**Linux进阶之路**

# 第一章Linux常识

## 1.1文件系统

Windows：NTFS、FAT16、FAT32

Linux：FAT32、ext3、ext4（RHEL6）、xfs（RHEL7）

Swap：交换空间（虚拟内存 缓解真实物理内存的不足）

## 1.2磁盘与分区表示

/dev/sda1 /dev/sdb3 /dev/sdc /dev/sdd

/dev/sda7：表示SCSI设备，第一块硬盘，第七个分区

hd：表示IDE设备

sd：表示SCSI设备

vd：表示虚拟化设备

## vim编辑技巧

1）命令模式

光标跳转

home或^ 、数字0 跳转到行首

end或者$ 跳转到行尾

1G或gg 跳转到文件的首行

G 跳转到文件的末尾行

复制/粘贴/删除

yy、#yy 复制光标处的一行、#行

p、P 粘贴到光标处之后、之前

x或delete 删除光标处的单个字符

dd、#dd 删除光标处的一行、#行

d^ 从光标之前删除至行首

d$ 从光标处删除至行尾

C（大写） 从光标处删除至行尾，进入插入模式

查找/撤销/保存

/word 向后查找字符串'world'

n,N 跳至后/前一个结果

u 撤销最近的一次操作

U 撤销对当前行的所有修改

Ctrl+r 取消前一次撤销操作

ZZ 保存修改并退出

2）末行模式

操作

:q! 放弃更改，强制退出

:wq 或者：x 保存并退出

:r /etc/passwd 读入其他文件内容

字符串替换

:s/old/new 替换当前第一个'old'

:s/old/new/g 替换当前所有的'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'

开关参数的控制

：set nu |nonu 显示/不显示行号

：set ai |noai 启用/关闭自动缩进

3）常见问题

缓存文件（交换文件）

解决办法：删除交换文件（以.开头的同名文件加上swap）

## 目录结构

#man hier

--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，/sbin中的程序只能管理员使用，/bin普通用户也能使用

/dev 存放硬盘、键盘、鼠标、光驱等各种设备文件

/root、/home/用户名 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

/peoc 存放内存中的映射数据，不占用磁盘空间

/tmp 存放系统运行过程中的一些临时文件

--权限的数值表示

• 权限的数值化

– 基本权限:r = 4,w = 2,x = 1

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

• 采用数值形式设置权限

– chmod [-R] nnn 文档...

– chmod [-R] xnnn 文档...

## system

• 一个更高效的系统&服务管理器

– 开机服务并行启动,各系统服务间的精确依赖

– 服务目录:/lib/systemd/system/

[root@svr7 /]# vim /lib/systemd/system/httpd.service

RHEL6:运行级别 200

0：关机

1：单用户模式（基本功能的实现，破解Linux密码）

2：多用户字符界面（不支持网络）

3：多用户字符界面（支持网络）服务器默认的运行级别

4：未定义

5：图形界面

6：重起

切换运行级别：init 数字

RHEL7：运行模式

字符模式：multi-user.target

图形模式：graphical.target

当前直接切换到图形模式

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

当前直接切换到字符模式

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

查看每次开机默认进入模式

# systemctl get-default

multi-user.target

设置永久策略，每次开机自动进入graphical.target

# systemctl set-default graphical.target

# reboot

## 1.3切换命令行界面

虚拟控制台切换( Ctrl + Alt + Fn 组合键)

tty1:图形桌面

tty2~tty6:字符控制台

命令行提示标识的含义

– [当前用户@主机名 工作目录]$

– 若当前用户是root,则最后一个字符为 #

– 否则,最后一个字符为 $

## 1.4颜色

蓝色：代表目录

黑色：代表文本文件

绿色：可运行的程序

红色：压缩包

1.5通配符

– \*:任意多个任意字符

– ?:单个字符

– [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

## 软连接与硬连接

都是快捷方式

--软连接

格式：ln -s /路径/源文档 /路径/快捷方式名称 #路径为绝对路径

软连接（常用）：快捷方式可以跨分区，源文件可以是目录

独占一个i节点

--硬连接

格式：ln /路径/源文档 /路径/快捷方式名称 #路径为绝对路径

硬连接：快捷方式不可以跨分区，源文件不可以是目录

硬连接与源文件i节点编号相同

i节点编号：标识硬盘上一块存储空间的编号

i节点

i节点表 数据存储区域

## 软件包安装

1）rpm包安装：

2）源码包安装

优点：可以选择安装的功能及路径，更加的灵活

--运行gcc与make变成可执行的程序

--步骤

安装gcc与make

# yum -y install gcc

make默认安装

tar解包

# tar -xf /tools/inotify-tools-3.13.tar.gz -C /

进入该目录，运行./configure配置

作用：指定安装目录/功能，检测本机是否安装gcc

例子

# ./configure --prefix=/opt/myrpm

参数：

--prefix #指定安装目录

make编译,生成可执行的二进制程序文件

#make

make install安装，将编译好的文件复制到安装目录

#make install

卸载

直接删除安装目录

## 1.5yum

服务：为客户端自动解决依赖关系，安装软件包

服务端：1.众多的软件包 2.仓库清单文件（repodata）

3.构建Web服务或FTP服务传递数据

1、服务端

众多的软件包、仓库数据文件(repodata)、搭建ftp服务共享内容

--安装ftp

#systemctl yum vsftpd

#systemctl restart vsftpd

#systemctl restart vsftpd

--挂载光盘到ftp目录

#ls /var/ftp

#vim /etc/fstab

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

2、客户端

客户端配置文件：/etc/yum.repos.d/\*.repo

注意：错误的配置文件会影响正确的配置文件

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

[rhel7] #仓库标识

name=rhel7.0 #仓库描述信息

baseurl= ftp://192.168.4.254/rhel7 #指定服务端位置

enabled=1 #是否启用该文件

gpgcheck=0 #是否检测红帽签名

Yum使用

# yum repolist #列出仓库信息

# yum -y install httpd #安装软件包

# yum -y install sssd

# yum -y install gcc

# yum -y install system-config-kickstart

# yum remove 软件名 #卸载

# yum clean all #清缓存

3、自定义yum仓库

具备互联网下载的包

--scp命令，依赖于ssh，把本地文件拷贝到远程主机上

格式：scp 本机文件绝对路径 远程主机用户名@远程主机ip地址：远程主机存放的路径

例子

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

root@192.168.4.7's password:

tools.tar.gz 100% 766KB 22.2MB/s 00:00

--解压tar包

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

--运行createrepo生成仓库清单文件

格式：createrepo 软件包路径

例子

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

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

boxes-1.1.1-4.el7.x86\_64.rpm oneko-1.2-19.fc24.x86\_64.rpm

cmatrix-1.2a-1.i386.rpm repodata

ntfs-3g-2014.2.15-6.el6.x86\_64.rpm sl-5.02-1.el7.x86\_64.rpm

--修改配置文件

位置：/etc/yum.repos.d/\*.repo

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

[myrpm]

name=rhel7.4

baseurl=file:///tools/other #软件源地址为本机写法

enabled=1

gpgcheck=0

--测试yum是否能用

#oneko & # &是把这个程序放入后台运行

## 1.6IP地址永久配置

方法一

配置：

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

ipv4.addresses '172.25.0.110/24 172.25.0.254'

connection.autoconnect yes

激活：nmcli connection up 'System eth0'

方法二

nmtui

1. Edit a connection（回车）#编辑连接

2. System eth0（回车） #选择网卡

3. Require IPv4 addressing for this connection

#利用IPv4进行激活

4. Automatically connect

#每次开机自动激活配置

## 1.7重定向输出

用途：将前面命令的输出结果，写入文本文件

>：覆盖重定向

>>：追加重定向

例子：

ls --help > /opt/ls.txt

管道

| ：将前面命令的输出结果，交由后面命令处理

## 1.8管道 |

用途 :将前面命令的输出结果，交由面后命令处理

例子：

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

## 1.9基本权限

1）访问方式(权限)

– 读取:允许查看内容-read r

– 写入:允许修改内容-write w

– 可执行:允许运行和切换-execute x

对于文本文件作用：

r:查看文件内容 cat less head tail

w:修改文件内容 vim 保存

x:双击运行该文件

2）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

3）使用 ls -l 命令

– ls -ld 文件或目录...

以 - 开头：代表文本文件

以 d 开头：代表目录

以 l 开头：快捷方式

## SELinux

Security-Emhamced Linux

增强安全linux

### 运行模式

-enforcing 强制 1

-permissive 宽松 0

-disabled 彻底禁用

任何一种模式变成disabled（彻底禁用）都要经历重启系统，反之也须重启

查看当前模式：getenforce

切换运行模式：setenforce 0|1 # permissive模式 | enforcing模式

配置文件：/etc/selinux/config

### SELinux布尔值

服务功能的开关 on或off

1、查看

[root@ client0 ~]# getsebool -a #查看

例子：

[root@client0 /]# getsebool -a | grep samba

samba\_create\_home\_dirs --> off

samba\_domain\_controller --> off

samba\_enable\_home\_dirs --> off

samba\_export\_all\_ro --> off

samba\_export\_all\_rw --> off

samba\_load\_libgfapi --> off

samba\_portmapper --> off

2、设置服务开

格式：setsebool [-P] 名称 on

例子：

[root@client0 /]# setsebool samba\_export\_all\_rw on

3、设置服务关

格式：setsebool [-P] 名称 off

提示：-p选项才能实现永久设置（注意内存需要2G以上）

例子：

[root@client0 /]# setsebool samba\_export\_all\_rw off

### SELinux端口

1、查看SELinux默认开放端口

格式：semanage port -l

#semanage port -l |grep http #查看含有http的默认开放端口

[root@client0 /]# semanage port -l | grep http

http\_cache\_port\_t tcp 8080, 8118, 8123, 10001-10010

http\_cache\_port\_t udp 3130

http\_port\_t tcp 80, 81, 443, 488, 8008, 8009, 8443, 9000

pegasus\_http\_port\_t tcp 5988

pegasus\_https\_port\_t tcp 5989

2、添加开放端口

格式:semanage port -a -t http\_port\_t -p tcp 端口号 #添加selinux开放端口

提示:内存需要2G以上

例子：

[root@client0 /]# semanage port -a -t http\_port\_t -p tcp 8909

3、删除开放端口

格式:semanage port -d -t http\_port\_t -p tcp 端口号

[root@client0 /]# semanage port -d -t http\_port\_t -p tcp 8909

### SELinux的安全上下文

1、查看所有SELinux标记

#semanage fcontext -l

#ls -Zd /var/www 查看目录的上下文

#ls -Zd /webroot 查看目录的上下文

2、更改标记

方式1

chcon --help |head -5 查看--reference=参考文件 文件...

chcon [-R] --reference=模板目录 新目录.....

例子

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

# ll -Z /webroot

## ssh 用户登陆访问限制

/etc/ssh/sshd\_config

[root@server0 ~]# vim /etc/ssh/sshd\_config

DenyUsers 用户名@地址(域名) .......

DenyUsers \*@192.168.4.\* \*@\*.jluo.com

AllowUsers 用户名@地址(域名) .......

AllowUsers \*@192.168.4.\* \*@\*.jluo.com

## 配置用户环境

1.别名

/root/.bashrc #只能管理员root使用

/home/普通用户家目录/.bashrc #只能该普通用户使用

/etc/bashrc #所有用均可使用

2.环境变量

位置：

/etc/profile

PATH:存储命令程序的路径

[root@desktop0 /]# echo $PATH

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

## 防火墙

功能：隔离作用

### 定义

分类：

硬件防火墙

品牌：华为，思科等

软件防火墙

系统服务firewalld

管理工具：firewall-cmd,firewall-config(图形)

### 防火墙配置

1）常用区域

-public ：仅允许访问本机的ssh，ping，dhcp服务

-trusted：允许任何访问

-block：阻塞任何来访请求，明确拒绝

-drop：丢弃任何来访的数据包，直接丢弃

数据包：数据 源IP 目标IP 目标端口号

2）防火墙进入区域规则：匹配及停止

--查看数据包源ip地址，再查看所有的区域中，是否有该源ip地址的规则，哪个区域有则进入哪个区域

--进入默认区域public

3）默认区域的修改

查看默认区域

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

修改默认区域

[root@server0 ~]# firewall-cmd --set-default-zone=区域名称

例如：

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

success

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

trusted

4）服务

查看public的所有服务

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

添加服务

[root@server0 ~]# firewall-cmd --zone=public --add-service=服务名称

例子：

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

success

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

success

[root@server0 ~]# firewall-cmd --zone=public --list-services #查看开放的服务

dhcpv6-client ftp http ssh

提示：--permanent

加入--permanent 永久修改，能直接修改配置文件,需重新加载防火墙配置文件

[root@server0 ~]# firewall-cmd --reload #重新加载防火墙配置文件

5）添加源ip规则设置

[root@desktop0 ~]# firewall-cmd --zone=block --add-source=源ip地址

例子：

[root@desktop0 ~]# firewall-cmd --zone=block --add-source=172.25.0.11

success

注意：一个ip只允许属于一个区域

查看活跃区域信息

[root@room9pc01 ~]# firewall-cmd --get-active-zones

public

interfaces: eth0

block

sources: 176.130.2.0/24

[root@room9pc01 ~]#

6）端口映射

端口号：编号 标识程序及应用、服务

1)互联网常见的协议

协议名称 默认端口号

http:超文本传输协议 80

https：安全（加密）的超文本传输协议 443

ftp：文本传输协议 21

DNS：域名解析协议 53

SMTP：用户发邮件协议 25

pop3：用户收邮件协议 110

telnet：远程管理协议 23

TFTP：简单的文本传输协议 69

SNMP：网络管理协议 161

注意：协议的端口可以有root改变，也可以具备多个端口号

2)端口映射

实现：172.25.0.11：5423 ---->172.25.0.11:80

[root@server0 ~]# firewall-cmd --permanent --zone=public --add-forward-port=port=5423:proto=tcp:toport=80

success

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

success

注意：本机访问自己无法使用防火墙映射

# 第二章 linux常用命令

命令行的一般格式

• 基本用法

– 命令字 [选项].. [参数1] [参数2]...

快速编辑技巧

• Tab键自动补全

– 可补齐命令字、选项、参数、文件路径、软件名、服务名

• 快捷键

– Ctrl + c:结束正在运行的命令

– Esc + . 或 ALT + . :粘贴上一个命令的参数

– Ctrl + l:清空整个屏幕

– Ctrl + u:清空至行首

– Ctrl + w:往回删除一个单词(以空格界定)

## 2.1 用户管理

用户帐号标识：UID

管理员UID为0

系统程序用户默认 1-999

所有普通用户UID默认从1000起

组帐号标识：GID

组帐号分类：基本组 附加组（从属组）

注意：所有用户至少属于一个组

基本组：由linux系统创建，由linux将用户加入，与用户同名

附加组（从属组）：由root用户创建

用户基本信息存在/etc/passwd文件

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

用户名：密码占位符：UID：基本组GID：用户描述信息：家目录：解释器

用户信息密码文件：/etc/shadow

root:$6$UiGI4Tc2$htsXYn5cJnOqv3P1VLcUSgfjDu2pL5yiJBuua6foZAHdwqeuLHfYUfS/vBn27Wjvoel8EJgtdsMjyquqvKAmf1:16261:0:99999:7:::

用户名：密码加密字段：上一次修改密码时间：自1970-1-1到上一次修改密码的时间经过的天数：：用户有效期

！！：该用户为锁定状态

修改文件/etc/shadow 可以达到目的

### 2.1.1 useradd

用途：添加用户

格式：useradd [选项] 用户名

常用选项

-u：用户id

-d：用户家目录

-s：用户登录解释器

-G：附加组

例子：

[root@client0 ~]# mkdir cluo

[root@client0 ~]# useradd -u 1002 -d /cluo -s /sbin/nologin cluo

### 2.1.2passwd

用途：修改用户密码

格式1：passwd #修改个人正在登录的用户密码

例子：

[root@client0 ~]# passwd

Changing password for user root.

