重要配置信息

在练习期间，除了您就坐位置的台式机之外，还将使用多个虚拟系统。您不具有台式机系统的根访问权，但具有对虚拟系统的完全 root 访问权。

系统信息

系统	IP地址	角色	LAN1	LAN2	练习环境
workstation	172.25.250.9	ansible 控制节点	Y		
bastion	172.25.250.254	网关	Y		
servera	172.25.250.10	CMD - server	Y	Y	
serverb	172.25.250.11	CMD - client	Y		
serverc	172.25.250.12	ansbile 托管节点	Y		
serverd	172.25.250.13	ansible 托管节点	Y		
servere		dhcp client		Y	N

这些系统的IP地址采用静态设置。请勿更改这些设置。主机名称解析已配置为解析上方列出的完全限定主机名，同时也解析主机短名称。

帐户信息

- 所有系统的root密码是**redhat**。请勿更改root密码
- 除非另有指定，否则这将是用于访问其他系统和服务的密码。
- 除非另有指定，否则此密码也应用于您创建的所有帐户或者任何需要设置密码的服务。
- 为方便起见，所有系统上已预装了SSH密钥，允许在不输入密码的前提下通过SSH进行root访问。请勿对系统上的root SSH配置文件进行任何修改。
- **Ansible控制节点上已创建了用户帐户**devops** 。**此帐户预装了SSH密钥，允许在Ansible控制节点和各个Ansible受管节点之间进行SSH登录。请勿对系统上的student SSH配置文件进行任何修改。您可以从root帐户使用su访问此用户帐户。

其他信息

- 一些练习项目可能需要修改Ansible主机清单。您要负责确保所有以前的清单组和项目保留下来，与任何其他更改共存。您还要有确保清单中所有默认的组和主机保留您进行的任何更改。
- 所有节点，yum存储库已正确配置。
- 一些项目需要额外的文件，这些文件已在以下位置提供：http://materials.example.com/classroom/ansible/
- 产品文档可从以下位置找到：https://docs.ansible.com/ansible/2.9/
- 其他资源也进行了配置，供您在练习期间使用。关于这些资源的具体信息将在需要这些资源的项目中提供。

虚拟系统管理

练习期间，您可以随时关闭或重新引导虚拟机系统。您可以从虚拟系统本身进行这项操作，也可以从物理系统控制虚拟系统。要从物理系统访问或控制练习系统，单击桌面上VM控制台图标。这会显示一个表格，包含每个虚拟机系统的对应按钮，单击特定虚拟机系统的按钮将弹出一个菜单，包含用来控制该系统选项：

- 启动节点点VM-如果指定的虚拟系统未在运行，该选项将启动指定系统。如果系统已经在运行-则该选项无任何作用。

- 重新引导节点VM-正常关闭练习虚拟系统，然后重启。
- 关闭节点VM-正常关闭指定虚拟系统。
- 关闭节点VM电源-立即关闭指定虚拟系统。
- VM控制台节点-这将打开一个窗口，用于连接到指定虚拟系统的控制台。请注意，如果将焦点移动到此窗口，控制台将抓住您的鼠标。要恢复鼠标，同时键入Ctrl+Alt。
- 重建节点VM-将当前VM还原为原始状态。系统将弹出一个单独的窗口，要求您确认操作。警告！！！您在VM上完成的所有操作都将丢失。仅当系统无法使用时才应使用这个功能。在使用这个功能之前，确保关闭VM。

重要评测信息

请注意，在评分之前，您的 Ansible 托管节点系统 将重置为练习开始时的初始状态，您编写的 Ansible playbook 将通过以 **student** 用户身份在控制节点上运行来加以应用。在 playbook 运行后，系统会重新启动您的托管节点，然后进行评估，以判断它们是否按照规定进行了配置。

请注意，在评分之前，您的 Ansible 托管节点系统 将重置为练习开始时的初始状态，您创建的 Ansible Tower 作业将通过以指定的用户身份运行来加以应用。在作业运行后，系统会重新启动 Ansible Tower 托管节点，然后进行评估，以判断它们是否按照规定进行了配置。

练习要求

1. 配置 IPV6 地址-C1

> 在您的练习系统上配置接口**eth0**使用下列IPv6地址：
>
> - [ ] servera 上的地址应该是**fddb:fe2a:ab1e::c0a8:64/64**
> - [ ] serverb 上的地址应该是**fddb:fe2a:ab1e::c0a8:65/64**
> - [ ] 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

Important - 重要

练习时，只需servera和serverb互相能 ping6 通即可

[root@servera]

```
查看连接配置文件名称
*# nmcli connection show | grep eth0
`Wired connection 1`  4ae4bb9e-8f2d-3774-95f8-868d74edcc3c  ethernet  eth0

修改网卡配置文件，永久生效
*# nmcli con mod "Wired connection 1" \
    ipv6.method manual \
    ipv6.addresses fddb:fe2a:ab1e::c0a8:64/64

立即生效
*# nmcli con up "Wired connection 1"
```

```
测试
*# ping -c4 172.25.250.254
64 bytes from 172.25.250.254: icmp_seq=1 ttl=64 time=1.16 ms
...
```

[root@serverb]

```
*# nmcli connection show | grep eth0
`Wired connection 1`  81e1324d-6dda-3ab6-b93f-dddb09b7f8ba  ethernet  eth0

*# nmcli con mod "Wired connection 1" \
    ipv6.method manual \
    ipv6.addresses fddb:fe2a:ab1e::c0a8:65/64
*# nmcli con up "Wired connection 1"
```

```
测试连接性-ipv6
*# ping6 -c4 fddb:fe2a:ab1e::c0a8:fe

测试连接性-ipv4
*# ping -c4 172.25.250.254
```

2. 配置 dhcp 服务器-C4

> Configure a DHCP server for IPv4 address assignment and provide fixed IP addresses to selected systems.
>
> - [ ] you deploy a DHCP server on **servera**
> - [ ] the second interface **eth1**
> - [ ] The DHCP server manages the **192.168.0.0/24** subnet
> - [ ] delivers IP addresses in the **192.168.0.200** to **192.168.0.254** range
> - [ ] gateway **192.168.0.1**
> - [ ] domain-name-servers **172.25.254.254**

- [ ] domain-search **example.net**
- [ ] default-lease-time **800**
- [ ] associates the **192.168.0.100** IP address to the **52:54:00:01:fa:0b** MAC address
- [ ] associates the **192.168.0.101** IP address to the **52:54:00:01:fa:0c** MAC address

