RHCE

姓名：坏坏

实验时间：2020年1月2日

实验类型：RHCE考证

实验环境：Redhat 7.0

一、安装红帽 RHEL7.X 操作系统；

/boot 500M

Swap 2048M

/ 20G

/dev/mapper/vg0-lvhome 512M

/dev/mapper/vg0-lv0 214M

二、考试环境概述

最新的红帽 RHEL7 官方 RHCSA/RHCE 考试秉承了之前考试的一贯形式，上机实战操作，学员根据考试机所提供的考试要求，独立操作完成，将考试要求的结果在虚拟机（KVM）中完成，但最新的 RHCE7.0 相对之前的 RHCE5/6.X 考试而言，最新的认证体系中加入虚拟化、集群、存储、web 安全，SQL 搭建配置查询等等，难度系数再次增大，自然含金量和社会认可度也再一次的大幅提升，作为参加红帽官方 RHCE 考试学员，一定要认真备考，做到万无一失！RHCE7.0 考试下午环境大概为：

1. 考试使用电脑真实机桌面将有2台 exam 考试虚拟机控制端为 Desktop 和 Server，分别可以控制各虚拟机的启动，关闭，重启等，考题同时也在真实机的桌面上；

2. server 和 desktop 同属于你的考试域 example.com 中的成员主机，考试还有一个攻击域my133t.org(172.25.0.0/16)网络，所有的环境信息请查阅考试环境说明；

3. 此次讲解的 RHCE7.0 考试环境，我们使用 VMwareWorkstation11 提供整套解决方案，下面是配置安装系统前的硬件配置需求：

4. 考试的网络环境为：

a. Server 和 Desktop 两台机器的root密码：tangkai
b. 上午 RHCSA 考试需要用户破密码，下午 RHCE 考试则不需要；
c. 主机名(Hostname)：server0.example.com
d. IP地址(IPaddress)：172.25.0.11
e. 子网掩码(Netmask)：255.255.255.0
f. 网关(Gateway)：172.25.0.254
g. 名称服务(Nameserver)：172.25.0.254

注意：下午的考试不需要额外配置 IP 等操作，需要完成 YUM 的配置！记得进 Server 和 Desktop 分别查看相对应的 IP、DNS、网关及主机名等相关信息！

当查看并配置完成后，点击下面的链接：http://classroom.example.com

三、考试时间

- 考试安排：
  - 上午RHCSA考试时间：2.5 小时　　满分 300分 210 分及格
  - 下午RHCE考试时间：3.5 小时　　满分 300分 210 分及格

四、配置系统信息

RHCE 不需要破解密码

五、配置 YUM

分别在 server 和 desktop 上配置 YUM ，指向：

http://classroom.example.com/content/rhel7.0/x86_64/dvd/

```
[root@desktop0 ~]# vim /etc/yum.repos.d/yum.repo
[Base]
name=RHCE
```

```
baseurl=http://classroom.example.com/content/rhel7.0/x86_64/dvd/
gpgcheck=0
enabled=1
```

六、配置 SELinux

SElinux 有三种模式，请将 server 与 desktop 运行于强制模式

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

七、配置 SSH

用户能够从域 exampl.com 内的客户端通过 SSH 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

```
//关闭iptables、ebtables，并禁止开机自启
[root@server0 ~]# systemctl disable iptables
[root@server0 ~]# systemctl disable ebtables
[root@server0 ~]# systemctl stop ebtables
[root@server0 ~]# systemctl stop iptables
//设置firewall开机自启并重新启动
[root@server0 ~]# systemctl enable firewalld
[root@server0 ~]# systemctl restart firewalld
//查看防火墙规则
 [root@server0 ~]# firewall-cmd --list-all
 //移除对ssh服务放行的的规则，重新加载，查看结果
 [root@server0 ~]# firewall-cmd --permanent --remove-service=ssh
 //添加防火墙富规则，放行指定域
 [root@server0 ~]# firewall-cmd --permanent --add-rich-rule 'rule family=ipv
4 source address=172.25.0.0/24 service name=ssh accept'
 [root@server0 ~]# firewall-cmd --reload
 [root@server0 ~]# firewall-cmd --list-all
```

八、命令别名及 IP 转发

在系统 server 和 desktop 上创建自定义命令为tk，此自定义命令将执行 /bin/ps aux，此命令对系统中所有用户有效

开启 IP 转发功能

```
[root@server0 ~]# echo "alias tk='/bin/ps aux'" >> /etc/bashrc
[root@server0 ~]# . /etc/bashrc
```

```
[root@server0 ~]# vim /etc/sysctl.conf
net.ipv4.ip_forward = 1
[root@server0 ~]# sysctl -p/etc/sysctl.conf
```

九、端口转发

在 server 上配置端口转发，在 172.25.0.0/24 中的系统，访问 server 的本地端口 9527 将被转发到 80，此设置永久生效

```
[root@server0 ~]# firewall-cmd --list-all
[root@server0 ~]# firewall-cmd --permanent --add-rich-rule 'rule family=ipv4 source address=172.25.0.0/24 forward-port port=9527 protocol=tcp to-port=80'
[root@server0 ~]# firewall-cmd --reload
[root@server0 ~]# firewall-cmd --list-all
```

