RH358综合练习

```
   System                          IP              Role
workstation.lab.example.com        172.25.250.9    Asible控制节点
servera.lab.example.com            172.25.250.10       非受管节点
serverb.lab.example.com            172.25.250.11       非受管节点
serverc.lab.example.com            172.25.250.12       受管节点
serverd.lab.example.com            172.25.250.13       受管节点

# 相关信息：
        root的密码是redhat，
        devops用户作为ansible的管理员，密码为 redhat
 /home/devops/ansible作为ansible的工作目录。
每台服务器默认防火墙开启，SELinux为enforcing状态

# 准备工作
        将files目录上传到f0的/content目录中
```

第1题：Configure unmanaged host firewall access(配置防火墙)

- Configure the firewall on servera and serverb such that:(在servera和serverb上配置防火墙，要求如下：)
  - Clients within 172.25.250.0/24 have remote SSH access to your virtual systems（来自172.25.250.0/24的客户端可以通过SSH访问)
  - Clients within 172.24.250.0/24 do NOT have remote SSH access to your virtual systems（来自172.24.250.0/24的客户端不允许通过SSH访问)
  - Configure denial rules to use a "reject" instead of a "drop"（使用 reject 规则，而不是 drop 规则)

```
# 知识点考察：
 1）配置系统防火墙，以允许访问特定的服务或端口
 2）配置系统防火墙，以仅允许或拒绝来自特定网域或IP子网的访问

# 解题：
        [root@servera ~]# firewall-cmd --permanent --remove-service=ssh
        [root@servera ~]# firewall-cmd --permanent --zone=public --add-rich-rule="rule family=ipv4 source address=172.25.250.0/24 service name=ssh accept"
        [root@servera ~]# firewall-cmd --permanent --zone=public --add-rich-rule="rule family=ipv4 source address=172.24.250.0/24 service name=ssh reject"
        [root@servera ~]# firewall-cmd --reload

        [root@serverb ~]# firewall-cmd --permanent --remove-service=ssh
        [root@serverb ~]# firewall-cmd --permanent --zone=public --add-rich-rule="rule family=ipv4 source address=172.25.250.0/24 service name=ssh accept"
        [root@serverb ~]# firewall-cmd --permanent --zone=public --add-rich-rule="rule family=ipv4 source address=172.24.250.0/24 service name=ssh reject"
        [root@serverb ~]# firewall-cmd --reload

# 验证方法：
        [root@servera ~]# firewall-cmd --list-all
            public (active)
              ..........
              rich rules:
                rule family="ipv4" source address="172.25.250.0/24" service name="ssh" accept
                rule family="ipv4" source address="172.24.250.0/24" service name="ssh" reject

# 查询帮助方法：
 [root@servera ~]# man -k firewalld  {查询帮助，example3示例}
                firewalld.richlanguage (5) - Rich Language Documentation
    [root@servera ~]# man firewalld.richlanguage  {查询帮助，搜索example得到example3示例}
            # rule family="ipv4" source address="192.168.0.0/24" service name="tftp" log prefix="tftp" level="info" limit value="1/m" accept
```

第2题：Configure IPv6 addresses (配置IPV6地址)

- Configure the eth0 on servera and serverb with the following IPv6 addresses： (在 servera 和 serverb 上配置接口 eth0 使用下列 IPv6 地址：)
  - servera has the address 2018:ac18::432/64 (servera 上的地址是 2018:ac18::432/64)
  - serverb has the address 2018:ac18::43c/64 (serverb 上的地址是 2018:ac18::43c/64)
  - Both systems are reachable from systems in the 2018:ac18/64 network (两台主机可以被2018:ac18/64 网络所访问)
  - The address assignments persist across system reboots(地址需要在重启后仍然保持)
  - Both systems are also reachable via IPv4 at their current addresses (两台服务器已有的 IPv4 地址保持不变)

```
# 知识点考察：
        配置网络接口

# 解题：
    # 查看eth0网卡对应的配置文件
        [root@servera ~]# nmcli connection show
        NAME                UUID                                  TYPE      DEVICE
        Wired connection 1  4ae4bb9e-8f2d-3774-95f8-868d74edcc3c  ethernet  eth0
    # 进行修改地址
        [root@servera ~]# nmcli connection modify "Wired connection 1" ipv6.method manual ipv6.addresses 2018:ac18::432/64
        [root@servera ~]# nmcli connection up "Wired connection 1"
        [root@servera ~]# firewall-cmd --permanent --add-rich-rule='rule family=ipv6 source address=2018:ac18::/64 accept'
        [root@servera ~]# firewall-cmd --reload


        [root@serverb ~]# nmcli connection modify "Wired connection 1" ipv6.method manual ipv6.addresses 2018:ac18::43c/64
        [root@serverb ~]# nmcli connection up "Wired connection 1"
        [root@serverb ~]# firewall-cmd --permanent --add-rich-rule='rule family=ipv6 source address=2018:ac18::/64 accept'
        [root@serverb ~]# firewall-cmd --reload

# 验证方法：
        1、互ping测试
            [root@servera ~]# ping 2018:ac18::43c
                PING 2018:ac18::43c(2018:ac18::43c) 56 data bytes
                64 bytes from 2018:ac18::43c: icmp_seq=1 ttl=64 time=2.17 ms
        2、网卡查看测试
            [root@servera ~]# ip a
                2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
                ……
                    inet6 2018:ac18::432/64 scope global noprefixroute
                ……
```

第3题：Implement a DHCP server(实施一个 DHCP 服务器)

- On servera configure a DHCP server to provide both fixed and dynamic IPv4 addresses for the 192.168.1.0/24 network as follows：

