

Redhat certified System Administrator (EX200)

servera

[Manage Networking]

Configure network and set the static parameters. Consider machine configured as DHCP, need to config it with static parameters.

```
IP-ADDRESS= 172.25.250.10
NETMASK= 255.255.255.0
GATEWAY= 172.25.250.254
Nameserver= 172.24.254.254
Hostname= servera.lab.example.com
```

Answer:

```
[root@test1 ~]# nmcli connection show
```

```
[root@test1 ~]# nmcli connection modify LAN ipv4.addresses 172.25.250.10/24 ipv4.gateway 172.25.250.254 ipv4.dns
172.25.254.254
```

```
[root@test1 ~]# nmcli connection modify LAN ipv4.method manual
```

```
[root@test1 ~]# nmcli connection up LAN
```

```
[root@test1 ~]# nmcli connection show
```

Set up hostname

```
=====
```

```
[root@test1 ~]# hostnamectl set-hostname servera.lab.example.com
```

```
[root@test1 ~]# hostnamectl status
```

[Installing and Updating Software Packages]

- Configure your system to use this location as a default repository (public/local repo):
- http://content.example.com/rhel9.0/x86_64/rhcsa-practice/rht
- http://content.example.com/rhel9.0/x86_64/rhcsa-practice/errata

Answer:

```
[root@servera ~]# vim /etc/yum.repos.d/exam.repo
```

```
[demo1]
```

```
name=demo1 repo
```

```
baseurl=http://content.example.com/rhel9.0/x86_64/rhcsa-practice/rht
```

```
enabled=1
```

```
gpgcheck=0
```

```
[demo2]
name=AppStram repo
baseurl= http://content.example.com/rhel9.0/x86_64/rhcsa-practice/errata
enabled=1
gpgcheck=0
```

```
[root@servera ~]#
[root@servera ~]#dnf repolist
repo id                repo name
demo1                  demo1 repo
demo2                  demo2 repo
```

[Managing Local Users and Groups]

Create the following users, groups and group memberships:

- A group named **sharegrp**
- A user **harry** who belongs to **sharegrp** as a secondary group
- A user **natasha** who also belongs to **sharegrp** as a secondary group
- A user **copper** who does not have access to an interactive shell on the system and who is not a member of **sharegrp**.
- **harry, natasha and copper** should have the password **redhat**

Answer:

```
[root@servera ~]# groupadd sharegrp
[root@servera ~]# useradd -G sharegrp harry
[root@servera ~]# useradd -G sharegrp natasha
[root@servera ~]# useradd -s /sbin/nologin copper
[root@servera ~]# passwd harry
[root@servera ~]# passwd natasha
[root@servera ~]# passwd copper
```

For verification :

```
[root@servera ~]# tail -5 /etc/passwd
[root@servera ~]# tail -5 /etc/group
[root@servera ~]# tail -5 /etc/shadow
```

[Controlling Access to Files]

Create collaborative directory **/var/shares** with the following characteristics:

- Group ownership of **/var/shares** should be **sharegrp**.
- The directory should be readable, writable and accessible to member of **sharegrp** but not to any other user. (It is understood that root has access to all files and directories on the system)

- Files created in `/var/shares` automatically have group ownership set to the `sharegrp` group.

Answer:

```
[root@servera ~]# mkdir -p /var/shares
[root@servera ~]# ls -ld /var/shares
[root@servera ~]# chgrp sharegrp /var/shares/
Or
[root@servera ~]# chown :sharegrp /var/shares/
[root@servera ~]# ls -ld /var/shares
```

[Accessing Linux File Systems]

Find all lines in the file `/usr/share/mime/packages/freedesktop.org.xml` that contain the string `ich`. Put a copy of these lines in the original order in the file `/root/lines`. `/root/lines` should contain no empty lines and all lines must be exact copies of the original lines in `/usr/share/mime/packages/freedesktop.org.xml`

Answer:

```
[root@servera ~]# grep ich /usr/share/mime/packages/freedesktop.org.xml > /root/lines
[root@servera ~]# cat /root/lines
<comment xml:lang="ast">Ficheru codificáu en BinHex de Machintosh</comment>
<comment xml:lang="fr">fichier codé Macintosh BinHex</comment>
<comment xml:lang="gl">ficheiro de Macintosh codificado con BinHex</comment>
<comment xml:lang="oc">fichièr encodat Macintosh BinHex</comment>
<comment xml:lang="pt">ficheiro codificado em BinHex de Macintosh</comment>
<comment xml:lang="fr">fichier boîte aux lettres</comment>
<comment xml:lang="gl">ficheiro de caixa de correo</comment>
.....output omitted.....
```

[Accessing Linux File Systems]

Find all the files owned by user `natasha` and redirect the output to `/tmp/output`.