New password:

BAD PASSWORD: The password is shorter than 8 characters

Retype new password:

passwd: all authentication tokens updated successfully.

格式2：passwd 用户名 #修改指定用户密码

例子：

[root@client0 ~]# passwd cluo

Changing password for user cluo.

New password:

BAD PASSWORD: The password is shorter than 8 characters

Retype new password:

passwd: all authentication tokens updated successfully.

非交互式修改用户密码：

echo ‘密码’ | passwd –stdin 用户名

例子：

[root@client0 ~]# echo '123' | passwd --stdin cluo

Changing password for user cluo.

passwd: all authentication tokens updated successfully.

### 2.1.3 usermod

用途：修改用户信息

格式：usermod [选项] 用户名

例子：

[root@client0 ~]# usermod -u 1003 -d /cluo -s /bin/bash cluo

### 2.1.4 userdel

用途：删除用户

格式：userdel [-r] 用户名 #加上-r连用户家目录一起删除

例子：

[root@client0 ~]# userdel -r cluo

## 2.2 组管理

组配置文件：/etc/group

lol:x:1003:jluo03

组名：密码占位符：组的GID：组的成员列表

### 2.2.1 groupadd

用途：创建组

格式：groupadd [-g 组id] 组名

例子：

[root@client0 ~]# groupadd -g 1005 lol

### 2.2.2 groupmod

用途：修改组信息

格式：groupmod [-g 组id] 组名

例子：

[root@client0 ~]# groupmod -g 1006 lol

### gpasswd

用途：添加/删除用户附加组

格式：gpasswd [选项] 用户 组名

-a：添加

-d：删除

例子：

[root@client0 ~]# gpasswd -a cluo lol

Adding user cluo to group lol

[root@client0 ~]# gpasswd -d cluo lol

Removing user cluo from group lol

### groupdel

用途：删除组信息

格式：groupdel 组名

例子

[root@client0 ~]# groupdel jk

## 链路聚合

用途：备份网卡 实现网卡双工工作

聚合方式

--轮询式（roundrobin）的流量负载均衡

--热备份（activebackup）连接冗余 #常用

### 热备份链路聚合

运行器的类型切换：man teamd.conf |grep runner

"runner": {"name": "activebackup"}

或者

"runner": {"name": "roundrobin"}

eth0 eth1

虚拟网卡team0

1.创建虚拟网卡team0

[root@server0 ~]# nmcli connection add type team con-name team0 ifname team0 autoconnect yes config '{"runner": {"name": "activebackup"}}'

Connection 'team0' (21f7c390-5968-4780-9efa-821b34385b73) successfully added.

2.添加成员

向虚拟网卡team0中添加eth1网卡

[root@server0 ~]# nmcli connection add type team-slave con-name team0-1 ifname eth1 master team0

Connection 'team0-1' (bb2ed3b1-9fd2-4341-bd1b-0ceaada7ae6c) successfully added.

向虚拟网卡team0中添加eth2网卡

[root@server0 ~]# nmcli connection add type team-slave con-name team0-2 ifname eth2 master team0

Connection 'team0-2' (1f347663-449b-44c1-acc6-580109c955d8) successfully added.

3.为虚拟网卡team0配置ip

[root@server0 ~]# nmcli connection modify team0 ipv4.method manual ipv4.addresses 172.25.0.211 connection.autoconnect yes

4.激活配置

激活顺序：虚拟网卡 虚拟网卡下的真实网卡

查看虚拟网卡状态

teamdctl 虚拟网卡名称 state

teamdctl team0 state

## 文件查看

## 文件/文件查找

### group

用途：查找指定条件的文件

格式：grep [选项] 查找条件 目标文件

-i：忽略大小写

-v：取反匹配

查找条件如：

word 包含字符串word

^word 以字符串word开头

word$ 以字符串word结尾

注意：这几个可以连用

例子：

[root@client0 ~]# grep root /etc/passwd

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

operator:x:11:0:operator:/root:/sbin/nologin

### find

1、简单使用

用途：查找指定调节的目录、文件

格式：find [目录] [条件] [-a|-o] [条件2].....

常用的条件表示

-type 类型（f,d,l,b,c）

-name “文档名称”

-size +10M #大于10M k(小写),M（大写）,G（大写）

-size -10M #小于10M

-user 用户名 #按照文档的所有者

-iname “文档名称” #忽略大小写

-group 组名 #按照文档的所属组

-maxdepth 层数 #限制目录查找的深度（最大层）

-a ：and

-o ：or

按文件修改时间

-mtime +10 #过去的10 天之前修改和创建的文档

-mtime -10 #过去的10 天之后修改和创建的文档

注意：find /etc/ -size -1M #size不能与1连用，可用1M=1024k

例子：

[root@client0 /]# find / -user jluo

/var/spool/mail/jluo

/home/jluo

/home/jluo/.mozilla

/home/jluo/.mozilla/extensions

/home/jluo/.mozilla/plugins

/home/jluo/.bash\_logout

/home/jluo/.bash\_profile

/home/jluo/.bashrc

2、find命令的-exec操作

用途：将find查找到的结果交由后面处理

格式：find .. .. -exec 处理命令{} \;

优势：以{}代表，每一个结果，逐个处理，遇到\;结束

例子：

[root@client0 /]# find / -user jluo -exec cp {} /jluo \;

[root@client0 /]# ll -A /jluo

-rw-r--r--. 1 root root 18 Oct 7 09:46 .bash\_logout

-rw-r--r--. 1 root root 193 Oct 7 09:46 .bash\_profile

-rw-r--r--. 1 root root 231 Oct 7 09:46 .bashrc

-rw-r-----. 1 root root 0 Oct 7 09:46 jluo

## 用户登陆分析

### users who w

users：查看已登陆的用户名

who：查看已登陆的用户信息（常用）

w：查看已登陆的用户信息（最详细，不常用）

例子

[root@svr7 named]# users

root

[root@svr7 named]# who

root pts/0 2018-09-25 08:25 (192.168.4.254)

[root@svr7 named]# w

16:52:24 up 8:33, 1 user, load average: 0.00, 0.01, 0.05

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

root pts/0 192.168.4.254 08:25 0.00s 0.71s 0.04s w

### last lastb

用途：查看最近登陆成功/失败的用户信息

例子

[root@svr7 named]# last

root tty1 Tue Sep 25 15:18 - 15:19 (00:00)

root pts/1 192.168.4.254 Tue Sep 25 15:16 - 16:38 (01:22)

root pts/1 192.168.4.254 Tue Sep 25 14:20 - 14:47 (00:26)

[root@svr7 named]# lastb

(unknown tty1 Thu Sep 20 09:24 - 09:24 (00:00)

root tty1 Thu Sep 20 09:23 - 09:23 (00:00)

(unknown ttyS0 Tue Jan 30 12:35 - 12:35 (00:00)

## 进程管理

1.简介

1）程序：静态的代码，仅仅占用的磁盘空间

2）进程：正在运行的代码，会占用cpu与内存

--父进程与子进程，后台整个进程，进程数

--进程唯一标识编号：PID

3）进程的前台后台调度

--将进程放入后台启动

在命令行末尾添加‘&’符号，不占用当前终端，放入后台执行

Ctrl + z ：将正在运行的程序暂停并放入后台

### pstree

systemd:Linux运行的第一个进程，由内核直接运行，是所有进程的父进程。

--pstree

命令：pstree #查看整个进程数(root)

命令：pstree [选项] [用户名] #查看指定用户名的进程数

选项

-u：指定用户名(可省略)

-p：列出对应PID编号

-a：显示完整的命令行

例子

[root@svr7 named]# pstree -ap

systemd,1 --switched-root --system --deserialize 21

├─ModemManager,676

│ ├─{ModemManager},688

│ └─{ModemManager},721

├─NetworkManager,673 --no-daemon

### ps

命令：ps aux 操作（进程的详细信息）

用途：详细列出正在运行的所有进程

[root@svr7 named]# ps aux

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

root 1 0.0 0.6 129416 6864 ? Ss 08:19 0:04 /usr/lib/system

root 2 0.0 0.0 0 0 ? S 08:19 0:00 [kthreadd]

命令：ps -elf 操作（进程的父进程的信息PPID）

用途：列出正在运行的所有进程

[root@svr7 named]# ps -elf

F S UID PID PPID C PRI NI ADDR SZ WCHAN STIME TTY TIME CMD

4 S root 1 0 0 80 0 - 32354 ep\_pol 08:19 ? 00:00:04 /usr/

1 S root 2 0 0 80 0 - 0 kthrea 08:19 ? 00:00:00 [kthr

### top

用途：进程动态排名

格式：top [-d 刷新妙数] [-u 用户名]

P：进程CPU排序

M：进程内存排序

q：退出

[root@svr7 named]# top -d 1

top - 14:48:12 up 6:29, 1 user, load average: 0.00, 0.01, 0.05

Tasks: 108 total, 1 running, 107 sleeping, 0 stopped, 0 zombie

%Cpu(s): 0.1 us, 0.0 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st

KiB Mem : 1016260 total, 596128 free, 166508 used, 253624 buff/cache

KiB Swap: 2097148 total, 2097148 free, 0 used. 654916 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

6220 root 20 0 0 0 0 S 0.1 0.0 0:00.01 kworker/0:0

6230 root 20 0 157696 2168 1524 R 0.1 0.2 0:00.02 top

1 root 20 0 129416 6864 2720 S 0.0 0.7 0:04.20 systemd

2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd

### pgrep

用途：查看指定进程的指令

格式：pgrep [选项].. 查询条件

选项

-l：输出进程名，而不仅仅是PID

-U：检索指定终端的进程

-t：检索指定终端的进程

-x：精确匹配完整的进程名

例子

[root@svr7 named]# pgrep -lu lisi -t pts/1

6562 bash

6601 top

[root@svr7 named]# pgrep -x named

4090

### jobs

用途：查看后台任务列表

格式：jobs [-l]

### fg

用途：将后台任务恢复到前台运行

格式：fg 编号

提示：编号在jobs名查看

### bg

用途：激活后台被挂起的任务

格式：bg 编号

提示：编号在jobs名查看

### sleep

用途：让当前终端休眠

格式：sleep [时间] #时间单位是秒

例子

[root@svr7 named]# sleep 100

[1]+ 已停止 sleep 100

[root@svr7 named]# jobs

[1]+ 已停止 sleep 100

[root@svr7 named]# bg 1

[1]+ sleep 100 &

[root@svr7 named]# jobs

[1]+ 运行中 sleep 100 &

[root@svr7 named]# fg 1

sleep 100

### kill

用途：杀死进程

--kill [-9] PID....

--killall [-9] 进程名...

--pkill 查找条件 #不介意使用

[-9]：强制杀死

例子：

[root@svr7 named]# sleep 4000 &

[1] 7003

[root@svr7 named]# jobs -l

[1]+ 7003 运行中 sleep 4000 &

[root@svr7 named]# kill 7003

[root@svr7 named]# jobs -l

[1]+ 7003 已终止 sleep 4000

[root@svr7 named]# jobs -l

[root@svr7 named]# sleep 5000 &

[1] 7004

[root@svr7 named]# sleep 500 &

[2] 7005

[root@svr7 named]# jobs -l

[1]- 7004 运行中 sleep 5000 &

[2]+ 7005 运行中 sleep 500 &

[root@svr7 named]# killall sleep

[1]- 已终止 sleep 5000

[2]+ 已终止 sleep 500

查看进程的命令：

pstree： 结构鲜明，可以查看整个进程树

ps aux： 查看正在运行的所有进程，信息非常全面

ps -elf 查看正在运行的所有进程，信息中有父进程的信息

top：动态，可以进行排名

pgrep：选项众多，用于检索进行信息，方便脚步中变量赋值

## 其他命令

tcpdump

格式：

tcpdump [ -adeflnNOpqStvx ] [ -c 数量 ] [ -F 文件名 ]

[ -i 网络接口 ] [ -r 文件名] [ -s snaplen ]

[ -T 类型 ] [ -w 文件名 ] [表达式 ]

1. tcpdump的选项介绍

-a 　将网络地址和广播地址转变成名字；

-d 　将匹配信息包的代码以人们能够理解的汇编格式给出；

-dd 　将匹配信息包的代码以c语言程序段的格式给出；

-ddd 　将匹配信息包的代码以十进制的形式给出；

-e 　在输出行打印出数据链路层的头部信息；

-f 　将外部的Internet地址以数字的形式打印出来；

-l 　使标准输出变为缓冲行形式；

-n 　不把网络地址转换成名字；

-t 　在输出的每一行不打印时间戳；

-v 　输出一个稍微详细的信息，例如在ip包中可以包括ttl和服务类型的信息；

-vv 　输出详细的报文信息；

-c 　在收到指定的包的数目后，tcpdump就会停止；

-F 　从指定的文件中读取表达式,忽略其它的表达式；

-i 　指定监听的网络接口；

-r 　从指定的文件中读取包(这些包一般通过-w选项产生)；

-w 　直接将包写入文件中，并不分析和打印出来；

-T 　将监听到的包直接解释为指定的类型的报文，常见的类型有rpc （远程过程

调用）和snmp（简单　网络管理协议；）

-nn 不进行端口名称的转换

Hexdump

用途:查看二进制文件的内容

格式：hexdump -C XXX(文件名) -C是参数 不同的参数有不同的意义

-C 是比较规范的 十六进制和ASCII码显示

-c 是单字节字符显示

-b 单字节八进制显示

-o 是双字节八进制显示

-d 是双字节十进制显示

-x 是双字节十六进制显示

### ulimit

用途：ulimit用于shell启动进程所占用的资源

格式：limit 选项　［size］

-H 设置硬资源限制.

-S 设置软资源限制.

-a 显示当前所有的资源限制.

-c size:设置core文件的最大值.单位:blocks

-d size:设置数据段的最大值.单位:kbytes

-f size:设置创建文件的最大值.单位:blocks

-l size:设置在内存中锁定进程的最大值.单位:kbytes

-m size:设置可以使用的常驻内存的最大值.单位:kbytes

-n size:设置内核可以同时打开的文件描述符的最大值.单位:n

-p size:设置管道缓冲区的最大值.单位:kbytes

-s size:设置堆栈的最大值.单位:kbytes

-t size:设置CPU使用时间的最大上限.单位:seconds

-v size:设置虚拟内存的最大值.单位:kbytes

-u <程序数目> 　用户最多可开启的程序数目

### netstat

用途：查看系统中启动的端口信息

格式：netstat [选项]

常用选项

-a：显示所有的端口信息

-t：显示tcp连接的端口

-n：以数字格式显示端口号

-u：显示udp连接的端口号

-l：显示服务正在监听的端口信息，如http启动后，会一直接听80端口

-p：显示监听端口的服务（程序）名称是什么

常用组合：netstat -tunlp

### ss

用途：查看系统中启动的端口信息

格式：ss [选项]

常用选项

-a显示所有端口的信息

-n以数字格式显示端口号

-t显示TCP连接的端口

-u显示UDP连接的端口

-l显示服务正在监听的端口信息，如httpd启动后，会一直监听80端口

-p显示监听端口的服务名称是什么（也就是程序名称）

常用选项ss -tunlp

注意：在RHEL7系统中可以使用ss命令替代netstat命令，功能一样，选项一样。

### umask

权限掩码：umask

[root@server0 ~]# umask

0022

特殊权限 属主权限 属组权限 其他人权限

### systemctl

• Linux系统和服务管理器

– 是内核引导之后加载的第一个初始化进程(PID=1)

– 负责掌控整个Linux的运行/服务资源组合

对于服务的管理

systemctl restart 服务名 #重起服务

systemctl start 服务名 #开启服务

systemctl stop 服务名 #停止服务

systemctl status 服务名 #查看服务当前的状态

systemctl enable 服务名 #设置服务开机自启动

systemctl disable 服务名 #设置服务不开机自启动

systemctl enable 服务名

#查看是否开机自启动

[root@svr7 /]# systemctl is-enabled httpd

disabled #禁止开机启动

[root@svr7 /]# systemctl enable httpd

[root@svr7 /]# systemctl is-enabled httpd

enabled #允许开机启动

### who

用途：查看当前所有用户登陆系统的信息

[root@svr7 named]# who

root tty1 2018-09-25 15:18

root pts/0 2018-09-25 08:25 (192.168.4.254)

root pts/1 2018-09-25 15:16 (192.168.4.254)

tty：源服务器命令行或者图形界面

pts：终端图形命令行

### which

用途：显示命令的绝对路径

格式：which 命令

例子：

[root@client0 /]# which pwd

/usr/bin/pwd

### du

用途：统计文件的占用空间

格式：du [选项].. [目录或者文件]...

