

# NSD ENGINEER DAY06

1. [案例1：postfix基础邮件服务](#)
2. [案例2：添加一个swap分区](#)
3. [案例3：配置聚合连接](#)
4. [案例4：Linux工程师 综合测试](#)

## 1 案例1：postfix基础邮件服务

### 1.1 问题

本例要求在虚拟机server0上配置 postfix 基础服务，具体要求如下：

1. 监听本机的所有接口
2. 将邮件域和邮件服务主机名都改为 example.com

然后在server0上使用mail命令测试发信/收信操作：

1. 由 root 给本机用户 mike 发一封测试邮件
2. 查收用户 mike 的邮箱，读取邮件内容，确保是从 root@example.com 发过来的

### 1.2 方案

电子邮箱：1234567@qq.com表示在互联网区域qq.com内的一台邮件服务器上属于用户1234567的一个电子邮箱（目录）。

postfix发信服务（TCP 25，SMTP）的功能：

- 为用户提供电子邮箱
- 为邮箱用户向其他邮件服务器发送邮件
- 为邮箱用户投递/存储收到的邮件

dovecot取信服务（TCP 110/143，POP3/IMAP）的功能：为邮箱用户提取邮件。

### 1.3 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：配置postfix基础邮件服务

##### 1) 安装postfix软件包

```
01. [root@server0 ~]# yum -y install postfix
02. ... ..
```

##### 2) 调整邮件服务配置

```
01. [root@server0 ~]# vim /etc/postfix/main.cf
02. ... ..
03. inet_interfaces = all           //监听接口
```

[Top](#)

04. mydomain = example.com //邮件域
05. myhostname = example.com //本服务器主机名

### 3) 启动postfix服务

01. [root@server0 ~]# systemctl restart postfix

### 4) 查看邮件服务监听状态

01. [root@server0 ~]# netstat -antpu | grep :25
02. tcp 0 0 0.0.0.0:25 0.0.0.0:\* LISTEN 1739/master
03. tcp6 0 0 :::25 :::\* LISTEN 1739/master

## 步骤二：使用mail命令发信/收信

### 1) 给用户root发一封测试邮件

01. [root@server0 ~]# echo '1111' | mail -s 'mail1' root

### 2) 由管理员收取指定用户root的邮件

01. [root@server0 ~]# mail -u root
02. Heirloom Mail version 12.5 7/5/10. Type ? for help.
03. "/var/mail/root": 1 message 1 new
04. >N 1 root Sat Nov 26 17:40 18/532 "mail"
05. & 1 //读取第1封邮件内容
06. Message 1:
07. From root@example.com Sat Nov 26 17:40:06 2016
08. Return-Path: <root@example.com>
09. X-Original-To: root
10. Delivered-To: root@example.com
11. Date: Sat, 26 Nov 2016 17:40:06 +0800
12. To: root@example.com
13. Subject: mail1 //检查邮件标题
14. User-Agent: Heirloom mailx 12.5 7/5/10
15. Content-Type: text/plain; charset=us-ascii

[Top](#)

```
16. From: root@example.com (root)
17. Status: R
18.
19. 1111 //检查邮件内容
20.
21. & q //退出mail程序
22. Held 1 message in /var/mail/root
23. [root@server0 ~]#
```

## 2 案例2：添加一个swap分区

### 2.1 问题

本例要求为虚拟机 server0 添加一个交换分区，相关要求如下：

1. 此交换分区的大小为 512MiB
2. 当系统启动时，swap分区应该可以自动挂载
3. 不要移除或更改其他已经存在于你系统中的交换分区

### 2.2 方案

交换分区不需要挂载点，在配置开机挂载时，挂载点直接写成swap即可。

### 2.3 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：格式化交换分区

1) 将提前准备的分区/dev/vdb7格式化为swap文件系统

```
01. [root@server0 ~]# mkswap /dev/vdb7
02. Setting up swapspace version 1, size = 524284 KiB
03. no label, UUID=80e358b9-b55d-4797-aaa4-41800aa00e3f
```