- (在 servera 上配置一个 dhcp 服务器，自动分配 192.168.1.0/24 网段地址，要求如下：)
  - Dynamic configuration（动态配置内容）：
  - Dynamic IPs must be between 192.168.1.125 and 192.168.1.200（动态 IP 地址范围：192.168.1.125-192.168.1.200）
  - DNS server address：（DNS服务器地址：192.168.1.6）
  - DNS domain name：（DNS域名：remote.example.com）
  - Default gateway： （默认网关：192.168.1.200）
  - Default lease time：600 （默认租期： 600）
  - Max lease time：3000 （最大租期： 3000）
- Fixed address information：（需要为以下 MAC 地址分配固定 IP 地址：）
  - IP 192.168.1.10 is assigned to hardware address 52:54:00:02:fa:0c （192.168.1.10地址分配给 mac 地址: 52:54:00:02:fa:0c）
  - IP 192.168.1.11 is assigned to hardware address 52:54:00:02:fa:0d （192.168.1.11 地址分配给 mac 地址: 52:54:00:02:fa:0d）

```
# 知识点考察：
        使用DHCP来分配IPv4地址

# 准备工作
    [root@servera ~]# nmcli connection modify "Wired connection 2" ipv4.addresses 192.168.1.100/24 ipv4.method manual
    [root@servera ~]# nmcli connection up "Wired connection 2"

# 解题：
 # 安装组件
                [root@servera ~]# yum -y install dhcp-server
 # 修改配置文件
        [root@servera ~]# vim /etc/dhcp/dhcpd.conf
            authoritative;   # 指定权威dhcp服务器
            subnet 192.168.1.0 netmask 255.255.255.0 {    # 指定dhcp分配网段
              range 192.168.1.125 192.168.1.200;          # 地址池范围
              option domain-name-servers 192.168.1.6;     # dns地址
              option domain-name "remote.example.com";    # dns域名
              option routers 192.168.1.200;               # 默认网关
              default-lease-time 600;                     # 默认租期
              max-lease-time 3000;                        # 最大租期
            }
            # 考试中只有第一个mac地址可以找到，是真实机器，第二个mac地址不存在，写上分配即可
            host serverc {                                # 为serverc配置网卡mac与ip的绑定关系
              hardware ethernet 52:54:00:02:fa:0c;        # 指定mac地址
              fixed-address 192.168.1.10;                 # 指定该mac可以得到的IP
            }
            host serverd {
              hardware ethernet 52:54:00:01:fa:0d;
              fixed-address 192.168.1.11;
            }
        # 放行并启动服务
            [root@servera ~]# systemctl enable --now dhcpd
            [root@servera ~]# firewall-cmd --permanent --add-service=dhcp
            [root@servera ~]# firewall-cmd --reload

# 验证方法：
    [root@serverc ~]# nmcli connection show
        ..........
```

```
        Wired connection 3  422eafb5-dfd0-32d2-b00a-c6ffbb4132ce  ethernet  eth2
    [root@serverc ~]# nmcli connection up "Wired connection 3"
        Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/4)
    [root@serverc ~]#  ip a
        ..........
        4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
            link/ether 52:54:00:02:fa:0c brd ff:ff:ff:ff:ff:ff
            inet 192.168.1.10/24 brd 192.168.1.255 scope global dynamic noprefixroute
eth2
        ..........

# 查询帮助方法：
 # 修改配置文件：vim /usr/share/doc/dhcp-server/dhcpd.conf.example
```

第4题：Implement a DNS server(实施一个 DNS 服务器)

- On serverb，BIND has been installed，and partially configured. Complete the BIND configuration such that your DNS server serves both forward and reverse lookups for：
- (在 servera 上，bind 已经安装并做了部分配置。完成 bind 的配置，实现正向和反向解析：)
  - serverc.rh358.example.com 192.168.1.10
  - serverd.rh358.example.com 192.168.1.11
  - Make sure that any upstream or downstream caching nameservers are appropriately updated（确保任何上游或下游缓存名称服务器都是适当的）

```
# 知识点考察：
        管理DNS和DNS服务器，完成的区域文件配置授权域名服务器
        配置IPv4地址正向与反向查询

# 准备工作
    [root@servera ~]# yum install -y bind
    [root@servera ~]# wget http://classroom.example.com/files/named.conf
    [root@servera ~]# wget http://classroom.example.com/files/rh358.zone
    [root@servera ~]# wget http://classroom.example.com/files/rh358-reverse.zone
    [root@servera ~]# cp named.conf /etc/named.conf
        cp: overwrite '/etc/named.conf'? y
    [root@servera ~]# cp *.zone  /var/named/

# 解题：
    [root@servera 4]# cd /var/named/
    # 修改zone文件
        [root@servera named]# vim rh358.zone
                $TTL 86400
                rh358.example.com. IN  SOA servera.rh358.example.com.
root.servera.rh358.example.com. (2025060301 1H 5M 1W 1M)  #修改序列版本号至当前日期然后末尾加上01
                @                               IN  NS
servera.rh358.example.com.
                @                               IN  MX          10
servera.rh358.example.com.
                @                               IN  A           192.168.1.100      # 修改A记录
                mail        3600    IN  CNAME
servera.rh358.example.com.
                kerberos    3600    IN  CNAME
servera.rh358.example.com.
                servera                         IN  A   192.168.1.100             # 新增三条A记录
                serverc                         IN  A   192.168.1.10
                serverd                         IN  A   192.168.1.11

                ;;;;; rh358.example.com zone information
                $ORIGIN rh358.example.com.
        [root@servera named]# vim rh358-reverse.zone
                $TTL 86400
```

```
                1.168.192.in-addr.arpa. IN  SOA servera.rh358.example.com.
root.servera.rh358.example.com. (2025060301 1H 5M 1W 1M)  #修改序列版本号至当前日期然后末
尾加上01
                @               IN  NS          servera.rh358.example.com.
                100             IN  PTR         servera.rh358.example.com. # 新增三条PTR
记录
                10              IN  PTR         serverc.rh358.example.com.
                11              IN  PTR         serverd.rh358.example.com.


                ;;;;;Reverse information for 192.168.1
                $ORIGIN 1.168.192.in-addr.arpa.
    # 放行并启动服务，并安装dig
        [root@servera named]# yum -y install bind-utils
        [root@servera named]# systemctl enable --now named.service
        [root@servera named]# firewall-cmd --permanent --add-service=dns
        [root@servera named]# firewall-cmd --reload

# 验证方法：
    [root@servera ~]# dig serverc.rh358.example.com @192.168.1.100
        serverc.rh358.example.com. 86400 IN     A       192.168.1.10
    [root@servera ~]# dig -x 192.168.1.11 @192.168.1.100
        11.1.168.192.in-addr.arpa. 86400 IN     PTR     serverd.rh358.example.com.
```

第5题：Share a directory via SMB(通过 Samba 共享目录)

- Configure SMB services on servera as follows： （在 servera 上配置 SMB 服务，满足以下要求：）
  - The SMB server is a member of the STAFF workgroup（samba 服务器属于 STAFF 工作组)
  - The service shares the /groupdir directory.The share's name is common（共享/groupdir 目录，共享名为 common）
  - The common share is available to lab.example.com domain clients only（只允许来自 lab.example.com 网络域的用户访问 common 共享)

