****

**云计算**

**Linux实操100例**

达内 ⬝ 云计算学院

2019年2月

目录

[一、 云计算系统管理 6](#_Toc29948)

[二、 Linux系统目录介绍 6](#_Toc19615)

[案例01：IP地址 7](#_Toc24258)

[案例02：ls命令练习 7](#_Toc26145)

[案例03：查看文件内容练习 7](#_Toc15325)

[案例04：cd命令练习 8](#_Toc18317)

[案例05：主机名与查看网卡命令练习 9](#_Toc19587)

[案例06：创建命令练习 9](#_Toc6532)

[案例07：复制、删除、移动 9](#_Toc25351)

[案例08：ls命令练习 10](#_Toc26714)

[案例09：mount挂载 10](#_Toc6265)

[案例10：别名与vim的使用 11](#_Toc22216)

[案例11：为虚拟机 server 手工配置静态地址参数 11](#_Toc17162)

[案例12：真机上操作：永久别名与vim的使用 12](#_Toc3879)

[案例13：虚拟机Server上操作：复制、删除、移动及vim文本编辑器 12](#_Toc9430)

[案例14：虚拟机Server上操作：ls与cat、head 13](#_Toc12200)

[案例15：虚拟机Server上操作：搭建Yum仓库 13](#_Toc15813)

[案例16：指定yum软件源 14](#_Toc4385)

[案例17：tar制作/释放归档压缩包 14](#_Toc16798)

[案例18：添加并测试用户账号 15](#_Toc1383)

[案例19：添加账号时设置不同属性 15](#_Toc31615)

[案例20：passwd设置密码 16](#_Toc20812)

[案例21：usermod修改用户 16](#_Toc29275)

[案例22：组账号基本管理 16](#_Toc1976)

[案例23：配置NTP网络时间客户端 17](#_Toc32307)

[案例24：chmod权限设置 17](#_Toc22144)

[案例25：chown归属设置 18](#_Toc15379)

[案例26：权限设置 19](#_Toc6876)

[案例27：绑定到LDAP验证服务 19](#_Toc32554)

[案例28：访问NFS共享 20](#_Toc689)

[案例29：指定yum软件源 21](#_Toc3959)

[案例30：虚拟机 server0上操作，复制、粘贴、移动 21](#_Toc26890)

[案例31：虚拟机Server上操作 22](#_Toc19151)

[案例32：虚拟机 server0上操作，查找并处理文件 22](#_Toc20339)

[案例33：虚拟机 server0上操作,查找并提取文件内容 23](#_Toc9053)

[案例34：虚拟机 server0上操作,tar制作/释放归档压缩包 24](#_Toc8979)

[案例35：虚拟机 server0上操作 24](#_Toc3776)

[案例36：组账号基本管理 25](#_Toc13244)

[案例37：虚拟机 server0上操作计划任务 26](#_Toc21360)

[二、云计算应用管理 26](#_Toc13382)

[案例01：为虚拟机 server0 配置以下静态地址参数 26](#_Toc20001)

[案例02：复制、粘贴、移动 27](#_Toc5868)

[案例03：server上操作,（MBR分区模式）规划分区 28](#_Toc22310)

[案例:04：在server上操作,分区使用 29](#_Toc28049)

[案例05：虚拟机 server0操作，构建 LVM 存储 30](#_Toc28725)

[案例06：虚拟机 server0操作，构建 LVM 存储(修改PE大小) 31](#_Toc29964)

[案例08：vim文本编辑器练习，命令模式下的基本操作（重点知识） 32](#_Toc25843)

[案例09：书写hello world脚本 33](#_Toc27626)

[案例10： 测试IP地址脚本 33](#_Toc7471)

[案例11：if多分支结构 33](#_Toc6227)

[案例12：for循环 34](#_Toc5565)

[案例13：设置SELinux保护 34](#_Toc15853)

[案例14：自定义用户环境 34](#_Toc23877)

[案例15：Shell脚本 34](#_Toc3055)

[案例16：实现本机server0 的Web服务 35](#_Toc7311)

[案例17：实现本机server0 的防火墙配置 35](#_Toc3271)

[案例18：实现本机server0 的防火墙配置 35](#_Toc11783)

[案例19：实现本机server0 的防火墙配置 36](#_Toc11266)

[案例20：配置聚合连接 36](#_Toc9873)

[案例21：为虚拟机 server0 配置以下静态地址参数 37](#_Toc19538)

[案例22：实现本机server0 的Web服务 37](#_Toc495)

[案例23：实现本机server0 的防火墙配置 38](#_Toc30442)

[案例24：配置Samba文件夹共享 38](#_Toc30520)

[案例25：配置Samba文件夹共享 39](#_Toc13953)

[案例26：普通NFS共享的实现 40](#_Toc24121)

[案例27：在server上操作,（MBR分区模式）规划分区 41](#_Toc965)

[案例28：vim文本编辑器练习，命令模式下的基本操作（重点知识） 43](#_Toc5306)

[案例29：在 server0、desktop0 上操作 43](#_Toc5631)

[案例30：在server上操作，搭建mariadb数据库系统 43](#_Toc2804)

[案例31：在server上操作，配置一个数据库 44](#_Toc16948)

[案例32：在server上操作，使用数据库查询 44](#_Toc6341)

[案例33：虚拟机 server0操作， /dev/vdb 上按以下要求建立分区： 45](#_Toc9235)

[案例34：发布iSCSI网络磁盘 47](#_Toc2394)

[案例35：为虚拟机 server 配置以下虚拟Web主机 49](#_Toc26464)

[案例36：为虚拟机 server 配置Web访问控制 49](#_Toc31088)

[案例37：虚拟机 server 使用自定Web根目录 50](#_Toc32199)

[案例38：为虚拟机 server 部署动态WSGI站点 51](#_Toc17501)

[案例39：配置安全Web服务 52](#_Toc11323)

[案例40：虚拟机 server0上操作,（GPT分区模式）规划分区 53](#_Toc24178)

[案例41：虚拟机 server0上操作,交换分区练习 54](#_Toc29769)

[三、系统&服务管理进阶 54](#_Toc12162)

[案例01：补充命令使用 54](#_Toc16649)

[案例02：软连接与硬连接 55](#_Toc22161)

[案例03：man手册、zip备份 56](#_Toc15505)

[案例04：虚拟机svr7中，自定义yum软件仓库 57](#_Toc18418)

[案例05：vim效率操作 57](#_Toc6686)

[案例06：搭建单区域DNS服务器 58](#_Toc350)

[案例07：特殊DNS解析 59](#_Toc16964)

[案例08：配置DNS子域授权 60](#_Toc11215)

[案例09：搭建并测试缓存DNS 61](#_Toc31209)

[案例10：配置并验证Split分离解析 62](#_Toc288)

[案例11：查看进程信息 63](#_Toc8665)

[案例12：进程调度及终止 64](#_Toc15300)

[案例13：系统日志分析 65](#_Toc23601)

[案例14：使用systemctl工具 66](#_Toc18702)

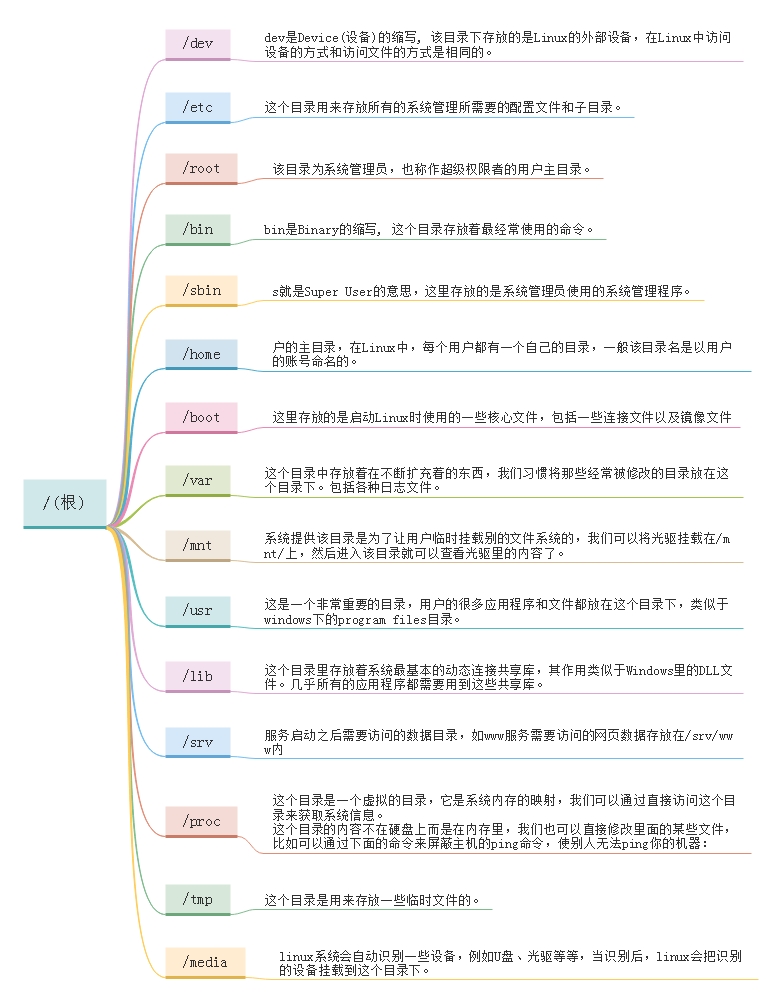
[案例15：PXE基础装机环境 67](#_Toc3578)

[案例16：rsync基本用法 70](#_Toc31456)

[案例17：rsync+SSH实时同步 71](#_Toc9887)

[案例18：配置Cobbler装机平台 72](#_Toc7105)

## Linux系统目录介绍



案例01：IP地址

1. IP地址的分类及范围

• IP地址：唯一标识计算机

– 一共32个二进制位

– 表示为4个十进制数,以 . 隔开

IP的分类:

• 用于一般计算机网络

– A类:1 ~ 127 网+主+主+主

– B类:128 ~ 191 网+网+主+主

– C类:192 ~ 223 网+网+网+主

• 组播及科研专用

– D类:224 ~ 239 组播

– E类:240 ~ 254 科研

案例02：ls命令练习

1. 查看根目录下内容

[root@server0 ~]# ls /

1. 显示根目录内容的详细属性

[root@server0 ~]# ls -l /

1. 显示/boot目录内容的详细属性

[root@server0 ~]# ls -l /boot/

1. 显示/boot的内容

[root@server0 ~]# ls /boot/

1. 显示/bin/bash程序，详细属性

[root@server0 ~]# ls -l /bin/bash

1. 显示/opt目录内容的详细属性

[root@server0 ~]# ls -l /opt/

案例03：查看文件内容练习

1. 查看/etc/passwd文件内容

[root@server0 ~]# cat /etc/passwd

1. 查看/etc/default/useradd文件内容

[root@server0 ~]# cat /etc/default/useradd

1. 查看内存信息

[root@server0 ~]# cat /proc/meminfo

1. 查看/etc/default/useradd文件内容，并显示行号

[root@server0 ~]# cat -n /etc/default/useradd

**案例04：cd命令练习**

1. 切换到根目录，利用pwd命令查看当前位置

[root@server0 ~]# cd /

[root@server0 /]# pwd

/

[root@server0 /]#

1. 切换到/root，利用pwd命令查看当前位置

[root@server0 /]# cd /root/

[root@server0 ~]# pwd

/root

[root@server0 ~]#

1. 切换到/boot，利用pwd命令查看当前位置

[root@server0 ~]# cd /boot/

[root@server0 boot]# pwd

/boot

[root@server0 boot]#

1. 切换到/opt，利用pwd命令查看当前位置

[root@server0 boot]# cd /opt/

[root@server0 opt]# pwd

/opt

[root@server0 opt]#

1. 切换到/tmp，利用pwd命令查看当前位置

[root@server0 opt]# cd /tmp/

[root@server0 tmp]# pwd

/tmp

[root@server0 tmp]#

1. 切换到/var，利用pwd命令查看当前位置

[root@server0 tmp]# cd /var/

[root@server0 var]# pwd

/var

[root@server0 var]#

1. 切换到/home，利用pwd命令查看当前位置

[root@server0 var]# cd /home/

[root@server0 home]# pwd

/home

[root@server0 home]#

1. 切换到/etc，利用pwd命令查看当前位置

[root@server0 home]# cd /etc/

[root@server0 etc]# pwd

/etc

[root@server0 etc]#

1. 切换到/proc，利用pwd命令查看当前位置

[root@server0 etc]# cd /proc/

[root@server0 proc]# pwd

/proc

[root@server0 proc]#

1. 切换到/etc/pki，利用pwd命令查看当前位置,再利用cd ..进行返回上一层目录

[root@server0 proc]# cd /etc/pki/

[root@server0 pki]# pwd

/etc/pki

[root@server0 pki]# cd ..

[root@server0 etc]# pwd

/etc

[root@server0 etc]#

**案例05：主机名与查看网卡命令练习**

1. 显示当前系统主机名

[root@server0 /]# hostname

server0.example.com

[root@server0 /]#

1. 显示当前系统内核版本

[root@server0 /]# uname -r

3.10.0-123.el7.x86\_64

[root@server0 /]#

1. 设置eth0网卡IP地址为192.168.10.1

[root@server0 /]# ifconfig eth0 192.168.10.1

[root@server0 /]# ifconfig

1. 查看当前系统eth0网卡IP地址信息

[root@server0 /]# ifconfig eth0

**案例06：创建命令练习**

1. 请在/root创建三个目录分别为student、file、example

[root@server0 /]# mkdir /root/student /root/file /root/example

[root@server0 /]# ls /root/

1. .请在/opt创建三个文本文件分别为1.txt、a.txt、nsd.txt

[root@server0 /]# touch /opt/1.txt /opt/a.txt /opt/nsd.txt

[root@server0 /]# ls /opt/

**案例07：复制、删除、移动**

1. 在目录/opt下创建一个子目录 nsd

[root@server0 /]# mkdir /opt/nsd

1. 在目录/opt/nsd/创建文件readme.txt,利用vim写入内容 I Love Linux

[root@server0 /]# vim /opt/nsd/readme.txt

I Love Linux

1. 将/etc/passwd 和 /etc/resolv.conf同时拷贝到/opt/nsd目录下

[root@server0 /]# cp /etc/passwd /etc/resolv.conf /opt/nsd

[root@server0 /]# ls /opt/nsd

1. 将文件 /etc/redhat-release复制到 /root/ 下，同时 改名为 version.txt

[root@server0 /]# cp /etc/redhat-release /root/version.txt

[root@server0 /]# ls /root/

1. 将文件 /root/version.txt 移动到/opt/nsd/目录下

[root@server0 /]# mv /root/version.txt /opt/nsd

[root@server0 /]# ls /opt/nsd

1. 将/home目录复制到/opt/nsd/目录下

[root@server0 /]# cp -r /home/ /opt/nsd

[root@server0 /]# ls /opt/nsd

**案例08：ls命令练习**

1. 查看根目录下内容

[root@server0 /]# ls /

1. 显示根目录本身详细属性

[root@server0 /]# ls -ld /

1. 切换到/boot目录，显示当前目录内容的详细属性，并加上易读的容量单位

[root@server0 /]# ls -lh /boot/

1. 显示/root的全部内容，包括隐藏文档

[root@server0 /]# ls -A /root/

1. 显示/bin/bash程序，详细属性

[root@server0 /]# ls -l /bin/bash

1. 显示/dev/cdrom光驱设备，详细属性

[root@server0 /]# ls -l /dev/cdrom

**案例09：mount挂载**

1. 在根目录下创建目录结构/rhel7/dvd

[root@server0 /]# mkdir -p /rhel7/dvd

[root@server0 /]# ls /

1. 利用图形将光盘文件rhel-server-7.4-x86\_64-dvd.iso放入光驱设备中
2. 将光驱设备挂载到/rhel7/dvd目录，以/rhel7/dvd目录作为其访问点

[root@server0 /]# mount /dev/cdrom /rhel7/dvd

[root@server0 /]# ls /rhel7/dvd

1. 查看/rhel7/dvd/Packages目录内容

[root@server0 /]# ls /rhel7/dvd/Packages

1. 将/rhel7/dvd/Packages目录中以vsftpd开头的软件包，拷贝到/opt下

[root@server0 /]# ls /rhel7/dvd/Packages/vsftpd\*

[root@server0 /]# cp /rhel7/dvd/Packages/vsftpd\* /opt

[root@server0 /]# ls /opt

**案例10：别名与vim的使用**

1. 为虚拟机定义一个别名，执行byebye可以实现关闭系统

[root@server0 /]# alias byebye='poweroff'

[root@server0 /]# alias

[root@server0 /]# byebye

1. 利用vim 修改文件/etc/hostname将其原有内容全部删除，写入新的内容为student.tmooc.cn

[root@server0 /]# vim /etc/hostname

student.tmooc.cn

案例练习，环境准备:

**rht-vmctl reset classroom**

**rht-vmctl reset server**

**rht-vmctl reset desktop**

**##########################################################################**

**案例11：为虚拟机 server 手工配置静态地址参数**

1. 主机名:server0.example.com

[root@server0 ~]# echo server0.example.com > /etc/hostname

[root@server0 ~]# cat /etc/hostname

server0.example.com

[root@server0 ~]# exit

登出

Connection to 172.25.0.11 closed.