2) 确认格式化结果

```
01. [root@server0 ~]# blkid /dev/vdb7
02. /dev/vdb7: UUID="80e358b9-b55d-4797-aaa4-41800aa00e3f" TYPE="swap"
```

#### 步骤二：配置交换分区的开机启用

修改/etc/fstab文件，添加交换分区记录：

[Top](#)

```
01. [root@server0 ~]# vim /etc/fstab
```

```
02.  .. ..
03.  /dev/vdb7  swap  swap  defaults  0 0
```

### 步骤三：确认挂载配置可用

#### 1) 检查启用新交换分区之前

```
01.  [root@server0 ~]# swapon -s
02.  [root@server0 ~]#
```

#### 2) 启用新交换分区

```
01.  [root@server0 ~]# swapon -a
```

#### 3) 检查启用新交换分区之后

```
01.  [root@server0 ~]# swapon -s
02.  Filename      Type      Size      Used      Priority
03.  /dev/vdb7      partition 524284    -1
```

## 3 案例3：配置聚合连接

### 3.1 问题

本例要求在两个虚拟机 server0、desktop0之间配置一个链路，要求如下：

1. 此链路使用接口 eth1 和 eth2
2. 此链路在其中一个接口失效时仍然能工作
3. 此链路在 server0 上使用下面的地址 172.16.3.20/255.255.255.0
4. 此链路在 desktop0 上使用下面的地址 172.16.3.25/255.255.255.0
5. 此链路在系统重启之后依然保持正常状态

### 3.2 方案

聚合连接（team）：指的是网络连接的捆绑/组队，通过将多个实际网卡（team-slave）整个为逻辑上的单个连接，实现负载均衡、热备份等单块网卡难以完成的特殊功能。

聚合连接的类型：热备份activebackup、轮询负载均衡roundrobin。

定义聚合连接的类型配置时，采用JSON语法标记，主要特点如下：

- 标记一个对象 —— { 对象 }
- 每一个对象 —— 名称:值
- 每一个字符串 —— "字符串"

[Top](#)

热备份-聚合连接（activebackup）：

```
01.  { "runner":{ "name":"activebackup" } }
```

负载均衡-聚合连接 (roundrobin) :

```
01.  { "runner":{ "name":"roundrobin" } }
```

### 3.3 步骤

除了所配置的IP地址不一样以外，在server0、desktop0主机上的其他操作相同。此处仅列出在server0上的配置过程。

实现此案例需要按照如下步骤进行。

#### 步骤一：准备练习用网卡环境

新建的聚合连接将组合新增加的两块网卡eth1、eth2。

```
01.  [root@server0 ~]# ifconfig
02.  eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
03.      inet 172.25.0.11 netmask 255.255.255.0 broadcast 172.25.0.255
04.      inet6 2003:ac18::305 prefixlen 64 scopeid 0x0<global>
05.      inet6 fe80::5054:ff:fe00:b prefixlen 64 scopeid 0x20<link>
06.      ether 52:54:00:00:00:0b txqueuelen 1000 (Ethernet)
07.      RX packets 172995 bytes 23870389 (22.7 MiB)
08.      RX errors 0 dropped 0 overruns 0 frame 0
09.      TX packets 54053 bytes 34274222 (32.6 MiB)
10.      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
11.
12.  eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
13.      ether 52:54:00:f8:86:c1 txqueuelen 1000 (Ethernet)
14.      RX packets 104217 bytes 5437855 (5.1 MiB)
15.      RX errors 0 dropped 0 overruns 0 frame 0
16.      TX packets 171 bytes 17171 (16.7 KiB)
17.      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
18.
19.  eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
20.      ether 52:54:00:38:79:d9 txqueuelen 1000 (Ethernet)
21.      RX packets 104118 bytes 5428927 (5.1 MiB)
22.      RX errors 0 dropped 2060 overruns 0 frame 0
23.      TX packets 0 bytes 0 (0.0 B)
24.      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

[Top](#)

## 步骤二：创建聚合连接配置

### 1) 新建聚合连接