- The common share is browseable（common 共享可以查看其中内容）
- The user barney has read access to the share， authenticating with the password redhat if necessary（barney 用户可以使用 redhat 密码，以只读方式访问 common 共享 ）

```
# 知识点考察：
        提供SMB文件共享

# 解题：
        #安装软件包
                [root@servera ~]# yum -y install samba policycoreutils-python-utils samba-client
        # 放行samba服务
                [root@servera ~]# firewall-cmd --permanent --add-service=samba
                [root@servera ~]# firewall-cmd --reload
    # 启动服务
                [root@servera ~]# systemctl enable --now smb
    # 创建共享目录及设置selinux文件安全上下文
        [root@servera ~]# mkdir /groupdir
        [root@servera ~]# semanage fcontext -a -t samba_share_t "/groupdir(/.*)?"
        [root@servera ~]# restorecon -R -v /groupdir/
    # 创建指定用户
        [root@servera ~]# useradd -s /sbin/nologin barney
        [root@servera ~]# echo redhat | passwd barney --stdin
        [root@servera ~]# chgrp barney /groupdir/
        [root@servera ~]# chmod 755 /groupdir/
        [root@servera ~]# smbpasswd -a barney
            New SMB password: redhat
            Retype new SMB password: redhat
            Added user barney.
    # 修改配置文件
        [root@servera ~]# vim /etc/samba/smb.conf
            [global]
                workgroup = STAFF
            [common]
                    path = /groupdir
                    browseable = yes
                    hosts allow = 127. 172.25.250.
        [root@servera ~]# systemctl restart smb

# 验证方法：
        [root@serverb ~]# smbclient //servera/common -U barney
            Enter SAMBA\barney's password:
            Try "help" to get a list of possible commands.
            smb: \> ls
```

```
  .                                   D        0  Wed Aug  2 11:21:05 2023
  ..                                  D        0  Wed Aug  2 11:21:05 2023

                10474476 blocks of size 1024. 8187540 blocks available
smb: \>

# 查询帮助方法：
    # 安装软件包:    [root@node1 ~]# yum provides semanage
                        policycoreutils-python-utils
        # 创建共享目录及设置selinux文件安全上下文: man semanage fcontext 查询 example获得selinux帮助
```

第6题：Create a multiuser SMB mount（创建一个多用户 Samba 挂载）

- On servera share the /data directory via SMB as follows：(在 servera 上通过 SMB 共享/data目录，要求如下：)
  - The share is named data（共享名为 data）
  - The shared data is only available to clients in the lab.example.com domain（只允许来自 lab.example.com 网络域的用户访问 data 共享）
  - The shared data is browseable（可以查看 data 共享中的内容）
  - magneter has read access to the share，authenticating with the password redhat（允许 magneter 用户以只读方式访问共享，密码为 redhat）
  - wolferyne has read and write access to the share，authenticating with the password redhat（允许 wolferyne 用户以读写方式访问共享，密码为 redhat
  - The SMB share is permanently mounted on serverb at /mnt/multi using the credentials of magneter（serverb 使用 magnetar 用户的凭据可将共享永久挂载到/mnt/multi 目录）

- The share allows anyone who can authenticate as wolferyne to temporarily acquire write permissions（使用 wolferyne 用户的凭据可以获取这个目录的写入权限）

```
# 知识点考察：
    提供基于文件的网络存储，限制对 SMB 共享的访问
    挂载 SMB 共享
    执行多用户 SMB 挂载
# 解题：
        # 服务端：servera
                # 创建相关用户及配置相关权限
                    [root@servera ~]# useradd -s /sbin/nologin magneter
                    [root@servera ~]# useradd -s /sbin/nologin wolferyne
                    [root@servera ~]# echo redhat | passwd magneter --stdin
                    [root@servera ~]# echo redhat | passwd wolferyne --stdin
                    [root@servera ~]# smbpasswd -a magneter
                    [root@servera ~]# smbpasswd -a wolferyne
                # 创建共享目录及修改selinux文件安全上下文
                    [root@servera ~]# mkdir /data
                    [root@servera ~]# setfacl -m u:wolferyne:rwx /data/
                    [root@servera ~]# setfacl -m u:magneter:rx /data/
                    [root@servera ~]# semanage fcontexset -a -t samba_share_t
"/data(/.*)?"
                    [root@servera ~]# restorecon -R -v /data/
                # 修改配置文件及重启服务
                    [root@servera ~]# vim /etc/samba/smb.conf
                        [global]
                            workgroup = STAFF
                        [data]
                            path = /data
                            browseable = Yes
                            hosts allow = 172.25.250.
                            write list =  wolferyne
                    [root@servera ~]# systemctl restart smb.service
        #客户端：serverb
                # 安装软件
                    [root@serverb ~]# yum -y install cifs-utils.x86_64
                # 创建挂载点
                    [root@serverb ~]# mkdir /mnt/multi
                # 创建相关用户,在a上用id查看，确保UID一致
                    [root@serverb ~]# useradd -u 1004 wolferyne
                    [root@serverb ~]# useradd -u 1003 magneter
                # 编辑smb挂载密码文件及修改权限
                    [root@serverb ~]# vim /root/smbpass.txt
```

```
                        username=magneter
                        password=redhat
                    [root@serverb ~]# chmod 600 /root/smbpass.txt
                # 配置开机自动挂载
                    [root@serverb ~]# echo "//servera/data /mnt/multi cifs defaults,_netdev,credentials=/root/smbpass.txt,multiuser 0 0" >> /etc/fstab
                # 挂载测试
                    [root@serverb ~]# mount -a

# 验证方法：
    [root@serverb ~]# df -h
        Filesystem      Size  Used Avail Use% Mounted on
        //servera/data   10G  2.2G  7.8G  23% /mnt/multi
    [root@serverb ~]# touch /mnt/multi/2.txt
        touch: cannot touch '/mnt/multi/2.txt': Permission denied
    [root@serverb ~]# su - wolferyne
    [wolferyne@serverb ~]$ cifscreds add servera
        Password: redhat
    [wolferyne@serverb ~]$ touch /mnt/multi/1.txt

# 查询帮助方法：
        # 配置开机自动挂载: man mount.cifs 搜索cred与user
```

第7题：Configure NFS services(配置 NFS 服务)

- Configure NFS on servera to export the /public directory with read only access to the lab.example.com domain only
- （在 servera 配置 NFS 服务，共享/public 目录，对 lab.example.com 域有只读权限）

```
# 知识点考察：
        提供基于文件的网络存储
# 解题：
        # 创建共享目录
        [root@servera ~]# mkdir /public
    # 修改配置文件
        [root@servera ~]# vim /etc/exports
            /public *.lab.example.com(ro,sync)
    # 启动并放行服务
        [root@servera ~]# systemctl enable --now nfs-server.service
        [root@servera ~]# firewall-cmd --permanent --add-service=nfs
        [root@servera ~]# firewall-cmd --permanent --add-service=mountd
        [root@servera ~]# firewall-cmd --permanent --add-service=rpc-bind
        [root@servera ~]# firewall-cmd --reload


# 验证方法：
    [root@servera ~]# showmount -e localhost
        Export list for localhost:
        /public *.lab.example.com
    [root@servera ~]# exportfs -v
        /public
*.lab.example.com(sync,wdelay,hide,no_subtree_check,sec=sys,ro,secure,root_squash,no_all_squash)
```

第8题：Mount an NFS share(挂载 NFS 共享)

