

说明

- 1 正式考试时，以下题目将会分布到两个主机。在考试过程中，请仔细阅读题目，明确题目要求该题在哪个虚拟机中作答。
 - 2 练习时，所有题目都将在一个虚拟机中执行
 - 3 root的密码是huayinetwork，training 用户密码： huayinetwork， 登录机器使用 training用户， 登录console界面使用root
 - 4 <https://registry.lab.example.com> 提供所需的容器镜像文件，账户 redhat 密码 Redhat@132
-

虚拟机操作

命令行：

- `vmctl status all` 查看所有虚拟机状态， `all` 可以换成对应的虚拟机，即表示查看指定虚拟机状态
- `vmctl start all` 启动所有虚拟机
- `vmctl reset all` 重置所有虚拟机

图形化操作：

- 使用 RDP 方式连接到主机，打开终端，输入 `virt-manager`，即可弹出虚拟机操作界面

在 `servera.lab.example.com` 上完成如下实验：

1、按要求配置网络

配置网络，要求如下：

- 主机名：servera.lab.example.com
- IP 地址：192.168.100.10/24
- 网关：192.168.100.1
- DNS：172.31.0.245

#在虚拟机的console界面里面配置，因为没有网络，不能使用SSH

\$ virt-manager

#输入的密码是training的密码

```
[training@yn2460 ~]$ ssh training@192.168.100.10
training@192.168.100.10's password:
```

```
[training@servera ~]$ sudo -i
[root@servera ~]#
[root@servera ~]# hostnamectl set-hostname servera.lab.example.com
[root@servera ~]# hostname
servera.lab.example.com
```

```
[root@servera ~]# exit
logout
[training@servera ~]$ exit
logout
Connection to 192.168.100.10 closed.
```

```
[training@yn2460 ~]$ ssh training@192.168.100.10
training@192.168.100.10's password:
Last login: Tue Mar 19 05:12:51 2024 from 192.168.100.1
[training@servera ~]$ hostname
servera.lab.example.com
```

#修改SSH的配置文件，允许root远程登录

```
[root@servera ~]# vi /etc/ssh/sshd_config
```

```
PermitRootLogin yes
```

```
[root@servera ~]# reboot
```

2、配置系统的软件仓库

给系统配置默认软件仓库，要求如下：

YUM 的两个存储库的地址分别是：

- <http://content.example.com:17242/rhel9.0/dvd/AppStream/>
- <http://content.example.com:17242/rhel9.0/dvd/BaseOS/>

```
[training@yn2460 ~]$ ssh root@192.168.100.10
```

```
root@192.168.100.10's password:
```

```
[root@centos7-vm01 ~]# cd /etc/yum.repos.d/
```

```
[root@centos7-vm01 yum.repos.d]# vi anyname.repo
```

```
[root@centos7-vm01 yum.repos.d]# cat anyname.repo
```

```
[BaseOS]
```

```
name=baseos
```

```
baseurl=http://content.example.com:17242/rhel9.0/dvd/BaseOS/
```

```
gpgcheck=0
```

```
#复制4行 修改baseurl
```

```
[AppStream]
```

```
name=appstream
```

```
baseurl=http://content.example.com:17242/rhel9.0/dvd/AppStream/
```

```
gpgcheck=0
```

```
#安装vim编辑器
```

```
[root@servera yum.repos.d]# yum install vim -y
```

3、配置 SELinux

非标准端口 82 上运行的 WEB 服务器在提供内容时遇到问题。根据需要调试并解决问题， 并使其满足以下条件:

- web 服务器能够提供 /var/www/html 中所有现有的 html 文件(注意: 不要删除或改动现有的文件内容)
- Web 服务器通过 82 端口访问
- Web 服务器在系统启动时自动启动

```
[root@servera ~]# systemctl status httpd
○ httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service;
disabled; v>
   Active: inactive (dead)
     Docs: man:httpd.service(8)

[root@servera ~]# rpm -qa |grep httpd
httpd-tools-2.4.51-7.el9_0.x86_64
redhat-logos-httpd-90.4-1.el9.noarch
httpd-filesystem-2.4.51-7.el9_0.noarch
httpd-2.4.51-7.el9_0.x86_64

#添加82端口到selinux 允许的http的端口
[root@servera ~]# semanage port -a -t http_port_t -p tcp 82

[root@servera ~]# semanage port -l |grep http_port_t
http_port_t                                tcp      82, 80, 81, 443, 488, 8008,
8009, 8443, 9000

[root@servera ~]# systemctl enable --now httpd
Created symlink /etc/systemd/system/multi-
user.target.wants/httpd.service →
/usr/lib/systemd/system/httpd.service.
[root@servera ~]#
[root@servera ~]#
```

```
[root@servera ~]# ss -ltpn |grep 82
LISTEN 0      511          *:82          *:~          users:
(("httpd",pid=2038,fd=4),("httpd",pid=2037,fd=4),
("httpd",pid=2036,fd=4),("httpd",pid=2034,fd=4))

[root@servera ~]# restorecon -RFv /var/www/html/

#测试
[root@servera ~]# cd /var/www/html/
[root@servera html]# ls
file1  file2  file3

[root@servera ~]# curl localhost:82/file1
<h1>Sidere mens eadem mutato</h1>

[root@servera ~]# curl localhost:82/file2
<h2>Generatim discite cultus</h2>

[root@servera ~]# curl localhost:82/file3
through hardships towards the sun
```

4、创建用户账户

创建下列用户、用户组，并按要求完成设置：

- 组名为 sysmgrs
- natasha 用户的附属组是 sysmgrs
- harry 用户的附属组是 sysmgrs
- john 用户的 shell 是非交互式 shell，且不是 sysmgrs 组的成员
- natasha、harry、john 的密码是 redhat

```
[root@servera ~]# groupadd sysmgrs

[root@servera ~]# useradd -G sysmgrs natasha
```

```
[root@servera ~]# useradd -G sysmgrs harry
```

```
[root@servera ~]# useradd -s /sbin/nologin john
```

```
[root@servera ~]# passwd natasha
```

Changing password for user natasha.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

```
[root@servera ~]# passwd harry
```

Changing password for user harry.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

```
[root@servera ~]# passwd john
```

Changing password for user john.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

#测试用户的密码是否正确

```
[root@servera ~]# su - natasha
```

```
[natasha@servera ~]$ su - john
```

Password:

This account is currently not available.

```
[natasha@servera ~]$ su - harry
```

Password:

```
[harry@servera ~]$ exit
```

logout

```
[natasha@servera ~]$ exit
```

logout

#检查用户的信息和属组

```
[root@servera ~]# grep sysmgrs /etc/group
```

```
sysmgrs:x:1001:natasha,harry
```

```
[root@servera ~]# id natasha
uid=1001(natasha) gid=1002(natasha)
groups=1002(natasha),1001(sysmgrs)
```

```
[root@servera ~]# id harry
uid=1002(harry) gid=1003(harry) groups=1003(harry),1001(sysmgrs)
```

```
[root@servera ~]# id john
uid=1003(john) gid=1004(john) groups=1004(john)
```