[root@room9pc01 ~]# ssh -X root@172.25.0.11

[root@server0 ~]# hostname

server0.example.com

[root@server0 ~]#

1. IP地址:172.25.0.150
2. 子网掩码:255.255.255.0
3. 默认网关:172.25.0.254

[root@server0 ~]# nmcli connection modify 'System eth0' ipv4.method manual ipv4.addresses '172.25.0.150/24 172.25.0.254' connection.autoconnect yes

[root@server0 ~]# nmcli connection up 'System eth0'

[root@room9pc01 ~]# ssh -X root@172.25.0.150

[root@server0 ~]# ifconfig | head -2

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 172.25.0.150 netmask 255.255.255.0 broadcast 172.25.0.255

[root@server0 ~]# route -n

1. 指定DNS服务器地址:172.25.254.254

[root@server0 ~]# echo nameserver 172.25.254.254 > /etc/resolv.conf

[root@server0 ~]# cat /etc/resolv.conf

nameserver 172.25.254.254

[root@server0 ~]#

**案例12：真机上操作：永久别名与vim的使用**

1. 定义一个永久别名（修改/root/.bashrc文件)，执行c可以实现还原虚拟机classroom
2. 定义一个永久别名（修改/root/.bashrc文件)，执行s可以实现还原虚拟机server
3. 定义一个永久别名（修改/root/.bashrc文件)，执行d可以实现还原虚拟机Desktop
4. 定义一个永久别名（修改/root/.bashrc文件)，执行gos可以远程管理server
5. 定义一个永久别名（修改/root/.bashrc文件)，执行god可以远程管理desktop

[root@room9pc01 ~]# vim /root/.bashrc #新开启一个终端验证

# .bashrc

alias gos='ssh -X root@172.25.0.11'

alias god='ssh -X root@172.25.0.10'

alias c='rht-vmctl reset classroom'

alias d='rht-vmctl reset desktop'

alias s='rht-vmctl reset server'

**案例13：虚拟机Server上操作：复制、删除、移动及vim文本编辑器**

1. 在目录/mnt下创建一个子目录public

[root@server0 ~]# mkdir /mnt/public

[root@server0 ~]# ls /mnt/

1. 在目录/mnt/public 创建文件linux.txt,利用vim写入内容 Study Linux

[root@server0 ~]# vim /mnt/public/linux.txt

Study Linux

1. 将/mnt/public/linux.txt文件复制到/root目录下，同时 改名为 study.txt

[root@server0 ~]# cp /mnt/public/linux.txt /root/study.txt

[root@server0 ~]# ls /root/

1. 利用vim 修改文件/etc/hostname将其原有内容全部删除，写入新的内容为www.qq.com

[root@server0 ~]# vim /etc/hostname

www.qq.com

1. 将/etc/passwd 、/etc/resolv.conf、/etc/hostname 同时拷贝到/mnt/public/目录下

[root@server0 ~]# cp /etc/passwd /etc/resolv.conf /etc/hostname /mnt/public/

[root@server0 ~]# ls /mnt/public/

1. 将文件 /mnt/public/hostname 重改名为 stu.txt

[root@server0 ~]# mv /mnt/public/hostname /mnt/public/stu.txt

[root@server0 ~]# ls /mnt/public/

1. 创建目录结构/mnt/public/test/vm

[root@server0 ~]# mkdir -p /mnt/public/test/vm

1. 将目录 /boot内容中以 vm 开头的 复制到/mnt/public/test/vm目录下

[root@server0 ~]# cp /boot/vm\* /mnt/public/test/vm

[root@server0 ~]# ls /mnt/public/test/vm

1. 将/home目录复制到/mnt/public/test/目录下

[root@server0 ~]# cp -r /home/ /mnt/public/test/

[root@server0 ~]# ls /mnt/public/test/

**案例14：虚拟机Server上操作：ls与cat、head**

1. 显示根目录下所有内容

[root@server0 ~]# ls /

1. 显示/etc目录下所有已tab结尾的文件

[root@server0 ~]# ls /etc/\*tab

1. 显示/etc/resolv.conf文件的详细属性并加上易读的单位

[root@server0 ~]# ls -lh /etc/resolv.conf

1. 显示/etc/sysconfig/network-scripts/ifcfg-eth0文件的详细属性并加上易读的单位

[root@server0 ~]# ls -lh /etc/sysconfig/network-scripts/ifcfg-eth0

1. 显示/etc/sysconfig/network-scripts/ifcfg-eth0文件的头4行内容

[root@server0 ~]# head -4 /etc/sysconfig/network-scripts/ifcfg-eth0

**案例15：虚拟机Server上操作：搭建Yum仓库**

1. 为 server0 指定可用的 yum 软件源

– YUM软件库的地址为http://classroom.example.com/content/rhel7.0/x86\_64/dvd

– 将此配置为虚拟机 server0 的默认软件仓库

– 确认可用的仓库列表

– 利用yum仓库安装sssd软件

– 利用yum仓库安装xeyes软件

[root@server0 ~]# rm -rf /etc/yum.repos.d/\*

[root@server0 ~]# vim /etc/yum.repos.d/rhel7.repo

[rhel7]

name=rhel7.0

baseurl=http://classroom.example.com/content/rhel7.0/x86\_64/dvd

enabled=1

gpgcheck=0

[root@server0 ~]# yum repolist

[root@server0 ~]# yum -y install sssd xeyes

**案例16：指定yum软件源**

1. 为 server0 指定可用的 yum 软件源

– YUM软件库的地址为 http://classroom.example.com/content/rhel7.0/x86\_64/dvd

– 将此配置为虚拟机 server0 的默认软件仓库

– 确认可用的仓库列表

– 利用yum仓库安装system-config-kickstart

[root@server0 ~]# rm -rf /etc/yum.repos.d/\*

[root@server0 ~]# vim /etc/yum.repos.d/rhel7.repo

[rhel7]

name=rhel7.0

baseurl=http://classroom.example.com/content/rhel7.0/x86\_64/dvd

enabled=1

gpgcheck=0

[root@server0 ~]# yum repolist

[root@server0 ~]# yum -y install system-config-kickstart

**案例17：tar制作/释放归档压缩包**

1. 首先创建/root/boothome/与/root/usrsbin/目录

[root@server0 ~]# mkdir /root/boothome mkdir /root/usrsbin

[root@server0 ~]# ls /root

1. 备份/boot、/home这两个文件夹，保存为boothome.tar.gz文件

[root@server0 ~]# tar -zcf /opt/boothome.tar.gz /boot/ /home

tar: 从成员名中删除开头的“/”

[root@server0 ~]# ls /opt/

1. 查看boothome.tar.gz文件内包含哪些内容

[root@server0 ~]# tar -tf /opt/boothome.tar.gz

1. 将boothome.tar.gz释放到文件夹/root/boothome/下

[root@server0 ~]# tar -xf /opt/boothome.tar.gz -C /root/boothome/

[root@server0 ~]# ls /root/boothome/

1. 备份/usr/sbin目录，保存为usrsbin.tar.bz2文件

[root@server0 ~]# tar -jcf /opt/usrsbin.tar.bz2 /usr/sbin/

tar: 从成员名中删除开头的“/”

tar: 从硬连接目标中删除开头的“/”

[root@server0 ~]# ls /usr/sbin/

1. 查看usrsbin.tar.bz2文件内包含哪些内容

[root@server0 ~]# tar -tf /opt/usrsbin.tar.bz2

1. 将usrsbin.tar.bz2释放到/root/usrsbin/文件夹下

[root@server0 ~]# tar -xf /opt/usrsbin.tar.bz2 -C /root/usrsbin/

[root@server0 ~]# ls /root/usrsbin/

**案例18：添加并测试用户账号**

1. 创建一个名为stu01的用户账号

[root@server0 ~]# useradd stu01

[root@server0 ~]# id stu01

uid=1001(stu01) gid=1001(stu01) 组=1001(stu01)

[root@server0 ~]# grep stu01 /etc/passwd

1. 检查/etc/passwd、/etc/shadow文件的最后一行

[root@server0 ~]# head -1 /etc/passwd

[root@server0 ~]# head -1 /etc/shadow

1. 检查/home/新增加的宿主目录（家目录）

[root@server0 ~]# ls /home/

1. 为用户stu01设置一个密码（123456）

[root@server0 ~]# echo 123456 | passwd --stdin stu01

**案例19：添加账号时设置不同属性**

1. 新建用户nsd01，宿主目录位于/opt/nsd01

[root@server0 ~]# useradd -d /opt/nsd01 nsd01

[root@server0 ~]# ls /opt/

[root@server0 ~]# grep nsd01 /etc/passwd

1. 新建系统账号sys01，将UID设为1234，登录Shell设为/sbin/nologin

[root@server0 ~]# useradd -u 1234 -s /sbin/nologin sys01

[root@server0 ~]# id sys01

uid=1234(sys01) gid=1234(sys01) 组=1234(sys01)

[root@server0 ~]# grep sys01 /etc/passwd

1. 为用户sys01设置密码，并测试是否能够登录

[root@server0 ~]# echo 123456 | passwd --stdin sys01

1. 新建用户admin，附加组设为adm、root

[root@server0 ~]# useradd -G adm,root admin

[root@server0 ~]# id admin

uid=1235(admin) gid=1235(admin) 组=1235(admin),0(root),4(adm)

**案例20：passwd设置密码**

1. 给用户nsd01设置密码123456

[root@server0 ~]# echo 123456 | passwd --stdin nsd01

1. 采用--stdin方式将用户nsd01的密码设为654321

[root@server0 ~]# echo 654321 | passwd --stdin nsd01

**案例21：usermod修改用户**

1. 新建一个用户nsd03，将宿主目录设为/opt/home03，并设置密码

[root@server0 ~]# useradd -d /opt/home03 nsd03

[root@server0 ~]# ls /opt/

[root@server0 ~]# grep nsd03 /etc/passwd

1. 设置nsd03密码为redhat

[root@server0 ~]# echo redhat | passwd --stdin nsd03

1. 将用户nsd03的宿主目录改为/home/nsd03

[root@server0 ~]# usermod -d /home/nsd03 nsd03

[root@server0 ~]# grep nsd03 /etc/passwd

1. 将用户sys01的登录Shell改为/bin/bash

[root@server0 ~]# usermod -s /bin/bash sys01

[root@server0 ~]# grep sys01 /etc/passwd

**案例22：组账号基本管理**

1. 新建组账号stugrp

[root@server0 ~]# groupadd stugrp

[root@server0 ~]# grep stugrp /etc/group

1. 为stugrp组添加三个成员用户（user01、root、zhangsan）

[root@server0 ~]# useradd user01

[root@server0 ~]# useradd zhangsan

[root@server0 ~]# gpasswd -a user01 stugrp

正在将用户“user01”加入到“stugrp”组中

[root@server0 ~]# gpasswd -a zhangsan stugrp

正在将用户“zhangsan”加入到“stugrp”组中

[root@server0 ~]# gpasswd -a root stugrp

正在将用户“root”加入到“stugrp”组中

[root@server0 ~]# grep stugrp /etc/group

1. 从stugrp组删除一个成员（user01）

[root@server0 ~]# gpasswd -d user01 stugrp

正在将用户“user01”从“stugrp”组中删除

[root@server0 ~]# grep stugrp /etc/group

**案例23：配置NTP网络时间客户端**

1. 配置虚拟机 server0，自动校对系统时间

NTP服务器位于 classroom.example.com

此客户机的时间与NTP服务器的时间保持同步

1.安装chrony软件包（客户端软件：专用于访问服务的软件）

[root@server0 ~]# yum -y install chrony

[root@server0 ~]# rpm -q chrony

2.修改程序主配置文件，指定服务端位置

在Linux大多数配置文件中，以#开头的行为注释行

[root@server0 ~]# vim /etc/chrony.conf

#server 0.rhel.pool.ntp.org iburst

#server 1.rhel.pool.ntp.org iburst

#server 2.rhel.pool.ntp.org iburst

server classroom.example.com iburst

3.重起程序

[root@server0 ~]# systemctl restart chronyd #重起chronyd

[root@server0 ~]# systemctl enable chronyd #设置为开机自起

daemon:超级守护进程

4.验证：

[root@server0 ~]# date -s "2008-10-1 12:00"

2008年 10月 01日 星期三 12:00:00 CST

[root@server0 ~]# date

[root@server0 ~]# systemctl restart chronyd

[root@server0 ~]# date

[root@server0 ~]# date

**案例24：chmod权限设置**

1. 以root用户新建/nsddir/目录，在该目录下新建文件readme.txt

[root@server0 ~]# mkdir /nsddir

[root@server0 ~]# echo abc > /nsddir/readme.txt

[root@server0 ~]# cat /nsddir/readme.txt

1. 使用户zhangsan能够在/nsddir/目录下创建/删除子目录

[root@server0 ~]# chmod o+w /nsddir

[root@server0 ~]# ls -ld /nsddir

drwxr-xrwx. 2 root root 23 10月 11 21:41 /nsddir

[root@server0 ~]# su - zhangsan

[zhangsan@server0 ~]$ mkdir /nsddir/test

[zhangsan@server0 ~]$ ls /nsddir/

readme.txt test

[zhangsan@server0 ~]$ exit

logout

[root@server0 ~]#

1. 使用户zhangsan能够修改/nsddir/readme.txt文件的容

[root@server0 ~]# chmod o+w /nsddir/readme.txt

[root@server0 ~]# su - zhangsan

[zhangsan@server0 ~]$ echo test >> /nsddir/readme.txt

[zhangsan@server0 ~]$ cat /nsddir/readme.txt

abc

test

[zhangsan@server0 ~]$ exit

logout

[root@server0 ~]#

**案例25：chown归属设置**

1. 新建/tarena1目录

-将属主设为gelin01，属组设为tarena组

[root@server0 ~]# mkdir /tarena1

[root@server0 ~]# useradd gelin01

[root@server0 ~]# groupadd tarena

[root@server0 ~]# chown gelin01:tarena /tarena1/

[root@server0 ~]# ls -ld /tarena1/

-使用户gelin01对此目录具有rwx权限，除所有者与所属组成员，其他人对此目录无任何权限

[root@server0 ~]# chmod o=--- /tarena1/

[root@server0 ~]# ls -ld /tarena1/

drwxr-x---. 2 gelin01 tarena 6 10月 11 21:42 /tarena1/

-使用户gelin02能进入、查看/tarena1文件夹（提示：将gelin02加入所属组）

[root@server0 ~]# useradd gelin02

[root@server0 ~]# gpasswd -a gelin02 tarena

正在将用户“gelin02”加入到“tarena”组中

[root@server0 ~]# su - gelin02

[gelin02@server0 ~]$ cd /tarena1/

[gelin02@server0 tarena1]$ ls

[gelin02@server0 tarena1]$ exit

logout

[root@server0 ~]#

1. 新建/tarena2目录

-将属组设为tarena

[root@server0 ~]# mkdir /tarena2

[root@server0 ~]# chown :tarena /tarena2

[root@server0 ~]# ls -ld /tarena2

-使tarena组的任何用户都能在此目录下创建、删除文件

[root@server0 ~]# chmod g+w /tarena2/

[root@server0 ~]# ls -ld /tarena2/

drwxrwxr-x. 2 root tarena 6 10月 11 21:45 /tarena2/

[root@server0 ~]# id gelin02

uid=1240(gelin02) gid=1242(gelin02) 组=1242(gelin02),1241(tarena)

[root@server0 ~]# su - gelin02

上一次登录：四 10月 11 21:44:39 CST 2018pts/1 上

[gelin02@server0 ~]$ mkdir /tarena2/test

[gelin02@server0 ~]$ ls /tarena2

test

[gelin02@server0 ~]$ exit

logout

[root@server0 ~]#

1. 新建/tarena/public目录

-使任何用户对此目录都有rwx权限

[root@server0 ~]# mkdir -p /tarena/public

[root@server0 ~]# chmod ugo=rwx /tarena/public

[root@server0 ~]# ls -ld /tarena/public

-拒绝zhangsan进入此目录，对此目录无任何权限

[root@server0 ~]# setfacl -m u:zhangsan:--- /tarena/public/

[root@server0 ~]# getfacl /tarena/public/

**案例26：权限设置**

1. 创建文件夹/data/test,设置目录的访问权限，使所有者和所属组具备读写执行的权限；其他人无任何权限。

[root@server0 ~]# mkdir -p /data/test

[root@server0 ~]# chmod ug=rwx,o=--- /data/test

[root@server0 ~]# ls -ld /data/test

drwxrwx---. 2 root root 6 10月 11 21:48 /data/test

1. 递归修改文件夹/data/test的归属使所有者为zhangsan，所属组为tarena。

[root@server0 ~]# chown -R zhangsan:tarena /data/test/

[root@server0 ~]# ls -ld /data/test/

drwxrwx---. 2 zhangsan tarena 6 10月 11 21:48 /data/test/

1. 请实现在test目录下，新建的所有子文件或目录的所属组都会是tarena。

[root@server0 ~]# chmod g+s /data/test/

[root@server0 ~]# ls -ld /data/test/

drwxrws---. 2 zhangsan tarena 6 10月 11 21:48 /data/test/

[root@server0 ~]# mkdir /data/test/nsd

[root@server0 ~]# ls -ld /data/test/nsd

drwxr-sr-x. 2 root tarena 6 10月 11 21:50 /data/test/nsd

[root@server0 ~]#

1. 为lisi创建ACL访问权限，使得lisi可以查看/etc/shadow文件

[root@server0 ~]# useradd lisi

[root@server0 ~]# setfacl -m u:lisi:r /etc/shadow

[root@server0 ~]# su - lisi

[lisi@server0 ~]$ cat /etc/shadow

[lisi@server0 ~]$ exit

logout

[root@server0 ~]#

**案例27：绑定到LDAP验证服务**

1. 使用系统 classroom.example.com 提供的LDAP服务

– 验证服务的基本DN是：dc=example,dc=com

– 账户信息和验证信息都是由 LDAP 提供的

– 连接要使用证书加密，证书可以在下面的链接下载：

http://classroom.example.com/pub/example-ca.crt

– 确认本地可以识别ldapuser0用户

1.安装一个客户端sssd软件,与LDAP 服务端沟通

[root@server0 /]# yum -y install sssd

2.安装authconfig-gtk 图形的工具 配置sssd软件

[root@server0 /]# yum -y install authconfig-gtk

[root@server0 /]# rpm -q authconfig-gtk

3.运行authconfig-gtk 图形的工具

[root@server0 /]# authconfig-gtk

//选择LDAP

dc=example,dc=com #指定服务端域名

classroom.example.com #指定服务端主机名

//勾选TLS加密

使用证书加密: http://classroom.example.com/pub/example-ca.crt

选择LDAP密码

4.重起客户端服务sssd服务，设置开机自启动

[root@server0 /]# systemctl restart sssd #重起sssd服务

[root@server0 /]# systemctl enable sssd #设置开机自启动

5.验证

[root@server0 ~]# grep 'ldapuser10' /etc/passwd

[root@server0 ~]# id ldapuser10

**案例28：访问NFS共享**

1. 查看classroom.example.com的NFS共享
2. 将classroom.example.com的NFS共享目录，挂载到本地/home/guests

1.查看共享classroom.example.com

[root@server0 ~]# showmount -e classroom.example.com

Export list for classroom.example.com:

/home/guests 172.25.0.0/255.255.0.0

2.访问共享内容,将服务端的共享文件夹数据，挂载到本地/home/guests以本地的/home/guests作为访问点

[root@server0 ~]# mkdir /home/guests

[root@server0 ~]# mount classroom.example.com:/home/guests/ /home/guests/

[root@server0 ~]# ls /home/guests/

[root@server0 ~]# su - ldapuser0

**案例29：指定yum软件源**

1. 为 server0 指定可用的 yum 软件源

– YUM软件库的地址为 http://classroom.example.com/content/rhel7.0/x86\_64/dvd

– 将此配置为虚拟机 server0 的默认软件仓库

– 确认可用的仓库列表

– 利用yum仓库安装httpd与vsftpd

[root@server0 ~]# rm -rf /etc/yum.repos.d/\*

[root@server0 ~]# vim /etc/yum.repos.d/rhel7.repo

[rhel7]

name=rhel7.0

baseurl=http://classroom.example.com/content/rhel7.0/x86\_64/dvd

enabled=1

gpgcheck=0

[root@server0 ~]# yum repolist

[root@server0 ~]# yum -y install httpd vsftpd

**案例30：虚拟机 server0上操作，复制、粘贴、移动**

1. 以root用户新建/nsddir/目录，在此目录下新建readme.txt文件，并进一步完成下列操作

[root@server0 ~]# mkdir /nsddir

[root@server0 ~]# touch /nsddir/readme.txt

[root@server0 ~]# ls /nsddir

1. 将“I am a girl”写入到文件readme.txt （男同学写男生）

[root@server0 ~]# echo I am a boy > /nsddir/readme.txt

[root@server0 ~]# cat /nsddir/readme.txt

I am a boy

[root@server0 ~]#

1. 将readme.txt重命名为mylove.txt

[root@server0 ~]# mv /nsddir/readme.txt /nsddir/mylove.txt

[root@server0 ~]# ls /nsddir/

1. 将/etc/passwd、/boot、/etc/group同时拷贝到/nsddir目录下

[root@server0 ~]# cp -r /etc/passwd /boot/ /etc/group /nsddir/

[root@server0 ~]# ls /nsddir/

1. 将ifconfig命令的前两行内容，追加写入mylove.txt

[root@server0 ~]# ifconfig | head -2

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 172.25.0.169 netmask 255.255.255.0 broadcast 172.25.0.255

