云计算系统管理

DAY01 开学典礼、云计算介绍、TCP/IP协议及配置

DAY02 Linux系统简介、安装RHEL7系统、RHEL7基本操作

DAY03 命令行基础、 目录和文件管理、 教学环境介绍

DAY04 软件包管理、配置网络、文本/文件查找

DAY05 管理用户和组、tar备份与恢复、NTP时间同步、cron计划任务

DAY06 权限和归属、使用LDAP认证、家目录漫游

DAY07 综合串讲 、综合练习

云计算应用管理

DAY01 分区规划及使用、LVM逻辑卷

DAY02 Shell脚本基础、使用变量、条件测试及选择、列表式循环

DAY03 系统安全保护、配置用户环境、配置高级连接、防火墙策略管理

DAY04 配置SMB共享、配置NFS共享

DAY05 ISCSI共享存储、数据库服务基础、管理表数据

DAY06 HTTP服务基础、网页内容访问、部署动态网站

DAY07 综合串讲、综合练习

系统&服务管理进阶

DAY01 扩展的几个应用、发布网络YUM源、vim编辑技巧、源码编译安装、systemctl控制

DAY02 DNS服务基础、特殊解析、DNS子域授权、缓存DNS

DAY03 Split分离解析、RAID磁盘阵列、进程管理、日志管理

DAY04 批量装机环境、配置PXE引导、kickstart自动应答

DAY05 rsync同步操作、inotify实时同步、Cobbler网络装机

###############################################################################

教学环境

真机：Linux操作系统 登陆的用户：root（最高权限） 密码：Taren1

虚拟机软件： 以真机的物理硬件，虚拟众多逻辑硬件

win2008：

密码为：Taren1

#########################################################

典型服务模式

• C/S,Client/Server架构

– 由服务器提供资源或某种功能

– 客户机使用资源或功能

###################################################

TCP/IP协议简介

• TCP/IP是最广泛支持的通信协议集合

– 包括大量Internet应用中的标准协议

– 支持跨网络架构、跨操作系统平台的通信

• 主机与主机之间通信的三个要素

– IP地址(IP address)

– 子网掩码(subnet mask)

– IP路由(IP router) 路由器

IP地址的概述

• 作用:用来标识一个计算机

• 地址组成(点分十进制):

– 一共32个二进制位

– 表示为4个十进制数,以 . 隔开

192.168.1.1

10.11.12.13

1.2.3.4

192.168.4.207

192.168.4.254

#####################################################

配置IP地址：

1.手工配置

2.自动配置（前提：网络中有DHCP服务器）

配置win2008的IP地址： 192.168.1.1

配置win2008的子网掩码：255.255.255.0

• 操作步骤:

– 右击桌面“网络”-->“属性”-->“更改适配器设置”

– 双击“本地连接”--> “属性”

– 双击“Internet协议版本4(TCP/IPv4)”

– 配置完成后,单击“确定”完成

查看IP地址：

• 图形方式

操作步骤:

– 右击桌面“网络”-->“属性”-->“更改适配器设置”

– 双击“本地连接”--> “详细信息”

• 命令方式

操作步骤:

– win键 + r ： 开始--->运行

– 回车

– 输入ipconfig 进行查看IP地址

– 输入exit 关闭命令行界面

################################################

• IP地址的分类

• 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

192.168.1.1------>只看第一个数字192------->为C类

##################################################

子网掩码：区分IP地址的网络位与主机位

用二进制的1表示网络位，用二进制的0表示主机位

IP地址由网络位与主机位组成

座机号： 区号+号码

010-12345678

0316-7351111

区号：标识一个区域

号码：在区域中座机的编号

192.168.1.1---->来自192.168.1区域（网络）---->编号位1的主机

192.168.2.1---->来自192.168.2区域（网络）---->编号位1的主机

192.168.10.1---》第一个数字为192---》C类地址---》网+网+网+主

11000000.10101000.00001010.00000001

11111111.11111111.11111111.00000000

192.168.10.1 255.255.255.0

也可以表示为 192.168.10.1/24 (24个网络位）

######################################################

两台机器互相通信

一、通过克隆产生一个新的机器

1.关闭win2008模版机器

操作：双击桌面“虚拟系统管理器”---->右击 “win2008”---->“克隆”---->“克隆”

2.两台机器均开启机器进入系统

3.密码均是：Taren1

二、 针对win2008-clone进行配置

1.修改桌面背景 双击桌面“控制面板”--->“显示”

2.进行配置IP地址： 192.168.1.2/24

三、测试通信

运行ping命令，专用于测试通信

默认进行4次，有去有回为一次ping

– win键 + r ： 开始--->运行

– 回车

命令格式：ping 对方的IP地址

关闭两台机器的防火墙：

双击桌面“控制面板”--->“Windows防火墙”--->“打开或关闭 Windows防火墙”

#######################################################

• 什么是网关（跨网络通信时，会用到网关）

– 从一个网络连接到另一个网络的“关口”

– 通常是一台路由器,或者防火墙/接入服务器

• DNS服务器作用：将域名解析为IP地址

www.qq.com -----》IP地址

#####################################################

• Linux命令

– 用来实现某一类功能的指令或程序

Linux执行命令，都要找到对应的程序

– 命令的执行依赖于解释器(默认解释器：/bin/bash)

用户------->解释器------->内核------->硬件

绿色：可以运行的程序

• Linux命令的分类

– 内部命令:属于解释器的一部分

– 外部命令:解释器之外的其他程序

命令行的一般格式

• 基本用法

– 命令字 [选项].. [参数1] [参数2]...

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

快速编辑技巧

• Tab键自动补全

– 可补齐命令字、选项、参数、文件路径、软件名、服务名

# ls /etc/sysconfig/network-scripts/ifcfg-eth0

# ls /et(tab)/sysco(tab)/netw(tab)-(tab)/ifcfg-eth0

# if 连续按两次tab

# ifco(tab)

# hostname

# cat /etc/redhat-release

• 快捷键

– Ctrl + c:结束正在运行的命令

– Esc + . 或 ALT + . :粘贴上一个命令的参数

[root@localhost ~]# ls /etc/redhat-release

[root@localhost ~]# cat ALT + .

[root@localhost ~]# cat -n ALT + .

[root@localhost ~]# ls -l ALT + .

– Ctrl + l:清空整个屏幕

– Ctrl + u:清空至行首

– Ctrl + w:往回删除一个单词(以空格界定)

#################################################

mount挂载操作

从光盘文件中获得众多的软件包

Windows：

光盘文件-------->光驱设备-------->CD驱动器（图标）访问点

Linux：

光盘文件-------->光驱设备-------->目录（访问点）

洞口（访问点）-------> 煤矿

在命令行访问设备，必须通过访问点进行访问

1.通过图形将光盘镜像文件，放入虚拟光驱设备

2.查看光驱设备

青色：快捷方式

[root@localhost ~]# ls /dev/cdrom

[root@localhost ~]# ls -l /dev/cdrom

[root@localhost ~]# ls /dev/sr0

3.进行挂载：让目录成为设备的访问点

• 使用mount命令

– 格式:mount 设备路径 挂载点目录

[root@localhost ~]# mkdir /dvd #创建目录dvd

[root@localhost ~]# ls /dvd

[root@localhost ~]# mount /dev/cdrom /dvd

mount: /dev/sr0 写保护，将以只读方式挂载

让/dvd成为/dev/cdrom 的访问点

[root@localhost ~]# ls /dvd/

[root@localhost ~]# ls /dvd/Packages

4.卸载

[root@localhost ~]# umount /dvd/

[root@localhost ~]# ls /dvd/

##################################################

注意事项：

1.卸载时，当前不要在访问点内

[root@localhost dvd]# umount /dvd/

umount: /dvd：目标忙。

2.挂载设备时，访问点最好时自己创建的目录

#################################################

目录和文件管理

• pwd — Print Working Directory

– 用途:查看当前工作目录

• cd — Change Directory

– 用途:切换工作目录

– 格式:cd [目标文件夹位置]

～：代表用户的家目录

~user表示用户user的家目录

/root：管理员的家目录

/home：存放所有普通用户的家目录

[root@localhost /]# cd ~root #去往root用户家目录

[root@localhost ～]# pwd

[root@localhost ～]# useradd dc #创建用户dc

[root@localhost ～]# cd ~dc #去往dc用户家目录

[root@localhost dc]# pwd

[root@localhost dc]# useradd harry #创建用户harry

[root@localhost dc]# cd ~harry #去往harry用户家目录

[root@localhost harry]# pwd

##############################################

. 表示当前目录，经常复制与移动

.. 表示父目录

###############################################

• ls — List

– 格式:ls [选项]... [目录或文件名]

• 常用命令选项

– -l :以长格式显示，显示详细属性

– -h:提供易读的容量单位(K、M等)

– -d:显示目录本身(而不是内容)的属性

– -A:包括名称以 . 开头的隐藏文档

[root@localhost ~]# ls --help #显示ls命令的帮助信息

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

[root@localhost ~]# ls -lh /boot

[root@localhost ~]# ls -ld /boot

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

[root@localhost ~]# ls /home

[root@localhost ~]# ls -lh /home

[root@localhost ~]# mkdir /mnt/.nsd1808

[root@localhost ~]# ls /mnt/

[root@localhost ~]# ls -A /mnt/

[root@localhost ~]# ls -A /root

[root@localhost ~]# ls -A /boot

[root@localhost ~]# man passwd #查看权威帮助信息

[root@localhost ~]# man ls #查看权威帮助信息

/example #全文搜索example

###############################################

使用通配符

• 针对不确定的文档名称,以特殊字符表示

– \*:任意多个任意字符

– ?:单个字符

[root@localhost /]# ls /root/a\* #列出以a开头

[root@localhost /]# ls /boot/vm\*

[root@localhost /]# ls /etc/\*tab #列出以tab结尾

[root@localhost /]# ls /etc/\*.conf

[root@localhost /]# ls /dev/tty\*

[root@localhost /]# ls /dev/tty?

[root@localhost /]# ls /dev/tty?？

[root@localhost /]# ls /dev/tty?？？

– [a-z]:多个字符或连续范围中的一个,若无则忽略

– {a,min,xy}:多组不同的字符串,全匹配

[root@localhost /]# ls /dev/tty[3-6]

[root@localhost /]# ls /dev/tty[1-3]

[root@localhost /]# ls /dev/tty{1,3,5,7,11,31}

[root@localhost /]# ls /dev/tty{1,3,5,7,11,31,S0}

[root@localhost /]# ls /dev/tty{S0,S3}

请列出/dev/下，tty20至tty30的设备文件？

[root@localhost /]# ls /dev/tty2[0-9] /dev/tty30

[root@localhost /]# ls /dev/tty{2[0-9],30}

############################################

别名的定义：简化复杂命令

• 查看已设置的别名

– alias [别名名称]

• 定义新的别名

– alias 别名名称= '实际执行的命令行'

• 取消已设置的别名

– unalias [别名名称]

[root@localhost /]# alias hn='hostname' #定义别名

[root@localhost /]# alias #查看当前系统生效的别名

[root@localhost /]# hn #执行成功

[root@localhost /]# unalias hn #删除别名

[root@localhost /]# hn #执行失败

[root@localhost /]# alias myls='ls -lh' #定义别名

[root@localhost /]# alias #查看当前系统生效的别名

[root@localhost /]# myls /root #执行成功

[root@localhost /]# unalias myls #删除别名

[root@localhost /]# myls /root #执行失败

###############################################

请定义一个自定义的命令qstat

执行该命令可以运行：/bin/ps -Ao pid,tt,user,fname,rsz

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

# qstat

################################################

mkdir 创建目录

• mkdir — Make Directory

– 格式:mkdir [-p] [/路径/]目录名 //-p表示连父目录一起创建

[root@localhost /]# mkdir -p /vod/movie/cartoon

[root@localhost /]# ls /

[root@localhost /]# ls /vod

[root@localhost /]# ls /vod/movie

[root@localhost /]# mkdir -p /mnt/aa/bb/cc/dd

[root@localhost /]# ls /mnt/

[root@localhost /]# ls /mnt/aa

[root@localhost /]# ls /mnt/aa/bb

[root@localhost /]# ls /mnt/aa/bb/cc

#################################################

[root@localhost /]# mkdir /root/test01 /mnt/test02

[root@localhost /]# ls /root/

[root@localhost /]# ls /mnt/

[root@server0 ~]# ls -R /mnt/aa #-R表示递归显示

递归:目录本身，以及目录下所有都算上

#################################################

• vim文本编辑器

– 格式: vim [[/目录/]文件名]

– 若目标文件不存在,则新建空文件并编辑

– 若目标文件已存在,则打开此文件并编辑

三个模式：命令模式 插入模式（编辑模式） 末行模式

[root@localhost /]# vim /mnt/haxi.txt

i 键

命 ----------------->插入模式(按Esc 键回到命令模式)

令

模

式 ----------------->末行模式(按Esc 键回到命令模式)

: 键

末行模式 ：wq 保存并退出

末行模式 ：q！ 强制不保存并退出

##############################################

rm 删除

• rm — Remove

– 格式:rm [选项]... 文件或目录...

• 常用命令选项

– -r、-f:递归删除(含目录)、强制删除

[root@localhost /]# mkdir /opt/abc /mnt/test05

[root@localhost /]# ls /opt/

[root@localhost /]# ls /mnt/

[root@localhost /]# rm -rf /opt/abc/ /mnt/test05/

[root@localhost /]# ls /opt/

[root@localhost /]# ls /mnt/

###############################################

mv 移动/重命名

• mv — Move

– 格式:mv 原文件... 目标路径

[root@localhost /]# rm -rf /mnt/\*

[root@localhost /]# mkdir /mnt

[root@localhost /]# mkdir /mnt/nsd01

[root@localhost /]# touch /mnt/a.txt

[root@localhost /]# ls /mnt/

[root@localhost /]# mv /mnt/a.txt /mnt/nsd01/

[root@localhost /]# ls /mnt

[root@localhost /]# ls /mnt/nsd01

重命名：路径不变的移动

[root@localhost /]# mv /mnt/nsd01/ /mnt/student

[root@localhost /]# ls /mnt/

[root@localhost /]# mv /mnt/student/ /mnt/file

[root@localhost /]# ls /mnt/

[root@localhost /]# mv /mnt/file /mnt/abcd

[root@localhost /]# ls /mnt/

################################################

cp 复制

• cp — Copy

– 格式:cp [选项]... 原文件... 目标路径

• 常用命令选项

– -r:递归,复制目录时必须有此选项

[root@localhost /]# rm -rf /mnt/\*

[root@localhost /]# ls /mnt/

[root@localhost /]# cp /etc/passwd /mnt/

[root@localhost /]# ls /mnt/

[root@localhost /]# cp /etc/redhat-release /mnt/

[root@localhost /]# ls /mnt/

[root@localhost /]# cp -r /boot/ /mnt/

[root@localhost /]# ls /mnt/

[root@localhost /]# ls /mnt/boot/

处理复制时出现重名，强制覆盖

在本次操作临时取消别名的操作

[root@localhost /]# \cp -r /boot/ /mnt/

[root@localhost /]# \cp -r /boot/ /mnt/

[root@localhost /]# cp -r /boot/ /mnt/

可以支持两个以上的参数，永远把最后一个参数作为目标，其他的所有参数都作为源

# cp -r /home/ /etc/fstab /tmp/ /opt/ /mnt/

# ls /mnt

复制与 点 的结合应用: 将源文档复制到，当前路径下

.表示 当前路径

# cd /etc/sysconfig/network-scripts

# cp /etc/fstab .

复制时候，可以重新命名，目标路径文档的名字

[root@localhost /]# rm -rf /mnt/\*

[root@localhost /]# cp /etc/redhat-release /mnt/

[root@localhost /]# ls /mnt/

[root@localhost /]# cp /etc/redhat-release /mnt/a

[root@localhost /]# ls /mnt/

[root@localhost /]# rm -rf /mnt/\*

[root@localhost /]# cp -r /home/ /mnt/test

[root@localhost /]# ls /mnt

[root@localhost /]# cp -r /home/ /mnt/test

[root@localhost /]# ls /mnt/test

请描述 两次复制的不同？

1.将/home目录复制到/mnt下重命名为test

2.将/home目录复制到/mnt/test/目录下

###############################################

查看时间date

[root@localhost /]# date

2018年 09月 03日 星期一 17:21:41 CST

[root@localhost /]# date -s "年-月-日 时：分：秒"

[root@localhost /]# date -s "2000-9-3 10:20"

[root@localhost /]# date

[root@localhost /]# date -s "2018-9-3 17:23"

交互式计算器： bc

运算法则： 加 + 减 - 乘 \* 除 / 取余数 %

##############################################

使用教学虚拟机

• 每个学员机上有三台预先配置好的虚拟机

– server —— 作为练习用服务器

– desktop —— 作为练习用客户机

– classroom —— 提供网关/DNS/软件素材/DHCP服务等资源

优先开机classroom，再去开启server、desktop

一、真机还原命令：优先还原classroom

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server[root@room9pc01 ~]# rht-vmctl reset desktop

二、进入 虚拟机server 与 虚拟机desktop

用户名：root 密码为：redhat

虚拟机server

主机名：server0.example.com

系统版本：RHEL7.0

eth0网卡IP地址：172.25.0.11

虚拟机desktop

主机名：desktop0.example.com

系统版本：RHEL7.0

eth0网卡IP地址：172.25.0.10

三、真机远程管理

[root@room9pc01 ~]# ping 172.25.0.11

[root@room9pc01 ~]# ping 172.25.0.10

远程管理的命令：ssh 登陆对方的用户名@对方的IP地址

[root@room9pc01 ~]# ssh root@172.25.0.11

Last login: Wed Sep 5 09:11:52 2018

[root@server0 ~]# ifconfig

[root@server0 ~]# exit

登出

Connection to 172.25.0.11 closed.

[root@room9pc01 ~]#

Ctrl + shift + t :新开一个终端

ssh -X:在远程管理时，本地运行对方的图形程序

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

[root@server0 ~]# firewall-config #防火墙图形工具

#################################################

四、为真机设置永久的别名，修改配置文件

设置永久的别名配置文件：/root/.bashrc 系统级配置文件

[root@room9pc01 ~]# vim /root/.bashrc

# .bashrc

alias gos='ssh -X root@172.25.0.11'

alias god='ssh -X root@172.25.0.10'

新开一个全新的终端

[root@room9pc01 ~]# gos

[root@room9pc01 ~]# god

################################################

软件包的管理

1.由classroom提供光盘RHEL7.0所有内容，具备软件包

http:超文本传输协议

虚拟机classroom搭建Web服务器，共享光盘RHEL7.0所有内容

真机打开浏览器：http://classroom.example.com/

classroom.example.com/content/rhel7.0/x86\_64/dvd

2.下载软件包

wget 下载工具 打开网页，按Ctrl+f 输入vsftpd

# cd /opt/

# wget http://classroom.example.com/content/rhel7.0/x86\_64/dvd/Packages/vsftpd-3.0.2-9.el7.x86\_64.rpm

# ls

3.安装软件包的命令

使用rpm命令管理软件,默认不允许用户做任何选择

• RPM Package Manager,RPM包管理器

– rpm -q 软件名... #查询当前系统软件是否安装

– rpm -ivh 软件名-版本信息.rpm... #安装软件包

– rpm -e 软件名... #卸载

[root@server0 opt]# rpm -ql vsftpd #列出安装清单

了解：导入红帽签名信息

[root@server0 /]# wget http://classroom.example.com/content/rhel7.0/x86\_64/dvd/RPM-GPG-KEY-redhat-release

[root@server0 /]# rpm --import /RPM-GPG-KEY-redhat-release

##############################################

[root@server0 /]# rpm -q vsftpd

[root@server0 /]# rpm -q firefox #查询软件是否安装

[root@server0 /]# rpm -ivh /opt/vsftpd-3.0.2-9.el7.x86\_64.rpm #安装软件

[root@server0 /]# rpm -e vsftpd #卸载软件

[root@server0 /]# rpm -q vsftpd #查询软件是否卸载

[root@server0 /]# rpm -ql firefox #查询安装清单

################################################

vsftpd-3.0.2-9.el7.x86\_64.rpm(包名)

vsftpd(软件名)

################################################

常见的错误信息：依赖关系错误

错误：依赖检测失败：

bind = 32:9.9.4-14.el7 被 bind-chroot-32:9.9.4-14.el7.x86\_64 需要

################################################

Yum软件包仓库，自动解决依赖关系

服务：为客户端自动解决依赖关系，安装软件包

服务端：虚拟机classroom.example.com

1.众多的软件包 2.仓库清单文件（repodata）

3.构建Web服务或FTP服务传递数据

客户端：虚拟机server0.example.com

客户端配置文件：/etc/yum.repos.d/\*.repo

错误的配置文件会影响正确的配置文件

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

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

[rhel7] #仓库标识

name=rhel7.0 #仓库描述信息

baseurl=http://classroom.example.com/content/rhel7.0/x86\_64/dvd/ #指定服务端位置

enabled=1 #是否启用该文件

gpgcheck=0 #是否检测红帽签名

#################################################

Yum使用

# yum repolist #列出仓库信息

# yum -y install httpd #安装软件包

# yum -y install sssd

# yum -y install gcc

# yum -y install system-config-kickstart

# yum remove 软件名 #卸载

# yum clean all #清缓存

#yum reinstall 包名 #重新安装包

##################################################

虚拟机Server：升级内核

1.下载内核软件包

# cd /opt

# wget http://classroom.example.com/content/rhel7.0/x86\_64/errata/Packages/kernel-3.10.0-123.1.2.el7.x86\_64.rpm

# ls

2.升级内核

# uname -r

# rpm -ivh /opt/kernel-3.10.0-123.1.2.el7.x86\_64.rpm

# reboot

# gos

# uname -r

3.10.0-123.1.2.el7.x86\_64

#################################################

配置网络参数

一、配置永久的主机名

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

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

A.tedu.cn

[root@server0 ~]# hostname

[root@server0 ~]# exit

登出

Connection to 172.25.0.11 closed.

