

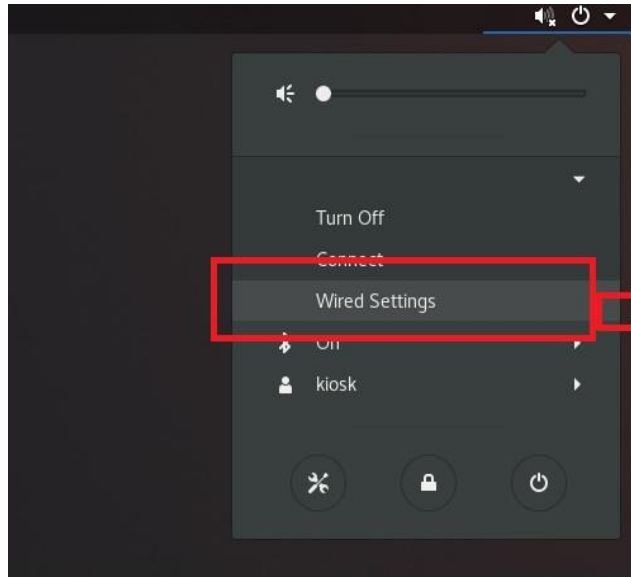
RHCSA EXAM

SOLUTIONS

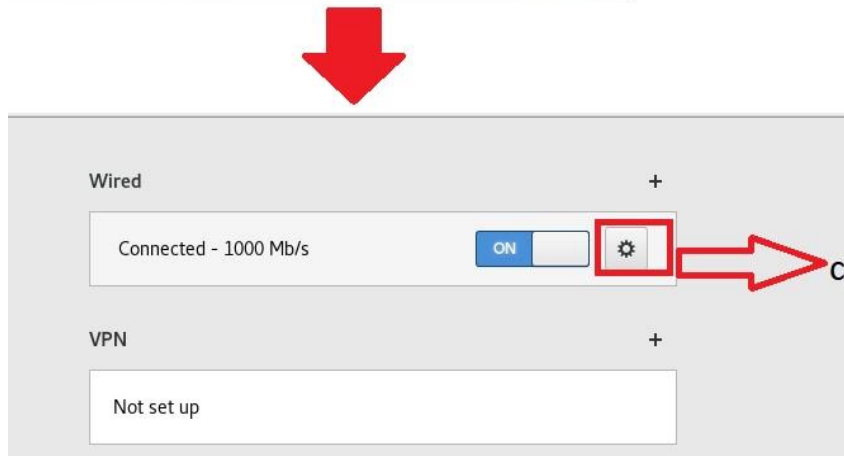
Machine 1 :- first question is set the network and hostname:-

- Start the first machine. Type root and password(root password is given in exam).
- First you have to setup the network(ipv4.address,gateway,dns is given in exam).
- Type **startx** for graphical login
- There are two way to setup network first graphical and second is from command line(nmcli)

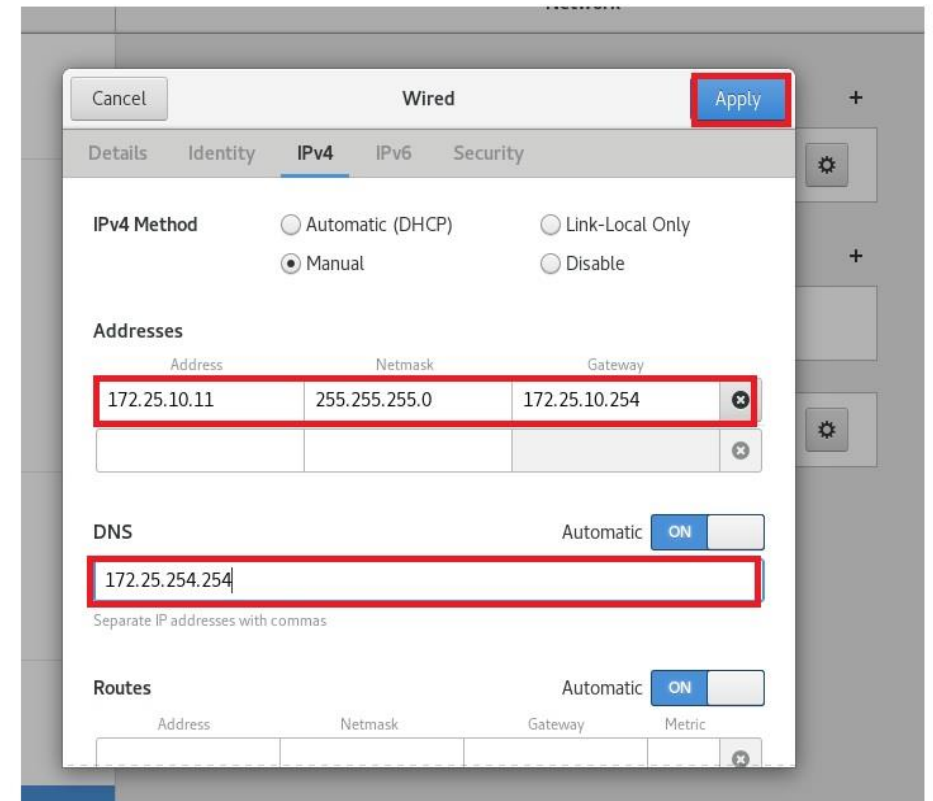
Graphical method:-



click on wired settings



click on setting



Enter ip address Netmask Gateway and DNS. Than click on APPLY.

