# 云计算应用管理

## Shell脚本基础

脚本:一个可以执行的文件,运行可以实现某种功能

创建用户dc = useradd dc

绿色:可执行程序

案例1:编写hello.sh脚本

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

echo Hello World

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

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

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

### 规范Shell脚本的一般组成

• #! 环境声明,以下代码由谁进行翻译执行

• # 注释文本

• 可执行代码

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

1)输出当前红帽系统的版本信息

2)输出当前使用的内核版本

3)输出当前系统的主机名

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

#!/bin/bash

echo Hello World

cat /etc/redhat-release

uname -r

hostname

ifconfig | head -2

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

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

### 重定向输出

> :只收集前面命令的正确输出

2>:只收集前面命令的错误输出

&>:正确与错误都收集

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

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

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

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

[root@server0 ~]# cat /opt/2.txt

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

[root@server0 ~]# cat /opt/2.txt

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

[root@server0 ~]# cat /opt/2.txt

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

### 单引号 ' ' : 取消所有特殊字符的意义,原样输出

[root@server0 ~]# echo '\* | & ^ % \*\*\*'

[root@server0 ~]# echo ', !'

### 反撇号 ` `或$( ) :将命令的输出,直接参与下一次命令的运行

[root@server0 ~]# date +%F

2019-07-06

[root@server0 ~]# cd /opt/

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

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

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

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

案例:创建用户并且设置密码的脚本

### 黑洞设备:/dev/null

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

#!/bin/bash

useradd nsd04 &> /dev/null

echo '用户nsd04创建成功'

echo 123 | passwd --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

为了增强脚本适应环境的变化以及需求的变化所有使用变量

### **变量**

:会变化的量 以不变的名称,存储可以变化的值

类似与容器

格式: 变量名=存储的值

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

#!/bin/bash

a=tc

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

交互式在命令行传递参数给脚本的代码

read -p '屏幕提示信息'

1.直接产生交互

2.记录用户在键盘上所有输入

3.将记录的内容交由变量储存

[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

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

vim错误解决:产生交换文件(缓存文件)

]# vim /opt/4.txt

]# rm -rf /opt/.4.txt.swp #删除交换文件

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

### 定义/赋值变量

• 设置变量时的注意事项

– 等号两边不要有空格

– 变量名只能由字母/数字/下划线组成,区分大小写

– 变量名不能以数字开头,不要使用关键字和特殊字符

– 若指定的变量名已存在,相当于为此变量重新赋值

### 基本格式

– 引用变量值:$变量名

– 查看变量值:echo $变量名、echo ${变量名}

[root@server0 ~]# a=rhel

[root@server0 ~]# echo $a

rhel

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

rhel

[root@server0 ~]# echo $a7

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

rhel7

[root@server0 ~]# b=7

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

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

### **环境变量 :变量名大写**

由系统定义并且赋值完成

USER=当前登录的用户身份

[root@server0 ~]# echo $USER

root

[root@server0 ~]# su - student

[student@server0 ~]$ echo $USER

student

[student@server0 ~]$ exit

logout

[root@server0 ~]# su - dc

[dc@server0 ~]$ echo $USER

dc

[dc@server0 ~]$ exit

logout

[root@server0 ~]#

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

### 位置变量

:由系统定义并且赋值完成

非交互式 在命令行传递参数给脚本的代码

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

#!/bin/bash

echo $1

echo $2

echo $3

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

[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 ~]# /root/user.sh jack

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

#!/bin/bash

cat -n $1 | head -$2

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

[root@server0 ~]# /root/2.sh /etc/passwd 3

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

### 预定义变量

:由系统定义并且赋值完成

$# 已加载的位置变量的个数,判断用户是否输入命令行参数

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

#!/bin/bash

echo $1

echo $2

echo $3

echo $#

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

$? 程序退出后的状态值,0表示正常,其他值异常

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

### 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:存在及为真

-d:存在并且必须为目录 才为真

-f:存在并且必须为文件 才为真

[root@server0 ~]# [ -e /aaa ]

[root@server0 ~]# echo $?

1

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

[root@server0 ~]# echo $?

0

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

[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

-lt:小于

-le:小于等于

[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

[root@server0 ~]# [ 10 -le 12 ]

[root@server0 ~]# echo $?

0

[root@server0 ~]#

### 字符串比对

==:一致为真

!=:不一致为真

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

[root@server0 ~]# echo $?

1

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

[root@server0 ~]# echo $?

0

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

[root@server0 ~]# echo $?

0

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

### if双分支处理

if [条件测试];then

命令序列xx

else

命令序列yy

fi

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

#!/bin/bash

if [ $1 -eq 1 ];then

echo hello

else

echo hi

fi

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

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

判断用户是否输入参数(位置变量)