十、聚合网络

在 server 和 desktop 之间配置链路聚合

此链路使用接口 slave1 和 slave2

此链路在一个接口失效后，仍然能工作

此链路在 server 上使用地址 192.168.0.1/24

此链路在 desktop 上使用地址 192.168.0.2/24

此链路在系统重启后依然保持正常状态

```
[root@desktop0 ~]# nmcli dev //查看所有网卡
//添加team
[root@desktop0 ~]# nmcli connection add type team con-name team0 ifname team0 config '{"runner":{"name":"activebackup"}}'
[root@desktop0 ~]# nmcli connection show //查看是否添加team0成功
//修改team0的IP
[root@desktop0 ~]# nmcli connection modify team0 ipv4.addresses 192.168.0.2/24 ipv4.method manual connection.autoconnect yes
//添加slave1
[root@desktop0 ~]# nmcli connection add type team-slave con-name slave1 ifname eth1 master team0
Connection 'slave1' (e9f49724-ba4d-46a0-982b-83d73a834d10) successfully added.
//添加slave2
[root@desktop0 ~]# nmcli connection add type team-slave con-name slave2 ifname eth2 master team0
```

```
Connection 'slave2' (096e23f1-60fe-48c7-ab9e-e3369b34a344) successfully added.
//启动team0
[root@desktop0 ~]# nmcli connection up team0
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/4)
//尝试ping通检查
[root@desktop0 ~]# ping 192.168.0.1
```

十一、IPv6 设置

在您的考试系统上配置接口,在你的默认网卡上使用如下IPv6地址

server 上的 IP 地址应该是 fd00:ba5e:ba11:10::1/64

desktop上的 IP 地址应该是 fd00:ba5e:ba11:10::2/64

两个系统必须能与网络 fd00:ba5e:ba11:10::cc 内的系统通信

地址必须在重启后依然生效

两个系统保持当前的 IPv4 地址并能通信

```
//修改默认网卡的ipv6地址
[root@server0 ~]# nmcli connection modify "System eth0" ipv6.addresses 'fd00:ba5e:ba11:10::1/64' ipv6.method manual connection.autoconnect yes
//重启网络服务
[root@server0 ~]# systemctl restart network
//查看是否有ipv6地址
[root@server0 ~]# ip a
//ping另一台客户机的ipv4地址和ipv6地址
[root@server0 ~]# ping 192.168.0.2
[root@server0 ~]# ping6 'fd00:ba5e:ba11:10::2'
```

```
[root@desktop0 ~]# nmcli connection modify "System eth0" ipv6.addresses 'fd00:ba5e:ba11:10::2/64' ipv6.method manual connection.autoconnect yes
[root@desktop0 ~]# systemctl restart network
[root@desktop0 ~]# ip a
[root@desktop0 ~]# ping 192.168.0.1
[root@desktop0 ~]# ping6 'fd00:ba5e:ba11:10::1'
```

十二、邮件服务

在 server 上配置邮件服务

这些系统不接受外部发来的邮件

在这些系统上本地发送任何邮件都会被路由到 classroom.example.com

从这些系统上发送的邮件显示来自于 example.com

您可以通过发送邮件到 harry 来测试您的配置

您可以通过访问http://classroom.example.com/email/harry来验证您的配置

发给harry的邮件同时能被natasha收到

```
[root@server0 ~]# rpm -qa | grep postfix //查看是否安装了邮件服务
postfix-2.10.1-6.el7.x86_64
[root@server0 ~]# systemctl restart postfix //重启服务
[root@server0 ~]# systemctl enable postfix //设置开机自启动
//修改配置文件
[root@server0 ~]# vim /etc/postfix/main.cf
myorigin = example.com //显示来自域
inet_interfaces = localhost
mydestination =
 mynetworks = 127.0.0.0/8
 relayhost = [classroom.example.com] //被路由到的地址
 [root@server0 ~]# systemctl restart postfix //重启服务
 //修改配置文件，使harry和natasha都可以收到邮件
 [root@server0 ~]# vim /etc/aliases
 harry: harry,natasha
 //将服务添加到防火墙规则中，使防火墙对邮件服务放行
 [root@server0 ~]# firewall-cmd --permanent --add-service=smtp
 success
 [root@server0 ~]# firewall-cmd --reload
 success
 [root@server0 ~]# firewall-cmd --list-all
```

