

考前准备

1) 不需要重启，切换到图像化界面

```
# systemctl isolate multi-user.target
# systemctl isolate graphical.target
```

2) 配置 yum 源

```
# vim /etc/yum.repos.d/yum.repo
[rhel7.lga]
name = rhel7.lga
baseurl = http://***/dvd
enabled = 1
gpgcheck =0
# yum clean all
# yum list
# yum repolist
```

3) 要求缩减/home 分区到 100M

```
# df -hT
/dev/mapper/wgroup-lvdata ext4      120M  1.6M  110M   2% /home
# umount /home/
# e2fsck -f /dev/wgroup/lvdata
# resize2fs /dev/wgroup/lvdata 100M
# lvreduce -L 100M /dev/wgroup/lvdata
# lvdisplay
# vim /etc/fstab
# mount -a
# df -hT
```

4) 扩展 home 为 512M

```
# df -hT
/dev/mapper/wgroup-lvdata xfs      101M  5.4M   96M   6% /home
# fdisk
t
5
8e
w
# partprobe
```

```
# pvcreate /dev/sdb5
# pvdisplay
# vgextend wgroup /dev/sdb5
# vgdisplay
# lvextend -L 512M /dev/wgroup/lvdata
# lvdisplay
    LV Size                512.00 MiB
# xfs_growfs /home/
    (xfs_growfs 采用的是挂载点。 Resize2fs 采用逻辑卷名称)
# vim /etc/fstab
/dev/wgroup/lvdata        /home    xfs      defaults    1        2
# umount /home
# df -hT
# mount -a
# df -hT
```

5) 缩减 VG

```
# pvmove /dev/sdb5
# pvdisplay
# vgreduce wgroup /dev/sdb5
# vgdisplay
```

6) 创建manager 组、创建2个用户natasha,harry 并要求其附属组 为manager组，创建第三个用户 strtlt 不允许login

```
# groupadd manager
# grep manager /etc/group
# useradd natasha 或者直接使用useradd -G manager natasha
# useradd harry 或者直接使用useradd -G manager harry
# passwd harry
# passwd natasha
# usermod -G manager harry
# usermod -G manager Natasha
# useradd -s /sbin/nologin strtlt
```

7) 扩展

```
# usermod -L harry 锁定
# usermod -U harry 解锁
# chage -M 90 harry 每90天修改密码
# chage -l harry
Maximum number of days between password change : 90
# chage -d 0 harry 强制第一次登录时修改密码
# date -d "+90 days"
```

```
# chage -E 2016-03-08 harry      设置用户harry90天后到期
```

- 8) 复制/etc/fstab 到/var/tmp/ 目录下 并设置 harry 用户对其可以读写、netasha不能做任何操作、其他用户可读、设置fstab所属组为manager

```
# cp /etc/fstab /var/tmp/  
# chgrp manager /var/tmp/fstab      - 需要确认对不对  
# setfacl -m u:harry:rw /var/tmp/fstab  
# setfacl -m u:netasha:- /var/tmp/fstab  
# setfacl -m o::r /var/tmp/fstab - 需要确认对不对  
# getfacl /var/tmp/fstab
```

- 9) 扩展

```
#setfacl -Rm g:manager:rwX /var/tmp  
递归方式更新组manager对tmp目录有读写，及目录的执行权限  
# setfacl -Rm u:test1:rX /var/tmp/  
递归方式对用户test1授予读和有条件的执行权限  
# setfacl -m d:g:groupuser1:rwX /var/tmp  
对组groupuser1组成员更新默认权限：读写执行  
# setfacl -m d:u:harry1:rx /var/tmp/ 更改用户harry1默认权限，读执行。  
# setfacl -m m::r /var/tmp/  
# setfacl -m d:m::r /var/tmp/ ACL掩码  
# setfacl -x u:test1 /var/tmp/ 删除用户test1的权限  
# setfacl -x g:manager /var/tmp/fstab 删除组manager的权限
```

- 10) Crontab: 用户netasha在14:23执行 echo “file_world”

```
# crontab -u natasha -e  
23      14      *      *      *      echo "file_world"  
# crontab -l -u natasha  
23      14      *      *      *      echo "file_world"
```

11) 扩展

```
# vim /etc/crontab
# crontab -r 删除任务
# crontab -u natasha -r 删除任务
# crontab -u natasha -l
```

12) 升级kernel, 并设置默认以新内核启动 :

```
# wget -O ftp://server1.domain20.example.com/pub/update/new.kernel
# vim /boot/grub/grub.cnf
# rpm -ivh kern*.rpm
# vim /boot/grub/grub.cnf
# reboot
```