Hint - 提示

- 考试环境中存在这台机器
- 练习环境中没有，可选择下面两种方式中的一种测试
1. 使用serverb的eth1网卡和serverc的eth1网卡

[root@servera]

```
# yum search dhcp
# yum -y install dhcp-server

# rpm -qc dhcp-server

-eth1=/etc/sysconfig/dhcpd
# cat /etc/sysconfig/dhcpd
# cp /usr/lib/systemd/system/dhcpd.service /etc/systemd/system/
# vim /etc/systemd/system/dhcpd.service
```

```
...
# ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -group dhcpd --no-pid
...
ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -group dhcpd --no-pid
eth1
```

```
# systemctl --system daemon-reload
重启不成功。因为/etc/dhcp/dhcpd.conf配置文件为空
# systemctl restart dhcpd.service

-/etc/dhcp/dhcpd.conf
# cat /etc/dhcp/dhcpd.conf
*# \cp /usr/share/doc/dhcp-server/dhcpd.conf.example /etc/dhcp/dhcpd.conf
# vim /etc/dhcp/dhcpd.conf
```

```
log-facility local7;

subnet 192.168.0.0 netmask 255.255.255.0 {
  range 192.168.0.200 192.168.0.254;
  option domain-name-servers 172.25.254.254;
  option domain-name "example.net";
  option routers 192.168.0.1;
  option broadcast-address 192.168.0.255;
```

```
  default-lease-time 800;
  max-lease-time 7200;
}

host passacaglia {
  hardware ethernet 52:54:00:01:fa:0b;
  fixed-address 192.168.0.100;
}

host fantasia {
  hardware ethernet 52:54:00:01:fa:0c;
  fixed-address 192.168.0.101;
}
```

```
# rpm -ql dhcp-server | grep service
# systemctl enable --now dhcpd

# systemctl status dhcpd
# grep -w dhcp /etc/services
# ss -anup | egrep '67|68'

# firewall-cmd --permanent --add-service=dhcp
# firewall-cmd --reload
```

[root@serverb]和[root@serverc]

```
判断是否存在配置文件
# nmcli con show
NAME                UUID                                  TYPE      DEV>
Wired connection 1  81e1324d-6dda-3ab6-b93f-dddb09b7f8ba  ethernet  eth>
Wired connection 3  f0f7c312-a06a-3850-bf2d-bb51ca746734  ethernet  eth>
ethernet-eth1       cd8550de-afa1-4acc-a0d1-4bf36e3d2b62  ethernet  -- >
Wired connection 2  706e063c-ee02-349e-8311-2c9837611ec4  ethernet  -- >

不存在，添加
# nmcli connection add type ethernet autoconnect yes ifname eth1
Connection 'ethernet-eth1' (cd8550de-afa1-4acc-a0d1-4bf36e3d2b62) successfully added.

# nmcli con up ethernet-eth1
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/50)

# ip a s eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
qlen 1000
    link/ether 52:54:00:01:fa:0b brd ff:ff:ff:ff:ff:ff
    inet **192.168.0.100/24** brd **192.168.0.255** scope global dynamic noprefixroute
eth1
       valid_lft 797sec preferred_lft 797sec
    inet6 fe80::cad5:44a8:4e02:bc41/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

```
# ip route
default via 172.25.250.254 dev eth0 proto static metric 106
default via 192.168.0.1 dev eth2 proto dhcp metric 107
default via **192.168.0.1** dev eth1 proto dhcp metric 108
172.25.250.0/24 dev eth0 proto kernel scope link src 172.25.250.11 metric 106
192.168.0.0/24 dev eth2 proto kernel scope link src 192.168.0.200 metric 107
192.168.0.0/24 dev eth1 proto kernel scope link src 192.168.0.100 metric 108

# cat /etc/resolv.conf
# Generated by NetworkManager
search lab.example.com example.com **example.net**
nameserver 172.25.250.254
nameserver **172.25.254.254**
```

3. 配置防火墙

> 在 **servera** 和 **serverb** 上分别设置，针对SSH
>
> - [ ] 允许**172.25.250.0/24** 的域对 servera 和 servera 进行**SSH**
> - [ ] 禁止**172.24.250.0/24** 的域对 servera 和 serverb 进行**SSH**

[root@servera]

```
# man -k fire
...
firewalld.richlanguage (5) - Rich Language Documentation
# man firewalld.richlanguage | grep rule.*service
...
           rule family="ipv4" source address="192.168.0.0/24" service name="tftp" log
prefix="tftp" level="info" limit value="1/m" accept
           rule family="ipv6" source address="1:2:3:4:6::" service name="radius" log
prefix="dns" level="info" limit value="3/m" reject

# firewall-cmd --list-all
*# firewall-cmd --permanent \
    --add-rich-rule='rule family="ipv4" source address="172.25.250.0/24" service
name="ssh" accept'
*# firewall-cmd --permanent \
    --add-rich-rule='rule family="ipv4" source address="172.24.250.0/24" service
name="ssh" reject'
*# firewall-cmd --permanent --remove-service=ssh

*# firewall-cmd --reload

# firewall-cmd --list-all
public (active)
...
  services: cockpit dhcpv6-client iscsi-target mountd nfs rpc-bind samba
```

```
...
  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
```

[root@serverb]

```
# firewall-cmd --list-all

*# firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source
address="172.25.250.0/24" service name="ssh" accept'
*# firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source
address="172.24.250.0/24" service name="ssh" reject'
*# firewall-cmd --permanent --remove-service=ssh

*# firewall-cmd --reload

# firewall-cmd --list-all
```

4. 完成主 DNS 配置-C3

> 当前我们已经配置好了DNS服务，要求
>
> - [ ] 在**servera**配置主 DNS
> - [ ] 配置正向解析**serverx**,**servery** 地址分别为**172.25.250.100**, **172.25.250.200**
> - [ ] 配置反向解析**serverx**,**servery**

Hint - 提示

- 考试时，这个题是排错题
- 默认该服务已安装
- 服务默认无法启动
- 相关文件全存在，就是权限不对
- 区域文件中，出现域名，记得点.结尾

[root@servera]

```
# yum search dns
# rpm -qc bind

*# vim /etc/named.conf
```

```
options {
```

```
      //listen-on port 53 { 127.0.0.1; };
        listen-on port 53 { any; };
      //listen-on-v6 port 53 { ::1; };
      //# 确认区域文件所在位置
        directory       "/var/named";
...
      //allow-query     { localhost; };
        allow-query     { any; };
...
      // 递归
        recursion yes;
...输出省略...
zone "lab.example.com" IN {
        type master;
//# 确认正向区域文件名称
        file "example.com.localhost";
        allow-update { none; };
};

zone "25.172.in-addr.arpa" IN {
        type master;
//# 确认反向区域文件名称
        file "25.172.loopback";
        allow-update { none; };
};
...输出省略...
```

```
# ls -l /var/named
*# chown :named /var/named/*

*# vim /var/named/example.com.localhost
```

```
$TTL 1D
@       IN SOA  servera.lab.example.com. rname.invalid. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
        NS      servera.lab.example.com.
servera         A       172.25.250.10
serverx         A       172.25.250.100
servery         A       172.25.250.200
```

```
*# vim /var/named/25.172.loopback
```

```
$TTL 1D
@       IN SOA  servera.lab.example.com. rname.invalid. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
        NS      servera.lab.example.com.
10.250        PTR     servera.lab.example.com.
100.250       PTR     serverx.lab.example.com.
200.250       PTR     servery.lab.example.com.
```

```
# yum search dns
# rpm -ql bind | grep service
*# systemctl enable --now named
# systemctl status named
...
   Loaded: loaded (/usr/lib/systemd/system/named.service; `enabled`...
   Active: `active` (running)