十三、Samba 服务

在 server 上配置 SAMBA 服务

您的 samba 服务器必须是 STAFF 工作组的一个成员

共享 /common 目录，共享名为 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

保存用户natasha必须能够读取共享中的内容，如果需要的话，验证密码是：tangkai

```
//安装samba服务，设置开机自启并启动服务
```

```
[root@server0 ~]# yum -y install samba*
[root@server0 ~]# systemctl enable smb nmb
[root@server0 ~]# systemctl restart smb nmb
//修改配置文件
[root@server0 ~]# vim /etc/samba/smb.conf
workgroup = STAFF
//此处设置selinux，直接取消注释，也可也使用命令设置
 setsebool -P samba_enable_home_dirs on
chcon -Rt samba_share_t /common
 //最后加上以下内容
[common]
 path = /common
 browseable = yes
 valid users =natasha
//防火墙放行服务
[root@server0 ~]# firewall-cmd --permanent --add-rich-rule 'rule family=ipv
4 source address=172.25.0.0/24 service name=samba accept'
success
[root@server0 ~]# firewall-cmd --permanent --add-rich-rule 'rule family=ipv
4 source address=172.25.0.0/24 service name=samba-client accept'
success
[root@server0 ~]# firewall-cmd --reload
success
[root@server0 ~]# firewall-cmd --list-all

//设置selinux，如果修改了配置文件此处则不用再放行
[root@server0 ~]# getsebool -a | grep samba | grep dirs
samba_create_home_dirs --> off
samba_enable_home_dirs --> off
use_samba_home_dirs --> off
[root@server0 ~]# setsebool -P samba_enable_home_dirs 1
[root@server0 ~]# getsebool -a | grep samba | grep dirs
samba_create_home_dirs --> off
samba_enable_home_dirs --> on
use_samba_home_dirs --> off
//创建共享的目录，并切换上下文
[root@server0 ~]# mkdir /common
[root@server0 ~]# chcon -Rt samba_share_t /common/
```

```
//设置natasha用户的samba共享密码
[root@server0 ~]# id natasha
uid=1001(natasha) gid=1001(natasha) groups=1001(natasha)
[root@server0 ~]# smbpasswd -a natasha
```

十四、多用户 samba 挂载

在 server 上通过 samba 共享目录 /storage

共享名为 share

共享目录只能被 example.com 域内的客户端使用

共享目录 share 必须可以被浏览

用户 sarah 能以读的方式访问此共享，访问密码是 tangkai

用户 kitty 能以读写的方式访问此共享，访问密码是 tangkai

此共享永久挂载在 desktop 上的 /mnt/dev 目录，并使用用户 sarah 进行认证，任何用户可临时通过kitty来获得读写权限

```
//创建共享目录，切换上下文
[root@server0 ~]# mkdir /storage
[root@server0 ~]# chcon -Rt samba_share_t /storage/
//修改配置文件
[root@server0 ~]# vim /etc/samba/smb.conf
 [share]
 path = /storage
 browseable = yes
 valid users =sarah, kitty
  writable = no
  write list = kitty
//重启服务，创建用户并设置samba密码
[root@server0 ~]# systemctl restart smb nmb
[root@server0 ~]# useradd sarah
[root@server0 ~]# useradd kitty
[root@server0 ~]# smbpasswd -a sarah
[root@server0 ~]# smbpasswd -a kitty
//设置sarah的读权限，kitty的读写权限
[root@server0 ~]# setfacl -m u:sarah:r-x /storage/
[root@server0 ~]# setfacl -m u:kitty:rwx /storage/
[root@server0 ~]# getfacl /storage/
//创建挂载点，修改配置文件，挂载
```

```
[root@desktop0 ~]# mkdir /mnt/dev
[root@desktop0 ~]# vim /etc/fstab
//172.25.0.11/share /mnt/dev cifs multiuser,username=sarah,password=tangkai,sec=ntlmssp 0 0
[root@desktop0 ~]# mount -a
[root@desktop0 ~]# df -h
//验证权限
[root@desktop0 ~]# cd /mnt/dev
//客户端安装cifs-utils
[root@desktop0 ~]# yum -y install cifs-utils.x86_64
[root@desktop0 ~]# su - student
[student@desktop0 ~]$ cifscreds add -u kitty 172.25.0.11
Password:
//验证权限
[student@desktop0 ~]# cd /mnt/dev
```

