# NSD Engineer全部笔记

## Day01

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

– 解释型程序

– 顺序、批量执行

案例1:Shell脚本的编写及测试

1. 编写一个面世问候 /root/helloworld.sh 脚本

– 显示出一段话 Hello World

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

echo Hello World

[root@server0 ~]# chmod +x /root/hello.sh #赋予执行权限

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

规范Shell脚本的一般组成

• #! 环境声明,以下代码由那个程序进行解释

• # 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

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

简单脚本技巧

重定向输出

>:只收集前面命令的正确输出信息

2>:只收集前面命令的错误输出信息

&>:收集前面命令的正确与错误信息

[root@server0 ~]# cat /etc/redhat-release

[root@server0 ~]# cat /xixi

[root@server0 ~]# cat /etc/redhat-release /xixi

]# cat /etc/redhat-release /xixi > /opt/1.txt

]# cat /opt/1.txt

]# cat /etc/redhat-release /xixi 2> /opt/2.txt

]# cat /opt/2.txt

]# cat /etc/redhat-release /xixi &> /opt/3.txt

]# cat /opt/3.txt

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

案例:书写一个创建用户设置密码的脚本

/dev/null:黑洞设备

[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

为了让脚本,适应多变需求,所以使用变量

变量:会变化的量 类似容器

以不变的名称,存储可以变化的值

定义变量:

变量名=存储的值

使用变量储存的值: $变量名

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

#!/bin/bash

a=dc

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

提高用户体验度,产生交互

read

1.直接产生交互,等待用户输入

2.记录用户在键盘上的输入

3.将记录的内容,赋值给一个变量储存

-p '屏幕提示信息'

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

#!/bin/bash

read -p '请您输入要创建的用户名:' a

read -p '请您输入要设置的密码:' b

useradd $a &> /dev/null

echo 用户$a创建成功

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

echo 用户$a密码设置成功

[root@server0 ~]#

什么是变量

• 以不变的名称存放的可能会变化的值

– 变量名=变量值

– 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

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

以下有效变量名为: B

A.2008file B.tun\_01 C a-b D a\_ccd\_9-abc

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 $a7

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

rhel7

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

单引号' ':取消所有特殊字符的意义,原样输出

[root@server0 ~]# echo '\*\*\*\*\*\*\*\*'

\*\*\*\*\*\*\*\*

[root@server0 ~]# echo '| ? \* {} [] > '

| ? \* {} [] >

[root@server0 ~]#

反撇号 ` ` 或 $( )

:将命令的输出,直接参与另一条命令的运行,作为另一条命令的参数

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

[root@server0 ~]# ls /opt/

[root@server0 ~]# mkdir /opt/`hostname`-`date +%F`

[root@server0 ~]# ls /opt/

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

[root@server0 ~]# ls /opt/

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

环境变量

变量名为大写字母,由系统定义并且赋值完成

USER=当前登录的用户名

[root@server0 ~]# echo $USER

root

[root@server0 ~]# su - student

[student@server0 ~]$ echo $USER

student

[student@server0 ~]$ exit

logout

[root@server0 ~]# su - nsd01

[nsd01@server0 ~]$ echo $USER

nsd01

[nsd01@server0 ~]$

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

位置变量

• 由系统定义并且赋值完成

• 在执行脚本时提供的命令行参数

– 表示为 $n,n为序号

– $1、$2、.. .. ${10}、${11}、.. ..

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

#!/bin/bash

#read -p '请您输入要创建的用户名:' a

#read -p '请您输入要设置的密码:' b

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 natasha

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

#!/bin/bash

cat -n $1 | head -$2

echo 您输入的文件为$1

echo 显示文件内容的头$2行

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

$? 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 $#

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

[root@server0 ~]# /root/2.sh haha xixi hehe lele

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

条件测试

[ 测试表达式 ] 每一部分都要有空格

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,且必须为目录才为真

-f:文档存在,且必须为文件才为真

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

[root@server0 ~]# echo $?

0

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

[root@server0 ~]# echo $?

1

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

[root@server0 ~]# echo $?

1

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

[root@server0 ~]# echo $?

0

[root@server0 ~]#

• 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 2 ]

[root@server0 ~]# echo $?