*# firewall-cmd --permanent --add-service=dns
*# firewall-cmd --reload
```

```
# yum provides host
# yum -y install bind-utils

# host serverx.lab.example.com 172.25.250.10
Using domain server:
Name: `172.25.250.10`
Address: 172.25.250.10#53
Aliases:
`serverx.lab.example.com has address 172.25.250.100`

# host 172.25.250.100 172.25.250.10
Using domain server:
Name: 172.25.250.10
Address: 172.25.250.10#53
Aliases:
`100.250.25.172.in-addr.arpa` domain name pointer `serverx.lab.example.com.`
```

```
# nslookup serverx.lab.example.com 172.25.250.10
Server:   172.25.250.10
Address:  172.25.250.10#53

Name: serverx.lab.example.com
Address: 172.25.250.100

# nslookup 172.25.250.100 172.25.250.10
100.250.25.172.in-addr.arpa name = serverx.lab.example.com.
```

```
# dig serverx.lab.example.com @172.25.250.10
...
;; flags: qr aa rd ra; QUERY: 1, ANSWER: `1`, AUTHORITY: 1, ADDITIONAL: 1
...
;; ANSWER SECTION:
`serverx.lab.example.com. 86400 IN  A 172.25.250.100`
...

# dig -x 172.25.250.100 @172.25.250.10
...
;; flags: qr aa rd ra; QUERY: 1, ANSWER: **1**, AUTHORITY: 1, ADDITIONAL: 2
..
;; ANSWER SECTION:
100.250.25.172.in-addr.arpa. 86400 IN PTR serverx.lab.example.com.
...
```

[root@serverb]

```
*# host servery.lab.example.com 172.25.250.10
Using domain server:
Name: `172.25.250.10`
Address: 172.25.250.10#53
Aliases:
`servery.lab.example.com has address 172.25.250.200

*# host 172.25.250.200 172.25.250.10
Using domain server:
Name: 172.25.250.10
Address: 172.25.250.10#53
Aliases:
`200.250.25.172.in-addr.arpa` domain name pointer `servery.lab.example.com.`
```

5. 准备 samba 共享-C10

> 在**servera**上准备 samba 共享
>
> - [ ] 您的SMB服务器必须是`STAFF`工作组的一个成员
> - [ ] 共享`/common`目录共享名必须为`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
>   - [ ] 要求`rob` 用户以只读的方式访问该目录，如果需要的话，验证的密码是`compede`
>   - [ ] `brian` 可以用读写的方式来访问该目录，brian密码为`postroll`
> - [ ] 共享`/releases`目录共享名必须为`releases`
>   - [ ] 挂载时用`opie`用户挂载,`opie` 用户以只读的方式访问该可浏览的目录
>   - [ ] 验证的密码是`haha`

[root@servera]

```
# yum search samba
# yum list samba samba-common samba-client
*# yum -y install samba samba-client

保证本地用户安全
*# id brian; id rob; id opie
# egrep 'brian|rob' /etc/passwd
*# usermod -s /sbin/nologin brian
*# usermod -s /sbin/nologin rob
*# useradd -s /sbin/nologin opie

添加 samba 用户
# pdbedit -L

*# smbpasswd -a brian
New SMB password: `postroll`
Retype new SMB password: `postroll`
Added user brian.
*# echo -e "compede\ncompede" | smbpasswd -a rob
New SMB password:
Retype new SMB password:
Added user rob.
*# (echo haha; echo haha) | smbpasswd -a opie
New SMB password:
Retype new SMB password:
Added user opie.

# pdbedit -L
brian
rob
opie

*# mkdir /common /releases
*# chown :brian /common
*# chmod 2775 /common

查上下文关系类型
m1# vim /etc/samba/smb.conf.example
m2# yum provides semanage
m2# yum -y install policycoreutils-python-utils
查上下文关系命令
# man semanage fcontext | grep \#

*# semanage fcontext -a -t samba_share_t "/common(/.*)?"
*# semanage fcontext -a -t samba_share_t "/releases(/.*)?"
*# restorecon -Rv /common /releases
Relabeled /common from unconfined_u:object_r:default_t:s0 to
unconfined_u:object_r:`samba_share_t`:s0
Relabeled /releases from unconfined_u:object_r:default_t:s0 to
unconfined_u:object_r:`samba_share_t`:s0
```

```
*# rpm -qc samba

*# vim /etc/samba/smb.conf
```

```
[global]
# workgroup = MYGROUP
  workgroup = STAFF
  ...
# 增加 1 行
  hostname lookups = yes
# 增加 7 行
[common]
  path = /common
# hosts allow = 127. *.example.com
  hosts allow = 127. .example.com
  browseable = Yes
  write list = @brian
[releases]
  path = /releases
```

```
*# systemctl enable --now smb nmb

*# firewall-cmd --permanent --add-service=samba
*# firewall-cmd --reload
```

```
*# smbclient -L //servera -N
Anonymous login successful
  Sharename       Type      Comment
  ---------       ----      -------
  `common`         Disk
  `releases`       Disk
  ...

*# smbclient -L //servera -U rob%compede
*# smbclient //servera/common \
    -U rob%compede \
    -c ls
*# smbclient //servera/common \
    -U brian%postroll \
    -c "put anaconda-ks.cfg"
putting file anaconda-ks.cfg as \anaconda-ks.cfg (1701.9 kb/s) (average 1701.9 kb/s)
```

6. 使用多用户访问 samba 共享目录-C10

> 在**serverb**上，要求通过smb 多用户的方式
>
> - [ ] 将共享目录`common`挂载到`/mnt/private`上,要求在对该共享目录挂载时以`rob`的身份进行操作

> - [ ] 将共享目录`releases`挂载到`/mnt/private2`上要求在对该共享目录挂载时，以`opie`的身份进行操作
> - [ ] 要求每次开机该共享目录可以自动挂载

[root@serverb]

```
# yum search cifs
# yum search samba
*# yum -y install cifs-utils samba-client

# smbclient -L //servera -N

*# mkdir /mnt/private /mnt/private2

# man mount.cifs
*# vim /root/cred.rob
```

```
username=rob
password=compede
```

```
*# vim /root/cred.opie
```

```
username=opie
password=haha
```

```
*# chmod 600 /root/cred.*
# ll /root/cred.*
-rw-------. 1 root root 30 Aug  7 13:31 /root/cred.rob
-rw-------. 1 root root 30 Aug  7 13:31 /root/cred.opie

*# vim /etc/fstab
```

```
...
//servera/common    /mnt/private cifs credentials=/root/cred.rob,multiuser 0 0
//servera/releases  /mnt/private2 cifs credentials=/root/cred.opie 0 0
```

```
*# mount –a
```

```
# df -ht cifs
Filesystem           Size  Used Avail Use% Mounted on
//servera/common      10G  2.7G  7.4G  27% /mnt/private
//servera/releases    10G  2.7G  7.4G  27% /mnt/private

# touch /mnt/private/ro.txt
touch: cannot touch '/mnt/private/ro.txt': Permission denied
# su - brian
$ cifscreds add servera
Password: `postroll`
$ touch /mnt/private/rw.txt
$ ls /mnt/private/rw.txt
-rwxr-xr-x. 1 brian brian 0 Aug  7 13:41 /mnt/private/rw.txt
```

7. 配置 NFS 服务-C10

> 在**servera**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
> - [ ] 目录**/protected**应该包含名为**project**拥有人为**student**的子目录
> - [ ] 用户**student**能以读写方式访问**/protected/project**

[root@servera]

```
-p, --parents
# mkdir -p /public /protected/project

# chown student /protected/project

# mandb && man -k nfs | grep 5
# man exports
# vim /etc/exports
```

```
/public       *.example.com(ro)
/protected    *.example.com(rw)
```

```
# systemctl list-unit-files | grep nfs
# systemctl enable --now nfs-server

# firewall-cmd --permanent --add-service=nfs
# firewall-cmd --reload
```

[root@serverb]

```
# showmount -e servera
clnt_create: **RPC**: Unable to receive
```

[root@servera]

```
# rpc<Tab><Tab>
# rpcinfo
   program version netid     address                service       owner
    100000    4    tcp6      ::.0.111               **portmapper** superuser
    100005    3    tcp6      ::.78.80               **mountd**     superuser
...输出省略...

