[第一阶段笔记]

Linux

目录

[一、 TCP/IP协议](#_Toc21139_WPSOffice_Level1) [1](#_Toc21139_WPSOffice_Level1)

[i 键](#_Toc7464_WPSOffice_Level1) [3](#_Toc7464_WPSOffice_Level1)

[一、永久配置主机名 /etc/hostname](#_Toc10398_WPSOffice_Level1) [4](#_Toc10398_WPSOffice_Level1)

[二、配置IP地址](#_Toc1129_WPSOffice_Level1) [4](#_Toc1129_WPSOffice_Level1)

[三、DNS服务器配置](#_Toc25265_WPSOffice_Level1) [5](#_Toc25265_WPSOffice_Level1)

[NTP时间同步](#_Toc28241_WPSOffice_Level1) [8](#_Toc28241_WPSOffice_Level1)

[1.安装客户端同步时间的软件chrony](#_Toc9659_WPSOffice_Level1) [8](#_Toc9659_WPSOffice_Level1)

[2.修改配置文件指定服务端位置](#_Toc18322_WPSOffice_Level1) [8](#_Toc18322_WPSOffice_Level1)

[3.重起服务](#_Toc17662_WPSOffice_Level1) [8](#_Toc17662_WPSOffice_Level1)

[4.验证：](#_Toc12887_WPSOffice_Level1) [8](#_Toc12887_WPSOffice_Level1)

[权限和归属](#_Toc20003_WPSOffice_Level1) [9](#_Toc20003_WPSOffice_Level1)

[r: cat less head tail grep](#_Toc20776_WPSOffice_Level1) [9](#_Toc20776_WPSOffice_Level1)

[w: vim 保存](#_Toc32527_WPSOffice_Level1) [9](#_Toc32527_WPSOffice_Level1)

[x: 可以执行](#_Toc25064_WPSOffice_Level1) [9](#_Toc25064_WPSOffice_Level1)

[acl访问控制列表](#_Toc32609_WPSOffice_Level1) [10](#_Toc32609_WPSOffice_Level1)

[分区规划](#_Toc29149_WPSOffice_Level1) [12](#_Toc29149_WPSOffice_Level1)

[一、分区规划](#_Toc8928_WPSOffice_Level1) [12](#_Toc8928_WPSOffice_Level1)

[一、制作逻辑卷](#_Toc31458_WPSOffice_Level1) [13](#_Toc31458_WPSOffice_Level1)

[逻辑卷的扩大,支持线上操作](#_Toc25702_WPSOffice_Level1) [14](#_Toc25702_WPSOffice_Level1)

[一、卷组有足够的剩余空间](#_Toc31536_WPSOffice_Level1) [14](#_Toc31536_WPSOffice_Level1)

[二、卷组没有足够的剩余空间](#_Toc1284_WPSOffice_Level1) [14](#_Toc1284_WPSOffice_Level1)

[ipv6地址配置](#_Toc25695_WPSOffice_Level1) [15](#_Toc25695_WPSOffice_Level1)

[SELinux概述](#_Toc22907_WPSOffice_Level1) [16](#_Toc22907_WPSOffice_Level1)

[配置聚合连接(网卡绑定、链路聚合)](#_Toc25490_WPSOffice_Level1) [17](#_Toc25490_WPSOffice_Level1)

[搭建基本Web服务](#_Toc24634_WPSOffice_Level1) [18](#_Toc24634_WPSOffice_Level1)

[1.安装一个可以提供Web功能软件](#_Toc21487_WPSOffice_Level1) [18](#_Toc21487_WPSOffice_Level1)

[2.启动httpd服务](#_Toc19578_WPSOffice_Level1) [18](#_Toc19578_WPSOffice_Level1)

[3.书写自己的页面文件](#_Toc30151_WPSOffice_Level1) [18](#_Toc30151_WPSOffice_Level1)

[搭建基本FTP服务(文件传输)](#_Toc22336_WPSOffice_Level1) [18](#_Toc22336_WPSOffice_Level1)

[防火墙策略的应用](#_Toc6180_WPSOffice_Level1) [18](#_Toc6180_WPSOffice_Level1)

[iSCSI磁盘的工作模式](#_Toc27966_WPSOffice_Level1) [21](#_Toc27966_WPSOffice_Level1)

[一 搭建基本的数据库](#_Toc3984_WPSOffice_Level1) [22](#_Toc3984_WPSOffice_Level1)

[案例5:使用数据库查询](#_Toc18668_WPSOffice_Level1) [23](#_Toc18668_WPSOffice_Level1)

[Web服务器](#_Toc27732_WPSOffice_Level1) [25](#_Toc27732_WPSOffice_Level1)

[一 搭建基本的Web服务](#_Toc16984_WPSOffice_Level1) [25](#_Toc16984_WPSOffice_Level1)

[案例4:使用自定Web根目录](#_Toc13191_WPSOffice_Level1) [28](#_Toc13191_WPSOffice_Level1)

[1)新建目录 /webroot,作为此站点新的网页目录](#_Toc10872_WPSOffice_Level1) [28](#_Toc10872_WPSOffice_Level1)

[搭建虚拟Web主机](#_Toc14746_WPSOffice_Level1) [31](#_Toc14746_WPSOffice_Level1)

[邮件服务](#_Toc30211_WPSOffice_Level1) [32](#_Toc30211_WPSOffice_Level1)

[配置NFS共享](#_Toc29258_WPSOffice_Level1) [37](#_Toc29258_WPSOffice_Level1)

[FTP服务](#_Toc23582_WPSOffice_Level1) [39](#_Toc23582_WPSOffice_Level1)

[案例4:编写一个判断脚本](#_Toc26910_WPSOffice_Level1) [39](#_Toc26910_WPSOffice_Level1)

[1)当运行/root/foo.sh redhat,输出为fedora](#_Toc28507_WPSOffice_Level1) [39](#_Toc28507_WPSOffice_Level1)

[2)当运行/root/foo.sh fedora,输出为redhat](#_Toc28379_WPSOffice_Level1) [39](#_Toc28379_WPSOffice_Level1)

[3)当没有任何参数或者参数不是redhat 或者fedora时,](#_Toc11236_WPSOffice_Level1) [39](#_Toc11236_WPSOffice_Level1)

[防火墙 处理 端口转发 功能](#_Toc11158_WPSOffice_Level1) [41](#_Toc11158_WPSOffice_Level1)

[建SERVICES教学环境](#_Toc3017_WPSOffice_Level1) [42](#_Toc3017_WPSOffice_Level1)

[第一台虚拟机,一般作为服务端 ,虚拟机A](#_Toc583_WPSOffice_Level1) [42](#_Toc583_WPSOffice_Level1)

[zip归档工具,跨平台的压缩工具](#_Toc31418_WPSOffice_Level1) [44](#_Toc31418_WPSOffice_Level1)

[搭建DNS服务器的基本步骤 ——](#_Toc23053_WPSOffice_Level1) [49](#_Toc23053_WPSOffice_Level1)

[第1个httpd进程由root运行，不面向Web用户](#_Toc3266_WPSOffice_Level1) [59](#_Toc3266_WPSOffice_Level1)

[第2个及以后的httpd进程是以apache用户身份运行，面向Web用户](#_Toc17030_WPSOffice_Level1) [59](#_Toc17030_WPSOffice_Level1)

[1）查看当前主机的CPU负载情况](#_Toc14274_WPSOffice_Level1) [59](#_Toc14274_WPSOffice_Level1)

[2）查看当前主机开启了多长时间、有几个用户登录、运行的进程数](#_Toc21475_WPSOffice_Level1) [59](#_Toc21475_WPSOffice_Level1)

[3）查看当前主机的CPU、内存、交换分区空闲比率](#_Toc3505_WPSOffice_Level1) [59](#_Toc3505_WPSOffice_Level1)

[4）观测当前占用CPU资源最多的进程叫什么名](#_Toc28835_WPSOffice_Level1) [59](#_Toc28835_WPSOffice_Level1)

[1）查看用户mike开启的进程名、PID号](#_Toc9711_WPSOffice_Level1) [60](#_Toc9711_WPSOffice_Level1)

[2）统计当前主机运行的进程的数量](#_Toc32278_WPSOffice_Level1) [60](#_Toc32278_WPSOffice_Level1)

[3）检查网站服务进程httpd是否已经开启](#_Toc5681_WPSOffice_Level1) [60](#_Toc5681_WPSOffice_Level1)

[4）找出名字为 gdm 的进程的PID号](#_Toc4637_WPSOffice_Level1) [60](#_Toc4637_WPSOffice_Level1)

[2）控制进程](#_Toc8145_WPSOffice_Level1) [60](#_Toc8145_WPSOffice_Level1)

[1）分离解析的原理](#_Toc18367_WPSOffice_Level1) [61](#_Toc18367_WPSOffice_Level1)

[2）RAID磁盘阵列的各个级别的特点](#_Toc7296_WPSOffice_Level1) [61](#_Toc7296_WPSOffice_Level1)

[3）查找进程的各种方法](#_Toc14732_WPSOffice_Level1) [61](#_Toc14732_WPSOffice_Level1)

[4）杀死进程的各种方法](#_Toc19887_WPSOffice_Level1) [61](#_Toc19887_WPSOffice_Level1)

[1）启用httpd服务](#_Toc22496_WPSOffice_Level1) [62](#_Toc22496_WPSOffice_Level1)

[2）挂载RHEL7的ISO文件到 /var/www/html/rh7dvd](#_Toc3120_WPSOffice_Level1) [62](#_Toc3120_WPSOffice_Level1)

[3）检查，确认软件源可访问](#_Toc17031_WPSOffice_Level1) [62](#_Toc17031_WPSOffice_Level1)

[1）客户机 请求，DHCP DISCOVERY](#_Toc471_WPSOffice_Level1) [62](#_Toc471_WPSOffice_Level1)

[2）服务器 提供，DHCP OFFER](#_Toc18524_WPSOffice_Level1) [62](#_Toc18524_WPSOffice_Level1)

[3）客户机 选择，DHCP REQUEST](#_Toc9504_WPSOffice_Level1) [62](#_Toc9504_WPSOffice_Level1)

[4）服务器 确认，DHCP ACK](#_Toc1222_WPSOffice_Level1) [62](#_Toc1222_WPSOffice_Level1)

[1）装包 dhcp](#_Toc12082_WPSOffice_Level1) [62](#_Toc12082_WPSOffice_Level1)

[2）配置](#_Toc4679_WPSOffice_Level1) [62](#_Toc4679_WPSOffice_Level1)

[3）起服务 dhcpd](#_Toc18091_WPSOffice_Level1) [62](#_Toc18091_WPSOffice_Level1)

[4）确认监听状态](#_Toc17028_WPSOffice_Level1) [63](#_Toc17028_WPSOffice_Level1)

[5）在客户机pc207上测试](#_Toc11803_WPSOffice_Level1) [63](#_Toc11803_WPSOffice_Level1)

[一、 TCP/IP协议](#_Toc3936_WPSOffice_Level1) [67](#_Toc3936_WPSOffice_Level1)

[二、 IP地址](#_Toc31519_WPSOffice_Level1) [67](#_Toc31519_WPSOffice_Level1)

[接口](#_Toc22055_WPSOffice_Level1) [69](#_Toc22055_WPSOffice_Level1)

[1） 双绞线分类：](#_Toc17265_WPSOffice_Level1) [69](#_Toc17265_WPSOffice_Level1)

[2） 双绞线标准与分类 ：](#_Toc13532_WPSOffice_Level1) [69](#_Toc13532_WPSOffice_Level1)

[一、 数据链路层](#_Toc22385_WPSOffice_Level1) [71](#_Toc22385_WPSOffice_Level1)

[三、交换机的基本配置](#_Toc25379_WPSOffice_Level1) [72](#_Toc25379_WPSOffice_Level1)

[一、什么是VLAN](#_Toc7659_WPSOffice_Level1) [72](#_Toc7659_WPSOffice_Level1)

[二、VLAN的优势](#_Toc27494_WPSOffice_Level1) [72](#_Toc27494_WPSOffice_Level1)

[三、trunk中继链接](#_Toc815_WPSOffice_Level1) [72](#_Toc815_WPSOffice_Level1)

[四、EthernetChannel（以太网通道）](#_Toc3612_WPSOffice_Level1) [73](#_Toc3612_WPSOffice_Level1)

[三、路由协议分类](#_Toc15788_WPSOffice_Level1) [75](#_Toc15788_WPSOffice_Level1)

[四、RIP路由协议](#_Toc10339_WPSOffice_Level1) [76](#_Toc10339_WPSOffice_Level1)

[配置DHCP服务](#_Toc31925_WPSOffice_Level1) [77](#_Toc31925_WPSOffice_Level1)

[一、 访问控制列表概述](#_Toc75_WPSOffice_Level1) [77](#_Toc75_WPSOffice_Level1)

[二、标准访问控制列表](#_Toc31256_WPSOffice_Level1) [77](#_Toc31256_WPSOffice_Level1)

[三、扩展访问控制列表](#_Toc11386_WPSOffice_Level1) [78](#_Toc11386_WPSOffice_Level1)

[一、 NAT（网络地址转换）](#_Toc28068_WPSOffice_Level1) [78](#_Toc28068_WPSOffice_Level1)

[二、 查看NAT转换条目](#_Toc14882_WPSOffice_Level1) [79](#_Toc14882_WPSOffice_Level1)

[三、 清除NAT转换条目](#_Toc6629_WPSOffice_Level1) [79](#_Toc6629_WPSOffice_Level1)

[四、 显示每个转换的数据包](#_Toc2900_WPSOffice_Level1) [79](#_Toc2900_WPSOffice_Level1)

**一、 TCP/IP协议**

1、 IP地址的分类

**– A类:1 ~ 126 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**

– **网络位**：代表本ip属于哪一个网络

– **主机位**：在该网络中编号

**子网掩码**：标识 ip地址的网络位与主机位，用1代表网络位 0代表主机位

**什么是Linux?**

一个操作系统：一群软件的集合，让计算机硬件正常工作.