```
01. [root@server0 ~]# nmcli con add con-name team0 type team ifname team0 confi
02. Connection 'team0' (8e61d730-50ff-4a7b-8ca0-fcf5955f6ea7) successfully added.
```

### 2) 配置IPv4地址

```
01. [root@server0 ~]# nmcli con modify team0 ipv4.method manual ipv4.addresses '1
```

### 3) 新建聚合成员连接

```
01. [root@server0 ~]# nmcli con add con-name team0-p1 type team-slave ifname eth
02. Connection 'team0-p1' (a62d23a2-9a2a-4855-8fbc-60ce1fd43f0b) successfully add
03. [root@server0 ~]# nmcli con add con-name team0-p2 type team-slave ifname eth
04. Connection 'team0-p2' (f4d4980e-8123-4840-89ac-1af148cc2eea) successfully adde
```

## 步骤三：激活聚合连接

### 1) 激活聚合连接

```
01. [root@server0 ~]# nmcli connection up team0
02. Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkMana
```

### 2) 激活聚合成员连接

```
01. [root@server0 ~]# nmcli connection up team0-p1
02. Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkMana
03. [root@server0 ~]# nmcli connection up team0-p2
04. Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkMana
```

[Top](#)

## 步骤四：确认聚合连接状态

### 1) 查看聚合连接地址

```
01. [root@server0 ~]# ifconfig team0
02. team0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
03.     inet 172.16.3.20 netmask 255.255.255.0 broadcast 172.16.3.255
04.     inet6 fe80::c80d:eff:fe08:ca57 prefixlen 64 scopeid 0x20<link>
05.     ether ca:0d:ef:08:ca:57 txqueuelen 0 (Ethernet)
06.     RX packets 0 bytes 0 (0.0 B)
07.     RX errors 0 dropped 36 overruns 0 frame 0
08.     TX packets 68 bytes 8695 (8.4 KiB)
09.     TX errors 0 dropped 1 overruns 0 carrier 0 collisions 0
```

### 2) 查看聚合连接运行状态

```
01. [root@server0 ~]# teamdctl team0 state
02. setup:
03.     runner: activebackup //运行模式/类型
04.     ports:
05.         eth1 //成员网卡1
06.         link watches:
07.             link summary: up
08.             instance[link_watch_0]:
09.                 name: ethtool
10.                 link: up
11.         eth2 //成员网卡2
12.         link watches:
13.             link summary: up
14.             instance[link_watch_0]:
15.                 name: ethtool
16.                 link: up
17.     runner:
18.         active port: eth1 //当前活动的成员网卡
19.
```

## 4 案例4：Linux工程师 综合测试

[Top](#)

### 4.1 问题

根据本文提供的练习步骤完成所有练习案例。

## 4.2 方案

开始练习之前，先依次重置虚拟机环境。

```
01. [root@room9pc13 ~]# rht-vmctl reset classroom
02. [root@room9pc13 ~]# rht-vmctl reset server
03. [root@room9pc13 ~]# rht-vmctl reset desktop
```

## 4.3 步骤

实现此案例需要按照如下步骤进行。

### 步骤01：配置SELinux

案例概述：

确保SELinux处于强制启用模式。

解题参考：

```
01. [root@server0 ~]# vim /etc/selinux/config //永久配置
02. SELINUX=enforcing
03. [root@server0 ~]# setenforce 1 //临时配置
```

### 步骤02：自定义用户环境（别名设置）

案例概述：

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。

解题参考：

```
01. [root@server0 ~]# vim /etc/bashrc //修改初始文件
02. alias qstat='/bin/ps -Ao pid,tt,user,fname,rsz' //设置别名
03.
04. [root@server0 ~]# source /etc/bashrc //或重登录后生效
05. [root@server0 ~]# qstat //确认别名可用
```

### 步骤03：配置防火墙端口转发

案例概述：

在系统server0、desktop0配置防火墙，要求如下：

[Top](#)



- 除了172.34.0.0/24网段以外，其它客户机都可以访问虚拟机server0、desktop0
- 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
- 上述设置必须永久有效

解题参考：

