**2018年11月1号**

信息交互

互联网的本质:客户端←-------→服务器

**互联网产业流程:**

开发------>UI设计------>web前端-------->测试------->运维(网络安全)------->网络营销

1. **运维:服务器运行与维护**

低层（单机操作）：

基础运维：电脑维修

域名解析服务器，将域名解析为IP地址

桌面运维：电信工作人员

低端（局部网络/服务）：

服务器运维工程师：web，DNS，邮件......

网络运维工程师：搭建局部网络，负责内部网络通信

中端（平台）进阶：

运维开发：自动化运维（处理重复作业，效率极高），涉及代码逻辑，走开发

全栈工程师：运维，开发，安全

云计算工程师：平台，涉及代码逻辑

高端

架构师：系统架构师、软件架构师

**二、开发：实现客户的需求，和人打交道**

低端（码农）

语言: java, c(c，c#, c++), phthon, .net, go, per

可替代性

中端开发市场：（代码熟练工）

学习多门语言

高端（大数据）

**三、**

DOS--------->windows

Unix

Linux

Unix生辰：1970-1-1

集群：将零散的资源整合

虚拟化：将整体的资源分割

对接：

阿里：系统的维护，安全维护，资源分配

内核：调动所有硬件资源（用户---->内核------>硬件）

**2018年11月2号**

**虚拟化技术**

虚拟机：安装在虚拟平台上，虚拟平台搭载在硬件上

（原生型）虚拟机：硬件--->虚拟平台--->虚拟机 可以类寄存式

（寄存式）虚拟机：硬件--->系统--->虚拟平台--->虚拟机

每个虚拟机可以安装不同的操作系统

**Tcp/ip协议：**

规定了主机之间通信的三要素：1）IP地址 2）子网掩码 3）IP路由

IP地址：标识网络节点的网络地址

IP地址的分类（根据第一字节的数字）

用于一般计算机网络

A类 1-126 网 主 主 主

B类 128-191 网 网 主 主

C类 192-223 网 网 网 主

组播和科研

D类 224-239 组播：先上传再下载

E类 240-254 科研

127.0.0.1--------代表本机 （127-----A，但是用不上）

0:番号-------代表1-254

255：广播

IP私有地址：

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

IP公共有地址：在因特网中使用的地址

子网掩码：标识网络位和主机位

A类：255.0.0.0

B类：255.255.0.0

C类：255.255.255.0

网关：从一个网络连接到另一个网络的“关口”，通常是一台路由器或者防火墙/接入服务器

Windows-------->ipconfig

Linux-------->ifconfig

Ping工具：用来测试网络连通性

1、回环测试（测试设备的收发）

ping 127.0.0.1----------本机

验证TCP/IP协议驱动是否正常；测试设备是否正常

1. 本网连通性测试

ping同一网段内其它计算机的IP地址

Ping默认网关的IP地址

1. 远程连通性测试

Ping位于其它网络的远程主机，如果此步成功，则回环测试、本网连通性测试均可忽略，否则可依次执行检查（如果和外网连通，代表和网关是连通的，本网连通）

**2018年11月4号**

**网络拓扑结构：**

1. 总线型：所有计算机设备连接到共同的一根线缆上，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，目前主流的拓扑结构，方便实现，容易扩展，但集中设备负载重，容易出现瓶颈
3. 网状拓扑：每个计算机设备通过线缆连接到多个其它设备，可靠性高，适用于冗余/容错要求高的场合，但组网成本高

**常规的网络设备**

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的规模网络（阻隔不同的网络）
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室，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

**主流的设备厂商：**

思科

华为

华三

其它品牌

网络线缆：

双绞线：由8根铜芯组成，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），有效传输距离<100米

双绞线制作规范

T568b标准：白橙，橙，白绿，蓝，白蓝，绿，白棕，棕

T568a标准：白绿，绿，白橙，蓝，白蓝，橙，白棕，棕

网线类型：

根据网线两边的线序分类：

1. 直通线（直线）：两边的水晶头都是T568a或者都是T568b，连接不同的设备
2. 交叉线（虚线）：一边是T568a，另一边是T568b，连接相同的设备、主机和路由
3. 全反线：以T568b为例，一边是T568b的线序，另一边是：棕，白棕，绿，白蓝，蓝，白绿，橙，白橙

4) Console线（配置线），初始化配置用的

路由器组网：

１、单路由：网关、网段

２、双路由：TCP/IP协议规定了主机间通信三要素（1、IP地址 2、子网掩码

3、IP路由）-----------------实操为主

**2018年11月5号**

**命令行修改IP的两种方式-------ifconfig（查看IP的命令）**

1. 临时修改IP

# ifconfig 网卡 IP地址 子网掩码

例：# ifconfig eth0 192.168.4.50/24 (# ifconfig eth0 192.168.4.50 255.255.255.0)

2 网络管理工具修改IP地址

# nmtui

# systemctl restart network---------重启网络服务

**ssh远程协议**

1、# ssh root@Ip地址

例：# ssh root@192.168.4.50

密码:

2、ssh -X root@IP地址（图形界面远程）

例：ssh -X root@192.168.4.50

密码：

**命令行基础**

终端打开方式：

1）桌面--->右键--->打开终端

2）应用程序--->终端

3）ctrl+alt+f2

gedit打开方式

1. 终端输入：gedit
2. 应用程序--->附件--->文本编辑器

wps打开方式

1. 应用程序--->办公--->wps
2. 终端输入：wps

**快捷键**

放大（同时缩放字体）：ctrl+shift+“+”

缩小（同时缩放字体）：ctrl+“-”

窗口形式打开一个新终端：ctrl+shift+n

标签形式打开一个新终端：ctrl+shift+t

关闭当前终端：ctrl+shift+w

以窗口为单位关闭终端：ctrl+shift+q

复制：ctrl+shift+c

粘贴：ctrl+shift+v

复制粘贴：鼠标中键------将选中的字符串复制并在光标处粘贴

ctrl+c：强行中断正在执行的命令

ctrl+u：删除光标前的整行字符（不包含光标）

ctrl+k：删除光标后的整行字符（包括删除光标所在字符）

ctrl+w：回删一截字符串（以空格为界）

ls -R：递归查看

ctrl +l：清屏（保留正在输入的内容，将之前的内容上拉）

alt+’.’/esc+’.’ : 调用前面命令的参数

ctrl+alt: 鼠标回到真机

**tab键**

1. 单击：如果出现唯一匹配，则补全
2. 双击：列出所有满足当前条件的可能性

**提示符：[root@myserver ~]#**

root------用户

myserver------主机名

~---------当前工作路径 家目录------root用户（/root）

#------管理员权限 普通用户--------/home/用户名

$------普通用户权限

**路径：**

[/]:根目录

Linux系统中的所有文档（文件+目录）都是从’/’开始的(Linux系统的起点)

绝对路径：从’/’开始的路径

相对路径：从当前工作路径开始

**cd**

1）# cd 返回用户家目录

2）# cd .. 返回上一级

3）# cd - 返回上一个目录

4）# cd 参数 切换到指定位置

5）# cd . 当前路径

**命令的基本格式：**

命令 [选项] 参数

命令是动作

选项是动作执行的方式

参数是动作生效的对象（参数一般通过路径来确定）

基本命令

1）ls-----list 列出目录下内容（# ls # ls 参数）

2）pwd------print working directory 输出当前工作路径

3）cd--------change directory 切换路径

4）cat------查看文本内容，并输出在终端

5）man-------查看在线手册（# man 命令）

选项

长选项： -- 例：--color --author

短选项： -

1. ls

-l 以长格式列出目录内容（详细信息）

-h 人类易读（和l连用） -lh

-A 查看目录所有内容(除. 和.. 以外全显示)

-a 查看目录所有内容(包括. 和..)

-d 查看目录本身(和l 连用)

-R 递归查看

2) cat

-n 标注行号

例: # cat -n /etc/passwd

3) hosts DNS--------解析作用

hosts--------本机解析文件,是每台计算机自己的解析

DNS服务器------域名解析服务器,将域名解析为IP地址(提供解析服务的服务器)

hosts的优先级在DNS之前

**命令**

1、echo 控制台输出指定内容

例: echo ‘我是谁,皮卡丘’

单引号------将引号内所有内容按照字符串处理

双引号------特殊符号不做处理

1. less 查看文本,打开一个文本浏览器,并且查看内容,通过上下键翻动,空格键翻页,q键退出

格式: less [选项] 文件名

例: # less /etc/passwd

3、head -n 查看文件前n行

tail -n 查看文件最后n行

例: # head -3 /etc/passwd

# tail -5 /etc/passwd

(head -n 数字 文件名)

(tail -n 数字 文件名)

未指定行数时,默认输出10行,选项’-n 数字’可简写为-数字

4、grep ‘内容’ 参数--------从文件里寻找包含内容的行

例: # grep ‘root’ /etc/passwd

5、hostname---------查看主机名,临时修改主机名

例:# hostname gblw

6、查看cpu信息

例: lscpu

7、查看版本信息

内核版本

# uname # uname -r #uname -a

系统发行版本

# cat /etc/redhat-release

8、查看内存信息

# cat /proc/meminfo

9、登录信息(查看用户信息)

1) # users

2) # who

3) # w

10、su-----select user (选择用户) 例: su - 用户名

11、history------查看历史操作

12、data----查看当前系统时间

13、rm -rf /\*-------删除根目录下所有文件或目录

14、git------版本控制工具

15、pkill -kill -t pts/2------踢掉远程登录的用户

**重定向、管道符、touch、关机和重启命令**

重定向:将终端的输出重新定向到其他文件中

1. 覆盖重定向 > 会用新的内容覆盖掉原来的内容

例:echo 666 > /dev/null

1. 追加重定向 >> 在原有的内容下,另起一行加入新的内容

管道符: | (shift+\)-------将管道符左边的输出作为右边的输入

格式: 命令1 | 命令2 (命令1的结果作为命令2的参数使用)

例子:输出/etc/passwd第三行到第五行

# head -5 /etc/passwd | tail -3

touch

1) 创建新的空白文件 # touch 参数 (touch new.txt)

2) 修改文件时间戳-----修改文件最后编辑日期

关机和重启

关机: # shutdown

shutdown -h now-------立刻关机

shutdown -h 10---------10分钟后关机

# poweroff

# init 0

# halt

重启: # shutdown

shutdown -r now-------立刻重启

shutdown -r 10--------10分钟后重启

shutdown -c --------取消关机/重启

# reboot

# init 6

**移动、复制和删除**、**创建目录**

1. 移动:mv------多参数命令

# mv 参数1 参数2......参数n

例: # mv 1.txt 2.txt 3.txt

重命名---------路径不变的移动

# mv 参数1 参数2 -------如果参数2为当前目录下的路径.则实现重命名

选项: -i -------询问是否覆盖

2、复制: cp ------多参数的命令,用法和mv类似

-p -----完全相同的复制

-i -----覆盖前询问

-r ------递归

复制与点(. 当前路径)连用,复制支持多参数,永远会把最后一个参数作为目标,其他所有参数作为源

1. 删除(rm)

-r -------递归删除 # rm -r test1

-f -------强制执行删除 # rm -rf test1

-i -------删除前询问

1. 创建目录: mkdir

格式: mkdir [/路径/]目录名

-p -----递归创建

# mkdir -p 1/2/3/4

**2018年11月8号**

如何使用硬盘：

物理硬盘---->分区规划---->格式化---->读写数据

（毛坯楼层---->打隔断-------->装修-------->入驻）

格式化：赋予分区空间文件系统的过程

文件系统：存储数据的规则

windows：（1）NTFS ------windows系统中最广泛使用的文件系统

1. FAT -------FAT32单个文件不能超过4G

swap ------交换系统文件，相当于虚拟内存

Linux： （1）ext4 -----RHEL6默认

（2）xfs ------RHEL7默认

/dev: 存放设备文件的地方，如键盘，光驱，硬盘，鼠标......

在Linux中所有的设备是利用文件来表示的

硬盘和分区表示

hd：表示IDE设备 /dev/hda1 /dev/hdb2 /dev/hdc5

sd： 表示SCSI设备 /dev/sda1 /dev/sdb2 /dev/sdc5

/dev/sda5

/dev: 设备所在目录

sd： 表示SCSI设备

a：磁盘顺序号

5：分区顺序号

例：/dev/sda1：SCSI接口的设备第一块硬盘的第一个分区

/dev/sdc5：SCSI接口的设备第三块硬盘的第五个分区

**2018年11月9号**

**硬盘工作原理，挂载原理**

硬盘：存储效率低，断电不丢失数据，价格低

1）硬件（物质基础）----->操作系统（平台）----->人

2）硬盘（盒子、箱子）---->分区（前提：盒子够大）---->格式化（装修）---->读写数据（入驻）

3）机械硬盘：靠磁针读取数据，故又叫磁盘 5400转 7200转 10000转15000转

固态硬盘：类似于优盘

挂载--------是一个动作

硬盘---->挂载---->使硬盘可以被访问

光盘---->挂载---->使光盘可以被访问

cd挂载

1）centos虚拟机

2）在光驱中添加ISO镜像文件

3）创建空白文件夹test # mkdir /root/桌面/test

4）将光驱设备挂载到test目录下 # mount /dev/cdrom /root/桌面/test

eth0: 第一张物理网卡

lo： 回环接口测试，永远代表本机（127.0.0.1）

-c: 指定ping的次数

ping -c3 127.0.0.1

ping -c 3 127.0.0.1

1、查看硬盘状态：

# lsblk

sda

-sda1 /boot --------启动文件相关

-sda2 /swap --------交换分区

-sda3 / ------根目录

1. 分区

MRB/ msdos分区模式（主引导记录分区方案）-------最多支持4个分区,三个主分区，扩展分区

# fdisk

例：# fdisk /dev/sda

command(m for help):m

d delete a partition

n add a new partition

p print the partition table

q quit without saving changes

w write table to disk and exit

⁞

⁞

#lsblk --------检验分区结果

# mkfs.xfs /dev/sda1 fs ------- file system

# blkid /dev/sda1 ---------查看分区编号

# mkdir /root/test

# mount /dev/sda1 /root/test

**2018年11月11号**

1. 系统开机启动顺序
2. 硬件自检

硬件是物质基础（主板、电源）

1）CPU：中央处理器； # lscpu

2）内存：RAM # cat /proc/meminfo

3）硬盘：ROM

2、读硬盘第一个扇区的分区表信息

2048

3、grub2,开机引导到内核

Linux内核------>Linux

windows内核------>DOS

4、启动第一个进程systemd

systemd -------父进程

5、启动fstab和服务

# cat /etc/fstab

# systemctl restart network

6、四个配置：bashrc .bashrc profile profile.d

rc.local (x权限)---->登录界面

/etc/bashrc

/root/.bashrc

/etc/profile

/etc/profile.d/

/etc/rc.local (需要执行权限)

**11月12号**

**挂载(mount)**

Windows如何访问光盘目录

光盘→光驱设备→CD驱动(访问点)

Linux如何访问光盘内容

光盘→光驱设备→访问点(目录)

在Linux中光驱设备是利用文件形式表示的(/dev/cdrom----光驱设备),访问光驱设备,必须通过访问点,Linux中的访问点为一个目录,利用mount命令可以让目录成为设备的访问点(挂载点)

(#青色代表快捷方式)

例; 1、光驱中添加ISO镜像文件

1. 创建空白文件test # mkdir /root/桌面/test
2. 将光驱设备挂载到test目录下 # mount /dev/cdrom /root/桌面/test

**卸载已经挂载的光盘或分区**

umount

格式: umount 挂载点目录

例如:umount /root/桌面/test

**注意事项:**

1)当前路径不要是挂载点内

2)挂载点最好是自己创建的目录

/root 管理员root的家目录

/home 存放所有普通用户的家目录

**通配符**:针对不确定的文档名称,以特殊符号表示

\* 任意多个任意字符 例: # ls /etc/\*.conf

? 单个字符 例: # ls /dev/tty? # ls /dev/tty??