# grep port.*mapper /etc/services
sunrpc          111/tcp         **portmapper** **rpcbind**      # RPC 4.0 portmapper TCP
sunrpc          111/udp         **portmapper** **rpcbind**      # RPC 4.0 portmapper UDP

# firewall-cmd --permanent --add-service=rpc-bind
# firewall-cmd --reload
```

[root@serverb ~]

```
# showmount -e servera
rpc **mount** export: RPC: Unable to receive; errno = No route to host
```

[root@servera]

```
# firewall-cmd --permanent --add-service=mountd
# firewall-cmd --reload
```

[root@serverb ~]

```
# showmount -e servera
Export list for servera:
/protected *.example.com
/public    *.example.com
```

8. 挂载一个 NFS 共享-C10

> 在**serverb**上挂载一个来自**servera**的NFS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**
> - [ ] 用户**student**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

[root@serverb]

```
# showmount -e servera
Export list for servera:
/protected *.example.com
/public    *.example.com

# mkdir /mnt/nfs{mount,secure}

# man fstab
# vim /etc/fstab
```

```
...输出省略...
servera:/public       /mnt/nfsmount   nfs ro  0 0
servera:/protected    /mnt/nfssecure  nfs rw  0 0
```

```
# mount –a
```

```
# df -h | grep nfs
servera:/public      10G  2.1G  7.9G  22% /mnt/nfsmount
servera:/protected   10G  2.1G  7.9G  22% /mnt/nfssecure

# ls /mnt/nfsmount
# su - student -c "touch /mnt/nfssecure/project/s.txt"
```

9. 配置 iSCSI 服务端-C11(50%)

> 配置**servera**提供一个iSCSI服务，磁盘名为**iqn.2014-11.com.example:servera**，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**3G**
> - [ ] 此服务只能被**serverb.lab.example.com**访问
> - [ ] 后端卷为**LVM**形式

[root@servera]

```
# lsblk
*# fdisk /dev/vda
Command (m for help): `n`
...输出省略...
Select (default p): `<Enter>`
Partition number (2-4, default 2): <Enter>
First sector (20971487-41943039, default 20971520): `<Enter>`
Last sector, +sectors or +size{K,M,G,T,P} (20971520-41943039, default 41943039): `<Enter>`
...输出省略...
Command (m for help): `w`

*# pvcreate /dev/vda2
```

```
*# vgcreate myvg /dev/vda2
*# lvcreate -n mylv -L 3G myvg

*# yum -y install target*

*# systemctl enable --now target

*# targetcli
/> help
/> ls
/> /backstores/block create iscsi_store /dev/myvg/mylv
/> /iscsi create iqn.2014-11.com.example:servera
/> /iscsi/iqn.2014-11.com.example:servera/tpg1/luns create /backstores/block/iscsi_store
/> /iscsi/iqn.2014-11.com.example:servera/tpg1/acls create iqn.2014-06.com.example:serverb
/> ls
o- /
.....................................................................................
[...]
  o- backstores
..................................................................... [...]
  | o- block ..................................................... [Storage
Objects: 1]
  | | o- `iscsi_store` ..................... [`/dev/myvg/mylv` (`3.0GiB`) write-thru
activated]
  | |   o- alua ........................................................... [ALUA
Groups: 1]
  | |     o- default_tg_pt_gp ..................................... [ALUA state:
Active/optimized]
  | o- fileio .................................................... [Storage
Objects: 0]
  | o- pscsi ..................................................... [Storage
Objects: 0]
  | o- ramdisk ................................................... [Storage
Objects: 0]
  o- iscsi ...................................................................
[Targets: 1]
  | o- `iqn.2014-11.com.example:servera` .....................................
[TPGs: 1]
  |   o- tpg1 ....................................................... [no-gen-acls,
no-auth]
  |     o- acls ..................................................................
[ACLs: 1]
  |     | o- `iqn.2014-06.com.example:serverb` ................................ [Mapped
LUNs: 1]
  |     |   o- mapped_lun0 ..................................... [lun0
block/iscsi_store (rw)]
  |     o- luns ..................................................................
[LUNs: 1]
  |     | o- lun0 ..................... [block/`iscsi_store` (/dev/myvg/mylv)
(default_tg_pt_gp)]
  |     o- portals ...............................................................
[Portals: 1]
```

```
  |       o- 0.0.0.0:`3260` ..................................................................
[OK]
  o- loopback ....................................................................
[Targets: 0]
/> exit

*# firewall-cmd --permanent --add-service=iscsi-target
*# firewall-cmd --reload
```

10. 配置 iSCSI 的客户端-C11

> 配置**serverb**使其能连接 在node1上提供的**iqn.2014-11.com.example:servera**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**
> - [ ] 此分区挂载在**/mnt/data**上，同时在系统启动的期间自动挂载

[root@serverb]

```
# yum search iscsi
*# yum -y install iscsi-initiator-utils

*# vim /etc/iscsi/initiatorname.iscsi
```

```
Inatialname=iqn.2014-06.com.example:serverb
```

```
*# systemctl enable --now iscsid iscsiuio

# man iscsiadm | grep -A 2 \\-m
*# iscsiadm --mode discoverydb --type sendtargets --portal servera --discover
172.25.250.10:3260,1 `iqn.2014-11.com.example:servera`

*# iscsiadm --mode node --targetname iqn.2014-11.com.example:servera --portal servera --
login
...
Login to [iface: default, target: iqn.2014-11.com.example:servera, portal:
172.25.250.10,3260] `successful.`

# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
`sda`    8:0    0   3G  0 disk
...