```
01. [root@server0 ~]# systemctl restart firewalld
02. [root@server0 ~]# systemctl enable firewalld
03. [root@server0 ~]# firewall-cmd --set-default-zone=trusted //默认全部允许
04. [root@server0 ~]# firewall-cmd --permanent --add-source=172.34.0.0/24 --zone=block //阻止个别网段
05.
06. [root@server0 ~]# firewall-cmd --permanent --zone=trusted --add-forward-port=port=5423:80
07. [root@server0 ~]# firewall-cmd --reload //重载防火墙策略
```

## 步骤04：配置链路聚合

案例概述：

在server0.example.com和desktop0.example.com之间按以下要求配置一个链路：

- 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

解题参考：

```
01. [root@server0 ~]# nmcli connection add con-name team0 type team ifname team0
02. [root@server0 ~]# nmcli connection add con-name team0-p1 type team-slave ifname team0-p1
03. [root@server0 ~]# nmcli connection add con-name team0-p2 type team-slave ifname team0-p2
04. [root@server0 ~]# nmcli con modify team0 ipv4.method manual ipv4.addresses "172.16.3.20/24"
05.
06. [root@server0 ~]# nmcli connection up team0 //激活聚合连接
07. [root@server0 ~]# nmcli con up team0-p1 //激活成员连接1
08. [root@server0 ~]# nmcli con up team0-p2 //激活成员连接2
09. [root@server0 ~]# teamdctl team0 state //确认连接状态
```

## 步骤05：配置IPv6地址

案例概述：

在您的考试系统上配置接口eth0使用下列 IPv6 地址：

[Top](#)

- 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

解题参考：

```

01. [root@server0 ~]# nmcli connection show //获知连接名称
02. NAME          UUID          TYPE          DEVICE
03. System eth0    5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03 802-3-ethernet eth0
04.
05. [root@server0 ~]# nmcli connection modify "System eth0" ipv6.method manual \
06.     ipv6.addresses 2003:ac18::305/64
07. [root@server0 ~]# nmcli connection up "System eth0"
08.
09. //设置固定主机名，避免误操作（若有必要，还可进一步配置静态IP地址/默认网关/DN
10. [root@server0 ~]# vim /etc/hostname
11. server0.example.com

```

## 步骤06：配置本地邮件服务

案例概述：

在系统 desktop0 上执行下列操作，将其配置为后端邮件服务：

- lab smtp-nullclient setup

在系统 server0 上配置邮件服务，满足以下要求：

- 这个系统不接收外部发送来的邮件
- 在这个系统上本地发送的任何邮件都会自动路由到 smtp0.example.com
- 从这个系统上发送的邮件显示来自于 desktop0.example.com
- 您可以在这个系统上发送邮件到本地用户student来测试您的配置，最终将会由系统 desktop0 上的用户 student 收到这封邮件

解题参考：

```

01. [root@server0 ~]# vim /etc/postfix/main.cf
02. relayhost = [smtp0.example.com] //后端邮件服务器
03. inet_interfaces = loopback-only //仅本机
04. myorigin = desktop0.example.com //发件来源域
05. mynetworks = 127.0.0.0/8 [::1]/128 //信任网络
06. mydestination = //此行的值设为空
07.
08. [root@server0 ~]# systemctl restart postfix
09. [root@server0 ~]# systemctl enable postfix
10.

```

[Top](#)

```

11. [root@server0 ~]# echo 'Mail Data.' | mail -s 'Test1' student
12.                                     //在server0发信测试
13. [root@server0 ~]# mail -u student           //在server0无邮件
14. No mail for student
15. [root@desktop0 ~]# mail -u student         //在desktop0上可收到这封邮件
16. ... ..

```

## 步骤07：通过Samba发布共享目录

案例概述：

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

解题参考：

