

# Lab Environment

Students should log in to Physical Computer (**FoundationX**).

You will have to access virtual machines to complete tasks

The virtual machine **servera** has the hostname **servera.lab.example.com**  
The virtual machine **serverb** has the hostname **serverb.lab.example.com**

Both machines have a standard user account, **student** with the password **student** and the **root password is redhat.** ldapuserX password is password.

**Dark Color Highlights are Imp Points....Light Color Highlights are Optional**

servera:

**Task 1:** Configure network and set the static hostname

IPADDR	:	172.25.250.10
NETMASK	:	255.255.255.0
DEFAULT GATEWAY	:	172.25.250.254
NAME SERVER	:	172.25.250.220
HOST NAME	:	servera.lab.example.com

**Task 2:**

This task is very important to install packages

**Create Repository**

**Create Repository for [http://content.example.com/rhel9.0/x86\\_64/dvd/BaseOS](http://content.example.com/rhel9.0/x86_64/dvd/BaseOS) with the name BaseOS**

**Create Repository for [http://content.example.com/rhel9.0/x86\\_64/dvd/AppStream](http://content.example.com/rhel9.0/x86_64/dvd/AppStream) with the name AppStream**

**Configure YUM repos with the given link ( 2 repos: 1st is Base and 2nd is AppStream )**

```
vim /etc/yum.repos.d/rhcsa.repo
[BaseOS]
name=BaseOS
baseurl= http://content.example.com/rhel9.0/x86\_64/dvd/BaseOS
gpgcheck=0
enabled=1

[AppStream]
name=AppStream
baseurl= http://content.example.com/rhel9.0/x86\_64/dvd/AppStream
gpgcheck=0
enabled=1
# yum repolist
```

**Task 3:**

Debug SELinux:

- A web server running on non standard port 82 is having issues serving content. Debug and fix the issues.
- \* The web server on your system can serve all the existing HTML files from /var/www/html \* Web service should automatically start at boot time.

```
#systemctl status httpd
#firewall-cmd - -list-all
#man semanage port
# semanage port -a -t http_port_t -p tcp 82
# systemctl start httpd
#systemctl status httpd
# firewall-cmd --list-ports
# firewall-cmd --add-port=82/tcp
# firewall-cmd --add-port=82/tcp --permanent
```

**Task 4:**

create the following user, groups, and group memberships:

- A group named sysadmin
- A user saara who belongs to sysadmin as a secondary group
- A user natasha who also belongs to sysadmin as a secondary group
  
- A user harry who does not have access to an interactive shell on system and who not a member of sysadmin
- saara,natasha and harry should all have the password of “Postroll”

```
#groupadd sysadmin  
  
#useradd -G sysadmin saara  
  
#useradd -G sysadmin natasha  
  
#useradd -s /sbin/nologin harry  
  
#echo Postroll | passwd --stdin saara  
  
#echo Postroll | passwd --stdin natasha  
  
#echo Postroll | passwd --stdin harry
```

**Task 5:**

Create a user deal with userid 2015. and assign the password Postroll

```
# useradd -u 2015 deal  
# echo Postroll | passwd deal
```

**Task 6:**

create a collaborative directory /shared/sysadmin with the following characteristics:

- Group ownership of /shared/sysadmin is sysadmin
- The directory should be readable, writable, and accessible to members of sysadmin, but not to any other user.
- Files created in /shared/sysadmin automatically have group ownership set to the sysadmin group

```
#mkdir -p /shared/sysadmin  
#chgrp sysadmin /shared/sysadmin  
#chmod 2770 /shared/sysadmin  
#ls -ld /shared/sysadmin
```

**Task 7:**

Case 1:

Search the files with extension .pl and copy the files to directory /root/pl.found  
sol:

```
# mkdir /root/pl.found  
# find / -name *.pl -exec cp {} /root/pl.found/ \;
```

Case 2:

Find a .conf extension files in /etc and copy files to /search

sol:

```
# mkdir /search  
# find /etc -name *.conf -exec cp -a {} /search/ \;
```

Case 3:

Search student user files and copy the files to directory /root/student.found

sol:

```
#mkdir /root/student.found  
# find / -user student -exec cp -rpf {} /root/student.found/ \;
```

**Task 8:**

- a. Search the keyword “sa” in /usr/share/dict/words file and store the lines in order in /root/sa.txt

Sol:

```
#grep -i sa /usr/share/dict/words | sort > /root/sa.txt
```

```
#cat /root/sa.txt
```

- b. Search the keyword “sa” in /usr/share/dict/words file and store the lines in the same order /root/sa.found

Sol:

```
#grep -i sa /usr/share/dict/words > /root/sa.found
```

```
#cat /root/sa.found
```

- c. search the string saara in the /etc/passwd file and save the output in /root/lines

Sol:

```
#grep saara /etc/passwd > /root/lines
```

```
#cat /root/lines
```

**Task 9:**

- a. Backup /usr/share directory to /root/usr.tar.gz or /root/usr.tgz

Sol:

```
# tar czvf /root/usr.tar.gz /usr/share (or)
```

```
# tar czvf /root/usr.tgz /usr/share
```

- b. Backup /usr/share directory to /root/usr.tar.bz2

Sol:

```
# tar cjvf /root/usr.tar.bz2 /usr/share (i.e bzip2) (small 'j')
```

- c. Backup /usr/share directory to /root/usr.tar.xz

Sol: (Hint man tar)

Sol:

```
# tar cfvf /root/usr.tar.xz /usr/share (Capital 'J')
```

**Task 10:**

**Type 1:**

The user saara must configure a cron job that runs daily at 5:30 PM local time and executes /bin/echo hello.

```
# su - saara  
$ crontab -e  
30 17 * * * /bin/echo hello
```

(Note: 24 hrs Clock ---- 17:30 )

```
$crontab -l
```

**Type 2:**

The user saara must configure a cron job that runs daily for every 2mins localtime and executes logger "ex200 prep"

```
# su - saara  
$ crontab -e  
*/2 * * * * logger "ex200 prep"  
$crontab -l
```

**Task 11:**

Configure your system so that it is an NTP client of

classroom.example.com

Note: Make your system NTP client to classroom.example.com (In exam host.domainX.example.com)

```
#yum install chrony -y  
#vim /etc/chrony.conf  
(comment if any server line found)  
server classroom.example.com iburst  
#systemctl restart chronyd  
#systemctl enable chronyd  
To Verify  
chronyc sources -V
```

**Task 12:**

Configure autofs to automount the home directories of LDAP users.

Note the following:

a.classroom.example.com NFS-exports /home/guests to your system

b.ldapuserX home directory is, classroom.example.com:/home/guests/ldapuserX

c.ldapuserX's home directory should be automounted locally beneath /home as /home/guests/ldapuserX.

d.Home directories must be writable by their users.

```
# yum install autofs -y
```

```
# showmount -e classroom.example.com (Based on given path)
```

/home/guests (You can find this mount point)

```
# vim /etc/auto.master
```

--> In this file copy line "/misc /etc/auto.misc" and paste it and modify the line as follows , but don't delete this line

```
/home/guests /etc/auto.misc
```

( In Exam /netdir or /rusers/ldapuserX, verify info using showmount command)

```
# vim /etc/auto.misc
```

--> Copy first full line starting with "#linux" and paste at end of the file, modify it as follow

```
Idapuser4 -rw,sync,fstype=nfs4 classroom.example.com:/home/guests/Idapuser4
```

( In Exam /netdir or /rusers/ldapuserX)

```
# systemctl enable --now autofs.service
```

To Verify:

```
#ssh Idapuser4@localhost
```

```
[Idapuser4@serverX ~]$ df -hT
```

**Task 13:**

Build **monitor** container image with the given Containerfile

<http://materials.example.com/Containerfile>

Create a container named monitor

**Sol:**

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

ctrl +d --> To exit

Note: You must do ssh to the user, don't switch user su - containers

In Exam read clearly about username to execute container tasks in additional info

ssh containers@servera

```
[containers@servera ~]$
```

Log in to the container registry using the podman login command.