*# fdisk /dev/sda
Command (m for help): `n`
...
Select (default p): `<Enter>`
...
```

```
Partition number (1-4, default 1): `<Enter>`
First sector (2048-6291455, default 2048): `<Enter>`
Last sector, +sectors or +size{K,M,G,T,P} (2048-6291455, default 6291455): `+2100M`
...
Command (m for help): `w`

*# mkfs.ext4 /dev/sda1
# blkid /dev/sda1
/dev/sda1: `UUID="85904e44-4ca9-4ad7-9b7d-587f1996d3df"` TYPE=`"ext4"` PARTUUID="c9460a37-
01"

*# mkdir /mnt/data

# man mount | grep net
*# vim /etc/fstab
```

```
...
UUID="85904e44-4ca9-4ad7-9b7d-587f1996d3df" /mnt/data ext4 _netdev 0 0
```

```
*# mount -a
```

```
# df -ht ext4
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       2.0G  6.2M  1.9G   1% /mnt/data

# reboot
# df -ht ext4
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       2.0G  6.2M  1.9G   1% /mnt/data
```

11. 搭建 MariaDB-C7

> 在**servera**上配置一个数据库服务器，然后执行下述步骤：
>
> - [ ] 仅**localhost** 登录。使用帐户**root**。密码**redhat**
> - [ ] 将http://materials/classroom/database-working/inventory.dump文件下载，并恢复**contacts** 库
> - [ ] 按以下要求设置数据库访问用户
>
> 用户名 **mary**，密码 **mary_password**，对 contacts 数据库所有数据有**选择**权限

[root@servera]

```
# yum search mariadb
*# yum –y install mariadb-server mariadb

*# systemctl enable --now mariadb

# mysql<Tab><Tab>
```

```
*# mysql_secure_installation
...
Enter current password for root (enter for none): `<Enter>`
Set root password? [Y/n] `<Enter>`
New password: `redhat`
Re-enter new password: `redhat`
Remove anonymous users? [Y/n] `<Enter>`
Disallow root login remotely? [Y/n] `<Enter>``
Remove test database and access to it? [Y/n] `<Enter>`
Reload privilege tables now? [Y/n] `<Enter>`

有显示
# netstat -vatnp | grep 3306
tcp  `LISTEN`  0  80  *:`3306`  *:*  users:(("mysqld",pid=9955,fd=22))

# yum -y install mariadb-test
A# grep -r skip-network /usr/share/
B# rpm -ql mariadb-test | grep help
# vim /usr/share/mysql-test/main/mysqld--help.result
...输出省略
--skip-networking   Don't allow connection with TCP/IP

查配置文件位置
# rpm -qc mariadb-server
*# vim /etc/my.cnf.d/mariadb-server.cnf
```

```
[mysqld]
# 添加一行
skip-networking=1
...
```

```
*# systemctl restart mariadb
无显示
# netstat -vatnp | grep 3306

# mysql -u root –predhat
```

```
MariaDB [(none)]> SHOW databases;
MariaDB [(none)]> HELP create;
MariaDB [(none)]> CREATE database contacts;
MariaDB [(none)]> HELP grant;
...输出省略...
CREATE USER 'jeffrey'@'localhost' IDENTIFIED BY 'mypass';
GRANT ALL ON db1.* TO 'jeffrey'@'localhost';
GRANT SELECT ON db2.invoice TO 'jeffrey'@'localhost';
GRANT USAGE ON *.* TO 'jeffrey'@'localhost' WITH MAX_QUERIES_PER_HOUR 90;
...输出省略...
MariaDB [(none)]> CREATE USER 'mary'@'localhost' IDENTIFIED BY 'mary_password';
MariaDB [(none)]> GRANT SELECT ON contacts.* TO 'mary'@'localhost';
MariaDB [(none)]> <Ctrl-D>
```

```
*# wget http://materials/classroom/database-working/inventory.dump

*# mysql -u root -predhat contacts < inventory.dump
```

测试方法1

```
# mysql -u mary -pmary_password contacts -e 'SHOW tables;'
+--------------------+
| Tables_in_contacts |
+--------------------+
| category           |
| manufacturer       |
| product            |
+--------------------+
```

测试方法2

```
# mysql -u mary -pmary_password
```

```
MariaDB [(none)]> USE contacts;
MariaDB [contacts]> SHOW tables;
MariaDB [contacts]> <Ctrl-D>
```

12. 数据查询填空1-C7

> 完成以下要求的查询并将结果填入相应的框格中。
>
> - [ ] 查询**RT-AC68U**产品的供应商名称为_____

Important - 重要

考试时，数据库里的表不同。答案是 **Pete**

[root@servera]

```
MariaDB [contacts]> help select;
MariaDB [contacts]> show tables;

MariaDB [contacts]> SELECT * FROM product;
MariaDB [contacts]> SELECT * FROM category;
MariaDB [contacts]> SELECT * FROM manufacturer;

MariaDB [contacts]> SELECT manufacturer.name
FROM product, manufacturer
WHERE id_manufacturer=manufacturer.id
 AND product.name="RT-AC68U";
```

13. 数据查询填空2-C7

> 完成以下要求的查询并将结果填入相应的框格中。
>
> - [ ] 查询类型是**Servers**，且供应商是**Lenovo**的产品有多少_____种

Important - 重要

考试时，数据库里的表不同。答案是 **2**

[root@servera]

```
MariaDB [contacts]> SELECT count(*)
  FROM category, manufacturer, product
  WHERE id_category=category.id
    AND id_manufacturer=manufacturer.id
    AND category.name="Servers"
    AND manufacturer.name="Lenovo";
MariaDB [contacts]> <Ctrl-D>
```

14. 实现一个 web 服务器-C9

> 在**servera**上配置一个站点http://www0.lab.example.com然后执行下述步骤：
>
> - [ ] 从http://materials/www0.html下载文件
> - [ ] 并且将文件重命名为**index.html**不要修改此文件的内容
> - [ ] 将文件**index.html**拷贝到您的web服务器的**DocumentRoot**目录下
> - [ ] 来自于**lab.example.com**域的客户端可以访问此web服务
> - [ ] 来自于**lab.example.org**域的客户端拒绝访问此Web服务

[root@servera]

```
# yum search http | grep ^http
# yum -y install httpd httpd-manual

# rpm -ql httpd | grep vhost
# cp /usr/share/doc/httpd/httpd-vhosts.conf /etc/httpd/conf.d/

# wget -O /var/www/html/index.html http://materials/www0.html

# rpm -ql httpd-manual | grep conf

# grep -i alias /etc/httpd/conf.d/manual.conf
Alias `/manual` /usr/share/httpd/manual
```

[kiosk@foundation]

> http://servera/manual
>
> `Upgrading to 2.4 from 2.2`

[root@servera]

```
*# vim /etc/httpd/conf.d/httpd-vhosts.conf
```

```
<VirtualHost *:80>
     DocumentRoot "/var/www/html"
     ServerName www0.lab.example.com
</VirtualHost>
```

```
# systemctl enable --now httpd

# firewall-cmd --permanent --add-service=http
# firewall-cmd –reload
```

```
# vim /etc/httpd/conf/httpd.conf
```

```
...
<Directory "/var/www/html">
...
    #Require all granted
    <RequireAll>
        Require host lab.example.com
        Require not host lab.example.org
    </RequireAll>
...
```

```
# systemctl restart httpd
```

[root@serverb]

```
# curl http://www0.lab.example.com
www0
```

15. 配置安全 web 服务-C8

> 站点http://www0.lab.example.com配置TLS加密
>
> - [ ] 一个已签名证书从http://materials/www0.lab.example.com.crt获取
> - [ ] 此证书的密钥从http://materials/www0.lab.example.com.key获取
> - [ ] 此证书的签名授权信息从http://materials/example-ca.crt获取

[root@servera]