Than connection OFF and ON

Command line (nmcli)

First check connection name:-

```
File Edit View Search Terminal Help
[root@servera ~]# nmcli connection show
NAME                UUID                                  TYPE      DEVICE
Wired connection 1  6bc56692-0f3b-3bf9-941f-8bc9f5ff7941 ethernet  enp1s0
```

Then we have command nmcli connection modify "connection name" ipv4.(double tab). then this show all options of ipv4.

```
[root@servera ~]# nmcli connection modify "Wired connection 1" ipv4.
ipv4.addresses      ipv4.dhcp-hostname  ipv4.dns-options    ipv4.ignore-auto-dns  ipv4.never-default
ipv4.dad-timeout    ipv4.dhcp-send-hostname  ipv4.dns-priority    ipv4.ignore-auto-routes  ipv4.route-metric
ipv4.dhcp-client-id  ipv4.dhcp-timeout      ipv4.dns-search      ipv4.mav-fail          ipv4.routes
ipv4.dhcp-fqdn       ipv4.dns               ipv4.gateway          ipv4.method             ipv4.route-table
[root@servera ~]# nmcli connection modify "Wired connection 1" ipv4.addresses 172.25.10.11/24 ipv4.gateway 172.25.10.254 ipv4.dns 172.25.254.254 ipv4.method static^C
```

Now enter ip address Gateway DNS and Method. And press enter.

```
File Edit View Search Terminal Help
[root@servera ~]# nmcli connection modify "Wired connection 1" ipv4.addresses 172.25.10.11/24 ipv4.gateway 172.25.10.254 ipv4.dns 172.25.254.254 ipv4.method static
```

Now first connection down and then up.

```
[root@servera ~]# nmcli connection down
```

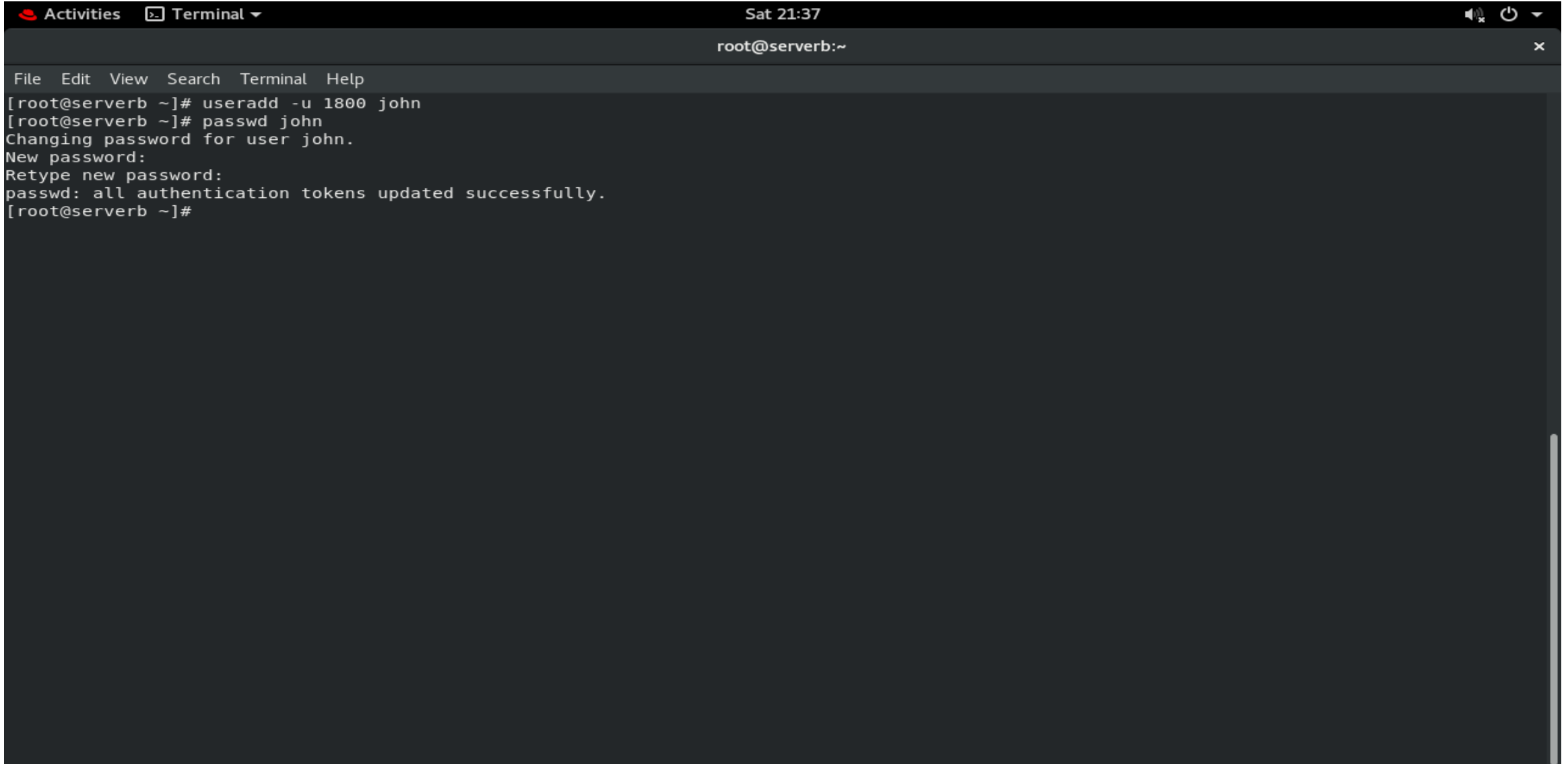
```
[root@servera ~]# nmcli connection up
```

