

# RHCSA EXAM QUESTIONS

## Lab setup:

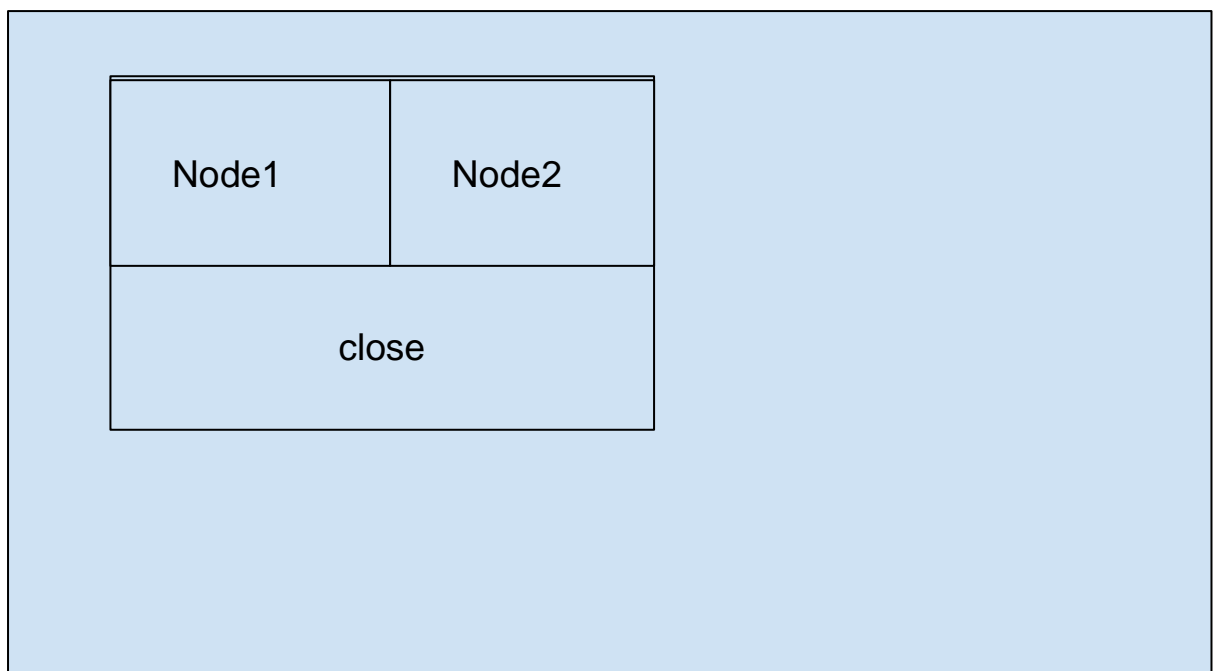
Activities(click on activities below options will be displayed with their respective icons)

Exam view

VM moniter

Terminal

- If we click on exam view exam question paper will be displayed
- If we click on vm monitor



- If we click on Node1

Start node1

Open node1
Close node1
Node1 console
Delete node1

- We need to click on **node1 console** (console of node1 will be opened)
- Similarly for node2

\*\*\*\*\*Note: Network question need to do in console and after that we need to do reboot\*\*\*\*\*

## 1. Configure the network.

**Assign hostname and ip address 1for your virtual machine.**

**Hostname - serverb.lab.example.com**

**Ip address - 172.25.250.11**

**Netmask - 255.255.255.0**

**Gateway - 172.25.250.254**

**NameServer - 172.25.250.254**

#nmcli connection (Note ethernet type)

```
#nmcli connection modify "Wired connection 1" ipv4.addresses
172.25.250.11/24 ipv4.gateway 172.25.250.254 ipv4.dns
172.25.250.254 ipv4.method static
```

#nmcli connection up "Wired connection 1"

#ping 172.25.250.11 (To check ping with ip)

#ping 172.25.250.254 (To check ping with gateway)

#ping 172.25.250.254 (To check ping with DNS)