13) 配置ldap认证、dc=domain20,dc=example.com,dc=com
ldap://host1.domain20.example.com 用户认证方式ldap

```
# yum install -y sssd krb5-workstation authconfig-gtk
# authconfig-gtk
dc=, dc=
# getent ldapuser*
```

14) 自动挂载用户家目录/ruser/ldapuser20 挂载/ruser

```
# getent passwd ldapuser1
/ruser/ldapuser20
# showmount -e classroom.example.com
classroom.example.com:/ruser
# yum install -y nfs-utils
# systemctl start nfs-secure;systemctl enable nfs-secure
# yum install -y autofs
# vim /etc/auto.master.d/demo.autofs
/ruser /etc/auto.direct
#vim /etc/auto.direct
ldapuser20 -rw, sync, sec=krb5p classroom.example.com:/ruser
# systemctl enable autofs
# systemctl start autofs
# ssh ldaperuser@localhost
# pwd
```

15) 扩展

用户名: ldapuserX 密码 kerberos

Classroom.example.com 正共享/server/guests

DesktopX 挂载点 /home/guests/ldapuserX

```
# showmount -e classroom.example.com
classroom.example.com:/server/guests
# yum install -y autofs
# vim /etc/auto.master.d/home.autofs
/home/guests/ /etc/auto.direct
#vim /etc/auto.direct
* -rw, sync classroom.example.com:/server/guests/&
# systemctl enable autofs
# systemctl start autofs
# ssh ldapuser@localhost
# pwd
```

16) NTP 客户端配置

```
# yum install -y system-config-date
# system-config-date
```

17) 从[ftp://server1.domain20.example.com/pub/x86_64/rhce/station.html](http://server1.domain20.example.com/pub/x86_64/rhce/station.html) 下载到本地/var/www/html目录下, 通过<http://station.domain20.example.com>以访问.

```
# wget -O ftp://server1.domain20.example.com/pub/x86\_64/rhce/station.html
# cp station.html /var/www/html/index.html
# ls -Z /var/www/html
# vim /etc/httpd/conf/httpd.conf
ServerName station.domain20.example.com
# systemctl start httpd
# systemctl enable httpd
# firewall-cmd --permanent --add-service=http
# firewall-cmd reload
# firefox http://station.domain20.example.com
```

18) 查找所有属于普通用户 samon 的文件, 并移动到/root/finder

```
# find / -user samon
# mkdir /root/finder
# mv /home/samon/file1 /root/finder
```

19) 扩展

```
# find / -iname '*message*'    i不区分大小写
# find -group student
# find -uid 1000
# find -gid 1000
# find / -user root -group mail 查找root用户和mail组拥有的文件
# find /home -perm 764
查找用户具有读写执行权限，组有读写，其他人只读权限的文件
# find /home -perm -324 查找用户至少有写执行，并且组至少写，并且其他人至少只读权限的文件
# find /home -perm /442 查找用户具有读权限，或者组至少读，或者其他用户至少读
# find -size 10M 文件大小等于10M
# find -size +10G 文件大于10G
# find -size -10K 小于10K
# find / -min 120 正好120分钟以前更新的所有文件
# find / -min +200 200分钟以前修改过的文件
# find / -min -150 不到150分钟以前修改的文件
# find /etc -type d 查找所有的目录
# find / -type l 查找所有的软连接
# find / -type b 查找所有的块设备的列表
# find / -type f -links +1 查找硬链接数大于1的所有普通文件
# find / -name "[A-Z]*" 以大写字母开头的文件
# find . -name "[a-z][a-z][0-9][0-9].txt" -print
查找以两个小写字母和两个数字开头的txt文件
```

20) 查找/var/share/doc/words中字符串、并复制到/root/linux.txt

```
# grep 'Constitution' readme.txt >> /root/linux.txt
```

21) 扩展

```
# ps aux | grep '^l' 以l开头的
# ps aux | grep 'l$' 以l结尾
# ps aux | grep 'c.\{2\}t' \{2\} 匹配中间2个字符
# ps aux | grep -v 'sleep' 不包含
# grep -r 递归
# grep -A 3 显示表示式匹配的之后行数
# grep -B 3 显示表示式匹配的之前行数
# ps aux | grep -e sleep -e bash

grep -i pattern files : 不区分大小写地搜索。默认情况区分大小写，
grep -l pattern files : 只列出匹配的文件名，
grep -L pattern files : 列出不匹配的文件名，
grep -w pattern files : 只匹配整个单词，而不是字符串的一部分(如匹配' magic'，
```