Set Hostname:-

- `# hostnamectl set-hostname server.example.com`
- `# exec bash`
- Stop firewall in exam only in first paper
- `#systemctl stop firewalld.service`

Add a user john with UID is 1800. Set password of john is thuctive

User add with UID



```
Activities Terminal ▾ Sat 21:37 root@serverb:~ x
File Edit View Search Terminal Help
[root@serverb ~]# useradd -u 1800 john
[root@serverb ~]# passwd john
Changing password for user john.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serverb ~]#
```

The image shows a terminal window titled "Terminal" with a dark theme. The window's title bar includes "Activities", "Terminal", and a dropdown arrow, along with system icons for volume, power, and network on the right. The terminal content shows a root user at a server named "serverb" with a tilde (~) as the home directory. The user runs the command "useradd -u 1800 john", followed by "passwd john". The system prompts for a new password and its retype, then confirms that all authentication tokens are updated successfully. The prompt returns to the root user at the server.

Create the following users, groups, and group memberships:-

A group named sysadmin. A user natasha who belongs to sysadmin as a secondary group. A user sarah who also belongs to sysadmin as a secondary group. A user harry who does not have access to an interactive shell on the system, and who is not a member of sysadmin. Natasha, Sarah and Harry should all have the password of thuctive

×

Create a collaborative directory “/common/admin” with the following characteristics: Group ownership of /common/admin is sysadmin. The directory should be readable, writable, and accessible to members of sysadmin, but not to any other user. (It is understood that root has access to all files and directories on the system.) Files created in /common/admin automatically have group ownership set to the sysadmin group.

root@serverb:~

File Edit View Search Terminal Help

```
[root@serverb ~]# mkdir -p /common/admin
[root@serverb ~]# chgrp sysadmin /common/admin/
[root@serverb ~]# ls -l /common/
total 0
drwxr-xr-x. 2 root sysadmin 6 Dec 28 03:31 admin
[root@serverb ~]# chmod 2770 /common/admin/
[root@serverb ~]# ls -l /common/
total 0
drwxrws---. 2 root sysadmin 6 Dec 28 03:31 admin
[root@serverb ~]# touch /common/admin/a
[root@serverb ~]# ls -l /common/admin/a
-rw-r--r--. 1 root sysadmin 0 Dec 28 03:32 /common/admin/a
[root@serverb ~]#
```

We're creating a collaborative directory with "mkdir" command which is used to make directory. -> In this command, "-p" option is used to make parent directory of current directory simultaneously. -> With "mkdir" command and "-p" option, we'll give directory name with parent directory name like "/common/admin" and both directory will be created simultaneously.

chgrp command is used to change group ownership of any directory.

"chmod" command is used to give permission to root, group or any other user. -> In this command, after "chmod" command first digit which is "2" is used to give special permission which is SGID by which all files in "/common/admin" directory automatically have group ownership and second digit which is "7" is used to give read, write and access to root user and third digit which is "7" is used to give read, write and access to group and last digit which is "0" is used to give no any permission of read, write and access to any other user. -> At last, we give path of directory on which we want to give permission.

Copy the file `/etc/fstab` to `/var/tmp`. Configure the permissions of `/var/tmp/fstab` so that:-

The file `/var/tmp/fstab` is owned by the root user. The file `/var/tmp/fstab` belong to the group root. The file `/var/tmp/fstab` should not be executable by anyone. The user natasha is able to read and write `/var/tmp/fstab`. The user sarah can neither write nor read `/var/tmp/fstab`. [Note: all other users (current or future) have the ability to read `/var/tmp/fstab`.


```
[root@serverb ~]# cp /etc/fstab /var/tmp/
[root@serverb ~]# chmod 774 /var/tmp/fstab #as per question set permission
[root@serverb ~]# setfacl -m u:natasha:rwX /var/tmp/fstab
[root@serverb ~]# setfacl -m u:sarah:--- /var/tmp/fstab
[root@serverb ~]# su - natasha
[natasha@serverb ~]$
```

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

First, we open the file "chrony.conf" with help of vim editor

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

Then we write server related entry. server servername/ip(given in exam) iburst .

