

01. 配置 SELinux

确保您的两个系统的 SELinux 处于强制启用模式。

```
[root@server0 ~]# vim /etc/selinux/config // 配置开机自动开启 SELinux
SELINUX=enforcing
[root@server0 ~]# setenforce 1 //临时开启 SELinux
[root@server0 ~]# getenforce //查看结果
Enforcing
```

02. 配置 SSH 访问控制

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 unknown.org 内的客户端不能访问您的两个虚拟机系统

```
[root@server0 ~]# vim /etc/ssh/sshd_config
...
DenyUsers *@*.unknown.org *@172.34.0.* // 网段地址以考试说明为准
[root@server0 ~]# systemctl restart sshd
```

注意：

不要把掩码写进去，写进去一定连接不上。

严格区分大小写

03. 自定义用户环境（别名设置）

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@server0 ~]# vim /etc/bashrc
alias qstat='/bin/ps -Ao pid,tt,user,fname,rsz'
[root@server0 ~]# su - student //切换为任意现有用户
[student@server0 ~]$ qstat //测试执行别名 qstat 都可用
```

注意：

配置文件/etc/bashrc

别名等号两边不要有空格

04. 配置防火墙端口转发

在系统 server0 配置端口转发，要求如下：

在 172.25.0.0/24 网络中的客户机，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 ~]# systemctl restart firewalld
[root@server0 ~]# systemctl enable firewalld
//调整防火墙信任区域（设为 trusted），简化对后续各种服务的防护
[root@server0 ~]# firewall-cmd --set-default-zone=trusted
//阻止未授权网络 unknown.org 的网段
[root@server0 ~]# firewall-cmd --permanent --add-source=172.34.0.0/24 --
zone=block
//实现端口 5423 ==> 80 端口转发
[root@server0 ~]# firewall-cmd --permanent --zone=trusted --add-
```

```
forwardport=port=5423:proto=tcp:toport=80
```

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

注意：

练习时，加永久生效，可能会导致内存不足卡死，重启即可，实在不行做一个交换分区也可解决。

考试时内存足够大，一般不会出现卡死现象，出现卡死，重启机器，或者做交换分区出来。防火墙一定记得，做完之后让配置生效，**reload**

05. 配置链路聚合

在您的两个系统之间配置一个聚合链路 team0:

此链路使用接口 eth1 和 eth2 (预先提供或自行添加接口)

此链路在一个接口失效时仍然能工作;

此链路在 server0 使用地址 172.16.3.20/255.255.255.0

此链路在 desktop0 使用地址 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

1) 新建聚合链接

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

```
[root@server0 ~]# nmcli connection add con-name team0-p1 type team-slave ifname eth1 master team0 //指定成员网卡 1
```

```
[root@server0 ~]# nmcli connection add con-name team0-p2 type team-slave ifname eth2 master team0 //指定成员网卡 2
```

```
[root@server0 ~]# nmcli con modify team0 ipv4.method manual ipv4.addresses "172.16.3.20/24" connection.autoconnect yes //为聚合连接配置 IP 地址
```

2) 设置自动连接 (通常情况下, 默认是已经设置好的)

```
[root@server0 ~]# nmcli con modify team0 connection.autoconnect yes
```

```
[root@server0 ~]# nmcli con modify team0-p1 connection.autoconnect yes
```

```
[root@server0 ~]# nmcli con modify team0-p2 connection.autoconnect yes
```

3) 激活连接并检查

```
[root@server0 ~]# nmcli connection up team0 //激活聚合连接
```

```
[root@server0 ~]# teamdctl team0 state //确认连接状态
```

注(1): 此考点在 desktop0 的配置与 server0 几乎完全相同 (只是 IP 地址不一样)。

网卡 eth1、eth2 考试时是现成的

添加链路聚合的操作 可以 `man nmcli-examples ==> Team1`

链路聚合的 runner 配置 可以 `man teamd.conf ==> "runner`

尽量复制粘贴不要手写

为聚合连接配置的 IP 地址千万不要与 eth0 在同一个网段

06. 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:

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

地址必须在重启后依旧生效

两个系统必须保持当前的 IPv4 地址并能通信

```
[root@server0 ~]# nmcli connection show //获知 eth0 的连接名称
NAME                UUID                                TYPE                DEVICE
System eth0         5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03 802-3-ethernet      eth0
[root@server0 ~]# nmcli connection modify "连接名" ipv6.method manual \
ipv6.addresses 2003:ac18::305/64
[root@server0 ~]# nmcli connection up "连接名"
```

