牛犇

第一阶段课程内容共 19 天

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

准备笔记本与笔，先讲解后练习，勤奋的练习

Win2008 虚拟机:

密码为 Taren1

什么是服务器

- 能够为其他计算机提供服务的更高级的电脑

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 协议及配置

- TCP/IP 是最广泛支持的通信协议
- 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 地址的概述

- 作用:用来标识一个主机的网络地址
- 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
- 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

IP 地址的组成: 网络位 与 主机位

网络位: 标识 网络 或者 区域

主机位: 标示 在该区域的第几台主机

子网掩码:用来标识 ip 地址的网络位与主机位

二进制的 1 标识网络位 0 标识主机位

- 默认子网掩码

- A 类地址,255.0.0.0

- B 类地址,255.255.0.0

- C 类地址,255.255.255.0

ip 地址：手工配置 与

自动获取（dhcp 配置 前提网络中必须要自动分配 ip 地址的服务器）

- 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

配置 Win2008 虚拟机，IP 地址与子网掩码

IP:192.168.10.1

子网掩码：255.255.255.0

192.168.10.1/24 ------->24 个网络位

虚拟机克隆：完全复制一台虚拟机

1.关闭虚拟机 win2008

2.右击虚拟机 win2008-----》克隆-----》左击右下角 克隆

3.配置虚拟机 win2008-clone

- 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

windows 防火墙

- 右击桌面网络 ---->属性 ----> Windows 防火墙----->打开和关闭 windows 防火墙

ipconfig windows 系统查看本机网络配置命令

了解内容：

网关地址：

- 什么是网关?
  - 从一个网络连接到另一个网络的“关口”
  - 不同网络之间通信
  - 不同网络之间通信借助路由器设备

DNS 服务器地址：域名的服务器

将域名解析为对应服务器的 ip 地址

192.168.10.2/24

ping 命令测试计算机之间通信

左下角 开始------->运行------->cmd------>确定

两台虚拟机关闭 windows 防火墙

- 右击桌面网络 ---->属性 ----> Windows 防火墙----->打开和关闭 windows 防火墙

ipconfig windows 系统查看本机网络配置命令

了解内容：

网关地址：

- 什么是网关?
  - 从一个网络连接到另一个网络的“关口”
  - 不同网络之间通信
  - 不同网络之间通信借助路由器设备

DNS 服务器地址：域名的服务器

将域名解析为对应服务器的 ip 地址

---

Linux 是一种操作系统

操作系统: 一堆软件的集合，让计算机硬件正常的工作

- UNIX 诞生,1970-1-1

Linux 的诞生

• Linux 之父,Linus Torwalds

用户---------->系统内核---------->硬件

- 版本号:主版本.次版本.修订号

• 是一套公开发布的完整 Linux 系统

- Linux 内核 + 各种应用软件

Linux 文件系统

• 基本作用

- 定义向磁盘介质上存储文档的方法和数据结构,以及

读取文档的规则

- 不同类型的文件系统,其存储/读取方式不一样

- 格式化操作就是建立新的文件系统

• 一块硬盘的“艺术”之旅

- 物理硬盘==>分区规划==>格式化==>读/写文档

• 典型的文件系统类型

- EXT4,第四代扩展文件系统,RHEL6 系列默认

- XFS,高级日志文件系统,RHEL7 系列默认

- SWAP,交换空间(虚拟内存)缓解物理内存的压力

/ 根目录： 所有数据都在此目录下（Linux 系统的起点）

/dev: 所有设备相关的数据

路径：  /dev/nsd/abc/test.txt

hd,表示 IDE 设备

sd,表示 SCSI 设备

/dev/sda  /dev/sdb  /dev/sdc  /dev/sdd

/dev/hda  /dev/hdb  /dev/hdc  /dev/hdd

/dev/sda1  /dev/sda5  /dev/sdc6

/dev/sdc6:SCSI 设备第三块硬盘第 6 个分区

鼠标回到真机：Ctrl+Alt

RHEL7 基本操作

获取命令行界面

- 虚拟控制台切换( Ctrl + Alt + Fn 组合键)
- tty1:图形桌面
- tty2~tty6:字符控制台

伪字符终端

- 右键“在终端中打开”
- 应用程序 --> 工具 --> 终端

以 #号结尾代表  管理员  root

以 $号结尾代表  普通用户

查看及切换目录

• 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]

• ls — List

- 格式:ls [选项]... [目录或文件名]...

```
[root@localhost ~]# pwd                    #显示当前所在位置
[root@localhost ~]# cd   /                 #切换到根目录下
[root@localhost /]# pwd
[root@localhost /]# ls                     #显示当前目录内容
bin   etc   lib64  opt   run   sys  var
boot  home  media  proc  sbin  tmp
dev   lib   mnt    root  srv   usr
[root@localhost /]# ls /home               #显示/home 目录下内容
[root@localhost /]# ls /root               #显示/root 目录下内容
[root@localhost /]# ls /dev                #显示/dev 目录下内容
[root@localhost /]# ls /etc                #显示/etc 目录下内容
[root@localhost /]# ls /var                #显示/var 目录下内容
```

蓝色：目录

黑色：文本文件

以 / 开始的绝对路径

以当前为参照的相对路径

```
[root@localhost /]# cd  /etc/pki/        #绝对路径
[root@localhost pki]# ls
[root@localhost pki]# cd  /etc/pki/CA    #绝对路径
[root@localhost CA]# pwd
[root@localhost CA]# ls
[root@localhost CA]# cd /etc/pki/     #绝对路径
[root@localhost pki]# pwd
[root@localhost pki]# ls
[root@localhost pki]# cd CA           #相对路径
[root@localhost CA]# pwd
[root@localhost CA]# cd  ..         #后退，返回上一级
[root@localhost pki]# pwd
[root@localhost pki]# cd  ..        #后退，返回上一级
[root@localhost etc]# pwd
```

cat 查看文本文件内容

/etc/redhat-release：存储当前系统的版本

```
[root@localhost etc]# cat   /etc/redhat-release
[root@localhost etc]# cat   /etc/passwd
[root@localhost etc]# cat   /etc/group
[root@localhost etc]# cat   /etc/shells
[root@localhost etc]# cat   /etc/default/useradd
[root@localhost etc]# cat   /etc/fstab
```

- 命令行的一般格式,每一部分之间都要有一个空格
  - 命令字　　[选项]...　　[参数 1]　　[参数 2]...

    操作　　如何执行该操作　　作用对象

- 列出内核版本

```
[root@svr7 桌面]# uname -r
3.10.0-327.el7.x86_64
[root@svr7 桌面]# cat  /etc/redhat-release
[root@svr7 桌面]# cat  -n  /etc/redhat-release
[root@svr7 桌面]# cat  -n  /etc/shells
[root@svr7 桌面]# cat  -n  /etc/passwd
```

ls 命令

- 常用命令选项

- -l :以长格式显示，显示详细属性

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

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

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

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

[root@localhost /]# ls  -l   /etc/fstab

[root@localhost /]# ls  -l   /etc/shadow
```

- 列出 CPU 处理器信息

```
[root@svr7 桌面]# lscpu
```

- 检查内存大小、空闲情况

```
[root@svr7 桌面]# cat /proc/meminfo
```

- 列出已激活的网卡连接信息

  eth0:第一张网卡名字

```
[root@localhost /]# ifconfig

[root@localhost /]# ifconfig eth0 192.168.1.1  #临时设置 IP

[root@localhost /]# ifconfig

[root@localhost /]# ping 127.0.0.1
```

  127.0.0.1/8 特殊 ip 地址：永远代表自己

  Ctrl+c ：结束正在运行的命令

• 列出当前系统的主机名称

[root@localhost ～]# hostname

[root@localhost ～]# hostname  nsd.tmooc.cn

[root@localhost ～]# hostname

  新开一个终端验证，提示符号变化

• 关机:poweroff

[root@svr7 ~]# poweroff

• 重启:reboot

[root@svr7 ~]# reboot

创建文档

• mkdir — Make Directory

- 格式:mkdir  [/路径/]目录名...

[root@nsd ~]# ls  /opt/

[root@nsd ~]# mkdir /opt/nsd01

[root@nsd ~]# ls  /opt/

[root@nsd ~]# mkdir nsd02

[root@nsd ~]# ls

[root@nsd ~]# mkdir  /root/abc   /opt/test

[root@nsd ~]# ls  /root/

[root@nsd ~]# ls  /opt/

创建文档

• touch 命令

- 用途:新建空文件

[root@nsd ~]# touch /opt/1.txt  /root/abc.txt

[root@nsd ~]# ls /opt/

[root@nsd ~]# ls /root/

[root@nsd ~]# touch  2.txt

[root@nsd ~]# ls

文本内容操作

• less 分屏阅读工具

• 格式:less [选项] 文件名...

• 按键盘上下键可以滚动

• 全文查找 /root

[root@nsd ~]# less /etc/passwd

• head、tail 命令

- 格式:head -n 数字 文件名

  tail -n 数字 文件名

```
[root@nsd ~]# head -3 /etc/passwd
[root@nsd ~]# head -4 /etc/passwd

[root@nsd ~]# head  /etc/passwd          #默认头十行内容

[root@nsd ~]# tail -3 /etc/passwd
[root@nsd ~]# tail -4 /etc/passwd
[root@nsd ~]# tail  /etc/passwd          #默认尾十行内容
```

- grep 工具
  - 用途:输出包含指定字符串的行
  - 格式:grep [选项]... ’查找条件’ 目标文件

```
[root@nsd ~]# grep  root  /etc/passwd
[root@nsd ~]# grep  bash  /etc/passwd

[root@nsd ~]# grep  bin   /etc/passwd

[root@nsd ~]# grep  sbin  /etc/passwd
```

补全键 Tab : 补全命令 与 补全路径

```
[root@nsd ~]# ls /etc/sysconfig/network-scripts/ifcfg-eth0
```

[root@nsd ~]# if（连续两次 tab）

[root@nsd ~]# ifco（tab）

# ls /et(tab)/sysco(tab)/netw(tab)-(tab)/ifc(tab)e(tab)

[root@nsd ~]# cat  /et(tab)/redh(tab)-r(tab)

绿色：可以执行的程序

- Linux 命令
  - 用来实现某一类功能的指令或程序
  - 命令的执行依赖于解释器(例如:/bin/bash)

    默认解释器：/bin/bash

    用户------->解释器------>内核--------->硬件

命令行的一般格式

- 基本用法
  - 命令字　[选项]...　[参数 1]　[参数 2]...
- 快捷键
  - Ctrl + c : 废弃当前编辑的命令行（结束正在执行的命令）
  - Esc + . : 粘贴上一个命令的参数
  - 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)

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

[root@nsd ~]# ls -l  Esc + .  #显示详细属性

[root@nsd ~]# cat  Esc + .  #显示文本文件内容

[root@nsd ~]# cat -n  Esc + .  #显示内容加上行号

Alt + .  =  Esc + .  #功能一致

mount 挂载操作

windows 系统显示光盘内容

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

Linux 系统显示光盘内容

光盘文件-------->光驱设备----------> 目录（访问点）

/dev/hdc  IDE 接口

/dev/sr0  SCSI 接口

煤矿-------->洞口

快捷方式：青色

1.通过图形鼠标选择，将光盘文件放入虚拟光驱设备中

2.查看光驱设备文件

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

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

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

3.创建访问点目录，挂载设备。  挂载的作用：提供设备的访问点

[root@nsd ~]# mkdir  /dvd

[root@nsd ~]# ls  /dvd

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

mount: /dev/sr0 写保护，将以只读方式挂载

[root@nsd ~]# ls  /dvd

[root@nsd ~]# ls  /dvd/Packages    #查看 Linux 所有安装包

4.卸载操作

[root@nsd ~]# ls  /dvd

[root@nsd ~]# umount  /dvd

[root@nsd ~]# ls  /dvd

注意事项：

1.[root@nsd dvd]# umount /dvd/

umount: /dvd：目标忙。

(有些情况下通过 lsof(8) 或 fuser(1) 可以

找到有关使用该设备的进程的有用信息)

2.挂载设备尽量挂载到自己创建的目录

ls 列出文档及属性

• 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

- -A:包括名称以 . 开头的隐藏文档

- -d:显示目录本身(而不是内容)的属性

- -h:提供易读的容量单位(K、M 等)

```
[root@nsd dvd]# ls  /root
[root@nsd dvd]# ls  -l  /root
[root@nsd dvd]# ls  -lh  /root
[root@nsd dvd]# ls  -l  /boot
[root@nsd dvd]# ls  -lh  /boot
[root@nsd dvd]# ls  -ld  /boot
[root@nsd dvd]# ls  -A  /root
[root@nsd dvd]# ls -lhA  /root
```

～：用户的家目录

/root：管理员 root 用户的家目录

/home：存放所有普通用户的家目录

~user 表示用户 user 的家目录

```
[root@nsd /]# cd ~root          #切换到 root 用户的家目录
[root@nsd ~]# pwd
/root
[root@nsd ~]# useradd  dc  #创建用户 dc
[root@nsd ~]# cd ~dc          #切换到 dc 用户的家目录
[root@nsd dc]# pwd
[root@nsd dc]# cd ~           #回到当前用户的家目录
[root@nsd ~]# pwd
/root
```

. 表示当前目录

.. 表示父目录

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@nsd ~]# ls /dev/tty*
[root@nsd ~]# ls /etc/*tab
[root@nsd ~]# ls /etc/*.conf
[root@nsd ~]# ls /boot/vm*
```

```
[root@nsd ~]# ls /boot/init*
[root@nsd ~]# ls /dev/tty?
[root@nsd ~]# ls /dev/tty??
[root@nsd ~]# ls /dev/tty???
```

- 针对不确定的文档名称,以特殊字符表示

- [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@nsd ~]# ls /dev/tty[3-8]
[root@nsd ~]# ls /dev/tty[0-9]
[root@nsd ~]# ls /dev/tty{1,3,5,7,9,27}
[root@nsd ~]# ls /dev/tty{1,3,5,7,9,27,S0}
[root@nsd ~]# ls /etc/{fs,init}tab
```

请显示/dev 目录下，tty20 到 tty30 所有的设备文件？

```
[root@nsd ~]# ls /dev/tty2[0-9]  /dev/tty30
[root@nsd ~]# ls /dev/tty{2[0-9],30}
```

别名的定义 ： 简化复杂的命令

```
[root@nsd ~]# alias  hn='hostname'       #定义别名
[root@nsd ~]# alias  myls='ls   -lh'     #定义别名
[root@nsd ~]# alias                      #显示所有定义的别名
```

```
[root@nsd ~]# hn
[root@nsd ~]# myls  /root/
[root@nsd ~]# unalias  hn        #删除别名 hn
[root@nsd ~]# hn
bash: hn: 未找到命令...
[root@nsd ~]#
```

复制、 删除、 移动

- mv — Move
  - 格式:mv [选项]... 原文件...  目标路径

```
[root@nsd ~]# mkdir  /mnt/nsd01
[root@nsd ~]# touch  /mnt/1.txt
[root@nsd ~]# ls  /mnt/
[root@nsd ~]# mv  /mnt/1.txt    /mnt/nsd01
[root@nsd ~]# ls  /mnt/
[root@nsd ~]# ls  /mnt/nsd01/
[root@nsd ~]# ls  /opt/
[root@nsd ~]# ls  /mnt/nsd01
[root@nsd ~]# mv  /opt/rh    /mnt/nsd01/
[root@nsd ~]# ls  /opt/
```

[root@nsd ~]# ls  /mnt/nsd01

重命名：路径不变的移动

[root@nsd ~]# ls  /mnt/

[root@nsd ~]# mv  /mnt/nsd01/   /mnt/haha

[root@nsd ~]# ls  /mnt/

[root@nsd ~]# mv  /mnt/haha/   /mnt/stu

[root@nsd ~]# ls  /mnt/

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

删除

   -r: 递归，目录本身以及目录下所有

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

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

• mkdir — Make Directory

- 格式:mkdir [-p] [/路径/]目录名...

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

[root@nsd ~]# mkdir -p  /opt/a/b/c/d

[root@nsd ~]# ls /opt/

[root@nsd ~]# ls /

[root@nsd ~]# rm -rf /vod

[root@nsd ~]# ls /

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

[root@nsd ~]# ls /opt/

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@nsd ~]# rm -rf  /opt/*

[root@nsd ~]# ls  /opt/

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

[root@nsd ~]# ls  /opt/

[root@nsd ~]# cp  /boot/   /opt/

cp: 略过目录"/boot/"

[root@nsd ~]# ls  /opt/

[root@nsd ~]# cp -r  /boot/   /opt/

[root@nsd ~]# ls  /opt/

支持多个参数：把最后一个参数作为目标，其他作为源

[root@nsd ~]# cp -r /etc/fstab  /etc/shadow /home/

/etc/redhat-release  /opt/

[root@nsd ~]# ls /opt/

强制覆盖

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

[root@nsd ~]# cp -r /boot/ /opt/

[root@nsd ~]# ls /opt/

[root@nsd ~]# cp -r /boot/  /opt/

   Ctrl+c 终止运行

[root@nsd ~]# \cp -r /boot/ /opt/   #在本次操作，临时取消别名

[root@nsd /]# cd /opt/

[root@nsd opt]# ls

boot

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

[root@nsd opt]# ls

boot  home

[root@nsd opt]#

[root@nsd opt]# cp  /etc/redhat-release   /opt/

[root@nsd opt]# ls  /opt/

[root@nsd opt]# cp  /etc/redhat-release   /opt/red

[root@nsd opt]# ls  /opt/

前提：

[root@nsd /]# rm -rf /opt/*

[root@nsd /]# ls /opt/

第一次命名操作的意义：复制/home 到/opt/ 重命名为 test

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

第二次命名操作的意义：复制/home 到/opt/test 目录下

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

vim 文本编辑工具

vim 工作模式：    命令模式      输入模式     末行模式

[root@server0 ~]# vim   /opt/haxi.txt

i 键

命 --------------》 输入模式（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

末行模式下　：wq　保存并退出

：q!　强制不保存并退出

教学环境介绍

- 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/软件素材等资源

首先保证 classroom 优先开启机器

server 与 desktop　用户 root 密码 为 redhat

还原三台虚拟机器

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

虚拟机 Server：　利用 root 密码 为 redhat

查询 Server 虚拟机 IP 地址：　172.25.0.11

查询 Server 虚拟机主机名：　server0.example.com

查询 Server 虚拟机系统版本：　7.0

虚拟机 Desktop：　利用 root 密码 为 redhat

查询 Desktop 虚拟机 IP 地址：　172.25.0.10

查询 Desktop 虚拟机主机名：

desktop0.example.com

查询 Desktop 虚拟机系统版本：  7.0

可以相互通信

真机远程管理

默认真机远程管理虚拟机 Desktop 和虚拟机 Server

ssh  用户名@对方 IP 地址

以什么样的用户身份，登陆到那个服务器

```
[root@room9pc01 ~]# ssh root@172.25.0.11
Last login: Tue Apr  3 10:12:33 2018 from 172.25.0.250
[root@server0 ~]# hostname
server0.example.com
[root@server0 ~]# ifconfig
[root@server0 ~]#  exit                #退出远程管理
```

在真机上：Ctrl + shfit + t  新开一个终端

分别远程管理虚拟机 Server0 与虚拟机 Desktop0

ssh  root@172.25.0.11

ssh  root@172.25.0.10

软件包的管理

虚拟机 Server0

1.关闭虚拟机 Server0，图形添加光驱设备

2.挂载/dev/cdrom 设备，具备软件包

[root@server0 ~]# mkdir /dvd

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

[root@server0 ~]# ls /dvd

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

• RPM Package Manager,RPM 包管理器

- rpm -q 软件名...

- rpm -ivh 软件名-版本信息.rpm...

- rpm -e 软件名...

[root@server0 ~]# rpm -q firefox #查询软件包是否安装

firefox-24.5.0-1.el7.x86_64

[root@server0 ~]# rpm -q haha

[root@server0 ~]# rpm -q zip

[root@server0 ~]# rpm -q vsftpd

# rpm -ivh /dvd/Packages/vsftpd-3.0.2-22.el7.x86_64.rpm

警告：/dvd/Packages/vsftpd-3.0.2-22.el7.x86_64.rpm: 头 V3 RSA/SHA256 Signature, 密钥 ID fd431d51: NOKEY

[root@server0 ~]# rpm -q vsftpd　　#查询是否安装成功

[root@server0 ~]# rpm -e vsftpd　　#卸载软件包

[root@server0 ~]# rpm -q vsftpd　　#查询是否卸载成功

了解：导入红帽的签名信息

# rpm --import /dvd/RPM-GPG-KEY-redhat-release

常见的报错：

错误：依赖检测失败

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 安装软件

服务端: classroom 虚拟机搭建 Web 服务，共享光盘所有内容

http://classroom.example.com/content/rhel7.0/x86_64/dvd/

客户端: Server 虚拟机

-X:在本地运行对方的图形程序

[root@server0 ~]# exit  #退出远程管理

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

Last login: Tue Apr  3 14:04:11 2018 from 172.25.0.250

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

配置文件的路径：/etc/yum.repos.d/*.repo

错误的配置文件，会影响正确配置文件

```
[root@server0 ~]# rm -rf /etc/yum.repos.d/*
[root@server0 ~]# vim /etc/yum.repos.d/dvd.repo
 [rhel7]                  #仓库标识
 name=nsd 1803            #仓库的描述信息
baseurl=http://classroom.example.com/content/rhel7.0/x86_6
4/dvd/                  #指定 Yum 仓库的位置
 enabled=1                #启用该文件
 gpgcheck=0               #关闭检测软件包签名功能
[root@server0 ~]# yum repolist    #列出仓库信息
```

Yum 的使用

安装： yum -y install 软件名

卸载： yum remove 软件名

```
[root@server0 ~]# yum -y install xeyes
[root@server0 ~]# yum -y install sssd
[root@server0 ~]# yum -y install gcc
[root@server0 ~]# yum -y install httpd
[root@server0 ~]# yum -y install system-config-kickstart
[root@server0 ~]# xeyes &          #放入后台
[root@server0 ~]# killall xeyes   #杀死该程序
```

清空 Yum 的缓存

```
[root@server0 ~]# yum clean all
```

下载软件包

- 使用 wget 下载工具
  - wget 软件包的 URL 网址
  - wget 软件包的 URL 网址 -O /目录路径/新文件名

```
# wget
http://classroom.example.com/content/rhel7.0/x86_64/errata
/Packages/kernel-3.10.0-123.1.2.el7.x86_64.rpm
```

默认下载到当前目录

升级内核

```
[root@server0 ~]# uname -r
3.10.0-123.el7.x86_64
# rpm -ivh kernel-3.10.0-123.1.2.el7.x86_64.rpm
[root@server0 ~]# uname -r
3.10.0-123.el7.x86_64
[root@server0 ~]# reboot
[root@room9pc01 ~]# ssh -X root@172.25.0.11
Last login: Tue Apr  3 14:14:57 2018 from 172.25.0.250
[root@server0 ~]# uname -r
```

配置网络

一、配置永久的主机名,配置文件/etc/hostname

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

A.tedu.cn

[root@server0 ~]# exit

登出

Connection to 172.25.0.11 closed.

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

[root@A ~]# hostname

A.tedu.cn

[root@A ~]#

二、配置永久 IP 地址，子网掩码，网关地址

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

1.查询识别的网卡名称

[root@A ~]# nmcli connection show

2.配置 IP 地址，子网掩码，网关地址

# nmcli  connection  modify  'System eth0'

ipv4.method  manual

ipv4.addresses  '172.25.0.100/24  172.25.0.254'

connection.autoconnect　yes

nmcli connection 修改网络配置 ’网卡名’

ipv4.方法　手工配置

ipv4.地址 ’IP 地址/子网掩码　网关地址’

每次开机自动启用配置

3.激活配置

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

关闭终端，新开一个全新的终端进行远程管理

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

三、配置 DNS 服务器地址

作用：将域名解析为对应的 IP 地址

/etc/resolv.conf　DNS 服务器地址的配置文件

vim 命令模式　dd　删除整行内容

10dd 删除 10 整行内容

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

nameserver　172.25.254.254

验证：

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

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

还原三台虚拟机器

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

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

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

用户管理

用户帐号： 1.可以登陆操作系统 2.实现访问控制（不同的用户权限不同）

组帐号： 方便对用户管理（权限方面）

唯一标识： UID GID

管理员的UID：0

一个用户必须至少属于一个组

组分类：基本组（私有组） 附加组（从属组 公共组）

添加用户

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:基本组的GID:描述信息:家目录:解释器程序

- 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 Shell、-G 附加组

```
[root@server0 ~]# useradd nsd01
[root@server0 ~]# id nsd01          #显示用户基本信息
[root@server0 ~]# useradd nsd03
[root@server0 ~]# grep nsd /etc/passwd
[root@server0 ~]# useradd -u 1200 nsd04   #创建用户指定 UID
[root@server0 ~]# grep nsd /etc/passwd
[root@server0 ~]# useradd nsd05
[root@server0 ~]# grep nsd /etc/passwd
[root@server0 ~]# useradd -d /opt/test nsd06  #指定家目录
[root@server0 ~]# grep nsd06 /etc/passwd
nsd06:x:1202:1202::/opt/test:/bin/bash
[root@server0 ~]# ls /opt/
[root@server0 ~]# ls -A /opt/test
[root@server0 ~]# groupadd tarena           #创建组
[root@server0 ~]# useradd -G tarena nsd07   #指定加入附加组
[root@server0 ~]# id nsd07
[root@server0 ~]# useradd -d /opt/abc -G tarena  nsd08
```

[root@server0 ~]# id nsd08

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

-s 登录 Shell（解释器）

用户 -------> 解释器 ------->内核------->硬件

/sbin/nologin :禁止用户登陆系统

[root@server0 ~]# useradd -s /sbin/nologin nsd09

[root@server0 ~]# grep nsd09 /etc/passwd

管道 操作 | :

将前面命令的输出结果，交由后面命令在处理一次，最后的输出以最后一条命令为准

# cat -n /etc/passwd

# cat -n /etc/passwd | head -12

# cat -n /etc/passwd | head -12 | tail -5

# ifconfig | head -2

# ls --help | less

# grep root /etc/passwd

# grep root /etc/passwd | grep bash

重定向

>:覆盖重定向

>>:追加重定向

将前面命令的输出结果，写入文本文件中

