习题参考答案：

案例1：设置SELinux保护

为虚拟机A、虚拟机B 配置SELinux

1）确保 SELinux 处于宽松模式（permissive）

2）在每次重新开机后，此设置必须仍然有效

虚拟机A

[root@svr7 /]# getenforce #查看当前运行模式

Enforcing

[root@svr7 /]# setenforce 0 #修改当前运行模式

[root@svr7 /]# getenforce

Permissive

[root@svr7 /]# vim /etc/selinux/config

SELINUX=permissive

虚拟机B：同上

案例2：实现虚拟机A 的Web服务

1. 利用httpd软件搭建Web服务，页面显示内容为 小蝌蚪找妈妈

虚拟机A：

[root@svr7 /]# yum -y install httpd

[root@svr7 /]# echo小蝌蚪找妈妈 > /var/www/html/index.html

[root@svr7 /]# cat /var/www/html/index.html

[root@svr7 /]# /usr/sbin/httpd

[root@svr7 /]# curl http://192.168.4.7

案例3：实现虚拟机A 的防火墙配置

1）修改虚拟机A防火墙配置，明确拒绝所有客户端访问（将默认区域设置为block）

2）在虚拟机B上,测试能否访问虚拟机A 的Web服务

3）在虚拟机B上,测试能否 ping通 虚拟机A

虚拟机A：修改默认区域

]# firewall-cmd --set-default-zone=block

]# firewall-cmd --get-default-zone

虚拟机B

]# curl http://192.168.4.7 #失败

]# ping 192.168.4.7 #失败

案例4：实现虚拟机A 的防火墙配置

1）修改虚拟机A防火墙配置，将默认区域修改为public

2）在虚拟机B上,测试能否访问虚拟机A的Web服务

3）在虚拟机B上,测试能否 ping通 虚拟机A

虚拟机A：修改默认区域

]# firewall-cmd --set-default-zone=public

]# firewall-cmd --get-default-zone

虚拟机B

]# curl http://192.168.4.7 #失败

]# ping 192.168.4.7 #成功

案例5：实现虚拟机A的防火墙配置

1）修改虚拟机A防火墙配置，将默认区域修改为public

2）修改虚拟机A永久的防火墙配置，在public区域中添加http协议

3）在虚拟机B上,测试能否访问虚拟机A 的Web服务

虚拟机A：修改默认区域

]# firewall-cmd --reload #加载防火墙永久策略

]# firewall-cmd --zone=public --list-all

]# firewall-cmd --permanent --zone=public --add-service=http

]# firewall-cmd --reload #加载防火墙永久策略

]# firewall-cmd --zone=public --list-all

虚拟机B

]# curl http://192.168.4.7 #成功

]# ping 192.168.4.7 #成功

一、环境准备

开启虚拟机A与虚拟机B

* **两台机器**设置SELinux运行模式

[root@svr7 ~]# **getenforce**

Enforcing

[root@svr7 ~]# **setenforce 0**

[root@svr7 ~]# **getenforce**

Permissive

[root@svr7 ~]# **vim /etc/selinux/config**

SELINUX=**permissive**

* **两台机器**设置防火墙（停止防火墙服务）

[root@svr7 ~]# **systemctl stop firewalld**

[root@svr7 ~]# **systemctl disable firewalld**

**一、服务的管理**

用户---》systemd---》服务

•上帝进程：systemd

•Linux系统和服务管理器

–是内核引导之后加载的**第一个初始化进程**（PID=1）

–负责掌控整个Linux的运行/服务资源组合

•一个更**高效**的系统&服务管理器

–开机服务**并行启动**，各系统服务间的精确依赖

–配置目录：/etc/systemd/system/

–服务目录：/lib/systemd/system/

–主要管理工具：systemctl

[root@svr7 ~]# systemctl -t service --all #列出所有的服务

•对于服务的管理（与手动启动有冲突）

**systemctl restart 服务名** #重起服务

systemctl start 服务名 #开启服务

systemctl stop 服务名 #停止服务

systemctl status 服务名 #查看服务当前的状态

**systemctl enable 服务名**  #设置服务开机自启动

systemctl disable 服务名 #设置服务禁止开机自启动

systemctl is-enabled 服务名 #查看服务是否开机自启

]# yum -y install httpd

]# > /etc/resolv.conf #清除文件内容

]# killall httpd #杀死手动启动的httpd

]# systemctl restart httpd #重启httpd服务

]# systemctl status httpd #查看服务httpd状态

]# systemctl enable httpd #设置httpd开机自启动

Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.

