

RHCSA RHEL 9 Exam Answers

| | |
|--|----------|
| | must |
| | Be aware |
| | optional |

1. Set hostname and Assign ip address

```
# hostname
# hostnamectl set-hostname node1.lab.network17.com
# hostname
# nmcli con sh
# nmcli con mod "Wired connection 1" ipv4.addresses 172.25.250.11/24 ipv4.gateway
172.25.250.254 ipv4.dns 172.25.250.254 ipv4.method manual connection.autoconnect
yes
# nmcli con up "Wired connection 1"
```

To verify:

```
# nmcli con sh "Wired connection 1"
# ping 172.25.250.11
# ping 172.25.250.254
```

```
# vim /etc/ssh/sshd_config
Goto the line #PermitRootLogin prohibit-password
remove # and change it to PermitRootLogin yes
:wq!
```

minimize or close the node1 virtual machine console

Take ssh from base machine to node1 VM

```
#ssh root@node1
```

2. Create a repository

```
# vim /etc/yum.repos.d/app.repo
```

```
[AppStream]
name=app
baseurl=http://classroom.example.com/content/rhel8.0/x86_64/dvd/AppStream
gpgcheck=0
enabled=1
[BaseOS]
name=base
```

```
baseurl=http://classroom.example.com/content/rhel8.0/x86_64/dvd/BaseOS
gpgcheck=0
enabled=1
:wq
# yum clean all
```

To verify:
yum repolist all

3. Configure webserver.

```
# semanage port -l |grep http
# semanage port -a -t http_port_t -p tcp 82
# semanage port -l |grep http (verify the port)
# firewall-cmd --list-ports
# firewall-cmd --permanent --add-port=82/tcp
# firewall-cmd --reload
# firewall-cmd --list-ports (verify the port)
```

```
# vim /etc/httpd/conf/httpd.conf (search the "/Listen") and verify the port
Listen 82
<virtualhost 172.25.250.11:82>
servername serverb.lab.example.com
documentroot /var/www/html
</virtualhost>
:wq -> just for verification. You don't edit
```

```
# httpd -t
# systemctl restart httpd.service
```

To verify:
curl http://172.25.250.11:82
(or)
http://node1.lab.example.com:82
(Works only in exam and mock test)

4. Users and groups management

```
# groupadd admin
# useradd -G admin harry
# useradd -G admin natasha
# useradd -s /sbin/nologin sarah
# passwd --stdin harry (or) echo "123" | passwd --stdin harry
```

```
# passwd --stdin natasha      (or)   echo "123" | passwd --stdin natasha
# passwd --stdin sarah       (or)   echo "123" | passwd --stdin sarah
```

To verify:

```
# id harry
# id natahsha
# su - sarah (the account should not available)
# tail -n 3 /etc/passwd (verify the users info)
# tail -n 4 /etc/group (verify the group info)
```

5.create a directory and assign the basic permissions.

```
# mkdir -p /common/admin
# ls -ld /common/admin
# chgrp admin /common/admin
# chmod 770 /common/admin
# chmod g+s /common/admin      (or by one step)      chmod 2770 /common/admin
```

To verify:

```
# ls -ld /common/admin
# su - harry
# touch /common/admin/file1
# ls -l /common/admin/file1
```

6. Configure the auto filesystem

```
# yum install autofs -y
# systemctl start autofs.service
# systemctl enable autofs.service
# getent passwd production5 (verify the centralized user)
# showmount -e 172.25.250.10 (this will show full path to mount)
# vim /etc/auto.master.d/name.autofs
/localhome /etc/auto.misc
:wq
add last line (don't edit the existing contents)
# vim /etc/auto.misc
add last line
production5 -rw,sync,fstype=nfs4 ip:/localhome/production5
:wq
# systemctl restart autofs
```

To verify:

```
# su - production5
# pwd
# df -hT
# exit
```

(logout from remote user)

7. Configure crontab

```
# crontab -eu harry
30 12 * * * /bin/echo "hello" (note: at 12:30 pm) or
*/3 * * * * /bin/echo "hello" (note: This is to print message hello for every 3 minutes)
*/2 * * * * logger user.debug "hello" (note: This is to print log message hello for every
2 minutes)
```