```
[root@server0 ~]# head -3 /etc/passwd

[root@server0 ~]# head -3 /etc/passwd  > /opt/pass.txt

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

[root@server0 ~]# hostname

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

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

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

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

[root@server0 ~]# head -2 /etc/passwd >> /opt/pass.txt

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

echo  在屏幕输出用户想输出的内容

```
[root@server0 ~]# echo A.tedu.cn

A.tedu.cn

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

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

[root@server0 ~]# echo nameserver 172.25.254.255

# echo nameserver 172.25.254.254 > /etc/resolv.conf
```

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

[root@server0 ~]# echo 123456 > /opt/1.txt

[root@server0 ~]# cat /opt/1.txt

设置登录密码

用户密码信息存放在 /etc/shadow 文件

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

用户名:密码加密字符串:上一次修改密码的时间

- 使用 passwd 命令

交互式： passwd [用户名]

非交互式：echo '密码' | passwd --stdin 用户名

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

[root@server0 ~]# su - nsd01 #临时切换用户身份

[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　　#退出临时身份，回到 root

logout

[root@server0 ~]#

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 Shell

- -G 附加组

[root@server0 ~]# useradd nsd11

[root@server0 ~]# id nsd11

[root@server0 ~]# grep nsd11 /etc/passwd

[root@server0 ~]# usermod -u 1300 -G tarena

-d /opt/nsd11 -s /sbin/nologin nsd11

[root@server0 ~]# grep nsd11 /etc/passwd

[root@server0 ~]# id nsd11

补充： 可以利用 vim 修改/etc/passwd 文件内容

删除用户

• 使用 userdel 命令

- userdel [-r] 用户名 #将该用户的家目录一起删除

[root@server0 ~]# userdel nsd01

[root@server0 ~]# userdel nsd05

管理组账号

组基本信息存放在 /etc/group 文件

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

root:x:0:

组名:x:GID:组的成员列表

添加组

组基本信息存放在 /etc/group 文件

• 使用 groupadd 命令

- groupadd [-g 组 ID] 组名

[root@server0 ~]# groupadd stugrp

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

[root@server0 ~]# useradd harry

[root@server0 ~]# useradd natasha

[root@server0 ~]# useradd jack

[root@server0 ~]# useradd kenji

管理组成员

• 使用 gpasswd 命令

- gpasswd -a 用户名 组名

- gpasswd -d 用户名 组名

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

[root@server0 ~]# gpasswd -a kenji stugrp   #添加用户到组

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

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

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

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

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

[root@server0 ~]# gpasswd -d kenji stugrp   #从组中删除用户

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

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

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

删除组

• 使用 groupdel 命令

- groupdel 组名

```
[root@server0 ~]# groupdel stugrp

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

tar 备份与恢复

归档和压缩 ： 1.方便对零散文件管理　　2.减少空间的占用

常见的压缩格式及命令工具:

gzip ----> .gz

bzip2 ---->.bz2

xz ----> .xz

- tar 集成备份工具
  - -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:指定释放的位置

格式： tar 选项　　/路径/压缩包名字　　被归档及压缩的文件

红色：压缩文件

```
#  tar  -zcf   file.tar.gz        /home/  /etc/passwd
#  ls
#  tar -zcf  /opt/nsd01.tar.gz     /home/ /etc/passwd
```

```
#  ls /opt/

#  mkdir /nsd

#  ls /nsd

#  tar -xf  /opt/nsd01.tar.gz    -C  /nsd/

#  ls /nsd

#  tar -tf /opt/nsd01.tar.gz
```

- -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）:指定释放的位置

```
[root@server0 ~]# tar -jcf /root/backup.tar.bz2 /usr/local/

[root@server0 ~]# ls /root/

[root@server0 ~]# tar -tf /root/backup.tar.bz2
```

#查看归档文件的内容

- tar -zcf 备份文件.tar.gz 被备份的文档....
- tar -jcf 备份文件.tar.bz2 被备份的文档....
- tar -Jcf 备份文件.tar.xz 被备份的文档....

NTP 时间同步

[root@server0 ~]# date

2018 年 04 月 04 日 星期三 16:10:18 CST

[root@server0 ~]# date +%Y　　#显示年份

[root@server0 ~]# date +%m　　#显示月份

[root@server0 ~]# date +%d　　#显示日期

[root@server0 ~]# date +%Y%m%d　　#显示年月日

[root@server0 ~]# date +%Y-%m-%d　#显示年月日

[root@server0 ~]# date +%F　　#显示年月日

修改时间：

[root@server0 ~]# date -s　"年-月-日　　时：分：秒"

[root@server0 ~]# date -s　"2008-9-6　12:10:11"

[root@server0 ~]# date

服务：NTP 服务器为客户机提供标准时间

服务端：classroom 时间同步服务器

客户端：虚拟机 Server

1.安装 chrony 客户端软件,与时间服务端沟通的软件

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

[root@server0 ~]# rpm -q chrony

chrony-1.29.1-1.el7.x86_64

2.修改配置文件/etc/chrony.conf,指定服务端位置

```
[root@server0 ~]# vim  /etc/chrony.conf
  #server 0.rhel.pool.ntp.org iburst   #开头加上# 注释
  #server 1.rhel.pool.ntp.org iburst   #开头加上# 注释
  #server 2.rhel.pool.ntp.org iburst   #开头加上# 注释
  server classroom.example.com iburst  #指定服务端位置
```

3. 启动客户端 chronyd 服务

daemon: 守护进程，守护程序

```
[root@server0 ~]# systemctl restart chronyd  #重起服务
[root@server0 ~]# systemctl enable chronyd   #设置随机自启动
```

4.验证：

```
[root@server0 ~]# date  -s "2000-10-1 10:12:30"
2000 年 10 月 01 日 星期日 10:12:30 CST
[root@server0 ~]# date
[root@server0 ~]# systemctl restart chronyd
[root@server0 ~]# date
[root@server0 ~]# date
```

还原三台虚拟机器

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

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

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

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

r： cat head tail less

w： vim

x： 可以运行该文件

• 权限适用对象(归属)

- 所有者(属主):拥有此文件/目录的用户-user u
- 所属组(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

查看权限

• 使用 ls -l 命令

- ls -ld  文件或目录...

  以 - 开头：文本文件

  以 d 开头：目录

  以 l 开头：快捷方式

• 使用 chmod 命令

- chmod [-R] 归属关系+-=权限类别 文档...

- [-R] 递归设置权限

```
[root@server0 ~]# mkdir /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=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01
[root@server0 ~]# chmod u=rwx,g=rwx,o=--- /nsd01
[root@server0 ~]# ls -ld /nsd01
[root@server0 ~]# chmod ugo=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01
```

管理员 root 具备一切权限

判别权限的方法：　　匹配及停止原则

1. 查看用户对于文档的身份，属于那个归属关系　　所有者>所属组>其他人

2. 查看相应归属关系位置的权限

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

Permission denied : 权限不足

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 zhangsan 能够在此目录下创建子目录

切换用户　su　-　zhangsan

chmod o+w　/nsddir/

2）使用户 zhangsan 不能够在此目录下创建子目录

chmod o-w　/nsddir/

3）使用户 zhangsan 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/

- 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...

```
[root@server0 ~]# mkdir /nsd03
[root@server0 ~]# ls -ld /nsd03
[root@server0 ~]# groupadd tedu
[root@server0 ~]# chown  zhangsan:tedu /nsd03
[root@server0 ~]# ls -ld /nsd03
[root@server0 ~]# chown root /nsd03      #修改所有者
[root@server0 ~]# ls -ld /nsd03
[root@server0 ~]# chown :zhangsan /nsd03   #修改所属组
[root@server0 ~]# ls -ld /nsd03
```

利用 root 用户新建/nsd05 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 /]# useradd  gelin02
```

[root@server0 /]# groupadd  tarena

[root@server0 /]# chown gelin01:tarena  /nsd05

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=--- /nsd05

3）使用户 gelin02 能进入、查看此目录

[root@server0 /]# gpasswd -a  gelin02  tarena  #用户加入组

4)将 gelin01 加入 tarena 组, 将 nsd05 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 /nsd05

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

- 继承父目录的所属组身份

[root@server0 ~]# mkdir /nsd07

[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# chown :tarena /nsd07

```
[root@server0 ~]# ls -ld /nsd07
[root@server0 ~]# mkdir /nsd07/test01
[root@server0 ~]# ls -ld /nsd07/test01
[root@server0 ~]# chmod g+s /nsd07
[root@server0 ~]# ls -ld /nsd07
[root@server0 ~]# mkdir /nsd07/test02
[root@server0 ~]# ls -ld /nsd07/test02
[root@server0 ~]# ls -ld /nsd07/test01
```

ACL 权限（ACL 策略）

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 /nsd09
[root@server0 ~]# chmod o=--- /nsd09
[root@server0 ~]# ls -ld /nsd09
```

```
[root@server0 ~]# su - zhangsan
[zhangsan@server0 ~]$ cd /nsd09
-bash: cd: /nsd09: Permission denied
[zhangsan@server0 ~]$ exit
logout
[root@server0 ~]# setfacl -m u:zhangsan:rx /nsd09   #设置
ACL
[root@server0 ~]# getfacl /nsd09    #查看 ACL 权限
[root@server0 ~]# su - zhangsan
[zhangsan@server0 ~]$ cd /nsd09
[zhangsan@server0 nsd09]$ pwd
[zhangsan@server0 nsd09]$ exit
logout
[root@server0 ~]# 还原三台虚拟机器
[root@room9pc01 ~]# rht-vmctl reset classroom
[root@room9pc01 ~]# rht-vmctl reset server
[root@room9pc01 ~]# rht-vmctl reset desktop
```

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

r： cat  head  tail  less

w： vim

x： 可以运行该文件

• 权限适用对象(归属)

- 所有者(属主):拥有此文件/目录的用户-user  u
- 所属组(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

查看权限

• 使用 ls -l 命令

- ls -ld  文件或目录...

以 - 开头：文本文件

以 d 开头：目录

以 l 开头：快捷方式

• 使用 chmod 命令

- chmod [-R] 归属关系+-=权限类别 文档...

- [-R] 递归设置权限

```
[root@server0 ~]# mkdir /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=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01
[root@server0 ~]# chmod u=rwx,g=rwx,o=--- /nsd01
[root@server0 ~]# ls -ld /nsd01
[root@server0 ~]# chmod ugo=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01
```

管理员 root 具备一切权限

判别权限的方法：　　匹配及停止原则

1. 查看用户对于文档的身份，属于那个归属关系　　所有者>所属组>其他人

2. 查看相应归属关系位置的权限

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

Permission denied : 权限不足

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 zhangsan 能够在此目录下创建子目录

切换用户 su - zhangsan

chmod o+w /nsddir/

2）使用户 zhangsan 不能够在此目录下创建子目录

chmod o-w /nsddir/

3）使用户 zhangsan 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/

- 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...

[root@server0 ~]# mkdir /nsd03

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

[root@server0 ~]# groupadd tedu

[root@server0 ~]# chown  zhangsan:tedu /nsd03

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

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

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

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

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

利用 root 用户新建/nsd05 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 /]# useradd  gelin02

[root@server0 /]# groupadd  tarena

[root@server0 /]# chown gelin01:tarena  /nsd05

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=--- /nsd05

3）使用户 gelin02 能进入、查看此目录

[root@server0 /]# gpasswd -a  gelin02  tarena  #用户加入组

4)将 gelin01 加入 tarena 组, 将 nsd05 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 /nsd05
```

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
- 继承父目录的所属组身份

```
[root@server0 ~]# mkdir /nsd07
[root@server0 ~]# ls -ld /nsd07
[root@server0 ~]# chown :tarena /nsd07
[root@server0 ~]# ls -ld /nsd07
[root@server0 ~]# mkdir /nsd07/test01
[root@server0 ~]# ls -ld /nsd07/test01
[root@server0 ~]# chmod g+s /nsd07
[root@server0 ~]# ls -ld /nsd07
[root@server0 ~]# mkdir /nsd07/test02
```

[root@server0 ~]# ls -ld /nsd07/test02

[root@server0 ~]# ls -ld /nsd07/test01

ACL 权限（ACL 策略）

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

• 使用 getfacl、setfacl 命令

- getfacl 文档...

- setfacl [-R] -m u:用户名:权限类别 文档...

- setfacl [-R] -m g:组名:权限类别 文档...

- setfacl [-R] -b 文档... #清除所有的 ACL

- setfacl -x u:用户名 文档... #删除指定的 ACL

- [-R] :递归设置 ACL 策略

[root@server0 ~]# mkdir /nsd10

[root@server0 ~]# setfacl -m u:zhangsan:rwx /nsd10

[root@server0 ~]# setfacl -m u:gelin01:rx /nsd10

[root@server0 ~]# setfacl  -m  u:gelin02:rx  /nsd10

[root@server0 ~]# getfacl  /nsd10

[root@server0 ~]# setfacl  -x  u:ge 还原三台虚拟机器

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

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

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

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

r： cat  head  tail  less

w： vim

x： 可以运行该文件

• 权限适用对象(归属)

- 所有者(属主):拥有此文件/目录的用户-user  u
- 所属组(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

查看权限

- 使用 ls -l 命令
  - ls -ld  文件或目录...
    以 - 开头：文本文件
    以 d 开头：目录
    以 l 开头：快捷方式
- 使用 chmod 命令
  - chmod [-R] 归属关系+-=权限类别 文档...
  - [-R] 递归设置权限

```
[root@server0 ~]# mkdir /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=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01
[root@server0 ~]# chmod u=rwx,g=rwx,o=--- /nsd01
[root@server0 ~]# ls -ld /nsd01
[root@server0 ~]# chmod ugo=rwx /nsd01
```

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

管理员 root 具备一切权限

判别权限的方法：　　匹配及停止原则

1. 查看用户对于文档的身份，属于那个归属还原三台虚拟机器

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

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

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

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

r：cat　head　tail　less

w：vim

x：可以运行该文件

• 权限适用对象(归属)

- 所有者(属主):拥有此文件/目录的用户-user　u
- 所属组(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

查看权限

• 使用 ls -l 命令

- ls -ld  文件或目录...

  以 - 开头：文本文件

  以 d 开头：目录

  以 l 开头：快捷方式

• 使用 chmod 命令

- chmod [-R] 归属关系+-=权限类别 文档...

- [-R] 递归设置权限

```
[root@server0 ~]# mkdir /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=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01
[root@server0 ~]# chmod u=rwx,g=rwx,o=--- /nsd01
[root@server0 ~]# ls -ld /nsd01
```

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

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

管理员 root 具备一切权限

判别权限的方法：　　匹配及停止原则还原三台虚拟机器

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

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

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

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

r：cat　head　tail　less

w：vim

x：可以运行该文件

• 权限适用对象(归属)

- 所有者(属主):拥有此文件/目录的用户-user　还原三台虚拟机器

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

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

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

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

r： cat  head  tail  less

w： vim

x： 可以运行该文件

• 权限适用对象(归属)

- 所有者(属主):拥有此文件/目录的用户-user  还原三台虚拟机器

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

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

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

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

r: cat head tail less

w: vim

x: 可以运行该文件

• 权限适用对象(归属)

- 所有者(属主):拥有此文件/目录的用户-user u
- 所属组(属组):拥有此文件/目录的组-group 还原三台虚拟机器

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

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

r：cat  head  tail  less

w：vim

x：可以运行该文件

还原三台虚拟机器

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

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

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

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

r：cat  head  tail  less

w：vim

x：可以运行该文件

• 权限适用对象(归属)

- 所有者(属主):拥有此文件/目录的用户-user 还原三台虚拟机器

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

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

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

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

r: cat head tail less

w: vim

x: 可以运行该文件

• 权限适用对象(归属)

- 所有者(属主):拥有此文件/目录的用户-user u
- 所属组(属组):拥有此文件/目录的组-group 还原三台虚拟机器

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

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

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

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

r： cat  head  tail  less

w： vim

x： 可以运行该文件

• 权限适用对象(归属)

- 所有者(属主):拥有此文件/目录的用户-user  u
- 所属组(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

查看权限

• 使用 ls -l 命令

- ls -ld  文件或目录...

以 - 开头：文本文件

以 d 开头：目录

以 l 开头：快捷方式

- 使用 chmod 命令

- chmod [-R] 归属关系+-=权限类别 文档...

- [-R] 递归设置权限

```
[root@server0 ~]# mkdir /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=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01
[root@server0 ~]# chmod u=rwx,g=rwx,o=--- /nsd01
[root@server0 ~]# ls -ld /nsd01
[root@server0 ~]# chmod ugo=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01
```

管理员 root 具备一切权限

判别权限的方法：　　匹配及停止原则

1. 查看用户对于文档的身份，属于那个归属关系　　所有者>所属组>其他人

2. 查看相应归属关系位置的权限

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

Permission denied : 权限不足

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 zhangsan 能够在此目录下创建子目录

切换用户 su - zhangsan

chmod o+w /nsddir/

2）使用户 zhangsan 不能够在此目录下创建子目录

chmod o-w /nsddir/

3）使用户 zhangsan 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/

- 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...

```
[root@server0 ~]# mkdir /nsd03
[root@server0 ~]# ls -ld /nsd03
[root@server0 ~]# groupadd tedu
[root@server0 ~]# chown  zhangsan:tedu /nsd03
[root@server0 ~]# ls -ld /nsd03
[root@server0 ~]# chown root /nsd03      #修改所有者
[root@server0 ~]# ls -ld /nsd03
[root@server0 ~]# chown :zhangsan /nsd03   #修改所属组
[root@server0 ~]# ls -ld /nsd03
```

利用 root 用户新建/nsd05 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 /]# useradd  gelin02
[root@server0 /]# groupadd  tarena
[root@server0 /]# chown gelin01:tarena  /nsd05
```

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=--- /nsd05

3）使用户 gelin02 能进入、查看此目录

[root@server0 /]# gpasswd -a  gelin02  tarena  #用户加入组

4)将 gelin01 加入 tarena 组, 将 nsd05 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 /nsd05

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

- 继承父目录的所属组身份

[root@server0 ~]# mkdir /nsd07

[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# chown :tarena /nsd07

[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test01

[root@server0 ~]# ls -ld /nsd07/test01

[root@server0 ~]# chmod g+s /nsd07

[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test02

[root@server0 ~]# ls -ld /nsd07/test02

[root@server0 ~]# ls -ld /nsd07/test01

ACL 权限（ACL 策略）

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　g
  - 其他用户:除所有者、所属组以外的用户-other　o

查看权限

- 使用 ls -l 命令
  - ls -ld　文件或目录...

    以 - 开头：文本文件

    以 d 开头：目录

    以 l 开头：快捷方式

• 使用 chmod 命令

- chmod [-R] 归属关系+-=权限类别 文档...

- [-R] 递归设置权限

```
[root@server0 ~]# mkdir /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=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01
[root@server0 ~]# chmod u=rwx,g=rwx,o=--- /nsd01
[root@server0 ~]# ls -ld /nsd01
[root@server0 ~]# chmod ugo=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01
```

管理员 root 具备一切权限

判别权限的方法：　　匹配及停止原则

1. 查看用户对于文档的身份，属于那个归属关系　　所有者>所属组>其他人

2. 查看相应归属关系位置的权限

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

Permission denied : 权限不足

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 zhangsan 能够在此目录下创建子目录

切换用户 su - zhangsan

chmod o+w /nsddir/

2）使用户 zhangsan 不能够在此目录下创建子目录

chmod o-w /nsddir/

3）使用户 zhangsan 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/

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

• 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...

```
[root@server0 ~]# mkdir /nsd03
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# groupadd tedu
[root@server0 ~]# chown  zhangsan:tedu /nsd03
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# chown root /nsd03      #修改所有者
[root@server0 ~]# ls -ld /nsd03
```

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

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

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

利用 root 用户新建/nsd05 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 /]# useradd  gelin02

[root@server0 /]# groupadd  tarena

[root@server0 /]# chown gelin01:tarena  /nsd05

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=--- /nsd05

3）使用户 gelin02 能进入、查看此目录

[root@server0 /]# gpasswd -a  gelin02  tarena  #用户加入组

4)将 gelin01 加入 tarena 组，将 nsd05 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 /nsd05

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

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

- 继承父目录的所属组身份

[root@server0 ~]# mkdir /nsd07

[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# chown :tarena /nsd07

[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test01

```
[root@server0 ~]# ls -ld /nsd07/test01

[root@server0 ~]# chmod g+s /nsd07
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test02
[root@server0 ~]# ls -ld /nsd07/test02

[root@server0 ~]# ls -ld /nsd07/test01
```

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

ACL 权限（ACL 策略）

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　u
  - 所属组(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

查看权限

• 使用 ls -l 命令

- ls -ld  文件或目录...

以 - 开头：文本文件

以 d 开头：目录

以 l 开头：快捷方式

• 使用 chmod 命令

- chmod [-R] 归属关系+-=权限类别 文档...

- [-R] 递归设置权限

```
[root@server0 ~]# mkdir /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=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01

[root@server0 ~]# chmod u=rwx,g=rwx,o=--- /nsd01
[root@server0 ~]# ls -ld /nsd01

[root@server0 ~]# chmod ugo=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01
```

##########

管理员 root 具备一切权限

判别权限的方法：　　匹配及停止原则

1. 查看用户对于文档的身份，属于那个归属关系　　所有者>所属组>其他人

2. 查看相应归属关系位置的权限

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

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

Permission denied : 权限不足

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

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 zhangsan 能够在此目录下创建子目录

切换用户 su - zhangsan

chmod o+w /nsddir/

2）使用户 zhangsan 不能够在此目录下创建子目录

chmod o-w /nsddir/

3）使用户 zhangsan 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/

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

• 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...

[root@server0 ~]# mkdir /nsd03

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

[root@server0 ~]# groupadd tedu

```
[root@server0 ~]# chown  zhangsan:tedu /nsd03
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# chown root /nsd03      #修改所有者
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# chown :zhangsan /nsd03   #修改所属组
[root@server0 ~]# ls -ld /nsd03
```

##########

利用 root 用户新建/nsd05 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 /]# useradd  gelin02
[root@server0 /]# groupadd  tarena
[root@server0 /]# chown gelin01:tarena  /nsd05
```

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=--- /nsd05

3）使用户 gelin02 能进入、查看此目录

[root@server0 /]# gpasswd -a  gelin02  tarena  #用户加入组

4)将 gelin01 加入 tarena 组, 将 nsd05 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 /nsd05

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

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

- 继承父目录的所属组身份

[root@server0 ~]# mkdir /nsd07

```
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# chown :tarena /nsd07
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test01
[root@server0 ~]# ls -ld /nsd07/test01

[root@server0 ~]# chmod g+s /nsd07
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test02
[root@server0 ~]# ls -ld /nsd07/test02

[root@server0 ~]# ls -ld /nsd07/test01
```

##########

ACL 权限（ACL 策略）

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
- 权限适用对象(归属)
  - 所有者(属主):拥有此文件/目录的用户-user　　u
  - 所属组(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

查看权限

- 使用 ls -l 命令
  - ls -ld　文件或目录...

以 - 开头：文本文件

以 d 开头：目录

以 l 开头：快捷方式

- 使用 chmod 命令

- chmod [-R] 归属关系+-=权限类别 文档...

- [-R] 递归设置权限

```
[root@server0 ~]# mkdir /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=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01

[root@server0 ~]# chmod u=rwx,g=rwx,o=--- /nsd01
[root@server0 ~]# ls -ld /nsd01

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

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

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

管理员 root 具备一切权限

判别权限的方法：　　匹配及停止原则

1. 查看用户对于文档的身份，属于那个归属关系　　所有者>所属组>其他人

2. 查看相应归属关系位置的权限

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

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

Permission denied : 权限不足

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

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 zhangsan 能够在此目录下创建子目录

切换用户　su　-　zhangsan

chmod o+w　/nsddir/

2）使用户 zhangsan 不能够在此目录下创建子目录

chmod o-w　/nsddir/

3）使用户 zhangsan 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/

##########

- 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...

```
[root@server0 ~]# mkdir /nsd03
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# groupadd tedu
[root@server0 ~]# chown  zhangsan:tedu /nsd03
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# chown root /nsd03      #修改所有者
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# chown :zhangsan /nsd03   #修改所属组
[root@server0 ~]# ls -ld /nsd03
```

##########

利用 root 用户新建/nsd05 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 /]# useradd  gelin02
[root@server0 /]# groupadd  tarena
[root@server0 /]# chown gelin01:tarena  /nsd05
```

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=--- /nsd05
```

3）使用户 gelin02 能进入、查看此目录

```
[root@server0 /]# gpasswd -a  gelin02  tarena  #用户加入组
```

4)将 gelin01 加入 tarena 组，将 nsd05 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 /nsd05
```

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

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

- 继承父目录的所属组身份

```
[root@server0 ~]# mkdir /nsd07
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# chown :tarena /nsd07
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test01
[root@server0 ~]# ls -ld /nsd07/test01

[root@server0 ~]# chmod g+s /nsd07
[root@server0 ~]# ls -ld /nsd07
```

[root@server0 ~]# mkdir /nsd07/test02

[root@server0 ~]# ls -ld /nsd07/test02

[root@server0 ~]# ls -ld /nsd07/test01

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

ACL 权限（ACL 策略）

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　g
  - 其他用户:除所有者、所属组以外的用户-other　o

查看权限

- 使用 ls -l 命令

- ls -ld  文件或目录...