- Configure serverb to mount the following NFS share from servera.lab.example.com： （在 serverb 上挂载一个来自 servera.lab.example.com 的 NFS 共享，并符合下列要求：）
  - /public is mounted to /mnt/nfsmount（/public 挂载在下面的目录上/mnt/nfsmount)
  - The file system is automatically mounted at boot（文件系统在系统启动时自动挂载 ）

```
# 知识点考察：
        将对 NFS 导出的访问限制到特定的客户端和网络

# 解题：
    # 创建挂载点
        [root@serverb ~]# mkdir /mnt/nfsmount
    # 配置开机自动挂载
        [root@serverb ~]# echo "servera.lab.example.com:/public /mnt/nfsmount nfs defaults 0 0 ">> /etc/fstab
    # 挂载测试
        [root@serverb ~]# mount -a

# 验证方法：
    [root@serverb ~]# df -h
        Filesystem                      Size  Used Avail Use% Mounted on
        servera.lab.example.com:/public  10G  2.2G  7.8G  23% /mnt/nfsmount
```

第9题：Implement a web server(实现一个 web 服务器)

- Implement a web server on servera for the site **http://servera.lab.example.com** and perform the following steps：
- (在 servera 上配置一个站点 **http://servera.lab.example.com** 然后执行下述步骤：)
  - Download **http://classroom.example.com/files/station.html**（从**http://classroom.example.com/files/station.html**下载网页文件）
  - Rename the downloaded file to index.html.Do NOT make any modifications to the content of this file（将文件重命名为 index.html 不要修改此文件的内容）
  - Copy this index.html to the DocumentRoot of your webserver（将文件 index.html 拷贝到您的 web 服务器的 DocumentRoot 目录下）
  - Clients within lab.example.com can access the webserver（来自于 lab.example.com 域的客户端可以访问此 web 服务）
  - Clients within my133t.org are denied access to the webserver（来自于 my133t.org 域的客户端拒绝访问此 Web 服务 ）

```
# 知识点考察：
        搭建apache服务器提供网站服务

# 解题：
        # 安装软件
            [root@servera ~]# yum -y install httpd
        # 下载文件并复制到网站目录
            [root@servera ~]# wget http://classroom.example.com/files/station.html
            [root@servera ~]# cp station.html /var/www/html/index.html
        # 修改配置文件
            [root@servera ~]# vim /etc/httpd/conf.d/station.conf
                <VirtualHost _default_:80>
                    DocumentRoot "/var/www/html"
                    ServerName servera.lab.example.com
                </VirtualHost>
                <Directory "/var/www/html">
                    <RequireAll>
                        Require host lab.example.com            # 允许与拒绝，可以换成serverb和serverc的域名来体验效果
                        Require not host  my133t.org
                    </RequireAll>
                </Directory>
    # 启动并放行服务
        [root@servera ~]# systemctl enable --now httpd
        [root@servera ~]# firewall-cmd --permanent --add-service=http
        [root@servera ~]# firewall-cmd --reload

# 验证方法：
        [root@serverb ~]# curl servera.lab.example.com
                Practice 9-10
# 查询帮助方法：
        # 修改配置文件：从 /usr/share/doc/httpd/httpd-vhosts.conf 与 /etc/httpd/conf/httpd.conf 两个文件查找帮助
```

第10题：Configure secure webservices(配置安全 web 服务)

- Configure TLS encryption for the web server for **https://servera.lab.example.com**
- （配置基于TLS加密的web服务器 **https://servera.lab.example.com** ）

- A signed certificate for the web server can be found at **http://classroom.example.com/files/servera.crt**(一个已签名证书从 **http://classroom.example.com/files/servera.crt** 获取)
- The associated key for the certificate is at **http://classroom.example.com/files/servera.key**(此证书的密钥从 **http://classroom.example.com/files/servera.key**获取)
- The certificate for the signing authority is at **http://classroom.example.com/files/cacert.crt**(此证书的签名授权信息从 **http://classroom.example.com/files/cacert.crt** 获取)

```
# 知识点考察：
        配置安全 Web 服务器 HTTPS

# 解题：
        # 安装软件
                [root@servera ~]# yum -y install httpd-manual mod_ssl
        # 下载相关文件并复制到指定位置
            [root@servera ~]# wget http://classroom.example.com/files/servera.crt
            [root@servera ~]# wget http://classroom.example.com/files/servera.key
            [root@servera ~]# wget http://classroom.example.com/files/cacert.crt
            [root@servera ~]# cp *.crt /etc/pki/tls/certs/
            [root@servera ~]# cp *.key /etc/pki/tls/private/
            [root@servera ~]# chmod 600 /etc/pki/tls/private/*
            [root@servera ~]# chmod 600 /etc/pki/tls/certs/*
    # 修改配置文件
        [root@servera ~]# vim /etc/httpd/conf.d/station.conf
            <VirtualHost *:443>
                ServerName servera.lab.example.com
                SSLEngine on
                DocumentRoot "/var/www/html"
                SSLCertificateFile "/etc/pki/tls/certs/servera.crt"
                SSLCertificateKeyFile "/etc/pki/tls/private/servera.key"
                SSLCertificateChainFile "/etc/pki/tls/certs/cacert.crt"
            </VirtualHost>
    # 重启并放行服务
        [root@servera ~]# firewall-cmd --permanent --add-service=https
        [root@servera ~]# firewall-cmd --reload
        [root@servera ~]# systemctl restart httpd

# 验证方法：
        # 网站访问 https://servera.lab.example.com

# 查询帮助方法：
        # 修改配置文件：访问 http://servera.lab.example.com/manual/ssl/ 点击 mod ssl config 获取帮助
```

第11题：Configure a virtual host(配置一个虚拟主机)

- Extend your web server on servera to include a virtual host for the site **http://www-a.lab.example.com** as follows:
- （在 servera 上为站点 **http://www-a.lab.example.com** 创建一个虚拟主机，要求如下：）
  - The DocumentRoot for the virtual host is /var/www/virtual （设置 DocumentRoot 为 /var/www/virtual）
  - The DocumentRoot contains a copy of **http://classroom.example.com/files/www.html** named index.html. Do not make any modification to the content of this file （从 **http://classroom.example.com/files/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 DocumentRoot 目录下）
  - The user barney can create content in /var/www/virtual （确保 barney 用户能够在/var/www/virtual 目录下写入内容）
  - The original web site **http://servera.lab.example.com** is still accessable （原始站点 **http://servera.lab.example.com** 必须仍然能够访问）

```
# 知识点考察：
        配置基于名称的虚拟主机

# 准备工作：
    [root@servera conf.d]# echo 192.168.1.100 www-a.lab.example.com >> /etc/hosts
#解题
        # 创建网站目录并设置权限，配置网页文件
            [root@servera ~]# mkdir /var/www/virtual
            [root@servera ~]# setfacl -m d:u:barney:rwx /var/www/virtual/
            [root@servera ~]# setfacl -m u:barney:rwx /var/www/virtual/
            [root@servera ~]# wget http://classroom.example.com/files/www.html
            [root@servera ~]# cp www.html /var/www/virtual/index.html
            [root@servera conf.d]# semanage fcontext -a -t httpd_sys_content_t
"/var/www/virtual(/.*)?"
            [root@servera conf.d]# restorecon -Rv /var/www/virtual/
        # 修改配置文件
            [root@servera ~]# cd /etc/httpd/conf.d/
            [root@servera conf.d]# cp station.conf  virtual.conf
            [root@servera conf.d]# vim virtual.conf
                <VirtualHost *:80>
                    DocumentRoot "/var/www/virtual"
                    ServerName www-a.lab.example.com
                </VirtualHost>
                <Directory "/var/www/virtual">
                    AllowOverride None
                    Require all granted
                </Directory>
        # 重启服务
                [root@servera ~]# systemctl restart httpd

# 验证方法：
        [root@servera ~]# curl www-a.lab.example.com
                Practice 11

# 查询帮助方法：
        # 修改配置文件 从 /usr/share/doc/httpd/httpd-vhosts.conf 与
/etc/httpd/conf/httpd.conf 两个文件查找帮助
```