注意：

此题一定要先修改静态主机名

做完之后记得开启，up

07. 配置本地邮件服务

在系统 server0 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 上发送邮件到本地用户 student 来测试您的配置，系统 desktop0.example.com 上的用户 student 将会收到这封邮件。

1) 在虚拟机 server0 上

```
[root@server0 ~]# hostnamectl set-hostname server0.example.com //正
确设置静态主机名，避免邮箱域识别错误
```

```
[root@server0 ~]# vim /etc/postfix/main.cf //调整邮件服务配置
relayhost = [smtp0.example.com] //后端邮件服务器
inet_interfaces = loopback-only //仅本机
myorigin = desktop0.example.com //发件来源域
mynetworks = 127.0.0.0/8, [::1]/128 //信任网络
mydestination = //此行的值设为空
local_transport = error:local delivery disabled //提供本地传输报错提
```

```
[root@server0 ~]# systemctl restart postfix
```

```
[root@server0 ~]# systemctl enable postfix
```

```
[root@server0 ~]# echo 'I am king.' | mail -s 'Test1' student //在 server0 发测试邮件
```

2) 在虚拟机 desktop0 上

注(1)：考试时 desktop0 的配置与 server0 完全相同（后端邮件服务器由考场提供，无需考生配置）

注(2)：练习时需在 desktop0 上执行 lab smtp-nullclient setup，将其配置为后端邮件服务器

注(3)：local_transport = error:报错信息，考试无要求可不写，练习时必须写

此题出错，检查防火墙，主机名，配置文件

```
[root@desktop0 ~]# hostnamectl set-hostname desktop0.example.com
```

//正确设置静态主机名，避免邮箱域识别错误

```
[root@desktop0 ~]# mail -u student //练习时在 desktop0 应能收到测试邮件
```

.. .. //考试时访问题目给出的网址，应能看到测试邮件

08. 配置默认 web 站点

为 `http://server0.example.com` 配置 Web 服务器:

从 `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 ~]# vim /etc/httpd/conf.d/00-default.conf //添加第一个(默认)虚拟主机
```

```
<VirtualHost *:80>
```

```
    ServerName server0.example.com
```

```
    DocumentRoot /var/www/html
```

```
</VirtualHost>
```

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

```
[root@server0 html]# wget http://classroom/pub/materials/station.html -O index.html
```

```
[root@server0 html]# systemctl restart httpd
```

```
[root@server0 html]# systemctl enable httpd
```

注意:

-O 一定不能丢, 需要指定位置!!!

配置文件区分大小写最好!!! 养成习惯!!!

给的 URL 连接写对!

09. 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` 的内容做任何修改

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

确保 `student` 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 `classroom.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/pub/materials/www.html -O index.html
```

```
[root@server0 virtual]# vim /etc/httpd/conf.d/01-www0.conf
```

```
<VirtualHost *:80>
```

```
    ServerName www0.example.com
```

```
    DocumentRoot /var/www/virtual
```

```
</VirtualHost>
```

```
[root@server0 virtual]# chown -R harry /var/www/virtual/  
[root@server0 virtual]# systemctl restart httpd
```

注意：

**-R 一定不能少!!! 权限不能改变! 出错之后先检查权限问题!!!
给的 URL 连接写对!**

10. 配置安全 web 服务

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

证书的密钥从 `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/example-ca.crt` 获取

1) 安装 `mod_ssl` 软件包

```
[root@server0 ~]# yum -y install mod_ssl
```

2) 下载并部署网站证书、密钥、根证书

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

```
[root@server0 certs]# wget http://classroom/pub/example-ca.crt //部署网站证书
```

```
[root@server0 certs]# wget http://classroom/pub/tls/certs/server0.crt //部署根证书
```

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

```
[root@server0 private]# wget http://classroom/pub/tls/private/server0.key //部署网站密  
钥
```

3) 调整虚拟 Web 主机配置 `ssl.conf`

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

```
<VirtualHost _default_:443>
```

```
    DocumentRoot "/var/www/html"           //设置站点根目录
```

```
    ServerName server0.example.com:443      //设置站点名称
```

```
    ... ..                                  //搜 localhost, 找到并修改以下行
```

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

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

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

```
</VirtualHost>
```

```
[root@server0 private]# systemctl restart httpd
```

注意：

此题需要下载网站证书、根证书、网站密钥三个文件! 而且下载的目录不一样!