(In exam, Read Additional Info to understand registry server details, username and password)

```
[containers@servera ~]$ podman login registry.lab.example.com
```

Username: admin

Password: redhat321

ssh containers@servera

```
[containers@servera ~]$ wget http://materials.example.com/Containerfile
```

\$ls -l

podman build -t monitor .

podman images

podman run -d --name monitor monitor

Note: loginctl enable-linger <username>

loginctl enable-linger containers

podman ps

**Task 14:**

Configure your host journal /srv/log to store all journal across reboot

Configure persistent volume /var/log/ from logserver container to /srv/log/ when container starts  
create container-logserver.service to run container persistently.

(In Exam you can find 2 volumes to map)

[containers@servera ~]\$**sudo -i or su -**

If given dirs doesn't exists, create it otherwise check ownership of the folder

```
#mkdir -p /srv/log  
#chown containers. /srv/log/
```

ctrl+d --> To exit

[containers@servera ~]\$

```
podman run -d --name logserver -v /srv/log:/var/log monitor
```

Optional : depends on the Question

```
-v /srv/log:/var/log -v /srv/data:/var/data
```

```
podman ps
```

```
mkdir -p .config/systemd/user
```

```
cd .config/systemd/user
```

[containers@servera user]\$

```
podman generate systemd --help
```

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

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

```
systemctl --user list-unit-files
```

```
systemctl --user enable container-logserver.service
```

[containers@servera ~]\$ sudo reboot / su - and then

#reboot

[student@workstation ~]\$ ssh containers@servera

[containers@servera ~]\$ podman ps

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS NAMES		
090e44c47a55	localhost/logserver:latest	/bin/rsyslog.sh	11 seconds ago
			Up 9 seconds ago
			logserver

**Task 15:**

Write a script file to copy all the files which are less than 10 Mb in size from /usr/share to /root/find and set gid bit on all those files

sol:

```
vim script.sh
#!/bin/bash
mkdir /root/find
find /usr/share -type f -size -10M -exec cp -rpf {} /root/find/ \;
chmod -R g+s /root/find
:wq!
chmod a+x script.sh
./script.sh
```

(or)

Write a script file to copy all the files which are greater than 30 kb in size and less than 70kb in size from /boot to /root/entersoft and set gid bit on all those files

sol:-

```
vim script.sh
#!/bin/bash
mkdir /root/entersoft
find /boot -type f -size +30k - size -70k -exec cp -rpf {} /root/ entersoV/ \;
chmod -R g+s /root/entersoV/ :wq!
chmod a+x script.sh
./script.sh
```

**Task 16:**

Files created by saara should have permissions as -> r----- on files and  
Directories created by saara should have permissions as -> d r-x --- on directories.

#su - saara

\$ls -la  
\$vim .bashrc  
Go to end of the file and add  
  
umask 277  
:wq!  
exit and login (or) source .bashrc

Test:

\$umask

**Task 17:**

Password policy

All local users must have password expiry for 20days (set expiry for 20 days)

To change for the particular user:

#chage -l <username>

Check present password expiry period and other password policy details by executing above commands

#chage -M 20 <username>

#chage -l <username>

To change for all future users:

vim /etc/login.defs

Go to this file and change

PASS\_MAX\_DAYS 20

:wq!

NOTE - changes made to login.def will have its effect on new users but it won't effect already existing users ..

So please read given questions in exam carefully and then answer them accordingly.

**Task 18:**

Case 1: All the users in production group should be able to run root user commands with sudo privileges without password prompt.

```
#visudo
```

Go to Line “## Same thing without a password” (Note – don’t delete any existing line, shift+g,o To Goto last line of the file)

Follow example from file

```
# %wheel ALL=(ALL) NOPASSWD: ALL ----> This Line already exists, Refer this, don't uncomment
```

```
%production ALL=(ALL) NOPASSWD: ALL
```

```
:wq!
```

Case 2: Configure user natasha should be able to run root user commands with sudo privileges without password prompt.

```
#visudo
```

Go to Line “## Same thing without a password” (Note – don’t delete any existing line, shift+g,o To Goto last line of the file)

Follow example from file

```
# %wheel ALL=(ALL) NOPASSWD: ALL ----> This Line already exists, Refer this, don't uncomment
```

Note: Don't use % for user

```
natasha ALL=(ALL) NOPASSWD: ALL
```

```
:wq!
```

**Task 19:**

Configure user saara so that she will be able to use sudo command

```
#usermod -aG wheel saara
```

```
#id saara
```

**Do it in serverb**

view console, we can't login without cracking root password

NOTE: In this Machine We can see 3 Disks

1. ./dev/vda
2. ./dev/vdb
3. ./dev/vdc

for ROOT filesystem

You need to use for Swap and LVM Partition. Will be used for vdo(RHEL8).