第12题：Configure web content access(配置 web 内容的访问)

- On your web server on servera create a directory named confidential under the DocumentRoot directory and configure it as follows:
- （在 servera 上的 web 服务器的 DocumentRoot 目录下创建 confidential 目录，要求如下：）
  - Copy **http://classroom.example.com/files/private.html** into this directory and rename it index.html. Do not change the content of this file
    - （从 **http://classroom.example.com/files/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）
  - The contents of confidential is visible to anyone browsing from servera (including localhost) but is not accessible from other locations
    - （这个网页的内容只有 servera 可以访问，其它地址不允许访问）

```
# 知识点考察：
        web网站的访问限制
# 解题
        # 创建web目录及网站内容
        [root@servera ~]# mkdir /var/www/html/confidential
        [root@servera ~]# wget http://classroom.example.com/files/private.html
        [root@servera ~]# cp private.html /var/www/html/confidential/index.html
    # 修改配置文件
        [root@servera ~]# vim /etc/httpd/conf.d/station.conf {新增以下内容，原内容不做修改}
            <Directory "/var/www/html/confidential">
                Require local
            </Directory>
    # 重启服务
        [root@servera ~]# systemctl restart httpd

# 测试
    [root@servera ~]# curl http://servera.lab.example.com/confidential/
        Practice 12
    [root@serverb ~]# curl http://servera.lab.example.com/confidential/
        <title>403 Forbidden</title>
```

第13题：Configure an iSCSI target(配置 iSCSI 服务端)

- Configure servera to provide an iscsl disk device named iqn.2021-09.com.example.lab:servera as follows： （ 配置 servera提供一个iSCSI服务， 磁盘名为iqn.2021-06.com.example.lab:servera，并符合下列要求：）
  - The iSCSI service use sport 3260 （iSCSI服务使用 3260 端口）
  - The target uses a 3GiB backing logical volume named iscsi_data （target 使用 3GiB 的后端逻辑卷存储，名为：iscsi_data）
  - The target is only available to serverb.lab.example.com （target 只供 serverb.lab.example.com 访问）
  - The service is available after a reboot（target 服务要求重启后仍然运行）

```
# 知识点考察：
        访问基于块的网络存储，提供和配置 iSCSI目标  配置 iSCSI 启动器，以持久连接 iSCSI 目标

# 解题：
        # 安装软件
                [root@servera ~]# yum -y install targetcli
        # 创建后端逻辑卷存储
            [root@servera ~]# pvcreate /dev/vdb
            [root@servera ~]# vgcreate  iscsi_vg /dev/vdb
            [root@servera ~]# lvcreate -L 3G -n iscsi_data iscsi_vg
        # 创建iscsi共享
            [root@servera ~]# targetcli
                /> backstores/block create iscsi_data /dev/iscsi_vg/iscsi_data
                /> iscsi/ create iqn.2023-08.com.example.lab:servera
                /> iscsi/iqn.2023-08.com.example.lab:servera/tpg1/acls create iqn.2023-08.com.example.lab:serverb
                /> iscsi/iqn.2023-08.com.example.lab:servera/tpg1/luns create /backstores/block/iscsi_data
                /> iscsi/iqn.2023-08.com.example.lab:servera/tpg1/portals/ delete 0.0.0.0 3260
                /> iscsi/iqn.2023-08.com.example.lab:servera/tpg1/portals/ create 172.25.250.10 3260
                /> exit
        # 启动并放行服务
            [root@servera ~]# systemctl enable --now  target.service
            [root@servera ~]# firewall-cmd --permanent --add-service=iscsi-target
            [root@servera ~]# firewall-cmd --reload

# 验证方法：
    [root@servera ~]# targetcli
        lstargetcli shell version 2.1.fb49
        Copyright 2011-2013 by Datera, Inc and others.
        For help on commands, type 'help'.

        /> ls
        o- / ..... [...]
          o- backstores ..... [...]
```

```
34          | o- block ..... [Storage Objects: 1]
35          | | o- iscsi_data ..... [/dev/iscsi_vg/iscsi_lv (3.0GiB) write-thru activated]
36          | |   o- alua ..... [ALUA Groups: 1]
37          | |     o- default_tg_pt_gp ..... [ALUA state: Active/optimized]
38          | o- fileio ..... [Storage Objects: 0]
39          | o- pscsi ..... [Storage Objects: 0]
40          | o- ramdisk ..... [Storage Objects: 0]
41          o- iscsi ..... [Targets: 1]
42          | o- iqn.2021-06.com.example.lab:servera ..... [TPGs: 1]
43          |   o- tpg1 ..... [no-gen-acls, no-auth]
44          |     o- acls ..... [ACLs: 1]
45          |     | o- iqn.2021-06.com.example.lab:serverb ..... [Mapped LUNs: 1]
46          |     |   o- mapped_lun0 ..... [lun0 block/iscsi_data (rw)]
47          |     o- luns ..... [LUNs: 1]
48          |     | o- lun0 ..... [block/iscsi_data (/dev/iscsi_vg/iscsi_lv) (default_tg_pt_gp)]
49          |     o- portals ..... [Portals: 1]
50          |       o- 172.25.250.10:3260 ..... [OK]
```

```
51          o- loopback
.....................................................................................
.................. [Targets: 0]
```

第14题：Configure an iSCSI initiator(配置 iSCSI 的客户端)