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

#!/bin/bash

if [ $# -eq 0 ];then

echo 您没有输入参数

else

echo 您输入了参数

fi

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

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

案例:利用read获取用户名进行判断

如果用户存在,则输出 用户已存在

如果用户不存在,则输出 用户不存在

[root@server0 ~]# vim /root/if03.sh

#!/bin/bash

read -p '请输入您要测试的用户名:' a

id $a &> /dev/null

if [ $? -eq 0 ];then

echo 用户$a已存在

else

echo 用户$a不存在

fi

[root@server0 ~]# /root/if03.sh

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

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

#!/bin/bash

read -p '请输入您要测试的IP地址:' ip

ping -c 2 $ip &> /dev/null #-c指定ping的次数

if [ $? -eq 0 ];then

echo $ip可以通信

else

echo $ip不可以通信

fi

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

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

### if多分支处理

if [条件测试1];then

命令序列xx

elif [条件测试2];then

命令序列yy

else

命令序列zz

fi

案例:利用read获取用户的成绩

如果大于等于90,则输出 优秀

如果大于等于80,则输出 良好

如果大于等于70,则输出 一般

如果大于等于60,则输出 合格

以上均不满足,则输出 再牛的肖邦也弹不出哥的悲伤

[root@server0 ~]# vim /root/if05.sh

#!/bin/bash

read -p '请输入您的成绩:' num

if [ $num -ge 90 ];then

echo 优秀

elif [ $num -ge 80 ];then

echo 良好

elif [ $num -ge 70 ];then

echo 一般

elif [ $num -ge 60 ];then

echo 合格

else

echo 再牛的肖邦也弹不出哥的悲伤

fi

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

### 列表式循环,解决重复性的操作

for 献血车 in 献血的队伍

do

抽血

done

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

#!/bin/bash

for a in zhangsan lisi wangwu haha xixi

do

echo $a

done

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

#!/bin/bash

for a in zhangsan lisi wangwu haha xixi

do

useradd $a &> /dev/null

echo $a用户创建成功

done

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

列表值可以不参与,循环代码的执行

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

#!/bin/bash

for a in zhangsan lisi wangwu haha xixi

do

echo I Love DC

done

### 造数工具:

制造连续范围的数字 {起始值..结束值}

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

#!/bin/bash

for a in {1..99}

do

sleep 0.5 #暂停0.5秒

echo I Love DC $a

done

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

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

案例4:编写一个判断脚本

在 server0 上创建 /root/foo.sh 脚本

1)当运行/root/foo.sh redhat,输出为fedora

2)当运行/root/foo.sh fedora,输出为redhat

3)当没有任何参数或者参数不是 redhat 或者fedora时,

其错误输出产生以下信息: /root/foo.sh redhat|fedora

案例4:编写一个判断脚本

在 server0 上创建 /root/foo.sh 脚本

1)当运行/root/foo.sh redhat,输出为fedora

2)当运行/root/foo.sh fedora,输出为redhat

3)当没有任何参数或者参数不是 redhat 或者fedora时,

其错误输出产生以下信息: /root/foo.sh redhat|fedora

答案：

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

#!/bin/bash

# 判断$1参数的脚本

if [ $# -eq 0 ];then #判断用户是否输入参数

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

exit 3 #脚本退出返回的状态值

elif [ $1 == redhat ];then #判断是否为redhat

echo fedora

elif [ $1 == fedora ];then #判断是否为fedora

echo redhat

else

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

exit 4 #脚本退出返回的状态值

fi

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 ~]# vim /root/userlist

duanwu

zhongqiu

zhsan

lisi

dc

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

#!/bin/bash

if [ $# -eq 0 ];then #判断是否有参数

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

exit 2

elif [ -f $1 ];then #判断文件是否存在

for a in $(cat $1)

do

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

echo $a创建成功

done

else

echo 'Input file not found' >&2

exit 3

fi

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

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

## 系统安全保护

### SELinux安全机制

• Security-Enhanced Linux

– 美国NSA国家安全局主导开发,一套增强Linux系统安

全的强制访问控制体系

– 集成到Linux内核(2.6及以上)中运行

– RHEL7基于SELinux体系针对用户、进程、目录和文件

提供了预设的保护策略,以及管理工具

### SELinux的运行模式

– enforcing(强制)、permissive(宽松)

– disabled(彻底禁用)

**任何模式切换成disabled模式,都要经历重启**

### 切换运行模式

– 临时切换:setenforce 1|0

– 固定配置:/etc/selinux/config 文件

虚拟机server0

[root@server0 ~]# getenforce #查看当前的状态

[root@server0 ~]# setenforce 0 #临时修改为宽松模式

[root@server0 ~]# getenforce

[root@server0 ~]# vim /etc/selinux/config

SELinux=permissive

虚拟机desktop0

[root@desktop0 ~]# getenforce

[root@desktop0 ~]# setenforce 0

[root@desktop0 ~]# getenforce

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

SELinux=permissive

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

## 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]

• 影响指定用户的 bash 解释环境

– **~/.bashrc**,每次开启 bash 终端时生效

• 影响所有用户的 bash 解释环境

– **/etc/bashrc**,每次开启 bash 终端时生效

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

alias hello='echo hello'

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

alias hi='echo hi'

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

alias dc='echo tc'

新开一个终端验证

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

虚拟机server0:搭建Web服务器

## Web服务:提供网页内容

实现Web服务的软件: httpd Nginx Tomcat

实现Web通信的协议: http(超文本传输协议)

### 1.安装httpd软件

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

### 2.重启程序(重启服务)

[root@server0 ~]# systemctl restart httpd

### 3.本机测试访问

[root@server0 ~]# firefox 172.25.0.11

### 4.书写自己的页面文件

默认存放路径:/var/www/html

默认首页文件名称:index.html

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

<marquee><font color=red><h1>NSD1906 haha

滚动 字体颜色 标题字体

[root@server0 ~]# firefox 172.25.0.11

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

## 虚拟机server0:搭建FTP服务

FTP服务:传输数据

FTP协议:文件传输协议

实现FTP服务软件:vsftpd

默认共享路径:/var/ftp

### 1.安装vsftpd软件

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

### 2.重启程序(重启服务)

[root@server0 ~]# systemctl restart vsftpd

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

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

## 防火墙策略管理

作用: 隔离 众多的策略,允许出站,严格控制入站

硬件防火墙

软件防火墙

firewalld服务基础

• 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ssh dhcp ping服务

– trusted:允许任何访问

– block:阻塞任何来访请求(明确拒绝,有回应客户端)

– drop:丢弃任何来访的数据包(没有回应,节省服务端资源)

根据数据包的源IP地址

数据包: 源IP地址 目标IP地址 数据

### 防火墙判定原则:

#### 1.查看数据包的源IP地址,然后查看自己所有的区域,那个区域中有该源IP地址的规则,则进入该区域

#### 2.进入默认区域(public)

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

防火墙默认区域的修改

虚拟机server:

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

虚拟机desktop:

]# ping 172.25.0.11 #可以通信,有回应

虚拟机server:

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

]# firewall-cmd --get-default-zone

虚拟机desktop:

]# ping 172.25.0.11 #不可以通信,有回应

虚拟机server:

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

]# firewall-cmd --get-default-zone

虚拟机desktop:

]# ping 172.25.0.11 #不可以通信,没有回应

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

## 互联网常见的协议:

http:超文本传输协议 默认端口:80

https:安全的超文本传输协议 默认端口:443

DNS:域名解析协议 默认端口:53

FTP:文件传输协议 默认端口:21

tftp:简单的文件传输协议 默认端口:69

telnet:远程管理协议(明文传输) 默认端口:23

ssh:远程管理协议(加密传输) 默认端口:22

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:简单的网络管理协议 默认端口:161

### 防火墙区域添加允许的协议

虚拟机server

]# firewall-cmd --set-default-zone=public #修改默认区域

]# firewall-cmd --zone=public --list-all #查看区域规则

]# firewall-cmd --zone=public --add-service=ftp #添加协议

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

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

虚拟机desktop

[root@desktop0 ~]# firefox 172.25.0.11 #访问成功

[root@desktop0 ~]# firefox ftp://172.25.0.11 #访问成功

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

### 防火墙规则的永久设置

– 永久(permanent)

]# firewall-cmd --reload #重新加载防火墙所有配置

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

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

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

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

]# firewall-cmd --reload

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

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

### 单独拒绝虚拟机desktop访问

虚拟机server:将虚拟机desktop的IP地址写入block

]# firewall-cmd --zone=block --add-source=172.25.0.10

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

虚拟机desktop:

]# firefox 172.25.0.11 #访问失败

]# firefox ftp://172.25.0.11 #访问失败

真机:

]# firefox 172.25.0.11 #可以进行访问

]# firefox ftp://172.25.0.11 #可以进行访问

虚拟机server: 删除规则

]# firewall-cmd --zone=block --remove-source=172.25.0.10

虚拟机desktop:

]# firefox 172.25.0.11 #可以进行访问

]# firefox ftp://172.25.0.11 #可以进行访问

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

### 端口:

协议或程序或服务的编号

利用root可以改变端口,而且一个程序可以具备多个端口

监听端口

#### 实现本机的端口映射

• 本地应用的端口重定向(端口1 --> 端口2)

客户端访问: firefox 172.25.0.11:5423--->172.25.0.11:80

虚拟机server:

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

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

]# firewall-cmd --reload

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

虚拟机desktop:

[root@desktop0 ~]# firefox 172.25.0.11:5423

## SWAP交换空间

准备交换空间(虚拟内存)

缓解真实物理内存的压力

• 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

一 利用fdisk进行分区,划分两个主分区分别为2G

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

[root@server0 ~]# lsblk

### 二 格式化为交换文件系统

[root@server0 ~]# mkswap /dev/vdb1 #专业格式化交换文件系统

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

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

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

### 三 启用交换分区

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

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

[root@server0 ~]# swapon -s #查看交换空间的成员信息

[root@server0 ~]# swapoff /dev/vdb1 #停用交换分区

[root@server0 ~]# swapon -s

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

[root@server0 ~]# swapon -s

### 四 开机自动启用交换分区

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

/dev/vdb1 swap swap defaults 0 0

/dev/vdb2 swap swap defaults 0 0

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

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

[root@server0 ~]# swapon -s

[root@server0 ~]# ***swapon -a*** #专门检测/etc/fstab交换分区

[root@server0 ~]# swapon -s

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

## 补充内容:触发挂载(autofs)

准备工作:在划分一个2G的主分区

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

[root@server0 ~]# partprobe #刷新分区表

[root@server0 ~]# lsblk

[root@server0 ~]# mkfs.xfs /dev/vdb3 #格式化文件系统

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

### 1.安装触发挂载的软件包

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

### 2.重启autofs程序

[root@server0 ~]# systemctl restart autofs

[root@server0 ~]# systemctl enable autofs

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

### 3.完成对/dev/vdb3触发挂载的配置

最终实现:访问路径/nsd/dc 挂载设备/dev/vdb3 到/nsd/dc

主配置文件:定义监控目录/etc/auto.master

格式:监控目录路径 挂载配置文件路径

挂载配置文件:定义挂载点目录与设备

cd -fstype=iso9660,ro,nosuid,nodev :/dev/cdrom

挂载点目录 -fstype=文件系统 :挂载设备

[root@server0 /]# mkdir /nsd #必须手动创建监控目录

[root@server0 /]# vim /etc/auto.master

/nsd /etc/test

格式:监控目录路径 挂载配置文件路径

[root@server0 /]# cp /etc/auto.misc /etc/test

[root@server0 /]# vim /etc/test #修改挂载配置文件

dc -fstype=xfs :/dev/vdb3

格式: 挂载点目录 -fstype=文件系统 :挂载设备

[root@server0 /]# systemctl restart autofs

[root@server0 /]# systemctl status autofs #查看服务状态

[root@server0 /]# ls /nsd/dc

[root@server0 /]# df –h

## iSCSI网络磁盘

• Internet SCSI,网际SCSI接口 默认端口:3260

– 一种基于C/S架构的虚拟磁盘技术

– 服务器提供磁盘空间,客户机连接并当成本地磁盘使用

### • backstore,后端存储

– 对应到服务端提供实际存储空间的设备,需要起一个管理名称

### • target,磁盘组

– 是客户端的访问目标,作为一个框架,由多个lun组成

### • lun,逻辑单元

– 每一个lun需要关联到某一个后端存储设备,在客户端

会视为一块虚拟硬盘

冰箱---->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

## 步骤

### 一 修改防火墙默认区域

虚拟机server

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

虚拟机desktop

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

### 二 划分一个5G的主分区

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

[root@server0 ~]# lsblk

虚拟机server服务端:

#### 1.安装软件包targetcli

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

#### 2.运行targetcli命令进行配置

[root@server0 ~]# targetcli

/> ls #查看相关配置

##### 1)建立后端存储backstore,后端存储

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

/> ls

##### 2)建立target磁盘组

/> iscsi/ create iqn.2019-07.com.example:server

/> ls

##### 3)进行lun关联

/> iscsi/iqn.2019-07.com.example:server/tpg1/luns create /backstores/block/nsd

/> ls

/> exit #退出

##### 4)设置访问控制,客户端声称的名字,符合iqn名称规范

[root@server0 ~]# targetcli

/> ls

/> iscsi/iqn.2019-07.com.example:server/tpg1/acls create iqn.2019-07.com.example:desktop

/> ls

##### 5)设置监听的IP地址

/> iscsi/iqn.2019-07.com.example:server/tpg1/portals create ip\_address=172.25.0.11

/> ls

/> exit #退出

### 三 重启服务(重启程序)

[root@server0 ~]# systemctl restart target

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

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

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

虚拟机desktop

#### 1.安装软件包(客户端软件)

• 所需软件包:iscsi-initiator-utils

Yum仓库可以补全软件名:

前提1:当前系统没有安装此软件

前提2:必须具备Yum缓存 (yum repolist 可以直接生成缓存)

[root@desktop0 ~]# yum repolist

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

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

[root@desktop0 ~]# yum -y install iscsi(tab)

#### 2.设置客户端生成的名字

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

InitiatorName=iqn.2019-07.com.example:desktop

#### 3.客户端刷新声称名字的服务

[root@desktop0 ~]# systemctl restart iscsid

Warning: Unit file of iscsid.service changed on

disk, 'systemctl daemon-reload' recommended.

[root@desktop0 ~]# systemctl daemon-reload

[root@desktop0 ~]# systemctl restart iscsid

#### 4.发现共享存储

Linux查看命令帮助的方法 : man iscsiadm

全文查找/example 按n项下跳转匹配,看到EXAMPLES

Ctrl - ：减小字体

Ctrl Shift + ：变大字体

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

##### 方法二:发现iSCSI磁盘

– iscsiadm -m discovery -t st -p 服务器地址[:端口]

#### 5.加载共享存储

[root@desktop0 ~]# systemctl restart iscsi

[root@desktop0 ~]# systemctl enable iscsi

[root@desktop0 ~]# lsblk

## 数据库服务基础

什么是数据库: 存放数据的仓库

数据库中有众多库,在每一个库中有众多的表格

• 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

database:数据库

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

虚拟机server:部署mariadb数据库服务器

• RHEL7 中的 MariaDB相关包

– mariadb-server :提供服务端有关的系统程序

[root@server0 ~]# systemctl restart mariadb

[root@server0 ~]# systemctl enable mariadb

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

[root@server0 ~]#

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

### 数据库的基本操作:

[root@server0 ~]# mysql #进入数据库

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

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

MariaDB [(none)]> show databases;

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

MariaDB [(none)]> show databases;

MariaDB [(none)]> create database nsd1906;

MariaDB [(none)]> show databases;

MariaDB [(none)]> use mysql; #切换到mysql库

MariaDB [mysql]> use nsd1906;

MariaDB [nsd1906]> use mysql;

MariaDB [mysql]> show tables; #查看当前库中有哪些表格

MariaDB [mysql]> exit #退出整个数据库

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

### 在Linux命令行,为数据库管理员设置密码

数据库管理员为root,对数据库有最高权限用户,存放在mysql库user表

Linux系统管理员为root,对系统有最高权限用户,存放/etc/passwd

#### • 为数据库账号修改密码

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

]# mysqladmin -u root password '123' #首次设置密码

]# mysql -u root -p #交互式进入

Enter password:

]# mysql -u root -p123 #非交互式进入

#### 已知旧密码修改新密码

]# mysqladmin -u root -p123 password '456'

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

##### MariaDB主配置文件:

[root@server0 ~]# ls /etc/my.cnf

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

***表格 : 增(insert) 删(delete) 改(update) 查(select)***

表记录 表字段

编号 姓名 联系方式 居住地址

1 dc 138..... 东村

2 tc 138..... 西村

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

[root@server0 ~]# ls

]# mysql -u root -p123 nsd1906 < /root/users.sql

]# mysql -u root -p123 #进入到数据库

MariaDB [(none)]> use nsd1906; #切换到nsd1906库

MariaDB [nsd1906]> show tables; #显示所有表格

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

| Tables\_in\_nsd1906 |

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

| base |

| location |

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

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

### 查(select)

格式: select 表字段 from 库名.表名 where 条件 ;

]# mysql -u root -p123 #进入到数据库

> use nsd1906; #切换到nsd1906库

> show tables; #显示所有表格

> select \* from base; #查看表所有字段及所有记录内容

> select \* from location; #查看表所有字段及所有记录内容

> use mysql;

> select \* from nsd1906.base;

> use nsd1906;

> select id,name from base;

> select \* from base where name='tom';

> select \* from base where password='456';

> select \* from base where id='3';

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

查询表结构: desc 表名;

MariaDB [(none)]> use nsd1906;

MariaDB [nsd1906]> desc base;

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

| Field |

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

| id |

| name |

| password |

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

MariaDB [nsd1906]> use mysql;

MariaDB [mysql]> desc user;

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

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

### 数据库授权

用户授权设置

• MariaDB [(none)]> 交互指令

– GRANT 权限列表 ON 数据库名.表名 TO 用户名@

客户机地址 IDENTIFIED BY '密码';

– 除了 root 用户,此nsd1906数据库只能被用户 lisi 查询,

此用户的密码为123

> grant select on nsd1906.\* to lisi@localhost identified by '123';

当lisi从localhost本地登录,输入密码123,将会获得nsd1906库中所有表格的查询权限;

> select user,host,password from user;

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

案例4:使用数据库查询

2. 在系统 server0 上使用数据库 nsd1906,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?