**Task 1:**

Cracking Root Password:

Click view console ---> Right corner press power button to restart. While restarting ....

Press 'e' ---> At the time of splash screen display ---> Go to linux16 line (Using Down Arrow) --->

Press "End" Button---> then press spacebar then type then press 'ctrl + x' to continue

```
rd.break console=tty1  
#mount -o remount,rw /sysroot  
#chroot /sysroot  
#passwd --stdin root  
(Enter given password 'redhat')  
#touch /.autorelabel  
#exit  
#exit
```

(Don't Forget to relabel , required for /etc/shadow relabel) (Exit from sysroot jail) (Exit from initramfs debug shell) will restart Automatically)

**Task 2:**

This task is very important to install packages

**Create Repository**

Create Repository for [http://content.example.com/rhel9.0/x86\\_64/dvd/BaseOS](http://content.example.com/rhel9.0/x86_64/dvd/BaseOS) with the name  
BaseOS

Create Repository for [http://content.example.com/rhel9.0/x86\\_64/dvd/AppStream](http://content.example.com/rhel9.0/x86_64/dvd/AppStream) with the name

**Configure YUM repos with the given link ( 2 repos: 1st is Base and 2nd is AppStream )**

```
vim /etc/yum.repos.d/rhcsa.repo
[BaseOS]
name=BaseOS
baseurl= http://content.example.com/rhel9.0/x86\_64/dvd/BaseOS
gpgcheck=0
enabled=1

[AppStream]
name=AppStream
baseurl= http://content.example.com/rhel9.0/x86\_64/dvd/AppStream
gpgcheck=0
enabled=1
# yum repolist
```

**Task 3:**

Add the SWAP Space with a size of 754 MiB

```
# lsblk
```

(To see ur device... /dev/vda (or) /dev/vdb (or) /dev/sda, In our Lab: /dev/vdb)

```
#fdisk /dev/vdb
:p (print)
```

Observe it is GPT, you can create 128 primary partitions

```
> n
>+754M
:press t > partition number >19 (if rhel9 --> 19)
:w
```

```
#partprobe /dev/vdb
#udevadm settle
#partprobe
#lsblk
```

EntersoftLabs Internal Document to Practice Lab for RH124 and 134 Tasks...Prepared by Nityanand Sir  
(Identify Newly created partition with 802Mib...Identify “**X**”. Don’t do any mistake here , number is important))

#mkswap /dev/vdb**X** ---> Make the swap on newly created Partition

**#blkid ( To see and copy UUID)**

#vim /etc/fstab -----> Create the mount point inside the fstab  
UUID=----- **swap swap defaults 0 0**

#swapon -a (To Activate fstab entries)

**To Verify:**

#free -m

#lsblk

**Task 4:**

Create a Logical Volume with the name **share** by using **100 Extents**. From the volume group **sharegroup** of **8MiB** extend size.

And mount it under **/mnt/share** with **vfat** File system.