[root@server0 ~]# ifconfig | head -2 >> /nsddir/mylove.txt

[root@server0 ~]# cat /nsddir/mylove.txt

1. 将主机名永久配置文件，拷贝到/nsddir目录下

[root@server0 ~]# cp /etc/hostname /nsddir/

[root@server0 ~]# ls /nsddir/

1. 将DNS永久配置文件，拷贝到/nsddir目录下

[root@server0 ~]# cp /etc/resolv.conf /nsddir/

[root@server0 ~]# ls /nsddir

**案例31：虚拟机Server上操作**

1. 新建目录结构/nsd/test

[root@server0 ~]# mkdir -p /nsd/test

[root@server0 ~]# ls /

1. 在目录/nsd/test创建文件nsd.txt并写入内容 NSD Student

[root@server0 ~]# echo NSD Student > /nsd/test/nsd.txt

[root@server0 ~]# cat /nsd/test/nsd.txt

1. 将/nsd/test/nsd.txt文件复制到/root目录下，同时 改名为 tedu.txt

[root@server0 ~]# cp /nsd/test/nsd.txt /root/tedu.txt

[root@server0 ~]# ls /root/

1. 利用vim 修改文件/etc/hostname将其原有内容全部删除，写入新的内容为student0.example.com

[root@server0 ~]# vim /etc/hostname

student0.example.com

1. 将/etc/passwd 、/etc/resolv.conf、/etc/hostname 同时拷贝到/nsd/test/目录下

[root@server0 ~]# cp /etc/passwd /etc/resolv.conf /etc/hostname /nsd/test

1. 将文件 /nsd/test/hostname 重改名为 hn.txt

[root@server0 ~]# mv /nsd/test/hostname /nsd/test/hn.txt

1. 创建目录结构/nsd/test/kernel

[root@server0 ~]# mkdir –p /nsd/test/kernel

1. 将目录 /boot内容中以 vm 开头的 复制到/nsd/test/kernel目录下

[root@server0 ~]# cp /boot/vm\* /nsd/test/kernel

[root@server0 ~]# ls /nsd/test/kerbel

**案例32：虚拟机 server0上操作，查找并处理文件**

1. 利用find查找所有用户 student 拥有的必须是文件,把它们拷贝到 /root/findfiles/(首先创建目录) 文件夹中

[root@server0 ~]# mkdir /root/findfiles

[root@server0 ~]# find / -user student –type f

[root@server0 ~]# find / -user student –type f –exec cp {} /root/findfiles \;

[root@server0 ~]# ls /root/findfiles

1. 利用find查找/boot目录下大于10M并且必须是文件，拷贝到/opt

[root@server0 ~]# find /boot -size +10M

[root@server0 ~]# find /boot –size +10M -exec cp {} /opt \;

[root@server0 ~]# ls /opt

1. 利用find查找/boot/ 目录下以 vm 开头且必须是文件，拷贝到/opt

[root@server0 ~]# find /boot –name "vm\*" -type f

[root@server0 ~]# find /boot –name "vm\*" -type f -exec cp {} /opt \;

[root@server0 ~]# ls /opt

1. 利用find查找/boot/ 目录下为快捷方式

[root@server0 ~]# find /boot –type l

1. 利用find查找/etc 目录下，以 tab 作为结尾的 必须是文件

[root@server0 ~]# find /etc –name "\*tab" -type f

**案例33：虚拟机 server0上操作,查找并提取文件内容**

1. 在文件 /usr/share/dict/words 中查找到所有包含字符串 seismic 的行,将输出信息,写入到/opt/nsd18.txt

[root@server0 ~]# grep seismic /usr/share/dict/words

1. 查看内核版本，将显示结果重定向到/root/version.txt

[root@server0 ~]# uname -r

3.10.0-123.el7.x86\_64

[root@server0 ~]# uname -r > /root/version.txt

[root@server0 ~]# cat /root/version.txt

3.10.0-123.el7.x86\_64

[root@server0 ~]#

1. 查看红帽系统版本，将显示结果追加到/root/version.txt

[root@server0 ~]# cat /etc/redhat-release

Red Hat Enterprise Linux Server release 7.0 (Maipo)

[root@server0 ~]# cat /etc/redhat-release >> /root/version.txt

[root@server0 ~]# cat /root/version.txt

1. 查看主机名将显示结果追加到/root/version.txt

[root@server0 ~]# hostname

server0.example.com

[root@server0 ~]# hostname >> /root/version.txt

[root@server0 ~]# cat /root/version.txt

3.10.0-123.el7.x86\_64

Red Hat Enterprise Linux Server release 7.0 (Maipo)

server0.example.com

1. 将/etc/fstab文件中以UUID开头的信息，写入到/root/fstab.txt

[root@server0 ~]# grep ^UUID /etc/fstab

UUID=9bf6b9f7-92ad-441b-848e-0257cbb883d1 / xfs defaults 1 1

[root@server0 ~]# grep ^UUID /etc/fstab > /root/fstab.txt

[root@server0 ~]# cat /root/fstab.txt

UUID=9bf6b9f7-92ad-441b-848e-0257cbb883d1 / xfs defaults 1 1

[root@server0 ~]#

1. 提取/etc/passwd以bash结尾的行，将其信息写入/opt/pass.txt

[root@server0 ~]# grep bash$ /etc/passwd

[root@server0 ~]# grep bash$ /etc/passwd > /opt/pass.txt

[root@server0 ~]# cat /opt/pass.txt

1. 复制/etc/login.defs文件到当前目录下，改名为init.txt

[root@server0 ~]# cp /etc/login.defs init.txt

[root@server0 ~]# ls

1. 提取init.txt文件里的有效配置（去除以#号开头，去除空行），保存为init2.txt

[root@server0 ~]# grep -v ^# init.txt

[root@server0 ~]# grep -v ^# init.txt | grep -v ^$

[root@server0 ~]# grep -v ^# init.txt | grep -v ^$ > init2.txt

**案例34：虚拟机 server0上操作,tar制作/释放归档压缩包**

1. 备份/boot、/usr/local这两个文件夹，保存为bootusr.tar.gz文件

[root@server0 ~]# tar -zcf /opt/bootusr.tar.gz /boot/ /usr/local/

tar: 从成员名中删除开头的“/”

[root@server0 ~]# ls /opt/

查看bootusr.tar.gz文件内包含哪些内容

[root@server0 ~]# tar –tf /opt/bootusr.tar.gz

1. 创建/root/bootusr/文件夹

[root@server0 ~]# mkdir /root/bootusr

1. 将bootusr.tar.gz释放到文件夹/root/bootusr/下

[root@server0 ~]# mkdir /root/bootusr

[root@server0 ~]# tar -xf /opt/bootusr.tar.gz -C /root/bootusr/

[root@server0 ~]# ls /root/bootusr/

1. 创建一个名为 /root/backup.tar.xz 的归档文件，其中包含 /tmp 目录中的内容

[root@server0 ~]# tar -Jcf /root/backup.tar.xz /tmp

tar: 从成员名中删除开头的“/”

tar: /tmp/.X11-unix/X0: 忽略套接字(socket)

tar: /tmp/.ICE-unix/1056: 忽略套接字(socket)

[root@server0 ~]# ls /root/

1. 查看 /root/backup.tar.xz 归档压缩包的内容

[root@server0 ~]# tar -tf /root/backup.tar.xz

**案例35：虚拟机 server0上操作**

1. 新建用户 alex，其用户ID为3456，密码是flectrag

[root@server0 ~]# useradd -u 3456 alex

[root@server0 ~]# echo flectrag | passwd --stdin alex

1. 创建下列用户、组以及组的成员关系：

– 一个名为 adminuser 的组

[root@server0 ~]# groupadd adminuser

[root@server0 ~]# grep adminuser /etc/group

– 一个名为 natasha 的用户，其属于 adminuser 组， 这个组是该用户的从属组

[root@server0 ~]# useradd -G adminuser natasha

[root@server0 ~]# id natasha

– 一个名为 harry 的用户，其属于 adminuser 组，这个 组是该用户的从属组

[root@server0 ~]# useradd -G adminuser harry

[root@server0 ~]# id harry

– 一个名为 sarah 的用户，其在系统中没有可交互的 Shell（/sbin/nologin），并且不是 adminuser 组的成员

[root@server0 ~]# useradd -s /sbin/nologin sarah

[root@server0 ~]# id sarah

uid=3459(sarah) gid=3460(sarah) 组=3460(sarah)

[root@server0 ~]# grep sarah /etc/passwd

– natasha 、harry、sarah 的密码都要设置为 flectra

[root@server0 ~]# echo flectra | passwd --stdin natasha

更改用户 natasha 的密码 。

passwd：所有的身份验证令牌已经成功更新。

[root@server0 ~]# echo flectra | passwd --stdin harry

更改用户 harry 的密码 。

passwd：所有的身份验证令牌已经成功更新。

[root@server0 ~]# echo flectra | passwd --stdin sarah

更改用户 sarah 的密码 。

passwd：所有的身份验证令牌已经成功更新。

**案例36：组账号基本管理**

1. 新建组账号stugrp

[root@server0 ~]# groupadd stugrp

1. 创建用户lily、zhangsan

[root@server0 ~]# useradd lily

[root@server0 ~]# useradd zhangsan

1. 为stugrp组添加三个成员用户（lily、root、zhangsan）

[root@server0 ~]# gpasswd -a lily stugrp

正在将用户“lily”加入到“stugrp”组中

[root@server0 ~]# gpasswd -a root stugrp

正在将用户“root”加入到“stugrp”组中

[root@server0 ~]# gpasswd -a zhangsan stugrp

正在将用户“zhangsan”加入到“stugrp”组中

[root@server0 ~]# grep stugrp /etc/group

从stugrp组删除一个成员（lily）

[root@server0 ~]# gpasswd -d lily stugrp

正在将用户“lily”从“stugrp”组中删除

[root@server0 ~]# grep stugrp /etc/group

**案例37：虚拟机 server0上操作计划任务**

为用户 natasha 配置一个定时任务

– 每天在本地时间 14:23 执行

– 需要完成的任务操作为 /bin/echo hiya

[root@server0 ~]# crontab -e -u natasha

23 14 \* \* \* /bin/echo hiya

[root@server0 ~]# crontab -l -u natasha

## 二、云计算应用管理

案例练习，环境准备:

**rht-vmctl reset classroom**

**rht-vmctl reset server**

**rht-vmctl reset desktop**

**##########################################################################**

**案例01：为虚拟机 server0 配置以下静态地址参数**

1. – 主机名:server0.example.com

[root@server0 ~]# echo server0.example.com > /etc/hostname

[root@server0 ~]# cat /etc/hostname

server0.example.com

[root@server0 ~]# exit

登出

Connection to 172.25.0.11 closed.

[root@room9pc01 ~]# ssh -X root@172.25.0.11

[root@server0 ~]# hostname

server0.example.com

[root@server0 ~]#

1. IP地址:172.25.0.170
2. 子网掩码:255.255.255.0
3. 默认网关:172.25.0.254

[root@server0 ~]# nmcli connection modify 'System eth0' ipv4.method manual ipv4.addresses '172.25.0.170/24 172.25.0.254' connection.autoconnect yes

[root@server0 ~]# nmcli connection up 'System eth0'

[root@room9pc01 ~]# ssh -X root@172.25.0.170

[root@server0 ~]# ifconfig | head -2

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 172.25.0.170 netmask 255.255.255.0 broadcast 172.25.0.255

[root@server0 ~]# route -n

1. DNS服务器:172.25.254.254

[root@server0 ~]# echo nameserver 172.25.254.254 > /etc/resolv.conf

[root@server0 ~]# cat /etc/resolv.conf

nameserver 172.25.254.254

[root@server0 ~]#

**案例02：复制、粘贴、移动**

1. 以root用户新建/example/目录，在此目录下新建nsd.txt文件，并进一步完成下列操作

[root@server0 ~]# mkdir /example

[root@server0 ~]# touch /example/readme.txt

[root@server0 ~]# ls /example

1. 将“We are all superhuman beings”写入到文件nsd.txt

[root@server0 ~]# echo We are all superhuman beings > /example/readme.txt

[root@server0 ~]# cat /example/readme.txt

We are all superhuman beings

[root@server0 ~]#

1. 将nsd.txt重命名为mylove.txt

[root@server0 ~]# mv /example/readme.txt /example/mylove.txt

[root@server0 ~]# ls /example

1. 将/etc/passwd、/boot、/etc/group同时拷贝到/example/目录下

[root@server0 ~]# cp -r /etc/passwd /boot/ /etc/group /example/

[root@server0 ~]# ls /example

1. 将ifconfig命令的前两行内容，追加写入mylove.txt

[root@server0 ~]# ifconfig | head -2

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 172.25.0.170 netmask 255.255.255.0 broadcast 172.25.0.255

[root@server0 ~]# ifconfig | head -2 >> /nsddir/mylove.txt

[root@server0 ~]# cat /example/mylove.txt

1. 将主机名永久配置文件，拷贝到/example/目录下

[root@server0 ~]# cp /etc/hostname /example

[root@server0 ~]# ls /example

1. 将DNS永久配置文件，拷贝到/example/目录下

[root@server0 ~]# cp /etc/resolv.conf /example

[root@server0 ~]# ls /example

1. 将开机自动挂载配置文件，拷贝到/example/目录下

[root@server0 ~]# cp /etc/fstab /example

[root@server0 ~]# ls /example

**案例03：server上操作,（MBR分区模式）规划分区**

添加一块80G的硬盘并规划分区：

-划分2个10G的主分区；1个12G的主分区;1个20G的逻辑分区。

[root@server0 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 253:0 0 10G 0 disk

└─vda1 253:1 0 10G 0 part /

vdb 253:16 0 10G 0 disk

vdc 253:32 0 80G 0 disk

[root@server0 ~]# fdisk /dev/vdc

……

命令(输入 m 获取帮助)：**n**

Partition type:

p primary (0 primary, 0 extended, 4 free)

e extended

Select (default p): **p**

分区号 (1-4，默认 1)：**1**

起始 扇区 (2048-167772159，默认为 2048)：  **#回车**

将使用默认值 2048

Last 扇区, +扇区 or +size{K,M,G} (2048-167772159，默认为 167772159)：**+10G**

分区 1 已设置为 Linux 类型，大小设为 10 GiB

命令(输入 m 获取帮助)：**n**

Partition type:

p primary (1 primary, 0 extended, 3 free)

e extended

Select (default p): **#回车**

Using default response p

分区号 (2-4，默认 2)： **#回车**

起始 扇区 (20973568-167772159，默认为 20973568)： **#回车**

将使用默认值 20973568

Last 扇区, +扇区 or +size{K,M,G} (20973568-167772159，默认为 167772159)：**+10G**

分区 2 已设置为 Linux 类型，大小设为 10 GiB

命令(输入 m 获取帮助)：**n**

Partition type:

p primary (2 primary, 0 extended, 2 free)

e extended

Select (default p): **#回车**

Using default response p

分区号 (3,4，默认 3)： **#回车**

起始 扇区 (41945088-167772159，默认为 41945088)： **#回车**

将使用默认值 41945088

Last 扇区, +扇区 or +size{K,M,G} (41945088-167772159，默认为 167772159)：**+12G**

分区 3 已设置为 Linux 类型，大小设为 12 GiB

命令(输入 m 获取帮助)：**n**

Partition type:

p primary (3 primary, 0 extended, 1 free)

e extended

Select (default e): **#回车**

Using default response e

已选择分区 4

起始 扇区 (67110912-167772159，默认为 67110912)： **#回车**

将使用默认值 67110912

Last 扇区, +扇区 or +size{K,M,G} (67110912-167772159，默认为 167772159)： **#回车**

将使用默认值 167772159

分区 4 已设置为 Extended 类型，大小设为 48 GiB

命令(输入 m 获取帮助)：**n**

All primary partitions are in use

添加逻辑分区 5

起始 扇区 (67112960-167772159，默认为 67112960)： **#回车**

将使用默认值 67112960

Last 扇区, +扇区 or +size{K,M,G} (67112960-167772159，默认为 167772159)：**+20G**

分区 5 已设置为 Linux 类型，大小设为 20 GiB

命令(输入 m 获取帮助)：**p**

命令(输入 m 获取帮助)：**w**

[root@server0 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 253:0 0 10G 0 disk

└─vda1 253:1 0 10G 0 part /

vdb 253:16 0 10G 0 disk

vdc 253:32 0 80G 0 disk

├─vdc1 253:33 0 10G 0 part

├─vdc2 253:34 0 10G 0 part

├─vdc3 253:35 0 12G 0 part

├─vdc4 253:36 0 1K 0 part

└─vdc5 253:37 0 20G 0 part

[root@server0 ~]#

**案例:04：在server上操作,分区使用**

1. 案例3中新添加80G硬盘的第一个逻辑分区

– 格式化成xfs文件系统，实现该分区开机自动挂载，挂载点为/mnt/xfs

[root@server0 ~]# mkfs.xfs /dev/vdc5

[root@server0 ~]# blkid /dev/vdc5

/dev/vdc5: UUID="79cd12c3-d552-4e35-857b-1d0a9b60dccb" TYPE="xfs"

[root@server0 ~]# mkdir /mnt/xfs

[root@server0 ~]# vim /etc/fstab

/dev/vdc5 /mnt/xfs xfs defaults 0 0

[root@server0 ~]# tail -1 /etc/fstab

/dev/vdc5 /mnt/xfs xfs defaults 0 0

[root@server0 ~]# df –h

[root@server0 ~]# mount -a

[root@server0 ~]# df -h

1. 案例3中新添加80G硬盘的第一个主分区

– 完成开机自动挂载，挂载点/mnt/mypart,文件系统为ext4

[root@server0 ~]# mkfs.ext4 /dev/vdc1

[root@server0 ~]# blkdid /dev/vdc1

[root@server0 ~]# mkdir /mnt/mypart

[root@server0 ~]# vim /etc/fstab

/dev/vdc1 /mnt/mypart ext4 defaults 0 0

[root@server0 ~]# tail -1 /etc/fstab

/dev/vdc1 /mnt/mypart ext4 defaults 0 0

[root@server0 ~]# df –h

[root@server0 ~]# mount -a

[root@server0 ~]# df -h

**案例05：虚拟机 server0操作，构建 LVM 存储**

1. 新建一个名为 systemvg 的卷组

[root@server0 ~]# vgs

No volume groups found

[root@server0 ~]# vgcreate systemvg /dev/vdc3

Physical volume "/dev/vdc3" successfully created

Volume group "systemvg" successfully created

[root@server0 ~]# vgs

1. 在此卷组中创建一个名为 vo 的逻辑卷，大小为8G

[root@server0 ~]# lvcreate -L 8G -n vo systemvg

Logical volume "vo" created

[root@server0 ~]# lvs

1. 将逻辑卷 vo 格式化为 xfs 文件系统

[root@server0 ~]# mkfs.xfs /dev/systemvg/vo

[root@server0 ~]# blkid /dev/systemvg/vo

1. 将逻辑卷 vo 挂载到 /vo 目录，并在此目录下建立一个测试文件 votest.txt，内容为“I AM KING.”