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

> use nsd1906;

> select \* from base;

> select \* from base where password='solicitous';

> select name from base where password='solicitous';

2)有多少人的姓名是 Barbara 同时居住在 Sunnyvale?

> select \* from base,location

where base.name='Barbara' and

location.city='Sunnyvale' and base.id=location.id;

> select count(\*) from base,location

where base.name='Barbara' and

location.city='Sunnyvale' and base.id=location.id;

> insert base values('6','Barbara','xixi'); #插入记录

> select \* from base;

> insert location values('6','Sunnyvale'); #插入记录

> select \* from location;

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

1. 禁止空密码root用户访问 mariadb 数据库

> use mysql;

> select user,host,password from user;

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

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

> select user,host,password from user;

> flush privileges; #刷新user表记录

验证登录失败:

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

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

## NFS文件共享

环境准备:防火墙设置

虚拟机server

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

虚拟机desktop

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

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

配置NFS共享

### NFS共享概述

• Network File System,网络文件系统

– 用途:为客户机提供共享使用的文件夹

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

• 所需软件包:nfs-utils

• 系统服务:nfs-server

虚拟机server:构建NFS服务

### 1.安装nfs-utils软件包

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

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

### 2.配置共享

[root@server0 ~]# mkdir /nsd

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

[root@server0 ~]# ls /nsd/

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

– 文件夹路径 客户机地址(权限) 客户机地址(权限) .. ..

/nsd \*(ro)

/opt \*(ro)

### 3.重启服务(重启程序)

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

虚拟机desktop:访问共享文件夹

### 1.挂载共享目录

]# mkdir /mnt/a /mnt/b

]# showmount -e 172.25.0.11 #查看服务端共享内容

]# mount 172.25.0.11:/nsd /mnt/a/

]# df -h

]# ls /mnt/a

]# mount 172.25.0.11:/opt/ /mnt/b

]# ls /mnt/b

]# df -ah #显示当前全部的挂载

### 2.开机自动挂载/etc/fstab

\_netdev:声明网络设备

指明挂载本设备需要有网络参数再进行挂载

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

172.25.0.11:/nsd /mnt/a nfs defaults,\_netdev 0 0

172.25.0.11:/opt /mnt/b nfs defaults,\_netdev 0 0

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

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

[root@desktop0 ~]# df -ah

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -ah

NFS配置参数说明

|  |  |
| --- | --- |
| rw | 可读写的权限 |
| ro | 只读的权限 |
| no\_root\_squash | 登入NFS主机，使用该共享目录时相当于该目录的拥有者，如果是root的话，那么对于这个共享的目录来说，他就具有root的权限，这个参数**『极不安全』**，不建议使用 |
| root\_squash | 登入NFS主机，使用该共享目录时相当于该目录的拥有者。但是如果是以root身份使用这个共享目录的时候，那么这个使用者（root）的权限将被压缩成为匿名使用者，即通常他的UID与GID都会变成nobody那个身份 |
| all\_squash | 不论登入NFS的使用者身份为何，他的身份都会被压缩成为匿名使用者，通常也就是nobody |
| anonuid | 可以自行设定这个UID的值，这个UID必需要存在于你的/etc/passwd当中 |
| anongid | 同anonuid，但是变成groupID就是了 |
| sync | 资料同步写入到内存与硬盘当中 |
| async | 资料会先暂存于内存当中，而非直接写入硬盘 |
| insecure | 允许从这台机器过来的非授权访问 |

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

## HTTP服务基础

Web服务:提供网页内容

• 基于 B/S (Browser/Server)架构的网页服务

– 服务端提供网页

– 浏览器下载并显示网页

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

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

• 软件包:httpd(Apache组织) #专门实现Web服务的软件

虚拟机server:

### 1.安装httpd软件

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

### 2.重启服务

[root@server0 ~]# systemctl restart httpd

### 3.访问测试

[root@server0 ~]# firefox 172.25.0.11

### 4.书写页面文件

]# echo '<h1>NSD1906 Web' > /var/www/html/index.html

]# cat /var/www/html/index.html

]# firefox 172.25.0.11

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

### 主配置文件配置字段:

– Listen:监听地址:端口(80)

– ServerName:本站点注册的DNS名称(空缺)

– DocumentRoot:网页根目录(/var/www/html)

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

虚拟机classroom:

DNS服务器的解析:将域名解析为IP地址

server0.example.com----->172.25.0.11

www0.example.com----->172.25.0.11

webapp0.example.com----->172.25.0.11

[root@desktop0 ~]# nslookup webapp0.example.com

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

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

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

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

虚拟机server

]# mkdir /var/www/myweb

]# echo '<h1>wo shi myweb' > /var/www/myweb/index.html

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

]# vim /etc/httpd/conf/httpd.conf

DocumentRoot "/var/www/myweb"

]# systemctl restart httpd

虚拟机desktop:测试

[root@desktop0 ~]# firefox 172.25.0.11

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

网络路径与实际服务器路径 网页文件的根目录

客户端:firefox http://172.25.0.11:80---->服务端172.25.0.11

----->httpd 80----->DocumentRoot /var/www/myweb--->index.html

客户端: firefox 172.25.0.11

服务端: /var/www/myweb

客户端: firefox 172.25.0.11/abc

服务端: /var/www/myweb/abc/index.html

DocmentRoot /var/www/myweb

客户端: firefox 172.25.0.11/var/www/myweb/abc

服务端: /var/www/myweb/var/www/myweb/abc

虚拟机server:

]# mkdir /var/www/myweb/abc

]# echo '<h1>wo shi abc' > /var/www/myweb/abc/index.html

]# firefox 172.25.0.11/abc

vsftpd----->/var/ftp

httpd------>/var/www/html

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

## 虚拟Web主机

### • 虚拟Web主机

– 由同一台服务器提供多个不同的Web站点

#### • 区分方式

– 基于域名的虚拟主机

– 基于端口的虚拟主机

– 基于IP地址的虚拟主机

#### • 配置文件路径

– /etc/httpd/conf/httpd.conf #主配置文件

– /etc/httpd/conf.d/\*.conf #调用配置文件

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

#### • 为每个虚拟站点添加配置

<VirtualHost IP地址:端口>

ServerName 此站点的DNS名称

DocumentRoot 此站点的网页根目录

</VirtualHost>

虚拟机server: vim命令模式下 u 可以撤销

]# vim /etc/httpd/conf.d/nsd01.conf #建立调用配置文件

<VirtualHost \*:80> #启用虚拟Web主机功能

ServerName server0.example.com #指定网站DNS名称

DocumentRoot /var/www/qq #指定网页文件根目录

</VirtualHost>

<VirtualHost \*:80>

ServerName www0.example.com

DocumentRoot /var/www/baidu

</VirtualHost>

]# mkdir /var/www/qq /var/www/baidu

]# echo '<h1>QQ 企鹅' > /var/www/qq/index.html

]# echo '<h1>baidu 百度' > /var/www/baidu/index.html

]# systemctl restart httpd

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

虚拟Web主机优先级,由上到下依次匹配,匹配即停止

***一旦使用虚拟Web主机功能,所有的网站都必须利用虚拟Web主机实现***

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

<VirtualHost \*:80>

ServerName www0.example.com

DocumentRoot /var/www/baidu

</VirtualHost>

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /var/www/qq

</VirtualHost>

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/myweb

</VirtualHost>

[root@server0 ~]# systemctl restart httpd

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

### Web访问控制,针对存放网页文件的目录

子目录默认继承父目录的Web访问控制

<Directory />

Require all denied #拒绝所有人访问

</Directory>

<Directory "/var/www">

Require all granted #允许所有人访问

</Directory>

案例:自定义Web的网页文件根目录

1.实现server0.example.com访问网页文件根目录修改为/webroot

]# vim /etc/httpd/conf.d/nsd01.conf

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /webroot

</VirtualHost>

]# mkdir /webroot

]# echo '<h1>wo shi webroot' > /webroot/index.html

2.修改访问控制

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

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

<Directory "/webroot"> #针对/webroot路径

Require all granted #允许所有人访问

</Directory>

[root@server0 ~]# systemctl restart httpd

3.SELinux策略

[root@server0 ~]# setenforce 0

[root@server0 ~]# getenforce

Permissive

[root@server0 ~]# vim /etc/selinux/config

SELINUX=permissive

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

环境准备:防火墙设置

虚拟机server

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

虚拟机desktop

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

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

案例:为虚拟机server配置以下虚拟Web主机

- 实现三个网站的部署

- 实现客户端访问server0.example.com网页内容为 卖女孩的小火柴

- 实现server0.example.com网页根目录为/var/www/abc01

- 实现客户端访问www0.example.com网页内容为 奔跑吧骆驼

- 实现www0.example.com网页根目录为/var/www/abc02

- 实现客户端访问webapp0.example.com网页内容为 20里春风不如你

- 实现webapp0.example.com网页根目录为/var/www/abc03

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

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

### 客户机地址限制

• 使用 <Directory> 配置区段

– 每个文件夹自动继承其父目录的ACL访问权限

– 除非针对子目录有明确设置

<Directory 目录的绝对路径>

.. ..

Require all denied|granted

Require ip IP或网段地址 .. ..

</Directory>

案例1: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,要求如下:

1.创建目录

]# cat /etc/httpd/conf.d/nsd01.conf #查看DocumentRoot

]# mkdir /var/www/abc01/private

2.从 http://classroom.example.com/pub/materials/private.html下载一个文件副本到这个目录,重命名为 index.html

不要对文件 index.html 的内容作任何修改

]# cd /var/www/abc01/private

]# wget http://classroom.example.com/pub/materials/private.html

]# mv private.html index.html

]# firefox server0.example.com/private

3)从虚拟机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

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

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

<Directory "/var/www/abc01/private">

Require ip 172.25.0.11 #仅允许172.25.0.11访问

</Directory>

[root@server0 /]# systemctl restart httpd

虚拟机desktop:

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

Forbidden

You don't have permission to access /private on this server.

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

案例2: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

]# mkdir /webroot

]# echo '<h1> wo shi webroot' > /webroot/index.html

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

<VirtUalHost \*:80>

ServerName server0.example.com

DocumentRoot /webroot

</VirtUalHost>

2)修改访问控制

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

<Directory "/webroot">

Require all granted #允许所有人可以访问

</Directory>

[root@server0 /]# systemctl restart httpd

3)修改SELinux策略:安全上下文值(打标签) 布尔值策略 非默认端口开放

神州旅游: 红色帽子

达内旅游: 绿色帽子

]# semanage --help

]# semanage fcontext -l | less #查看所有上下文值

]# ls -Zd /var/www/ #专查看目录上下文值

]# ls -Zd /webroot/

• 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

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

]# ls -Zd /webroot/

测试:

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

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

## 部署动态网站

### 静态网站的运行

• 服务端的原始网页 = 浏览器访问到的网页

– 由Web服务软件处理所有请求

– 文本(txt/html)、图片(jpg/png)等静态资源

### 动态网站的运行

• 服务端的原始网页 ≠ 浏览器访问到的网页

– 由Web服务软件接受请求,动态程序转后端模块处理

– PHP网页、Python网页、JSP网页......