#reboot

## 2. Create a repository

[http://classroom.example.com/rhel8.0/x86\\_64/devd/AppStream](http://classroom.example.com/rhel8.0/x86_64/devd/AppStream)

[http://classroom.example.com/rhel8.0/x86\\_64/devd/BaseOS](http://classroom.example.com/rhel8.0/x86_64/devd/BaseOS)

In console of node1 (we need to do this in console because there is no ip or hostname assigned to take remote connectivity)

#vi /etc/yum.repos.d/local.repo

[1 (any name, should be in square braces)]

name=AppStream (any name)

baseurl:[http://classroom.example.com/rhel8.0/x86\\_64/devd/AppStream](http://classroom.example.com/rhel8.0/x86_64/devd/AppStream)

enabled=1

gpgcheck=0

[2 (any name, should be in square braces)]

name=BaseOS (any name)

baseurl:[http://classroom.example.com/rhel8.0/x86\\_64/devd/AppStream](http://classroom.example.com/rhel8.0/x86_64/devd/AppStream)

enabled=1

gpgcheck=0

esc:wq

#yum clean all (it clears all cache)

#yum repolist all (it lists the repositories)

#yum install httpd -y (if it won't install the issue may be in repo file or setting ip address or dns or gateway)

## 3. Configure the Selinux

(a) Your webcontent has been configured in port 82 at the /var/www/html directory (Don't alter or remove any files in this directory) Make the content accessible.

```
#semanage port -l | grep http (check whether port 82 is enabled or if not use below command to add)
```

```
#semanage port -a -t http_port_t -p tcp 82 (-a=add, -t= type, -p=protocol)
```

```
#semanage port -l | grep http (verify post 82 is added or not)
```

```
#firewall-cmd --permanent --add-port=82/tcp
```

```
#firewall-cmd --reload
```

```
#firewall-cmd --list-all (check port 82 is added or not)
```

```
#yum install httpd -y
```

```
#systemctl start httpd
```

```
#systemctl enable httpd
```

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

(Shortcut to go to last line of the file **shift+g**)

```
<virtualhost 172.25.250.11:82>
```

```
servername servera.lab.example.com
```

```
documentroot /var/www/html
```

```
</virtualhost>
```

Esc:wq

```
#httpd -t (it should show syntax ok)
```

```
#systemctl restart httpd
```

```
#curl http://servera.lab.example.com:82
```

```
#curl http://servera.lab.example.com:82/file1
```

```
#curl http://servera.lab.example.com:82/file2
```

## **Extra work**

Normally you will get error. If you want to get actual output in curl command

Go to file

```
#vi /etc/httpd/conf/httpd.conf
```

```
37 # Listen.  Allows you to bind Apache to specific IP addresses and/or  
38 # ports, instead of the default.  See also the <VirtualHost>  
39 # directive.  
40 #  
41 # Change this to Listen on specific IP addresses as shown below to  
42 # prevent Apache from glomming onto all bound IP addresses.  
43 #  
44 #Listen 12.34.56.78:80  
45 Listen 82  
46  
47 #  
48 # Dynamic Shared Object (DSO) Support  
49 #  
50 # To be able to use the functionality of a module which was built as a DSO you
```

And goto line no 45

Edit Listen value to 82 (**Listen 82**)

Make sure your vm is pinging to google.com(#ping google.com)

#### **4. Create the following users, groups and group memberships:**

**(a) A group named admin.**

**(b) A user harry who belongs to admin as a secondary group.**

**(c) A user natasha who belongs to admin as a secondary group.**

**(d) A user sarah who does not have access to an interactive shell on the system and who is not a member of admin.**

**(e) The users harry, natasha, sarah should all have password of password.**

```
#groupadd admin
```

```
#useradd -G admin harry (-G= secondary group, -g=primary group)
```

```
#useradd -G admin natasha
```

```
#useradd -s /sbin/nologin sarah (-s=shell)
```

```
#passwd --stdin harry
```

```
#passwd --stdin natasha
```

```
#passwd --stdin sarah
```