[root@server0 ~]# mkdir /vo

[root@server0 ~]# mount /dev/systemvg/vo /vo

[root@server0 ~]# echo I AM KING > /vo/votest.txt

[root@server0 ~]# ls /vo/

1. 实现逻辑卷vo开机自动挂载到/vo

[root@server0 ~]# vim /etc/fstab

/dev/systemvg/vo /vo xfs defaults 0 0

[root@server0 ~]# tail -1 /etc/fstab

/dev/systemvg/vo /vo xfs defaults 0 0

[root@server0 ~]# df –h

[root@server0 ~]# mount -a

[root@server0 ~]# df -h

**案例06：虚拟机 server0操作，构建 LVM 存储(修改PE大小)**

1. 新的逻辑卷命名为 database，其大小为50个PE的大小，属于 datastore 卷组
2. 在 datastore 卷组中其PE的大小为1M

[root@server0 ~]# fdisk /dev/vdc

欢迎使用 fdisk (util-linux 2.23.2)。

更改将停留在内存中，直到您决定将更改写入磁盘。

使用写入命令前请三思。

命令(输入 m 获取帮助)：**n**

All primary partitions are in use

添加逻辑分区 6

起始 扇区 (109058048-167772159，默认为 109058048)： **#回车**

将使用默认值 109058048

Last 扇区, +扇区 or +size{K,M,G} (109058048-167772159，默认为 167772159)：**+5G**

命令(输入 m 获取帮助)：**p**

命令(输入 m 获取帮助)：**w**

[root@server0 ~]# partprobe

[root@server0 ~]# lsblk

[root@server0 ~]# vgcreate -s 1M datastore /dev/vdc6

Physical volume "/dev/vdc6" successfully created

Volume group "datastore" successfully created

[root@server0 ~]# vgs

[root@server0 ~]# lvcreate -l 50 -n database datastore

Logical volume "database" created

[root@server0 ~]# lvs

1. 使用 EXT4 文件系统对逻辑卷 database 格式化，此逻辑卷应该在开机时自动挂载到 /mnt/database 目录

[root@server0 ~]# mkfs.ext4 /dev/datastore/database

[root@server0 ~]# vim /etc/fstab

/dev/datastore/database /mnt/database ext4 defaults 0 0

[root@server0 ~]# tail -1 /etc/fstab

/dev/datastore/database /mnt/database ext4 defaults 0 0

[root@server0 ~]# df –h

[root@server0 ~]# mount -a

[root@server0 ~]# df -h

**案例:07：虚拟机 server0操作，扩展逻辑卷**

将/dev/systemvg/vo逻辑卷的大小扩展到20G

[root@server0 ~]# fdisk /dev/vdc

欢迎使用 fdisk (util-linux 2.23.2)。

更改将停留在内存中，直到您决定将更改写入磁盘。

使用写入命令前请三思。

命令(输入 m 获取帮助)：**n**

All primary partitions are in use

添加逻辑分区 7

起始 扇区 (161488896-167772159，默认为 161488896)：**#回车**

将使用默认值 161488896

Last 扇区, +扇区 or +size{K,M,G} (161488896-167772159，默认为 167772159)：**#回车**

将使用默认值 167772159

命令(输入 m 获取帮助)：w

[root@server0 ~]# partprobe

[root@server0 ~]# lsblk

[root@server0 ~]# vgextend systemvg /dev/vdc7

[root@server0 ~]# vgs

[root@server0 ~]# lvextend -L 20G /dev/systemvg/vo

[root@server0 ~]# lvs

[root@server0 ~]# df –h

[root@server0 ~]# xfs\_growfs /dev/systemvg/vo

[root@server0 ~]# df –h

**案例08：vim文本编辑器练习，命令模式下的基本操作（重点知识）**

1. 将/etc/passwd的前20行，写入到文件/opt/pass20.txt

[root@server0 ~]# head -20 /etc/passwd > /opt/pass20.txt

1. 将/etc/shadow的前10行,追加写入到文件/opt/pass20.txt

[root@server0 ~]# head -10 /etc/shadow >> /opt/pass20.txt

1. 利用/opt/pass20.txt，进项如下操作：（练习操作，练习三遍）

– 命令模式下移动光标：键盘上下左右键、Home键、End键

– 命令模式下行间跳转：到全文的第一行（1G或gg）、到全文的最后一行（G）、到全文的第10行（10G）

1. 命令模式下复制、粘贴：（练习操作，练习三遍）

-复制1行（yy）、复制3行（3yy）

-粘贴到当前行之后（小写p）

1. 命令模式下删除：（练习操作，练习三遍）

-删除单个字符（x）

-删除到行首（d^）、删除到行尾（d$）

-删除1行（dd）、删除3行（3dd）

1. 命令模式下查找关键词：（练习操作，练习三遍）

-搜索（/word）切换结果（n、N）

1. 进入插入模式操作（练习操作，练习三遍）

-在命令模式下大写的C，可以删除光标之后，并且进入插入模式

**案例09：书写hello world脚本**

[root@server0 ~]# vim /root/hello.sh

#!/bin/bash

echo hello world

[root@server0 ~]# chmod +x /root/hello.sh

[root@server0 ~]# /root/hello.sh #绝对路径来运行

**案例10： 测试IP地址脚本**

利用read读入用户输入IP地址，进行判断

如果本机能够ping，则输出可以通信

如果本机不能够ping，则输出不可以通信

[root@server0 /]# vim /root/ip.sh

#!/bin/bash

read -p '请输入您要测试的IP地址:' ip

ping -c 2 $ip &> /dev/null

if [ $? -eq 0 ];then

echo 可以通信

else

echo 不可以通信

fi

[root@server0 /]#

**案例11：if多分支结构**

利用read实现，用户输入考试成绩

如果成绩 大于等于90，则输出 优秀

如果成绩 大于等于80，则输出 良好

如果成绩 大于等于70，则输出 一般

如果成绩 大于等于60，则输出 合格

以上均不满足：则输出 一首凉凉送给你

[root@server0 /]# vim /root/nsd.sh

#!/bin/bash

read -p '请输入您的成绩：' num

if [ $num -ge 90 ];then

echo 优秀

elif [ $num -ge 80 ];then

echo 良好

elif [ $num -ge 70 ];then

echo 一般

elif [ $num -ge 60 ];then

echo 合格

else

echo '一首凉凉送给你!'

fi

[root@server0 /]#

**案例12：for循环**

在屏幕输出20遍: I Love girl

[root@server0 ~]# vim /root/for.sh

#!/bin/bash

for i in {1..20}

do

echo I Love girl

done

[root@server0 ~]# chmod +x /root/for.sh

[root@server0 ~]# /root/for.sh

**案例13：设置SELinux保护**

为虚拟机 server0、虚拟机desktop0 配置SELinux

-确保 SELinux 处于强制启用模式（permissive）

-在每次重新开机后，此设置必须仍然有效

[root@server0 ~]# getenforce #查看当前SELinux的运行模式

Enforcing

[root@server0 ~]# setenforce 0 #修改当前运行模式

[root@server0 ~]# getenforce

Permissive

[root@server0 ~]# vim /etc/selinux/config #每次开机运行模式

SELINUX=permissive

**案例14：自定义用户环境**

为系统 server0 创建自定义命令

-自定义命令的名称为 qstat

-此自定义命令将执行以下操作： /bin/ps -Ao pid,tt,user,fname,rsz

-此自定义命令对系统中的所有用户都有

[root@server0 ~]# vim /etc/bashrc #全局配置文件

alias qstat='/bin/ps -Ao pid,tt,user,fname,rsz'

**案例15：Shell脚本**

为系统 server0 书写脚本/root/user.sh

运行脚本，可以判断用户输入的用户是否存在

如果存在，输出用户基本信息（id 用户名）

如果用户，不存在则创建用户，并输出用户创建成功

[root@server0 ~]# vim /root/user.sh

#!/bin/bash

read -p '请输入您要测试的用户名:' user

id $user &> /dev/null

if [ $? -eq 0 ];then

echo $user已存在

id $user

else

useradd $user

echo $user 创建成功

fi

[root@server0 ~]#

**案例16：实现本机server0 的Web服务**

利用httpd软件搭建Web服务，页面显示内容为 我要像风一样自由

[root@server0 ~]# yum -y install httpd

[root@server0 ~]# echo '<h1>我要像风一样自由' > /var/www/html/index.html

[root@server0 ~]# cat /var/www/html/index.html

<h1>我要像风一样自由

[root@server0 ~]# systemctl restart httpd

[root@server0 ~]# systemctl enable httpd

ln -s '/usr/lib/systemd/system/httpd.service' '/etc/systemd/system/multi-user.target.wants/httpd.service'

[root@server0 ~]# firefox 172.25.0.11

**案例17：实现本机server0 的防火墙配置**

1. 修改虚拟机 server0防火墙配置，明确拒绝所有客户端访问（默认区域设置为block）

[root@server0 ~]# firewall-cmd --set-default-zone=block

[root@server0 ~]# firewall-cmd --get-default-zone

1. 在虚拟机 desktop0上，测试能否访问server0 的Web服务

无法访问，有回应

1. 在虚拟机 desktop0上，测试能否 ping通 虚拟机 server0

无法通信，有回应

**案例18：实现本机server0 的防火墙配置**

1. 修改虚拟机 server0防火墙配置，将默认区域修改为public

[root@server0 ~]# firewall-cmd --set-default-zone=public

[root@server0 ~]# firewall-cmd --get-default-zone

1. 在虚拟机 desktop0上，测试能否访问server0 的Web服务

可以访问成功

1. 在虚拟机 desktop0上，测试能否 ping通 虚拟机 server0

可以成功通信

**案例19：实现本机server0 的防火墙配置**

1. 修改虚拟机 server0防火墙配置，将默认区域修改为public

[root@server0 ~]# firewall-cmd --set-default-zone=public

[root@server0 ~]# firewall-cmd --get-default-zone

1. 修改虚拟机 server0防火墙配置，在public区域中添加http服务

[root@server0 ~]# firewall-cmd --permanent --zone=public --add-service=http

success

[root@server0 ~]# firewall-cmd --reload

success

[root@server0 ~]# firewall-cmd --zone=public --list-all

1. 在虚拟机 desktop0上测试能否访问server0 的Web服务

可以访问成功

**案例20：配置聚合连接**

在两个虚拟机之间配置一个链路，要求如下：

– 此链路使用接口 eth1 和 eth2

– 此链路在其中一个接口失效时仍然能工作

– 此链路在 server0 上使用下面的地址 172.16.3.20/255.255.255.0

– 此链路在 desktop0 上使用下面的地址 172.16.3.25/255.255.255.0

– 此链路在系统重启之后依然保持正常状态

1.制作虚拟网卡 team0, 参考 man teamd.conf 全文查找/example 按n跳转匹配项

# nmcli connection add type team

autoconnect yes con-name team0 ifname team0

config '{"runner": {"name": "activebackup"}}'

# nmcli connection 添加 类型为 team 的设备

每次开机自动启用 配置文件命名为 team0 网卡显示的名字为 team0

team0网卡内部成员工作模式为 '{"runner": {"name": "activebackup"}}' （热备）

# ifconfig

2. 为 team0添加成员

# nmcli connection add type team-slave con-name team0-1 ifname eth1 master team0

# nmcli connection add type team-slave con-name team0-2 ifname eth2 master team0

# 添加 类型为 team-slave 的设备 配置文件命名为 team0-1 网卡为eth1

主设备为team0

3.配置team0的ip地址与激活

# nmcli connection modify team0 ipv4.method manual

ipv4.addresses 172.16.3.20/24 connection.autoconnect yes

# nmcli connection up team0 #激活team0网卡

# nmcli connection up team0-1 #激活team0-1成员

# nmcli connection up team0-2 #激活team0-2成员

**案例21：为虚拟机 server0 配置以下静态地址参数**

1. 主机名:server0.example.com

[root@server0 ~]# echo server0.example.com > /etc/hostname

[root@server0 ~]# cat /etc/hostname

server0.example.com

[root@server0 ~]# exit

登出

Connection to 172.25.0.11 closed.

[root@room9pc01 ~]# ssh -X root@172.25.0.11

[root@server0 ~]# hostname

server0.example.com

[root@server0 ~]#

1. IP地址:172.25.0.100
2. 子网掩码:255.255.255.0
3. 默认网关:172.25.0.254

[root@server0 ~]# nmcli connection modify 'System eth0' ipv4.method manual ipv4.addresses '172.25.0.100/24 172.25.0.254' connection.autoconnect yes

[root@server0 ~]# nmcli connection up 'System eth0'

[root@room9pc01 ~]# ssh -X root@172.25.0.100

[root@server0 ~]# ifconfig | head -2

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 172.25.0.100 netmask 255.255.255.0 broadcast 172.25.0.255

[root@server0 ~]# route -n

1. DNS服务器:172.25.254.254

[root@server0 ~]# echo nameserver 172.25.254.254 > /etc/resolv.conf

[root@server0 ~]# cat /etc/resolv.conf

nameserver 172.25.254.254

[root@server0 ~]#

**案例22：实现本机server0 的Web服务**

利用httpd软件搭建Web服务，页面显示内容为 卖女孩的小火柴

[root@server0 ~]# yum -y install httpd

[root@server0 ~]# echo '<h1>卖女孩的小火柴' > /var/www/html/index.html

[root@server0 ~]# cat /var/www/html/index.html

<h1>卖女孩的小火柴

[root@server0 ~]# systemctl restart httpd

[root@server0 ~]# systemctl enable httpd

ln -s '/usr/lib/systemd/system/httpd.service' '/etc/systemd/system/multi-user.target.wants/httpd.service'

[root@server0 ~]# firefox 172.25.0.11

**案例23：实现本机server0 的防火墙配置**

1. 修改虚拟机 server0防火墙配置，将默认区域修改为trusted

[root@server0 ~]# firewall-cmd --set-default-zone=trusted

[root@server0 ~]# firewall-cmd --get-default-zone

1. trusted区域设置端口转发策略：从desktop0上访问server0的5423端口，与访问server0的80端口效果一样

虚拟机server0:

[root@server0 ~]# firewall-cmd --permanent --zone=trusted

--add-forward-port=port=5423:proto=tcp:toport=80

[root@server0 ~]# firewall-cmd --zone=trusted --list-all

[root@server0 ~]# firewall-cmd --reload

[root@server0 ~]# firewall-cmd --zone=trusted --list-all

1. 必须利用客户端desktop0测试：

[root@desktop0 ~]# firefox 172.25.0.11:5423

**案例24：配置Samba文件夹共享**

1. 防火墙默认区域设置为trusted，修改SElinux状态为Enforcing

[root@server0 ~]# getenforce #查看当前SELinux的运行模式

Enforcing

[root@server0 ~]# vim /etc/selinux/config #每次开机运行模式

SELINUX=Enforcing

[root@server0 ~]# firewall-cmd --set-default-zone=trusted

[root@server0 ~]# firewall-cmd --get-default-zone

1. 在 server0 上发布共享文件夹

-此服务器必须是 STAFF 工作组的一个成员

-发布目录 /devops，共享名为 devops

-用户 harry 对共享 devops可读可写，密码是 123

1.安装软件包samba

[root@server0 ~]# yum –y install samba

2.建立samba共享帐号，专用于验证samba共享服务用户名

[root@server0 ~]# useradd -s /sbin/nologin harry

[root@server0 ~]# pdbedit -a harry #添加为samba共享帐号

[root@server0 ~]# pdbedit -L #列出所有的samba共享帐号

3.发布目录 /devops,共享名为 devops

[root@server0 ~]# mkdir /devops

[root@server0 ~]# echo test > /devops/1.txt

[root@server0 ~]# ls /devops

[root@server0 ~]# vim /etc/samba/smb.conf #按G(大写)到最后\

1)此服务器必须是 STAFF 工作组的一个成员

workgroup = STAFF

[devops]

path = /devops

write list = harry

4.重起smb服务

[root@server0 ~]# systemctl restart smb

[root@server0 ~]# systemctl enable smb

ln -s '/usr/lib/systemd/system/smb.service' '/etc/systemd/system/multi-user.target.wants/smb.service'

5.SELinux的 布尔值（功能的开关 on与off）

– 需要加 -P 选项才能实现永久设置

[root@server0 ~]# getsebool -a | grep samba #查看SELinux布尔值

[root@server0 ~]# setsebool samba\_export\_all\_ro on #修改SELinux布尔值

[root@server0 ~]# getsebool -a | grep samba

6.设置本机目录的基本的权限

[root@server0 ~]# setfacl -m u:harry:rwx /devops/

[root@server0 ~]# getfacl /devops/

1. 在 desktop0 上访问共享文件夹

-访问 server0 上发布共享文件夹,实现开机自动挂载到/mnt/smb

-实现对/mnt/smb目录可以读和写

1.书写配置文件完成开机自动挂载

[root@desktop0 ~]# vim /etc/fstab

//172.25.0.11/devops /mnt/smb cifs

defaults,user=harry,pass=123,\_netdev 0 0

[root@desktop0 ~]# mkdir /mnt/smb

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h

[root@desktop0 ~]# touch /mnt/nsd02/test.txt

**案例25：配置Samba文件夹共享**

在 server0 上发布共享文件夹

-发布目录 /public，共享名为 nb

-用户 kenji 对共享 nb可读可写，密码是 321

1.安装软件包samba

[root@server0 ~]# yum –y install samba

2.建立samba共享帐号，专用于验证samba共享服务用户名

[root@server0 ~]# useradd -s /sbin/nologin kenji

[root@server0 ~]# pdbedit -a kenji #添加为samba共享帐号

[root@server0 ~]# pdbedit -L #列出所有的samba共享帐号

3.发布目录 /public,共享名为 public

[root@server0 ~]# mkdir /public

[root@server0 ~]# echo test > /public/1.txt

[root@server0 ~]# ls /public

[root@server0 ~]# vim /etc/samba/smb.conf #按G(大写)到最后\

[nb]

path = /public

write list = kenji

4.重起smb服务

[root@server0 ~]# systemctl restart smb

[root@server0 ~]# systemctl enable smb

ln -s '/usr/lib/systemd/system/smb.service' '/etc/systemd/system/multi-user.target.wants/smb.service'