- Configure serverb so that it connects to the iqn.2023-08.com.example.lab:servera target on servera as follows:
- （ 配置 serverb 使其能连接在 servera 上提供的 iqn.2023-08.com.example.lab:servera 并符合以下要求：）
  - The iSCSI device is automatically available on system boot （ iSCSI 设备在系统启动的期间自动加载）
  - The iSCSI block device contains a 2100 MiB partition that is formatted as ext4（ 块设备 iSCSI 上包含一个大小为 2100MiB 的分区，并格式化为 ext4）
  - The partition is mounted to /mnt/data and is automatically mounted to this directory at system boot（ 此分区挂载在/mnt/data 上，同时在系统启动的期间自动挂载）

```
# 知识点考察：
        访问基于块的网络存储，将对 iSCSI 服务的访问限制到特定的客户端和访问

# 解题：
        # 安装软件{此软件包可tab，以iscsi开头的软件包仅有一个}
                [root@serverb ~]# yum -y install iscsi-initiator-utils
        # 修改配置文件
            [root@serverb ~]# cat /etc/iscsi/initiatorname.iscsi
                InitiatorName=iqn.2021-06.com.example.lab:serverb
        # 启动服务
            [root@serverb ~]# systemctl enable --now  iscsid.service
        # 加入iscsi共享到本地
            [root@serverb ~]# iscsiadm --mode discoverydb --type sendtargets --portal servera.lab.example.com --discover
                172.25.250.10:3260,1 iqn.2021-06.com.example.lab:servera
            [root@serverb ~]# iscsiadm --mode node --targetname iqn.2021-06.com.example.lab:servera  --portal 172.25.250.10:3260 --login
                Logging in to [iface: default, target: iqn.2021-06.com.example.lab:servera, portal: 172.25.250.10,3260]
                Login to [iface: default, target: iqn.2021-06.com.example.lab:servera, portal: 172.25.250.10,3260] successful.
            [root@serverb ~]# lsblk
                NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
                sda      8:0    0   3G  0 disk
        # 进行分区
            [root@serverb ~]# fdisk /dev/sda
            [root@serverb ~]# lsblk
                NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
                sda      8:0    0    3G  0 disk
                └─sda1   8:1    0  2.1G  0 part
        # 格式化并配置自动挂载
            [root@serverb ~]# mkfs.ext4 /dev/sda1
            [root@serverb ~]# mkdir /mnt/data
            [root@serverb ~]# blkid
                /dev/sda1: UUID="49715030-f326-4b74-949d-3c31e4789033" TYPE="ext4" PARTUUID="6ae89eb5-01"
            [root@serverb ~]# echo 'UUID="49715030-f326-4b74-949d-3c31e4789033" /mnt/data ext4 defaults,_netdev 0 0 ' >> /etc/fstab
            [root@serverb ~]# mount -a

# 验证方法：
```

```
        [root@serverb ~]# df -h
            Filesystem                      Size  Used Avail Use% Mounted on
            /dev/sda1                       2.0G  6.2M  1.9G   1% /mnt/data

# 查询帮助方法：
        # 加入iscsi共享到本地：查询帮助 man iscsiadm 搜索example
```

💡 **#注意：**

此题完成后，可重启serverb测试/etc/fstab文件是否有误

第15题：Configure a database（配置一个数据库）

Create a MariaDB database, named Contacts on servera such that the following conditions exist: （在 servera 上配置一个名字是 Contacts 数据库服务器，然后执行下述步骤：）

- The database contains the contents of the database dump from **http://classroom.example.com/files/users.mdb**（将 **http://classroom.example.com/files/users.mdb**文件下载，并恢复 Contacts 库)
- The database is accessible from localhost only （数据库只允许 localhost 访问）
- Other than the root user, this database only allows queries from the user student.This user has the password redhat （除数据库 root 用户外，此数据库只允许 student 用户查询访问，用户密码是 redhat）
- The root user has the password redhat and can not log in without using a password （数据库的 root 用户密码是 redhat，不允许免密登录数据库）

```
# 知识点考察：
        将 MariaDB 服务器访问限制到特定的网络地址
    创建 MariaDB 数据库
    管理 MariaDB 数据库用户和访问权限
    从备份导入 MariaDB 数据库

# 解题：
    # 安装组件并修改配置
        [root@servera ~]# yum -y install mariadb-server
        [root@servera ~]# cat /etc/my.cnf.d/mariadb-server.cnf  # 在 [mysqld]下新增一行配置
            [mysqld]
            ……
            bind-address=127.0.0.1
    # 放行并启动服务
        [root@servera ~]# firewall-cmd --permanent --add-service=mysql
        [root@servera ~]# firewall-cmd --reload
        [root@servera ~]# systemctl enable --now mariadb
    # 安全加固
        [root@servera ~]# mysql_secure_installation  # 进行安全加固，除了设置密码外均为 y 回车
    # 进入mysql
        [root@servera ~]# mysql -uroot -p
            Enter password: redhat
    # 创建用户
        MariaDB [(none)]> create user student@localhost identified by 'redhat';
    # 配置权限
        MariaDB [(none)]>  GRANT SELECT ON Contacts.* TO 'student'@'localhost';

    # 创建数据库
        MariaDB [(none)]> create database Contacts;
    # 导入数据库
        [root@servera ~]# wget  http://classroom.example.com/files/users.mdb
        [root@servera ~]# mysql -u root -predhat Contacts < users.mdb

# 验证方法：
        [root@servera ~]# mysql -uroot -predhat
            MariaDB [(none)]> use Contacts;
            MariaDB [Contacts]> show tables;
                +--------------------+
```

```
| Tables_in_Contacts |
+--------------------+
| User_Contacts      |
| User_Logins        |
| User_Names         |
+--------------------+

# 查询帮助方法：
        # 创建用户:  MariaDB [(none)]> show create user;
```

第16 & 17题：Query a database(数据查询填空)

- Use the database Contacts on servera and the appropriate SQL queries to provide the following information （完成以下要求的查询并将结果填入相应的框格中）：
  - Enter the first name of the person whose password is pass56 （在 Contacts 数据库中进行关联查询：密码为 pass56的用户的 first name 是？）
  - Number of people named Wilson and Zhao in Qingdao（在 Contacts 数据库中进行关联查询：在Qingdao居住的名字为"Zhao"和"Wilson"的有几人？）

```
# 知识点考察：
        mariadb数据库联合查询

# 解题：
 [root@servera ~]# mysql -u root -predhat
        MariaDB [(none)]> use Contacts;  {可通过select * from 表名 查询所有的架构}
        MariaDB [Contacts]> select * from User_Logins,User_Names
            -> where User_Names.user_id = User_Logins.id
            -> and User_Logins.user_Pass like 'pass56';
                +----+------------+-----------+---------+-----------+------------+
                | id | User_Login | User_Pass | user_id | last_name | first_name |
                +----+------------+-----------+---------+-----------+------------+
                | 56 | student56  | pass56    |      56 | Wuliu     | Sun        |
                +----+------------+-----------+---------+-----------+------------+

        MariaDB [Contacts]> select * from User_Names,User_Contacts
            -> where User_Contacts.id = User_Names.user_id
            -> and User_Contacts.Location like "Qingdao";            # 此题采用简单做法，只过滤出了Qingdao的人，需要自己数一下名字为 "Zhao"和"Wilson" 的人数为多少
                +---------+-----------+------------+----+----------+---------------+-------------+-------------+
                | user_id | last_name | first_name | id | Location | Email         | Telephone   | Fax         |
                +---------+-----------+------------+----+----------+---------------+-------------+-------------+
                |      83 | Basan     | Zhao       | 83 | Qingdao  | zhao2@abc.com | 13100000005 | 13200000005 |
                |      19 | Shijiu    | Zhao       | 19 | Qingdao  | zhao@abc.com  | 13100000001 | 13200000001 |
                +---------+-----------+------------+----+----------+---------------+-------------+-------------+
# 答：
         16题  密码为 pass56 的用户的 first name 是 Sun ，
         17题  在Qingdao居住的名字为"Zhao"和"Wilson"的有2人
```