1

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

[root@server0 ~]# echo $?

0

[root@server0 ~]# [ 10 -gt 9 ]

[root@server0 ~]# echo $?

0

[root@server0 ~]# [ 9 -ge 9 ]

[root@server0 ~]# echo $?

• 字符串比对

==:一致为真

!=:不一致为真

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

[root@server0 ~]# echo $?

1

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

[root@server0 ~]# echo $?

1

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

[root@server0 ~]# echo $?

0

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

[root@server0 ~]# echo $?

0

[root@server0 ~]#

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

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

案例:利用read获取,用户所输入的用户名

进行判断,如果用户名当前系统已存在,则输出 用户已存在

进行判断,如果用户名当前系统不存在,则输出 用户不存在

[root@server0 ~]# vim /root/if02.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 $a用户已存在

else

echo $a用户不存在

fi

[root@server0 ~]#

案例:利用read获取,用户所输入的IP地址

进行判断,如果当前系统可以与该IP地址通信,则输出 可以ping通

进行判断,如果当前系统不可以与该IP地址通信,则输出 不可以ping通

[root@server0 ~]# vim /root/if03.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不可以通信

fi

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

if多分支处理

if [条件测试1];then

命令序列xx

elif [条件测试2];then

命令序列yy

elif [条件测试3];then

命令序列aa

..........

else

命令序列zz

fi

案例:利用read获取,用户所输入的成绩

进行判断

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 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,反复执行类似的操作

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

for 献血车 in zhangsan lisi wangwu

do

抽血

done

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

#!/bin/bash

for a in zhangsan lisi wangwu natasha harry jack

do

useradd $a &> /dev/null

echo $a创建成功

done

[root@server0 ~]#

列表值可以不参与循环执行的操作

造数工具: {起始..结束}

{1..20}:制造1到20之间连续的数字

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

#!/bin/bash

for a in {1..20}

do

echo Hello World

done

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

案例4:编写一个判断脚本

在 server0 上创建 /root/foo.sh 脚本

1)当运行/root/foo.sh redhat,输出为fedora

2)当运行/root/foo.sh fedora,输出为redhat

3)当没有任何参数或者参数不是 redhat 或者fedora时,

其错误输出产生以下信息:

/root/foo.sh redhat|fedora

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

#!/bin/bash

if [ $# -eq 0 ];then #判断用户是否输入参数

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

exit 38 #脚本退出返回值

elif [ $1 == redhat ];then #判断参数是否是redhat

echo fedora

elif [ $1 == fedora ];then #判断参数是否是fedora

echo redhat

else

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

exit 48 #脚本退出返回值

fi

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

## Day02

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.example.com/pub/materials/userlist

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

#!/bin/bash

if [ $# -eq 0 ];then #判断用户是否输入参数

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

exit 2

elif [ -f $1 ];then　　　　　　　　＃判断用户输入的文件是否存在

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)

所有状态变成disabled(彻底禁用),都要经历重启系统

• 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server

[root@server0 ~]# getenforce #查看SELinux当前系统状态

Enforcing

[root@server0 ~]# setenforce 0 #修改SELinux状态,为宽松模式

[root@server0 ~]# getenforce

Permissive

[root@server0 ~]# vim /etc/selinux/config#SELinux状态永久配置文件

SELINUX=permissive

虚拟机desktop

[root@desktop0 ~]# getenforce

Enforcing

[root@desktop0 ~]# setenforce 0

[root@desktop0 ~]# getenforce

Permissive

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

SELINUX=permissive

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

用户个性化配置文件

• 影响指定用户的 bash 解释环境

– ~/.bashrc,每次开启 bash 终端时生效

全局环境配置

• 影响所有用户的 bash 解释环境

– /etc/bashrc,每次开启 bash 终端时生效

虚拟机server0

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

alias hello='echo hello'

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

alias hi='echo hi'

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

alias dc='echo dc'

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

搭建基本Web服务,提供一个网页内容

http协议:超文本传输协议,专用于传输页面内容

服务端:虚拟机server

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'