]# systemctl is-enabled httpd #查看httpd是否是开机自启动

]# systemctl disable httpd #关闭httpd开机自启动

Removed symlink /etc/systemd/system/multi-user.target.wants/httpd.service.

]# systemctl is-enabled httpd #查看httpd是否是开机自启动

**管理运行级别**

RHEL6:运行级别 300

0：关机 0个服务

1：单用户模式（基本功能的实现，破解Linux密码） 50个服务

2：多用户字符界面（不支持网络） 80个服务

**3：多用户字符界面（支持网络）服务器默认运行级别** 100个服务

4：未定义 0个服务

**5：图形界面 300个服务**

6：重起 0个服务

切换运行级别：init 数字

RHEL7：运行模式（运行级别）

字符模式：multi-user.target

图形模式：graphical.target

当前直接切换到字符模式

]# systemctl isolate multi-user.target #相当于原来的init 3

当前直接切换到图形模式

]# systemctl isolate graphical.target #相当于原来的init 5

查看每次开机默认进入模式

[root@svr7 /]# systemctl get-default

设置永久策略，每次开机自动进入multi-user.target

[root@svr7 /]# systemctl set-default multi-user.target

[root@svr7 /]# reboot

**二、Web服务器简介**

* 基于 **B/S** （Browser/Server）架构的网页服务
  + 服务端提供网页
  + 浏览器下载并显示网页
* **H**yper **T**ext **M**arkup **L**anguage，超文本标记语言
* **H**yper **T**ext **T**ransfer **P**rotocol，超文本传输协议

三步骤策略：装包、配置、启服务

实现Web功能软件：httpd、Nginx、Tomcat

httpd由软件基金会Apache

**虚拟机A：构建基本Web服务**

]# **yum -y install httpd**

]# **rpm -q httpd**

]# **echo NSD Web Server > /var/www/html/index.html**

]# **> /etc/resolv.conf**

]# **systemctl restart httpd** #启动服务

]# **curl http://192.168.4.7**  #测试访问

NSD Web Server

* 提供的默认配置
  + **Listen：监听地址:端口（80）**
  + **ServerName：本站点注册的DNS名称（空缺）**
  + **DocumentRoot：网页根目录（/var/www/html）**
  + DirectoryIndex：起始页/首页文件名（index.html）

**主配置文件：/etc/httpd/conf/httpd.conf**

常见错误：服务启动失败

[root@svr7 ~]# systemctl restart httpd

Job for httpd.service failed because the control process exited with error code. See "systemctl status httpd.service" and "**journalctl -xe**" for details.

[root@svr7 ~]# journalctl -xe

**DocumentRoot：网页文件根目录（/var/www/html）**

虚拟机A

]# **mkdir /var/www/myweb**

]# **echo wo shi myweb > /var/www/myweb/index.html**

]# **vim /etc/httpd/conf/httpd.conf**

…….此处省略一万字

**DocumentRoot "/var/www/myweb"**

…….此处省略一万字

]# **systemctl restart httpd** #重启服务

]# **curl 192.168.4.7**

wo shi myweb

* **基于网页文件目录，进行访问控制**

当子目录没有规则，默认继承上一级目录规则

针对此目录有单独配置，则不继承上一级目录规则

<Directory />

Require all denied #拒绝所有人访问

</Directory>

<Directory "/var/www">

Require all granted #允许所有人访问

</Directory>

/webroot/abc/test #拒绝所有人访问

/var/www/myweb #允许所有人访问

**虚拟机A**：

]# **mkdir /webroot**

]# **echo wo shi webroot > /webroot/index.html**

]# **vim /etc/httpd/conf/httpd.conf**

…….此处省略一万字

**DocumentRoot "/webroot" #修改配置内容**

**<Directory "/webroot"> #针对于/webroot路径**

**Require all granted #允许所有人访问**

**</Directory>**

…….此处省略一万字

]# **systemctl restart httpd**  #重启服务

]# **curl http://192.168.4.7**

wo shi webroot

客户端浏览器上访问http://192.168.4.7----->服务端192.168.4.7-----》http----》httpd软件---》/etc/httpd/conf/httpd.conf---》DocumentRoot-----》/webroot----》index.html

客户端：http://192.168.4.7

服务端：/webroot

**http://192.168.4.7= DocumentRoot 设置路径**

客户端：http://192.168.4.7/abc

服务端：/webroot/abc

DocumentRoot设置为/webroot

客户端：http://192.168.4.7/webroot/abc/test

服务端：/webroot/webroot/abc/test

* **网络路径与实际路径**

DocumentRoot ---》/webroot

网络路径：curl http://192.168.4.7

实际服务：/webroot

访问过程：客户端curl http://192.168.4.7--->服务端192.168.4.7---》80---》httpd---》/etc/httpd/conf/httpd.conf---》DocumentRoot ---》/webroot---》index.html