[root@room9pc01 ~]# gos

[root@A ~]# hostname

二、配置永久的IP地址、子网掩码、网关地址

网卡配置文件/etc/sysconfig/network-scripts/ifcfg-eth0

通过命令nmcli connection #可以修改网卡配置文件

1.查看命令识别网卡的名称

[root@A ~]# nmcli connection show

2.进行配置永久的IP地址、子网掩码、网关地址

# nmcli connection modify 'System eth0' ipv4.method manual

ipv4.addresses '172.25.0.110/24 172.25.0.254'

connection.autoconnect yes

# nmcli connection 修改 '识别的网卡名'

ipv4.方法 手工配置

ipv4.addresses 'IP地址/子网掩码 网关地址'

每次开机自动激活配置

3.激活配置

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

方式2：nmtui 修改IP地址为172.25.0.120/24

1. Edit a connection（回车）#编辑连接

2. System eth0（回车） #选择网卡

3. Require IPv4 addressing for this connection

#利用IPv4进行激活

4. Automatically connect

#每次开机自动激活配置

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

查看网关命令：

[root@A ~]# route -n #以数字的方式显示

Gateway

172.25.0.254

#################################################

三、配置永久的DNS服务器地址

[root@A ~]# vim /etc/resolv.conf

nameserver 172.25.254.254

#################################################

使用教学虚拟机

• 每个学员机上有三台预先配置好的虚拟机

– server —— 作为练习用服务器

– desktop —— 作为练习用客户机

– classroom —— 提供网关/DNS/软件素材/DHCP服务等资源

优先开机classroom，再去开启server、desktop

一、真机还原命令：优先还原classroom

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server[root@room9pc01 ~]# rht-vmctl reset desktop

二、进入 虚拟机server 与 虚拟机desktop,检测Yum是否可用

# yum clean all #清空缓存

# yum repolist #列出仓库信息

###################################################

管理用户和组

用户帐号作用：

1. 访问控制的设置，不同的用户具备不同权限

2. 提高安全可靠性

组帐号作用：方便管理用户帐号

用户帐号标识： UID

管理员UID为0

系统程序用户 默认1～999

所有普通用户UID默认从1000起始

组帐号标识： GID

组帐号分类： 基本组 附加组（从属组）

所有用户至少属于一个组

基本组：由Linux系统创建，由Linux将用户加入，与用户同名

附加组（从属组）：由root用户创建

###############################################

/etc/passwd:用户基本信息存放在 /etc/passwd 文件

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

root:x:0:0:root:/root:/bin/bash

用户名:密码占位符:UID:基本组GID:用户描述信息:家目录:解释器

• 使用 useradd 命令

– useradd [选项]... 用户名

• 常用命令选项

– -u 用户id、-d 家目录路径、-s 登录解释器、-G 附加组

用户名:密码占位符:UID:基本组GID:用户描述信息:家目录:解释器

[root@server0 /]# useradd nsd01

[root@server0 /]# id nsd01 #查看基本信息

[root@server0 /]# grep nsd /etc/passwd

[root@server0 /]# useradd nsd02

[root@server0 /]# grep nsd /etc/passwd

[root@server0 /]# useradd -u 1600 nsd05 #指定UID

[root@server0 /]# grep nsd /etc/passwd

[root@server0 /]# id nsd05

# useradd -d /opt/nsd07 nsd07 #指定用户家目录创建

# grep nsd07 /etc/passwd

# ls /opt/

-s：登录解释器

/sbin/nologin :禁止用户登陆系统

[root@server0 /]# useradd -s /sbin/nologin nsd08

[root@server0 /]# grep nsd08 /etc/passwd

[root@server0 /]# groupadd tarena

[root@server0 /]# useradd -G tarena harry #指定附加组

[root@server0 /]# id harry

#############################################

重定向输出:将前面命令的输出结果，写入文本文件

>：覆盖重定向

>>：追加重定向

[root@server0 /]# ls --help

[root@server0 /]# ls --help > /opt/ls.txt

[root@server0 /]# vim /opt/ls.txt

[root@server0 /]# less /opt/ls.txt

[root@server0 /]# hostname > /opt/ls.txt

[root@server0 /]# cat /opt/ls.txt

[root@server0 /]# hostname >> /opt/ls.txt

[root@server0 /]# cat /opt/ls.txt

[root@server0 /]# ifconfig eth0 >> /opt/ls.txt

[root@server0 /]# echo hello world

[root@server0 /]# echo hello world > /opt/ls.txt

[root@server0 /]# cat /opt/ls.txt

[root@server0 /]# echo A.tedu.cn > /etc/hostname

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

[root@server0 /]# hostname

管道 | :将前面命令的输出结果，交由面后命令处理

[root@A ~]# head -12 /etc/passwd

[root@A ~]# head -12 /etc/passwd | tail -5

[root@A ~]# cat -n /etc/passwd

[root@A ~]# cat -n /etc/passwd | head -12 | tail -5

[root@A ~]# ifconfig | less

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

[root@A ~]# echo 1+2 | bc

##################################################

• 使用 passwd 命令

– passwd [用户名]

[root@A ~]# passwd nsd01

更改用户 nsd01 的密码 。

新的 密码： #输入密码

无效的密码： 密码是一个回文

重新输入新的 密码： #输入密码

passwd：所有的身份验证令牌已经成功更新。

[root@A ~]# su - nsd01 #临时切换用户身份： su - 用户名

[nsd01@A ~]$ passwd

Changing password for user nsd01.

Changing password for nsd01.

(current) UNIX password: #输入旧密码

New password: #输入新密码

Retype new password: #重新输入新密码

passwd: all authentication tokens updated successfully.

[nsd01@A ~]$ exit

################################################

非交互式设置用户密码

– echo '密码' | passwd --stdin 用户名

[root@A ~]# echo 123 | passwd --stdin nsd01

[root@A ~]# echo 123 | passwd --stdin nsd02

################################################

用户密码信息存放在 /etc/shadow 文件

用户名：密码加密字符串:上一次修改密码的时间

自1970-1-1到上一次修改密码时间经过的天数

#############################################

修改用户属性

• 使用 usermod 命令

– usermod [选项]... 用户名

• 常用命令选项

– -u 用户id、-d 家目录路径、-s 登录解释器、-G 附加组

[root@A ~]# useradd nsd10

[root@A ~]# grep nsd10 /etc/passwd

# usermod -u 1700 -s /sbin/nologin -d /opt/abc

-G tarena nsd10

[root@A ~]# grep nsd10 /etc/passwd

[root@A ~]# id nsd10

###############################################

删除用户

• 使用 userdel 命令

– userdel [-r] 用户名 # -r:连同用户家目录一并删除

[root@server0 ~]# userdel alex

[root@server0 ~]# id alex

id: alex: no such user

###############################################

管理组账号

组基本信息存放在 /etc/group 文件

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

stugrp:x:2002:

组名：密码占位符：组的GID：组的成员列表

• 使用 groupadd 命令

– groupadd [-g 组ID] 组名

[root@A ~]# groupadd stugrp

[root@A ~]# grep stugrp /etc/group #查看组的信息

stugrp:x:2002:

[root@A ~]# useradd kenji

[root@A ~]# useradd chihiro

[root@A ~]# useradd jack

[root@A ~]# useradd natasha

################################################

• 使用 gpasswd 命令

– gpasswd -a 用户名 组名

– gpasswd -d 用户名 组名

[root@A ~]# gpasswd -a kenji stugrp

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

[root@A ~]# gpasswd -a natasha stugrp

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

[root@A ~]# gpasswd -d kenji stugrp

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

[root@A ~]# gpasswd -a jack stugrp

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

[root@A ~]# gpasswd -a chihiro stugrp

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

删除组

• 使用 groupdel 命令

– groupdel 组名

###############################################

创建一个用户Linux有那些文件会改变？

[root@A ~]# grep nsd20 /etc/passwd

[root@A ~]# ls /home/

[root@A ~]# grep nsd20 /etc/shadow

[root@A ~]# grep nsd20 /etc/group

##############################################

tar备份与恢复

归档及压缩：1.零散的文件归成一个 2.占用硬盘空间变小

常见的压缩格式及命令工具:

.gz ---> gzip

.bz2 ---> bzip2

.xz ---> xz

制作归档及压缩（打包----->tar包）

格式：tar 选项 /路径/tar包名字 /路径/源文件 /路径/源文件

解包

格式：tar 选项 /路径/tar包名字 /路径/释放的位置

– -c:创建归档

– -x:释放归档

– -f:指定归档文件名称

– -z、-j、-J:调用 .gz、.bz2、.xz 格式的工具进行处理

– -t:显示归档中的文件清单

– -C（大写）:指定释放位置

打包：

tar -zcf /路径/tar包名字 /路径/源文件

tar -jcf /路径/tar包名字 /路径/源文件

tar -Jcf /路径/tar包名字 /路径/源文件

选项顺序要求：-f必须在最后

解包：

tar -xf /路径/tar包名字 -C /路径/指定释放位置

红色：压缩包

[root@A ~]# rm -rf /opt/\*

[root@A ~]# rm -rf /mnt/\*

# tar -zcf /opt/nsd02.tar.gz /home/ /boot/

# ls /opt/

# tar -jcf /opt/test01.tar.bz2 /home/ /boot/

# ls /opt/

# tar -Jcf /opt/file01.tar.xz /home/ /boot/

# ls /opt/

# tar -xf /opt/nsd02.tar.gz -C /mnt/

# ls /mnt/

# ls /mnt/boot/

################################################

查看tar包内的内容

[root@A ~]# tar -tf /opt/file01.tar.xz

案例2:创建一个备份包

使用 tar 工具完成以下备份任务:

– 创建一个名为 /root/backup.tar.bz2 的归档文件

– 其中包含 /usr/local 目录中的内容

– tar 归档必须使用 bzip2 进行压缩

# tar -jcf /root/backup.tar.bz2 /usr/local/

# ls /root/

# tar -tf /root/backup.tar.bz2 #查看包内容

################################################

NTP时间同步

NTP网络时间协议

• Network Time Protocol

– NTP服务器为客户机提供标准时间

– NTP客户机需要与NTP服务器保持沟通

服务端：虚拟机classroom

客户端：虚拟机Server

1.安装chrony软件包（客户端软件：专用于访问服务的软件）

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

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

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

daemon:超级守护进程

4.验证：

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

2008年 10月 01日 星期三 12:00:00 CST

[root@A ~]# date

[root@A ~]# systemctl restart chronyd

[root@A ~]# date

[root@A ~]# date

############################################

cron计划任务

周期性任务

• 软件包:cronie、crontabs

• 系统服务:crond

• 日志文件:/var/log/crond 排除故障

• 配置任务格式

– 分 时 日 月 周 任务命令行(绝对路径)

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

\*:匹配范围内任意时间

,:分隔多个不连续的时间点

-:指定连续时间范围

/n:指定时间频率,每n

每分钟记录当前系统的时间，写入到/opt/time.txt

[root@A ~]# date >> /opt/time.txt

[root@A ~]# cat /opt/time.txt

[root@A ~]# crontab -e -u root #以root书写计划任务

[root@A ~]# crontab -l -u root

\* \* \* \* \* date >> /opt/time.txt

[root@A ~]# ls /var/spool/cron/ #任务文件存放路径

[root@A ~]# cat /var/spool/cron/root #查看任务文件内容

[root@A ~]# cat /opt/time.txt #查看结果

################################################

使用教学虚拟机

• 每个学员机上有三台预先配置好的虚拟机

– server —— 作为练习用服务器

– desktop —— 作为练习用客户机

– classroom —— 提供网关/DNS/软件素材/DHCP服务等资源

优先开机classroom，再去开启server、desktop

一、真机还原命令：优先还原classroom

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server[root@room9pc01 ~]# rht-vmctl reset desktop

二、进入 虚拟机server 与 虚拟机desktop,检测Yum是否可用

# yum clean all #清空缓存

# yum repolist #列出仓库信息

###################################################

基本权限

• 访问方式(权限)

– 读取:允许查看内容-read r

– 写入:允许修改内容-write w

– 可执行:允许运行和切换-execute x

对于文本文件作用：

r:查看文件内容 cat less head tail

w:修改文件内容 vim 保存

x:双击运行该文件

• 权限适用对象(归属)

– 所有者:拥有此文件/目录的用户-user u

– 所属组:拥有此文件/目录的组-group g

– 其他用户:除所有者、所属组以外的用户-other o

• 使用 ls -l 命令

– ls -ld 文件或目录...

以 - 开头：代表文本文件

以 d 开头：代表目录

以 l 开头：快捷方式

[root@server0 /]# ls -ld /etc/

[root@server0 /]# ls -l /etc/passwd

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

[root@server0 /]# ls -ld /home/student

[root@server0 /]# ls -ld /tmp

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

[root@server0 /]# ls -l /etc/shadow

###########################################

设置基本权限

• 使用 chmod 命令

– chmod [-R] 归属关系+-=权限类别 文档...

[-R]:递归修改权限

[root@server0 /]# mkdir /nsd01

[root@server0 /]# ls -ld /nsd01

[root@server0 /]# chmod u-w /nsd01

[root@server0 /]# ls -ld /nsd01

[root@server0 /]# chmod g+w /nsd01

[root@server0 /]# ls -ld /nsd01

[root@server0 /]# chmod o=--- /nsd01

[root@server0 /]# ls -ld /nsd01

[root@server0 /]# chmod u=rwx,g=rx,o=rx /nsd01

[root@server0 /]# ls -ld /nsd01

[root@server0 /]# chmod ugo=rwx /nsd01

[root@server0 /]# ls -ld /nsd01

################################################

Linux判断用户具备的权限： 匹配及停止

1.查看用户对于，文档处于什么身份 所有者>所属组>其他人

2.查看相应权限位置的权限

Permission denied:权限不足

##########################################

目录的 r 权限:能够 ls 浏览此目录内容

目录的 w 权限:能够执行 rm/mv/cp/mkdir/touch/等更改目录内容的操作

目录的 x 权限:能够 cd 切换到此目录

###############################################

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

1）使用户lisi能够在此目录下创建子目录 切换用户 su - lisi

chmod o+w /nsddir/

2）使用户lisi不能够在此目录下创建子目录

chmod o-w /nsddir/

3）使用户lisi能够修改readme.txt文件

chmod o+w /nsddir/readme.txt

4）调整此目录的权限，使所有用户都不能cd进入此目录

chmod u-x,g-x,o-x /nsddir/

5）为此目录及其下所有文档设置权限 rwxr-x---

chmod -R u=rwx,g=rx,o=--- /nsddir/

##############################################

设置文档归属

• 使用 chown 命令

– chown [-R] 属主 文档...

– chown [-R] :属组 文档...

– chown [-R] 属主:属组 文档..

[-R]:递归修改归属关系

[root@server0 /]# useradd zhangsan

[root@server0 /]# groupadd tarena

[root@server0 /]# mkdir /nsd03

[root@server0 /]# ls -ld /nsd03

[root@server0 /]# chown zhangsan:tarena /nsd03

[root@server0 /]# ls -ld /nsd03

[root@server0 /]# chown lisi /nsd03 #修改所有者

[root@server0 /]# ls -ld /nsd03

[root@server0 /]# chown :root /nsd03 #修改所属组

[root@server0 /]# ls -ld /nsd03

#################################################

利用root用户新建/nsd06目录，并进一步完成下列操作

1）将属主设为gelin01，属组设为tarena组

[root@server0 /]# useradd gelin01

[root@server0 /]# groupadd tarena

[root@server0 /]# chown gelin01:tarena /nsd06

2）使用户gelin01对此目录具有rwx权限，其他人对此目录无任何权限

[root@server0 /]# chmod o=--- /nsd06

3）使用户gelin02能进入、查看此目录

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

4）将gelin01加入tarena组, 将nsd06目录的权限设为rw-r-x---

再测试gelin01用户能否进入此目录

[root@server0 /]# chmod u=rw,g=rx /nsd06

##################################################

附加权限(特殊权限)

Set GID

• 附加在属组的 x 位上

– 属组的权限标识会变为 s

– 适用于目录,Set GID可以使目录下新增的文档自动设

置与父目录相同的属组

– 让新增的子文档，自动继承父目录的所属组

[root@server0 /]# mkdir /nsd09

[root@server0 /]# chown :tarena /nsd09 #修改所属组权限

[root@server0 /]# ls -ld /nsd09

[root@server0 /]# mkdir /nsd09/test01

[root@server0 /]# ls -ld /nsd09/test01

[root@server0 /]# chmod g+s /nsd09 #添加Set GID权限

[root@server0 /]# ls -ld /nsd09

[root@server0 /]# mkdir /nsd09/test02

[root@server0 /]# ls -ld /nsd09/test02

[root@server0 /]# touch /nsd09/test02/2.txt

[root@server0 /]# ls -l /nsd09/test02/2.txt

############################################

acl访问控制列表

acl策略的作用

• 文档归属的局限性

– 任何人只属于三种角色:属主、属组、其他人

– 无法实现更精细的控制

• acl访问策略

– 能够对个别用户、个别组设置独立的权限

– 大多数挂载的EXT3/4、XFS文件系统默认已支持

[root@server0 /]# mkdir /nsd11

[root@server0 /]# chmod o=--- /nsd11

[root@server0 /]# su - zhangsan

[zhangsan@server0 ~]$ cd /nsd11 #切换失败

-bash: cd: /nsd11: Permission denied

[zhangsan@server0 ~]$ exit

[root@server0 /]# setfacl -m u:zhangsan:rx /nsd11

[root@server0 /]# getfacl /nsd11

[root@server0 /]# su - zhangsan

[zhangsan@server0 ~]$ cd /nsd11 #切换成功

[zhangsan@server0 nsd11]$ pwd

[zhangsan@server0 nsd11]$ exit

[root@server0 /]#

##################################################

创建一个文件1.txt，如果想要让lisi对文件具备读写权限

你有几种办法，请写出方案？

第一种：如果lisi为其他人 chmod o=rw 1.txt

第二种：chown 修改所属组，将lisi所属的组设置为1.txt所属组

chmod g=rw 1.txt

第三种：chown 修改所有者，将lisi设置为1.txt的所有者

chmod u=rw 1.txt

第四种：通过ACL权限实现 setfacl -m u:lisi:rw 1.txt

#################################################

• 使用 getfacl、setfacl 命令

– getfacl 文档...