5、配置 crontab 计划任务

该作业每两隔两分钟执行以下命令：

- `logger "Come from ex200"`，以用户 natasha 身份运行

i 注：每天 14: 23 分 natasha 执行 echo “Hi rhcsa” 也是考察要点

```
[root@servera ~]# crontab -e -u natasha
*/2 * * * * natasha logger "Come from ex200"
23 14 * * * natasha echo "Hi rhcsa"
```

#检验计划任务

```
[root@servera ~]# cd /var/spool/cron/
[root@servera cron]# ls
```

natasha

```
[root@servera cron]# cat natasha
*/2 * * * * natasha logger "Come from ex200"
23 14 * * * natasha echo "Hi rhcsa"
```

#通过日志校验，等待2分钟

```
[root@servera cron]# tail -f /var/log/cron
Mar 19 07:50:02 servera CROND[2395]: CMDEND (natasha logger "Come
from ex200")
```

6、按要求创建目录

创建具有特殊权限的目录，要求如下：

- `/home/managers` 目录属于 `sysmgrs` 组
- 此目录可以被 `sysmgrs` 的组成员读取、写入和访问，但是其他任何用户不具备这些权限。（不包括 `root` 用户）
- 在 `/home/managers` 目录中创建的文件的所有组自动变成 `sysmgrs` 组

```
[root@servera ~]# mkdir -p /home/managers
[root@servera ~]# chgrp sysmgrs /home/managers/

[root@servera ~]# chmod g+ws /home/managers/

[root@servera ~]# chmod o=--- /home/managers/

#查看权限
[root@servera ~]# ls -ld /home/managers/
drwxrws---. 2 root root 6 Mar 19 07:56 /home/managers/

#测试特别权限
[root@servera ~]# su - natasha
Last login: Tue Mar 19 07:57:31 EDT 2024 on pts/0
[natasha@servera ~]$ cd /home/managers/
[natasha@servera managers]$ touch myfile
[natasha@servera managers]$ ll
total 0
-rw-r--r--. 1 natasha sysmgrs 0 Mar 19 07:58 myfile
```

7、配置 NTP

配置 `servera` 作为 NTP 的客户端，跟时间服务器 `172.31.0.245` 同步时间


```
[root@servera ~]# vim /etc/chrony.conf
pool 172.31.0.245 iburst

[root@servera ~]# systemctl enable chronyd
[root@servera ~]# systemctl restart chronyd.service

#校验ntp
[root@servera ~]# chronyc sources
MS Name/IP address          Stratum Poll Reach LastRx Last sample

=====
=====
^? 172.31.0.245              3      6      1      2  -1900us[-1900us]
+/-      21ms
```

8、配置 autofs

配置 autofs，按照以下要求自动挂载远程用户的家目录，要求如下：

- NFS 服务器 172.31.0.242 的共享目录是 /rhome,此文件系统包含为用户 ldapuser 预配置的家目录
- ldapuser 的家目录是 172.31.0.242:/rhome/ldapuser
- ldapuser 的家目录应该自动挂载到 /rhome 下的 /rhome/ldapuser
- 家目录允许用户写入
- ldapuser 的密码是 redhat

```
[root@servera ~]# yum install nfs-utils autofs -y

#查看nfs server共享的目录
[root@servera ~]# showmount -e 172.31.0.242
Export list for 172.31.0.242:
/rhome          *

#showmount -e 能看见的目录就是监控目录
[root@servera ~]# vim /etc/auto.master
```

```
/rhome      /etc/auto.misc
#监控目录   监控动作

[root@servera ~]# vim /etc/auto.misc
ldapuser    -fstype=nfs,rw 172.31.0.242:/rhome/ldapuser

[root@servera ~]# systemctl enable --now autofs
Created symlink /etc/systemd/system/multi-
user.target.wants/autofs.service →
/usr/lib/systemd/system/autofs.service.

#测试
[root@servera ~]# su - ldapuser
```

9、配置用户账户

配置用户账户 alex，用户的 ID 为 1234，此用户的密码应当为 redhat.

```
[root@servera ~]# useradd -u 1234 alex

[root@servera ~]# passwd alex
Changing password for user alex.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.

[root@servera ~]# id alex
uid=1234(alex) gid=1234(alex) groups=1234(alex)
```

10、查找文件

查找属与 harry 用户的所有文件，并拷贝到 `/root/findfiles` 目录

```
[root@servera ~]# mkdir /root/findfiles
```

#命令是find的表达式 一定要以 \; 结束

```
[root@servera ~]# find / -user harry -exec cp -r {} /root/findfiles/ \;
```

```
find: '/proc/3829/task/3829/fd/5': No such file or directory
```

```
find: '/proc/3829/task/3829/fdinfo/5': No such file or directory
```

```
find: '/proc/3829/fd/6': No such file or directory
```

```
find: '/proc/3829/fdinfo/6': No such file or directory
```

```
cp: cannot overwrite non-directory '/root/findfiles/harry' with directory '/home/harry'
```

#报错是正常的

```
[root@servera ~]# cd /root/findfiles/
```

```
[root@servera findfiles]# ls
```

```
harry
```

```
[root@servera findfiles]# ls -a
```

```
.  ..  .bash_history  .bash_logout  .bash_profile  .bashrc  harry  .zshrc
```

```
[root@servera findfiles]#
```

11、查找字符串

找出文件 `/usr/share/dict/words` 中包含字符串 `wood` 的所有行，将其按原始顺序导入到文件 `/root/select.txt` 中，文件 `/root/select.txt` 中不得包含空行

```
[root@servera ~]# grep "wood" /usr/share/dict/words > /root/select.txt
```

#校验

```
[root@servera ~]# cat select.txt
```

```
agalawood
```

```
agalwood
```

12、创建归档

创建一个名为 `/root/backup-YYYY-MM-DD.tar.bz2` 格式的 `tar` 包，用来压缩 `/var/log/` 目录。

i 需要同时掌握创建 xz(-J)、gzip (-z)、bzip2(-j) 格式的压缩包的能力

```
[root@servera ~]# yum install bzip2 -y
```

```
[root@servera ~]# tar -jcf /root/backup-YYYY-MM-DD.tar.bz2 /var/log/  
tar: Removing leading `/' from member names
```

```
[root@servera ~]# file backup-YYYY-MM-DD.tar.bz2
```

```
backup-YYYY-MM-DD.tar.bz2: bzip2 compressed data, block size = 900k
```