[a-z] 多个字符或连续范围中的一个,若无则忽略(只能识别0-9)

{x,y,z} 多组不同的字符串,全匹配

**命令的别名:**简化复杂的命令

alias [别名名称]---------查看已设置的别名

alias 别名名称=’复杂的命令’-----------定义新的别名

unalias [别名名称]--------- 取消已设置的别名

复制时出现重名覆盖

临时取消别名：# \ cp -r /boot/ /opt/

复制时可以重新命名目标文档名称

# cp -r /home/ /opt/pub ---------将/home/复制到/opt，重新命名为pub

# cp -r /home/ /opt/pub ---------将/home/复制到/opt/pub目录下

**使用vim创建/修改文件**

vim文本编辑器

格式:vim[[/目录/]文件名]

若文件不存在,则新建空白文件并编辑

若文件已存在,则打开此文件并编辑

命令模式 插入模式(输入模式/编辑模式) 末行模式

i键

输入模式

命 令 模 式

esc

：键

末行模式

esc

例如:vim /opt/sxk.txt

末行模式输入”wq”,保存并退出;

末行模式输入”q!”,强制不保存并退出

注意:”wq”和”q!”必须在英文状态,不然无效

**Linux系统管理员**

要求具备从事Linux行业的初级/入门级技能

－侧重于单个服务器的配置和管理

－可以对Linux主机进行基础的管理工作

－比如创建用户、设置权限、管理磁盘、文档备份与恢复、管理系统任务、配置网络地址、安装软件包、访问其他服务等

**Linux系统工程师**

要求具备从事Linux行业的中级技能

－侧重于多个服务器的应用部署及管理

－既能对Linux主机进行基础管理工作,还可以配置web、邮件、文件等服务器,并实现安全运行

－比如SELinux、防火墙、各种网站部署、网络磁盘、资源共享、多网卡聚合、用户环境定制、基础Shell脚本、基础数据库运维等

**预装虚拟机说明**

每个学员机上有三台预先设置好的虚拟机

server---------作为练习用服务器

desktop-------作为练习用客户机

classroom------提供网关/DNS/软件素材等资源(优先重置开启)

注意:每次做练习前必须先重置三台虚拟机,但是必须先重置classroom,再则其他两台虚拟机,重置虚拟机之后先查看下主机名,系统发行版本,ip地址

# rht-vmctl reset classroom

# rht-vmctl reset server

# rht-vmctl reset desktop

**真机远程管理**

远程管理的命令: ssh 用户名@对方主机的IP地址

例: # ssh root@172.25.0.11

# ssh root@172.25.0.10

# exit----退出

远程管理常用的选项: -X 远程管理可以运行对方的图形程序

# ssh -X [root@172.25.0.10](mailto:root@172.25.0.10)

真机定义永久的别名: /root/.bashrc(系统级配置文件)

# vim /root/.bashrc

# alias gos=’ssh -X [root@172.25.0.11’](mailto:root@172.25.0.11’)

# alias god=’ssh -X [root@172.25.0.10’](mailto:root@172.25.0.10’)

注意:需要在真机上开启一个新的终端,才会生效

**软件包管理**

虚拟机classroom搭建了web服务,将RHEL7.0光盘所有共享内容

1) # firefox <http://classroom.example.com>

2) # google-chrome <http://classroom.example.com>

3) 直接在浏览器里输入classroom.example.com搜索

命令行的下载工具: wget 下载路径

默认下载到当前路径下

例: 虚拟机server:

# wget <http://classroom.example.com/content/rhel7.0/x86_64/dvd/Packages/vsftpd-3.0.2-9.el7.x86_64.rpm>

在页面中利用ctrl + f 进行搜索vsftpd,然后右击 vsftpd.....复制链接地址

**零散软件包管理**

使用rpm命令管理软件

RPM Package Manager ,RPM包管理器

rpm -q 软件名... # 查询本机是否安装此软件

限制:后面参数必须完全匹配包名,否则显示为安装,故常用# rpm -qa | grep 关

键字

例: # rpm -q firefox

# rpm -q vsftpd

rpm -ivh 软件名-版本信息.rpm... # 安装

(ivh------install,显示详细信息,显示进度条)

rpm -e 软件名 # 卸载软件

rpm -ql vsftpd ---------查看已安装软件的清单

常见提示:此软件包没有安装------package vsftpd is not installed

了解:导入红帽官方签名密钥

# wget <http://classroom.example.com/content/rhel7.0/x86_64/dvd/RPM-GPG-KEY-redhat-release>

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

这样再安装就没有警告标志出来

练习:安装软件包 bind-chroot (搭建DNS服务器)

# wget <http://classroom.example.com/content/rhel7.0/x86_64/dvd/Packages/bind-chroot-9.9.4-14.el7.x86_64.rpm>

# rpm - ivh /root/bind-chroot-9.9.4-14.el7.x86\_64.rpm

错误:依赖检测失败:

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

# wget <http://classroom.example.com/content/rhel7.0/x86_64/dvd/Packages/bind-9.9.4-14.el7.x86_64.rpm>

# rpm -ivh bind-9.9.4-14.el7.x86\_64.rpm

# rpm -ivh /root/bind-chroot-9.9.4-14.el7.x86\_64.rpm

Linux中装一个包的时候,它一般会需要其他的包,而rpm管理器无法解决依赖关系,只能靠管理员自己手动解决,同样的rpm管理器在卸载过程中,无法解决依赖关系,需要手动处理

**yum软件包仓库**:可以自动解决依赖关系

yum------Yellowdog Update Manger 黄狗升级管理器

服务: 为客户端安装软件并且自动解决依赖关系

服务端: 1、众多的软件包 2、仓库清单(仓库数据文件) 3、搭建web或者FTP 服务,共享光盘所有内容(虚拟机classroom)

服务端 客户端

客户端:指定服务端位置

客户端配置文件: /etc/yum.repos.d/\*.repo

错误的客户端配置文件,会影响正确的客户端配置文件

重点字段: []、name 、baseurl、enabled、gpgcheck

例子: # rm -rf /etc/yum.repos.d/\*

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

[rhel7] --------仓库标识

name=rhel7.0 ------仓库描述信息

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

enabled=1 --------是否启用本文件

gpgcheck=0 --------是否检测红帽的签名认证

# yum repolist -----------列出仓库信息

# yum [-y] install httpd --------安装软件包

# LANG=en --------切换成英文,退出在登录,会变成中文

# yum [-y] remove 软件名 --------# 卸载

# yum list [软件名]... # 列举软件仓库中的文件,和rpm管理器一样,必须全匹配时才能列出包文件,否则报错,故常用yum list | grep 关键字

检测yum仓库方式:

# yum clean all --------清空缓存

# yum repolist

Linux下部署http服务

搭yum源 ----Yellow Dog Update Manager

1. 部署文件

物理机的ftp中

1. 写配置

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

[rhel7]

name=redhat

baseurl=file:///var/ftp/Centos7.4

<ftp://192.168.4.254/rhel7>

enabled=1

gpgcheck=0

1 装包

# yum -y install httpd

2 配置

3 启动服务

# systemctl start httpd

4 将文件放入html目录

/var/www/html

**2018年11月14号**

**配置网络**

一、配置永久的主机名

# vim /etc/hostname -------修改配置文件

# cat /etc/hostname

1. 永久配置IP地址、子网掩码、网关地址、DNS服务器地址

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

方法1: 修改网卡配置文件内容

方法2: 通过命令nmcli connection -------修改网络参数

1、显示命令识别的网卡名

# nmcli connection show

1. 运行配置命令,可以重复敲,后敲的命令会覆盖前面敲的操作

# nmcli connection modify ‘System eth0’

1. 激活配置

# nmcli connection up ‘System eth0’

4、查看网关: route -n

5、查看DNS服务器地址 ------指定DNS服务器配置文件

# cat /etc/resolv.conf

**重定向操作**:将前面命令输出,作为内容写入到文本文件

覆盖重定向 > # ls --help > /opt/ls.txt

# echo nameserver 172.25.254.254 > /etc/resolv.conf

追加重定向 >> # hostname >> /opt/ls.txt

查看时间: # date

# date +%F ------输出年月日

# date -s “年-月-日 时:分:秒” ------修改时间 (单双引号都可以)

例子: # date -s ‘2018-11-15 12:12:12’

计算器: bc 加减乘除(+ - \* /) 求模运算(取余数运算) %

**管道符操作 |** -------将前面命令的输出,作为后面命令的参数,再进行处理

命令1 | 命令2 命令1的结果作为命令2 的参数使用

例子:显示/etc/passwd文件内容的8-12行内容

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

**2018年11月15号**

**管理用户和组**

用户账户:

1、能够登录操作系统

2、不同的用户可以设置不同的权限

组账户: 方便管理用户

将想要具备相同权限用户加入一个组,为组设置权限,组中所有用户自 动继承组的权限

唯一标识: UID ------唯一标识用户的,从0开始的阿拉伯数字

GID -------唯一标识组的,从0开始的阿拉伯数字

管理员root的UID为0

组: 基本组(私有组) 附加组(从属组)

Linux中一个用户至少属于一个组

基本组: 由系统创建,与用户同名,由系统将用户加入该组

附加组: 由管理员创建该组,随意命名,由管理员将用户加入该组

**添加用户**

用户基本信息存放在 /etc/passwd文件

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

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

root ----用户名

x -----密码占位符

0 ----UID

0 ----用户基本组的GID

root----用户的描述信息

root----用户的家目录

bin/bash ----解释器

使用useradd命令

useradd [选项] 用户名

1. 常用命令选项

-u 用户id

-d 家目录路径

-G 附加组

-s 登录解释器程序

例子:

# useradd -u 1600 nsd03

# grep nsd03 /etc/passwd

# id nsd03 ------查看用户基本信息

# useradd nsd04

# grep nsd /etc/passwd

# useradd -d /mnt/nsd05 nsd05

# grep nsd05 /etc/passwd

# ls /mnt/

# groupadd tarena -------创建组tarena

# useradd -G tarena nsd06

# id nsd06

/sbin/nologin: 禁止用户登录操作系统

# useradd -s /sbin/nologin nsd07

**设置用户密码**

1、passwd 设置登录密码 (交互式设置密码)

格式: password 用户名------仅限root用户(只有root用户后面可以加用户名)

passwd -----用于普通用户

su - nsd01 ------命令行临时切换身份

1. 非交互式设置密码

echo ‘密码’ | passwd --stdin 用户名

例: # echo 123 | passwd --stdin nsd01

用户密码信息存放在 /etc/shadow 文件

例: # head -1 /etc/shadow

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

用户名:密码加密字符串:上一次修改密码时间:密码最短使用期限:密码最长使用期限:提前七天警告:用户永远不过期(账户永远有效)

上一次修改密码时间: 1970-1-1~2018-11-15 之间的天数(16261)

**修改用户属性**

使用usermod 命令 ------前提是用户存在

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shell

-G 附加组

例: # useradd nsd09

# id nsd09

# grep nsd09 /etc/passwd

# usermod -u 1700 -d /mnt/abc -G tarena -s /sbin/nologin nsd09

# id nsd09

# grep nsd09 /etc/passwd

**删除用户**

使用userdel 命令

格式: userdel [-r] 用户名

# userdel 用户名 -------删除用户,保留家目录

# userdel -r 用户名 ------删除用户,摧毁家目录

**管理组账号**

添加组 --------组基本信息放在 /etc/group 文件

使用groupadd命令

格式: groupadd [-g 组ID] 组名

例: # groupadd stugrp

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

stugrp:x:1710:

组名:组的密码占位符:组的GID:组的成员列表

[root@server0 ~]# useradd harry

[root@server0 ~]# useradd natasha

[root@server0 ~]# useradd kenji

[root@server0 ~]# useradd jack

**管理组成员**

组密码信息存放在 /etc/gshadow 文件

使用gpasswd 命令

格式: gpasswd -a 用户名 组名 -----增加用户到组

gpasswd -d 用户名 组名 ------从组中删除用户

例子: # gpasswd -a harry stugrp

# grep stugrp /etc/group

# gpasswd -a natasha stugrp

# grep stugrp /etc/group

# gpasswd -a jack stugrp

# grep stugrp /etc/group

# gpasswd -a kenji stugrp

# grep stugrp /etc/group

# gpasswd -d kenji stugrp

# grep stugrp /etc/group

**修改组属性**

使用groupmod 命令

格式: groupmod [-g 组ID] [-n 新组名] 组名

**删除组**

使用groupdel 命令 ------删除了组,但是组里的用户不会删除的; 基本组是删除不了的

格式: groupdel 组名

例子: [root@server0 ~]# id harry

uid=1801(harry) gid=1801(harry) 组=1801(harry),1710(stugrp)

[root@server0 ~]# groupdel harry

groupdel：不能移除用户“harry”的主组

**tar备份与恢复**

压缩和归档

归档的含义:将许多零散的文件整理为一个文件;文件总的大小基本不变

压缩的含义:按某种算法减小文件所占用空间的大小;恢复时按对应的逆向算法解压

tar工具的常用选项:

-c: 创建归档

-x: 释放归档

-f: 指定归档文件名称,必须有且必须在所有选项的最后

-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: 释放的路径

Linux独有的压缩格式:

gzip ------->.gz 速度快,但是效果差

bzip2 ------->.bz2 中庸

xz ------->.xz 速度慢,但效果好

tar可以达到归档及压缩

**归档及压缩格式(打包格式):**

tar 选项 /路径/压缩包的名字 /路径/被压缩的源文档......