-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）

### date

查看/调整系统日期时间

查看格式：date [选项]

+%F:显示年月日

+%D:

– date +%F、date +%R

– date +"%Y-%m-%d %H:%M:%S"

– date -s "yyyy-mm-dd HH:MM:SS" #修改时间

### history

用途：操作历史命令

#history #查看历史命令

#history -c #清空命令

#!str #查看最近以str开头的命令（常用）

#!n #n为整数，查看最近第n行命令（不常用

### wc

用途：统计命令

格式：wc -l 文件名

例子：

[root@client0 /]# wc -l /etc/passwd

43 /etc/passwd

## 文件压缩、解压

常见的压缩格式

.gz-->gzip (速度快，压缩效率不是特别高)

.bz2-->bzip2 （中等）

.xz-->xz （速度慢，压缩效率高）

### Zip

跨平台压缩

--压缩

格式：zip [-r] 备份文件zip 被归档的文档.....

-r：被归档由目录须加上该选项

例子

# zip /opt/test.zip /root /boot

# ls /opt/

test.zip

--减压缩

格式：unzip 压缩文件 -d 减压存放目录

-d ：指定存放减压文件目录

例子

#unzip /opt/test.zip -d /mnt

# ls /mnt

boot root

### tar

用途：打包压缩文件

格式：tar 选项 /路径/tar包名字 /路径/源文件 /路径/源文件 ........

用途：解压

格式：tar 选项 /路径/tar包名字 [-C] /路径/释放的位置

选项

-c：创建归档

-x：释放归档

-f：指定归档文件名称 必须在选项最后

-z -j -J：调用.gz .bx2 .xz格式的工具进行处理

-t：显示归档中的文件清单

-C：指定释放位置

-p：保持归档文件内的绝对路径

例子：

[root@client0 /]# tar zcf /jluo/test.tar.gz /root /etc/passwd /etc/group

[root@client0 /]# ls /jluo

test.tar.gz

[root@client0 /]# tar -xf /jluo/test.tar.gz -C /jluo

[root@client0 /]# ls /jluo

etc root test.tar.gz

## 权限归属

### 权限分类

1、基本权限

1）访问权限

读取：r #查看文件内容 cat less head tail

写入：w #修改 vim 保存

可执行：x #运行

2）权限使用对象

所有组：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

3)字母表示

drwxr-xr-x. 2 root root 6 1月 12 2018 Desktop

第1字母：

-：文本文件

d：目录

l：快捷键

中间9个字母

所有者：前3个字母

所属组：中间3个字母

其他用户：后3个字母

4)linux判断用户具备的权限

查看用户对于文档处于什么身份：所有者>所属组>其他人 特点：匹配及停止

查看相应权限位置的权限

2、附加权限（特殊权限）

set GID

附加在属组的x上

属组的权限标识会变成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

Set UID（了解）

附加在属主的x位置上

属主的权限标识会变成s

适用于可执行的文件，Set UID可以让使用者具有文件属主的身份及部分权限

Stichy Bit

附加在其他人的x位上

其他人的权限标识会变成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（禁止用户操作别人的文档）

3、acl访问控制列表

能够对个别用户，个别组设置独立的权限

大多数挂载的EXT4/4 XFS默认支持

格式：

setfacl [-R] -m u:dc:rx 目录名 #设置特殊用户对该目录有读执行权限

setfacl [-R] -m g:组名:rx 目录名 #设置特殊组对该目录有读执行权限

setfacl -x u:dc 目录名 #移除该目录指定特殊用户的权限

setfacl -b 目录名 #移除该目录的所有acl权限

getfacl 目录名 #查看该目录的acl权限

例子：

[root@client0 ~]# ll /jluo

drwxr-xr-x. 2 root root 6 Oct 7 08:48 test

[root@client0 ~]# setfacl -m u:cluo:rwx /jluo/test/

[root@client0 ~]# ll /jluo

drwxrwxr-x+ 2 root root 6 Oct 7 08:48 test

[root@client0 ~]# getfacl /jluo/test/

getfacl: Removing leading '/' from absolute path names

# file: jluo/test/

# owner: root

# group: root

user::rwx

user:cluo:rwx #此处为acl权限

group::r-x

mask::rwx

other::r-x

### chmod

用途：设置文档的基本权限（rwx）

格式：chmod [-R] 归属关系 选项=权限类别 文档…….

-R:递归修改权限(修改此目录下的目录和文件都修改)

归属关系

u：所有者

g：所属组

o：其他人

选项

+：添加 - ：删除

权限类别：r、w、x

例子：

[root@client0 /]# ll /jluo/

-rw-r--r--. 1 root root 779344 Oct 7 07:41 test.tar.gz

[root@client0 /]# chmod ugo=rwx /jluo/test.tar.gz

[root@client0 /]# ll /jluo/

-rwxrwxrwx. 1 root root 779344 Oct 7 07:41 test.tar.gz

### chown

用途：设置文档归属关系（所有者、所属组）

格式：chown [-R] :属组 文档....

格式：chown [-R] 属主 文档....

格式：chown [-R] 属主：属组 文档....

例子：

[root@client0 /]# chown cluo:lol /jluo/test.tar.gz

[root@client0 /]# ll /jluo

-rwxrwxrwx. 1 cluo lol 779344 Oct 7 07:41 test.tar.gz

2.1 pwd

用途：显示当前用户所在目录

格式：pwd

2.2cd

用途：切换工作目录

格式：cd [目标文件夹位置]

2.3ls

用途：显示当前目录内容

格式：ls [选项]… [目录或者文件名]…

选项

-l：查看目录里文件详细信息

-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)

-A：包括名称以 . 开头的隐藏文档

2.4uname

用途：列出内核版本

格式：uname [选项]

-r ：列出详细内核版本号

例子：

uname -r

3.10.0-693.el7.x86\_64

含义：主版本. 次版本. 修订号. 企业版Linux 版本7. 64位操作系统

2.5lscpu

用途：列出linuxCPU处理器信息

格式：lscpu

例子：

[root@svr7 桌面]#Lscpu

2.6hostname

1）用途：查看主机名

格式：hostname

例子：

[root@localhost /]# hostname

localhost.localdomain

2）用途：修改主机名

格式：hostname 主机名

例子：

[root@localhost /]# hostname jluo.com

[root@localhost /]#hostname

Jluo.com

2.7ifconfig

用途：查看网卡连接信息

格式：ifconfig

例子：

[root@A ~]# ifconfig

2.8poweroff

用途：关机

格式：poweroff

例子：

[root@A ~]#poweroff

2.9reboot

用途：重启

格式：reboot

例子：

[root@A ~]reboot

2.10mkdir

用途：新建目录

格式：mkdir [-p] 目录名称

-p：表示连父目录一起创建

例子：

[root@A ~]mkdir –p aa/bb/cc/dd

2.11touch

用途：新建空文件

格式：touch 文件名….

例子：

[root@A ~]touch jluo.txt

2.12less

用途：分屏阅读（支持前后翻页）

格式：less [选项] 文件名…..

基本交互操作：

/：查找（n,N跳转匹配内容）

N：上一页

N：下一页

例子：

[root@A ~]less /etc/passwd

2.13head

用途：查看前几行文本内容

格式：head –n 文件名 n为数字

例子：

[root@A ~]head -3 /etc/group

2.14tail

用途：查看后几行文本内容

格式：tail –n 文件名 n为数字

例子：

[root@A ~]tail -4 /etc/passwd

2.15grep

用途：输出包含指定字符串的行

格式：grep ‘关键字’ 目标文件

例子:

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

2.16mount

1）挂载 mount

用途：让目录成为设备的访问点

格式：mount 设备路径 挂载点目录

例子：

[root@A ~]mount /dev/cdrom /dvd

2)卸载 umount

用途：卸载挂载点目录

格式：umount挂载点目录

例子：

[root@A ~]umount /dvd

注意事项：

1.卸载时，当前不要在访问点内

2.挂载设备时，访问点最好时自己创建的目录

2.17alias

1）设置别名

用途：设置别名

格式：alias 别名名称=’实际执行的命令行’

2）取消别名

格式：ualias 别名名称

2.18vim

用途：vim编辑器可以编辑所有文本文件,也可以新建文本文件

格式：vim 文件名

– 若目标文件不存在,则新建空文件并编辑

– 若目标文件已存在,则打开此文件并编辑

三个模式：命令模式 插入模式（编辑模式） 末行模式

i 键

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

令

模

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

: 键

末行模式 ：wq 保存并退出

末行模式 ：q！ 强制不保存并退出

2.19rm

用途：删除目录和文件

格式：rm [选项]… 文件或者目录

-r：递归删除（包含目录）

-f：强制删除

[root@A ~]rm –rf /jluo

2.20mv

用途：移动文件或者重命名

格式：mv 原文件….. 目标路径

例子：

[root@A ~]mv /opt/jluo /mnt

2.21cp

用途：复制文件或者目录

格式：cp [选项]… 原文件……. 目标路径

-r:递归,复制目录时必须有此选项

例子：

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

注意：

处理复制时出现重名，强制覆盖

在本次操作临时取消别名的操作： \cp [选项]… 原文件……. 目标路径

例子：

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

2.22date

1）查看当前时间

格式：date

例子：

[root@localhost /]# date

2018年 09月 03日 星期一 17:21:41 CST

2）修改时间

格式：date -s "年-月-日 时：分：秒"

例子：

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

[root@localhost /]# date

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

2.23bc

用途：计算机

格式：bs

运算法则： 加 + 减 - 乘 \* 除 / 取余数 %

例子：

[root@localhost /]#bc

2.24wget

用途：下载工具

格式：wget 原文件地址（网址）

例子：

[root@At/]wgethttp://classroom.example.com/content/rhel7.0/x86\_64/dvd/Packages/vsftpd-3.0.2-9.el7.x86\_64.rpm

2.25rpm

用途：安装和卸载软件

格式：rpm 选项 软件名…

-q：查询当前系统软件是否安装

-ivh：安装软件包

-e：卸载

-ql：列出软件安装清单

2.26passwd

1）修改自己密码

格式：passwd

2）修改其他用户密码（管理员专享）

格式：passwd 用户名

3）非交互式设置密码

echo ‘密码’ | passwd –stdin 用户名

2.27useradd

用途：添加用户

格式：useradd [选项]….. 用户名

-u 用户id：指定用户id

-d 家目录路径:指定用户家目录路径

-s 登录解释器：指定用户解释器

-G 附加组：添加用户附加组

例子：

[root@server0 /]#useradd –u 1005 –d /opt/jluo01 –s /bin/nologin –G test jluo01

2.28usermod

用途：修改用户属性

适用对象：管理员

格式：usermod [选项]…. 用户名

常用命令选项

– -u 用户id、-d 家目录路径、-s 登录解释器、-G 附加组

2.29userdel

用途：删除用户

适用对象：管理员

格式：userdel [-r] 用户名

-r：连同用户家目录一并删除

2.30groupadd

用途：添加组

适用对象：管理员（root）

格式：groupadd [-g 组ID] 组名

例子：

[root@A ~]# groupadd stugrp

2.31gpasswd

用途：添加组成员或者删除组成员

适用对象：管理员（root）

格式：gpasswd 选项 用户名 组名

-a：添加用户到组上

-d：讲用户从组上删除

2.32groupdel

用途：删除组名

适用对象：管理员（root）

格式：groupdel 组名

例子：

[root@A ~]# groupdel stugrp

2.33tar

1）常见的压缩格式及命令工具:

.gz ---> gzip

.bz2 ---> bzip2

.xz ---> xz

2）用途：对文件或者目录打包

对象：所有用户

格式：tar 选项 /路径/tar包名字 /路径/释放的位置

选项：

– -c:创建归档

– -x:释放归档

– -f:指定归档文件名称

– -z、-j、-J:调用 .gz、.bz2、.xz 格式的工具进行处理

– -t:显示归档中的文件清单

– -C（大写）:指定释放位置

2.34chmod

2.35

2.36

2.37

2.38

2.39

2.40

# 第三章系统配置文件

## 3.1系统版本信息

文件位置：/etc/redhat-release

例子：

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

## 3.2系统内存信息

文件位置：/proc/meminfo

例子：

[root@svr7 桌面]#cat /proc/meminfo

MemTotal: 16166612 kB

......

## 3.3linux解释器

位置：/etc/shell

例子:

[root@A ~]cat /etc/shell

默认解释器：/bin/bash

## 3.4保存别名文件

位置：~/.bashrc ,/etc/bashrc

例子：

[root@A ~]cat /root/.bashrc

## 3.5DNS服务器文件

位置：/etc/resolv.conf

例子：

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

nameserver 172.25.254.254

## 3.6用户信息文件

位置：/etc/passwd

例子：

## 3.7组信息文件

位置：/etc/group

例子：

组名：密码占位符：组的GID：组的成员列表

## 3.8用户密码信息文件

位置：/etc/shadow

例子：

3.9

# 第四章 基础服务配置

## 4.1NTP时间同步

NTP：Network Time Protocol 网络时间协议

4.1.1NTP服务器：

– NTP服务器为客户机提供标准时间

4.1.2NTP客户端

– NTP客户机需要与NTP服务器保持沟通

1. 软件

chrony ：客户端软件：专用于访问服务的软件

2）安装：[root@A ~]# yum -y install chrony

3）修改程序主配置文件，指定服务端位置

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

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

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

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

server classroom.example.com iburst

classroom.example.com：位NTP服务端域名（没有设置域名以地址写入）

4）重启程序

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

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

5）验证

[root@A ~]# date -s "2008-10-1 12:00" #设置错误时间

[root@A ~]# date #查看错误时间

[root@A ~]# systemctl restart chronyd #重启chrony

[root@A ~]# date #查看时间是否修改成功

## 4.2 cron计划任务

期性任务

• 软件包:cronie、crontabs

• 系统服务:crond

• 日志文件:/var/log/crond 排除故障

• 配置任务格式

分 时 日 月 周 任务命令行(绝对路径)

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

\*:匹配范围内任意时间

,:分隔多个不连续的时间点

-:指定连续时间范围

/n:指定时间频率,每n

格式：

crontab -e -u root #以root书写计划任务

打开chrony写入：\* \* \* \* \* date >> /opt/time.txt

crontab -l –u root #查看root计划任务

例子：每一分钟都写入到time.txt上

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

打开chrony写入：\* \* \* \* \* date >> /opt/time.txt

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

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

[root@A ~]# cat /opt/time.txt #查看结果

## autofs触发挂载

#mkdir -p 挂载点父目录

#mkdir -p /home/guests

#yum -y install autofs

#vim /etc/auto.master #autofs服务的主配置文件

被监控的目录 路径 自定义操作策略文件路径

/home/guest /etc/auto.guests

#vim /etc/auto.guests #操作策略文件路径

挂载点名 -挂载参数 服务器地址：NFS共享目录的路径

ldapuser0 -rw classroom.example.com:/home/guests/ldapuser0

\* -rw classroom.example.com:/home/guests/&

&：与前面\*参数相同

#systemctl restart autofs

#systemctl enable autofs

## 4.2 DNS服务

### DNS服务基础

1）功能

--正向解析

根据注册的域名查找对应的ip地址

--反向解析

根据ip地址查找对应的注册域名，不常用

2）域名体系

IANA--互联网数字分配机构

Internet Assigned Numbers Authority

整个域名系统的最高权威机构

CINIC--中国互联网络信息中心

China Internet Network Information Center

主管国家顶级域.cn

--域名代理/注册/购买服务商

新网--http://www.xinnet.com

万网--http://www.net.cn

中国互联--http://hulian.top

www.qq.com.