```

01. [root@server0 ~]# yum -y install samba
02. [root@server0 ~]# mkdir /common
03. [root@server0 ~]# setsebool -P samba_export_all_rw=on //取消SELinux限制
04. [root@server0 ~]# useradd harry ; pdbedit -a harry //启用共享账号并设密码
05.
06. [root@server0 ~]# vim /etc/samba/smb.conf
07. [global]
08. workgroup = STAFF //修改此行，指定工作组名
09. [common]
10. path = /common
11. hosts allow = 172.25.0.0/24 //只允许指定网段访问
12. [root@server0 ~]# systemctl restart smb
13. [root@server0 ~]# systemctl enable smb

```

## 步骤08：配置多用户Samba挂载

案例概述：

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

[Top](#)

解题参考：

在server0上——

```
01. [root@server0 ~]# mkdir /devops
02. [root@server0 ~]# useradd kenji ; pdbedit -a kenji
03. [root@server0 ~]# useradd chihiro ; pdbedit -a chihiro
04.
05. [root@server0 ~]# setfacl -m u:chihiro:rwX /devops/ //调整目录权限
06.
07. [root@server0 ~]# vim /etc/samba/smb.conf
08. .. ..
09. [devops]
10.     path = /devops
11.     write list = chihiro
12.     hosts allow = 172.25.0.0/24 //只允许指定网域访问
13. [root@server0 ~]# systemctl restart smb
```

在desktop0上——

```
01. [root@desktop0 ~]# yum -y install samba-client cifs-utils
02. [root@desktop0 ~]# smbclient -L server0 //查看对方提供了哪些共享
03. .. .. //无需密码，直接按Enter键确认
04. [root@desktop0 ~]# mkdir /mnt/dev //创建挂载点
05. [root@desktop0 ~]# vim /etc/fstab
06. //server0.example.com/devops /mnt/dev cifs username=kenji,password=atenort
07.
08. [root@desktop0 ~]# mount -a //检查配置并挂载资源
```

验证多用户访问（在desktop0上）：普通用户切换为chihiro 身份即可读写。

```
01. [root@desktop0 ~]# su - student //切换到普通用户
02. [student@desktop0 ~]$ su - chihiro
03. [student@desktop0 ~]$ cifscreds add -u chihiro server0 //提交新认证凭据
04. Password: //提供Samba用户chihiro的密码
05. [student@desktop0 ~]$ touch /mnt/dev/b.txt //确认有写入权限
```

[Top](#)

步骤09：配置NFS共享服务

案例概述：

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
- 网络用户 ldapuser0 能以读写方式访问 /protected/project

解题参考：

[练习环境：lab nfskrb5 setup]

```
01. [root@server0 ~]# mkdir -p /public /protected/project //创建共享目录
02. [root@server0 ~]# chown ldapuser0 /protected/project/ //调整目录访问权限
03.
04. [root@server0 ~]# wget -O /etc/krb5.keytab \
05.     http://classroom.example.com/pub/keytabs/server0.keytab //下载并部署服务端
06. [root@server0 ~]# vim /etc/exports //配置NFS共享
07. /public 172.25.0.0/24(ro)
08. /protected 172.25.0.0/24(rw,sec=krb5p)
09. [root@server0 ~]# systemctl start nfs-secure-server nfs-server //启用两个服务
10. [root@server0 ~]# systemctl enable nfs-secure-server nfs-server
11. [root@server0 ~]# exportfs -rv //必要时更新共享配置
```

## 步骤10：挂载NFS共享

案例概述：

在 desktop0 上挂载一个来 classroom.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

解题参考：

[练习环境：lab nfskrb5 setup]

```
01. [root@desktop0 ~]# mkdir -p /mnt/nfsmount /mnt/nfssecure
02. [root@desktop0 ~]# wget -O /etc/krb5.keytab \
03.     http://classroom.example.com/pub/keytabs/desktop0.keytab //下载部署客户端密
04. [root@desktop0 ~]# systemctl start nfs-secure //启用安全NFS的客户端服务
05. [root@desktop0 ~]# systemctl enable nfs-secure
06.
```