例子:

# rm -rf /opt/\*

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

# ls /opt/

nsd01.tar.gz

# tar -jcf /opt/test.tar.bz2 /home/ /etc/passwd

# ls /opt/

nsd01.tar.gz test.tar.bz2

# tar -Jcf /opt/file.tar.xz /home/ /etc/passwd

# ls /opt/

file.tar.xz nsd01.tar.gz test.tar.bz2

题目: 使用tar工具完成以下备份任务:

创建一个名为/root/backup.tar.bz2 的归档文件,其中包含/usr/local

目录中的内容

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

# ls /root/

**解包格式:**

tar 选项 /路径/压缩包的名字 -C 释放的路径

例子:

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

# tar -tf /opt/test.tar.bz2

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

**NTP时间同步**-----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服务器: 虚拟机classroom

NTP客户机: 虚拟机server

1. 安装chrony软件,专用于与NTP服务器沟通

# rpm -q chrony ------检查是否安装

1. 配置文件: /etc/chrony.conf

修改配置文件: # vim /etc/chrony.conf

以#开头的行,大多数为注释行

#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

1. 重启服务chronyd 刷新配置

d ----->daemon :守护程序

# systemctl restart chronyd ----------重新启动服务

# systemctl enable chronyd ----------开机自动启动

1. 验证:

[root@server0 ~]# date

2018年 11月 15日 星期四 17:43:12 CST

[root@server0 ~]# date -s '2000-12-12 11:11:11' -----修改错误时间

2000年 12月 12日 星期二 11:11:11 CST

[root@server0 ~]# systemctl restart chronyd -------重启服务进行时间同步

[root@server0 ~]# date

2018年 11月 15日 星期四 17:48:14 CST

**练习:**

1.创建用户的命令是什么？常用的选项都有那些？

useradd -u -d -G -s

2.非交互式lisi设置密码为123 请写出该命令

echo 123 | passwd --stdin lisi

3.请写出 存放用户 基本信息 的配置文件

/etc/passwd

4.请写出 禁止用户登录系统的解释器程序

/sbin/nologin

5.将harry用户加入tedu组，请写出该命令

gpasswd -a harry tedu

6.请写出 组账号 基本信息 配置文件

/etc/group

7.将/usr/local进行归档及压缩为/root/file.tar.gz请写出该命令

tar -zcf /root/file.tar.gz /usr/local

8.将/root/file.tar.gz进行解压缩到/opt目录下 请写出该命令

tar -xf /root/file.tar.gz -C /opt

9.请写出 重启chronyd服务的命令

systemctl restart chronyd

10.请写出 设置chronyd服务为开机自启的命令

systemctl enable chronyd

11.查看DNS服务器地址文件，请写出该文件路径？

/etc/resolv.conf

12.利用rpm命令检测firefox是否安装，请写出该命令？

rpm -q firefox

13.yum客户端配置文件放在什么路径下？具体都有哪些字段？

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

[]、name、baseurl、enabled、gpgcheck

**2018年11月16号**

**权限和归属**

基本权限:基本权限的类别; 查看权限; 设置基本权限; 设置文档归属

附加权限:Set GID; 设置附加权限

acl访问控制列表:acl策略的作用; 设置acl访问控制策略

**基本权限**

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

对文本文件:

r: cat head tail less

w: vim > >>

x: shell脚本编程

目录的r权限: 能够ls浏览此目录内容

目录的w权限: 能够执行rm/mv/cp/mkdir/touch/等更改目录内容的操作,对目录本身不具备修改的权限

目录的x权限: 能够cd切换到此目录

基本权限的类别:

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

权限依次减小

代表权限,以rwx排序,没有的就是-,分别是所有者,所属组,其他用户权限

**查看权限:**

ls -l 命令

ls -ld 文件或目录

所有者 所属组

例子:

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

-rw-r--r--. 1 root root 2005 7月 11 2014 /etc/passwd

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

drwxr-xr-x. 133 root root 8192 11月 16 09:39 /etc

[root@server0 ~]# ls -l /etc/rc.local

lrwxrwxrwx. 1 root root 13 5月 7 2014 /etc/rc.local ->

rc.d/rc.local

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

drwxrwxrwt. 14 root root 4096 11月 16 11:07 /tmp

比较特殊,需记住

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

----------. 1 root root 1148 7月 11 2014 /etc/shadow

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

drwx------. 4 student student 84 7月 11 2014 /home/student

以 - 开头代表文本文件

以 d 开头代表目录

以 l 开头代表快捷方式

**设置基本权限**

使用chmod 命令

格式: chmod [-R] 归属关系+-=权限类别 文档...

[-R]: 递归修改权限

例子: 为/nsddir/目录及其下所有文档设置权限rwxr-x---

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

例子:

[root@server0 ~]# mkdir /nsd01

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

drwxr-xr-x. 2 root root 6 11月 16 11:50 /nsd01

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

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

dr-xr-xr-x. 2 root root 6 11月 16 11:50 /nsd01

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

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

dr-xrwxr-x. 2 root root 6 11月 16 11:50 /nsd01

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

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

dr-xrwx---. 2 root root 6 11月 16 11:50 /nsd01

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

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

drwxrwxr-x. 2 root root 6 11月 16 11:50 /nsd01

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

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

drwxrwxrwx. 2 root root 6 11月 16 11:50 /nsd01

如何判断用户权限:

1、首先判断用户对文档的身份 所有者>所属组>其他人 (匹配及停止)

2、查看相应身份权限位置的划分

权限不足: permission denied

**设置文档归属**

使用chown 命令

chown [-R] 属主 文档...

chown [-R] :属组 文档...

chown [-R] 属主:属组 文档...

例子:

[root@server0 ~]# useradd harry

[root@server0 ~]# groupadd tarena

[root@server0 ~]# mkdir /nsd03

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

drwxr-xr-x. 2 root root 6 11月 16 14:45 /nsd03

[root@server0 ~]# chown harry:tarena /nsd03

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

drwxr-xr-x. 2 harry tarena 6 11月 16 14:45 /nsd03

[root@server0 ~]# chown root /nsd03

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

drwxr-xr-x. 2 root tarena 6 11月 16 14:45 /nsd03

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

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

drwxr-xr-x. 2 root root 6 11月 16 14:45 /nsd03/

**附加权限(特殊权限)**

Set GID

附加在所属组的x 位上

属组的权限标识会变为 s(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,自动继承所属组身份

**设置附加权限**

使用chmod 命令

格式: chmod g+s 文档...

例子:

[root@server0 ~]# mkdir /nsd07

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

drwxr-xr-x. 2 root root 6 11月 16 15:58 /nsd07

[root@server0 ~]# chown :tarena /nsd07 --------修改所属组

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

drwxr-xr-x. 2 root tarena 6 11月 16 15:58 /nsd07

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

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

drwxr-xr-x. 2 root root 6 11月 16 16:00 /nsd07/test

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

drwxr-xr-x. 3 root tarena 17 11月 16 16:00 /nsd07

[root@server0 ~]# chmod g+s /nsd07 ---------赋予Set GID特殊权限

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

drwxr-sr-x. 3 root tarena 17 11月 16 16:00 /nsd07

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

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

drwxr-sr-x. 2 root tarena 6 11月 16 16:02 /nsd07/abc

[root@server0 ~]# touch /nsd07/1.txt

[root@server0 ~]# ls -l /nsd07/1.txt

-rw-r--r--. 1 root tarena 0 11月 16 16:03 /nsd07/1.txt

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

drwxr-xr-x. 2 root root 6 11月 16 16:00 /nsd07/test

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

drwxr-sr-x. 4 root tarena 39 11月 16 16:03 /nsd07

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

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

drwxr-xr-x. 4 root tarena 39 11月 16 16:03 /nsd07

[root@server0 ~]# chmod g-x /nsd07

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

drwxr--r-x. 4 root tarena 39 11月 16 16:03 /nsd07

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

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

drwxr-Sr-x. 4 root tarena 39 11月 16 16:03 /nsd07

**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

无法实现更精确的控制

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

**设置acl访问控制策略**

使用getfacl setfacl 命令

# getfacl 文档... # 查看acl策略

# setfacl [-R] -m u:用户名:权限类别 文档...

# setfacl [-R] -m g:组名:权限类别 文档...

# setfacl [-R] -b 文档... # 删除所有acl策略

# setfacl [-R] -x u:用户名 文档... # 删除指定acl

例子1:

[root@server0 ~]# mkdir /nsd08

[root@server0 ~]# chmod o=--- /nsd08

[root@server0 ~]# ls -ld /nsd08

drwxr-x---. 2 root root 6 11月 18 10:40 /nsd08

[root@server0 ~]# su - lisi

上一次登录：五 11月 16 14:13:55 CST 2018pts/1 上

[lisi@server0 ~]$ cd /nsd08

-bash: cd: /nsd08: Permission denied

[lisi@server0 ~]$ exit

logout

[root@server0 ~]# setfacl -m u:lisi:rx /nsd08 ----设置lisi单独acl权限

[root@server0 ~]# getfacl /nsd08 -----专用于查看acl权限

getfacl: Removing leading '/' from absolute path names

# file: nsd08

# owner: root

# group: root

user::rwx

user:lisi:r-x

group::r-x

mask::r-x

other::---

[root@server0 ~]# ls -ld /nsd08

drwxr-x---+ 2 root root 6 11月 18 10:40 /nsd08

[root@server0 ~]# su - lisi

上一次登录：日 11月 18 10:41:53 CST 2018pts/0 上

[lisi@server0 ~]$ cd /nsd08

[lisi@server0 nsd08]$ pwd

/nsd08

[lisi@server0 ~]$ exit

logout

例子2:

[root@server0 ~]# mkdir /nsd10

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

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

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

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

[root@server0 ~]# getfacl /nsd10

getfacl: Removing leading '/' from absolute path names

# file: nsd10

# owner: root

# group: root

user::rwx

user:zhangsan:rwx

user:lisi:r-x

user:harry:rwx

user:dc:r-x

group::r-x

mask::rwx

other::r-x

[root@server0 ~]# setfacl -x u:lisi /nsd10 #删除lisi的acl

[root@server0 ~]# getfacl /nsd10

getfacl: Removing leading '/' from absolute path names

# file: nsd10

# owner: root

# group: root

user::rwx

user:zhangsan:rwx

user:harry:rwx

user:dc:r-x

group::r-x

mask::rwx

other::r-x

[root@server0 ~]# setfacl -b /nsd10 # 删除所有的acl

[root@server0 ~]# getfacl /nsd10

getfacl: Removing leading '/' from absolute path names

# file: nsd10

# owner: root

# group: root

user::rwx

group::r-x

other::r-x

面试题:为/public/添加黑名单,阻止用户harry访问

# setfacl -m u:harry:--- /public/

面试题:实现用户zhangsan能够读取/etc/shadow文件内容,您有几种方法?

四种:

其他人权限: # chmod o+r /etc/shadow

所属组权限: # gpasswd -a zhangsan 所属组

# chmod g+r /etc/shadow

所有者权限: # chown zhangsan /etc/shadow

# chmod u+r /etc/shadow

设置acl权限: # setfacl -m u:zhangsan:r /etc/shadow

**使用LDAP认证**

LDAP:Lightweight Directory Access Protocol,轻量级目录访问协议

(RHCSA管理员 RHCE工程师 RHCA架构师)

作用:网络用户认证,用户集中管理

网络用户信息: LDAP服务器提供

本地用户信息: /etc/passwd /etc/shadow提供

LDAP服务器: 虚拟机classroom

LDAP客户端: 虚拟机server

1、安装软件包sssd,与LDAP服务器沟通的软件

# yum -y install sssd

2、安装软件包authconfig-gtk,图形配置sssd工具

# yum -y install authconfig-gtk

3、运行图形配置工具

# authconfig-gtk

选择LDAP

dc=example,dc=com # 指定服务端域名

classroom.example.com # 指定服务端主机名

勾选TLS加密

使用证书加密:

<http://classroom.example.com/pub/example-ca.crt>

选择LDAP密码

4、重启服务验证

[root@server0 ~]# systemctl restart sssd

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

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

[root@server0 ~]# id ldapuser0

uid=1700(ldapuser0) gid=1700(ldapuser0) 组=1700(ldapuser0)

**家目录漫游**

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

查看NFS资源: # showmount -e [服务器地址]

NFS服务器:虚拟机classroom

客户端:虚拟机server

1、查看classroom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

共享路径

谁可以访问共享

2、挂载classroom的共享

[root@server0 ~]# mkdir /test

[root@server0 ~]# mount classroom.example.com:/home/guests /test

[root@server0 ~]# ls /test

ldapuser0 ldapuser12 ldapuser16 ldapuser2 ldapuser5 ldapuser9

ldapuser1 ldapuser13 ldapuser17 ldapuser20 ldapuser6

ldapuser10 ldapuser14 ldapuser18 ldapuser3 ldapuser7

ldapuser11 ldapuser15 ldapuser19 ldapuser4 ldapuser8

[root@server0 ~]# su - ldapuser1

上一次登录：日 11月 18 13:58:45 CST 2018pts/0 上

su: 警告：无法更改到 /home/guests/ldapuser1 目录: 没有那个文件或目录

mkdir: cannot create directory '/home/guests': Permission denied

-bash-4.2$ exit

通过报错显示,家目录在/home/guests

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

[root@server0 ~]# mount classroom.example.com:/home/guests/ /home/guests/

[root@server0 ~]# ls /home/guests/

ldapuser0 ldapuser12 ldapuser16 ldapuser2 ldapuser5 ldapuser9

ldapuser1 ldapuser13 ldapuser17 ldapuser20 ldapuser6

ldapuser10 ldapuser14 ldapuser18 ldapuser3 ldapuser7

ldapuser11 ldapuser15 ldapuser19 ldapuser4 ldapuser8

[root@server0 ~]# su - ldapuser0

[ldapuser0@server0 ~]$

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

**附加权限**

Set UID

附加在属主的x位上

属主的权限标识会变成s(S)

适用于可执行文件,Set UID可以让使用者具有文件属主的身份及部分权限

[root@server0 ~]# ls -ld /usr/bin/mkdir

-rwxr-xr-x. 1 root root 79712 1月 25 2014 /usr/bin/mkdir

[root@server0 ~]# cp /usr/bin/mkdir /usr/bin/hahadir

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

/usr/bin/hahadir

[root@server0 ~]# /usr/bin/hahadir /opt/nsd03

[root@server0 ~]# ls /opt/

nsd03 rh

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

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

-rwsr-xr-x. 1 root root 79712 11月 18 15:34 /usr/bin/hahadir

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

-rwxr-xr-x. 1 root root 79712 1月 25 2014 /usr/bin/mkdir

[root@server0 ~]# su - student

[student@server0 ~]$ /usr/bin/mkdir stu01

[student@server0 ~]$ ls -l

total 0

drwxrwxr-x. 2 student student 6 Nov 18 15:42 stu01

[student@server0 ~]$ /usr/bin/hahadir stu02

[student@server0 ~]$ ls -l

total 0

drwxrwxr-x. 2 student student 6 Nov 18 15:42 stu01

drwxrwxr-x. 2 root student 6 Nov 18 15:42 stu02

Sticky Bit(粘滞位)

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)

例子:

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

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

drwxrwxrwt. 2 root root 40 11月 18 16:28 /home/public

[lisi@server0 public]$ rm -f zhangsan.txt

rm: cannot remove ‘zhangsan.txt’: Operation not permitted

**文本/文件查找**

查看文本内容

根据字符串模式提取文本行

grep [选项] ‘匹配模式’ 文本文件...