5.SELinux的 布尔值（功能的开关 on与off）

– 需要加 -P 选项才能实现永久设置

[root@server0 ~]# getsebool -a | grep samba #查看SELinux布尔值

[root@server0 ~]# setsebool samba\_export\_all\_ro on #修改SELinux布尔值

[root@server0 ~]# getsebool -a | grep samba

6.设置本机目录的基本的权限

[root@server0 ~]# setfacl -m u:kenji:rwx /devops/

[root@server0 ~]# getfacl /devops/

1. 在 desktop0 上访问共享文件夹

-访问 server0 上发布共享文件夹,实现开机自动挂载到/mnt/nsd

-实现对/mnt/nsd目录可以读和写

1.书写配置文件完成开机自动挂载

[root@desktop0 ~]# vim /etc/fstab

//172.25.0.11/nb /mnt/nsd cifs

defaults,user=kenji,pass=123,\_netdev 0 0

[root@desktop0 ~]# mkdir /mnt/nsd

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h

[root@desktop0 ~]# touch /mnt/nsd/test.txt

**案例26：普通NFS共享的实现**

1）在 server0 上配置NFS服务

-只读的方式共享目录 /mynfs，只能被172.25.0.0/24网段中的客户机系统访问

1.安装软件nfs-utils

[root@server0 ~]# yum -y install nfs-utils

[root@server0 ~]# rpm -q nfs-utils

2.发布共享 /abc

[root@server0 ~]# mkdir /mynfs

[root@server0 ~]# echo 123 > /mynfs/123.txt

[root@server0 ~]# ls /mynfs

[root@server0 ~]# vim /etc/exports

/mynfs 172.25.0.0/24(ro)

3.重起nfs-server服务

[root@server0 ~]# systemctl restart nfs-server

[root@server0 ~]# systemctl enable nfs-server

1. 在 desktop0 上访问NFS共享目录

-将 server0 的 /mynfs 挂到本地 /mnt/nfsmount

-这些文件系统在系统启动时自动挂载

[root@desktop0 ~]# vim /etc/fstab

172.25.0.11:/mynfs /mnt/nfsmount nfs defaults,\_netdev 0 0

[root@desktop0 ~]# mkdir /mnt/nfsmount

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h

**案例27：在server上操作,（MBR分区模式）规划分区**

添加一块80G的硬盘并规划分区：

-划分2个10G的主分区；1个12G的主分区;1个20G的逻辑分区

[root@server0 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 253:0 0 10G 0 disk

└─vda1 253:1 0 10G 0 part /

vdb 253:16 0 10G 0 disk

vdc 253:32 0 80G 0 disk

[root@server0 ~]# fdisk /dev/vdc

……

命令(输入 m 获取帮助)：**n**

Partition type:

p primary (0 primary, 0 extended, 4 free)

e extended

Select (default p): **p**

分区号 (1-4，默认 1)：**1**

起始 扇区 (2048-167772159，默认为 2048)：  **#回车**

将使用默认值 2048

Last 扇区, +扇区 or +size{K,M,G} (2048-167772159，默认为 167772159)：**+10G**

分区 1 已设置为 Linux 类型，大小设为 10 GiB

命令(输入 m 获取帮助)：**n**

Partition type:

p primary (1 primary, 0 extended, 3 free)

e extended

Select (default p): **#回车**

Using default response p

分区号 (2-4，默认 2)： **#回车**

起始 扇区 (20973568-167772159，默认为 20973568)： **#回车**

将使用默认值 20973568

Last 扇区, +扇区 or +size{K,M,G} (20973568-167772159，默认为 167772159)：**+10G**

分区 2 已设置为 Linux 类型，大小设为 10 GiB

命令(输入 m 获取帮助)：**n**

Partition type:

p primary (2 primary, 0 extended, 2 free)

e extended

Select (default p): **#回车**

Using default response p

分区号 (3,4，默认 3)： **#回车**

起始 扇区 (41945088-167772159，默认为 41945088)： **#回车**

将使用默认值 41945088

Last 扇区, +扇区 or +size{K,M,G} (41945088-167772159，默认为 167772159)：**+12G**

分区 3 已设置为 Linux 类型，大小设为 12 GiB

命令(输入 m 获取帮助)：**n**

Partition type:

p primary (3 primary, 0 extended, 1 free)

e extended

Select (default e): **#回车**

Using default response e

已选择分区 4

起始 扇区 (67110912-167772159，默认为 67110912)： **#回车**

将使用默认值 67110912

Last 扇区, +扇区 or +size{K,M,G} (67110912-167772159，默认为 167772159)： **#回车**

将使用默认值 167772159

分区 4 已设置为 Extended 类型，大小设为 48 GiB

命令(输入 m 获取帮助)：**n**

All primary partitions are in use

添加逻辑分区 5

起始 扇区 (67112960-167772159，默认为 67112960)： **#回车**

将使用默认值 67112960

Last 扇区, +扇区 or +size{K,M,G} (67112960-167772159，默认为 167772159)：**+20G**

分区 5 已设置为 Linux 类型，大小设为 20 GiB

命令(输入 m 获取帮助)：**p**

命令(输入 m 获取帮助)：**w**

[root@server0 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 253:0 0 10G 0 disk

└─vda1 253:1 0 10G 0 part /

vdb 253:16 0 10G 0 disk

vdc 253:32 0 80G 0 disk

├─vdc1 253:33 0 10G 0 part

├─vdc2 253:34 0 10G 0 part

├─vdc3 253:35 0 12G 0 part

├─vdc4 253:36 0 1K 0 part

└─vdc5 253:37 0 20G 0 part

[root@server0 ~]#

**案例28：vim文本编辑器练习，命令模式下的基本操作（重点知识）**

1. 将/etc/passwd的前10行，写入到文件/opt/pass30.txt

[root@server0 ~]# head -10 /etc/passwd

[root@server0 ~]# head -10 /etc/passwd > /opt/pass30.txt

1. 将/etc/shadow的前20行,追加写入到文件/opt/pass30.txt

[root@server0 ~]# tail -20 /etc/passwd

[root@server0 ~]# tail -20 /etc/passwd >> /opt/pass30.txt

1. 利用/opt/pass30.txt，进项如下操作：（练习操作，练习三遍）

– 命令模式下移动光标：键盘上下左右键、Home键、End键

– 命令模式下行间跳转：到全文的第一行（1G或gg）、到全文的最后一行（G）、到全文的第12行（10G）

命令模式下复制、粘贴：（练习操作，练习三遍）

-复制2行（yy）、复制4行（4yy）

-粘贴到当前行之后（小写p）

1. 命令模式下删除：（练习操作，练习三遍）

-删除单个字符（x）

-删除到行首（d^）、删除到行尾（d$）

-删除第3行（光标在第三行dd）、同时删除3行（3dd）

1. 命令模式下查找关键词：（练习操作，练习三遍）

-搜索（/word）切换结果（n、N）

1. 进入插入模式操作（练习操作，练习三遍）

-在命令模式下大写的C，可以删除光标之后，并且进入插入模式

**案例29：在 server0、desktop0 上操作**

将防火墙默认区域设置为trusted

虚拟机Server0

[root@server0 ~]# firewall-cmd --set-default-zone=trusted

虚拟机Desktop0

[root@desktop0 ~]# firewall-cmd --set-default-zone=trusted

**案例30：在server上操作，搭建mariadb数据库系统**

在 server0 上安装 MariaDB 数据库系统

-安装 mariadb-server软件包

-启动 mariadb 服务

1.安装mariadb-server:提供服务端有关的系统程序

[root@server0 ~]# yum -y install mariadb-server

2.重起服务

[root@server0 ~]# systemctl restart mariadb

[root@server0 ~]# systemctl enable mariadb

**案例31：在server上操作，配置一个数据库**

1. 为mariadb数据库root设置登陆密码为 haxi

在Linux系统命令行，为数据库管理员设置密码haxi

[root@server0 ~]# mysqladmin -u root password 'haxi'

1. 新建一个数据库名为 nsd

[root@server0 ~]# mysql -u root –phaxi

> show databases; #显示所有的库

> create database nsd; #创建nsd库

> show databases; #显示所有的库

> exit #退出

1. 除了 root 用户,此nsd数据库只能被用户 lisi 查询,此用户的密码为123(用户的授权)

[root@server0 ~]# mysql -u root –phaxi

> grant select on nsd.\* to lisi@localhost identified by '123';

1. 数据库 nsd 中应该包含来自数据库复制的内容（导入数据）

-数据库文件的 URL为:http://classroom/pub/materials/users.sql

1.Linux命令行，下载备份文件

[root@server0 ~]# wget http://classroom.example.com/pub/materials/users.sql

[root@server0 ~]# ls

2.Linux命令行，导入数据到nsd库

[root@server0 ~]# mysql -u root -phaxi nsd < users.sql

[root@server0 ~]# mysql -u root -phaxi

MariaDB [(none)]> use nsd; #进入nsd库

MariaDB [nsd]> show tables; #显示所有表格

+---------------+

| Tables\_in\_nsd |

+---------------+

| base |

| location |

+---------------+

2 rows in set (0.00 sec)

**案例32：在server上操作，使用数据库查询**

1. 密码是 solicitous 的人的名字?（写出查询命令）

[root@server0 ~]# mysql -u root -phaxi

MariaDB [(none)]> use nsd; #进入nsd库

MariaDB [nsd]> select \* from base where password='solicitous';

1. 有多少人的姓名是 Barbara 同时居住在 Sunnyvale? （写出查询命令及结果）

[root@server0 ~]# mysql -u root -phaxi

MariaDB [(none)]> use nsd; #进入nsd库

MariaDB [nsd]> select \* from base,location

where base.name='Barbara' and

location.city='Sunnyvale' and

base.id=location.id;

MariaDB [nsd]> select count(\*) from base,location #统计匹配条件记录

where base.name='Barbara' and

location.city='Sunnyvale' and

base.id=location.id;

1. 在base表中追加记录 id为6，name为Barbara，password为900（写出插入表记录命令）

[root@server0 ~]# mysql -u root -phaxi

MariaDB [(none)]> use nsd; #进入nsd库

MariaDB [nsd]> insert base values ('6','Barbara','900');

MariaDB [nsd]> select \* from base;

1. 在location表中追加记录 id为6，city为Sunnyvale（写出插入表记录命令）

[root@server0 ~]# mysql -u root -phaxi

MariaDB [(none)]> use nsd; #进入nsd库

MariaDB [nsd]> insert location values ('6','Sunnyvale');

MariaDB [nsd]> select \* from location;

1. 再次查询有多少人的姓名是 Barbara 同时居住在 Sunnyvale? （写出查询命令及结果）

[root@server0 ~]# mysql -u root -phaxi

MariaDB [(none)]> use nsd; #进入nsd库

MariaDB [nsd]> select \* from base,location

where base.name='Barbara' and

location.city='Sunnyvale' and

base.id=location.id;

MariaDB [nsd]> select count(\*) from base,location #统计匹配条件记录

where base.name='Barbara' and

location.city='Sunnyvale' and

base.id=location.id;

**案例33：虚拟机 server0操作， /dev/vdb 上按以下要求建立分区：**

采用默认的 msdos 分区模式

– 第1个分区 /dev/vdb1 的大小为 3G

– 第2个分区 /dev/vdb2 的大小为 200M

– 第3个分区 /dev/vdb3 的大小为 100M

– 第4个分区 /dev/vdb4为扩展分区

– 在划分二个分区逻辑分区/dev/vdb[5-6]，

– 分区大小依次为500M、2000M

[root@server0 ~]# lsblk

[root@server0 ~]# fdisk /dev/vdb

……

命令(输入 m 获取帮助)：**n**

Partition type:

p primary (0 primary, 0 extended, 4 free)

e extended

Select (default p): **p**

分区号 (1-4，默认 1)：**#回车**

起始 扇区 (2048-167772159，默认为 2048)：  **#回车**

将使用默认值 2048

Last 扇区, +扇区 or +size{K,M,G} (2048-167772159，默认为 167772159)：**+3G**

命令(输入 m 获取帮助)：**n**

Partition type:

p primary (1 primary, 0 extended, 3 free)

e extended

Select (default p): **#回车**

Using default response p

分区号 (2-4，默认 2)： **#回车**

起始 扇区 (20973568-167772159，默认为 20973568)： **#回车**

将使用默认值 20973568

Last 扇区, +扇区 or +size{K,M,G} (20973568-167772159，默认为 167772159)：**+200M**

命令(输入 m 获取帮助)：**n**

Partition type:

p primary (2 primary, 0 extended, 2 free)

e extended

Select (default p): **#回车**

Using default response p

分区号 (3,4，默认 3)： **#回车**

起始 扇区 (41945088-167772159，默认为 41945088)： **#回车**

将使用默认值 41945088

Last 扇区, +扇区 or +size{K,M,G} (41945088-167772159，默认为 167772159)：**+100M**

命令(输入 m 获取帮助)：**n**

Partition type:

p primary (3 primary, 0 extended, 1 free)

e extended

Select (default e): **#回车**

Using default response e

已选择分区 4

起始 扇区 (67110912-167772159，默认为 67110912)： **#回车**

将使用默认值 67110912

Last 扇区, +扇区 or +size{K,M,G} (67110912-167772159，默认为 167772159)： **#回车**

将使用默认值 167772159

命令(输入 m 获取帮助)：**n**

All primary partitions are in use

添加逻辑分区 5

起始 扇区 (67112960-167772159，默认为 67112960)： **#回车**

将使用默认值 67112960

Last 扇区, +扇区 or +size{K,M,G} (67112960-167772159，默认为 167772159)：**+500M**

命令(输入 m 获取帮助)：**n**

All primary partitions are in use

添加逻辑分区 6

起始 扇区 (67112960-167772159，默认为 67112960)： **#回车**

将使用默认值 67112960

Last 扇区, +扇区 or +size{K,M,G} (67112960-167772159，默认为 167772159)：**+2000M**

命令(输入 m 获取帮助)：**p**

命令(输入 m 获取帮助)：**w**

[root@server0 ~]# lsblk

**案例34：发布iSCSI网络磁盘**

1. 配置 server0 提供 iSCSI 服务，要求如下：

-磁盘名为iqn.2016-02.com.example:server0

-服务端口为 3260

-使用 iscsi\_store（后端存储的名称） 作其后端卷，其大小为 3GiB

-此磁盘服务只能被 desktop0.example.com 访问，在Server0上配置客户端ACL为iqn.2016-02.com.example:desktop0

1.安装软件targetcli（服务端软件）

[root@server0 ~]# yum -y install targetcli

2.运行targetcli可以再交互式界面，配置iSCSI共享存储

[root@server0 ~]# targetcli

A:创建与命名后端存储

/> backstores/block create name= iscsi\_store dev=/dev/vdb1

后端存储 块设备 创建 命名 具体设备

/> ls

B:创建target,磁盘组

/> iscsi/ create iqn.2016-02.com.example:server0

/> ls

C:lun,逻辑单元

/> iscsi/iqn.2016-02.com.example:server0/tpg1/

luns create /backstores/block/iscsi\_store

/> ls

D:配置访问控制，设置访问服务时，客户端声称的名字

/> iscsi/iqn.2016-02.com.example:server0/tpg1/acls

create iqn.2016-02.com.example:desktop0

/> ls

E:客户端访问本机的IP地址及端口

[root@server0 ~]# targetcli

/> iscsi/iqn.2016-02.com.example:server0/tpg1/

portals create 172.25.0.11

/> ls

/> exit 退出

三、重起服务target

[root@server0 ~]# systemctl restart target

[root@server0 ~]# systemctl enable target

1. 配置虚拟机 desktop0 使用 server0 提供 iSCSI 服务

一、安装所需软件包iscsi-initiator-utils

yum安装能够补全包名：

1、前提由Yum的缓存（yum repolist）

2、当前系统没有安装的，才可以补全

[root@desktop0 ~]# rpm -q iscsi-initiator-utils

二、修改配置文件，用来指定客户端声称的名字

1.修改配置文件

# vim /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.2016-02.com.example:desktop0

2.需重启服务 iscsid 以更新IQN标识

[root@desktop0 ~]# systemctl restart iscsid

Warning: Unit file of iscsid.service changed on

disk, 'systemctl daemon-reload' recommended.

[root@desktop0 ~]# systemctl daemon-reload

[root@desktop0 ~]# systemctl restart iscsid

三、发现服务端共享存储

1.书写发现的命令, 参考# man iscsiadm 搜索全文 /example

Ctrl - ：减小字体

Ctrl Shift + ：变大字体

# iscsiadm --mode discoverydb --type sendtargets --

portal 172.25.0.11 --discover

2.本机识别服务端共享

[root@desktop0 ~]# lsblk

[root@desktop0 ~]# systemctl restart iscsi

[root@desktop0 ~]# lsblk

[root@desktop0 ~]# systemctl enable iscsi

**案例35：为虚拟机 server 配置以下虚拟Web主机**

实现三个网站的部署

-客户端访问server0.example.com网页内容为 大圣归来

-客户端访问www0.example.com网页内容为 大圣又归来

-客户端访问webapp0.example.com网页内容为 大圣累了

1.建立新的调用配置文件

[root@server0 ~]# vim /etc/httpd/conf.d/nsd01.conf

<VirtualHost \*:80> #在所有网卡启用80端口

ServerName www0.example.com #指定网站域名

DocumentRoot /var/www/nsd01 #指定网页文件路径

</VirtualHost>

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd02

</VirtualHost>

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /var/www/nsd03

</VirtualHost>

[root@server0 ~]# mkdir /var/www/nsd01 /var/www/nsd02 /var/www/nsd03

[root@server0 ~]# echo 大圣归来 > /var/www/nsd01/index.html

[root@server0 ~]# echo 大圣又归来 > /var/www/nsd02/index.html

[root@server0 ~]# echo 大圣累了 > /var/www/nsd03/index.html

[root@server0 ~]# systemctl restart httpd

[root@server0 ~]# firefox www0.example.com

[root@server0 ~]# firefox webapp0.example.com

[root@server0 ~]# firefox server0.example.com

**案例36：为虚拟机 server 配置Web访问控制**

在 Web 网站 http://server0.example.com 的 DocumentRoot 目录下创建一个名为 private 的子目录，要求如下：

-在server0.example.com的DocumentRoot目录下，private的子目录里书写网页文件index.html内容为 大圣偷偷归来

-此页面只能在本机浏览，但是从其他系统不能访问这个目录的内容

虚拟机Server0

[root@server0 ~]# mkdir /var/www/myweb/private

[root@server0 ~]# echo大圣偷偷归来 > /var/www/myweb/private/index.html

[root@server0 ~]# vim /etc/httpd/conf.d/nsd02.conf

<Directory "/var/www/myweb/private">

Require ip 172.25.0.11

</Directory>

[root@server0 ~]# systemctl restart httpd

客户端虚拟机Desktop：访问测试

[root@desktop0 ~]# firefox server0.example.com/private

网页报错信息:

Forbidden

You don't have permission to access /private/ on this server.