To verify:

```
# crontab -lu harry
```

8.NTP configuration

```
# vim /etc/chrony.conf
put '#' for existing iburst argument line (or)
put '#' for existing server or pool line (In exam)
add a new line
server classroom.example.com iburst
:wq
# systemctl restart chronyd.service
# timedatectl
# chronyc sources -v
```

9. find the owner of the file

```
# mkdir /root/find.user
# find / -user sarah -type f
# find / -user sarah -type f -exec cp -prv {} /root/find.user \; 2>/dev/null
# ls -a /root/find.user
```

10. grep the "home" string from /etc/passwd

```
# touch /root/search.txt

# grep "home" /etc/passwd
# grep "home" /etc/passwd > /root/search.txt
# cat /root/search.txt
```

11. Create a user with UID and set given password

```
# useradd -u 1326 alies
# passwd --stdin alies (or) echo "123" | passwd --stdin alies
```

12.backup the /var/tmp file

Note : yum install -y gzip

```
# tar zcvf /root/test.tar.gz /var/tmp (its gzip compression tool if they ask for .gz extension)
```

Note : yum install -y bzip2

```
# tar jcvf /root/test.tar.bz2 /var/tmp (its bzip2 compression tool if they ask for .bz2 extension)
```

To verify:

```
#ls
```

13. Container Question 1

Open a new Tab in the terminal

```
#ssh walhalla@node1
$ wget <Use the given url to download the ContainerFile>
$ ls
$ podman login <given registry link>
un :admin
pw: redhat321
$ podman build -t monitor .
```

To verify:

```
$ podman images
```

14. Container Question 2 (needn't to verify 😎 follow these steps)

Go back to Tab 1

```
# mkdir /opt/files /opt/processed
# chmod 777 /opt/files /opt/processed
# chown walhalla: walhalla /opt/files /opt/processed
```

Switch back to Tab 2

```
$ podman run -d --name ascii2pdf -v /opt/files :/opt/incoming:Z -v /opt/processed:/opt/outcoming:Z localhost/monitor:latest
$ podman ps
$ mkdir -p ~/.config/systemd/user
$ cd ~/.config/systemd/user
$ podman generate systemd --name ascii2pdf --files --new
$ podman ps
$ systemctl --user daemon-reload
$ systemctl --user start container-ascii2pdf.service
$ systemctl --user enable container-ascii2pdf.service
$ podman ps
```

Switch to Tab 1

```
#loginctl enable-linger walhalla
# reboot
```

16.1 Set the Permission

```
#su – natasha
#ls -a (you can find out the .bash_profile file)
#vim .bash_profile
add last line (don't disturb the existing contents)
umask 277 (enter the calculated value)
:wq!
#source .bash_profile
```

```
#mkdir test
#ls -ld test (To verify the permission)
#touch testfile
#ls -l testfile (To verify the permission)
```

16.2 Set the Password expire date

```
#vim /etc/login.defs
(find the PASS_MAX_DAYS line and change the value )
PASS_MAX_DAYS 99999 → PASS_MAX_DAYS 20
:wq!
```

16.3 Assign Sudo Privilege

```
#vim /etc/sudoers
:se nu (then go to line 111 the enter)
%admin ALL=(ALL) NOPASSWD: ALL
```

16.4 Configure the application

```
# su – alies
# ls -a
# vim .bash_profile (Go to the last line and type)
RHCSA="Welcome to Advantage Pro"
export RHCSA
echo $RHCSA
:wq!

# source .bash_profile
# exit
```

To verify:

```
# su – alies (verify the message was showing)
```

16.5 Create the script file

```
#mkdir /root/myfiles
# touch /usr/local/bin/mysearch

#vim /usr/local/bin/mysearch
#!/bin/sh
find /usr/share/ -type f -size -1M -exec cp -prvf {} /root/myfiles/ \;
(or)
find /usr/share/ -type f -size -1M -perm /4000 -exec cp -prvf {} /root/myfiles/ \; (For u+s)
(or)
find /usr/share/ -type f -size +800M -size -900M -perm /2000 -exec cp {} /root/myfiles/ \;
(For g+s)
:wq! (save the file)

#chmod a+x /usr/local/bin/mysearch
# mysearch
```

To verify:

```
# ls -a /root/myfiles/
```

After you Complete all the question in VM-1 you need to reboot and wait to system bootup then verify all outputs.