进入Ansible题目

```
# 准备工作 {devops密码为redhat}
    [root@foundation0 ~]# ssh devops@workstation
    [devops@workstation ~]$ ssh-keygen
    [devops@workstation ~]$ ssh-copy-id serverc
    [devops@workstation ~]$ ssh-copy-id serverd
    [devops@workstation ~]$ mkdir ansible
    [devops@workstation ~]$ cd ansible/
    [devops@workstation ansible]$ sudo yum -y install ansible rhel-system-roles
    [devops@workstation ansible]$ echo -e 'serverc\nserverd' >> inventory
    [devops@workstation ansible]$ vim ansible.cfg
        [defaults]
            inventory=/home/devops/ansible/inventory
        [privilege_escalation]
            become=True
            become_method=sudo
            become_user=root
            become_ask_pass=False
    [devops@workstation ansible]$ ansible all -m ping
```

第18题：Configure managed host firewall access(配置受管主机的防火墙)

Create a playbook called /home/student/ansible/firewall.yml on control that creates a firewall on serverc and serverd as follows： （在 ansible 控制节点的/home/devops/ansible目录中创建firewall.yml在 serverc 和 serverd 上分别设置： ）

- The playbook runs on serverc.lab.example.com and serverd.lab.example.com （Playbook 运行在 serverc.lab.example.com 和 serverd.lab.example.com ）
- Clients within 172.25.250.0/24 have remote SSH access to serverc and serverd （来自 172.25.250.0/24 的客户端可以通过 ssh 连接访问 serverc 和 serverd ）
- Clients within 172.24.250.0/24 do NOT have remote SSH access to either serverc or serverd （来自 172.24.250.0/24 的客户端不允许通过 ssh 访问 serverc 和 serverd ）

```
# 知识点考察：
        通过 ansible 配置系统防火墙，以允许访问特定的服务或端口
        通过 ansible 配置系统防火墙，以仅允许或拒绝来自特定网域或 IP 子网的访问

# 解题：
        # 编辑剧本文件
            [devops@workstation ansible]$ vim firewall.yml
                - name: configure firewalld rich rule
                  hosts: serverc,serverd
                  tasks:
                          - name: accept
                            firewalld:
                                    zone: public
                                    rich_rule: rule family="ipv4" source address="172.25.250.0/24" service name="ssh" accept
                                    permanent: yes
                                    immediate: yes
                                    state: enabled

                          - name: reject
                            firewalld:
                                    zone: public
                                    rich_rule: rule family="ipv4" source address="172.24.250.0/24" service name="ssh" reject
                                    permanent: yes
                                    immediate: yes
                                    state: enabled
                          - name: ssh
                            firewalld:
                                    zone: public
                                    service: ssh
                                    permanent: yes
                                    state: disabled
                                    immediate: yes
        # 执行剧本
            [devops@workstation ansible]$ ansible-playbook firewall.yml

# 验证方法：
            [devops@workstation ansible]$ ansible serverc -m shell -a "firewall-cmd --list-all"
```

```
                serverc | CHANGED | rc=0 >>
                ……
                  rich rules:
                        rule family="ipv4" source address="172.25.250.0/24" service name="ssh" accept
                        rule family="ipv4" source address="172.24.250.0/24" service name="ssh" reject
# 查询帮助方法：
        # 编辑剧本文件： ansible-doc firewalld 搜索EXAMPLE，rich_rule规则可复制servera历史记录中的规则配置
```

第19题：Configure NGINX services(配置 NGINX 服务)

Create a playbook called /home/devops/ansible/nginx.yml on control that creates an NGINX server on serverc and serverd as follows:（在 ansible 控制节点的/home/devops/ansible 目录中创建 nginx.yml，在 serverc和 serverd 运行 nginx server，要求如下：）

- The playbook runs on serverc and serverd
- The NGINX servers use **http://classroom.example.com/files/webserver.conf.j2** as their configuration file （ Nginx 服务器使用 **http://classroom.example.com/files/webserver.conf.j2** 作为配置文件 ）
- The DocumentRoot directory for the NGINX servers is /srv/www/html （ Nginx 的网站家目录是/srv/www/html ）
- The DocumentRoot directory contains an index.html file that is generated from **http://classroom.example.com/files/index.html.j2**（ 网站的 index.html 来自于 **http://classroom.example.com/files/index.html.j2**）
- Do NOT change the content of index.html （ 不要修改其中内容 ）
- The appropriate SELinux contexts are set and firewall ports are open for access from lab.example.com （ 在防火墙上打开相应端口和设置相应的 SELinux contexts ）
- After the playbook has been run it is possible to browse the following sites: （ 可以通过访问 **http://serverc.lab.example.com** 和 **http://serverd.lab.example.com** 访问网页 ）
  - **http://serverc.lab.example.com**
  - **http://serverd.lab.example.com**

```
# 知识点考察：
        通过 ansible 安装和配置 nginx
# 解题：
        # 下载j2模板文件
                [devops@workstation ansible]$ wget http://classroom.example.com/files/webserver.conf.j2
                [devops@workstation ansible]$ wget http://classroom.example.com/files/index.html.j2
        # 编辑剧本文件
                [devops@workstation ansible]$ vim nginx.yml
            ---
            - name: config nginx server
              hosts: serverc,serverd
              tasks:
                      - name: install nginx
                        yum:
                                name: '@nginx:1.16'
                                state: present
                      - name: create web dir
                        file:
                                path: /srv/www/html
                                owner: root
                                group: root
                                state: directory
                                mode: '0755'
                                setype: httpd_sys_content_t
                      - name: edit web-page
                        template:
                                src: index.html.j2
                                dest: /srv/www/html/index.html
                                setype: httpd_sys_content_t
                      - name: copy nginx_conf
                        template:
                                src: webserver.conf.j2
                                dest: /etc/nginx/conf.d/nginx.conf
                      - name: firewalld
                        firewalld:
                                    service: http
                                    permanent: yes
                                    state: enabled
```

```
                                    immediate: yes
                    - name: start service
                      systemd:
                              name: nginx
                              state: started
                              enabled: yes
        # 执行测试
                [devops@workstation ansible]$ ansible-playbook nginx.yml

# 验证方法：
    [devops@workstation ansible]$ curl http://serverc.lab.example.com
        <html>
        <head>
        <title>Welcome to RH358 course</title>
        </head>
        <body>
        <h3>Success! Looking at serverc.lab.example.com</h3>
        Practice 19
        </body>
        </html>
    [devops@workstation ansible]$ curl http://serverd.lab.example.com
        <html>
        <head>
        <title>Welcome to RH358 course</title>
        </head>
        <body>
        <h3>Success! Looking at serverd.lab.example.com</h3>
        Practice 19
        </body>
        </html>

# 查询帮助方法：
        # 编辑剧本文件.yum模块软件包名： 在serverc中使用 yum module list | grep nginx 查询最新版}
```