而不是' magical'),
grep -C number pattern files : 匹配的上下文分别显示[number]行,
grep pattern1 | pattern2 files : 显示匹配 pattern1 或 pattern2 的行,
grep pattern1 files | grep pattern2 : 显示既匹配 pattern1 又匹配 pattern2 的行。

grep -n pattern files 即可显示行号信息

grep -c pattern files 即可查找总行数

这里还有些用于搜索的特殊符号:

\< 和 \> 分别标注单词的开始与结尾。

例如:

grep man * 会匹配 'Batman'、'manic'、'man' 等,

grep '\<man' * 匹配'manic' 和'man', 但不是'Batman',

grep '\<man\>' 只匹配'man', 而不是'Batman' 或'manic' 等其他的字符串。

'^': 指匹配的字符串在行首,

'\$': 指匹配的字符串在行尾,

1、 参数:

-I : 忽略大小写

-c : 打印匹配的行数

-l : 从多个文件中查找包含匹配项

-v : 查找不包含匹配项的行

-n: 打印包含匹配项的行和行标

2、 RE (正则表达式)

\ 忽略正则表达式中特殊字符的原有含义

^ 匹配正则表达式的开始行

\$ 匹配正则表达式的结束行

\< 从匹配正则表达式的行开始

\> 到匹配正则表达式的行结束

[] 单个字符; 如[A] 即A符合要求

[-] 范围 ; 如[A-Z]即A, B, C一直到Z都符合要求

. 所有的单个字符

* 所有字符, 长度可以为0

22) 设置用户natasha 对目录/home/cntrs有2770 ,设置manager 组用户对目录有读写执行权限, 其他人没有权限 (root除外)

```
# chmod 2770 /home/cntrs
```

23) 扩展

```
Setuid = u+s 4
```

```
Setgid = g+s 2
```

```
Sticky = o+t 1
```

24) 创建一个 512m 的 swap，并实现开机自动挂载

```
# fdisk /dev/sdb
P
t
2 - 选择对应的 partition
L
82
w
# partprobe
# mkswap /dev/sdb2
# free          (此时 swap 空间是原有的)
# swapon -s      (此时不会有新建的 swap)
# swapon /dev/sdb2 (启动新建的交换空间)
# swapon -s      (可以查看到新的 swap)
# free          (可以看到 swap 空间增加 )
# swapoff /dev/sdb2      (禁用新 swap)
# blkid /dev/sdb2
  UUID="868aa45b-d4e9-4f6e-a7e8-493cecc99f83"
# vim /etc/fstab
  UUID=868aa45b-d4e9-4f6e-a7e8-493cecc99f83          swap          swap          defaults
0 0
# swapon -a      (使用刚添加的/etc/fstab 测试启动交换空间)
# swapon -s      (验证)
重启后机器，使用 swapon -s 验证
# swapon -s
```

25) 创建一个 VG 名字为 wgroup。一个 LV 名字为 wshare, 要求 PE 大小为 8M LV 中的 PE 个数为 100 个，格式化为 vfat ， 并设置开机自动挂载到/mnt/wshare

```
# fdisk /dev/sdb
- t
- 1
- L
- 8e (LVM)
- t
- 3
  8e
  w
# pvcreate /dev/sdb1 /dev/sdb3
# pvdisplay
```



```

# vgcreate -s 8M wgroup /dev/sdb1 /dev/sdb3          (-s 8M 根据题目要求确认)
# vgdisplay
    PE Size              8.00 MiB
# lvcreate -l 10 -n wshare wgroup                    (-l 根据题目要求)
# lvdisplay
    Current LE           10
# mkfs.vfat /dev/wgroup/wshare
# mkdir -p /mnt/share
# mount /dev/wgroup/wshare /mnt/share/
# df -hT
/dev/mapper/wgroup-wshare vfat      80M    0   80M   0% /mnt/share
# umount /mnt/share
# vim /etc/fstab
/dev/wgroup/wshare      /mnt/share      vfat      defaults 1      2

```

下午考试

1) 两台SELINUX 启动设置enable

```
# vim /etc/selinux/config
SELINUX=enforcing
# getenforce
# setenforce 0    (Permissive)
# setenforce 1    (Enforcing)
```

2) 配置 SSH 访问

按以下要求配置 SSH 访问:

用户能够从域 group3.example.com 内的客户端通过 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

```
# vim /etc/hosts.allow
sshd : 193.168.181.231
# vim /etc/hosts.deny
sshd : 192.168.122.29
```

3) 扩展

```
# vim /etc/ssh/sshd_config
PasswordAuthentication no
# 重启 sshd 服务
```

4) 自定义用户环境

在系统system1和system2上创建自定义命令名为qstat此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

```
# vim /etc/profile (vim /etc/bashrc)
alias qstat='/bin/ps -Ao pid,tt,user,fname,rsz'
# . /etc/profile
```

5) 扩展

去除别名

```
# unalias qstat
```

6) 配置端口转发:

在系统system1 配置端口转发, 要求如下:

在172.24.3.0/24网络中的系统, 访问system1的本地端口5423将被转发到80

```
# firewall-config
```

source addresses.