– setfacl -m u:用户名:权限类别 文档...

– setfacl -m g:组名:权限类别 文档...

– setfacl -x u:用户名 文档... #删除指定的ACL

– setfacl -b 文档... #删除所有的ACL

[-R]：递归设置ACL权限

# mkdir /nsd12

# setfacl -m u:zhangsan:rwx /nsd12 #设置用户ACL

# setfacl -m u:lisi:rx /nsd12 #设置用户ACL

# setfacl -m u:dc:rwx /nsd12 #设置用户ACL

# setfacl -x u:zhangsan /nsd12 #删除zhangsan的ACL

# getfacl /nsd12

# setfacl -b /nsd12 #删除所有的ACL

# getfacl /nsd12

################################################

[root@server0 /]# setfacl -m u:lisi:--- /public/

[root@server0 /]# getfacl /public/

#################################################

使用LDAP认证

主要实现网络用户认证，达到用户的集中管理

本地用户：由本地创建，用于本地登陆 本地/etc/passwd

网络用户 : 由LDAP服务器提供用户名、密码信息

服务端：LDAP服务器 虚拟机classroom

客户端：虚拟机server

1.安装sssd软件包，与LDAP服务端沟通的软件

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

2.安装authconfig-gtk图形的工具，配置sssd软件包

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

3.运行authconfig-gtk图形的工具

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

[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

最后选择“应用” dc为域控制器

4.重起服务，验证

[root@server0 ~]# systemctl restart sssd

[root@server0 ~]# systemctl enable sssd

[root@server0 ~]# grep ldapuser0 /etc/passwd

[root@server0 ~]# id ldapuser0

[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

###############################################

家目录漫游

什么是NFS共享

• Network File System,网络文件系统

– 由NFS服务器将指定的文件夹共享给客户机

– 客户机将此共享目录 mount 到本地目录,访问此共享

资源就像访问本地目录一样方便

– 类似于 EXT4、XFS等类型,只不过资源在网上

NFS共享服务端：虚拟机classroom

客户端：虚拟机Server 访问虚拟机classroom的共享

1. 查看NFS资源

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

Export list for classroom.example.com:

/home/guests 172.25.0.0/255.255.0.0

服务端的共享路径 共享给那些客户端

2.访问虚拟机classroom的共享

# umount /mnt

# mkdir /home/guests

# mount classroom.example.com:/home/guests /home/guests

# ls /home/guests

# mount 服务端:服务端共享路径 本地挂载点

#################################################

使用教学虚拟机

• 每个学员机上有三台预先配置好的虚拟机

– server —— 作为练习用服务器

– desktop —— 作为练习用客户机

– classroom —— 提供网关/DNS/软件素材/DHCP服务等资源

优先开机classroom，再去开启server、desktop

一、真机还原命令：优先还原classroom

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server[root@room9pc01 ~]# rht-vmctl reset desktop

二、进入 虚拟机server 与 虚拟机desktop,检测Yum是否可用

# yum clean all #清空缓存

# yum repolist #列出仓库信息

###################################################

Set UID

• 附加在属主的 x 位上

– 属主的权限标识会变为 s

– 适用于可执行文件,Set UID可以让使用者具有文件属主的身份及部分权限

[root@server0 ~]# cp /usr/bin/touch /usr/bin/xixich

[root@server0 ~]# chmod u+s /usr/bin/xixich

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

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

[root@server0 ~]# su - student

[student@server0 ~]$ /usr/bin/touch 1.txt

[student@server0 ~]$ /usr/bin/xixich 2.txt

[student@server0 ~]$ ls -l

[student@server0 ~]$ exit

#################################################

Sticky Bit

• 附加在其他人的 x 位上

– 其他人的权限标识会变为 t

– 适用于开放 w 权限的目录,可以阻止用户滥用 w 写入

权限(禁止操作别人的文档)

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

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

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

[root@server0 ~]# chmod o+t /home/public

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

###############################################

查找文本内容

• 根据字符串模式提取文本行

– grep [选项] '匹配模式' 文本文件...

– 命令行 | grep [选项] '匹配模式'

• 常用命令选项

– -v,取反匹配

– -i,忽略大小写

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

[root@server0 ~]# grep -v root /etc/passwd

[root@server0 ~]# grep -i ROOT /etc/passwd

• 常用的匹配模式

– word 包含字符串word

– ^word 以字符串word开头

– word$ 以字符串word结尾

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

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

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

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

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

[root@server0 ~]# grep ^127 /etc/hosts

显示/etc/login.defs配置文件有效信息（去除空行 去除注释行）

# grep -v ^# /etc/login.defs

# grep -v ^# /etc/login.defs | grep -v ^$

# grep -v ^# /etc/login.defs | grep -v ^$ > /opt/1.txt

#################################################

查找文件

• 根据预设的条件递归查找对应的文件

– find [目录] [条件1] [-a|-o] [条件2] ...

– 常用条件表示:

-type 类型(f文本文件、d目录、l快捷方式)

-name "文档名称"

-size +|-文件大小(k、M、G)

-user 用户名

# find /boot/ -type d #查找/boot目录下是目录的

# find /boot/ -type f #查找/boot目录下是文件的

# find /boot/ -type l #查找/boot目录下是快捷方式

# ls /boot/grub/menu.lst

# ls -l /boot/grub/menu.lst

统计命令

[root@server0 ~]# wc -l /etc/passwd #统计行数

统计/etc目录下以".conf"结尾的有多少个？

# find /etc/ -name "\*.conf"

# find /etc/ -name "\*.conf" | wc -l

# find /etc/ -name "\*tab"

# find /etc/ -name "\*tab" | wc -l

# find /dev/ -name "tty\*"

# find /dev/ -name "tty\*" | wc -l

[root@server0 ~]# mkdir /root/nsd1808

[root@server0 ~]# touch /root/nsd01.txt

[root@server0 ~]# touch /root/nsd02.txt

# find /root/ -name "nsd\*"

# find /root/ -name "nsd\*" -type f #默认两个条件都满足

# find /root/ -name "nsd\*" -type d

#############################################

-size +10M #大于10M

-size -10M #小于10M

k(小写) M(大写) G(大写)

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

[root@server0 ~]# ls -lh /boot/initr\*

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

##############################################

-user 用户名 #按照文档的所有者

/proc/:所占用的空间不是硬盘，而是内存

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

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

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

• 根据名称查找,忽略大小写

– -iname

[root@server0 /]# find /etc/ -iname "PASSWD"

[root@server0 /]# find /etc/ -name "PASSWD"

• 根据账号名称或所属组

– -user、-group

[root@server0 /]# find /home/ -group student

• 限制目录查找的深度(最大层数)

– -maxdepth

# find /etc/ -maxdepth 1 -name "\*.conf"

# find /etc/ -maxdepth 2 -name "\*.conf"

# find /etc/ -maxdepth 3 -name "\*.conf"

• 使用find命令的 -exec 操作

– find .. .. -exec 处理命令 {} \;

– 优势:以 {} 代替每一个结果,逐个处理,遇 \; 结束

# rm -rf /opt/\*

# find /etc/ -name "\*tab"

# find /etc/ -name "\*tab" -exec cp {} /mnt/ \;

# ls /mnt/

# find /boot -size +10M

# find /boot -size +10M -exec cp {} /opt/ \;

# ls /opt

• 使用find命令完成以下任务

– 找出所有用户 student 拥有的文件

– 把它们拷贝到 /root/findfiles/ 文件夹中

# mkdir /root/findfiles

# find / -user student -type f

# find / -user student -type f

-exec cp {} /root/findfiles/ \;

# ls -A /root/findfiles/

############################################

• 根据文件修改时间，所有的时间都是过去时间

-mtime +10 #过去的 10天之前修改和创建的文档

-mtime -10 #过去的 10天之内修改和创建的文档

var:存放经常变化的数据

[root@server0 /]# find /var/log/ -mtime -10

[root@server0 /]# find /var/log/ -mtime +10

[root@server0 /]# find /var/log/ -mtime +1000

##############################################

分 时 日 月 周 执行动作（提高执行的可靠性，书写绝对路径）

[root@server0 /]# which find #查询命令所对应的程序

首先激活网卡，再去修改DNS服务器配置文件

使用教学虚拟机

• 每个学员机上有三台预先配置好的虚拟机

– server —— 作为练习用服务器

– desktop —— 作为练习用客户机

– classroom —— 提供网关/DNS/软件素材/DHCP服务等资源

优先开机classroom，再去开启server、desktop

一、真机还原命令：优先还原classroom

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server[root@room9pc01 ~]# rht-vmctl reset desktop

二、进入 虚拟机server 与 虚拟机desktop,检测Yum是否可用

# yum clean all #清空缓存

# yum repolist #列出仓库信息

###################################################

扇区的大小为512字节

分区规划及使用

一块硬盘的“艺术”之旅

• 识别硬盘 => 分区规划=> 格式化 => 挂载使用

毛坯楼层 => 打隔断 => 装修 => 入驻

一、识别硬盘

[root@server0 ~]# lsblk

NAME SIZE TYPE MOUNTPOINT

vda 10G disk

└─vda1 10G part /

vdb 10G disk

二、分区规划

• MBR/msdos 分区模式

– 分区类型：主分区、扩展分区、逻辑分区

– 1~4个主分区,或者 3个主分区+1个扩展分区(n个逻辑分区)

– 最大支持容量为 2.2TB 的磁盘

– 扩展分区不能格式化

/dev/sda5：SCSI接口设备第一块硬盘的第一个逻辑分区

• GPT分区模式，最大到18EB

1EB=1000PB

1PB=1000TB

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

n 创建新的分区---->回车---->回车--->回车--->在last结束时 +2G

p 查看分区表

n 创建新的分区---->回车---->回车--->回车--->在last结束时 +1G

d 删除分区

w 保存并退出

[root@server0 ~]# lsblk

[root@server0 ~]# ls /dev/vdb[1-2]

三、格式化，赋予空间 文件系统（规则）

# mkfs.ext4 /dev/vdb1 #格式化ext4文件系统

# blkid /dev/vdb1 #查看文件系统类型

# mkfs.xfs /dev/vdb2 #格式化xfs文件系统

# blkid /dev/vdb2 #查看文件系统类型

四、挂载使用分区

[root@server0 ~]# mkdir /mypart1 /mypart2

[root@server0 ~]# mount /dev/vdb1 /mypart1

[root@server0 ~]# mount /dev/vdb2 /mypart2

[root@server0 ~]# df -h #显示已经挂载的设备使用情况

五、开机自动挂载

配置/etc/fstab

设备路径 挂载点 类型 参数 备份标记 检测顺序

[root@server0 ~]# tail -2 /etc/fstab

/dev/vdb1 /mypart1 ext4 defaults 0 0

/dev/vdb2 /mypart2 xfs defaults 0 0

[root@server0 ~]# umount /mypart1

[root@server0 ~]# umount /mypart2

[root@server0 ~]# df -h

[root@server0 ~]# mount -a

检测/etc/fstab开机自动挂载配置文件,格式是否正确

检测/etc/fstab中,书写完成,但当前没有挂载的设备,进行挂载

[root@server0 ~]# df -h

六、继续分区

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

p 查看分区表

n 创建主分区----->回车----->回车---->回车----->在last结束时 +1G

p 查看分区表

n 创建扩展分区

----->回车---->起始回车----->结束回车 将所有空间给扩展分区

p 查看分区表

n 创建逻辑分区----->起始回车------>结束+1G

n 创建逻辑分区----->起始回车------>结束+1G

p 查看分区表

w 保存并退出

[root@server0 ~]# lsblk

[root@server0 ~]# partprobe #刷新 新的分区表

[root@server0 ~]# lsblk

############################################

总结：

1.识别硬盘 lsblk

2.划分分区 fdisk

3.刷新分区表 partprobe

4.格式化 mkfs.ext4 mkfs.xfs---->blkid查看文件系统

5.利用mount挂载使用 ----> /etc/fstab完成开机自动挂载

################################################

为虚拟机添加新的硬盘60G

1.关闭虚拟机server

2.利用图形 为虚拟机server添加60G硬盘

开机之后：

[root@room9pc01 ~]# gos

[root@server0 ~]# lsblk

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

################################################

新加的60G硬盘进行分区

三个主分区10G，划分扩展分区， 两个逻辑分区10G

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

p 查看分区表

n 创建主分区----->回车----->回车---->回车----->在last结束时 +10G

连续创建3个10G主分区

......

n 创建扩展分区

----->回车---->起始回车----->结束回车 将所有空间给扩展分区

p 查看分区表

n 创建逻辑分区----->起始回车------>结束+10G

n 创建逻辑分区----->起始回车------>结束+10G

p 查看分区表

w 保存并退出

[root@server0 ~]# lsblk

###############################################

LVM逻辑卷的管理

作用：1.可以整合分散的空间 2.容量大小可以扩大

– 零散空闲存储 ---->整合的虚拟磁盘 ---->虚拟的分区

物理卷 PV 卷组 VG 逻辑卷LV

将众多的物理卷( PV),组成卷组(VG),再从卷组中划分出逻辑卷(LV)

###################################################

一、创建逻辑卷

successfully:成功

1.直接创建卷组

格式：vgcreate 卷组的名 设备路径1 设备路径2 .......

# vgcreate systemvg /dev/vdc1 /dev/vdc2

# vgs #查看卷组信息

# pvs #查看物理卷信息

2.通过卷组划分逻辑卷

格式：lvcreate -n 逻辑卷名 -L 逻辑卷大小 卷组名

# lvcreate -n mylv -L 16G systemvg

# lvs #查看逻辑卷信息

# vgs

# ls -l /dev/systemvg/mylv

# mkfs.ext4 /dev/systemvg/mylv #格式化ext4

# blkid /dev/systemvg/mylv #查看设备文件系统

# vim /etc/fstab

/dev/systemvg/mylv /lvm ext4 defaults 0 0

# mount -a #检测是否书写正确

# df -h #查看挂载情况

#################################################

逻辑卷的扩展

一、卷组有足够的剩余空间

1.扩展空间的大小

[root@server0 ~]# vgs

[root@server0 ~]# lvextend -L 18G /dev/systemvg/mylv

[root@server0 ~]# lvs

2.扩展文件系统的大小

resize2fs:扩展ext4文件系统

xfs\_growfs:扩展xfs文件系统

[root@server0 ~]# df -h

[root@server0 ~]# resize2fs /dev/systemvg/mylv

[root@server0 ~]# df -h

二、卷组没有足够的剩余空间

1.扩展卷组

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

[root@server0 ~]# vgs

2.扩展空间的大小

[root@server0 ~]# vgs

[root@server0 ~]# lvextend -L 25G /dev/systemvg/mylv

[root@server0 ~]# lvs

3.扩展文件系统的大小

resize2fs:扩展ext4文件系统

[root@server0 ~]# df -h

[root@server0 ~]# resize2fs /dev/systemvg/mylv

[root@server0 ~]# df -h

#################################################

了解内容：缩减空间

ext4文件系统支持缩减

xfs文件系统不支持缩减

#################################################

卷组划分空间的单位：PE 默认4M

1.查看卷组PE大小

[root@server0 ~]# vgdisplay systemvg

PE Size 4.00 MiB

2.修改方式

卷组已经存在，修改PE的大小

# vgchange -s PE大小 卷组名

创建卷组的时候设置PE大小

# vgcreate -s PE大小 卷组名 设备路径

创建逻辑卷使用PE的个数创建

# lvcreate -l PE的个数 -n 逻辑卷 卷组名

2. 基于刚建立的 2000MiB 分区构建新的 LVM 存储

– 新的逻辑卷命名为 database,大小为50个物理扩展单元

(Physical Extent),属于 datastore 卷组

– 在 datastore 卷组中的所有逻辑卷,其物理扩展单元

(Physical Extent)的大小为16MiB

– 使用 EXT3 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

# vgchange -s 16M systemvg

# lvcreate -L 800M -n database systemvg

# lvcreate -l 50 -n lvtest02 systemvg

# lvs

###############################################

1.查看卷组PE大小

[root@server0 ~]# vgdisplay systemvg

[root@server0 ~]# vgchange -s 1M systemvg

[root@server0 ~]# vgdisplay systemvg

2.利用PE的个数创建逻辑卷

[root@server0 ~]# lvcreate -l 80 -n lvtest systemvg

[root@server0 ~]# lvs

#################################################

逻辑卷的删除

1.先删除逻辑卷本身（先卸载）

2.再删除卷组

3.最后删除物理卷（可选）

# lvremove /dev/systemvg/mylv #删除逻辑卷

# umount /lvm

# lvremove /dev/systemvg/mylv

# lvs

# lvremove /dev/systemvg/lvtest

# lvs

# vgremove systemvg #删除卷组

# vgs

# pvremove /dev/vdc[1-3] #删除物理卷

# pvs

################################################

使用教学虚拟机

• 每个学员机上有三台预先配置好的虚拟机

– server —— 作为练习用服务器

– desktop —— 作为练习用客户机

– classroom —— 提供网关/DNS/软件素材/DHCP服务等资源

优先开机classroom，再去开启server、desktop

一、真机还原命令：优先还原classroom

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server[root@room9pc01 ~]# rht-vmctl reset desktop

二、进入 虚拟机server 与 虚拟机desktop,检测Yum是否可用

# yum clean all #清空缓存

# yum repolist #列出仓库信息

###################################################

Shell脚本基础

脚本：一个可以运行文件，可以实现某种功能

案例：

请利用Shell语言实现，运行一个文件，可以再命令行

输出:hello world

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

echo hello world

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

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

################################################

规范Shell脚本的一般组成

• #! 环境声明(以下的代码有那个程序进行翻译)

• # 注释文本

• 可执行代码

案例：

1)输出当前红帽系统的版本信息

2)输出当前使用的内核版本

3)输出当前系统的主机名

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

#!/bin/bash

echo hello world

hostname

cat /etc/redhat-release

uname -r

ifconfig | head -2 | tail -1

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

############################################

重定向输出

>：只收集前面命令的正确输出信息，写入文本文件中

2>：只收集前面命令的错误输出信息，写入文本文件中

&>：收集前面命令的正确与错误输出信息，写入文本文件中

# echo 123 > /opt/1.txt

# cat /opt/1.txt /2.txt #有正确信息与错误信息的输出

# cat /opt/1.txt /2.txt > /mnt/a.txt

# cat /mnt/a.txt

# cat /opt/1.txt /2.txt 2> /mnt/a.txt

# cat /mnt/a.txt

# cat /opt/1.txt /2.txt &> /mnt/a.txt

# cat /mnt/a.txt

#################################################

案例:

书写创建用户zhangsan，并设置密码为123脚本

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

#!/bin/bash

abc=nsd12

useradd $abc &> /dev/null

echo $abc创建成功

echo 123 | passwd --stdin $abc &> /dev/null

echo $abc密码设置成功

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

黑洞设备：/dev/null 专用于收集无用的信息

为了增加脚本的灵活度，适用多变的环境------>变量

变量：类似于容器， 以不变的名称，存储可以变化的值

变量名=储存的值 #为变量赋值

#################################################

read :

1.产生交互

2.记录用户在键盘上的输入

3.将记录的信息赋值给变量

-p ：屏幕的提示信息

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

#!/bin/bash

read -p '请输入您要创建的用户:' abc

useradd $abc &> /dev/null

echo $abc创建成功

echo 123 | passwd --stdin $abc &> /dev/null

echo $abc密码设置成功

################################################

• 以不变的名称存放的可能会变化的值

– 变量名=变量值

– 方便以固定名称重复使用某个值

– 提高对任务需求、运行环境变化的适应能力

• 设置变量时的注意事项

– 若指定的变量名已存在,相当于为此变量重新赋值

– 等号两边不要有空格

– 变量名只能由字母/数字/下划线组成,区分大小写

– 变量名不能以数字开头,不要使用关键字和特殊字符

• 基本格式

– 引用变量值:$变量名

– 查看变量值:echo $变量名、 echo ${变量名}

[root@server0 ~]# a=rhel

[root@server0 ~]# echo $a

rhel

[root@server0 ~]# echo $a7

[root@server0 ~]# echo ${a}7

#################################################

变量的种类

一、环境变量：变量名一般都大写，由系统定义完成，用户直接使用

USER=永远储存当前系统登陆的用户名

[root@server0 ~]# echo $USER

[root@server0 ~]# su - dc

[dc@server0 ~]$ echo $USER

[dc@server0 ~]$ exit

logout

[root@server0 ~]# su - student

[student@server0 ~]$ echo $USER

[student@server0 ~]$ exit

logout

[root@server0 ~]#

位置变量:由系统定义完成，用户直接使用,非交互式

• 在执行脚本时,提供的命令行参数

– 表示为 $n,n为序号

– $1、$2、$3 、$4.. .. ${10}、${11}、.. ..