十五、配置 NFS 服务

在 server 上配置 NFS

以只读的方式共享 /public ,同时只能被 example.com 内用户访问

以读写的方式共享 /protected 能被 example.com 内用户访问

访问 /protected 需要通过 kerberos 安全加密，您可以使用下边链接的秘钥：

http://classroom.example.com/pub/keytabs/server0.keytab

目录 /protected 应该包含名为 project 拥有人为 ldapuser12 的子目录

用户 ldapuser12 能以读写的方式访问 /protected/project

```
[root@server0 ~]# yum -y install nfs*
//创建共享目录，修改权限
[root@server0 ~]# mkdir /public /protected
[root@server0 ~]# vim /etc/exports
/public 172.25.0.0/24(ro)
/protected 172.25.0.0/24(rw,sec=krb5p)
//查看是否配置好ldap
[root@server0 ~]# su - ldapuser12
//配置时间同步，配置ldap
[root@server0 ~]# vim /etc/chrony.conf
server classroom.example.com iburst
[root@server0 ~]# systemctl restart chronyd
```

```
//创建目录，下载密钥
[root@server0 ~]# cd /etc/openldap/
[root@server0 openldap]# ls
certs ldap.conf schema
[root@server0 openldap]# mkdir cacerts
[root@server0 openldap]# cd cacerts/
[root@server0 cacerts]# wget http://classroom.example.com/pub/example-ca.crt
//配置kerberos认证，下载密钥
[root@server0 cacerts]# cd
[root@server0 ~]# authconfig-tui
[root@server0 ~]# su - ldapuser12
[root@server0 ~]# wget -O /etc/krb5.keytab http://classroom.example.com/pub/keytabs/server0.keytab
//创建project目录，更改属主
[root@server0 ~]# cd /protected/
[root@server0 protected]# mkdir project
[root@server0 protected]# chown -R ldapuser12 project/

//查看ldapuser12是否具有读写权限
[root@server0 protected]# ll
drwxr-xr-x. 2 ldapuser12 root 6 Dec 30 08:46 project
//修改配置文件，设置相关服务开机自启动，并启动相关服务
[root@server0 ~]# vim /etc/sysconfig/nfs
RPCNFSDARGS="-V 4.2"
[root@server0 ~]# systemctl enable nfs-secure nfs-secure-server nfs-server
[root@server0 ~]# systemctl restart nfs-secure nfs-secure-server nfs-server
//防火墙放行服务
[root@server0 ~]# firewall-cmd --permanent --add-rich-rule 'rule family=ipv4 source address=172.25.0.0/24 service name=nfs accept'
success
[root@server0 ~]# firewall-cmd --permanent --add-rich-rule 'rule family=ipv4 source address=172.25.0.0/24 service name=rpc-bind accept'
success
[root@server0 ~]# firewall-cmd --reload
success
```

在 desktop 上挂载来自于 server0 的 NFS 共享

/public 挂载在目录 /mnt/nfsmount 上

/protected 挂载在目录 /mnt/nfssecure ， 并使用安全的方式，秘钥：http://classroom.example.com/pub/keytabs/desktop0.keytab

用户 ldapuser12 能在 /mnt/nfssecure/project 上创建文件

这些文件系统在系统启动时自动挂载

```
//先验证是否安装好ldap
[root@desktop0 cacerts]# su - ldapuser12
//配置时间同步
[root@desktop0 cacerts]# vim /etc/chrony.conf
[root@desktop0 cacerts]# systemctl restart chronyd.service
//安装配置ldap
[root@desktop0 cacerts]# yum -y install authconfig* sssd* krb5*
[root@desktop0 cacerts]# cd /etc/openldap/
[root@desktop0 cacerts]# mkdir cacerts
 [root@desktop0 cacerts]# cd cacerts/
 [root@desktop0 cacerts]# wget http://classroom.example.com/pub/example-ca.crt
 [root@desktop0 cacerts]# authconfig-tui
 [root@desktop0 cacerts]# su - ldapuser12
 //创建挂载点
 [root@desktop0 ~]# mkdir /mnt/{nfsmount,nfssecure}
 [root@desktop0 ~]# ls /mnt/
 dev nfsmount nfssecure
 //下载密钥，修改配置文件
 [root@desktop0 ~]# wget -O /etc/krb5.keytab http://classroom.example.com/pub/keytabs/desktop0.keytab
 [root@desktop0 ~]# vim /etc/sysconfig/nfs
 RPCNFSDARGS="-V 4.2"
 //修改开机自动挂载
 [root@desktop0 ~]# vim /etc/fstab
 172.25.0.11:/public /mnt/nfsmount nfs ro 0 0
 172.25.0.11:/protected /mnt/nfssecure nfs defaults,v4.2,sec=krb5p 0 0
 //设置开机自启动，并启动服务，挂载
 [root@desktop0 ~]# systemctl enable nfs-secure
 ln -s '/usr/lib/systemd/system/nfs-secure.service' '/etc/systemd/system/nfs.target.wants/nfs-secure.service'
 [root@desktop0 ~]# systemctl restart nfs-secure
 [root@desktop0 ~]# mount -a
```

```
31
```

验证

```
//验证public只读
[root@desktop0 ~]# cd /mnt/nfsmount/
[root@desktop0 nfsmount]# touch a
touch: cannot touch ‘a’: Read-only file system
//验证ldapuser12能以读写方式访问/protected/project
[root@desktop0 ~]# ssh ldapuser12@localhost
。。。
ldapuser12@localhost's password: kerberos
。。。
 -bash-4.2$ id
 uid=1712(ldapuser12) gid=1712(ldapuser12) groups=1712(ldapuser12) context=u
nconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
 -bash-4.2$ df -h
 -bash-4.2$ cd /mnt/nfssecure/
 -bash-4.2$ ll
 total 0
 drwxr-xr-x. 2 ldapuser12 root 6 Dec 30 08:46 project
 -bash-4.2$ cd project/
 -bash-4.2$ touch a //测试是否可写
 -bash-4.2$ ls
 a
```

自动挂载脚本

最后根据需要挂载的具体情况决定

```
[root@desktop0 ~]# mkdir /scripts
[root@desktop0 ~]# cd /scripts/
[root@desktop0 scripts]# vim automount.sh
[root@desktop0 scripts]# chmod +x automount.sh
[root@desktop0 scripts]# cat automount.sh
#!/bin/bash

while true;do
status=$(df -h | grep -E '/mnt/dev|/mnt/nfsmount|/mnt/nfssecure' | wc -l)

  if [ $status -ne 3 ];then
  mount -a
```

```
 else
 break
 fi
done
[root@desktop0 ~]# vim /etc/rc.local
nohup /bin/bash /scripts/automount.sh
```