3.本机测试

[root@server0 ~]# firefox 172.25.0.11

4.书写自己的页面

默认存放文件文件路径:/var/www/html/

默认网页文件名称:index.html

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

<marquee><font color=red><h1>NSD1904hahaxixi

滚动 字体颜色 标题字体

[root@server0 ~]#

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

搭建基本的FTP服务

FTP:文件传输协议

默认FTP服务共享位置:/var/ftp/

虚拟机server

1.安装vsftpd软件

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

2.重启vsftpd服务

[root@server0 ~]# systemctl restart vsftpd

[root@server0 ~]# systemctl enable vsftpd

3.本机验证

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

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

防火墙策略管理

作用:隔离,进行入站过滤

硬件防火墙

软件防火墙

firewalld服务基础

Linux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

– trusted:允许任何访问

– block:拒绝任何来访请求(明确拒绝回应)

– drop:丢弃任何来访的数据包(直接丢弃不给回应,节省服务器资源)

数据包: 源IP地址 目标IP地址 数据

防火墙的判定规则

1.查看请求数据包,中客户端来源IP地址,查看自己所有区域规则,如果某一个区域有该源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

虚拟机server

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

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

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 #修改默认区域为drop

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

虚拟机desktop

]# ping 172.25.0.11 #不可以通信,没有回应

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

常见的互联网协议:

http:超文本传输协议 默认端口:80

FTP:文本传输协议 默认端口:21

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

telnet:远程管理协议 默认端口:23

tftp:简单的文本传输协议 默认端口:69

smtp:发邮件协议 默认端口:25

pop3:收邮件协议 默认端口:110

snmp:简单的网络管理协议 默认端口:161

在默认区域添加协议:

虚拟机server:

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

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

]# firewall-cmd --zone=public --list-all #查看区域详细规则

虚拟机desktop

]# firefox 172.25.0.11 #失败

]# firefox ftp://172.25.0.11 #失败

虚拟机server:

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

]# firewall-cmd --zone=public --add-service=http #添加允许的协议

]# firewall-cmd --zone=public --add-service=ftp #添加允许的协议

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

虚拟机desktop

]# firefox 172.25.0.11 #成功

]# firefox ftp://172.25.0.11 #成功

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

防火墙永久策略

– 永久(permanent)

]# firewall-cmd --reload #重新加载防火墙所有配置

]# 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 #重新加载防火墙所有配置

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

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

单独拒绝虚拟机desktop的访问

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

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

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

端口:编号 标识协议或服务 可以具备多个

由root指定

数据包: 源IP地址 目标IP地址 端口号 数据

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)

– 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://172.25.0.11:5423 ---> http://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

]# firefox 172.25.0.11:5423

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

## Day03

一 环境设置

虚拟机server

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

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

虚拟机desktop

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

[root@desktop0 ~]# firewall-cmd --get-default-zone

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

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

一 划分一个主分区3G

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

[root@server0 ~]# lsblk

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

二 服务端选择虚拟机server,构建iSCSI共享存储

• 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

• ISCSI Qualified Name(iqn) 名称规范

– iqn.yyyy-mm.倒序域名:自定义标识

– 用来识别 target 磁盘组,也用来识别客户机身份

• 名称示例

– iqn.2016-02.com.example:server0

• 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------->纸质的包装 木质的箱子

1.安装targetcli软件包

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

2.配置

a)建立backstore,后端存储(冰箱)

[root@server0 ~]# targetcli

/> ls

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

/> ls

b)建立target磁盘组(建立木质的箱子)

iqn.yyyy-mm.倒序域名:自定义标识

/> iscsi/ create iqn.2019-05.com.example:server0

/> ls

c)进行lun关联(冰箱----->木质的箱子)

/> iscsi/iqn.2019-05.com.example:server0/tpg1/luns create /backstores/block/nsd

/> ls

d)设置访问控制ACL,设置客户端声称的名字(符合iqn命名规范)

/> iscsi/iqn.2019-05.com.example:server0/tpg1/acls create

iqn.2019-05.com.example:abc

/> ls

e)指定本机开放共享服务的IP地址及端口

/> iscsi/iqn.2019-05.com.example:server0/tpg1/portals create 172.25.0.11

/> ls

[root@server0 ~]# systemctl restart target #重启服务

[root@server0 ~]# systemctl enable target #设置开机自动

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

[root@server0 ~]#

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

虚拟机desktop:客户端

1.安装客户端软件

• 所需软件包:iscsi-initiator-utils

Yum安装软件时可以补全包名:

前提1:必须有Yum缓存, yum repolist 可以直接生成缓存

前提2:必须是当前系统没有安装

[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.配置客户端声称的名字

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

InitiatorName=iqn.2019-05.com.example:abc

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.发现服务端共享存储

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

172.25.0.11:3260,1 iqn.2019-05.com.example:server0

5.使用服务端共享存储

[root@desktop0 ~]# lsblk

[root@desktop0 ~]# systemctl restart iscsi #重启客户端服务

[root@desktop0 ~]# lsblk

[root@desktop0 ~]# systemctl enable iscsi #设置开机自启动服务

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

总结:

服务端:

1.修改防火墙默认区域trusted

2.安装targetcli软件

3.运行targetcli命令进行配置

1)建立后端存储

2)建立target磁盘

3)进行lun关联

4)设置ACL访问控制

5)设置启用IP地址 默认端口为3260

4.重启target服务

客户端:

1.修改防火墙默认区域trusted

2.所需软件包:iscsi-initiator-utils

3.修改配置文件/etc/iscsi/initiatorname.iscsi

4.客户端刷新声称名字的服务 systemctl restart iscsid

5.发现共享存储

]# iscsiadm --mode discoverydb --type sendtargets --portal 172.25.0.11 --discover

6.使用共享存储

]# systemctl enable iscsi

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

数据库服务基础

什么是数据库:存放数据的仓库

– 社区开源版 MariaDB

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 ~]# yum -y install mariadb-server

[root@server0 ~]# systemctl restart mariadb

在数据库中,有众多的库,在每一个库中有很多的表格

一 数据库的基本操作

[root@server0 ~]# mysql

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

MariaDB [(none)]> create database nsd01; #创建库nsd01

MariaDB [(none)]> show databases;

MariaDB [(none)]> drop database nsd01; #删除nsd01库

MariaDB [(none)]> show databases;

MariaDB [(none)]> create database nsd1904;

MariaDB [(none)]> show databases;

MariaDB [(none)]> exit

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

二 为数据库管理员设置密码

数据库管理员root,针对数据库具有最高权限,信息存放mysql库中user表

系统管理员root,针对系统具有最高权限,信息存放/etc/passwd

命令格式: 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

Enter password:

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

已知旧密码修改新密码

[root@server0 ~]# mysqladmin -u root -p123 password '321'

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

数据库的主配置文件:/etc/my.cnf

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

• 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

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

]# ls

]# mysql -u root -p123 nsd1904 < /root/users.sql

• 查看库中有哪些表格

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

MariaDB [(none)]> use nsd1904; #切换到nsd1904库

MariaDB [nsd1904]> show tables; #显示当前库有那些表格

表格: 表字段 表记录

编号 姓名 联系方式 居住地址

1 dc 138..... 东村

2 tc 138..... 西村

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

查(select)

格式: select 表字段,表字段..... from 库名.表名称

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

MariaDB [(none)]> use nsd1904; #进入nsd1904库

MariaDB [(none)]> show tables; #显示库中所有表格

MariaDB [(none)]> select \* from base;

MariaDB [(none)]> select \* from location;

MariaDB [(none)]> select name,password from base;

MariaDB [(none)]> use mysql;

MariaDB [(none)]> select \* from nsd1904.base;

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

查看表结构: desc 表名字;

MariaDB [(none)]> desc nsd1904.base;

MariaDB [(none)]> desc mysql.user;

MariaDB [nsd1904]> select user,host,password from mysql.user;

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

条件查询:

MariaDB [nsd1904]> use nsd1904;

MariaDB [nsd1904]> select \* from base;

MariaDB [nsd1904]> select \* from base where password='123';

MariaDB [nsd1904]> select \* from base where password='456';

MariaDB [nsd1904]> select \* from base where name='tom';

MariaDB [nsd1904]> select \* from base where id='4';

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

除了root用户,此nsd1904库只能被用户lisi查询,此用户的密码为123

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';

grant select on nsd1904.\* to lisi@localhost identified by '123';

当lisi通过本地localhost登录,输入密码为123,将会获得nsd1904库中所有表格的查询权限.

