**CSA-200—小戴哥哥—**

1. 设置主机名

[root@server0 ~]# hostnamectl set-hostname server0.example.com

1. 配置IP

[root@server0 ~]# nmcli connection modify "System eth0" ipv4.method manual ipv4.addresses "172.25.0.11/24 172.25.0.254" ipv4.dns 172.25.254.254 connection.autoconnect yes

[root@server0 ~]# nmcli connection up "System eth0"

3.为您的系统指定一个默认的软件仓库

YUM的软件库源为 http://content.example.com/rhel7.0/x86\_64/dvd，将此配置为您的系统的默认软件仓库。

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

[dvd]

name=dvd

baseurl= http://content.example.com/rhel7.0/x86\_64/dvd

enabled=1

gpgcheck=0

[root@server0 ~]# yum repolist

1. 调整逻辑卷的大小

自行建立一个200MiB的逻辑卷 /dev/systemvg/vo，格式化为ext3文件系统并挂载到 /vo 。然后将逻辑卷vo和其文件系统大小调整到 300 MiB。要确保文件系统中的内容保持完整。请注意：分区大小很少能够完全符合要求的大小，所以大小在 270 MiB 和 330 MiB 之间都是可以接受的。

[root@server0 ~]# fdisk /dev/vdb——n——回车到last:+500M——w保存

[root@server0 ~]# 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

└─vdb1 253:17 0 500M 0 part

[root@server0 ~]# vgcreate systemvg /dev/vdb1

[root@server0 ~]# lvcreate -n vo -L 200M systemvg

[root@server0 ~]# mkfs.ext3 /dev/systemvg/vo

[root@server0 ~]# lvextend -L 300M /dev/systemvg/vo

[root@server0 ~]# resize2fs /dev/systemvg/vo

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

/dev/systemvg/vo /vo ext3 defaults 0 0

[root@server0 ~]# mkdir /vo

[root@server0 ~]# mount -a

5.创建用户帐户

创建下列用户、组以及和组的成员关系：

一个名为adminuser的组

一个名为natasha的用户，其属于adminuser，这个组是该用户的从属组

一个名为harry的用户，属于adminuser，这个组是该用户的从属组

一个名为sarah的用户，其在系统中没有可交互的shell，并且不是adminuser组的成员用户

natasha、harry、和sarah的密码都要设置为flectrag

[root@server0 ~]# groupadd adminuser

[root@server0 ~]# useradd -G adminuser natasha

[root@server0 ~]# useradd -G adminuser harry

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

[root@server0 ~]# echo flectrag | passwd --stdin natasha

[root@server0 ~]# echo flectrag | passwd --stdin harry

[root@server0 ~]# echo flectrag | passwd --stdin sarah

6.配置文件/var/tmp/fstab的权限

拷贝文件/etc/fstab到/var/tmp/fstab，配置文件/var/tmp/fstab的权限：

文件/var/tmp/fstab的拥有者是root用户

文件/var/tmp/fstab属于root组

文件/var/tmp/fstab对任何人都不可执行

用户natasha能够对文件/var/tmp/fstab执行读和写操作

用户harry 对文件/var/tmp/fstab既不能读，也不能写

所有其他用户（当前的和将来的）能够对文件/var/tmp/fstab进行读操作

[root@server0 ~]# cp /etc/fstab /var/tmp/fstab

[root@server0 ~]# setfacl -m u:natasha:rwx /var/tmp/fstab

[root@server0 ~]# setfacl -m u:harry:- /var/tmp/fstab

7.配置一个cron任务

为用户natasha配置一个定时任务，每天在本地时间14:23时执行命令 /bin/echo hiya

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

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

[root@server0 ~]# systemctl restart crond.service