十六、在 server 上配置一个 web 站点 http://server0.example.com

从 http://classroom.example.com/pub/example.html 下载文件，并重命名为 index.html ，不要修改文件内容

将文件 index.html 拷贝到您的 DocumentRoot 目录下

来自于 example.com 的客户端可以访问该 web 服务器

来自于 my133t.org 的客户端的访问会被拒绝

```
//下载安装apache服务
[root@server0 ~]# yum -y install httpd
//下载html文件至指定目录，修改命名
[root@server0 ~]# cd /var/www/html/
[root@server0 html]# wget -O index.html http://classroom.example.com/pub/exa
mple.html
[root@server0 html]# cd
//防火墙放行服务
[root@server0 ~]# firewall-cmd --permanent --add-rich-rule 'rule family=ipv4
source address=172.25.0.0/24 service name=http accept'
success
[root@server0 ~]# firewall-cmd --reload
success
//设置开机自启动，并重新启动服务
[root@server0 ~]# systemctl enable httpd.service
ln -s '/usr/lib/systemd/system/httpd.service' '/etc/systemd/system/multi-us
er.target.wants/httpd.service'
[root@server0 ~]# systemctl restart httpd.service
```

十七、为站点 http://server0.example.com 配置 TLS 加密

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

证书的秘钥从 http://classroom.example.com/pub/tls/private/server0.key 获取

证书的签名授权信息从 http://classroom.example.com/pub/tls/certs/www0.crt 获取

```
//安装mod_ssl模块
[root@server0 httpd]# yum -y install mod_ssl
//创建ssl目录，下载证书至该目录
[root@server0 ~]# cd /etc/httpd/
[root@server0 httpd]# mkdir ssl
[root@server0 httpd]# cd ssl/
[root@server0 ssl]# wget
 http://classroom.example.com/pub/tls/certs/server0.crt
 http://classroom.example.com/pub/tls/certs/www0.crt
  http://classroom.example.com/pub/tls/private/server0.key
 //修改配置文件
 [root@server0 ~]# vim /etc/httpd/conf.d/ssl.conf
 DocumentRoot "/var/www/html"
 ServerName server0.example.com:443
 SSLCertificateFile /etc/httpd/ssl/server0.crt
 SSLCertificateKeyFile /etc/httpd/ssl/server0.key
 SSLCertificateChainFile /etc/httpd/ssl/www0.crt
 //防火墙放行https
 [root@server0 ~]# firewall-cmd --permanent --add-rich-rule 'rule family=ipv
4 source address=172.25.0.0/24 service name=https accept'
 success
 [root@server0 ~]# firewall-cmd --reload
 success
 //重新启动服务
 [root@server0 ~]# systemctl restart httpd
```

十八、在 server0 上扩展您的 WEB 服务器

为站点 http://www.example.com 创建一个虚拟主机

设置 DocumentRoot 为 /var/www/virtual

从 http://classroom.example.com/pub/www.html 下载文件，并重命名为 index.html，不要修改文件内容

将文件 index.html 拷贝到 DocumentRoot 目录下

确保 floyd 用户能够在 /var/www/virtual 下创建文件

```
//创建目录，进入下载www.html
[root@server0 ~]# mkdir /var/www/virtual
```

```
[root@server0 ~]# cd /var/www/virtual/
[root@server0 virtual]# wget -O index.html http://classroom.example.com/pub/www.html
//创建floyd用户，设置权限
[root@server0 virtual]# useradd floyd
[root@server0 virtual]# setfacl -m u:floyd:rwx /var/www/virtual/
[root@server0 virtual]# getfacl /var/www/virtual/
//复制虚拟主机配置文件，到conf.d目录下，并修改配置文件
[root@server0 virtual]# cd /etc/httpd/conf.d/
[root@server0 conf.d]# cp /usr/share/doc/httpd-2.4.6/httpd-vhosts.conf .
[root@server0 conf.d]# vim httpd-vhosts.conf
<VirtualHost *:80>
 DocumentRoot "/var/www/virtual"
 ServerName www.example.com
</VirtualHost>
<VirtualHost *:80>
 DocumentRoot "/var/www/html"
 ServerName server0.example.com
</VirtualHost>
[root@server0 conf.d]# systemctl restart httpd.service
```

十九、Web 访问控制

在您 server 上的 web 服务器的 DocumentRoot 目录下创建一个名为 private 的目录
从 http://classroom.example.com/pub/private.html 下载文件到这个目录，并重命名为 index.html，不要修改文件内容
从 server 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

```
//进入html目录下，创建private目录，将指定文件下载到该目录下
[root@server0 conf.d]# cd /var/www/html/
[root@server0 html]# mkdir private
[root@server0 html]# cd private/
[root@server0 private]# wget -O index.html http://classroom.example.com/pub/private.html
//修改虚拟主机配置文件，在server0虚拟主机中添加配置文件
[root@server0 private]# vim /etc/httpd/conf.d/httpd-vhosts.conf
<Directory "/var/www/html/private">
```

```
 Require ip 172.25.0.11
</Directory>
//重启服务，并验证
[root@server0 private]# systemctl restart httpd.service
[root@server0 private]# curl http://server0.example.com/private/index.html
private
```