13、创建容器镜

使用 <http://content.example.com:17242/el9/Containerfile> 按照以下要求创建容器

- 使用用户 training 身份，创建一个名为 watcher 的容器镜像
- 不要对 Containerfile 进行任何更改

#以training用的身份登录

```
[training@yn2460 ~]$ ssh training@192.168.100.10  
training@192.168.100.10's password:
```

```
[training@servera ~]$ curl -o Containerfile \  
http://content.example.com:17242/el9/Containerfile
```

```
[training@servera ~]$ ls  
Containerfile
```

```
[training@servera ~]$ cat Containerfile  
FROM redhat/ubi9:9.2  
MAINTAINER nobody@redhat.com
```

```

RUN rm -f /etc/yum.repos.d/ubi.repo && curl -o
/etc/yum.repos.d/rhel_dvd.repo
http://content.example.com:17242/el9/rhel_dvd.repo && yum -y install
\
    enscript \
    ghostscript \
    http://content.example.com:17242/el9/cert_file_watcher.noarch.rpm

ENTRYPOINT ["/usr/local/bin/file_watcher.py"]

# 不要忘了 . , .表示在当前目录寻找Containerfile
[training@servera ~]$ podman build -t watcher .
#选择下载基础镜像的站点, 在联系环境里面选在docker.io 考试的时候不需要选择直接
执行
...
Successfully tagged localhost/watcher:latest
7a2e2c13b55e7caf2d3fc5cac4131fe772c180364dda37fdd4edeb96dfff42ef

#校验
[training@servera ~]$ podman images

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
localhost/watcher	latest	7a2e2c13b55e	5 seconds ago	356 MB
docker.io/redhat/ubi9	9.2	20cef057605e	5 months ago	217 MB

14、将容器配置为服务

在 servera 上创建一个 rootless 容器，并配置为systemd服务自动启动，要求如下

- 容器叫做 magic
- 容器使用在其他项目中创建的 watcher 容器镜像
- 该服务面向 training 以systemd服务运行
- 服务名称为 container-magic
- 系统重启后，容器无需干预自动运行

- 将本地目录/opt/files 附加到容器的 /opt/incoming 目录
- 本地目录的 /opt/processed 附加到容器的 /opt/outgoing 目录
- 如果服务正常，则放入/opt/files的任何纯文本文件将自动转换为pdf文件，并使用相同文件 名置入 /opt/outgoing

#创建挂在到容器的里面的目录

```
[training@servera ~]$ sudo mkdir -p /opt/{files,processed}
[sudo] password for training:

[training@servera ~]$ sudo chown training:training /opt/*
[training@servera ~]$ ls -l /opt/
total 0
drwxr-xr-x. 2 training training 6 Mar 19 09:26 files
drwxr-xr-x. 2 training training 6 Mar 19 09:26 processed
[training@servera ~]$ sudo chmod 777 training:training /opt/*
```

#创建systemd所需要的文件目录

```
[training@servera ~]$ cd
[training@servera ~]$ mkdir -p .config/systemd/user
[training@servera ~]$ cd .config/systemd/user/
```

#启动容器， :Z是设置selinux， 在考试的时候selinux是必须要打开的

```
[training@servera user]$ podman run -d --name magic -v
/opt/files:/opt/incoming:Z -v /opt/processed:/opt/outgoing:Z
localhost/watcher
eb3d6fcb6eed36b75771114daab0b90967e32b363cb4713f8672a02250c4f1a7
```

#查看容器

```
[training@servera user]$ podman ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
eb3d6fcb6eed	localhost/watcher:latest		5 seconds ago
Up 6 seconds ago		magic	

#停止容器。准备做systemd的配置

```
[training@servera user]$ podman stop magic
```

#生成systemd配置文件

```
[training@servera user]$ podman generate systemd --name magic --new --files /home/training/.config/systemd/user/container-magic.service
```

```
[training@servera user]$ ls  
container-magic.service
```

#启动服务和设置开机启动

```
[training@servera user]$ systemctl --user daemon-reload  
[training@servera user]$ systemctl --user enable --now container-magic.service  
Created symlink  
/home/training/.config/systemd/user/default.target.wants/container-magic.service → /home/training/.config/systemd/user/container-magic.service.
```

```
[training@servera user]$ systemctl --user status container-magic.service
```

```
● container-magic.service - Podman container-magic.service  
   Loaded: loaded (/home/training/.config/systemd/user/container-magic.servi>  
   Active: active (running) since Tue 2024-03-19 09:33:33 EDT; 7s ago  
     Docs: man:podman-generate-systemd(1)  
  Process: 6099 ExecStartPre=/bin/rm -f /run/user/1000/container-magic.servi>  
    Main PID: 6115 (conmon)  
       Tasks: 3 (limit: 11107)  
     Memory: 1.2M  
        CPU: 461ms  
    CGroup: /user.slice/user-1000.slice/user@1000.service/app.slice/container>  
            └─6112 /usr/bin/slrp4netns --disable-host-loopback --mtu=65520 ->  
                └─6115 /usr/bin/conmon --api-version 1 -c f3c27324c6a5442860551ee>
```

```
[training@servera user]$  
  
#测试  
[training@servera ~]$ cd /opt/files/  
[training@servera files]$ echo "1234456" > test2  
  
[training@servera files]$ cd ../processed/  
[training@servera processed]$ ls  
test2  
[training@servera processed]$ file test2  
test2: PDF document, version 1.4
```

15、添加 sudo 免密操作

允许 sysmgrs 成员 sudo 时不需要密码

```
[training@yn2460 ~]$ ssh root@192.168.100.10  
root@192.168.100.10's password:  
  
[root@servera ~]# visudo  
%wheel          ALL=(ALL)          ALL  
%sysmgrs        ALL=(ALL)          NOPASSWD: ALL  
  
#测试  
[root@servera ~]# su - natasha  
Last login: Tue Mar 19 07:58:21 EDT 2024 on pts/0  
  
[natasha@servera ~]$ id  
uid=1001(natasha) gid=1002(natasha)  
groups=1002(natasha),1001(sysmgrs)  
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
  
[natasha@servera ~]$ fdisk -l  
fdisk: cannot open /dev/vda: Permission denied
```



```
fdisk: cannot open /dev/sr0: Permission denied
fdisk: cannot open /dev/mapper/rhel-root: Permission denied
fdisk: cannot open /dev/mapper/rhel-swap: Permission denied
fdisk: cannot open /dev/mapper/exvg-rhel: Permission denied
```

#免密码

```
[natasha@servera ~]$ sudo fdisk -l
Disk /dev/vda: 25 GiB, 26843545600 bytes, 52428800 sectors
Units: sectors of 1 * 512 = 512 bytes
```

16、配置新用户的密码策略

创建新用户时，默认密码策略为 90 天后密码会过期

```
[root@servera ~]# vim /etc/login.defs
#在文件的131行 将9999 改为90
PASS_MAX_DAYS    90

#验证 创建一个用户，修改密码，查看shadow文件中的密码最长使用期限是不是90天
[root@servera ~]# useradd examtest
[root@servera ~]# passwd examtest
Changing password for user examtest.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.

[root@servera ~]# grep examtest /etc/shadow
examtest:$6$Uk4PWHgYKksAAdjz$C/lnTUGNr6DMfCCtKbtQ2/bJVT2mKFzZKOnIMmw
s576tnQfoYKcWI.SjMWx6pmVWJhjON6rkoP0W/Ezj00bkf1:19801:0:90:7:::
```