Services Ports Masquerading **Port Forwarding**

Add entries to forward ports either from one port to another on the same system or from one system to another system. Forwarding to another system is masqueraded. Port forwarding is IPv4 only.

Port	Protocol	To Port	To Address
5432	tcp	80	

Port Forwarding

Please select the source and destination options according to your needs.

Source

Protocol: tcp

Port / Port Range: 5432

Destination

If you enable local forwarding, you have to specify a port. This port has to be different to the source port.

☒ Local forwarding

☐ Forward to another port

IP address:

Port / Port Range: 80

Cancel OK

7) 配置链路聚合

在system1.group3.example.com和system2.group3.example.com之间按以下要求配置一个链路:

此链路在system1使用下面的地址172.16.3.20/255.255.255.0

此链路在system2使用下面的地址172.16.3.25/255.255.255.0

System1上做

```
# ip link
```

```
# nmcli connection add type team con-name team1 ifname team1 config '{"runner" : "activebackup"}'
```

—这块要注意 activebackup 以备份方式, broadcast 传输来自所有端口的每个包
Roundrobin 轮循方式 loadbalance 监控流量, 哈希均衡, 看题目要求, 作相应改动。

```

# nmcli connection show
# nmcli connection modify team1 ipv4.addresses '172.16.3.20/24'
# nmcli connection modify team1 ipv4.method manual
# nmcli con add type team-slave con-name team1-port1 ifname eth1 master team1
# nmcli con add type team-slave con-name team1-port2 ifname eth2 master team1
# ip a      -- 看到team1 ip显示为172.16.3.20/24即可
# nmcli connection show 看到列表device列eth1和eth2都要有，才表示聚合成功
# nmcli con up team1
# teamdctl team1 state
Server2上做
# nmcli con add type team con-name team1 ifname team1 config
  '{ "runner": { "name": "activebackup" } }'
# nmcli con mod team1 ipv4.addresses '172.16.3.25/24'
# nmcli con mod team1 ipv4.method manual
# nmcli con add type team-slave con-name team1-port1 ifname eth1 master team1
# nmcli con add type team-slave con-name team1-port2 ifname eth2 master team1
# ip a
# nmcli con up team1
# teamdctl team1 state

```

8) 配置IPv6地址

在您的考试系统上配置接口 eth0 使用下列 Ipv6 地址：

A system1 上的地址应该是 2003:ac18::305/64

B system2 上的地址应该是 2003:ac18::30a/64

C 两个系统必须能与网络 2003:ac18/64 内的系统通信

D 地址必须在重启后依旧生效

E 两个系统必须保持当前的 Ipv4 地址并能通信。

方法1 采用 nm-connection-editor 图形化界面配置

在 system1 上做

```

# nmcli connection add con-name eth0 type ethernet ifname eth0
# nmcli connection mod eth0 ipv6.addresses "2003:ac18::305/64"
# nmcli connection modify eth0 ipv6.method manual
# nmcli con up eth1
# ping6 2003:ac18::30a

```

System2 上做

```

# nmcli con add con-name eth1 type ethernet ifname eth0
# nmcli con mod eth1 ipv6.addresses '2003:ac18::30a/64'
# nmcli con mod eth1 ipv6.method manual
# nmcli con down eth1
# nmcli con up eth1
# ping6 2003:ac18::305

```

9) 配置本地邮件服务

A 这些系统不接收外部发送来的邮件

B 在这些系统上本地发送的任何邮件都会自动路由到 `server1.group3.example.com`

C 在这些系统上发送的邮件显示来自于 `group3.example.com`

您可以通过发送邮件到本地用户 ‘arthur’ 来测试您的配置，系统

`server1.group3.example.com` 已经配置把此用户的邮件转到下列 URL

`http://server1.group3.example.com /received_mail/3`

此题在两台服务器上做的步骤是相同的

方法 1 修改 `/etc/postfix/main.cf`

方法 2

```
# postconf -e "inet_interfaces = loopback-only"
# postconf -e "relayhost = [server1.group3.example.com]"
# postconf -e "myorigin = group3.example.com"
# postconf -e "mynetworks = 127.0.0.0/8 [::1]/128" -- 应该不需要修改
# postconf -e "mydestination = " (此处没有内容，表示不接收外部发来的邮件)
# postconf -e "local_transport = error: local delivery disabled"
# systemctl restart postfix
# systemctl enable postfix
# firewall-cmd --permanent --add-service=smtp
# firewall-cmd -reload
# mail -u Arthur
# mail -s "test" arthur@group3.example.com -s 是主题
# firefox http://server1.group3.example.com/received_mail/3
# mutt -f imapX.example.com
```