```
# yum search ssl
# yum -y install mod_ssl

查证书文件所在位置
# rpm -qc mod_ssl
# grep Chain /etc/httpd/conf.d/ssl.conf
#SSLCertificateChainFile /etc/pki/tls/certs/server-chain.crt
# grep localhost /etc/httpd/conf.d/ssl.conf
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key

# wget -P /etc/pki/tls/certs/ http://materials/example-ca.crt
# wget -P /etc/pki/tls/certs/ http://materials/www0.lab.example.com.crt
# wget -P /etc/pki/tls/private/ http://materials/www0.lab.example.com.key

# vim /etc/httpd/conf.d/ssl.conf
```

```
...
<VirtualHost _default_:443>
#DocumentRoot "/var/www/html"
DocumentRoot "/var/www/html"
#ServerName www.example.com:443
ServerName www0.lab.example.com
...
SSLCertificateFile /etc/pki/tls/certs/www0.lab.example.com.crt
SSLCertificateKeyFile /etc/pki/tls/private/www0.lab.example.com.key
SSLCertificateChainFile /etc/pki/tls/certs/example-ca.crt
...
```

```
# systemctl restart httpd

# firewall-cmd --permanent --add-service=https
# firewall-cmd –reload
```

[root@serverb]

```
# curl http://www0.lab.example.com
www0
# curl -k https://www0.lab.example.com
www0
```

16. 配置虚拟主机-C8

> 在**servera**上扩展您的web服务器，
> 为站点http://webapp0.lab.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://materials/webapp0.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放到虚拟机的DocumentRoot目录下，确保**floyd**用户能够在/var/www/virtual目录下创建文件
>
> 注意：原始站点http://www0.lab.example.com必须仍然能够访问，
> 名称服务器 bastion.lab.example.com 提供对主机名 webapp0.lab.exmple.com 的域名解析

[root@servera]

```
*# mkdir /var/www/virtual

# wget -O /var/www/virtual/index.html http://materials/webapp0.html

# id floyd
# chown floyd /var/www/virtual

# vim /etc/httpd/conf.d/httpd-vhosts.conf
```

```
...
<VirtualHost *:80>
     DocumentRoot "/var/www/virtual"
     ServerName webapp0.lab.example.com
</VirtualHost>
```

```
# systemctl restart httpd
```

[root@serverb]

```
# curl http://www0.lab.example.com
www0
# curl http://webapp0.lab.example.com
webapp0
# curl –k https://www0.example.com
www0
```

17. 配置 web 内容的访问-C8

> 在您的**servera**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://materials/permission.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
> - [ ] 从 **servera** 上，任何人都可以浏览**private**的内容，但是从其它系统不能访问这个目录的内容

[root@servera]

```
*# mkdir /var/www/{html,virtual}/private
# tree /var/www/{html,virtual}

*# wget http://materials/permission.html \
    -O /var/www/html/private/index.html

*# wget http://materials/permission.html \
    -O /var/www/virtual/private/index.html

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

```
...
# 新增加 3 行 * 2
<Directory "/var/www/html/private">
    Require host servera.lab.example.com
</Directory>
<Directory "/var/www/virtual/private">
    Require host servera.lab.example.com
</Directory>
# 下面行为参照物
<Directory "/var/www">
...
```

```
*# systemctl restart httpd
```

[root@servera]

```
# curl http://www0.lab.example.com/private/
permission

# curl http://webapp0.lab.example.com/private -L
permission
```

[root@serverb]

```
# curl http://www0.lab.example.com
www0
# curl -k https://www0.lab.example.com
www0
# curl http://webapp0.lab.example.com
webapp0

# curl http://www0.lab.example.com/private/
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 `Forbidden`</title>
</head><body>
<h1>Forbidden</h1>
<p>You don\'t have permission to access /private/ on this server.<br />
</p>
</body></html>

# curl http://webapp0.lab.example.com/private/
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 `Forbidden`</title>
</head><body>
<h1>Forbidden</h1>
<p>You don\'t have permission to access /private/ on this server.<br />
</p>
</body></html>
```

18. 通过 ansible 布署 Nginx - C8

> 在**serverc**和**serverd**上配置一个站点，然后执行下述步骤：
>
> - [ ] 在**playbooks**目录下，创建剧本**nginx.yml**
> - [ ] 从http://foundation0.ilt.example.com/nginx.conf.j2下载jinja模板文件
> - [ ] 从http://materials/nginx.html下载文件，并且将文件重命名为**index.html**不要修改此文件的内容
> - [ ] 将文件**index.html**拷贝到您的 web 服务器的 DocumentRoot 目录下**/www/src/html**

[root@serverb] 不安装在serverc和serverd，也不要安装在servera就可以

```
# yum -y install nginx

参考**配置文件**, **权限**
# rpm -qc nginx
# vim /etc/nginx/nginx.conf

# ls -ldZ /usr/share/nginx/html
```

[devops@workstation]

```
*$ cd playbooks

*$ ansible --version
config file = `/home/devops/playbooks/ansible.cfg`
...
*$ ansible-inventory --graph
@all:
  |--@ungrouped:
  |  |--`serverc`
  |  |--`serverd`
*$ ansible all -a whoami
serverc | CHANGED | rc=0 >>
`root`
serverd | CHANGED | rc=0 >>
`root`
```

```
*# wget http://foundation0.ilt.example.com/nginx.conf.j2

# vim nginx.conf.j2
```

```
{# 下3行，需添加 #}
events {
    worker_connections 1024;
}

{# http 行，需添加 #}
http {
  server {
      listen       80 default_server;
      listen       [::]:80 default_server;
      {# ; 分号结尾 #}
      server_name  {{ ansible_fqdn }};
      root         /www/src/html;

      include /etc/nginx/default.d/*.conf;
      location / {
      }
      error_page 404 /404.html;
          location = /40x.html {
```

```
        }
        error_page 500 502 503 504 /50x.html;
            location = /50x.html {
        }
    }
{# 和 http 成对匹配 #}
}
```

*# vim nginx.yml

```
---
- name: nginx
  hosts: serverc,serverd
# become: yes
  tasks:
  - name: install the latest version
    yum:
      name: nginx
      state: latest
  - name: Template a file
    template:
      src: nginx.conf.j2
      dest: /etc/nginx/nginx.conf
  - name: Start service
    service:
      name: nginx
      state: started
      enabled: yes
  - name: mkdir
    file:
      path: /www/src/html
      state: directory
  - name: Download
    get_url:
      url: http://materials/nginx.html
      dest: /www/src/html/index.html
      # 考点
      setype: httpd_sys_content_t
  - firewalld:
      service: http
      permanent: yes
      state: enabled
      # 立即生效
      immediate: yes
```

*$ ansible-playbook nginx.yml

[root@serverb]

```
# curl http://serverc.lab.example.com
nginx
# curl http://serverd.lab.example.com
nginx
```

19. 通过 ansible 配置 firewall

> 在 **serverc** 和 **serverd** 上分别设置，针对SSH
>
> - [ ] 在**playbooks**目录下，创建剧本**firewall.yml**
> - [ ] 允许**172.25.250.0/24** 的域对 serverc 和 serverd 进行**SSH**
> - [ ] 禁止**172.24.250.0/24** 的域对 serverc 和 serverd 进行**SSH**