**案例37：虚拟机 server 使用自定Web根目录**

调整 Web 站点 http://server0.example.com 的网页目录，要求如下：

-新建目录 /webroot，作为此站点新的网页目录

-确保站点 http://server0.example.com 仍然可访问

虚拟机Server0：

1.新建目录 /webroot,作为此站点新的网页目录

[root@server0 ~]# mkdir /webroot

[root@server0 ~]# echo wo shi webroot > /webroot/index.html

2.修改配置文件/etc/httpd/conf.d/nsd01.conf

<VirtualHost \*:80>

ServerName www0.example.com

DocumentRoot /var/www/nsd01

</VirtualHost>

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd02

</VirtualHost>

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /webroot

</VirtualHost>

3.修改配置文件/etc/httpd/conf.d/nsd02.conf(访问控制)

<Directory "/var/www/myweb/private">

Require ip 172.25.0.11

</Directory>

<Directory "/webroot">

Require all granted

</Directory>

4.SELinux安全上下文（标签值）

• 参照标准目录,重设新目录的属性

– chcon [-R] --reference=模板目录 新目录

[root@server0 ~]# ls -Zd /webroot/

[root@server0 ~]# chcon -R --reference=/var/www /webroot/

[root@server0 ~]# ls -Zd /webroot/

5.重起服务

[root@server0 ~]# systemctl restart httpd

客户端desktop:访问测试

[root@server0 ~]# firefox server0.example.com

**案例38：为虚拟机 server 部署动态WSGI站点**

为站点 webapp0.example.com 配置提供动态Web内容，要求如下：

-此虚拟主机侦听在端口8909

-测试网页从以下地址下载，不要作任何更改

http://classroom/pub/materials/webinfo.wsgi

-从浏览器访问 http://webapp0.example.com:8909 可接收到动态生成的 Web 页面

1.部署Python的页面

[root@server0 ~]# vim /etc/httpd/conf.d/nsd01.conf

查看webapp0.example.com的DocumentRoot路径

# cd /var/www/nsd02/

# wget http://classroom.example.com/pub/materials/webinfo.wsgi

2.方便用户访问,网页跳转（网页别名）

[root@server0 /]# vim /etc/httpd/conf.d/nsd01.conf

......

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd02

Alias / /var/www/nsd02/webinfo.wsgi

#当用户访问网页文件根目录时，将webinfo.wsgi呈现

</VirtualHost>

......

[root@server0 /]# systemctl restart httpd

[root@server0 /]# firefox webapp0.example.com

3.翻译Python页面

[root@server0 /]# yum -y install mod\_wsgi

[root@server0 /]# vim /etc/httpd/conf.d/nsd01.conf

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd02

WsgiScriptAlias / /var/www/nsd02/webinfo.wsgi

</VirtualHost>

[root@server0 /]# systemctl restart httpd

[root@server0 /]# firefox webapp0.example.com

UNIX时间戳：自1970-1-1 0:0:0算起，到达当前时间所经历的秒数

4.修改虚拟Web主机的端口

[root@server0 /]# vim /etc/httpd/conf.d/nsd01.conf

<VirtualHost \*:80>

ServerName www0.example.com

DocumentRoot /var/www/nsd01

</VirtualHost>

Listen 8909

<VirtualHost \*:8909>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd02

WsgiScriptAlias / /var/www/nsd02/webinfo.wsgi

</VirtualHost>

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /var/www/nsd03

</VirtualHost>

5.修改SELinux开放的端口，-a添加 -t 类型 -p 协议

# semanage port -l | grep http

# semanage port -a -t http\_port\_t -p tcp 8909

# semanage port -l | grep http #查看是否添加成功

# systemctl restart httpd

[root@desktop0 ~]# firefox webapp0.example.com:8909

**案例39：配置安全Web服务**

为站点 http://server0.example.com 配置TLS加密

一个已签名证书从以下地址获取 http://classroom/pub/tls/certs/server0.crt

此证书的密钥从以下地址获取 http://classroom/pub/tls/private/server0.key

此证书的签名授权信息从以下地址获取 http://classroom/pub/example-ca.crt

虚拟机Server0：

1.部署网站证书（营业执照）

[root@server0 /]# cd /etc/pki/tls/certs/

[root@server0 certs]# wget http://classroom.example.com/pub/tls/certs/server0.crt

[root@server0 certs]# ls

2.部署数字证书授权中心信息（根证书）（公安局信息）

[root@server0 /]# cd /etc/pki/tls/certs/

[root@server0 certs]# wget http://classroom.example.com/pub/example-ca.crt

[root@server0 certs]# ls

3.部署私钥文件,用于解密数据

[root@server0 certs]# cd /etc/pki/tls/private/

[root@server0 private]# wget http://classroom.example.com/pub/tls/private/server0.key

[root@server0 certs]# ls

4.安装支持安全Web的软件

[root@server0 /]# yum -y install mod\_ssl

5.修改配置文件

[root@server0 /]# echo security example > /var/www/html/index.html

[root@server0 /]# vim /etc/httpd/conf.d/ssl.conf

补充：vim末行模式下 输入 : set nu 显示行号

59 DocumentRoot "/var/www/html"

60 ServerName server0.example.com:443

指定网站证书

100 SSLCertificateFile /etc/pki/tls/certs/server0.crt

指定私钥文件

107 SSLCertificateKeyFile /etc/pki/tls/private/server0.key

指定根证书（数字授权中心信息）

122 SSLCACertificateFile /etc/pki/tls/certs/example-ca.crt

6.重起 httpd 服务

[root@server0 /]# systemctl restart httpd

7.验证

[root@desktop0 ~]# firefox https://www0.example.com

**案例40：虚拟机 server0上操作,（GPT分区模式）规划分区**

关闭虚拟机，图形添加一块60G的硬盘并规划分区：

-划分4个10G的主分区

-划分1个12G的主分区

[root@server0 ~]# lsblk

[root@server0 ~]# parted /dev/sdb

......

(parted) mktable gpt #设置为GPT分区模式

(parted) print #显示分区表信息

(parted) mkpart

分区名称？ []? nsd

文件系统类型？ [ext2]? ext4

起始点？ 0

结束点？ 10G

警告: The resulting partition is not properly aligned for best performance.

忽略/Ignore/放弃/Cancel? i #选择忽略

(parted) print

......

Number Start End Size File system Name 标志

1 17.4kB 10.0GB 10000MB nsd

(parted) mkpart nsd ext4 10G 20G

(parted) mkpart nsd ext4 20G 30G

(parted) mkpart nsd ext4 30G 40G

(parted) mkpart nsd ext4 40G 52G

(parted) print

Number Start End Size File system Name 标志

1 17.4kB 10.0GB 10000MB nsd

2 10.0GB 20.0GB 9999MB nsd

3 20.0GB 30.0GB 10.0GB nsd

4 30.0GB 40.0GB 10.0GB nsd

5 40.0GB 52.0GB 12.0GB nsd

**案例41：虚拟机 server0上操作,交换分区练习**

利用案例2划分的分区，制作10G的交换分区，此交换分区开机自启动

1.格式化交换文件系统

[root@server0 ~]# mkswap /dev/vdc1 #格式化交换文件系统

[root@server0 ~]# blkid /dev/vdc1 #查看文件系统类型

2.启用交换分区

[root@server0 ~]# swapon /dev/vdc1 #启用交换分区

[root@server0 ~]# swapon -s #查看交换空间信息

3.停用交换分区

[root@server0 ~]# swapoff /dev/vdc1

[root@server0 ~]# swapon -s

4.开机自动启用交换分区

[root@server0 ~]# vim /etc/fstab

/dev/vdc1 swap swap defaults 0 0

[root@server0 ~]# swapon -s

[root@server0 ~]# swapon -a #专用于检测swap分区开机自动启动

[root@server0 ~]# swapon -s

## 三、系统&服务管理进阶

**案例01：补充命令使用**

1. 采用数值形式将目录/root的权限调整为 rwx------

1.查看原来的权限

[root@svr7 ~]# ls -ld /root/

dr-xr-x---. 22 root root 4096 3月 26 14:59 /root/

2.修改为新权限

[root@svr7 ~]# chmod 700 /root/

3.确认权限设置结果

[root@svr7 ~]# ls -ld /root/

drwx------. 22 root root 4096 3月 26 14:59 /root/

1. 将记录的历史命令条数更改为 200 条

1.调整记录条数

修改配置文件/etc/profile，找到HISTSIZE行，将此变量的值修改为200：

[root@svr7 ~]# vim /etc/profile

.. ..

HISTSIZE = 200

2.确认设置结果

所有用户重新登录以后即可生效：

[root@svr7 ~]# su - root

[root@svr7 ~]# echo $HISTSIZE

200

1. 统计 /boot、/etc/pki 目录占用的空间大小

1.分别统计结果

[root@svr7 ~]# du -sh /boot/ /etc/pki/

130M /boot/

1.5M /etc/pki/

2.比较du与ls查看文件大小的差异（默认块大小4096字节）：

[root@svr7 ~]# ls -lh /etc/inittab //数据大小511字节

-rw-r--r--. 1 root root 511 Sep 16 2015 /etc/inittab

[root@svr7 ~]# du -sh /etc/inittab //实际占用4KB磁盘空间

4.0K /etc/inittab

1. 以格式“yyyy-mm-dd HH:MM” 显示系统时间

[root@svr7 ~]# date +"%F %R"

2026-12-26 16:23

**案例02：软连接与硬连接**

1. 新建文件 file1，内容为 linux.tedu.cn
2. 为 file1 建立软连接 file1-s，对比两文件内容
3. 为 file1 建立硬连接 file1-h，对比两文件内容
4. 删除文件 file1 ，再查看文件 file1-s、file1-h 内容

1.新建一个测试文件

[root@svr7 ~]# vim /root/file1

AAAA

2.为文件file1建立软连接file1-s并测试

[root@svr7 ~]# ln -s /root/file1 /root/file1-s

[root@svr7 ~]# cat /root/file1-s

linux.tedu.cn

3.为文件file1建立硬连接file1-h并测试

[root@svr7 ~]# ln /root/file1 /root/file1-h

[root@svr7 ~]# cat /root/file1-h

linux.tedu.cn

4.对比原始文件、软连接、硬连接的属性

[root@svr7 ~]# ls -l /root/f\*

-rw-r--r--. 2 root root 14 Jan 6 12:14 file1-h

lrwxrwxrwx. 1 root root 12 Jan 6 12:16 file1-s -> file1

-rw-r--r--. 2 root root 14 Jan 6 12:14 file1

步骤二：原始文件删除测试

1.当原始文件被删除时，软连接将会失效，而硬连接仍然可访问文件数据

[root@svr7 ~]# rm -rf file1

[root@svr7 ~]# cat /root/file1-s

cat: file1-s: No such file or directory

[root@svr7 ~]# cat /root/file1-h

linux.tedu.cn

2.不支持为目录创建硬连接，但可以为目录建立软连接

[root@svr7 ~]# ln /etc/sysconfig/network-scripts/ /etc/network

ln: '/etc/sysconfig/network-scripts/': hard link not allowed for directory

[root@svr7 ~]# ln -s /etc/sysconfig/network-scripts/ /etc/interface

[root@svr7 ~]# ls -l /etc/interface

lrwxrwxrwx. 1 root root 31 Jan 6 12:28 /etc/interface -> /etc/sysconfig/network-scripts/

**案例03：man手册、zip备份**

1. 查阅passwd命令、/etc/passwd配置文件的手册页

1.查看passwd命令的手册页

[root@svr7 ~]# man passwd

PASSWD(1) User utilities PASSWD(1)

NAME

passwd - update user's authentication tokens

SYNOPSIS

passwd [-k] [-l] [-u [-f]] [-d] [-e] [-n mindays] [-x maxdays] [-w

warndays] [-i inactivedays] [-S] [--stdin] [username]

DESCRIPTION

The passwd utility is used to update user's authentication token(s).

This task is achieved through calls to the Linux-PAM and Libuser API.

Essentially, it initializes itself as a "passwd" service with Linux-

PAM and utilizes configured password modules to authenticate and then

update a user's password.

.. ..

2.查看/etc/passwd配置文件的手册页

[root@svr7 ~]# man 5 passwd

PASSWD(5) Linux Programmer's Manual PASSWD(5)

NAME

passwd - password file

DESCRIPTION

The /etc/passwd file is a text file that describes user login

accounts for the system. It should have read permission allowed for

all users (many utilities, like ls(1) use it to map user IDs to user‐

names), but write access only for the superuser.

In the good old days there was no great problem with this general

read permission. Everybody could read the encrypted passwords, but

the hardware was too slow to crack a well-chosen password, and more‐

over the basic assumption used to be that of a friendly user-commu‐

nity. These days many people run some version of the shadow password

suite, where /etc/passwd has an 'x' character in the password

.. ..

1. 使用zip打包/usr/share/doc/qemu-kvm/目录

1.将目录/usr/share/doc/qemu-kvm/备份为/root/qemu-kvm.zip

[root@svr7 ~]# zip -r /root/qemu-kvm.zip /usr/share/doc/qemu-kvm/

adding: usr/share/doc/qemu-kvm/ (stored 0%)

adding: usr/share/doc/qemu-kvm/COPYING (deflated 62%)

adding: usr/share/doc/qemu-kvm/COPYING.LIB (deflated 65%)

adding: usr/share/doc/qemu-kvm/Changelog (deflated 61%)

adding: usr/share/doc/qemu-kvm/LICENSE (deflated 45%)

adding: usr/share/doc/qemu-kvm/README (deflated 4%)

[root@svr7 ~]# ls /root

**案例04：虚拟机svr7中，自定义yum软件仓库**

1. 虚拟机svr7创建目录 /myrpm
2. tools.tar.gz中的RPM包，复制到/myrpm目录下
3. 利用/myrpm目录的RPM包进行搭建自定义Yum仓库

1. 真机的数据传递到虚拟机svr7中

[root@room9pc01 ~]# scp /root/桌面/tools.tar.gz root@192.168.4.7:/root/

2.在虚拟机svr7中验证

[root@svr7 ~]# ls /root

3.进行tar解包

[root@svr7 ~]# mkdir /myrpm

[root@svr7 ~]# tar -xf /root/tools.tar.gz -C /myrpm

[root@svr7 ~]# ls /myrpm/tools/

[root@svr7 ~]# ls /myrpm/tools/other/

4.生成仓库数据文件

[root@svr7 ~]# createrepo /myrpm/tools/other/

[root@svr7 ~]# ls /myrpm/tools/other/

5.书写yum客户端配置文件

[root@svr7 ~]# vim /etc/yum.repos.d/rhel7.repo

.......

[myrpm]

name=myrpm

baseurl=file:///myrpm/tools/other/ #指定本地路径

enabled=1

gpgcheck=0

[root@svr7 ~]# yum repolist

**案例05：vim效率操作**

1. 将文件 /etc/passwd 复制为 /opt/nsd.txt，然后打开 /opt/nsd.txt 文件，练习命令模式下的切换/复制/删除/查找操作

步骤一：vim命令模式下的切换/复制/删除/查找

1.建立练习文件

将文件 /etc/passwd 复制为 /opt/nsd.txt：

[root@svr7 ~]# cp /etc/passwd /opt/nsd.txt

2.使用vim打开练习文件，默认处于命令模式

[root@svr7 ~]# vim /opt/nsd.txt

.. ..

3.在命令模式下完成下列操作

切换操作：G 最后一行，5G 第5行，gg 第一行。

复制操作：按2yy复制2行，7G移动到第7行，p 粘贴。

删除操作：25G 移动到第25行，200dd 从此行开始删除200行（不够就剩下全删）。

查找操作：gg 第一行，/adm 查找关键词adm，n 跳转到下一个结果。

4.保存并退出编辑器

ZZ 保存退出。

1. 将文件 /etc/man\_db.conf 复制到 /opt 目录下，然后打开 /opt/man\_db.conf 文件，将第50~100行内的“man”替换为“MAN”，在 vim 中设置显示行号查看效果

步骤二：vim末行模式下的替换/设置操作

1.建立练习文件

将文件 /etc/man\_db.conf 复制到 /opt/ 目录下：

[root@svr7 ~]# cp /etc/man\_db.conf /opt/

2.使用vim打开练习文件，输入:切换到末行模式

[root@svr7 ~]# vim /opt/man\_db.conf

.. ..

:

3.在末行模式下完成下列操作

输入 :set nu ，确认后显示行号。

输入 :50,100 s/man/MAN/g ，确认将第50~100行内的“man”替换为“MAN”。

4.保存并退出编辑器

输入 :wq ，确认后保存并退出编辑器。

**案例06：搭建单区域DNS服务器**

1. 在虚拟机svr7上，搭建DNS服务器，可以解析以下域名

pc207.tedu.cn ---> 192.168.4.207

www.tedu.cn ---> 192.168.4.100

1.安装软件包

[root@svr7 ~]# yum -y install bind-chroot bind

bind-9.9.4-29.el7.x86\_64 //域名服务包

bind-chroot-9.9.4-29.el7.x86\_64 //提供虚拟根支持

2.修改配置文件/etc/named.conf

[root@svr7 ~]# cp /etc/named.conf /root/named.bak

[root@svr7 ~]# vim /etc/named.conf

options {

directory "/var/named"; #指定地址库文件路径

};

zone "tedu.cn" IN { #指定本机负责解析的域名

type master; #本机为权威主DNS服务器

file "tedu.cn.zone"; #指定地址库文件名字

};

3.创建地址库文件/var/named/tedu.cn.zone

[root@svr7 ~]# cd /var/named

[root@svr7 ~]# cp -p named.localhost tedu.cn.zone #保持属性不变

[root@svr7 ~]# ls -l tedu.cn.zone

[root@svr7 ~]# vim tedu.cn.zone #没有以点结尾,默认补全tedu.cn.