10) 通过SMB共享目录

在 `system1` 上配置 SMB 服务，您的 SMB 服务器必须是 STAFF 工作组的一个成员

共享 `/common` 目录共享名必须为 `common`，只有 `domain4.exmaple.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 `andy` 必须能够读取共享中的内容，如果需要的话，验证的密码是 `redhat`

其步骤为：

装包，建目录，改目录类型值，修改配置文件，提升用户为 `smb` 用户，启服务，开端口。

——在 `system1` 上做——

```
# yum -y install samba samba-client cifs*
```

包最好一个个装，才知道有没有装上

```
# mkdir /common
```

```
# chcon -R -t samba_share_t /common 改类型值（注意 R 一定在前）
```

```
# setenforce 1
```

```
# 或者 semanage fcontext -a -t samba_share_t /common
```

```
# restorecon -vvFR /common/
```

```

# vim /etc/samba/smb.conf
[global]
workgroup = STAFF
[common] -- 添加
path = /common
hosts allow = 172.24.3. 如果域名, 则是: .domain4.example.com
browseable = yes
write list=andy 如果 andy 可写, 则加这句, 这句不是必须, 看题目
# useradd andy 看结果, 如果已经有, 则不要建立, 如果没有, 则要建立
# echo redhat | passwd --stdin andy 用这个的目的, 是知道自己打的是什么密码,
# smbpasswd -a andy 如果是组, 则 smbpasswd -a +andy
# systemctl start smb
# systemctl enable smb
# firewall-cmd --permanent --add-service=samba
# firewall-cmd --reload
# chmod o+w /common 如果要给写权限, 要把/common 给上其他人的写权限

——在 system2 上测试
# yum -y install samba-client cifs-utils (包一个个装, 才知道装没有装成功)
# useradd andy
# smbclient //172.24.3.0/common -U andy //登录测试

```

11) 配置多用户SMB挂载

在 system1 共享通过 SMB 目录/devops 满足以下要求

- A 共享名为 devops
- B 共享目录 devops 只能被 group3.example.com 域中的客户端使用
- C 共享目录 devops 必须可以被浏览
- D 用户 kenji 必须能以读的方式访问此共享, 访问密码是 atenorth
- E 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 system2.group3.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

```

——在 system1 上做——
# mkdir /devops
# chcon -t samba_share_t /devops
# vim /etc/samba/smb.conf
在 share 下添加
[devops]
path = /devops
hosts allow = 172.24.3.
browseable = yes
writable = no

```

```

        write list = chihiro
:wq
# chmod o+w /devops  共享目录 devops 必须可以被浏览
# useradd kenji  看结果, 如果已经有, 则不要建立, 如果没有, 则要建立
# echo atenorth | passwd --stdin kenji  这时 kenji 可以用 esc 键+ . 来完成
# useradd chihiro  看结果, 如果已经有, 则不要建立, 如果没有, 则要建立
# echo atenorth | passwd --stdin chihiro  这时 kenji 可以用 esc 键+ . 来完成
# smbpasswd -a kenji
# smbpasswd -a chihiro
# systemctl restart smb

——在 system2 上做——
# yum -y install cifs*
# useradd kenji  看结果, 如果已经有, 则不要建立, 如果没有, 则要建立
# echo atenorth | passwd --stdin kenji  useradd chihiro
# echo atenorth | passwd --stdin chihiro
# mkdir /mnt/dev
# smbclient //172.24.3.0/ -U kenji //测试下
# vim /etc/fstab
//system1/devops /mnt/dev cifs
    defaults,multiuser,username=kenji,password=atenorth,sec=ntlmssp 0 0
# mount -a
# su - chihiro
# cifscreds add system1
# su - kenji
# cifscreds add system1 -u chihiro

```

12) 配置NFS服务

在 system1 配置 NFS 服务, 要求如下:

- A 以只读的方式共享目录/public 同时只能被 group3.example.com 域中的系统访问
- B 以读写的方式共享目录/protected 能被 group3.example.com 域中的系统访问
- C 访问 /protected 需要通过 kerberos 安全加密, 您可以使用下面的 URL 提供的密钥
http://host.group3.example.com/materials/nfs_server.keytab
- D 目录 /protected 应该包含名为 project 拥有人为 krishna 的子目录
- E 用户 krishna 能以读写方式访问 /protected/project

```

# yum install -y nfs-utils
# mkdir /public
# mkdir -p /protected/project - 看题目要求
# chcon -Rt public_content_t /protected
# ls -ldZ /protected/project 看看类型值改了没有
# chcon -t pulbic_content_t /public

```

```
# wget -O /etc/krb5.keytab
http://host.group3.example.com/materials/nfs_server.keytab 注意大写的 O
# cp /etc/sysconfig/nfs /etc/sysconfig/nfs_bak 备份一个配置文件，在做实验过程中发现有配错现象，但无法恢复原来的，所以保险一些。
# vim /etc/sysconfig/nfs
    RPCNFSDARGS=" -V 4.2"
# vim /etc/exports
/public 172.24.3.0/255.255.255.0(ro, sync)如果写成域为: .group3.example.com
/protected 172.24.3.0/255.255.255.0(sec=krb5p, rw)
# exportfs -r
# chown krishna /protected/project - /public 要求只读，所以不用更改属组
                                chown :nfsnobody /public
                                chmod g+w /public
# systemctl start nfs-server nfs-secure-server
# systemctl enable nfs-server nfs-secure-server
# firewall-cmd --permanent --add-service=nfs
# firewall-cmd --reload
# showmount -e 127.0.0.1
```

13) 挂载一个NFS共享

在 system2 上挂载一个来自 system1.group3.example.com 的 NFS 共享，并符合下列要求：

A /public 挂载在下面的目录上/mnt/nfsmount

B /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 如下：

http://host.group3.example.com/materials/nfs_client.keytab

C krishna 能够在/mnt/nfssecure/project 上创建文件

D 这些文件系统在系统启动时自动挂载

```
# yum install -y nfs-utils
# wget -O /etc/krb5.keytab
http://host.group3.example.com/materials/nfs\_client.keytab
# mkdir /mnt/nfsmount
# mkdir /mnt/nfssecure
# vim /etc/fstab
system1:/public /mnt/nfsmount nfs defaults 0 0
system1:/protected /mnt/nfssecure nfs defaults, sec=krb5p, v4.2 0 0
# 手动挂载 mount -o sec=krb5p, v4.2 serverX.example.com/protected
/mnt/nfssecure
# systemctl restart nfs-secure
# systemctl enable nfs-secure
# mount -a 如果出错，查看日志、上课实机，需要安装 kerberos 认证
```


14) 实现一个Web服务器

为站点http://system1.domain1.example.com 创建一个web服务器在system1

15) 配置虚拟主机

在system1上扩展你的web服务器，为站点 http://www.group3.example.com 创建一个虚拟机

```
——在 system1 上做——
yum install -y wget httpd links （建议一个一个安装）
useradd floyd
mkdir -pv /var/www/virtual
cd /var/www/virtual
wget -O index.html http://server1.group3.example.com/materials/www.html
cd /var/www/html
wget -O index.html http://rhgls.group3.example.com/materials/station.html
vim /etc/httpd/conf/zh.conf
输入：
<VirtualHost *:80>
    DocumentRoot /var/www/html
    ServerName system1.group3.example.com
</VirtualHost>
按 esc，光标移到第一行，4yy，光标移到最后一行，p，会复制上面四行，再进行修改：
<VirtualHost *:80>
    DocumentRoot /var/www/virtual
    ServerName www.group3.example.com
</VirtualHost>
保存退出，注意，这个文件不区分大小写
systemctl restart httpd
systemctl enable httpd
firewall-cmd --permanent --add-port=80/tcp
firewall-cmd --permanent --add-port=443/tcp
firewall-cmd --reload
setfacl -m u:Floyd:rwX /var/www/virtual 第14题的目录写权限
以下可用图形化界面做，firewall-config 在有 rich 的那个 tab 做。以下保留，
可以看着这个来配置。
firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source
address=172.25.1.0/24 service name="http" log level=notice reject'
firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source
address=172.24.3.0/24 service name="http" log level=notice accept'
```

——在 system2 上, 再开两个终端——

一个终端 links [http:// system1.group3.example.com](http://system1.group3.example.com), 看到内容, 关掉

另一个终端 links [http:// www.group3.example.com](http://www.group3.example.com), 看到内容, 关掉