  以 - 开头：文本文件

  以 d 开头：目录

  以 l 开头：快捷方式

• 使用 chmod 命令

- chmod [-R] 归属关系+-=权限类别 文档...

- [-R] 递归设置权限

```
[root@server0 ~]# mkdir /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=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01
```

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

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

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

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

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

管理员 root 具备一切权限

判别权限的方法：　　匹配及停止原则

1. 查看用户对于文档的身份，属于那个归属关系　　所有者>所属组>其他人

2. 查看相应归属关系位置的权限

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

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

Permission denied : 权限不足

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

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 zhangsan 能够在此目录下创建子目录

切换用户　su　-　zhangsan

chmod o+w　/nsddir/

2）使用户 zhangsan 不能够在此目录下创建子目录

chmod o-w　/nsddir/

3）使用户 zhangsan 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/

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

• 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...

```
[root@server0 ~]# mkdir /nsd03
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# groupadd tedu
[root@server0 ~]# chown  zhangsan:tedu /nsd03
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# chown root /nsd03      #修改所有者
[root@server0 ~]# ls -ld /nsd03

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

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

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

利用 root 用户新建/nsd05 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 /]# useradd  gelin02

[root@server0 /]# groupadd  tarena

[root@server0 /]# chown gelin01:tarena  /nsd05

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=--- /nsd05

3）使用户 gelin02 能进入、查看此目录

[root@server0 /]# gpasswd -a  gelin02  tarena  #用户加入组

4)将 gelin01 加入 tarena 组，将 nsd05 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 /nsd05

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

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

- 继承父目录的所属组身份

[root@server0 ~]# mkdir /nsd07

[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# chown :tarena /nsd07

[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test01

[root@server0 ~]# ls -ld /nsd07/test01

```
[root@server0 ~]# chmod g+s /nsd07
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test02
[root@server0 ~]# ls -ld /nsd07/test02

[root@server0 ~]# ls -ld /nsd07/test01
```

##########

ACL 权限（ACL 策略）

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  u
  - 所属组(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

查看权限

- 使用 ls -l 命令

- ls -ld  文件或目录...

  以 - 开头：文本文件

  以 d 开头：目录

  以 l 开头：快捷方式

- 使用 chmod 命令

- chmod [-R] 归属关系+-=权限类别 文档...

- [-R] 递归设置权限

```
[root@server0 ~]# mkdir /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=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01

[root@server0 ~]# chmod u=rwx,g=rwx,o=--- /nsd01
[root@server0 ~]# ls -ld /nsd01

[root@server0 ~]# chmod ugo=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01
```

##########

管理员 root 具备一切权限

判别权限的方法：　　匹配及停止原则

1. 查看用户对于文档的身份，属于那个归属关系　　所有者>所属组>其他人

2. 查看相应归属关系位置的权限

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

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

Permission denied : 权限不足

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

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 zhangsan 能够在此目录下创建子目录

切换用户 su - zhangsan

chmod o+w /nsddir/

2）使用户 zhangsan 不能够在此目录下创建子目录

chmod o-w /nsddir/

3）使用户 zhangsan 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/

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

• 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...

[root@server0 ~]# mkdir /nsd03

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

[root@server0 ~]# groupadd tedu

[root@server0 ~]# chown  zhangsan:tedu /nsd03

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

[root@server0 ~]# chown root /nsd03       #修改所有者
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# chown :zhangsan /nsd03   #修改所属组
[root@server0 ~]# ls -ld /nsd03
```

##########

利用 root 用户新建/nsd05 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 /]# useradd  gelin02
[root@server0 /]# groupadd  tarena
[root@server0 /]# chown gelin01:tarena  /nsd05
```

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=--- /nsd05
```

3）使用户 gelin02 能进入、查看此目录

[root@server0 /]# gpasswd -a  gelin02  tarena  #用户加入组

4)将 gelin01 加入 tarena 组, 将 nsd05 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 /nsd05

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

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

- 继承父目录的所属组身份

[root@server0 ~]# mkdir /nsd07

[root@server0 ~]# ls -ld /nsd07

```
[root@server0 ~]# chown :tarena /nsd07
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test01
[root@server0 ~]# ls -ld /nsd07/test01

[root@server0 ~]# chmod g+s /nsd07
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test02
[root@server0 ~]# ls -ld /nsd07/test02

[root@server0 ~]# ls -ld /nsd07/test01
```

##########

ACL 权限（ACL 策略）

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 u

- 所属组(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

查看权限

• 使用 ls -l 命令

- ls -ld 文件或目录...

以 - 开头：文本文件

以 d 开头：目录

以 l 开头：快捷方式

• 使用 chmod 命令

- chmod [-R] 归属关系+-=权限类别 文档...

- [-R] 递归设置权限

```
[root@server0 ~]# mkdir /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=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01

[root@server0 ~]# chmod u=rwx,g=rwx,o=--- /nsd01
[root@server0 ~]# ls -ld /nsd01

[root@server0 ~]# chmod ugo=rwx /nsd01
[root@server0 ~]# ls -ld /nsd01
```

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

管理员 root 具备一切权限

判别权限的方法：　　匹配及停止原则

1. 查看用户对于文档的身份，属于那个归属关系　　所有者>所属组>其他人

2. 查看相应归属关系位置的权限

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

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

Permission denied : 权限不足

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

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 zhangsan 能够在此目录下创建子目录

切换用户  su  -  zhangsan

chmod o+w  /nsddir/

2）使用户 zhangsan 不能够在此目录下创建子目录

chmod o-w  /nsddir/

3）使用户 zhangsan 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/

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

- 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...

```
[root@server0 ~]# mkdir /nsd03
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# groupadd tedu
[root@server0 ~]# chown  zhangsan:tedu /nsd03
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# chown root /nsd03      #修改所有者
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# chown :zhangsan /nsd03   #修改所属组
[root@server0 ~]# ls -ld /nsd03
```

##########

利用 root 用户新建/nsd05 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 /]# useradd  gelin02
```

[root@server0 /]# groupadd  tarena

[root@server0 /]# chown gelin01:tarena  /nsd05

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=--- /nsd05

3）使用户 gelin02 能进入、查看此目录

[root@server0 /]# gpasswd -a  gelin02  tarena  #用户加入组

4)将 gelin01 加入 tarena 组, 将 nsd05 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 /nsd05

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

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
- 继承父目录的所属组身份

```
[root@server0 ~]# mkdir /nsd07
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# chown :tarena /nsd07
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test01
[root@server0 ~]# ls -ld /nsd07/test01

[root@server0 ~]# chmod g+s /nsd07
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test02
[root@server0 ~]# ls -ld /nsd07/test02
```

[root@server0 ~]# ls -ld /nsd07/test01

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

ACL 权限（ACL 策略）

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

1. 查看用户对于文档的身份，属于那个归属关系　　所有者>所属组>其他人
2. 查看相应归属关系位置的权限

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

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

Permission denied : 权限不足

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

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 zhangsan 能够在此目录下创建子目录

切换用户　su　-　zhangsan

chmod o+w　/nsddir/

2）使用户 zhangsan 不能够在此目录下创建子目录

chmod o-w　/nsddir/

3）使用户 zhangsan 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/

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

• 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...

```
[root@server0 ~]# mkdir /nsd03
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# groupadd tedu
[root@server0 ~]# chown  zhangsan:tedu /nsd03
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# chown root /nsd03      #修改所有者
[root@server0 ~]# ls -ld /nsd03

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

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

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

利用 root 用户新建/nsd05 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 /]# useradd  gelin02

[root@server0 /]# groupadd  tarena

[root@server0 /]# chown gelin01:tarena  /nsd05

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=--- /nsd05

3）使用户 gelin02 能进入、查看此目录

[root@server0 /]# gpasswd -a  gelin02  tarena  #用户加入组

4)将 gelin01 加入 tarena 组，将 nsd05 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 /nsd05

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

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

- 继承父目录的所属组身份

[root@server0 ~]# mkdir /nsd07

[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# chown :tarena /nsd07

[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test01

[root@server0 ~]# ls -ld /nsd07/test01

```
[root@server0 ~]# chmod g+s /nsd07
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test02
[root@server0 ~]# ls -ld /nsd07/test02

[root@server0 ~]# ls -ld /nsd07/test01
```

##########

ACL 权限（ACL 策略）

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 访问策略关系　　所有者>所属组>其他人

2. 查看相应归属关系位置的权限

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

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

Permission denied : 权限不足

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

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 zhangsan 能够在此目录下创建子目录

切换用户 su - zhangsan

chmod o+w /nsddir/

2）使用户 zhangsan 不能够在此目录下创建子目录

chmod o-w /nsddir/

3）使用户 zhangsan 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/

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

• 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...

[root@server0 ~]# mkdir /nsd03

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

[root@server0 ~]# groupadd tedu

[root@server0 ~]# chown  zhangsan:tedu /nsd03

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

[root@server0 ~]# chown root /nsd03       #修改所有者
[root@server0 ~]# ls -ld /nsd03

[root@server0 ~]# chown :zhangsan /nsd03   #修改所属组
[root@server0 ~]# ls -ld /nsd03
```

##########

利用 root 用户新建/nsd05 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 /]# useradd  gelin02
[root@server0 /]# groupadd  tarena
[root@server0 /]# chown gelin01:tarena  /nsd05
```

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=--- /nsd05
```

3）使用户 gelin02 能进入、查看此目录

[root@server0 /]# gpasswd -a  gelin02  tarena  #用户加入组

4)将 gelin01 加入 tarena 组, 将 nsd05 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 /nsd05

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

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

- 继承父目录的所属组身份

[root@server0 ~]# mkdir /nsd07

[root@server0 ~]# ls -ld /nsd07

```
[root@server0 ~]# chown :tarena /nsd07
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test01
[root@server0 ~]# ls -ld /nsd07/test01

[root@server0 ~]# chmod g+s /nsd07
[root@server0 ~]# ls -ld /nsd07

[root@server0 ~]# mkdir /nsd07/test02
[root@server0 ~]# ls -ld /nsd07/test02

[root@server0 ~]# ls -ld /nsd07/test01
```

##########

ACL 权限（ACL 策略）

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 lin01  /nsd10

[root@server0 ~]# getfacl  /nsd10

[root@server0 ~]# setfacl  -b  /nsd10

[root@server0 ~]# getfacl  /nsd10

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

使用 LDAP 认证

网络用户：由网络中一台服务器提供用户名、密码

本地用户：/etc/passwd

作用：集中管理网络中的用户帐号

什么是 LDAP?

• 轻量级目录访问协议

- Lightweight Directory Access Protocol

- 提供的信息包括:用户名、密码、通信录、主机名映射记录、.....

LDAP 服务器：classroom.example.com

客户端：虚拟机 Server

1.安装 sssd 客户端软件，与服务端 LDAP 服务器沟通

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

2.安装图形工具配置 sssd 软件

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

3.运行 authconfig-gtk 图形工具，配置 sssd 软件

```
[root@server0 ~]# exit
```

登出

```
[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

4.验证

[root@server0 ~]# systemctl restart sssd

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

[root@server0 ~]# id ldapuser0

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

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

NFS 服务端：classroom.example.com

客户端：虚拟机 Server

1. 查看服务端有那些共享文件夹

```
# showmount -e classroom.example.com
```

2.访问共享文件夹，利用 mount 提供访问点

```
# mount classroom.example.com:/home/guests  /mnt
# ls /mnt/
```

```
# umount /mnt/
# mkdir /home/guests
# mount classroom.example.com:/home/guests/  /home/guests
# ls /home/guests
```

# su - ldapuser12

还原三台虚拟机器

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

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

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

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

附加权限

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

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 /public
[root@server0 ~]# chmod ugo=rwx /public
[root@server0 ~]# ls -ld /public

[root@server0 ~]# chmod o+t /public
[root@server0 ~]# ls -ld /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...

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

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

```
[root@server0 ~]# grep -i ROOT /etc/passwd   #忽略大小写
```

```
[root@server0 ~]# grep -v root /etc/passwd   #取反查找
```

- ^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 bash$ /etc/passwd
```

```
[root@server0 ~]# grep ^root /etc/shadow
```

匹配空行

```
[root@server0 ~]# grep  ^$  /etc/default/useradd
```

```
[root@server0 ~]# grep -v ^$  /etc/default/useradd
```

显示文件的内容，有效的配置（去除注释以#开头的行、去除空行）

```
# grep -v  ^#  /etc/default/useradd
# grep -v  ^#  /etc/default/useradd   |  grep -v ^$

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

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

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

[root@server0 ~]# find /boot/ -type f  #文本文件

[root@server0 ~]# find /boot/ -type d  #目录

[root@server0 ~]# find /boot/ -type l  #快捷方式

[root@server0 ~]# ls -l /boot/grub/menu.lst

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

/etc/passwd

/etc/pam.d/passwd

[root@server0 ~]# find /etc/ -name "*tab"

```
[root@server0 ~]# find /etc/ -name "*tab*"
```

请查找/etc 以 .conf 结尾（包含子目录）

```
[root@server0 ~]# find /etc -name "*.conf"
```

请查找/etc 以 .conf 结尾（不包含子目录）

```
[root@server0 ~]# ls /etc/*.conf
```

```
[root@server0 ~]# touch /root/nsd01.txt
[root@server0 ~]# touch /root/nsd02.txt
[root@server0 ~]# mkdir /root/nsd1803
```

```
[root@server0 ~]# find /root/ -name "nsd*"
```

```
[root@server0 ~]# find /root/ -name "nsd*"  -type f
```

```
[root@server0 ~]# find /root/ -name "nsd*"  -type d
```

```
##########################################################
```

```
  -size +10M
```

```
[root@server0 ~]# find  /boot/  -size  +10M
[root@server0 ~]# find  /boot/  -size  +300k
```

```
  -user 用户
```

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

```
  -group 组名
```

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

```
##########################################################
```

查找文件

- 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

```
# find /boot/ -name "vm*"
# find /boot/ -name "vm*"  -exec cp  {}  /opt  \;
# ls /opt/
```

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

```
# find /root/ -name "nsd*"  -type  f
# find /root/ -name "nsd*"  -type  f  -exec cp  {} /opt  \;
# ls /opt/
```

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

- 根据名称查找,忽略大小写

- -iname

```
[root@server0 ~]# find /etc/ -name "PASSWD"
[root@server0 ~]# find /etc/ -iname "PASSWD"
/etc/passwd
/etc/pam.d/passwd
[root@server0 ~]#
```

- 限制目录查找的深度(最大层数)

- -maxdepth

```
[root@server0 ~]# find /etc/ -maxdepth 1  -name "*.conf"
[root@server0 ~]# find /etc/ -maxdepth 2  -name "*.conf"
[root@server0 ~]# find /etc/ -maxdepth 3  -name "*.conf"

[root@server0 ~]# find /etc/ -maxdepth 1  -name "passwd"
[root@server0 ~]# find /etc/ -maxdepth 2  -name "passwd"
```

查找文件

- 根据文件修改时间，都是过去时间

-mtime +10  #过去 10 天之前

-mtime -10  #最近 10 天之内

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

[root@server0 ~]# find  /var/log/  -mtime  +100

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

cron 计划任务

cron 任务概述

- 用途:按照设置的时间间隔为用户反复执行某一项固

定的系统任务

- 软件包:cronie、crontabs
- 系统服务:crond
- 日志文件:/var/log/crond

如何编写 crontab 任务记录

分　时　日　月　周　　任务命令行(绝对路径)

```
*     *     *     *     *

0     8     *     *     5

30    23    *     *     *
```

*: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 ...

```
30    23    *     *     1, 3, 5

30    23    *     *     2-5

0     */2   *     *     *
```

- 使用 crontab 命令

- 编辑:crontab -e [-u 用户名]
- 查看:crontab -l [-u 用户名]
- 清除:crontab -r [-u 用户名]

每分钟记录当前系统的时间，写入文件/opt/time.txt

```
[root@server0 ~]# date >> /opt/time.txt
[root@server0 ~]# cat /opt/time.txt

[root@server0 ~]# crontab -e -u root
[root@server0 ~]# crontab -l -u root
*/1   *   *   *   *   date >> /opt/time.txt
```

分　　时　　日　　月　　周

还原三台虚拟机器

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

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

附加权限

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

- 传递所有者身份

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

磁道：track

扇区：sector  默认 512 字节

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

一块硬盘的“艺术”之旅

- 识别硬盘 => 分区规划 => 格式化 => 挂载使用

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

分区规划

MBR/msdos：主引导记录模式

- 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

GPT：128 个主分区， 最大支持容量为 18EB 的磁盘

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

- 修改硬盘的分区表
  - fdisk  硬盘设备

常用交互指令:

m 列出指令帮助

p 查看现有的分区表

n 新建分区

d 删除分区

q 放弃更改并退出

w 保存更改并退出

一、查看系统所有的磁盘设备

```
[root@server0 ~]# lsblk
```

二、划分新的分区

```
[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 +2G

d 删除分区

w 保存并退出

[root@server0 ~]# lsblk

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

/dev/sda5 ： 第一块SCSI硬盘的第5个分区

第一块SCSI硬盘的第1个逻辑分区

三、格式化分区

make file system

[root@server0 ~]# mkfs.ext4 /dev/vdb1 #格式化ext4文件系统

[root@server0 ~]# mkfs.xfs /dev/vdb2 #格式化xfs文件系统

[root@server0 ~]# blkid /dev/vdb1 #查看文件系统及UUID

[root@server0 ~]# blkid /dev/vdb2 #查看文件系统及UUID

四、挂载使用

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

mount: 挂载点 /mypart1 不存在

[root@server0 ~]# mkdir  /mypart1

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

[root@server0 ~]# mkdir  /mypart2

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

[root@server0 ~]# df -h   #查看正在挂载的分区使用情况

disk  file

五、继续划分分区,最后一个主分区

[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

w 保存分区

partprobe : 刷新  划分的新分区

[root@server0 ~]# lsblk

[root@server0 ~]# ls /dev/vdb3

六、继续划分分区，扩展分区与逻辑分区

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

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

p 查看分区表

n 创建新的分区----->回车----->在 last 结束时 +1G

p 查看分区表

n 创建新的分区----->回车----->在 last 结束时 +1G

p 查看分区表

w 保存分区

```
[root@server0 ~]# partprobe
[root@server0 ~]# lsblk
```

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

总结：

1.查看磁盘　lsblk

2.划分分区　fdisk

3.刷新　　partprobe

4.格式化　　mkfs.ext4　　mkfs.xfs

5.查看文件系统　blkid

6.挂载使用　　mount

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

将虚拟机关闭，图形添加新的硬盘60G

```
[root@server0 ~]# poweroff
[root@room9pc01 ~]# ssh -X root@172.25.0.11
[root@server0 ~]# lsblk
 ......
```

vdc    253:32   0  60G  0 disk

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

实现开机自动挂载

- 修改/etc/fstab 配置文件

  file  system  table

- 配置文件 /etc/fstab 的记录格式

  设备路径　　挂载点　　类型　　参数　　备份标记　　检测顺序

vim 命令模式：按 o 可以新起一行，进入插入模式

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

/dev/vdb1   /mypart1   ext4   defaults 0 0
/dev/vdb2   /mypart2   xfs    defaults 0 0
```

[root@server0 ~]# df -h

[root@server0 ~]# mount -a

[root@server0 ~]# df  -h

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

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

综合分区

将 /dev/vdc 进行分区

3 个 10G 主分区

2 个 10G 逻辑分区

[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

作用： 1.整合分散的空间

2.逻辑卷空间可以扩大

新建逻辑卷： 将众多的物理卷(pv)组成卷组(vg)，再从卷组中划分逻辑卷(lv)

面粉---------->大面团--------->小面团------>蒸-------->吃

砖---------->大房子---------->打隔段------>装修------->入住

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

一、逻辑卷的创建

1.建立 vg 卷组

格式：vgcreate  卷组名  设备路径

[root@server0 ~]# vgcreate  nsd  /dev/vdc[1-2]  #创建卷组

[root@server0 ~]# vgs  #查看卷组基本信息

[root@server0 ~]# pvs  #查看物理卷基本信息

2.创建 lv 逻辑卷

格式：lvcreate -n  逻辑卷名  -L  大小  卷组名

[root@server0 ~]# lvcreate  -n  vo  -L 16G  nsd

```
[root@server0 ~]# lvs    #查看逻辑卷基本信息
[root@server0 ~]# vgs
```

3.格式化，挂载使用

```
[root@server0 ~]# mkfs.ext4 /dev/nsd/vo
[root@server0 ~]# blkid /dev/nsd/vo
[root@server0 ~]# vim /etc/fstab

/dev/nsd/vo  /mylvm    ext4   defaults  0  0

[root@server0 ~]# mount -a
[root@server0 ~]# df -h
```

##########

逻辑卷的扩大，支持线上工作

一、卷组有足够的剩余空间

1. 扩建逻辑卷的空间

```
[root@server0 ~]# lvextend -L 18G /dev/nsd/vo
[root@server0 ~]# lvs
```

2. 扩建逻辑卷的文件系统

resize2fs : ext4 文件系统扩展命令

xfs_growfs : xfs 文件系统扩展命令

```
[root@server0 ~]# df -h
[root@server0 ~]# resize2fs /dev/nsd/vo
[root@server0 ~]# df -h
```

二、卷组没有足够的剩余空间

1.扩展卷组

```
[root@server0 ~]# vgextend nsd /dev/vdc3
[root@server0 ~]# vgs
```

2. 扩建逻辑卷的空间

[root@server0 ~]# lvextend -L 25G /dev/nsd/vo

[root@server0 ~]# lvs

3. 扩建逻辑卷的文件系统

[root@server0 ~]# df -h

[root@server0 ~]# resize2fs /dev/nsd/vo

[root@server0 ~]# df -h

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

逻辑卷可以缩小，但是强烈不建议

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

PE：卷组划分空间的单位

[root@server0 ~]# vgdisplay nsd  #显示卷组详细信息,看 PE 的大小

[root@server0 ~]# vgchange -s 1M nsd  #修改卷组 PE 的大小

[root@server0 ~]# vgdisplay nsd  #显示卷组详细信息,看 PE 的大小

```
[root@server0 ~]# lvcreate -L 250M -n lvtest02 nsd


[root@server0 ~]# lvcreate -l 50 -n lvtest03 nsd
[root@server0 ~]# lvs
```

-l:指定 PE 的个数

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

逻辑卷的删除

首先删除 LV 逻辑卷，在删除 VG 卷组，最后删除 PV 物理卷

```
[root@server0 ~]# lvremove /dev/nsd/vo
[root@server0 ~]# umount /mylvm/
[root@server0 ~]# lvremove /dev/nsd/vo
Do you really want to remove active logical volume vo?
 [y/n]: y
  Logical volume "vo" successfully removed
```

```
[root@server0 ~]# lvs

[root@server0 ~]# vgs
```

还原三台虚拟机器

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

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

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

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

Shell 脚本：可以执行文件，可是实现某种功能

shell 语言

- 提前设计可执行语句,用来完成特定任务的文件
  - 解释型程序
  - 顺序、批量执行

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

echo hello world
```

[root@server0 ~]# chmod +x /root/hello.sh  #所有人加执行权限

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

规范 Shell 脚本的一般组成

- #! 环境声明
- # 注释文本
- 可执行代码

一般脚本为提高执行的效率，采用非交互式

请书写可以创建用户的脚本，并且为用户设置密码为 123

```
[root@server0 ~]# vim  /root/user.sh
 #!/bin/bash
 useradd  nsd02
 echo  123  |  passwd  --stdin  nsd02
```

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

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

编写一个能输出系统信息的 /root/sysinfo.sh 脚本

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/sysinfo.sh
 #!/bin/bash
 cat /etc/redhat-release
 uname -r
 hostname
 ifconfig | head -2

[root@server0 ~]#
```

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

简单脚本技巧

管道传递

- 使用 | 管道操作
  - 将前一条命令的标准输出交给后一条命令处理

重定向输出

> :将前面命令的正确输出，写入到文本文件中,只收集正确信息

2> :将前面命令的错误输出，写入到文本文件中,只收集错误信息

&> :将前面命令的正确与错误输出，写入到文本文件中,正确与错误都收集

```
[root@server0 ~]# echo 123 > /opt/1.txt
[root@server0 ~]# cat /opt/1.txt
```

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

[root@server0 ~]# cat /opt/1.txt /etc/
[root@server0 ~]# cat /opt/1.txt /etc/ > /opt/a.txt
[root@server0 ~]# cat /opt/a.txt

[root@server0 ~]# cat /opt/1.txt /etc/ 2> /opt/a.txt
[root@server0 ~]# cat /opt/a.txt

[root@server0 ~]# cat /opt/1.txt /etc/ &> /opt/a.txt
[root@server0 ~]# cat /opt/a.txt
```

补充：

’ ’ ：取消所有特殊字符意义

$[ ]：运算 + - * / %(取余数运算，求模)

定律 ： 余数一定小于除数

```
[root@server0 ~]# echo  $[1+1]
[root@server0 ~]# echo  $[1-2]
[root@server0 ~]# echo  $[3*2]
[root@server0 ~]# echo  $[10/2]
[root@server0 ~]# echo  $[10/3]
```

$( ) 与 ` ` 反撇号：将命令的输出结果，作为参数