Find all files that are larger than `5MiB` in the `/etc` directory and copy them to `/find/largedir` and redirect the output to `/find/largefiles`

Answer:

```
[root@servera ~]# find / -user natasha -type f > /tmp/output
[root@servera ~]# cat /tmp/output
/var/spool/mail/natasha
/mnt/shares/natasha

[root@servera ~]# find /etc -size +5M -exec cp {} /find/largedir \;
[root@servera ~]# cd /find/largedir; ls
policy.31
hwdb.bin
```

OR

```
[root@servera ~]# find /etc -size +5M > /find/largedir
```

[Managing Local Users and Groups]

Create a user **fred** with a user ID 3945. Give the password as **iamredhatman**

Answer:

```
[root@servera ~]# useradd -u 3945 fred
```

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

For Verification:

```
[root@servera ~]# tail -1 /etc/passwd
```

```
[root@servera ~]# tail -1 /etc/shadow
```

[Managing Files from the Command Line]

Search the string **nologin** in the `/etc/passwd` file and save the output in `/root/strings`

Answer:

```
[root@servera ~]# grep nologin /etc/passwd > /root/strings
```

```
[root@servera ~]# cat /root/strings
```

```
bin:x:1:1:bin:/bin:/sbin/nologin
```

```
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

```
adm:x:3:4:adm:/var/adm:/sbin/nologin
```

```
.....output omitted.....
```

[Configuring NTP/Time Synchronization]

Configure your system so that it is an NTP client of **classroom.example.com**

Answer:

```
[root@servera ~]# dnf install chrony
```

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

```
server classroom.example.com iburst [ note: # iburst line and write this line ]
```

```
[root@servera ~]# systemctl enable chronyd --now
```

```
[root@servera ~]# systemctl restart chronyd
```

```
[root@servera ~]# systemctl status chronyd
```

```
[root@servera ~]# chronyc sources -v
```

[Scheduling Future Tasks]

The user **natasha** must configure a cron job that runs daily at 14:23 local time or also the same cron job will run after every 2 minutes and executes:

```
/bin/echo hello
```

Answer:

```
[root@servera ~]# crontab -u natasha -e
23 14 * * * /bin/echo hello
*/2 * * * * /bin/echo hello
```

```
[root@servera ~]# crontab -l -u natasha
23 14 * * * /bin/echo hello
*/2 * * * * /bin/echo hello
```

```
[root@servera ~]# systemctl status crond.service
```

[Archiving and Transferring Files & SELinux]

Create a backup file named **/root/backup.tar.bz2** or **/root/backup.tar.gz2**. The backup file should contain the content of **/usr/local** and should be zipped with **bzip2** or **gzip2** compression format.

Furthermore, ensure SELinux is in enforcing mode. If it is not, change SELinux to enforcing mode.

Answer:

```
[root@servera ~]# tar cjvf /root/backup.tar.bz2 /usr/local
tar: Removing leading `/' from member names
/usr/local/
/usr/local/bin/
/usr/local/etc/
.....output omitted.....
```

```
[root@servera ~]# ls
backup.tar.bz2
```

[Create a Bash Script]

Questions: Create a script file name **find.sh**. when you run this script, it will find all files from 30K to 60k file size from the directory **/etc** directory & copies those files to **/root/data** directory.

ANSWER:

```
[root@servera ~]# cat find.sh
```

```
#!/bin/bash
```

```
# Directory where files will be copied
```

```
DEST_DIR="/root/data"
```

```
# Check if destination directory exists, if not create it
```

```
if [ ! -d "$DEST_DIR" ]; then  
    mkdir -p "$DEST_DIR"  
fi
```

```
# Find files in /etc with size between 30 KB and 60 KB and copy them to /root/data  
find /etc -type f -size +30k -size -60k -exec cp {} "$DEST_DIR" \;
```

```
echo "Files copied successfully to $DEST_DIR"
```

```
[root@servera ~]# chmod a+x find.sh
```

[Managing SELinux Security]

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.

Answer:

```
[root@servera ~]# curl http://servera.lab.example.com:82  
curl: (7) Failed to connect to servera.lab.example.com port 82: Connection refused
```

```
[root@servera ~]# systemctl status httpd
```

```
● httpd.service - The Apache HTTP Server  
Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)  
Active: failed (Result: exit-code) since Sun 2022-07-03 18:23:38 +06; 1min 19s ago  
Docs: man:httpd.service(8)  
Process: 801 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=1/FAILURE)  
Main PID: 801 (code=exited, status=1/FAILURE)  
Status: "Reading configuration..."
```