--根域.

www.qq.com. 最后的点为根域

一级域名：.com .cn .edu .hk .tw .us

二级域名：.cn.com .us.com .jp.com

三级域名：sina.cn.com

完整的主机名：web.sina.cn.com ftp.sina.cn.com

FQDN：完全合格主机名

### 单区域DNS

服务端：

3）安装软件包

bind 域名服务包

bind-chroot 提供虚拟根支持

--bind服务端程序

主要执行程序：/usr/sbin/named

系统服务：named

默认端口:tcp/udp 53

运行时的虚拟根环境：/var/named/chroot

--主配置文件: /etc/named.conf

作用

设置本服务器负责解析的域名

全局配置部分

-设置监听地址/端口、地址库存放位置等

options {

listen-on port 53 { 192.168.4.7; }; #监听地址和端口，

listen-on-v6 port 53 { ::1; };

directory "/var/named"; #地址文件的默认位置

dump-file "/var/named/data/cache\_dump.db";

statistics-file "/var/named/data/named\_stats.txt";

memstatistics-file "/var/named/data/named\_mem\_stats.txt";

allow-query { all; }; #允许任何客户机查询，删掉默认允许任何客户机访问

} 提示：参数中{}两边必须有空格，行尾';'结尾不能有空格

区域配置部分

-定义DNS区域、类型、地址文件路径等

-关键IN表示Internet，可省略

zone "tedu.cn" IN { #定义正向区域

type master; #区域为主DNS slave则为辅 hint只能为根域名使用，普通域名不能使用

file "tedu.cn.zone"; #地址库文件 命名规则：域名.zone

};

include "/etc/named.rfc1912.zones"; #可载入其他配置

include "/etc/named.root.key";

--地址库文件: /var/named/.....

4)修改配置文件

例子：

# cp /etc/named.conf /root/ #做好备份

# vim /etc/named.conf

options {

directory "/var/named";

};

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

5）新建地址库文件

cp -p 源文件 目标文件

-p：保持文件属性不变的拷贝

例子

#cp -p /var/named/named.localhost /var/named/tedu.cn.zone #保持属性不变拷贝

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

$TTL 1D

@ IN SOA @ rname.invalid. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

tedu.cn. NS svr7.tedu.cn.

svr7.tedu.cn. A 192.168.4.7

www.tedu.cn. A 1.1.1.1

可以改为：后面不加'.'，默认补全维护的域名如:tedu.cn,建议书写下面格式

tedu.cn. NS svr7

svr7 A 192.168.4.7

www A 1.1.1.1

提示：域名须要顶格写，不能有空格

6）重启

# systemctl restart named #启动NDS服务

# systemctl enable named #设置开机自启

客户端：

1）配置

# echo 'nameserver 192.168.4.7' > /etc/resolv.conf #告诉DNS服务器的地址

2）测试

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

Server: 192.168.4.7

Address: 192.168.4.7#53

Name: www.tedu.cn

Address: 1.1.1.1

### 多区域DNS

服务端：

# 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";

};

#cp -p /var/named/named.localhost /var/named/tedu.cn.zone #保持属性不变拷贝

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

$TTL 1D

@ IN SOA @ rname.invalid. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

tedu.cn. NS svr7.tedu.cn. #

svr7.tedu.cn. A 192.168.4.7

www.tedu.cn. A 1.1.1.1

# cp -p /var/named/tedu.cn.zone /var/named/qq.com.zone

# vim /var/named/qq.com.zone

$TTL 1D

@ IN SOA @ rname.invalid. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

qq.com. NS svr7

svr7 A 192.168.4.7

www A 2.2.2.2

jluo A 8.8.8.9

客户端：

1）配置

# echo 'nameserver 192.168.4.7' > /etc/resolv.conf #告诉DNS服务器的地址

2）测试

--测试域名1

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

Server: 192.168.4.7

Address: 192.168.4.7#53

Name: www.tedu.cn

Address: 1.1.1.1

--测试域名2

[root@pc207 ~]# nslookup www.qq.com

Server: 192.168.4.7

Address: 192.168.4.7#53

Name: www.qq.com

Address: 2.2.2.2

--DNS查询结果的轮询

### 泛域名解析

1）无规律：

-- \*

$TTL 1D

@ IN SOA @ rname.invalid. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

qq.com. NS svr7

svr7 A 192.168.4.7

www A 2.2.2.2

jluo A 8.8.8.9

\* A 9.9.9.9 #泛域名,匹配所有

--域名解析

qq.com

qq.com. NS svr7

svr7 A 192.168.4.7

www A 2.2.2.2

jluo A 8.8.8.9

\* A 9.9.9.9 #泛域名,匹配所有

qq.com. A 3.3.3.3

2）有规律的泛域名解析

需求：

pc1.tedu.cn---------->192.168.20.1

pc2.tedu.cn---------->192.168.20.2

pc3.tedu.cn---------->192.168.20.3

pc4.tedu.cn---------->192.168.20.4

.

.

pc50.tedu.cn---------->192.168.20.50

函数：$GENERATE 造数 起始点-结束点

例子

$GENERATE 1-50 pc$.tedu.cn. A 192.168.20.$

测试：

[root@pc207 ~]# nslookup pc4.tedu.cn

Server: 192.168.4.7

Address: 192.168.4.7#53

Name: pc4.tedu.cn

Address: 192.168.20.4

[root@pc207 ~]# nslookup pc8.tedu.cn

Server: 192.168.4.7

Address: 192.168.4.7#53

Name: pc8.tedu.cn

Address: 192.168.20.8

3）解析记录的别名

---CNAME

ftp A 4.4.4.4

tts CNAME ftp

测试：

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

Server: 192.168.4.7

Address: 192.168.4.7#53

tts.tedu.cn canonical name = ftp.tedu.cn. #加CNAME后的效果

Name: ftp.tedu.cn

Address: 4.4.4.4

### 子域授权

1.父域：www.qq.com(二级域名) #已经配置好了

2.子域：www.bj.qq.com(三级域名) #在另一台服务器上操作

# vim /etc/named.conf

options {

directory "/var/named";

};

zone "bj.qq.com" IN {

type master;

file "bj.qq.com.zone";

};

# cp -p /var/named/named.localhost /var/named/bj.qq.com.zone

#vim /var/named/bj.qq.com.zone

$TTL 1D

@ IN SOA @ rname.invalid. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

bj.qq.com. NS pc207

pc207 A 192.168.4.207

www A 20.30.40.50

测试：

[root@pc207 ~]# nslookup www.bj.qq.com 192.168.4.207

Server: 192.168.4.207

Address: 192.168.4.207#53

Name: www.bj.qq.com

Address: 20.30.40.50

[root@pc207 ~]# nslookup www.qq.com 192.168.4.7

Server: 192.168.4.7

Address: 192.168.4.7#53

Name: www.qq.com

Address: 2.2.2.2

[root@pc207 ~]# nslookup www.bj.qq.com 192.168.4.7

Server: 192.168.4.7

Address: 192.168.4.7#53

\*\* server can't find www.bj.qq.com: NXDOMAIN

子域授权配置

父域配置：

# vim /var/named/qq.com.zone

$TTL 1D

@ IN SOA @ rname.invalid. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

qq.com. NS svr7

bj.qq.com NS pc207 #填写

svr7 A 192.168.4.7

pc207 A 192.168.4.207 #填写

www A 2.2.2.2

jluo A 8.8.8.9

测试

[root@pc207 ~]# nslookup www.bj.qq.com 192.168.4.7

Server: 192.168.4.7

Address: 192.168.4.7#53

Non-authoritative answer: #非权威解答

Name: www.bj.qq.com

Address: 20.30.40.50

递归解析（默认允许）：客户端询问DNS解析，DNS服务器与其他NDS服务器交流，最终将解析结果带回来的过程。

recursion yes; 默认允许递归

recursion no; 禁止递归

#dig 域名

dig 域名

迭代解析：DNS服务器与其他DNS服务器交流的过程

3.主机映射记录文件

/etc/hosts（起到DNS解析的功能，只为本机提供解析功能）

#nslookup 不读/etc/hosts文件

客户端解析DNS域名优先级：

1./etc/hosts

2./etc/resolv.conf DNS服务器

### 缓存DNS

构建方式：

1)全局转发（建议）

将请求转发给指定的公共DNS（其他缓存），请求递归

--真机搭建yum仓库

# mount /iso/CentOS-7-x86\_64-DVD-1708.iso /dvd

# vim /etc/yum.repos.d/mydvd.repo

[CentOS7]

name=centos7

baseurl=file:///dvd

enabled=1

gpgcheck=0

--安装软件包

# yum -y install bind bind-chroot

--查看真机DNS服务地址

--修改DNS主配置文件，指定缓存dns服务器配置（全局转发式）

options {

directory "/var/named";

forwarders { 192.168.1.1 }; #转发给DNS服务器

};

测试

[root@pc207 ~]# nslookup www.baidu.com 192.168.4.254

Server: 192.168.4.254

Address: 192.168.4.254#53

Non-authoritative answer:

www.baidu.com canonical name = www.a.shifen.com.

Name: www.a.shifen.com

Address: 14.215.177.38

Name: www.a.shifen.com

Address: 14.215.177.39

2)根域迭代

依次向根、一级、二级...域的DNS服务器迭代

### split分离解析(视图解析)

1.当收到客户机的DNS查询请求的时候

--能够区分客户机的来源地址

--为不同类别的客户机提供不同的解析结果（ip地址）

--不同的客户端解析同一个域名时，解析结果不同

--为客户端提供最近的资源

2.环境及需求

– 权威DNS:svr7.tedu.cn 192.168.4.7

– 负责区域:tedu.cn

– A记录分离解析 —— 以 www.tedu.cn 为例

客户机来自解析结果

192.168.4.207 -------> 192.168.4.100

其他地址 -------> 1.2.3.4

3.分离解析配置

1）注意事项

--所有的客户端都要必须找到自己的分类，否则DNS直接报错

--类别的view匹配，由上即下，匹配即停止

--所有的zone都必须在view中

例子

view "nsd" { #视图名称可以自己取，不能取中文

match-clients{ 192.168.4.207; }; #配置客户端来源地址，可以配置多个

zone "tedu.cn" IN{

type master;

file "tedu.cn.zone";

};

};

view "other" {

match-clients{ any; };

zone "tedu.cn" IN{

type master;

file "tedu.cn.other";

};

};

2)acl地址列表

--为大批量的客户机地址建立列表

-- 若地址比较少,也可以不建立列表

如：

acl test{ 192.168.4.207;192.168.110.1/24; };

view "nsd" { #视图名称可以自己取，不能取中文

match-clients{ test; }; #配置客户端来源地址，可以配置多个

zone "tedu.cn" IN{

type master;

file "tedu.cn.zone";

};

};

view "other" {

match-clients{ any; };

zone "tedu.cn" IN{

type master;

file "tedu.cn.other";

};

};

3）多区域split分离解析

--注意事项：每一个view中，zone的个数都必须一样

acl test{ 192.168.4.207;192.168.110.1/24; };

view "nsd" {

match-clients{ test; };

zone "tedu.cn" IN{

type master;

file "tedu.cn.zone";

};

zone "baidu.com" IN{

type master;

file "baidu.com.zone";

};

zone "qq.com" IN{

type master;

file "qq.com.zone";

};

};

view "other" {

match-clients{ any; };

zone "tedu.cn" IN{

type master;

file "tedu.cn.other";

};

zone "baidu.com" IN{

type master;

file "baidu.com.zone";

};

zone "qq.com" IN{

type master;

file "qq.com.zone";

};

};

## 4.3LDAP

4.3.1 功能

主要实现网络用户认证，达到用户的集中管理

本地用户：由本地创建，用于本地登陆 本地/etc/paaswd

网络用户：由LDAP服务器提供用户名，密码信息

4.3.2 服务端

4.3.2 客户端

1、安装软件包

#yum -y install sssd # sssd：与LDAP服务器沟通的软件

#yum -y install authconfig-gtk # authconfig-gtk：图像工具，配置sssd软件包

2、家目录漫游

1）查看服务端共享

格式：showmount -e 服务器地址 #查看服务端共享

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

Export list for classroom.example.com:

/home/guests 172.25.0.0/255.255.0.0

1. 挂载

#mkdir /home/guests

#mount classroom.example.com: /home/guests /home/guests #临时挂载

## 4.4 SMB服务

提示：

客户端访问服务端资源排错步骤

1.防火墙策略

2.服务本身的访问控制

3.安全增强SELinux

4.服务端本地目录的权限

### 基础

1.SMB

samba服务基础，linux与windows跨平台的共享

windows:共享文件，直接点击属性，共享即可，隐藏即在共享名后面添加$符号，访问时//ip地址

例如： 共享文件test 共享名test$ 访问 //ip地址/test$ 要是没有$，直接访问地址即可

用途：为客户机提供共享使用的文件夹

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

系统服务：smb

samba用户：专门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（可以不设置密码，建议设置禁止登陆系统）

使用pdbedit管理工具

添加samba用户：pdbedit -a 用户名

删除samba用户：pdbedit -x 用户名

查看samba用户：pdbedit -L

### 部署

1、服务端

1）安装samba软件包

#yum -y install samba

2）创建samba共享帐号

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

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

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

[root@server0 ~]# pdbedit -a harry #123

[root@server0 ~]# pdbedit -a kenji #123

[root@server0 ~]# pdbedit -a chihiro #123

3）创建共享目录

发布目录：/common

共享名为：/common

4)修改配置文件

位置：/etc/samba/smb.conf

[common]

path = /common

提示：等号两边有空格

5)重启smb服务

[root@server0 ~]# systemctl restart smb

[root@server0 ~]# systemctl enable smb

6)SELinux布尔值（服务功能的开关 on或off）

查看

[root@server0 ~]# getsebool -a

设置

开

setsebool [-p] 名称 on

关

setsebool [-p] 名称 off

-p选项才能实现永久设置（注意内存需要2G以上）

例子：

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

7）修改本地目录权限

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

2、客户端

利用挂载mount访问

--安装软件：cifs-utils（让本机支持cifs文件系统）

--挂载访问：

格式：[root@desktop0 ~]# mount -o user=用户名,pass=密码 //ip地址/共享目录 挂载目录

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

4.完成开机自动挂载

\_netdev:声明网络设备

配置完所有的网络参数后，再进行挂载该设备

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

3、共享文件夹的多用户访问（了解）

SMB客户端的multiuser挂载技术

--客户端管理员只需要做一次挂载

--客户端在访问挂载点时，若需要不同权限，可以临时切换为新的共享用户（无需挂载）

multiuser,sec=ntlmssp

## 4.5NFS

network file server

用途：为客户机提供共享使用文件夹

协议：NFS（TCP/UDP 2049），RPC（TCP/UDP 111）

服务器

1)查看

所需软件包：nfs-utils （一般默认安装）

系统服务：nfs-server

2)exports配置文件

/etc/exports

-文件夹路径 客户机地址（权限）客户机地址（权限）.....