Linux服务器操作系统

• UNIX诞生  **1970-1-1**

• Linux之父,Linus Torwalds

Linux系统内核：软件

– **版本号**:**主版本.次版本.修订号**

• 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) 缓解真实物理内存的压力

Linux目录结构，树型结构

根目录：所有的数据都在此目录下（Linux系统的起点）

**/dev** : 所有设备文件相关的数据都在此目录下

**hd**,表示IDE设备

**sd**,表示SCSI设备

**vd**,表示virtual设备 虚拟化设备

/dev/sda6 :该设备表示的含义？

表示**SCSI**设备第一块硬盘第6个分区

**黑色**：文本文件

**蓝色**：目录

**绿色**：可以执行的程序

**青色**：快捷方式

**Linux命令**

– 命令的执行依赖于解释器(例如:/bin/bash)

默认的解释器：/bin/bash

• h**ostname** 查看及设置主机名

• **lscpu** 列出CPU处理器信息

• **poweroff**关机

• **reboot**重启

• **mkdir**新建目录

• **&** 放入后台

• **cat** 查看文本文件的内容

• **pwd** 查看当前工作目录

• **cd** 切换工作目录

• **uname -r** 查看内核版本

• **cd** – 用途:切换工作目录

绝对路径： 以根目录开始的路径

相对路径： 不以根目录开始的路径，以当前为参照的路径

• **ls** – **-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)

**查看文本文件内容**

**cat** #不适合 查看大文件

# **less** /etc/passwd #适合查看大文件，分屏阅读工具

按 q 退出

**查看文本文件部分内容**

**head** -3 /etc/passwd #显示文本文件内容的前3行

**tail** -4 /etc/passwd #显示文本文件内容的尾部4行

**查看文本文件，包含指定字符串的行**

**grep** '指定字符串' /路径/文本文件

**快速编辑技巧**

• 快捷键

– **Ctrl + l**: 清空整个屏幕

– **Ctrl + c**: 结束正在运行的命令

– **Esc + .** : 粘贴上一个命令的参数

– **Ctrl + u**:清空至行首

– **Ctrl + w**:往回删除一个单词(以空格界定)

**mount挂载操作**

Linux：

光盘文件------->光驱设备-------->访问点(随意的目录)

访问点： 在访问设备资源时

1.通过图形将光盘文件放入光驱设备

2.查看光驱设备文件

3.访问光驱设备内容

4.**卸载**

**注意**：

1.挂载时 挂载点目录（访问点）尽量不要是 根目录 已经存在的目录，自己创建新目录

2.卸载时 当前路径，不能是挂载点目录（访问点）

**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

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

**经典案例：**# ls /dev/tty{2[0-9],30}

• 别名的定义，简化较为复杂的命令

# alias aa='ls -ld' #设置命令的别名

# unalias hn #删除命令的别名

**复制、删除、移动**

**rm** 删除

– **-r、-f**:递归删除(含目录)、强制删除

**mv** 移动/改名

**cp** 复制

– -r:递归,复制目录时必须有此选项

cp：可以在复制时，改变源文件在目标路径中名字

**.** : 代表当前路径

cp：可以加三个以上的参数，将最后一个参数作为目标，其他作为源文件

**vim** 文本编辑器

命令模式 输入模式（插入模式） 末行模式

i 键

命---------------------》输入模式（Esc 键回到命令模式）

令

模

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

: 键

末行模式 ：wq 保存并退出

：q！ 强制不保存并退出

– **rpm -q**  软件名... 查看

– **rpm -ivh** 软件名-版本信息.rpm... 安装

– **rpm -e**  软件名... 卸载

[root@server0 ~]# rpm -ivh /dvd/Packages/vsftpd-3.0.2-22.el7.x86\_64.rpm

不能解决依赖关系！

**Yum软件包仓库配置及使用**

服务：为客户端自动解决依赖关系，并且安装软件

服务端: 虚拟机classroom 搭建完成Web服务 共享光盘所有内容

验证： 浏览器中

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

客户端：虚拟机server

修改配置文件 ， 指定服务端位置

/etc/yum.repos.d/\*.repo

正确的文件与错误的文件会相互影响

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

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

**[rhel7] #仓库标识**

**name=nsd rhel7 #仓库描述信息**

**baseurl=http://classroom.example.com/content/rhel7.0/x86\_64/dvd/ #服务端具体位置路径**

**enabled=1 #是否启用该文件**

**gpgcheck=0 #是否检测软件包签名信息**

[root@server0 ~]# yum repolist #列出仓库信息

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

[root@server0 ~]# yum remove 软件名 #卸载软件

[root@server0 ~]# yum clean all #清缓存

**配置网络地址**

**/etc** : 存放系统各种配置文件

一、永久配置主机名 **/etc/hostname**

二、配置IP地址

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

**nmcli连接管理**

**1.**查看**nmcli**命令识别的网卡名字

[root@A ~]# nmcli connection show

**2.配置ip地址**

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

**ipv4.method manual ipv4.addresses '172.25.0.110/24 172.25.0.254' connection.autoconnect yes**

nmcli 连接网络配置 '网卡识别的名字' ipv4.method 配置ip地址的方法，auto自动获取 manual手工配置 ipv4.addresses 'ip地址/子网掩码 网关地址' connection.autoconnect yes 每次开机自动启用

**3.激活配置**

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

[root@A ~]# **ifconfig**  #查看ip地址

[root@A ~]# **route** #查看网关地址 gateway

successfully : 成功

**三、DNS服务器配置**

作用： 域名解析为对应IP地址

DNS服务器: 虚拟机classroom

配置文件: **/etc/resolv.conf**

虚拟机Server

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

nameserver 172.25.254.254

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

**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...**

• 常用命令选项

– **-v**,取反匹配

– **-i**,忽略大小写

– ^word 以字符串word开头

– word$ 以字符串word结尾

**重定向输出**

> ： 覆盖重定向 将前面命令的输出，写入文本文件中 自带touch功能

>> ： 追加重定向

**管理用户和组**

**用户**：1.可以登陆操作系统 2.可以实现访问控制（不同的用户具备不同权限）

**组** ： 方便对用户管理（将用户加入组）

**唯一标识**： UID 编号 GID 编号

**管理员用户root**： UID 0

**普通用户UID**： 从UID 1000

组 ： 基本组（私有组） 附加组（从属组 公共组）

一个用户至少属于一个组

**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GID:用户描述信息:家目录:解释器程序(shell)**

• 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解释器 、-G 附加组**

[root@server0 ~]# useradd -d /mnt/abc nsd06 #指定家目录创建用户

[root@server0 ~]# useradd -u 1100 nsd02 #指定UID创建用户

[root@server0 ~]# **id** nsd03 #查看用户信息

**/sbin/nologin** : 禁止用户登陆操作系统

[root@server0 ~]# useradd -s /sbin/nologin nsd07

创建组：]# **groupadd** tarena #创建tarena组

**管道操作**： **|**

作用：将前面命令的输出结果，交由后面命令处理，最后输出最后命令的结果

passwd 命令

**非交互式设置密码** ： echo 密码 | passwd --stdin 用户名

用户密码信息存放在 **/etc/shadow** 文件

• **usermod** 命令 修改用户信息

– 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解释器、-G 附加组

# usermod -u 1300 -d /opt/abc -s /sbin/nologin -G root natasha

补充： 可以vim修改/etc/passwd 文件内容

**删除用户**

• **userdel** 命令

– userdel [-r] 用户名 # -r：连同用户家目录一并删除

**管理组账号**

组基本信息存放在 **/etc/group** 文件

[root@server0 ~]# groupadd stugrp

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

**stugrp:x:1002:**

**组名：组的密码占位符：本组的GID编号：本组成员的列表**

使用 **gpasswd** 命令

– **gpasswd -a 用户名 组名** #用户加入组

– **gpasswd -d 用户名 组名** #从组中删除用户

**删除组**

– **groupdel 组名**

**tar备份与恢复**

• 归档的含义

– 将许多零散的文件整理为一个文件

– 文件总的大小基本不变

• 压缩的含义

– 按某种算法减小文件所占用空间的大小

– 恢复时按对应的逆向算法解压

Linux独有的压缩工具：

**gzip** ------》 \*.gz

**bzip2** ------》 \*.bz2

**xz**  ------》 \*.xz

• tar 集成备份工具

**tar 选项 tar包名字 被归档的文件**

[root@server0 /]# tar -cPf /opt/file01.tar /home/ /mnt/

[root@server0 /]# tar -xf /opt/file01.tar -C /nsd/

**-c : 创建归档 -P：保持路径不变进行归档 -f：指定归档包的名字**

**-x：释放归档 -C：指定释放位置 -t : 查看tar包内容**

[root@server0 /]# tar -cPf /root/test.tar /opt/1.txt

[root@server0 /]# tar -tf /root/test.tar #查看tar包里面内容

[root@server0 /]# tar -xPf /root/test.tar #绝对路径释放

– **-z、-j、-J** : 调用 **.gz**、**.bz2**、**.xz** 格式的工具进行处理

[root@server0 ~]# tar -zcPf /opt/nsd.tar.gz /home/

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

[root@server0 ~]# tar -jcPf /opt/abc.tar.bz2 /home/

[root@server0 ~]# tar -JcPf /opt/home.tar.xz /home/

**查看日期时间**

- **date**

date **+%F** #显示年-月-日 date **+%Y** #显示年

date **+%m** #显示月 date **+%d** #显示当天是多少号

date **+%H** #显示时 date **+%M**  #显示分

date  **-s**  "年-月-日 时：分：秒"

**NTP时间同步**

NTP网络时间协议

1.安装客户端同步时间的软件**chrony**

2.修改配置文件指定服务端位置

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

以 # 开头的行，是注释

#server 0.rhel.pool.ntp.org iburst #前面加上#注释

#server 1.rhel.pool.ntp.org iburst #前面加上#注释

#server 2.rhel.pool.ntp.org iburst #前面加上#注释

server classroom.example.com iburst

3.重起服务

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

[root@server0 ~]# systemctl enable chronyd #随机自启动

4.验证：

[root@server0 ~]# date -s "2008-10-3"

2008年 10月 03日 星期五 00:00:00 CST

[root@server0 ~]# date

[root@server0 ~]# systemctl restart chronyd

**权限和归属**

• 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**

文本文件:

r: cat less head tail grep

w: vim 保存

x: 可以执行

• 权限适用对象(归属)

– **所有者**:拥有此文件/目录的用户-user **u**

– **所属组**:拥有此文件/目录的组-group  **g**

– **其他用户**:除所有者、所属组以外的用户-other  **o**

**查看权限**

– **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

**设置基本权限**

– **chmod [-R] 归属关系+-=权限类别 文档...**

**-R** : 递归修改目录下所有内容，及子目录所有内容

Permission denied : 权限不足

目录的 **r** 权限:能够 ls 浏览此目录内容

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

目录的 **x** 权限:能够 cd 切换到此目录

**设置文档归属**

• 使用 **chown** 命令

– **chown [-R] 属主 文档...**

**– chown [-R] :属组 文档...**

**– chown [-R] 属主:属组 文档...**

**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

– 继承所属组身份

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)

**acl访问控制列表**

• **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 /]# setfacl -m u:zhangsan:rx /nsd11

– **getfacl 文档... #查看ACL访问控制列表**

– **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] -x u:用户名 文档... #删除指定ACL**