[root@server0 ~]# systemctl enable crond.service

8.创建一个共享目录

创建一个共享目录/home/admins ，特性如下：

/home/admins目录的组所有权是adminuser

adminuser组的成员对目录有读写和执行的权限。除此之外的其他所有用户没有任何权限（root用户能够访问系统中的所有文件和目录）

在/home/admins目录中创建的文件，其组所有权会自动设置为属于adminuser组

[注]此处所谓的共享目录并不是指网络共享，只是某个组成员共用

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

[root@server0 ~]# chown ldapuser0 /home/admins

[root@server0 ~]# chmod 2770 /home/admins

9.安装内核的升级

从指定的地址 http://classroom.example.com/content/rhel7.0/x86\_64/errata/Pa ckages/ 找到新版kernel的rpm安装文件，下载并用来升级内核，同时要满足下列要求：

当系统重新启动之后升级的内核要作为默认的内核

原来的内核要被保留，并且仍然可以正常启动

[root@server0 ~]# firefox http://classroom.example.com/content/rhel7.0/x86\_64/errata/Packages/

[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 ~]# yum -y install kernel-3.10.0-123.1.2.el7.x86\_64.rpm

[root@server0 ~]# reboot

[root@server0 ~]# uname -r

10.绑定到外部验证服务

系统 classroom.example.com 提供了一个 LDAP 验证服务。您的系统需要按照以下要求绑定到这个服务上：

验证服务器的基本 DN 是：dc=example,dc=com

帐户信息和验证信息都是由 LDAP 提供的

连接要使用证书进行加密，证书可以在下面的链接中下载 ：http://classroom.example.com/pub/example-ca.crt

当正确完成配置后，用户 ldapuser0 应该能够登录到您的系统中，但是没有主目录。当您完成 autofs的题目之后，才能生成主目录

用户ldapuser0的密码是password

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

[root@server0 ~]# authconfig-tui

使用LDAP 使用LDAP认证 使用TLS

服务器： classroom.example.com

基础 DN： dc=example,dc=com

[root@server0 cacerts]# wget http://classroom.example.com/pub/example-ca.crt

[root@server0 cacerts]# authconfig-tui(刷新一下)

[root@server0 cacerts]# systemctl restart sssd

[root@server0 cacerts]# systemctl enable sssd

[root@server0 cacerts]# id ldapuser0(以下显示为成功)

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

11.autofs的配置

按照下述要求配置 autofs 用来自动挂载 LDAP 用户的主目录：

classroom.example.com（172.25.0.11）通过 NFS 输出 /home/guests 目录到您的系统，这个文件系统包含了用户ldapuser0的主目录，并且已经预先配置好了

ldapuser0用户的主目录是 classroom.example.com:/home/guests/ldapuser0

ldapuser0的主目录应该按需挂载到本地的/home/guests/ldapuser0 目录下

用户对其主目录必须是可写的

ldapuser0用户的密码是password

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

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

/home/guests /etc/guests.rule

[root@server0 ~]# vim /etc/guests.rule

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

[root@server0 ~]# systemctl restart autofs

[root@server0 ~]# systemctl enable autofs

12.配置NTP网络时间客户端

配置您的系统，让其作为 NTP 服务器 classroom.example.com 的客户端

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

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

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

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

server classroom.example.com iburst(2-5行#注释，第6行改掉)