例子

/abc \*(ro) #可以查看

3)重新加载配置文件命令：exportfs -r

4)重启

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

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

客户端

1.查看服务器共享目录

showmount -e 地址

例如：

[root@desktop0 nsd01]# showmount -e 172.25.0.11

Export list for 172.25.0.11:

/abc 172.25.0.1/24

2.挂载

vim /etc/fstab

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

## 4.6 ISCSI共享存储

internet SCSI 网络SCSI接口

----一种基于c/s构架的虚拟磁盘技术

---服务器提供磁盘空间，客户机连接并当成本地磁盘使用

默认端口：3260

ISCSI Qualified Name (iqn)命名规范

-- iqn-yyyy-mm.倒序域名：自定义标示

-- 用来识别target，磁盘组,也用来识别客户机的身份

-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

配置文件

/etc/target/saveconfig.json

【服务端】：

1.划分新的分区

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

2.安装targetcli

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

3.运行targetcli可以在交互式界面，配置SCSI共享存储

[root@server0 /]# targetcli

1)创建与命名后端存储

/> backstores/block create name=nsd dev=/dev/vdb1

后端存储 块设备

2)创建target，磁盘组

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

3)lun，逻辑单元

/> iscsi/iqn.2018-09.com.example.server0/tpg1/luns create /backstores/block/nsd

4）配置访问控制,设置客户端声称的名字

/> iscsi/iqn.2018-09.com.example.server0/tpg1/acls create iqn.2018-09.com.example:desktop0

5）客户端访问本机的ip地址及端口(端口默认：3260)

/> iscsi/iqn.2018-09.com.example.server0/tpg1/portals create 172.25.0.11

4.重启服务

[root@server0 /]# systemctl restart target

[root@server0 /]# systemctl enable target

【客户端】

1.安装软件 iscsi-initiator-utils

yum安装能够补全包名：

--前提由yum的缓存（yum repolist）

--当前系统没有安装才可以补全

2.修改配置文件，用来

位置：vim /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.2018-09.com.example:desktop0

3.重启刷新配置文件

[root@server0 /]# systemctl restart iscsid

4.识别发现共享存储

1）书写发现的命令

man iscsiadm 进入搜索 大写example

Ctrl+- ：放小字体

2)运行命令

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

172.25.0.11:3260,1 iqn.2018-09.com.example.server0 #出现此命令则成功

3)本机识别服务共享

[root@server0 /]# systemctl restart iscsi

4）分区，格式化，挂载

[root@server0 /]#fdisk 设备路径

[root@server0 /]#mkfs.文件类型 设备名(要写详细路径)

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

## 4.7 PXE批量装机

1.DHCP服务

1)原理

--DHCP地址分配的四次会话(以广播方式进行，先到先得)

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

一个网络中只能有一个DHCP服务器

--主要分配以下入网参数

IP地址/子网掩码/广播地址

默认网关地址、DNS服务器地址

--服务端基本概念

组期：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地址范围

--配置dhcp地址分配服务

装软件包：dhcp

配置文件：/etc/dhcp/dhcpd.conf

启服务：dhcpd

2）部署DHCP

--装包

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

--配置文件

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

subnet 192.168.4.0 netmask 255.255.255.0 { #网段

range 192.168.4.100 192.168.4.200; #地址池

option domain-name-servers 192.168.4.7; #dhcp服务地址

# option domain-name "internal.example.org"; #dhcp服务域名

option routers 192.168.4.254; #网关

# option broadcast-address 10.5.5.31; #关播地址

default-lease-time 600; #最小连接时长，单位秒

max-lease-time 7200; #最小连接时长，单位秒

}

--重启服务

[root@svr7 ~]# systemctl restart dhcpd

2.网络装机

1）优势

--规模化

同时装配多台主机

--自动化

自动装系统、配置各种服务

--远程实现

不需要光盘、u盘等物理安装介质

2）PXE网络

PXE，Pre-boot eXecution Environment

预启动执行环境，在操作系统之前执行，可用于远程安装

--工作模式

PXE client集成在网卡的启动芯片中

当计算机引导时，从网卡芯片中把PXE clent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

--需要服务套件

DHCP，分配ip地址，定位引导程序

TFTP（简单的文件传输协议，69），提供引导程序下载

HTTP（FTP/NFS），提供yum安装源

开机启动项顺序：1.硬盘 2.光驱设备 3.移动硬盘 4.网络安装

--客户机应具备的条件

网卡芯片必须支持PXE协议

主板支持从网卡启动

3）搭建DHCP服务

--装包

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

--配置文件

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

subnet 192.168.4.0 netmask 255.255.255.0 { #网段

range 192.168.4.100 192.168.4.200; #地址池

option domain-name-servers 192.168.4.7; #dhcp服务地址

# option domain-name "internal.example.org"; #dhcp服务域名

option routers 192.168.4.254; #网关

# option broadcast-address 10.5.5.31; #关播地址

default-lease-time 600; #最小连接时长，单位秒

max-lease-time 7200; #最大连接时长，单位秒

next-server 192.168.4.7; #设置下一个服务地址

filename "pxelinux.0"; #指定网卡引导文件名称

}

pxelinux.0：网卡引导文件（安装说明书），二进制文件

需安装软件获得该文件

--重启

[root@svr7 ~]# systemctl restart dhcpd

4）搭建tftp服务

tftp：简单文件传输协议，端口号：69

默认共享文件路径：/var/lib/tftpboot/

--安装软件tftp-server

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

--启服务

[root@svr7 ~]# systemctl restart tftp

5）部署相应文件

--部署pxelinux.0文件

----安装获取pxelinux.0文件软件

[root@svr7 ~]#yum provides \*/pxelinux.0 #获取pxelinux.0文件软件名称

[root@svr7 ~]#yum -y install syslinux #安装软件

[root@svr7 ~]#rpm -ql syslinux |grep pxelinux.0 #获取pxelinux.0文件路径

/usr/share/syslinux/gpxelinux.0

/usr/share/syslinux/pxelinux.0

[root@svr7 ~]# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/ #复制

[root@svr7 ~]# ll /var/lib/tftpboot/

总用量 28

-rw-r--r--. 1 root root 26826 9月 26 10:27 pxelinux.0

--部署菜单文件

pxelinux.0-->/var/lib/tftpboot/pxelinux.cfg/default

----创建目录

[root@svr7 ~]# mkdir /var/lib/tftpboot/pxelinux.cfg

----挂载镜像文件

[root@svr7 ~]# mount /dev/cdrom /mnt

mount: /dev/sr0 写保护，将以只读方式挂载

[root@svr7 ~]# ls /mnt

addons GPL media.repo RPM-GPG-KEY-redhat-release

EFI images Packages TRANS.TBL

EULA isolinux repodata

extra\_files.json LiveOS RPM-GPG-KEY-redhat-beta

----拷贝pxelinux.cfg文件

[root@svr7 ~]# cp /mnt/isolinux/isolinux.cfg /var/lib/tftpboot/pxelinux.cfg/default

[root@svr7 ~]# ll /var/lib/tftpboot/pxelinux.cfg/default

-r--r--r--. 1 root root 3166 9月 26 10:59 /var/lib/tftpboot/pxelinux.cfg/default

[root@svr7 /]# chmod u+w /var/lib/tftpboot/pxelinux.cfg/default

[root@svr7 /]# ll /var/lib/tftpboot/pxelinux.cfg/default

-rw-r--r--. 1 root root 3166 9月 26 10:59 /var/lib/tftpboot/pxelinux.cfg/default

----修改菜单文件

[root@svr7 /]# vim /var/lib/tftpboot/pxelinux.cfg/default

1 etefault vesamenu.c32 #默认加载图形模块

2 timeout 600 #读秒时间 1/10 这个为：60秒

10 menu background splash.png #指定背景图片

11 menu title NSD1808 PXE Server #菜单显示标题

61 label linux

62 menu label Install Linux 7.4 #选项内容

63 menu default #读秒结束后默认选择

63 kernel vmlinuz #加载内核

64 append initrd=initrd.img #加载驱动

提示：后面全部删除

--部署图形的模块及背景模块

图形的模块：vesamenu.c32

背景模块：splash.png

#cp /mnt/isolinux/vesamenu.c32 /mnt/isolinux/splash.png /var/lib/tftpboot/

--部署启动内核与驱动程序

启动内核：vmlinuz

驱动程序：initrd.img

[root@svr7 /]# cp /mnt/isolinux/vmlinuz /mnt/isolinux/initrd.img /var/lib/tftpboot/

[root@svr7 /]# ll /var/lib/tftpboot/

总用量 54524

-r--r--r--. 1 root root 49763300 9月 26 11:12 initrd.img

-rw-r--r--. 1 root root 26826 9月 26 10:35 pxelinux.0

drwxr-xr-x. 2 root root 21 9月 26 10:59 pxelinux.cfg

-r--r--r--. 1 root root 186 9月 26 11:08 splash.png

-r--r--r--. 1 root root 153104 9月 26 11:08 vesamenu.c32

-r-xr-xr-x. 1 root root 5875184 9月 26 11:12 vmlinuz

6)搭建httpd

[root@svr7 /]# yum -y install httpd #安装服务

[root@svr7 /]# mkdir /var/www/html/rhel7

[root@svr7 /]# mount /dev/cdrom /var/www/html/rhel7 #一般直接拷贝，不用挂载

[root@svr7 /]# systemctl restart httpd #重启服务

[root@svr7 /]# firefox 192.168.4.7/rhel7

7)实现无人值守安装-->生成应答文件

--安装图形工具（system-config-kickstart）进行生成应答文件

[root@svr7 /]# yum -y install system-config-kickstart

--运行system-config-kickstart

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

[development] #表识名称必须写成development，运行system-config-kickstart

name=rhel7.4 #才能识别安装软件包，否则不能识别

baseurl=http://192.168.4.254/rhel7

enabled=1

gpgcheck=0

[root@svr7 /]# system-config-kickstart #运行

[root@svr7 ~]# cat /root/ks.cfg #生成的应答文件

--共享ks应答文件

[root@svr7 ~]# cp /root/ks.cfg /var/www/html

[root@svr7 ~]# firefox 192.168.4.7/ks.cfg

--修改文件

[root@svr7 ~]# vim /var/lib/tftpboot/pxelinux.cfg/default

label linux

menu label Install Red Hat Enterprise Linux 7.4

menu default

kernel vmlinuz

append initrd=initrd.img ks=http://192.168.4.7/ks.cfg

8）安装后脚本

--安装完自动搭建yum

方式一

echo '

[rhel7]

name=rhel7.4

baseurl=http://192.168.4.254/rhel7

enabled=1

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

提示：现在不用该方式

方式二

把文件rhel7.repo部署到PXE服务机的http上

cd /etc/yum.repos.d/

wget http://192.168.4.207/rhel7.repo

提示：现在常用这种方式

9）总结

--安装DHCP

yum -y install dhcp --> vim /etc/dhcp/dhcpd.conf

--> next-server,filename -->systemctl restart dhcpd

--安装TFTP

yum -y install tftp-server -->systemctl restart tftp

默认共享位置：/var/lib/tftpboot/

--安装http

yum -y install httpd -->mkdir /var/www/html/rhel7

-->systemctl restart httpd

--拷贝文件

mount /dev/cdrom /mnt

cp /mnt/\* /var/www/html/rhel7

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

cp vesamenu.c32 splash.png vmlinuz initrd.img /var/lib/tftpboot/

--pxelinux.0

yum provides \*/pxelinux.0 -->yum -y install syslinux -->rpm -ql syslinux | grep pxelinux.0

cp pxelinux.0 /var/lib/tftpboot/

--ks.cfg

yum -y install systemc-config-kickstart -->system-config-kickstart

-->ks.cfg

cp ks.cfg /var/www/html/

## Cobbler装机平台

基于CentOS7系统下部署Cobbler

1.基本概念

Cobbler是一款快速的网络系统部署工具

集中管理所需服务，如DHCP，DNS，TFTP，Web

Cobbler内部集成了一个镜像版本仓库、

Cobbler内部集成了一个ks应答文件仓库

Cobbler还提供了包括yum源管理，Web界面管理，API接口，电源管理等功能

--CentOS7系统设置

\*设置防火墙为trusted

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

\*当前及永久设置SELinux状态为permissive

[root@localhost ~]# setenforce 0 #当前临时关闭

[root@localhost ~]# getenforce

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

SELINUX=permissive

2.安装

安装软件 cobbler cobbler-web dhcp tftp-server pykickstart httpd tftp-server

cobbler #cobbler程序包

cobbler-web #cobbler的web服务包

pykickstart #cobbler检查kickstart语法错误

httpd #Apache web服务

dhcp #dhcp服务

tftp-server #tftp服务

1）解压Cobbler.zip

# unzip /Cobbler.zip -d / #解压到根目录

# unzip Cobbler/cobbler.zip -d /opt

2)安装cobbler主程序、工具包等

# yum -y install /opt/cobbler/\*.rpm

3.配置

1）配置Cobbler

位置：/etc/cobbler/settings

[root@Cobbler /]# vim /etc/cobbler/settings

next\_server: 192.168.4.168 #设置下一个服务器地址（该处为本机）

server: 192.168.4.168 #设置本机为cobbler服务器

manage\_dhcp: 1 #设置cobbler管理dhcp服务

pxe\_just\_once: 1 #防止客户端重复安装操作系统

提示：

开机启动： 匹配及停止

1.硬盘启动 2.光驱设备 3.U盘 4.网络引导

2）配置cobbler的dhcp

[root@Cobbler /]# vim /etc/cobbler/dhcp.template

:%s /192.168.1/192.168.4/g #全文替换

subnet 192.168.4.0 netmask 255.255.255.0 {

option routers 192.168.4.5;

option domain-name-servers 192.168.4.1;

option subnet-mask 255.255.255.0;

range dynamic-bootp 192.168.4.100 192.168.4.254;

default-lease-time 21600;

max-lease-time 43200;

next-server $next\_server;

3)绝对路径解压cobbler\_boot.tar.gz

# tar -xPf /Cobbler/cobbler\_boot.tar.gz #P以绝对路径解压

# ls /var/lib/cobbler/loaders/ #解压默认放置路径

COPYING.elilo COPYING.yaboot grub-x86\_64.efi menu.c32 README

COPYING.syslinux elilo-ia64.efi grub-x86.efi pxelinux.0 yaboot

4）启动服务

[root@Cobbler /]# systemctl restart cobblerd

[root@Cobbler /]# systemctl enable cobblerd

[root@Cobbler /]# systemctl restart httpd

[root@Cobbler /]# systemctl enable httpd

[root@Cobbler /]# systemctl restart tftp

[root@Cobbler /]# systemctl enable tftp

[root@Cobbler /]# systemctl restart rsyncd

[root@Cobbler /]# systemctl enable rsyncd

5)同步刷新cbbler配置

[root@Cobbler /]# cobbler sync

[root@Cobbler /]# firefox https://192.168.4.168/cobbler\_web

用户名：cobbler

密 码：cobbler

6)cobbler应用

--导入镜像文件

格式：cobbler import --path=挂载点 --name=导入系统命名(随意起)

提示：

cobbler导入的镜像放在：/var/www/cobbler/ks\_mirror

例子：

#mount /dev/cdrom /dvd #挂载

#cobbler import --path=/dvd --name=CnetOS7 #导入镜像

#cobbler list #查看有哪些系统

--删除操作

看有哪些系统

格式：cobbler list

删除菜单信息

格式：cobbler profile remove --name=名称

删除镜像信息

格式：cobbler distro remove --name=名称

例子：

[root@Cobbler /]# cobbler list

distros:

CentOS7-x86\_64

RHEL7-x86\_64

profiles:

CentOS7-x86\_64

RHEL7-x86\_64

[root@Cobbler /]# cobbler profile remove --name=RHEL7-x86\_64

[root@Cobbler /]# cobbler distro remove --name=RHEL7-x86\_64

[root@Cobbler /]# cobbler list

distros:

CentOS7-x86\_64

profiles:

CentOS7-x86\_64

7)自定义应答文件

提示：

开头文件注释行要删除

必须将ks文件放入Cobbler默认存放位置，即：/var/lib/cobbler/kickstart/

--运行system-config-kickstart生成ks文件

[root@cobbler ~]# system-config-kickstart #生成ks文件

--修改kickstart文件

格式：cobbler profile edit --name=名称 --kickstart=/var/lib/cobbler/kickstarts/自定义.cfg

例子：

[root@cobbler-stu251 kickstarts]# cobbler profile edit --name=RHLE7-x86\_64 --kickstart=/var/lib/cobbler/kickstarts/ks7.cfg