11. 配置 web 内容的访问

在您的 `server0` 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://classroom.example.com/pub/materials/private.html` 下载一个文件副本到这个目录，并且得命名为 `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/pub/materials/private.html -O index.html
```

```
[root@server0 private]# vim /etc/httpd/conf.d/00-default.conf
...
<Directory /var/www/html/private>
    Require ip 127.0.0.1 ::1 172.25.0.11 //仅允许本机 IP 访问
</Directory>
```

```
[root@server0 private]# systemctl restart httpd
```

注意：

建议为每一个虚拟站点准备独立的配置文件，

默认站点的配置文件名最好以 00 开头

httpd -t 检查语法

指定目录下载，而且-O 不能少！

12. 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> 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/pub/materials/webinfo.wsgi
```

```
[root@server0 webapp0]# vim /etc/httpd/conf.d/02-alt.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 //突破 SELinux 限制
```

```
[root@server0 webapp0]# systemctl restart httpd
```

注意：

动态网站这个，别忘了监听端口 Listen 8909

测试访问动态网站时 http://域名:8909/ 也需要跟对应的端口

13. 通过 Samba 发布共享目录

在 server0 上通过 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

```
[root@server0 ~]# yum -y install samba
[root@server0 ~]# mkdir /common
[root@server0 ~]# setsebool -P samba_export_all_rw=on //取消 SELinux 限制
[root@server0 ~]# useradd harry //此账号考试时或已提供
[root@server0 ~]# pdbedit -a harry //启用共享账号 harry、设置密码
[root@server0 ~]# vim /etc/samba/smb.conf
[global]
workgroup = STAFF //修改此行，指定工作组名
[common]
path = /common
hosts allow = 172.25.0.0/24 //只允许指定网段访问
```

```
[root@server0 ~]# systemctl restart smb nmb
```

```
[root@server0 ~]# systemctl enable smb nmb
```

注(1)：为安全起见，建议直接调整共享目录的安全属性（而不调整全局布尔值）

```
[root@server0 ~]# semanage fcontext -a -t samba_share_t '/common(/.*)?'
```

```
[root@server0 ~]# restorecon -R /common //设置目录 SELinux 属性
```

-P 可能会引起内存不足，重启即可，selinux 需要取消

工作组名不能少！

14. 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，该密码是 atenth

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

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

1) 在虚拟机 server0 上

```
[root@server0 ~]# mkdir /devops //若无目录，应提前建好
[root@server0 ~]# useradd kenji //此账号考试时默认应已提供
[root@server0 ~]# pdbedit -a kenji //启用共享账号 kenji、设置密码
[root@server0 ~]# useradd chihiro //此账号考试时默认应已提供
[root@server0 ~]# pdbedit -a chihiro //启用共享账号 chihiro、设置密码
[root@server0 ~]# setfacl -m u:chihiro:rwX /devops/ //设置写入权限
```

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

```
[devops]
```

```
path = /devops
```

```
write list = chihiro
```

```
hosts allow = 172.25.0.0/24 //只允许指定网段访问
```

```
[root@server0 ~]# systemctl restart smb nmb //重启
```

注(1): 为安全起见, 建议直接调整共享目录的安全属性 (而不调整全局布尔值)

```
[root@server0 ~]# semanage fcontext -a -t samba_share_t '/devops(/.*)?'
```

```
[root@server0 ~]# restorecon -R /devops //设置目录 SELinux 属性
```

2) 在虚拟机 desktop0 上

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

```
[root@desktop0 ~]# smbclient -L server0 //检查对方提供了哪些共享
```

```
... //无需密码, 直接按 Enter 键确认
```

```
[root@desktop0 ~]# mkdir /mnt/dev //创建好挂载点
```

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

```
//server0.example.com/devops /mnt/dev cifs
```

```
username=kenji,password=atenorth,multiuser,sec=ntlmssp_netdev 0 0
```

```
[root@desktop0 ~]# mount -a //检查挂载配置
```

注意:

装包是 samba-client cifs-utils 两个包, 缺一不可

自动挂载格式要写对, 尤其是 sec=ntlmssp_netdev, 下划线不能少

3) 验证结果 (在 desktop0 上)

```
[root@desktop0 ~]# su - student //切换到普通用户
```

```
[student@desktop0 ~]$ cifscreds add -u chihiro server0 //临时切换为 chihiro 身份
```

```
Password: //提供 Samba 用户 chihiro 的密码
```

```
[student@desktop0 ~]$ touch /mnt/dev/b.txt //确认有写入权限 (新建文件)
```

注(2): 为安全起见, 建议不要将密码直接存入/etc/fstab 文件

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

```
//server0.example.com/devops /mnt/dev cifs
```

```
credentials=/root/smb.pass,multiuser,sec=ntlmssp_netdev 0 0
```

```
[root@desktop0 ~]# vim /root/smb.pass
```

```
username=kenji
```

```
password=atenorth
```