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.rhel.pool.ntp.org iburst
#server 1.rhel.pool.ntp.org iburst
#server 2.rhel.pool.ntp.org iburst
#server 3.rhel.pool.ntp.org iburst
server classroom.example.com iburst

# Ignore stratum in source selection.
stratumweight 0

# Record the rate at which the system clock goes (loses time
```

Noe we'll restart the service of "chronyd.service" and enable this service by which all configuration will remain same after reboot of system.

By this command, we can cross check the connection with server and also can check that system time is synchronized with server time or not

```
root@servera ~]# systemctl restart chronyd.service
root@servera ~]# chronyc sources -v
210 Number of sources = 1


.-- Source mode '^' = server, '=' = peer, '#' = local clock.
/ .. Source state '*' = current synced, '+' = combined , '-' = not combined,
/ '?' = unreachable, 'x' = time may be in error, '~' = time too variable.

|                                     | .- xxxx [ yyyy ] +/- zzzz
| Reachability register (octal) -.   |      xxxx = adjusted offset,
| Log2(Polling interval) --.        |      yyyy = measured offset,
|                                   \   |      zzzz = estimated error.
|                               |       |
|                               |       |
MS Name/IP address             Stratum Poll Reach LastRx Last sample
=====
** classroom.example.com         8     6    17    13   -59us[ -597us] +/- 4269us
root@servera ~]#
```



Find the files in your system which is owned by Simone user & copy all the files on /root/found Directory

root@servera:~

File Edit View Search Terminal Help



```
[root@servera ~]# mkdir /root/found
[root@servera ~]# find / -user simone -exec cp -rvf {} /root/found \;
find: '/proc/2012/task/2012/fd/6': No such file or directory
find: '/proc/2012/task/2012/fdinfo/6': No such file or directory
find: '/proc/2012/fd/7': No such file or directory
find: '/proc/2012/fdinfo/7': No such file or directory
'/var/spool/mail/simone' -> '/root/found/simone'
cp: cannot overwrite non-directory '/root/found/simone' with directory '/home/simone'
'/home/simone/.bash_logout' -> '/root/found/.bash_logout'
'/home/simone/.bash_profile' -> '/root/found/.bash_profile'
'/home/simone/.bashrc' -> '/root/found/.bashrc'
[root@servera ~]#
```



-> By "mkdir" command, We make a new directory where we can copy all find files.

find / -user (user_name) -exec cp -rvf {} (path of directory where we want to copy) \; -> "find" is a command to find any file or directory. -> After the "find" command, we use "/" which means this command will find the required file in whole system because whole directory or file is made in "/". -> After "/", We use "-user" option which is used to specify the name of user and we're using this option when we want to find that files which is owned by any user. -> After specify the user name, we use "-exec" option by which we can add or join another command with previous command. -> After the "-exec" option, we use "cp" command for copy all files at given path of directory. In "cp" command, we use certain following option- -a = This option is used to append command with previous command. -r = This option is used to copy files recursively. -v = This option is used to print verbose information on screen means process shows on display. -f = This option is used to copy all files forcefully. -> After all options, we give path of that directory where we want to copy all file which is find and owned by specified user.

Find the string strato from
/usr/share/dict/words/file and save the result
in /searchfile.

```
[root@servera ~]# grep 'strato' /usr/share/dict/words > /searchfile
[root@servera ~]# cat /searchfile
```

Using automounter service mount
RemoteuserX onto the provided folder
/ourhome/RemoteuserX

First:- Install package autofs.X86_64

```
root@servera ~]# yum install -y autofs.x86_64
```

Second:- Start and enable autofs.service.

then create a file /etc/auto.master.d/xyz.autofs(create file whatever name you want and ext must be autofs.)

```
[root@servera ~]# systemctl start autofs.service
[root@servera ~]# systemctl enable autofs.service
Created symlink /etc/systemd/system/multi-user.target.wants/autofs.service → /usr/lib/systemd/system/autofs.service.
[root@servera ~]# vim /etc/auto.master.d/xyz.autofs
```

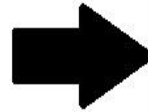
third :- entry in this file

/- /etc/auto.misc

```
/- /etc/auto.misc
```

Now open /etc/auto.misc file and entry in that file servername/ip and directory is given in exam:
(local mount point) -ro,soft,intr servername/ip:directory which is shared from server
save file. then restart the service.

```
# the following entries are samples to pique your imagination
#linux      -ro,soft,intr      ftp.example.org:/pub/linux
#boot       -fstype=ext2       /dev/hda1
#floppy     -fstype=auto       /dev/fd0
#floppy     -fstype=ext2       /dev/fd0
#e2floppy   -fstype=ext2       /dev/fd0
#jaz        -fstype=ext2       /dev/sdc1
#removable  -fstype=ext2       /dev/hdd
/remoteuser -ro,soft,intr      classroom.example.com:/home/remoteuser/test123
```



```
[root@servera ~]# systemctl restart autofs.service
[root@servera ~]# cd /remoteuser/
[root@servera remoteuser]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	892M	0	892M	0%	/dev
tmpfs	915M	0	915M	0%	/dev/shm
tmpfs	915M	17M	899M	2%	/run
tmpfs	915M	0	915M	0%	/sys/fs/cgroup
/dev/vda1	10G	1.5G	8.6G	15%	/
tmpfs	183M	0	183M	0%	/run/user/0
classroom.example.com:/home/remoteuser/test123	10G	1.6G	8.5G	16%	/remoteuser

```
[root@servera remoteuser]#
```

#systemctl restart autofs.service and go to local dirt. and check with df -h

The user natasha must configure a cron job that runs daily at 14:23 local time and executes in /bin/echo hiii.

- We have command crontab to set cronjob
- #crontab -eu Natasha (e- edit u- user)
23 14 * * * /bin/echo hiii
- crontab -lu Natasha (l- list & Show the cronjob of user natasha)

Start httpd service. Your apache is configure on 82 port. Set selinux context respectively to start httpd service.