--

[root@cobbler ~]# cobbler profile report

[root@cobbler ~]# cobbler sync #同步配置

## rsync同步服务

Remote Sync,远程同步

1.rsync命令用法

格式：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

1)本地同步

rsync [选项..] 本地目录1 本地目录2 #把目录1同步到目录2下

rsync [选项..] 本地目录1/ 本地目录2 #把目录1下的数据同步到目录2下

-a :归档模式，相当于-rlptgoD

-v :显示详细信息

-z :传输过程中启用压缩/解压

-n :测试同步过程，不做实际修改 #与-avz连用

--delete :删除目标文件夹内多余的文档 #与-avz连用

2)远程同步

rsync+ssh

--命令

rsync user@host:远程目录/ #列出ssh服务端资源

rsync [选项..] user@host:远程目录 本地目录 #下行（下载） 不常用

rsync [选项..] 本地目录 user@host:远程目录 #上行（上传） 常用

-a :归档模式，相当于-rlptgoD

-v :显示详细信息

-z :传输过程中启用压缩/解压

-n :测试同步过程，不做实际修改 #与-avz连用

--delete :删除目标文件夹内多余的文档 #与-avz连用

提示：同步目录时在目录后加/，表示同步目录里的数据；同步目录不加/，表示同步此目录。

2.实时同步

1)部署公钥、私钥，实现ssh无密码验证

[root@svr7 /]# ssh-keygen #虚拟机A生成公钥、私钥 一路回车

[root@svr7 /]# ls /root/.ssh #公钥、私钥生成的位置

id\_rsa id\_rsa.pub known\_hosts

[root@svr7 /]# ssh-copy-id root@192.168.4.207 #传递公钥到目标主机（入虚拟机B）

[root@pc207 ~]# ls /root/.ssh #查看上传到虚拟机B的公钥

authorized\_keys #上传到客户机的公钥文件

2)inotify实时同步

--源码安装inotify-tools工具，实现监控目录内容的变化

[root@svr7 /]# mkdir /myrpm

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

[root@svr7 /]# yum -y install gcc make #安装编译环境gcc make

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

[root@svr7 inotify-tools-3.13]# ./configure #生成Makefile文件

[root@svr7 inotify-tools-3.13]# make #编译成二进制文件

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

[root@svr7 inotify-tools-3.13]# which inotifywait #查看命令位置

/usr/local/bin/inotifywait

--inotifywait基本用法

inotifywait [选项] 目标文件夹

-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

-qq:无屏幕输出信息

-e :指定监控的modify move create delete attrib等事件类别

--书写shell脚本

\*利用while循环来反复检查单次监控结果

while [条件]

do

循环执行的操作

done

while inotifywait监控操作 #状态返回值为0即条件成立进入循环

do

循环执行的rsync同步操作

done

例子

[root@svr7 /]# vim /root/rsync.sh #新建脚本

#!/bin/bash

while inotifywait -rqq /opt

do

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

done

[root@svr7 /]# chmod +x /root/rsync.sh #给予文件执行权限

[root@svr7 /]# /root/rsync.sh #运行脚本达到实时同步数据功能

# web服务配置

## nginx

1，源码安装nginx软件包

#yum -y install gcc pcre-devel openssl-devel

gcc:支持c语言,c语言解释器

pcre-devel:支持正则

openssl-devel:支持安全ssl协议，https

[root@proxy ~]# useradd -s /sbin/nologin nginx #出于安全创建运行nginx的用户

[root@proxy ~]# tar -xf nginx-1.10.3.tar.gz

[root@proxy ~]# cd nginx-1.10.3

[root@proxy nginx-1.10.3]# ./configure \

> --prefix=/usr/local/nginx \ //指定安装路径

> --user=nginx \ //指定用户

> --group=nginx \ //指定组

> --with-http\_ssl\_module //开启SSL加密功能

.. ..

--with-模块 #开启模块

--without-模块　　　　　　　　 #禁用模块

2 升级nginx

1）编译新版本nginx软件

[root@proxy ~]# tar -zxvf nginx-1.12.2.tar.gz

[root@proxy ~]# cd nginx-1.12.2

[root@proxy nginx-1.12.2]# ./configure \

> --prefix=/usr/local/nginx \

> --user=nginx \

> --group=nginx \

> --with-http\_ssl\_module

[root@proxy nginx-1.12.2]# make

特别提示：千外不能make install ,否则就真的覆盖已有版本了

2) 备份老的nginx主程序，并使用编译好的新版本nginx替换老版本

[root@proxy nginx-1.12.2]# mv /usr/local/nginx/sbin/nginx \

>/usr/local/nginx/sbin/nginxold

[root@proxy nginx-1.12.2]# cp objs/nginx /usr/local/nginx/sbin/ //拷贝新版本

–杀死老nginx，启动nginx

killadd nginx #停止老nignx

nginx 　　　　　　　#启动

--make upgrade

[root@proxy nginx-1.12.2]# make upgrade //升级

/usr/local/nginx/sbin/nginx -t

nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok

nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful

kill -USR2 `cat /usr/local/nginx/logs/nginx.pid`

sleep 1

test -f /usr/local/nginx/logs/nginx.pid.oldbin

kill -QUIT `cat /usr/local/nginx/logs/nginx.pid.oldbin`

[root@proxy ~]# /usr/local/nginx/sbin/nginx –v //查看版本

3 配置文件

1)日志文件

[root@proxy jluo]# ls /usr/local/nginx/logs/

access.log #每访问一次就有一行记录

error.log 　　　　　　　　　　　　　　　　#服务器的错误记录文件

nginx.pid　　　　　　　　　　　　　　　　#服务器运行就有这个文件

2)nginx.conf

--全局配置文件

worker\_processes 1; #开启进程数

events {

worker\_connections 1024;　　　　　　#一个进程并发量

}

--一个web服务配置

http{

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

##http普通网站

server {

listen 80;

server\_name localhost;

localtion / {

root html;

index index.html index.htm;

}

error\_page 500 502 503 504 /50x.html

location = /50x.html {

root html;

}

}

}

4 nginx用户认证

访问Web页面需要进行用户认证

用户名为：tom，密码为：123456

1）修改/usr/local/nginx/conf/nginx.conf

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

server {

listen 80;

server\_name localhost;

auth\_basic "Input Password:"; //认证提示符,不能是汉字

auth\_basic\_user\_file "/usr/local/nginx/pass"; //认证密码文件

location / {

root html;

index index.html index.htm;

}

}

2）生成密码文件，创建用户及密码

使用htpasswd命令创建账户文件，需要确保系统中已经安装了httpd-tools。[root@proxy ~]# yum -y install httpd-tools

[root@proxy ~]# htpasswd -c /usr/local/nginx/pass tom //创建密码文件

New password:

Re-type new password:

Adding password for user tom

[root@proxy ~]# htpasswd /usr/local/nginx/pass jerry //追加用户，不使用-c选项

New password:

Re-type new password:

Adding password for user jerry

[root@proxy ~]# cat /usr/local/nginx/pass

3）重启Nginx服务

[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload //重新加载配置文件

#请先确保nginx是启动状态，否则运行该命令会报错,报错信息如下：

#[error] open() "/usr/local/nginx/logs/nginx.pid" failed (2: No such file or directory

5基于域名的虚拟主机

实现两个基于域名的虚拟主机，域名分别为www.a.com和www.b.com

对域名为www.a.com的站点进行用户认证，用户名称为tom，密码为123456

1）修改Nginx服务配置，添加相关虚拟主机配置如下

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

server {

listen 80; //端口

server\_name www.a.com; //域名

　　　 auth\_basic "Input Password:"; //认证提示符

auth\_basic\_user\_file "/usr/local/nginx/pass"; //认证密码文件

location / {

root html; //指定网站根路径

index index.html index.htm;

}

}

… …

server {

listen 80; //端口

server\_name www.b.com; //域名

location / {

root www; //指定网站根路径

index index.html index.htm;

}

}

2）创建网站根目录及对应首页文件

[root@proxy ~]# mkdir /usr/local/nginx/www

[root@proxy ~]# echo "www" > /usr/local/nginx/www/index.html

3）重启nginx服务

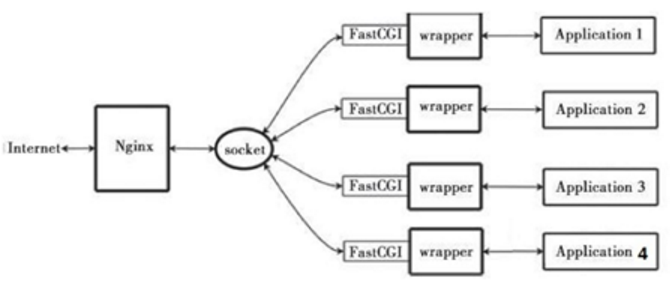
[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

#请先确保nginx是启动状态，否则运行该命令会报错,报错信息如下：

#[error] open() "/usr/local/nginx/logs/nginx.pid" failed (2: No such file or directory)

6　配置Fast-CGI支持PHP网页

创建PHP测试页面，测试使用PHP连接数据库的效果



注意，FastCGI的内存消耗问题，一个PHP-FPM解释器将消耗约25M的内存。

1）查看php-fpm配置文件（实验中不需要修改该文件）

[root@proxy etc]# vim /etc/php-fpm.d/www.conf

[www]

listen = 127.0.0.1:9000 //PHP端口号

pm.max\_children = 32 //最大进程数量

pm.start\_servers = 15 //最小进程数量

pm.min\_spare\_servers = 5 //最少需要几个空闲着的进程

pm.max\_spare\_servers = 32 //最多允许几个进程处于空闲状态

1. 修改Nginx配置文件并启动服务location实现动静分离

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

location / { #匹配地址栏,/优先级最低

root html;

index index.php index.html index.htm;

#设置默认首页为index.php，当用户在浏览器地址栏中只写域名或IP，不说访问什么页面时，服务器会把默认首页index.php返回给用户

}

location ~ \.php$ {　　　　　　　　　　#匹配地址栏.php结尾，支持正则,～(正则匹配)匹配即可．php，区分大小写

root html;

fastcgi\_pass 127.0.0.1:9000; #将请求转发给本机9000端口，PHP解释器

fastcgi\_index index.php;

#fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

include fastcgi.conf;

}

[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

#请先确保nginx是启动状态，否则运行该命令会报错,报错信息如下：

#[error] open() "/usr/local/nginx/logs/nginx.pid" failed (2: No such file or directory)

7　地址重写

所有访问a.html的请求，重定向到b.html;

所有访问192.168.4.5的请求重定向至www.tmooc.cn；

所有访问192.168.4.5/下面子页面，重定向至www.tmooc.cn/下相同的页面；

实现firefox与curl访问相同页面文件，返回不同的内容

rewrite regex replacement flag

rewrite 旧地址 新地址 [选项]

1）修改配置文件(访问a.html重定向到b.html)

--修改Nginx服务配置

location / {

root html;

index index.html index.htm;

rewrite /a\.html /b.html; #修改\.转意.

}

--重新加载配置文件

[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

--客户端测试

[root@client ~]# firefox <http://192.168.4.5/a.html>

2)访问a.html重定向到b.html（跳转地址栏）

--修改Nginx服务配置

location / {

root html;

index index.html index.htm;

rewrite /a\.html /b.html redirect; #修改\.转意.

}

--重新加载配置文件

[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

--客户端测试

[root@client ~]# firefox http://192.168.4.5/a.html

3)修改配置文件(访问192.168.4.5，重定向至www.tmooc.cn)

--修改Nginx服务配置

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

server {

listen 80;

server\_name localhost;

rewrite ^/ <http://www.tmooc.cn/>; #^/匹配根下的所有都会跳转到该页面

location / {

root html;

index index.html index.htm;

# rewrite /a.html /b.html redirect;

}

}

--重新加载配置文件

[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

4)修改配置文件(访问192.168.4.5/下面子页面，重定向至www.tmooc.cn/下相同的页面)

--修改Nginx服务配置

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

server {

listen 80;

server\_name localhost;

rewrite ^/(.\*)$ http://www.tmooc.cn/$1;

location / {

root html;

index index.html index.htm;

# rewrite /a.html /b.html redirect;

}

}

--重新加载配置文件

[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

５)修改配置文件(实现curl和火狐访问相同链接返回的页面不同)

用途：实现部署不同页面(手机端，电脑端)

--知识小记：

没有用’-’表示

192.168.4.254 - 　tom 　[31/Oct/2018:21:46:44 +0800] "　GET / HTTP/1.1" 　200 2 "-" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36"

#log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

# '$status $body\_bytes\_sent "$http\_referer" '

# '"$http\_user\_agent" "$http\_x\_forwarded\_for"';

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf　　　　　　　　　　　#加入如下三行

#这里，~符号代表正则匹配，匹配即可，\*符号代表不区分大小写，==精确匹配

if ($http\_user\_agent ~\* firefox) { //识别客户端firefox浏览器

rewrite ^(.\*)$ /firefox/$1;

}

--地址重写格式【总结】

rewrite 旧地址 新地址 [选项];

last 不再读其他rewrite

break 不再读其他语句，结束请求

redirect 临时重定向

permament 永久重定向

8　Nginx反向代理

调度处理，自动健康检查

\*\*部署实施后端Web服务器

1）部署后端Web1服务器

后端Web服务器可以简单使用yum方式安装httpd实现Web服务，为了可以看出后端服务器的不同，可以将两台后端服务器的首页文档内容设置为不同的内容。

[root@web1 ~]# yum -y install httpd

[root@web1 ~]# echo "192.168.2.100" > /var/www/html/index.html

[root@web1 ~]# systemctl restart httpd

[root@web1 ~]# firewall-cmd --set-default-zone=trusted

[root@web1 ~]# setenforce 0

2）部署后端Web2服务器

[root@web2 ~]# yum -y install httpd

[root@web2 ~]# echo "192.168.2.200" > /var/www/html/index.html

[root@web2 ~]# systemctl restart httpd

[root@web2 ~]# firewall-cmd --set-default-zone=trusted

[root@web2 ~]# setenforce 0

\*\*配置Nginx服务器，添加服务器池，实现反向代理功能

1）修改/usr/local/nginx/conf/nginx.conf配置文件

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

http {

.. ..

#使用upstream定义后端服务器集群，集群名称任意(如webserver)

#使用server定义集群中的具体服务器和端口

upstream webserver {

server 192.168.2.100:80;

server 192.168.2.200:80;

}

.. ..

server {

listen 80;

server\_name localhost;

location / {

#通过proxy\_pass将用户的请求转发给webserver集群

proxy\_pass http://webserver;

}

}

2）重启nginx服务

[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

#请先确保nginx是启动状态，否则运行该命令会报错,报错信息如下：

#[error] open() "/usr/local/nginx/logs/nginx.pid" failed (2: No such file or directory)

3）客户端使用浏览器访问代理服务器测试轮询效果

[root@client ~]# curl http://192.168.4.5 //使用该命令多次访问查看效果

[root@client ~]# curl http://192.168.4.5 //使用该命令多次访问查看效果

9 配置upstream服务器集群池属性

1）设置失败次数，超时时间，权重

weight可以设置后台服务器的权重，max\_fails可以设置后台服务器的失败次数，fail\_timeout可以设置后台服务器的失败超时时间。

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

http {

.. ..

upstream webserver {

server 192.168.2.100 weight=1 max\_fails=1 fail\_timeout=30;

server 192.168.2.200 weight=2 max\_fails=2 fail\_timeout=30;

server 192.168.2.101 down;

}

#weight设置服务器权重值，默认值为1

#max\_fails设置最大失败次数

#fail\_timeout设置失败超时时间，单位为秒

#down标记服务器已关机，不参与集群调度

.. ..

server {

listen 80;

server\_name localhost;

location / {

proxy\_pass http://webserver;

}

}

2）重启nginx服务

[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

#请先确保nginx是启动状态，否则运行该命令会报错,报错信息如下：

#[error] open() "/usr/local/nginx/logs/nginx.pid" failed (2: No such file or directory)

3）关闭一台后端服务器（如web1）

[root@web1 ~]# systemctl stop httpd

4）客户端使用浏览器访问代理服务器测试轮询效果

[root@client ~]# curl http://192.168.4.5 //使用该命令多次访问查看效果

5）再次启动后端服务器的httpd（如web1）

[root@web1 ~]# systemctl start httpd

6）客户端再次使用浏览器访

问代理服务器测试轮询效果

[root@client ~]# curl http://192.168.4.5 //使用该命令多次访问查看效果

７)配置upstream服务器集群的调度算法(防止重复登录等)

--设置相同客户端访问相同Web服务器

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

http {

.. ..

upstream webserver {

#通过ip\_hash设置调度规则为：相同客户端访问相同服务器

ip\_hash;

server 192.168.2.100 weight=1 max\_fails=2 fail\_timeout=10;

server 192.168.2.200 weight=2 max\_fails=2 fail\_timeout=10;

}

.. ..

server {

listen 80;

server\_name www.tarena.com;

location / {

proxy\_pass http://webserver;

}

}

ip\_hash　　#对前三个8字节ip进行hash(md5sum加密)，md5sum%２判断是否是同一个ip地址

--重启nginx服务

[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

#请先确保nginx是启动状态，否则运行该命令会报错,报错信息如下：

#[error] open() "/usr/local/nginx/logs/nginx.pid" failed (2: No such file or directory)

--客户端使用浏览器访问代理服务器测试轮询效果

[root@client ~]# curl http://192.168.4.5 //使用该命令多次访问查看效果

10Nginx的TCP/UDP调度器

ngx\_stream\_core\_module模块，使用—with-stream开启该模块

提示：nginx1.9开始才支持

配置Nginx服务器，添加服务器池，实现TCP/UDP反向代理功能

1）部署nginx服务器

编译安装必须要使用--with-stream参数开启4层代理模块。

[root@proxy ~]# yum –y install gcc pcre-devel openssl-devel //安装依赖包

[root@proxy ~]# tar -xf nginx-1.12.2.tar.gz

[root@proxy ~]# cd nginx-1.12.2

[root@proxy nginx-1.12.2]# ./configure \

> --with-http\_ssl\_module //开启SSL加密功能

> --with-stream //开启4层反向代理功能

[root@proxy nginx-1.12.2]# make && make install //编译并安装

2）修改/usr/local/nginx/conf/nginx.conf配置文件

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

stream {

upstream backend {

server 192.168.2.100:22; //后端SSH服务器的IP和端口

server 192.168.2.200:22;

}

server {

listen 12345; //Nginx监听的端口

proxy\_connect\_timeout 1s;

proxy\_timeout 3s;

proxy\_pass backend;

}

}

http {

.. ..