17、创建 shell 脚本

创建一个名为 simple 的脚本

- 该脚本放置在 `/usr/local/bin` 下

- 该脚本用于获取当前系统进程的快照，要求按照顺序输出进程的所有者，进程的PID，进程消耗的虚拟内存，实际内存，以及CPU的百分比，并其中以cpu的百分比进行排序，消耗CPU最多的进程在最后显示。

```
[root@servera ~]# vim /usr/local/bin/simple
[root@servera ~]# cat /usr/local/bin/simple
#!/bin/bash
ps -axo user,pid,vsz,rss,%cpu --sort=pcpu

[root@servera ~]# chmod a+x /usr/local/bin/simple

#测试
[root@servera ~]# simple
USER          PID     VSZ   RSS %CPU
root           2        0     0  0.0
...
root          1 173156 17504  0.1
root         4311        0     0  0.1
```

18、运行一个容器

在 <http://content.example.com:17242/dockerImg/rsyslog-custom-cert.tar> 中提供一个 rsyslog-custom-cert.tar 的镜像文件：

- 将镜像存放在 <https://registry.lab.example.com/library/> 下
- 使用 <https://registry.lab.example.com/library/rsyslog-custom-cert> 镜像运行容器 `syslogserver`
- 将生成的日志存放在 `/home/training/syslog/` 目录下，并使用logger命令发送 "This is my syslog container" 到日志中

```
[training@servera ~]$ curl -o rsyslog-custom-cert.tar \
http://content.example.com:17242/dockerImg/rsyslog-custom-cert.tar

[training@servera ~]$ ls rsyslog-custom-cert.tar
```

rsyslog-custom-cert.tar

```
[training@servera ~]$ podman load -i rsyslog-custom-cert.tar
```

#查看镜像是否存在

```
[training@servera ~]$ podman images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
localhost/rsyslog-custom-cert	latest	199c43916a5c	3 years ago	228 MB
...	...			

#为镜像打tag

```
[training@servera ~]$ podman tag localhost/rsyslog-custom-  
cert:latest \  
registry.lab.example.com/library/rsyslog-custom-cert
```

```
[training@servera ~]$ podman images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
localhost/rsyslog-custom-cert	latest	199c43916a5c	3 years ago	228 MB
registry.lab.example.com/library/rsyslog-custom-cert	latest	199c43916a5c	3 years ago	228 MB

#登录registry

```
[training@servera ~]$ podman login registry.lab.example.com  
Username: redhat  
Password:  
Login Succeeded!
```

#将镜像推到仓库

```
[training@servera ~]$ podman push  
registry.lab.example.com/library/rsyslog-custom-cert:latest  
Getting image source signatures  
Copying blob 2ac3801886b8 done  
Copying blob eb7bf34352ca done  
Copying blob a071269a675a done
```

```
Copying blob 92538e92de29 done
Copying config 199c43916a done
Writing manifest to image destination
Storing signatures
```

```
[training@servera ~]$ mkdir -p /home/training/syslog
```

#skopeo是一个容器的工具，可以远程检车容器镜像

```
[training@servera ~]$ sudo yum install -y skopeo
```

```
[training@servera ~]$ skopeo inspect \
docker://registry.lab.example.com/library/rsyslog-custom-cert
```

#启动容器，镜像本地不存在就会下载

```
[training@servera ~]$ podman run -d --privileged --name
syslogserver -v /home/training/syslog:/var/log:Z
registry.lab.example.com/library/rsyslog-custom-cert:latest
```

```
2a4977c5561bdeb00146930e8bcc846a7c6ec547ec2ecbc29603c2cb986518a4
```

#查看syslogserver容器是都正常

```
[training@servera ~]$ podman ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
NAMES					
45f9e70c4530	localhost/watcher:latest		4 minutes ago	Up 4 minutes ago	
magic					
2a4977c5561b	registry.lab.example.com/library/rsyslog-custom-cert:latest	/bin/rsyslog.sh	6 seconds ago	Up 6 seconds ago	
syslogserver					

#登录容器发送测试日志，在存储卷里面查看数据

```
[training@servera ~]$ podman exec -it syslogserver logger "rhcsa
test log"
```

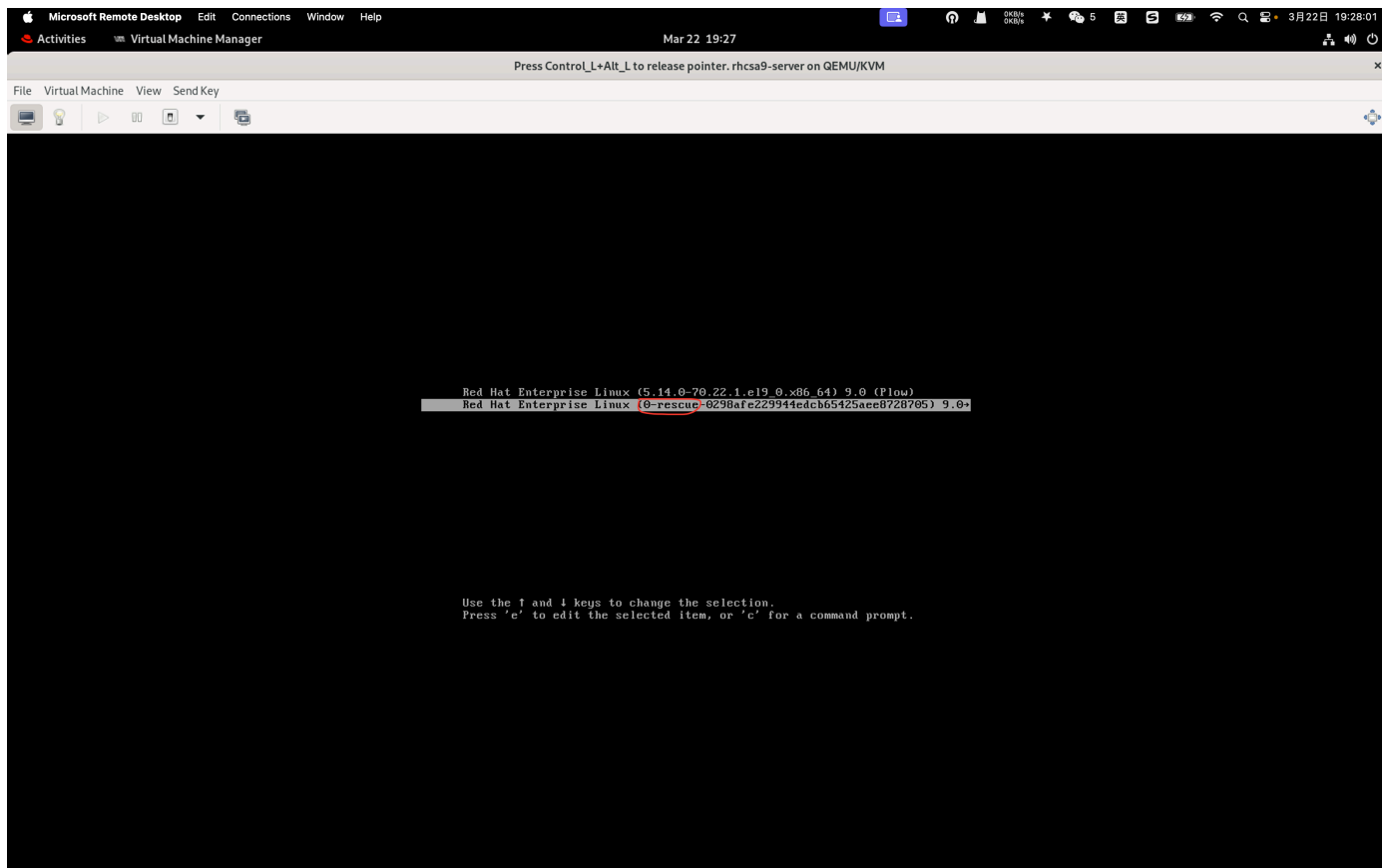
```
[training@servera ~]$ cd /home/training/syslog/
[training@servera syslog]$ ls
messages