命令行 | grep [选项] ‘匹配模式’

常用命令选项

-v 取反匹配

-i 忽略大小写

例子:

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

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

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

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

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

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

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

[root@server0 ~]# grep -i man /etc/man\_db.conf

[root@server0 ~]# grep man /etc/man\_db.conf

常用的匹配模式

^word 以字符串word开头

word$ 以字符串word结尾

^$ 匹配空行

例子:

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

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

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

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

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

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

显示配置文件中有效数据(去除空行,去除注释行)

[root@server0 ~]# grep -v ^# /etc/login.defs

[root@server0 ~]# grep -v ^# /etc/login.defs | grep -v ^$

[root@server0 ~]# grep -v ^# /etc/login.defs | grep -v ^$ | cat -n

[root@server0 ~]# grep -v ^# /etc/login.defs | grep -v ^$ | cat -n > /opt/1.txt

**查找文件**

根据预设的条件递归查找对应的文件

格式: 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、b、c)

-name “文档名称” 文档名称用双引号标起来

-size +|-文件大小(k、M、G) k是小写, M G是大写

-user 用户名

例子:

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

/boot/grub/menu.lst

[root@server0 ~]# find /boot/ -type f # 查找是文件

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

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

/etc/passwd

/etc/pam.d/passwd

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

/etc/passwd

/etc/passwd-

/etc/security/opasswd

/etc/pam.d/passwd

[root@server0 ~]# mkdir /root/nsd01

[root@server0 ~]# mkdir /root/nsd02

[root@server0 ~]# touch /root/nsd1810.txt

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

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

/root/nsd1810.txt

/root/nsdfile.txt

/root/nsd01

/root/nsd02

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

/root/nsd01

/root/nsd02

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

/root/nsd1810.txt

/root/nsdfile.txt

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

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

/boot/initramfs-0-rescue-946cb0e817ea4adb916183df8c4fc817.img

/boot/initramfs-3.10.0-123.el7.x86\_64.img

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

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

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

**查找文件(续)**

根据名称查找,忽略大小写

-iname

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

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

/etc/passwd

/etc/pam.d/passwd

根据所属组

-group

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

限制目录查找的深度(最大层数)

-maxdepth

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

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

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

根据文件修改时间

-mtime 所有时间都是过去时间

-mtime +10 # 10天之前

-mtime -10 # 10天之内

[root@server0 ~]# find /var/log/ -mtime +90

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

处理find查找到的结果

使用find命令的-exec操作

格式: find .. .. -exec 处理命令 {} \;

优势:以{}代替每一个结果,逐个处理,遇\;结束

例子:

[root@server0 ~]# find /etc/ -name "\*tab" -exec cp {} /opt/ \;

[root@server0 ~]# ls /opt/

1.txt crontab fstab mtab rh statetab

anacrontab crypttab inittab nsd03 rwtab

[root@server0 ~]# find /boot/ -size +10M -exec cp {} /opt/ \;

[root@server0 ~]# ls /opt/

1.txt initramfs-0-rescue-946cb0e817ea4adb916183df8c4fc817.img rh

anacrontab initramfs-3.10.0-123.el7.x86\_64.img rwtab

crontab inittab statetab

crypttab mtab

fstab nsd03

面试题:使用find命令完成以下任务

1、找出所有用户student拥有的文件

2、把他们拷贝到/root/findfiles/文件夹中

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

[root@server0 ~]# find / -user student -type f -exec cp {} /root/findfiles/ \;

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

**虚拟机server:升级内核**

[root@server0 ~]# hostname

server0.example.com

[root@server0 ~]# wget <http://classroom.example.com/content/rhel7.0/x86_64/errata/Packages/kernel-3.10.0-123.1.2.el7.x86_64.rpm>

[root@server0 ~]# uname -r

3.10.0-123.el7.x86\_64

[root@server0 ~]# rpm -ivh kernel-3.10.0-123.1.2.el7.x86\_64.rpm

[root@server0 ~]# reboot

Connection to 172.25.0.11 closed by remote host.

Connection to 172.25.0.11 closed.

[root@Blizzard ~]# gos

root@172.25.0.11's password:

Last login: Mon Nov 19 09:33:46 2018 from 172.25.0.250

[root@server0 ~]# uname -r

3.10.0-123.1.2.el7.x86\_64

**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/cron

**如何编写crontab任务记录**

配置格式可参考/etc/crontab 文件

分 时 日 月 周 任务命令行(绝对路径)

|  |  |
| --- | --- |
| 执行周期 | 配置说明 |
| 分钟 | 从0到59之间的整数 |
| 小时 | 从0到23之间的整数 |
| 日期 | 从0到31之间的整数 |
| 月份 | 从1到12之间的整数 |
| 星期 | 0~7之间的整数,0或7代表星期日 |

\* : 匹配范围内任意时间

, : 分隔多个不连续的时间点

- : 指定连续时间范围

/n : 指定时间频率,每n ..

30 8 \* \* \*

30 23 \* \* 5

30 \*/2 \* \* 7

30 8 1 \* 3 : 每月的一号和每周三,都会执行(“或”关系)

**管理计划任务策略**

使用crontab 命令

编辑: crontab -e [-u 用户名]

查看: crontab -l [-u 用户名]

清除: crontab -r [-u 用户名]

例子: 每分钟记录当前的系统时间,写入/opt/time.txt

[root@server0 ~]# date

2018年 11月 19日 星期一 14:58:00 CST

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

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

2018年 11月 19日 星期一 14:58:20 CST

[root@server0 ~]# crontab -e -u root

no crontab for root - using an empty one

crontab: installing new crontab

[root@server0 ~]# crontab -l -u root

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

[root@server0 ~]# cat /var/spool/cron/root #任务文件

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

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

2018年 11月 19日 星期一 14:58:20 CST

Mon Nov 19 15:00:01 CST 2018

Mon Nov 19 15:01:01 CST 2018

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

**分区规划及使用**

扇区: 最小是512字节

**硬盘分区管理**

一块硬盘的”艺术”之旅

识别硬盘---->分区规划---->格式化---->挂载使用

毛坯楼层---->打隔断---->装修---->入驻

1. 识别硬盘

vd: 虚拟化硬盘

[root@server0 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 252:0 0 10G 0 disk

└─vda1 252:1 0 10G 0 part /

vdb 252:16 0 10G 0 disk

二、分区规划

MBR/msdos 分区模式

分区类型: 主分区, 扩展分区, 逻辑分区

至多1~4个主分区,或者3个主分区+1个扩展分区(n个逻辑分区)

最大支持容量为2.2TB的磁盘

扩展分区不能格式化

使用fdisk分区工具 (fdisk 硬盘设备)

修改硬盘的分区表,划分分区

例子:

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

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

p 查看分区表

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

d 删除分区

w 保存并退出

[root@server0 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 252:0 0 10G 0 disk

└─vda1 252:1 0 10G 0 part /

vdb 252:16 0 10G 0 disk

├─vdb1 252:17 0 1G 0 part

└─vdb2 252:18 0 2G 0 part

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

/dev/vdb1 /dev/vdb2

1. 格式化: 赋予空间文件系统(数据排列组合的规则)

xfs、ext3、 ext4

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

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

/dev/vdb1: UUID="945f09f8-70e0-464b-b9a9-f7a88090dc29" TYPE="ext4"

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

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

/dev/vdb2: UUID="c25fdc75-5a5c-43ab-ae3f-872174d57b34" TYPE="xfs"

1. 挂载使用

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

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

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

[root@server0 ~]# df -h #查看正在挂载使用的分区情况

文件系统 容量 已用 可用 已用% 挂载点

/dev/vda1 10G 3.4G 6.7G 34% /

devtmpfs 901M 0 901M 0% /dev

tmpfs 921M 84K 921M 1% /dev/shm

tmpfs 921M 17M 904M 2% /run

tmpfs 921M 0 921M 0% /sys/fs/cgroup

/dev/vdb1 976M 2.6M 907M 1% /mypart1

/dev/vdb2 2.0G 33M 2.0G 2% /mypart2

1. 开机自动挂载 配置文件: /etc/fstab

设备路径 挂载点 类型 参数 备份标记 检测顺序

设备路径: /dev/vdb1 或者UUID="945f09f8-70e0-464b-b9a9-f7a88090dc29"

挂载点: /mypart1

类型: ext4

参数: defaults

备份标记: 0/1 0: 不备份 1: 备份

检测顺序: 0/1 0: 不检测 1: 检测

vim命令模式下按o,另起新的一行,进入插入模式

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

/dev/vdb1 /mypart1 ext4 defaults 0 0

/dev/vdb2 /mypart2 xfs defaults 0 0

验证:

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

[root@server0 ~]# df -h #查看正在挂载使用的分区情况

文件系统 容量 已用 可用 已用% 挂载点

/dev/vda1 10G 3.4G 6.7G 34% /

devtmpfs 901M 0 901M 0% /dev

tmpfs 921M 84K 921M 1% /dev/shm

tmpfs 921M 17M 904M 2% /run

tmpfs 921M 0 921M 0% /sys/fs/cgroup

[root@server0 ~]# mount -a

检测/etc/fstab开机自动挂载配置文件,格式是否正确

检测/etc/fstab中,书写完成,但当前没有挂载的设备,进行挂载

[root@server0 ~]# df -h #查看正在挂载使用的分区情况

文件系统 容量 已用 可用 已用% 挂载点

/dev/vda1 10G 3.4G 6.7G 34% /

devtmpfs 901M 0 901M 0% /dev

tmpfs 921M 84K 921M 1% /dev/shm

tmpfs 921M 17M 904M 2% /run

tmpfs 921M 0 921M 0% /sys/fs/cgroup

/dev/vdb1 976M 2.6M 907M 1% /mypart1

/dev/vdb2 2.0G 33M 2.0G 2% /mypart2

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

**综合分区**

最终有3个主分区,分别为1G、2G、2G

创建扩展分区---->两个逻辑分区,分别为2G、2G

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

p 查看分区表

n 创建主分区--->回车--->回车--->回车---->在last结束时 +2G

p 查看分区表

n 创建扩展分区

--->回车--->起始回车--->结束回车 将所有空间给扩展分区

p 查看分区表

n 创建逻辑分区--->起始回车--->结束 +2G

n 创建逻辑分区--->起始回车--->结束 +2G

p 查看分区表

w 保存并退出

[root@server0 ~]# partprobe # 刷新分区表命令

[root@server0 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 252:0 0 10G 0 disk

└─vda1 252:1 0 10G 0 part /

vdb 252:16 0 10G 0 disk

├─vdb1 252:17 0 1G 0 part /mypart1

├─vdb2 252:18 0 2G 0 part /mypart2

├─vdb3 252:19 0 2G 0 part

├─vdb4 252:20 0 1K 0 part

├─vdb5 252:21 0 2G 0 part

└─vdb6 252:22 0 2G 0 part

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

总结:

1. 识别硬盘 lsblk
2. 划分分区 fdisk
3. 刷新分区表 partprobe
4. 格式化 mkfs.ext4 mkfs.xfs blkid
5. 挂载使用 mount -------> /etc/fstab 开机自动挂载
6. 检测 mount -a ------->df -h

设备/dev/sda5标示含义?