First install httpd package:-

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

Now Stop firewall :-

```
# systemctl stop firewalld.service
```

Now open man page of semanage port

```
root@servera ~]# man semanage port
```

Now go to examples:-

```
File Edit View Search Terminal Help
```

EXAMPLE

List all port definitions

```
# semanage port -l
```

Allow Apache to listen on tcp port 81

```
# semanage port -a -t http_port_t -p tcp 81
```

Allow sshd to listen on tcp port 8991

```
# semanage port -a -t ssh_port_t -p tcp 8991
```

SEE ALSO

Copy line:- `semanage port -a -t http_port_t -p tcp 82`

then paste and replace 81 with 82.

Now restart/enable httpd service.

```
root@servera ~]# man semanage port
```

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

```
root@servera ~]# systemctl restart httpd
```

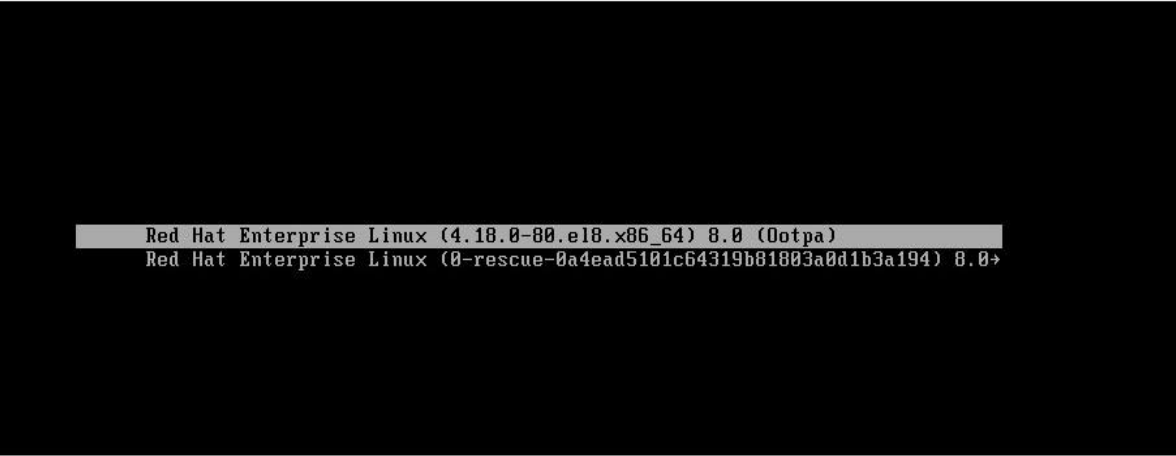
```
root@servera ~]# systemctl enable httpd
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
```

Secondary machine (machine 2)

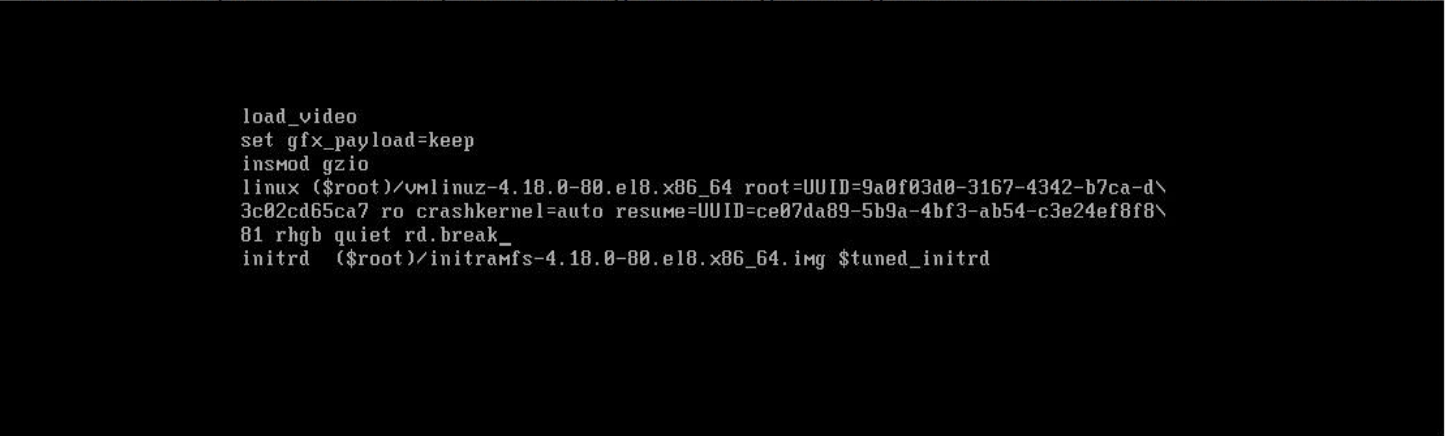
First question is crack the root password:-

when our virtual machine selecting the kernel in starting then there is a option for editing by pressing “e” button on keyboard by which we can done editing in between booting process and also able to crack password of root.



```
Red Hat Enterprise Linux (4.18.0-80.el8.x86_64) 8.0 (Ootpa)
Red Hat Enterprise Linux (0-rescue-0a4ead5101c64319b81803a0d1b3a194) 8.0→
```

when we're in editing mode by pressing “e” button then there's a coding on screen. when you'll go down in coding then there's a line which started by “linux(\$root)”. In this end of line, we write “**rd.break**” and press “**Ctrl + x**” by which booting will start again and give a command line interface where we can run further commands which is required to crack password of root.



```
load_video
set gfx_payload=keep
insmod gzio
linux ($root)/vmlinuz-4.18.0-80.el8.x86_64 root=UUID=9a0f03d0-3167-4342-b7ca-d\
3c02cd65ca7 ro crashkernel=auto resume=UUID=ce07da89-5b9a-4bf3-ab54-c3e24ef8f8\
81 rhgb quiet rd.break_
initrd ($root)/initramfs-4.18.0-80.el8.x86_64.img $tuned_initrd
```

Step 1:-On command line interface, we write command- **mount -o remount,rw /sysroot** - By this command, we're going to remount with read and write permission on `/sysroot`. -> In this command,"-o" option is used for give option for remount.