第20题：Configure local mail services（配置本地的邮件服务）

- Create a playbook called /home/devops/ansible/nullclient.yml on control that configures mail on both serverc and serverd according to the following requirements:
- （在 ansible 控制节点的/home/devops/ansible 目录中创建 nullclient.yml，要求如下：）
  - The playbook runs on serverc and serverd （Playbook 管理 serverc a）
- You may test your configuation by sending email to the local user ‘student'. The system bastion.lab.example.com has been configured to drop mail for this user into bastion.lab.example.com/mail/student （您可以通过发送邮件到本地用户 student 来测试您的配置，系统 bastion.lab.example.com 已经配置，可以通过如下网址测试 http://bastion.lab.example.com/received_mail/1）

```
# 知识点考察：
    配置电子邮件传输，配置电子邮件服务器，以将电子邮件转发到出站邮件中继
    使用邮件客户端读取或发送电子邮件
    通过 ansible 使用 RHEL 系统角色
# 准备工作
    [student@workstation ~]$ lab smtp-automation start
# 解题：
        # 编写剧本文件
            [devops@workstation ansible]$ vim nullclient.yml
                - name: Configure Null Client Email Service
                  hosts: serverc,serverd
                  vars:
                    postfix_conf:
                      relayhost: "smtp.lab.example.com"
                      inet_interfaces: "loopback-only"
                      mynetworks: "127.0.0.0/8 [::1]/128"
                      myorigin: "lab.example.com"
                      mydestination: ""
                  roles:
                    - linux-system-roles.postfix
        # 执行
                [devops@workstation ansible]$ ansible-playbook nullclient.yml

# 测试
    # 验证方法：
        [root@serverc ~]# yum -y install mailx
        [root@serverc ~]# echo "hello,this is serverc" | mail -s 'demo' student
    # 登录bastion查看
        [root@bastion ~]# cat /var/www/html/mail/student
            ……
        From: Student User <student@lab.example.com>
        hello,this is serverc

    # 登录网页查看准备工作
        [root@bastion ~]# setenforce 0
        [root@bastion ~]# ln -s /var/spool/mail /var/www/html/maail
        [root@bastion ~]# chmod a+r /var/www/html/mail/student
            而后访问下载student即可查看
```

```
# 查询帮助方法：
        # 编写剧本文件.查看角色写法：   vim /usr/share/ansible/roles/rhel-system-roles.postfix/README.md
        # 编写剧本文件.变量vars键 ：   vim /usr/share/doc/postfix/README_FILES/STANDARD_CONFIGURATION_README  ## 需要先安装postfix才可以查看
```

下载此文件可查看

第21题：Configure printing services(配置打印服务)

- Create a playbook called /home/devops/ansible/printing.yml on control that configures printing on both serverc and serverd according to the following requirements: （在 ansible 控制节点的/home/devops/ansible 目录中创建 printing.yml，要求如下：）
  - The playbook runs on serverc and serverd （Playbook 管理 serverc 的打印队列）
  - The systems create a default print queue named my-printer （创建名为 my-printer 的默认打印队列）
  - The print queue forwards print jobs to an IPP printer using the URI of ipp://serverc.lab.example.com:631/printers/rht-printer (打印队列将打印作业发送到IPP打印机：ipp://serverc.lab.example.com:631/printers/rht-printer）

- You may test your configuration by sending a print job to the default print queue. The above remote IPP server has been configured to drop your print jobs into http://serverd/ippserver/ipp-everywhere-pdf （配置完成可以在 http://serverd/ippserver/ipp-everywhere-pdf检查打印作业）

```
# 知识点考察：a
    通过ansible管理打印机和打印文件，创建和管理网络打印机的打印机队列
    通过ansible管理现有的打印机队列

# 准备工作：
        # 该题目与19题冲突，先停止serverc及serverd的nginx
            [devops@workstation ansible]$ ansible all -m shell -a 'systemctl stop nginx'
        # 等待系统启动后
            [student@workstation ~]$ lab printing-config start
        # 此题servera 和 serverb 来替代 c，d两台机器，满足题目需求
            [devops@workstation ansible]$ echo -e 'servera\nserverb' >> inventory

# 解题：
        # 编辑剧本文件
            [devops@workstation ansible]$ vim  printing.yml
                - hosts: servera,serverb
                  tasks:
                    - name: install
                      yum:
                        name: cups,avahi
                        state: present
                    - name: start services
                      service:
                        name: "{{ item }}"
                        state: started
                        enabled: yes
                      loop:
                        - avahi-daemon
                        - cups
                    - name: open firewalld
                      firewalld:
                        service: mdns
                        state: enabled
                        permanent: yes
                        immediate: yes
                    - name: Create print queue
                      command: lpadmin -p my-printer -v ipp://serverc.lab.example.com:631/printers/rht-printer -m everywhere -E
                    - name: make default printer
```

```
                    command: lpadmin -d my-printer    # 注意command两处的的queue named（my-printer），考试中会修改名称，注意同步修改
        # 执行测试
            [devops@workstation ansible]$ ansible-playbook printing.yml

# 验证方法：
    [devops@workstation ansible]$ ansible servera -m shell -a 'lpstat -d;lpstat -v'
        servera | CHANGED | rc=0 >>
        system default destination: my-printer
        device for my-printer: ipp://serverc.lab.example.com:631/printers/rht-printer
    [devops@workstation ansible]$ ansible servera -m shell -a 'lp /etc/passwd'
        servera | CHANGED | rc=0 >>
        request id is my-printer-2 (1 file(s))
    # 而后登录 http://serverd/ippserver/ipp-everywhere-pdf/ 网址查看

# 查询帮助方法：
        # 编辑剧本文件.create print queue.command:
                                [root@serverc ~]# yum -y install cups*
                                [root@serverc ~]# man lpadmin 搜索example拿到案例进行修改
```