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

为站点 webapp0.example.com 配置提供动态Web内容,要求如下:

#### 1.部署Python页面文件

]# cat /etc/httpd/conf.d/nsd01.conf #查看DocumentRoot

]# cd /var/www/abc03

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

]# ls

#### 2.页面跳转(页面别名 地址重写)方便用户的访问

当用户访问 webapp0.example.com将webinfo.wsgi页面进行呈现

格式: Alias 网络路径 实际路径

Alias / /var/www/abc03/webinfo.wsgi

当用户访问的时网页文件根目录时,跳转到/var/www/abc03/webinfo.wsgi

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

<VirtUalHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/abc03

Alias / /var/www/abc03/webinfo.wsgi

当用户访问的是网页文件根目录时,跳转到/var/www/abc03/webinfo.wsgi

</VirtUalHost>

[root@server0 /]# systemctl restart httpd

[root@desktop0 ~]# firefox webapp0.example.com

#### 3.翻译Python页面代码

[root@server0 /]# yum -y install mod\_wsgi

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

<VirtUalHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/abc03

WsgiScriptAlias / /var/www/abc03/webinfo.wsgi

</VirtUalHost>

[root@server0 /]# systemctl restart httpd

页面内容显示:

UNIX时间戳:从1970-1-1 0:0:0到达现在时间,所经历的秒数

[root@desktop0 ~]# firefox webapp0.example.com

#### 4.此虚拟Web主机侦听在端口8909

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

Listen 8909 #设置httpd程序监听8909端口

<VirtUalHost \*:8909> #设置虚拟Web主机监听8909端口

ServerName webapp0.example.com

DocumentRoot /var/www/abc03

WsgiScriptAlias / /var/www/abc03/webinfo.wsgi

</VirtUalHost>

]# semanage port -l | grep http #查看关于http开放的端口

]# semanage port -a -t http\_port\_t -p tcp 8909

此命令会占用较多内存, -a 添加 -t 类型 -p 协议

]# systemctl restart httpd

测试验证: firefox webapp0.example.com:8909

#### 总结:访问时端口优先级最高

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

## 安全Web服务

### PKI公钥基础设施

• 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、......)

– Certificate Authority,数字证书授权中心:负责证书

的申请/审核/颁发/鉴定/撤销等管理工作

### **数据证书授权中心**

:虚拟机classroom

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

虚拟机server:

### 为站点 www0.example.com 配置安全加密的Web内容

#### 1.部署网站证书(营业执照)

]# cd /etc/pki/tls/certs/

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

]# ls

#### 2.部署网站的根证书(工商局的信息)

]# cd /etc/pki/tls/certs/

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

]# ls

#### 3.部署解密数据的私钥

]# cd /etc/pki/tls/private/

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

]# ls

#### 4.安装可以进行加密支持的软件mod\_ssl

[root@server0 /]# yum -y install mod\_ssl

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

补充:vim末行模式 输入set nu 可以开启行号

59 DocumentRoot "/var/www/abc02"

60 ServerName www0.example.com:443

指定网站证书

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

指定解密的私钥

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

指定网站的根证书

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

[root@server0 /]# systemctl restart httpd

[root@server0 /]# firefox https://www0.example.com

我已了解可能的风险--->添加例外---->确认安全例外

## Samba

补充:Samba共享服务(Windows与Linux跨平台的共享)

smb(用户验证通信协议) cifs(Samba独有文件系统)

程序名(服务名):smb

虚拟机server:

### 服务端

#### 1.安装软件包

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

#### 2.创建Samba共享帐号

1.建立在本地用户的基础上

2.采用独立的密码,与用户登录系统的密码不同

[root@server0 /]# useradd harry

[root@server0 /]# pdbedit -a harry #添加为samba帐号

new password:

retype new password:

[root@server0 /]# pdbedit -L #查看所有的samba帐号

[root@server0 /]# pdbedit -x harry #删除samba帐号harry

[root@server0 /]# pdbedit -a harry

#### 3.创建共享目录及配置共享

[root@server0 /]# mkdir /public

[root@server0 /]# echo haha > /public/1.txt

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

命令模式: G(大写) 到全文的末行

[nsd] #设置共享名

path = /public #设置实际共享的路径

[root@server0 /]# systemctl restart smb

#### 4.修改SELinux策略: 布尔值策略(服务功能的开关)

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

[root@server0 /]# setsebool samba\_export\_all\_ro on

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

### **客户端**:

虚拟机desktop0

#### 1.安装软件包,支持cifs文件系统

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

#### 2.进行挂载访问

]# mkdir /mnt/smb

格式:mount -o user=用户名,pass=密码 //服务器IP地址/共享名 挂载点目录

]# mount -o user=harry,pass=123 //172.25.0.11/nsd /mnt/smb/

]# df -h

]# ls /mnt/smb/

#### 3.开机自动挂载/etc/fstab

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

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

[root@desktop0 ~]# df -h

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h

## 设置网络参数

### 一 设置永久的主机名,配置文件为/etc/hostname

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

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

nsd.tedu.cn

[root@server0 ~]# hostname

nsd.tedu.cn

[root@server0 ~]# exit

登出

Connection to 172.25.0.11 closed.