15. 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, 同时只能被 example.com 域中的系统访问

以读写的方式共享目录/protected, 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

注(1): 考试时 LDAP+Kerberos 环境预先已配置好, 用户 ldapuser0 可以直接使用

注(2): 练习时需在 server0 上执行 lab nfskrb5 setup, 由脚本快速配置好 LDAP+Kerberos 环境

```
[root@server0 ~]# mkdir -p /public /protected/project
[root@server0 ~]# chown ldapuser0 /protected/project/
[root@server0 ~]# wget -O /etc/krb5.keytab http://classroom/pub/keytabs/server0.keytab
[root@server0 ~]# vim /etc/exports
/public 172.25.0.0/24(ro)
/protected 172.25.0.0/24(rw,sec=krb5p)
[root@server0 ~]# systemctl start nfs-secure-server nfs-server //启用两个系统服务
[root@server0 ~]# systemctl enable nfs-secure-server nfs-server
```

注意:

服务不能起错, 需要启动 nfs-secure-server nfs-server 两个服务

16. 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

注(1): 考试时 LDAP+Kerberos 环境预先已配置好, 用户 ldapuser0 可以直接使用

注(2): 练习时需在 desktop0 上执行 lab nfskrb5 setup, 由脚本快速配置好 LDAP+Kerberos 环境

```
[root@desktop0 ~]# mkdir -p /mnt/nfsmount /mnt/nfssecure
[root@desktop0 ~]# wget -O /etc/krb5.keytab
http://classroom/pub/keytabs/desktop0.keytab
```

```
[root@desktop0 ~]# systemctl start nfs-secure //启用安全 NFS 的客户端服务
[root@desktop0 ~]# systemctl enable nfs-secure
```

注意:

启动 nfs-secure 服务

```
[root@desktop0 ~]# showmount -e server0 //检查对方提供了哪些共享
```

Export list for server0:

```
/protected 172.25.0.0/24
/public 172.25.0.0/24
```

```
[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 ~]# mount -a //测试挂载配置
```

```
[root@desktop0 ~]# ssh ldapuser0@desktop0 //SSH 登入以领取通行证
ldapuser0@desktop0's password: //密码 kerberos (见考试说明)
[ldapuser0@desktop0 ~]$ touch /mnt/nfssecure/project/a.txt //写入测试
```

注(3): 为避免 desktop0 重启后挂载延迟, 可以考虑做如下开机设置

```
[root@desktop0 ~]# chmod +x /etc/rc.d/rc.local //为开机脚本添加 x 权限
[root@desktop0 ~]# vim /etc/rc.d/rc.local //在文件末尾添加“补刀”操作
for i in {1..20}
do
systemctl status nfs-secure || systemctl restart nfs-secure
mount -a
sleep 5
done &
```

17. 配置 iSCSI 服务端

配置 server0 提供 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

1) 准备磁盘空间

```
[root@server0 ~]# fdisk /dev/vdb
Command (m for help): n // n 新建分区
Partition number (1-128, default 1): //接受默认分区编号
First sector (34-20971486, default 2048): //起始位置默认
Last sector, +sectors or +size{K,M,G,T,P} (2048-20971486, default 20971486): +3G
//结束位置设置 +3G
Command (m for help): w // w 保存分区更改
...
[root@server0 ~]# partprobe /dev/vdb //刷新分区表
```

2) 安装、配置 iSCSI 磁盘

```
[root@server0 ~]# yum -y install targetcli
[root@server0 ~]# targetcli
/> backstores/block create iscsi_store /dev/vdb1 //定义后端存储
/> /iscsi create iqn.2016-02.com.example:server0 //创建 iqn 对象
/> /iscsi/iqn.2016-02.com.example:server0/tpg1/acls create iqn.2016-02.com.example:desktop0 //授权客户机 (的 IQN)
/> /iscsi/iqn.2016-02.com.example:server0/tpg1/luns create /backstores/block/iscsi_store //绑定存储
/> /iscsi/iqn.2016-02.com.example:server0/tpg1/portals create 172.25.0.11 3260 //指定监听地址 (本机 IP 及端口)
/> saveconfig //保存配置结果 (缺省)
/> exit
```

3) 启用 target 服务

```
[root@server0 ~]# systemctl restart target
```

```
[root@server0 ~]# systemctl enable target
```

