[计算机基础 2](#_Toc573453109)

[什么是Linux 2](#_Toc1829525158)

[什么是互联网网络: 2](#_Toc685130888)

[服务器: 3](#_Toc401154809)

[WAN与LAN 3](#_Toc497390849)

[Linux 的应用 3](#_Toc1543182982)

[系统开机启动顺序： 3](#_Toc31690220)

[硬件: 3](#_Toc677124201)

[管理用户和组 4](#_Toc1063979505)

[IP相关 4](#_Toc650763892)

[TCP/IP协议: 4](#_Toc1820506162)

[IP地址: 5](#_Toc255859793)

[子网掩码: 6](#_Toc590281619)

[IP路由: 6](#_Toc1518199854)

[DNS服务器地址 : 7](#_Toc1529096677)

[Linux基础命令 7](#_Toc691952033)

[命令行基本概念: 7](#_Toc57604)

[基础命令介绍: 7](#_Toc640957682)

[快捷键: 13](#_Toc1860248937)

[通配符: 14](#_Toc819880940)

[特殊符号: 14](#_Toc1363821629)

[网络部分 15](#_Toc2044137901)

[网络拓扑结构 15](#_Toc1697754433)

[常见网络设备 16](#_Toc833293712)

[交换机工作原理 16](#_Toc1687930041)

[交换机的命令模式 16](#_Toc1339425892)

[网络线缆 18](#_Toc1145090700)

[各种服务 19](#_Toc2011001693)

[服务的概念 19](#_Toc1807374392)

[ftp(文件传输协议)服务搭建: 19](#_Toc2046644698)

[HTTP(超文本传输协议)服务搭建: 20](#_Toc1253007814)

[NTP(网络时间协议)服务: 20](#_Toc233343853)

[Linux目录含义 20](#_Toc1728686208)

[/ (根目录) : 20](#_Toc1938138702)

[dev : 20](#_Toc634498663)

[proc: 21](#_Toc78593410)

[etc: 21](#_Toc1333838036)

[安装软件包 22](#_Toc666188883)

[RPM Package Manager, RPM包管理器: 22](#_Toc755717611)

[Yellow Dog Update Manager -- 用来自动解决依赖关系 22](#_Toc250333894)

[源码装包: 23](#_Toc1316952776)

[硬件相关 23](#_Toc428740125)

[硬盘分区: 23](#_Toc506193687)

[U盘挂载: 24](#_Toc1907234395)

[常用软件 24](#_Toc1946939979)

[curl : 字符浏览器 24](#_Toc2035290365)

[vim : 24](#_Toc451702780)

[git: 25](#_Toc1946997583)

[单用户模式: 26](#_Toc528764399)

[进入单用户模式: 26](#_Toc164468069)

[修改root密码: 26](#_Toc619394875)

[重要的配置文件: 26](#_Toc1892586028)

[锁定项配置文件: 26](#_Toc61122323)

[网卡配置文件: 27](#_Toc169665661)

[DNS配置文件: 27](#_Toc578396092)

[报错实例解决 27](#_Toc1749052364)

[ssh运行程序出现 cannot connect to X server 27](#_Toc1509091553)

# 计算机基础

## 什么是Linux

Linux是一个操作系统

操作系统=内核+外围应用软件 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

--标准读音:“哩呐科斯”

发行版的名称/版本由发行方决定

--Red Hat Enterprise Linux 5/6/7

--Suse Linux Enterprise 12

--Debian Linux 7.8

--Ubuntu Linux 14.10/15.04

## 什么是互联网网络:

计算机网络的功能:数据通信,资源共享,增加可靠性,提高系统处理能力

## 服务器:

机架式服务器最多 , 以厚度表示大小 , 1U = 4.445cm

CPU : Intel Xeon (至强)

内存 : ECC纠错技术

硬盘 : 转速10000RPM/15000RPM

服务器一般托管到IDC (Inernet Data Center)

上网采用的结构 : C/S(Client/Server) B/S(Browser/Server)

## WAN与LAN

广域网(wide-aera Network)

-范围：几十到几千千米

-作用：用于连接远距离的计算机网络

-典型应用：Internet

局域网(Local Area Network)

-范围：1KM左右

-作用：用于连接较短距离内的计算机

-典型应用：企业网、校园网

## Linux 的应用

企业服务器

嵌入式系统

高性能的大型运算

## 系统开机启动顺序：

1.硬件自检

2.读硬盘第一个扇区的分区表信息

3.grub2，开机引导到内核

4.启动第一个进程systemd

5.启动fstab和服务

6.四个配置：bashrc，.bashrc，profile，profile.d，rc.local(x权限)->登录界面

## 硬件:

-主板

-电源

-CPU

-内存:RAM 断电丢数据

-硬盘:ROM

## 管理用户和组

唯一标识用户:UID

管理员root的UID : 0

组:方便对用户的管理,方便对权限管理

唯一标识组:GID

Linux用户至少属于一个组

组的分类

基本组 : 创建用户时系统会自动创建同名组

附加组(从属组) : 自己创建并加入的组

## RAID磁盘阵列

• 廉价冗余磁盘阵列

– Redundant Arrays of Inexpensive Disks

– 通过硬件/软件技术,将多个较小/低速的磁盘整合成一个大磁盘

– 阵列的价值:提升I/O效率、硬件级别的数据冗余

– 不同RAID级别的功能、特性各不相同

• RAID 0,条带模式

– 同一个文档分散存放在不同磁盘

– 并行写入以提高效率

– 无可靠性提升

– 至少由两块磁盘组成

• RAID 1,镜像模式

– 一个文档复制成多份,分别写入不同磁盘

– 多份拷贝提高可靠性,效率无提升

– 至少由两块磁盘组成

• RAID5,高性价比模式

– 相当于RAID0和RAID1的折中方案

– 需要至少一块磁盘的容量来存放校验数据

– 至少由三块磁盘组成

• RAID6,高性价比/可靠模式

– 相当于扩展的RAID5阵列,提供2份独立校验方案

– 需要至少两块磁盘的容量来存放校验数据

– 至少由四块磁盘组成

• RAID 0+1/RAID 1+0 (RAID 10)

– 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

– 至少由四块磁盘组成

## 进程管理

程序:静态的代码 占用磁盘空间

进程:动态的代码 占用内存与CPU

父进程/子进程 结构:树型结构

进程编号:PID

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:所有进程的父进程 (上帝进程)

• 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 (PPID 父进程的PID)

– 列出正在运行的所有进程

补充:

计算当前系统中正在运行的进程有多少个

#wc 统计函数 -l 统计行数

进程动态排名

• top 交互式工具

– 格式:top [-d 刷新秒数] [-U 用户名]

输入 P(大写) 进行CPU排序

输入 M(大写) 进行内存排序

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

进程的前后台调度

命令行末尾添加“&”符号,不占用当前终端,正在运行放入后台

• Ctrl + z 组合键

– 挂起当前进程(暂停并转入后台)

• jobs 命令

– 查看后台任务列表

• fg 命令

– 将后台任务恢复到前台运行

• bg 命令

– 激活后台被挂起的任务

杀死进程

• 干掉进程的不同方法

– Ctrl+c 组合键,中断当前命令程序

– kill [-9] PID...

– killall [-9] 进程名...

– pkill 查找条件

## 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

– 信息安全控制的“依据”

• 由系统服务rsyslog统一记录/管理

– 日志消息采用文本格式

– 主要记录事件发生的时间、主机、进程、内容

• 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

• 通用分析工具

– tail、tailf、less、grep等文本浏览/检索命令

– awk、sed等格式化过滤工具

用户登录分析

• users、who、w 命令

– 查看已登录的用户信息,详细度不同

• last、lastb 命令

– 查看最近登录成功/失败的用户信息

日志消息的优先级

• Linux内核定义的事件紧急程度

– 分为 0~7 共8种优先级别

– 其数值越小,表示对应事件越紧急/重要

0 EMERG（紧急） 会导致主机系统不可用的情况

1 ALERT（警告） 必须马上采取措施解决的问题

2 CRIT（严重） 比较严重的情况

3 ERR（错误） 运行出现错误

4 WARNING（提醒） 可能会影响系统功能的事件

5 NOTICE（注意） 不会影响系统但值得注意

6 INFO（信息） 一般信息

7 DEBUG（调试） 程序或系统调试信息等

使用journalctl工具

• 提取由 systemd-journal 服务搜集的日志

– 主要包括内核/系统日志、服务日志

• 常见用法

– journalctl | 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"

systemd介绍

• Linux系统和服务管理器

– 是内核引导之后加载的第一个初始化进程(PID=1)

– 负责掌控整个Linux的运行/服务资源组合

• 一个更高效的系统&服务管理器

– 开机服务并行启动,各系统服务间的精确依赖

– 配置目录:/etc/systemd/system/

– 服务目录:/lib/systemd/system/

– 主要管理工具:systemctl

对于服务的管理

systemctl restart 服务名 #重起服务

systemctl start 服务名 #开启服务

systemctl stop 服务名 #停止服务

systemctl status 服务名 #查看服务当前的状态

systemctl enable 服务名 #设置服务开机自启动

systemctl disable 服务名 #设置服务不开机自启动

RHEL6 运行级别 200

0：关机 0

1：单用户模式（基本功能的实现，破解Linux密码） 50

2：多用户字符界面（不支持网络） 80

3：多用户字符界面（支持网络）服务器默认的运行级别 100

4：未定义

5：图形界面 180

6：重起 0

切换运行级别：init 0

RHEL7 运行模式

字符模式：multi-user.target

图形模式：graphical.target

当前直接切换到图形模式

[root@svr7 /]# systemctl isolate graphical.target

当前直接切换到字符模式

[root@svr7 /]# systemctl isolate multi-user.target

查看每次开机默认进入模式

# systemctl get-default

multi-user.target

设置永久策略，每次开机自动进入graphical.target

# systemctl set-default graphical.target

## rsync同步操作

• 命令用法

– rsync [选项...] 源目录 目标目录

• 同步与复制的差异

– 复制:完全拷贝源到目标

– 同步:增量拷贝,只传输变化过的数据

• rsync操作选项

– -n:测试同步过程,不做实际修改

– --delete:删除目标文件夹内多余的文档

– -a:归档模式,相当于-rlptgoD

– -v:显示详细操作信息

– -z:传输过程中启用压缩/解压

• 本地同步

– rsync [选项...] 本地目录1 本地目录2 //同步整个文件夹

– rsync [选项...] 本地目录1/ 本地目录2 //只同步目录下的数据

rsync+SSH同步

• 与远程的 SSH目录保持同步

– 下行:rsync [...] user@host:远程目录 本地目录

– 上行:rsync [...] 本地目录 user@host:远程目录

两台服务器之间实时同步

一 部署公钥私钥ssh无密码验证

1.在虚拟机svr7生成公钥 私钥

# ssh-keygen

/root/.ssh/known\_hosts:记录曾经远程管理过的服务器信息

1. 将公钥进行拷贝到目标机

#ssh-copy-id root@192.168.4.207

1. 查看公钥是否拷贝成功
2. 验证无密码的同步

二 运行一个程序,监控目录内容的变化

• Linux内核的 inotify 机制

– 提供事件响应式的文件系统通知机制

– 安装 inotify-tools 控制工具可调用此机制实现监控

• 基本用法

– inotifywait [选项] 目标文件夹

• 常用命令选项

– -m,持续监控(捕获一个事件后不退出)

– -r,递归监控、包括子目录及文件

– -q,减少屏幕输出信息

– -e,指定监视的 modify、move、create、delete、attrib 等事件类别

# inotifywait -mrq /opt/

三 书写一个shell脚本,完成实时同步

while inotifywait -rqq /opt/

do

rsync -az --delete /opt/ root@192.168.4.207:/opt/

done

# 权限和归属

## 基本权限的类别

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

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

对于文本文件:

r : 用cat , less , tail , head 查看内容

w : 用vim > >>写入

x : 一般是Shell脚本

对于目录:

r : 能够 ls 浏览此目录内容

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

x : 能够 cd 切换到此目录

权限前以 - 开头:文本文件

以 d 开头:目录

以 l 开头:快捷方式

Permission denied:权限不足

Linux判断用户具备的权限: 匹配及停止

1.判断用户对于文档,属于什么身份 所有者>所属组>其他人

2.查看文档,相应角色的权限

## 附加权限

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

– 传递所属组身份(让子文档自动继承父目录的所属组身份)

Set UID

附加在属主的 x 位上

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

- 属主的权限标识会变为 s

- 传递所有者身份

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)

权限的数值表示

• 权限的数值化

– 基本权限:r = 4,w = 2,x = 1

– 附加权限:SUID = 4,SGID = 2,Sticky Bit = 1

• 采用数值形式设置权限

– chmod [-R] nnn 文档...