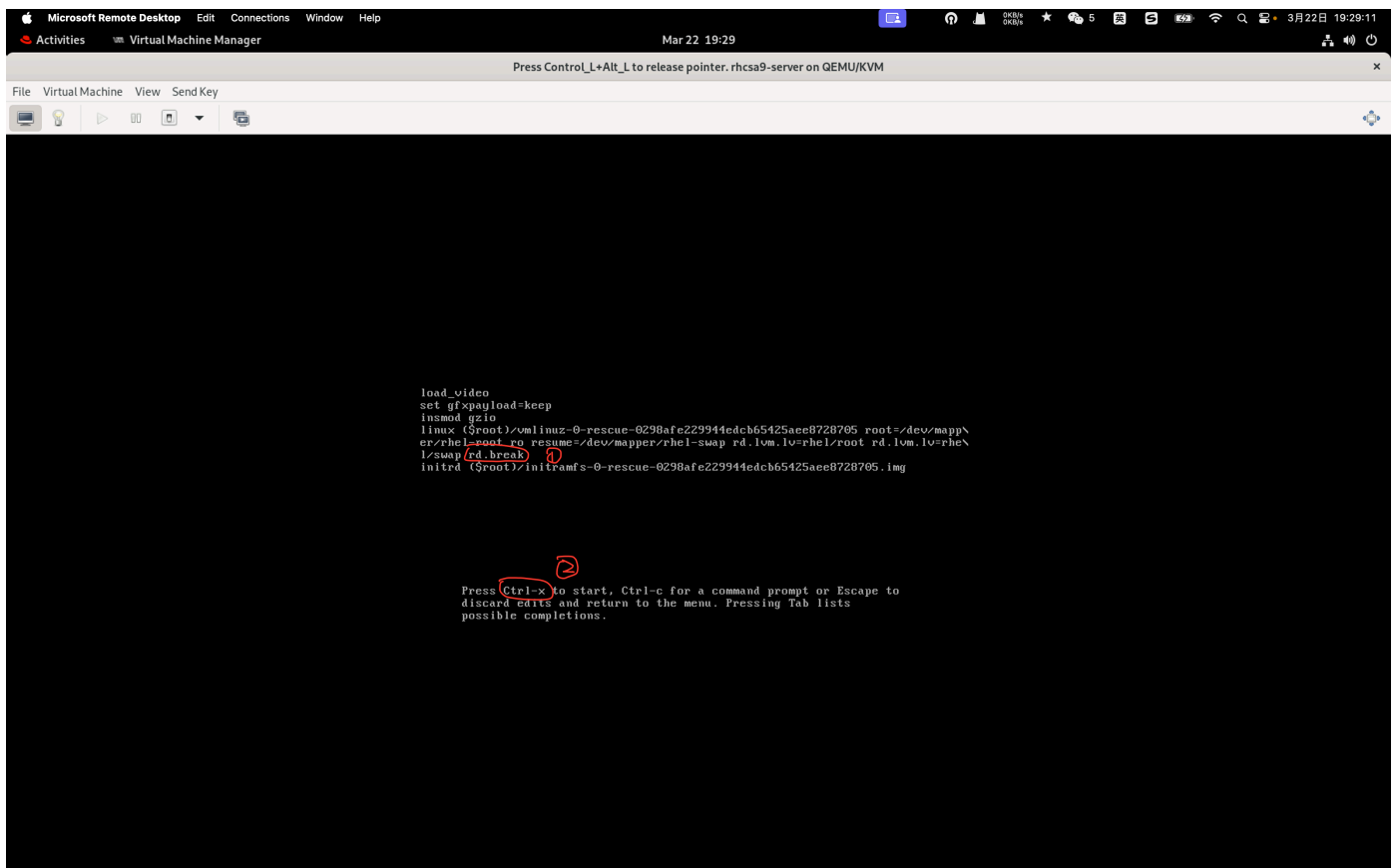
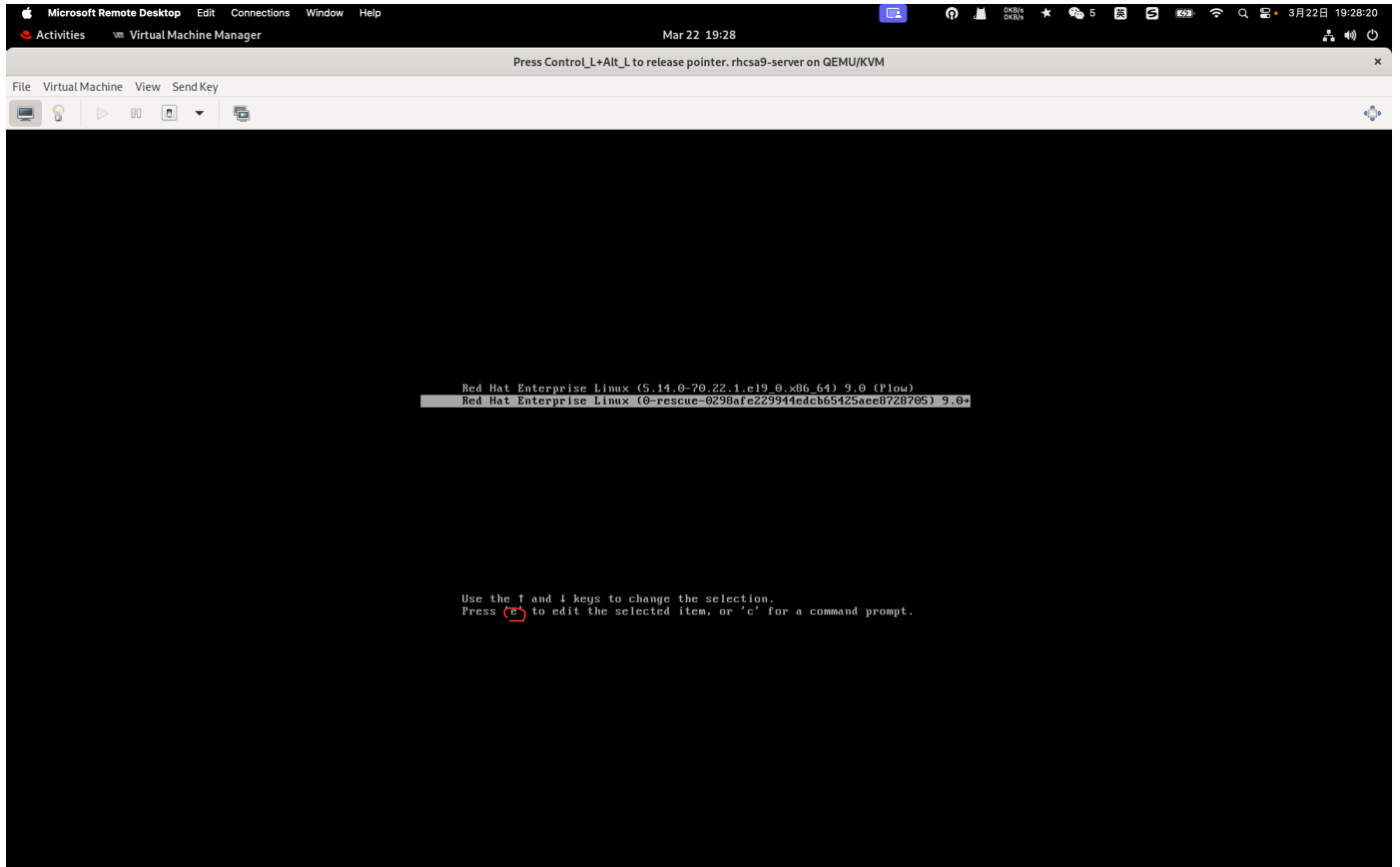
[training@servera syslog]$ tail messages
... ..
Mar 19 15:49:27 2a4977c5561b root: rhcsa test log