```
[root@server0 opt]# cd /opt

[root@server0 opt]# mkdir  $(date +%F)
[root@server0 opt]# ls

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

[root@server0 opt]# mkdir  `hostname`-`date +%F`
[root@server0 opt]# ls
```

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

黑洞设备:/dev/null

```
[root@server0 /]# vim /root/user.sh
#!/bin/bash
useradd  nsd05   &> /dev/null
echo  nsd05 用户创建成功
echo  123 |  passwd  --stdin  nsd05  &> /dev/null
echo  nsd05 用户密码设置成功

[root@server0 /]#
```

变量：提高脚本的灵活度，适用多变的环境

变量：会变化的量，以不变的名称，存储可以变化的值（容器）

变量的定义：　　变量名=存储的值

```
[root@server0 /]# vim /root/user.sh
 #!/bin/bash
 abc=kenji
 useradd $abc &> /dev/null
 echo $abc 用户创建成功
 echo 123 | passwd --stdin $abc &> /dev/null
 echo $abc 用户密码设置成功
```

##########

交互式,降低脚本使用的难度，用户在键盘上的输入，存放到变量中

read : 记录用户在键盘上的输入，并且存放到变量中

```
[root@server0 /]# cat /root/user.sh
 #!/bin/bash
 read  -p  '请输入您要创建的用户名：'     abc
 useradd  $abc  &> /dev/null
 echo  $abc 用户创建成功
```

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

echo  $abc 用户密码设置成功

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

使用变量

定义/赋值变量

- 变量名只由字母/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
- 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

查看/引用变量

• 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 ${a}

rhel

[root@server0 /]# echo $a7

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

rhel7

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

环境变量 : 由系统定义赋值完成，用户直接调用

USER ： 永远储存当前登陆的用户名

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

位置变量：由系统定义赋值完成，用户直接调用

方便向脚本中传递命令行参数

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

#!/bin/bash

cat -n $1 | head -$2

```
[root@server0 /]# /root/test.sh  /etc/passwd  3


[root@server0 /]# vim /root/123.sh
#!/bin/bash
echo $1
echo $2
echo $3
[root@server0 /]# /root/123.sh  haha   abcd   rhel7


############################################################
```

预定义变量：由系统定义赋值完成，用户直接调用

```
   $#  已加载的位置变量的个数
   $*  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/test.sh
 #!/bin/bash
 cat -n $1  | head -$2
```

```
 echo $#

 echo $*

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


[root@server0 /]# vim  /root/123.sh

 #!/bin/bash

 echo $1

 echo $2

 echo $3

 echo $#

[root@server0 /]# /root/123.sh   haha  xixi  hehe  lele
```

##########

条件测试

- 检查文件状态

  -e : 判断文档是否存在，存在为真

  -d : 存在且为目录，才为真

  -f : 存在且为文件，才为真

-r ： 存在且具备读权限，才为真

-w ： 存在且具备写权限，才为真

-x ： 存在且具备执行权限，才为真

```
[root@server0 /]# [  -e  /etc  ]
[root@server0 /]# echo $?
 0
[root@server0 /]# [ -e  /eta  ]
[root@server0 /]# echo $?
 1
[root@server0 /]# [ -f /etc/passwd ]
[root@server0 /]# echo $?
 0
[root@server0 /]# [ -f /root ]
[root@server0 /]# echo $?
 1
[root@server0 /]# [ -d /root ]
[root@server0 /]# echo $?
 0
```

- 比较整数大小(带 e 字母都有等于二字，g 代表大于，l 代表小于)

  -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  1  ]        #大于
[root@server0 /]# echo $?
 1
[root@server0 /]# [  1 -ge  1  ]        #大于等于
[root@server0 /]# echo $?
 0
[root@server0 /]# [ 0 -eq  10 ]         #等于
[root@server0 /]# echo $?
 1
[root@server0 /]# [ 0 -lt  10 ]         #小于
```

```
[root@server0 /]# echo $?
 0
[root@server0 /]# [ 0 -ne  10 ]          #不等于
[root@server0 /]# echo $?
 0
```

- 字符串比对

  == ： 字符串相等为真

  != ： 字符串不相等为真

```
[root@server0 /]# [  root  ==   redhat  ]
[root@server0 /]# echo $?
   1
[root@server0 /]# [  root  !=   redhat  ]
[root@server0 /]# echo $?
   0
[root@server0 /]# [  $USER  ==   root  ]
[root@server0 /]# echo $?
```

0

##########

if 选择结构

if 双分支处理

```
if [条件判断];then
    命令序列 xx
else
    命令序列 yy
fi
```

案例 1：

请书写一个脚本，判断当前登陆的用户是否为 root

如果 是 则输出 当前是管理员

如果 不是 则输出 当前是普通用户

```
[root@server0 /]# vim /opt/if02.sh
 #!/bin/bash
 if  [ $USER == root ];then
     echo 当前是管理员
 else
     echo 当前是普通用户
 fi

[root@server0 /]# /opt/if02.sh
```

案例 2：

计算机随机产生一个 0 到 9 之间的数字

用户输入一个 0 到 9 之间的数字

如果　用户输入的数字与 计算机随机产生 相等，则输出　您猜对了

如果　用户输入的数字与 计算机随机产生 不相等，则输出　猜错了

$RANDOM :系统储存随机数字

定律 ： 余数一定小于除数

```
[root@server0 /]# vim  /root/num.sh
 #!/bin/bash
 read -p   '请输入0到9之间的一个数字:'       num1
 num2=$[$RANDOM%10]

 if  [   $num1   -eq  $num2  ];then
    echo  您猜对了
 else
    echo  猜错了
    echo  正确的数字为$num2
 fi
```

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

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 aa
else
   命令序列 zz
fi
```

案例 3：

书写一个成绩判断的脚本

用户输入成绩，0 到 100 之间

如果，成绩 大于等于 90 以上　则输出　优秀

如果，成绩 大于等于 80 以上　则输出　良好

如果，成绩 大于等于 70 以上　则输出　合格

如果，成绩 大于等于 60 以上　则输出　仍需努力

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/nsd.sh
#!/bin/bash
read   -p   '请输入您的成绩：'        num

if [  $num   -ge   90  ];then
  echo  优秀
elif [  $num  -ge  80  ];then
  echo  良好
elif [  $num  -ge  70  ];then
  echo  合格
elif [  $num  -ge  60  ];then
  echo  仍需努力
else
  echo  再牛的肖邦，也弹不出哥的悲伤
fi
```

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

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                    #判断是否输入了位置参数
   echo  '/root/foo.sh   redhat|fedora' >&2
                                    #把正确变成错误输出
       exit  2   #脚本退出返回值
 elif  [  $1  ==  redhat  ];then
   echo fedora
```

```
elif  [  $1  ==  fedora  ];then
   echo redhat

else
   echo   '/root/foo.sh   redhat|fedora'  >&2
   exit  3  #脚本退出返回值
fi
```

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

循环： 重复性执行一个操作

for 循环处理

- 遍历/列表式循环
- 根据变量的不同取值,重复执行 xx 处理

```
   for 变量名 in 值列表
   do
        循环的操作
   done
```

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

```
  for  a   in  zhangsan   lisi  dc  tc  dz  tz
  do
        useradd  $a
  done
```

{20..80} :造数工具,制造 20 到 80 之间所有的数字

{ 起始..结束}

```
[root@server0 /]#  vim  /root/for.sh
#!/bin/bash
for  a  in  {1..20}
do
  useradd  stu$a
  echo  stu$a 创建成功
done
```

```
########################################
  循环可以与循环执行的操作无关
########################################
[root@server0 /]# vim  /root/num.sh
 #!/bin/bash
 for  a  in  {1..3}
 do
 read   -p   '请输入0到9之间的一个数字:'      num1
 num2=$[$RANDOM%10]

 if  [   $num1  -eq   $num2   ];then
    echo   您猜对了
 else
    echo   猜错了
    echo   正确的数字为$num2
 fi

 Done
```

还原三台虚拟机器

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

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

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

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

防火墙策略管理

基本 Web 服务,FTP 服务

1.服务端虚拟机 Server，安装可以提供 Web 服务与 FTP 服务软件

# yum -y  install  httpd  vsftpd

2.虚拟机 Server 操作，启动 httpd 与 vsftpd 服务，设置为开机自起服务

# systemctl  restart  httpd   #重起服务

# systemctl  enable  httpd    #设置开机自起服务

# systemctl  restart  vsftpd  #重起服务

```
# systemctl  enable  vsftpd   #设置开机自起服务
```

3.虚拟机 Server 操作,书写页面文件

```
# echo '<h1>NSD1803'  >  /var/www/html/index.html
```

4.虚拟机 Server 操作,访问测试

```
# firefox  172.25.0.11
# firefox  ftp://172.25.0.11
```

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

防火墙策略管理

作用：隔离，过滤入站请求，允许出站

RHEL7 的防火墙体系

- 系统服务:firewalld
- 管理工具:firewall-cmd、firewall-config（图形工具）

匹配规则的原则： 匹配及停止

预设安全区域

- 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 sshd、DHCP、ping 少数几个服务
  - 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，节省资源

防火墙决定，客户端请求进入某个区域的规则：

1.查看客户端请求中源 IP 地址，再看所以有区域中，哪一个区域有该源 IP 地址的规则，则进入该区域

2.进入默认区域，public

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

互联网常见的服务协议

http :超文本传输协议 80

FTP ：文件传输协议 21

https ： 安全的超文本传输协议　　443

DNS　　:域名解析协议　　53

telnet :远程管理协议　　23

smtp　 ： 邮件协议，用户发邮件协议　 25

pop3　 ： 邮件协议，用户收邮件协议　 110

tftp　 ： 简单文件传输协议　　69

默认区域服务的添加

虚拟机 Server

```
# firewall-cmd --get-default-zone            #查看默认区域
# firewall-cmd --zone=public  --list-all     #查看区域规则
# firewall-cmd --zone=public  --add-service=http  #添加服务
# firewall-cmd --zone=public  --add-service=ftp
# firewall-cmd --zone=public  --list-all
```

虚拟机 Desktop

```
# firefox 172.25.0.11          #可以访问
# firefox ftp://172.25.0.11  #可以访问
```

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

策略的永久配置

- 永久(permanent)

虚拟机 Server

# firewall-cmd --reload　　#重新加载防火墙配置，模拟重起机器

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

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

# firewall-cmd --permanent --zone=public --add-service=ftp

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

# firewall-cmd --reload

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

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

添加源 IP 的规则设置

虚拟机 Server

```
 # firewall-cmd --zone=block --list-all
 # firewall-cmd --zone=block    --add-source=172.25.0.10
 # firewall-cmd --zone=block --list-all
```

虚拟机 Desktop

```
 # ping -c 2 172.25.0.11         #失败
 # firefox   172.25.0.11         #失败
 # firefox   ftp://172.25.0.11   #失败
```

真机访问

```
 # ping -c 2 172.25.0.11         #成功
 # firefox   172.25.0.11         #成功
 # firefox   ftp://172.25.0.11   #成功
```

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

工作时防火墙，设置的方式

严格：默认区域为 drop，把允许的 IP 单独放入 trusted

宽松：默认区域为 trusted，把拒绝的 IP 单独放入 drop

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

端口：编号，标识作用，标识每个服务

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 desktop-----》172.25.0.11:5423-----》服务端 Server

服务端 Server-----172.25.0.11:5423 移交

-----172.25.0.11:80

虚拟机 Server

# firewall-cmd --reload

虚拟机 Desktop

# firefox 172.25.0.11  #可以访问

# firefox 172.25.0.11:5423  #不可以访问

虚拟机 Server

# firewall-cmd --permanent

--zone=public

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

#添加 - 转发 - 端口 = 将端口 5423 协议为 tcp 转发到 80

# firewall-cmd --reload

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

虚拟机 Desktop

# firefox 172.25.0.11:5423  #可以访问

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

配置聚合连接（网卡绑定、链路聚合）

eth1　　eth2

虚拟网卡 team（组队）

1.创建 虚拟网卡　　man teamd.conf　#查看帮助信息

# nmcli　connection　add　type　team

con-name　team0　ifname　team0　autoconnect　yes

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

# ifconfig　#查看是否有 team0 网卡

# nmcli　connection 添加　类型为　team（绑定类型）

配置文件名为　team0　ifconfig 显示网卡名为　team0　每次开机自起

配置网卡绑定工作模式　　热备份方式

#　如果敲错误

# nmcli  connection  delete team0

2.添加成员（添加奴隶）

# 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  类型为  team-slave 配置文件名 team0-1

网卡为  eth1  添加到 team0 中

#如果敲错误  nmcli connection delete  team0-1

3.配置 team0 的 IP 地址

# nmcli connection modify team0 ipv4.method manual

ipv4.addresses 192.168.1.1/24 connection.autoconnect yes

4.激活所有配置

```
# nmcli connection up team0

# nmcli connection up team0-1

# nmcli connection up team0-2
```

如果激活失败

```
  # nmcli connection delete team0

  # nmcli connection delete team0-1

  # nmcli connection delete team0-2
```

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

终极验证:

```
[root@server0 ~]# teamdctl team0 state  #查看 team0 详细信息


[root@server0 ~]# ifconfig eth1 down    #禁用网卡


[root@server0 ~]# teamdctl team0 state
```

还原三台虚拟机器

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

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

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

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

修改防火墙的默认区域为 trusted

虚拟机 Server

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

虚拟机 desktop

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

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

Samba 服务基础

配置 SMB 共享，跨平台的共享，Windows 与 Linux 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)、CIFS(TCP 445)

• 所需软件包:samba

• 系统服务:smb

Samba 共享帐号：默认情况下，访问 Samba 共享必须通过用户验证

专门用于访问 Samba 共享时验证的用户与密码

与系统用户为同一个用户，但是密码时 samba 独立密码

• 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 .. ..

服务端，虚拟机 Server：

1.安装可以提供 smb 共享功能的软件

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

2.建立 Samba 共享验证的用户

[root@server0 ~]# useradd harry

[root@server0 ~]# useradd kenji

[root@server0 ~]# useradd chihiro

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

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

[root@server0 ~]# pdbedit  -L        #查看所有 Samba 共享帐号
```

3.创建共享目录与文件

```
[root@server0 ~]# mkdir /common
[root@server0 ~]# ls /
[root@server0 ~]# ls /common/
[root@server0 ~]# echo 123 > /common/123.txt
[root@server0 ~]# ls /common/
```

4.修改配置/etc/samba/smb.conf

vim 末行模式　：set  nu　　#添加行号

命令模式　　G　到全文的最后

```
[global]
   89 行  workgroup = STAFF         #指定工作组名

[common]                            #指定共享名
```

```
path = /common          #指定共享文件夹的实际绝对路径
```

5.重起 smb 服务，刷新配置

```
[root@server0 ~]# systemctl restart smb    #重起服务
[root@server0 ~]# systemctl enable smb     #设置为开机自起
```

6.服务端,虚拟机 Server：

- 需要加 -P 选项才能实现永久设置

SELinux 策略：布尔值（所服务功能的开关）

1. 查看 samba 布尔值

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

2.修改 SELinux 策略的布尔值

```
[root@server0 ~]# setsebool samba_export_all_ro on
```

3. 查看 samba 布尔值

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

客户端虚拟机 Desktop：

1.安装客户端软件，访问 samba 共享

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

2. 查看服务端 samba 共享，目的看 共享名

[root@desktop0 ~]# smbclient -L //172.25.0.11

Enter root's password:  #直接敲回车

Sharename

common

3.以 harry 身份，访问服务端 samba 共享

[root@desktop0 ~]# smbclient -U harry //172.25.0.11/common

Enter harry's password:  #输入密码

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

smb: \>

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

客户端虚拟机 desktop

使用 mount 挂载访问

- 所需软件包:cifs-utils　　#支持 cifs 协议

```
# yum -y install cifs-utils
# mkdir /mnt/samba

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

# df  -h
```

开机自动挂载

_netdev : 网络设备

先启动网络服务，具备网络参数后，再进行挂载

```
[root@desktop0 ~]# yum -y install cifs-utils

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

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


 //172.25.0.11/common  /mnt/samba  cifs

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


[root@desktop0 ~]# mount  -a

[root@desktop0 ~]# df  -h        #查看是否挂载成功
```

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

读写的 Samba 共享

服务端虚拟机 Server：

1.部署共享

```
[root@server0 ~]# mkdir  /devops

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

[root@server0 ~]# ls /devops
```

```
[root@server0 ~]# vim /etc/samba/smb.conf  #修改配置文件
  # 追加写入
 [devops]                          #共享名
   path = /devops                  #路径为/devops
   write list = chihiro            #允许 chihiro 用户可写

[root@server0 ~]# systemctl restart smb
```

2.修改 SELinux 策略

```
[root@server0 ~]# getsebool -a | grep samba
[root@server0 ~]# setsebool samba_export_all_rw on
[root@server0 ~]# getsebool -a | grep samba
```

3.用户本身的本地权限

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

客户端虚拟机 desktop

1.实现开机自动挂载

```
[root@desktop0 ~]# mkdir /mnt/pub
[root@desktop0 ~]# vim  /etc/fstab

  //172.25.0.11/devops  /mnt/pub    cifs
  defaults,user=chihiro,pass=123,_netdev  0 0

[root@desktop0 ~]# mount -a
[root@desktop0 ~]# df -h
```

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

总结:客户端访问服务端资源

1.服务本身的访问控制

2.本的目录的权限

3.防火墙

4.SELinux

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

multiuser 机制,专门为普通用户设计，专为客户端设计

- 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

必要的时候,任何普通用户都可以通过命令切换成权限较大的用户

来临时获取写的权限

客户端虚拟机 Desktop：

```
[root@desktop0 ~]# vim  /etc/fstab
//172.25.0.11/devops  /mnt/pub   cifs
defaults,user=kenji,pass=123,_netdev,multiuser,sec=ntlmssp
    0   0
[root@desktop0 ~]# umount  /mnt/pub
[root@desktop0 ~]# mount -a

[root@desktop0 ~]# su - student
[student@desktop0 ~]$ cifscreds add -u chihiro 172.25.0.11
```

[student@desktop0 ~]$ ls /mnt/pub

[student@desktop0 ~]$ exit

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

配置 NFS 共享, Linux 与 Linux 的共享

- 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)

只读的 NFS 共享

服务端虚拟机 server

1.所需软件包 : nfs-utils

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

nfs-utils-1.3.0-0.el7.x86_64

[root@server0 ~]# rpm -qa | grep nfs  #显示所有已安装，进行过滤

2.NFS 共享主配置文件/etc/exports

[root@server0 ~]# mkdir /nsd

[root@server0 ~]# echo haha > /nsd/abc.txt

[root@server0 ~]# ls /nsd

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

 /nsd      *(ro)                  #共享目录路径        客户端(权限)

3.重起 nfs 服务，设置为开机自起

[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:/nsd   /mnt/nfs  nfs  defaults,_netdev  0  0

```
[root@desktop0 ~]# mkdir /mnt/nfs
[root@desktop0 ~]# mount -a
[root@desktop0 ~]# df -h
```

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

环境变量

PATH：与执行命令相关

在执行命令时，需要找到命令所对应的程序，Linux系统会到PATH变量值的所有路径去寻找，如果找到就执行，没有就不执行

```
[root@server0 ~]# vim /opt/hello.sh
 #!/bin/bash
 echo hello world

[root@server0 ~]# chmod +x /opt/hello.sh
[root@server0 ~]# hello.sh      #执行失败

[root@server0 ~]# echo $PATH
```

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

[root@server0 ~]# hello.sh      #执行成功

还原三台虚拟机器

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

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

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

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

修改防火墙的默认区域为 trusted

虚拟机 Server

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

虚拟机 desktop

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

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

 iSCSI 网络磁盘

一、fdisk 划分 分区

[root@server0 ~]# lsblk

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

三个主分区 ， 分别 2 个 G 大小

两个逻辑分区 ， 分别 1 个 G 大小

[root@server0 ~]# lsblk

[root@server0 ~]# ls /dev/vdb[1-6]

二、搭建 iscsi 共享存储

- 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
- 端口：3260

思路：example：快递打包送货

服务端：　TCL 王牌大彩电　2 寸--------->装箱 ------->　运输

客户端：　　------》箱子 ------->拆开

共享存储服务端：/dev/vdb1(nsd)-->装箱(Target 磁盘组)--》传输

共享存储客户端：　箱子(Target 磁盘组)----->拆开

使用 targetcli 建立配置

- ISCSI Qualified Name 名称规范
  - iqn.yyyy-mm.倒序域名:自定义标识

虚拟机 server0_搭建 iscsi 服务端：

1.安装可以提供共享存储功能的软件 targetcli

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

2.进行配置共享存储

1）建立后端存储 backstore,指定后端存储是那个分区，起名

```
[root@server0 ~]# targetcli
/> ls                          #查看所有配置
/> backstores/block  create  nsd  /dev/vdb1
```

后端存储/类型为块设备　　创建　　名字　　实际设备路径

2）创建箱子并起名（建立 iqn 磁盘组）

```
/> iscsi/  create  iqn.2018-16.com.example:server0
```

3）将后端存储，放入箱子中（磁盘组 绑定 后端存储 luns）

```
/> iscsi/iqn.2018-16.com.example:server0/tpg1/luns
    create   /backstores/block/nsd
```

4）设置 ACL 访问控制(客户端访问时声称的名字)

/>iscsi/iqn.2018-16.com.example:server0/tpg1/acls

create  iqn.2018-16.com.example:test

5）设置提供服务的 IP 地址

/>iscsi/iqn.2018-16.com.example:server0/tpg1/portals

create  172.25.0.11

/> exit  #保存并退出

3.重起服务

[root@server0 ~]# systemctl  restart  target

[root@server0 ~]# systemctl  enable  target

虚拟机 desktop0_客户端：

1.安装客户端软件

# yum repolist  #产生缓存

# yum -y install iscsi-initiator-utils.i686

2.修改配置文件，指定客户端声称的名字

# vim /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.2018-16.com.example:test

vim 命令模式下：按 大写的 C 删除光标之后

3. 刷新  客户端声称的名字（客户端 iqn）刷新 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

4.寻找服务端，发现 iSCSI 磁盘  man iscsiadm

'Ctrl'+'shift'+'+' :变大

'Ctrl'+'-'

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

172.25.0.11 --discover

172.25.0.11:3260,1 iqn.2018-16.com.example:server0

5.重起客户端 iscsi 服务

[root@desktop0 ~]# lsblk

[root@desktop0 ~]# systemctl restart iscsi

[root@desktop0 ~]# lsblk

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

数据库服务基础

数据库:存放数据的仓库

数据库中有很多的库，每一个库中有很多的表格，每一个表格有很多的记录

表格：表记录　　表字段（表头）

一、安装部署 MariaDB 数据库

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

二、启动 mariadb 服务

```
[root@server0 ~]# systemctl  restart  mariadb
[root@server0 ~]# systemctl  enable  mariadb
```

三、使用 MariaDB 数据库

```
[root@server0 ~]# mysql          #进入数据库
  > show databases;              #查看所有库
  > create  database  nsd1803;   #创建库
  > show databases;

  > drop  database  nsd1803;     #删除库
  > show  databases;
  > exit                         #退出

[root@server0 ~]#
```

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

设置数据库管理员的密码

数据库管理员 和 系统管理员，不是同一个用户

数据库管理员：root 对所有库有完全控制权限， mysql 库 user 表

系统管理员：root 对 Linux 系统有完全控制权限，/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'

```
[root@server0 ~]# mysqladmin -u root  password '123'

[root@server0 ~]# mysql  -u root   -p
Enter password:

[root@server0 ~]# mysql  -uroot  -p123  #非交互式进入

[root@proxy ~]# mysqldump -uroot -predhat mydb table > table.sql
//备份数据库中的某个数据表
```

[root@proxy ~]# mysqldump -uroot -predhat mydb > mydb.sql

//备份某个数据库

[root@proxy ~]# mysqldump -uroot -predhat --all-databases > all.sql

//备份所有数据库

[root@proxy ~]# mysql -uroot -predhat mydb  < table.sql

//还原数据表

[root@proxy ~]# mysql -uroot -predhat mydb  < mydb.sql

//还原数据库

[root@proxy ~]# mysql -uroot -predhat < all.sql

//还原所有数据库

数据库主配置文件 /etc/my.cnf

了解：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

导入数据

http://classroom.example.com/pub/materials/users.sql

```
# wget 。。。。。。        #下载数据库文件
# mysql -uroot -p123

MariaDB [(none)]> create database nsd1803;
MariaDB [(none)]> exit

# mysql -uroot -p123 nsd1803 < users.sql
# mysql -uroot -p123

MariaDB [nsd1803]> use nsd1803;  #进入 nsd1803 库
MariaDB [nsd1803]> show tables;  #查看所有表格
```

对于表格操作：

增(insert)　　删 (delete)　　改 (update)　　查 (select)

查 (select) :  select 表字段 from 表名;

select * from 表名;

MariaDB [mysql]> use nsd1803;  #进入 nsd1803 库

MariaDB [nsd1803]> show tables;  #查看所有表格

> select * from base;  #查询所有字段内容

> select name,password from base;

> select * from location;

查询表结构：desc 表名;

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

数据库的授权：

除了 root 用户,此 nsd1803 数据库只能被用户 lisi 查询,

此用户的密码为 123

- 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';

```
# mysql -uroot  -p123
```

```
> grant select  on  nsd1803.*  to  lisi@localhost
  identified by  '123';
```