二十、在 server 上实现动态 web 内容

动态内容由名为 alt.example.com 的虚拟主机提供

虚拟主机侦听端口为 8909

从 http://classroom.example.com/pub/webapp.wsgi 下载一个脚本，然后放在适当的位置，不要修改文件内容

客户端访问 http://alt.example.com:8909 时，应该接收到动态生成的 web 页面

http://alt.example.com:8909 必须能被 example.com 内所有的系统访问

```
//修改虚拟主机配置文件，监听8909端口
[root@server0 ~]# vim /etc/httpd/conf.d/httpd-vhosts.conf
Listen 8909
<VirtualHost *: 8909>
 ServerName alt.example.com: 8909
 WSGIScriptAlias / "/var/www/wsgi/webapp.wsgi"
</VirtualHost>
//在www目录下创建wsgi，将指定文件下载到该目录下
[root@server0 ~]# cd /var/www/
[root@server0 www]# mkdir wsgi
[root@server0 www]# cd wsgi/
[root@server0 wsgi]# wget http://classroom.example.com/pub/webapp.wsgi
[root@server0 wsgi]# ls
webapp.wsgi
//安装指定模块软件包mod_wsgi，防火墙放行8909端口
[root@server0 wsgi]# yum -y install mod_wsgi
[root@server0 wsgi]# firewall-cmd --permanent --add-rich-rule 'rule family=
ipv4 source address=172.25.0.0/24 port port=8909 protocol=tcp accept'
[root@server0 wsgi]# firewall-cmd --reload
[root@server0 wsgi]# firewall-cmd --list-all
//selinux放行8909端口
[root@server0 wsgi]# semanage port -l | grep http
```

```
http_cache_port_t tcp 8080, 8118, 8123, 10001-10010
http_cache_port_t udp 3130
http_port_t tcp 80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t tcp 5988
pegasus_https_port_t tcp 5989
[root@server0 wsgi]# semanage port -a -t http_port_t -p tcp 8909
[root@server0 wsgi]# semanage port -l | grep http
http_cache_port_t tcp 8080, 8118, 8123, 10001-10010
http_cache_port_t udp 3130
http_port_t tcp 8909, 80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t tcp 5988
pegasus_https_port_t tcp 5989
//重新启动服务
[root@server0 wsgi]# systemctl restart httpd
```

二十一、配置 server 提供一个 iSCSI 共享服务

磁盘名为 iqn.2014-09.com.example:server

服务端口为 3260

使用 iscsi_store 作为其后端卷其大小为 3G

此服务只能被 desktop.example.com 访问

```
//创建一个3G的主分区
[root@server0 ~]# lsblk
[root@server0 ~]# fdisk /dev/vdb
[root@server0 ~]# partprobe /dev/vdb
//安装target包并进行配置
[root@server0 ~]# yum -y install target*
[root@server0 ~]# targetcli
/> ls
o- / ......................................................... [...]
  o- backstores ............................................. [...]
  | o- block ................................. [Storage Objects: 0]
  | o- fileio ................................ [Storage Objects: 0]
  | o- pscsi ................................. [Storage Objects: 0]
  | o- ramdisk ............................... [Storage Objects: 0]
  o- iscsi ........................................... [Targets: 0]
  o- loopback ........................................ [Targets: 0]
```

```
17
18 //创建iscsi_store，将创建的主分区分配给iscsi_store
19 /> cd /backstores/block
20 /backstores/block> create iscsi_store /dev/vdb1
21
22 //创建磁盘名为iqn.2014-09.com.example:server的磁盘
23 /backstores/block> cd /iscsi
24 /iscsi> create iqn.2014-09.com.example:server
25
26 //设置acl只允许desktop.example.com访问
27 /iscsi> cd iqn.2014-09.com.example:server/tpg1/acls
28 /iscsi/iqn.20...ver/tpg1/acls> create iqn.2014-09.com.example:desktop
29
30 //将iscsi_store添加到luns中
31 /iscsi/iqn.20...e:server/tpg1> cd luns
32 /iscsi/iqn.20...ver/tpg1/luns> create /backstores/block/iscsi_store
33
34 //设置服务端口，允许本机
35 /iscsi/iqn.20...e:server/tpg1> cd portals
36 /iscsi/iqn.20.../tpg1/portals> create 172.25.0.11 3260
37
38 //检查，并退出保存
39 /iscsi/iqn.20.../tpg1/portals> cd /
40 /> ls
41 o- / ......................................................... [...]
42  o- backstores ............................................ [...]
43  | o- block ................................ [Storage Objects: 1]
44  | | o- iscsi_store ... [/dev/vdb1 (3.0GiB) write-thru activated]
45  | o- fileio ............................... [Storage Objects: 0]
46  | o- pscsi ................................ [Storage Objects: 0]
47  | o- ramdisk .............................. [Storage Objects: 0]
48  o- iscsi ........................................... [Targets: 1]
49  | o- iqn.2014-09.com.example:server .................. [TPGs: 1]
50  | o- tpg1 ............................... [no-gen-acls, no-auth]
51  | o- acls ......................................... [ACLs: 1]
52  | | o- iqn.2014-09.com.example:desktop .... [Mapped LUNs: 1]
53  | | o- mapped_lun0 ......... [lun0 block/iscsi_store (rw)]
54  | o- luns ......................................... [LUNs: 1]
```

```
  | | o- lun0 ............... [block/iscsi_store (/dev/vdb1)]
  | o- portals ................................ [Portals: 1]
  | o- 172.25.0.11:3260 ............................. [OK]
  o- loopback ..................................... [Targets: 0]
/> exit
//设置开机自启动，并启动服务
[root@server0 ~]# systemctl enable target
[root@server0 ~]# systemctl restart target
//防火墙放行3260端口，并刷新
[root@server0 ~]# firewall-cmd --permanent --add-rich-rule 'rule family=ipv
4 source address=172.25.0.0/24 port port=3260 protocol=tcp accept'
[root@server0 ~]# firewall-cmd --reload
```