```

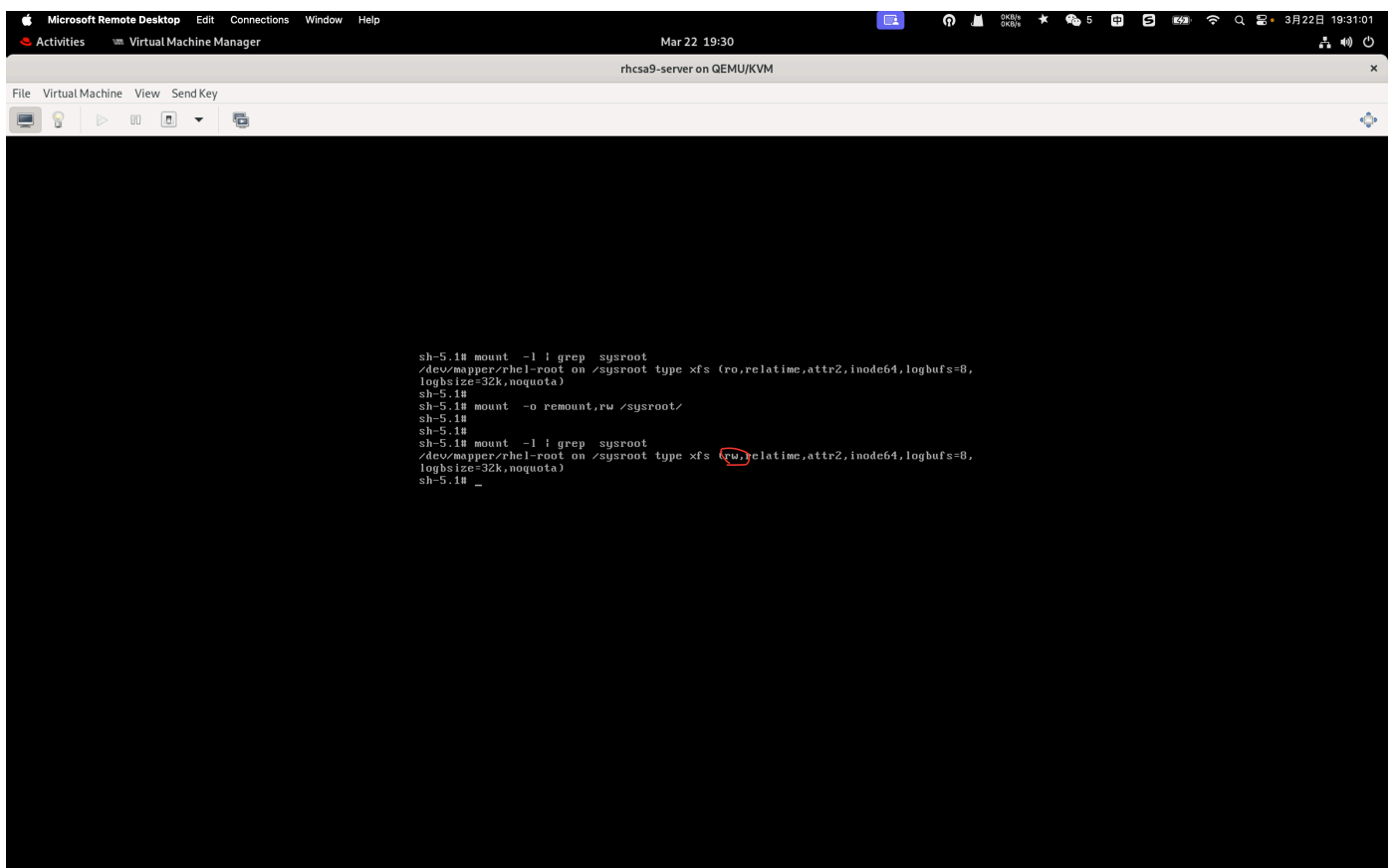
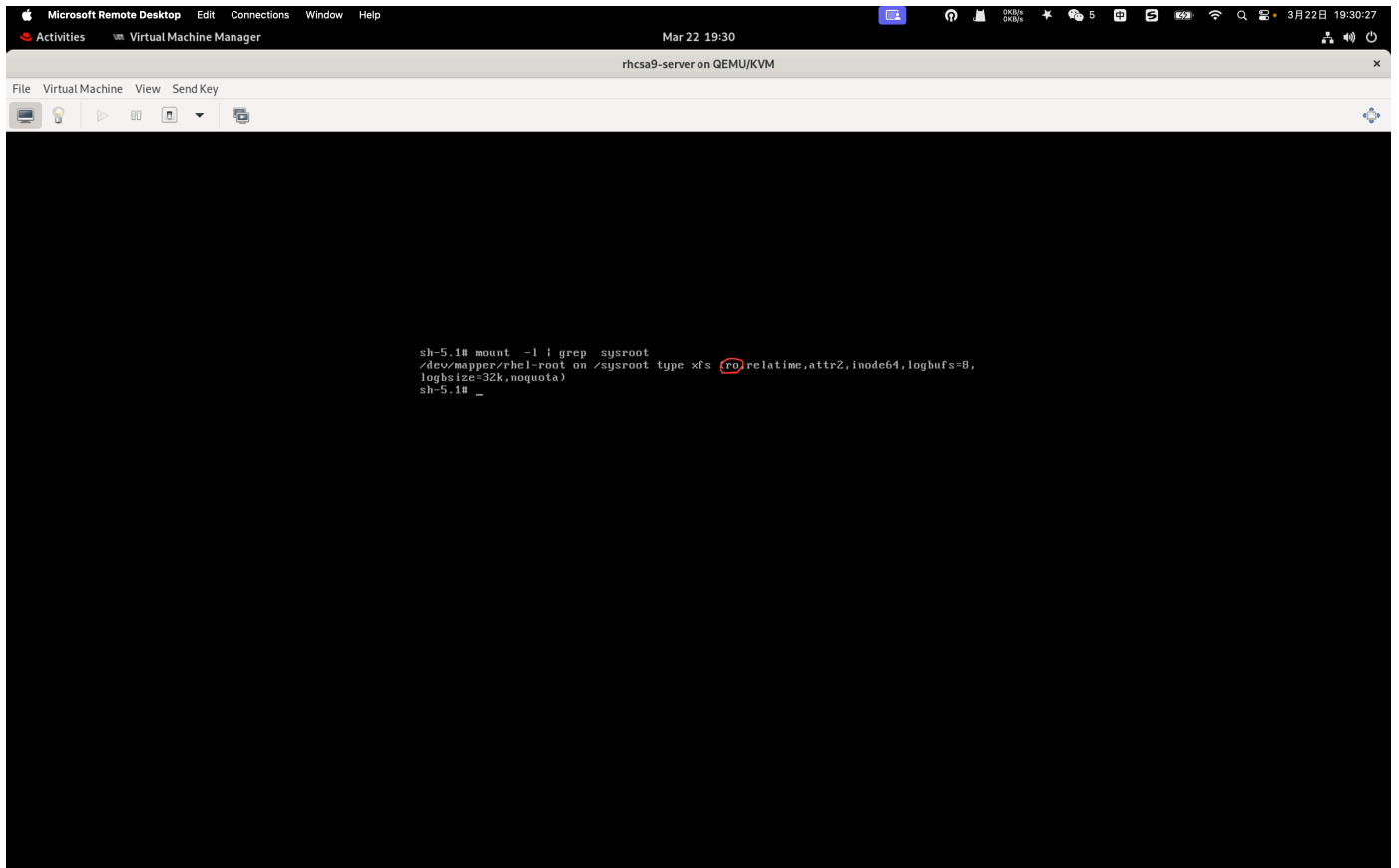
在 serverb.lab.example.com 上完成如下实验

1、设置 root 密码

将 serverb.lab.example.com 主机的密码设置成 redhat