远程连接数据库

```
[root@client ~]# mysql -h192.168.4.5 -uhaha -p123456
```

当 lisi 用户从本地登陆输入密码 123，将会获得 nsd1803 库中所有表的查询权限

验证： # mysql -ulisi -p123 #lisi 可以登陆即可

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

案例 5:使用数据库查询

1. 在系统 server0 上使用数据库 nsd1803,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?

```
MariaDB [mysql]> use nsd1803;
MariaDB [nsd1803]> show tables;
```

有条件的查询： where

```
> select * from  base  where password='solicitous';
> select name from base where password='solicitous';
> select * from  base  where name='tom';
> select * from  base  where password='456';
```

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;
> desc user; #查看表结构


> select user,host,password from user  where password='';
```

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

> flush  privileges;    #刷新策略

> select  user,host,password  from  user;

还原三台虚拟机器

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

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

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

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

虚拟机 Server 与虚拟机 desktop,修改防火墙的默认区域为 trusted

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

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 Transfer Protocol,超文本传输协议(http)

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

http 协议的端口：80

默认网页文件目录：/var/www/html

默认网页文件的名字: index.html

一、搭建基本的 Web 服务

虚拟机 Server0

1.安装软件 httpd

2.书写一个页面文件

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

3.重起 httpd 服务，设置开机自起动

客户端访问测试：

虚拟机 desktop0

# firefox 172.25.0.11

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

- 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:classroom.example.com

server0.example.com

www0.example.com

webapp0.example.com

虚拟机 Server：

1.修改配置文件/etc/httpd/conf/httpd.conf

ServerName server0.example.com:80　　#把开头的#去掉

2.重起 httpd 服务

虚拟机 Desktop：

# firefox server0.example.com

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

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

虚拟机 Server0

1.修改配置文件 /etc/httpd/conf/httpd.conf

DocumentRoot "/var/www/myweb"

```
# mkdir /var/www/myweb
# echo '<h1>wo shi myweb' > /var/www/myweb/index.html
# systemctl restart httpd
```

虚拟机 Desktop0

```
# firefox server0.example.com
```

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

网页文件的根目录 DocumentRoot /var/www/myweb

客户端 firefox 172.25.0.11----》服务端：/var/www/myweb

客户端 firefox 172.25.0.11/test----》
/var/www/myweb/test/index.html

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

虚拟 Web 主机

作用：让一台 Web 服务器，提供多个页面

搭建方式：

1.基于域名的虚拟 Web

2.基于端口的虚拟 Web

3.基于 IP 地址的虚拟 Web

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

基于域名的虚拟 Web

容器类型的配置

<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>



<VirtualHost  *:80>
    ServerName   www.qq.com
    DocumentRoot /var/www/qq
</VirtualHost>

<VirtualHost  *:80>
    ServerName   www.baidu.com
    DocumentRoot /var/www/baidu
</VirtualHost>
```

- 配置文件路径
  - /etc/httpd/conf/httpd.conf
  - /etc/httpd/conf.d/*.conf

虚拟机 Server0：

1.修改调用配置文件

```
[root@server0 ~]# vim  /etc/httpd/conf.d/nsd01.conf
<VirtualHost  *:80>
  ServerName   www0.example.com
  DocumentRoot  /var/www/nsd01
</VirtualHost>

<VirtualHost  *:80>
  ServerName   webapp0.example.com
  DocumentRoot  /var/www/nsd02
</VirtualHost>

# mkdir  /var/www/nsd01  /var/www/nsd02
# echo '<h1>wo shi nsd01' > /var/www/nsd01/index.html
# echo '<h1>wo shi nsd02' > /var/www/nsd02/index.html
```

2.重起 httpd 服务

注意：一旦使用了虚拟 Web 主机功能，所有的站点都必须使用虚拟 Web 来实现

书写配置文件，完成虚拟 Web 主机的配置

1. 追加写入/etc/httpd/conf.d/nsd01.conf

```
<VirtualHost  *:80>
   ServerName   server0.example.com
   DocumentRoot  /var/www/myweb
</VirtualHost>
```

2.重起 httpd 服务

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

网页内容访问

- 每个文件夹自动继承其父目录的 ACL 访问权限
- 除非针对子目录有明确设置

```
<Directory 目录的绝对路径>
    .. ..
```

```
    Require all denied|granted
    Require ip IP或网段地址 .. ..
</Directory>
```

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

虚拟机Server0：

1.创建目录，写入网页文件

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

2.书写新的调用配置文件/etc/httpd/conf.d/nsd02.conf

```
<Directory  "/var/www/myweb/private">    #针对路径
    Require ip 172.25.0.11                #仅允许172.25.0.11
```

</Directory>

3.重起 httpd 服务

虚拟机 Desktop0： firefox server0.example.com/private

虚拟机 Server0： firefox server0.example.com/private

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

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:

1) 新建目录 /webroot,作为此站点新的网页文件根目录

# mkdir /webroot

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

2) 修改配置文件/etc/httpd/conf.d/nsd01.conf

<VirtualHost *:80>

```
  ServerName     server0.example.com
  DocumentRoot   /webroot                #修改网页文件根目录
</VirtualHost>
```

3) 追加写入 /etc/httpd/conf.d/nsd02.conf

```
<Directory  "/webroot">
    Require all granted       #允许所有
</Directory>
```

4）SELinux 安全上下文（标识，标签）

- chcon [-R] --reference=模板目录     新目录

```
# ls -Zd /webroot/       #查看目录标签值

# chcon -R  --reference=/var/www    /webroot/
# ls -Zd /webroot/
```

5) 重起 httpd 服务

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

部署动态网站

静态的网站

- 服务端的原始网页 = 浏览器访问到的网页
- 由 Web 服务软件处理所有请求
- 文本(txt/html)、图片(jpg/png)等静态资源

动态的网站

- 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.....

虚拟机 Server0：

1.部署 Python 动态页面文件

```
# vim /etc/httpd/conf.d/nsd01.conf
```

查看 webapp0.example.com 的 DocumentRoot 目录在那里

# cd /var/www/nsd02

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

2. 方便用户访问，页面的跳转

当客户端访问 webapp0.example.com-》
/var/www/nsd02/webinfo.wsgi

修改配置文件 /etc/httpd/conf.d/nsd01.conf

<VirtualHost *:80>
ServerName webapp0.example.com
DocumentRoot /var/www/nsd02
Alias / /var/www/nsd02/webinfo.wsgi

#当检测到客户端访问网页文件根目录时 跳转页面到
webinfo.wsgi

</VirtualHost>

重起 httpd 服务，客户端验证：firefox  webapp0.example.com

3. 安装一个可以解释 Python 页面程序，配置解释该页面

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

[root@server0 /]# rpm -ql mod_wsgi #列出软件安装清单

Unix 时间戳：从 1970-1-1 0:0:0 算起，到达当前时间经过的秒数

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

<VirtualHost *:80>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd02

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

</VirtualHost>

[root@server0 /]# systemctl  restart  httpd

4.修改端口为8909,修改配置文件 /etc/httpd/conf.d/nsd01.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

6.重起 httpd 服务

客户端访问测试 # firefox  webapp0.example.com:8909

还原三台虚拟机器

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

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

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

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

两台虚拟机防火墙默认区域设置为 trusted

搭建虚拟 Web

虚拟机 Server0：

1.安装 httpd 软件

2.书写网页文件/var/www/html/index.html

3.建立调用配置文件/etc/httpd/conf.d/nsd01.conf

```
<VirtualHost  *:80>
        ServerName     server0.example.com
        DocumentRoot   /var/www/html/
</VirtualHost>
```

4.重起 httpd 服务

虚拟机 Desktop0:

访问测试 firefox  server0.example.com

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

 安全 Web 服务器

   https 协议： 443 端口

虚拟 Server0：

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，证书颁发机构的信息）

# 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.安装可以支持加密 Web 的软件

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

[root@server0 /]# rpm -q mod_ssl

5.修改配置文件

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

末行模式下  :set  nu  #添加行号

59 DocumentRoot  "/var/www/html"

60 ServerName  server0.example.com:443

#指定网站证书

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

#指定密钥

107 SSLCertificateKeyFile /etc/pki/tls/private/server

0.key

#指定根证书

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

6.重起服务

[root@server0 /]# systemctl  restart  httpd

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

虚拟机 Desktop0：

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

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

parted(分区工具)

 要支持大容量（18EB）,需改用  gpt 分区模式可以有 128 个主分区

[root@server0 /]# lsblk

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

(parted) mktable gpt  #指定分区的模式

(parted) print        #输出所有分区信息

(parted) mkpart       #划分新的分区

分区名称？ []？ haha #分区的名字

文件系统类型？ [ext2]？ext4 #分区的文件系统类型，不起实际作用

起始点？ 0 #起始点 ， 第一个分区起始点为 0

结束点？ 4G #结束点

警告: The resulting partition is not properly aligned

for best performance.

忽略/Ignore/放弃/Cancel？ Ignore #忽略

(parted) print #输出所有分区信息

(parted) unit GB #设置显示单位

(parted) print #输出所有分区信息

(parted) quit #退出

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

(parted) print

(parted) unit GB #设置单位

(parted) print

(parted) mkpart　　#划分新的分区

分区名称？　[]? haha　　#分区的名字

文件系统类型？　[ext2]?ext4　#分区的文件系统类型，不起实际作用

起始点？ 4G　　#起始点 ， 上一个分区的结束

结束点？ 8G　　#结束点

(parted) print

(parted) quit

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

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

交换空间: 缓解真实物理内存的压力

由硬盘的空间来组成

- 交换分区:以空闲分区充当的交换空间

1.格式化交换分区

```
[root@server0 /]# mkswap /dev/vdb1

[root@server0 /]# mkswap /dev/vdb2
```

2.启用交换分区/dev/vdb1

```
[root@server0 /]# swapon  /dev/vdb1    #启用

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

[root@server0 /]# swapon  /dev/vdb2

[root@server0 /]# swapon  -s
```

3.停用交换分区

```
[root@server0 /]# swapoff  /dev/vdb1   #停用

[root@server0 /]# swapon  -s
```

4.实现开机自动启用交换分区/dev/vdb1

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

/dev/vdb1  swap   swap  defaults 0 0

/dev/vdb2  swap   swap  defaults 0 0
```

```
[root@server0 /]# swapoff /dev/vdb1

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

[root@server0 /]# swapon -s


[root@server0 /]# swapon  -a   #自动检测/etc/fstab 文件 swap 分区

[root@server0 /]# swapon  -s
```

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

基础邮件服务

SMTP:用户发邮件协议  25

pop3:用户收邮件协议  110

- 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

DNS 服务器：classroom.example.com 以 server0.example.com 为例

搭建基本邮件服务

虚拟机 Server0：

1. 安装 postfix 提供邮件功能的软件

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

[root@server0 /]# rpm  -q  postfix

2.修改配置文件/etc/postfix/main.cf

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

  vim 末行模式      ：set  nu

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

116 inet_interfaces = all              #允许本机所有网卡

164 mydestination = server0.example.com  #判断为本域邮件

3.重起 postfix 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

• mail 收信操作

- mail [-u 用户名]

1.创建用户

[root@server0 /]# useradd zhangsan

[root@server0 /]# useradd lisi

2.收发邮件, -s 邮件的主题 -r 收件人

[root@server0 /]# mail -s ’哈哈’ -r zhangsan lisi

AAAAAAAAAAAAA

BBBBBBBBBBBBB

.

EOT

[root@server0 /]# mail -u lisi

Heirloom Mail version 12.5 7/5/10.  Type ? for help.

"/var/mail/lisi": 1 message 1 new

>N  1 zhangsan@server0.exa  Wed Apr 18 14:29  19/631

& 1

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

  # echo xixi  |  mail -s 'xixi' -r zhangsan  lisi

删除 classroom  server  desktop 虚拟机器

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

搭建教学环境

1. 创建两台虚拟机

[root@room9pc01 ~]# clone-vm7

Enter VM number: 8

[root@room9pc01 ~]# clone-vm7

Enter VM number: 9

2.将两台虚拟机，修改名字，一个取名 A　一个取名 B

3.利用 root 用户，密码：123456　登陆两台虚拟系统

4. 设置虚拟机 A

为 eth0IP 地址：192.168.4.7/24

永久的主机名：svr7.tedu.cn

设置虚拟机 B

为 eth0IP 地址：192.168.4.207/24

永久的主机名：pc207.tedu.cn

5.默认情况，教学环境。真机可以直接与 clone-vm7 创建的虚拟机通信，虚拟机 eth0 的 ip 地址必须为 192.168.4.0/24 网络中的 ip 地址

6.进行真机远程管理两台虚拟机

7.Yum 仓库的设置

服务端：真机搭建 Web 服务，共享光盘所有内容

1.安装 httpd 软件

```
# rpm -q httpd
```

2.启动 httpd 服务,设置开机起动

```
# systemctl status httpd        #查看当前的状态
# systemctl is-enabled httpd    #查看是否是开机自起动
# firefox 127.0.0.1             #自己访问自己
```

3.挂载访问测试

```
# ls /var/www/html/rhel7

# mount /iso/rhel-server-7.4-x86_64-dvd.iso
/var/www/html/rhel7/
```

# ls /var/www/html/rhel7

# firefox 127.0.0.1/rhel7

客户端： 虚拟机 A 虚拟机 B

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

[rhel7]

name=rhel7

baseurl=http://192.168.4.254/rhel7

enabled=1

gpgcheck=0

真机上完成开机自动挂载， 光盘文件默认的文件系统为 iso9660

vim /etc/fstab

/var/lib/libvirt/images/iso/rhel-server-7.4-x86_64-dvd.iso /var/www/html/rhel7 iso9660 defaults 0 0

验证：

1.将所有的 loop 设备全部卸载

```
[root@room9pc01 ~]# df -h
/dev/loop0      3.8G  3.8G     0  100% /var/ftp/rhel7
/dev/loop1      3.8G  3.8G     0  100% /var/www/html/rhel7

[root@room9pc01 ~]# umount /var/ftp/rhel7
[root@room9pc01 ~]# umount /var/www/html/rhel7
[root@room9pc01 ~]# df -h     #查看是否卸载成功
[root@room9pc01 ~]# mount -a
[root@room9pc01 ~]# df -h     #查看是否挂载成功
```

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

自定义 Yum 仓库

1.具备非光盘的软件包

将真机的数据传递给虚拟机 A

# scp /路径/源文件　　root@IP 地址 :/路径/

真机上操作：

# scp /root/桌面/tools.tar.gz　root@192.168.4.7:/root/

root@192.168.4.7's password:

2.验证在虚拟机 A 上,查看/root 目录下是否有 tools.tar.gz

3.虚拟机 A 上，tar 解包到根目录下

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

[root@svr7 ~]# ls /

[root@svr7 ~]# ls /tools/

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

4.建立仓库数据文件

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

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

5.虚拟机 A  /tools/other 是一个仓库

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

```
[rhel7]
name=rhel7
baseurl=http://192.168.4.254/rhel7
enabled=1
gpgcheck=0


[other]
name=hahaxixi
baseurl=file:///tools/other      #指定本机为 Yum 仓库
enabled=1
gpgcheck=0
```

[root@svr7 ~]#  yum repolist

6. 使用 Yum 仓库

小火车

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

星球大战背景

```
[root@svr7 ~]# yum -y install cmatrix
```

猫

```
[root@svr7 ~]# yum -y install oneko
[root@svr7 ~]# killall oneko        #杀死所有的 oneko 程序
[root@svr7 ~]# oneko &              #放入后台运行
```

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

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

```
[root@svr7 ~]# mkdir /nsd01
[root@svr7 ~]# ls -ld /nsd01

[root@svr7 ~]# chmod 777 /nsd01
[root@svr7 ~]# ls -ld /nsd01

[root@svr7 ~]# chmod 007 /nsd01
[root@svr7 ~]# ls -ld /nsd01
```

[root@svr7 ~]# chmod 700 /nsd01

[root@svr7 ~]# ls -ld /nsd01

[root@svr7 ~]# chmod 755 /nsd01

[root@svr7 ~]# ls -ld /nsd01

- 附加权限:SUID = 4,SGID = 2,Sticky Bit = 1

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

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

- !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      #查看历史命令

[root@svr7 ~]# history -c   #清空历史命令

[root@svr7 ~]# cat /etc/redhat-release
[root@svr7 ~]# ls -l /etc/redhat-release
[root@svr7 ~]# history

[root@svr7 ~]# !ls      #执行最近一条历史命令以 ls 开头的

[root@svr7 ~]# !cat
```

##########

统计目录的大小

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

[root@svr7 ~]# du -sh /root/
```

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

[root@svr7 ~]# du -sh /

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

zip归档工具,跨平台的压缩工具

命令格式： zip -r /路径/压缩包名 /路径/被压缩的源文件

# zip -r /root/test.zip /home/ /mnt/ /etc/passwd

# ls /root

命令格式： unzip /路径/压缩包名 -d /释放的路径

# unzip /root/test.zip -d /opt

# ls /opt

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

软连接与硬连接，快捷方式

制作快捷方式： ln -s /路径/源文件 /路径/快捷方式的名字

[root@svr7 /]# ln -s /etc/redhat-release /abc

[root@svr7 /]# ls -l /abc

[root@svr7 /]# cat /abc

i 节点: 编号标识硬盘的一块存储区域

[root@svr7 /]# echo 123 > /opt/A

[root@svr7 /]# cat /opt/A

123

[root@svr7 /]# ln -s /opt/A /opt/B

[root@svr7 /]# ls /opt/

ln,创建软连接

- 软连接 --> 原始文档 --> i 节点 --> 文档数据
- ln -s 原始文件或目录 软连接文件

  若原始文件或目录被删除,连接文件将失效

  软连接可存放在不同分区/文件系统

• ln,创建硬连接

- 硬连接 --> i 节点 --> 文档数据
- ln 原始文件 硬连接文件

  若原始文件被删除,连接文件仍可用

  硬连接与原始文件必须在同一分区/文件系统

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

[root@svr7 /]# echo 123 > /opt/A

[root@svr7 /]# cat /opt/A

[root@svr7 /]#  ln -s /opt/A   /opt/B   #软连接

[root@svr7 /]#  ln  /opt/A   /opt/C     #硬连接

[root@svr7 /]#  ls -i  /opt/

[root@svr7 /]#  rm -rf /opt/A   #当删除源文件，只有硬连接可用

[root@svr7 /]#  ls -i  /opt/

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

wc 统计命令

wc -l : 只统计行数

[root@svr7 /]# wc  -l  /etc/passwd

请统计/etc/以 .conf 结尾的文件有多少个（包含子目录）

```
[root@svr7 /]# find /etc/ -name  "*.conf"  |  wc -l

[root@svr7 /]# find /etc/ -name  "*tab"  |  wc -l

[root@svr7 /]# find /boot/ -name "vm*"
[root@svr7 /]# find /boot/ -name "vm*" | wc -l
```

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

vim 文本编辑器

光标跳转

| | |
|---|---|
| Home 键 或 ^、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 |
| 12G | 跳转到文件的第 12 行 |

```
[root@svr7 /]# head -20 /etc/passwd > /opt/test.txt
```

[root@svr7 /]# vim /opt/test.txt

复制/粘贴/删除

复制 yy、#yy 复制光标处的一行、#行

粘贴 p 粘贴到光标处之后

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，并且进入插入模式

u 撤销上一次操作

[root@svr7 /]# vim /opt/test.txt

查找/撤销/保存

/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

:r /etc/filesystems 读入其他文件内容

[root@svr7 /]# echo 123 > /opt/1.txt

[root@svr7 /]# echo abc > /opt/2.txt

[root@svr7 /]# vim /opt/1.txt

末行模式下　　: r　　/opt/2.txt

:r　　/etc/passwd

字符串替换

:s/root/admin　　替换光标所在行第一个“root”

:s/root/admin/g　　替换光标所在行所有的“root”

:1,10s/root/admin/g　替换第 1-10 行所有的“root”

:%s/root/admin/g　　替换文件内所有的“root

[root@svr7 /]# vim /opt/test.txt

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

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

源码编译安装

rpm 包：rpm　yum　　简单　不灵活

利用 gcc 与 make

源码包---------->可以执行的程序 ---------->

运行安装

灵活，功能与安装位置 可以自由选择

源码包编译安装：

步骤 1：安装 gcc 与 make 编译开发工具

[root@svr7 /]# yum -y install gcc make

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 /mnt/

# ls /mnt/

# cd /mnt/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 1：检测系统是否安装 gcc

作用 2：功能与安装位置 可以自由选择

--prefix= 指定安装位置

```
# cd /mnt/inotify-tools-3.13/
# ./configure  --prefix=/opt/myrpm
```

常见报错：没有安装 gcc

```
checking for gcc... no
checking for cc... no
checking for cl.exe... no
```

步骤 4:make 编译,生成可执行的二进制程序文件

步骤 5:make install 安装,将编译好的文件复制到安装目录

验证：

```
[root@svr7 inotify-tools-3.13]# ls /opt/
[root@svr7 inotify-tools-3.13]# ls /opt/myrpm/
[root@svr7 inotify-tools-3.13]# ls /opt/myrpm/bin/
```

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

systemctl 控制

服务的控制

开启服务　systemctl　start　服务名

停止服务　systemctl　stop　服务名

重起服务　systemctl　restart　服务名

设置开机自起　　systemctl　enable　服务名

设置开机不自起　systemctl　disable　服务名

查看当前是不是开机自起　systemctl　is-enabled　服务名

模式的控制

纯文本字符模式 ： multi-user.target

图形模式 ： graphical.target

当前切换模式

```
# systemctl isolate  graphical.target   #切换到图形模式
# systemctl isolate  multi-user.target   #切换到字符模式
```

修改默认进入的模式

```
# systemctl get-default   #查看默认进入的模式

# systemctl set-default graphical.target  #修改默认的模式

# systemctl get-default

# reboot  #重起验证
```

检测虚拟机 A 与虚拟机 B 的 Yum 仓库

```
[root@svr7 ~]# yum clean all    #清空缓存
[root@svr7 ~]# yum repolist     #列出仓库信息
```

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

DNS 服务：域名解析　　将域名解析 ip 地址

DNS 服务器的功能

- 正向解析:根据注册的域名查找其对应的 IP 地址
- 反向解析:根据 IP 地址查找对应的注册域名,不常用

所有完整的域名都要以点结尾：　www.qq.com.

www.baidu.com.

.　根域

一级域名：　.cn　.us　.hk　.tw　.kr　.com　.net

二级域名:　　.com.cn　.net.cn　.org.cn .edu.cn ......

三级域名:　　nb.com.cn　dawai.com.cn　haha.com.cn

haxi.com.cn

完整的主机名:　www.nb.com.cn　ftp.nb.com.cn

tts.nb.com.cn

• Full Qualified Domain Name,完全合格主机名(FQDN)

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

- 系统服务: named
- 默认端口: 53
- 运行时的虚拟根环境:/var/named/chroot/

• 主配置文件:/etc/named.conf　#设置本机负责解析的域名 tedu.cn

• 地址库文件:/var/named/　　# 主机名与 IP 地址的对应记录

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

搭建基本的 DNS 服务

服务端：虚拟机 A

1.安装一个可以提供域名解析的软件

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

bind-9.9.4-29.el7.x86_64　　//域名服务包

bind-chroot-9.9.4-29.el7.x86_64　　//提供虚拟根支持，牢笼政策

2.修改主配置文件/etc/named.conf

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

```
options {
    directory   "/var/named";     #设置地址库文件路径
};
zone  "tedu.cn"  IN  {            #设置负责解析的域名
```

```
        type master;                    #设置本机为主DNS服务器
        file "tedu.cn.zone";            #设置地址库文件名字
    };
```

3. 建立地址库文件/var/named/tedu.cn.zone

-p:权限不变进行拷贝

```
# cp -p /var/named/named.localhost /var/named/tedu.cn.zone
```

```
# vim  /var/named/tedu.cn.zone
    tedu.cn.        NS      svr7            #指定本区域DNS服务器
    svr7            A       192.168.4.7     #指定DNS服务器的IP
地址
    www             A       1.1.1.1
    ftp             A       2.2.2.2
```

```
# systemctl restart named   #重起服务
```

客户端：虚拟机 B

验证 DNS 解析

```
# echo nameserver 192.168.4.7 > /etc/resolv.conf
# nslookup   www.tedu.cn  #测试域名解析
# nslookup   ftp.tedu.cn  #测试域名解析
```

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

搭建多区域的 DNS

解析 qq.com 域名

1.修改主配置文件

```
zone  "qq.com"  IN  {
    type master;
    file "qq.com.zone";
};
```

2.建立地址库文件 /var/named/qq.com.zone

| | | |
|---|---|---|
| qq.com. | NS | svr7 |
| svr7 | A | 192.168.4.7 |
| www | A | 30.30.30.30 |
| ftp | A | 4.4.4.4 |

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

特殊的解析记录

一、DNS 的轮询

虚拟机 A：

[root@svr7 ~]# vim /var/named/tedu.cn.zone

| | | |
|---|---|---|
| tedu.cn. | NS | svr7 |
| svr7 | A | 192.168.4.7 |
| www | A | 192.168.4.10 |

```
www             A       192.168.4.20
www             A       192.168.4.30
ftp             A       2.2.2.2
```

```
[root@svr7 ~]# systemctl restart named
```

虚拟机 B：验证

```
[root@pc207 ~]# ping -c 1 www.tedu.cn
[root@pc207 ~]# ping -c 1 www.tedu.cn
[root@pc207 ~]# ping -c 1 www.tedu.cn
```

二、泛域名解析

虚拟机 A：