[devops@workstation]

```
$ cd playbooks
$ ansible --version
$ ansible-inventory --graph
$ ansible all -a whoami

$ ansible-doc -l | grep fire
$ ansible-doc firewalld
https://docs.ansible.com/ansible/2.9/reference_appendices/playbooks_keywords.html

$ echo set number ts=2 sw=2 et > ~/.vimrc

*$ vim firewall.yml
```

```
---
- hosts: serverc, serverd
  tasks:
  - firewalld:
      rich_rule: "{{ item }}"
      permanent: yes
      state: enabled
      immediate: yes
    loop:
    - 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
  - firewalld:
      service: ssh
      permanent: yes
      state: disabled
      immediate: yes
```

```
*$ ansible-playbook firewall.yml
...
PLAY RECAP ````````````````````````````````````````````
serverc : ok=3 changed=3    unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
serverd : ok=3 changed=3    unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

```
$ ansible serverc,serverd -a 'firewall-cmd --list-all'
serverc | CHANGED | rc=0 >>
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0
  sources:
  services: cockpit dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  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
...
```

20. 通过 ansible 配置空邮件客户端 - C6

> 在系统 **serverc** 和 **serverd** 上配置邮件服务，满足以下要求：
>
> - [ ] 在**playbooks**目录下，创建剧本**nullclients.yml**
> - [ ] 这些系统不能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**smtp.lab.example.com**
> - [ ] 从这些系统上发送的邮件显示来自于**lab.example.com**
> - [ ] 您可以通过发送邮件到本地用户**student**来测试您的配置，系统**smtp.lab.example.com** 已经配置
> - [ ] 可以通过如下网址测试**http://bastion/mail**

Important - 重要

不要太相信 README

[root@servera]

```
# yum search mail | grep -i M.*T.*A
...
```

```
`postfix.x86_64` : Postfix Mail Transport Agent

*# yum -y install postfix

# rpm -ql postfix | grep -i stand
/usr/share/doc/postfix/README_FILES/STANDARD_CONFIGURATION_README
# vim /usr/share/doc/postfix/README_FILES/STANDARD_CONFIGURATION_README
...
 69     1 /etc/postfix/main.cf:
 70     2     myhostname = hostname.example.com
 71     3     myorigin = $mydomain
 72     4     relayhost = $mydomain
 73     5     inet_interfaces = loopback-only
 74     6     mydestination =

永久生效
# vim /etc/postfix/main.cf
已经生效，立即生效
# postconf | egrep '^inet.*int'
# postconf | egrep '^relayh|^mydes'
# postconf | grep ^myori

b=before
a=after
5b inet_interfaces = localhost
5a inet_interfaces = loopback-only
6b mydestination = $myhostname, localhost.$mydomain, localhost
6a mydestination =
4b relayhost =
4a relayhost = [smtp.lab.example.com]
3b myorigin = $myhostname
3a myorigin = lab.example.com
```

[devops@workstation]

```
$ yum search role
*$ sudo yum -y install rhel-system-roles
$ ansible-galaxy list
...
- `rhel-system-roles.postfix`, (unknown version)

$ rpm -ql rhel-system-roles | grep postfix
...
`/usr/share/doc/rhel-system-roles/postfix/README.md`

只能参考，不要全抄
$ vim /usr/share/doc/rhel-system-roles/postfix/README.md
 20 ```yaml
 21 ---
 22 - hosts: all
 23   vars:
```

```
 24     postfix_conf:
 25       relay_domains: "$mydestination"
 26       relay_host: "example.com"
 27   roles:
 28     - linux-system-roles.postfix
 29 ```

*$ vim nullclients.yml
```

```
---
- hosts: serverc,serverd
  become: true
  vars:
    postfix_conf:
      inet_interfaces: "loopback-only"
      relayhost: "[smtp.lab.example.com]"
      mydestination: ""
      myorigin: "lab.example.com"
  roles:
    - linux-system-roles.postfix
```

```
*$ ansible-playbook nullclients.yml
```

[root@server{c..d}]

```
# yum provides mail
# yum -y install mailx

给本地用户发邮件
nr=内容
-s, subject标题
# echo nr1 | mail -s local student
# echo nr2 | mail -s domain student@lab.example.com
```

> 测试方法A: **firefox http://bastion/mail**
>
> 考试环境是，student链接可以直接打开
>
> ```
> Name  Last modified Size  Description
>   ...
>   `student`   2021-08-01 11:09  `6.9K`
> ```

测试方法B：练习环境支持

```
# mutt -f imap://smtp.lab.example.com
/home/student/Mail does not exist. Create it? ([yes]/no): `<Enter>`
(r)eject, accept (o)nce, (a)ccept always `a`
Username at imap.lab.example.com: `student`
Password for tom@imap.lab.example.com: `student`
```

21. 通过 ansible 布署打印机-C5(33%)

> - [ ] 在**playbooks**目录下，创建剧本**printer-create.yml**
> - [ ] 安装打印机为默认打印机
> - [ ] printer queue **new-printer**
> - [ ] URI 地址**ipp://serverc:631/printers/rht-printer**

Hint - 提示

- 考试环境中不需要配置servera
- 考试环境中只需要配置workstation剧本，管理serverc、serverd

[root@servera] 手动配置

```
查找打印机服务
Ma
*# yum provides lpadmin
Mb
*# yum search print

发现打印机服务
Ma
*# yum search discover
Mb
*# rpm -ql firewalld | grep mdns
/usr/lib/firewalld/services/mdns.xml
*# cat /usr/lib/firewalld/services/mdns.xml
...输出省略...
  <description>mDNS provides the ability to use DNS programming interfaces, packet formats
and operating semantics in a small network without a conventional DNS server. If you plan
to use `Avahi`

*# yum install cups avahi

# rpm -ql cups | grep service
/usr/lib/systemd/system/cups.service
*# systemctl enable --now cups
```

```
# rpm -ql avahi | grep service
/usr/lib/systemd/system/avahi-daemon.service
*# systemctl enable --now avahi-daemon

-new-printer
# man lpadmin
-p  named printer
-E  Enables
-v  "device-uri"
-m  model
-d  default printer
lpadmin -p myprinter -E -v ipp://myprinter.local/ipp/print -m everywhere

*# lpadmin -p new-printer -E -v ipp://serverc:631/printers/rht-printer -m everywhere
ipp://serverc.lab.example.com:631/printers/rht-printer

-d, default
*# lpadmin -d new-printer

*# firewall-cmd --permanent \
    --add-service=ipp-client \
    --add-service=mdns
*# firewall-cmd --reload
```

```
# man lpstat
-e  Shows all available destinations
-d  Shows the current default destination
-v  Shows the printers and what device they are attached to
# lpstat -e
new-printer

# lpstat -d
system default destination: new-printer

# lpstat -v
```

[devops@workstation] 自动配置

```
确认生效的配置文件，是当前目录中的ansible.cfg
*$ cd playbooks
*$ ansible --version
config file = `home/devops/playbooks/ansible.cfg`
...