案例：利用位置变量实现

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

#!/bin/bash

useradd $1 &> /dev/null

echo $1创建成功

echo 123 | passwd --stdin $1 &> /dev/null

echo $1密码设置成功

[root@server0 ~]# /root/user.sh nsd20

[root@server0 ~]# vim /root/1.sh

#!/bin/bash

cat -n $1 | head -$2

[root@server0 ~]# /root/1.sh /etc/passwd 3

##############################################

预定义变量:由系统定义完成，用户直接使用

• 用来保存脚本程序的执行信息

– 直接使用这些变量

– 不能直接为这些变量赋值

$# 已加载的位置变量的个数(计算用户输入位置变量的个数)

$\* 所有位置变量的值

$? 程序退出后的状态值,0表示正常,其他值异常

[root@server0 ~]# vim /root/2.sh

#!/bin/bash

echo $1

echo $2

echo $3

echo $# #输出一共有多少个参数（求和）

echo $\* #输出所有位置变量的值

[root@server0 ~]# /root/2.sh haha xixi hehe lele

[root@server0 ~]# vim /root/1.sh

#!/bin/bash

cat -n $1 | head -$2

[root@server0 ~]# /root/1.sh /etc/passwd 3

################################################

自定义变量:用户自主设置、修改及使用

###############################################

单引号 ' ' : 取消所有特殊字符的含义

###############################################

常用的测试选项

• 检查文档状态

-e:文档存在为真

-d:文档存在,且为目录才为真

-f:文档存在,且为文件才为真

-r:文档存在,且具备读取权限才为真

-w:文档存在,且具备写入权限才为真

-x:文档存在,且具备执行权限才为真

[root@server0 ~]# [ -d /etc/passwd ]

[root@server0 ~]# echo $?

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

[root@server0 ~]# [ -e /home ]

[root@server0 ~]# [ -f /home ]

[root@server0 ~]# [ -f /etc/passwd ]

• 比较整数大小

-gt：大于

-ge：大于等于

-eq：等于

-ne：不等于

-lt：小于

-le：小于等于

[root@server0 ~]# [ 1 -gt 2 ]

[root@server0 ~]# echo $?

[root@server0 ~]# [ 1 -eq 2 ]

[root@server0 ~]# echo $?

[root@server0 ~]# [ 1 -ge 1 ]

[root@server0 ~]# echo $?

• 字符串比对

==:两个字符串相等为真

!=:两个字符串不相等为真

[root@server0 ~]# [ student == root ]

[root@server0 ~]# echo $?

[root@server0 ~]# [ student != root ]

[root@server0 ~]# echo $?

[root@server0 ~]# [ $USER == root ]

[root@server0 ~]# echo $?

################################################

if双分支处理

if [条件测试];then

命令序列xx

else

命令序列yy

fi

案例：

利用read实现交互

用户输入一个用户名，脚本判断用户所输入的用户，是否存在

如果存在，输出用户存在

如果不存在，输出用户不存在

[root@server0 ~]# vim /root/if01.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 #判断上一条命令是否成功

echo $user用户已存在

else

echo $user用户不存在

fi

案例：

利用位置变量实现命令行传递参数

参数为IP地址，脚本判断用户所输入IP地址，是否本机可以ping通

如果可以ping，输出 可以通信

如果不可以ping，输出 不可以通信

[root@server0 ~]# vim /root/if02.sh

#!/bin/bash

ping -c 2 $1 &> /dev/null

if [ $? -eq 0 ];then

echo $1 本机可以通信

else

echo $1 本机不可以通信

fi

[root@server0 ~]# /root/if02.sh 192.168.1.1

优化：

[root@server0 ~]# vim /root/if02.sh

#!/bin/bash

if [ $# -eq 0 ];then #判断用户是否输入参数

echo 格式错误，请在脚本运行时添加参数

echo 参数为IP地址

else

ping -c 2 $1 &> /dev/null

if [ $? -eq 0 ];then

echo $1 本机可以通信

else

echo $1 本机不可以通信

fi

fi

#############################################

for循环结构

循环:让计算机重复执行一个操作

for 变量名 in 值列表

do

命令序列

done

for a in zhangsan lisi wangwu dc

do

useradd $a

echo 123 | passwd --stdin $a

done

造数： {起始数值..结束数值}

[root@server0 /]# cat /root/for01.sh

#!/bin/bash

for a in {1..20}

do

useradd stu$a &> /dev/null

echo stu$a创建成功

done

[root@server0 /]# /root/for01.sh

###############################################

$( ) 或 反撇号` `

:将命令的输出，作为另一个命令的参数，参与下一个命令的运行

[root@server0 /]# date +%F

2018-09-12

[root@server0 mnt]# cd /mnt

[root@server0 mnt]# mkdir nsd-`date +%F`

[root@server0 mnt]# ls

[root@server0 mnt]# mkdir nsd-`date +%F`

[root@server0 mnt]# ls

[root@server0 mnt]# mkdir $(hostname)-$(date +%F)

[root@server0 mnt]# ls

#################################################

扩展案例：

[root@server0 /]# vim /root/user.txt

yuanxiao

qingming

duanwu

wuyi

liuyi

wusi

......

请利用for循环：

该文件/root/user.txt内容，每一行为一个用户的名称

创建/root/user.txt文件中所有用户

################################################

################################################

if多分支处理

if [条件测试1];then

命令序列xx

elif [条件测试2];then

命令序列yy

elif [条件测试3];then

命令序列cc

......

else

命令序列zz

fi

案例:

利用read交互，让用户输入考试成绩

进行如下判断

如果成绩大于90 则 输出 优秀

如果成绩大于80 则 输出 良好

如果成绩大于70 则 输出 一般

如果成绩大于60 则 输出 合格

以上条件均不满足 ， 则 输出 再牛的肖邦，也弹不出哥的悲伤

[root@server0 ~]# vim /root/if03.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 再牛的肖邦，也弹不出哥的悲伤

fi

###############################################

使用教学虚拟机

• 每个学员机上有三台预先配置好的虚拟机

– server —— 作为练习用服务器

– desktop —— 作为练习用客户机

– classroom —— 提供网关/DNS/软件素材/DHCP服务等资源

优先开机classroom，再去开启server、desktop

一、真机还原命令：优先还原classroom

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server[root@room9pc01 ~]# rht-vmctl reset desktop

二、进入 虚拟机server 与 虚拟机desktop,检测Yum是否可用

# yum clean all #清空缓存

# yum repolist #列出仓库信息

#################################################

案例4:编写一个判断脚本

在 server0 上创建 /root/foo.sh 脚本

1)当运行/root/foo.sh redhat,输出为fedora

2)当运行/root/foo.sh fedora,输出为redhat

3)当没有任何参数或者参数不是 redhat 或者fedora时,

其错误输出产生以下信息: /root/foo.sh redhat|fedora

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

#!/bin/bash

if [ $# -eq 0 ];then

echo '/root/foo.sh redhat|fedora' >&2 #变成错误输出

exit 36 #程序退出返回值

elif [ $1 == redhat ];then

echo fedora

elif [ $1 == fedora ];then

echo redhat

else

echo '/root/foo.sh redhat|fedora' >&2 #变成错误输出

exit 25 #程序退出返回值

fi

#################################################

扩展案例：

[root@server0 /]# vim /root/user.txt

yuanxiao

qingming

duanwu

wuyi

liuyi

wusi

......

请利用for循环：

该文件/root/user.txt内容，每一行为一个用户的名称

创建/root/user.txt文件中所有用户

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

#!/bin/bash

for a in `cat /root/user.txt`

do

useradd $a &> /dev/null

echo $a创建成功

done

################################################

案例5:编写一个批量添加用户脚本

在 server0 上创建 /root/batchusers 脚本

1)此脚本要求提供用户名列表文件作为参数

2)如果没有提供参数,此脚本应该给出提示

Usage: /root/batchusers,退出并返回相应值

3)如果提供一个不存在的文件,此脚本应该给出提

示 Input file not found,退出并返回相应值

4)新用户的登录Shell为 /bin/false,无需设置密码

5)用户列表测试文件:

[root@server0 ~]# cat /root/userlist

zhangsan

lisi

harry

natasha

dc

tc

jack

[root@server0 ~]# cat /root/batchusers

#!/bin/bash

if [ $# -eq 0 ];then

echo 'Usage: /root/batchusers' >&2

exit 2

elif [ -f $1 ];then

for i in `cat $1`

do

useradd -s /bin/false $i &> /dev/null

echo $i创建成功

done

else

echo 'Input file not found' >&2

exit 3

fi

###############################################

系统安全保护

• Security-Enhanced Linux

– 美国NSA国家安全局主导开发,一套增强Linux系统安

全的强制访问控制体系

– 集成到Linux内核(2.6及以上)中运行

– RHEL7基于SELinux体系针对用户、进程、目录和文件

提供了预设的保护策略,以及管理工具

• SELinux的运行模式

– enforcing(强制)、permissive(宽松)

– disabled(彻底禁用)

任何一种运行模式，变成disabled(彻底禁用)都要经历重起系统

切换运行模式

– 临时切换:setenforce 1|0

– 固定配置:/etc/selinux/config 文件

修改两台虚拟机 Server与 Desktop

[root@server0 ~]# getenforce

Enforcing #强制模式

[root@server0 ~]# setenforce 0 #修改当前状态为宽松模式

[root@server0 ~]# getenforce

Permissive

[root@server0 ~]# vim /etc/selinux/config

SELINUX=permissive

###############################################

虚拟机Server：搭建基本Web服务

1.安装httpd(Apache)软件包（服务端软件）

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

2.重起程序（服务）

[root@server0 ~]# systemctl restart httpd

[root@server0 ~]# systemctl enable httpd #开机自起

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

3.验证本机访问测试

[root@server0 ~]# firefox 172.25.0.11

4.书写一个页面

网页文件默认存放路径:/var/www/html/

默认网页文件的名字:index.html

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

<marquee><font color=red><h1>NSD1808 阳光明媚

滚动效果 字体颜色 最大字体

[root@server0 ~]# firefox 172.25.0.11

###############################################

虚拟机Server：搭建FTP服务:文件数据传输

1.安装软件包vsftpd

2.重起vsftpd服务

[root@server0 ~]# systemctl restart vsftpd

[root@server0 ~]# systemctl restart vsftpd

默认共享数据路径：/var/ftp

3.本机测试

[root@server0 ~]# firefox ftp://172.25.0.11

################################################

防火墙： 隔离作用

硬件防火墙

软件防火墙

firewalld服务

• 系统服务:firewalld

• 管理工具:firewall-cmd、firewall-config

[root@server0 ~]# systemctl restart firewalld

[root@server0 ~]# firewall-config &

##############################################

• 根据所在的网络场所区分,预设保护规则集

– public:仅允许访问本机的ssh、ping、dhcp服务

– trusted:允许任何访问

– block:阻塞任何来访请求，明确拒绝

– drop:丢弃任何来访的数据包，直接丢弃

数据包： 数据 源ip地址 目标IP地址

防火墙进入区域的规则：匹配及停止

1.查看数据包种源ip地址，再查看所有的区域中，是否有该源IP地址的规则，那个区域有，则进入该区域

2.进入默认区域为public

###############################################

默认区域的修改

虚拟机Server：

# firewall-cmd --get-default-zone #查看默认区域

虚拟机Desktop：

# ping 172.25.0.11 #可以通信

虚拟机Server：

# firewall-cmd --set-default-zone=block #修改默认区域

# firewall-cmd --get-default-zone

虚拟机Desktop：

# ping 172.25.0.11 #不可以通信，有回应

虚拟机Server：

# firewall-cmd --set-default-zone=drop #修改默认区域

# firewall-cmd --get-default-zone

虚拟机Desktop：

# ping 172.25.0.11 #不可以通信，没有回应

#################################################

添加服务

虚拟机Server：

# firewall-cmd --set-default-zone=public

# firewall-cmd --get-default-zone

# firewall-cmd --zone=public --list-all

列出 public区域中规则

# firewall-cmd --zone=public --add-service=http

为 public区域,添加允许的协议http

# firewall-cmd --zone=public --list-all

虚拟机Desktop：

# firefox 172.25.0.11 #访问成功

# firefox ftp://172.25.0.11 #访问失败

################################################

– 永久(permanent)

虚拟机Server

# firewall-cmd --reload #重新加载防火墙配置

# firewall-cmd --zone=public --list-all

# firewall-cmd --permanent --zone=public

--add-service=http

# firewall-cmd --zone=public --list-all

# firewall-cmd --reload

虚拟机Desktop

# firefox 172.25.0.11

#############################################

添加源IP规则设置

虚拟机Server：单独拒绝172.25.0.10访问

# firewall-cmd --zone=block --add-source=172.25.0.10

##############################################

数据包： 数据 源ip地址 目标IP地址 目标端口号 80

端口号： 编号 标识程序及应用、服务

互联网常见协议：

http：超文本传输协议 默认端口号:80

https：安全的超文本传输协议 默认端口号:443

FTP：文件传输协议 默认端口号:21

DNS：域名解析协议 默认端口号:53

SMTP：用户发邮件协议 默认端口号:25

pop3：用户收邮件协议 默认端口号:110

telnet：运程管理协议 默认端口号:23

TFTP：简单的文件传输协议 默认端口号:69

SNMP：网络管理协议 默认端口号:161

协议的端口可以有root改变，也可以具备多个端口

################################################

实现本机的端口映射

• 本地应用的端口重定向(端口1 --> 端口2)

– 从客户机访问 端口1 的请求,自动映射到本机 端口2

– 比如,访问以下两个地址可以看到相同的页面:

客户端访问172.25.0.11:5423 ----> 172.25.0.11:80

虚拟机Server

# firewall-cmd --permanent --zone=public

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

# firewall-cmd --reload

# firewall-cmd --zone=public --list-all

虚拟机Desktop

# firefox 172.25.0.11:5423

###############################################

配置IPv6地址

IPv4地址：32个二进制组成,利用点分隔,用4个十进制数表示

IPv6地址：128个二进制组成,利用冒号分隔8部分,

最终每一部分利用4个16进制数表示

每段内连续的前置 0 可省略、连续的多个 : 可简化为 ::

例如: 2003:ac18:0000:0000:0000:0000:0000:0305

2003:ac18::305

虚拟机Server0

# nmcli connection modify 'System eth0'

ipv6.method manual ipv6.addresses 2003:ac18::305/64

connection.autoconnect yes

# nmcli connection up 'System eth0'

# ifconfig | head -4

# ping6 2003:ac18::305

##############################################

配置聚合连接(链路聚合)

备份网卡 eth1 eth2

虚拟网卡team0 192.168.1.1

1.创建虚拟网卡team0 参考 man teamd.conf--》/example

# nmcli connection add type team

con-name team0 ifname team0

autoconnect yes

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

# nmcli connection 添加 类型为 team（组队）

配置文件名 team0 网卡名 team0

每次开机自动启用

工作模式为 热备份

# ifconfig

# nmcli connection delete team0 #删除team0配置

2.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

# nmcli connection 添加 类型 team-成员

配置文件名为 team0-2 网卡名 eth2 主设备 为team0

# nmcli connection delete team0-1

# nmcli connection delete team0-2

3. 为虚拟网卡team0配置IP地址

# nmcli connection modify team0

ipv4.method manual

ipv4.addresses 192.168.1.1/24

connection.autoconnect yes

4.激活配置

[root@server0 ~]# nmcli connection up team0

[root@server0 ~]# nmcli connection up team0-1

[root@server0 ~]# nmcli connection up team0-2

5.专用于查看链路聚合的命令

[root@server0 ~]# ifconfig eth1 down

[root@server0 ~]# teamdctl team0 state

使用教学虚拟机

• 每个学员机上有三台预先配置好的虚拟机

– server —— 作为练习用服务器

– desktop —— 作为练习用客户机

– classroom —— 提供网关/DNS/软件素材/DHCP服务等资源

优先开机classroom，再去开启server、desktop

一、真机还原命令：优先还原classroom

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server

[root@room9pc01 ~]# rht-vmctl reset desktop

二、进入 虚拟机server 与 虚拟机desktop,检测Yum是否可用

# yum clean all #清空缓存

# yum repolist #列出仓库信息

#################################################

案例3：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

id $1 &> /dev/null

if [ $? -eq 0 ];then

id $1

else

useradd $1

echo $1创建成功

fi

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

#!/bin/bash

useradd $1 &> /dev/null

if [ $? -eq 0 ];then

echo $1创建成功

else

id $1

fi

###############################################

练习搭建Yum仓库：

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

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

[rhel7]

name=rhel7.0

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

enabled=1

gpgcheck=0

[root@server0 ~]# yum clean all

[root@server0 ~]# yum repolist

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

##############################################

环境的初步设置：修改防火墙默认区域为允许所有

虚拟机Server

# firewall-cmd --set-default-zone=trusted

虚拟机Desktop

# firewall-cmd --set-default-zone=trusted

###############################################

Samba服务基础，Linux与Windows跨平台的共享

• Samba 软件项目

– 用途:为客户机提供共享使用的文件夹

– 协议:SMB(TCP 139 用户验证)、CIFS(TCP 445 传输数据)

• 所需软件包:samba

• 系统服务:smb

• Samba用户 —— 专用来访问共享文件夹的用户

– 采用独立设置的密码

– 但需要提前建立同名的系统用户(可以不设密码）

• 使用 pdbedit 管理工具

– 添加用户:pdbedit -a 用户名

– 查询用户:pdbedit -L [用户名]

– 删除用户:pdbedit -x 用户名

• 修改 /etc/samba/smb.conf

[自定共享名]

path = 文件夹绝对路径

; public = no|yes //默认no

; browseable = yes|no //默认yes

; read only = yes|no //默认yes

; write list = 用户1 .. .. //默认无

; valid users = 用户1 .. .. //默认任何用户

; hosts allow = 客户机地址 .. ..

; hosts deny = 客户机地址 .. ..