[root@server0 ~]# systemctl restart chronyd.service

[root@server0 ~]# systemctl enable chronyd.service

13.配置一个用户帐户

创建一个名为alex的用户，用户ID是 3456，密码是flectrag

[root@server0 ~]# useradd -u 3456 alex

[root@server0 ~]# echo flectrag | passwd --stdin alex

14.添加一个swap分区

在您的系统中添加一个大小为 512 MiB 的swap分区：

当您的系统启动时，swap 分区应该可以自动挂载

不要移除或者修改其他已经存在于您的系统中的 swap 分区

[root@server0 ~]# fdisk /dev/vdb——n——回车到last:+512M——w保存

[root@server0 ~]# partprobe

[root@server0 ~]# 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

├─vdb1 253:17 0 500M 0 part

└─vdb2 253:18 0 512M 0 part

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

正在设置交换空间版本 1，大小 = 524284 KiB

无标签，UUID=c8bcdbe7-31eb-4e96-87a4-9496b8b01b80

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

/dev/vdb2 swap swap defaults 0 0

[root@server0 ~]# swapon -a

[root@server0 ~]# swapon -s

文件名 类型 大小 已用 权限

/dev/vdb2 partition 524284 0 -1

15.查找文件

找出所有用户student拥有的文件，并且把它们拷贝到/root/findfiles 目录中

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

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

16.查找一个字符串

在文件/usr/share/dict/words中查找到所有包含字符串seismic的行：

将找出的行按照原文的先后顺序拷贝到/root/wordlist文件中

/root/wordlist文件不要包含空行，并且其中的所有行的内容都必须是 /usr/share/dict/words文件中原始行的准确副本

[root@server0 ~]# grep seismic /usr/share/dict/words > /root/wordlist

17.创建一个逻辑卷

根据下面的要求创建一个新的逻辑卷：

逻辑卷命名为database，属于datastore卷组，并且逻辑卷的大小为50个物理扩展单元 (physical extent)

在datastore卷组中的逻辑卷，物理扩展单元 (physical extent) 大小应为 16 MiB

使用ext3文件系统对新的逻辑卷进行格式化，此逻辑卷应该在系统启动的时候自动挂载在 /mnt/database 目录下

[root@server0 ~]# fdisk /dev/vdb——n——回车到last:+1G——w保存

[root@server0 ~]# partprobe

[root@server0 ~]# 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

├─vdb1 253:17 0 500M 0 part

├─vdb2 253:18 0 512M 0 part [SWAP]

└─vdb3 253:19 0 1G 0 part

[root@server0 ~]# vgcreate -s 16M systemvg /dev/vdb3

Physical volume "/dev/vdb3" successfully created

Volume group "systemvg" successfully created

[root@server0 ~]# lvcreate -n vo -L 200M systemvg

Rounding up size to full physical extent 208.00 MiB

Logical volume "vo" created

[root@server0 ~]# mkfs.ext3 /dev/systemvg/vo

[root@server0 ~]# lvextend -L 300M /dev/systemvg/vo

Rounding size to boundary between physical extents: 304.00 MiB

Extending logical volume vo to 304.00 MiB

Logical volume vo successfully resized

[root@server0 ~]# resize2fs /dev/systemvg/vo

resize2fs 1.42.9 (28-Dec-2013)

Resizing the filesystem on /dev/systemvg/vo to 311296 (1k) blocks.

The filesystem on /dev/systemvg/vo is now 311296 blocks long.

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

/dev/systemvg/vo /vo ext3 defaults 0 0

[root@server0 ~]# mkdir /vo

[root@server0 ~]# mount -a

18.创建一个归档

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归档必须使用bzip2进行压缩

[root@server0 ~]# tar -jcPf /root/backup.tar.bz2 /usr/local/

## CE-300—小戴哥哥—

## 配置SELinux

## 确保您的两个虚拟机的SELinux处于强制启用模式

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

## SELINUX=enforcing

[root@serverX ~]# setenforce 1

[root@serverX ~]# getenforce

Enforcing

## 2.配置SSH访问

按以下要求配置 SSH 访问：

· 用户能够从域 example.com 内的客户端 SSH 远程访问您的两个虚拟机系统

· 在域 my133t.org 内的客户端不能访问您的两个虚拟机系统

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

最底下

DenyUsers \*@\*.my133t.org \*@172.34.0.\*

[root@serverX ~]# systemctl restart sshd

[root@serverX ~]# systemctl enable sshd

## 3.自定义用户环境 在系统 server0 和 desktop0 上创建自定义命令为 qstat：

· 此自定义命令将执行以下命令：/bin/ps -Ao pid,tt,user,fname,rsz

· 此命令对系统中所有用户有效

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

底下添加

alias qstat='/bin/ps -Ao pid,tt,user,fname,rsz'