Step-2. **chroot /sysroot** -> By this command, we get an interactive shell with special root directory because we want an interactive shell on which we can easily run command to change password of root.

Step-3. **passwd root** -> This is command for changing password of any user so we're going to change password of root.

Step-4. **touch /.autorelabel**

Step-5. **Exit** -> Now by write "exit" on command line by which we can exit from interactive shell.

Step-6. **Exit** -> Now again we write "exit" on command line interface by which we can exit from command line interface and booting process will again started with new root password.

```
Generating "/run/initramfs/rdsosreport.txt"
```

```
Entering emergency mode. Exit the shell to continue.
```

```
Type "journalctl" to view system logs.
```

```
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot  
after mounting them and attach it to a bug report.
```

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

```
switch_root:/# chroot /sysroot
```

```
sh-4.4# passwd
```

```
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-4.4# touch /.autorelabel
```

```
sh-4.4# _
```

After reset root password login as root.

- Type root and password and login
- After login with root for graphical login type startx and enter.
- And First we have to create repo in exam .

Set a recommended tuning profile for your system.
(profile already available)

File Edit View Search Terminal Help

```
[root@servera ~]# yum install -y tuned
Repository 'AppStream' is missing name in configuration, using id.
Repository 'BaseOS' is missing name in configuration, using id.
Last metadata expiration check: 0:17:17 ago on Sat 28 Dec 2019 04:21:51 AM IST.
Package tuned-2.10.0-15.el8.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@servera ~]# tuned-adm
active      list      off      profile  recommend  verify
[root@servera ~]# tuned-adm recommend
virtual-guest
[root@servera ~]# tuned-adm profile virtual-guest
[root@servera ~]# tuned-adm active
Current active profile: virtual-guest
[root@servera ~]#
```

First install tuned package. Then start and enable tuned service.

Then command tuned-adm

recommend:- what is recommend for your system

profile:- to set profile what is recommend.

active :- check what profile is active

list:- to list all profile

Create a backup.tar.(bz2 or gz) of /etc
directory in /home location

tar -cvjf (file_name.tar.bz2 { or -z for gzip file }) (path of that directory where we want to save backup file)

-> “tar” command is used to create backup of any file or directory.

-> With “tar” command, we use some following options:

-c = This option is used to create a new archive.

-v = This option is used for verbosely list files processed (display processing on screen)

-f = This option is used for creating backup in a single file.

-j = This option is used for .bzip2 compression technique

or -z = This option is used for gz compression technique.

-> After providing the option, We provide the file name of .tar file and after this we provide path of that directory where we want to save .tar file.

```
root@servera ~]# tar -cvjf /home/backup.tar.bz2 /etc
```

```
[root@servera ~]# tar -cvzf /home/backup.tar.gz /etc
```

check:- **ls (path of that directory where we saved the backup file)** -> “ls” command is used to show file of any directory

```
[root@servera ~]# cd /home/  
[root@servera home]# ls  
backup.tar.bz2  backup.tar.gz  john  simone  student
```

System info:-

```
root@servera:~  
File Edit View Search Terminal Help  
[root@servera ~]# lsblk  
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT  
vda          252:0    0   10G  0 disk  
└─vda1       252:1    0   10G  0 part /  
vdb          252:16   0    5G  0 disk  
└─vdb1       252:17   0    1G  0 part  
    └─test-test1 253:0    0  252M  0 lvm  /test  
vdc          252:32   0    5G  0 disk
```

In Exam we have three disk.

vda

vdb

vdc

We have to use only vdb and vdc.vda is reserved for system.

Vdb there is already a lvm mounted.In exam there is 4 questions of partitions.[swap , lvm create, lvm extend. (vdo or stratis)]

swap and lvm create is performed on vdb. Vdo or stratis is performed on vdc.

For LVM extend there is already a lvm partition created on vdb you have to extend that partition.

Create a SWAP partition of 250 megabyte and make available at next reboot. Partition already available.

```
Command (m for help):
```

mkswap (path of swap type partition) -> "mkswap" is command to format new swap partition by which we can activate the swap partition on run time

```
[root@servera ~]# mkswap /dev/vdb2
Setting up swapspace version 1, size = 250 MiB (262139904 bytes)
no label, UUID=95d3f182-57e5-403f-9048-2b87a765c2c6
```