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

MariaDB [(none)]> grant select on nsd1904.\* to lisi@localhost identified by '123';

MariaDB [(none)]> select user,host,password from mysql.user;

MariaDB [(none)]> exit

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

MariaDB [(none)]> show databases;

MariaDB [(none)]> exit

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

案例4:使用数据库查询

2. 在系统 server0 上使用数据库 nsd1904,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 nsd1904;

> select \* from base;

> select \* 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','789');

> select \* from base;

> insert location values ('6','Sunnyvale');

> 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='';

#删除表中password值为空的记录

> delete from user where password='';

> select user,host,password from user;

> flush privileges; #刷新user表记录

> exit

验证登录失败:

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

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

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

## Day04

配置NFS共享

• 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

一 环境设置

虚拟机server

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

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

虚拟机desktop

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

[root@desktop0 ~]# firewall-cmd --get-default-zone

二 虚拟机server构建NFS共享

1.安装nfs-utils软件包

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

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

[root@server0 ~]#

2.修改配置文件

[root@server0 ~]# mkdir /nsd01 #创建共享目录

[root@server0 ~]# echo haha > /nsd01/a.txt #建立测试文件

[root@server0 ~]# ls /nsd01

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

文件夹路径 客户机地址(权限) 客户机地址(权限) .. ..

/nsd01 \*(ro) #将/nsd01目录共享给所有客户端,可以只读方式访问

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

[root@server0 ~]# systemctl enable nfs-server #设置开机自启动

虚拟机desktop

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

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

[root@desktop0 ~]# showmount -e 172.25.0.11 #显示服务端共享路径

[root@desktop0 ~]# mount 172.25.0.11:/nsd01 /mnt/test/

[root@desktop0 ~]# df -h

开机自动挂载

\_netdev:声明网络设备,Linux先部署网络所有参数,在进行挂载该设备

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

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

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

[root@desktop0 ~]# df -h

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h

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

HTTP服务基础

Web通信基本概念

• 基于 B/S (Browser/Server)架构的网页服务

– 服务端提供网页

– 浏览器下载并显示网页

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

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

提供Web服务的软件

httpd(Apache)、nginx、tomcat

• 软件包:httpd

• 系统服务:httpd

• 提供的默认配置

– Listen:监听地址:端口(80)

– ServerName:本站点注册的DNS名称(空缺)

– DocumentRoot:网页根目录(/var/www/html)

虚拟机server:构建基本的Web服务

1.安装httpd

2.重启httpd服务

]# systemctl restart httpd

3.设置httpd服务开机自启

4.书写一个网站内容

]# echo '<h1>NSD1904 Web1' > /var/www/html/index.html

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

]# firefox 172.25.0.11

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

DocumentRoot:设置存放网页文件的路径(设置网页文件根目录)

虚拟机server:

]# mkdir /var/www/myweb

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

]# vim /etc/httpd/conf/httpd.conf #全文命令模式下 搜索 /Doc

DocumentRoot "/var/www/myweb"

]# systemctl restart httpd

]# firefox 172.25.0.11

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

实际路径与网络路径(浏览器中输入的路径)

客户端:firefox 172.25.0.11:80 ----->服务端 httpd---->主配置 DocumenRoot /var/www/myweb -----> /var/www/myweb

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

客户端:firefox 172.25.0.11/nsd

服务端:/var/www/myweb/nsd

DocumenRoot /var/www/myweb

客户端:firefox 172.25.0.11/var/www/myweb

服务端:/var/www/myweb/var/www/myweb

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

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

Web访问控制,默认继承父目录访问控制

访问控制的配置字段:

<Directory "/var/www">

Require all granted #允许所有人访问

</Directory>

<Directory "/var/www">

Require ip 172.25.0.10 #仅允许172.25.0.10访问

</Directory>

<Directory "/var/www">

Require all denied #拒绝所有人访问

</Directory>

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

<Directory "/var/www">

Require all denied #拒绝所有人访问

</Directory>

<Directory "/var/www/test">

Require all granted #允许所有人访问

</Directory>

/var/www/路径下的页面,拒绝所有人访问

/var/www/test路径下的页面,允许所有人访问

/var/www/abc路径下的页面,拒绝所有人访问

/var/www/abc/nsd路径下的页面,拒绝所有人访问

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

案例:

修改网页文件,存放路径为/webroot

1.创建路径写入网页文件

]# mkdir /webroot

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

]# ls /webroot/

2.修改配置文件

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

DocumentRoot "/webroot"

<Directory "/webroot"> #针对/webroot进行访问控制配置

Require all granted #允许所有人访问

</Directory>

[root@server0 ~]# setenforce 0 #将SELinux设置宽松模式

[root@server0 ~]# getenforce #查看SELinux状态

[root@server0 ~]# systemctl restart httpd

[root@server0 ~]# firefox 172.25.0.11

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

虚拟Web主机

– 由同一台服务器提供多个不同的Web页面

• 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

域名可以DNS解析 :

DNS服务器虚拟机classroom

server0.example.com

www0.example.com

webapp0.example.com

• 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>

• 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

虚拟机server

[root@server0 ~]# vim /etc/httpd/conf.d/nsd01.conf #调用配置文件

<VirtualHost \*:80> #启用虚拟Web主机功能

ServerName server0.example.com #设置访问的域名

DocumentRoot /var/www/qq #设置存放网页文件目录

</VirtualHost>

<VirtualHost \*:80> #启用虚拟Web主机功能

ServerName www0.example.com #设置访问的域名

DocumentRoot /var/www/baidu #设置存放网页文件目录

</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

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

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

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

一旦使用了虚拟Web主机功能,所有的网站都必须使用虚拟Web主机实现

如果直接使用IP地址进行访问,默认会将第一个虚拟Web主机页面内容呈现

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

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /var/www/qq

</VirtualHost>

<VirtualHost \*:80>

ServerName www0.example.com

DocumentRoot /var/www/baidu

</VirtualHost>

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/myweb

</VirtualHost>

[root@server0 ~]# systemctl restart httpd

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

## Day05

一 环境设置

虚拟机server

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

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

虚拟机desktop

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

[root@desktop0 ~]# firewall-cmd --get-default-zone

二 虚拟机server构建基于域名的虚拟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

- 实现客户端访问www0.example.com网页内容为 奔跑吧骆驼

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

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

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

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /var/www/nsd01

</VirtualHost>

<VirtualHost \*:80>

ServerName www0.example.com

DocumentRoot /var/www/nsd02

</VirtualHost>

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd03

</VirtualHost>

]# cd /var/www/

]# mkdir nsd01 nsd02 nsd03

]# echo '<h1>卖女孩的小火柴' > /var/www/nsd01/index.html

]# echo '<h1>奔跑吧骆驼' > /var/www/nsd02/index.html

]# echo '<h1>20春风不如你' > /var/www/nsd03/index.html

]# systemctl restart httpd

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

文件夹权限

• 针对 DocumentRoot 网页目录的权限控制

• 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

案例1:虚拟机server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.查看server0.example.com的DocumentRoot目录路径

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

]# mkdir /var/www/nsd01/private

]# echo '<h1>wo shi private' > /var/www/nsd01/private/index.html

]# firefox server0.example.com/private

2.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.d/nsd02.conf

<Directory "/var/www/nsd01/private">

Require ip 172.25.0.11 #仅允许172.25.0.11进行访问

</Directory>

[root@server0 ~]# systemctl restart httpd

虚拟机desktop0:访问测试权限不足

]# 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

]# ls /webroot/

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

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /webroot

</VirtualHost>

........

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

........

<Directory "/webroot">

Require all granted

</Directory>

[root@server0 ~]# systemctl restart httpd

SELinux三大策略: 1.布尔值 2.安全上下文 3.非默认端口的开放

[root@server0 ~]# setenforce 1 #开启SELinux

[root@server0 ~]# getenforce #查看SELinux状态

SELinux安全上下文(标签值)

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

[root@server0 ~]# 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

[root@server0 ~]# chcon -R --reference=/var/www /webroot/

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

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

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......