##################################################

搭建Samba共享

服务端虚拟机Server：

1.安装软件包samba

2.创建samba共享帐号

# useradd -s /sbin/nologin harry

# useradd -s /sbin/nologin kenji

# useradd -s /sbin/nologin chihiro

# pdbedit -a harry #将harry添加为samba共享帐号

# pdbedit -a kenji

# pdbedit -a chihiro

# pdbedit -L #列出所有的Samba共享帐号

3.创建共享目录

[root@server0 ~]# mkdir /common

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

[root@server0 ~]# ls /common/

4.修改配置文件/etc/samba/smb.conf

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

# vim命令模式 按 G（大写）

此服务器必须是 STAFF 工作组的一个成员

workgroup = STAFF

[common] #共享名

path = /common #共享实际路径

5.重起smb服务

[root@server0 ~]# systemctl restart smb

[root@server0 ~]# systemctl enable smb

6.SELinux布尔值（服务功能的开关 on 或 off）

– 需要加 -P 选项才能实现永久设置

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

[root@server0 ~]# setsebool samba\_export\_all\_ro on

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

客户端：虚拟机Desktop0

1.所需软件包samba-client（客户端软件）

[root@desktop0 ~]# yum -y install samba-client

2.利用smbclient访问共享

# smbclient -L 172.25.0.11 #查看对方有哪些共享

Enter root's password: #直接回车

# smbclient -U harry //172.25.0.11/common

Enter harry's password: #输入harry的共享密码

Domain=[STAFF] OS=[Unix] Server=[Samba 4.1.1]

smb: \>

#############################################

客户端访问服务端资源：

1.防火墙策略

2.服务本身的访问控制

3.安全增强SELinux

4.服务端本地目录的权限

#############################################

客户端：虚拟机Desktop

利用mount挂载，更加科学方式进行访问

1.安装软件cifs-utils(让本机支持cifs文件系统)

# yum -y install cifs-utils

2.挂载访问

# mkdir /mnt/nsd

# mount -o user=harry,pass=123 //172.25.0.11/common /mnt/nsd/

# df -h

# ls /mnt/nsd

3.开机自动挂载

\_netdev :声明网络设备

配置完所有的网络参数后，再进行挂载该设备

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

//172.25.0.11/common /mnt/nsd cifs

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

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

[root@desktop0 ~]# df -h

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h

##############################################

可以读写的Samba共享

服务端：虚拟机Server

1.建立新的共享目录

[root@server0 ~]# mkdir /devops

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

[root@server0 ~]# ls /devops/

2.修改主配置文件

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

[devops]

path = /devops

write list = chihiro

3.重起smb服务

[root@server0 ~]# systemctl restart smb

4.SELinux布尔值（服务功能的开关 on 或 off）

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

[root@server0 ~]# setsebool samba\_export\_all\_rw on

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

5.修改本地目录的权限

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

[root@server0 ~]# getfacl /devops/

客户端：虚拟机Desktop

[root@desktop0 ~]# tail -1 /etc/fstab

//172.25.0.11/devops /mnt/dev cifs

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

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h

#################################################

案例：

虚拟机Server上实现Samba共享

发布共享目录为/public,共享名 test

设置harry用户对此共享可读写

客户端虚拟机Desktop，访问虚拟机Server共享

利用harry进行验证

完成开机自动挂载到本地目录/mnt/class

虚拟机Server：

[root@server0 ~]# mkdir /public

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

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

[test]

path = /public

write list = harry

[root@server0 ~]# systemctl restart smb

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

虚拟机desktop：

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

//172.25.0.11/test /mnt/class cifs defaults,user=harry,pass=123,\_netdev 0 0

###############################################

多用户Samba共享

SMB客户端的 multiuser 挂载技术

– 客户端管理员只需要作一次挂载

– 客户端在访问挂载点时,若需要不同权限,可以临时

切换为新的共享用户(无需重新挂载)

• 使用 cifscreds 提交新的用户凭据并测试

– cifscreds add -u 共享用户名 服务器地址

挂载参数调整

– multiuser,提供对客户端多个用户身份的区分支持

– sec=ntlmssp,提供NT局域网管理安全支持

###############################################

配置NFS共享(Linux与Linux平台之间)

NFS共享概述

• Network File System,网络文件系统

– 用途:为客户机提供共享使用的文件夹

– 协议:NFS(TCP/UDP 2049)、RPC(TCP/UDP 111)

• 所需软件包:nfs-utils

• 系统服务:nfs-server

搭建基本的NFS共享

1.检测软件包是否安装

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

nfs-utils-1.3.0-0.el7.x86\_64

2.修改主配置文件 /etc/exports

– 文件夹路径 客户机地址(权限)

[root@server0 ~]# mkdir /abc #创建共享目录

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

[root@server0 ~]# ls /abc/

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

/abc \*(ro)

3.重起服务nfs-server

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

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

客户端:虚拟机Desktop

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

172.25.0.11:/abc /mnt/nsd01 nfs defaults,\_netdev 0 0

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h

#################################################

配置用户环境

alias别名设置

• 查看已设置的别名

– alias [别名名称]

• 定义新的别名

– alias 别名名称= '实际执行的命令行'

• 取消已设置的别名

– unalias [别名名称]

• 影响指定用户的 bash 解释环境

– ~/.bashrc,每次开启 bash 终端时生效

• 影响所有用户的 bash 解释环境

– /etc/bashrc,每次开启 bash 终端时生效

[root@server0 ~]# vim /root/.bashrc

alias hello='echo hello'

[root@server0 ~]# vim /home/student/.bashrc

alias hi='echo hi'

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

alias myls='ls -lhd'

新开一个终端：验证root可以执行那些别名--->hello,myls

验证student可以执行那些别名--->hi,myls

##############################################

环境变量： 变量名一般大写，由系统定义完成

PATH：存储命令程序的路径

[root@server0 ~]# echo $PATH

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin

[root@server0 ~]# vim /opt/abc.sh

#!/bin/bash

echo hello world

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

[root@server0 ~]# /opt/abc.sh

[root@server0 ~]# abc.sh #执行失败

[root@server0 ~]# cp /opt/abc.sh /usr/bin

[root@server0 ~]# abc.sh #执行成功

#############################################

使用教学虚拟机

• 每个学员机上有三台预先配置好的虚拟机

– server —— 作为练习用服务器

– desktop —— 作为练习用客户机

– classroom —— 提供网关/DNS/软件素材/DHCP服务等资源

优先开机classroom，再去开启server、desktop

一、真机还原命令：优先还原classroom

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server[root@room9pc01 ~]# rht-vmctl reset desktop

二、进入 虚拟机server 与 虚拟机desktop,检测Yum是否可用

# yum clean all #清空缓存

# yum repolist #列出仓库信息

##############################################

环境的初步设置：修改防火墙默认区域为允许所有

虚拟机Server

# firewall-cmd --set-default-zone=trusted

虚拟机Desktop

# firewall-cmd --set-default-zone=trusted

###############################################

iSCSI网络磁盘 默认端口：3260

• Internet SCSI,网际SCSI接口

– 一种基于C/S架构的虚拟磁盘技术

– 服务器提供磁盘空间,客户机连接并当成本地磁盘使用

• backstore,后端存储（大冰箱）

– 对应到服务端提供实际存储空间的设备,需要起一个

管理名称

• target,磁盘组（木质包装）

– 是客户端的访问目标,作为一个框架,由多个lun组成

• lun,逻辑单元（大冰箱纸制包装）放入（木质包装）

– 每一个lun需要关联到某一个后端存储设备,在客户端

会视为一块虚拟硬盘

大冰箱 -----> 纸制包装 -----> 木质包装

• ISCSI Qualified Name(iqn) 名称规范

– iqn.yyyy-mm.倒序域名:自定义标识

– 用来识别 target 磁盘组,也用来识别客户机身份

• 名称示例

– iqn.2016-02.com.example:server0

– iqn.2016-02.com.example:desktop0

服务端：虚拟机Server

一、划分新的分区

[root@server0 ~]# lsblk

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

划分一个5G的主分区

[root@server0 ~]# lsblk

[root@server0 ~]# ls /dev/vdb1

二、部署iSCSI共享存储

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（大冰箱----->纸制包装）

/> backstores/block create name=nsd 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（木质包装）

/> iscsi/ create iqn.2018-09.com.example:server0

/> ls

C:lun,逻辑单元（大冰箱纸制包装）放入（木质包装）

/> iscsi/iqn.2018-09.com.example:server0/tpg1/

luns create /backstores/block/nsd

/> 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.2018-09.com.example:server0/tpg1/acls

create iqn.2018-09.com.example:desktop0

/> ls

E:客户端访问本机的IP地址及端口

[root@server0 ~]# targetcli

/> iscsi/iqn.2018-09.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

客户端：虚拟机Desktop

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.2018-09.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

#################################################

数据库服务基础

什么是数据库 : 存放数据的仓库

在数据库中，有很多的库，在每一个库中会有很多的表格

一、部署mariadb数据库服务器，默认端口 3306

• RHEL7 中的 MariaDB 相关包

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

二、数据库简单的使用

[root@server0 ~]# mysql

> show databases; #显示所有的库

> create database nsd1808; #创建nsd1808库

> show databases; #显示所有的库

> drop database nsd1808; #删除nsd1808库

> show databases; #显示所有的库

> exit #退出

Bye

[root@server0 ~]# mysql

> create database nsd; #创建nsd库

> show databases; #显示所有的库

三、为数据库管理员设置密码

数据库管理员root，用户信息存储mysql库中表user

系统管理员root，用户信息存储/etc/passwd

• 为数据库账号修改密码

– mysqladmin [-u用户名] [-p[旧密码]] password '新密码'

在Linux系统命令行，为数据库管理员设置密码123

# mysqladmin -u root password '123'

# mysql -u root -p #交互式登陆

Enter password:

# mysql -u root -p123 #非交互式登陆

命令行操作已知旧密码，修改新的密码

# mysqladmin -u root -p123 password '321'

了解：/etc/my.cnf

• 禁止监听,只服务于本机

[root@server0 ~]# vim /etc/my.cnf

[mysqld]

skip-networking //跳过网络监听

.. ..

[root@server0 ~]# systemctl restart mariadb

################################################

四、数据库恢复数据

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

# mysql -u root -p123 nsd < users.sql

###############################################

五、进行表格的操作（表字段 表记录）

增 insert 删 delete 改 update 查select

[root@server0 ~]# mysql -u root -p321

MariaDB [(none)]> show databases;

MariaDB [(none)]> use nsd; #进入nsd库

查寻格式：select 表字段 from 表名

> show tables; #查看所有表格

> select \* from base; #查询base所有字段内容

> select \* from location;

> desc 表名 #查询表结构

查询内容多的表格

[root@server0 ~]# mysql -u root -p123

MariaDB [nsd]> use mysql;

MariaDB [mysql]> show tables; #查看所有表格

MariaDB [mysql]> desc user ; #查询表结构

MariaDB [mysql]> select user,host,password from user;

MariaDB [mysql]> select \* from nsd.base;

有条件的查询 where 表字段='值'

[root@server0 ~]# mysql -u root -p123

> use nsd;

> select \* from base

> select \* from base where password='123';

> select \* from base where password='tarena';

> select \* from base where name='tom' and

password='123';

> select \* from base where password='456' or

password='123';

################################################

六、数据库的授权

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

用户授权设置

– GRANT 权限列表 ON 数据库名.表名 TO 用户名@

客户机地址 IDENTIFIED BY '密码';

grant select on nsd.\* to lisi@localhost

identified by '123';

当lisi在localhost进行登陆，输入密码123，

将会获得nsd库中所有表的查询权限

MariaDB [(none)]> grant select on nsd.\* to

lisi@localhost identified by '123';

MariaDB [(none)]> select user,host,password from

mysql.user;

[root@server0 ~]# mysql -u lisi -p123

MariaDB [(none)]> exit

################################################

案例5:使用数据库查询

2. 在系统 server0 上使用数据库 nsd,并使用相

应的 SQL 查询以回答下列问题:

1)密码是 solicitous 的人的名字?

> use nsd;

> select \* from base ;

> select \* from base where password='solicitous';

> select name from base where password='solicitous';

2)有多少人的 姓名是 Barbara 同时居住在 Sunnyvale?

> select \* from base,location

where base.name='Barbara' and

location.city='Sunnyvale' and

base.id=location.id;

> select count(\*) from base,location

where base.name='Barbara' and

location.city='Sunnyvale' and

base.id=location.id;

> insert base values ('6','Barbara','123456');

> select \* from base;

> insert location values ('6','Sunnyvale');

> select \* from location;

1. 禁止空密码 root用户访问 mariadb 数据库

> use mysql;

> select user,host,password from user;

> select user,host,password from user where

password='';

> delete from user where password=''; #删除记录

> select user,host,password from user;

> flush privileges; #刷新数据所有策略

命令行测试： mysql -u root -h server0.example.com

登陆失败

###############################################

使用教学虚拟机

• 每个学员机上有三台预先配置好的虚拟机

– server —— 作为练习用服务器

– desktop —— 作为练习用客户机

– classroom —— 提供网关/DNS/软件素材/DHCP服务等资源

优先开机classroom，再去开启server、desktop

一、真机还原命令：优先还原classroom

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server[root@room9pc01 ~]# rht-vmctl reset desktop

二、进入 虚拟机server 与 虚拟机desktop,检测Yum是否可用

# yum clean all #清空缓存

# yum repolist #列出仓库信息

###############################################

HTTP服务基础

• 基于 B/S (Browser/Server)架构的网页服务

– 服务端提供网页

– 浏览器下载并显示网页

• Hyper Text Markup Language(HTML),超文本标记语言

• Hyper Text Transfer Protocol（HTTP）,超文本传输协议

HTTP默认端口 ： 80

#################################################

环境的初步设置：修改防火墙默认区域为允许所有

虚拟机Server

# firewall-cmd --set-default-zone=trusted

虚拟机Desktop

# firewall-cmd --set-default-zone=trusted

#################################################

构建独立的Web服务器

虚拟机Server

1.安装软件

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

2.书写页面文件

# echo '<h1>Hello World' > /var/www/html/index.html

# cat /var/www/html/index.html

3.重起httpd服务

# systemctl restart httpd

# systemctl enable httpd

################################################

• 提供的默认配置

– Listen:监听地址:端口(80)

– ServerName:本站点注册的DNS名称(空缺)

– DocumentRoot:网页根目录(/var/www/html)

– DirectoryIndex:起始页/首页文件名(index.html)

#################################################

– ServerName:本站点注册的DNS名称(空缺)

DNS服务器：域名解析 classroom.example.com

今日可用域名： server0.example.com

www0.example.com

webapp0.example.com

[root@desktop0 ~]# nslookup server0.example.com

[root@desktop0 ~]# nslookup www0.example.com

[root@desktop0 ~]# nslookup webapp0.example.com

###############################################

– ServerName:本站点注册的DNS名称(空缺)

虚拟机Server0：

1.修改主配置文件/etc/httpd/conf/httpd.conf

95行 ServerName server0.example.com:80

2.重起httpd服务

[root@server0 ~]# systemctl restart httpd

虚拟机Desktop0：测试访问

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

#################################################

– DocumentRoot:网页文件的根目录(/var/www/html)

虚拟机Server0：

1.修改主配置文件/etc/httpd/conf/httpd.conf

122行 DocumentRoot "/var/www/myweb"

2.常见网页文件路径，书写新的页面

# mkdir /var/www/myweb

# echo '<h1>NSD1808 Web' > /var/www/myweb/index.html

3.重起httpd服务

[root@server0 ~]# systemctl restart httpd

虚拟机Desktop0：测试访问

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

##################################################

网络路径与服务器实际路径

网络路径：在浏览器输入的路径

firefox server0.example.com---》DNS

firefox 172.25.0.11---->服务端 80---->httpd

-->主配置文件----->DocumentRoot /var/www/myweb

浏览器： server0.example.com

服务端：/var/www/myweb

浏览器： server0.example.com/abc

服务端：/var/www/myweb/abc

##############################################

虚拟机Server0：

# mkdir /var/www/myweb/abc/

# echo '<h1>wo shi abc' > /var/www/myweb/abc/index.html

虚拟机Desktop0：

# firefox server0.example.com/abc

###############################################

虚拟Web主机

作用：一台服务器，提供多个不同的Web页面

• 区分方式

– 基于域名的虚拟主机

– 基于端口的虚拟主机

– 基于IP地址的虚拟主机

• 配置文件路径

– /etc/httpd/conf/httpd.conf (主配置文件)

– /etc/httpd/conf.d/\*.conf(调用配置文件)

<VirtualHost IP地址:端口>

ServerName 此站点的DNS名称

DocumentRoot 此站点的网页根目录

</VirtualHost>

##############################################

案例：构建基于域名的虚拟主机

虚拟机Server0:

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

<VirtualHost \*:80> #本机所有IP启用虚拟Web

ServerName www0.example.com #网站的域名

DocumentRoot /var/www/nsd01 #网页文件路径

</VirtualHost>

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd02

</VirtualHost>

# mkdir /var/www/nsd01 /var/www/nsd02

# echo '<h1>百度' > /var/www/nsd01/index.html

# echo '<h1>企鹅' > /var/www/nsd02/index.html

# systemctl restart httpd

一旦使用虚拟Web主机功能，所有网站都必须用虚拟Web主机完成

[root@server0 ~]# vim /etc/httpd/conf.d/nsd.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 /var/www/myweb

</VirtualHost>

################################################

Web访问控制（DocumentRoot）

针对 DocumentRoot 网页目录的权限控制

– httpd 运行身份(用户/组):apache

服务配置

• 使用 <Directory> 配置区段

– 每个文件夹自动继承其父目录的访问权限

– 除非针对子目录有明确设置

/opt 拒绝所有人访问

/opt/test 拒绝所有人访问

/opt/class 单独书写配置允许

<Directory 目录的绝对路径>

.. ..

Require all denied|granted

Require ip IP或网段地址 .. ..

</Directory>

案例3:配置网页内容访问

在 Web 网站 http://server0.example.com 的

DocumentRoot 目录下创建一个名为 private 的子目录

1.查看DocumentRoot 路径

# cat /etc/httpd/conf.d/nsd.conf

2.创建相应目录

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

3.书写一个网页文件

# echo '<h1> wo shi private' > /var/www/myweb/private/index.html

# cat /var/www/myweb/private/index.html

4.从 server0 上,任何人都可以浏览 private 的内容,

但是从其他系统不能访问这个目录的内容

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

<Directory /var/www/myweb/private>

Require ip 172.25.0.11

</Directory>

5.重起httpd服务

客户端与服务端均做访问测试：

firefox server0.example.com/private

验证最终结果：虚拟机desktop被拒绝，虚拟机server可以访问

#################################################

案例4: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:

虚拟机Server0：

1)新建目录 /webroot,作为此站点新的网页文件目录

# mkdir /webroot

# echo '<h1>wo shi webroot' > /webroot/index.html

# cat /webroot/index.html

2)修改配置文件vim /etc/httpd/conf.d/nsd.conf

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /webroot

</VirtualHost>

3)修改配置文件，添加服务的访问控制

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

<Directory /webroot>

Require all granted #允许所有人访问

</Directory>

4)重起httpd服务

[root@server0 ~]# systemctl restart httpd

5)SELinux的安全上下文（标签起到标识作用）

[root@server0 ~]# ls -Zd /var/www/

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

• 方式1:参照标准目录,重设新目录的属性

– chcon [-R] --reference=模板目录 新目录

# chcon -R --reference=/var/www /webroot

# 将/webroot修改为/var/www一样的标签值

# ls -Zd /webroot/

##############################################

部署动态网站

动态网站的运行

• 服务端的原始网页 ≠ 浏览器访问到的网页

– 由Web服务软件接受请求,动态程序转后端模块处理

– PHP网页、Python网页、JSP网页

为站点 webapp0.example.com 配置提供动态Web内

容,要求如下:

1.下载Python网页文件，到站点 webapp0.example.com网页根目录

# cat /etc/httpd/conf.d/nsd.conf #查看网页文件路径

# cd /var/www/nsd02

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

2.方便用户的访问，网页跳转

配置字段格式： Alias 网络路径 实际本地路径

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

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd02

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

#当客户端访问网页文件根目录时，跳转到webinfo.wsgi页面

</VirtualHost>

[root@server0 /]# systemctl restart httpd

3.安装mod\_wsgi软件包,专用于翻译Python代码页面

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

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

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd02

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

</VirtualHost>

[root@server0 /]# systemctl restart httpd

网页内容显示：显示当前系统的 Unix时间戳

Unix时间戳：自1970-1-1 0:0:0到当前系统时间经历秒数

4.此虚拟主机侦听在端口8909

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

Listen 8909

<VirtualHost \*:8909>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd02

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

</VirtualHost>

5.SELinux非默认端口的开放

# semanage port -l | grep http

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

-a:添加 -t：类型 -p：协议

[root@server0 /]# systemctl restart httpd

客户端验证：

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

##################################################

虚拟Web主机匹配优先级：由上到下，匹配及停止

端口的优先级大于域名

################################################

###############################################

客户端访问服务端资源：

1.防火墙

2.服务本身的权限

3.本地目录的权限

4.SELinux安全的策略

###############################################

使用教学虚拟机

• 每个学员机上有三台预先配置好的虚拟机

– server —— 作为练习用服务器

– desktop —— 作为练习用客户机

– classroom —— 提供网关/DNS/软件素材/DHCP服务等资源

优先开机classroom，再去开启server、desktop

一、真机还原命令：优先还原classroom

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server[root@room9pc01 ~]# rht-vmctl reset desktop

二、进入 虚拟机server 与 虚拟机desktop,检测Yum是否可用

# yum clean all #清空缓存

# yum repolist #列出仓库信息

###############################################

HTTP服务基础

• 基于 B/S (Browser/Server)架构的网页服务

– 服务端提供网页

– 浏览器下载并显示网页

• Hyper Text Markup Language(HTML),超文本标记语言

• Hyper Text Transfer Protocol（HTTP）,超文本传输协议

HTTP默认端口 ： 80

#################################################

环境的初步设置：修改防火墙默认区域为允许所有

虚拟机Server

# firewall-cmd --set-default-zone=trusted

虚拟机Desktop

# firewall-cmd --set-default-zone=trusted

#################################################

搭建基于域名的虚拟Web主机

实现两个网站

server0.example.com 与 webapp0.example.com

server0.example.com---》页面内容 My Web01

webapp0.example.com---》页面内容 My Web02

页面均由虚拟机Server提供

[root@server0 ~]# 安装httpd

# mkdir /var/www/nsd01 /var/www/nsd02

# echo '<h1>My Web1' > /var/www/nsd01/index.html

# echo '<h1>My Web2' > /var/www/nsd02/index.html

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

<VirtualHost \*:80>

Servername server0.example.com

DocumentRoot /var/www/nsd01

</VirtualHost>

<VirtualHost \*:80>

Servername webapp0.example.com

DocumentRoot /var/www/nsd02

</VirtualHost>

[root@server0 ~]# systemctl restart httpd

################################################

安全的Web服务器

• Public Key Infrastructure,公钥基础设施

– 公钥:主要用来加密数据

– 私钥:主要用来解密数据(与相应的公钥匹配)

– 数字证书:证明拥有者的合法性/权威性(单位名称、

有效期、公钥、颁发机构及签名、......)