```
[root@svr7 ~]# vim /var/named/tedu.cn.zone
```

```
*               A       10.20.30.40
tedu.cn.        A       50.60.70.80
```

[root@svr7 ~]# systemctl restart named

虚拟机 B：

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

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

[root@pc207 ~]# nslookup  tedu.cn

三、有规律的泛域名解析

stu1.tedu.cn--------》192.168.10.1

stu2.tedu.cn--------》192.168.10.2

stu3.tedu.cn--------》192.168.10.3

stu4.tedu.cn--------》192.168.10.4

.....

stu50.tedu.cn--------》192.168.10.50

$GENERATE 造数工具

虚拟机 A：

[root@svr7 ~]# vim /var/named/tedu.cn.zone

```
*                A        10.20.30.40
tedu.cn.         A        50.60.70.80
$GENERATE  1-50  stu$   A  192.168.10.$
```

```
[root@svr7 ~]# systemctl restart named
```

虚拟机 B：

```
[root@svr7 ~]# nslookup  stu32.tedu.cn
[root@svr7 ~]# nslookup  stu50.tedu.cn
[root@svr7 ~]# nslookup  stu51.tedu.cn
```

四、解析记录的别名

虚拟机 A：

```
[root@svr7 ~]# vim /var/named/tedu.cn.zone
```

```
  tts            CNAME    ftp
```

```
[root@svr7 ~]# systemctl restart named
```

虚拟机 B：

[root@svr7 ~]# nslookup  tts.tedu.cn

[root@svr7 ~]# nslookup  ftp.tedu.cn

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

DNS 子域授权

父域:  www.tedu.cn  虚拟机 A

子域:  www.bj.tedu.cn  虚拟机 B

一、虚拟机 B  搭建子域的 DNS

1.安装一个可以提供域名解析的软件

[root@pc207 ~]# yum -y install bind-chroot bind

2.修改主配置文件/etc/named.conf

  options {

```
        directory   "/var/named";
    };
    zone  "bj.tedu.cn"  IN  {
        type master;
        file "bj.tedu.cn.zone";
    };
```

3. 建立地址库文件/var/named/bj.tedu.cn.zone

```
# cp -p /var/named/named.localhost
/var/named/bj.tedu.cn.zone

# vim  /var/named/bj.tedu.cn.zone
    bj.tedu.cn.        NS        pc207
    pc207               A        192.168.4.207
    www                 A        1.2.3.4
# systemctl restart named   #重起服务
```

```
[root@pc207 /]# nslookup www.bj.tedu.cn 192.168.4.207
```

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

子域授权

向 父域的DNS虚拟机A 解析 子域的域名 可以得到 虚拟机B 的结果

虚拟机A：

1.修改地址库文件/var/named/tedu.cn.zone

| | | |
|---|---|---|
| tedu.cn. | NS | svr7 |
| bj.tedu.cn. | NS | pc207 |
| svr7 | A | 192.168.4.7 |
| pc207 | A | 192.168.4.207 |

2.重起named服务

测试验证

[root@pc207 /]# nslookup www.bj.tedu.cn 192.168.4.7

Server: 192.168.4.7

Address: 192.168.4.7#53

Non-authoritative answer:  #非权威解答

Name: www.bj.tedu.cn

Address: 1.2.3.4

DNS 解析过程：

递归解析：指 DNS 服务器与其他 DNS 服务器交互，最终将解析结果带回来的过程

迭代解析：指 DNS 服务器与其他 DNS 服务器交互，最终将告知下一个 DNS 服务

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

验证迭代查询

- dig 命令,更专业的 DNS 测试工具

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

缓存 DNS

作用：缓存解析结果，提高解析速度

搭建方式：

1.全局转发，所有的DNS解析请求全部转发给公网DNS

2.根域迭代，所有的DNS解析请求全部发给根域DNS服务器

真机上实现缓存DNS服务器

1.搭建Yum仓库

```
# mkdir /dvd
# mount /iso/CentOS-7-x86_64-DVD-1708.iso  /dvd

# ls /dvd
# rm -rf /etc/yum.repos.d/*
# vim /etc/yum.repos.d/dvd.repo

  [dvd]
  name=dvd
  baseurl=file:///dvd
  enabled=1
  gpgcheck=0
```

```
[root@room9pc01 ~]# yum -y install bind bind-chroot
```

2.确认真机的DNS服务器

```
[root@room9pc01 ~]# cat /etc/resolv.conf
 nameserver  172.40.1.10
```

3.真机修改配置文件/etc/named.conf

```
 options {
         directory        "/var/named";
         forwarders {    172.40.1.10;    };  #转发给
172.40.1.10
 };
```

```
[root@room9pc01 ~]# systemctl restart named
```

4.在虚拟机上解析

```
[root@svr7 /]# nslookup www.qq.com 192.168.4.254
```

[root@svr7 /]# nslookup www.360.com 192.168.4.254

检测虚拟机 A 与虚拟机 B 的 Yum 仓库

[root@svr7 ~]# yum clean all    #清空缓存

[root@svr7 ~]# yum repolist     #列出仓库信息

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

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

搭建基本的 DNS 服务

服务端：虚拟机 A

1.安装一个可以提供域名解析的软件

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

2.修改主配置文件/etc/named.conf

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

```
  options {
     directory   "/var/named";
  };
  zone  "sina.com"  IN  {
```

```
        type master;
        file "sina.com.zone";
    };
```

3. 建立地址库文件/var/named/sina.com.zone  -p:权限不变进行拷贝

```
# cp -p /var/named/named.localhost /var/named/sina.com.zone
# vim  /var/named/sina.com.zone
    sina.com.       NS      svr7            #指定本区域DNS服务器
    svr7            A       192.168.4.7   #指定DNS服务器的IP地址
    www             A       19.18.16.17
# systemctl restart named   #重起服务
```

客户端：虚拟机B 验证DNS解析

```
 # echo nameserver 192.168.4.7 > /etc/resolv.conf
 # nslookup   www.sina.com  #测试域名解析
```

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

DNS分离解析

什么是分离解析

- 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)
  - 根据客户端的不同，解析同一个域名，得到的解析结果不同
  - 目的：为客户端提供网络最近的服务器资源

案例需求及要点

- 环境及需求
  - 权威 DNS:svr7.tedu.cn 192.168.4.7
  - 负责区域:sina.com
  - A 记录分离解析 —— 以 www.sina.com 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

1.由上到下依次匹配， 匹配及停止

2.使用 view 时，所有的客户端都必须在分类中

3.所有的 zone 都必须在 view 中

4.每一个 view 中的 zone 必须相同

```
view "lan" {
  match-clients {  192.168.4.207; };
  zone "sina.com" IN {
  ......  sina.com.lan;
  };
192.168.4.100
};
view "other" {
   match-clients { any; };
```

```
    zone "sina.com"  IN  {
    ......   sina.com.other;
    };
};
```

虚拟机 A：

1.修改配置文件/etc/named.conf

```
view "nsd" {
 match-clients {   192.168.4.207;    };
 zone "sina.com" IN {
        type master;
        file "sina.com.nsd";
    };
  };
```

```
view "other" {
 match-clients {    any;    };
 zone "sina.com" IN {
        type master;
        file "sina.com.other";
   };
 };
```

2.建立 sina.com.nsd、sina.com.other 地址库文件 ， 写入不同的解析结果

sina.com.nsd 解析结果为 192.168.4.100

sina.com.other 解析结果为 1.2.3.4

3.修改/etc/named.conf

补充：vim 可视模式 在命令模式下 按 ctrl+v 进入可视模式 可以选择列

按 大写的 I 进入插入模式，然后输入内容 按 Esc 退回到命令模式

```
#############################################################
##
acl 地址列表  简化 match-clients 匹配

acl test { 192.168.4.207; 192.168.200; 192.168.4.250;
192.168.1.1; };

view "nsd" {
 match-clients {   test;    };
 zone "sina.com" IN {
        type master;
        file "sina.com.nsd";
  };
};

#############################################################
补充： vim 默认配置文件   ~/.vimrc
          初始化 vim 操作，每次使用 vim 命令首先执行的内容
```

```
[root@svr7 /]# vim /root/.vimrc


  set nu           #开启行号
  set ai           #启用自动缩进
  set tabstop=2    #一个 tab 键等于 两个空格
```

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

RAID 磁盘管理

- 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

- 至少由 2 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
  - 至少由 2 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
  - 至少由 3 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
  - 至少由 4 块磁盘组成

- RAID 0+1/RAID 1+0
  - 整合 RAID 0、RAID 1 的优势
  - 并行存取提高效率、镜像写入提高可靠性
  - 至少由 4 块磁盘组成

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

进程管理

程序：静态的代码，占用磁盘空间

进程：动态的代码，会占用 CPU  内存

进程的唯一标识：PID

父进程与子进程  树形结构

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

查看进程树

- pstree — Processes Tree

- 格式:pstree [选项] [PID 或用户名]

• 常用命令选项

- -a:显示完整的命令行

- -p:列出对应 PID 编号

systemd 是所有进程的父进程

```
[root@svr7 /]# pstree              #显示进程树

[root@svr7 /]# pstree  lisi        #显示用户 lisi 的进程

[root@svr7 /]# pstree -p lisi      #显示进程 PID

[root@svr7 /]# pstree -ap lisi     #显示进程完整的信息
```

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

• ps aux 操作

- 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

- 列出正在运行的所有进程

PPID:父进程的 PID 号

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

进程动态排名

• top 交互式工具

- 格式:top [-d 刷新秒数] [-U 用户名]

按 大写的 P 按 cpu 降序

按 大写的 M 按内存降序

```
[root@svr7 /]# top -d  1


[root@svr7 /]# ps aux
[root@svr7 /]# ps -elf
[root@svr7 /]# ps aux   |  wc  -l   #统计进程数
[root@svr7 /]# ps -elf  |  wc  -l   #统计进程数
```

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

检索进程

- pgrep — Process Grep
  - 用途:pgrep [选项]... 查询条件

- 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  atd

[root@svr7 /]# pgrep -l  cron


[root@svr7 /]# pgrep -lU lisi


[root@svr7 /]# pgrep -lU lisi | wc -l


[root@svr7 /]# pgrep -lx atd
```

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

控制进程

```
[root@svr7 /]# sleep 900 &   #正在运行放入后台

[root@svr7 /]# jobs          #查看后台进程信息

[root@svr7 /]# sleep 800

^Z                           #按 Ctrl+z 正在运行放入后台

[2]+  已停止                 sleep 800

[root@svr7 /]# jobs

[root@svr7 /]# bg 2        #将后台编号为 2 的进程继续运行
```

```
[root@svr7 /]# jobs
[root@svr7 /]# fg 1          #将后台编号为1的进程恢复到前台
sleep 900
^C                           #按Ctrl+c结束
[root@svr7 /]# jobs
[root@svr7 /]# fg 2          #将后台编号为2的进程恢复到前台
sleep 800
^C                           #按Ctrl+c结束
[root@svr7 /]# jobs
```

杀死进程

- Ctrl+c 组合键,中断当前命令程序
- kill [-9] PID... 、kill [-9] %后台任务编号
- killall [-9] 进程名...
- pkill 查找条件

-9：强制杀

- killall -9 -u 用户名 #杀死该用户开启的所有进程

强制 踢出 一个用户

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

日志管理

日志的功能

- 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

- 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

tailf ：实时跟踪日志信息

- awk、sed 等格式化过滤工具

- 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   #最近登陆成功 2 条记录

[root@svr7 /]# lastb  -2  #最近登陆失败 2 条记录

- 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 |

DNS 分离解析综合实验：

虚拟机 C 搭建 Web 服务器

1.安装 httpd

2.修改配置文件/etc/httpd/conf.d/nsd01.conf

```
# mkdir /var/www/qq /var/www/163
# echo woshi Web1 QQ > /var/www/qq/index.html
# echo woshi Web1 163 > /var/www/163/index.html

# cat /etc/httpd/conf.d/nsd01.conf
<VirtualHost  *:80>
   ServerName www.qq.com
   DocumentRoot /var/www/qq
</VirtualHost>
<VirtualHost  *:80>
   ServerName www.163.com
   DocumentRoot /var/www/163
</VirtualHost>
```

虚拟机 D 搭建 Web 服务器

1.安装 httpd

2.修改配置文件/etc/httpd/conf.d/nsd01.conf

```
# mkdir /var/www/qq /var/www/163
# echo woshi Web2 QQ > /var/www/qq/index.html
# echo woshi Web2 163 > /var/www/163/index.html

# cat /etc/httpd/conf.d/nsd01.conf
<VirtualHost  *:80>
   ServerName www.qq.com
   DocumentRoot /var/www/qq
</VirtualHost>
<VirtualHost  *:80>
   ServerName www.163.com
   DocumentRoot /var/www/163
</VirtualHost>
```

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

实现的效果

192.168.4.7-> www.qq.com  www.163.com---》Web1（10）

192.168.4.207-> www.qq.com  www.163.com---》Web2（20）

在虚拟机 A 搭建 DNS 分离解析

```
view "nsd" {
 match-clients {   192.168.4.7;    };
 zone "qq.com" IN {
  type master;
  file "qq.com.nsd";
  };
 zone "163.com" IN {
  type master;
  file "163.com.nsd";
  };
};

view "other" {
 match-clients {   any;    };
```

```
 zone "qq.com" IN {
  type master;
  file "qq.com.other";
  };
 zone "163.com" IN {
  type master;
  file "163.com.other";
  };
};
```

指定 DNS 服务器

```
# echo nameserver 192.168.4.7 > /etc/resolv.conf
```

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

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 | grep 关键词

- journalctl -u 服务名 [-p 优先级]

- journalctl -n 消息条数

- journalctl --since="yyyy-mm-dd HH:MM:SS" --until="yyyy-mm-dd HH:MM:SS"

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

搭建 DHCP 服务

DHCP 概述及原理

• 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

- DHCP 地址分配的四次会话,通过广播的方式，先到先得

  一个网络中不能有多台 DHCP

虚拟机 A：

1.安装软件包

[root@svr7 /]# yum -y install dhcp

2.修改主配置文件

[root@svr7 /]# vim  /etc/dhcp/dhcpd.conf

末行模式  :r /usr/share/doc/dhcp*/dhcpd.conf.example

```
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;
}
```

```
[root@svr7 /]# systemctl restart dhcpd
```

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

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

- 网络装机服务端需要哪些服务组件?
  - 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

网络装机服务器虚拟机 A

一、dhcp 配置，指定 next-server 下一个服务器

1.修改配置文件

[root@svr7 /]# vim /etc/dhcp/dhcpd.conf

.......

  next-server 192.168.4.7;

  filename   "pxelinux.0";       #指定网卡引导文件名

}

2.重起 dhcpd 服务

[root@svr7 /]# systemctl restart dhcpd

pxelinux.0：安装说明书,二进制文件，安装一个软件自动生成

pxelinux.0-----》读取菜单文件

/var/lib/tftpboot/pxelinux.cfg/default

二、搭建 tftp 服务，传输引导文件

tftp:简单的文件传输协议  端口：69

默认共享路径：/var/lib/tftpboot

1.安装软件包 tftp-server

2.重起 tftp 服务

[root@svr7 /]# systemctl restart tftp

[root@svr7 /]# systemctl enable tftp

3.部署 pxelinux.0 文件

#查询仓库中软件生成 pxelinux.0

```
#  yum  provides  */pxelinux.0
#  yum -y install syslinux
#  rpm -ql syslinux      #查看软件包安装清单
#  rpm -ql syslinux | grep pxelinux.0

# cp /usr/share/syslinux/pxelinux.0  /var/lib/tftpboot/
```

```
# ls /var/lib/tftpboot/
```

4. 部署菜单文件

```
# mkdir  /var/lib/tftpboot/pxelinux.cfg
# ls  /var/lib/tftpboot/

# mount  /dev/cdrom   /mnt/
# ls  /mnt/isolinux/

# cp /mnt/isolinux/isolinux.cfg
/var/lib/tftpboot/pxelinux.cfg/default

# ls -l  /var/lib/tftpboot/pxelinux.cfg/default
# chmod 644 /var/lib/tftpboot/pxelinux.cfg/default
# ls -l  /var/lib/tftpboot/pxelinux.cfg/default
```

5. 部署启动内核  启动驱动程序

vmlinuz 启动内核

initrd.img  启动驱动程序

# cp /mnt/isolinux/vmlinuz /mnt/isolinux/initrd.img /var/lib/tftpboot/

# ls /var/lib/tftpboot/

6. 部署　图形的模块　　背景图片

　　vesamenu.c32 图形的模块

　　splash.png　背景图片

[root@svr7 /]# cp /mnt/isolinux/vesamenu.c32 /mnt/isolinux/splash.png /var/lib/tftpboot/

[root@svr7 /]# ls /var/lib/tftpboot/

 initrd.img　pxelinux.cfg　vesamenu.c32

 pxelinux.0　splash.png　　vmlinuz

7.修改菜单文件内容

 # vim　/var/lib/tftpboot/pxelinux.cfg/default

　default vesamenu.c32　#加载图形的模块

　timeout 60　　　　#读秒的时间

```
    .......
  menu background splash.png       #指定背景图片
  menu title PXE Server NSD1803    #指定标题
    .......


 label linux
    menu label Install RHEL7    #显示选项内容
    kernel vmlinuz              #指定启动内核
    append initrd=initrd.img    #指定启动驱动程序
```

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

总结：

dhcp----》IP、next-server、pxelinux.0

tftp----》pxelinux.0

pxelinux.0----》default

default----》vesamenu.c32 、vmlinuz、initrd.img

简单测试: 新建一台虚拟机 选择 PXE 网络引导 安装

网络类型选择为 private1

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

三、搭建 Web 服务，共享光盘所有内容

1.安装软件包

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

2.创建目录

[root@svr7 /]# mkdir /var/www/html/rhel7

[root@svr7 /]# ls /var/www/html/rhel7

[root@svr7 /]# mount /dev/cdrom  /var/www/html/rhel7

[root@svr7 /]# ls /var/www/html/rhel7

[root@svr7 /]# systemctl restart httpd

[root@svr7 /]# systemctl enable httpd

3.测试

[root@svr7 /]# firefox 192.168.4.7/rhel7

四、部署应答文件，无人值守安装

1.图形生成应答文件的工具  system-config-kickstart

# yum  -y install system-config-kickstart

2.修改 Yum 仓库标识

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

[development]

[root@svr7 /]# system-config-kickstart

查看软件包是否可以选择

3.查看应答文件

[root@svr7 /]# ls /root/ks.cfg

/root/ks.cfg

4.利用 Web 服务，共享应答文件

[root@svr7 /]# cp /root/ks.cfg  /var/www/html

[root@svr7 /]# ls /var/www/html

5.修改菜单文件指定应答文件

```
 # vim /var/lib/tftpboot/pxelinux.cfg/default
.........
label linux
  menu label Install RHEL7
  menu  default                    #读秒之后默认的选择
  kernel vmlinuz
  append initrd=initrd.img ks=http://192.168.4.7/ks.cfg
```

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：192.168.4.168/24

4.搭建 Yum 仓库

5.主机名：PXE.tedu.cn

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

搭建一键装机平台

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  192.168.4.168;
  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
  kernel rhel7/vmlinuz
  append initrd=rhel7/initrd.img

label vesa
  menu label Install RHEL6
```

```
  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，从本地硬盘启动
  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://192.168.4.168/rhel6

[root@PXE /]# firefox  http://192.168.4.168/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://192.168.4.168/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
  kernel rhel7/vmlinuz
  append initrd=rhel7/initrd.img ks=http://192.168.4.
168/ks7.cfg

label vesa
  menu label Install RHEL6
```

kernel rhel6/vmlinuz

append initrd=rhel6/initrd.img ks=http://192.168.4.168/ks6.cfg

rsync 同步服务

- 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

- 命令用法
  - rsync [选项...]  源目录/  目标目录
  - rsync [选项...]  源目录  目标目录

  - -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

本地目录的同步

```
[root@svr7 ~]# mkdir /dir1
[root@svr7 ~]# mkdir /stu01

# cp /etc/passwd /etc/group /etc/fstab    /dir1/

[root@svr7 ~]# ls /dir1/
[root@svr7 ~]# rsync -avz /dir1/  /stu01/
[root@svr7 ~]# ls /stu01/

[root@svr7 ~]# cp /etc/resolv.conf /dir1/
[root@svr7 ~]# rsync -avz /dir1/  /stu01/
[root@svr7 ~]# ls /stu01/
[root@svr7 ~]# echo haha >> /dir1/group
```

[root@svr7 ~]# rsync -avz /dir1/  /stu01/

[root@svr7 ~]# echo xixi > /dir1/resolv.conf

[root@svr7 ~]# rsync -avz /dir1/  /stu01/

[root@svr7 ~]# ls /stu01/

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

rsync+SSH 远程同步

  rsync  /本地路径/源文件/    用户名@对方的 IP 地址:/目标路径

  rsync   用户名@对方的 IP 地址:/目标路径        /本地路径/路径/

虚拟机 A:

# rm -rf /opt/*

# touch /opt/{1..5}.txt

# ls /opt/

# rsync --delete -avz  /opt/  root@192.168.4.207:/opt/

虚拟机 B:

# ls /opt/

虚拟机 A:

# touch /opt/6.txt

# rsync --delete -avz  /opt/  root@192.168.4.207:/opt/

虚拟机 B:

# ls /opt/

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

 实时的远程同步

一、取消用户名密码的验证

  1.虚拟机 A  生成公钥  私钥

 [root@svr7 ~]# ssh-keygen        #一路回车

 [root@svr7 ~]# ls  /root/.ssh/  #查看公钥  私钥

2.虚拟机 A  传递公钥 到对方服务器

[root@svr7 ~]# ssh-copy-id  root@192.168.4.207

3.虚拟机 A 验证

[root@svr7 ~]# ssh  root@192.168.4.207

二、源码包编译安装 inotify-tools(监控目录内容变化)

1.真机上传到虚拟机 A

# scp /var/ftp/NSD1803/SERVICES/Day01/tools.tar.gz root@192.168.4.7:/

2.安装 gcc make

3.解包 tar

# tar -xf /tools/inotify-tools-3.13.tar.gz  -C  /

# cd /inotify-tools-3.13/

4.配置./configure

[root@svr7 inotify-tools-3.13]# ./configure

5.编译与安装

[root@svr7 inotify-tools-3.13]# make

[root@svr7 inotify-tools-3.13]# make install

[root@svr7 /]# inotifywait    #具备命令程序

No files specified to watch!

6.程序的使用

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

[root@svr7 /]# inotifywait -mrq /opt/

三、书写 rsync 脚本

死循环脚本

while [条件判断]
do
循环的语句
done

```
[root@svr7 /]# vim  /root/rsync.sh

#!/bin/bash
 while  inotifywait -rqq /opt/
 do
  rsync --delete -az  /opt/  root@192.168.4.207:/opt
 done  &

[root@svr7 /]# chmod  +x  /root/rsync.sh
```

一、搭建 CentOS 虚拟机

新建一台虚拟机，放入 CentOS7 的光盘

内存为 2G，硬盘 10G

网络类型为 private1

分区自动选择，设置 root 密码，建立普通用户

二、虚拟机设置

1.设置防火墙为 trusted

2.当前及永久设置 SELinux 状态为 permissive

3.配置 IP 地址：192.168.4.180/24

4.利用本地/dev/cdrom 挂载,搭建本地 Yum 仓库

5.主机名：cobbler.tedu.cn

三、利用 scp 真机传递 Cobbler.zip 包到虚拟机 192.168.4.180 中

# scp /root/桌面/Cobbler.zip  root@192.168.4.180:/root/

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

[root@cobbler /]# unzip /root/Cobbler.zip -d /

[root@cobbler /]# cd /Cobbler/

[root@cobbler Cobbler]# unzip cobbler.zip　　#解压到当前目录

[root@cobbler Cobbler]# ls

[root@cobbler Cobbler]# cd cobbler

[root@cobbler cobbler]# ls

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

五、安装cobbler主程序、工具包等

[root@cobbler /]# yum -y install /Cobbler/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@svr7 /]# vim /etc/cobbler/settings

next_server:  192.168.4.180　　#设置下一个服务器还为本机

server: 192.168.4.180　　#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

开机启动：匹配及停止

1.硬盘启动　2.光驱设备　3.U 盘　4.网络引导

3.配置 cobbler 的 dhcp

[root@svr7 /]# vim /etc/cobbler/dhcp.template

:%s  /192.168.1/192.168.4/g

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

# 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
[root@svr7 /]# systemctl enable rsyncd
```

6.同步刷新 cobbler 配置

```
[root@svr7 /]# cobbler sync
[root@svr7 /]# firefox https://192.168.4.180/cobbler_web
```

用户名：cobbler

密码：cobbler

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

cobbler import  --path=挂载点  --name=导入系统命名（随意起）

导入安装镜像数据

```
[root@Cobbler ~]# mount /dev/cdrom  /dvd
mount: /dev/sr0 is write-protected, mounting read-only
```

```
[root@room9pc01 /]# cobbler list   #查看有哪些系统
```