**– setfacl [-R] -b 文档... #清空ACL**

**使用LDAP认证**

**LDAP服务器**：网络用户 用户的集中管理，用户信息由LDAP服务器提供

本地用户： 用户信息由/etc/passwd

**什么是LDAP?**

**1.**轻量级目录访问协议

**2.**为一组客户机集中提供可登录的用户账号

– 网络用户:用户名、密码信息存储在 LDAP 服务端

– 这些客户机都加入同一个 LDAP 域

**服务端**:LDAP服务器classroom.example.com

**客户端**：虚拟机Server

**1.安装客户端软件sssd**，与LDAP网络用户服务器沟通

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

**2.安装图形工具 authconfig-gtk** 配置**sssd**

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

**3.运行图形的工具进行配置**

[root@server0 ~]# authconfig-gtk

**选择LDAP**

**dc=example,dc=com #指定服务端域名**

**classroom.example.com #指定服务端主机名**

**勾选TLS加密**

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

**选择LDAP密码**

**4.重起服务**

[root@server0 ~]# systemctl restart sssd #重起服务

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

**5.验证:**

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

[root@server0 ~]# id ldapuser0

**家目录漫游**

• Network File System,网络文件系统

共享: 本地数据传递给网络中其他主机

– 由NFS服务器将指定的文件夹共享给客户机

– 客户机将此共享目录 mount 到本地目录,访问此共享

资源就像访问本地目录一样方便

– 类似于 EXT4、XFS等类型,只不过资源在网上

服务端:classroom.example.com NFS服务器共享所有用户的家目录

**客户端:虚拟机Server0**

**查看NFS资源**

# showmount -e classroom.example.com

**挂载NFS资源**

# mkdir /nfs

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

# ls /nfs

# umount /nfs

# mkdir /home/guests

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

# ls /home/guests

# su - ldapuser0

**find查找文件**

• 根据名称查找,忽略大小写： -iname

• 根据账号名称或所属组： -user、-group

# find /home/ -group student

• 限制目录查找的深度(最大层数)： -maxdepth

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

**内核升级**

下载软件包

• 使用wget下载工具

– wget 软件包的URL网址

在虚拟机Server：

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

默认下载到当前路径

# uname -r

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

# reboot

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

[root@server0 ~]# uname -r

3.10.0-123.1.2.el7.x86\_

**分区规划**

扇区的大小：默认 512 字节

• 识别硬盘 => 分区规划 => 格式化 => 挂载使用

一、**分区规划**

MBR分区模式 最大2TB空间

– **三种分区**类型：**主分区 扩展分区 逻辑分区**

– 1~4个主分区,或者 **3个主分区+1个扩展分区(n个逻辑分区)**

– 最大支持容量为 2.2TB 的磁盘

– 扩展分区不能格式化

– **理论上**：最多有**4**个主分区

/dev/sdc5 表示 scsi类型的磁盘，第三块，第一个逻辑分区

**1.查看本机识别硬盘**

[root@server0 ~]# **lsblk** #列出所有 块设备

**2.分区指令 fdisk**

[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

**3.格式化文件系统**

**mkfs** 工具集

**– mkfs.ext3 分区设备路径**

**– mkfs.ext4 分区设备路径**

**– mkfs.xfs 分区设备路径**

**– mkfs.vfat -F 32 分区设备路径**

**-f :强制格式化**

**4.挂载使用**

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

[root@server0 ~]# df -h #查看所有 正在挂载使用 的分区的信息

**5.开机自动挂载分区**

• 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**

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

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

/dev/vdb1 /mypart1 ext4 defaults 0 0

[root@server0 ~]# mount -a #检测/etc/fstab文件格式，并尝试挂载设备

[root@server0 ~]# df -h

**总结分区：**

**1.查看识别硬盘 lsblk**

**2.划分分区 fdisk**

**3.刷新新的分区 partprobe**

**4.格式化 mkfs.ext4**

**5.查看文件系统类型 blkid**

**6.挂载使用 mount**

**7.查看挂载使用 df -h**

**8.完成开机自动挂载 /etc/fstab**

**9.验证书写内容 mount -a**

**LVM逻辑卷**

作用： 1. 可以整合分散的空间

2. 逻辑卷可以容量扩大

将众多的物理卷（pv），组成卷组（vg），再从卷组中划分逻辑卷（lv），再去格式化逻辑卷，进行挂载

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

**一、制作逻辑卷**

1.**创建卷组** ： **vgcreate 卷组的名字 组成卷组的成员**

[root@server0 ~]# vgcreate myvg /dev/vdc[1-2]

[root@server0 ~]# pvs #显示物理卷的信息

[root@server0 ~]# vgs #显示卷组的信息

2.**创建逻辑卷** : **lvcreate -n 逻辑卷的名字 -L 大小 卷组的名字**

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

[root@server0 ~]# lvs #显示逻辑卷的信息

[root@server0 ~]# vgs #显示卷组的信息

3.**使用逻辑卷**

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

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

[root@server0 ~]# mkdir /lvm

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

/dev/myvg/vo /lvm ext4 defaults 0 0

[root@server0 ~]# mount -a

[root@server0 ~]# df -h

**逻辑卷的扩大**,支持线上操作

**一、卷组有足够的剩余空间**

1.**空间的扩展**

[root@server0 ~]# vgs

[root@server0 ~]# lvextend -L 18G /dev/myvg/vo

[root@server0 ~]# lvs

2.**文件系统的扩展**

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/myvg/vo

[root@server0 ~]# df -h

二、**卷组没有足够的剩余空间**

1.**扩展卷组空间**

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

[root@server0 ~]# vgs

2.**空间的扩展**

[root@server0 ~]# vgs

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

[root@server0 ~]# lvs

3.**文件系统的扩展**

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/myvg/vo

[root@server0 ~]# df -h

了解:**逻辑卷的缩小,有风险**

1.先缩减文件系统的大小,在缩减空间的大小

ext4文件系统**可以**缩减

xfs文件系统**不支持**缩减

**PE:卷组划分空间的单位**

• 创建卷组的时候设置PE大小

PE:卷组划分空间的单位 默认4M

[root@server0 ~]# vgdisplay 卷组名 #查看卷组详细信息

• 创建卷组的时候设置PE大小

– **vgcreate -s PE大小 卷组名 空闲分区...**

• 修改卷组的时候设置PE大小

**– vgchange -s PE大小 卷组名 空闲分区...**

• 创建逻辑卷的时候指定PE个数

– **lvcreate -l PE个数 -n 逻辑卷名 卷组名**

[root@server0 ~]# vgdisplay myvg

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

请划分一个逻辑卷名字lvtest01,大小为50个PE的总和

[root@server0 ~]# lvcreate -l 50 -n lvtest01 myvg

[root@server0 ~]# lvs

**逻辑卷的删除**

**先删除 逻辑卷 然后在删除 卷组**

[root@server0 ~]# umount /lvm

[root@server0 ~]# lvremove /dev/myvg/vo #删除 逻辑卷

Do you really want to remove active logical volume vo? [y/n]:y

[root@server0 ~]# lvs

[root@server0 ~]# vgremove myvg #删除 卷组

[root@server0 ~]# vgs

[root@server0 ~]# pvremove /dev/vdc[1-3] #删除 卷组

[root@server0 ~]# pvs

**ipv6地址配置**

ip地址：唯一标识网络中主机地址

**ipv4地址:**

**32个二进制 分成4段 最后用 点 分隔 用10进制表示**

**ipv6地址：**

**128个二进制 分成8段 最后用 冒号 分隔 用16进制表示**

每段内连续的前置 0 可省略、连续的多个 : 可简化为 ::

虚拟机server0：

# nmcli connection modify 'System eth0' ipv6.method manual

ipv6.addresses 2003:ac18::305/64 connection.autoconnect yes

# nmcli connection up 'System eth0'

# ping6 2003:ac18::305

**SELinux概述**

– RHEL7基于SELinux体系针对用户、进程、目录和文件

提供了预设的保护策略,以及管理工具

• SELinux的运行模式

– **enforcing(强制)**、**permissive(宽松)**

– **disabled(彻底禁用)**

任何状态变成disabled(彻底禁用)，都必须通过重起reboot

• **切换运行模式**

– 临时切换:**setenforce 1|0**

– 固定配置:**/etc/selinux/config** 文件

两台虚拟机均做

[root@server0 ~]# getenforce

Enforcing

[root@server0 ~]# setenforce 0

[root@server0 ~]# getenforce

Permissive

[root@server0 ~]# vim /etc/selinux/config #下一次开机后生效

SELINUX=permissive

**用户初始化文件**

• 影响指定用户的 bash 解释环境

– **~/.bashrc**,每次开启 bash 终端时生效

• 影响所有用户的 bash 解释环境

– **/etc/bashrc**,每次开启 bash 终端时生效

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

alias hello='echo hello'

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

alias hi='echo hi'

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

alias haha='echo haha'

root可以执行：

hello haha

student可以执行：

hi haha

**配置聚合连接**(网卡绑定、链路聚合)

作用：网卡设备的冗余

**热备份**(activebackup)连接冗余

**1**.**创建虚拟网卡team0 (参考 man teamd.conf) /example 全文搜索**

# **nmcli connection add type team**

**con-name team0 ifname team0 autoconnect yes**

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

# nmcli connection 添加 类型为 team

配置文件名字 team0 ifconfig显示的名字为 team0 每次开机自动启用配置 热备份方式

**2.添加成员**

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

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

# nmcli connection 添加 类型为 team的奴隶设备 配置文件名为 team0-1 奴隶的网卡名为 eth1 加入的组织为 team0

**3**.**配置IP地址**

[root@server0 ~]# nmcli connection modify team0 ipv4.method manual ipv4.addresses 192.168.1.1/24 connection.autoconnect yes

**4.激活**

# **nmcli connection up team0**

# **nmcli connection up team0-1**

# **nmcli connection up team0-2**

5.**验证查看**

# **teamdctl team0 state**

# ifconfig eth1 down

# teamdctl team0 state

**如果配置错误，可以从头来，删掉从新做**

# nmcli connection delete team0

# nmcli connection delete team0-1

# nmcli connection delete team0-2

**搭建基本Web服务**

服务端：虚拟机Server0

**1.安装一个可以提供Web功能软件**

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

**2.启动httpd服务**

# systemctl restart httpd #重起httpd服务

# systemctl enable httpd #设置httpd服务，开机自启动

**3.书写自己的页面文件**

默认存放网页文件的路径：/var/www/html

默认网页文件名字：index.html

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

<marquee><font color=red><h1>NSD1801 万岁 阳光明媚

客户端：虚拟机Server0

[root@server0 ~]# firefox 172.25.0.11

**搭建基本FTP服务(文件传输)**

服务端：虚拟机Server

**1.安装一个可以提供FTP功能软件**

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

**2.启动httpd服务**

# systemctl restart vsftpd #重起vsftpd服务

# systemctl enable vsftpd #设置vsftpd服务，开机自启动

**3.默认vsftpd共享路径: /var/ftp**

客户端：虚拟机Server

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

**防火墙策略的应用**

• 隔离作用

• 允许出站，过滤入站

• 硬件防火墙

• 软件防火墙

**RHEL7的防火墙体系**

• 系统服务:**firewalld**

• 管理工具:**firewall-cmd、firewall-config**(图形)

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等少数几个服务

– **trusted** : 允许任何访问

– **block** : 阻塞任何来访请求 (明确回应拒绝)

– **drop** : 丢弃任何来访的数据包 (没有明确回应，直接丢弃)

**防火墙的判断规则：匹配及停止**

1.客户端请求中的来源IP地址，查看本身所有区域的规则，如果有一个区域规则有该IP地址的规则，则进入该区域

2.进入默认区域 （默认区域一般为public）

默认区域的案例

虚拟机Server0：

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

虚拟机desktop0：

[root@desktop0 ~]# ping -c 2 172.25.0.11 #可以通行

虚拟机Server0：

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

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

虚拟机desktop0：

[root@desktop0 ~]# ping -c 2 172.25.0.11 #不可以通信，有回应

虚拟机Server0：

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

# firewall-cmd --get-default-zone

虚拟机desktop0：

[root@desktop0 ~]# ping -c 2 172.25.0.11 #不可以通信，没有回应

服务案例

虚拟机Server0

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

# **firewall-cmd --zone=public --list-all #查看区域策略**

虚拟机Desktop0

[root@desktop0 ~]# firefox 172.25.0.11 #不可以

[root@desktop0 ~]# firefox ftp://172.25.0.11 #不可以

虚拟机Server0

# **firewall-cmd --zone=public --add-service=http #添加协议**

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

虚拟机Desktop0

[root@desktop0 ~]# firefox 172.25.0.11 #可以

[root@desktop0 ~]# firefox ftp://172.25.0.11 #不可以

虚拟机Server0

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

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

虚拟机Desktop0

[root@desktop0 ~]# firefox 172.25.0.11 #可以

[root@desktop0 ~]# firefox ftp://172.25.0.11 #可以

**防火墙策略永久配置**

– 永久(**permanent**)

虚拟机Server0

# **firewall-cmd --reload**  #重新加载防火墙所有配置

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

# **firewall-cmd --permanent --zone=public --add-service=http** #设置永久

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

# firewall-cmd --reload #重新加载防火墙所有配置

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

**防火墙对于客户端源IP控制**

拒绝172.25.0.10访问本机的所有服务,其他客户端都允许

虚拟机desktop0

[root@desktop0 ~]# firefox 172.25.0.11 #可以

虚拟机Server0

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

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

# firewall-cmd --reload success

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

虚拟机desktop0

[root@desktop0 ~]# firefox 172.25.0.11 #不可以

**iSCSI磁盘的工作模式**

• 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

• ISCSI Qualified Name 名称规范

– iqn.yyyy-mm.倒序域名:自定义标识

– 用来识别 target 磁盘组,也用来识别客户机身份

iqn.2018-3.com.example:server0

服务端:虚拟机Server0

**1.划分分区/dev/vdb**

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

n ----->创建新的分区 5G

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

/dev/vdb1

[root@server0 ~]# lsblk

**2.安装服务端软件包,提供共享存储服务程序**

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

**3.运行targetcli,交互式的界面**

• 建立后端存储: /backstore/block create 后端存储名 实际设备路径

• 建立磁盘组(相当于制作存放分区的箱子): /iscsi create 磁盘组的IQN名称

• 将共享的分区放入箱子中: /iscsi/磁盘组名/tpg1/luns create 后端存储路径

• 指定提供服务的ip地址: /iscsi/磁盘组名/tpg1/portals create IP地址

• 访问控制的设置:/iscsi/磁盘组名/tpg1/acls create 客户端声称的名字

[root@server0 ~]# targetcli

/> backstores/block create iscsi\_nsd /dev/vdb1

/> iscsi/ create iqn.2018-03.com.example:server0

/> iscsi/iqn.2018-03.com.example:server0/tpg1/luns create /backstores/block/iscsi\_nsd