………

tedu.cn. NS svr7 #tedu.cn.区域有svr7负责

svr7 A 192.168.4.7 #svr7的IP地址为192.168.4.7

www A 192.168.4.100

pc207 A 192.168.4.207

4.重起named服务

[root@svr7 ~]# systemctl restart named

客户端验证:

1.指定DNS服务器位置

[root@pc207 ~]# echo nameserver 192.168.4.7 > /etc/resolv.conf

2.域名解析测试

[root@pc207 ~]# nslookup www.tedu.cn

[root@pc207 ~]# nslookup pc207.tedu.cn

**案例07：特殊DNS解析**

1. 为站点 www.tedu.cn 提供DNS轮询解析，三台Web服务器节点的IP地址分别为：192.168.4.100、192.168.4.110、192.168.4.120

1.修改DNS服务器上tedu.cn区域的地址库文件，在末尾添加轮询地址记录

[root@svr7 ~]# vim /var/named/tedu.cn.zone

.. ..

www A 192.168.4.100

www A 192.168.4.110

www A 192.168.4.120

2.重启系统服务named

[root@svr7 named]# systemctl restart named

3.在客户机pc207上测试轮询记录

针对目标www.tedu.cn执行多次查询，观察第1条结果的变化：

[root@pc207 ~]# host www.tedu.cn

www.tedu.cn has address 192.168.4.100 //第1个结果为192.168.4.100

www.tedu.cn has address 192.168.4.110

www.tedu.cn has address 192.168.4.120

[root@pc207 ~]# host www.tedu.cn

www.tedu.cn has address 192.168.4.120 //第1个结果为192.168.4.120

www.tedu.cn has address 192.168.4.110

www.tedu.cn has address 192.168.4.100

[root@pc207 ~]# host www.tedu.cn

www.tedu.cn has address 192.168.4.110 //第1个结果为192.168.4.110

www.tedu.cn has address 192.168.4.120

www.tedu.cn has address 192.168.4.100

1. 配置泛域名解析实现以下解析记录：任意名称.tedu.cn ---> 119.75.217.56

1.修改DNS服务器上指定区域的地址库文件，在末尾添加\*通配地址记录

[root@svr7 ~]# vim /var/named/tedu.cn.zone

.. ..

\* A 119.75.217.56

2.重启系统服务named

[root@svr7 named]# systemctl restart named

3.在客户机pc207上测试多对一的泛域名解析记录

当查询未知站点（地址库中没有明确记录）时，以 \* 对应的IP地址反馈：

[root@pc207 ~]# host station123.tedu.cn

station123.tedu.cn has address 119.75.217.56

[root@pc207 ~]# host movie.tedu.cn

movie.tedu.cn has address 119.75.217.56

[root@pc207 ~]# host tts8.tedu.cn

tts8.tedu.cn has address 119.75.217.56

1. 利用解析记录的别名，实现tts.tedu.cn记录解析结果与pc207.tedu.cn结果一致

1.修改DNS服务器上指定区域的地址库文件

[root@svr7 ~]# vim /var/named/tedu.cn.zone

.. ..

tts CNAME pc207

2.重启系统服务named

[root@svr7 named]# systemctl restart named

3.在客户机pc207上测试

[root@pc207 ~]# nslookup tts.tedu.cn

**案例08：配置DNS子域授权**

1. 构建父DNS（tedu.cn）服务器
2. 构建子DNS（bj.tedu.cn）服务器
3. 在父DNS上配置子域授权
4. 测试子域授权查询

虚拟机pc207

1.安装软件包

[root@pc207 ~]# yum -y install bind-chroot bind

2.修改配置文件/etc/named.conf

[root@pc207 ~]# cp /etc/named.conf /root/named.bak

[root@pc207 ~]# vim /etc/named.conf

options {

directory "/var/named"; #指定地址库文件路径

};

zone "bj.qq.com" IN { #指定本机负责解析的域名

type master; #本机为权威主DNS服务器

file "bj.qq.com.zone"; #指定地址库文件名字

};

3.创建地址库文件/var/named/bj.qq.com.zone

[root@pc207 ~]# cd /var/named

[root@pc207 ~]# cp –p named.localhost bj.qq.com.zone

[root@pc207 ~]# vim bj.qq.com.zone

bj.qq.com. NS pc207

pc207 A 192.168.4.207

www A 5.5.5.5

4.重起named服务

[root@pc207 ~]# systemctl restart named

子域授权:通过父域的DNS服务器为虚拟机svr7,能够解析www.bj.qq.com

虚拟机svr7:

[root@svr7 /]# vim /var/named/qq.com.zone

qq.com. NS svr7

bj.qq.com. NS pc207

svr7 A 192.168.4.7

pc207 A 192.168.4.207

………

[root@svr7 /]# systemctl restart named

[root@svr7 /]# nslookup www.bj.qq.com 192.168.4.7

Server: 192.168.4.7

Address: 192.168.4.7#53

Non-authoritative answer: #非权威解答

Name: www.bj.qq.com

Address: 5.5.5.5

**案例09：搭建并测试缓存DNS**

利用真机安装 bind、bind-chroot 包，搭建并测试基于全局转发器的缓存DNS

真机搭建缓存DNS服务器

1.搭建Yum仓库

[root@room9pc01 ~]# cat /etc/redhat-release

CentOS Linux release 7.4.1708 (Core)

[root@room9pc01 ~]# ls /iso/

CentOS-7-x86\_64-DVD-1708.iso

[root@room9pc01 ~]# mkdir /dvd/

[root@room9pc01 ~]# mount /iso/CentOS-7-x86\_64-DVD-1708.iso /dvd/

[root@room9pc01 ~]# ls /dvd/

[root@room9pc01 ~]# cd /etc/yum.repos.d/

[root@room9pc01 ~]# mkdir repo

[root@room9pc01 ~]# mv \*.repo repo

[root@room9pc01 ~]# vim dvd.repo

[dvd]

name=CentOS7.4

baseurl=file:///dvd #指定本地Yum

enabled=1

gpgcheck=0

[root@room9pc01 ~]# yum -y install bind bind-chroot

2.查看达内的DNS服务器地址

[root@room9pc01 /]# cat /etc/resolv.conf

nameserver 172.40.1.10

[root@room9pc01 /]#

3.修改DNS服务器主配置文件，指定转发给内网DNS服务器

[root@room9pc01 /]# vim /etc/named.conf

options {

directory "/var/named";

forwarders { 172.40.1.10; };

};

4. 重起named服务

[root@room9pc01 /]# systemctl restart named

5.在虚拟机上验证：

[root@svr7 /]# nslookup www.taobao.com 192.168.4.254

**案例10：配置并验证Split分离解析**

从主机192.168.4.207查询时，结果为：www.tedu.cn ---> 192.168.4.100

从其他客户端查询时，www.tedu.cn ---> 1.2.3.4

步骤一：配置Split分离解析

1．为tedu.cn区域建立两份解析记录文件

第一份解析记录文件提供给客户机192.168.4.207，对应目标域名www.tedu.cn的A记录地址为192.168.4.100。相关操作及配置如下：

[root@svr7 ~]# cd /var/named/

[root@svr7 named]# cp -p tedu.cn.zone tedu.cn.zone.lan

[root@svr7 named]# vim tedu.cn.zone.lan

$TTL 1D

@ IN SOA @ rname.invalid. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

tedu.cn. NS svr7.tedu.cn.

svr7 A 192.168.4.7

pc207 A 192.168.4.207

www A 192.168.4.100

第二份解析记录文件提供给其他客户机，对应目标域名www.tedu.cn的A记录地址为1.2.3.4。相关操作及配置如下：

[root@svr7 named]# cp -p tedu.cn.zone tedu.cn.zone.other

[root@svr7 named]# vim tedu.cn.zone.other

$TTL 1D

@ IN SOA @ rname.invalid. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

tedu.cn. NS svr7.tedu.cn.

svr7 A 192.168.4.7

pc207 A 192.168.4.207

www A 1.2.3.4

2）修改named.conf配置文件，定义两个view，分别调用不同解析记录文件

[root@svr7 ~]# vim /etc/named.conf

options {

directory "/var/named";

};

acl "mylan" { //名为mylan的列表

192.168.4.207;

};

view "mylan" {

match-clients { mylan; }; //检查客户机地址是否匹配此列表

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone.lan";

};

};

view "other" {

match-clients { any; }; //匹配任意客户机地址

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone.other";

};

};

3）重启named服务

[root@svr7 ~]# systemctl restart named

步骤二：测试分离解析效果

1）从mylan地址列表中的客户机查询

在客户机192.168.4.207上查询www.tedu.cn，结果是 192.168.4.100：

[root@pc207 ~]# host www.tedu.cn 192.168.4.7

Using domain server:

Name: 192.168.4.7

Address: 192.168.4.7#53

Aliases:

www.tedu.cn has address 192.168.4.100

2）从其他客户机查询

在DNS服务器本机或CentOS真机上查询www.tedu.cn时，结果为 1.2.3.4：

[root@svr7 ~]# host www.tedu.cn 192.168.4.7

Using domain server:

Name: 192.168.4.7

Address: 192.168.4.7#53

Aliases:

www.tedu.cn has address 1.2.3.4

**案例11：查看进程信息**

1. 找出进程 rsyslogd 的 PID 编号值

[root@svr7 ~]# pgrep -l rsyslogd

718 rsyslogd

[root@svr7 ~]#

1. 列出由进程 systemd 开始的子进程树结构信息

[root@svr7 ~]# pstree -ap 1

systemd,1 --switched-root --system --deserialize 21

├─ModemManager,704

│ ├─{ModemManager},710

│ └─{ModemManager},715

├─NetworkManager,697 --no-daemon

│ ├─{NetworkManager},742

│ └─{NetworkManager},745

├─abrt-dbus,31118 -t133

│ ├─{abrt-dbus},31119

…………

1. 找出进程 sshd 的父进程的 PID 编号/进程名称

[root@svr7 ~]# ps -elf | grep sshd #父进程PPID为1 所以该进程为systemd

4 S root 912 1 0 80 0 - 26499 poll\_s 12月18 ? 00:00:00 /usr/sbin/sshd –D

………

1. 查看当前系统的CPU负载/进程总量信息

[root@svr7 ~]# top -d 1

top - 16:39:40 up 4 days, 7:51, 2 users, load average: 0.00, 0.01, 0.05

……………

**案例12：进程调度及终止**

1. 运行“sleep 600”命令，再另开一个终端，查出sleep程序的PID并杀死
2. 运行多个vim程序并都放入后台，然后杀死所有vim进程
3. su切换为zhsan用户，再另开一个终端，强制踢出zhsan用户

步骤一：根据PID杀死进程

1.开启sleep测试进程

[root@svr7 ~]# sleep 600

//.. .. 进入600秒等待状态

2.找出进程sleep的PID

另开一个终端，ps aux并过滤进程信息（第2列为PID值）：

[root@svr7 ~]# ps aux | grep sleep

root 32929 0.0 0.0 4312 360 pts/1 S+ 17:25 0:00 sleep 600

3.杀死指定PID的进程

[root@svr7 ~]# kill -9 32929

返回原终端会发现sleep进程已经被杀死：

[root@svr7 ~]# sleep 600

Killed

步骤二：根据进程名杀死多个进程

1.在后台开启多个vim进程

[root@svr7 ~]# vim a.txt &

[1] 33152

[root@svr7 ~]# vim b.txt &

[2] 33154

[1]+ 已停止 vim a.txt

[root@svr7 ~]# vim c.txt &

[3] 33155

[2]+ 已停止 vim b.txt

2.确认vim进程信息

[root@svr7 ~]# jobs -l

[1] 33152 停止 (tty 输出) vim a.txt

[2]- 33154 停止 (tty 输出) vim b.txt

[3]+ 33155 停止 (tty 输出) vim c.txt

3.强制杀死所有名为vim的进程

[root@svr7 ~]# killall -9 vim

[1] 已杀死 vim a.txt

[2]- 已杀死 vim b.txt

[3]+ 已杀死 vim c.txt

4.确认杀进程结果

[root@svr7 ~]# jobs -l

[root@svr7 ~]#

步骤三：杀死属于指定用户的所有进程

1.登入测试用户zhsan

[root@svr7 ~]# useradd zhsan

[root@svr7 ~]# su - zhsan

[zhsan@svr7 ~]$

2.另开一个终端，以root用户登入，查找属于用户zhsan的进程

[root@svr7 ~]# pgrep -u zhsan

33219

[root@svr7 ~]# pstree -up 33219 //检查进程树

bash(33219,zhsan)

3.强制杀死属于用户zhsan的进程

[root@svr7 ~]# killall -9 -u zhsan

[root@svr7 ~]#

4.返回原来用户zhsan登录的终端，确认已经被终止

[zhsan@svr7 ~]$ 已杀死

[root@svr7 ~]#

**案例13：系统日志分析**

1. 列出所有与服务httpd相关的消息

[root@svr7 ~]# journalctl -u httpd

-- Logs begin at Thu 2017-01-05 15:50:08 CST, end at Fri 2017-01-06 18:01:01 CST. --

Jan 06 14:57:16 svr7 systemd[1]: Starting The Apache HTTP Server...

Jan 06 14:57:16 svr7 httpd[23812]: AH00557: httpd: apr\_sockaddr\_info\_get() failed for svr7

Jan 06 14:57:16 svr7 httpd[23812]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directi

Jan 06 14:57:16 svr7 systemd[1]: Started The Apache HTTP Server.

Jan 06 17:53:44 svr7 systemd[1]: Stopping The Apache HTTP Server...

1. 列出最近2条成功/不成功的用户登录消息

查看成功登录的事件消息：

[root@svr7 ~]# last -2

zhsan pts/2 192.168.4.207 Fri Jan 6 18:00 - 18:00 (00:00)

root pts/2 192.168.4.110 Fri Jan 6 17:26 - 17:59 (00:33)

wtmp begins Thu Aug 4 00:10:16 2016

查看失败登录的事件消息：

[root@svr7 ~]# lastb -2

anonymou ssh:notty 192.168.4.207 Fri Jan 6 18:00 - 18:00 (00:00)

anonymou ssh:notty 192.168.4.207 Fri Jan 6 18:00 - 18:00 (00:00)

btmp begins Fri Jan 6 18:00:34 2017

**案例14：使用systemctl工具**

1. 重启 httpd、crond服务，查看状态
2. 禁止 httpd 服务开机自启，并停用此服务
3. 设置默认级别为 multi-user.target 并确认

1.重启系统服务httpd、crond、bluetooth

[root@svr7 ~]# systemctl restart httpd crond

2.查看上述服务的状态

[root@svr7 ~]# systemctl status httpd crond

\* httpd.service - The Apache HTTP Server

Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)

Active: active (running) since Fri 2017-01-06 18:18:20 CST; 18s ago

.. ..

\* crond.service - Command Scheduler

Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)

Active: active (running) since Fri 2017-01-06 18:18:19 CST; 19s ago

.. ..

步骤二：禁止 httpd 服务开机自启，并停用此服务

1）停用httpd服务

[root@svr7 ~]# systemctl stop httpd

2）禁止httpd服务开机自启

[root@svr7 ~]# systemctl disable httpd

[root@svr7 ~]# systemctl is-enabled httpd //检查结果

disabled

步骤三：设置默认级别为 multi-user.target 并确认

1）查看默认运行级别

[root@svr7 ~]# systemctl get-default

graphical.target

2）将默认运行级别设置为multi-user.target

[root@svr7 ~]# systemctl set-default multi-user.target

Removed symlink /etc/systemd/system/default.target.

Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/multi-user.target.

3）确认配置结果

[root@svr7 ~]# systemctl get-default

multi-user.target

根据此处的设置，重启此虚拟机后图形桌面将不再可用。

**案例15：PXE基础装机环境**

1. 搭建dhcp、tftp、httpd服务实现PXE网络装机服务器
2. 部署应答文件实现无人值守安装RHEL7操作系统

一、搭建配置DHCP服务

1.指定下一个服务器地址及引导文件

[root@svr7 /]# vim /etc/dhcp/dhcpd.conf

subnet 192.168.4.0 netmask 255.255.255.0 {

range 192.168.4.20 192.168.4.100;

option domain-name-servers 192.168.4.7;

option routers 192.168.4.254;

default-lease-time 600;

max-lease-time 7200;

next-server 192.168.4.7; #指定下一个服务器地址

filename "pxelinux.0"; #指定引导文件名称

}

2.重起服务

[root@svr7 /]# systemctl restart dhcpd

pxelinux.0:网卡引导文件，安装说明书，二进制文件

安装一个软件，会自动生成该文件

二、配置TFTP服务

tftp：简单的文件传输协议

默认端口为： 69

默认共享路径：/var/lib/tftpboot

1.安装tftp-server软件

[root@svr7 /]# yum -y install tftp-server

2.重起tftp服务

[root@svr7 /]# systemctl restart tftp

[root@svr7 /]# systemctl enable tftp

3.部署pxelinux.0文件

[root@svr7 /]# yum provides \*/pxelinux.0 #查询那个软件包产生该文件

[root@svr7 /]# yum -y install syslinux #安装软件

[root@svr7 /]# rpm -ql syslinux | grep pxelinux.0 #查看安装清单

[root@svr7 /]# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/

[root@svr7 /]# ls /var/lib/tftpboot/

4.部署菜单文件

默认存放路径/var/lib/tftpboot/pxelinux.cfg/

pxelinux.0--》/var/lib/tftpboot/pxelinux.cfg/default

[root@svr7 /]# mkdir /var/lib/tftpboot/pxelinux.cfg

[root@svr7 /]# mount /dev/cdrom /mnt

[root@svr7 /]# ls /mnt/

[root@svr7 /]# cp /mnt/isolinux/isolinux.cfg /var/lib/tftpboot/pxelinux.cfg/default

[root@svr7 /]# ls -l /var/lib/tftpboot/pxelinux.cfg/default

[root@svr7 /]# chmod u+w /var/lib/tftpboot/pxelinux.cfg/default

[root@svr7 /]# ls -l /var/lib/tftpboot/pxelinux.cfg/default

5.部署图形的模块(vesamenu.c32)与背景图片(splash.png)

[root@svr7 /]# cp /mnt/isolinux/vesamenu.c32 /mnt/isolinux/splash.png /var/lib/tftpboot/

[root@svr7 /]# ls /var/lib/tftpboot/

pxelinux.0 pxelinux.cfg splash.png vesamenu.c32

6.部署 启动内核（vmlinuz） 与 驱动程序（initrd.img）

[root@svr7 /]# cp /mnt/isolinux/vmlinuz /mnt/isolinux/initrd.img /var/lib/tftpboot/

[root@svr7 /]# ls /var/lib/tftpboot/

initrd.img pxelinux.cfg vesamenu.c32

pxelinux.0 splash.png vmlinuz

[root@svr7 /]#

7.修改菜单文件/var/lib/tftpboot/pxelinux.cfg/default

[root@svr7 /]# vim /var/lib/tftpboot/pxelinux.cfg/default

1 default vesamenu.c32 #默认加载图形的模块

2 timeout 60 #读秒时间为6秒

9 menu background splash.png #背景图片

10 menu title NSD1809 PXE Server !!! #标题

60 label linux

61 menu label Install RHEL7 #选项的内容

menu default #读秒结束后默认选择

62 kernel vmlinuz #指定启动内核

63 append initrd=initrd.img #指定启动的驱动程序

################################################

总结：

1.DHCP--->IP地址、next-server、filename

2.tftp--->pxelinux.0

3.pxelinux.0--->default

4.default---》

vesamenu.c32/splash.png/vmlinuz/initrd.img

测试：新建一台虚拟机，采用PXE网络引导安装

注意网络类型选择为：private1

##################################################

三、搭建httpd服务，共享光盘所有内容

1.安装软件包

[root@svr7 /]# yum -y install httpd

2.重起服务

[root@svr7 /]# systemctl restart httpd

3.共享光盘所有内容

[root@svr7 /]# mkdir /var/www/html/rhel7/

[root@svr7 /]# mount /dev/cdrom /var/www/html/rhel7/

mount: /dev/sr0 写保护，将以只读方式挂载

[root@svr7 /]# ls /var/www/html/rhel7/

4.访问测试

[root@svr7 /]# firefox 192.168.4.7/rhel7

###############################################

四、部署无人值守安装，生成应答文件

1.通过图形的system-config-kickstart软件可以生成

[root@svr7 /]# yum -y install system-config-kickstart

2.运行图形的system-config-kickstart进行图形配置

[root@room9pc01 ~]# ssh -X root@192.168.4.7

root@192.168.4.7's password:

[root@svr7 ~]# system-config-kickstart

首先检测 “软件包选择” 是否可以进行选择

需要光盘的源的Yum仓库支持, 要求 仓库标识必须为 [development]

[root@svr7 ~]# vim /etc/yum.repos.d/rhel7.repo

[development]

.......