– chmod [-R] xnnn 文档...

## acl访问控制列表

acl策略的作用

文档归属的局限性

– 任何人只属于三种角色:属主、属组、其他人

– 无法实现更精细的控制

acl访问策略

– 能够对个别用户、个别组设置独立的权限

– 大多数挂载的EXT3/4、XFS文件系统默认已支持

# IP相关

## TCP/IP协议:

1.TCP/IP族是获最广泛支持的通信协议集合,包括大量的Internet应用中的标准协议,支持跨网络架构,跨操作系统平台的通信

1. TCP/IP协议的作用:

规定主机之间通信的三要素

3主机与主机之间通信的三要素

IP

子网掩码

IP路由

1. TCP/IP五层模型

应用层

传输层

网络层 路由器

数据链路层 交换机

物理层 网卡

小链接:

1. ARPAnet : DoD (美国国防部) 出资创办了高级研究项目署 , 即ARPA , 他把当时美国三所大学的4台计算机连接起来

## IP地址:

1. IP地址(点分十进制): 32位的2进制数 因为这么长的2进制数对人来说非常不友好,为了方便人为记忆和表示 , 每8位转换成一个10进制数的表示方式

任何0 - 255 间的数字都可以表示成8个0和1的组合

00000000 -> 0

00000001 -> 1

00000010 -> 2

00000011 -> 3

11111111 -> 255

2.IP地址的作用？

--表示一个网络节点的网络地址

--唯一标识网络中的一台主机

1. IP地址的分类？

A类 : 前8位作为网络位 , 第1位必须是0

B类 : 前16位作为网络位 , 前2位必须是10

C类 : 前24位作为网络位 , 前3位必须是110

D类 : 前4位必须是1110 , 用于多播(组播)

E类 : 前5位必须是11110 , 保留用于互联网研究

1. IP地址的版本

--ipv4

--ipv6

1. 私有IP地址:预留给企业内部使用

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

1. IP地址的表示

--如192.168.1.1 255.255.255.0 = 192.168.1.1/24

--” / ”后面的数字表示网络位的位数

1. 自动分配IP地址

--DHCP服务

1. 配置IP的4种方法

--临时配置IP:

#ifconfig 设备名 IP地址 子网掩码

--永久配置IP:

1 ) nmtui

2 ) 更改配置文件

/etc/sysconfig/network-scripts/ifcfg-设备名

3 ) # nmcli connection modify eth0 ipv4.method manual ipv4.addresses 176.19.2.200/24 autoconnect yes

# nmcli connection up eth0

配置IPv6地址

• IPv4 地址表示

– 32个二进制位,点分隔的十进制数

– 例如:172.25.0.11、127.0.0.1

• IPv6 地址表示

– 128个二进制位,冒号分隔 的十六进制数

– 每段内连续的前置 0 可省略、连续的多个 : 可简化为 ::

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

2003:ac18::305

nmcli connection modify 'System eth0'

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

connection.autoconnect yes

ping6 2003:ac18::305

小链接:

1. MAC地址: MAC地址是物理地址具有唯一性
2. 单播 : 一对一 ; 多播(组播) : 一对部分 ; 广播 : 一对全部

3.127.0.0.1 是本地环回地址 , 表示"我自己"的意思

4.0与255不能充当主机位,0代表网段,255代表广播地址

## 子网掩码:

1.子网掩码的作用？

--可以区分IP地址的网络部分和主机部分,二进制数模式下将网络部份全部写成1,主机部分全部写成0

1. 网络位的作用

--标识一个网络(网段)

小链接:

1. 网段(网络ID): 由二进制的IP地址与其对应的子网掩码同位相乘

## IP路由:

1. 什么是网关:

一个网络连接到下一个网络的关口,通常为一个路由器,实质为路由器连接局域网的端口的IP地址.

1. 网关的作用:

解决不同网络的通信

小链接:

1.采用交换机连接的网络是相同网络,采用路由器连接的网络是不同网络

## DNS服务器地址 :

1. 概述:DNS服务可以理解为网络查号台,它可以将好记的名字与不好记的IP地址进行匹配,解析
2. DNS服务器(域名服务器)的作用:

将在浏览器中输入的域名,解析为对应IP地址

# Linux基础命令

## 命令行基本概念:

1. 提示符的组成:

[root@pokemon ~]#

root:用户名

pokemon:主机名

~:当前工作路径

#:用户提示符,如果是root , 用户提示符是# , 如果是非root用户 , 用户提示符是$

1. 命令的基本格式

命令 [选项]... [参数]...

3.路径的表示方法？

--绝对路径 : /root/bin/pycharm

--相对路径 : bin/pycharm

## 基础命令介绍:

1. #ping:测试网络连通性

-c : 连接次数

--ping 127.0.0.1 测试自己的TCP/IP协议是否正常

--ping 网关地址 测试局域网通信是否正常

--ping 远程主机 测试外网通信

--ping [主机名] [-4/-6] 可以选择用ipv4或ipv6进行连接

--ping命令是双向的,有来有回

1. #ifconfig:显示或设置网络设备

语法：ifconfig [网络设备][down up -allmulti -arp -promisc][add<地址>][del<地址>][<硬件地址>] [media<网络媒介类型>][mem\_start<内存地址>][metric<数目>][mtu<字节>][netmask<子网掩码>][tunnel<地址>][-broadcast<地址>] [-pointopoint<地址>]

参数：

[网络设备] 网络设备的名称

　down 关闭指定的网络设备

　up 启动指定的网络设备

　-arp 打开或关闭指定接口上使用的ARP协议。前面加上一个负号用于关闭该选项

　-allmuti 关闭或启动指定接口的无区别模式。前面加上一个负号用于关闭该选项

　-promisc 关闭或启动指定网络设备的promiscuous模式。前面加上一个负号用于关闭该选项 www.2cto.com

　add<地址> 设置网络设备IPv6的IP地址

　del<地址> 删除网络设备IPv6的IP地址

　media<网络媒介类型> 设置网络设备的媒介类型

　mem\_start<内存地址> 设置网络设备在主内存所占用的起始地址

　metric<数目> 指定在计算数据包的转送次数时，所要加上的数目

　mtu<字节> 设置网络设备的MTU

　netmask<子网掩码> 设置网络设备的子网掩码

　tunnel<地址> 建立IPv4与IPv6之间的隧道通信地址

　-broadcast<地址> 将要送往指定地址的数据包当成广播数据包来处理

　-pointopoint<地址> 与指定地址的网络设备建立直接连线，此模式具有保密功能

--eth0:第一张网卡

--lo(localhost):回环接口

--virbr0(virtual):虚拟网卡

1. #whoami:显示当前用户
2. #hostname:显示主机名

5.#lscpu:查看CPU

1. #free:查看内存
2. #uname:查看内核版本
3. #w , #who , #users : 查看已登录用户
4. #history:查看历史命令

--调用历史命令: “!” + 命令编号

--清空历史命令: -c

--执行最近一次以某字段开头的历史命令: “!”+字段

--调整历史命令的数量 /etc/profile HISTSIZE=1000 //默认记录1000条

10.#kill:杀死进程中的程序

1. #ps:查看后台进程

--一般搭配-aux选项使用表示查询以用户为主的现行终端机下的所有进程的完整信息

1. #cd:切换目录

" . " : 代表当前目录

" .. " : 代表父目录

" - " : 返回之前所在目录

“~” : 家目录

1. #touch:创建空白文件
2. #cat:在终端页面查看选定文件内容

--显示行号: -n

--显示行号但不包括空白行: -b

1. #less:进入文件查看内容(不会在终端中产生内容)
2. #more:一次性查看
3. #vim:使用vim编辑器打开文件
4. #head:显示从顶端开始到指定行数的内容(默认10行)

--head -n 5 /root/123 = head -5 /root/123

1. #tail:显示从底端开始到指定行数的内容(默认10行)
2. #sed:规定行查看

--sed -n ‘开始行,结束行p’ 参数

--sed -n ‘指定行...p’ 参数

--sed -n ‘指定行p;开始行,结束行p’ 参数

1. #pwd:显示当前目录
2. #mkdir:创建目录

--递归创建选项:-p 如:mkdir -p 11/22/33

1. #ls:查看目录内容

-l:长格式显示

-h：以人类可以读取的方式显示

-d：只显示目录本身

-R：递归

-A: 显示隐藏文件

1. #mount:挂载

--mount /dev/cdrom zrb/

挂载的意义:通过mount挂载命令，让目录成为设备的访问点

--umount:卸载

-a:检测挂载 检测fstab内的内容并运行

1. #grep:输出包含指定字符的行

-i:忽略大小写

-v:反向查找

常用的匹配模式

– word 包含字符串word

– ^word 以字符串word开头

– word$ 以字符串word结尾

– ^$ 表示空行

1. #find:查找

– find [目录] [条件1] [-a|-o] [条件2] ...