Note: --stdin is not mandatory, if we use it no need to retype password and also it shows the password you typed.

```
[root@serverb ~]# passwd --stdin harry
Changing password for user harry.
harry123
passwd: all authentication tokens updated successfully.
[root@serverb ~]# passwd harry
Changing password for user harry.
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serverb ~]#
```

**5. Create a collaborative directory /common/admin with the following characteristics:**

**(a) Group ownership of /common/admin is admin.**

**(b) The directory should be readable, writable and accessible to members of admin, but not any other user.**

**(It is understood that root has access to all files and directories on the system.)**

**(c) Files created in /common/admin automatically have group ownership set to the admin group.**

```
#mkdir -p /common/admin      (-p=parent directory)
#chgrp admin /common/admin
#chmod 770 /common/admin
#chmod g+s /common/admin
#ls -ld /common/admin
#su - harry
$touch /common/admin/file1
$ls -ltr /common/admin/file1 (you need to get group ownership as admin)
$logout
```

**6. Configure autofs to automount the home directories of production5 domain users. Note the following:**

**(a) [servera.lab.example.com](http://servera.lab.example.com)([172.25.250.10](http://172.25.250.10)) NFS -exports /home-directories to your system.**

**(b) production5 home directory is [servera.lab.example.com](http://servera.lab.example.com):/home-directories/production5**

**(c) production5 home directory should be automounted locally mapped to localhome in your system.**

**(d)home directories must be writable by their users.**

```
#yum install autofs -y
#systemctl start autofs
#systemctl enable autofs
#getent passwd production5
#showmount -e servera.lab.example.com (it should show exported files)
#vim /etc/auto.master
    (To go to last line Shift+g)
    /localhome /etc/auto.misc
Esc:wq
#vi /etc/auto.misc
(To go to last line Shift+g)
Production5 -rw,soft,intr servera.lab.example.com:/home-directories/production5
Esc:wq
#systemctl restart autofs
#su - production5
$pwd (o/p: /localhome/production5 it should match with home directory of production5)
production5:x:1005:1006:./localhome/production5:/bin/bash
$df -hT (you should get mountpoint)
$pwd
$logout
```

**7. Set a Cron job for harry on 12.30 at noon print /bin/echo on "hello".**

```
#crontab -eu harry
    30 12 * * * /bin/echo "hello"
#crontab -lu harry (it should show crontabs of that user)
(-l=list, -u=user, -e=edit)
```

**8. Configure Acl permission**

**copy the file /etc/fstab to /var/tmp. Configure the permission of /var/tmp/fstab so that:**

**(a)The file /var/tmp/fstab is owned by root user**

**(b)The file /var/tmp/fstab belongs to the group root.**

**(c)The file /var/tmp/fstab should not be executable by anyone.**

**(d)The user harry is able to read and write by /var/tmp/fstab.**

**(e)The user natasha can neither read nor write /var/tmp/fstab.**

**(f)All other users (current/future) have the ability to read /var/tmp/fstab**

```
#cp /etc/fstab /var/tmp
#setfacl -m u:harry:rw /var/tmp/fstab
#setfacl -m u:natasha:--- /var/tmp/fstab
#getfacl /var/tmp/fstab
```

## **9.Configure the NTP**

**a) Configure your system so that it is an NTP client of [classroom.example.com](http://classroom.example.com).**

```
#vim /etc/chrony.conf
    Server classrom.lab.example.com iburst
Esc:wq
#systemctl restart chronyd.service
#chronyc sources
```

## **10. Locate the Files**

**(a) Find the owner of the file sarih and copy the file to given path of /root/find.user**

```
#mkdir /root/find.user
#find / -user sarah -type f
#find / -user sarah -type f -exec cp -pr {} /root/find.user \;
#ls -a /root/find.user
```

## **12. Create a user account**

**a) Create a new user with UID 1326 and user name as alies.**

```
#useradd -u 1326 alies
```

## **13. Create a archive file**



**(a) Backup the /var/tmp as /root/test.tar.gz**

```
#tar -zcvf /root/test.tar.gz /var/tmp    (-j for bzip2, -z for gzip)
#ls
```