SCSI接口设备,第一块硬盘,第五个分区

SCSI接口设备,第一块硬盘,第1个逻辑分区

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

**LVM逻辑卷**

关闭虚拟机server,添加新80G的硬盘

[root@server0 ~]# poweroff

[root@Blizzard ~]# gos

[root@server0 ~]# lsblk # 查看新的硬盘设备

分区要求:划分3个主分区,1个扩展分区,3个逻辑分区

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

p 查看分区表

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

连续创建3个10G主分区

......

n 创建扩展分区

--->回车--->起始回车--->结束回车 将所有空间给扩展分区

p 查看分区表

n 创建逻辑分区--->起始回车--->结束 +10G

n 创建逻辑分区--->起始回车--->结束 +10G

n 创建逻辑分区--->起始回车--->结束 +10G

p 查看分区表

w 保存并退出

[root@server0 ~]# lsblk

LVM逻辑卷

作用: 1,整合分散的空间 2,空间可以进行扩大

ps: 没有内容的分散分区才可以整合,有存储数据的分散分区不能整合

新建逻辑卷

LVM工作方式

在”分区--->格式化”中间增加的一个逻辑层

零散空间存储--------------整合的虚拟磁盘-------------------虚拟的分区

物理卷 Physical Volume

逻辑卷Logical Volume

卷组 Volume Group

由众多的物理卷(PV)组合成卷组(VG),从卷组中划分多个逻辑卷(LV)

LVM管理工具集



1. 建立逻辑卷
2. 创建卷组

格式: vgcreate 卷组名 空闲的分区

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

Physical volume "/dev/vdc1" successfully created

Physical volume "/dev/vdc2" successfully created

Volume group "systemvg" successfully created

[root@server0 ~]# pvs # 查看物理卷基本信息

PV VG Fmt Attr PSize PFree

/dev/vdc1 systemvg lvm2 a-- 10.00g 10.00g

/dev/vdc2 systemvg lvm2 a-- 10.00g 10.00g

[root@server0 ~]# vgs # 查看卷组基本信息

VG #PV #LV #SN Attr VSize VFree

systemvg 2 0 0 wz--n- 19.99g 19.99g

1. 通过卷组创建逻辑卷

格式: lvcreate -L 逻辑卷的大小 -n 逻辑卷名 卷组名

[root@server0 ~]# lvcreate -L 16G -n mylv systemvg

Logical volume "mylv" created

[root@server0 ~]# lvs # 查看逻辑卷基本信息

LV VG Attr LSize Pool Origin Data% Move Log Cpy%Sync Convert

mylv systemvg -wi-a----- 16.00g

3、使用逻辑卷

[root@server0 ~]# mkfs.xfs /dev/systemvg/mylv

[root@server0 ~]# vim /etc/fstab # 开机自动挂载

/dev/systemvg/mylv /lvm xfs defaults 0 0

[root@server0 ~]# mount -a

[root@server0 ~]# mkdir /lvm

[root@server0 ~]# df -h #查看正在挂载使用的分区情况

文件系统 容量 已用 可用 已用% 挂载点

/dev/vda1 10G 3.4G 6.7G 34% /

devtmpfs 901M 0 901M 0% /dev

tmpfs 921M 84K 921M 1% /dev/shm

tmpfs 921M 17M 904M 2% /run

tmpfs 921M 0 921M 0% /sys/fs/cgroup

/dev/vdb2 2.0G 33M 2.0G 2% /mypart2

/dev/vdb1 976M 2.6M 907M 1% /mypart1

/dev/mapper/systemvg-mylv 16G 33M 16G 1% /lvm

二、逻辑卷的扩展,支持线上工作

1、卷组有足够的剩余空间

A: 扩展空间

[root@server0 ~]# vgs

VG #PV #LV #SN Attr VSize VFree

systemvg 2 1 0 wz--n- 19.99g 3.99g

[root@server0 ~]# lvs

LV VG Attr LSize Pool Origin Data% Move Log Cpy%Sync Convert

mylv systemvg -wi-ao---- 16.00g

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

[root@server0 ~]# lvs

LV VG Attr LSize Pool Origin Data% Move Log Cpy%Sync Convert

mylv systemvg -wi-ao---- 18.00g

B: 扩展文件系统

刷新xfs文件系统: xfs\_growfs

刷新ext4文件系统: resize2fs

[root@server0 ~]# df -h | tail -1

/dev/mapper/systemvg-mylv 16G 33M 16G 1% /lvm

[root@server0 ~]# xfs\_growfs /dev/systemvg/mylv

[root@server0 ~]# df -h | tail -1

/dev/mapper/systemvg-mylv 18G 33M 18G 1% /lvm

1. 卷组没有足够的剩余空间

A: 扩展卷组

[root@server0 ~]# vgextend systemvg /dev/vdc{3,5}

Physical volume "/dev/vdc3" successfully created

Physical volume "/dev/vdc5" successfully created

Volume group "systemvg" successfully extended

[root@server0 ~]# vgs

VG #PV #LV #SN Attr VSize VFree

systemvg 4 1 0 wz--n- 39.98g 21.98g

B: 扩展逻辑卷空间

[root@server0 ~]# lvs

LV VG Attr LSize Pool Origin Data% Move Log Cpy%Sync Convert

mylv systemvg -wi-ao---- 18.00g

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

Extending logical volume mylv to 25.00 GiB

Logical volume mylv successfully resized

[root@server0 ~]# lvs

LV VG Attr LSize Pool Origin Data% Move Log Cpy%Sync Convert

mylv systemvg -wi-ao---- 25.00g

C: 扩展逻辑卷文件系统

刷新xfs文件系统: xfs\_growfs

刷新ext4文件系统: resize2fs

[root@server0 ~]# df -h | tail -1

/dev/mapper/systemvg-mylv 18G 33M 18G 1% /lvm

[root@server0 ~]# xfs\_growfs /dev/systemvg/mylv

[root@server0 ~]# df -h | tail -1

/dev/mapper/systemvg-mylv 25G 33M 25G 1% /lvm

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

了解: 逻辑卷也可以缩减

ext4文件系统支持缩减

xfs文件系统不支持缩减

卷组划分空间的单位: PE 默认一个4M大

[root@server0 ~]# vgdisplay systemvg | head -16 | tail -1

PE Size 4.00 MiB

题目: 请划分一个逻辑卷250M,名字设置为lvredhat

[root@server0 ~]# lvcreate -L 250M -n lvredhat systemvg

Rounding up size to full physical extent 252.00 MiB

Logical volume "lvredhat" created

[root@server0 ~]# lvs

LV VG Attr LSize Pool Origin Data% Move Log Cpy%Sync Convert

lvredhat systemvg -wi-a----- 252.00m

mylv systemvg -wi-ao---- 25.00g

创建卷组的时候设置PE大小

格式: vgcreate -s PE大小 卷组名 空闲分区...

卷组已存在,修改卷组的PE大小

格式: vgchange -s PE大小 卷组名

创建逻辑卷的时候指定PE个数

格式: lvcreate -l PE个数 -n 逻辑卷名 卷组名

例子:

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

[root@server0 ~]# vgdisplay | head -16 | tail -1

PE Size 1.00 MiB

[root@server0 ~]# lvcreate -l 100 -n lvtest systemvg #根据PE个数创建逻辑卷

Logical volume "lvtest" created

[root@server0 ~]# lvs

LV VG Attr LSize Pool Origin Data% Move Log Cpy%Sync Convert

lvredhat systemvg -wi-a----- 252.00m

lvtest systemvg -wi-a----- 100.00m

mylv systemvg -wi-ao---- 25.00g

逻辑卷的删除

删除顺序: 1 删除逻辑卷---->2 删除卷组

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

Logical volume systemvg/mylv contains a filesystem in use.

# 逻辑卷正在使用

[root@server0 ~]# umount /lvm

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

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

Logical volume "mylv" successfully removed

[root@server0 ~]# lvs

LV VG Attr LSize Pool Origin Data% Move Log Cpy%Sync Convert

lvredhat systemvg -wi-a----- 250.00m

lvtest systemvg -wi-a----- 100.00m

[root@server0 ~]# lvremove /dev/systemvg/lvredhat

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

Logical volume "lvredhat" successfully removed

[root@server0 ~]# lvremove /dev/systemvg/lvtest

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

Logical volume "lvtest" successfully removed

[root@server0 ~]# vgremove /dev/systemvg (=vgremove systemvg)

Volume group "systemvg" successfully removed

[root@server0 ~]# vgs

No volume groups found

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

练习:

1. 划分分区的命令是什么?

fdisk

1. 一般在磁盘中一个扇区为多少字节?

512字节

1. Linux中新硬盘经历哪些步骤才能存档?

识别硬盘--->划分新分区--->格式化--->挂载

1. MBR常见的分区类型有哪三种?

主分区,扩展分区,逻辑分区

1. 刷新新的分区表命令是什么?

partprobe

1. 格式化ext4文件系统的命令? 格式化xfs文件系统的命令?

mkfs.ext4 mkfs.xfs

1. 查看分区的文件系统类型的命令是什么?

blkid

1. 如何在开机状态下,检测/etc/fstab文件系统是否书写正确,命令是?

mount -a

1. 查看正在挂载的分区,使用情况的命令是什么?

df -h

1. 开机挂载配置文件六个字段分别是什么?

设备路径 挂载点,类型,参数,备份标记,检测顺序

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

vim的一些快捷操作

命令模式下移动光标:键盘上下左右键,home键,End键

命令模式下行间跳转:到全文的第一行(1G或gg)、到全文的最后一行(G)、到全文的第10行(10G)

命令模式下复制粘贴:

复制一行(yy), 复制3行(3yy)

粘贴到当前行之后(小写p)

命令模式下删除:

删除单个字符(x)

删除到行首(d^), 删除到行尾(d$)

删除1行(dd), 删除3行(3dd)

命令模式下查找关键字:

搜索(/word) 切换(n、N)

补充: 在命令模式下大写的C,可以删除光标之后,并且进入输入模式

真机: 还原与启动优先虚拟机classroom

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

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

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

周期性计划任务: 每天8点30分重置虚拟机

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

30 8 \* \* \* /usr/local/bin/rht-vmctl reset classroom

30 8 \* \* \* /usr/local/bin/rht-vmctl reset server

30 8 \* \* \* /usr/local/bin/rht-vmctl reset desktop

**SHELL脚本基础**

脚本: 一个可以运行的文件,可以实现某种功能

|  |  |
| --- | --- |
| 常见的脚本语言 | Bash shell |
| Python/Perl/Ruby |
| JSP/PHP/ASP/CGI |
| JavaScipt |

什么是shell脚本:

提前设计可执行语句,用来完成特定任务的文件

案例1、第一个shell脚本

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

echo hello word

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

([root@server0 ~]# chmod +x /root/hello.sh)

[root@server0 ~]# /root/hello.sh # 绝对路径执行

hello word

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

#! /usr/bin/python

print 'hello word'

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

hello word

规范shell脚本的一般组成

1) #! 环境声明 (以下所有代码,有什么程序执行翻译)

#! /bin/bash

2) # 注释文本

3) 可执行代码

案例2、第二个shell脚本

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

#! /bin/bash

echo hello word

cat /etc/redhat-release

uname -r

hostname

ifconfig | head -2

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

hello word

Red Hat Enterprise Linux Server release 7.0 (Maipo)

3.10.0-123.el7.x86\_64

server0.example.com

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 172.25.0.11 netmask 255.255.255.0 broadcast 172.25.0.255

**简单脚本技巧**

1) 管道传递

2) 重定向输出

> 只收集前面命令正确输出结果

2> 只收集前面命令错误输出结果

&> 收集前面命令,正确与错误输出结果

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

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

123456

[root@server0 ~]# cat /opt/a

cat: /opt/a: 没有那个文件或目录

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

123456

cat: /opt/a: 没有那个文件或目录

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

cat: /opt/a: 没有那个文件或目录

[root@server0 ~]# cat /mnt/nsd.txt

123456

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

123456

[root@server0 ~]# cat /mnt/nsd.txt

cat: /opt/a: 没有那个文件或目录

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

[root@server0 ~]# cat /mnt/nsd.txt

123456

cat: /opt/a: 没有那个文件或目录

黑洞设备: /dev/null

案例3: 书写一个可以创建用户的脚本,

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

#! /bin/bash

useradd nsd04 &> /dev/null

echo 用户nsd04创建成功

echo 123 | pass --stdin nsd04 &> /dev/null

echo 用户nsd04密码设置成功

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

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

用户nsd04创建成功

用户nsd04密码设置成功

为了增加脚本的灵活性,以及适应多变的需求所以使用变量

变量:会变化的量,类似于容器,以不变的名称存储可以变化的值

变量名=会变化的值,方便以固定名称重复使用某个值,提高对任务需求、运行环境变化的适应能力

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

#! /bin/bash

a=dc # 定义变量a储存值dc

useradd $a &> /dev/null # $a使用变量a储存的值

echo 用户$a创建成功

echo 123 | pass --stdin $a &> /dev/null

echo 用户$a密码设置成功

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

用户dc创建成功

用户dc密码设置成功

定义/赋值变量

设置变量时的注意事项

1) 若指定的变量已存在,相当于为此变量重新赋值

2) 等号两边不要有空格

3) 变量名只能由字母/数字/下划线组成,区分大小写

4) 变量名不能以数字开头,不要使用关键字和特殊字符

查看/引用变量

基本格式

引用变量值: $变量名

查看变量值: echo $变量名、echo ${变量名}

[root@server0 ~]# a=rhel

[root@server0 ~]# echo $a

rhel

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

rhel

[root@server0 ~]# echo $a7 # 未定义的变量为空值

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

rhel7

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

补充:

单引号 ‘ ’ -----取消所有特殊意义字符,变为普通字符

$()或反撇号 ` ` ------将命令的输出,作为参数直接参与下一条命令

[root@server0 ~]# rm -rf /opt/\*

[root@server0 ~]# cd /opt

[root@server0 opt]# date +%F

2018-11-21

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

[root@server0 opt]# ls

2018-11-21

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

[root@server0 opt]# ls

2018-11-21 MySQL-2018-11-21

[root@server0 opt]# mkdir `hostname`

[root@server0 opt]# ls

2018-11-21 MySQL-2018-11-21 server0.example.com

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

[root@server0 opt]# ls

2018-11-21 MySQL-2018-11-21 nsd-2018-11-21 server0.example.com

**变量的种类:**

根据变量的用途不同区分:

|  |  |
| --- | --- |
| 类型 | 说明 |
| 环境变量 | 变量名一般都大写,用来设置用户/系统环境 |
| 位置变量 | bash内置,存储执行脚本时提供的命令行参数 |
| 预定义变量 | bash内置,可直接调用的特殊值,不能直接修改 |
| 自定义变量 | 用户自主设置,修改及使用 |

**环境变量**: 变量名一般都大写,由系统定义完成,并且赋值完成

常见的环境变量:PWD、PATH、USER、LOGNAME、SHELL、HOME

USER=当前登录系统的用户名

[root@server0 ~]# echo $USER

root

[root@server0 ~]# su - student

[student@server0 ~]$ echo $USER

student

[root@server0 ~]# which nmcli # 查找命令程序存在哪

/usr/bin/nmcli

[root@server0 ~]# which mkdir

/usr/bin/mkdir

[root@server0 ~]# which echo

/usr/bin/echo

[root@server0 ~]# /usr/bin/echo 123

123

**位置变量**: 由系统定义完成,并且赋值完成

作用: 在执行脚本时,提供命令行的参数

标示为$n,n为序号

$1、$2、.. ..${10}、${11}、.. ..