二十二、配置 desktop 使其能连接在 server 上提供的 iscsi

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 1500MB 的分区，并格式化为 ext4

此分区挂载在 /mnt/netdev 上同时在系统启动的期间自动加载

```
//安装iscsi-init相关包
[root@desktop0 ~]# yum -y install iscsi-init*
//修改配置文件，与在服务端配置的acl标识一致
[root@desktop0 ~]# vim /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.2014-09.com.example:desktop
//设置开机自启动，并启动服务
[root@desktop0 iscsi]# systemctl enable iscsid
[root@desktop0 iscsi]# systemctl start iscsid
//发现iscsi，并连接iscsi
[root@desktop0 iscsi]# man iscsiadm
[root@desktop0 iscsi]# iscsiadm --mode discoverydb --type sendtargets --por
tal 172.25.0.11 --discover
172.25.0.11:3260,1 iqn.2014-09.com.example:server
[root@desktop0 iscsi]# iscsiadm --mode node --targetname iqn.2014-09.com.ex
ample:server --portal 172.25.0.11:3260 --login
Logging in to [iface: default, target: iqn.2014-09.com.example:server, port
al: 172.25.0.11,3260] (multiple)
Login to [iface: default, target: iqn.2014-09.com.example:server, portal: 1
72.25.0.11,3260] successful.
//在sda上创建一个1500M的主分区
```

```
[root@desktop0 iscsi]# lsblk //查看是否有sda
[root@desktop0 iscsi]# fdisk /dev/sda
[root@desktop0 iscsi]# partprobe /dev/sda
[root@desktop0 iscsi]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 3G 0 disk
└─sda1 8:1 0 1.5G 0 part
vda 253:0 0 10G 0 disk
└─vda1 253:1 0 10G 0 part /
vdb 253:16 0 10G 0 disk
//格式化为ext4，并设置开机自动挂载
[root@desktop0 iscsi]# mkfs.ext4 /dev/sda1
[root@desktop0 iscsi]# mkdir /mnt/netdev
[root@desktop0 iscsi]# blkid /dev/sda1
/dev/sda1: UUID="e6989b1a-eb89-412b-ba26-b058b2f6734b" TYPE="ext4"
[root@desktop0 iscsi]# vim /etc/fstab
 UUID="e6989b1a-eb89-412b-ba26-b058b2f6734b" /mnt/netdev ext4 defaults,_net
dev 0 0
[root@desktop0 iscsi]# mount -a
//检验
[root@desktop0 iscsi]# df -h
```

二十三、编写一个位于 /root/program 的 shell 脚本

当执行 /root/programtang 时，终端显示 kai

当执行 /root/programkai 时，终端显示 tang

当仅执行 /root/program 不加参数，或者加上其他参数时，终端显示标准错误输出 /root/program tang|kai

```
[root@desktop0 ~]# cat program
#!/bin/bash/

case $1 in
 'tang')
 echo 'kai'
 ;;
 'kai')
 echo 'tang'
```

```
 ;;
 *)
 echo '/root/program tang|kai'
 ;;
esac
//检验
```

二十四、写一个创建用户的脚本

脚本名为 /root/mkuser ，脚本执行时需要添加一个参数，

请在 http://classroom.example.com/pub/user 下载下来，这个 user 就是参数

如果没有参数，将提示：Usage：/root/mkuser

如果参数为不存在的文件，则提示：Input file not found

如果存在，则创建用户，用户不需要设置密码，用户的 shell 为 /bin/false

```
[root@desktop0 ~]# cat mkuser
#!/bin/bash

if [ $# -eq 0 ];then
 echo "Usage: /root/mkuser"
else
 if [ ! -f $1 ];then
 echo "Input file not found."
 else
  for name in $(cat $1);do
  /usr/sbin/useradd -s /bin/false $name &>/dev/null
  done
  fi
fi
//检验
```