– Certificate Authority,数字证书授权中心:负责证书

的申请/审核/颁发/鉴定/撤销等管理工作

虚拟机Server：搭建安全的Web服务

1.部署网站证书（营业执照）

# cd /etc/pki/tls/certs

# wget http://classroom.example.com/pub/tls/certs/server0.crt

# ls

2.部署根证书（公安局信息）

# cd /etc/pki/tls/certs

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

# ls

3.部署私钥（用于解密）

# cd /etc/pki/tls/private/

# wget http://classroom.example.com/pub/tls/private/server0.key

# ls

4.安装支持安全https的软件

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

5.修改配置文件

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

#补充，在vim末行模式 ，开启行号 ：set nu

59 DocumentRoot "/var/www/html"

60 ServerName www0.example.com:443

#指定网站证书

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

#指定解密的密钥

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

#指定根证书

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

6.书写一个网站用于测试 DocumentRoot "/var/www/html"

# echo '<h1>Hello world' > /var/www/html/index.html

7.重起httpd服务

[root@server0 /]# systemctl restart httpd

验证：

# firefox https://www0.example.com

鼠标点击选择 我已了解可能的风险----->确认安全例外

###############################################

邮件服务

• 电子邮件服务器的基本功能

– 为用户提供电子邮箱存储空间(用户名@邮件域名)

– 处理用户发出的邮件 —— 传递给收件服务器

– 处理用户收到的邮件 —— 投递到邮箱

SMTP：用户发邮件协议 25

pop3：用户收邮件协议 110

DNS服务器：classroom.example.com

yg@server0.example.com

xln@server0.example.com

虚拟机Server0：

1.安装 postfix软件

[root@server0 /]# rpm -q postfix

postfix-2.10.1-6.el7.x86\_64

2.修改主配置文件/etc/postfix/main.cf

[root@server0 /]# vim /etc/postfix/main.cf

#在末行模式下 :set nu 开启行号

99 myorigin = server0.example.com #默认补全域名后缀

116 inet\_interfaces = all #允许所有使用邮件功能

164 mydestination = server0.example.com

#利用值来判断为本域邮件

3.重起服务

[root@server0 /]# systemctl restart postfix

################################################

使用mail命令发信/收信

• mail 发信操作

– mail -s '邮件标题' -r 发件人 收件人[@收件域]...

• mail 收信操作

– mail [-u 用户名]

[root@server0 /]# useradd yg

[root@server0 /]# useradd xln

# echo haha | mail -s 'test' -r yg xln

# mail -u xln

>N 1 yg@server0.example.com

& 1 #输入邮件的编号，查看邮件内容

################################################

parted大容量分区，专用于gpt分区方案

128个主分区，最大容量到18EB

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

(parted) mktable gpt #指定分区模式为GPT

(parted) print #输出分区表

(parted) mkpart #划分 分区

分区名称？ []? xixi #分区名称 随意起

文件系统类型？ [ext2]? ext4 #分区文件系统 无实际作用

起始点？ 0 # 起始点

结束点？ 4G # 结束点

忽略/Ignore/放弃/Cancel? Ignore #输入i选择忽略

(parted) unit GB # 使用GB作为显示单位

(parted) print

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? ext4

起始点？ 4G

结束点？ 6G

(parted) print

###############################################

[root@server0 ~]# ls /dev/vdb1 /dev/vdb2

什么是交换空间

• 相当于虚拟内存,

– 当物理内存不够用时,使用磁盘空间来模拟内存

– 在一定程度上缓解内存不足的问题

– 交换分区:以空闲分区充当的交换空间

[root@server0 ~]# mkswap /dev/vdb1 #格式化交换文件系统

[root@server0 ~]# mkswap /dev/vdb2 #格式化交换文件系统

[root@server0 ~]# swapon /dev/vdb2 #启用的命令

[root@server0 ~]# swapon -s #查看交换空间的成员

[root@server0 ~]# swapoff /dev/vdb1 #停用交换分区

[root@server0 ~]# swapon -s

[root@server0 ~]# swapoff /dev/vdb2

[root@server0 ~]# swapon -s

#################################################

交换分区的开机自动启用

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

/dev/vdb1 swap swap defaults 0 0

/dev/vdb2 swap swap defaults 0 0

[root@server0 ~]# swapon -a #检测书写是否正确

[root@server0 ~]# swapon -s

##############################################

读写Samba共享

服务端：

1.安装软件samba

2.创建共享帐号

3.修改配置文件/etc/samba/smb.conf

[ 共享名]

path = 实际共享目录路径

write list = 可写共享帐号名

4.创建 实际共享目录路径

5.重起smb服务

6.设置本地目录的权限

7.SELinux布尔值

8.防火墙设置

客户端：

1.安装软件cifs-utils

2.实现挂载

################################################

逻辑卷的管理

将众多的分区，组成卷组，从卷组中划分逻辑卷

逻辑卷的扩展：

1.卷组有剩余空间：扩展逻辑卷的空间，扩展文件系统

2.卷组没有有剩余空间：扩展卷组，扩展逻辑卷的空间，扩展文件系统

################################################

配置聚合链接

1.创建虚拟机网卡team0 config '{}'

2.为team0添加两个成员

3.为team0配置IP地址

4.激活配置

#################################################

iscsi网络磁盘

服务端：

1.创建分区

2.安装targetcli

3.创建后端存储

4.创建target磁盘组，起iqn名字

5.将后端存储放入target磁盘组，lun关联

6.设置客户端声称的名字

7.设置本机IP地址启用iscsi网络磁盘

8.重起target服务

客户端：

1.安装软件包iscsi-initiator-utils

2.修改配置文件,指定客户端声称的名字

/etc/iscsi/initiatorname.iscsi

3.重起iscsid服务，刷新iqn标识

4.运行iscsiadm命令进行发现服务端共享存储

5.重起iscsi识别共享存储

################################################

动态Web

1.下载或书写一个python

2.进行网页的跳转 Alias 网络路径 实际本地路径

3.安装mod\_wsgi软件支持翻译python代码

4.书写配置WsgiScriptAlias 网络路径 实际本地路径

5.重起httpd服务

6.书写配置文件

Listen 8909

<VirtualHost \*:8909>

7.修改SELinux端口

8.重起httpd服务

一、搭建新的教学环境

真机上操作:新建两台虚拟机

[root@room9pc01 ~]# clone-vm7

Enter VM number: 8

[root@room9pc01 ~]# clone-vm7

Enter VM number: 9

利用鼠标图形将虚拟机名字，修改为A与B，进行开机启动

利用root用户，密码为 123456进入新建两台虚拟机

虚拟机A:作为练习使用的服务端

1.配置永久的主机名:svr7.tedu.cn

2.配置eth0永久的IP地址:192.168.4.7/24

虚拟机B:作为练习使用的客户端

1.配置永久的主机名:pc207.tedu.cn

2.配置eth0永久的IP地址:192.168.4.207/24

##############################################

二、为新的教学环境搭建Yum仓库

真机上搭建服务端 ：

1.众多的软件包 2.仓库数据文件 3.搭建FTP服务共享内容

1）搭建FTP服务

[root@room9pc01 ~]# rpm -q vsftpd

vsftpd-3.0.2-22.el7.x86\_64

[root@room9pc01 ~]# systemctl restart vsftpd

[root@room9pc01 ~]# systemctl enable vsftpd

2)挂载光盘内容到/var/ftp/rhel7

[root@room9pc01 ~]# ls /var/ftp/rhel7/

[root@room9pc01 ~]# vim /etc/fstab

/var/lib/libvirt/images/iso/rhel-server-7.4-

x86\_64-dvd.iso /var/ftp/rhel7 iso9660 defaults 0 0

3)最终测试

# firefox ftp://192.168.4.254/rhel7

客户端：两台虚拟机A与虚拟机B

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

[rhel7]

name=rhel7.4

baseurl=ftp://192.168.4.254/rhel7

enabled=1

gpgcheck=0

[root@svr7 ~]# yum -y install httpd

###############################################

真机制作别名

[root@room9pc01 ~]# head -3 /root/.bashrc

# .bashrc

alias goa='ssh -X root@192.168.4.7'

alias gob='ssh -X root@192.168.4.207'

[root@room9pc01 ~]#

###############################################

利用真机clone-vm7产生产两台虚拟机，利用root密码123456

1.搭建虚拟机C服务端

要求：配置永久主机名svr8.tedu.cn

配置eth0永久 IP地址：192.168.4.8/24

使用真机ftp服务器作为Yum源

2.搭建虚拟机D客户端

要求：配置永久主机名pc208.tedu.cn

配置eth0永久 IP地址：192.168.4.208/24

使用真机ftp服务器作为Yum源

################################################

目录结构

• 认识Linux的目录层次:

– man hier

常见一级目录的用途

主要用途

/boot 存放系统引导必需的文件,包括内核、启动配置

/bin、/sbin 存放各种命令程序

/dev 存放硬盘、键盘、鼠标、光驱等各种设备文件

/etc 存放Linux系统及各种程序的配置文件

/root、/home/用户名 分别是管理员root、普通用户的默认家目录

/var 存放日志文件、邮箱目录等经常变化的文件

/proc 存放内存中的映射数据,不占用磁盘

/tmp 存放系统运行过程中使用的一些临时文件

权限的数值表示

• 权限的数值化

– 基本权限:r = 4,w = 2,x = 1

– 附加权限:SUID = 4,SGID = 2,Sticky Bit = 1

• 采用数值形式设置权限

– chmod [-R] nnn 文档...

– chmod [-R] xnnn 文档...

[root@svr7 /]# mkdir /nsd01

[root@svr7 /]# ls -ld /nsd01

[root@svr7 /]# chmod 700 /nsd01

[root@svr7 /]# ls -ld /nsd01

[root@svr7 /]# chmod 007 /nsd01

[root@svr7 /]# ls -ld /nsd01

[root@svr7 /]# chmod 750 /nsd01

[root@svr7 /]# ls -ld /nsd01

[root@svr7 /]# chmod 755 /nsd01

[root@svr7 /]# ls -ld /nsd01

############################################

历史命令

• 管理/调用曾经执行过的命令

– history:查看历史命令列表

– history -c:清空历史命令

– !n:执行命令历史中的第n条命令

– !str:执行最近一次以str开头的历史命令

• 调整历史命令的数量

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

HISTSIZE=1000 //默认记录1000条

[root@svr7 /]# history -c

[root@svr7 /]# cat /etc/redhat-release

[root@svr7 /]# uname -r

[root@svr7 /]# ifconfig | head -2

[root@svr7 /]# history

[root@svr7 /]# !cat #执行历史命令最近一条以cat开头的命令

[root@svr7 /]# !u

################################################

实用小命令工具

• du,统计文件的占用空间

– du [选项]... [目录或文件]...

– -s:只统计每个参数所占用的总空间大小

– -h:提供易读容量单位(K、M等)

[root@svr7 /]# du -sh /

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

[root@svr7 /]# du -sh /\*

#################################################

• date,查看/调整系统日期时间

– date +%F、date +%R

– date +"%Y-%m-%d %H:%M:%S"

– date -s "yyyy-mm-dd HH:MM:SS" #修改时间

##############################################

vim编辑技巧

命令模式操作

光标跳转

Home 键 或 ^键、数字 0 跳转到行首

End 键 或“$”键 跳转到行尾

1G 或 gg 跳转到文件的首行

G 跳转到文件的末尾行

复制/粘贴/删除

yy、#yy 复制光标处的一行、#行

p、P 粘贴到光标处之后、之前

x 或 Delete键 删除光标处的单个字符

dd、#dd 删除光标处的一行、#行

d^ 从光标处之前删除至行首

d$ 从光标处删除到行尾

C(大写) 从光标处删除到行尾，进入插入模式

[root@svr7 ~]# cp /etc/passwd /opt/abc

[root@svr7 ~]# vim /opt/abc

/word 向后查找字符串“word”

n、N 跳至后/前一个结果

u 撤销最近的一次操作

Ctrl + r 取消前一次撤销操作(反撤销)

ZZ(大写) 保存修改并退出

##############################################

末行模式操作

保存/退出/文件操作

:q! 放弃已有更改后强制退出

:wq 或 :x 保存已有修改后退出

:r /etc/filesystems 读入其他文件内容

[root@svr7 ~]# echo 123456 > /opt/1.txt

[root@svr7 ~]# echo abcdef > /opt/a.txt

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

：r /opt/1.txt #读入/opt/1.txt文件内容

：r /etc/fstab #读入/etc/fstab文件内容

字符串替换

:s/root/admin 替换当前行第一个“root”

:s/root/admin/g 替换当前行所有的“root”

:1,11 s/root/admin/g 替换第1-11行所有的“root”

:% s/root/admin/g 替换文件内所有的root

[root@svr7 ~]# cp /etc/passwd /opt/test

[root@svr7 ~]# vim /opt/test

sed 非交互式编辑器 vim 交互式编辑器

开关参数的控制

:set nu|nonu 显示/不显示行号

:set ai|noai 启用/关闭自动缩进

##################################################

vim常见问题：缓存文件(交换文件),删除交换文件

[root@svr7 ~]# rm -rf /opt/.1.txt.swp

[root@svr7 ~]# vim /opt/1.txt

##################################################

自定义Yum仓库

1.具备互联网下载的包，真机传递tools.tar.gz到虚拟机A

ssh+cp=scp

# scp /root/桌面/tools.tar.gz root@192.168.4.7:/root

2.虚拟机A解tar包

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

[root@svr7 ~]# ls /

[root@svr7 ~]# ls /tools/

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

3.运行createrepo生成仓库数据文件

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

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

4.书写客户端配置文件

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

[myrpm]

name=rhel7.4

baseurl=file:///tools/other #指定本机为Yum服务端

enabled=1

gpgcheck=0

[root@svr7 ~]# yum repolist

################################################

[root@svr7 ~]# yum -y install sl

查询rpm包安装的清单

[root@svr7 ~]# rpm -ql sl

[root@svr7 ~]# yum -y install oneko

[root@svr7 ~]# oneko & #放入后台运行

[root@svr7 ~]# killall oneko #杀死所有的oneko

#################################################

源码编译安装

rpm包： rpm -ivh yum

源码包-----运行gcc与make--->可以执行的程序---》运行安装即可

源码包的优势：可以选择安装的功能及路径，更加的灵活

虚拟机A：

步骤1: 安装gcc与make

[root@svr7 ~]# yum -y install gcc

步骤2: tar解包,释放源代码至指定目录

# tar -xf /tools/inotify-tools-3.13.tar.gz -C /

# ls /

# ls /inotify-tools-3.13/

步骤3: ./configure 配置,指定安装目录/功能模块等选项

作用：检测本机是否安装gcc

# cd /inotify-tools-3.13/

# ./configure --prefix=/opt/myrpm #指定安装位置

常见错误提示：没有安装gcc

checking for gcc... no

checking for cc... no

checking for cl.exe... no

configure: error: no acceptable C compiler found in $PATH

步骤4: make 编译,生成可执行的二进制程序文件

# cd /inotify-tools-3.13/

# make

步骤5: make install 安装,将编译好的文件复制到安装目录

# cd /inotify-tools-3.13/

# make install

# ls /opt/

# ls /opt/myrpm

# ls /opt/myrpm/

# ls /opt/myrpm/bin/

###############################################

[root@svr7 ~]# man 5 passwd

5:配置文件的帮助信息

################################################

zip归档工具,跨平台的压缩工具

• 归档+压缩操作

– zip [-r] 备份文件.zip 被归档的文档...

• 释放归档+解压操作

– unzip 备份文件.zip [-d 目标文件夹]

[root@svr7 /]# zip -r /opt/test.zip /root/ /boot/

[root@svr7 /]# ls /opt

[root@svr7 /]# unzip /opt/test.zip -d /mnt/

[root@svr7 /]# ls /mnt/

[root@svr7 /]# zip -r /opt/nsd.zip /etc/passwd

[root@svr7 /]# ls /opt

[root@svr7 /]# unzip /opt/nsd.zip -d /mnt/

[root@svr7 /]# ls /mnt/

[root@svr7 /]# zip -r /opt/abc.zip /home

[root@svr7 /]# ls /opt

[root@svr7 /]# unzip /opt/abc.zip -d /mnt/

[root@svr7 /]# ls /mnt/

#################################################

软连接与硬连接

快捷方式(软连接)：ln -s /路径/源文档 /路径/快捷方式的名

快捷方式(硬连接)：ln /路径/源文档 /路径/快捷方式的名

[root@svr7 /]# cat /etc/redhat-release

[root@svr7 /]# ln -s /etc/redhat-release /abc

[root@svr7 /]# ls /

[root@svr7 /]# ls -l /abc

[root@svr7 /]# cat /abc

############################################

[root@svr7 /]# rm -rf /opt/\*

[root@svr7 /]# cd /opt/

[root@svr7 opt]# echo 123456 > /opt/A.txt

[root@svr7 opt]# ln -s /opt/A.txt /opt/B.txt

[root@svr7 opt]# ls

A.txt B.txt

[root@svr7 opt]# ln /opt/A.txt /opt/C.txt

[root@svr7 opt]# ls

[root@svr7 opt]# ls -i

##############################################

i节点编号：标识硬盘上一块存储空间的编号

读取数据1.txt：1.txt----》i节点-----》数据对应区域 取数据

软连接（常用）：快捷方式可以跨分区，源文件可以是目录

硬连接：快捷方式不可以跨分区，源文件不可以是目录

软连接独占一个i节点编号，硬连接与源文件i节点编号一致

##############################################

lisi----》38 lisi 38房间

tom----》39 39房间 纸条 其实我是lisi

#################################################

###############################################

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

– 运行时的虚拟根环境:/var/named/chroot/ #牢笼政策

– 系统服务:named

– 默认端口: 53

• 主配置文件: /etc/named.conf #设置本服务器负责解析的域名

• 地址库文件: /var/named/ #主机名与IP地址的对应关系

###############################################

在虚拟机A搭建DNS服务器

负责解析baidu.com域名的DNS

最终在虚拟机B测试解析www.baidu.com----->192.168.10.1

虚拟机A：

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

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

zone "baidu.com" IN {

type master;

file "baidu.com.zone";

};

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

[root@svr7 named]# cp -p tedu.cn.zone baidu.com.zone

[root@svr7 named]# vim baidu.com.zone

baidu.com. NS svr7

svr7 A 192.168.4.7

www A 192.168.10.1

[root@svr7 named]# systemctl restart name

虚拟机B:

# vim /etc/resolv.conf

nameserver 192.168.4.7

# nslookup www.baidu.com

#################################################

分离解析概述（视图解析）

• 当收到客户机的DNS查询请求的时候

– 能够区分客户机的来源地址

– 为不同类别的客户机提供不同的解析结果(IP地址)