例1: [root@server0 ~]# vim /root/1.sh

#! /bin/bash

echo $1

echo $2

echo $3

[root@server0 ~]# chmod +x /root/1.sh

[root@server0 ~]# /root/1.sh haha xixi dcc

haha

xixi

dcc

例2:

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

#! /bin/bash

useradd $1 &> /dev/null

echo 用户$1创建成功

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

echo 用户$1密码设置成功

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

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

用户zhangsan创建成功

用户zhangsan密码设置成功

例3:

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

#! /bin/bash

cat -n $1 | head -$2

echo 显示$1文件的头$2行

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

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

2 bin:x:1:1:bin:/bin:/sbin/nologin

3 daemon:x:2:2:daemon:/sbin:/sbin/nologin

显示/etc/passwd文件的头3行

让用户体验更好,降低脚本使用难度,让脚本运行产生交互

read:

1) 产生交互,会等待用户输入内容

2) 记录用户在键盘上的输入

3) 将用户在键盘上的输入,赋值给变量储存

-p ‘屏幕输出信息’

例子:

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

#! /bin/bash

read -p '请输入您要创建的用户名:' a

useradd $a &> /dev/null

echo 用户$a创建成功

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

echo 用户$a密码设置成功

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

请输入您要创建的用户名:harry

用户harry创建成功

用户harry密码设置成功

**预定义变量**:

由系统定义完成,并且赋值完成

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

|  |  |
| --- | --- |
| 变量名 | 含义 |
| $# | 已加载的位置变量的个数(统计一共输入多少个位置变量) |
| $\* | 输出所有位置变量的值 |
| $? | 程序退出后的状态值,0表示正常,其他值异常 |

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

#! /bin/bash

echo $1

echo $2

echo $3

echo $# # 统计一共输入了多少个位置变量的值

echo $\* # 输出所有位置变量的值

[root@server0 ~]# chmod +x /root/2.sh

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

0

[root@server0 ~]# /root/2.sh dc tc dz tz

dc

tc

dz

4

dc tc dz tz

[root@server0 ~]#

[root@server0 ~]# useradd tom

useradd：用户“tom”已存在

[root@server0 ~]# echo $? # 非0值代表上条命令失败

9

[root@server0 ~]# useradd natasha

[root@server0 ~]# echo $? # 为0值代表上条命令成功

0

[root@server0 ~]# id natasha

uid=1007(natasha) gid=1007(natasha) 组=1007(natasha)

[root@server0 ~]# echo $?

0

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

**条件测试及选择**

[ 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,且为目录为真

-f: 文档存在,且为文件为真

-r: 文档存在,且对文档具备读取权限为真

-w: 文档存在,且对文档具备写权限为真

-x: 文档存在,且对文档具备执行权限为真

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

[root@server0 ~]# echo $?

0

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

[root@server0 ~]# echo $?

0

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

[root@server0 ~]# echo $?

1

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

[root@server0 ~]# echo $?

1

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

[root@server0 ~]# echo $?

0

比较整数大小

-gt: 大于

-ge: 大于等于

-eq: 等于

-ne: 不等于

-le: 小于等于

-lt: 小于

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

[root@server0 ~]# echo $?

0

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

[root@server0 ~]# echo $?

0

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

[root@server0 ~]# echo $?

1

字符串对比

==: 相等一致为真

!=: 不相等为真

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

[root@server0 ~]# echo $?

1

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

[root@server0 ~]# echo $?

0

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

[root@server0 ~]# echo $?

0

**if单分支处理**

单分支: 当条件满足时,作xx处理

**if双分支处理**

双分支: 当条件满足/不满足时,分别作xx,yy处理

if 条件测试 ; then

命令序列xx

else

命令序列yy

fi

例子1:

[root@server0 ~]# vim /root/if01.sh

#! /bin/bash

if [ $1 -eq 2 ] ; then

echo hello

else

echo hi

fi

[root@server0 ~]# chmod +x /root/if01.sh

[root@server0 ~]# /root/if01.sh 1

hi

[root@server0 ~]# /root/if01.sh 2

hello

例子2:

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

#! /bin/bash

if [ $# -eq 0 ] ; then

echo 您没有输入任何的命令行参数

else

echo 您输入了命令行参数

echo 您一共输入了$#个参数

echo 他们的值为$\*

fi

[root@server0 ~]# chmod +x /root/if02.sh

[root@server0 ~]# /root/if02.sh dc tc dz tz

您输入了命令行参数

您一共输入了4个参数

他们的值为dc tc dz tz

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

您没有输入任何的命令行参数

**if多分支处理**

多分支: 当条件1满足时,作xx处理;否则继续检查条件2,若成立则作yy处理;否则,作zz处理

if 条件测试1 ; then

命令序列xx

elif 条件测试2 ; then

命令序列yy

elif 条件测试3 ; then

命令序列aa

elif 条件测试4 ; then

命令序列ss

........

else

命令序列zz

fi

案例: 书写一个成绩脚本

用户输入考试成绩,利用read -p 手法进行获得用户成绩,

如果成绩大于等于90,则输出优秀;

如果成绩大于等于80,则输出良好;

如果成绩大于等于70,则输出一般

如果成绩大于等于60,则输出合格

如果以上条件均不满足,则输出 再牛的肖邦也弹不出哥的悲伤!

[root@server0 ~]# vim /root/if04.sh

#! /bin/bash

read -p '请输入您的成绩:' num

if [ $num -ge 90 ] ; then

echo 您的成绩为优秀

elif [ $num -ge 80 ] ; then

echo 您的成绩为良好

elif [ $num -ge 70 ] ; then

echo 您的成绩为一般

elif [ $num -ge 60 ] ; then

echo 您的成绩为合格

else

echo 再牛的肖邦也弹不出哥的悲伤!

fi

[root@server0 ~]# chmod +x /root/if04.sh

[root@server0 ~]# /root/if04.sh

请输入您的成绩:90

您的成绩为优秀

[root@server0 ~]# /root/if04.sh

请输入您的成绩:75

您的成绩为一般

[root@server0 ~]# /root/if04.sh

请输入您的成绩:34

再牛的肖邦也弹不出哥的悲伤!

**for循环结构**

for循环处理

遍历/列表式循环: 根据变量的不同取值,重复执行xx处理

for 变量名 in 值列表

do

命令序列

done

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

#! /bin/bash

for a in zhangsan lisi wangwu dc tc tz

do

useradd $a &> /dev/null

echo $a创建成功

done

[root@server0 ~]# chmod +x /root/for01.sh

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

zhangsan创建成功

lisi创建成功

wangwu创建成功

dc创建成功

tc创建成功

tz创建成功

造数工具: {起始点..结束点}

[root@server0 ~]# echo {1..10}

1 2 3 4 5 6 7 8 9 10

[root@server0 ~]# rm -rf /opt/\*

[root@server0 ~]# touch /opt/nsd{1..10}.txt

[root@server0 ~]# ls /opt

nsd10.txt nsd2.txt nsd4.txt nsd6.txt nsd8.txt

nsd1.txt nsd3.txt nsd5.txt nsd7.txt nsd9.txt

例子: 创建用户nsd1~nsd50

[root@server0 ~]# vim /root/for2.sh

#! /bin/bash

for a in {1..50}

do

useradd nsd$a &> /dev/null

echo nsd$a创建成功

done

[root@server0 ~]# chmod +x /root/for2.sh

[root@server0 ~]# /root/for2.sh

nsd1创建成功

.......

nsd49创建成功

nsd50创建成功

案例: 书写一个判断IP通信的脚本

用户输入IP地址,利用read -p 方式获得用户输入的IP地址,

判断如果本机可以与该IP地址通信,则输出 可以通信

判断如果本机不可以与该IP地址通信,则输出 无法通信

[root@server0 ~]# vim /root/for03.sh

#! /bin/bash

read -p '请输入您要测试的IP地址' ip

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

if [ $? -eq 0 ] ; then

echo $ip可以通信

else

echo $ip无法通信

fi

[root@server0 ~]# chmod +x /root/for03.sh

[root@server0 ~]# /root/for03.sh

请输入您要测试的IP地址:172.25.0.11

172.25.0.11可以通信

请输入您要测试的IP地址:172.25.2.11

172.25.2.11无法通信

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

案例4: 编写一个判断脚本

在server0上创建/root/foo.sh脚本

1) 当运行/root/foo.sh redhat , 输出fedora

2) 当运行/root/foo.sh fedora ,输出redhat

3) 当没有任何参数或者参数不是redhat 或者fedora时,其错误输出产生以下信息: /root/foo.sh redhat|fedora

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

#! /bin/bash

if [ $# -eq 0 ] ; then

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

exit 3 # 程序运行退出状态值

elif [ $1 == redhat ] ; then

echo fedora

elif [ $1 == fedora ] ;then

echo redhat

else

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

exit 4 # 程序运行退出状态值

fi

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

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

fedora

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

redhat

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

/root/foo.sh redhat|fedora

案例5: 编写一个批量添加用户脚本

在server0上创建/root/batchusers脚本

1. 此脚本要求提供用户名列表文件作为参数
2. 如果没有提供参数,此脚本应该给出提示 Usage:/root/batchusers,退出并返回相应值
3. 如果提供一个不存在的文件,此脚本应该给出提示 Input file not found ,退出并返回相应值
4. 新用户的登录shell为/bin/false ,无需设置密码
5. 用户列表测试文件: <http://classroom>/pub/materials/userlist

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

duanwu

zhongqiu

zhsan

lisi

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

#! /bin/bash

if [ $# -eq 0 ] ; then

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

exit 2

elif [ -f $1 ] ; then

for i in `cat $1`

do

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

echo $i 创建成功

done

else

echo 'Input file not found' >&2

exit 3

fi

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

[root@server0 ~]# /root/batchusers

Usage:/root/batchusers

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

duanwu 创建成功

zhongqiu 创建成功

zhsan 创建成功

lisi 创建成功

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

Input file not found

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

**系统安全防护**

SELinux安全机制

SELinux概述(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运行模式的切换

SELinux的运行模式:

enforcing (强制)

permissive (宽松)

disabled (彻底禁用)

任何模式变成disabled(彻底禁用),或者disabled变成其他模式,都需要经历重启系统

切换运行模式

临时切换: setenfore 1|0 1----强制模式 0----宽松模式

固定配置: /etc/selinux/config 文件

虚拟机server0与虚拟机desktop0

[root@server0 ~]# getenforce # 查看目前SELinux运行的模式

Enforcing

[root@server0 ~]# setenforce 0 # 临时修改模式

[root@server0 ~]# getenforce

Permissive

[root@server0 ~]# setenforce 1

[root@server0 ~]# getenforce

Enforcing

[root@desktop0 ~]# vim /etc/selinux/config # 永久修改

SELINUX=permissive

**防火墙策略管理**

**搭建基本的web服务**

1、安装httpd软件包

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

2、重启httpd服务

[root@server0 ~]# systemctl restart httpd

[root@server0 ~]# systemctl enable httpd

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

(ln -s # 软连接(创建快捷方式)

3、访问测试

[root@server0 ~]# firefox 172.25.0.11

1. 书写网页文件

默认存放网页文件路径: /var/www/html/

默认网页文件名称: index.html

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

<marquee><font color=blue><h1>闭关五个月,秒变高富帅哦

[root@server0 ~]# firefox 172.25.0.11

**搭建基本FTP服务**

FTP: 文件传输协议

1、安装vsftpd软件包

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

1. 重启vsftpd服务

[root@server0 ~]# systemctl restart vsftpd

[root@server0 ~]# systemctl enable vsftpd

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

1. 本机访问测试

FTP默认共享路径: /var/ftp/

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



[root@server0 ~]# cp /etc/redhat-release /var/ftp/

[root@server0 ~]# ls /var/ftp/

pub redhat-release

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



**防火墙规则管理**

作用: 过滤与隔离

规则: 允许出站,过滤入站

硬件防火墙:

软件防火墙:

RHEL7的防火墙体系

系统服务: firewalld

管理工具: firewall-cmd、firewall-config

预设安全区域  
 根据所在的网络场所区分,预设保护规则集

public: 仅允许访问本机的sshd、DHCP、ping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)

drop: 丢弃任何来访的数据包(没有回应,直接丢弃,节省资源)

数据包内容: 源IP地址, 目标IP地址, 数据、目标端口

防火墙判定的规则: 匹配及停止

1) 查看数据包中源IP地址,查看自己所有区域规则,那一个区域有该IP地址规则,则进入该区域

2) 进入默认区域 (public) root用户可以修改默认区域

默认区域的修改:

虚拟机server0:

[root@server0 ~]# firewall-cmd --get-default-zone # 查看默认区域

public

虚拟机desktop0:

[root@desktop0 ~]# ping -c1 172.25.0.11 # 可以通信

PING 172.25.0.11 (172.25.0.11) 56(84) bytes of data.

64 bytes from 172.25.0.11: icmp\_seq=1 ttl=64 time=0.147 ms

虚拟机server0:

[root@server0 ~]# firewall-cmd --set-default-zone=block #修改默认区域

success

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

block

虚拟机desktop0:

[root@desktop0 ~]# ping -c1 172.25.0.11 # 不可以通信,有回应

PING 172.25.0.11 (172.25.0.11) 56(84) bytes of data.

From 172.25.0.11 icmp\_seq=1 Destination Host Prohibited

虚拟机server0:

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

success

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

drop

虚拟机desktop0:

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

PING 172.25.0.11 (172.25.0.11) 56(84) bytes of data.