– 常用条件表示:

-type 类型(f 文本文件、d 目录、l 快捷方式)

-name "文档名称"

-size +/-文件大小(k、M、G)

-user 用户名

-iname 同name但忽略大小写

-group 根据属组

-maxdepth 限制目录深度

-mtime 根据文件修改时间

+10 :查找10天之前创建或修改的文档

-10 :查找最近10天之内创建或修改的文档

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

1. #fdisk:磁盘分区

--p:打印分区表

--n:新建一个分区

--d:删除一个分区

--q:退出不保存

--w:把分区写进分区表,保存并退出

1. #lsblk:查看磁盘分区
2. #mkfs:格式化

mkfs.格式 + 硬盘路径

1. #df:显示正在挂载的

1. #partprobe:刷新分区表
2. #scp:远程文件复制

--scp /root/mytxt.txt 192.168.4.128:/root/

1. #ip a s:查看网络配置
2. #nmcli:网络配置

nmcli connection show + 端口名

nmcli connection show : 查看命令识别的网卡名

nmcli connection modify 'System eth0' 选择变更目标

ipv4.method manual 改变ipv4的IP模式改为手动

ipv4.addresses '172.25.0.110/24 172.25.0.254' 添加IP地址

Ipv4.dns 172.25.254.254 添加DNS

connection.autoconnect yes 每次开机自动启用本网卡所有配置

nmcli connection up ‘System eth0’ : 更新

1. #reboot:重启命令
2. #poweroff:关机
3. #hostnamectl : 查看主机名

修改静态主机名可以实现永久修改主机名

静态、瞬态或灵活主机名，分别使用“--static”，“--transient”或“--pretty”选项

set-hostname选项可以更改主机名

1. #useradd :创建用户

-u :指定UID

-g :指定GID

-d :指定家目录

-G :指定附加组

-s :指定解释器

/sbin/nologin :禁止用户登录的解释器

1. #id:查看用户
2. #su:切换用户

su - 用户名

1. #usermod:改变用户属性
2. #userdel:删除用户

-r : 删除用户信息并删除家目录

1. #passwd 更改用户密码

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

1. #groupadd : 添加组

-g指定组ID

1. #gpasswd:管理用户至组

-a:添加

-d:删除

1. #groupmod:修改组属性

-g:组ID

-n:组名

1. #groupdel:删除组
2. #rm:删除

-r:递归

-f:强制

1. #cp:复制

-r:递归

经常与” . ”联用代表复制到当前

强制覆盖: 在命令前加” \ ”临时取消别名

多参数的的拷贝:永远把最后作为目标

拷贝是可以同时改名的

1. #mv:移动

重命名用法: mv /opt/1.txt /opt/2.txt 将原文件路径不变,改变移动后的文件名就可以做到重命名

1. #alias:别名

ailas [别名名称] : 查看现有别名

ailas 别名=’实际执行命令’ 添加别名

unailas 别名 删除别名

1. #crontab:时程表

f1 f2 f3 f4 f5 program

minute hour day month week command

minute： 表示分钟，可以是从0到59之间的任何整数。

hour：表示小时，可以是从0到23之间的任何整数。

day：表示日期，可以是从1到31之间的任何整数。

month：表示月份，可以是从1到12之间的任何整数。

week：表示星期几，可以是从0到7之间的任何整数，这里的0或7代表星期日。

command：要执行的命令，可以是系统命令，也可以是自己编写的脚本文件。

crontab 选项 [-n 用户名]

-e:编辑

-l:查看

-r:删除

如 : \*/3 \* \* \* \* /root/zrb/zrbshell/backg.sh 每3分钟运行一次backg.sh

1. #wget:

1. #nslookup:DNS测试

nslookup + 域名

1. #echo:输出

-e:处理特殊字符

1. #date:查看时间

+%Y :年份

+%m :月份

+%d :日

+%F :年月日

+%a :短星期

+%A :完整星期

+%b :完整月份

+%B :中文月份

-s ‘年-月-日 时:分:秒’ :修改时间

1. #man:帮助文档

5:配置文件帮助

/:代表即将进行搜索

/-h:搜索的内容为“-h”

按“N”：向上匹配

按“n”：向下匹配

按“q”：退出

1. #tar:备份和恢复

常见压缩格式及工具:

.gz ---> gzip

.bz2 ---> bzip2

.xz ---> xz

选项:

-c:创建归档

-x:释放归档

-f:指定tar包的名字(必须在所有选项的后面)

-z , -j , -J:调用.gz , .bz2 , .xz格式的工具

-t:显示归档的文件清单

-C:选定释放路径

压缩 : tar 选项 压缩包的名字 被压缩的文档...

释放 : tar 选项 压缩包的名字 -C 释放的路径

1. #zip:归档工具(跨平台)

zip [-r] 备份文件.zip 被归档的文档...

1. #unzip:释放归档

unzip 备份文件.zip [-d 目标文件夹]

1. #systemctl:服务
2. #bc:计算器
3. #chmod:访问权限

-R:递归

r = 4 w = 2 x = 1

1. #chown:所属权限

-R:递归

chown 属主 : 属组 修改对象

chown 属主 修改对象

chown : 属组 修改对象

1. #setfacl:设置acl

– setfacl + 选项 + u/g:用户名:权限类别 + 文档...

-m:修改指定的acl

-x:删除指定的acl

-b:清空acl

1. #getfacl:查看acl
2. #showmount:查看NFS资源

showmount -e [服务器地址]

1. #du:统计文件的占用空间

– du [选项]... [目录或文件]...

– -s:只统计每个参数所占用的总空间大小

– -h:提供易读容量单位(K、M等)

1. # ln:创建快捷方式

创建软连接

--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,支持目录

创建硬链接

--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

硬链接与原始文件必须在同一分区/文件系统,不支持目录

小链接

1. 显示提示选项 : -i 一般系统会自己定义别名
2. date的起始年份是1970-1-1

## 快捷键:

1. Tab:自动补全

--有唯一匹配值时,单击补全字符串

--匹配值不为一时,双击列出所有可能性

1. alt+[ . ] / esc+[ . ]:调用上一个命令的参数
2. 上下键:调用上/下一个命令行
3. ctrl + c:强行中断正在执行的命令
4. ctrl + shift +c/v:复制/粘贴
5. 鼠标中键:复制粘贴选中的内容
6. ctrl + u:将光标前的内容删除
7. ctrl + k:将光标后包括光标的内容删除
8. ctrl + w:以空格为界将光标前的内容删除
9. ctrl + l:清屏
10. ctrl + shift + ‘=’:放大终端
11. ctrl + ‘-’ 缩小终端
12. ctrl + shift + t:以标签的形式开启终端,用alt + 数字切换
13. ctrl + shift + n:以窗口的形式开启终端,用alt + tab切换
14. ctrl + shift + w/d:关闭正在使用的终端
15. ctrl + shift + q:以窗口为单位关闭终端
16. ctrl + r :检索敲过的命令

## 通配符:

1. \* : 任意多个字符

2.? : 单个字符

1. [] : 多个字符或连续性的一个
2. {} : 多组不同的字符串,全匹配

连续批量在[n-m]不能使用时,用{n..m} 如mkdir 连续的文件夹时

## 特殊符号:

1. > : 覆盖重定向
2. >> : 追加重定向
3. | : 管道 (将前一条命令的结果作为后一条命令的参数)
4. \a : 发出警告声
5. \b : 删除前一个字符
6. \c : 最后不加上换行符号
7. \f : 换行但光标仍旧停留在原来的位置
8. \n : 换行且光标移至行首

9. \r : 光标移至行首，但不换行

10. \t : 插入tab

11. \v : 与\f相同

1. \\ : 插入\字符

13. \nnn : 插入nnn（八进制）所代表的ASCII字符

# 网络部分

## 网络拓扑结构

1.总线型

--所有计算机设备连接到共用的一根线缆上，可靠性不高

--多节点通信容易冲突，目前基本已弃用

2.星型

--所有计算机设备通过线缆连接到集中的设备上，目前主流的拓扑

--方便实现、容易扩展，但集中设备负载重，容易出现瓶颈

3.网状

--每个计算机设备通过线缆连接到多个其他设备，可靠性高

--适用于冗余/容错要求高的场合，但组网成本高

1. 环型

5.点对点

## 常见网络设备

1.交换机

用于组建星型网络，将多台计算机/交换机连接到一起

交换机的背板带宽可承载所有计算机同时通信

2.路由器

用于组建星型网络或网状网络，可以连接多台交换机或路由器

从而实现多个小网络组合成更大规模的网络

3.交换&路由一体机

通常为无线宽带路由器，适用于家庭或小型办公环境

设备接入ISP运营商网络，为其他终端设备提供有线/无线接入

## 交换机工作原理

1.学习

MAC地址表是交换机通过学习接收的数据帧的源MAC地址来形成的

2.广播

如果目标地址在MAC地址表中没有，交换机就向除接收到该数据帧的端口外的其他所有端口广播该数据帧

3.转发

交换机根据MAC地址表单播转发数据帧

4.更新

交换机MAC地址表的老化时间是300秒

交换机如果发现一个帧的入端口和MAC地址表中源MAC地址的所在端口不同，交换机将MAC 地址重新学习到新的端口

## 交换机的命令模式

1.配置前的连接

Console电缆

物理连接

计算机COM口

交换机路由器Console口

软件连接

1. Cisco交换机的命令行

用户模式

switch>

特权模式(一般用于查看配置信息)

switch>enable

switch#

全局配置模式(所做的配置对整个设备生效)

switch#configure terminal

switch(config)#

端口模式

switch(config)# interface f0/1

switch(config-if)#

模式间的转换

退回到上一个模式 exit

退到特权模式: end <ctrl+z>

强制终止 : <Ctrl+shift+^>

配置主机名

switch(config)#hostname pokemon

pokemon#

查看交换机配置

pokemon# show running-config

显示版本信息

pokemon# show version

配置enable明文口令

pokemon>enable

pokemon#configure t

pokemon(config)#enable password 7852398521

pokemon(config)#exit

pokemon#show run

pokemon#configure t

pokemon(config)#no enable password

配置加密口令

pokemon>enable

pokemon#configure t

pokemon(config)#enable secret 7852398521

pokemon(config)#exit

pokemon#show run

pokemon#configure t

pokemon(config)#no enable secret

配置Console口令

pokemon>enable

pokemon#configure t

pokemon(config)#line console 0

pokemon(config-line)#password 789

pokemon(config-line)#login

pokemon(config-line)#end

pokemon#show run

pokemon#exit

旗标

可以让每个登录你设备的人收到你设置的信息

pokemon>enable

pokemon#configure t

pokemon(config)#banner motd #

输入信息以#号结尾

## 网络线缆

1.双绞线概述

目前使用最普遍、性价比最优的一种传输介质

由8根铜芯线组成，分成4对两两绞合组成

包括UTP（无屏蔽双绞线）、STP（屏蔽双绞线）

2.双绞线制作规范

T568b标准：白橙、橙、白绿、蓝、白蓝、绿、白棕、棕 1、3发送 2、6接收

T568a标准：白绿、绿、白橙、蓝、白蓝、橙、白棕、棕

直通线：两边均是568b或者568a 不同设备相连

交叉线：一边568a一边568b 相同设备相连

全反线：一边568b，另一边完全相反（Console）

小链接

1. 单工、半双工与全双工

--单工

只有一个信道，传输方向只能是单向的

--半双工

只有一个信道，在同一时刻，只能是单向传输

--全双工

双信道，同时可以有双向数据传输

1. 主机与路由器用交叉 交换机之间直通交叉都可以

## 配置聚合连接(链路聚合 网卡组队 网卡绑定)

采用虚拟网卡设计,对网卡设备的备份

热备份(activebackup)连接冗余

1.创建虚拟网卡team0 参考: man teamd.conf #全文搜索/example

nmcli connection add type team

ifname team0 con-name team0 autoconnect yes

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

nmcli connection 添加 类型 team(组队)

网卡名 team0 配置文件名 team0 每次开机自动启用

配置运行模式 热备份模式

ifconfig #查看team0网卡信息

cat /et(tab)/sysco(tab)/netw(tab)-(tab)/ifcfg-team0

生成的网卡配置文件信息

如果有敲错，一定要删除 : nmcli connection delete team0

1. 添加成员

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

nmcli connection 添加 类型为 team的成员 配置文件名 team0-1 网卡为 eth1 主设备为 team0

3.配置team0的IP地址

查看team0信息的指令 teamdctl team0 state

# 各种服务

## 服务的概念

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

服务搭建的步骤

1. 装包
2. 配置
3. 起服务

服务的种类及端口号:

-ftp : 21

-ssh : 22

-Telnet : 23

-SMTP : 25

-DNS : 53

-DHCP/UDP : 67/68

-tftp : 69

-http : 80

-NTP : 123

-https : 443

-pop3 :110

-SNMP:161

-SMB:139

-CIFS:445

-NFS:2049

-RPC:111

## ftp(文件传输协议)服务搭建:

( 1 ) 连接主机网络 : 在虚拟网络接口出连接主机网络,为虚拟机添加IP地址

( 2 ) 在CDROM中加载光盘 用mount命令将光盘挂载到自定义目录

( 3 ) 用rpm -qa | grep vsftpd 查看是否安装 vsftpd 如果没有则在挂载目录里的Packages文件夹中用 rpm -ivh vsftpd-3.0.2-22.el7.x86\_64.rpm 来安装

( 4 ) 安装完毕后在/etc目录中会生成vsftpd目录,该目录中有三大配置文件,编辑vsftpd.conf来使得用户可以匿名登录

( 5 ) 用systemctl start vsftpd 启动vsftpd服务

( 6 ) 关闭防火墙 iptsbles -F

## HTTP(超文本传输协议)服务搭建:

/var/www/html

DNS:域名解析协议

## NTP(网络时间协议)服务:

• Network Time Protocol

– NTP服务器为客户机提供标准时间

– NTP客户机需要与NTP服务器保持沟通

NTP客户机:

1.安装chrony软件

2.修改主配置文件,指定NTP服务

Server 172.25.254.254 iburst 指定服务端

3.重启服务

systemctl restart chronyd

systemctl enable chronyd 开机自启服务

## LDAP服务:

什么是LDAP?

轻量级目录访问协议

– Lightweight Directory Access Protocol

– 由服务器来集中存储并向客户端提供的信息,存储方

式类似于DNS分层结构

– 提供的信息包括:用户名、密码、通信录、主机名映射记录、......

用户的集中管理

LDAP用户:网络用户,由网络中LDAP服务器统一提供用户信息

LDAP服务器:

LDAP客户机:

1. 安装sssd软件
2. 安装图形工具authconfig-gtk进行配置

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

1. 重起sssd服务
2. 验证:

## Samba服务基础:跨平台的共享 Windows与Linux

• 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

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

• 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 与Windows的guest来宾用户有关

; browseable = yes|no //默认yes 与隐藏共享有关

; read only = yes|no //默认yes

; write list = 用户1 .. .. //默认无

; valid users = 用户1 .. .. //默认为所有Samba共享帐号

; hosts allow = 客户机地址 .. ..

; hosts deny = 客户机地址 .. ..

服务器工作组的配置

workgroup = STAFF

服务机:

1. 安装软件samba
2. 创建共享帐号
3. 创建共享目录
4. 修改配置文件/etc/samba/smb.conf
5. 重起smb服务,设置为开机自起服务
6. 修改SELinux策略,布尔值(服务功能开和关)

需要加 -P 选项才能实现永久设置,足够内存

setsebool + 修改的配置

getsebool -a :查看全部布尔型策略

1. 给予用户权限

客户机

1. 安装客户端软件:samba-client
2. 使用客户端软件提供的命令,访问服务端共享

smbclient -L 172.25.0.11

smbclient -U harry //172.25.0.11/common

客户端更加方便的访问:

1. 安装cifs-utils,支持cifs协议及相应文件系统
2. 进行挂载

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

1. 开机自动挂载:修改配置/etc/fstab

\_netdev:声明网络设备

在挂载该设备时,需要配置本机IP地址及网络服务正常才能挂载本设备

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

客户端访问服务端

1.防火墙

2.安全SELinux

3.服务本身的访问控制

4.本地目录权限

总结:读写Samba共享的搭建

服务端:

1.设置防火墙默认区域为trusted

2.安装软件包samba

3.常见共享用户的帐号 pdbedit

4.创建共享目录

5.修改配置文件/etc/samba/smb.conf发布共享

指明 write list = 可写用户

6.重起smb服务

7.开放SELinux读写布尔值

8.设置本地目录的权限

客户端:

1.设置防火墙默认区域为trusted

2.安装软件cifs-utils

3.挂载访问

1. .实现开机自动挂载

了解内容:multiuser机制

• SMB客户端的 multiuser 挂载技术

– 管理员只需要作一次挂载

– 客户端在访问挂载点时,若需要不同权限,可以临时

切换为新的共享用户身份(无需重新挂载)

• mount.cifs 的挂载参数

– 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

cifscreds add/update -u 共享用户名 服务器地址

## NFS共享服务:

什么是NFS共享

Network File System,网络文件系统

– 由NFS服务器将指定的文件夹共享给客户机

– 客户机将此共享目录 mount 到本地目录,访问此共享

资源就像访问本地目录一样方便

– 类似于 EXT4、XFS等类型,只不过资源在网上

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

• 所需软件包:nfs-utils

• 系统服务:nfs-server

NFS服务器:

1. 安装nfs-utils软件
2. 创建共享目录
3. 修改配置文件/etc/exports 发布共享

文件夹路径 客户机地址(权限)

/nfs \*(ro/rw)

1. 重起服务nfs-server

NFS客户机

1. 用showmount -e + 服务器地址 查看共享目录
2. mount挂载共享目录

– mount 服务器地址:目录路径 本地挂载点

完成开机自动挂载/etc/fstab

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

## iSCSI网络磁盘:

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

• backstore,后端存储

– 对应到服务端提供实际存储空间的设备,需要起一个管理名称

• target,磁盘组

– 是客户端的访问目标,作为一个框架,由多个lun组成

• lun,逻辑单元

– 每一个lun需要关联到某一个后端存储设备,在客户端

会视为一块虚拟硬盘

• ISCSI Qualified Name 名称规范(iqn命名规范)

– 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

服务端

一 划分一个主分区

二 安装软件targetcli

三 配置共享存储

1.创建后端存储,为其设置名字为nsd(购买大冰箱,进行纸制的包装)

2.创建target磁盘组 (打造的木质箱子)

3.Lun关联( 将后端存储 放入 target磁盘组 )

4. 配置ACL访问控制

5.启用网络接口及端口号

四 重起服务target

客户端

1. 安装软件包 iscsi-initiator-utils
2. 修改配置文件,指定客户端声称的名字

3.重起iscsid服务,刷新客户端声称的名字

4.发现服务端共享存储

5.重起客户端服务,识别加载共享存储

总结:

服务端:

1.设置防火墙默认区域为trusted

2.安装软件包targetcli

3.运行targetcli命令执行配置

a:创建后端存储

b:创建磁盘组

c:进行关联lun

d:配置ACL访问控制,客户端声称的名字

e:启用网络接口,启用端口

4.重起服务target

客户端:

1.设置防火墙默认区域为trusted

2.安装软件包 iscsi-initiator-utils

3.修改配置文件,指定客户端声称的名字

4.重起iscsid刷新客户端声称的名字

5.执行命令发现服务端共享存储

6.重起客户端服务iscsi,识别加载共享存储

## cron计划任务:

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

‘ \* ‘ : 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

在/var/spool/cron/目录下会生成任务文件

## Web服务:

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,超文本标记语言(html)

• Hyper Text Transfer Protocol,超文本传输协议(http)默认端口:80

• 软件包:httpd

• 系统服务:httpd

• 提供的默认配置

– Listen:监听地址:端口(80)

– ServerName:本站点注册的DNS名称(空缺)

– DocumentRoot:网页根目录(/var/www/html)

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

一 搭建基本的Web服务

1. 安装软件httpd

2.书写一页面文件

3.重起httpd服务

4.验证

DocumentRoot:设置网页文件存放的根目录(/var/www/html)

1.建立新网页存放路径 mkdir /var/www/myweb

2.修改配置文件 DocumentRoot "/var/www/myweb"

3.重起httpd服务

4.验证

LAMP: Linux Apache MySQL/MariaDB PHP/Python

LNMP: Linux Nginx MySQL/MariaDB PHP/Python

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(构建方式)

– 基于域名的虚拟主机

– 基于端口的虚拟主机

– 基于IP地址的虚拟主机

• 配置文件路径

– /etc/httpd/conf/httpd.conf #主配置文件

– /etc/httpd/conf.d/\*.conf #调用配置文件

• 为每个虚拟站点添加配置

<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>

虚拟Web主机,会由上及下 进行匹配,写在上面的优先级高

一旦使用了虚拟Web主机的功能,所有的站点都必须,要使用虚拟web主机功能呈现

httpd 运行身份(用户/组):apache

客户机地址限制

• 使用 <Directory> 配置区段

– 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>

使用自定Web根目录

1)新建目录 s写入页面文件

2)配置/webroot作为此站点新的网页文件根目录

3)修改SELinux策略,安全上下文 (标签)

semanage fcontext -l 查看安全上下文

• 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

• 方式2:将新目录增加到预设的标准web目录范围

– semanage fcontext -a -t httpd\_sys\_content\_t ‘新目录(/.\*)?’

Restorecon -R 新目录

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

– 文本(txt/html)、图片(jpg/png)等静态资源

动态网站的运行

• 服务端的原始网页 ≠ 浏览器访问到的网页

– 由Web服务软件接受请求,动态程序转后端模块处理

– PHP网页、Python网页、JSP网页......

部署动态Web

1.为站点 webapp0.example.com 配置提供动态Web内容

2.方便客户端用户访问动态页面,页面跳转(页面别名 地址重写)

配置: Alias 网络路径 实际服务器本地路径

当客户端访问 网络路径 ,页面跳转到 实际服务器本地路径

Example:

Alias /nsd /var/www/html/abc.html

当客户端访问 浏览器输入的路径为/nsd ,页面跳转到/var/www/html/abc.html

Alias / /var/www/qq/webinfo.wsgi

当客户端访问 浏览器输入的路径为webapp0.example.com,页面跳转到/var/www/qq/webinfo.wsgi

3.安装软件mod\_wsgi,翻译Python页面的代码

WsgiScriptAlias / /var/www/qq/webinfo.wsgi

4.此虚拟主机侦听在端口8909

Listen 8909

1. 修改SELinux策略, 非默认端口开放
2. semanage port -a -t http\_port\_t -p tcp 8909

-a:添加 -t:类型 -p:协议

Web服务

端口的优先级,优于域名,也优于IP地址

安全Web服务

https:安全的超文本传输协议 443

– 公钥:主要用来加密数据

– 私钥:主要用来解密数据(与相应的公钥匹配)

– 数字证书:证明拥有者的合法性/权威性(单位名称、

有效期、公钥、颁发机构及签名、......)

– Certificate Authority,数字证书授权中心:负责证书

的申请/审核/颁发/鉴定/撤销等管理工作

部署Web服务器

1.安装httpd软件

2.部署网站证书(营业执照) cd /etc/pki/tls/certs/

3.部署根证书(工商局信息) cd /etc/pki/tls/certs/

4.部署解密的私钥 cd /etc/pki/tls/private/

5.安装mod\_ssl软件,支持安全加密的Web软件

6.修改配置文件 /etc/httpd/conf.d/ssl.conf

59 DocumentRoot "/var/www/html"

60 ServerName www0.example.com:443

#指定网站证书

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

#指定解密的私钥匙文件

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

#指定根证书(工商局信息)

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

7.重起httpd服务

8.测试验证

## DNS解析服务:

• 为什么需要DNS系统

– www.baidu.com 与 119.75.217.56,哪个更好记?