[root@serverX ~]# source /etc/bashrc

[root@serverX ~]# qstat

## 4.配置防火墙端口转发 在系统 server0 上配置端口转发，要求如下：

· 在 172.25.0.0/24 网络中的系统，访问 server0 的本地端口 5423 将被转发到80

· 此设置必须永久有效

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

[root@server0~]#firewall-cmd --permanent --add-source=172.34.0.0/24 --zone=block

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

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

## 5.配置链路聚合 在 server0.example.com 和 desktop0.example.com 之间按以下要求配置一个链路team0：

· 此链路使用接口 eth1 和 eth2

· 此链路在一个接口失效时仍然能工作

· 此链路在 server0 使用下面的地址 172.16.3.20/255.255.255.0

· 此链路在 desktop0 使用下面的地址 172.16.3.25/255.255.255.0

· 此链路在系统重启之后依然保持正常状态

Server0

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

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

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

[root@server0 ~]# nmcli connection modify team0 ipv4.method manual ipv4.addresses "172.16.3.20/24" connection.autoconnect yes

[root@server0 ~]# nmcli connection up team0

[root@server0 ~]# nmcli connection up team0-1

[root@server0 ~]# nmcli connection up team0-2

Desktop0

[root@desktop0 ~]# nmcli connection add con-name team0 type team ifname team0 config '{"runner":{"name":"activebackup"} }'

[root@desktop0 ~]# nmcli connection add con-name team0-1 type team-slave ifname eth1 master team0

[root@desktop0 ~]# nmcli connection add con-name team0-2 type team-slave ifname eth2 master team0

[root@desktop0 ~]# nmcli connection modify team0 ipv4.method manual ipv4.addresses "172.16.3.25/24" connection.autoconnect yes

[root@desktop0 ~]# nmcli connection up team0

[root@desktop0 ~]# nmcli connection up team0-1

[root@desktop0 ~]# nmcli connection up team0-2

## 6.配置IPv6地址 在您的两个考试系统上配置接口 eth0 使用下列 IPv6 地址：

· server0 上的地址应该是 2003:ac18::305/64

· desktop0 上的地址应该是 2003:ac18::306/64

· 两个系统必须能与网络 2003:ac18/64 内的系统通信

· 地址必须在重启系统后依旧生效

· 两个系统必须保持当前的 IPv4 地址并仍然能够通信

Server0:

[root@server0 ~]# nmcli connection modify "System eth0" ipv6.method manual ipv6.addresses "2003:ac18::305/64" connection.autoconnect yes

[root@server0 ~]# nmcli connection up "System eth0"

Desktop0:  
[root@server0 ~]# nmcli connection modify "System eth0" ipv6.method manual ipv6.addresses "2003:ac18::306/64" connection.autoconnect yes  
[root@server0 ~]# nmcli connection up "System eth0"

## 7.配置本地邮件服务 在系统 server0 和 desktop0 上配置邮件服务，满足以下要求：

· 这些系统不接收外部发送来的邮件

· 在这些系统上本地发送的任何邮件都会自动路由到 smtp0.example.com

· 从这些系统上发送的邮件显示来自于 desktop0.example.com

· 您可以通过在 server0.example.com 发送邮件到本地用户 student 来测试您的配置，并将系统 desktop0.example.com 配置为后端邮件服务器，确保可接收并投递来自 server0 的邮件

环境:lab smtp-nullclient setup

Server0:(改主机名)

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

myorigin = desktop0.example.com

inet\_interfaces = loopback-only

mydestination =

mynetworks = 127.0.0.0/8 [::1]/128

relayhost = [smtp0.example.com]