^C

--- 172.25.0.11 ping statistics ---

14 packets transmitted, 0 received, 100% packet loss, time 12999ms

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

常见互联网应用协议:

http: 超文本传输协议 默认端口: 80

https: 安全超文本传输协议 默认端口: 443

ftp: 文件传输协议 默认端口: 21

tftp: 简单文件传输协议 默认端口: 69

DNS: 域名解析协议 默认端口: 53

telnet: 远程管理协议 默认端口: 23

smtp: 邮件协议(用户发邮件) 默认端口: 25

pop3: 邮件协议(用户收邮件) 默认端口: 110

snmp: 网络管理协议 默认端口: 161

查看防火墙规则列表

列表查看操作

firewall-cmd --list-all [--zone=区域名]

firewall-cmd --list-all-zones

firewall-cmd --get-zones

firewall-cmd --get-services

firewall-cmd --get-default-zone

**public区域的服务的协议添加**

虚拟机server0:

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

success

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

public

[root@server0 ~]# firewall-cmd --list-all --zone=public #查看区域规则

public (default, active)

interfaces: eth0

sources:

services: dhcpv6-client ssh

ports:

masquerade: no

forward-ports:

icmp-blocks:

rich rules:

虚拟机desktop0:

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

虚拟机server0: 在public区域添加规则,添加服务访问协议http

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

success

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

public (default, active)

interfaces: eth0

sources:

services: dhcpv6-client http ssh

ports:

masquerade: no

forward-ports:

icmp-blocks:

rich rules:

虚拟机desktop0:

[root@desktop0 ~]# firefox 172.25.0.11 # 可以访问

**防火墙实现永久策略**

配置规则的位置

运行时 (--runtime)

永久 (--permanent)

虚拟机server0:

[root@server0 ~]# firewall-cmd --reload #重新加载防火墙所有永久策略

success

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

public (default, active)

interfaces: eth0

sources:

services: dhcpv6-client ssh

ports:

masquerade: no

forward-ports:

icmp-blocks:

rich rules:

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

success

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

success

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

public (default, active)

interfaces: eth0

sources:

services: dhcpv6-client http ssh

ports:

masquerade: no

forward-ports:

icmp-blocks:

rich rules:

临时删除:

[root@server0 ~]# firewall-cmd --zone=public --remove-service=http

永久删除:

[root@server0 ~]# firewall-cmd --permanent --zone=public --remove-service=http

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

**源IP地址规则的添加**

虚拟机server0: 单独拒绝desktop0访问

[root@server0 ~]# firewall-cmd --zone=block --add-source=172.25.0.10

success

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

block

interfaces:

sources: 172.25.0.10

services:

ports:

masquerade: no

forward-ports:

icmp-blocks:

rich rules:

实现本机的端口映射

案例: 本地应用的端口重定向(端口1---->端口2)

从客户机访问端口1的请求,自动映射到本机端口2

比如,访问以下两个地址可以看到相同的页面:

<http://server0.example.com:5423/>

[http://server0.example.com/](http://server0.example.com:5423/)

端口: 服务或程序或协议的编号

数据包内容: 源IP地址, 目标IP地址, 数据、目标端口

root可以改变服务器或程序或协议的端口,也可以让一个服务器具备多个端口

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

虚拟机server0:

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

public (default, active)

interfaces: eth0

sources:

services: dhcpv6-client http ssh

ports:

masquerade: no

forward-ports:

icmp-blocks:

rich rules:

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

success

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

success

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

public (default, active)

interfaces: eth0

sources:

services: dhcpv6-client http ssh

ports:

masquerade: no

forward-ports: port=5423:proto=tcp:toport=80:toaddr=

icmp-blocks:

rich rules:

虚拟机desktop0:

[root@desktop0 ~]# firefox 172.25.0.11:5423

**配置用户环境**

alias别名设置

查看已设置的别名

alias [别名名称]

定义新的别名

alias 别名名称=’实际执行的命令行’

取消已设置的别名

unalias [别名名称]

用户初始化文件

影响指定用户的bash解释环境

~/.bashrc, 每次开启bash终端时生效

影响所有用户的bash解释环境

/etc/bashrc,每次开启bash终端时生效 (需要重启终端)

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

alias hello='echo hello'

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

alias hi='echo hi'

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

alias abc='echo abc'

[root@server0 ~]# abc 重新远程管理测试(gos)

abc

[root@server0 ~]# hello

hello

[root@server0 ~]# hi

bash: hi: 未找到命令...

[root@server0 ~]# su - student

[student@server0 ~]$ hi

hi

[student@server0 ~]$ abc

abc

[student@server0 ~]$ hello

bash: hello: command not found...

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

真机: 还原与启动优先虚拟机classroom

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

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

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

环境准备: 设置两台虚拟机防火墙默认区域为trusted

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

success

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

success

**配置SMB共享(Samba共享)**

为了完成windows平台与Linux平台跨平台的共享

Samba软件项目

用途: 为客户机提供共享使用的文件夹

协议: SMB(TCP 139)、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)

必须建立系统用户,登录系统的密码可以设置,也可以不设置

必须建立系统用户,可以不用登录系统(useradd -s /sbin/nologin dc)

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

**搭建基本Samba服务**

虚拟机server0

1、安装软件(服务端软件: 提供服务)

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

1. 建立Samba共享账号

[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 #将harry添加为Samba共享账号

[root@server0 ~]# pdbedit -a kenji #将kenji添加为Samba共享账号

[root@server0 ~]# pdbedit -a chihiro #将chihiro添加为Samba共享账号

[root@server0 ~]# pdbedit -L # 查看所有Samba共享账号

harry:1001:

chihiro:1003:

kenji:1002:

3、修改配置文件进行配置

1) 创建共享目录

[root@server0 ~]# mkdir /common

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

2) 修改配置文件 /etc/samba/smb.conf

[自定共享名]

path = 文件夹绝对路径

public = no|yes 默认no

browseable = yes|no 默认yes

read only = yes|no 默认yes

write list = 用户1.... 默认无

valid users = 用户1.... 默认任何用户

hosts allow = 客户机地址....

hosts deny = 客户机地址....

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

此服务必须是STAFF工作组的一个成员

workgroup = STAFF #指定工作组

[common] # 共享名

path = /common # 共享的实际路径

4、重起smb服务

[root@server0 ~]# systemctl restart smb

[root@server0 ~]# systemctl enable smb

1. SELinux策略设置(布尔值策略:功能的开关)

需要加-P选项才能实现永久设置,消耗内存

[root@server0 ~]# getsebool -a | grep samba | head -5 | tail -2 #查看布尔值

samba\_export\_all\_ro --> off

samba\_export\_all\_rw --> off

[root@server0 ~]# setsebool samba\_export\_all\_ro on # 修改布尔值,开启

[root@server0 ~]# getsebool -a | grep samba | head -5 | tail -2

samba\_export\_all\_ro --> on

samba\_export\_all\_rw --> off

虚拟机desktop0

1. 安装软件samba-client(客户端软件: 访问服务端服务)

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

1. 列出共享资源

smbclient -L 服务器地址

[root@desktop0 ~]# smbclient -L 172.25.0.11

Enter root's password: # 直接回车,不用输密码

Anonymous login successful

1. 访问服务端共享

连接到共享文件夹

格式: smbclient -U 用户名 //服务器地址/共享名

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

Enter harry's password: # 输入harry共享密码

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

smb: \> ls

NT\_STATUS\_ACCESS\_DENIED listing \\* # 显示错误

smb: \> ls # SELinux策略设置以后就OK了,就不会报错了

. D 0 Tue Nov 27 15:08:11 2018

.. D 0 Tue Nov 27 15:07:55 2018

1.txt N 4 Tue Nov 27 15:08:11 2018

40913 blocks of size 262144. 28589 blocks available

**客户端访问服务端影响的方面**

1 防火墙

2 服务本身的访问控制

3 安全增强的SELinux 一票否定权,必须全部满足

4 本地目录的权限

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

**访问Samba共享,更加科学方便**

虚拟机desktop0

1 安装软件包

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

2 创建挂载点目录,挂载实现

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

[root@desktop0 ~]# mount -o username=harry,passwd=123 //172.25.0.11/common /mnt/nsd/

(或者[root@desktop0 ~]# mount -o user=harry,pass=123 //172.25.0.11/common /mnt/nsd/)

[root@desktop0 ~]# df -h | tail -1 #查看挂载是否成功

//172.25.0.11/common 10G 3.1G 7.0G 31% /mnt/nsd

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

1.txt

3 开机自动挂载配置文件(/etc/fstab)

\_netdev: 声明网络设备

作用: 在启动网络服务之后,再进行挂载本设备

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

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

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

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h | tail -1

//172.25.0.11/common 10G 3.1G 7.0G 31% /mnt/nsd

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

**读写Samba共享**

虚拟机server0

1 创建共享目录

[root@server0 ~]# mkdir /devops

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

2 修改配置文件 /etc/samba/smb.conf

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

[devops]

path = /devops

write list = chihiro

3 重启smb服务

[root@server0 ~]# systemctl restart smb

4 修改SELinux布尔值

[root@server0 ~]# getsebool -a | grep samba | head -5 | tail -1

samba\_export\_all\_rw --> off

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

5 本地目录权限的修改

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

[root@server0 ~]# getfacl /devops/ | head -5 | tail -1

getfacl: Removing leading '/' from absolute path names

user:chihiro:rwx

虚拟机desktop0

1 挂载访问

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

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

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

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h | tail -2

//172.25.0.11/common 10G 3.1G 7.0G 31% /mnt/nsd

//172.25.0.11/devops 10G 3.1G 7.0G 31% /mnt/dev

[root@desktop0 ~]# touch /mnt/dev/c.txt

touch: 无法创建"/mnt/dev/c.txt": 权限不够

[root@desktop0 ~]# touch /mnt/dev/c.txt # 服务端目录权限修改以后就OK了

[root@desktop0 ~]# ls /mnt/dev/

a.txt b.txt c.txt

在虚拟机server0:

发布共享/public,共享名为abc

允许kenji用户对此共享可读写权限

在虚拟机desktop0:

将虚拟机server0发布的共享abc,开机自动挂载到/mnt/test

虚拟机server0

[root@server0 ~]# mkdir /public

[root@server0 ~]# echo 123456 > /public/3.txt

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

[abc]

path = /public

write list = kenji

[root@server0 ~]# systemctl restart smb

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

虚拟机desktop0

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

//172.25.0.11/abc /mnt/test cifs defaults,user=kenji,pass=123,\_netdev 0 0

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h | head | tail -3

//172.25.0.11/common 10G 3.1G 7.0G 31% /mnt/nsd

//172.25.0.11/devops 10G 3.1G 7.0G 31% /mnt/dev

//172.25.0.11/abc 10G 3.1G 7.0G 31% /mnt/test

[root@desktop0 ~]# ls /mnt/test/

3.txt

[root@desktop0 ~]# cat /mnt/test/3.txt

123456

[root@desktop0 ~]# touch /mnt/test/5.txt

[root@desktop0 ~]# ls /mnt/test/

3.txt 5.txt

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

**multiuser机制,**针对客户端所有普通用户设计 (了解一下就可以)

作用: 在客户端访问Samba共享时,如果需要切换身份,multiuser机制可以让客户端在无需重新挂载的前提下,直接切换临时身份

SMB客户端的multiuser挂载技术

1) 管理员只需要作一次挂载

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

实现方式

1) 挂载SMB共享时启用multiuser支持

2) 使用cifscreds临时切换身份

挂载参数调整

mount.cifs的挂载参数

1) multiuser 提供对客户多个用户身份的区分支持

2) sec=ntlmssp 提供NT局域网管理安全支持

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

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

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

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h

切换共享用户身份

在客户端以普通用户登入

[root@desktop0 ~]# su - student

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

ls: cannot access /mnt/dev: Permission denied

使用cifscreds提交新的用户凭据并测试

cifscreds add -u 共享用户名 服务器地址

[student@desktop0 ~]$ cifscreds add -u chihiro 172.25.0.11

Password:

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

a.txt b.txt c.txt

[student@desktop0 ~]$ touch /mnt/dev/d.txt

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

a.txt b.txt c.txt d.txt

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

**配置NFS共享**(Linux平台与Linux平台)

NFS共享概述

Network File System,网络文件系统

用途: 为客户机提供共享使用的文件夹

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

所需软件包: nfs-utils

系统服务: nfs -server

虚拟机server0:

1、安装软件(默认已安装的)

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

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

1. 修改配置文件 /etc/exports

文件夹路径 客户机地址(权限) 客户机地址(权限)......

1) 创建共享的目录

[root@server0 ~]# mkdir /student

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

2) 修改配置文件

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

/student \*(ro) # 任何人都可以访问,以只读模式

3) 重启nfs-server服务

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

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

虚拟机desktop0:

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

172.25.0.11:/student /mnt/stu nfs defaults,\_netdev 0 0

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

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h | tail -1

172.25.0.11:/student 10G 3.1G 7.0G 31% /mnt/stu

[root@desktop0 ~]# cat /mnt/stu/abc.txt

haha

[root@desktop0 ~]# touch /mnt/stu/123.txt

touch: 无法创建"/mnt/stu/123.txt": 只读文件系统

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

复习:

1. Linux中黑洞设备是什么?

/dev/null

2、Linux中正确信息与错误信息都收集的重定向是什么?

&>

3、下列变量名中有效的shell变量名是( C )

A: -2-time B: \_2$3

C: trust\_no\_1 D: 2004file

1. vim编辑器有几种工作模式,分别是什么?

3种 命令模式、插入模式(编辑模式)、末行模式

1. 开机挂载配置文件六个字段分别为什么?

/etc/fstab 设备路径,挂载点,文件系统类型,参数,备份标记,检测顺序

1. 搭建smb服务,需要安装什么软件?

samba

1. 配置nfs共享,配置文件在哪里?

/etc/exports

1. yum客户端配置文件放在什么路径下? 具体都有哪些字段?

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

[] name baseurl enabled gpgcheck

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

真机:还原与启动优先虚拟机classroom

真机: 还原与启动优先虚拟机classroom

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

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

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

环境准备: 设置两台虚拟机防火墙默认区域为trusted

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

success

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

success

**iscsi网络磁盘 默认端口:3260**

Internet SCSI,网际SCSI接口

一种基于C/S架构的虚拟磁盘技术

服务器提供磁盘空间,客户机连接并当成本地磁盘使用

虚拟机server0:

一、利用fdisk命令划分一个新的分区5G,用于共享iscsi磁盘

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

[root@server0 ~]# lsblk

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

/dev/vdb1

iscsi磁盘的构成

backstore------后端存储 /dev/vdb1 大电视

对应到服务端提供实际存储空间的设备,需要起一个管理名称

target------磁盘组 木质箱子

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 (将大电视+纸质箱子 放入 木质箱子)

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

举例: 大电视------>纸质包装------>木质箱子

ISCSI Qualified Name名称规范(iqn)

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

示例: iqn.2018-11.com.example:server0

1. 安装targetcli软件包,提供共享iscsi磁盘

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

1. 运行targetcli进行配置

[root@server0 ~]# targetcli #进入共享iscsi磁盘配置模式

1、建立backstore(后端存储) (大电视---->纸质包装)

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

Created block storage object nsd using /dev/vdb1.

后端存储/块设备 创建 名称(纸质包装) 设备(大电视)

2、建立target(磁盘组) (建立木质的箱子)

/> iscsi/ create iqn.2018-11.com.example:server0

Created target iqn.2018-11.com.example:server0.

Created TPG 1.

3、配置lun关联 (将大电视+纸质包装放入木质的箱子)

/> iscsi/iqn.2018-11.com.example:server0/tpg1/luns create /backstores/block/nsd

Created LUN 0.

4、配置访问控制: 设置客户端声称的名字

[root@server0 ~]# targetcli

/> iscsi/iqn.2018-11.com.example:server0/tpg1/acls create iqn.2018-11.com.example:desktop0

Created Node ACL for iqn.2018-11.com.example:desktop0

Created mapped LUN 0.

5、启用IP地址与端口

/> iscsi/iqn.2018-11.com.example:server0/tpg1/portals / create 172.25.0.11

Using default IP port 3260

Created network portal 172.25.0.11:3260.

本机的172.25.0.11端口为3260提供,相应的共享存储服务

/> exit

Global pref auto\_save\_on\_exit=true

Last 10 configs saved in /etc/target/backup.