/> iscsi/iqn.2018-03.com.example:server0/tpg1/portals create 172.25.0.11

/> iscsi/iqn.2018-03.com.example:server0/tpg1/acls create

iqn.2018-03.com.example:desktop0

/> ls

/> exit

**4.启动target服务**

[root@server0 ~]# systemctl restart target #重起target服务

[root@server0 ~]# systemctl enable target #设置为开机自启动

**客户端:虚拟机Desktop0**

**1.安装客户端软件,访问iscsi共享存储**

[root@desktop0 ~]# yum repolist #生成Yum的缓存

[root@desktop0 ~]# yum -y install iscsi-initiator-utils.i686

**2.设置客户端访问时,声称的名字**

[root@desktop0 ~]# vim /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.2018-03.com.example:desktop0

**3.刷新 客户端iqn名字**

[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

[root@desktop0 ~]# systemctl enable iscsid

**4.加载iscsi共享存储,访问服务端( Ctrl+ -号 变小 Ctrl+shift +号 变大)**

[root@desktop0 ~]# iscsiadm --mode discoverydb --type sendtargets --portal 172.25.0.11 --discover

172.25.0.11:3260,1 iqn.2018-03.com.example:server0

# systemctl restart iscsi #重起iscsi服务加载共享存储

[root@desktop0 ~]# lsblk

**数据库服务基础**

什么是数据库:存放数据的仓库

数据库中有很多的小库,每一个库中有很多的表格,每一个表格中有很多的记录

表字段:表头

表记录:表内容

部署mariadb数据库服务器

• RHEL7 中的 MariaDB 相关包

– mariadb-server:提供服务端有关的系统程序

端口:3306

**一 搭建基本的数据库**

**1.安装mariadb-server**

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

**2.开启数据库服务**

[root@server0 ~]# systemctl restart mariadb

[root@server0 ~]# systemctl enable mariadb

**3.进入 MariaDB数据库 基本操作**

[root@server0 ~]# mysql

MariaDB [(none)]> show databases; #查看所有的库

MariaDB [(none)]> create database nsd; #创建库nsd

MariaDB [(none)]> show databases;

MariaDB [(none)]> drop database nsd; #删除库nsd

MariaDB [(none)]> show databases;

MariaDB [(none)]> create database nsd1802;

MariaDB [(none)]> show databases;

MariaDB [(none)]> exit

**4. 设置MariaDB数据库管理员的密码**

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

数据库管理员root 与 系统管理员root不为同一个用户

数据库管理员root:mysql库中user表

系统管理员root:/etc/passwd

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

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

数据库主配置文件:/etc/my.cnf

**5.导入数据库的数据**

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

[root@server0 ~]# mysql -u root -p123 nsd1802 < users.sql

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

MariaDB [(none)]> show databases;

| nsd1802

MariaDB [(none)]> use nsd1802; #进入库nsd1802

MariaDB [nsd1802]> show tables; #查看当前库的所有表格

对于表有四个操作: 增(insert) 删(delete) 改(update) 查(select)

MariaDB [nsd1802]> select \* from base; #显示base表所有字段内容

MariaDB [nsd1802]> select \* from location; #显示location表所有字段内容

查看表结构: desc 表名;

MariaDB [mysql]> select \* from nsd1802.base;

**6.数据库的授权,不需要创建本地用户**

MariaDB [(none)]> 交互指令

– GRANT 权限列表 ON 数据库名.表名 TO 用户名@

客户机地址 IDENTIFIED BY '密码';

grant select on nsd1802.\* to lisi@localhost identified by '123';

当lisi用户从本地localhost登陆输入密码123,将会获得nsd1802库中所有表的查询权限

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

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

MariaDB [(none)]> grant select on nsd1802.\* to lisi@localhost identified by '123';

MariaDB [(none)]> exit

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

**案例5:使用数据库查询**

2. 在系统 server0 上使用数据库 nsd1802,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?

有条件的查询: where

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

MariaDB [(none)]> use nsd1802;

MariaDB [nsd1802]> select \* from base;

MariaDB [nsd1802]> select \* from base where password='solicitous';

MariaDB [nsd1802]> select name from base where password='solicitous';

MariaDB [nsd1802]> select \* from base where name='tom';

**2)**有多少人的姓名是 Barbara 同时居住在 Sunnyvale? (多表联合查询)

MariaDB [nsd1802]> use nsd1802

MariaDB [nsd1802]> select \* from base,location

where base.name='Barbara' and location.city='Sunnyvale'

and base.id=location.id;

MariaDB [nsd1802]> select count(\*) from base,location

where base.name='Barbara' and location.city='Sunnyvale'

and base.id=location.id;

MariaDB [nsd1802]> insert base values (6,'Barbara','321');

MariaDB [nsd1802]> select \* from base;

MariaDB [nsd1802]> insert location values (6,'Sunnyvale');

MariaDB [nsd1802]> select \* from location;

MariaDB [nsd1802]> select \* from base,location

where base.name='Barbara' and location.city='Sunnyvale'

and base.id=location.id;

**1.** 禁止空密码root用户访问 mariadb 数据库(user表为登陆数据库用户的信息表)

MariaDB [nsd1802]> use mysql;

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

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

MariaDB [mysql]> delete from user where password=''; #删除表记录

MariaDB [mysql]> flush privileges; #刷新数据库策略

MariaDB [mysql]> exit

[root@server0 ~]# mysql -u root -h server0.example.com #测试网络登陆

**Web服务器**

独立Web主机

Web通信基本概念

• 基于 B/S (Browser/Server)架构的网页服务

– 服务端提供网页

– 浏览器下载并显示网页

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

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

• 软件包:**httpd**  (Apache软件基金会)

• 系统服务:httpd

• 提供的默认配置

**一 搭建基本的Web服务**

虚拟机Server0

**1.安装软件包**

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

**2.重起httpd服务**

[root@server0 ~]# systemctl restart httpd

[root@server0 ~]# systemctl enable httpd

**3.书写一个自己的页面文件**

[root@server0 ~]# echo '<h1>nsd1802' > /var/www/html/index.html

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

<h1>nsd1802

虚拟机Desktop0

[root@desktop0 ~]# firefox 172.25.0.11

– ServerName:本站点注册的DNS名称(空缺)

– DocumentRoot:网页文件根目录(/var/www/html)

DNS服务器:classroom.example.com

server0.example.com

www0.example.com

webapp0.example.com

通过ServerName指定本网站的域名为server0.example.com

服务端Server0

**1.修改配置文件/etc/httpd/conf/httpd.conf**

95行 ServerName server0.example.com:80

**2.重起httpd服务**

[root@server0 ~]# systemctl restart httpd

虚拟机Desktop0

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

– DocumentRoot:网页文件根目录(/var/www/html)

虚拟机Server0

**1.建立新的网页文件根目录**

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

[root@server0 ~]# echo '<h1>wo shi myweb' > /var/www/myweb/index.html

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

<h1>wo shi myweb

**2.修改配置文件**

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

DocumentRoot "/var/www/myweb"

**3.重起httpd服务**

[root@server0 ~]# systemctl restart httpd

虚拟机Desktop0

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

客户端浏览器路径 与 服务器真实路径的对照

客户端浏览器路径 firefox 172.25.0.11 -> DocumentRoot目录下/var/www/myweb

客户端 firefox 172.25.0.11/abc -> DocumentRoot目录下/var/www/myweb/abc

客户端 firefox 172.25.0.11/test -> DocumentRoot目录下/var/www/myweb/test

**虚拟主机的含义及类型**

• 虚拟Web主机

– 由同一台服务器提供多个不同的Web站点

• 区分方式

– 基于域名的虚拟主机

– 基于端口的虚拟主机

– 基于IP地址的虚拟主机

搭建 基于域名的虚拟主机

www0.example.com ----> 172.25.0.11 --> wo shi www0

webapp0.example.com ----> 172.25.0.11 -->wo shi webapp0

**• 配置文件路径**

– /etc/httpd/conf/httpd.conf (主配置文件)

– /etc/httpd/conf.d/\*.conf (调用配置文件)

• 为每个虚拟站点添加配置,容器式配置语句

<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>

虚拟机Server0

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

<VirtualHost \*:80> #在本机所有IP地址的80端口启用虚拟web功能

ServerName www0.example.com #指定网站域名

DocumentRoot /var/www/haha #指定网页文件根目录

</VirtualHost>

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/xixi

</VirtualHost>

[root@server0 ~]# mkdir /var/www/haha /var/www/xixi

[root@server0 ~]# echo '<h1>wo shi haha' > /var/www/haha/index.html

[root@server0 ~]# echo '<h1>wo shi xixi' > /var/www/xixi/index.html

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

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

[root@server0 ~]# systemctl restart httpd

虚拟机desktop0

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

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

一旦启用虚拟Web主机功能,所有的站点都必须用虚拟Web来实现

虚拟机Server0

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

<VirtualHost \*:80> #在本机所有IP地址的80端口启用虚拟web功能

ServerName www0.example.com #指定网站域名

DocumentRoot /var/www/haha #指定网页文件根目录

</VirtualHost>

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/xixi

</VirtualHost>

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /var/www/myweb

</VirtualHost>

**配置httpd服务访问控制**

在 Web 网站 http://server0.example.com 的

DocumentRoot 目录下创建一个名为 private 的子目录

仅允许本机可以浏览,其他客户端不可浏览

虚拟机Server0

# mkdir /var/www/myweb/private

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

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

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

<Directory /var/www/myweb/private>

Require ip 172.25.0.11

</Directory>

[root@server0 ~]# systemctl restart httpd

虚拟机Desktop0

# firefox server0.example.com/private #访问失败

虚拟机server0

# firefox server0.example.com/private #访问成功

– 每个文件夹自动继承其父目录的访问控制

– 除非针对子目录有明确设置

<Directory />

Require all denied #拒绝所有

</Directory>

<Directory "/var/www">

Require all granted #允许所有

</Directory>

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目录

虚拟机Server0

**1.建立新的网页文件目录**

[root@server0 ~]# mkdir /webroot

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

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

**2.修改的网页文件根目录**

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

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /webroot

</VirtualHost>

**3.修改访问控制**

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

<Directory /webroot>

Require all granted #允许所有

</Directory>

**4.重起服务**

[root@server0 ~]# systemctl restart httpd

**5.SELinux策略的控制,安全上下文(标签)**

[root@server0 ~]# ls -Zd /var/www/ #查看目录的上下文值

[root@server0 ~]# ls -Zd /webroot/ #查看目录的上下文值

• 方式1:参照标准目录,重设新目录的属性

– chcon [-R] --reference=模板目录 新目录

[root@server0]# chcon -R --reference=/var/www /webroot/ #修改标签值

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

虚拟机Desktop0

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

**部署动态网站**

**静态网站的运行**

• 服务端的原始网页 = 浏览器访问到的网页

– 由Web服务软件处理所有请求

**动态网站的运行**

• 服务端的原始网页 ≠ 浏览器访问到的网页

– 由Web服务软件接受请求,动态程序转后端模块处理

虚拟机Server0

**1.下载Python页面,以webapp0.example.com**

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

[root@server0 ~]# cd /var/www/xixi

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

[root@server0 xixi]# ls

[root@server0 xixi]# cat webinfo.wsgi

**2.修改虚拟Web主机配置,方便用户的访问,网页的跳转(网页的别名)**

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

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/xixi

Alias / /var/www/xixi/webinfo.wsgi

当客户端 访问到网页文件的根目录时 ,将/var/www/xixi/webinfo.wsgi呈现给客户端

</VirtualHost>

[root@server0 xixi]# systemctl restart httpd

客户端:虚拟机Desktop0 测试 firefox webapp0.example.com

**3.翻译Python页面**

[root@server0 /]# yum -y install mod\_wsgi #安装可以识别Python代码程序

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

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/xixi

WsgiScriptAlias / /var/www/xixi/webinfo.wsgi

</VirtualHost>

最终显示页面的内容为 UNIX时间戳 : 1970-1-1 0:0:0 到 指定时间 经过的秒数

[root@server0 xixi]# systemctl restart httpd

客户端:虚拟机Desktop0 测试 firefox webapp0.example.com

**4.此虚拟主机侦听在端口8909**

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

Listen 8909

<VirtualHost \*:8909>

ServerName webapp0.example.com

DocumentRoot /var/www/xixi

WsgiScriptAlias / /var/www/xixi/webinfo.wsgi

</VirtualHost>

[root@server0 /]# systemctl restart httpd

Job for httpd.service failed. See 'systemctl status httpd.service' and 'journalctl -xn' for details.