```
Jul 03 18:23:38 servera.lab.example.com systemd[1]: Starting The Apache HTTP Server...  
Jul 03 18:23:38 servera.lab.example.com httpd[801]: (13)Permission denied: AH00072: make_sock: could not bind to address [::]:82  
Jul 03 18:23:38 servera.lab.example.com httpd[801]: (13)Permission denied: AH00072: make_sock: could not bind to address 0.0.0.0:82  
Jul 03 18:23:38 servera.lab.example.com httpd[801]: no listening sockets available, shutting down  
Jul 03 18:23:38 servera.lab.example.com httpd[801]: AH00015: Unable to open logs  
Jul 03 18:23:38 servera.lab.example.com systemd[1]: httpd.service: Main process exited, code=exited, status=1/FAILURE  
Jul 03 18:23:38 servera.lab.example.com systemd[1]: httpd.service: Failed with result 'exit-code'.  
Jul 03 18:23:38 servera.lab.example.com systemd[1]: Failed to start The Apache HTTP Server.
```

```
[root@servera ~]# systemctl start httpd  
[root@servera ~]# journalctl -xe
```

If you want to allow httpd to bind to network port 82

Then you need to modify the port type.

Do

```
# semanage port -a -t PORT_TYPE -p tcp 82
```

where PORT_TYPE is one of the following: http_cache_port_t, http_p>

```
***** Plugin catchall (1.49 confidence) suggests *****>
```

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

```
[root@servera ~]# semanage port -a -t http_port_t -p tcp 82
```

```
[root@servera ~]# semanage port -l | grep http
http_cache_port_t   tcp    8080, 8118, 8123, 10001-10010
http_cache_port_t   udp    3130
http_port_t         tcp    82, 80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t tcp    5988
```

```
[root@servera ~]# firewall-cmd --permanent --add-port=82/tcp
[root@servera ~]# firewall-cmd --reload
[root@servera ~]# firewall-cmd --list-all
[root@servera ~]# systemctl restart httpd
[root@servera ~]# curl http://servera.lab.example.com:82
```

This is Webservice

[Set the Password expire date]

The password for all new users in servera.lab.example.com should expires after 30 days.

Answer:

```
[root@servera ~]# vim /etc/login.defs
```

```
# Password aging controls:
#
#     PASS_MAX_DAYS   Maximum number of days a password may be used.
#     PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#     PASS_MIN_LEN     Minimum acceptable password length.
#     PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS 30  #( Modify this line & set the vlaue 30. default value was 99999 )
PASS_MIN_DAYS 0
PASS_MIN_LEN 5
PASS_WARN_AGE 7

#
# Min/max values for automatic uid selection in useradd
#
UID_MIN 1000
UID_MAX 60000
# System accounts
:wq
```

[Autofs Configuration]

Configure autofs to automount the home directories of user remoteuser15. Note the following:

- utility.lab.example.com (172.24.10.10), NFS-exports /netdir to your system, where user is remoteuser15
- remoteuser15's home directory is utility.lab.example.com:/netdir/remoteuser15
- remoteuser15's home directory should be auto mounted locally beneath /netdir as /netdir/remoteuser15

- Home directories must be writable by their users while you are able to login as any of the remoteuser15 only home directory that is accessible from your system

Answer:

```
[root@servera ~]# dnf install autofs
[root@servera ~]# systemctl enable --now autofs
[root@servera ~]# vim /etc/auto.master.d/demo.autofs
    /netdir /etc/auto.demo
[root@servera ~]# vim /etc/auto.demo
    remoteuser15 -rw,sync utility.lab.example.com:/netdir/remoteuser15
[root@servera ~]# systemctl restart autofs
[root@servera ~]# df -h
[root@servera ~]# su - remoteuser15
[remoteuser15@servera ~]$ ls
[remoteuser15@servera ~]$ touch file1
```

[Question for second node server.lab.example.com]

Add a Swap partition

Add an additional swap partition of 512 MiB to your system. The swap partition should automatically mount when your system boots. Do not remove or otherwise alter any existing swap partition on your system.

Answer:

```
# parted /dev/vdb print
    Model: Virtio Block Device (virtblk)
    Disk /dev/vdb: 5369MB
    Sector size (logical/physical): 512B/512B
    Partition Table: gpt
    Disk Flags:
    Number Start End Size File system Name Flags
    1 1049kB 1001MB 1000MB data
# parted /dev/vdb mkpart myswap linux-swap 1001MB 1501MB
# udevadm settle
# mkswap /dev/vdb2
# swapon /dev/vdb2
# vim /etc/fstab ; append the following line
/dev/vda2 swap swap defaults 0 0
# swapon -a
# free -m (for check the memory status)
```

[Create a logical volume]

Create a new logical volume according to the following requirements:

- The logical volume is named database and belongs to the datastore volume group and has a size of 50 extents.
- Logical volume in the datastore volume group should have an extent size of 16 MiB.
- Format the new logical volume with vfat filesystem. The logical volume should be mounted automatically mounted under /mnt/database at system boot time.

Answer:

```
# parted /dev/vdb print

Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
Number Start End Size File system Name Flags
1 1049kB 1001MB 1000MB data
2 1001MB 1501MB 499MB myswap swap # partprobe

# parted /dev/vdb mkpart primary 1001MB 2001MB
# parted /dev/vdb set 3 lvm on
# udevadm settle
# pvcreate /dev/vdb3
# vgcreate -s 16M datastore /dev/vdb3
# vgdisplay
# lvcreate -n database -L 800M datastore
# lvdisplay
# mkfs.vfat /dev/datastore/database
# mkdir /mnt/database
# vim /etc/fstab ; append the following line

/dev/datastore/database /mnt/database ext4 defaults 0 0

# mount -a
# df -h
```

[LVM partition resize]

LVM partition resize re-size LVM partition 850MB. Where LV name is database. Partition size must be within approximately 830MB to 865MB and usable.

Answer:

```
# df -h (for check the current LV size)
# lvdisplay (check the current LV Size & LV path)
# lvextend -r -L 850M /dev/datastore/database
# lvdisplay
# df -h
```

[TUNING SYSTEM PERFORMANCE]

Change the current tuning profile for server to default profile

Answer:

```
# dnf install tuned -y
# systemctl enable tuned
```

```
# systemctl start tuned

# tuned-adm recommend
  virtual-guest
# tuned-adm profile virtual-guest
# tuned-adm active
# tuned-adm profile_info ( check summary information of the current active tuned profile )
```

UMASK Configure

Questions: Set permission for user daffy. User will get the permission below for file & directory when he creates new files or directory.

```
-rw-----
drwx-----
```

Answer:

```
[root@servera~]# su - daffy
[daffy@servera~]$ vim .bashrc
  umask 0077 ( append lis line to .bashrc file , save and exit from .bashrc )
[daffy@servera~]$ exit
[root@servera~]#
```

SUDO Configuration

Question: Configure sudo permissions for users who are members of the admin group, allowing them to use sudo without a password.

Answer:

```
[root@servera~]#vim /etc/sudoers.d/exam

%admin ALL=(ALL) NOPASSWD: ALL (insert this line and save and exit )
```

Container Questions

Questions 01: Create a container image named *monitor* from a Containerfile from below link <http://fromwebserver/Containerfile> . All this task done using student user.

Answer:

```
[student@hosta ~]$ podman login <exam registry url > ( man
username: yourusername
password: your registry login pass

[student@hosta ~]$ mkdir demo
[student@hosta ~]$ cd demo
[student@hosta demo]$ wget http://fromwebserver/Containerfile
```

```
[student@hosta demo]$  
[student@hosta demo]$ podman build -t process_files .  
[student@hosta demo]$ podman images  
REPOSITORY          TAG      IMAGE ID      CREATED      SIZE  
localhost/monitor   latest   aa30769bf294 30 seconds ago 489 MB
```

Questions 02: Create rootless container and do volume mapping which they asked you in the question and run container as a service from normal user account, the service must be enable so it could start automatically after reboot.

- Create a container named as 'ascii2pdf' using the previously created container image from previous question monitor
- Map the '/opt/files' to container '/opt/incoming'
- Map the '/opt/processed' to container '/opt/outgoing'
- Create systemd service as container-ascii2pdf.service
- Make service active after all server reboots

Answer:

```
#mkdir /opt/files /opt/processed  
#chown student:student /opt/files /opt/processed  
# ssh stuednt@localhost
```

```
$podman run -d --name ascii2pdf -v /opt/files:/opt/incoming:Z -v /opt/processed:/opt/outgoing:Z  
localhost:monitor:latest
```

```
$ mkdir ~/.config/systemd/user  
$cd ~/.config/systemd/user  
$podman generate systemd --name ascii2pdf --new --files  
$podman stop ascii2pdf  
$podman rm ascii2pdf  
$systemctl --user daemon-reload  
$loginctl enable-linger  
$systemctl --user enable container-ascii2pdf.service --now  
$ podman ps
```