虚拟机server新建一个动态站点站点名称为 webapp0.example.com

1.部署Python页面

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

[root@server0 ~]# cd /var/www/nsd03

[root@server0 nsd03]# wget http://classroom.example.com/pub/materials/webinfo.wsgi

[root@server0 nsd03]# cat webinfo.wsgi

2.方便用户的访问,页面跳转(页面别名 地址重写)

浏览器中输入: firefox webapp0.example.com

服务端:/var/www/nsd03/webinfo.wsgi

Alias 网络路径 服务端实际路径

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

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd03

Alias / /var/www/nsd03/webinfo.wsgi

#当客户端直接访问网页文件根目录时,呈现/var/www/nsd03/webinfo.wsgi页面内容

</VirtualHost>

[root@server0 /]# systemctl restart httpd

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

3.安装mod\_wsgi软件包,专用于翻译Python页面

[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/nsd03

WsgiScriptAlias / /var/www/nsd03/webinfo.wsgi

</VirtualHost>

[root@server0 /]# systemctl restart httpd

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

显示内容为 UNIX时间戳:从1970-1-1 0:0:0到达当前系统时间,所经历的秒数

4.此虚拟webapp0.example.com主机侦听在端口8909

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

Listen 8909 #让httpd程序监听8909端口

<VirtualHost \*:8909> #本虚拟Web主机使用8909端口

ServerName webapp0.example.com

DocumentRoot /var/www/nsd03

WsgiScriptAlias / /var/www/nsd03/webinfo.wsgi

</VirtualHost>

5.SELinux非默认端口的开放

-l:列出 -a:添加 -t:类型 -p:协议

]# semanage port -l | grep http

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

]# systemctl restart httpd

]# firefox webapp0.example.com:8909

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

1.端口的优先级最高

2.默认将第一个虚拟Web主机内容呈现

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

安全Web服务,实现https(完全的超文本传输协议)

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

虚拟机server:构建安全的Web服务

1.部署网站证书

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

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

]# ls

2.部署根证书

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

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

]# ls

3.部署私钥(解密数据)

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

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

]# ls

4.安装支持安全的软件

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

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

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

vim末行模式 :set nu #开启行号功能

59 DocumentRoot "/var/www/html"

60 ServerName www0.example.com:443

#指定网站证书的路径及名称

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

#指定解密的私钥

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

#指定根证书

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

5.书写一个网站

]# echo '<h1>haha xixi' > /var/www/html/index.html

6.重启httpd服务

]# systemctl restart httpd

7.验证

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

我已了解可能的风险--->添加例外---->确认例外

## Day06

一 环境设置

虚拟机server

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

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

虚拟机desktop

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

[root@desktop0 ~]# firewall-cmd --get-default-zone

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

二 网络参数配置

方式一:nmtui

方式二:

1.设置永久设置主机名,修改/etc/hostname配置文件

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

[root@server0 ~]# hostname

A.tedu.cn

[root@server0 ~]# exit

登出

Connection to 172.25.0.11 closed.

[student@room9pc01 ~]$ gos

Last login: Fri May 17 09:04:25 2019 from 172.25.0.250

[root@A ~]#

2.配置IP地址,子网掩码,网关地址

1)查看识别的网卡名称

[root@A ~]# nmcli connection show

名称

System eth0

2)修改IP地址,子网掩码,网关地址

]# nmcli connection modify 'System eth0'

ipv4.method manual

ipv4.addresses '172.25.0.112/24 172.25.0.254'

connection.autoconnect yes

]# nmcli connection 修改 '识别的网卡名'

ipv4.方法 手工配置IP地址

ipv4.地址 'IP地址/子网掩码 网关地址'

每次开机自动启用

3)激活配置

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

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

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

DNS服务器地址:将域名解析为对应的IP地址

/etc/resolv.conf #指定DNS服务器最终有效配置文件

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

]# cat /etc/resolv.conf

nameserver 172.25.254.254

]# nslookup www0.example.com #测试域名解析

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

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

配置IPv6地址

• IPv4 地址表示

– 32个二进制位,点分隔的十进制数

– 例如:172.25.0.11、127.0.0.1

• IPv6 地址表示

– 128个二进制位,冒号分隔的十六进制数

– 每段内连续的前置 0 可省略、连续的多个 : 可简化为 ::