with blkid command check uuid of partition and for mount permanently entry in /etc/fstab.

```
[root@servera ~]# blkid
/dev/vda1: UUID="884f47c9-a69d-4c5b-915d-6b7c9c74c923" TYPE="xfs" PARTUUID="16a1e057-01"
/dev/vdb1: UUID="mBogTP-34QN-qDPD-xyUN-y7YJ-ycL5-dc2aqG" TYPE="LVM2_member" PARTUUID="35c52092-01"
/dev/mapper/test-test1: UUID="6f28d683-6ebe-4819-b456-be105ae9ecfd" TYPE="ext4"
/dev/vdb2: UUID="95d3f182-57e5-403f-9048-2b87a765c2c6" TYPE="swap" PARTUUID="35c52092-02"
[root@servera ~]# vim /etc/fstab
```

```
UUID=884f47c9-a69d-4c5b-915d-6b7c9c74c923 / xfs defaults 0 0
UUID="6f28d683-6ebe-4819-b456-be105ae9ecfd" /test ext4 defaults 0 0
UUID="95d3f182-57e5-403f-9048-2b87a765c2c6" swap swap defaults 0 0
```

swapon (path of swap type partition) -> "swapon" command is used to active swap partition on run time.

free -m -> "free" command is used to check or display free and used amount of memory in system. -> "-m" option is used to display free and used amount of memory in system in Megabyte(mb).

```
[root@servera ~]# swapon -a
[root@servera ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda         252:0    0   10G  0 disk
└─vda1      252:1    0   10G  0 part /
vdb         252:16    0    5G  0 disk
└─vdb1      252:17    0    1G  0 part
   └─test-test1 253:0    0  252M  0 lvm  /test
└─vdb2      252:18    0  250M  0 part [SWAP]
vdc         252:32    0    5G  0 disk
vdd         252:48    0    5G  0 disk
[root@servera ~]#
```


Create the volume group with name myvol with 8 MiB P.E. and create the lvm name mydatabase with the 100 P.E. and format this lvm with vfat and create a directory /database and mount this lvm permanently on /database

```
Command (m for help):
```


First:- **pvcreate (path of created partition of LVM type)** -> With help of “pvcreate” command, we’re creating a physical volume of LVM type partition by which we can create volume group of physical volume

```
[root@servera ~]# pvcreate /dev/vdb3  
Physical volume "/dev/vdb3" successfully created.
```

Second:- **vgcreate -s (size of P.E.) (V.G. name) (path of P.V. partition)** -> With help of “vgcreate” command, we’re creating volume group by physical volume partition and we can add one or more physical volume in a single volume group to increase size of volume group by other command.

-> After the “vgcreate” command, we use “-s” option by which we can set the physical extent size on physical volumes of specified volume group. By default, P.E. size is 4Mb.

-> We can check the all created volume group by “vgdisplay” command.

```
Physical volume "/dev/vdb3" successfully created.  
[root@servera ~]# vgcreate -s 8M myvol /dev/vdb3  
Volume group "myvol" successfully created
```

Third:- **lvcreate -L (size of L.V.) -n (name of L.V.) (path of V.G.)** -> With help of “lvcreate” command, we’re creating logical volume from volume group.

-> After the command, we use some following options- -L = This option is used to provide size of new logical volume. -n = This option is used to give name of new logical volume.

-> We can check the all created logical volume by “lvdisplay” command.

```
[root@servera ~]# lvcreate -L 800M -n mydatabase myvol  
Logical volume "mydatabase" created.
```

Fourth:- **mkfs.(file_system) (path of newly created logical volume partition)** -> “mkfs” command is used to make file system of any partition and by separating “.”, we provide file system which we want to provide to new logical volume partition. After the command, we give path of that L.V. partition which we want to format.

```
[root@servera ~]# mkfs.vfat /dev/myvol/mydatabase  
mkfs.fat 4.1 (2017-01-24)
```

Now make directory where the question is asking for mount:-

```
root@servera ~]# mkdir /database
```

Now use command blkid to copy uuid and for permanently mount entry in /etc/fstab.

```
root@servera ~]# blkid
/dev/vda1: UUID="884f47c9-a69d-4c5b-915d-6b7c9c74c923" TYPE="xfs" PARTUUID="16a1e057-01"
/dev/vdb1: UUID="mBogTP-34QN-qDPD-xyUN-y7YJ-ycL5-dc2aqG" TYPE="LVM2_member" PARTUUID="35c52092-01"
/dev/mapper/test-test1: UUID="6f28d683-6ebe-4819-b456-be105ae9ecfd" TYPE="ext4"
/dev/vdb2: UUID="95d3f182-57e5-403f-9048-2b87a765c2c6" TYPE="swap" PARTUUID="35c52092-02"
/dev/vdb3: UUID="wuobHW-1jeI-gS1F-GHCm-Whz4-BRFp-cyFcZH" TYPE="LVM2_member" PARTUUID="35c52092-03"
/dev/mapper/myvol-mydatabase: UUID="35ED-E9EB" TYPE="vfat"
root@servera ~]#
```

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

```
# units generated from this file.
#
UUID=884f47c9-a69d-4c5b-915d-6b7c9c74c923 / xfs defaults 0 0
UUID="6f28d683-6ebe-4819-b456-be105ae9ecfd" /test ext4 defaults 0 0
UUID="95d3f182-57e5-403f-9048-2b87a765c2c6" swap swap defaults 0 0
UUID="35ED-E9EB" /database vfat defaults 0 0
```

then mount -a -> We have to reboot our system or run "mount -a" command by which partition will be mount on specified mount point. In this command, "-a" is used of mount all that partition on specified mount point which we specified in "/etc/fstab" file for permanent mounting.

```
root@servera ~]# mount -a
root@servera ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda                                 252:0    0   10G  0 disk
├─vda1                             252:1    0   10G  0 part /
└─vdb                               252:16   0    5G  0 disk
   ├─vdb1                           252:17   0    1G  0 part
   │   └─test-test1                 253:0    0  252M  0 lvm  /test
   └─vdb2                           252:18   0  250M  0 part [SWAP]
      └─vdb3                         252:19   0   807M  0 part
         └─myvol-mydatabase          253:1    0   800M  0 lvm  /database
```