网络路径：curl http://192.168.4.7/abc

实际服务： /webroot/abc/index.html

DocumentRoot ---》/webroot

网络路径：curl http://192.168.4.7/webroot/abc

实际服务：/webroot/webroot/abc/index.html

虚拟机A

]# **mkdir /webroot/abc**

]# **echo wo shi abc > /webroot/abc/index.html**

]# **curl http://192.168.4.7/abc/**

* **Listen：监听IP地址: 监听端口（80）**

端口:数字编号起到标识作用，标识协议

http协议默认端口：80

ftp协议默认端口：21

建议自定义端口时大于1024，端口的极限65535

[root@svr7 ~]# vim /etc/httpd/conf/httpd.conf

…….此处省略一万字

**Listen 80**

**Listen 8000**

…….此处省略一万字

[root@svr7 ~]# **systemctl restart httpd**

[root@svr7 ~]# **curl http://192.168.4.7:8000**

**三、虚拟Web主机**

* 虚拟Web主机
  + 由同一台服务器提供多个不同的Web站点
* 区分方式
  + 基于**域名**的虚拟主机
  + 基于**端口**的虚拟主机
  + 基于IP地址的虚拟主机
* 配置文件路径
  + /etc/httpd/conf/httpd.conf #主配置文件
  + /etc/httpd/conf.d/\*.conf #调用配置文件
* 为每个虚拟站点添加配置

<VirtualHost IP地址:端口>

ServerName 此站点的DNS名称

DocumentRoot 此站点的网页根目录

</VirtualHost>

虚拟机A：

]# **vim /etc/httpd/conf.d/haha.conf**

**<VirtualHost \*:80>** #在所有IP地址监听80

**ServerName www.qq.com**  #网站的域名

**DocumentRoot /var/www/qq** #网页文件路径

**</VirtualHost>**

**<VirtualHost \*:80>**

**ServerName www.lol.com**

**DocumentRoot /var/www/lol**

**</VirtualHost>**

]# mkdir /var/www/qq /var/www/lol

]# echo wo shi QQ > /var/www/qq/index.html

]# echo wo shi lol > /var/www/lol/index.html

]# systemctl restart httpd

采用/etc/hosts文件直接解析域名，**只为本机解析**

]# **vim /etc/hosts**

…….此处省略一万字

**192.168.4.7 www.qq.com www.lol.com**

]# **curl http://www.qq.com**

]# **curl http://www.lol.com**

一旦使用虚拟Web主机功能，所有的网站都必须使用虚拟Web进行呈现

虚拟机A

[root@svr7 ~]# vim /etc/httpd/conf.d/haha.conf

…….此处省略一万字

<VirtualHost \*:80>

ServerName www.xixi.com

DocumentRoot /webroot

</VirtualHost>

[root@svr7 ~]# systemctl restart httpd

[root@svr7 ~]# vim /etc/hosts

…….此处省略一万字

192.168.4.7 www.qq.com www.lol.com www.xixi.com

[root@svr7 ~]# curl http://www.xixi.com

* **基于端口的虚拟Web主机**

[root@svr7 ~]# vim /etc/httpd/conf.d/haha.conf

<VirtualHost \*:80>

ServerName **www.qq.com**

DocumentRoot /var/www/qq

</VirtualHost>

**Listen 8080**

<VirtualHost \*:**8080**>

ServerName **www.qq.com**

DocumentRoot /var/www/lol

</VirtualHost>

[root@svr7 ~]# systemctl restart httpd

[root@svr7 ~]# curl http://www.qq.com:8080

[root@svr7 ~]# curl http://www.qq.com

四、NFS服务基础

* Network File System，网络文件系统
  + 用途：为客户机提供共享使用的文件夹
  + 协议：NFS（ 2049）、RPC（ 111）
* 所需软件包：nfs-utils
* 系统服务：nfs-server

虚拟机A：服务端

[root@svr7 ~]# rpm -q nfs-utils

nfs-utils-1.3.0-0.54.el7.x86\_64

[root@svr7 ~]# mkdir /public #创建共享目录

[root@svr7 ~]# echo haha > /public/h.txt

[root@svr7 ~]# echo xixi > /public/x.txt

[root@svr7 ~]# ls /public/

[root@svr7 ~]# vim /etc/exports

文件夹路径 客户机地址(权限)