[root@room9pc01 /]# cobbler import --path=/dvd --name=CentOS7

[root@room9pc01 /]# mkdir /rhel7

[root@room9pc01 /]# mount /iso/rhel-server-7.4-x86_64-dvd.iso /rhel7/

[root@room9pc01 /]# cobbler import --path=/rhel7 --name=RedHat7

cobbler 导入的镜像放在：/var/www/cobbler/ks_mirror

[root@room9pc01 /]# cobbler profile report #查看 cobbler 导入信息

[root@room9pc01 /]# killall -9 dnsmasq #虚拟化服务会干扰 DHCP 服务

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

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/CentOS-7.3-x86_64.cfg

[root@cobbler ~]# cobbler profile report

[root@cobbler ~]# cobbler sync

网络

一、 TCP/IP 协议

1、TCP/IP 通信协议是目前最完整、最被广泛支持的通信协议，它可以让不同网络架构、不同操作系统的计算机之间通信，是 Internet

的标准通信协议

2、主机与主机之间通信三个要素

IP 地址（IP address）

子网掩码（subnet mask）

IP 路由（IP router）

二、 IP 地址

1、 作用：用来标识一个节点的网络地址

2、 组成：32 位，以 4 个十进制数来表示，之间用 . 隔开

网络位+主机位

3、 分类：

A 1 ~ 126 网+主+主+主

B 128 ~ 191 网+网+主+主

C 192 ~ 223 网+网+网+主

D 224 ~ 239 组播(多播)

E 240 ~ 254 科研

4、默认子网掩码

A 类 255.0.0.0

B 类 255.255.0.0

C 类 255.255.255.0

5、网关

就是一个网络连接到另一个网络的“关口”

6、公有 IP 地址和私有 IP 地址

公有地址，也可称为公网地址，通过它直接访问因特网，它是广域网范畴内的。

私有地址，也可称为专网地址，专门为组织机构内部使用，它是局域网范畴内的，出了所在局域网是无法访问因特网的。

私有地址范围：

A 类　10.0.0.1 ~ 10.255.255.254

B 类　172.16.0.1 ~ 172.31.255.254

C 类　192.168.0.1 ~ 192.168.255.254

Windows 系统中查看 IP 地址的配置信息

右击桌面网络图标—属性—更改适配器设置—双击本地连接—单击“详细信息

或者开始 | 命令提示符 | 运行 ipconfig

或 ipconfig /all

计算机网络的功能

数据通信、资源共享、增加数据可靠性、提高系统处理能力

计算机网络的发展

60 年代:分组交换

70-80 年代:TCP/IP

90 年后:Web 技术

网络协议与标准

1、协议：一组控制数据通信的规则，协议的三要素：语法、语义，同步。

2、标准：一致同意的规则可以理解为标准

ISO（国际标准化组织）在网络通信中创建了 OSI（开放系统互联）模型。

ANSI（美国国家标准化局）

ITU-T（国际电信联盟-电信标准部）

IEEE（电气和电子工程师学会）

按照网络规模和使用范围分类为

WAN：广域网　　LAN:局域网

网络设备

交换机、路由器

网络安全设备：防火墙、VPN 设备

网络设备生产厂商：cisco 思科，华为。

网络拓扑结构

1、 点对点

两台设备之间有一条单独的连接

2、 星型拓扑

1） 优点：易于实现、易于网络扩展、易于故障排查

2） 缺点：中心节点压力大、组网成本较高

3、网型拓扑结构

1） 各个节点至少与其他两个节点相连

2） 可靠性高、组网成本高

带宽

单位为比特/秒，记为 bit/s、 b/s、 bps

8bit=1byte

OSI

1、　　国际标准化组织（ISO）

开放系统互连参考模型 OSI

OSI 是一个开放式体系结构，它规定将网络分为七层

2、协议分层

为了降低网络设计的复杂性，将协议进行了分层设计

应用层：网络服务与最终用户的一个接口

表示层：数据的表现形式，如加密、压缩。

会话层：建立、管理、中止会话，例如断点续传。

传输层：定义传输数据的协议端口号，以及流控和差错校验，实现了程序与程序的互连，可靠与不可靠的传输。

网络层：进行逻辑地址寻址，实现不同网络之间的通信，定义了 IP 地址，为数据传输选择最佳路径，路由器工作在网络层。

数据链路层：建立逻辑连接、进行硬件地址寻址、差错校验等功能、通过 MAC 地址实现数据的通信，帧包装、帧传输、帧同步。交换机工作在数据链路层。

物理层：建立、维护、断开物理连接，定义了接口及介质，实现了比特流的传输。

数据解封装过程

数据在传输的过程中从上层入下层进行封装，对于接收方从底层到顶层进行解封装。

TCP/IP 协议族的组成

应用层：HTTP、https、FTP、TFTP、SMTP 、pop3、SNMP、DNS 、telnet

传输层：TCP、UDP

网络层：ICMP、IGMP、IP、ARP、RARP

PDU（协议数据单元）

传输层　　段　segment

网络层　　包　packet

数据链路层 帧　frame

物理层　　比特 bit

相应层次的设备

应用层　　计算机

传输层　　防火墙

网络层　　路由器

数据链路层 交换机

物理层　　网卡

接口

以太网接口：

RJ-45 水晶头

光纤接口：

FC 、ST、SC

LC 窄体方形光纤接头（目前主流）

MT-RJ

双绞线

1） 双绞线分类：

屏蔽双绞线 （STP）

线外包裹一层金属网膜，用于电磁环境非常复杂的工业环境中

非屏蔽双绞线 （UTP）

2） 双绞线标准与分类 ：

Cat5 五类双绞线，适用于 100Mbps 的网络

Cat 5e 衰减更小，适用于 100Mbps 的网络，串扰更少，性能优于 5 类线 （超五类）

Cat 6 适用于 1000Mbps 的网络

Cat 7 适用于 10000Mbps 的网络

4、双绞线的连接规范

1）线序

T568A：

白绿、绿、白橙、蓝、白蓝、橙、白棕、棕

T568B：

白橙、橙、白绿、蓝、白蓝、绿、白棕、棕

2）线缆的连接：

标准网线（直连线或直通线）：用于连接不同设备（A-A，B-B）

交叉网线：用于连接相同设备 （A-B）

全反线 ：不用于以太网的连接，主要用于计算机的串口和路由器或交换机的 console（控制口）相连

特例：计算机直接连接路由器用交叉线 交换机与交换机相连使用交叉或直连线

5、物理层设备

网卡、中继器

交换机的工作模式：

Switch>用户模式

Switch>enable

Switch#特权模式

Switch#configure  terminal

Switch(config)#全局配置模式

Switch(config)#interface fastEthernet 0/1

Switch(config-if)#接口模式

exit 返回上一模式

end 直接退到特权模式

常用命令：

Switch(config)#hostname S1 修改主机名为 S1

Switch#show  running-config 查看配置信息

配置 enable 明文口令

全局配置模式：enable  password  *123*

保存交换机的配置

特权： copy  running-config  startup-config

或 write

恢复设备出厂默认值

特权：erase  startup-config

重启：reload

设备配置的准备工作

空闲一段时间后，重回初始界面的问题

switch(config)#line con 0

switch(config-line)#exec-timeout 0 0

配置输出日志同步

Switch(config)#line console 0

Switch(config-line)#logging synchronous

禁用 DNS 查询

switch(config)#no ip domain-lookup

一、 数据链路层

MAC 地址

用来识别一个以太网上的某个单独的设备或一组设备

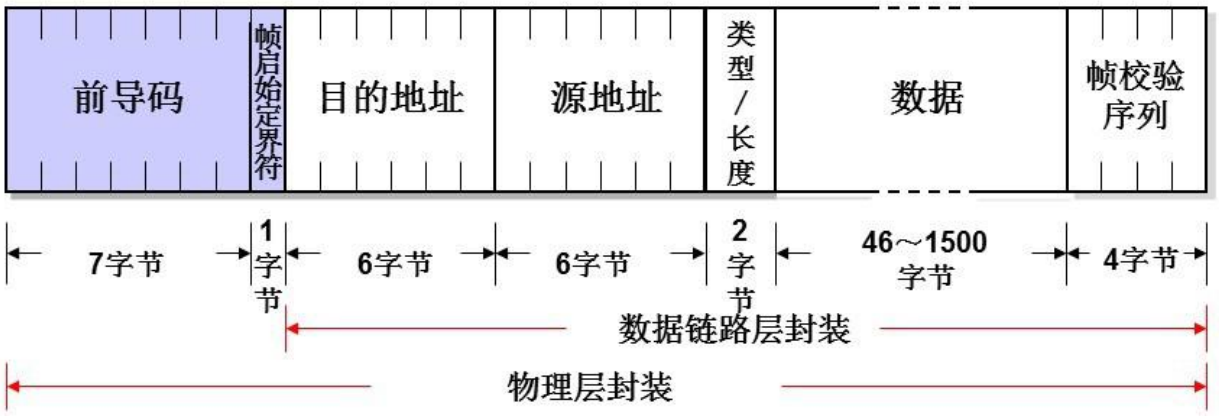
MAC 地址长度 48 位(6 个字节)，前 24 位代表厂商，后 24 位代表网卡编号，

MAC 地址的第 8 位为 0 时表示该 MAC 地址为单播地址，为 1 时表示组播地址。48 位都为 1 表示广播地址。

Ipconfig  /all 查看 MAC 地址

注：一块物理网卡的地址一定是一个单播地址，也就是第 8 位一定为 0

数据链路层的帧格式




帧校验序列（FCS）：从目的地址开始到数据结束这部分的校验和。

类型/长度：用来标识上层协议的类型或数据的字节长度。

交换机

1、交换机是用来连接局域网的主要设备， 交换机分割冲突域，实现全双工通信

2、交换机的工作原理

学习，广播，转发，更新

3、交换机以太网接口双工模式

单工：两个数据站之间只能沿单一方向传输数据

半双工：两个数据站之间可以双向数据传输，但不能同时进行

全双工：两个数据站之间可双向且同时进行数据传输

4、冲突与广播域

广播域指接收同样广播消息的节点的集合

交换机分割冲突域，但是不分割广播域，即交换机的所有端口属于同一个广播域

三、交换机的基本配置

1、常用命令

1）查看 MAC 地址表

特权：show mac-address-table

一、 什么是 VLAN

虚拟局域网

二、 VLAN 的优势

广播控制、安全性、带宽利用、延迟

创建 VLAN 的方法

在全局配置模式中：vlan  2（创建 vlan2）

Name  名字（给 vlan2 命名）

接口加入 vlan

1）进入将要加入 vlan 的接口然后输入

switchport  access  vlan  3

2)、同时将多个接口加入 vlan 2

全局： interface  range  f0/1 - 10

switchport  access  vlan  2

5、查看 vlan 信息

特权：show  vlan

五、trunk 中继链接

1、作用：实现交换机之间的单一链路传递多个 vlan 的信息

2、链路类型：

接入链路(access)： 可以承载 1 个 vlan

中继链路(trunk)：可以承载多个 vlan

3、vlan 的标识

1）ISL(cisco 私有的标记方法)

2）IEEE 802.1q(公有的标记方法)

4、trunk 的配置

接口模式：switchport mode trunk(配置为中继链路)

5、在 trunk 链路上移除某 vlan

进入 trunk 接口：switchport trunk allowed vlan remove 3

6、在 trunk 链路上 添加某 vlan

进入 trunk 接口：switchport trunk allowed vlan add 3

7、查看接口模式

特权模式：show interface f0/5 switchport

六、EthernetChannel（以太网通道）

1、功能：多条线路负载均衡，带宽提高

容错，当一条线路失效时，其他线路通信，不会丢包

2、以太网通道的配置：

全局：interface range f0/6 - 8

switchport mode trunk

channel-group 1 mode on

查看以太网通道的配置： show etherchannel summary

路由器原理及静态路由

1、路由

跨越从源主机到目标主机的一个互联网络来转发数据包的过程

2、路由表

路由器根据路由表做路径选择

3、路由表的获得

1)、直连路由：配置 IP 地址，端口 UP 状态，形成直连路由。

2)、非直连网段：需要静态路由或动态路由，将网段添加到路由表中。

4、静态路由

1)、特点：

由管理员手工配置的，是单向的，因此需要在两个网络之间的边缘路由器上需要双方对指，否则就会造成流量有去无回，缺乏灵活性，适用于小型网络。

2)、配置

全局模式：

ip route 目标网络 ID 子网掩码 下一跳 IP

缺省路由（默认路由）

缺省路由是一种特殊的静态路由

简单地说,缺省路由就是在没有找到任何匹配的具体路由条目的情况下才使用的路由，适用于只有一个出口的末节网络（比如企业的网关出口），优先级最低，可以做为其他路由的补充。

全局:ip  route  0.0.0.0  0.0.0.0  下一跳

代表任意网络 ID  代表任意子网掩码

查看路由表

特权：show  ip  route

C 直连路由

S 静态路由

S*默认路由

==========

三层交换技术

1、作用

使用三层交换技术实现 VLAN 间通信

三层交换=二层交换+三层转发

2、虚拟接口（SVI）

三层交换机上配置的 VLAN 接口为虚接口

3、三层交换机的配置

1)、在三层交换机启用路由功能

全局：ip  routing

2)、配置虚拟接口的 IP 地址

全局：interface  vlan  *2*

ip  address  *192.168.2.254  255.255.255.0*

no  shutdown

3）在三层交换机上配置 Trunk 并指定接口封装为 802.1q

接口模式：switchport  trunk  encapsulation  dot1q

switchport  mode  trunk

4)、配置路由接口

进入接口：no  switchport

==========

动态路由

1、动态路由特点

根据网络拓扑或流量变化，由路由器通过路由协议自动设置，减少

了管理任务，但占用了网络带宽

适合 ISP 服务商、广域网、园区网等大型网络

==========

OSPF 协议

- Open Shortest Path First （ 开放式最短路径优先）
- OSPF 区域
- 为了适应大型的网络，OSPF 在 AS 内划分多个区域
- 每个 OSPF 路由器只维护所在区域的完整链路状态信息
- 区域 ID
- 区域 ID 可以表示成一个十进制的数字
- 也可以表示成一个 IP
- 骨干区域 Area 0
- 负责区域间路由信息传播

启动 OSPF 路由进程

Router(config)# router ospf *process-id*

指定 OSPF 协议运行的接口和所在的区域

Router(config-router)# network *address inverse-mask* area

*area-id*

TCP 和 UDP 协议

1、TCP

传输控制协议

可靠的、面向连接的协议

传输效率低

2、UDP

用户数据报协议

不可靠的、无连接的服务

传输效率高

2、　TCP 的三次握手与四次断开

TCP 的应用

| 端口 | 协议 | 说　　明 |
| --- | --- | --- |
| 21 | FTP | FTP 服务器所开放的控制端口 |
| 23 | TELNET | 用于远程登录，可以远程控制管理目标计算机 |
| 25 | SMTP | SMTP 服务器开放的端口，用于发送邮件 |
| 80 | HTTP | 超文本传输协议 |

| | | |
|---|---|---|
| 53 | DNS | 域名服务，当用户输入网站的名称后，由 DNS 负责将它解析成 IP 地址，这个过程中用到的端口号是 53 |

三、UDP

1、UDP 首部格式

| | |
|---|---|
| 源端口号（16） | 目标端口号（16） |
| UDP 长度（16） | UDP 校验和（16） |

UDP 长度：用来指出 UDP 的总长度

校验和：用来完成对 UDP 数据的差错检验，它是 UDP 协议提供的唯一的可靠机制

2、UDP 端口及应用

| 端口 | 协议 | 说明 |
|---|---|---|
| 69 | TFTP | 简单文件传输协议 |
| 123 | NTP | 网络时间协议 |
| 53 | DNS | 域名服务 |

3、UDP 的流控和差错控制

UDP 缺乏可靠机制

UDP 只有校验和来提供差错控制

需要上层协议来提供差错控制：例如 TFTP 协议

访问控制列表概述

1、访问控制列表（ACL）：

读取第三层、第四层包头信息

根据预先定义好的规则对包进行过滤

2、 访问控制列表的处理过程

如果匹配第一条规则，则不再往下检查，路由器将决定该数据包允许通过或拒绝通过。

如果不匹配第一条规则，则依次往下检查，直到有任何一条规则匹配。

如果最后没有任何一条规则匹配，则路由器根据默认的规则将丢弃该数据包。

3、访问控制列表的类型：

1） 标准访问控制列表

基于源 IP 地址过滤数据包

列表号是 1～99

2） 扩展访问控制列表

基于源 IP 地址、目的 IP 地址、指定协议、端口等来过滤数据包

列表号是 100～199

二、标准访问控制列表

1、标准访问控制列表的创建

全局：access-list 1 deny 192.168.1.1 0.0.0.0

全局：access-list 1 permit 192.168.1.0 0.0.0.255

通配符掩码：也叫做反码。用二进制数 0 和 1 表示，如果某位为 1，表明这一位不需要进行匹配操作，如果为 0 表明需要严格匹配。

隐含拒绝语句：

access-list 1 deny 0.0.0.0 255.255.255.255

2、 将 ACL 应用于接口

接口模式：ip access-group 列表号 in 或 out

注：access-list 1 deny 192.168.1.1 0.0.0.0 或写为

access-list 1 deny host 192.168.1.1

access-list 1 deny 0.0.0.0 255.255.255.255 或写为

access-list 1 deny any

3、 删除已建立的访问控制列表

全局：no access-list 列表号

4、 接口上取消 ACL

接口模式：no ip access-group 列表号 in 或 out

5、 查看访问控制列表

特权：show access-lists

5、删除 ACL

全局：no access-list 列表号

注：不能删除单条 ACL 语句，只能删除整个 ACL。

一、 NAT（网络地址转换）

1、 作用：通过将内部网络的私有 IP 地址翻译成全球唯一的公网 IP 地址，使内部网络可以连接到互联网等外部网络上。

2、 优点：

节省公有合法 IP 地址

处理地址重叠

安全性

3、NAT 的缺点

延迟增大

配置和维护的复杂性

4、NAT 实现方式

1）　　静态转换

IP 地址的对应关系是一对一，而且是不变的，借助静态转换，能实现外部网络对内部网络中某些特设定服务器的访问。

静态 NAT 配置：

配置接口 IP 及路由

全局：

Ip nat inside source static 192.168.1.1 61.159.62.131

在内外接口上启用 NAT：

出口配置：ip　nat　outside

入口配置：ip　nat　inside

2）　端口多路复用（PAT）

通过改变外出数据包的源 IP 地址和源端口并进行端口转换，内部网络的所有主机均可共享一个合法 IP 地址实现互联网的访问，节约 IP。

PAT 的配置：

全局：ip nat inside source list 1 interface f0/1 overload

5、NAT 两种实现方式的区别：

静态转换的对应关系一对一且不变，并且没有节约公用 IP，只隐藏了主机的真实地址。

端口多路复用可以使所有内部网络主机共享一个合法的外部 IP 地址，从而最大限度地节约 IP 地址资源。

开启 nat 排错功能

Router#debug  ip  nat

S 表示源地址

D 表示目的地址

192.168.1.2->61.159.62.130 表示将 192.168.1.2 转换为 61.159.62.130

关闭 nat 排错功能

Router#undebug  ip  nat

STP 简介

STP — Spanning Tree Protocol(生成树协议)

逻辑上断开环路，防止广播风暴的产生

当线路故障，阻塞接口被激活，恢复通信，起备份线路的作用

选择根网桥

选择交换网络中网桥 ID 最小的交换机成为根网桥，网桥 ID 是一个八字的字段，前两个字节十进制数为网桥优先级，后六个字是网桥的 MAC 地址，优先级小的被选择为根网桥，如优先级相同则 MAC 地址小的为根网桥。

网桥优先级的取值范围 0-65535 默认值为 32768

查看交换机 mac 地址

Switch#show version

Base ethernet MAC Address : 0001.9751.0467

VLAN 与 STP（生成树）之间的关系：

PVST+（增强的每 vlan 生成树）

PVST+配置的意义

配置网络中比较稳定的交换机为根网桥

利用 PVST+实现网络的负载分担

四、PVST+的配置命令

1、启用生成树命令（此命令可以不用输入，默认交换机会开启）

全局：spanning-tree  vlan  2

2、指定根网桥（主根或次根）

改优先级

全局：spanning-tree vlan *1* priority *优先级的值*

注意： 优先级的值是 4096 的倍数；

或者在全局模式（推荐此方式）：spanning-tree vlan *2* root primary

spanning-tree vlan *2* root secondary

3、查看某个 vlan 的生成树的配置

特权：show spanning-tree vlan 1

Root ID Priority 20481 表示根网桥的优先级

Bridge ID Priority 24577 表示当前设备的优先级

BLK 表示阻塞接口

FWD 表示转发接口

==========

一、热备份路由选择协议（HSRP）

1、 作用：Cisco 私有协议 ，确保了当网络边缘设备或接入链路出现故障时，用户通信能迅速并透明地恢复，以此为 IP 网络提供冗

余性。通过使用同一个虚拟 IP 地址和虚拟 MAC 地址，LAN 网段上的两台或者多台路由器可以作为一台虚拟路由器对外提供服务。HSRP 使组内的 cisco 路由器能互相监视对方的运行状态。（Cisco 私有协议）