– 不同的客户端解析同一个域名时，解析结果不同

– 为客户端提供最近的资源

• 环境及需求

– 权威DNS:svr7.tedu.cn 192.168.4.7

– 负责区域:tedu.cn

– A记录分离解析 —— 以 www.tedu.cn 为例

客户机来自解析结果

192.168.4.207 -------> 192.168.4.100

其他地址 -------> 1.2.3.4

###############################################

分离解析的配置：

view "联通" {

match-clients { 来源地址1; .. ..; }

zone "12306.cn" IN {

...... 地址库1;

}; };

view "铁通" {

match-clients { 来源地址2; .. .. ; }

zone "12306.cn" IN {

...... 地址库2;

}; };

分离解析注意事项：

1.所有的客户端必须到要找到自己的分类

2.类别的view匹配,由上及下，匹配及停止

3.所有的zone都必须在view中

view "nsd" {

match-clients { 192.168.4.207; }；

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";---->192.168.4.100

}; };

view "other" {

match-clients { any; }；

zone "tedu.cn" IN {

type master;

file "tedu.cn.other";---->1.2.3.4

}; };

然后在/var/named建立相应的地址库文件，写入不同的解析结果

################################################

针对来源地址定义acl列表

– 若地址比较少,也可以不建立列表

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

options {

directory "/var/named";

};

acl test { 192.168.4.207; 192.168.7.0/24;

192.168.10.1; };

view "nsd" {

match-clients { test; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

################################################

多区域的分离解析

注意事项：每一个view中，zone的个数一致

options {

directory "/var/named";

};

acl test { 192.168.4.207; 192.168.7.0/24; 192.168.10.1; };

view "nsd" {

match-clients { test; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "qq.com" IN {

type master;

file "qq.com.zone";

};

};

view "other" {

match-clients { any; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.other";

};

zone "qq.com" IN {

type master;

file "qq.com.other";

};

};

##################################################

RAID磁盘阵列

• 廉价冗余磁盘阵列

– Redundant Arrays of Inexpensive Disks

– 通过硬件/软件技术,将多个较小/低速的磁盘整合成一个大磁盘

– 阵列的价值:提升I/O效率、硬件级别的数据冗余

– 不同RAID级别的功能、特性各不相同

• RAID 0,条带模式

– 同一个文档分散存放在不同磁盘

– 并行写入以提高效率,无容错功能

– 至少需要两块磁盘组成

• RAID 1,镜像模式

– 一个文档复制成多份,分别写入不同磁盘

– 多份拷贝提高可靠性,效率无提升

– 至少需要两块磁盘组成

• RAID5,高性价比模式

– 相当于RAID0和RAID1的折中方案

– 需要至少一块磁盘的容量来存放校验数据

– 至少需要三块磁盘组成

• RAID6,高性价比/可靠模式

– 相当于扩展的RAID5阵列,提供2份独立校验方案

– 需要至少两块磁盘的容量来存放校验数据

– 至少需要四块磁盘组成

################################################

进程管理

一、进程的简介

1.程序：静态的代码，仅仅占用磁盘空间

2.进程：正在运行的代码，会占用CPU与内存

父进程与子进程，后台整个进程，进程树

进程唯一标识编号：PID

二、查看进程

systemd:Linux运行的第一个进程，由内核直接运行

所有进程的父进程

• 常用命令选项

– -a:显示完整的命令行

– -p:列出对应PID编号

– -u:指定用户名

[root@svr7 /]# pstree #查看整个进程树

[root@svr7 /]# pstree -u lisi #查看lisi开启的进程

[root@svr7 /]# pstree -pu lisi

[root@svr7 /]# pstree -apu lisi

[root@svr7 /]# pstree lisi

[root@svr7 /]# pstree -ap lisi

[root@svr7 /]#

• ps aux 操作(进程的信息详细)

– 列出正在运行的所有进程

用户 进程ID %CPU %内存 虚拟内存 固定内存 终端 状态 起始时间 CPU时间 程序指令

• ps -elf 操作 (进程的父进程的信息PPID)

– 列出正在运行的所有进程

[root@svr7 /]# ps aux | grep lisi

进程动态排名

• top 交互式工具

– 格式:top [-d 刷新秒数] [-U 用户名]

[root@svr7 /]# top -d 1

按键盘大写的P，进行CPU排序

按键盘大写的M，进行内存排序

按键盘的q,退出

查询进程的指令

• pgrep — Process Grep

• 常用命令选项

– -l:输出进程名,而不仅仅是 PID

– -U:检索指定用户的进程

– -t:检索指定终端的进程

– -x:精确匹配完整的进程名

[root@svr7 /]# pgrep -l crond #-l显示进程名字

[root@svr7 /]# pgrep -l log

[root@svr7 /]# pgrep crond

[root@svr7 /]# a=`pgrep crond` #将命令的输出赋值给变量

[root@svr7 /]# echo $a

[root@svr7 /]# pgrep -U lisi #检索指定用户

[root@svr7 /]# pgrep -lU lisi

[root@svr7 /]# pgrep -lx systemd #检索进程名，严格匹配

pts:图形命令行

[root@svr7 /]# who #查看正在登陆用户信息

[root@svr7 /]# pgrep -lU lisi

[root@svr7 /]# pgrep -lU lisi -t pts/2

-t：指定检索的终端

###############################################

查看进程的命令：

pstree： 结构鲜明，可以查看整个进程树

ps aux： 查看正在运行的所有进程，信息非常全面

ps -elf 查看正在运行的所有进程，信息中有父进程的信息

top：动态，可以进行排名

pgrep：选项众多，用于检索进行信息，方便脚步中变量赋值

#################################################

进程的前后台调度

将进程放入后台启动

– 在命令行末尾添加“&”符号,不占用当前终

– Ctrl + z 组合键：将正在运行的进程(暂停并转入后台)

• jobs 命令

– 查看后台任务列表

• fg 命令

– 将后台任务恢复到前台运行

• bg 命令

– 激活后台被暂停的任务

[root@svr7 /]# sleep 1000 & #正在运行放入后台

[root@svr7 /]# jobs

[root@svr7 /]# sleep 2000

^Z #输入Ctrl+z暂停放入后台

[2]+ 已停止 sleep 2000

[root@svr7 /]# jobs #查看后台任务

[root@svr7 /]# bg 2 #让后台编号为2的任务 继续运行

[root@svr7 /]# jobs

[root@svr7 /]# fg 2 #让后台编号为2的任务 恢复到前台

sleep 2000

^C #输入Ctrl+c结束进程

[root@svr7 /]# jobs

[root@svr7 /]# fg 1 #让后台编号为1的任务 恢复到前台

sleep 1000

^C

[root@svr7 /]# jobs

#################################################

杀死进程

• 干掉进程的不同方法

– Ctrl+c 组合键,中断当前命令程序

– kill [-9] PID...

– killall [-9] 进程名...

– pkill 查找条件

[-9]:强制杀死

[root@svr7 /]# sleep 1000 &

[root@svr7 /]# sleep 2000 &

[root@svr7 /]# sleep 3000 &

[root@svr7 /]# jobs -l

[root@svr7 /]# kill 14986

[root@svr7 /]# jobs -l

[root@svr7 /]# jobs -l

[root@svr7 /]# killall sleep #杀死所有叫sleep

[root@svr7 /]# jobs -l

[root@svr7 /]# sleep 4000 &

[root@svr7 /]# jobs

[root@svr7 /]# killall -9 sleep

[root@svr7 /]# jobs

##############################################

日志管理

• 系统和程序的“日记本”

– 记录系统、程序运行中发生的各种事件

– 通过查看日志,了解及排除故障

– 信息安全控制的依据

• 由系统服务rsyslog统一记录/管理

• 常见的日志文件

/var/log/messages 记录内核消息、各种服务的公共消息

/var/log/dmesg 记录系统启动过程的各种消息

/var/log/cron 记录与cron计划任务相关的消息

/var/log/maillog 记录邮件收发相关的消息

/var/log/secure 记录与访问限制相关的安全消息

日志分析

• 通用分析工具

– tail、tailf、less、grep等文本浏览/检索命令

– awk、sed等格式化过滤工具

tailf:实时跟踪信息

[root@svr7 /]# echo 123 > /opt/1.txt

[root@svr7 /]# tailf /opt/1.txt

123

haha

用户登录分析

• users、who、w 命令

– 查看已登录的用户信息,详细度不同

• last、lastb 命令

– 查看最近登录成功/失败的用户信息

[root@svr7 /]# users

[root@svr7 /]# who

[root@svr7 /]# w

[root@svr7 /]# last -2

[root@svr7 /]# lastb -2

###########################################

• Linux内核定义的事件紧急程度

– 分为 0~7 共8种优先级别

– 其数值越小,表示对应事件越紧急/重要

0 EMERG（紧急） 会导致主机系统不可用的情况

1 ALERT（警告） 必须马上采取措施解决的问题

2 CRIT（严重） 比较严重的情况

3 ERR（错误） 运行出现错误

4 WARNING（提醒） 可能会影响系统功能的事件

5 NOTICE（注意） 不会影响系统但值得注意

6 INFO（信息） 一般信息

7 DEBUG（调试） 程序或系统调试信息等

使用journalctl工具

• 提取由 systemd-journal 服务搜集的日志

– 主要包括内核/系统日志、服务日志

• 常见用法

– journalctl -u 服务名 [-p 优先级]

– journalctl | grep 关键词

– journalctl -n 消息条数

– journalctl --since="yyyy-mm-dd HH:MM:SS" --

until="yyyy-mm-dd HH:MM:SS"

[root@svr7 /]# yum -y install httpd

[root@svr7 /]# systemctl restart httpd

显示httpd服务，日志消息在6级别以上的信息

[root@svr7 /]# journalctl -u httpd -p 6

显示named服务，日志消息在6级别以上的信息

[root@svr7 /]# journalctl -u named -p 6

###########################################

systemctl控制

• Linux系统和服务管理器

– 是内核引导之后加载的第一个初始化进程(PID=1)

– 负责掌控整个Linux的运行/服务资源组合

对于服务的管理

systemctl restart 服务名 #重起服务

systemctl start 服务名 #开启服务

systemctl stop 服务名 #停止服务

systemctl status 服务名 #查看服务当前的状态

systemctl enable 服务名 #设置服务开机自启动

systemctl disable 服务名 #设置服务不开机自启动

systemctl enable 服务名

#查看是否开机自启动

[root@svr7 /]# systemctl is-enabled httpd

disabled #禁止开机启动

[root@svr7 /]# systemctl enable httpd

[root@svr7 /]# systemctl is-enabled httpd

enabled #允许开机启动

[root@svr7 /]#

systemd

• 一个更高效的系统&服务管理器

– 开机服务并行启动,各系统服务间的精确依赖

– 服务目录:/lib/systemd/system/

[root@svr7 /]# vim /lib/systemd/system/httpd.service

RHEL6:运行级别 200

0：关机

1：单用户模式（基本功能的实现，破解Linux密码）

2：多用户字符界面（不支持网络）

3：多用户字符界面（支持网络）服务器默认的运行级别

4：未定义

5：图形界面

6：重起

切换运行级别：init 数字

RHEL7：运行模式

字符模式：multi-user.target

图形模式：graphical.target

当前直接切换到图形模式

[root@svr7 /]# systemctl isolate graphical.target

当前直接切换到字符模式

[root@svr7 /]# systemctl isolate multi-user.target

查看每次开机默认进入模式

# systemctl get-default

multi-user.target

设置永久策略，每次开机自动进入graphical.target

# systemctl set-default graphical.target

# reboot

############################################

部署DHCP服务器

• Dynamic Host Configuration Protocol

– 动态主机配置协议,由 IETF(Internet 网络工程师任

务小组)组织制定,用来简化主机地址分配管理

• 主要分配以下入网参数

– IP地址/子网掩码/广播地址

– 默认网关地址、DNS服务器地址

• 装软件包 dhcp

• 配置文件 /etc/dhcp/dhcpd.conf

• 起服务 dhcpd

• DHCP地址分配的四次会话（以广播方式进行，先到先得）

– DISCOVERY --> OFFER --> REQUEST --> ACK

一个网络中只能有一个DHCP服务器

• 服务端基本概念

– 租期:允许客户机租用IP地址的时间期限,单位为秒

– 作用域:分配给客户机的IP地址所在的网段

– 地址池:用来动态分配的IP地址的范围

一、部署 DHCP服务

1.安装dhcp软件包

[root@svr7 /]# yum -y install dhcp

2.修改配置文件

[root@svr7 /]# vim /etc/dhcp/dhcpd.conf

：r /usr/share/doc/dhcp\*/dhcpd.conf.example

subnet 192.168.4.0 netmask 255.255.255.0 { #分配的网段

range 192.168.4.100 192.168.4.200; #分配IP范围

option domain-name-servers 192.168.4.7; #DNS服务器

option routers 192.168.4.254; #网关地址

default-lease-time 600;

max-lease-time 7200;

}

3.重起服务

[root@svr7 /]# systemctl restart dhcpd

##################################################

网络装机

• 规模化:同时装配多台主机

• 自动化:装系统、配置各种服务

• 远程实现:不需要光盘、U盘等物理安装介质

什么是PXE网络

• PXE,Pre-boot eXecution Environment

– 预启动执行环境,在操作系统之前运行

– 可用于远程安装

• 工作模式

– PXE client 集成在网卡的启动芯片中

– 当计算机引导时,从网卡芯片中把PXE client调入内存

执行,获取PXE server配置、显示菜单,根据用户选

择将远程引导程序下载到本机运行

• 需要哪些服务组件?

– DHCP服务,分配IP地址、定位引导程序

– TFTP服务,提供引导程序下载

– HTTP服务,提供yum安装源

开机启动项： 1.硬盘 2.光驱设备 3.移动存储设备 4.网络安装

##################################################

二、配置DHCP服务

1.修改配置文件

[root@svr7 /]# vim /etc/dhcp/dhcpd.conf

subnet 192.168.4.0 netmask 255.255.255.0 {

range 192.168.4.100 192.168.4.200;

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"; #指定网卡引导文件名称

}

[root@svr7 /]# systemctl restart dhcpd

pxelinux.0: 网卡引导文件（安装说明书）二进制文件

安装一个软件获得该文件

#################################################

三、搭建tftp服务

tftp：简单的文件传输协议 69

默认共享路径：/var/lib/tftpboot/

1.安装软件包tftp-server

[root@svr7 /]# yum -y install tftp-server

2.重起tftp服务

[root@svr7 /]# systemctl restart tftp

3.部署pxelinux.0文件

# yum provides \*/pxelinux.0 #查询仓库中那个软件包产生

# yum -y install syslinux

# rpm -ql syslinux | grep pxelinux.0 #查询安装清单

# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/

# ls /var/lib/tftpboot/

4.部署菜单文件

pxelinux.0-->/var/lib/tftpboot/pxelinux.cfg/default

[root@svr7 /]# mkdir /var/lib/tftpboot/pxelinux.cfg

[root@svr7 /]# mount /dev/cdrom /mnt

mount: /dev/sr0 写保护，将以只读方式挂载

[root@svr7 /]# ls /mnt/

# cp /mnt/isolinux/isolinux.cfg /var/lib/tftpboot/pxelinux.cfg/default

# ls -l /var/lib/tftpboot/pxelinux.cfg/default

# chmod u+w /var/lib/tftpboot/pxelinux.cfg/default

# ls -l /var/lib/tftpboot/pxelinux.cfg/default

5.部署图形的模块及背景图片

图形的模块：vesamenu.c32

背景图片：splash.png

# cp /mnt/isolinux/vesamenu.c32 /mnt/isolinux/splash.png /var/lib/tftpboot/

# ls /var/lib/tftpboot/

6.部署启动内核与驱动程序

启动内核：vmlinuz

驱动程序：initrd.img

# cp /mnt/isolinux/vmlinuz /mnt/isolinux/initrd.img /var/lib/tftpboot/

# ls /var/lib/tftpboot/

7.修改菜单文件

# vim /var/lib/tftpboot/pxelinux.cfg/default

# 开启行号

1 default vesamenu.c32 #默认加载图形的模块

2 timeout 600 #读秒时间 1/10

10 menu background splash.png #指定背景图片

11 menu title NSD1808 PXE Server ！ #菜单显示的标题

61 label linux

62 menu label Install RHEL7.4 #选项内容

63 menu default #读秒结束后默认选择

64 kernel vmlinuz #加载内核

65 append initrd=initrd.img #加载驱动

#################################################

总结：

1.DHCP---》IP地址-----》next-server----》pxelinux.0

2.tftp--》pxelinux.0

3.pxelinux.0-->/var/lib/tftpboot/pxelinux.cfg/default

4.default->vesamenu.c32\splash.png\vmlinuz\initrd.img

###################################################

[root@svr7 /]# systemctl restart dhcpd

[root@svr7 /]# systemctl restart tftp

简单的验证：

1.新建一台全新的虚拟机，安装的方式为PXE网络引导

网络类型选择为private1,其他选择均默认即可

#################################################

四、搭建httpd服务

1.安装软件包

[root@svr7 /]# yum -y install httpd

2.将光盘内容放入DocumentRoot（/var/www/html）

[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

3.重起服务

[root@svr7 /]# systemctl restart httpd

[root@svr7 /]# firefox 192.168.4.7/rhel7

################################################

五、实现无人值守安装，生成应答文件

1.安装一个图形的工具(system-config-kickstart)进行生成

# yum -y install system-config-kickstart

2.运行system-config-kickstart 进行配置

# system-config-kickstart

首先确认 软件包的选择 是否可以使用

需要本机Yum的支持，必须要求Yum仓库标识 [development]

# vim /etc/yum.repos.d/rhel7.repo

[development]

name=......

baseurl=ftp://192.168.4.254/rhel7

........

# system-config-kickstart

在此确认 软件包的选择 是否可以使用

[root@svr7 ~]# ls /root/

[root@svr7 ~]# vim /root/ks.cfg

###############################################

六、共享ks应答文件

1.通过httpd服务共享

[root@svr7 ~]# cp /root/ks.cfg /var/www/html/

[root@svr7 ~]# ls /var/www/html/

[root@svr7 ~]# firefox 192.168.4.7/ks.cfg

2.通过菜单文件进行指定

# vim /var/lib/tftpboot/pxelinux.cfg/default

........

label linux

menu label Install RHEL7.4

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----》pxelinux.0

2.tftp--》pxelinux.0

3.pxelinux.0-->/var/lib/tftpboot/pxelinux.cfg/default

4.default->vesamenu.c32\splash.png\vmlinuz\initrd.img

-> ks.cfg

5. ks.cfg->语言、时区、分区....