二十五、在你的机器上创建一个 mariadb 数据库

数据库名为 contacts

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/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：redhat
root 用户密码为 redhat ， 同时不允许空密码登陆

```
//下载数据库
[root@server0 ~]# wget http://classroom.example.com/pub/users.mdb
[root@server0 ~]# ls
anaconda-ks.cfg users.mdb
//安装及相关的包，设置开机自启动，并启动服务
[root@server0 ~]# yum -y install mariadb*
[root@server0 ~]# systemctl enable mariadb
[root@server0 ~]# systemctl start mariadb
//数据库初始化
 [root@server0 ~]# mysql_secure_installation
 Set root password? [Y/n] y //设置root密码
 New password: redhat
 Re-enter new password: redhat
 Password updated successfully!
 Reloading privilege tables..
  ... Success!

 Remove anonymous users? [Y/n] y //禁止匿名登陆
  ... Success!
 //这里输入Y是不允许root用户登录
 Disallow root login remotely? [Y/n] n //允许root登录，
  ... skipping.
 //移除测试数据库和访问权限
 Remove test database and access to it? [Y/n] y //移除测试数据库和访问权限
  - Dropping test database...
  ... Success!
  - Removing privileges on test database...
  ... Success!

 Reload privilege tables now? [Y/n] y //刷新权限表
  ... Success!

 Thanks for using MariaDB!
 //登录数据库
 [root@server0 ~]# mysql -uroot -predhat
```

```
//添加contacts数据库
MariaDB [(none)]> create database contacts;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> use contacts;
Database changed
//数据库文件并执行
MariaDB [contacts]> source users.mdb;
MariaDB [contacts]> show tables;
+--------------------+
| Tables_in_contacts |
+--------------------+
| u_loc |
| u_name |
| u_passwd |
+--------------------+
3 rows in set (0.00 sec)
//查看表结构
MariaDB [contacts]> desc u_loc;
+----------+-------------+------+-----+---------+----------------+
| Field | Type | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+----------------+
| uid | int(11) | NO | PRI | NULL | auto_increment |
| location | varchar(50) | NO | | NULL | |
+----------+-------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)

MariaDB [contacts]> desc u_name;
+-----------+-------------+------+-----+---------+----------------+
| Field | Type | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+----------------+
| userid | int(11) | NO | PRI | NULL | auto_increment |
| firstname | varchar(50) | NO | | NULL | |
| lastname | varchar(50) | NO | | NULL | |
+-----------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

```

```
MariaDB [contacts]> desc u_passwd;
+----------+-------------+------+-----+---------+----------------+
| Field | Type | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+----------------+
| uid | int(11) | NO | PRI | NULL | auto_increment |
| password | varchar(50) | NO | | NULL | |
+----------+-------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)
//创建raikon用户，密码为redhat，并且只拥有查询contacts数据库的权限
MariaDB [contacts]> grant select on contacts.* to 'raikob'@'localhost' iden
tified by 'redhat';
Query OK, 0 rows affected (0.00 sec)
//刷新权限表
MariaDB [contacts]> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

使用相应的 SQL 查询以回答下列问题：

- 密码是 fadora 的人的名字是什么?

```
//查询密码为“fadora”的uid
MariaDB [contacts]> select * from u_passwd where password='fadora';
+-----+----------+
| uid | password |
+-----+----------+
| 8 | fadora |
+-----+----------+
1 row in set (0.00 sec)
//查询密码对应uid的人的姓名
MariaDB [contacts]> select * from u_name where userid = 8;
+--------+-----------+----------+
| userid | firstname | lastname |
+--------+-----------+----------+
| 8 | John | Clinton |
+--------+-----------+----------+
1 row in set (0.00 sec)
//一条查询语句查询
MariaDB [contacts]> select n.firstname,n.lastname from u_name as n,u_passwd
as p where p.password = 'fadora' and p.uid = n.userid;
```

```
+-----------+----------+
| firstname | lastname |
+-----------+----------+
| John | Clinton |
+-----------+----------+
1 row in set (0.00 sec)

MariaDB [contacts]>
```

- 有多少人的姓名是 John， 同时居住在 Santa Clara ？

```
//查询姓名为John的人
MariaDB [contacts]> select * from u_name where firstname = 'John';
+--------+-----------+-------------+
| userid | firstname | lastname |
+--------+-----------+-------------+
| 8 | John | Clinton |
| 10 | John | li |
| 15 | John | Clinton |
| 21 | John | Jackson |
| 22 | John | Obama |
| 23 | John | Clinton |
| 24 | John | Walker Bush |
| 25 | John | wang |
+--------+-----------+-------------+
8 rows in set (0.01 sec)
//查询居住地址为Santa Clara
MariaDB [contacts]> select * from u_loc where location = 'Santa Clara';
+-----+-------------+
| uid | location |
+-----+-------------+
| 3 | Santa Clara |
| 4 | Santa Clara |
| 8 | Santa Clara |
| 15 | Santa Clara |
| 20 | Santa Clara |
| 21 | Santa Clara |
| 24 | Santa Clara |
+-----+-------------+
```

```
7 rows in set (0.00 sec)
//一条语句查询并统计
MariaDB [contacts]> select count(*) from u_name as n,u_loc as l where n.fir
stname = 'john' and l.location = 'Santa Clara' and n.userid = l.uid;
+----------+
| count(*) |
+----------+
| 4 |
+----------+
1 row in set (0.00 sec)

MariaDB [contacts]>
```