最底下添加：

local\_transport = error:123

[root@serverX ~]# systemctl restart postfix

[root@serverX ~]# systemctl enable postfix

[root@serverX ~]# mail -s “dai” student </etc/passwd

Desktop0:（改主机名）

[root@serverX ~]# mail -u student

## 8.通过 Samba 发布共享目录 在 server0 通过 SMB 共享/common 目录：

· 您的 SMB 服务器必须是 STAFF 工作组的一个成员

· 共享名必须为 common

· 只有 example.com 域内的客户端可以访问 common 共享

· common 必须是可以浏览的

· 用户 harry 必须能够读取共享中的内容，如果需要的话，验证的密码是 migwhisk

Server0：

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

[root@serverX ~]# mkdir /common

[root@serverX ~]# useradd harry

[root@serverX ~]# pdbedit -a harry

[root@serverX ~]# migwhisk(2次)

[root@serverX ~]# getsebool -a | grep samba

[root@serverX ~]# setsebool -P samba\_export\_all\_ro=on

[root@serverX ~]# setsebool -P samba......\_all\_rw=on

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

XXXgroup = STAFF

左下角：

[common]

Path = /common

Hosts allow = 172.25.0.0/24

[root@serverX ~]# systemctl restart smb

[root@serverX ~]# systemctl enable smb

## 9.配置多用户Samba挂载 在 server0 通过 SMB 共享目录/devops，并满足以下要求：

· 共享名为 devops

· 共享目录 devops 只能被 example.com 域中的客户端使用

· 共享目录 devops 必须可以被浏览

· 用户 kenji 必须能以读的方式访问此共享，该问密码是 atenorth

· 用户 chihiro 必须能以读写的方式访问此共享，访问密码是 atenorth

· 此共享永久挂载在 desktop0.example.com 上的/mnt/dev 目录，并使用用户 kenji 作为认证。任何用户可以通过用户 chihiro 来临时获取写的权限

Server0:

[root@serverX ~]# mkdir /devops

[root@serverX ~]# useradd kenji

[root@serverX ~]# useradd chihiro

[root@serverX ~]# pdbedit -a kenji

[root@serverX ~]# atenorth(2次)

[root@serverX ~]# pdbedit -a chihiro

[root@serverX ~]# atenorth(2次)

[root@serverX ~]# setfac -m u:chirhiro:rwx /devops/

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

XXXgroup = STAFF

左下角

[common]

Path = /common

Hosts allow = 172.25.0.0/24

[devops]

Path = /devops

Hosts allow = 172.25.0.0/24

Write list = chihiro

[root@serverX ~]# systemctl restart smb

[root@serverX ~]# systemctl enable smb

Desktop0:

[root@desktop0X ~]# yum -y install samba-client.x86\_64 cifs-utils.x86\_64

[root@desktop0X ~]# mkdir /mnt/dev

[root@desktop0X ~]# vim /etc/fstab

//server0.example.com/devops /mnt/dev cifs user=kenji,pass=atenorth,multiuser,sec=ntlmssp,\_netdev 0 0

[root@desktop0X ~]#mount -a

## 10.配置NFS共享服务 在 server0 配置 NFS 服务，要求如下：

· 以只读的方式共享目录 /public，只能被 example.com 域中的系统访问

· 以读写的方式共享目录 /protected，只能被 example.com 域中的系统访问

· 访问 /protected 需要通过 Kerberos 安全加密，您可以使用下面 URL 提供的密钥：<http://classroom.example.com/pub/keytabs/server0.keytab>

· 目录 /protected 应该包含名为 project 拥有人为 ldapuser0 的子目录

· 用户 ldapuser0 能以读写方式访问 /protected/project

环境:lab nfskrb5 setup

[root@server0 ~]# Mkdir -p /public /protected/project

[root@server0 ~]# chown ldapuser0 /protected/project

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

/public 172.25.0.0/24(ro)

/protected 172.25.0.0/24(rw,sec=krb5p)

[root@server0 ~]# wget -O /etc/krb5.keytab <http://classroom.example.com/pub/keytabs/server0.keytab>