```

07. [root@desktop0 ~]# showmount -e server0 //查看对方提供了哪些共享
08. Export list for server0:
09. /protected 172.25.0.0/24
10. /public 172.25.0.0/24
11. [root@desktop0 ~]# vim /etc/fstab //配置开机挂载
12. ... ..
13. server0.example.com:/public /mnt/nfsmount nfs _netdev 0 0
14. server0.example.com:/protected /mnt/nfssecure nfs sec=krb5p,_netdev 0 0
15. [root@desktop0 ~]# mount -a //检查配置并挂载资源
16.
17. [root@desktop0 ~]# ssh ldapuser0@desktop0 //SSH登入以获取通行证
18. ldapuser0@desktop0's password: //密码kerberos
19. [ldapuser0@desktop0 ~]$ touch /mnt/nfssecure/project/a.txt //写入测试

```

## 步骤11：实现一个web服务器

案例概述：

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
- 来自于 example.com 域的客户端可以访问此Web服务
- 拒绝来自于 my133t.org 域（172.34.0.0/24）的客户端访问此Web服务

解题参考：

```

01. [root@server0 ~]# yum -y install httpd
02. [root@server0 ~]# vim /etc/httpd/conf.d/00-default.conf
03. <VirtualHost *:80> //添加第一个（默认）虚拟主机
04.     ServerName server0.example.com
05.     DocumentRoot /var/www/html
06. </VirtualHost>
07. [root@server0 ~]# cd /var/www/html/ //下载并部署给定的首页文件
08. [root@server0 html]# wget -O index.html \
09.     http://classroom.example.com/pub/materials/station.html
10.
11. [root@server0 html]# systemctl restart httpd
12. [root@server0 html]# systemctl enable httpd

```

[Top](#)

## 步骤12：配置安全web服务

案例概述：

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
- 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

解题参考：

```
01. [root@server0 ~]# yum -y install mod_ssl //安装模块包
02. [root@server0 ~]# cd /etc/pki/tls/certs/ //下载并部署证书、密钥
03. [root@server0 certs]# wget http://classroom.example.com/pub/example-ca.crt
04. [root@server0 certs]# wget \
05.     http://classroom.example.com/pub/tls/certs/server0.crt
06. [root@server0 certs]# cd /etc/pki/tls/private/
07. [root@server0 private]# wget \
08.     http://classroom.example.com/pub/tls/private/server0.key
09.
10. [root@server0 private]# vim /etc/httpd/conf.d/ssl.conf
11. <VirtualHost _default_:443>
12.     DocumentRoot "/var/www/html"
13.     ServerName server0.example.com:443
14.     .. .. //修改第100、107、122行
15.     SSLCertificateFile /etc/pki/tls/certs/server0.crt
16.     SSLCertificateKeyFile /etc/pki/tls/private/server0.key
17.     SSLCACertificateFile /etc/pki/tls/certs/example-ca.crt
18. </VirtualHost>
19. [root@server0 private]# systemctl restart httpd
```

### 步骤13：配置虚拟主机

案例概述：

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

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` 已经提供对主机名 `www0.example.com` 的域名解析。

解题参考：

[Top](#)

```

01. [root@server0 ~]# mkdir /var/www/virtual
02. [root@server0 ~]# cd /var/www/virtual/ //下载并部署给定的首页文件
03. [root@server0 virtual]# wget -O index.html \
04.    http://classroom.example.com/pub/materials/www.html
05.
06. [root@server0 virtual]# vim /etc/httpd/conf.d/01-www0.conf
07. <VirtualHost *:80>
08.     ServerName www0.example.com
09.     DocumentRoot /var/www/virtual
10. </VirtualHost>
11. [root@server0 virtual]# systemctl restart httpd

```

## 步骤14：配置web内容的访问

### 案例概述：

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

### 解题参考：

```

01. [root@server0 ~]# mkdir /var/www/html/private
02. [root@server0 ~]# cd /var/www/html/private/ //下载并部署给定的首页文
03. [root@server0 private]# wget -O index.html \
04.    http://classroom.example.com/pub/materials/private.html
05.
06. [root@server0 private]# vim /etc/httpd/conf.d/00-default.conf
07. .. ..
08. <Directory /var/www/html/private>
09.     Require ip 127.0.0.1 ::1 172.25.0.11 //仅允许本机IP访问
10. </Directory>
11. [root@server0 private]# systemctl restart httpd

```

## 步骤15：实现动态WEB内容

### 案例概述：

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

[Top](#)



- 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

解题参考：