Resize the Lvm partition "home" to 350MiB

File Edit View Search Terminal Help

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

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
vda	252:0	0	10G	0	disk	
└─vda1	252:1	0	10G	0	part	/
vdb	252:16	0	5G	0	disk	
└─vdb1	252:17	0	1G	0	part	
└─test-test1	253:0	0	252M	0	lvm	/test
└─vdb2	252:18	0	250M	0	part	[SWAP]
└─vdb3	252:19	0	807M	0	part	
└─myvol-mydatabase	253:1	0	800M	0	lvm	/database
vdc	252:32	0	5G	0	disk	
vdd	252:48	0	5G	0	disk	

```
[root@servera ~]# lvresize -L 352M /dev/test/test1
```

```
Size of logical volume test/test1 changed from 252.00 MiB (63 extents) to 352.00 MiB (88 extents).
```

```
Logical volume test/test1 successfully resized.
```

```
[root@servera ~]# resize2fs /dev/test/test1
```

```
resize2fs 1.44.3 (10-July-2018)
```

```
Filesystem at /dev/test/test1 is mounted on /test; on-line resizing required
```

```
old_desc_blocks = 2, new_desc_blocks = 3
```

```
The filesystem on /dev/test/test1 is now 360448 (1k) blocks long.
```

```
[root@servera ~]# lvextend -L 452M /dev/test/test1
```

```
Size of logical volume test/test1 changed from 352.00 MiB (88 extents) to 452.00 MiB (113 extents).
```

```
Logical volume test/test1 successfully resized.
```

```
[root@servera ~]# resize2fs /dev/test/test1
```

```
resize2fs 1.44.3 (10-July-2018)
```

```
Filesystem at /dev/test/test1 is mounted on /test; on-line resizing required
```

```
old_desc_blocks = 3, new_desc_blocks = 4
```

```
The filesystem on /dev/test/test1 is now 462848 (1k) blocks long.
```

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

lvextend/lvresize -L (required_size in Kb or Mb or Gb) (path_of_LVM_partition)

-> "lvextend" command is used to extend lvm partition with required size of partition.

-> OR "lvresize command is used to resize of lvm partition "

-> With this command, we use "-L" option which is used to extend or set size of partition in form of kilobyte, megabyte or gigabyte and many more. If we use "+" sign with value then value is added in actual size of partition but if we not use "+" sign with value then value is set as actual size of LVM partition.

resize2fs (path_of_LVM_partition) -> "resize2fs" command is used to resize the ext2/ext3/ext4 file system after extend the size of LVM partition. It's used to enlarge or shrink a mounted or unmounted file system located in system.

if lvm is formatted with xfs then we have command:-
xfs_growfs (mount point of lvm)

You have been provided with a disk drive attached to your system, make use of it to create a VDO device with a logical size of 50GB. format this with xfs format and mount on /mnt/vdo0.

In exam there is disk given in exam(/dev/vdc) there you have to perform vdo task:-
First you have to install two packages:- vdo , kmod-kvdo

```
[root@server1 ~]# yum install -y vdo kmod-kvdo
```

Now open man page of vdo:- `man vdo`>>>example>> there is command to create vdo >> copy that command and changes as per questions like:- `vdoname device` and `vdoLogicalSize`
then format the vdo partition in xfs filesystem with `-K` option.

EXAMPLES

Creation of a VDO device named `vdo0`, with a 10 terabyte thinly-provisioned logical address size:

```
# vdo create --name=vdo0 --device=/dev/sdb1 --vdoLogicalSize=10T
Creating VDO vdo0
Starting VDO vdo0
Starting compression on VDO vdo0
VDO instance 1 volume is ready at /dev/mapper/vdo0
#
```

Of course, as with any thinly-provisioned device, it may not hold 10 terabytes of user data even if

```
[root@server1 ~]# man vdo
```

```
[root@server1 ~]# vdo create --name=vdo0 --device=/dev/vdc --vdoLogicalSize=50G
Creating VDO vdo0
Starting VDO vdo0
Starting compression on VDO vdo0
VDO instance 0 volume is ready at /dev/mapper/vdo0
```

```
[root@server1 ~]# mkfs.xfs -K /dev/mapper/vdo0
meta-data=/dev/mapper/vdo0      isize=512    agcount=4, agsize=3276800 blks
      =                       sectsz=4096   attr=2, projid32bit=1
      =                       crc=1        finobt=1, sparse=1, rmapbt=0
      =                       reflink=1
data      =                       bsize=4096   blocks=13107200, imaxpct=25
      =                       sunit=0       swidth=0 blks
naming    =version 2           bsize=4096   ascii-ci=0, ftype=1
log       =internal log       bsize=4096   blocks=6400, version=2
      =                       sectsz=4096   sunit=1 blks, lazy-count=1
realtime  =none               extsz=4096   blocks=0, rtextents=0
[root@server1 ~]#
```