16) 配置安全web服务

为站点<http://system1.group3.example.com> 配置TLS加密, 一个已签名的证书从<http://host.group3.example.com/materials/system1.crt> 获取, 此证书的密钥从<http://host.group3.example.com/materials/system1.key> 获取, 此证书的签名授权信息从<http://host.group3.example.com/materials/group3.crt> 获取

```
# yum -y install mod_ssl httpd(一个一个安装, 能看到安装成功没有)
# cd /etc/pki/tls/certs 进入到这个目录, 准备下载各证书, 把证书下到这个目录
的好处是, 改配置文件时, 改的东西比较少。
# wget http://host.group3.example.com/materials/system1.crt (证书)
# wget http://host.group3.example.com/materials/system1.key (密钥)
# wget http://host.group3.example.com/materials/group3.crt (公钥)
# chmod 0600 system1.key 改密钥属性, 让别人不能读写
# cd /etc/httpd/conf.d 进行配置文件目录
# ls 看一下, 有个 ssl.conf 文件, 为防止出错, 复制一下
# cp ssl.conf sslbak
# vim ssl.conf
    在命令行状态下, 打 /SSLCer 回车,
    找到: SSLCertificateFile 这个公钥, 看下相对应的路径, 最后文件改成我们下
    的公钥名称 即/etc/pki/tls/certs/group3.crt
    按 esc, n
    找到: SSLCertificatekeyFile 这个是私钥, 看下路径, 改最后那个表示私钥文件
    的字符串, 变成我们下的那个公钥文件 /etc/pki/tls/cert/system1.key
    按 esc, n
    接下来这个是证书, 被 #引掉, 把# 去掉, 同上, 改路径和文件名, 改成如下:
    SSLCertificateChainFile /et/pki/tls/certs/system1.crt
    命令行状态, Shift+g 到最后, 按 o
    <VirtualHost *:443>
        DocumentRoot /var/www/html
        ServerName system1.group3.example.com
    </VirtualHost>
    <Directory /var/www/html>
        Require all granted
    </Directory>
    保存退出
```

```
# systemctl restart httpd
# Systemctl enable httpd
# firewall-cmd -perment - add-service=http - add-service=https
# firewall-cmd -perment - add-service=http - add-service=http
# firewall-cmd -reload
——在 system2 上——
    首先下证书
# wget http://host.group3.example.com/materials/system1.crt (证书)
# yum install -y firefox
# firefox &
# 进入 firefox 浏览器后, 选择菜单 edit -- preferences
# 选择 advanced 选项框, 把 greneral 都选上, 再选 Certificates
# Select one automatically 选上, view certificates, authorities
点 import, 选择 desktop, 选择下的证书, 点 open , 选中 Trust this CA to identify
websits 点击 ok!ok!close
# 然后浏览器中输入 http:// system1.group3.example.com 能看到结果
```

17) 配置web内容的访问

在你的system1 上的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://server1.group3.example.com/materials/private.html> 下载一个文件副本到这个目录, 并且重命名为 index.html

不要对这个文件的内容做任何修改

从 system1 上, 任何人都可以浏览 private的内容, 但是从其它系统不能访问这个目录的内容

```
——在 system1 上做——
# mkdir /var/www/html/private
# cd /var/www/html/private
# wget -O index.html http://server1.example.com/materials/private
# cd /etc/httpd/conf.d
#vim zh.conf 这是之前做 12 和 14 题时定义的配置文件, 在/var/www/html 的那个
virtualhost 窗口中, 加入
    <VirtualHost>
    .....
    <Directory /var/www/virtual/private >
        Order allow,deny
        Allow from 172.25.10.11(这是 system1 服务器地址)
    </Directory>
</VirtualHost>
```

18) 实现动态Web内容

在 system1 上配置提供动态Web内容，要求如下：

动态内容由名为 alt.group3.example.com 的虚拟主机提供

虚拟主机侦听在端口8909

从 <http://server1.group3.example.com/materials/webinfo.wsgi> 下载一个脚本，然后放在适当的位置

```
——在 system1 上做——  
# yum -y install mod_wsgi  
# mkdir /var/www/alt  
# cd /var/www/alt  
# wget http://server1.group3.example.com/materials/webinfo.wsgi  
# vim /etc/httpd/conf.d/zh.conf  
在这个配置文件中加入：  
listen 8909  
    <VirtualHost *:8909>  
        ServerName alt.group3.example.com  
        WSGIScriptAlias / /var/www/alt/webinfo.wsgi  
    </VirtualHost>  
# semanage port -a -t http_port_t -p tcp 8909  
    //这步非常重要，如果不加入，httpd 是起不来的  
# systemctl restart httpd  
# setenforce 1  
# firewall-cmd --permanent --add-port=8909/tcp  
# firewall-cmd -reload  
——在 system2 上打开 firefox 查看——
```