}

3）重启nginx服务

[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

#请先确保nginx是启动状态，否则运行该命令会报错,报错信息如下：

#[error] open() "/usr/local/nginx/logs/nginx.pid" failed (2: No such file or directory)

4）客户端使用访问代理服务器测试轮询效果

[root@client ~]# ssh 192.168.4.5 -p 12345 //使用该命令多次访问查看效果

11自定义报错页面

1）优化前，客户端使用浏览器访问不存在的页面，会提示404文件未找到

[root@client ~]# firefox http://192.168.4.5/xxxxx //访问一个不存在的页面

2）修改Nginx配置文件，自定义报错页面

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

error\_page 404 /40x.html; //自定义错误页面

.. ..

[root@proxy ~]# vim /usr/local/nginx/html/40x.html //生成错误页面

Oops,No NO no page …

[root@proxy ~]# nginx -s reload

#请先确保nginx是启动状态，否则运行该命令会报错,报错信息如下：

#[error] open() "/usr/local/nginx/logs/nginx.pid" failed (2: No such file or directory)

3）优化后，客户端使用浏览器访问不存在的页面，会提示自己定义的40x.html页面

[root@client ~]# firefox http://192.168.4.5/xxxxx //访问一个不存在的页面

12如何查看服务器状态信息（非常重要的功能）

1）编译安装时使用--with-http\_stub\_status\_module开启状态页面模块

[root@proxy ~]# yum -y install gcc pcre-devel openssl-devel //安装常见依赖包

[root@proxy ~]# tar -zxvf nginx-1.12.2.tar.gz

[root@proxy ~]# cd nginx-1.12.2

[root@proxy nginx-1.12.2]# ./configure \

> --with-http\_ssl\_module //开启SSL加密功能

> --with-stream //开启TCP/UDP代理模块

> --with-http\_stub\_status\_module //开启status状态页面

[root@proxy nginx-1.12.2]# make && make install //编译并安装

2）启用Nginx服务并查看监听端口状态

ss命令可以查看系统中启动的端口信息，该命令常用选项如下：

-a显示所有端口的信息

-n以数字格式显示端口号

-t显示TCP连接的端口

-u显示UDP连接的端口

-l显示服务正在监听的端口信息，如httpd启动后，会一直监听80端口

-p显示监听端口的服务名称是什么（也就是程序名称）

注意：在RHEL7系统中可以使用ss命令替代netstat命令，功能一样，选项一样。

[root@proxy ~]# /usr/local/nginx/sbin/nginx

[root@proxy ~]# netstat -anptu | grep nginx

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 10441/nginx

[root@proxy ~]# ss -anptu | grep nginx

3）修改Nginx配置文件，定义状态页面

[root@proxy ~]# cat /usr/local/nginx/conf/nginx.conf

… …

location /status {

stub\_status on;

#allow IP地址;

#deny IP地址;

}

… …

[root@proxy ~]# nginx

4）优化后，查看状态页面信息

[root@proxy ~]# curl http://192.168.4.5/status

Active connections: 1

server accepts handled requests

10 10 3

Reading: 0 Writing: 1 Waiting: 0

Active connections：当前活动的连接数量。

Accepts：已经接受客户端的连接总数量。

Handled：已经处理客户端的连接总数量。

（一般与accepts一致，除非服务器限制了连接数量）。

Requests：客户端发送的请求数量。

Reading：当前服务器正在读取客户端请求头的数量。

Writing：当前服务器正在写响应信息的数量。

Waiting：当前多少客户端在等待服务器的响应。

Http1.0 一次连接一次请求

http1.1 一次连接多次请求

keepalive\_timeout 30; 　#保持连接时间，超时断开tcp连接

测试：

[root@client ~]# yum -y isntall httpd-tool

[root@client ~]# ab -c 200 -n 200 http://192.168.4.5/

c：client 客户端

n：number 访问数量

100个客户端访问200次

13优化Nginx并发量

1）优化前使用ab高并发测试

[root@proxy ~]# ab -n 2000 -c 2000 http://192.168.4.5/

Benchmarking 192.168.4.5 (be patient)

socket: Too many open files (24) //提示打开文件数量过多

2）修改Nginx配置文件，增加并发量

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

worker\_processes 2; //与CPU核心数量一致

events {

worker\_connections 65535; //每个worker最大并发连接数,尽量写大一点

use epoll;

}

.. ..

[root@proxy ~]# nginx -s reload

3）优化Linux内核参数（最大文件数量）

[root@proxy ~]# ulimit -a //查看所有属性值

[root@proxy ~]# ulimit -Hn 100000 //设置硬限制（临时规则）

[root@proxy ~]# ulimit -Sn 100000 //设置软限制（临时规则）

[root@proxy ~]# vim /etc/security/limits.conf

.. ..

\* soft nofile 100000

\* hard nofile 100000

#该配置文件分4列，分别如下：

#用户或组 硬限制或软限制 需要限制的项目 限制的值

4）优化后测试服务器并发量（因为客户端没调内核参数，所以在proxy测试）

[root@proxy ~]# ab -n 2000 -c 2000 http://192.168.4.5/

5)ss

ss是Socket Statistics的缩写。顾名思义，ss命令可以用来获取socket统计信息，它可以显示和netstat类似的内容。但ss的优势在于它能够显示更多更详细的有关TCP和连接状态的信息，而且比netstat更快速更高效。

ss -l 显示本地打开的所有端口

ss -pl 显示每个进程具体打开的socket

ss -t -a 显示所有tcp socket

ss -u -a 显示所有的UDP Socekt

ss -o state established '( dport = :smtp or sport = :smtp )' 显示所有已建立的SMTP连接

ss -o state established '( dport = :http or sport = :http )' 显示所有已建立的HTTP连接

ss -x src /tmp/.X11-unix/\* 找出所有连接X服务器的进程

ss -s 列出当前socket详细信息:

14 优化Nginx数据包头缓存

1）优化前，使用脚本测试长头部请求是否能获得响应

[root@proxy ~]# cat lnmp\_soft/buffer.sh

#!/bin/bash

URL=http://192.168.4.5/index.html?

for i in {1..5000}

do

URL=${URL}v$i=$i

done

curl $URL //经过5000次循环后，生成一个长的URL地址栏

[root@proxy ~]# ./buffer.sh

.. ..

<center><h1>414 Request-URI Too Large</h1></center> //提示头部信息过大

2）修改Nginx配置文件，增加数据包头部缓存大小

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

http {

client\_header\_buffer\_size 1k; //默认请求包头信息的缓存

large\_client\_header\_buffers 4 4k; //大请求包头部信息的缓存个数与容量,生产环境够用了

.. ..

}

[root@proxy ~]# nginx -s reload

3）优化后，使用脚本测试长头部请求是否能获得响应

[root@proxy ~]#cat cat buffer.sh

#!/bin/bash

URL=http://192.168.4.5/index.html?

for i in {1..5000}

do

URL=${URL}v$i=$i

done

curl $URL

[root@proxy ~]# ./buffer.sh

15浏览器本地缓存静态数据

1）使用Firefox浏览器查看缓存

以Firefox浏览器为例，在Firefox地址栏内输入about:cache将显示Firefox浏览器的缓存信息，如图-3所示，点击List Cache Entries可以查看详细信息

1. 清空firefox本地缓存数据，如图-4所示

3）修改Nginx配置文件，定义对静态页面的缓存时间

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

server {

listen 80;

server\_name localhost;

location / {

root html;

index index.html index.htm;

}

location ~\* \.(jpg|jpeg|gif|png|css|js|ico|xml)$ {

expires 30d; //定义客户端缓存时间为30天

}

}

[root@proxy ~]# cp /usr/share/backgrounds/day.jpg /usr/local/nginx/html

[root@proxy ~]# nginx -s reload

#请先确保nginx是启动状态，否则运行该命令会报错,报错信息如下：

#[error] open() "/usr/local/nginx/logs/nginx.pid" failed (2: No such file or directory)

4）优化后，使用Firefox浏览器访问图片，再次查看缓存信息

[root@client ~]# firefox http://192.168.4.5/day.jpg

在Firefox地址栏内输入about:cache，查看本地缓存数据，查看是否有图片以及过期时间是否正确。

16日志切割

日志文件越来越大怎么办？单个文件10G? 如何切割？（非常常见的面试题）

步骤：

1. 把旧的日志重命名

2.kill USR1 PID(nginx的进程PID号)

1）手动执行

备注：/usr/local/nginx/logs/nginx.pid文件中存放的是nginx的进程PID号。

[root@proxy ~]# mv access.log access2.log

[root@proxy ~]# kill -USR1 $(cat /usr/local/nginx/logs/nginx.pid)

2）自动完成

每周5的03点03分自动执行脚本完成日志切割工作。

[root@proxy ~]# vim /usr/local/nginx/logbak.sh

#!/bin/bash

date=`date +%Y%m%d`

logpath=/usr/local/nginx/logs

mv $logpath/access.log $logpath/access-$date.log

mv $logpath/error.log $logpath/error-$date.log

kill -USR1 $(cat $logpath/nginx.pid)

[root@proxy ~]# crontab -e

03 03 \* \* 5 /usr/local/nginx/logbak.sh

17对页面进行压缩处理

提示：对多媒体文件不要压缩，太大了

1. 修改Nginx配置文件

[root@proxy ~]# cat /usr/local/nginx/conf/nginx.conf

http {

.. ..

gzip on; //开启压缩

gzip\_min\_length 1000; //当返回内容大于此值时才会使用gzip进行压缩,以K为单位,当值为0时，所有页面都进行压缩

gzip\_comp\_level 4; //压缩比率　　（1－9），数字越高，花费时间越多，压缩效率越好，一般选中间

gzip\_types text/plain text/css application/json application/x-javascript text/xml application/xml application/xml+rss text/javascript;

//对特定文件压缩，类型参考mime.types，/usr/local/nginx/conf/mime.types

.. ..

}

17服务器内存缓存

1）如果需要处理大量静态文件，可以将文件缓存在内存，下次访问会更快。

http {

open\_file\_cache max=2000 inactive=20s;

open\_file\_cache\_valid 60s;

open\_file\_cache\_min\_uses 5;

open\_file\_cache\_errors off; #关闭缓存过期报错

//设置服务器最大缓存2000个文件句柄，关闭20秒内无请求的文件句柄

//文件句柄的有效时间是60秒，60秒后过期

//只有访问次数超过5次会被缓存

## http服务

提示：网页文件默认目录：/var/www/html

1、基于b/s架构服务的网页服务

服务器提供网页

浏览器下载并显示网页

HTML：Hyper Text Markup Language 超文本标记语言

HTTP：Hyper Text Transfer Protocol 超文本传输协议

--默认端口 80

2、安装启服务

--安装软件

#yum -y install httpd

--启动软件

# systemctl restart httpd

# systemctl enable httpd 设置开机自启

3、默认设置

主配置文件：/etc/httpd/conf/httpd.conf

调用配置文件：/etc/httpd/conf.d/\*.conf

1）Listen:监听地址端口（默认80）

Listen 80

2）ServerName：本站点注册的DNS域名（空缺，默认注释了）

ServerName server0.example.com:80

#systemctl restart httpd

#firefox server0.example.com

3）DoucumentRoot:网页文件根目录（默认：/var/www/html）

例子

DocumentRoot "/var/www/myweb"

--网络路径与服务器实际路径

firefox server0.example.com---》DNS

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

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

浏览器： server0.example.com

服务端：/var/www/myweb

浏览器： server0.example.com/abc

服务端：/var/www/myweb/abc

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

4、虚拟主机

1)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

2)作用

一台主机提供多个web页面

3)基于域名的虚拟主机

<VirtualHost ip地址：端口号> #ip地址一般写\*

ServerName 站点的DNS名称

DocumentRoot 站点的网页根目录

</VirtualHost>

--配置虚拟主机

建议写入调用配置自己定义的文件中(如:/etc/httpd/conf.d/virtualhost.conf)

例子：

vim /etc/httpd/conf.d/virtualhost.conf

<VirtualHost \*:80>

ServerName www.jluo.com

DocumentRoot /var/www/test

</VirtualHost>

注意:一旦使用虚拟主机功能,所有网站都要使用虚拟web主机完成

--虚拟主机匹配原则

由上到下,匹配即停止

端口优先级大于域名

5、web访问控制(DocumentRoot)

httpd运行身份(用户/组):apache

1)服务配置

--使用<Directory>配置区段

特点:

每个文件夹自动继承其父目录的访问权限

除非针对个别需单独写配置文件

<Directory /webroot>

.. ..

Require all denied|granted #拒绝所有访问|允许所有访问

</Directory>

<Directory /webroot>

.. ..

Require ip ip地址或者网段... #只允许ip地址或者该网段的地址访问

</Directory>

提示:访问控制配置内容建议新建一个专门配置访问控制的配置文件进行配置

2)客户端访问服务端资源

--防火墙

--服务本身权限

--本地目录权限

--SELinux安全策略

3)SELinux的安全上下文(标签:起到标识作用)

--查看所有SELinux标记

#semanage fcontext -l

#ls -Zd /var/www 查看目录的上下文

#ls -Zd /webroot 查看目录的上下文

--格式

方式1

chcon --help |head -5 查看--reference=参考文件 文件...

chcon [-R] --reference=模板目录 新目录.....