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

2003:ac18::305/64

1)查看识别的网卡名称

[root@A ~]# nmcli connection show

2)修改IP地址,子网掩码,网关地址

]# nmcli connection modify 'System eth0'

ipv6.method manual

ipv6.addresses 2003:ac18::305/64

connection.autoconnect yes

]# nmcli connection 修改 '识别的网卡名'

ipv6.方法 手工配置IP地址

ipv6.地址 IP地址/子网掩码

每次开机自动启用

3)激活配置

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

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

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

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

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

链路聚合(聚合链路 网卡绑定)

作用:提高网卡设备的可靠性,提供冗余

eth1 eth2

虚拟网卡:team0

192.168.1.1

1.建立虚拟网卡team0,参考 man teamd.conf 全文搜索 /example

]# nmcli connection add type team

con-name team0 ifname team0 autoconnect yes

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

]# nmcli connection 添加 类型 team

配置文件名 team0 网卡名 team0 每次开机自动启用

链路聚合管理方式为 热备份方式

]# ifconfig #查看生成的虚拟网卡team0

生成配置文件位置:/etc/sysconfig/network-scripts/

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 team0-1 网卡名 eth1 主设备为 team0

生成配置文件位置:/etc/sysconfig/network-scripts/

3.配置IP地址激活

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

[root@A ~]# nmcli connection up team0 #激活配置

[root@A ~]# teamdctl team0 state #专用于查看team0信息

[root@A ~]# ifconfig eth2 down #禁用eth2网卡

[root@A ~]# teamdctl team0 state

如果无法激活,删除所有配置重新来过

[root@A ~]# nmcli connection delete team0

[root@A ~]# nmcli connection delete team0-1

[root@A ~]# nmcli connection delete team0-2

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

parted分区工具,进行gpt分区模式

128个主分区 最大空间支持18EB

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

(parted) mktable gpt #指定分区表类型为gpt

(parted) print #输出分区表信息

(parted) mkpart #划分新的分区

分区名称？ []? haha #分区名称随意写

文件系统类型？ [ext2]? ext4 #文件系统随意些,不起实际作用

起始点？ 0 #分区起始点

结束点？ 2G #分区结束点

警告: The resulting partition is not properly aligned

for best performance.

忽略/Ignore/放弃/Cancel? Ignore #选择ignore忽略

(parted) print

(parted) unit GB #使用GB最为单位

(parted) print

(parted) mkpart

分区名称？ []? haha

文件系统类型？ [ext2]? ext4

起始点？ 2G

结束点？ 5G

(parted) print

(parted) quit

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

基础邮件服务

邮件域名:

server0.example.com

邮箱帐号:

yg@server0.example.com

• 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

快速部署postfix邮件服务器

1.装postfix软件包

[root@A ~]# rpm -q postfix

postfix-2.10.1-6.el7.x86\_64

2.修改配置文件/etc/postfix/main.cf

vim末行模式 :set nu #开启行号

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

116 inet\_interfaces = all #允许本机所有网卡提供邮件服务

164 mydestination = server0.example.com #此值判断为本域邮件

3.重启服务

[root@A ~]# systemctl restart postfix

测试邮件收发

[root@A ~]# useradd yg

[root@A ~]# useradd xln

•发信操作: mail -s '邮件标题' -r 发件人 收件人

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

ahahxixihehelele

.

EOT

[root@A ~]# echo AAA | mail -s 'test02' -r yg xln

•收信操作: mail [-u 用户名]

[root@A ~]# mail -u xln

>N 1 yg@server0.example.c Fri May 17 15:28 18/565

& 1 #输入邮件编号1回车查看邮件的内容

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

使用LDAP认证,实现网络用户认证,达到用户的集中管理

　　RHCSA　红帽认证管理员

　　RHCE　红帽认证工程师

　　RHCA　红帽认证架构师

网络用户:用户信息会放在网络中LDAP服务器

本地用户:用户信息会放在/etc/passwd

LDAP服务器:虚拟机classroom

客户端:虚拟机server

1.安装sssd软件,与LDAP服务器沟通

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

[root@server0 /]# rpm -q sssd

2.安装authconfig-gtk图形的工具,进行配置客户端sssd软件

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

[root@server0 /]# authconfig-gtk

完全合格的主机名 www.baidu.com　　　www.qq.com

classroom.example.com

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

3.重起服务

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

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

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

[root@server0 ~]# id ldapuser0 #验证LDAP用户信息

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