[student@room9pc01 ~]$ gos

Last login: Sat Jul 13 08:35:21 2019 from 172.25.0.250

[root@nsd ~]#

### 二 永久设置设置IP地址 子网掩码 网关地址

#### 1.查看识别的网卡名

[root@nsd ~]# nmcli connection show

名称

System eth0

#### 2.设置永久的IP地址 子网掩码 网关地址

]# nmcli connection modify 'System eth0'

ipv4.method manual

ipv4.addresses '172.25.0.110/24 172.25.0.254'

connection.autoconnect yes

]# nmcli connection 修改 '网卡名'

ipv4.方法 手工设置

ipv4.地址 'IP地址/子网掩码 网关地址'

每次开机自动启用配置

#### 3.激活配置

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

[root@nsd ~]# ifconfig | head -2

[root@nsd ~]# route #查看网关地址

Gateway

172.25.0.254

### 三 DNS服务器的地址

永久配置文件:/etc/resolv.conf

]# echo 'nameserver 172.25.254.254' > /etc/resolv.conf

]# cat /etc/resolv.conf

nameserver 172.25.254.254

]# nslookup server0.example.com

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

### IPv6地址的组成

• IPv4 地址表示

– 32个二进制位,点分隔的十进制数

– 例如:172.25.0.11、127.0.0.1

• IPv6 地址表示

– 128个二进制位,冒号分隔的十六进制数

– 每段内连续的前置 0 可省略、连续的多个 : 可简化为 ::

– 例如:

2003:ac18:0000:0000:0000:0000:0000:0305

2003:ac18::305

[root@nsd ~]# nmcli connection modify 'System eth0' ipv6.method manual ipv6.addresses 2003:ac18::305/64 connection.autoconnect yes

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

successfully(成功)

[root@nsd ~]# ifconfig | head -4

[root@nsd ~]# ping6 2003:ac18::305

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

### 聚合连接(链路聚合 网卡绑定)

作用:实现网卡的高可用

eth1 eth2

虚拟网卡:team0 192.168.1.1

#### 1.生成虚拟的网卡team0 参考:man teamd.conf

全文搜索/example 按 n跳转匹配

]# nmcli connection add type team

con-name team0 ifname team0 autoconnect yes

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

]# nmcli connection 添加 类型 组队

配置文件名 team0 网卡名 team0 开机自动启用

配置运行方式为 活跃备份方式

]# ifconfig team0

#### 2.添加成员

]# nmcli connection add type team-slave

con-name team0-1 ifname eth1 master team0

]# nmcli connection add type team-slave

con-name team0-2 ifname eth2 master team0

]# nmcli connection 添加 类型 team-成员

配置文件名 team0-2 网卡名为 eth2 主设备 team0

#### 3.team0配置IP地址

]# nmcli connection modify team0

ipv4.method manual ipv4.addresses 192.168.1.1/24

connection.autoconnect yes

]# nmcli connection up team0

]# teamdctl team0 state #专用于查看team0信息

]# ifconfig eth2 down

]# teamdctl team0 state

如果有敲错,删除所有重新开始

]# nmcli connection delete team0 #删除配置文件

]# nmcli connection delete team0-1 #删除配置文件

]# nmcli connection delete team0-2 #删除配置文件

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

## 基础邮件服务

前提:修改IP地址,符合DNS解析结果

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

ipv4.method manual

ipv4.addresses '172.25.0.11/24 172.25.0.254'

connection.autoconnect yes

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

### 查看DNS服务器地址:

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

nameserver 172.25.254.254

[root@nsd ~]# nslookup server0.example.com #测试解析

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

### 电子邮件通信

• 电子邮件服务器的基本功能

– 为用户提供电子邮箱存储空间(用户名@邮件域名)

– 处理用户发出的邮件 —— 传递给收件服务器

– 处理用户收到的邮件 —— 投递到邮箱

DNS服务器:虚拟机classroom

yg@qq.com

yg@server0.example.com

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

虚拟机server:构建邮件服务器

#### 1.安装postfix软件,实现邮件服务

[root@nsd ~]# rpm -q postfix

postfix-2.10.1-6.el7.x86\_64

#### 2.修改配置文件

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

vim末行模式: set nu 开启行号

99 myorigin = server0.example.com #默认补全的域名后缀

116 inet\_interfaces = all #在本机所有网卡启用邮件功能

164 mydestination = server0.example.com

#判断为本域邮件的依据

#### 3.重启邮件服务

[root@nsd ~]# systemctl restart postfix

#### 使用mail命令发信/收信

• mail 发信操作

– mail -s '邮件标题' -r 发件人 收件人

[root@nsd ~]# useradd yg

[root@nsd ~]# useradd xln

[root@nsd ~]# mail -s 'test01' -r yg xln

hahaxixihehelele

.

EOT

[root@nsd ~]# mail -u xln #查看xln的邮件

>N 1 yg@server0.example.c Sat Jul 13 14:2118/569

& 1 #输入邮件编号1查看邮件内容

[root@nsd ~]# echo '123' | mail -s 'abc' -r yg xln

[root@nsd ~]# mail -u xln

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

## parted分区工具,进行GPT分区模式

MBR分区模式: fdisk分区工具

最多有4个主分区,划分3个主分区,1个扩展分区,n个逻辑分区

最大的空间支持:2.2TB

GPT分区模式:parted分区工具

最多有128个主分区

最大的空间支持:18EB

1EB=1024PB

1PB=1024TB

[root@nsd ~]# lsblk

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

(parted) mktable gpt #指定分区模式gpt

(parted) print #输出分区表信息

(parted) mkpart #划分新的分区

分区名称？ []? nsd #随意起名字

文件系统类型？ [ext2]? xfs #随意写不起决定性作用

起始点？ 0

结束点？ 2G

忽略/Ignore/放弃/Cancel? Ignore #输入 i 忽略

(parted) unit GB #使用GB作为单位显示

(parted) print

(parted) mkpart

分区名称？ []? nsd

文件系统类型？ [ext2]? xfs

起始点？ 2G

结束点？ 4G

(parted) print