**Note: For Practise 14 and 15 Questions**

**a) In Our lab do this in "servere".**

**b) In the Redhat lab do this in "servera".**

**14. Create the container as a system startup service.**

**(a) Create the container name as logserver with the images rsyslog are stored in registry on paradise user**

**(b) The container should be configured as system startup services**

**(c) The container directory is container\_ journal should be created on paradise user**

**15. Configure the Container as persistent storage and create logs for container**

**(a) Configure the container with the persistent storage that mounted on /var/log/journal to /home/paradise/container**

**(b) The container directory contains all journal files**

\*\*\*\*\* Take two terminals one is for root and another is for user. Note: "#" indicates root and "\$" indicates user. Take ssh to paradise user \*\*\*\*\*

```
#ls /var/log
```

```
#vim /etc/systemd/journald.conf
```

```
[journal]
```

```
storage=persistent ("Replace #storage=auto with storage=persistent")
```

```
Esc :wq
```

```
#systemctl restart systemd-journald.service
```

```
#ls /var/log (o/p: need to find journal directory)
```

```
#ls /var/log/journal/f2.....(use tab) (need to find system.journal file)
```

```
$mkdir ~/container_journal
```

```
#cp -rvf /var/log/journal/f2..../*.journal /home/paradise/container_journal
```

```
$podman login registry.lab.example.com  
(Provide credentials from important configuration)
```

```
$podman search rsyslog
```

```
$podman pull registry.lab.example.com/rhel8/rsyslog
```

```
$podman images
```

```
$podman run -d --name logserver -v /home/paradise/container_journal:/var/log/journal:Z  
registry.lab.example.com/rhel8/rsyslog
```

```
$podman ps
```

```
$mkdir -p /home/paradise/.config/systemd/user
```

```
$cd /home/paradise/.config/systemd/user
```

```
$podman generate systemd --name logserver --files --new
```

```
$podman stop logserver
```

```
$podman rm logserver
```

```
$podman ps (we should not find logserver)
```

```
$systemctl --user daemon-reload
```

```
$systemctl --user start container-logserver.service
```

```
$podman ps
```

```
#loginctl show-user paradise (if linger=NO then run below command)
```

```
#loginctl enable-linger paradise
```

```
#loginctl show-user paradise (o/p :linger=YES)
```

```
#reboot
```

After reboot

---

```
#ssh paradise@serverip
```

```
#podman ps (need to find logserver container)
```

## 16.1 Set the permission

**(a) All new creating files for user natasha as -r----- as default permission.**

**(b) All new creating directories for user natasha as dr-x----- as default permission.**

```
#su - natasha
```

```
$umask
```

```
$ls -la
```

```
$vim .bash_profile
```

```
    umask 277
```

```
Esc:wq
```

```
$source .bash_profile
```

```
$umask
```

```
$mkdir dir1
```

```
$ls -l ( o/p directory should have dr-x----- permissions)
```

```
$touch file1
```

```
$ls -l ( o/p file should have -r----- permissions)
```

```
$logout
```

## 16.2 Set the Password expire date

**(a) The password for all new users in [serverb.lab.example.com](http://serverb.lab.example.com) should expires after 20 days.**

```
#vim /etc/login.defs
```

```
    PASS_MAX_DAYS    20
```

```
Esc:wq
```

## 16.3 Assign Sudo Privilege

**(a) Assign the Sudo Privilege for Group "admin" and Group members can administrate without any password.**

```
#vim /etc/sudoers
```

```
%admin ALL=(ALL) NOPASSWD=ALL
Esc:wq!
```

## 16.4 Create the script file

(a) Create a mysearch script to locate file under /usr/share having size less than 1M  
(b) After executing the mysearch script file and listed (searched) files has to be copied under /root/myfiles.

```
#mkdir /root/myfiles
#vim mysearch
#!/bin/bash
find /usr/share -type f -size -1M -exec cp {} /root/myfiles \;
(if >1M put::: -size 1M
if the range is in between 100M and 200M then::: -size 100M -size -200M)
Esc:wq
#chmod +x mysearch
#./mysearch
#ls -a /root/myfiles
#reboot (mandatory)
```

server lab [example.com](http://example.com) should be contain this following details