Configuration saved to /etc/target/saveconfig.json

1. 服务端重启服务

[root@server0 ~]# systemctl restart target

[root@server0 ~]# systemctl enable target

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

客户端 虚拟机desktop0:

1、安装软件包,访问共享存储服务

所需软件包: iscsi-initiator-utils -------教学环境已安装

yum仓库安装包的补全:

1) 本机没有安装此包;

2) yum具备缓存 (yum repolist-----可以生成yum具备的缓存)

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

[root@desktop0 ~]# yum -y install iscsi-initiator-utils.i686 ---32位的包

[root@desktop0 ~]# rpm -q iscsi-initiator-utils

iscsi-initiator-utils-6.2.0.873-21.el7.x86\_64

1. 进行配置客户端声称的名字

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

InitiatorName=iqn.2018-11.com.example:desktop0

1. 重启iscsid服务,专门刷新客户端声称的名字 (不需设置开机自启)

[root@desktop0 ~]# systemctl restart iscsid

1. 运行命令,发现共享存储------通过man iscsiadm帮助进行参考,

/example -------全文查找example, 然后按n进行跳转

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

172.25.0.11:3260,1 iqn.2018-11.com.example:server0

1. 重启使用iscsi服务,加载共享存储

[root@desktop0 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 253:0 0 10G 0 disk

└─vda1 253:1 0 10G 0 part /

vdb 253:16 0 10G 0 disk

[root@desktop0 ~]# systemctl restart iscsi

[root@desktop0 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 5G 0 disk

vda 253:0 0 10G 0 disk

└─vda1 253:1 0 10G 0 part /

vdb 253:16 0 10G 0 disk

[root@desktop0 ~]# systemctl enable iscsi

总结:

虚拟机server0: 服务端

1, 修改防火墙默认区域为trusted

2, 安装软件包targetcli

3, 运行targetcli进行配置

1) 建立后端存储

2) 建立target磁盘组(iqn命名)

3) 进行lun关联

4) 配置acl访问控制,设置客户端声称的名字(iqn命名)

5) 启用IP地址与端口(设置本机利用哪一个IP地址,进行提供服务)

4, 重启target服务

虚拟机desktop0: 客户端

1 防火墙设置为默认区域trusted

2 安装所需软件包: iscsi-initiator-utils

3 修改配置文件,指定客户端声称的名字(/etc/iscsi/initiatorname.iscsi)

4 重启iscsid服务,刷新客户端声称的名字

5 运行命令,发现共享存储

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

6 重启iscsi服务,加载共享存储

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

**MariaDB数据库的管理**

DB: database

数据库: 存放数据的仓库,一批数据的集合,主流的数据库多用来存放关系型表格数据

关系型数据: 以二维表格记录大量实体的属性信息

DBMS: 数据库管理系统

在数据库中会有很多的库.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

部署mariadb数据库服务器, 默认端口: 3306

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

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

[root@server0 ~]# systemctl restart mariadb

[root@server0 ~]# systemctl enable mariadb

**数据库的基本操作**

**数据库的增删查**

MariaDB [(none)]> 交互指令

列出数据库: show databases;

使用/选择数据库: use 数据库名;

列出库里有哪些表: show tables;

创建数据库: create database 数据库名;

删除数据库: drop database 数据库名;

[root@server0 ~]# mysql #进入数据库

MariaDB [(none)]> show databases; # 显示所有库

MariaDB [(none)]> create database nsd1811; 创建nsd1811库

MariaDB [(none)]> show databases; #显示所有库

MariaDB [(none)]> drop database nsd1811; # 删除nsd1811库

MariaDB [(none)]> show databases;

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

MariaDB [(none)]> show databases;

MariaDB [(none)]> exit

**为数据库管理员设置密码**

数据库管理员: 账户: root 用户信息存放在MySQL库中user表中

系统管理员: 账户: root 用户信息放在/etc/passwd

格式: mysqladmin [-u 用户名] [-p[旧密码]] password ‘新密码’

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

[root@server0 ~]# mysql -u root -p # 交互式进入

Enter password:

[root@server0 ~]# mysql -u root -p123 # 非交互式验证

禁止监听,只服务于本机

[root@server0 ~]# vim /etc/my.cnf # 主配置文件

[mysqld]

skip-networking # 跳过网络监听

.. ..

[root@server0 ~]# systemctl restart mariadb

**数据表的增删查**

创建新的数据表

create table 表名(字段1名 类型(长度), 字段2名 类型(长度), .. ..);

查看表结构

describe 表名;

删除数据表

drop table 表名;

**表格的操作: 表记录 表字段**

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

insert插入表记录

insert into [数据库.]表名 values(值1,值2, .. ..);

select查询表记录

select \* from [数据库.]表名;

select 字段1,字段2,... from [数据库.]表名;

select .. .. where 字段1=’值’ and|or 字段2=’值’;

update修改表记录

update [数据库名.]表名 set 字段名=新值 where 条件语句;

delete删除表记录

delete from [数据库.]表名 where条件语句;

多表组合查询

统计查询结果数量: select count(\*) .. ..;

关联查询: .. ..from 表1,表2 where 表1.字段1=’值’ and 表2.字段2=’值’ and 表1.字段1=表2.字段2;

虚拟机classroom: 数据库的备份文件

[root@server0 ~]# wget http://classroom.example.com/pub/materials/users.sql

数据库的导出和导入操作

导出/备份数据库

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

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

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

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

MariaDB [(none)]> use nsd; # 切换到nsd库

MariaDB [nsd]> show tables; # 显示有哪些表格

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

MariaDB [nsd]> select \* from location; #查看location表所有表字段内容

where: 有条件的查询

MariaDB [nsd]> select \* from base where password='123';

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

MariaDB [nsd]> select password from base;

MariaDB [nsd]> select name,password from base;

查看表结构desc

MariaDB [nsd]> desc base; # 查看base表结构

+----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------+-------------+------+-----+---------+-------+

| id | int(11) | YES | | NULL | |

| name | varchar(20) | YES | | NULL | |

| password | varchar(24) | YES | | NULL | |

+----------+-------------+------+-----+---------+-------+

**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: 除了root用户,数据库nsd只能被用户lisi查询, 此用户密码为123

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

Query OK, 0 rows affected (0.00 sec)

# 当lisi从localhost登录,输入密码123,将会获得nsd库所有表的查询权限

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

MariaDB [mysql]> exit

Bye

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

MariaDB [(none)]> show databases;

案例: 使用数据库查询

1 在系统server0上使用数据库nsd, 并使用相应的SQL查询以回答下列问题:

1. 密码是solicitous的人的名字?

MariaDB [(none)]> use nsd;

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

2) 有多少人的姓名是Barbara同时居住在sunnyvale?

MariaDB [nsd]> select count(\*) from base,location where base.name='Barbara' and location.city='sunnyvale' and base.id=location.id;

2 禁止空密码root用户访问mariadb数据库

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

Query OK, 5 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges; # 刷新数据库user表记录

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit

[root@server0 ~]# mysql -u root -h server0.example.com # 验证

ERROR 1130 (HY000): Host 'server0.example.com' is not allowed to connect to this MariaDB server

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

真机:还原与启动优先虚拟机classroom

真机: 还原与启动优先虚拟机classroom

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

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

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

环境准备: 设置两台虚拟机防火墙默认区域为trusted

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

success

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

success

web常用计算机语言: Apache(httpd)、Nginx、Tomcat

**HTTP服务基础**

独立web主机

web通信基本概念

RHEL7中的web服务

独立web站点的快速部署

访问web站点

虚拟web主机

虚拟主机的含义及类型

配置一个虚拟站点

对默认web站点的影响

web通信基本概念: 基于B/S(Browser/Server)架构的网页服务

作用: 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)

1. 搭建基本的web服务器

虚拟机server0:

1 安装httpd软件包

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

2 重启httpd服务,设置开机自启服务

[root@server0 ~]# systemctl restart httpd

[root@server0 ~]# systemctl enable httpd

3 写网站

[root@server0 ~]# echo '<h1>NSD1811 web' > /var/www/html/index.html

虚拟机desktop0:

1 利用firefox 172.25.0.11测试

[root@desktop0 ~]# firefox 172.25.0.11



RHEL7中的web服务

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: 网页文件根目录(/var/www/html)

DirectoryIndex: 起始页/首页文件名(index.html)

DNS服务器:虚拟机classroom只能解析下面三种域名:

server0.example.com

www0.example.com

webapp0.example.com

测试DNS解析: nslookup 域名

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

Server: 172.25.254.254

Address: 172.25.254.254#53

Name: server0.example.com

Address: 172.25.0.11

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

Server: 172.25.254.254

Address: 172.25.254.254#53

www0.example.com canonical name = server0.example.com.

Name: server0.example.com

Address: 172.25.0.11

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

Server: 172.25.254.254

Address: 172.25.254.254#53

webapp0.example.com canonical name =

server0.example.com.

Name: server0.example.com

Address: 172.25.0.11

DocumentRoot: 网页目录(/var/www/html)

虚拟机server0:

1 建立新的网页文件存放路径

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

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

2 修改配置文件

[root@server0 ~]# vim /etc/httpd/conf/httpd.conf #修改主配置文件

DocumentRoot "/var/www/html"

DocumentRoot "/var/www/myweb"

3 重启httpd服务

[root@server0 ~]# systemctl restart httpd

虚拟机desktop0:

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



**网络路径和实际路径对应关系**

网络路径: 在浏览器输入的路径

实际路径: 服务器存放网页文件的路径

客户端: firefox server0.example.com

服务端: /var/www/myweb

firefox server.example.com------>DNS----->172.25.0.11

172.25.0.11:80----->服务端----->httpd----->DocumentRoot----->/var/www/myweb(网页文件根目录)

虚拟机server0:

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

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

虚拟机desktop0:

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



**虚拟web主机**

作用: 一台服务器可以提供多个web站点

区分方式:

1) 基于域名的虚拟主机

2) 基于端口的虚拟主机

3) 基于IP地址的虚拟主机

配置一个虚拟站点

配置文件路径:

1) /etc/httpd/conf/httpd.conf 主配置文件

2) /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> **记住**

.. ..

虚拟机server0:

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

<VirtualHost \*:80>

ServerName www0.example.com

DocumentRoot /var/www/qq

</VirtualHost>

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/baidu

</VirtualHost>

[root@server0 ~]# mkdir /var/www/qq /var/www/baidu

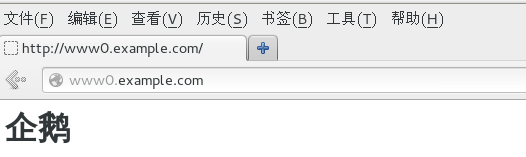
[root@server0 ~]# echo '<h1>企鹅' > /var/www/qq/index.html

[root@server0 ~]# echo '<h1>百度' > /var/www/baidu/index.html

[root@server0 ~]# systemctl restart httpd

虚拟机desktop0:

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



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



**一旦使用虚拟web主机功能,所有的站点都必须用虚拟web主机来实现**

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

<VirtualHost \*:80>

ServerName www0.example.com

DocumentRoot /var/www/qq

</VirtualHost>

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/baidu

</VirtualHost>

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /var/www/myweb

</VirtualHost>

**配置目录访问**

文件夹权限: 针对DocumentRoot网页目录的权限控制

httpd运行身份(用户/组): apache

能够提取哪些网页资源

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或granted

Require ip或网段地址.. ..

</Directory>

禁止任何客户机访问

<Directory />

Require all denied

</Directory>

允许任何客户机访问

<Directory /var/www/html>

Require all granted

</Directory>

仅允许部分客户机访问

<Directory /var/www/html/private>

Require ip 172.25.0.11

</Directory>

案例3: 配置网页内容访问

在web网站http://server0.example.com的DocumentRoot目录下创建一个名为private的子目录

1. 从http://classroom/pub/materials/private.html下载一个文件副本到这个目录,重命名为index.html
2. 不要对文件index.html的内容作任何修改
3. 从server0上, 任何人都可以浏览private的内容,但是从其他系统不能访问这个目录的内容

查看http://server0.example.com的DocumentRoot目录

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

建立相应的网页文件目录

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

案例4: 使用自定web根目录

调整web站点http://server0.example.com的网页目录,要求如下:

1) 新建目录/webroot , 作为此站点新的网页目录

2) 从http://classroom/pub/materials/station.html 下载一个文件副本到这个目录,重命名为index.html

3) 不要对文件index.html的内容做任何修改

4) 确保站点http://server0.example.com 仍然可以访问

添加新的web目录

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 (后面了解)

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

engineer-day06 下午 246分钟

**部署动态网站**

LAMP: Linux Apache MariaDB/MySQL php/Python

LNMP: Linux Nginx MariaDB/MySQL php/Python

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服务端软件接受请求,动态程序转后端模块处理

PHP网页、Python网页、JSP网页.....

1 部署动态网站文件

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

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

[root@server0 baidu]# wget http://classroom.example.com/pub/materials/webinfo.wsgi

2 为了方便用户的访问,用户可以直接输入webapp0.example.com得到相应的Python页面,所以使用页面跳转(页面别名/地址重写)

Alias 网络路径 实际路径

3 安装mod\_wsgi软件,专用于Python页面代码翻译

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

4 修改配置文件进行翻译

engineer-Day06 下午 178分钟

端口的优先级最高;

虚拟web由上及下的匹配,.在最上面的优先级最高

复习:

1 vim编辑器命令模式,删除光标之后如何操作

C(大写) d$

2 开机挂载配置文件与指定DNS服务器配置文件,请写出路径

/etc/fstab /etc/resolv.conf

3 Samba文件系统是什么

cifs

4 配置nfs共享,配置文件在哪里?

/etc/exports

5 yum客户端配置文件放在什么路径下? 具体都有哪些字段?

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

[] name baseurl enabled gpgcheck

6 配置Apache的httpd服务时,虚拟web主机区分方式都有哪些?

基于域名方式 基于端口方式 基于IP地址方式

7 配置Apache的httpd服务,指定网页文件根目录,配置字段是什么?

DocumentRoot

8 httpd服务进行虚拟web主机配置,具体格式是什么

<VirtualHost \*:80>

ServerName

DocumentRoot

</VirtualHost>

9 请写出您熟知的Linux命令,至少10条以上?

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

**安全web服务(**https 默认端口443**)**

虚拟机server0: 搭建安全的web

1 安装mod\_ssl软件包,支持安全加密https

engineer-day07 上28分钟