修改SELinux策略,查看SELinux允许的端口

[root@server0 /]# semanage port -l | grep http

添加8909端口

[root@server0 /]# semanage port -a -t http\_port\_t -p tcp 8909

-a:添加

-t:类型

-p:协议

[root@server0 /]# systemctl restart httpd

客户端:虚拟机Desktop0 测试 firefox webapp0.example.com:8909

**搭建虚拟Web主机**

虚拟机Server0

**1.安装httpd**

**2.修改调用配置文件/etc/httpd/conf.d/nsd01.conf**

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /var/www/html

</VirtualHost>

**3.书写一个简单的页面文件**

echo NSD1802 > /var/www/html/index.html

**4.重起服务httpd**

虚拟机Desktop0

elinks -dump server0.example.com

搭建安全的Web服务,加密传输数据

协议 https 安全的超文本传输协议 端口 443

虚拟机Server0

**1.部署网站证书**(营业执照)

# cd /etc/pki/tls/certs/ #存放网站证书的路径

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

# ls

**2.部署根证书**(公安局信息 颁发证书的服务器信息)

# cd /etc/pki/tls/certs/

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

**3.部署解密的密钥文件**

# cd /etc/pki/tls/private/

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

**4**.**安装支持安全Web软件 mod\_ssl**

# yum -y install mod\_ssl

**5**.**修改配置文件指定网站证书 根证书 密钥文件**

# vim /etc/httpd/conf.d/ssl.conf

末行模式 : set nu #开启行号功能

59行 DocumentRoot "/var/www/html"

60行 ServerName server0.example.com:443

100行 SSLCertificateFile /etc/pki/tls/certs/server0.crt #指定网站证书

107行 SSLCertificateKeyFile /etc/pki/tls/private/server0.key #指定密钥

122行 SSLCACertificateFile /etc/pki/tls/certs/example-ca.crt #指定根证书

**6**.**重起httpd服务**

# systemctl restart httpd

**7**.**验证虚拟机Desktop0**

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

**邮件服务**

• 电子邮件服务器的基本功能

– 为用户提供电子邮箱存储空间(用户名@邮件域名)

– 处理用户发出的邮件 —— 传递给收件服务器

– 处理用户收到的邮件 —— 投递到邮箱

DNS服务器 : classroom.example.com

server0.example.com

**搭建基本邮件服务器**

**1.安装postfix**

[root@server0 ~]# rpm -q postfix

**2**.**修改配置文件** **/etc/postfix/main.cf**

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

末行模式下:  **set nu**  #开启行号

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

116行 inet\_interfaces = all #允许所有网络接口164行 mydestination = server0.example.com #判断为本域邮件

**3.重起postfix服务**

[root@server0 ~]# systemctl restart postfix

SMTP协议:用户发邮件的协议 端口 25

pop3协议:用户收邮件的协议 端口 110

**测试收发邮件**

[root@server0 ~]# useradd yg

[root@server0 ~]# useradd xln

• **mail 发信操作**

– **mail -s '邮件标题' -r 发件人[@收件域] 收件人[@收件域]...**

• **mail 收信操作**

– **mail [-u 用户名]**

[root@server0 ~]# mail -s 'test' -r yg xln

[root@server0 ~]# mail -u xln

**准备交换分区**

什么是交换空间

• 相当于虚拟内存,

– 当物理内存不够用时,使用磁盘空间来模拟内存

– 在一定程度上缓解内存不足的问题

– 交换分区:以空闲分区充当的交换空间

虚拟机Server0

**1.划分新的分区,两个2G的主分区**

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

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

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

[root@server0 ~]# lsblk

**2.格式化 交换文件系统**

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

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

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

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

**3.启用 交换分区**

[root@server0 ~]# swapon -s #查看交换分区的组成 成员

[root@server0 ~]# swapon /dev/vdb1 #启用交换分区

[root@server0 ~]# swapon /dev/vdb2 #启用交换分区

[root@server0 ~]# swapon -s

[root@server0 ~]# swapoff /dev/vdb2 #停用交换分区

[root@server0 ~]# swapon -s

**4.完成开机自动启用swap分区**

[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 ~]# swapon -s

[root@server0 ~]# swapon -a #检测/etc/fstab文件交换分区书写内容

[root@server0 ~]# swapon -s

**Samba 软件项目**

(配置SMB共享, 跨平台的共享可以实现Linux与Linux之间,也可以实现Linux与windows之间)

– 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**

• **管理共享账号**

Samba用户 —— 专用来访问共享文件夹的用户

– 采用独立设置的密码

– 但需要提前建立同名的系统用户(可以不设密码)

• **使用 pdbedit 管理工具**

– 添加用户:**pdbedit -a** 用户名

– 查询用户:**pdbedit -L** [用户名]

– 删除用户:**pdbedit -x** 用户名服务端虚拟机Server0

• **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Server0：

**1.安装samba软件**

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

**2.建立Samba共享使用的帐号**

[root@server0 ~]# useradd harry

[root@server0 ~]# useradd kenji

[root@server0 ~]# useradd chihiro

[root@server0 ~]# pdbedit -a chihiro #将本地用户添加为Samba共享帐号

[root@server0 ~]# pdbedit -a harry #将本地用户添加为Samba共享帐号

[root@server0 ~]# pdbedit -a kenji #将本地用户添加为Samba共享帐号

[root@server0 ~]# pdbedit -L #显示所有Samba共享帐号

**3.创建共享目录**

[root@server0 ~]# mkdir /common

[root@server0 ~]# echo 123 > /common/1.txt

**4. 修改配置文件**

[root@server0 ~]# vim /etc/samba/smb.conf #vim命令模式 /workgroup

workgroup = STAFF #设置工作组

[common] #共享名

path = /common #共享的实际路径

**5.重起smb服务**

[root@server0 ~]# systemctl restart smb #重起smb服务

[root@server0 ~]# systemctl enable smb #设置smb服务开机自启动

**6. SELinux 布尔值 (开与关 on 与 off) 功能的开关**

– 需要加 **-P** 选项才能实现永久设置

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

#修改布尔值

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

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

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

**客户端访问服务端资源影响方面**

**1. 防火墙策略**

**2. 服务本身的访问控制**

**3. 服务端SELinux策略限制**

**4. 服务端目录的本地权限**

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

**客户端虚拟机Desktop0:了解内容**

**1**.安装客户端软件**samba-client**

**2**.利用**smbclient**进行访问

列出共享资源

[root@desktop0 ~]# smbclient -L 172.25.0.11

Enter root's password: #直接回车

连接到共享文件夹

– smbclient -U 用户名 //服务器地址/共享名

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

Enter harry's password:

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

smb: \>

**方便科学的访问利用mount挂载**

**-o**:添加选项

客户端虚拟机Desktop0:

**1**.**安装支持samba挂载的软件,文件系统cifs-utils**

# yum -y install cifs-utils

**2.挂载访问**

# mkdir /mnt/nsd

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

# df -h

# ls /mnt/nsd

**/etc/fstab完成samba开机自动挂载**

**\_netdev**:网络设备(先启动网络服务配置好ip地址,然后再挂载该设备)

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

**//172.25.0.11/common /mnt/nsd cifs defaults,user=harry,pass=123,\_netdev 0 0**

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

[root@desktop0 ~]# df -h

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h

**读写Samba共享**

服务端虚拟机server0

**1.修改配置文件/etc/samba/smb.conf**

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

[devops]

path = /devops

write list = chihiro #允许chihiro可以写

**2.创建共享目录结构**

[root@server0 ~]# mkdir /devops

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

[root@server0 ~]# ls /devops/

**3**.**重起smb服务**

[root@server0 ~]# systemctl restart smb

**4.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

**5.修改本地权限**

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

[root@server0 ~]# getfacl /devops/

**客户端Desktop0:挂载访问**

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

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

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

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h

了解:多用户的samba访问,专为不同权限的普通用户设计

在 desktop0 上访问 server0 提供的共享 devops

**1)将此共享永久挂载在 /mnt/dev 目录**

**2)挂载时以用户 kenji 作为认证**

**3)必要的时候,任何普通用户都可以通过用户 chihiro 来临时获取写的权限**

[root@desktop0 /]# umount /mnt/stu/

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

//172.25.0.11/devops /mnt/stu cifs defaults,user=kenji,pass=123,\_netdev,multiuser,sec=ntlmssp 0 0

[root@desktop0 /]# mount -a

[root@desktop0 /]# df -h

[root@desktop0 /]# su - natasha

[natasha@desktop0 stu]$ cd /mnt/stu

[natasha@desktop0 stu]$ cifscreds add -u chihiro 172.25.0.11

Password:

[natasha@desktop0 stu]$ ls

**配置NFS共享**

Linux与Linux之间的共享

NFS共享概述

• Network File System,网络文件系统

– 用途:为客户机提供共享使用的文件夹

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

• 所需软件包**:nfs-utils**

• 系统服务:**nfs-server**

**搭建基本NFS服务**

服务端虚拟机Server0:

**1.安装软件nfs-utils**

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

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

**2.修改配置文件/etc/exports**

[root@server0 ~]# mkdir /public

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

[root@server0 ~]# ls /public/

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

/public \*(ro) #共享路径 客户端地址(权限)

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

虚拟机Desktop0:

[root@desktop0 /]# mkdir /mnt/nfs

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

172.25.0.11:/public /mnt/nfs nfs defaults,\_netdev 0 0

[root@desktop0 /]# mount -a

[root@desktop0 /]# df -h

**环境变量**

USER:存放登陆的用户身份

PATH:储存命令的搜寻路径

执行命令时,都会区域PATH变量值中去寻找,该命令的执行程序,

如果能找到就执行,不能找到报告命令没有找到

[root@server0 ~]# echo $PATH

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

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

#!/bin/bash

echo hello

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

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

[root@server0 ~]# ls /usr/bin/hello

[root@server0 ~]# hello

**划分分区**  **GPT**

GPT : 分区模式 128个主分区 最大18EB容量

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

(parted) mktable gpt #设置分区模式

(parted) print #显示分区表内容

(parted) mkpart #划分新的分区指令

分区名称？ []? haha #指定分区名称

文件系统类型？ [ext2]? ext4 #指定文件系统类型,不起实际作用

起始点？ 0

结束点？ 1G

警告: The resulting partition is not properly aligned

for best performance.

忽略/Ignore/放弃/Cancel? Ignore #忽略GPT分区模式,占有的磁盘空间

(parted) print

(parted) unit GB #使用GB作为单位

(parted) mkpart

分区名称？ []? haha

文件系统类型？ [ext2]? ext4

起始点？ 1G

结束点？ 2G

(parted) quit

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

**FTP服务**

服务端虚拟机Server0:

默认共享路径:/var/ftp

默认访问用户身份:ftp

FTP客户端访问FTP服务端时,登陆的位置为该用户,在服务器上的家目录

**1.安装vsftpd软件**

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

**2.启动vsftpd服务**

[root@server0 ~]# systemctl restart vsftpd

[root@server0 ~]# systemctl enable vsftpd

客户端Desktop0:

**1.安装客户端软件lftp**

[root@desktop0 ~]# yum -y install lftp

[root@desktop0 ~]# lftp 172.25.0.11

lftp 172.25.0.11:~> ls

请计算Linux系统一共有多少个用户?

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

41 /etc/passwd

请计算Linux系统/etc/有多少个tab结尾的文件?(不包含子目录)

[root@server0 /]# ls /etc/\*tab | wc -l

请计算Linux系统/etc/有多少个tab结尾的文件?(包含子目录)

[root@server0 /]# find /etc/ -name "\*tab" | wc -l

请计算Linux系统/etc/有多少个.conf结尾的文件?(包含子目录)

[root@server0 /]# find /etc/ -name "\*.conf" | wc -l

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 #判断位置变量的个数是否为0个

echo '/root/foo.sh redhat|fedora' >&2 #将正确的输出结果变成错误的输出

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 #将正确的输出结果变成错误的输出

exit 3 #脚本退出返回值

fi

**在 server0 上创建 /root/batchusers 脚本**

1)此脚本要求提供用户名列表文件作为参数

2)如果没有提供参数,此脚本应该给出提示 Usage:/root/batchusers,退出并返回相应值

if [ $# -eq 0 ];then

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

exit 2

fi

3)如果提供一个不存在的文件,此脚本应该给出提示 Input file not found,退出并返回相应值

if [ -f $1 ];then

for i in `cat $1`

do

useradd -s /bin/false $i

echo $i成功

done

else

echo 'Input file not found' >&2

exit 2

fi

4)新用户的登录Shell为 /bin/false ,无需设置密码

5)用户列表测试文件:

http://classroom/pub/materials/userlist

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

#!/bin/bash

if [ $# -eq 0 ];then

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

exit 2

fi

if [ -f $1 ];then

for i in `cat $1`

do

useradd -s /bin/false $i

echo $i成功

done

else

echo 'Input file not found' >&2

exit 2

Fi

**防火墙 处理 端口转发 功能**

虚拟机Server0:基本的Web服务

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

[root@server0 ~]# echo '<h1>NSD1802' > /var/www/html/index.html

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

[root@server0 ~]# systemctl restart httpd

[root@server0 ~]# systemctl enable httpd

[root@server0 ~]# firefox 172.25.0.11

实现默认区域为public,在默认区域里添加允许客户端访问http

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

# firewall-cmd --reload

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

服务或协议的端口 : 标识服务或协议

常见的协议：

**http** 超文本传输协议 默认端口  **80**

**https**  安全的超文本传输协议 默认端口 **443**

**ftp** 文件传输协议 默认端口 **21**

**tftp** 简单文件传输协议 默认端口 **69**

**telnet** 远程管理协议 默认端口 **23**

**dns** 域名解析协议 默认端口 **53**

**smtp** 邮件协议 默认端口 **25**

**pop3** 收邮件协议 默认端口 **110**

**snmp**  简单的管理协议 默认端口 **161**

虚拟机Desktop0 访问 虚拟机Server0 5423端口

将访问 虚拟机Server0 5423端口的请求 转发到 虚拟机Server0 80端口

虚拟Desktop0 ------->172.25.0.11:5423

虚拟机Server0 -----172.25.0.11:5423------>172.25.0.11:80

虚拟机Server0

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

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

# firewall-cmd --reload

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

虚拟机Desktop0

# firefox 172.25.0.11:5423

**建SERVICES教学环境**

第一台虚拟机,一般作为服务端 ,虚拟机A

真机 快速产生一台虚拟机: **clone-vm7(clone-auto7)**

[root@room9pc01 ~]# clone-vm7

Enter VM number: 5

**1**.**将虚拟机的名字修改为A**

**2.以root用户登陆虚拟机A,密码为123456**

虚拟机A要求:

**3.配置永久的主机名:svr7.tedu.cn**

[root@localhost ~]# echo svr7.tedu.cn > /etc/hostname

[root@localhost ~]# hostname svr7.tedu.cn

**4**.**配置eth0网卡 永久的IP地址: 192.168.4.7/24**

**5**.**检测防火墙是否开启,关闭防护墙**

# systemctl status firewalld

**6.检测SELinux是否开启,关闭SELinux**

# getenforce

Disabled

**为虚拟机A搭建Yum仓库**

[root@room9pc01 ~]# ssh -X root@192.168.4.7

root@192.168.4.7's password:

Last login: Thu Mar 29 09:09:37 2018 from 192.168.4.254

真机作为Yum服务端

**1.利用FTP共享光盘所有内容**

[root@room9pc01 ~]# rpm -q vsftpd #安装软件包

vsftpd-3.0.2-22.el7.x86\_64

[root@room9pc01 ~]# systemctl status vsftpd #查看服务是否开启

默认共享的路径:/var/ftpSESERVICESSERVICESRVICES

# firefox ftp://192.168.4.254/rhel7

# mkdir /var/ftp/rhel7

# mount /var/lib/libvirt/images/iso/rhel-server-7.4-x86\_64-dvd.iso /var/ftp/rhel7SERVICES

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

/var/lib/libvirt/images/iso/rhel-server-7.4-x86\_64-dvd.iso /var/ftp/rhel7 iso9660 defaults 0 0

SERVICES

**虚拟机A作为Yum客户端**

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

[rhel7]

name=dvd

baseurl=ftp://192.168.4.254/rhel7

enabled=1

gpgcheck=0

[root@svr7 ~]# yum -y install mariadb-server

[root@svr7 ~]# 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 SERVICES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

虚拟机A操作

**权限的数值表示, 利用 4 2 1 三个数字表示 r w x**

[root@svr7 ~]# mkdir /nsd01

[root@svr7 ~]# ls -ld /nsd01

[root@svr7 ~]# chmod 700 /nsd01

**历史命令**

• 管理/调用曾经执行过的命令

– **history**:查看历史命令列表

– **history -c**:清空历史命令

– !**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

**du,统计文件的占用空间SERVICES**

– **du** **[选项]... [目录或文件]...**

– **-s**:只统计每个参数所占用的总空间大小

– **-h**:提供易读容量单位(K、M等)

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

1.5G /root/

43M /etc/

12K /home/

**zip归档工具,跨平台的压缩工具**

• 归档+压缩操作

– **zip [-r] 备份文件.zip 被归档的文档...**

• 释放归档+解压操作

– **unzip 备份文件.zip [-d 目标文件夹]**

[root@svr7 ~]# zip -r /opt/test.zip /home/ /mnt/

[root@svr7 ~]# mkdir /nsd

[root@svr7 ~]# unzip /opt/test.zip -d /nsd

[root@svr7 ~]# ls /nsd

**5**:显示配置文件的帮助信息

[root@svr7 ~]# man 5 passwd

**实用小命令工具**

• **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

**制作快捷方式**

格式: **ln -s /路径/源文件 /路径/快捷方式名称**

[root@svr7 ~]# cat /etc/redhat-release

[root@svr7 ~]# ln -s /etc/redhat-release /abc

[root@svr7 ~]# ls -l /abc

[root@svr7 ~]# cat /abc

**i节点: 硬盘上文件唯一的编号**

**只有硬连接,不会产生新的文件**

[root@svr7 ~]# rm -rf /opt/\*

[root@svr7 ~]# echo 123 > /opt/A

[root@svr7 ~]# cat /opt/A

[root@svr7 ~]# ln -s /opt/A /opt/A1 #制作软连接

[root@svr7 ~]# ls /opt/

[root@svr7 ~]# ls -l /opt/A1

[root@svr7 ~]# ln /opt/A /opt/A2 #制作硬连接

[root@svr7 ~]# ls /opt/

[root@svr7 ~]# cat /opt/A2

[root@svr7 ~]# cp /opt/A /opt/A3

[root@svr7 ~]# ls /opt/

[root@svr7 ~]# cat /opt/A3

[root@svr7 ~]# ls -i /opt #查看文件 i节点 编号

**自定义yum仓库**

**1.有光盘之外的包**

**2.将真机tools文件夹,传递到虚拟机A根目录下**

真机上实现

# scp -r /root/桌面/tools root@192.168.4.7:/

**3.到虚拟机A验证**

# ls /

# ls /tools

**4.生成仓库数据文件**

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

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

**5.修改配置文件/etc/yum.repos.d/dvd.repo**

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

[rhel7]

name=dvd

baseurl=ftp://192.168.4.254/rhel7

enabled=1

gpgcheck=0

[myrpm]

name=myrpm

baseurl=file:///tools/other

enabled=1

gpgcheck=0

[root@svr7 ~]# yum repolist

**显示已安装软件包的安装清单**

**rpm -ql 软件名**

**vim编辑技巧**

命令模式操作

[root@svr7 ~]# head -6 /etc/passwd > /opt/abc.txt

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

光标跳转

**Home**键 或 **^** 跳转到行首

**End**键 或**“$”**键 跳转到行尾

**1G** 或 **gg**  跳转到文件的首行

**G**  跳转到文件的末尾行

**复制/粘贴/删除**

**yy、#yy**  复制光标处的一行、#行

粘贴 **p**  粘贴到光标处之后、之前

**x** 或 **Delete**键 删除光标处的单个字符

**dd、#dd**  删除光标处的一行、#行

**d^** 从光标处之前删除至行首

**d$**  从光标处删除到行尾

**C(大写的)** 从光标处删除到行尾,并且进入输入模式

**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

**ZZ**  保存修改并退出

**末行模式操作**

**:q!** 放弃已有更改后强制退出

**:wq** 或 **:x**  保存已有修改后退出

**:r /etc/filesystems** 读入其他文件内容

[root@svr7 ~]# echo 123 > /opt/1.txt

[root@svr7 ~]# echo abc > /opt/2.txt

[root@svr7 ~]# vim /opt/2.txt

**:r /etc/redhat-release**  #读入其他文件内容

**:r /opt/1.txt**  #读入其他文件内容

**字符串替换**

**:s/old/new**  替换当前光标所在行第一个“old”

**:s/old/new/g**  替换当前光标所在行所有的“old”

**:n,m s/old/new/g** 替换第n-m行所有的“old”

**:% s/old/new/g** 替换文件内所有的“old

[root@svr7 ~]# head -10 /etc/passwd > /opt/abc.txt

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

:s/root/admin/

:s/root/admin/g

:1,8 s/root/admin/g

:%s /root/admin/g

**开关参数的控制**

**:set nu** 或 **nonu**  显示/不显示行号

**:set ai** 或 **noai** 启用/关闭自动缩进

**vim不允许多终端,打开同一个文件**

**vim不能创建目录**

**源码编译安装**

rpm包------->rpm yum----->安装

源码包------gcc make-------> 可以执行的程序--------->运行安装

• 主要优点

– 获得软件的最新版,及时修复bug

– 软件功能可按需选择/定制,有更多软件可供选择

– 源码包适用各种平台

**步骤1**:**安装gcc 与 make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**

[root@svr7 ~]# tar -xf /tools/inotify-tools-3.13.tar.gz -C /opt/

[root@svr7 ~]# ls /opt/

[root@svr7 ~]# ls /opt/inotify-tools-3.13/

[root@svr7 ~]# cd /opt/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: 指定安装目录/功能模块等选项

作用2: 检测本机是否安装gcc与make

[root@svr7 ~]# cd /opt/inotify-tools-3.13/

[root@svr7 inotify-tools-3.13]# ./configure --prefix=/mnt/myrpm

**报错提示: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**

[root@svr7 inotify-tools-3.13]# make

**步骤5: make install 安装,将编译好的文件复制到安装目录**

[root@svr7 inotify-tools-3.13]# make install

[root@svr7 inotify-tools-3.13]# ls /mnt/myrpm/

bin include lib share

**systemctl控制**

[root@svr7 /]# systemctl restart 服务名 #重起服务

[root@svr7 /]# systemctl enable 服务名 #设置服务开机自启动

[root@svr7 /]# systemctl status 服务名 #查看服务当前的状态

[root@svr7 /]# systemctl start 服务名

[root@svr7 /]# systemctl stop 服务名

[root@svr7 /]# systemctl disable 服务名 #设置服务开机不再自启动

**模式控制**

字符模式: multi-user.target

图形模式: graphical.target

• 切换到文本/图形模式

– systemctl isolate multi-user.target

– systemctl isolate graphical.target

• 查看默认模式

– systemctl get-default

• 设置默认模式

– systemctl set-default multi-user.target

– systemctl set-default graphical.target

[root@svr7 /]# systemctl isolate graphical.target #当前临时切换

[root@svr7 /]# reboot

[root@svr7 /]# systemctl set-default graphical.target #修改默认

[root@svr7 /]# systemctl get-default #查看默认模式

[root@svr7 /]# reboot

浏览器 http：//172.40.50.113/note

vncviewer 172.40.50.113:7000

classroom

提前搭建实验环境：

svr7.tedu.cn ==》 192.168.4.7/24

pc207.tedu.cn ==》 192.168.4.207/24

svr7作为主要的DNS服务器

pc207作为测试用的客户机

路由器，只认识IP地址，不认识 www.baidu.com

网站 www.baidu.com 的IP地址在哪里？

问某个DNS服务器 类似于 114查号台

注册一个域名 zylinux.cn，

获得的权利：

决定 \*.zylinux.cn 的IP地址是多少

我的Web服务器 放在 某一个电信机房，IP地址 60.72.83.94

添加一个 www.zylinux.cn ==》60.72.83.94

**搭建DNS服务器的基本步骤** ——

**1. 装包**

# yum -y install bind bind-chroot

**2. 配置**

主配置文件 /etc/named.conf 【管理哪些区域、地址库文件在哪】

# cat /etc/named.conf

options { directory "/var/named"; };

zone "tedu.cn" {

type master;

file "tedu.cn.zone";

};

# named-checkconf 【若配置正确，无任何输出】

地址库（区域数据）文件： /var/named/自定义文件名

# vim /var/named/tedu.cn.zone

.... // 文件头参考 named.localhost 文件的内容复制过来

@ NS svr7.tedu.cn.

www A 192.168.4.100

svr7 A 192.168.4.7

pc207 A 192.168.4.207

# named-checkzone tedu.cn /var/named/tedu.cn.zone

zone tedu.cn/IN: loaded serial 0

OK

chroot 切换根目录（提供一个伪造的根）【笼环境】

==》把 /var/named/chroot/ 目录当成 / 根目录

把192.168.4.7配置为 tedu.cn 区域的DNS服务器

www.tedu.cn ===》 192.168.4.100

svr7.tedu.cn ===》 192.168.4.7

pc207.tedu.cn ===》 192.168.4.207

一个服务器的IP地址，为它注册多个域名

www.tedu.cn

www.tmooc.cn

www.tarena.com.cn

tts8.tmooc.cn

.. ..

**3. 起服务**（在svr7上，!!! 关闭防火墙）

# systemctl restart named

客户机测试（在pc207主机上查询DNS ——

**1）为客户机指定默认DNS**

# nmcli con mod eth0 ipv4.dns 192.168.4.7

# nmcli con up eth0

**2）测试DNS查询结果**

# host www.tedu.cn

www.tedu.cn has address 192.168.4.100

6. //地址库里有记录的可以查到结果

# host wwww.tedu.cn

Host wwww.tedu.cn not found: 3(NXDOMAIN)

//地址库里没有记录的查不到结果

互联网网站访问 www.tedu.cn 场景：

**1）浏览器**（客户机）—— 192.168.4.7

a. 输入 http://www.teud.cn/ 并回车

b. 浏览器向DNS查询“www.tedu.cn 的IP地址是多少？”

c. 封装一个HTTP访问请求发给目标IP地址 192.168.4.100

d. 收到来自网站服务器提供的网页

**2）DNS服务器**（新网）—— 192.168.4.7

a. 收到来自客户机的查询请求

b. 检查客户机的问题是否属于自己管理的DNS区域（地址库内查找是否有一条 www 的A类型的记录）

c. 告诉客户机 www.tedu.cn 的IP地址（192.168.4.100）

**3）网站服务器**（达内）—— 192.168.4.100

付费注册DNS域名 —— 让DNS服务器告诉其他任何客户机“www.tedu.cn 的IP地址是 192.168.4.100”

a. 收到来自浏览器提交的HTTP访问请求

b. 由本机的Web服务程序来处理这个HTTP访问请求

c. 按照浏览器的请求提供相应的网页

DNS轮询（负载均衡）—— 比如百度、网易、……

特点：同一个域名 --> 多个IP地址

www.tedu.cn -->

192.168.4.100 可以承受 10000个用户同时访问

192.168.4.110 +10000并发负载能力

192.168.4.120 +10000并发负载能力

实现DNS轮询的关键：修改所在地址库的解析记录

www A 192.168.4.100

www A 192.168.4.110

www A 192.168.4.120

泛域名解析 ——

特点：多个域名（后缀相同） --> 同一个IP地址

www.jd.com

或者 jd.com

3w.jd.com

sanw.jd.com

当用户访问一个域名时，只要后缀是正确的，都能够找到正确的网站

实现泛域名解析的关键：修改所在地址库的解析记录

\* A 192.168.4.100 ---> wwwwww.tedu.cn

@ A 6. 192.168.4.100 ----> tedu.cn

DNS服务器 ：. 根域（上级域）

DNS服务器6. ：.cn 一级（上级域）

DNS服务器 svr7：tedu.cn 二级（上级域）

www.tedu.cn 192.168.4.100

DNS服务器 pc207：bj.tedu.cn 三级（下级域）

www.bj.tedu.cn 1.2.3.4

亲密/疏远：

属于同一个机构管理的上下级区域，一般称为 父子域

子域授权配置关键（修改父区域地址库）：

子域域名. NS 子DNS的FQDN.