Node 2 Answers

1. Break the password on server-a VM

- ❖ open console and reboot machine
- ❖ select the line which says rescue in the grub loader
- ❖ Press "e" and Move the cursor to the kernel command line which starts with "linux" then press "End" key in the keyboard).
- ❖ Add rw init=/bin/bash at end of the line and Press Ctrl+x to boot with the changes.

```
#echo "123" | passwd --stdin root
# touch /.autorelabel
# exec /sbin/init
Login with the root user password
# vim /etc/ssh/sshd_config
Goto the line #PermitRootLogin prohibit-password
remove # and change it to PermitRootLogin yes
:wq!
minimize or close the node1 virtual machine console
```

Take ssh from base machine to node2 VM

```
#ssh root@node2
```

2 configure repository

```
# vi /etc/yum.repos.d/app.repo
```

```
[AppStream]
name=app
baseurl=http://classroom.example.com/content/rhel8.0/x86_64/dvd/AppStream
enabled=1
gpgcheck=0
[BaseOS]
name=base
baseurl=http://classroom.example.com/content/rhel8.0/x86_64/dvd/BaseOS
enabled=1
gpgcheck=0
:wq
```

To verify:

```
# yum clean all
# yum repolist
# yum install vim -y (only for exam not now)
```

3. Create a swap partition

```
# free -m      (or)      swapon -s
#lsblk
# fdisk /dev/vdb
-> Command (m for help): n
-> Select (default p): p
-> Partition number (3-4, default 3): 3
-> First sector (4196352-10485759, default 4196352):
-> Last sector, +sectors or +size{K,M,G,T,P} (4196352-10485759, default 10485759): +512M
-> Created a new partition 2 of type 'Linux' and of size 512 MiB.
-> Command (m for help): t
-> Partition number (1,2,3 default 3): 3
-> Hex code (type L to list all codes): swap
-> Changed type of partition 'Linux' to 'Linux swap / Solaris'.
-> Command (m for help): w
```

```
# udevadm settle
#lsblk
# mkswap /dev/vdb3
# vim /etc/fstab
add last line (don't change the existing contents)
UUID= swap swap defaults 0 0
:wq
# swapon -a
```

To verify:

```
# free -m      (or)      swapon -s
```

```
#lsblk
```

4. Create a lvm partition

```
# lsblk
```

```
# fdisk /dev/vdb
```

```
-> Command (m for help): n
```

```
-> Select (default p): p
```

```
-> Partition number (2-4, default 2): 2
```

```
-> First sector (2048-10485759, default 2048):
```

```
-> Last sector, +sectors or +size{K,M,G,T,P} (2048-10485759, default 10485759): +2G
```

```
-> Created a new partition 1 of type 'Linux' and of size 2 GiB.
```

```
-> Command (m for help): t
```

```
-> Selected partition :2
```

```
-> Hex code (type L to list all codes): lvm
```

```
-> Changed type of partition 'Linux' to 'Linux LVM'
```

```
-> Command (m for help): w
```

```
# udevadm settle
```

```
#lsblk
```

```
# pvcreate /dev/vdb2
```

```
# vgcreate -s 8M datastore /dev/vdb2
```

```
# vgdisplay datastore
```

```
# lvcreate -l 50 -n database datastore
```

```
# lvdisplay /dev/datastore/database
```

```
# mkfs -t ext3 /dev/datastore/database
```

```
# mkdir /mnt/database
```

```
#lsblk -fs
```

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

```
add last line (don't disturb the existing contents)
```

```
UUID=          /mnt/database      ext3  defaults      0      0
```

```
:wq
```

```
# mount -a
```

To verify:

```
# df -hT or lsblk (verify the lvm partition)
```

5. Resize the lvm partition.

```
# lsblk
```

```
# df -hT
```

```
# lvresize -l 100 -r /dev/mapper/datastore-database
```

```
(Depends on question) (or)
```

```
# lvresize -L 100M -r /dev/mapper/datastore-database
```

To verify:

```
# lsblk
```

```
# df -hT
```

7.set the tuned profile

```
# systemctl status tuned.service  
# tuned-adm active (check the current active profile)  
# tuned-adm recommend  
  virtual-guest -> (This one is recommended)  
# tuned-adm profile virtual-guest  
# tuned-adm active
```

After you Complete all the question in Node 2 you need to reboot and wait to system bootup then verify all outputs.