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

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

## 11.挂载NFS共享 在 desktop0 上挂载一个来自 server0.example.com 的共享，并符合下列要求：

· /public 挂载在下面的目录上 /mnt/nfsmount

· /protected 挂载在下面的目录上 /mnt/nfssecure 并使用安全的方式，密钥下载 URL：<http://classroom.example.com/pub/keytabs/desktop0.keytab>

· 用户 ldapuser0 能够在/mnt/nfssecure/project 上创建文件

· 这些文件系统在系统启动时自动挂载

[root@desktop0 ~]# mkdir /mnt/nfssecure /mnt/nfsmount

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

server0.example.com:/public /mnt/nfsmount nfs \_netdev 0 0

server0.example.com:/protected /mnt/nfssecure nfs sec=krb5p,\_netdev

0 0

[root@desktop0 ~]# wget -O /etc/krb5.keytab <http://classroom.example.com/pub/keytabs/desktop0.keytab>

[root@desktop0 ~]# systemctl restart nfs-secure

[root@desktop0 ~]# systemctl enable nfs-secure

[root@desktop0 ~]# mount -a

## 12.实现一个Web服务器 为 http://server0.example.com 配置 Web 服务器：

· 从URL地址 <http://classroom.example.com/pub/materials/station.html> 下载一个主页文件，并将该文件重命名为 index.html

· 将文件 index.html 拷贝到您的 web 服务器的 DocumentRoot 目录下

· 不要对文件 index.html 的内容进行任何修改

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

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

[root@server0 html~]# wget <http://classroom.example.com/pub/materials/station.html> -O index.html

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

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /var/www/html

</VirtualHost>

[root@server0 ~]# systemctl restart httpd

[root@server0 ~]# systemctl enable httpd

## 13.配置安全Web服务 为站点 http://server0.example.com 配置TLS加密：

· 一个已签名证书从http://classroom.example.com/pub/tls/certs/server0.crt 获取

· 此证书的密钥从 http://classroom.example.com/pub/tls/private/server0.key获取

· 此证书的签名授权信息从 http://classroom.example.com/pub/example-ca.crt获取

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

[root@server0 ~]# cd /etc/pki/tls/certs

[root@server0 certs~]# wget http://classroom.example.com/pub/tls/certs/server0.crt

[root@server0 certs~]# wget  http://classroom.example.com/pub/example-ca.crt

[root@server0 certs~]# cd ..

[root@server0 tls~]# cd private

[root@server0 private~]# wget http://classroom.example.com/pub/tls/private/server0.key

[root@server0 private~]# vim /etc/hhtpd/conf.d/ssl.conf

更改100行 localhost.cret——server0.crt

107行 localhost.key——server0.key

122行 ca-bundle.crt——example-ca.crt

去掉122行的#注释

[root@server0 private~]# systemctl restart httpd

[root@server0 private~]# systemctl enable httpd

14.配置虚拟主机  
在 server0 上扩展您的 web 服务器，为站点 http://www0.example.com 创建一个虚拟主机，然后执行下述步骤：

· 设置 DocumentRoot 为/var/www/virtual

· 从 <http://classroom.example.com/pub/materials/www.html> 下载文件并重命名为index.html

· 不要对文件 index.html 的内容进行任何修改

· 将文件 index.html 放到虚拟主机的 DocumentRoot 目录下

· 确保 fleyd 用户能够在/var/www/virtual 目录下创建文件

注意：原始站点 http://server0.example.com 必须仍然能够访问，名称服务器 example.com 提供对主机名 www0.example.com 的域名解析。

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

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

[root@server0 virtual~]# wget <http://classroom.example.com/pub/materials/www.html>  -O index.html

[root@server0 virtual~]# useradd fleyd

[root@server0 virtual~]# setfacl -m u:fleyd:rwx /var/www/virtual/

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

...

...