子DNS的FQDN. A 子DNS的IP地址

当一台DNS6.服务器（tedu.cn）收到一个查询非本DNS管辖的域名（www.bj.tedu.cn）的请求时 ——

情况1：未配置子域授权，告诉客户机“not found”

情况2-父子域（递归查询）：配置了子域授权，帮客户机去查询并把结果告诉客户机

情况3-上下级域（迭代查询）：配置了子域授权，不愿意/不方便帮客户机去查询结果，但是会告知客户机应该去问另一个DNS服6.务器

缓存DNS服务器（比如教学环境CentOS真机 192.168.4.254）

1）能正常上网（至少能查询各种域名）

2）客户机（比如pc207）必须能正常访问本DNS

实现过程 ——

0）简单准备工作

停用与DNS服务冲突的其他服务（dnsmasq）

# pkill -9 dnsmasq

1）装包 bind bind-chroot

2）配置（指向可用的公共DNS服务器地址）

# vim /etc/named.conf

options { forwarders { 172.40.1.10; }; };

3）起服务 named

客户机测试：

# host www.qq.com 192.168.4.254

今日重点：

1）理解DNS的作用

2）搭建基本的DNS服务器

3）客户机如何使用指定的DNS

4）缓存DNS怎么部署

**1）理解DNS系统的作用**

互联网的指路人，提供信息查询“网站域名 --》IP地址”

知道一个www.baidu.com的IP地址

不等于

能访问 www.baidu.com 的网页

FQDN，完全合格的域名 ==》互联网的一个服务器的DNS地址

www.pku.edu.cn

www 站点名（对应一台服务器）

pku.edu.cn 区域名（对应一个公司/单位/机构/组织）

**2）学会搭建一台权威DNS服务器**

为 tedu.cn 区域提供域名解析（意思：当有其他客户机来问你这台DNS服务器 “以 tedu.cn 结尾的那些域名的IP地址是多少”时，你这台DNS应该能提供正确的答案）

svr7.tedu.cn 的 IP地址是 192.168.4.7

pc207.tedu.cn 的 IP地址是 192.168.4.207

www.tedu.cn 的 IP地址是 192.168.4.100

配置关键：

# vim /etc/named.conf 【负责的区域、地址库文件在哪】

options {

directory "存放地址库文件的默认目录";

};

zone "区域1名" {

type master;

file "地址库1文件名";

};

zone "区域2名" {

type master;

file "地址库2文件名";

};

# vim /.../地址库文件名 【XX站点的IP地址是多少？】

文件头（参考模板文named.localhost件复制）

@ NS 本DNS服务器的FQDN.

站点1名 A 站点1的IP地址

站点2名 A 站点2的IP地址

.. ..

**3）学会使用DNS服务器**

为Linux主机指定默认使用哪一台DNS服务器：

# nmcli con mod "连接名" ipv4.dns DNS服务器的IP地址

# nmcli con mod up "连接名" 【永久】

# vim /etc/resolv.conf 【临时】

nameserver DNS服务器的IP地址

测试DNS服务器时可以手动临时指定使用哪一台DNS服务器：

# host XX域名 DNS服务器的IP地址

**4）学会搭建一台缓存DNS服务器**

前提条件：

1）客户机能访问到缓存DNS

2）缓存DNS服务器能够正常访问其他的可用的公共DNS

配置关键：

# vim /etc/named.conf

options {

forwarders { 公共DNS的IP地址; };

};

**DNS的子域授权** ——

告诉上一级DNS服务器：

子域叫什么名，负责子域内站点解析的下一级DNS服务器的FQDN.

下一级DNS服务器的FQDN. 的IP地址是多少

对于上一级DNS服务器来说，

如果 子域授权+ 愿意为客户机查询非本机管理的域名 ==》递归（缓存DNS服务器）

如果 子域授权+ 不愿意为客户机查询非本机管理的域名 ==》迭代（有下级子域权威DNS）

recursion yes|no;

DNS地址库的特殊应用 ——

DNS轮询，一个域名解析为多个IP地址

# vim 地址库文件

.. ..

www A IP地址1

www A IP地址2

www A IP地址3

泛域名，多个未知域名解析为同一个IP地址

# vim 地址库文件

.. ..

\* A IP地址

练习：扩展DNS服务器 svr7的功能

除了负责 tedu.cn 区域以外，

还能够为教学环境的 example.com 提供域名解析服务

实验目标（提供解析结果）：

classroom.example.com ==》172.25.254.254

content.example.com ==》172.25.254.254

server0.example.com ==》172.25.0.11

www0.example.com ==》172.25.0.11

webapp0.example.com ==》172.25.0.11

desktop0.example.com ==》172.25.0.10

smtp0.example.com ==》172.25.0.10

在服务器svr7上：

**1）调整主配置**

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

options {

directory "/var/named";

};

zone "tedu.cn" {

type master;

file "tedu.cn.zone";

};

zone "example.com" {

type master;

file "example.com.zone";

};

**2）建地址库**

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

[root@svr7 named]# cp -p tedu.cn.zone example.com.zone

[root@svr7 named]# vim example.com.zone

.. ..

@ NS svr7.tedu.cn.

classroom A 172.25.254.254

content A 172.25.254.254

server0 A 172.25.0.11

www0 A 172.25.0.11

webapp0 A 172.25.0.11

desktop0 A 172.25.0.10

smtp0 A 172.25.0.10

**3）起服务**

[root@svr7 named]# systemctl restart named

两个问题：

1）options全局设置只需要一段

2）为什么 example.com 区域的NS记录还是 svr7.tedu.cn. 【一个DNS服务器管理多个区域时，没必要在每一个区域都注册一个域名】

今日目标：

1）理解分离解析原理，并通过实验验证

2）了解RAID阵列的几种方式和各自的特点

3）学会查看和控制系统的进程信息

4）学会通过日志去了解系统或服务发生的事件

场景1：

www.12306.cn

==》 IP地址1（联通）、IP地址2（移动）、IP地址3（电信）、……

当一个联通宽带用户访问 www.12306.cn 时，面临的问题？

单纯的DNS轮询难以解决

需要DNS更加智能 ——

1）当联通的客户机来访问www.12306.cn，告诉客户机IP地址1

2）当移动的客户机来访问www.12306.cn，告诉客户机IP地址2

1）当电信的客户机来访问www.12306.cn，告诉客户机IP地址3

智能DNS（Split分离解析）

Content Delivery Network，内容分发网络

铁道部：负责提供Web网站/数据库内容，假设 2 台

CDN服务商（蓝汛）：负责为铁道部在全国部署200台缓存服务器，由这些缓存服务器面向购票用户

购票用户：===》CDN服务商的缓存服务器

铁道部成本降低

CDN服务商赚钱

用户访问速度块

CDN服务商需要做的事情：

当用户查询www.12306.cn时，把客户机引导到最近/最快的那一台缓存服务器上

**配置关键步骤：**

1）建立2份地址库

/var/named/tedu.cn.zone.1 【www ==》192.168.4.100】

/var/named/tedu.cn.zone.2 【www ==》1.2.3.4】

2）在 /etc/named.conf 创建2个视图

options {

directory "/var/named";

};

view "mylan" { //第1个视图设置

match-clients { 192.168.4.207; 192.168.7.0/24; };

zone "tedu.cn" {

type master;

file "tedu.cn.zone.1";

};

};

view "other" { //第2个视图设置

match-clients { any; };

zone "tedu.cn" {

type master;

file "tedu.cn.zone.2";

};

};

了解RAID阵列的几种方式和各自的特点

RAID —— 把一堆便宜磁盘整合成一块大磁盘

主要实现两个目标

1）提高磁盘读写速度

2）提供硬件备份

其他好处：

拥有更大容量的磁盘、性价比更高

阵列如何实现（RAID级别）？

500GB X 6块 构建 RAID5 阵列

==》2.5TB

500GB X 6块 构建 RAID6 阵列

==》2TB

500GB X 6块 构建 RAID0 阵列

==》3TB

500GB X 6块 构建 RAID1 阵列

==》500GB

500GB X 6块 构建 RAID10 阵列

==》3块组条带 x2 镜像 1.5TB

==》2块组条带 x3 镜像 1.0TB

学会查看和控制系统的进程信息 ——

Process，进程（内存中正在进行的程序）

整个Linux操作系统是由大量的进程（每一个进程都由一个编号PID）一起提供服务的

系统开机 ==》内核 ==》系统管理器（PID 1） ==》其他程序。。。

父进程

子进程

### 查看进程信息 ——

为了更好的掌握系统的状态

pstree 看进程树

练习：

1）查看Linux系统的整个进程树结构，显示各进程的PID号

# pstree -p

2）从上述结果中找出sshd进程的PID值

# pstree -p | grep sshd

3）根据sshd进程的PID值查看进程树，列出各进程的命令行

# pstree -ap 937

ps，进程快照，用来为系统中所有进程拍照

练习：

1）安装并启用httpd服务

# yum -y install httpd

# systemctl restart httpd

2）列出系统中所有的httpd进程，比较每个进程属于哪个用户

# ps aux | grep httpd

3）列出系统中所有httpd进程的父进程号，观测差别

# ps -elf | grep httpd

4）查看前一步结果中PID编号最小（辈分最大）的httpd进程的进程树

# pstree -p 9634

关于httpd网站服务器：

第1个httpd进程由root运行，不面向Web用户

第2个及以后的httpd进程是以apache用户身份运行，面向Web用户

知道一个进程的父进程有什么作用？

比如病毒进程

如果一个进程被杀死，这个进程会自动把所有子进程都终止

top，查看进程资源占用情况，动态展示（实时录像）

# top

top - 15:53:09 up 6:23, 3 users, load average: 0.00, 0.01, 0.05

Tasks: 150 total, 2 running, 148 sleeping, 0 stopped, 0 zombie

%Cpu(s): 0.1 us, 0.1 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st

KiB Mem : 2049108 total, 1267212 free, 303156 used, 478740 buff/cache

KiB Swap: 2097148 total, 2097148 free, 0 used. 1556160 avail Mem

排名信息。。。

CPU处理器的平均负载：最近1分钟，5分钟，15分钟

load average: 0.00, 0.01, 0.05

%Cpu(s): 99.7 id 空闲

按数字1切换多核模式

按q键退出

练习：

1）查看当前主机的CPU负载情况

2）查看当前主机开启了多长时间、有几个用户登录、运行的进程数

3）查看当前主机的CPU、内存、交换分区空闲比率

4）观测当前占用CPU资源最多的进程叫什么名

pgrep 检索进程信息

练习：

1）查看用户mike开启的进程名、PID号

# pgrep -l -U mike

2）统计当前主机运行的进程的数量

# pgrep . | wc -l

3）检查网站服务进程httpd是否已经开启

# pgrep -l httpd

4）找出名字为 gdm 的进程的PID号

# pgrep -x gdm

pgrep 关键词 ==》只要进程名包括这个关键词，就符合条件

pgrep -x 关键词 ==》必须进程名等于这个关键词，才符合条件

例如：有2个进程 —— 张三、张三丰

# pgrep 张三

张三的PID

张三丰的PID

# pgrep -x 张三

张三的PID

2）控制进程

进程前后台调度 ——

测试用的命令工具格：

# sleep 秒数

fg，forground 前台

bg，background 后台

jobs，列出任务

Ctrl z 暂停当前任务

杀死进程（选择进程不同的死法）——

根据进程的PID号

# kill PID .. ..

根据后台进程的任务编号

# kill %任务编号 .. ..

根据进程的名称

# killall 进程名

根据进程所属的用户（符合XX条件）

# pkill -U 用户名

!!! 不带信号 -9 表示正常杀死进程

!!! 带信号 -9 表示强制杀死

图形桌面下的点杀工具 xkill ——

按 Alt F2 调出运行命令的工具，输入 xkill 并回车

今日重点：

1）分离解析的原理

2）RAID磁盘阵列的各个级别的特点

3）查找进程的各种方法

4）杀死进程的各种方法

准备装机环境 ——

在真机上提供软件安装源 http://192.168.4.254/rh7dvd

**1）启用httpd服务**

# systemctl restart httpd

**2）挂载RHEL7的ISO文件到 /var/www/html/rh7dvd**

# mkdir -p /var/www/html/rh7dvd

# vim /etc/fstab

ISO文件的绝对路径 /var/www/html/rh7dvd iso9660 defaults 0 0

# mount /var/www/html/rh7dvd

# ls /var/www/html/rh7dvd/ //确认目录下有资源

**3）检查，确认软件源可访问**

浏览器访问 http://192.168.4.254/rh7dvd，可看到软件目录资源

常见问题：执行mount -a 挂载不成功

原因：同一个镜像挂载到多个位置，如果其中一个位置已经挂载，那么在执行mount -a时不再挂载新的位置

解决办法：

重起，

或者 mount 挂载点

或者 卸载这个设备的所有挂载点，再重新mount -a

DHCP服务器，为客户机动态提供IP地址相关参数

一台计算机如何从DHCP服务器获取IP地址

1）客户机 请求，DHCP DISCOVERY

2）服务器 提供，DHCP OFFER

3）客户机 选择，DHCP REQUEST

4）服务器 确认，DHCP ACK

快速搭建DHCP服务器（svr7） ——

1）装包 dhcp

# yum -y install dhcp

2）配置

# vim /etc/dhcp/dhcpd.conf

subnet 192.168.4.0 netmask 255.255.255.0 {

range 192.168.4.10 192.168.4.200;