->--url="http://192.168.4.7/rhel7"

#################################################

rsync同步操作

• 命令用法

– rsync [选项...] 源目录 目标目录

• 同步与复制的差异

– 复制:完全拷贝源到目标

– 同步:增量拷贝,只传输变化过的数据

• 本地同步

– rsync [选项...] 本地目录1 本地目录2

– rsync [选项...] 本地目录1/ 本地目录2

[root@svr7 ~]# mkdir /todir

[root@svr7 ~]# rsync -a /boot /todir //同步整个文件夹

[root@svr7 ~]# rsync -a /boot/ /todir/ //只同步目录下的数据

• rsync操作选项

– -n:测试同步过程,不做实际修改

– --delete:删除目标文件夹内多余的文档

– -a:归档模式,相当于-rlptgoD

– -v:显示详细操作信息

– -z:传输过程中启用压缩/解压

[root@svr7 ~]# mkdir /nsd02

[root@svr7 ~]# mkdir /todir

[root@svr7 ~]# touch /nsd02/{1..5}.txt

[root@svr7 ~]# ls /nsd02

[root@svr7 ~]# rsync -avz /nsd02/ /todir/

[root@svr7 ~]# ls /todir

[root@svr7 ~]# echo ahah > /nsd02/1.txt

[root@svr7 ~]# rsync -avz /nsd02/ /todir/

[root@svr7 ~]# cp /etc/passwd /nsd02

[root@svr7 ~]# rsync -avz /nsd02/ /todir/

# touch /todir/7.txt

# rsync -avz --delete /nsd02/ /todir/

# ls /nsd02

# ls /todir

# cp /etc/fstab /todir

# rsync -avz --delete /nsd02/ /todir/

# ls /nsd02

# ls /todir

#################################################

实现远程同步

rsync+ssh

• 与远程的 SSH目录保持同步

– 下行:rsync [...] user@host:远程目录 本地目录

– 上行:rsync [...] 本地目录 user@host:远程目录

虚拟机A：

#ls /opt

#rsync -avz --delete /opt/ root@192.168.4.207:/opt/

#cp /etc/group /opt

#rsync -avz --delete /opt/ root@192.168.4.207:/opt/

虚拟机B：

#ls /opt

#################################################

实时同步

一、部署公钥、私钥 实现ssh无密码认证

1.虚拟机A生成 公钥、私钥

[root@svr7 ~]# ssh-keygen #一路回车

2.公钥、私钥生成的位置

[root@svr7 ~]# ls /root/.ssh/

3.传递公钥到虚拟机B

[root@svr7 ~]# ssh-copy-id root@192.168.4.207

# rsync -avz --delete /opt/ root@192.168.4.207:/opt/

wget ----》 /root/.ssh ---》mv authorized\_keys

二、如何知道目录内容的变化，监控目录内容的变化

1.真机传递软件包到虚拟机A中

# scp /root/桌面/tools.tar.gz root@192.168.4.7:/

2.虚拟机A安装inotify-tools工具，实现监控目录内容的变化

# tar -xf /tools.tar.gz -C /root/

# mkdir /myrpm

# tar -xf /root/tools/inotify-tools-3.13.tar.gz -C /myrpm

# ls /myrpm/

步骤1: 安装gcc与make

步骤2: ./configure #首先切换到,解压目录内

步骤3: make #首先切换到,解压目录内

步骤4: make install #首先切换到,解压目录内

[root@svr7 /]# which inotifywait #查询命令所对应的程序

/usr/local/bin/inotifywait

[root@svr7 /]# echo $PATH

##################################################

inotifywait监控

• 基本用法

– inotifywait [选项] 目标文件夹

• 常用命令选项

– -m,持续监控(捕获一个事件后不退出)

– -r,递归监控、包括子目录及文件

– -q,减少屏幕输出信息

– -e,指定监视的 modify、move、create、delete、

attrib 等事件类别

三、书写Shell脚本

死循环 while

while [条件]

do

循环执行的操作

done

##################################################

[root@svr7 /]# vim /root/rsync.sh

#!/bin/bash

while inotifywait -rqq /opt

do

rsync -az --delete /opt/ root@192.168.4.207:/opt/

done

[root@svr7 /]# /root/rsync.sh #运行脚本

##################################################

-rqq #两个q没有输出信息

################################################

Cobbler装机平台,不同版本的多系统的安装

一 安装一个CentOS虚拟机

真机:

[root@room9pc01 ~]# ls /iso/

CentOS-7-x86\_64-DVD-1708.iso

虚拟机要求:

1.硬盘大小50G

2.网络类型private1

3.软件包选择 "带GUI的服务器"

4.分区选择 "自动分区"

5.配置IP地址:192.168.4.168/24

6.将CentOS放入光驱设备,搭建本地Yum仓库

7.配置主机名:Cobbler.tedu.cn

[root@Cobbler /]# mkdir /dvd

[root@Cobbler /]# mount /dev/cdrom /dvd/

[root@Cobbler /]# ls /dvd/

[root@Cobbler /]# cat /etc/yum.repos.d/dvd.repo

[CentOS]

name=CentOS7

baseurl=file:///dvd

enabled=1

gpgcheck=0

[root@Cobbler /]#

二、虚拟机设置

1.设置防火墙为trusted

# firewall-cmd --set-default-zone=trusted

2.当前及永久设置SELinux状态为permissive

[root@Cobbler ~]# setenforce 0 #当前临时关闭

[root@Cobbler ~]# getenforce

[root@Cobbler ~]# vim /etc/selinux/config

SELINUX=permissive

三、利用scp真机传递Cobbler.zip包到虚拟机192.168.4.168中

# scp /root/桌面/Cobbler.zip root@192.168.4.168:/root/

#################################################

重设root密码 破解密码

案例1:为虚拟机 server 重设管理密码

1)重启系统,进入 recovery 恢复模式(修复模式)

按e键，找到 linux16 行，末尾添加 rd.break console=tty0

按 ctrl + x 启动

2)以可写方式重新挂载 /sysroot,并切换到此环境

switch\_root# mount -o remount,rw /sysroot

switch\_root# chroot /sysroot

sh-3.2#

3)将root用户的密码设置为 redhat

# echo redhat | passwd --stdin root

4)重设SELinux安全标签(安全增强版Linux)

# touch /.autorelabel #让 SElinux 失忆

5)先后执行 exit、reboot 完成修复

# exit

# reboot

##################################################

四、搭建Cobbler装机平台

Cobbler概述软件，管理dhcp、tftp、Web服务

自由的导入镜像与ks应答文件

1.解压Cobbler.zip包

# unzip /root/Cobbler.zip -d /

# cd /Cobbler/

# ls

# unzip /Cobbler/cobbler.zip -d /opt/

# ls /opt/cobbler

五、安装cobbler主程序、工具包等

[root@cobbler /]# yum -y install /opt/cobbler/\*.rpm

################################################################

cobbler网络装机部署

1.安装软件 cobbler cobbler-web dhcp tftp-server pykickstart httpd tftp-server

cobbler #cobbler程序包

cobbler-web #cobbler的web服务包

pykickstart #cobbler检查kickstart语法错误

httpd #Apache web服务

dhcp #dhcp服务

tftp-server #tftp服务

2.配置cobbler

[root@cobbler /]# vim /etc/cobbler/settings

next\_server: 192.168.4.168 #设置下一个服务器还为本机

server: 192.168.4.168 #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

3.配置cobbler的dhcp

[root@svr7 /]# vim /etc/cobbler/dhcp.template

:%s /旧/新/g #全文替换

:%s /192.168.1/192.168.4/g #全文替换

5 次替换，共 4 行

4.绝对路径解压cobbler\_boot.tar.gz #众多的引导文件

# tar -tf /Cobbler/cobbler\_boot.tar.gz #查看包里面内容

# tar -xPf /Cobbler/cobbler\_boot.tar.gz #绝对路径释放

# ls /var/lib/cobbler/loaders/

5.启动相关服务

[root@svr7 /]# systemctl restart cobblerd

[root@svr7 /]# systemctl enable cobblerd

[root@svr7 /]# systemctl restart httpd

[root@svr7 /]# systemctl enable httpd

[root@svr7 /]# systemctl restart tftp

[root@svr7 /]# systemctl enable tftp

[root@svr7 /]# systemctl restart rsyncd #同步服务

[root@svr7 /]# systemctl enable rsyncd

6.同步刷新cobbler配置

[root@svr7 /]# cobbler sync #检测配置是否正确

......

\*\*\* TASK COMPLETE \*\*\*

# firefox https://192.168.4.168/cobbler\_web

用户名：cobbler

密码：cobbler

##########################################################################################################

cobbler应用

cobbler import --path=挂载点 --name=导入系统命名（随意起）

导入安装镜像数据

# mount /dev/cdrom /dvd

# ls /dvd/

# cobbler import --path=/dvd --name=CentOS7

cobbler导入的镜像放在：/var/www/cobbler/ks\_mirror

[root@cobbler /]# cobbler list #查看有哪些系统

distros:

CentOS7-x86\_64 #安装客户端至少2G内存

profiles:

CentOS7-x86\_64

##############################################

删除

# cobbler list

# cobbler profile remove --name=CentOS7-x86\_64 #删除菜单信息

# cobbler distro remove --name=CentOS7-x86\_64 #删除镜像信息

###########################################

# umount /dvd/

# mount /dev/cdrom /dvd 将光盘换成RHEL7

# ls /dvd/

安装完成机器 默认root 密码 cobbler

############################################################################################################

自定义应答文件：开头注释行删除

[root@cobbler ~]# system-config-kickstart #生成ks文件

必须默认kickstart文件存放位置：/var/lib/cobbler/kickstarts/

[root@cobbler ~]# cobbler list

修改kickstart文件：

[root@cobbler ~]# cobbler profile edit --name=CentOS7.4-A --kickstart=/var/lib/cobbler/kickstarts/自定义.cfg

[root@cobbler ~]# cobbler profile report

[root@cobbler ~]# cobbler sync #同步配置

############################################################################################################

Cobbler装机平台,不同版本的多系统的安装

一 安装一个CentOS虚拟机

真机:

[root@room9pc01 ~]# ls /iso/

CentOS-7-x86\_64-DVD-1708.iso

虚拟机要求:

1.硬盘大小50G

2.网络类型private1

3.软件包选择 "带GUI的服务器"

4.分区选择 "自动分区"

5.配置IP地址:192.168.4.168/24

6.将CentOS放入光驱设备,搭建本地Yum仓库

7.配置主机名:Cobbler.tedu.cn

[root@Cobbler /]# mkdir /dvd

[root@Cobbler /]# mount /dev/cdrom /dvd/

[root@Cobbler /]# ls /dvd/

[root@Cobbler /]# cat /etc/yum.repos.d/dvd.repo

[CentOS]

name=CentOS7

baseurl=file:///dvd

enabled=1

gpgcheck=0

[root@Cobbler /]#

二、虚拟机设置

1.设置防火墙为trusted

# firewall-cmd --set-default-zone=trusted

2.当前及永久设置SELinux状态为permissive

[root@Cobbler ~]# setenforce 0 #当前临时关闭

[root@Cobbler ~]# getenforce

[root@Cobbler ~]# vim /etc/selinux/config

SELINUX=permissive

三、利用scp真机传递Cobbler.zip包到虚拟机192.168.4.168中

# scp /root/桌面/Cobbler.zip root@192.168.4.168:/root/

#################################################

重设root密码

案例1:为虚拟机 server 重设管理密码

1)重启系统,进入 recovery 恢复模式(修复模式)

按e键，找到 linux16 行，末尾添加 rd.break console=tty0

按 ctrl + x 启动

2)以可写方式重新挂载 /sysroot,并切换到此环境

switch\_root# mount -o remount,rw /sysroot

switch\_root# chroot /sysroot

sh-3.2#

3)将root用户的密码设置为 redhat

# echo redhat | passwd --stdin root

4)重设SELinux安全标签(安全增强版Linux)

# touch /.autorelabel #让 SElinux 失忆

5)先后执行 exit、reboot 完成修复

# exit

# reboot

##################################################

四、搭建Cobbler装机平台

Cobbler概述软件，管理dhcp、tftp、Web服务

自由的导入镜像与ks应答文件

1.解压Cobbler.zip包

# unzip /root/Cobbler.zip -d /

# cd /Cobbler/

# ls

# unzip /Cobbler/cobbler.zip -d /opt/

# ls /opt/cobbler

五、安装cobbler主程序、工具包等

[root@cobbler /]# yum -y install /opt/cobbler/\*.rpm

################################################################

cobbler网络装机部署

1.安装软件 cobbler cobbler-web dhcp tftp-server pykickstart httpd tftp-server

cobbler #cobbler程序包

cobbler-web #cobbler的web服务包

pykickstart #cobbler检查kickstart语法错误

httpd #Apache web服务

dhcp #dhcp服务

tftp-server #tftp服务

2.配置cobbler

[root@cobbler /]# vim /etc/cobbler/settings

next\_server: 192.168.4.168 #设置下一个服务器还为本机

server: 192.168.4.168 #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

3.配置cobbler的dhcp

[root@svr7 /]# vim /etc/cobbler/dhcp.template

:%s /旧/新/g #全文替换

:%s /192.168.1/192.168.4/g #全文替换

5 次替换，共 4 行

4.绝对路径解压cobbler\_boot.tar.gz #众多的引导文件

# tar -tf /Cobbler/cobbler\_boot.tar.gz #查看包里面内容

# tar -xPf /Cobbler/cobbler\_boot.tar.gz #绝对路径释放

# ls /var/lib/cobbler/loaders/

5.启动相关服务

[root@svr7 /]# systemctl restart cobblerd

[root@svr7 /]# systemctl enable cobblerd

[root@svr7 /]# systemctl restart httpd

[root@svr7 /]# systemctl enable httpd

[root@svr7 /]# systemctl restart tftp

[root@svr7 /]# systemctl enable tftp

[root@svr7 /]# systemctl restart rsyncd #同步服务

[root@svr7 /]# systemctl enable rsyncd

6.同步刷新cobbler配置

[root@svr7 /]# cobbler sync #检测配置是否正确

......

\*\*\* TASK COMPLETE \*\*\*

# firefox https://192.168.4.168/cobbler\_web

用户名：cobbler

密码：cobbler

##########################################################################################################

cobbler应用

cobbler import --path=挂载点 --name=导入系统命名（随意起）

导入安装镜像数据

# mount /dev/cdrom /dvd

# ls /dvd/

# cobbler import --path=/dvd --name=CentOS7

cobbler导入的镜像放在：/var/www/cobbler/ks\_mirror

[root@cobbler /]# cobbler list #查看有哪些系统

distros:

CentOS7-x86\_64 #安装客户端至少2G内存

profiles:

CentOS7-x86\_64

##############################################

删除

# cobbler list

# cobbler profile remove --name=CentOS7-x86\_64 #删除菜单信息

# cobbler distro remove --name=CentOS7-x86\_64 #删除镜像信息

###########################################

# umount /dvd/

# mount /dev/cdrom /dvd 将光盘换成RHEL7

# ls /dvd/

安装完成机器 默认root 密码 cobbler

############################################################################################################

自定义应答文件：开头注释行删除

[root@cobbler ~]# system-config-kickstart #生成ks文件

必须默认kickstart文件存放位置：/var/lib/cobbler/kickstarts/

[root@cobbler ~]# cobbler list

修改kickstart文件：

[root@cobbler ~]# cobbler profile edit --name=CentOS7.4-A --kickstart=/var/lib/cobbler/kickstarts/自定义.cfg

[root@cobbler ~]# cobbler profile report

[root@cobbler ~]# cobbler sync #同步配置

安装后脚本:安装系统后自动搭建Yum

服务端PXE服务器：1.共享光盘所有内容 2.搭建FTP或Web服务

客户端：书写一个配置文件

方式一：

echo '

[rhel7]

name=rhel7.4

baseurl=http://192.168.4.7/rhel7

enabled=1

gpgcheck=0' > /etc/yum.repos.d/rhel7.repo

方式二：

[root@svr7 ~]# cd /var/www/html/

[root@svr7 html]# cat rhel7.repo

[rhel7]

name=rhel7.4

baseurl=http://192.168.4.7/rhel7

enabled=1

gpgcheck=0

[root@svr7 ~]# vim /var/www/html/ks.cfg

%post --interpreter=/bin/bash

useradd lisi

echo 123 | passwd --stdin lisi

cd /etc/yum.repos.d/

wget http://192.168.4.7/rhel7.repo

%end

##############################################

在真机上，利用clone-vm7新建一台虚拟机，名字:PXE-Server

1.设置防火墙为trusted

2.当前及永久关闭SELinux

3.配置IP地址：192.168.4.122/24

4.搭建Yum仓库

5.主机名：PXE.tedu.cn

######################################################

搭建一键装机平台，搭建一键装机不同的系统

一 、

搭建DHCP服务

1.安装dhcp

2.修改配置文件

subnet 192.168.4.0

netmask 255.255.255.0 {

range 192.168.4.180 192.168.4.230;

next-server 192.168.4.122;

filename "pxelinux.0";

}

3.重起dhcpd服务，设置开机自起动

#####################################################

二、搭建tftp

1.安装tftp-server

2.启动tftp服务，设置开机自起动

3. 部署pxeliunx.0

# yum provides \*/pxelinux.0

# rpm -ql syslinux | grep pxelinux.0

# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/

# ls /var/lib/tftpboot/

4. 部署光盘内容

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

[root@PXE ~]# mkdir /var/www/html/rhel6

[root@PXE ~]# mkdir /var/www/html/rhel7

[root@PXE ~]# mount /dev/cdrom /mnt/ #光盘为rhel7

[root@PXE ~]# ls /mnt/

[root@PXE ~]# cp -r /mnt/\* /var/www/html/rhel7/

[root@PXE ~]# umount /mnt

[root@PXE ~]# mount /dev/cdrom /mnt/ #光盘为rhel6

[root@PXE ~]# ls /mnt/

[root@PXE ~]# cp -r /mnt/\* /var/www/html/rhel6/

[root@PXE ~]# du -sh /var/www/html/rhel7/

[root@PXE ~]# du -sh /var/www/html/rhel6/

5.部署rhel7、rhel6 的启动内核与驱动程序

# mkdir /var/lib/tftpboot/rhel6

# mkdir /var/lib/tftpboot/rhel7

# cp /var/www/html/rhel6/isolinux/vmlinuz

/var/www/html/rhel6/isolinux/initrd.img

/var/lib/tftpboot/rhel6/

# cp /var/www/html/rhel7/isolinux/vmlinuz

/var/www/html/rhel7/isolinux/initrd.img

/var/lib/tftpboot/rhel7/

# ls /var/lib/tftpboot/rhel7

# ls /var/lib/tftpboot/rhel6

6.部署图形模块与背景

# cp /var/www/html/rhel6/isolinux/vesamenu.c32

/var/lib/tftpboot/

# rpm -ql syslinux | grep jpg

# cp /usr/share/doc/syslinux-4.05/sample/syslinux\_splash.jpg /var/lib/tftpboot/

# ls /var/lib/tftpboot/

7.部署菜单文件

# mkdir /var/lib/tftpboot/pxelinux.cfg

# cp /var/www/html/rhel6/isolinux/isolinux.cfg

/var/lib/tftpboot/pxelinux.cfg/default

# chmod 644 /var/lib/tftpboot/pxelinux.cfg/default

# ls -l /var/lib/tftpboot/pxelinux.cfg/default

8.修改菜单文件内容

# vim /var/lib/tftpboot/pxelinux.cfg/default

......

menu background syslinux\_splash.jpg

menu title Welcome to PXE Server!

......

label linux

menu label Install RHEL7.2

kernel rhel7/vmlinuz

append initrd=rhel7/initrd.img

label vesa

menu label Install RHEL6.7

kernel rhel6/vmlinuz

append initrd=rhel6/initrd.img

label local

menu label Boot from local drive

menu default

#读秒结束最后默认选择

localboot 0xffff

#####################################################

三、搭建httpd服务共享光盘所有内容

1.启动服务

[root@PXE /]# systemctl restart httpd

[root@PXE /]# systemctl enable httpd

2.测试访问

[root@PXE /]# firefox http://192.168.4.122/rhel6

[root@PXE /]# firefox http://192.168.4.122/rhel7

####################################################

四、生成ks文件

[root@PXE /]# yum -y install system-config-kickstart

修改Yum客户端配置文件的标示名

[development]

[root@PXE /]# system-config-kickstart #先看“软件包选择”

软件包选择： 在“桌面”一栏选择----->第一个为GNOME 则为rhel7

RHEL7的文件系统为xfs

#####################################################

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

执行向rhel6的光盘

[development]

name=rhel7

baseurl=http://192.168.4.122/rhel6

enabled=1

gpgcheck=0

[root@PXE ~]# yum clean all #清空Yum缓存

[root@PXE ~]# system-config-kickstart

软件包选择： 在“桌面”一栏选择----->第一个为KDE桌面 则为rhel6

RHEL6的文件系统为ext4

[root@PXE ~]# ls /root/ks\*

/root/ks6.cfg /root/ks7.cfg

#############################################

五、指定ks应答文件

1.共享ks应答文件

[root@PXE ~]# cp /root/ks\* /var/www/html/

[root@PXE ~]# ls /var/www/html/

2.修改菜单文件

# vim /var/lib/tftpboot/pxelinux.cfg/default

label linux

menu label Install RHEL7.2

kernel rhel7/vmlinuz

append initrd=rhel7/initrd.img ks=http://192.168.4.122/ks7.cfg

label vesa

menu label Install RHEL6.7

kernel rhel6/vmlinuz

append initrd=rhel6/initrd.img ks=http://192.168.4.122/ks6.cfg

####################################################