19) 创建一个脚本

要在 system1 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

```
# vim /root/foo.sh  
#!/bin/bash  
case $1 in  
    redhat)  
        echo fedora  
        ;;  
    fedora)  
        echo redhat  
    *)  
        echo "Invalid argument: $1" >> /dev/stderr  
    esac
```

```

        ;;
    *)
        echo "/root/foo.sh redhat|fedora"
        ;;
esac
# chmod u+x /root/foo.sh

```

20) 创建一个添加用户的脚本

在system1上创建一个脚本，名为/root/batchusers，此脚本能实现为系统system1创建本地用户，并且这些用户名来自一个包含用户名列表的文件，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

```

——在 system1 上做——
# cd /root
# wget http://server1.group3.example.com/materials/userlist
# vim /root/batchusers
#!/bin/bash
if [ $# -eq 0 ];then
    echo "Usage: /root/batchusers"
    exit 1
elif [ ! -f $1 ];then
    echo "Input file not found"
    exit 1
fi
for I in $( cat $1 )
do
    useradd -s /bin/false $I
done
# chmod +x /root/batchusers
# ./root/batchusers &

```

21) 配置iSCSI服务器

配置system1提供一个iSCSI服务，磁盘名为iqn.2014-09.com.example.group3:system1，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3G

此服务只能被system2.group3.example.com访问

```

在system1上做
# yum -y install targetcli
# fdisk -l 看分区情况
# fdisk /dev/vda 先分一个 3G 的盘出来
# n 回车 回车 回车 + 3G 最后 w
# partprobe
# targetcli
# /backstores/block create block1 /dev/vda3 //此处的 vda3 是我在上面分出来的
# /iscsi create iqn.2014-09.com.example.group3:system1 (粗线部分服务器名)

```

```
# /iscsi/iqn.2014-09.com.example.group3:system1/tpgl/acls/ create
iqn.2014-09.com.example.group3:system2 （粗线部分客户机名）
# /iscsi/iqn.2014-09.com.example.group3:system1/tpgl/luns create
/backstores/block/block1
# /iscsi/iqn.2014-09.com.example.group3:system1/tpgl/portals/ create
172.24.3.5 （粗线部分是服务器ip地址，最好这样写）
# systemctl enable target
# systemctl start target
# firewall-cmd --permanent --add-port=3260/tcp
# firewall-cmd -reload
```

22) 配置iSCSI的客户端

配置system2使其能连接在system1的上提供的iqn.2014-09.com.example.group3:system1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2100MB的分区，并格式化为ext4

```
# yum -y install iscs*
# vim /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.2014-09.com.example.group3:system2(客户机，标识自己)
# systemctl restart iscsi
# systemctl enable iscsi
# iscsiadm -m discovery -t st -p system1 (服务器)发现服务器
# iscsiadm -m node -T iqn.2014-09.com.example.group3:system1 -l
（此块如果登录不了，要重新启动电脑，再起服务 systemctl restart iscsi, systemctl
enable iscsi, 再发现，再登录）
# fdisk -l 看到有 s 开头的盘，如 sda
fdisk /dev/sda 分一个 2100M 的分区
# partprobe
# mkfs.ext4 /dev/sda1
# mkdir /mnt/data
# vim /etc/fstab
/dev/sda1 /mnt/data ext4 _netdev 0 0
# mount -a
```

23) 配置一个数据库

在system1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://server1.group3.example.com/materials/users.mdb.

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

```
# yum install -y mariadb
```

```
# yum install -y mariadb-client
# wget http://server1.group3.example.com/materials/users.mdb
# systemctl start mariadb
# systemctl enable mariadb
# mysql
MariaDB [(none)]> show databases;
MariaDB [(none)]> create database Contacts;
MariaDB [(none)]> use Contacts
MariaDB [Contacts]> source /root/users.mdb
MariaDB [Contacts]> show tables;
MariaDB [Contacts]> grant select on Contacts.* to Raikon@localhost identified
by 'atenorth';
Exit
# mysql_secure_installation //使用向导来设置root密码。
输入当前密码并回车
Y
密码
然后一路Y
# mysql -u root -p
```

24) 数据库查询

在系统system1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：

```
# mysql -u root -p
# MariaDB [(none)]> use Contacts;
# MariaDB [Contacts]> show tables;
# MariaDB [Contacts]> select * from &&& where ;
```