### 1. Assign root user password as northale.

In console:

Click on ctrl+alt+del(it will be there at the top right corner)

Press e

Rd.break

Ctrl+x

```
#mount -o remount,rw /sysroot
```

```
#chroot /sysroot
```

```
#passwd root
```

Type password:

Retype password:

```
#touch /.autorelabel
```

```
#exit
```

```
#exit
```

## 2. Create a repository file

[http://classroom.example.com/content/thel8.0/x86\\_64/dvd/AppStream](http://classroom.example.com/content/thel8.0/x86_64/dvd/AppStream)

[http://classroom.example.com/content/thel8.0/x86\\_64/dvd/BaseOS](http://classroom.example.com/content/thel8.0/x86_64/dvd/BaseOS)

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

```
[1(any name, should be in square braces)]
```

```
name=AppStream(any name)
```

```
baseurl:http://classroom.example.com/content/thel8.0/x86\_64/dvd/AppStream
```

```
enabled=1
```

```
gpgcheck=0
```

```
[2(any name, should be in square braces)]
```

```
name=BaseOS(any name)
```

```
baseurl:http://classroom.example.com/content/thel8.0/x86\_64/dvd/BaseOS
```

```
enabled=1
```

```
gpgcheck=0
```

```
esc:wq
```

```
#yum clean all (it clears all cache)
```

```
#yum repolist all (it lists the repositories)
```

```
#yum install httpd -y (if it won't install the issue may be in repo file or setting  
ip address or dns or gateway)
```

## 3. Create a swap partition 512MB size.

```
#lsblk
```

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

```
n (for new)
```

```
Press Enter (for primary)
```

```
Press Enter (for partition )
```

```
Press Enter (for starting size)
```

```
+512M (need to provide size as given in question)
```

```
t (type)
```

```
        82 (for swap)
        W (to save and exit)
#udevadm settle

#lsblk (to check)

#mkswap /dev/vdb1

#vim /etc/fstab
        /dev/vdb1    swap    swap    defaults    0 0
Esc :wq

#free -h

#swapon -a

#free -h
```

**4. Create one logical volume named database and it should be on datastore volume group with size 50 extent and assign the filesystem as ext3.**

**the datastore volume group extend should be 8MiB (mount the logical volume under mount point /mnt/database.**

```
#fdisk /dev/vdb
n
Press Enter
Press Enter
Press enter
+2G
t
8e
w

#udevadm settle

#lsblk

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

```
#lvcreate -l 50 -n database datastore  
(-l for extends and -L for size)
```

```
#mkfs.ext3 /dev/datastore/database
```

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

```
#vi /etc/fstab
```

```
    /dev/datastore/database /mnt/database ext3 defaults 0 0
```

```
Esc :wq
```

```
#mount -a
```

```
#df -hT
```

## **5. Create the vectra volume using the VDO with the logical size 50GB and mount under test directory**

```
#yum install vdo -y
```

```
#systemctl start vdo
```

```
#systemctl enable vdo
```

```
#lsblk (we need unused disk)
```

```
#vdo create --name=vectra --device=/dev/vdc --vdoLogicalSize=50G
```

```
#vdo list
```

```
#mkfs.xfs /dev/mapper/vectra
```

```
#mkdir /test
```

```
#vi /etc/fstab
```

```
    /dev/mapper/vectra /test xfs defaults,x-systemd.requires=vdo.service 0 0
```

```
Esc :wq
```

```
#mount -a
```

```
#df -hT
```

```
#systemctl restart vdo
```

**6. Resize the logical volume size of 100 extent on /mnt/database directory.**

```
#lvextend -l 100 -r /mnt/database
```

**7. Set the recommended tuned profile for your system.**

```
#yum install tuned -y  
#systemctl start tuned  
#systemctl enable tuned  
#tuned-adm recommended  
#tuned-adm profile virtual-guest  
#systemctl restart tuned  
#tuned-adm active
```