/public \*(ro) #允许所有客户端进行只读访问

]# systemctl restart rpcbind #涉及动态端口，nfs-server依赖此服务

]# systemctl restart nfs-server

虚拟机B：客户端

[root@pc207 ~]# rpm -q nfs-utils

nfs-utils-1.3.0-0.54.el7.x86\_64

[root@pc207 ~]# mkdir /mnt/mynfs

]# mount 192.168.4.7:/public /mnt/mynfs

]# ls /mnt/mynfs

]# df -h #查看正在挂载的设备信息

**实现开机自动挂载**

\_netdev：声明网络设备，系统在具备网络参数后，再进行挂载本设备

[root@pc207 ~]# vim /etc/fstab

……此处省略一万字

192.168.4.7**:**/public /mnt/mynfs nfs defaults**,**\_netdev 0 0

[root@pc207 ~]# umount /mnt/mynfs/

[root@pc207 ~]# ls /mnt/mynfs/

[root@pc207 ~]# mount -a

[root@pc207 ~]# ls /mnt/mynfs/

五、触发挂载（了解内容）

* 由 autofs 服务提供的“按需访问”机制
  + 只要访问挂载点就会触发响应，自动挂载指定设备
  + 闲置超过时限（默认5分钟）后，会自动卸载

[root@pc207 ~]# yum -y install autofs

[root@pc207 ~]# systemctl restart autofs

[root@pc207 ~]# ls /misc

[root@pc207 ~]# ls /misc/cd

触发挂载实现，必须多级的目录结构： /监控目录/挂载点目录

* 主配置文件 /etc/auto.master
  + 监控点目录 挂载配置文件的路径
* 挂载配置文件，比如 /etc/auto.misc
  + 触发点子目录 -挂载参数 :设备名