Sol:

use bc in the new terminal, and then calculate

**(First Determine Size : 100 extents x 8 MiB = 800 MiB, while creating add +8 MiB for Exact Size  
Note: you can add +4 MB or +8 MB or +16 MB or +32 MB based on extent size)**

#lsblk

(Read Available Device to execute fdisk)

**Step 1: Create Partition and change type to LVM (8e)**

#fdisk /dev/vdb

:p (print) > n

>>+808M

Then Change Type

:press **t** >partition number > 30

:w

#partprobe /dev/vdb

#udevadm settle

#lsblk (Identify Newly created partition with 802Mib...Don’t do mistake here)

## Step 2: Create LVM with the Name Share

```
#pvcreate /dev/vdbX (X or Identify the Number from lsblk
```

```
#vgcreate -s 8M sharegroup /dev/vdbX
```

(Note: By default Physical Extent Size i.e PE is 4 Mib...>

Check with

```
#vgdisplay sharegroup
```

this command will show PE Size info as 8Mib

```
#lvcreate -n share -l 100 sharegroup
```

(Note: In Question, Number Extents required 100..i.e  $100 \times 8 = 800$ Mib, since PE Size is 8 MiB)

(small l represent size in extents and Capital L represent size in MB)(Should be very careful)

```
#lvdisplay /dev/sharegroup/share (To check Size and info)
```

## Step 3: Write File System based on Objective

```
#mkfs.vfat /dev/sharegroup/share
```

```
#mkdir /mnt/share
```

```
#blkid -----> Get the UUID of it
```

```
#vim /etc/fstab -----> Create the mount point inside the fstab
```

```
UUID=----- /mnt/share vfat defaults 0 0
```

```
#mount -a
```

```
#df -hT ----> verify
```

**Task 5:**

Resize the Logical volume /dev/mapper/myvol-vo to 100MiB.

NOTE: Size in between 90M and 110M is acceptable.

Sol:

Note: If it is resize or new size keyword found, then check

```
#lvdisplay /dev/mapper/myvol-vo
```

resize cases:

Case 1:

If the resize or new size is greater than existing size, then use

```
#lvextend -r -L 100M /dev/mapper/myvol-vo      (If Size,Exact Size Rounded)
```

```
#lvextend -r -l 100 /dev/mapper/myvol-vo        (If extents,To round to 100 extents)
```

Case 2:

If the resize or new size is lesser than existing size, then use

```
#lvreduce -r -L 100M /dev/mapper/myvol-vo      (If Size,Exact Size Rounded)
```

```
#lvreduce -r -l 100 /dev/mapper/myvol-vo        (If extents,To round to 100 extents)
```

Note: If it is add size / add no.of extents keyword found

```
#lvextend -r -L +100M /dev/mapper/myvol-vo     (To Add 100M to Current Size)
```

```
#lvextend -r -l +100 /dev/mapper/myvol-vo       (To add 100 extents)
```

Note: If it is remove or subtract size / remove or subtract no.of extents keyword found

```
#lvreduce -r -L -100M /dev/mapper/myvol-vo     (To reduce 100 M from Current Size)
```

```
#lvreduce -r -l -100 /dev/mapper/myvol-vo       (To subtract 100 extents)
```

**Task 6:**

Configure System Tuning:

- Choose the recommended 'tuned' profile for your system and set it as the active.

```
# rpm -q tuned  
# yum install tuned (If it is not installed)  
  
# systemctl enable tuned  
  
# systemctl start tuned  
  
# tuned-adm active  
# tuned-adm recommend  
# tuned-adm profile <copy and paste from the above>  
  
# tuned-adm active
```

***For Experts, others can skip these tasks, Self Study Tasks***

***Practice LVMs to Become Expert for Real Time Scenarios***

**Type 1:**

Create a Logical volume **fedora** with **60 extents** from a volume group **redhat** of **16MB extent size**. Mount it under /mnt/fedora with **xfs** file system persistently.

**Type 2:**

Create a new logical volume according to the following requirements.

- \* The **logical volume** is named **engineering** and belongs to the **development** volume group and has a size of **20 extents**
  - \* logical volume in development volume group should have an **extend size of 32MiB**.
  - \* Format the new logical volume with the **ext3** file system. The logical volume should be automatically mounted under /mnt/engineering at system boot time.

**Type 3:**

Create a Logical volume **ubuntu** with **250MB** from a volume group **debian** of **1000MB** size.

Mount it under /mnt/ubuntu with ext4 file system persistently.

**Type 4: (Before doing this task create lvm using type 3 task above)**

- a. Resize the logical volume **ubuntu** so that it should be in between 450MB to 500MB
- b. Resize the logical volume **ubuntu** so that it should be in between 130MB to 150MB