<VirtualHost \*:80>

ServerName www0.example.com

DocumentRoot /var/www/virtual

</VirtualHost>

[root@server0 virtual~]# systemctl restart httpd

[root@server0 virtual~]# systemctl enable httpd

## 15.配置Web内容访问 在您的 server0 web服务器的 DocumentRoot 目录下创建一个名为 private 的目录，要求如下：

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

· 不要对这个文件的内容做任何修改

· 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

[root@server0 ~]# mkdir /var/www/html/private

[root@server0 ~]# cd /var/www/html/private

[root@server0 private~]# wget <http://classroom.example.com/pub/materials/private.html> -O index.html

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

<Directory /var/www/html/private>

Require ip 127.0.0.1 ::1 172.25.0.11

</Directory>

[root@server0 private~]# syetemctl restart httpd

[root@server0 private~]# systemctl enabled httpd

## 16.实现动态Web内容 在 server0 上配置提供动态Web内容，要求如下：

· 动态内容由名为 webapp0.example.com 的虚拟主机提供

· 虚拟主机侦听在端口 8909

· 从 <http://classroom.example.com/pub/materials/webinfo.wsgi> 下载一个脚本，然后放在适当的位置，无论如何不要修改此文件的内容

· 客户端访问 [http://webapp0.example.com:8909](http://webapp0.example.com:8909/) 可接收到动态生成的 Web 页

· 此 http://webapp0.example.com:8909/必须能被 example.com 域内的所有系统访问

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

[root@server0 ~]# mkdir /var/www/webapp0

[root@server0 ~]# cd /var/www/webapp0

[root@server0 webapp0~]# wget <http://classroom.example.com/pub/materials/webinfo.wsgi>

[root@server0 webapp0~]# vim /etc/httpd//conf.d/dai.conf

...

...

Listen 8909

<VirtualHost \*:8909>

ServerName webapp0.example.com

DocumentRoot /var/www/webapp0

WSGIScriptAlias / /var/www/webapp0/webinfo.wsgi

</VirtualHost>

[root@server0 webapp0~]# semanage port -a -t http\_port\_t -p tcp 8909

[root@server0 webapp0~]# systemctl restart httpd

[root@server0 webapp0~]# systemctl enable httpd

## 17.创建一个脚本 在 server0 上创建一个名为/root/foo.sh 的脚本，让其提供下列特性：

· 当运行/root/foo.sh redhat，输出为 fedora

· 当运行/root/foo.sh fedora，输出为 redhat

· 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的信息：/root/foo.sh redhat|fedora

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

#!/bin/bash

If [ “$1” = “redhat” ];then

Echo fedora

Elif [ “$1” = “fedora” ];then

Echo redhat

Else

Echo “/root/foo.sh redhat | fedora” >&2

Fi

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

## 18.创建一个添加用户的脚本 在 server0 上创建一个脚本，名为/root/batchusers，此脚本能实现为系统 server0 创建本地用户，并且这些用户的用户名来自一个包含用户名的文件，同时满足下列要求：

· 此脚本要求提供一个参数，此参数就是包含用户名列表的文件

· 如果没有提供参数，此脚本应该给出下面的提示信息 Usage: /root/batchusers <userfile> 然后退出并返回相应的值

· 如果提供一个不存在的文件名，此脚本应该给出下面的提示信息 Input file not found 然后退出并返回相应的值

· 创建的用户登陆Shell为/bin/false，此脚本不需要为用户设置密码

· 您可以从下面的 URL 获取用户名列表作为测试用：<http://classroom.example.com/pub/materials/userlist>

[root@server0 ~]#wget -O /root/userlist <http://classroom.example.com/pub/materials/userlist>

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

#!/bin/bash

if [ “$#” -eq 0 ];then

Echo “Usage: /root/batchusers <userfile>”

Exit 1

Fi

if [ ! -f $1 ];then

Echo “ Input file not found”

Exit 2

Fi

For name in $(cat $1)