2、HSRP 组成员

活跃路由器、备份路由器、虚拟路由器（即该 lan 上的网关）、其他路由器

HSRP 的配置

1、配置为 HSRP 的成员

进入路由器的网关接口

standby 1 ip 虚拟网关 IP

2、配置 HSRP 的优先级

standby *1* priority *优先级*

*优先级范围 0-255，默认为 100*

3、查看 HSRP 摘要信息

特权： show standby brief

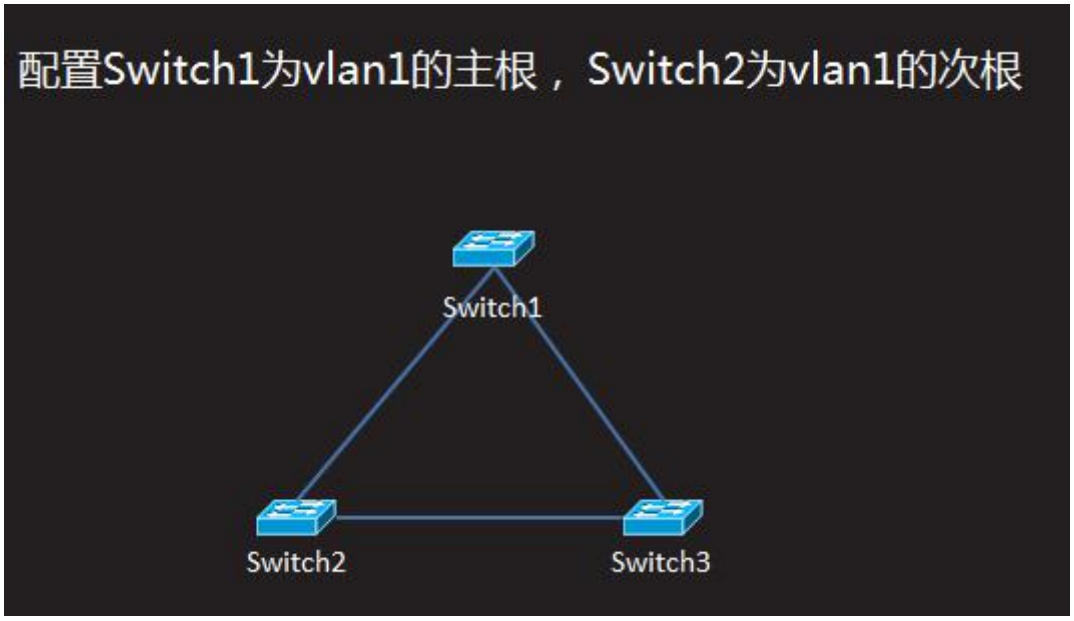
4、HSRP 端口跟踪

standby 1 track f0/1

5、HSRP 占先权

standby *1* preempt

练习 1




1，在 Switch1 中配置

Switch(config)#spanning-tree vlan 1 priority 24576

或

Switch(config)#spanning-tree vlan 1 root primary

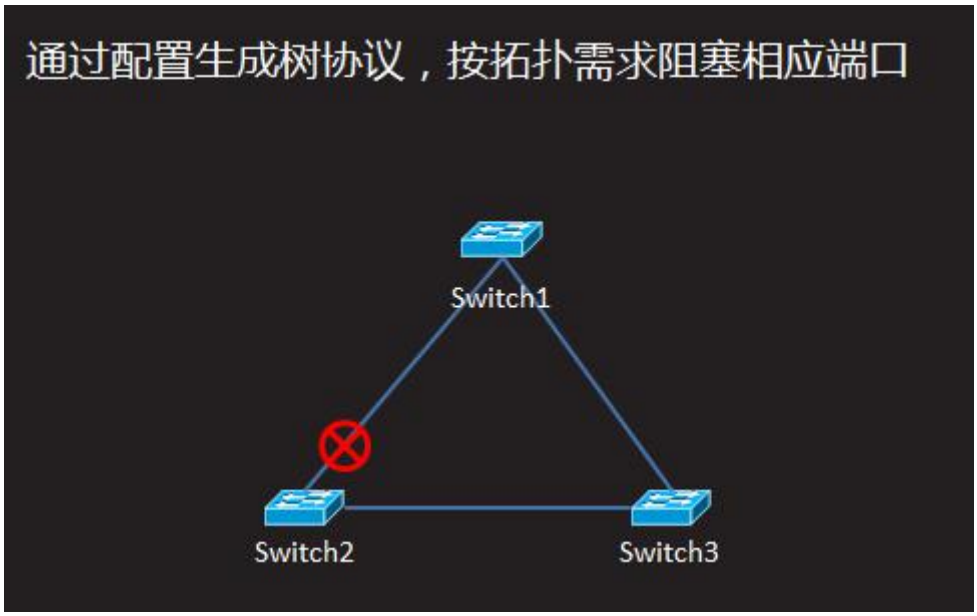
2，在 Switch2 中配置

Switch(config)#spanning-tree vlan 1 priority 28672

或

Switch(config)#spanning-tree vlan 1 root secondary

练习 2



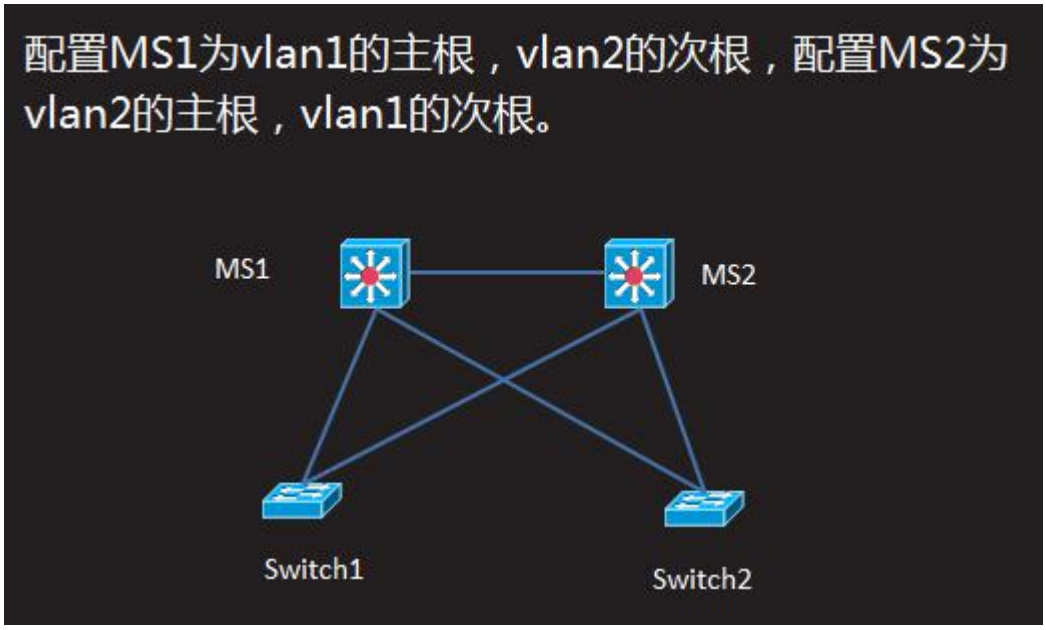

1，在 Switch3 中配置

Switch(config)#spanning-tree vlan 1 root primary

2，在 Switch1 中配置

Switch(config)#spanning-tree vlan 1 root secondary

**练习 3**




1，在所有交换机中创建 vlan2

Switch(config)#vlan 2

2，将拓扑中所有交换机之间都配置为中继链路

MS1

Switch(config)#interface range fastEthernet 0/1-3

Switch(config-if-range)#switchport trunk encapsulation dot1q

Switch(config-if-range)#switchport mode trunk

MS2

Switch(config)#interface range fastEthernet 0/1-3

Switch(config-if-range)#switchport trunk encapsulation dot1q

Switch(config-if-range)#switchport mode trunk

Switch1

Switch(config)#interface range fastEthernet 0/1-2

Switch(config-if-range)#switchport mode trunk

Switch2

Switch(config)#interface range fastEthernet 0/1-2

Switch(config-if-range)#switchport mode trunk

3，在 MS1 中配置

Switch(config)#spanning-tree vlan 1 root primary

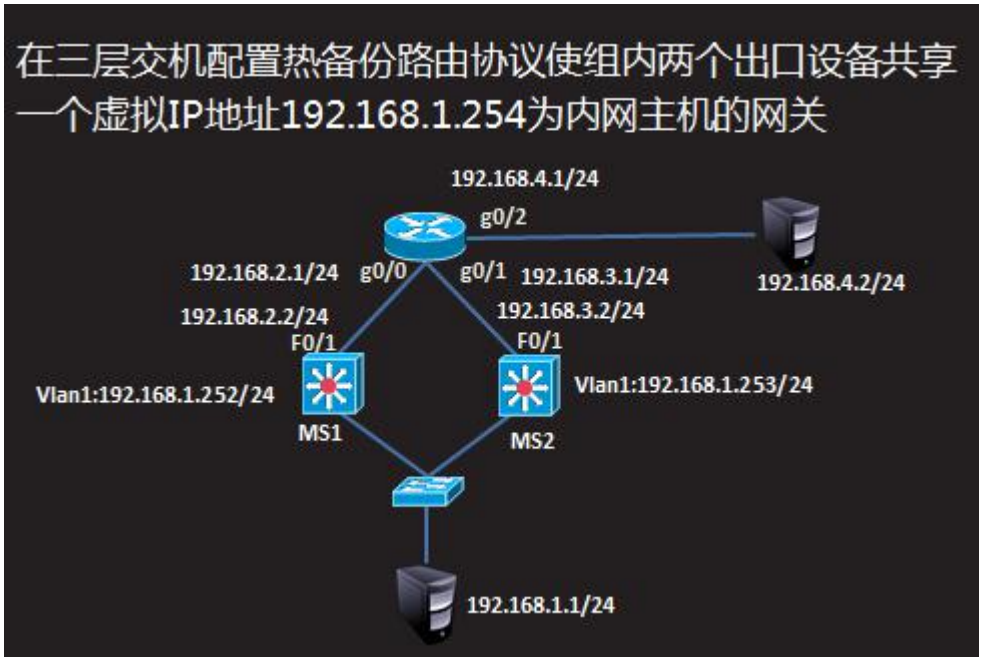
Switch(config)#spanning-tree vlan 2 root secondary

4，在 MS2 中配置

Switch(config)#spanning-tree vlan 2 root primary

Switch(config)#spanning-tree vlan 1 root secondary

**练习 4**




本实验暂不考虑 NAT 问题。

1，为所有 pc 设备配置 ip 与网关，内网主机网关为 192.168.1.254

外网主机网关为 192.168.4.1

2，为所有网络设备配置接口的 ip 地址

路由器

Router(config)#interface gigabitEthernet 0/0

Router(config-if)#ip address 192.168.2.1 255.255.255.0

Router(config-if)#no shutdown

Router(config)#interface gigabitEthernet 0/1

Router(config-if)#ip address 192.168.3.1 255.255.255.0

Router(config-if)#no shutdown

Router(config)#interface gigabitEthernet 0/2

```
Router(config-if)#ip address 192.168.4.1 255.255.255.0
Router(config-if)#no shutdown
```

MS1

```
Switch(config)#interface fastEthernet 0/1
Switch(config-if)#no switchport
Switch(config-if)#ip address 192.168.2.2 255.255.255.0
Switch(config)#interface vlan 1
Switch(config-if)#ip address 192.168.1.252 255.255.255.0
Switch(config-if)#no shutdown
```

MS2

```
Switch(config)#interface fastEthernet 0/1
Switch(config-if)#no switchport
Switch(config-if)#ip address 192.168.3.2 255.255.255.0
Switch(config)#interface vlan 1
Switch(config-if)#ip address 192.168.1.253 255.255.255.0
Switch(config-if)#no shutdown
```

3，配置动态路由技术使全网互通

路由器

```
Router(config)#router ospf 1
Router(config-router)#network 192.168.4.0 0.0.0.255 area 0
Router(config-router)#network 192.168.3.0 0.0.0.255 area 0
Router(config-router)#network 192.168.2.0 0.0.0.255 area 0
```

MS1

```
Switch(config)#ip routing
Switch(config)#router ospf 1
Switch(config-router)#network 192.168.1.0 0.0.0.255 area 0
Switch(config-router)#network 192.168.2.0 0.0.0.255 area 0
```

MS2

```
Switch(config)#ip routing
Switch(config)#router ospf 1
Switch(config-router)#network 192.168.1.0 0.0.0.255 area 0
Switch(config-router)#network 192.168.3.0 0.0.0.255 area 0
```

4，在两台三层交换机中开启热备份功能，使用 192.168.1.254 作为虚拟路由器的地址，并由 MS1 承担活跃路由器。

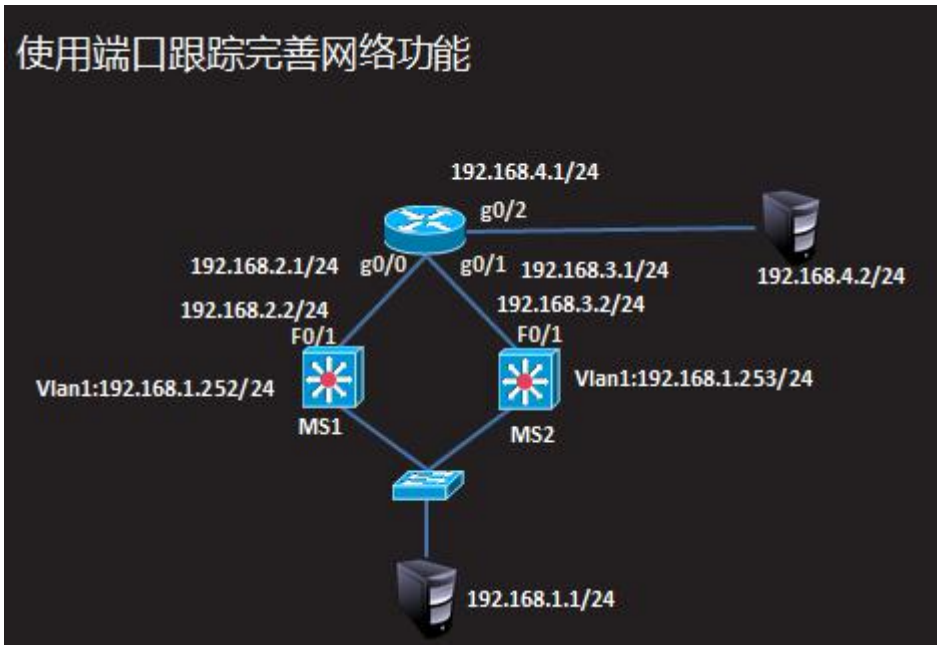
MS1

```
Switch(config)#interface vlan 1
Switch(config-if)#standby 1 ip 192.168.1.254
Switch(config-if)#standby 1 priority 105
```

MS2

```
Switch(config)#interface vlan 1
Switch(config-if)#standby 1 ip 192.168.1.254
```

**练习 5**




为三层交换机设置端口跟踪与占先权

MS1

Switch(config)#interface vlan 1

Switch(config-if)#standby 1 track fastEthernet 0/1

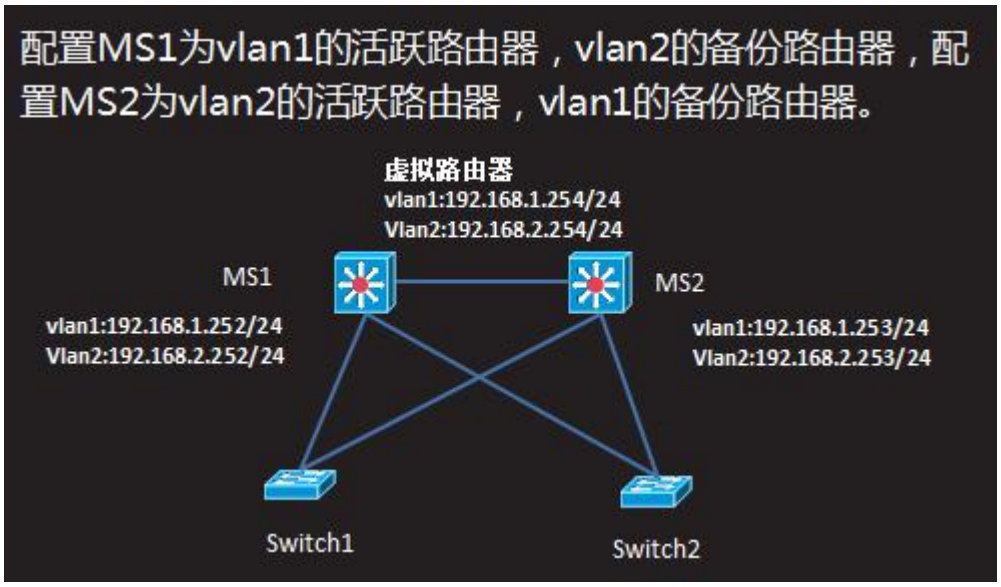
Switch(config-if)#standby 1 preempt

MS2

Switch(config)#interface vlan 1

Switch(config-if)#standby 1 preempt

**练习 6**




注意：此实验需要在 **练习 3** 的基础之上进行配置

1，先配置两台三层交换机的 ip 地址

MS1

```
Switch(config)#interface vlan 1
Switch(config-if)#ip address 192.168.1.252 255.255.255.0
Switch(config-if)#no shutdown
Switch(config)#interface vlan 2
Switch(config-if)#ip address 192.168.2.252 255.255.255.0
```

MS2

```
Switch(config)#interface vlan 1
Switch(config-if)#ip address 192.168.1.253 255.255.255.0
Switch(config-if)#no shutdown
```

```
Switch(config)#interface vlan 2

Switch(config-if)#ip address 192.168.2.253 255.255.255.0
```

2，开启热备份功能

MS1

```
Switch(config)#interface vlan 1

Switch(config-if)#standby 1 ip 192.168.1.254

Switch(config-if)#standby 1 priority 105

Switch(config-if)#standby 1 preempt


Switch(config)#interface vlan 2

Switch(config-if)#standby 2 ip 192.168.2.254

Switch(config-if)#standby 2 preempt
```

MS2

```
Switch(config)#interface vlan 1

Switch(config-if)#standby 1 ip 192.168.1.254

Switch(config-if)#standby 1 preempt
```

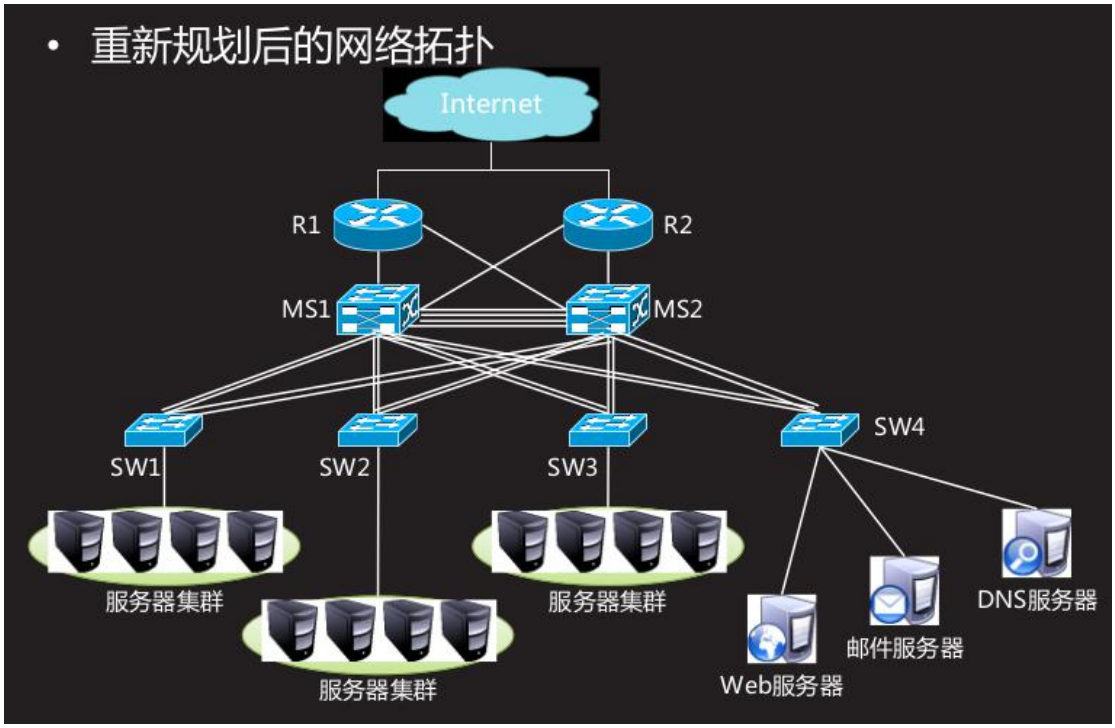
Switch(config)#interface vlan 2

Switch(config-if)#standby 2 ip 192.168.2.254

Switch(config-if)#standby 2 priority 105

Switch(config-if)#standby 2 preempt

综合项目




二层交换机

分别创建 VLAN10、20、30、40

sw1 将 f0/5 接口加入 vlan10

Switch(config)#interface fastEthernet 0/5

Switch(config-if)#switchport access vlan 10

sw2 将 f0/5 接口加入 vlan20

Switch(config)#interface fastEthernet 0/5

Switch(config-if)#switchport access vlan 20

sw3 将 f0/5 接口加入 vlan30

Switch(config)#interface fastEthernet 0/5

Switch(config-if)#switchport access vlan 30

sw4 将 f0/5 接口加入 vlan40

Switch(config)#interface fastEthernet 0/5

Switch(config-if)#switchport access vlan 40

每台设备捆绑以太通道,将 f0/1 与 f0/2 捆绑为通道 1,f0/3 与 f0/4 捆绑为通道 2

Switch(config)#interface range f0/1-2

Switch(config-if-range)#channel-group 1 mode on

Switch(config)#interface range f0/3-4

Switch(config-if-range)#channel-group 1 mode on

查看以太通道汇总信息

Switch#show etherchannel summary

依次进入所有二层交换机的以太通道接口，配置中继链路

Switch(config)#interface port-channel 1

Switch(config-if)#switchport mode trunk

Switch(config)#interface port-channel 2

Switch(config-if)#switchport mode trunk

==========

三层交换机

每台设备分别创建 VLAN10、20、30、40

1-2 口捆绑为通道 1

3-4 口捆绑为通道 2

5-6 口捆绑为通道 3

7-8 口捆绑为通道 4

9-10 口捆绑为通道 5

依次进入三层交换机的 4 个通道接口，配置中继链路（两台三层交换机配置相同）

```
Switch(config)#interface port-channel 1
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)#switchport mode trunk

Switch(config)#interface port-channel 2
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)#switchport mode trunk

Switch(config)#interface port-channel 3
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)#switchport mode trunk

Switch(config)#interface port-channel 4
Switch(config-if)# switchport trunk encapsulation dot1q
```

Switch(config-if)#switchport mode trunk

Switch(config)#interface port-channel 5

Switch(config-if)# switchport trunk encapsulation dot1q

Switch(config-if)#switchport mode trunk

配置三层交换机 vlan10、20、30、40 的 ip 地址

Switch(config)#interface vlan 10

S  witch(config-if)#ip address 192.168.10.252 255.255.255.0

Switch(config)#interface vlan 20

Switch(config-if)#ip address 192.168.20.252 255.255.255.0

Switch(config)#interface vlan 30

Switch(config-if)#ip address 192.168.30.252 255.255.255.0

Switch(config)#interface vlan 40

Switch(config-if)#ip address 192.168.40.252 255.255.255.0

注意：另外一台三层交换机配置的 ip 地址是 253

Switch(config)#interface vlan 10

Switch(config-if)#ip address 192.168.10.253 255.255.255.0

Switch(config)#interface vlan 20

Switch(config-if)#ip address 192.168.20.253 255.255.255.0

Switch(config)#interface vlan 30

Switch(config-if)#ip address 192.168.30.253 255.255.255.0

Switch(config)#interface vlan 40

Switch(config-if)#ip address 192.168.40.253 255.255.255.0

==========

=

配置生成树协议，产生负载均衡效果。

MS1 配置 PVST+  使其成为 vlan10、20 的主根 vlan30、40 的次根

Switch(config)#spanning-tree vlan 10 root primary

Switch(config)#spanning-tree vlan 20 root primary

Switch(config)#spanning-tree vlan 30 root secondary

Switch(config)#spanning-tree vlan 40 root secondary

MS2 配置 PVST+  使其成为 vlan30、40 的主根 vlan10、20 的次根

Switch(config)#spanning-tree vlan 30 root primary

Switch(config)#spanning-tree vlan 40 root primary

Switch(config)#spanning-tree vlan 10 root secondary

Switch(config)#spanning-tree vlan 20 root secondary

配置热备份路由协议，完善负载均衡效果。

MS1 配置 HSRP  使其成为 vlan10、20 的活跃路由器 vlan30、40 的备份路由器

Switch(config)#interface vlan 10

Switch(config-if)#standby 10 ip 192.168.10.254

Switch(config-if)#standby 10 priority 105

Switch(config-if)#standby 10 preempt

Switch(config)#interface vlan 20

Switch(config-if)#standby 20 ip 192.168.20.254

Switch(config-if)#standby 20 priority 105

Switch(config-if)#standby 20 preempt

Switch(config)#interface vlan 30

Switch(config-if)#standby 30 ip 192.168.30.254

Switch(config)#interface vlan 40

Switch(config-if)#standby 40 ip 192.168.40.254

查看热备份状态

Switch#show standby brief

MS2 配置 HSRP  使其成为 vlan30、40 的活跃路由器 vlan10、20 的备份路由器

Switch(config)#interface vlan 30

Switch(config-if)#standby 30 ip 192.168.30.254 综合项目

Switch(config-if)#standby 30 priority 105

Switch(config-if)#standby 30 preempt

Switch(config)#interface vlan 40

Switch(config-if)#standby 40 ip 192.168.40.254

Switch(config-if)#standby 40 priority 105

Switch(config-if)#standby 40 preempt

Switch(config)#interface vlan 10

Switch(config-if)#standby 10 ip 192.168.10.254

Switch(config)#interface vlan 20

Switch(config-if)#standby 20 ip 192.168.20.254

开启两台三层交换机的路由功能，并设置每个服务器所在 vlan 的网关

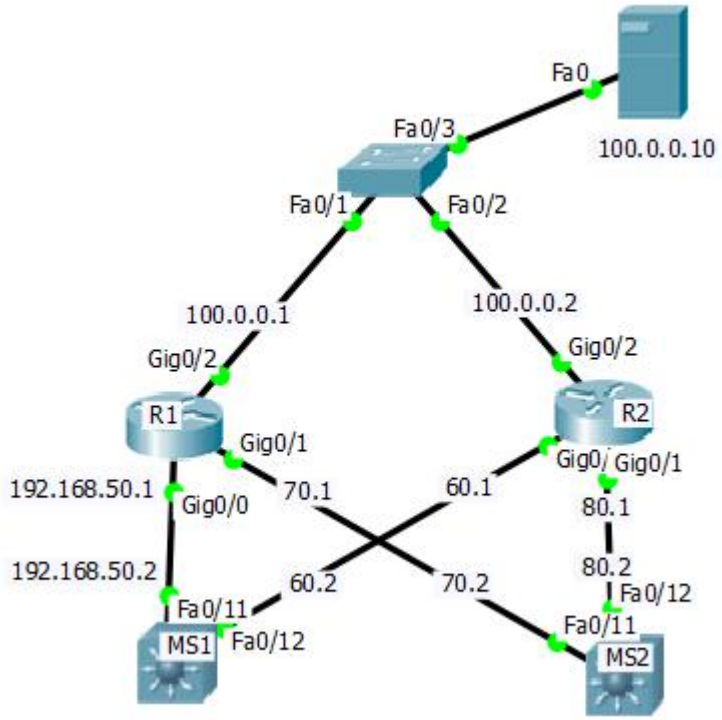
Switch(config)#ip routing

然后测试目前网络是否可以达成全网互通。

==========

==

按图为路由器与三层交换机相连的接口配置 ip




综合项目

配置动态路由协议，使所有内网互通。

在 ms1 中开启 ospf 动态路由，并宣告直连网段

Switch(config)#router ospf 1

Switch(config-router)#network 192.168.10.0 0.0.0.255 area 0

Switch(config-router)#network 192.168.20.0 0.0.0.255 area 0

Switch(config-router)#network 192.168.30.0 0.0.0.255 area 0

Switch(config-router)#network 192.168.40.0 0.0.0.255 area 0

Switch(config-router)#network 192.168.50.0 0.0.0.255 area 0

Switch(config-router)#network 192.168.60.0 0.0.0.255 area 0

在 ms2 中开启 ospf 动态路由，并宣告直连网段

Switch(config)#router ospf 1 综合项目

Switch(config-router)#network 192.168.10.0 0.0.0.255 area 0

Switch(config-router)#network 192.168.20.0 0.0.0.255 area 0

Switch(config-router)#network 192.168.30.0 0.0.0.255 area 0

Switch(config-router)#network 192.168.40.0 0.0.0.255 area 0

Switch(config-router)#network 192.168.70.0 0.0.0.255 area 0

Switch(config-router)#network 192.168.80.0 0.0.0.255 area 0

在 r1 中开启 ospf 动态路由，并宣告直连网段

Router(config)#router ospf 1

Router(config-router)#network 192.168.50.0 0.0.0.255 area 0

Router(config-router)#network 192.168.70.0 0.0.0.255 area 0

在 r2 中开启 ospf 动态路由，并宣告直连网段

Router(config)#router ospf 1

Router(config-router)#network 192.168.60.0 0.0.0.255 area 0

Router(config-router)#network 192.168.80.0 0.0.0.255 area 0

查看所有三层设备路由表，应该是统一状态

show ip route

配置 r1 与 r2 的 nat 功能,使内网服务器 40.1 映射到外网 100.0.0.3，并在接口中开启

Router(config)#ip nat inside source static 192.168.40.1 100.0.0.3

Router(config)#in g0/2

Router(config-if)#ip nat outside

Router(config-if)#in range g0/0-1

Router(config-if-range)#ip nat inside

在 r1 与 r2 中配置默认路由之后，使用 ospf 宣告自己是默认信息源（表示自己有通往外网的默认路由）

Router(config)#ip route 0.0.0.0 0.0.0.0 100.0.0.10

Router(config)#router ospf 1

Router(config-router)#default-information originate

验证从外网可以访问内网的 web 服务。