– 互联网中的114查号台/导航员

• 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

所有的域名必须以点结尾 www.baidu.com.

根域名: .

一级域名: .cn .us .hk .tw .jp .kr ........

二级域名: .com.cn .edu.cn .org.cn .net.cn .........

三级域名: nb.com.cn haha.com.cn xixi.com.cn ......

完整合格的主机名: www.nb.com.cn ftp.nb.com.cn

完整合格的主机名: 主机头部+域名

• Full Qualified Domain Name(FQDN),完全合格主机名

– = 站点名.域名后缀

– = 站点名. .. .. .二级域.一级域

BIND域名服务

• BIND(Berkeley Internet Name Daemon)

– 伯克利 Internet 域名服务

– 官方站点:https://www.isc.org/

[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)

– 系统服务:named

– DNS默认端口:TCP/UDP 53

– 主配置文件:/etc/named.conf #设置本机负责解析的域名

– 地址库文件:/var/named/ #主机名与IP地址对应关系

1.安装软件包

2.修改主配置文件 /etc/named.conf

options {

directory "/var/named"; #指定地址库文件存放路径

};

zone "tedu.cn" IN { #指定负责解析的域名为tedu.cn

type master; #指定本机为权威服务器

file "tedu.cn.zone"; #指定地址库文件名称

};

3.建立地址库文件 /var/named/tedu.cn.zone #保证named用户有读权限

#所有域名都必须以点结尾

#不以点结尾,自动补全本地址库文件负责的域名

tedu.cn. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.1

ftp A 192.168.4.2

4.重起named服务

构建多区域的DNS服务器

1.修改主配置文件

vim /etc/named.conf

options {

directory "/var/named";

};

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "qq.com" IN {

type master;

file "qq.com.zone";

};

2.建立地址库文件/var/named/qq.com.zone

vim /var/named/qq.com.zone

qq.com. NS svr7

svr7 A 192.168.4.7

www A 1.2.3.4

1. 重起named服务

特殊的解析记录

1.基于DNS的轮询

[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 192.168.4.25

2.泛域名解析

[root@svr7 /]# vim /var/named/tedu.cn.zone

tedu.cn. NS svr7

svr7 A 192.168.4.7

tedu.cn. A 10.20.30.40

\* A 11.12.13.14

3.有规律的泛域名解析

pc1.tedu.cn ----> 192.168.10.1

pc2.tedu.cn ----> 192.168.10.2

pc3.tedu.cn ----> 192.168.10.3

........

pc50.tedu.cn ----> 192.168.10.50

内置函数: $GENERATE 起始点-结束点

[root@svr7 /]# vim /var/named/tedu.cn.zone

$GENERATE 1-50 pc$ A 192.168.10.$

4.解析记录的别名

tts CNAME ftp

tts解析结果与ftp一致

DNS子域授权

父域: www.qq.com

子域: www.bj.qq.com

父域的qq.com 由虚拟机A解析

子域的bj.qq.com 由虚拟机B解析

子域构建DNS服务器

1.安装软件包

2.修改主配置文件

options {

directory "/var/named";

};

zone "bj.qq.com" IN {

type master;

file "bj.qq.com.zone";

};

3.建立地址库文件

4.重起named服务

5.测试解析

子域授权:

让父域的DNS服务器,可以解析子域DNS负责的域名

修改地址库文件

[root@svr7 /]# vim /var/named/qq.com.zone

qq.com. NS svr7

bj.qq.com. NS pc207

svr7 A 192.168.4.7

pc207 A 192.168.4.207

www A 1.2.3.4

Non-authoritative answer: #非权威解答

递归解析:

将首选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:

将首选DNS服务器,与其他DNS服务器交互

recursion no; #禁止递归解析

缓存DNS服务器:缓存解析记录,加快解析

在真机上操作:

2.查看真机DNS服务器

3.真机修改配置文件

options {

directory "/var/named";

forwarders { 172.40.1.10; }; #转发给172.40.1.10

};

Split分离解析(视图解析)

• 当收到客户机的DNS查询请求的时候

– 能够区分客户机的来源地址

– 为不同类别的客户机提供不同的解析结果(IP地址)

– 针对不同的客户机就近提供服务器

BIND的view视图

• 根据源地址集合将客户机分类

– 不同客户机获得不同结果(待遇有差别)

– 由上及下进行匹配,匹配及停止

– 分类要合理,所有的客户端都要找到自己的分类

– 所有的zone都要在view中

view "a" {

match-clients { 192.168.4.1; }

zone "12306.cn" IN {

12306.cn.zone;

};

};

view "b" {

match-clients { 192.168.4.2; }

zone "12306.cn" IN {

12306.cn.other;

};

};

view "c" {

match-clients { any; }

zone "12306.cn" IN {

12306.cn.any;

};

};

acl定义变量abc

acl abc { 192.168.4.207; 192.168.7.0/24; };

view "nsd" {

match-clients { abc; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "qq.com" IN {

type master;

file "qq.com.zone";

};

};

## DHCP服务:

概述及原理

主要分配一下入网参数

--IP地址/子网掩码/广播地址

--默认网关地址,DNS服务器地址

--PXE引导设置(TFTP服务器地址,引导文件名)

DHCP地址分配的四次会话(广播的形式)

-- DISCOVERY --> OFFER --> REQUEST --> ACK

pc广播要IP 录用 回应 确认有无重复IP

一个网络中只能有一台DHCP

服务端基本概念

--租期:允许客户机租用IP地址的时间期限,单位为秒

--作用域:分配给客户机的IP地址所在的网段

--地址池:用来动态分配的IP地址的范围

1. 安装软件包:dhcp
2. 修改配置文件 /etc/dhcp/dhcpd.conf

读入 /usr/share/doc/dhcp\*/dhcpd.conf.example 文件

subnet 192.168.4.0 netmask 255.255.255.0 {

range 192.168.4.20 192.168.4.100;

option domain-name-servers 192.168.4.7;

option routers 192.168.4.254;

default-lease-time 600;

max-lease-time 7200;

}

subnet : 网段

netmask : 子网掩码

range : 地址池

option domain-name-servers : 给别人配置的DNS

option routers : 网关

default-lease-time : 默认租期

max-lease-time : 最大租期

1. 起服务

## 基础邮件服务:

• 电子邮件服务器的基本功能

– 为用户提供电子邮箱存储空间(用户名@邮件域名)

– 处理用户发出的邮件 —— 传递给收件服务器

– 处理用户收到的邮件 —— 投递到邮箱

SMTP:用户发邮件协议 默认端口 25

pop3:用户收邮件协议 默认端口 110

搭建邮件服务器

1.安装软件包postfix

2.修改主配置文件 /etc/postfix/main.cf

99 myorigin = server0.example.com #收件人与发件人补全的域名后缀

116 inet\_interfaces = all #本机所有网络接口都提供邮件功能

164 mydestination = server0.example.com #判定收件人域名后缀 为本域邮件

destination:目标

1. 重起服务postfix

创建普通用户测试邮件收发

使用mail命令发信/收信

• 发信操作

– mail -s '邮件标题' -r 发件人 收件人

• 收信操作

– mail [-u 用户名]

# PXE网络装机

## 网络装机的优势

规模化:同时装配多台主机

自动化:装系统,配置各种服务

远程实现:不需要光盘,U盘等物理安装介质

PXE: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 集成在网卡的芯片里

当计算机引导时,从网卡芯片中把PXE client调入内存执行,获取PXE server配置,显示菜单,根据用户选择将远程引导程序下载到本机运行

需要那些服务组件

DHCP服务,分配IP地址,定位引导程序

TFTP服务,提供引导下载(众多的引导文件)

HTTP服务,提供yum源

部署网络装机服务器

## 一 部署DHCP服务

1. 修改配置文件

在括号中添加next-server 192.168.4.7; 指定下一个服务的IP地址

filename “pxelinux.0”; 指定网卡引导文件的名称

pxelinux.0:网卡引导文件(网络安装说明书) 二进制文件,安装一个软件产生

## 二 搭建TFTP服务

简单的文件传输协议,默认端口69

默认共享路径: /var/lib/tftpboot

1. 安装软件包tftp-server
2. 重启服务
3. 部署pxelinux.0文件

用yum provides 选项来进行查询pxelinux.0的所属包

provides(提供) 查询某个命令或文件是哪个包提供的

复制pxelinux.0到tftp的共享文件夹

1. 部署菜单文件

从光盘下的isolinux文件夹里拷贝pxelinux.cfg到共享路径下新建的pxelinux.cfg文件夹下

1. 部署图形模块

将图形支持文件 vesamenu.c32 和 背景图片 splash.png 拷贝到共享路径

1. 部署启动内核和驱动程序

将启动内核文件 vmlinuz 和 启动所需的驱动程序 initrd.img 拷贝到共享路径

7.修改菜单文件

## 三 搭建http服务,共享光盘全部内容

1.安装httpd包,启动服务

2.创建挂载目录,进行挂载光驱设备

将光盘文件放入http默认的文件夹下

## 四 部署无人值守安装 ,生成应答文件

1.图形的工具进行生成 system-config-kickstart

2.运行进行配置

首先查看 "软件包选择" 是否可以使用

需要Yum仓库支持,必须要求 光盘的仓库标识为 [development]

3.查看root目录下ks文件生成

4.利用Web服务共享应答文件

5.修改菜单文件,指定ks应答文件位置

ks=http://192.168.4.7/ks.cfg

总结:

1. DHCP--->IP地址 next-server filename

2. tftp---> pxelinux.0

3. pxelinux.0--->读取菜单文件

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

4. default---->

vesamenu.c32 splash.png vmlinuz initrd.img

ks=http://192.168.4.7/ks.cfg

5. ks文件----> 时区 密码 分区 .......