[root@svr7 ~]# system-config-kickstart

首先检测 “软件包选择” 是否可以进行选择

#################################################

五、应答文件

[root@svr7 ~]# ls /root/ks.cfg

/root/ks.cfg

[root@svr7 ~]# vim /root/ks.cfg

利用httpd服务，将应答文件共享

[root@svr7 ~]# cp /root/ks.cfg /var/www/html/

[root@svr7 ~]# ls /var/www/html/

ks.cfg rhel7

[root@svr7 ~]# firefox 192.168.4.7/ks.cfg

修改菜单文件，指定应答文件位置

[root@svr7 /]# vim /var/lib/tftpboot/pxelinux.cfg/default

label linux

menu label Install RHEL7

menu default

kernel vmlinuz

append initrd=initrd.img ks=http://192.168.4.7/ks.cfg

##################################################

总结：

1.DHCP--->IP地址、next-server、filename

2.tftp--->pxelinux.0

3.pxelinux.0--->default

4.default---》

vesamenu.c32/splash.png/vmlinuz/initrd.img

ks=http://192.168.4.7/ks.cfg

5.ks.cfg----》--url="http://192.168.4.7/rhel7"

################################################

**案例16：rsync基本用法**

1. 将目录 /boot 同步到目录 /todir 下
2. 将目录 /boot 下的文档同步到目录 /todir 下
3. 在目录 /boot 下新增文件 a.txt，删除 /todir 下的子目录 grub2，再次同步使 /todir 与 /boot 一致

1．将目录 /boot 同步到目录 /todir 下

[root@svr7 ~]# ls -l /todir //同步前

ls: 无法访问/todir: 没有那个文件或目录

[root@svr7 ~]# rsync -a /boot /todir //将目录1作为目录2的子目录

[root@svr7 ~]# ls -l /todir //检查同步结果

总用量 4

dr-xr-xr-x. 4 root root 4096 11月 30 18:50 boot

2．将目录 /boot 下的文档同步到目录 /todir 下

[root@svr7 ~]# rm -rf /todir //清理掉目录2

[root@svr7 ~]# rsync -a /boot/ /todir //将目录1下的文档同步到目录2下

[root@svr7 ~]# ls -l /todir //检查同步结果

总用量 126708

-rw-r--r--. 1 root root 126426 10月 30 2015 config-3.10.0-327.el7.x86\_64

drwxr-xr-x. 2 root root 4096 11月 30 18:50 extlinux

drwx------. 6 root root 104 12月 9 09:58 grub2

.. ..

3．同步效果测试

在目录/boot下新增文件a.txt，删除/todir下的子目录 grub2：

[root@svr7 ~]# touch /boot/a.txt

[root@svr7 ~]# rm -rf /todir/grub2/

现在目录/boot和/todir目录下的内容已经不一致了：

[root@svr7 ~]# ls -ld /boot/a.txt /todir/a.txt

ls: 无法访问/todir/a.txt: 没有那个文件或目录

-rw-r--r--. 1 root root 0 1月 11 21:09 /boot/a.txt

[root@svr7 ~]# ls -ld /boot/grub2 /todir/grub2

ls: 无法访问/todir/grub2: 没有那个文件或目录

drwx------. 6 root root 104 12月 9 09:58 /boot/grub2

再次同步使/todir与/boot一致：

[root@svr7 ~]# rsync -a /boot/ /todir/

确认同步结果：

[root@svr7 ~]# ls -ld /boot/a.txt /todir/a.txt

-rw-r--r--. 1 root root 0 1月 11 21:09 /boot/a.txt

-rw-r--r--. 1 root root 0 1月 11 21:09 /todir/a.txt

[root@svr7 ~]# ls -ld /boot/grub2 /todir/grub2

drwx------. 6 root root 104 12月 9 09:58 /boot/grub2

drwx------. 6 root root 104 12月 9 09:58 /todir/grub2

**案例17：rsync+SSH实时同步**

1. 实现本机/opt目录与远程主机/opt目录，使用inotifywait工具进行实时同步

实时同步

一、部署公钥 私钥，实现远程ssh无密码验证

虚拟机svr7：

1.生成公钥 私钥

[root@svr7 ~]# ssh-keygen #一路回车

[root@svr7 ~]# ls /root/.ssh/

id\_rsa id\_rsa.pub known\_hosts

2.传递公钥 到虚拟机B

[root@svr7 ~]# ssh-copy-id root@192.168.4.207

3.虚拟机pc207：查看

[root@pc207 ~]# ls /root/.ssh/

authorized\_keys

4.虚拟机svr7：测试无密码同步

[root@svr7 ~]# rsync --delete -avz /opt/ root@192.168.4.207:/opt/

二、虚拟机svr7 监控目录内容的变化

• Linux内核的 inotify 机制

– 提供事件响应式的文件系统通知机制

– 安装 inotify-tools 控制工具可调用此机制实现监控

1.真机传递tools.tar.gz到虚拟机svr7

[root@svr7 ~]# scp /root/桌面/tools.tar.gz root@192.168.4.7:/root

2.虚拟机A验证：

[root@svr7 ~]# ls /root/

tools.tar.gz

[root@svr7 ~]#

3.安装源码包

[root@svr7 ~]# mkdir /nsd

[root@svr7 ~]# tar -xf /root/tools.tar.gz -C /nsd

[root@svr7 ~]# tar -xf /nsd/tools/inotify-tools-3.13.tar.gz -C /root/

步骤1:安装gcc与make

[root@svr7 ~]# yum –y install gcc make

步骤2: ./configure 配置,指定安装目录/功能模块等选项

[root@svr7 ~]# cd /root/inotify-tools-3.13/

[root@svr7 ~] # ./configure

步骤4:make 编译,生成可执行的二进制程序文件

[root@svr7 ~]# make

步骤5:make install 安装,将编译好的文件复制到安装目录

[root@svr7 ~]# make install

[root@svr7 ~]# ls /usr/local/bin/inotifywait #验证程序产生

###############################################

三、书写Shell脚本完成，实时同步

死循环： while

while [条件]

do

重复执行的代码

done

[root@svr7 /]# vim /root/rsync.sh

#!/bin/bash

while inotifywait -rqq /opt

do

rsync --delete -az /opt/ root@192.168.4.207:/opt/

done &

[root@svr7 /]#

################################################

**案例18：配置Cobbler装机平台**

1. 虚拟机名设置为Cobbler
2. IP地址设置为：192.168.4.123/24
3. 关闭SELinux安全机制
4. 设置防火墙默认区域为trusted

1. 在虚拟机中解压cobbler.zip包

此cobbler.zip包，起初应该在真机上，可以通过scp命令传递到虚拟机。

1）将真机cobbler.zip包传递到虚拟机中/root目录下，注意真机cobbler.zip绝对路径

[root@room9pc01 桌面]# scp -r /root/桌面/Cobbler/ root@192.168.4.123:/root/

The authenticity of host '192.168.4.123 (192.168.4.123)' can't be established.

ECDSA key fingerprint is SHA256:TFNqSD+oJMsA88kLwSdLSKZhSigkQIIAfrXLdKoUaJA.

ECDSA key fingerprint is MD5:f9:b8:7d:8d:ca:4e:20:0d:10:c4:72:a5:9f:42:28:8e.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '192.168.4.123' (ECDSA) to the list of known hosts.

root@192.168.4.123's password:

cobbler\_web.png 100% 78KB 6.7MB/s 00:00

cobbler.zip 100% 9781KB 61.1MB/s 00:00

cobbler\_boot.tar.gz 100% 416KB 45.7MB/s 00:00

[root@room9pc01 桌面]#

2）确认虚拟机中的目录内容

[root@cob ~]# ls /root/Cobbler/

cobbler\_boot.tar.gz cobbler\_web.png cobbler.zip

[root@cob ~]#

2. 搭建Yum仓库，安装Cobbler

1）将cobbler.zip包，解压缩到根目录下

[root@cob ~]# unzip /root/Cobbler/cobbler.zip -d /

……

[root@cob ~]# ls /cobbler/

……

2)搭建Yum仓库，利用yum安装所有rpm软件包

[root@cob ~]# mount /dev/cdrom /mnt/ #首先通过图形将CentOS光盘放入光驱设备

mount: /dev/sr0 写保护，将以只读方式挂载

[root@cob ~]# ls /mnt/

[root@cob ~]# rm -rf /etc/yum.repos.d/\*

[root@cob ~]# vim /etc/yum.repos.d/dvd.repo

[CentOS7]

name=CentOS 7.4

baseurl=file:///mnt

enabled=1

gpgcheck=0

[root@cob ~]# yum repolist

……

源标识 源名称 状态

CentOS7 CentOS 7.4 3,894

repolist: 3,894

[root@cob ~]# yum -y install /cobbler/\*.rpm

3. 设置防火墙与SELinux

1）设置防火墙默认区域为trusted

[root@cob ~]# firewall-cmd --set-default-zone=trusted

2）修改SELinux状态

[root@cob ~]# setenforce 0

[root@cob ~]# getenforce

Permissive

[root@cob ~]# vim /etc/selinux/config

……

SELINUX=permissive

……

4. 配置cobbler

1）修改配置文件/etc/cobbler/settings

[root@cob ~]# vim /etc/cobbler/settings

……

next\_server: 192.168.4.123 #设置下一个服务器还为本机

server: 192.168.4.123 #设置本机为cobbler服务器

manage\_dhcp: 1 #设置cobbler管理dhcp服务

pxe\_just\_once: 1 #防止客户端重复安装操作系统,增加默认从本机硬盘启动菜单

2）配置cobbler的dhcp分配网段及IP地址范围

[root@cob ~]# vim /etc/cobbler/dhcp.template

……

:%s /192.168.1/192.168.4/g #将所有192.168.1替换为192.168.4

3）绝对路径解压cobbler\_boot.tar.gz

[root@cob ~]# tar -tf /root/Cobbler/cobbler\_boot.tar.gz #众多的引导文件

……

[root@cob ~]# tar -xPf /root/Cobbler/cobbler\_boot.tar.gz

[root@cob ~]# ls /var/lib/cobbler/loaders/ #默认cobbler存放引导文件路径

5. 启动相关服务

1）启动cobblerd主服务

[root@cob ~]# systemctl restart cobblerd

[root@cob ~]# systemctl enable cobblerd

Created symlink from /etc/systemd/system/multi-user.target.wants/cobblerd.service to /usr/lib/systemd/system/cobblerd.service.

2）启动httpd主服务，主要提供Web页面、装机光盘内容，ks应答文件等

[root@cob ~]# systemctl restart httpd

[root@cob ~]# systemctl enable httpd

Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.

3）启动tftp主服务，主要提供pxelinux.0、菜单文件等

[root@cob ~]# systemctl restart tftp

[root@cob ~]# systemctl enable tftp

Created symlink from /etc/systemd/system/sockets.target.wants/tftp.socket to /usr/lib/systemd/system/tftp.socket.

4）启动rsyncd主服务，主要提供cobbler各个组件之间同步数据

[root@cob ~]# systemctl restart rsyncd

[root@cob ~]# systemctl enable rsyncd

Created symlink from /etc/systemd/system/multi-user.target.wants/rsyncd.service to /usr/lib/systemd/system/rsyncd.service.

6. 同步刷新cobbler所有配置

[root@cob ~]# cobbler sync

task started: 2018-03-22\_200534\_sync

task started (id=Sync, time=Thu Mar 22 20:05:34 2018)

running pre-sync triggers

cleaning trees

removing: /var/lib/tftpboot/grub/images

copying bootloaders

copying: /var/lib/cobbler/loaders/pxelinux.0 -> /var/lib/tftpboot/pxelinux.0

copying: /var/lib/cobbler/loaders/menu.c32 -> /var/lib/tftpboot/menu.c32

copying: /var/lib/cobbler/loaders/yaboot -> /var/lib/tftpboot/yaboot

copying: /usr/share/syslinux/memdisk -> /var/lib/tftpboot/memdisk

copying: /var/lib/cobbler/loaders/grub-x86.efi -> /var/lib/tftpboot/grub/grub-x86.efi

copying: /var/lib/cobbler/loaders/grub-x86\_64.efi -> /var/lib/tftpboot/grub/grub-x86\_64.efi

copying distros to tftpboot

copying images

generating PXE configuration files

generating PXE menu structure

rendering DHCP files

generating /etc/dhcp/dhcpd.conf

rendering TFTPD files

generating /etc/xinetd.d/tftp

cleaning link caches

running post-sync triggers

running python triggers from /var/lib/cobbler/triggers/sync/post/\*

running python trigger cobbler.modules.sync\_post\_restart\_services

running: dhcpd -t -q

received on stdout:

received on stderr:

running: service dhcpd restart

received on stdout:

received on stderr: Redirecting to /bin/systemctl restart dhcpd.service

running shell triggers from /var/lib/cobbler/triggers/sync/post/\*

running python triggers from /var/lib/cobbler/triggers/change/\*

running python trigger cobbler.modules.scm\_track

running shell triggers from /var/lib/cobbler/triggers/change/\*

\*\*\* TASK COMPLETE \*\*

7. 导入系统光盘镜像数据

1）导入CentOS系统光盘镜像，cobbler默认提供ks应答文件，但应答文件为最小化安装，命令格式：

cobbler import --path=挂载点 --name=导入系统命名 --arch=操作系统架构

cobbler导入的镜像放在：/var/www/cobbler/ks\_mirror

[root@cob ~]# cobbler import --path=/mnt --name=CentOS7 --arch=x86\_64

task started: 2018-03-22\_201215\_import

task started (id=Media import, time=Thu Mar 22 20:12:15 2018)

……

Keeping repodata as-is :/var/www/cobbler/ks\_mirror/CentOS7-x86\_64/repodata

\*\*\* TASK COMPLETE \*\*\*

[root@cob ~]# ls /var/www/cobbler/ks\_mirror/

CentOS7-x86\_64 config

[root@cob ~]#

2）首先卸载光驱设备挂载，通过图形将光驱设备中的光盘，换成RHEL7光盘如图-1所示，导入RHEL7系统光盘镜像，cobbler默认提供ks应答文件，但应答文件为最小化安装

[root@cob ~]# umount /mnt/ #卸载光驱设备，将光盘换成RHEL7



图-1

[root@cob ~]# mount /dev/cdrom /mnt/

mount: /dev/sr0 写保护，将以只读方式挂载

[root@cob ~]# ls /mnt/

addons extra\_files.json isolinux Packages RPM-GPG-KEY-redhat-release

EFI GPL LiveOS repodata TRANS.TBL

EULA images media.repo RPM-GPG-KEY-redhat-beta

[root@cob ~]# cobbler import --path=/mnt --name=RedHat --arch=x86\_64

task started: 2018-03-22\_202531\_import

task started (id=Media import, time=Thu Mar 22 20:25:31 2018)

……

Keeping repodata as-is :/var/www/cobbler/ks\_mirror/RedHat-x86\_64/addons/ResilientStorage/repodata

\*\*\* TASK COMPLETE \*\*\*

[root@cob ~]# ls /var/www/cobbler/ks\_mirror/

CentOS7-x86\_64 config RedHat-x86\_64

[root@cob ~]#

8. 新建虚拟机测试

1）新建一台虚拟机测试：

选择pxe安装方式

注意如果安装CentOS系统虚拟机内容必须为2G以上，安装RedHat内存1G以上，硬盘均在9G以上

测试虚拟机网络类型选择为：private1 如图-2与图-3所示。

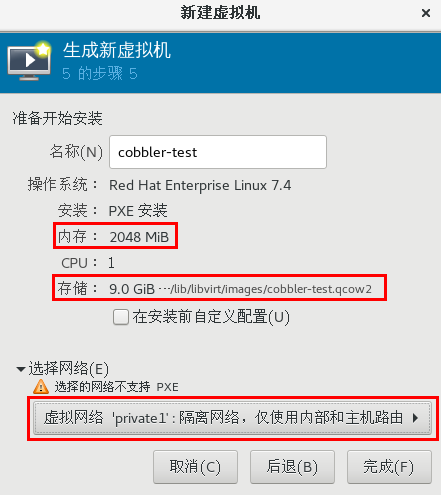


图-2

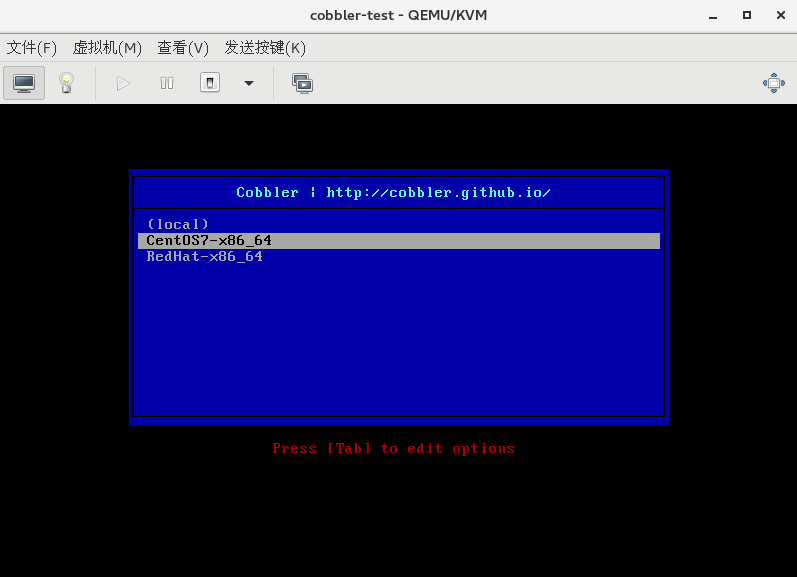


图-3