*$ cat ansible.cfg
[defaults]
inventory=./inventory
host_key_checking = False

确认主机清单，包含serverc和serverd
```

```
*$ ansible-inventory --graph
@all:
  |--@ungrouped:
  |  |--`serverc`
  |  |--`serverd`
```

确认远程管理身份是root。-m command 此模块是默认的，可省略；-a arg参数就是我们会的命令

```
*$ ansible all -a whoami
#$ ansible ungrouped -a whoami
#$ ansible serverc,serverd -a whoami
serverc | CHANGED | rc=0 >>
`root`
serverd | CHANGED | rc=0 >>
`root`
```

查文档，模块名和用法

```
$ ansible-doc -l | grep fire
$ anisble-doc yum
$ anisble-doc service
$ anisble-doc firewalld
$ ansible-doc shell
/EX
```

提高yml文件，修改效率。降低出错机率

```
*$ cat > ~/.vimrc <<EOF
set number ts=2 et cuc sw=2
EOF

*$ vim printer-create.yml
```

```
- hosts: all
  # 注意提权
  become: true
  tasks:
  # 安装
  - name: ensure a list of packages installed
    yum:
      name: "{{ packages }}"
    vars:
      packages:
      - cups
      - avahi
  # 启动服务
  - name: Start service httpd, if not started
    service:
      name: cups
      state: started
      enabled: yes
    loop:
    - cups
    - avahi-daemon
```

```
  # 添加打印机
  - name: Run expect to wait for a successful PXE boot>
    shell: |
      lpadmin -p new-printer -E -v ipp://serverc:631/printers/rht-printer -m everywhere
      lpadmin -d new-printer
  # 防火墙
  - firewalld:
      service: "{{ item }}"
      permanent: yes
      state: enabled
      immediate: yes
    loop:
    - ipp-client
    - mdns
```

```
注意recap下面的输出，没有红色
*$ ansible-playbook printer-create.yml
...输出省略...
PLAY RECAP ****************************************************
serverc : ok=6  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
serverd : ok=6  changed=5  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

```
*$ ansible all -a 'lpstat -d'
serverc | CHANGED | rc=0 >>
system default destination: new-printer
serverd | CHANGED | rc=0 >>
system default destination: new-printer

$ ansible all -a 'lpstat -v'
serverc | CHANGED | rc=0 >>
device for **new-printer: ipp://serverc:631/printers/rht-printer**
device for rht-printer: ipp://serverd:8000/ipp/print/ipp-everywhere-pdf
serverd | CHANGED | rc=0 >>
device for **new-printer: ipp://serverc:631/printers/rht-printer**
```

APPENDENCES

A1. 红帽认证服务管理和自动化专家练习

> 持续时间 **4.00 小时**

> ### 练习要点
>
> 除了以下所列目标外，参加红帽认证服务管理和自动化专家练习的考生还应参考红帽认证工程师练习（RHCE）的练习目标，并应能够胜任 RHCE 级别的任务，因为本练习可能要求考生掌握这些技能。

参加红帽认证服务管理和自动化专家练习的考生应能够独立完成下列任务。考生应能够手动和利用 Ansible 自动化执行这些任务。

- **管理网络服务**
  - **配置网络客户端，以使用**动态或**静态分配的地址**
  - **使用 IPv4 和 IPv6 工作**
- **管理防火墙服务**
  - **配置系统防火墙，以允许访问特定的服务或端口**
  - **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**
- **管理 SELinux**
- 配置指定服务的 SELinux 布尔值
- **配置文件或目录的 SELinux 上下文**
- **管理系统进程**
- **将系统进程配置为在系统引导时启动**
- 阻止系统进程启动
- **管理链路聚合**
- 创建由两个网络接口构成的网络组接口
- 使网络组接口在系统重启后保持有效
- 为网络组接口分配网络地址
- 配置 teamd 运行程序
- **管理 DNS**
- 配置缓存名称服务器
- **利用部分完成的区域文件配置授权域名服务器**
- **配置 IPv4 和 IPv6 地址正向与反向查询**
- **管理 DHCP**
- **配置指定地址范围内的地址分配**
- **将特定地址分配配置给指定的主机**
- **配置 IPv4** 和 IPv6 **地址分配**
- **管理打印机**
- **创建和管理网络打印机的打印机队列**
- 管理现有的打印机队列
- **管理电子邮件服务**
- **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**
- **管理 MariaDB 数据库服务器**
  - **安装和配置基本的 MariaDB 服务**

    - **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**
    - 添加记录到现有的 MariaDB 数据库
    - **提交对 MariaDB 数据库的简单 SQL 查询**
    - 创建 MariaDB 备份
    - **从备份导入 MariaDB 数据库**
- **管理 HTTPD Web 访问**
- **安装和配置 Apache**
- **安装和配置 NGINX**
- **配置替代文档根目录**
- 配置替代 Web 访问端口
- **配置基于名称的虚拟主机**
- **配置安全 Web 服务器（HTTPS）**
- 提供静态缓存以缩短 HTTP 响应时间
- 配置 HTTP HAProxy 负载平衡器
- 终止 HTTPS 连接
- **管理 iSCSI**
- **提供和配置 iSCSI 目标**
- **配置 iSCSI 启动器，以持久连接 iSCSI 目标**
- **将对 iSCSI 服务的访问限制到特定的客户端和访问**
- **管理 NFS**
- **配置持久 NFS 导出**
- **配置 NFS 客户端，以持久挂载 NFS 导出**
- **将对 NFS 导出的访问限制到特定的客户端和网络**
- **管理 SMB**
    - **配置 SMB 共享**
    - **创建和管理 SMB 用户**
    - **创建仅限 SMB 用户**
    - **限制对 SMB 共享的访问**
    - **挂载 SMB 共享**
    - **执行多用户 SMB 挂载**
- **使用 Ansible 配置标准服务**
    - **创建和修改 playbook**

> - 了解和使用清单文件
> - 在 playbook 中使用变量
> - 使用 RHEL 系统角色
>
> 对于所有实际任务操作型的红帽练习，您的所有系统配置必须在重启后仍然有效（无需人工干预）。

A2. 补充

- 英文考试
- nginx：
  j2模板有问题。是只有server { 开始的，没有外面两层，把外面两层补上。要对照 nginx.conf.default 把外面两层补上
- DNS：
  ex358-net.zone 一个是ex358-example.zone
- mariadb：
  名字有 firstname 和lastname 两个字段 我查询只有 lastname 有 这个名 所以 要么用 or 查 要么 分别查一下两个字段 最后人数是 1
- smb：
  两道题第一题共享一个A 目录用a 用户 可读没提挂载的事。第二题是共享B目录用c 用户可读d用户可写。 并且挂载 两个提目录不同，英文没斜体或粗体，一不注意就搞错
- ansible:

  提权

A3. 相關鏈接

ID	鏈接	作用
1	https://pan.baidu.com/s/1NKnX4QFWTM9Oj852zwSmXg?pwd=trhf	培訓環境
2	https://pan.baidu.com/s/1mI5li29nchmx2qt9BFBGxQ?pwd=ck2o	相關文檔
3	https://www.redhat.com/zh/services/training/ex358-red-hat-certified-specialist-services-management-automation-exam?section=%E8%80%83%E8%AF%95%E7%9B%AE%E6%A0%87	红帽认证服务管理和自动化专家考试
4		
5		