```

01. [root@server0 ~]# yum -y install mod_wsgi
02. [root@server0 ~]# mkdir /var/www/webapp0
03.
04. [root@server0 ~]# cd /var/www/webapp0 //下载并部署给定的动态WEB程序
05. [root@server0 webapp0]# wget
06. http://classroom.example.com/pub/materials/webinfo.wsgi
07.
08. [root@server0 webapp0]# vim /etc/httpd/conf.d/02-webapp0.conf
09. Listen 8909 //增加对新端口的监听
10. <VirtualHost *:8909>
11.     ServerName webapp0.example.com
12.     DocumentRoot /var/www/webapp0
13.     WSGIScriptAlias / /var/www/webapp0/webinfo.wsgi //访问Web根自动转向程
14. </VirtualHost>
15.
16. [root@server0 webapp0]# semanage port -a -t http_port_t -p tcp 8909
17. //开启非标准端口
18. [root@server0 webapp0]# systemctl restart httpd

```

## 步骤16：配置一个数据库

案例概述：

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。

解题参考：

### 1) 安装、配置

```

01. [root@server0 ~]# yum -y install mariadb-server mariadb
02. [root@server0 ~]# vim /etc/my.cnf
03. [mysqld]

```

[Top](#)

```
04. skip-networking //添加此行，跳过网络
05. [root@server0 ~]# systemctl restart mariadb
06. [root@server0 ~]# systemctl enable mariadb
```

## 2) 设密码、建库

```
01. [root@server0 ~]# mysqladmin -u root -p password 'atenorth' //设置密码
02. [root@server0 ~]# mysql -u root -p
03. MariaDB [(none)]> CREATE DATABASE Contacts;
04. MariaDB [(none)]> GRANT select ON Contacts.* to Raikon@localhost IDENTIFIED BY 'a
05. MariaDB [(none)]> DELETE FROM mysql.user WHERE Password=''; //删除空密码则
06. //!!注意：设好root密码再做
07. MariaDB [(none)]> QUIT
```

## 3) 导入库

```
01. [root@server0 ~]# wget http://classroom.example.com/pub/materials/users.sql
02. [root@server0 ~]# mysql -u root -p Contacts < users.sql
```

## 步骤17：数据库查询（填空）

### 案例概述：

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？

### 解题参考：

```
01. [root@server0 ~]# mysql -u root -p
02. Enter password:
03. MariaDB [Contacts]> USE Contacts;
04. MariaDB [Contacts]> SELECT name FROM base WHERE password='solicitous';
05. +-----+
06. | name |
07. +-----+
08. | James |
09. +-----+
10. MariaDB [Contacts]> SELECT count(*) FROM base,location WHERE base.name='Bar
```

[Top](#)

11. 1
12. MariaDB [Contacts]> QUIT

## 步骤18：创建一个脚本

案例概述：

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

解题参考：

```
01. [root@server0 ~]# vim /root/foo.sh
02. #!/bin/bash
03. if [ "$1" = "redhat" ]; then
04.     echo "fedora"
05. elif [ "$1" = "fedora" ]; then
06.     echo "redhat"
07. else
08.     echo "/root/foo.sh redhat|fedora" >&2
09. fi
10. [root@server0 ~]# chmod +x /root/foo.sh
```

## 步骤19：创建一个添加用户的脚本

案例概述：

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>

解题参考：

```
01. [root@server0 ~]# vim /root/batchusers
02. #!/bin/bash
```

[Top](#)

```
03.  if [ $# -eq 0 ]; then
04.  echo "Usage: /root/batchusers <userfile>"
05.  exit 1
06.  fi
07.  if [ ! -f $1 ]; then
08.  echo "Input file not found"
09.  exit 2
10.  fi
11.  for name in $(cat $1)
12.  do
13.      useradd -s /bin/false $name
14.  done
15.  [root@server0 ~]# chmod +x /root/batchusers
```