next-server 192.168.4.7; //PXE装机服务器地址

filename "pxelinux.0"; //启动文件

}

3）起服务 dhcpd

# systemctl restart dhcpd

4）确认监听状态

# netstat -anptu | grep :67

5）在客户机pc207上测试

# dhclient -d eth0

选项含义：

-a，所有连接

-n，以数字方式显示地址和端口

-p，列出对应的进程和PID

-t，列出TCP类型的连接

-u，列出UDP类型的连接

DHCP服务监听端口：UDP 67

BOOTROM 启动芯片

PXE 应用：

方式1：远程装操作系统

方式2：远程运行操作系统（无盘工作站）

实现PXE网络装机 ——

基本条件：

1. DHCP地址分配服务

2. yum 软件源

PXE引导配置：

3. 启用TFTP引导服务，提供用来启动客户机的文件资料

1）装包、起服务

# yum -y install tftp-server

# systemctl restart tftp

2）提供网卡启动文件 pxelinux.0

# yum -y install syslinux

# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/pxelinux.0

—— 建议在客户机pc207先测试，确保可以下载

# yum -y install tftp

!!! 测试方法：tftp 服务器地址 -c get 文件名

# tftp 192.168.4.7 -c get pxelinux.0

# ls -lh pxelinux.0 //确认下载结果

3）提供用来安装XX系统的内核文件（及运行内核所需的初始化文件）

#### 访问 http://192.168.4.254/isolinux/

#### 找到 vmlinuz、initrd.img 文件的下载地址，复制使用

# mkdir /var/lib/tftpboot/rhel7

# cd /var/lib/tftpboot/rhel7

# wget http://vmlinuz文件的下载地址

# wget http://initrd.img文件的下载地址

# ls /var/lib/tftpboot/rhel7/ //确认部署结果

initrd.img vmlinuz

4）提供pxelinux.0启动程序的配置文件（用来显示启动菜单）

#### 访问 http://192.168.4.254/isolinux/

#### 找到 isolinux.cfg 文件的下载地址，复制使用

# mkdir /var/lib/tftpboot/pxelinux.cfg/

# cd /var/lib/tftpboot/pxelinux.cfg/

# wget http://isolinux.cfg文件的下载地址 -O default

# cd /var/lib/tftpboot/

# wget http://vesamenu.c32图形模块支持文件的下载地址

# wget http://splash.png背景图片文件的下载地址

# ls /var/lib/tftpboot/ //确认部署结果

isolinux.cfg rhel7 vesamenu.c32

pxelinux.0 splash.png

# ls pxelinux.cfg/ //确认菜单配置文件

default

!!!! TFTP服务的默认资源目录

/var/lib/tftpboot/

!!!! 查找哪一个软件包能够提供 XXXX 文件

# yum provides "\*/XXXX"

4. 提供装机菜单

# vim /var/lib/tftpboot/pxelinux.cfg/default

default vesamecu.cfg //菜单显示由vesa...负责

timeout 600 //倒计时等待时间（1/10秒）

label linux

menu label ^Install Red Hat Enterprise Linux 7.2

kernel rhel7/vmlinuz 【第63行】

append initrd=rhel7/initrd.img inst.stage2=http://192.168.4.254/rh7dvd 【第64行】

练习：新建一台虚拟机，测试PXE网络装机的效果

1）虚拟机定制：从网络安装、内存2G、硬盘20G、网络private1

2）启动后能成功看到选择菜单

3）选择第一项后，成功加载图形安装程序

4）手动指定分区方式、软件仓库（http://192.168.4.）

客户机的引导过程：

一台需要装RHEL7系统的客户机（裸机）

==》电源 ==》主板开机BIOS自检 ==》优先从网络启动

==》网卡芯片里的PXE Client程序

==》获取IP地址、按照DHCP指示去找TFTP服务器

==》向TFTP服务器索取 pxelinux.0 并下载

==》在内存中运行 pxelinux.0

==》从TFTP资源目录 pxelinux.cfg/default 下载到菜单配置

==》按照default文件配置显示一个选择界面

==》如果客户机的用户选择“Install Red Hat Enterprise Linux 7.2”，那么

==》找TFTP服务器的 rhel7/ 目录下载内核、初始化文件

==》在内存中运行vmlinuz内核文件，由内核获取系统控制权限

==》根据配置的inst.stage2参数指定的位置去获取安装程序

==》在客户机运行安装程序

==》由安装程序主导XX系统安装过程

kernel panic，内核 恐慌

常见错误：

1）配置文件 default 错写为 deafult

2）inst.stage2指定的软件源位置写错 http://.../rhdvd

3）DHCP服务配置中

subnet 192.168.4.0 netmask 254.254.254.0

4）tftp服务没开启

5）……

NIC，Network Interface Card，网卡

实现自动应答的关键配置：

1）需要一份提前准备好的应答文件，客户机必须能够下载

确认访问地址：http://192.168.4.254/ks-rhel7.cfg

2）修改 default 菜单配置使用这份应答文件

# vim /var/lib/tftpboot/pxelinux.cfg/default

label linux

.. ..

kernel rhel7/vmlinuz

append initrd=rhel7/initrd.img ks=http://1 92.168.4.254/ks-rhel7.cfg

全自动裸机装配（装系统、配置系统）

安装后脚本 —— 使PXE客户机在完成系统安装之后自动执行XX操作

比如：自动配置好yum源

# rm -rf /etc/yum.repos.d/\*.repo

!!!!

### echo 'yum源配置文件内容' > /etc/yum.repos.d/rhel7dvd

# echo '[rhel7]

name=Red Hat Enterprise Linux 7.2

baseurl=http://192.168.4.254/rh7dvd

gpgcheck=0' > /etc/yum.repos.d/rhel7.repo

**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**
3. **优点**：**易于实现、易于网络扩展、易于故障排查**
4. **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）

1. **双绞线标准与分类** ：

Cat5五类双绞线，传输频率为100MHz，适用于100Mbps的网络

Cat 5e衰减更小，串扰更少，性能优于5类线 （超五类）

Cat 6传输频率为200MHz

Cat 7传输频率为600MHz

**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**

**1、2**发送,**3、6**接收

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一般用交叉线**

**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**

Switch#**show version**   **查看IOS版本信息**

配置enable明文口令

全局配置模式：enable password *123*

配置enable密文口令

全局配置模式： enable secret *456*

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

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

**禁用DNS查询**

switch(config)#**no ip domain-lookup**

接口模式：**shutdown禁用此接口**

1. **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

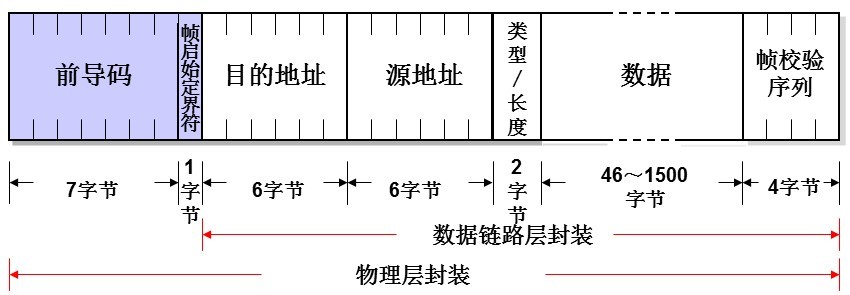
**MAC地址长度48位(6个字节)，前24位代表厂商，后24位代表网卡编号，**

**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后续数据的字节长度。

**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

初始状态

根据源MAC地址学习

除源端口外的端口广播未知数据帧

接收方回应

交换机实现单播通信（转发）

更新：老化时间300秒

交换机对应端口的MAC 地址发生变化时

**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

全局配置模式创建vlan

全局：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

全局： **interface range f0/1 – 10**

switchport access vlan 2将1-10口同时加入vlan2

**3、**查看**vlan**信息

特权：**show vlan**

**三、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**：

1）接入链路: 可以承载1个 vlan

2）中继链路：可以承载多个 vlan

**3、vlan的标识**

1）ISL(cisco私有的标记方法)

ISL外部封装头部26个字节，尾部4个字节共30字节

2）IEEE 802.1q(公有的标记方法)

内部封装在标准以太网帧内插入了4个字节，其中12位vlan标识。

**4、ISL和802.1Q 的异同**

相同点:都是显示了VLAN的信息

不同点:

IEEE 802.1Q是公有的标记方式，ISL是Cisco私有的

ISL采用外部标记的方法，802.1Q采用内部标记的方法

ISL标记的长度为30字节，802.1Q标记的长度为4字节

**5、trunk的配置**

接口模式：switchport mode trunk(直接配置为trunk)

access（配置为接入链路）

**6、在trunk链路上移除某vlan**

进入trunk接口：switchport trunk allowed vlan remove 3中继链路不允许传送vlan 3的数据

**7、在trunk链路上 添加某vlan**

进入trunk接口：switchport trunk allowed vlan add 3

**8、查看接口模式**

特权：show interface f0/5 switchport

**四、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

1. 查看以太网通道的配置：

特权：show etherchannel summary

1. 以太网道必须遵循以下一些规则：

1）参与捆绑的端口必须属于同一个vlan,如果是在中继模式下，要求所有参加捆绑的端口都配置成中继模式。

**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

3)、浮动路由

配置浮动静态路由，需设置管理距离大于1，从而成为备份路由，实现链路备份的作用。

**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、基于CEF 的快速转发**

主要包含两个转发用的信息表：

1）转发信息库（FIB）：FIB类似于路由表，包含路由表中转发信息的镜像。当网络的拓扑发生变化时，路由表将被更新，而FIB也将随之变化。

2）邻接关系表：每个FIB条目，邻接关系表中都包含相应的第2层地址。

**3、虚拟接口**（SVI）

三层交换机上配置的VLAN接口为虚接口

**4、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

减少了管理任务

占用了网络带宽

**三、路由协议分类**

1、按应用范围的不同，路由协议可分为两类：

1）在一个AS内的路由协议称为[内部网关协议](http://baike.baidu.com/view/706465.htm" \t "_blank)（interior gateway protocol），正在使用的[内部网关路由协议](http://baike.baidu.com/view/4217560.htm" \t "_blank)有以下几种：RIP-1，RIP-2，IGRP，EIGRP，IS-IS和OSPF。

2）AS之间的路由协议称为[外部网关协议](http://baike.baidu.com/view/543338.htm" \t "_blank)（exterior gateway protocol）。

外部网关协议（External Gateway Protocol，EGP，也叫域 间路由协议）。域间路由协议有两种：外部网关协议（EGP）和边界网关协议（BGP）

注：AS自治系统（Autonomous System，指一个[互连网络](http://baike.baidu.com/view/545553.htm" \t "_blank)，就是把整个Internet划分为许多较小的网络单位，这些小的网络有权自主地决定在本系统中应采用何种路由协议）

3）以情况下，需要使用BGP：

· 当你需要从一个AS发送[流量](http://baike.baidu.com/view/66422.htm" \t "_blank)到另一个AS时；

2、按照路由执行的算法动态路由协议的分类

1）距离矢量路由协议

依据从源网络到目标网络所经过的路由器的个数选择路由

RIP、IGRP

1. 链路状态路由协议

综合考虑从源网络到目标网络的各条路径的情况选择路由

OSPF、IS-IS

**四、RIP路由协议**

RIP是距离-矢量路由选择协议

RIP度量值为跳数 ，最大跳数为15跳，16跳为不可达

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*

**配置DHCP服务**

1、全局ip dhcp pool 名字（定义地址池）

2、network 192.168.1.0 255.255.255.0（动态分配IP地址段）

3、default-router 192.168.1.254（动态分配的网关地址）

4、dns-server 202.106.0.20（动态分配的DNS服务器地址）

5、全局：ip dhcp excluded-address 192.168.1.1（预留已静态分配的IP地址）

1. **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

1. 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

1. **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

1. 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

1. 删除已建立的访问控制列表

全局：no access-list 列表号

1. 接口上取消ACL

接口模式：no ip access-group 列表号in 或out

1. 查看访问控制列表

特权：show access-lists

**三、扩展访问控制列表**

1、作用

可以根据源IP地址，目的IP地址，指定协议，端口等过滤数据包。

2.扩展访问控制列表号:100-199

1. eq等于、lt小于、gt大于

4、扩展访问控制列表案例：

例1：全局： access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255

（允许192.168.1.0网络访问192.168.2.0网络的所有服务）

全局： access-list 101 deny ip any any

（拒绝所有）

例2：全局： access-list 101 deny tcp 192.168.1.0 0.0.0.255 host 192.168.2.2 eq 80

（拒绝192.168.1.0网段访问192.168.2.2的TCP的80端口）

全局：access-list 101 permit ip any any（允许访问所有）

5、删除扩展ACL

全局：no access-list 列表号

注：扩展与标准ACL不能删除单条ACL语句，只能删除整个ACL。

6、扩展ACL应该应用在离源地址最近的路由器上。

1. **NAT**（网络地址转换）
2. 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。
3. 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

不支持某些应用，可以通过静态NAT映射来避免

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：

进入出口配置：ip 　nat 　outside

进入入口配置：ip 　nat 　inside

端口映射：

ip nat inside source static tcp 192.168.1.6 80 61.159.62.133 80

1. 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。

1. **查看NAT转换条目**

特权：show ip nat translations显示当前存在的转换

1. **清除NAT转换条目**

1、特权： clear ip nat translation \* 清除NAT转换条目中的所有所条目

注：静态NAT条目不会被清除

1. **显示每个转换的数据包**

特权：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