--url="http://192.168.4.7/rhel7"

# Cobbler装机平台

不同版本的多系统的安装

一 安装一个CentOS虚拟机

虚拟机要求:

1.硬盘大小50G

2.网络类型private1

3.软件包选择 "带GUI的服务器"

4.分区选择 "自动分区"

5.管理员root密码为 1

6.创建普通用户lisi

7.将CentOS放入光驱设备,搭建本地Yum仓库

8.配置IP地址:192.168.4.66/24

9.配置主机名:Cobbler.tedu.cn

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

1. 利用scp真机传递Cobbler.zip包到虚拟机192.168.4.66中
2. 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

# 数据库

## 数据库服务基础

什么是数据库:存放数据的仓库

数据库中有很多的库,在每一个库中,会有很多的表格

• 常见的关系型 数据库管理系统

– 微软的 SQL Server

– IBM的 DB2

– 甲骨文的 Oracle、MySQL

– 社区开源版 MariaDB

一 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

– mariadb:提供客户端及管理工具

二 数据库的基本操作

mysql 进入数据库

show databases 显示所有的库

create database 库名 创建库

drop database 库名 删除库

use 进入库

show tables 查看表

desc 表名 查看表结构

• 为数据库账号修改密码

数据库管理员:账户名 root 信息放在mysql库中,user表

系统管理员:账户名 root 信息放在/etc/passwd

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

• 禁止监听网络,只服务于本机

/etc/my.cnf 数据库主配置文件

skip-networking //跳过网络监听

• 导入/恢复到数据库

mysql [-u用户名] [-p[密码]] 数据库名 < 备份文件.sql

用户授权设置

授权命令:

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';

# Linux目录含义

目录结构

• 认识Linux的目录层次:

– man hier

## / (根目录) :

## /boot:

存放系统引导必须的文件,包括内核,自动配置

## /bin /sbin:

存放各种命令

## /dev :

存放设备相关的数据(硬盘 键盘 鼠标 显示器)

## /root /home:

家目录

## /var:

存放日志文件,邮箱目录等经常变化的文件

## /proc:

用于存放内存中的映射数据,不占用磁盘

1. cpu信息文件 : cpuinfo
2. 内存信息文件 : meminfo

## /tmp:

存放系统运行过程中使用的一些临时文件

## /etc:

用于存放配置文件

1. 版本信息文件 : redhat-release
2. 自动挂载文件 : fstab

设备路径 挂载点 类型 参数 备份标记 检测顺序

UUID=3a32272f-5b5c-4c49-b203-7307de2df412 /mypart2 xfs defaults 0 1

1. 主机名:hostname
2. 用户基本信息:passwd

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

用户名:密码占位符:UID:基本组的GID值:用户描述信息:用户的家目录:解释器

1. DNS服务器地址:resolv.conf

nameserver 176.19.8.254

1. 用户的密码信息:shadow

root :$6$UiGI4Tc2$ht...KAmf1:16261:0:99999:7:::

用户名:密码加密字符串 :上一次修改密码的时间

1. 组的基本信息:group

adminuser :x :1507 :

组名 :组的密码占位符:组的GID:组的成员列表

8.组的密码信息:gshadow

1. 主机名映射文件:hosts

作用:只为本机直接提供解析结果

# 安装软件包

## RPM Package Manager, RPM包管理器:

--rpm -q 查看是否已经安装某个包

限制 : 参数必须完全匹配

--rpm -qa 显示所有安装的包

配合 | grep

--rpm -ivh

-i 安装rpm包

-v 显示安装过程

-h 显示哈希标记

--rpm -e 卸载软件包

小链接:

1. 升级内核

-下载新的内核软件包kernel

-安装新的内核软件包

-重启生效

## Yellow Dog Update Manager -- 用来自动解决依赖关系

1.修改配置文件

/etc/yum.repos.d/xxx.repo

[rhel7]

name=redhat7.4

baseurl=ftp://192.168.4.1/rhel7.4 或 file:///var/ftp/rhel7

enabled=1

gpgcheck=0

1. 验证配置是否正确

--yum repolist

清除缓存

--yum clean all

1. 列举软件仓库中的文件

--yum list [软件名]...

和rpm管理器一样,必须全匹配时才能列出包文件,否则报错

--yum list | grep 关键字

1. 安装软件

--yum [-y] install 软件名...

1. 卸载软件

--yum [-y] remove 软件名...

## 源码装包:

源码包----开发工具gcc与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

一 准备源码包

二 源码包编译安装

步骤1:安装开发工具gcc 和 make

步骤2: tar解包,释放源代码至指定目录

步骤3: ./configure 配置,指定安装目录/功能模块等选项

检测gcc是否安装

--prefix=指定安装路径

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

# 硬件相关

## 硬盘分区:

1. 分区类型：主分区、扩展分区、逻辑分区

2.传统的分区方式

MBR 主引导记录模式

最大支持的空间2.2Tb

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)

扩展分区不能格式化,不能存储数据

GPT: 最多可以划分128个主分区

最大空间支持到 18EB 容量

1TB=1000GB

1PB=1000TB

1EB=1000PB

3.名称规则

/dev/sda5

hd,表示IDE设备 sd,表示SCSI设备

a,表示该接口的第一块硬盘,以此类推b为第二块(磁盘序号)

5,第5个分区(分区序号)

4.Linux文件系统

文件系统是对文件存储设备的空间进行组织和分配，负责文件存储并对存入的文件进行保护和检索的系统。

典型的文件系统

-EXT4,第四代扩展文件系统,RHEL6系列默认 (适合小文件)

-XFS,高级日志文件系统,RHFL系列默认 (适合大文件)

-SWAP,交换空间(虚拟内存)

1. 格式化：赋予空间文件系统过程

分区规划及使用

扇区默认的大小为512字节

一块硬盘：识别硬盘 -> 分区 -> 格式化（文件系统） -> 挂载

一 识别硬盘 : lsblk

二 分区规划

使用交互式fdisk分区工具 : fdisk /dev/vdb

三 格式化文件系统(存储的数据的规则)

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

blkid : 查看文件系统类型

- f :强制格式化

四 挂载使用

mount : 挂载

df : 显示正在挂载设备的使用情况

五 开机自动挂载设备

六 综合分区

Partprobe : 刷新分区表信息,识别新的分区

小链接:

1. vim技巧:命令模式进入插入模式 按 o 可以另起一行进入插入模式
2. mount -a : 检验挂载

parted分区工具

parted /dev/vdb

(parted) mktable gpt #指定分区模式

(parted) print #输出分区表

(parted) mkpart #划分新的分区

分区名称？ []? haha #指定分区的名称,随意写

文件系统类型？ [ext2]? ext4 #分区文件系统类型,不起实际作用

起始点？ 0

结束点？ 2G

忽略/Ignore/放弃/Cancel? Ignore #忽略 硬盘存储分区表空间

(parted) print

(parted) unit GB #使用单位为GB

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

1.格式化交互文件系统 mkswap

2.启用交换分区 swapon -s #查看组成交换空间成员信息

3.停用交换分区 swapoff

4.开机自动启用交换分区 /dev/vdb1 swap swap defaults 0 0

swapon -a #专用于检测交换分区是否书写正确的指令

## LVM逻辑卷:

作用: 1.整合分散的空间 2.空间可以按照需求扩大

零散空闲存储 ---- 整合的虚拟磁盘 ---- 虚拟的分区

物理卷 Physical Volume (PV)

卷组 Volume Group (VG)

逻辑卷 Logical Volume (LV)

将众多的物理卷(PV),组建成卷组(VG),从卷组中划分出逻辑卷(LV)

一 创建逻辑卷

successfully(成功)

1.创建卷组

命令格式: vgcreate 卷组名 设备路径......

vgs : 查看卷组信息

pvs : 查看物理卷信息

2.创建逻辑卷

命令格式:lvcreate -L 大小 -n 逻辑卷名称 基于的卷组

lvs #查看逻辑卷信息

1. 使用逻辑卷

与使用硬盘一样

二 逻辑卷的扩展

1.卷组有足够的剩余空间

A:扩展逻辑卷的空间

逻辑卷的扩张:lvextend -L 18G /dev/systemvg/vo

B:扩展文件系统

resize2fs:扩展ext4文件系统

xfs\_growfs:扩展xfs文件系统

1. 卷组没有足够的剩余空间

A:扩展卷组 : vgextend systemvg /dev/vdc3

B:扩展逻辑卷的空间 : lvextend -L 25G /dev/systemvg/vo

C:扩展文件系统 : xfs\_growfs /dev/systemvg/vo

了解内容:

逻辑卷空间可以进行缩小

ext4文件系统支持缩小

xfs文件系统不支持缩小

卷组划分空间的单位: 默认4M 称为 PE

vgdisplay systemvg : 显示卷组详细信息

• 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

了解:逻辑卷删除

删除的顺序:

首先删除逻辑卷,在删除卷组,最后删除物理卷(可以不删除物理卷)

1. 先卸载逻辑卷 lvremove /dev/systemvg/vo
2. 删除卷组(没有任何一个逻辑卷,基于此卷组创建)

vgremove systemvg

## U盘挂载:

1.以root用户登陆

先加载USB模块 modprobe usb-storage

用fdisk -l 看看U盘的设备

假如U盘是sda1

1. 确定在 目录 /mnt 下建立了 文件夹 /usb,如果未建立可键入一下命令:mkdir /mnt/usb\_disk,成功后进行下一步。

3.载入 u 盘，需键入以下命令：mount /dev/sda /mnt/usb\_disk

成功后，即可使用 u 盘了, 文件就在目录 /mnt/usb 下。

4.卸载u盘：在使用完u盘后，在拔出前需要先键入卸载U盘命令

命令如下：umount /mnt/usb

# 常用软件

## curl : 字符浏览器

## vim :

认识vim编辑器:

1.vim编辑器工作模式

命令模式

输入模式

末行模式

2.光标间的跳转

gg：跳转到文件的首行

G：跳转到文件的行末尾

1. 快捷键

dd:删除

2dd:删除两行

u:撤销

yy:复制

2yy:复制两行

p:粘贴

1. 保存退出

:wq 或 :x 保存并退出

:q! 强制退出

:w 写入

1. 字符串替换

行内替换

:s /n#查找目标/替换目标 替换当前行第一个查找目标

:s /查找目标/替换目标/g 替换当前行所有的查找目标

区域内替换

:n,m s/查找目标/替换目标/g 替换第n-m行所有的查找目标

:%s /查找目标/替换目标/g 替换文件内所有的查找目标

1. 行号的开关

:set number 显示行号

:set nonumber 不显示行号

1. 查找

/ + 查找内容

n：下一个匹配

N：上一个匹配

:nohl 取消高亮

1. vim教程:

#vimtutor

## git:

1.创建仓库

2.克隆到本地

3.将需要的文件放到git目录下

4.配置:

# git config --global user.email 账号

5. 同步

# git pull

6.如果目录下有新的文件

# git add -A .

7.提交更改

# git commit

输入备注

8.上传

# git push

# 单用户模式:

## 进入单用户模式:

1. 开机之后在内核上敲击e，然后编辑选项
2. 在linux16这一行，将ro后的内容改成rw rd.break

3.ctrl + x

## 修改root密码:

#mount -o remount,rw /sysroot //重新挂载/sysroot，使用rw方式进行挂载

# chroot /sysroot //进入系统的根目录

# passwd //重置密码

# touch /.autorelabel //创建文件,让系统重新启动时能够识别修改

# exit 退出chroot模式

# reboot //重启系统

# 重要的配置文件:

## 锁定项配置文件:

/etc/dconf/db/local.d

## 网卡配置文件:

/etc/sysconfig/network-scripts/ifcfg-p8p1

## DNS配置文件:

/etc/resolv.conf

## SELinux配置文件:

/etc/selinux/config

## Samba配置文件:

/etc/samba/smb.conf

## 数据库主配置文件:

/etc/my.cnf

## httpd主配置文件:

/etc/httpd/conf/httpd.conf

# sshd\_config 的选项

/etc/ssh/sshd\_config

描述

     sshd(8) 默认从 /etc/ssh/sshd\_config 文件(或通过 -f 命令行选项指定的文件)读取配置信息。

     配置文件是由"指令 值"对组成的，每行一个。空行和以'#'开头的行都将被忽略。

     如果值中含有空白符或者其他特殊符号，那么可以通过在两边加上双引号(")进行界定。

     [注意]值是大小写敏感的，但指令是大小写无关的。

     当前所有可以使用的配置指令如下：

## **AcceptEnv**

             指定客户端发送的哪些环境变量将会被传递到会话环境中。[注意]只有SSH-2协议支持环境变量的传递。

             细节可以参考 ssh\_config(5) 中的 SendEnv 配置指令。

             指令的值是空格分隔的变量名列表(其中可以使用'\*'和'?'作为通配符)。也可以使用多个 AcceptEnv 达到同样的目的。

             需要注意的是，有些环境变量可能会被用于绕过禁止用户使用的环境变量。由于这个原因，该指令应当小心使用。

             默认是不传递任何环境变量。

## **AddressFamily**

             指定 sshd(8) 应当使用哪种地址族。取值范围是："any"(默认)、"inet"(仅IPv4)、"inet6"(仅IPv6)。

## **AllowGroups**

             这个指令后面跟着一串用空格分隔的组名列表(其中可以使用"\*"和"?"通配符)。默认允许所有组登录。

             如果使用了这个指令，那么将仅允许这些组中的成员登录，而拒绝其它所有组。

             这里的"组"是指"主组"(primary group)，也就是/etc/passwd文件中指定的组。

             这里只允许使用组的名字而不允许使用GID。相关的 allow/deny 指令按照下列顺序处理：

             DenyUsers, AllowUsers, DenyGroups, AllowGroups

## **AllowTcpForwarding**

             是否允许TCP转发，默认值为"yes"。

             禁止TCP转发并不能增强安全性，除非禁止了用户对shell的访问，因为用户可以安装他们自己的转发器。

## **AllowUsers**

             这个指令后面跟着一串用空格分隔的用户名列表(其中可以使用"\*"和"?"通配符)。默认允许所有用户登录。

             如果使用了这个指令，那么将仅允许这些用户登录，而拒绝其它所有用户。

             如果指定了 USER@HOST 模式的用户，那么 USER 和 HOST 将同时被检查。

             这里只允许使用用户的名字而不允许使用UID。相关的 allow/deny 指令按照下列顺序处理：

             DenyUsers, AllowUsers, DenyGroups, AllowGroups

## **AuthorizedKeysFile**

             存放该用户可以用来登录的 RSA/DSA 公钥。

             该指令中可以使用下列根据连接时的实际情况进行展开的符号：

             %% 表示'%'、%h 表示用户的主目录、%u 表示该用户的用户名。

             经过扩展之后的值必须要么是绝对路径，要么是相对于用户主目录的相对路径。

             默认值是".ssh/authorized\_keys"。

## **Banner**

             将这个指令指定的文件中的内容在用户进行认证前显示给远程用户。

             这个特性仅能用于SSH-2，默认什么内容也不显示。"none"表示禁用这个特性。

## **ChallengeResponseAuthentication**

             是否允许质疑-应答(challenge-response)认证。默认值是"yes"。

             所有 login.conf(5) 中允许的认证方式都被支持。

## **Ciphers**

             指定SSH-2允许使用的加密算法。多个算法之间使用逗号分隔。可以使用的算法如下：

             "aes128-cbc", "aes192-cbc", "aes256-cbc", "aes128-ctr", "aes192-ctr", "aes256-ctr",

             "3des-cbc", "arcfour128", "arcfour256", "arcfour", "blowfish-cbc", "cast128-cbc"

             默认值是可以使用上述所有算法。

## **ClientAliveCountMax**

             sshd(8) 在未收到任何客户端回应前最多允许发送多少个"alive"消息。默认值是 3 。

             到达这个上限后，sshd(8) 将强制断开连接、关闭会话。

             需要注意的是，"alive"消息与 TCPKeepAlive 有很大差异。

             "alive"消息是通过加密连接发送的，因此不会被欺骗；而 TCPKeepAlive 却是可以被欺骗的。

             如果 ClientAliveInterval 被设为 15 并且将 ClientAliveCountMax 保持为默认值，

             那么无应答的客户端大约会在45秒后被强制断开。这个指令仅可以用于SSH-2协议。

## **ClientAliveInterval**

             设置一个以秒记的时长，如果超过这么长时间没有收到客户端的任何数据，

             sshd(8) 将通过安全通道向客户端发送一个"alive"消息，并等候应答。

             默认值 0 表示不发送"alive"消息。这个选项仅对SSH-2有效。

## **Compression**

             是否对通信数据进行加密，还是延迟到认证成功之后再对通信数据加密。

             可用值："yes", "delayed"(默认), "no"。

## **DenyGroups**

             这个指令后面跟着一串用空格分隔的组名列表(其中可以使用"\*"和"?"通配符)。默认允许所有组登录。

             如果使用了这个指令，那么这些组中的成员将被拒绝登录。

             这里的"组"是指"主组"(primary group)，也就是/etc/passwd文件中指定的组。

             这里只允许使用组的名字而不允许使用GID。相关的 allow/deny 指令按照下列顺序处理：

             DenyUsers, AllowUsers, DenyGroups, AllowGroups

## **DenyUsers**

             这个指令后面跟着一串用空格分隔的用户名列表(其中可以使用"\*"和"?"通配符)。默认允许所有用户登录。

             如果使用了这个指令，那么这些用户将被拒绝登录。

             如果指定了 USER@HOST 模式的用户，那么 USER 和 HOST 将同时被检查。

             这里只允许使用用户的名字而不允许使用UID。相关的 allow/deny 指令按照下列顺序处理：

             DenyUsers, AllowUsers, DenyGroups, AllowGroups

## **ForceCommand**

             强制执行这里指定的命令而忽略客户端提供的任何命令。这个命令将使用用户的登录shell执行(shell -c)。

             这可以应用于 shell 、命令、子系统的完成，通常用于 Match 块中。

             这个命令最初是在客户端通过 SSH\_ORIGINAL\_COMMAND 环境变量来支持的。

## **GatewayPorts**

             是否允许远程主机连接本地的转发端口。默认值是"no"。

             sshd(8) 默认将远程端口转发绑定到loopback地址。这样将阻止其它远程主机连接到转发端口。

             GatewayPorts 指令可以让 sshd 将远程端口转发绑定到非loopback地址，这样就可以允许远程主机连接了。

             "no"表示仅允许本地连接，"yes"表示强制将远程端口转发绑定到统配地址(wildcard address)，

             "clientspecified"表示允许客户端选择将远程端口转发绑定到哪个地址。

## **GSSAPIAuthentication**

             是否允许使用基于 GSSAPI 的用户认证。默认值为"no"。仅用于SSH-2。

## **GSSAPICleanupCredentials**

             是否在用户退出登录后自动销毁用户凭证缓存。默认值是"yes"。仅用于SSH-2。

## **HostbasedAuthentication**

             这个指令与 RhostsRSAAuthentication 类似，但是仅可以用于SSH-2。推荐使用默认值"no"。

             推荐使用默认值"no"禁止这种不安全的认证方式。

## **HostbasedUsesNameFromPacketOnly**

             在开启 HostbasedAuthentication 的情况下，

             指定服务器在使用 ~/.shosts ~/.rhosts /etc/hosts.equiv 进行远程主机名匹配时，是否进行反向域名查询。

             "yes"表示 sshd(8) 信任客户端提供的主机名而不进行反向查询。默认值是"no"。

## **HostKey**

             主机私钥文件的位置。如果权限不对，sshd(8) 可能会拒绝启动。

             SSH-1默认是 /etc/ssh/ssh\_host\_key 。

             SSH-2默认是 /etc/ssh/ssh\_host\_rsa\_key 和 /etc/ssh/ssh\_host\_dsa\_key 。

             一台主机可以拥有多个不同的私钥。"rsa1"仅用于SSH-1，"dsa"和"rsa"仅用于SSH-2。

## **IgnoreRhosts**

             是否在 RhostsRSAAuthentication 或 HostbasedAuthentication 过程中忽略 .rhosts 和 .shosts 文件。

             不过 /etc/hosts.equiv 和 /etc/shosts.equiv 仍将被使用。推荐设为默认值"yes"。

## **IgnoreUserKnownHosts**

             是否在 RhostsRSAAuthentication 或 HostbasedAuthentication 过程中忽略用户的 ~/.ssh/known\_hosts 文件。

             默认值是"no"。为了提高安全性，可以设为"yes"。

## **KerberosAuthentication**

             是否要求用户为 PasswordAuthentication 提供的密码必须通过 Kerberos KDC 认证，也就是是否使用Kerberos认证。

             要使用Kerberos认证，服务器需要一个可以校验 KDC identity 的 Kerberos servtab 。默认值是"no"。

## **KerberosGetAFSToken**

             如果使用了 AFS 并且该用户有一个 Kerberos 5 TGT，那么开启该指令后，

             将会在访问用户的家目录前尝试获取一个 AFS token 。默认为"no"。

## **KerberosOrLocalPasswd**

             如果 Kerberos 密码认证失败，那么该密码还将要通过其它的认证机制(比如 /etc/passwd)。

             默认值为"yes"。

## **KerberosTicketCleanup**

             是否在用户退出登录后自动销毁用户的 ticket 。默认值是"yes"。

## **KeyRegenerationInterval**

             在SSH-1协议下，短命的服务器密钥将以此指令设置的时间为周期(秒)，不断重新生成。

             这个机制可以尽量减小密钥丢失或者黑客攻击造成的损失。

             设为 0 表示永不重新生成，默认为 3600(秒)。

## **ListenAddress**

             指定 sshd(8) 监听的网络地址，默认监听所有地址。可以使用下面的格式：

                   ListenAddress host|IPv4\_addr|IPv6\_addr

                   ListenAddress host|IPv4\_addr:port

                   ListenAddress [host|IPv6\_addr]:port

             如果未指定 port ，那么将使用 Port 指令的值。

             可以使用多个 ListenAddress 指令监听多个地址。

## **LoginGraceTime**

             限制用户必须在指定的时限内认证成功，0 表示无限制。默认值是 120 秒。

## **LogLevel**

             指定 sshd(8) 的日志等级(详细程度)。可用值如下：

             QUIET, FATAL, ERROR, INFO(默认), VERBOSE, DEBUG, DEBUG1, DEBUG2, DEBUG3

             DEBUG 与 DEBUG1 等价；DEBUG2 和 DEBUG3 则分别指定了更详细、更罗嗦的日志输出。

             比 DEBUG 更详细的日志可能会泄漏用户的敏感信息，因此反对使用。

## **MACs**

             指定允许在SSH-2中使用哪些消息摘要算法来进行数据校验。

             可以使用逗号分隔的列表来指定允许使用多个算法。默认值(包含所有可以使用的算法)是：

             hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160,hmac-sha1-96,hmac-md5-96