```
Microsoft Remote Desktop  Edit  Connections  Window  Help
Activities  Virtual Machine Manager

Mar 22 19:32

rhcsa9-server on QEMU/KVM

File  Virtual Machine  View  Send Key

sh-5.1# mount -l | grep sysroot
/dev/mapper/rhel-root on /sysroot type xfs (ro,relatime,attr2,inode64,logbufs=8,logbsize=32k,noquota)
sh-5.1#
sh-5.1# mount -o remount,rw /sysroot/
sh-5.1#
sh-5.1# mount -l | grep sysroot
/dev/mapper/rhel-root on /sysroot type xfs (rw,relatime,attr2,inode64,logbufs=8,logbsize=32k,noquota)
sh-5.1#
sh-5.1# passwd
sh: passwd: command not found
sh-5.1# chroot /sysroot
sh-5.1#
sh-5.1# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
sh-5.1#
```

```
Microsoft Remote Desktop  Edit  Connections  Window  Help
Activities  Virtual Machine Manager

Mar 22 19:32

rhcsa9-server on QEMU/KVM

File  Virtual Machine  View  Send Key

sh-5.1# mount -l | grep sysroot
/dev/mapper/rhel-root on /sysroot type xfs (ro,relatime,attr2,inode64,logbufs=8,logbsize=32k,noquota)
sh-5.1#
sh-5.1# mount -o remount,rw /sysroot/
sh-5.1#
sh-5.1# mount -l | grep sysroot
/dev/mapper/rhel-root on /sysroot type xfs (rw,relatime,attr2,inode64,logbufs=8,logbsize=32k,noquota)
sh-5.1#
sh-5.1# passwd
sh: passwd: command not found
sh-5.1# chroot /sysroot
sh-5.1#
sh-5.1# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
sh-5.1# touch /.autorelabel
sh-5.1#
```


2、调整逻辑卷的大小

将名字为 rhel 的逻辑卷的大小调整到 256M，确保文件系统的内容保持不变。

调整后的逻辑卷的大小范围在 236 M 到 260M 的范围内都是可以接受的。

```
[root@servera ~]# lvs
  LV   VG   Attr          LSize   Pool Origin Data%  Meta%  Move Log
Cpy%Sync Convert
  rhel exvg -wi-ao---- 96.00m

  root rhel -wi-ao---- <17.00g

  swap rhel -wi-ao---- 2.00g
```

```
[root@servera ~]# lvextend -L 256M /dev/exvg/rhel
  Size of logical volume exvg/rhel changed from 96.00 MiB (24
extents) to 256.00 MiB (64 extents).
  Logical volume exvg/rhel successfully resized.
```

```
[root@servera ~]# lvs
  LV   VG   Attr          LSize   Pool Origin Data%  Meta%  Move Log
Cpy%Sync Convert
  rhel exvg -wi-ao---- 256.00m

  root rhel -wi-ao---- <17.00g

  swap rhel -wi-ao---- 2.00g
```

#确定文件系统，使用什么工具拉伸文件系统

```
[root@servera ~]# df -Th
Filesystem              Type      Size  Used Avail Use% Mounted on
... ..
/dev/mapper/exvg-rhel xfs        91M   5.7M   85M   7% /rhelmnt
tmpfs                   tmpfs     178M    0   178M   0% /run/user/0
```

#拉伸文件系统

```
[root@servera ~]# xfs_growfs /dev/mapper/exvg-rhel
```

```
[root@servera ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
...	...				
/dev/mapper/exvg-rhel	251M	7.3M	244M	3%	/rhelmnt

3、创建交换分区

向 serverb.lab.example.com 添加一个额外的交换分区 768 MiB

- 交换分区应在系统启动时自动挂载
- 不要删除或以任何方式改动系统上的任何现有交换分区

```
[root@servera ~]# fdisk -l
```

... ..

#系统里面已经存在了三个分区，并且都是主分区

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/vda1	*	2048	2099199	2097152	1G	83	Linux
/dev/vda2		2099200	41943039	39843840	19G	8e	Linux LVM
/dev/vda3		41943040	42147839	204800	100M	83	Linux

#先创建一个扩展分区，然后创建两个逻辑分区 (3P + 1E)

```
[root@servera ~]# fdisk /dev/vda
```

#常见扩展分区，使用所有的空间

Command (m for help): n

Partition type

p primary (3 primary, 0 extended, 1 free)

e extended (container for logical partitions)

Select (default e): e

Selected partition 4

First sector (42147840-52428799, default 42147840):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (42147840-52428799, default 52428799):

Created a new partition 4 of type 'Extended' and of size 4.9 GiB.