注意：

iscsi 创建出错后，需要删除之后再做，delete。

18. 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 iscsi-initiator-utils
[root@desktop0 ~]# vim /etc/iscsi/initiatorname.iscsi // 设置本机 iqn 名称
InitiatorName=iqn.2016-02.com.example:desktop0
[root@desktop0 ~]# systemctl restart iscsid //起 iscsid 服务以更新 iqn 名称

[root@desktop0 ~]# iscsiadm -m discovery -t st -p server0 //发现磁盘
[root@desktop0 ~]# iscsiadm -m node -L all // 手动连接磁盘测试
[root@desktop0 ~]# systemctl enable iscsi //将 iscsi 服务设为开机自启
[root@desktop0 ~]# lsblk //确认多出的磁盘，比如/dev/sda
```

```
[root@desktop0 ~]# fdisk /dev/sda
Command (m for help): n // n 新建分区
Partition number (1-128, default 1): //接受默认分区编号
First sector (34-20971486, default 2048): //起始位置默认
Last sector, +sectors or +size{K,M,G,T,P} (2048-20971486, default 20971486): +2100M
//结束位置设置 +2100M
Command (m for help): w // w 保存分区更改
```

```
..
[root@desktop0 ~]# partprobe /dev/sda //刷新分区表
[root@desktop0 ~]# mkfs.ext4 /dev/sda1 //按要求格式化分区
[root@desktop0 ~]# mkdir /mnt/data //创建挂载点
[root@desktop0 ~]# vim /etc/fstab //提前用 blkid 找到分区 UUID
UUID="6ff20bb3-7543-4fa0-b4fa-bdc99a1e63ce" /mnt/data ext4 _netdev 0 0
```

```
[root@desktop0 ~]# mount -a
[root@desktop0 ~]# sync ; reboot -f //先存盘再强制重启，避免卡死
```

注意：

做完之后一定重启虚拟机！

19. 编写简单脚本

有口碑找飞哥 联系电话：18138713024

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 //将正常输出变为错误输出
    exit 2 //返回非零值
fi
[root@server0 ~]# chmod +x /root/foo.sh
```

20. 创建批量添加用户的脚本

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 ~]# 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
```

注意：

Shell 脚本

脚本的文件名和路径不要写错

foo.sh 不要写成 foo

batchusers 不要写成 batchusers.sh

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。

1) 安装、配置

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

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

```
[mysqld]
```

```
skip-networking          //添加此行，跳过网络
```

```
[root@server0 ~]# systemctl restart mariadb
```

```
[root@server0 ~]# systemctl enable mariadb
```

2) 设密码、建库

```
[root@server0 ~]# mysqladmin -u root password 'atenorth' //无密码情况下，设新密码
```

```
[root@server0 ~]# mysql -u root -patenorth
```

```
MariaDB [(none)]> CREATE DATABASE Contacts;
```

```
MariaDB [(none)]> GRANT select ON Contacts.* to Raikon@localhost IDENTIFIED BY 'atenorth';
```

```
MariaDB [(none)]> DELETE FROM mysql.user WHERE Password=""; //删除空密码的账号
```

///!!注意：设好 root 密码再做

```
MariaDB [(none)]> QUIT
```

3) 导入库

```
[root@server0 ~]# wget http://classroom/pub/materials/users.sql
```

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

注意：

数据库名、用户名不要写错（区分大小写）

删除空密码的 root 用户时，千万要提前为 root 用户设好密码

必要时重建数据库系统 ——

```
# systemctl stop mariadb
```

```
# rm -rf /var/lib/mysql/
```

```
# yum -y reinstall mariadb-server
```

```
# systemctl restart mariadb
```

22. 数据库查询（填空）

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 -patenorth
```

```
Enter password: MariaDB [(none)]> USE Contacts;           //选库
```

```
MariaDB [Contacts]> SHOW TABLES;                       //了解有哪些表
```

```
MariaDB [Contacts]> DESC XX 表名;                       //了解 XX 表有哪些列
```

```
MariaDB [Contacts]> SELECT name FROM base WHERE password='solicitous';
```

```
+-----+
```

```
| name |
```

```
+-----+
```

```
| James |
```

```
+-----+
```

```
MariaDB [Contacts]> SELECT count(*) FROM base,location WHERE  
base.name='Barbara' AND location.city='Sunnyvale' AND base.id=location.id;
```

```
1
```

```
MariaDB [Contacts]> QUIT
```

注意事项:

1、认真阅读考试说明

2、下午不需要破密操作,连接虚拟机之后,先修改主机名、改防火墙、IP 一定是自动分配,不需要你改成静态!

3、做题顺序:先做 iscsi,做完重启虚拟机,做完之后可以先做自己易出错的题