例子

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

# ll -Z /webroot

-rw-r--r--. root root system\_u:object\_r:httpd\_sys\_content\_t:s0 index.html

标签

killall 程序

#killall firefox

6.部署动态网站

1)编写python网页文件,放到webapp0.example.com根目录下

vim /webinfo.wsgi

#!/usr/bin/env python

import time

def application (environ, start\_response):

response\_body = 'UNIX EPOCH time is now: %s\n' % time.time()

status = '200 OK'

response\_headers = [('Content-Type', 'text/plain'),

('Content-Length', '1'),

('Content-Length', str(len(response\_body)))]

start\_response(status, response\_headers)

return [response\_body]

显示当前系统的Unix时间戳

Unix时间戳:自1970-1-1 0:0:0到当前系统时间经历的秒数

2)方便用户访问,网页跳转

--修改配置文件

配置字段格式: Alias 网络路径 实际本地路径

例子

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd02

Alias / /var/www/nsd02/webinfo.wsgi #页面跳转

</VirtualHost>

3)安装mod\_wsgi软件包,专门用于翻译python代码页面

# yum -y install mod\_wsgi

--配置

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd02

WsgiScriptAlias / /var/www/nsd02/webinfo.wsgi #页面跳转

</VirtualHost>

4)此虚拟主机侦听在端口8909

--修改配置文件内容

Listen 8909

<VirtualHost \*:8909>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd02

WsgiScriptAlias / /var/www/nsd02/webinfo.wsgi #页面跳转

</VirtualHost>

--SELinux非默认端口的开放

#semanage port -l |grep http

命令:semanage port -a -t http\_port\_t -p tcp 端口号 #添加selinux开放端口

提示:内存需要2G以上

7、安全web

Public Key Infrastructure,公钥基础设施

公钥：主要用来加密数据

私钥：主要用来解密数据（与相应的公钥匹配）

数字证书：证明拥有者的合法性/权威性（单位名称、有效期、公钥、颁发机构及签名）

Certificaate 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

部署网站证书

数字证书存放位置：/etc/pki/tls/certs/

--下载证书存放位置

#cd /etc/pki/tls/certs/ #数字证书存放目录

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

--部署根证书

#cd /etc/pki/tls/certs/

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

--部署私钥(解密)

#cd /etc/pki/tls/private/

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

--安装支持https的mod\_ssl软件

# yum -y install mod\_ssl

--修改配置文件

位置：/etc/httpd/conf.d/ssl.conf

提示：在末行模式下输入 set nu 开启行号,关闭 set no nu

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

59 DocumentRoot "/var/www/html" #去掉注释

60 ServerName www0.example.com:443 #去掉注释

100 SSLCertificateFile /etc/pki/tls/certs/server0.crt #证书位置

107 SSLCertificateKeyFile /etc/pki/tls/private/server0.key #证书密钥

122 SSLCACertificateFile /etc/pki/tls/certs/example-ca.crt #去掉注释，修改根证书路径

--书写用于测试网站

#echo '<h1>https test</h1>' > /var/www/html/index.html

--重启服务

#systemctl restart httpd

# 第六章 磁盘管理

## RAID磁盘阵列

热备：不需要人操作

冷备：需要人操作

• 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

1.RAID0,条带模式

--同一个文档分散存放在不同磁盘

--并行写入以提高效率，无容错功能

--至少需要两块磁盘组成

--存储利用率100%

2.RAID1,镜像模式

--一个文档复制多份，分别写入不同磁盘

--多份拷贝提高可靠性，效率无提升

--至少需要两块磁盘组成

--存储利用率小于或等与50%

3.RAID5，高性价比模式

--相当于RAID0和RAID1的折中方案

--至少需要一块磁盘的容量来存放校验数据

--至少需要三块磁盘组成

--存储利用率是(n-1)/n

4.RAID6,高性价比/可靠模式

--相当于扩展的RAID5磁盘阵列，提供2份独立检验方案

--需要至少两块磁盘的容量来存放校验数据

--至少需要4块磁盘组成

--存储利用率是(n-2/n)

## 分区基础

扇区的大小为512字节（一般情况下）

1、分区的一般步骤：

识别硬盘 >分区规划>格式化>挂载使用

单位：

1TB=1000GB

1TiB=1024GiB

1EB=1000PB

1PB=1000TB

2、分区模式

1)MBR/msdos 分区模式

分区类型：主分区，扩展分区，逻辑分区

--1-4个主分区，或者0-3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

--扩展分区不支持格式化

2）GPT分区模式

--可以分128个主分区，最大可支持18EP硬盘分区

## MBR/msdos 分区模式

1、识别硬盘

[root@client0 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

fd0 2:0 1 4K 0 disk

sda 8:0 0 20G 0 disk

├─sda1 8:1 0 1G 0 part /boot

└─sda2 8:2 0 19G 0 part

├─centos-root 253:0 0 17G 0 lvm /

└─centos-swap 253:1 0 2G 0 lvm [SWAP]

sdb 8:16 0 10G 0 disk #此硬盘为刚添加的硬盘

sr0 11:0 1 4.2G 0 rom

2、分区规划

[root@client0 ~]# fdisk /dev/sdb

Command (m for help): n

Partition type:

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

e extended

Select (default p):

Using default response p

Partition number (1-4, default 1):

First sector (2048-20971519, default 2048):

Using default value 2048

Last sector, +sectors or +size{K,M,G} (2048-20971519, default 20971519): +2G

Command (m for help): p

Device Boot Start End Blocks Id System

/dev/sdb1 2048 4196351 2097152 83 Linux

Command (m for help): w

[root@client0 ~]# lsblk #查看下

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

fd0 2:0 1 4K 0 disk

sda 8:0 0 20G 0 disk

├─sda1 8:1 0 1G 0 part /boot

└─sda2 8:2 0 19G 0 part

├─centos-root 253:0 0 17G 0 lvm /

└─centos-swap 253:1 0 2G 0 lvm [SWAP]

sdb 8:16 0 10G 0 disk

└─sdb1 8:17 0 2G 0 part #刚才分区的

sr0 11:0 1 4.2G 0 rom

提示：

1）要是显示出来没有，运行命令：partprobe 刷新分区

2）要是创建出错了，如sda2分错了，要先按照分区号从大到小的顺序依次删除

3、格式化，赋予空间文件系统

[root@client0 ~]# mkfs.ext4 /dev/sdb1 #格式化ext4格式

[root@client0 ~]# blkid /dev/sdb1 #查看分区id

/dev/sdb1: UUID="6034a802-3223-41cc-b4ad-598d071d7d6e" TYPE="ext4"

提示：

ext4：小文件存储

xfs：大文件存储

4、挂载使用分区

[root@client0 /]# mount /dev/sdb1 /cluo/ #挂载

[root@client0 /]# df -h #显示正在挂载的设备使用情况

Filesystem Size Used Avail Use% Mounted on

/dev/mapper/centos-root 17G 3.4G 14G 20% /

devtmpfs 899M 0 899M 0% /dev

tmpfs 916M 0 916M 0% /dev/shm

tmpfs 916M 9.8M 906M 2% /run

tmpfs 916M 0 916M 0% /sys/fs/cgroup

/dev/sda1 1014M 170M 845M 17% /boot

tmpfs 184M 0 184M 0% /run/user/0

/dev/sdb1 2.0G 6.0M 1.8G 1% /cluo #挂载成功的分区

5、开机自动挂载

配置：/etc/fstab

0:不备份 1备份

建议：0 0：不检测 1：检测

-设备路径 挂载点 类型 参数 备份标记 检测顺序

#vim /etc/fstab

UUID="f6cee922-f44d-4012-b761-4f40230a078a" /cluo ext4 defaults 0 0

[root@ client0 ~]# mount -a

检测/etc/fstab开机自动挂载配置文件格式是否正确

检测/etc/fstab中书写完成，但当前没有挂载的设备进行挂载

## GPT分区模式

parted大容量分区

1、特点

专门用于gpt分区方案，可以分128个主分区，最大可支持18EP硬盘分区

2、分区命令

--分区顺序

#parted 设备名称

(parted)mktable gpt #清除整块硬盘

(parted)print #输出分区表

(parted)mkpart #划分分区

(parted)unit GB #设置显示单位

例子

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

(parted) mktable gpt

(parted) mkpart

分区名称？ []? lvdisk1 #分区名称

文件系统类型？ [ext2]? ext4 #文件类型，无实际意义

起始点？ 0G

结束点？ 4G

(parted) mkpart

分区名称？ []? lvdisk2

文件系统类型？ [ext2]? ext4

起始点？ 4G

结束点？ 6G

(parted) print

Number Start End Size File system Name 标志

1 1049kB 4000MB 3999MB lvdisk1

2 4000MB 6000MB 2000MB lvdisk2

## LVM逻辑卷的管理

1、作用

1）可以整合分散的空间

2）容量大小可以扩大

-->零散的空闲存储-->整合的虚拟磁盘-->虚拟分区

--（物理卷PV）-->卷组（VG）-->逻辑卷(LV)

将众多的物理卷（PV），组成卷组（VG），再从卷组中划出逻辑卷（LV）

P：Physical 物理

V：volume 卷

G：group 组

L：Logical 逻辑的

2、逻辑卷创建

pvs #查看物理卷信息

vgs #查看卷组信息

lvs #查看逻辑卷信息

1）创建物理卷PV

格式：pvcreate 设备路径1 设备路径2......

#pvcreate /dev/vdc1 /dev/vdc2

2）创建卷组VG 可以直接创建卷组

格式：vgcreate 卷组名称 设备路径1 设备路径2......

#vgcreate systemvg /dev/vdc[1-2]

3）创建逻辑卷LV

格式：lvcreate -n 逻辑卷名 -L 逻辑卷大小 卷组名

#lvcreate -n mylv -l 16G systemvg

4）格式化逻辑卷

#mkfs.ext4 /systemvg/mylv #格式化逻辑卷

#blkid /systemvg/mylv #查看设备文件系统

#vim /etc/fstab #挂载逻辑卷

3、逻辑卷的扩展

1）卷组有足够的剩余空间

1.1扩展空间大小

lvextend -L 18G /dev/systemvg/mylv

1.2扩展文件系统大小

resize2fs #扩展ext4文件系统

xfs\_growfs #扩展xfs文件系统

2）卷组没有足够的剩余空间

扩展卷组

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

然后接着第一步

4、缩减空间（了解内容）

ext4：支持缩减

xfs ：不支持缩减

5、卷组划分空间单位

单位：PE （默认1PE=4MiB）

1）修改方式

卷组已存在

vgchange -s PE大小 卷组名

vgchange -s 16M systemvg

卷组不存在

vgcreate -s PE大小 卷组名 设备路径

利用PE个数创建逻辑卷

lvcreate -l 个数 -n 逻辑卷名 卷组名

例如：

lvcreate -l 80 -n lvtest systemvg

6、逻辑卷的删除

先删除LV-->VG-->PV

1）先删除逻辑卷本身（先卸载）

umount 设备路径

lvremove 设备路径

2）再删除卷组

先删除卷组下的所有逻辑卷，再删除卷组

格式：vgremove 卷组名

3）最后可删除物理卷

## 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

交换文件：以文件模拟的设备充当的交换空间

--命令

#swapon -s #查看交换空间

#mkswap 设备名 #格式交换空间

例子

#mkswap /dev/vdb1

#swapon 设备名 #启用该交换分区

例子

#swapon /dev/vdb1

#swapoff 设备名 #停用该交换分区

例子

#swapoff /dev/vdb1

--交换分区开机自动启用

#vim /etc/fstab

/dev/vdb1 swap swap defaults 0 0

#swapon -a #检测swap交换空间配置

# 第七章 shell脚本

脚本：一个可运行的文件，可以实现某种功能

## 组成

#！环境声明(这个程序由哪一个解释器解释，默认/bin/bash)

#注释文本

可执行代码

例子：

#！/bin/bash

#我的第一个程序

echo this is my first shell

## 2.技巧

1）| 管道

2)重定向

>:只收集前面命令的正确输出信息写入文本文件中(覆盖源文本文件内容)

2>:只收集前面命令的错误输出信息写入文本文件中(覆盖源文本文件内容)

&>:只收集前面命令的正确和错误输出信息都写入文本文件中(覆盖源文本文件内容)

>>:

>&2:将输出变为错误输出

## 3.黑洞设备

位置:/dev/null

## 4.语法

read:产生交互,记录用户在键盘上的输入,将记录的信息赋值给变量

-p:提示信息

read -p '提示信息' 变量

### 变量

1)定义变量

变量名=变量值

注意:

--等号两边不能由空格

--只能由字母,数字,下划线组成,区分大小写

--只能由字母,下划线开头

--变量名不能是关键字

2)引用变量

$变量名

${变量名}

$()或反撇号~ ~:将命令的输出,作为另一个命令的参数,参与下一个命令的运行

如:mkdir data-$(date +%F)

3)变量类型

--环境变量:变量名一般大写,由系统定义完成,用户直接使用

如:

USER=永远存储当前系统用户名

--位置变量:由系统定义完成,用户直接使用(非交互式)

在执行脚本时,提供命令行参数

表示$n,n为序号

如:

$1,$2,$3........,${44}.....

--预定义变量:由系统定义完成,用户直接使用

用来保存脚本程序的执行信息

直接使用这些变量

不能直接为这些变量赋值

如:

$# 已加载的位置变量的个数(计算用户输入位置变量个数)

$\* 所有位置变量的值

$? 程序退出后的状态值,0标识正常,非0即异常

--自定义变量:用户自主设置,修改,使用

### 条件测试及选择

单引号 '' :取消所有特殊字符的含义

双引号 "":

1)测试选项

注意:[ ] 两边都要由空格,每个参数都至少有一个空格

--检查文档状态

-e:文档存在为真

[ -e /etc ]

-d:文档存在,并且为目录才为真

[ -d /etc ]

-f:文档存在,并且为文件才为真

-r:文档存在,并且具备读取权限才为真

-w:文档存在,并且具写权限才为真

-x:文档存在,并且具备执行权限才为真

--比较整数大小

-gt:大于

-ge:大于等于

-eq:等于

-ne:不等于

-lt:小于

-le:小于等于

--字符串比对

==:两个字符串相等为真

!=:两个字符串不相等为真

### 条件处理

1)if双分支

if [条件测试];then

命令序列xx

else

命令序列yy

fi

2)if多分支

if [条件测试1];then

命令序列xx

elif [条件测试2];then

命令序列zz

elif [条件测试3];then

命令序列ff

else

命令序列yy

fi

### 循环

1)for

for 变量名 in 值列表

do

命令序列

done

造数:{起始数值..结束数值}

# 第八章 数据库

数据库：存放数据的仓库

--社区开源版-MariaDB

--甲骨文的-oracle mysql

-IBM的DB2

-微软的sqlserver

## MariaDB

1)MariaDB

mariadb-server:提供服务端有关的系统程序

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

2)重启服务

[root@server0 /]# systemctl restart mariadb

[root@server0 /]# systemctl enable mariadb

2.简单的使用

1)[root@server0 /]# mysql

2）命令

MariaDB [(none)]> show databases; #查看所有数据库

MariaDB [(none)]> create database nsd1808; #建库

MariaDB [(none)]> drop database nsd1808; #删库

MariaDB [(none)]> exit #退出数据库

3）格式 无空格

mysqladmin [-u 用户名] [-p旧密码] passwd '新密码' #设置密码

例子：[root@server0 /]# mysqladmin -u root password '123'

[root@server0 /]# mysql -u root -p #交互式登陆数据库

[root@server0 /]# mysql -u root -p密码 #非交互式登陆

例子：[root@server0 /]# mysql -u root -p123

4)主配置文件

/etc/my.conf

5)进行表格操作

----查看表格字段

desc 表格名

----单表查询

格式：

select 字段名... from 表名 [where 条件.........];

例子：

select \* from base ;

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

当前在mysql库里想要查询库nsd中的表base 库名.表名

----多表查询

格式：

select table1.字段名(可多个)，table2.字段名(可有多个).... from table1,table2... [where 条件.......];

提示--要是单表中的字段是特有的可省略表名，可直接写字段名

例子：

----插入

insert into 表名(表字段名1，表字段名2....)values(值1，值2,......);

例子

MariaDB [jluo]> insert into student(id,name,age)values(1,'jack',18);

----删除

delete from 表名 [where 条件....]

例子

delete from base where id=1;

----修改

格式：

update 表名 set 字段名=值..... [where 条件....];

例子：

update student set name='小强',age=18 where id=1400;

6）数据库的授权

--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’

例子：

MariaDB [nsd]> grand select on nsd.\* to lisi@localhost identified by '123';

7)禁止空密码登陆

flush privileges :刷新数据库所有策略

----删除user表中密码为空的用户，需刷新数据库(flush privileges)

测试：

[root@server0 /]# mysql -u root -h 127.0.0.1

ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)