#常见逻辑分区 大小800M (> 768M)

Command (m for help): n

All primary partitions are in use.

Adding logical partition 5

First sector (42149888-52428799, default 42149888):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (42149888-52428799, default 52428799): +800M

Created a new partition 5 of type 'Linux' and of size 800 MiB.

#创建逻辑分区, 使用全部的空间

Command (m for help): n

All primary partitions are in use.

Adding logical partition 6

First sector (43790336-52428799, default 43790336):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (43790336-52428799, default 52428799):

Created a new partition 6 of type 'Linux' and of size 4.1 GiB.

Command (m for help): w

The partition table has been altered.

Syncing disks.

[root@servera ~]# fdisk -l

Disk /dev/vda: 25 GiB, 26843545600 bytes, 52428800 sectors

Units: sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: dos

Disk identifier: 0xbfbaefc1

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/vda1	*	2048	2099199	2097152	1G	83	Linux
/dev/vda2		2099200	41943039	39843840	19G	8e	Linux LVM

/dev/vda3	41943040	42147839	204800	100M	83	Linux
/dev/vda4	42147840	52428799	10280960	4.9G	5	Extended
/dev/vda5	42149888	43788287	1638400	800M	83	Linux
/dev/vda6	43790336	52428799	8638464	4.1G	83	Linux

```
[root@servera ~]# pvcreate /dev/vda5
```

```
Physical volume "/dev/vda5" successfully created.
```

```
Not creating system devices file due to existing VGs.
```

```
[root@servera ~]# vgcreate -s 1M swapvg /dev/vda5
```

```
Not creating system devices file due to existing VGs.
```

```
Volume group "swapvg" successfully created
```

```
[root@servera ~]# lvcreate -l 768 -n swaplv swapvg
```

```
Logical volume "swaplv" created.
```

```
[root@servera ~]# lvs
```

LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move
Log Cpy%Sync Convert								
rhel	exvg	-wi-ao----	96.00m					
root	rhel	-wi-ao----	<17.00g					
swap	rhel	-wi-ao----	2.00g					
swaplv	swapvg	-wi-a-----	768.00m					

```
[root@servera ~]# mkswap /dev/swapvg/swaplv
```

```
Setting up swapspace version 1, size = 768 MiB (805302272 bytes)
no label, UUID=ff356978-c6e0-4eb2-b41d-fd8667f97fdb
```

```
[root@servera ~]# vi /etc/fstab
```

```
[root@servera ~]# cat /etc/fstab
```

```
... ..
```

```
/dev/swapvg/swaplv none swap defaults 0 0
```

```
[root@servera ~]# mount -a
```

```
[root@servera ~]# swapoff -a
```

```
[root@servera ~]# swapon -a
[root@servera ~]# swapon -s
```

Filename	Type	Size	Used	Priority
/dev/dm-1				partition 2097148 0 -2
/dev/dm-3				partition 786428 0 -3

```
[root@servera ~]# free -m
```

	total	used	free	shared	buff/cache
Mem:	1774	170	1439	13	164
available	1450				
Swap:	2815	0	2815		

4、创建逻辑卷

根据如下要求，创建新的逻辑卷：

- 逻辑卷的名字 myvol，卷组是 share，大小是 60 个 PE size
- share 的 PE size 是 32 MiB
- 格式化成 fat32 文件系统。并在系统启动时自动挂载到/mnt/volume

#该包里面包含了mkfs.vfat命令，如果没有该命令就安装dosfstools

```
[root@servera yum.repos.d]# yum install dosfstools -y
```

```
[root@servera ~]# pvcreate /dev/vda6
```

Physical volume `"/dev/vda6"` successfully created.

Not creating system devices file due to existing VGs.

```
[root@servera ~]# vgcreate -s 32M share /dev/vda6
```

Not creating system devices file due to existing VGs.

Volume group `"share"` successfully created

```
[root@servera ~]# lvcreate -l 60 -n myvol share
```

Logical volume `"myvol"` created.

```
[root@servera ~]# lvs
```

```

LV      VG      Attr      LSize   Pool Origin Data%  Meta%  Move
Log Cpy%Sync Convert
...    ...
myvol   share  -wi-a----- <1.88g

[root@servera yum.repos.d]# mkfs.vfat /dev/share/myvol
mkfs.fat 4.2 (2021-01-31)

[root@servera yum.repos.d]# mkdir /mnt/volume

[root@servera ~]# vim /etc/fstab
[root@servera ~]# cat /etc/fstab
...    ...
/dev/share/myvol      /mnt/volume  vfat  defaults 0 0

[root@servera yum.repos.d]# mount -a

[root@servera yum.repos.d]# df -Th /dev/share/myvol
Filesystem              Type  Size  Used Avail Use% Mounted on
/dev/mapper/share-myvol vfat  1.9G  4.0K  1.9G   1% /mnt/volume

```

5、给系统配置默认存储库

YUM 的两个存储库的地址分别是：

- <http://content.example.com:17242/rhel9.0/dvd/AppStream/>
- <http://content.example.com:17242/rhel9.0/dvd/BaseOS/>

```

[training@yn2460 ~]$ ssh root@192.168.100.10
root@192.168.100.10's password:

[root@centos7-vm01 ~]# cd /etc/yum.repos.d/

```

```
[root@centos7-vm01 yum.repos.d]# vi anyname.repo
[root@centos7-vm01 yum.repos.d]# cat anyname.repo
[BaseOS]
name=baseos
baseurl=http://content.example.com:17242/rhel9.0/dvd/BaseOS/
gpgcheck=0

#复制4行 修改baseurl
[AppStream]
name=appstream
baseurl=http://content.example.com:17242/rhel9.0/dvd/AppStream/
gpgcheck=0

#安装vim编辑器
[root@servera yum.repos.d]# yum install vim -y
```

6、配置系统调优

为您的系统选择建议的 tuned 配置集并将它设为推荐值。

```
[root@servera ~]# yum install tuned -y

[root@servera ~]# systemctl is-active tuned
inactive
[root@servera ~]# systemctl enable --now tuned
[root@servera ~]# systemctl is-active tuned
active

[root@servera ~]# systemctl is-enabled tuned
enabled

#tuned根据系统环境 推荐调优方案
[root@servera ~]# tuned-adm recommend
virtual-guest

#应用调优方案virtual-guest
```

```
[root@servera ~]# tuned-adm profile virtual-guest
```

#查看当前生效的调优方案

```
[root@servera ~]# tuned-adm active
```

```
Current active profile: virtual-guest
```

#重启一下系统，观察系统是否正常

```
[root@servera ~]# reboot
```

#重启之后检查各个题目是否正常