Do

Useradd -s /bin/false $name

Done

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

## 19.配置iSCSI服务端 配置 server0 提供一个 iSCSI 服务，磁盘名为 iqn.2016-02.com.example:server0，并符合下列要求：

· 服务端口为 3260

· 使用 iscsi\_store 作其后端卷，其大小为 3GiB

· 此服务只能被 desktop0.example.com 访问

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

[root@server0 ~]#fdisk /dev/vdb(分区+3G)

[root@server0 ~]#targetcli

/>backstores/block create iscsi\_store /dev/vdb1

/>iscsi/ create iqn.2016-02.com.example:server0

/>iscsi/iqn.2016-02.com.example:server0/tpg1/acls create iqn.2016-02.com.example:desktop0

/>iscsi/iqn.2016-02.com.example:server0/tpg1/luns create /backstores/block/iscsi\_store

/>iscsi/iqn.2016-02.com.example:server0/tpg1/portals create 172.25.0.11 3260

/>saveconfing

/>exit

[root@server0 ~]# systemctl restart target

[root@server0 ~]# systemctl enable target

## 20.配置iSCSI客户端 配置 desktop0 使其能连接 server0 上提供的 iqn.2016-02.com.example:server0，并符合以下要求：

· iSCSI设备在系统启动的期间自动加载

· 块设备iSCSI上包含一个大小为 2100MiB 的分区，并格式化为ext4文件系统

· 此分区挂载在 /mnt/data 上，同时在系统启动的期间自动挂载

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

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

InitiatorName=iqn.2016-02.com.example:desktop0

[root@desktop0 ~]# systemctl restart iscsid

[root@desktop0 ~]# systemctl enable iscsid

[root@desktop0 ~]# man iscsiadm(在example里面找)

更改IP ：iscsiadm --mode discoverydb --type sendtargets --portal 172.25.0.11 --discover

[root@desktop0 ~]# systemctl restart iscsi

[root@desktop0 ~]# systemctl enable iscsi

[root@desktop0 ~]# lsblk

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

[root@desktop0 ~]# fdisk /dev/sda——n——回车到last:+2100M——w保存

[root@desktop0 ~]# mkfs.ext4 /dev/sda1

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

/dev/sda1 /mnt/data/ ext4 defaults 0 0

[root@desktop0 ~]#mount -a

## 21.配置一个数据库 在 server0 上创建一个 MariaDB 数据库，名为 Contacts，并符合以下条件：

· 数据库应该包含来自数据库复制的内容，复制文件的 URL 为：<http://classroom.example.com/pub/materials/users.sql>

· 数据库只能被 localhost 访问

· 除了root用户，此数据库只能被用户 Raikon 查询，此用户密码为 atenorth

· root用户的密码为 atenorth，同时不允许空密码登陆

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

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

第二行添加：skip-networking

[root@server0 ~]# systemctl restart mariadb

[root@server0 ~]# systemctl enable mariadb

[root@server0 ~]#mysqladmin -u root password ‘atenorth’

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

验证密码：atenoth

MariaDB[(none)]>create database Contacts;

MariaDB[(none)]>grant select on Contacts.\* to Rainkon@localhost identified by ‘atenorth’;

MariaDB[(none)]>delete from mysql.user where password=’’;

MariaDB[(none)]>quit

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

[root@server0 ~]# mysql -u root -p Contacts < users.sql

验证密码：atenorth

22.数据库查询  
在系统 server0 上使用数据库 Contacts，并使用相应的 SQL 查询以回答下列问题：

· 密码是 solicitous 的人的名字？

· 有多少人的姓名是 Barbara 同时居住在 Sunnyvale？

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

验证密码：atenorth

MariaDB [(none)]>use Contacts

MariaDB [Contacts]> select name from base where password=’solicitous’;

MariaDB [Contacts]> select count(\*) from base,location where base.name=’Baraba’ and location.city=’Sunnyvale’ and base.id=location.id;

MariaDB [Contacts]>quit