## **Match**

             引入一个条件块。块的结尾标志是另一个 Match 指令或者文件结尾。

             如果 Match 行上指定的条件都满足，那么随后的指令将覆盖全局配置中的指令。

             Match 的值是一个或多个"条件-模式"对。可用的"条件"是：User, Group, Host, Address 。

             只有下列指令可以在 Match 块中使用：AllowTcpForwarding, Banner,

             ForceCommand, GatewayPorts, GSSApiAuthentication,

             KbdInteractiveAuthentication, KerberosAuthentication,

             PasswordAuthentication, PermitOpen, PermitRootLogin,

             RhostsRSAAuthentication, RSAAuthentication, X11DisplayOffset,

             X11Forwarding, X11UseLocalHost

## **MaxAuthTries**

             指定每个连接最大允许的认证次数。默认值是 6 。

             如果失败认证的次数超过这个数值的一半，连接将被强制断开，且会生成额外的失败日志消息。

## **MaxStartups**

             最大允许保持多少个未认证的连接。默认值是 10 。

             到达限制后，将不再接受新连接，除非先前的连接认证成功或超出 LoginGraceTime 的限制。

## **PasswordAuthentication**

             是否允许使用基于密码的认证。默认为"yes"。

## **PermitEmptyPasswords**

             是否允许密码为空的用户远程登录。默认为"no"。

## **PermitOpen**

             指定TCP端口转发允许的目的地，可以使用空格分隔多个转发目标。默认允许所有转发请求。

             合法的指令格式如下：

                   PermitOpen host:port

                   PermitOpen IPv4\_addr:port

                   PermitOpen [IPv6\_addr]:port

             "any"可以用于移除所有限制并允许一切转发请求。

## **PermitRootLogin**

             是否允许 root 登录。可用值如下：

             "yes"(默认) 表示允许。"no"表示禁止。

             "without-password"表示禁止使用密码认证登录。

             "forced-commands-only"表示只有在指定了 command 选项的情况下才允许使用公钥认证登录。

                                   同时其它认证方法全部被禁止。这个值常用于做远程备份之类的事情。

## **PermitTunnel**

             是否允许 tun(4) 设备转发。可用值如下：

             "yes", "point-to-point"(layer 3), "ethernet"(layer 2), "no"(默认)。

             "yes"同时蕴含着"point-to-point"和"ethernet"。

## **PermitUserEnvironment**

             指定是否允许 sshd(8) 处理 ~/.ssh/environment 以及 ~/.ssh/authorized\_keys 中的 environment= 选项。

             默认值是"no"。如果设为"yes"可能会导致用户有机会使用某些机制(比如 LD\_PRELOAD)绕过访问控制，造成安全漏洞。

## PidFile

             指定在哪个文件中存放SSH守护进程的进程号，默认为 /var/run/sshd.pid 文件。

## Port

             指定 sshd(8) 守护进程监听的端口号，默认为 22 。可以使用多条指令监听多个端口。

             默认将在本机的所有网络接口上监听，但是可以通过 ListenAddress 指定只在某个特定的接口上监听。

## PrintLastLog

             指定 sshd(8) 是否在每一次交互式登录时打印最后一位用户的登录时间。默认值是"yes"。

## PrintMotd

             指定 sshd(8) 是否在每一次交互式登录时打印 /etc/motd 文件的内容。默认值是"yes"。

## Protocol

             指定 sshd(8) 支持的SSH协议的版本号。

             '1'和'2'表示仅仅支持SSH-1和SSH-2协议。"2,1"表示同时支持SSH-1和SSH-2协议。

## PubkeyAuthentication

             是否允许公钥认证。仅可以用于SSH-2。默认值为"yes"。

## RhostsRSAAuthentication

             是否使用强可信主机认证(通过检查远程主机名和关联的用户名进行认证)。仅用于SSH-1。

             这是通过在RSA认证成功后再检查 ~/.rhosts 或 /etc/hosts.equiv 进行认证的。

             出于安全考虑，建议使用默认值"no"。

## RSAAuthentication

             是否允许使用纯 RSA 公钥认证。仅用于SSH-1。默认值是"yes"。

## ServerKeyBits

             指定临时服务器密钥的长度。仅用于SSH-1。默认值是 768(位)。最小值是 512 。

## StrictModes

             指定是否要求 sshd(8) 在接受连接请求前对用户主目录和相关的配置文件进行宿主和权限检查。

             强烈建议使用默认值"yes"来预防可能出现的低级错误。

## Subsystem

             配置一个外部子系统(例如，一个文件传输守护进程)。仅用于SSH-2协议。

             值是一个子系统的名字和对应的命令行(含选项和参数)。比如"sft /bin/sftp-server"。

## SyslogFacility

             指定 sshd(8) 将日志消息通过哪个日志子系统(facility)发送。有效值是：

             DAEMON, USER, AUTH(默认), LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7

## TCPKeepAlive

             指定系统是否向客户端发送 TCP keepalive 消息。默认值是"yes"。

             这种消息可以检测到死连接、连接不当关闭、客户端崩溃等异常。

             可以设为"no"关闭这个特性。

## UseDNS

             指定 sshd(8) 是否应该对远程主机名进行反向解析，以检查此主机名是否与其IP地址真实对应。默认值为"yes"。

## UseLogin

             是否在交互式会话的登录过程中使用 login(1) 。默认值是"no"。

             如果开启此指令，那么 X11Forwarding 将会被禁止，因为 login(1) 不知道如何处理 xauth(1) cookies 。

             需要注意的是，login(1) 是禁止用于远程执行命令的。

             如果指定了 UsePrivilegeSeparation ，那么它将在认证完成后被禁用。

## UsePrivilegeSeparation

             是否让 sshd(8) 通过创建非特权子进程处理接入请求的方法来进行权限分离。默认值是"yes"。

             认证成功后，将以该认证用户的身份创建另一个子进程。

             这样做的目的是为了防止通过有缺陷的子进程提升权限，从而使系统更加安全。

## X11DisplayOffset

             指定 sshd(8) X11 转发的第一个可用的显示区(display)数字。默认值是 10 。

             这个可以用于防止 sshd 占用了真实的 X11 服务器显示区，从而发生混淆。

## X11Forwarding

             是否允许进行 X11 转发。默认值是"no"，设为"yes"表示允许。

             如果允许X11转发并且sshd(8)代理的显示区被配置为在含有通配符的地址(X11UseLocalhost)上监听。

             那么将可能有额外的信息被泄漏。由于使用X11转发的可能带来的风险，此指令默认值为"no"。

             需要注意的是，禁止X11转发并不能禁止用户转发X11通信，因为用户可以安装他们自己的转发器。

             如果启用了 UseLogin ，那么X11转发将被自动禁止。

## X11UseLocalhost

             sshd(8) 是否应当将X11转发服务器绑定到本地loopback地址。默认值是"yes"。

             sshd 默认将转发服务器绑定到本地loopback地址并将 DISPLAY 环境变量的主机名部分设为"localhost"。

             这可以防止远程主机连接到 proxy display 。不过某些老旧的X11客户端不能在此配置下正常工作。

             为了兼容这些老旧的X11客户端，你可以设为"no"。

## XAuthLocation

             指定 xauth(1) 程序的绝对路径。默认值是 /usr/X11R6/bin/xauth

时间格式

     在 sshd(8) 命令行参数和配置文件中使用的时间值可以通过下面的格式指定：time[qualifier] 。

     其中的 time 是一个正整数，而 qualifier 可以是下列单位之一：

           <无>    秒

           s | S   秒

           m | M   分钟

           h | H   小时

           d | D   天

           w | W   星期

     可以通过指定多个数值来累加时间，比如：

           1h30m   1 小时 30 分钟 (90 分钟)

文件

     /etc/ssh/sshd\_config

             sshd(8) 的主配置文件。这个文件的宿主应当是root，权限最大可以是"644"。

# SELinux安全机制

## Security-Enhanced Linux

– 美国NSA国家安全局主导开发,一套增强Linux系统安全的强制访问控制体系

– 集成到Linux内核(2.6及以上)中运行

– RHEL7基于SELinux体系针对用户、进程、目录和文件

提供了预设的保护策略,以及管理工具

## SELinux运行模式的种类

• SELinux的运行模式

– enforcing(强制)、permissive(宽松)

– disabled(彻底禁用)

变成disabled状态,必须经历重起系统

## 切换运行模式

– 临时切换:setenforce 1|0

– 查看当前模式:getenforce

– 固定配置:/etc/selinux/config 文件

# 防火墙

## 根据所在的网络场合区分,预设保护规则集

--public:仅允许访问本机的sshd,dhcp,ping等少数几个服务

--trusted:允许任何访问

--block:阻塞任何来访请求(明确拒绝,给予回应)

--drop:丢弃任何来访的数据包(直接丢弃,不给予回应,节省资源)

## 防火墙的判定规则:

1. 查看客户端请求中源IP地址然后查阅知己哪个区域,又该IP地址的规则则进入该区域
2. 进入默认区域public

更改默认区域: firewall-cmd --set-default-zone= 更改的区域

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

查看区域的规则:firewall-cmd --zone=public --list-all

添加区域的规则:firewall-cmd --zone=public --add-添加的规则类型=添加的规则

永久添加 --permanent

规则类型:

1.service:服务规则

2.source:IP地址规则

3.forward-port=port=5423:proto=tcp:toport=80:端口转发规则

# Shell脚本

## Shell脚本基础

脚本:可以执行的一个文件,实现某种功能

规范Shell脚本的一般组成

#! 环境声明

# 注释文本

可执行代码

## 重定向输出

> :只收集前面命令的正确输出

2>:只收集前面命令的错误输出

&>:收集前面命令的正确输出与错误输出

>&2:重定向到标准错误输出

## 定义/赋值变量

• 设置变量时的注意事项

– 变量名只能由字母/数字/下划线组成,区分大小写

– 变量名不能以数字开头,不要使用关键字和特殊字符

– 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

• 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}

## 根据变量的用途不同区分

环境变量: 变量名一般都大写,用来设置用户/系统环境

由系统定义完成,由系统赋值完成

USER=当前系统登陆的用户名

位置变量: bash内置,存储执行脚本时提供的命令行参数

由系统定义完成,由系统赋值完成

表示为 $n,n为序号

$1、$2、.. .. ${10}、${11}、.. ..

为了降低脚本的使用难度,提高用户的体验度,采用交互式传递参数方法

read作用:

1.产生交互

2.自动记录用户在键盘上的输入

3.自动将用户在键盘上的输入,赋值给一个变量储存

预定义变量: bash内置,可直接调用的特殊值,不能直接修改

由系统定义完成,由系统赋值完成

$# 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

自定义变量: 用户自主设置、修改及使用

## 条件测试及选择

条件测试

[ 测试表达式 ] #每一部分之间都要有空格

常用的测试选项

检查文件状态

-e:文档存在为真

-d:文档存在,并且为目录才为真

-f:文档存在,并且为文件才为真

-r:文档存在,并且对其有读取权限为真

-w:文档存在,并且对其有写入权限为真

-x:文档存在,并且对其有执行权限为真

比较整数大小(带e的都有等于二字,g是大于,l是小于)

-gt:大于

-ge:大于等于

-eq:等于

-ne:不等于

-lt:小于

-le:小于等于

字符串比对

==:相等一致为真

!=:不相等为真

if选择结构

if双分支处理

if [条件测试];then

命令序列xx

else

命令序列yy

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cc

else

命令序列zz

fi

for循环结构:可以重复执行一个操作

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

命令序列

done

for循环中,变量的值可以不参与循环执行的动作

小链接:

1. 黑洞设备:/dev/null
2. 造数工具:{起始点..结束点} {1..50}生成1到50之间所有的数字
3. exit 设定程序退出返回值
4. 单引号 ' ' :取消特殊字符的意义,原样输出
5. 反撇号 ` ` 或 $( ) :将命令的输出,直接参数其他命令运行

# 报错实例解决

## ssh运行程序出现 cannot connect to X server

第一步：

export DISPLAY=:0.0

第二步：

xhost +local:root