案例：虚拟机B访问/myauto/nsd,光驱设备挂载/ myauto/nsd

[root@pc207 ~]# **yum -y install autofs**

[root@pc207 ~]# **vim /etc/auto.master**

……此处省略一万字

**/myauto /opt/xixi.txt #增加一行新的配置**

……此处省略一万字

[root@pc207 ~]# **cp /etc/auto.misc /opt/xixi.txt**

[root@pc207 ~]# **vim /opt/xixi.txt #此文件仅保留一行**

**nsd -fstype=iso9660 :/dev/cdrom**

[root@pc207 ~]# **systemctl restart autofs**

[root@pc207 ~]# **ls /myauto/**

[root@pc207 ~]# **ls /myauto/nsd**

* **触发挂载进阶autofs与NFS**

虚拟机B访问/myauto/nfs,虚拟机A 的nfs共享/public挂载到/myauto/nfs

[root@pc207 ~]# yum -y install autofs

[root@pc207 ~]# vim /etc/auto.master

……此处省略一万字

**/myauto /opt/xixi.txt**

……此处省略一万字

[root@pc207 ~]# vim /opt/xixi.txt

nsd -fstype=iso9660 :/dev/cdrom

**nfs -fstype=nfs 192.168.4.7:/public**

[root@pc207 ~]# systemctl restart autofs

[root@pc207 ~]# ls /myauto/nfs

课后习题：

案例1:为虚拟机A 配置以下虚拟Web主机

- 实现三个网站的部署

- 实现客户端访问server0.example.com网页内容为 大圣归来

- 实现客户端访问www0.example.com网页内容为 大圣又归来

- 实现客户端访问webapp0.example.com网页内容为 大圣累了

案例2:为虚拟机A 使用自定Web根目录

调整 Web 站点 http://www0.example.com 的网页目录，要求如下：

1）新建目录 /webroot，作为此站点新的网页目录（提示：修改Web服务的访问控制）

2）确保站点 http://www0.example.com 仍然可访问

案例3：为虚拟机A 部署站点

为站点 webapp0.example.com 进行配置，要求如下：

1）此虚拟主机侦听在端口8909

2）从浏览器访问 http://webapp0.example.com:8909

案例4：普通NFS共享的实现

1.在虚拟机A上配置NFS服务，只读的方式共享目录 /public

2.在虚拟机B上访问NFS共享目录

a）将虚拟机A 的 /public 挂到本地 /mnt/nfsmount

b）这些文件系统在系统启动时自动挂载

案例5： autofs触发挂载

在虚拟机B上访问NFS共享目录，将 虚拟机A 的 /public完成触发挂载到本地 /mnt/nfsauto

* **参考答案**

案例1:为虚拟机A 配置以下虚拟Web主机

- 实现三个网站的部署

- 实现客户端访问server0.example.com网页内容为 大圣归来

- 实现客户端访问www0.example.com网页内容为 大圣又归来

- 实现客户端访问webapp0.example.com网页内容为 大圣累了

虚拟机A：

[root@A ~]# systemctl stop firewalld

[root@A ~]# setenforce 0

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

[root@A ~]# mkdir /var/www/nsd01

[root@A ~]# mkdir /var/www/nsd02

[root@A ~]# mkdir /var/www/nsd03

[root@A ~]# echo '大圣归来' > /var/www/nsd01/index.html

[root@A ~]# echo '大圣又归来' > /var/www/nsd02/index.html

[root@A ~]# echo '大圣累了' > /var/www/nsd03/index.html

[root@A ~]# vim /etc/httpd/conf.d/vhost.conf

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /var/www/nsd01

</VirtualHost>

<VirtualHost \*:80>

ServerName www0.example.com

DocumentRoot /var/www/nsd02

</VirtualHost>

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd03

</VirtualHost>

[root@A ~]# systemctl restart httpd

虚拟机B：测试

[root@B ~]# vim /etc/hosts

192.168.4.7 server0.example.com www0.example.com webapp0.example.com

[root@B ~]# curl server0.example.com

[root@B ~]# curl www0.example.com

[root@B ~]# curl webapp0.example.com

案例2:为虚拟机A 使用自定Web根目录

调整 Web 站点 http://www0.example.com 的网页目录，要求如下：

1）新建目录 /webroot，作为此站点新的网页目录

2）确保站点 http://www0.example.com 仍然可访问

虚拟机A：

[root@A ~]# vim /etc/httpd/conf.d/vhost.conf

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /var/www/nsd01

</VirtualHost>

<VirtualHost \*:80>

ServerName www0.example.com

DocumentRoot **/webroot**

</VirtualHost>

<VirtualHost \*:80>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd03

</VirtualHost>

[root@A ~]# mkdir /webroot

[root@A ~]# echo 'wo shi webroot' > /webroot/index.html

[root@A ~]# vim /etc/httpd/conf.d/myacl.conf

<Directory /webroot>

Require all granted

</Directory>

[root@A ~]# systemctl restart httpd

虚拟机B:

[root@B ~]# curl www0.example.com

wo shi webroot

[root@B ~]#

案例3：为虚拟机A 部署站点

为站点 webapp0.example.com 进行配置，要求如下：

1）此虚拟主机侦听在端口8909

2）从浏览器访问 http://webapp0.example.com:8909

虚拟机A：

[root@A ~]# vim /etc/httpd/conf.d/vhost.conf

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /var/www/nsd01

</VirtualHost>

<VirtualHost \*:80>

ServerName www0.example.com

DocumentRoot /webroot

</VirtualHost>

Listen 8909

<VirtualHost \*:8909>

ServerName webapp0.example.com

DocumentRoot /var/www/nsd03

</VirtualHost>

[root@A ~]# systemctl restart httpd

虚拟机B：

[root@B ~]# curl webapp0.example.com:8909

大圣累了

案例4：普通NFS共享的实现

1.在虚拟机A上配置NFS服务，只读的方式共享目录 /public

2.在虚拟机B上访问NFS共享目录

a）将虚拟机A 的 /public 挂到本地 /mnt/nfsmount

b）这些文件系统在系统启动时自动挂载

虚拟机A：

[root@A ~]# yum -y install nfs-utils

[root@A ~]# mkdir /public

[root@A ~]# touch /public/1.txt

[root@A ~]# ls /public/

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

/public \*(ro)

[root@A ~]# systemctl restart rpcbind #先重启

[root@A ~]# systemctl restart nfs-server #后重启

虚拟机B：

[root@B ~]# mkdir /mnt/nfsmount

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

192.168.4.7:/public /mnt/nfsmount nfs defaults,\_netdev 0 0

[root@B ~]# mount -a

[root@B ~]# df -h | tail -1

192.168.4.7:/public 17G 3.5G 14G 21% /mnt/nfsmount

[root@B ~]#

案例5： autofs触发挂载

在虚拟机B上访问NFS共享目录，将 虚拟机A 的 /public完成触发挂载到本地 /mnt/nfsauto

[root@pc207 ~]# yum -y install autofs

[root@pc207 ~]# vim /etc/auto.master

……此处省略一万字

**/mnt /etc/nsd.conf**

……此处省略一万字

[root@pc207 ~]# vim **/etc/nsd.conf**

**nfsauto -fstype=nfs 192.168.4.7:/public**

[root@pc207 ~]# systemctl restart autofs

[root@pc207 ~]# ls /mnt/nfsauto