REDHAT ENTERPRISE LINUX ADMINISTRATION

Contents

Module 1: Setting up Training Lab and Course Introduction

Installing CentOS-8(RHEL-8) Virtual Machine using Oracle VirtualBox

1. Download and install Oracle VirtualBox and Oracle VirtualBox Extension Pack
   https://www.virtualbox.org/wiki/Downloads

2. Create an Account in the RHEL Developer Portal and Download RHEL 8 from the Developer Portal
   https://developers.redhat.com/products/rhel/download
   *Take note of your RHEL Developer Account Name and Password, we will use it for registration and activation

3. Create a new virtual machine in virtualbox
   - ✓ Redhat 64-bit
   - ✓ 2048-4096MB RAM
   - ✓ 50GB Disk Storage

4. Install RHEL 8 developer into the new vm using the RHEL8 dvd iso
   - ✓ Server with a gui
   - ✓ No additional server components
   - ✓ Set timezone to Asia Manila Philippines
   - ✓ No partitioning
   - ✓ Create new user
   - ✓ Set root password

5. Setup virtualbox network to bridge
6. Enable network interface in RHEL8 after login
7. Search→subscriptions
   - ✓ Register RHEL8
   - ✓ Attach Subscription
   - ✓ Using cli:

```
# subscription-manager register --username <username> --password <password> --auto-attach
```

If the command is unable to attach a subscription, it will indicate that in the output. Then, you can attach the subscription from the Customer Portal instead:

```
# subscription-manager register
# subscription-manager attach --auto
# subscription-manager refresh
```

8. [*Optional but recommended] perform RHEL8 update and install virtualbox guest additions
   # yum update -y
   # dnf install tar bzip2 kernel-devel-$(uname -r) kernel-headers perl gcc make elfutils-libelf-devel
   - Install virtualbox guest additions
   - Set poweroptions to never blank screen when idle

9. Clone the VM
10. Edit hostname and Virtualmachine mac address

Get new repos for centos 8 after dec 31 2021:

```
# cd /etc/yum.repos.d/
# sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-*
# sed -i 's|#baseurl=
```

Setting root password (Interrupting Boot Process)

Procedure:

1. Start the system
2. Wait for grub menu, press e to edit
3. Find the line starting with linux and enter the rd.break at the end
4. Press ctrl+x to boot the system with these kernel boot parameters
5. At this stage, root file system is mounted in read-only mode to /sysroot and must be remounted with r/w permissions.
6. To mount the /sysroot with r/w permissions:

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

7. Switch /sysroot to / file system

```
# chroot /sysroot
```

8. To set root password

```
# passwd
```

9. To relabel the selinux contexts

```
# touch /.autorelabel
# exit
# exit
```

10. Reboot normally and Test root password

Configuring SELinux to enforcing mode

Procedure:

1. Check selinux mode

```
# getenforce
```

2. Set the variable selinux=enforcing using vim

```
# vim /etc/selinux/config
```

3. Restart

```
# systemctl reboot
```

4. Recheck status of selinux

```
# sestatus      Or      # getenforce
```

5. Manual page of selinux

```
# man selinux
```

Introducing Software Repositories in RHEL-8 (CentOS-8)

In rhel8, packages are distributed through 2 different repos: BaseOS and AppStream
BaseOS: Contains RPM packages which provide core functionality needed by all installations
AppStream: Contains RPM packages for user space applications

DNF (Dandified Yum, Yum version 4) is package manager for RPM based packages, faster than old yum and is the default package manager for RHEL 8.

```
# man dnf
```

YUM (Yellowdog Updater Modifier) is an older package manager

```
# man yum
```

Setting Up Local YUM/DNF Repositories

Set up local yum (dnf) repositories , BaseOS and AppStream at location /repo/BaseOS and /repo/AppStream on System. Repo should be created with yum group information to use yum group.

Command	Action/Description
dnf repolist	To list repositories configured on system
dnf grouplist hidden	To list available group packages
mkdir -p /repo/BaseOS /repo/AppStream	To create directories for repositories
cp -irv /run/media/pbajaj/CentOS*/BaseOS/* /repo/BaseOS	Copy packages and repodata for BaseOS
cp -irv /run/media/pbajaj/CentOS*/AppStream/* /repo/AppStream	Copy packages and repodata for AppStream
man dnf	Manual page for dnf

Configuring System to Use Local YUM/DNF Repositories

Configure System to use BaseOS & AppStream repositories present at /repo/BaseOS and /repo/AppStream respectively.

Command	Action/Description
vim /etc/yum.repos.d/system.repo **[BaseOS]** name = BaseOS baseurl = ///repo/BaseOS gpgcheck = 0 enabled =1 **[AppStream]** name = AppStream baseurl = ///repo/AppStream gpgcheck = 0 enabled = 1 :wq	Creating **.repo** file to configure **System** to use repos.
dnf clean all	To clear cache
dnf repolist	To list repositories configured on **System**
dnf grouplist hidden	To list group available packages
man yum.conf	Manual page for **yum.conf**

Introducing Services Configured on IPA Server

IPA (Identity-Policy-Authentication) Server machine is pre-configured to provide below services:

1. DNS Server
   To provide Hostname resolution.

Forward DNS lookup	**Result (Output)**
host system.example.com	192.168.99.10
host ipaserver.example.com	192.168.99.254
Reverse DNS lookup	
host 192.168.99.10	system.example.com
host 192.168.99.254	ipaserver.example.com

*We can also use nslookup or dig for same purpose.

2. 389 Directory Server
   Provides LDAP functionality to host LDAP users. Four users have been created on Directory Server.

Username	**Home Directory**
ldap	/home/ldapuser/ldap
ldap1	/ldap /home/ldap1
ldap2	/ldap /home/ldap2
smb1	/home/smb1/PRINCE BAJAJ 2

3. NTP Server
   To provide time services

Command	**Action/Description**
# vim /etc/chrony.conf	To allow network to use time services
allow 192.168.99.0/24	
:wq	
# systemctl restart chronyd	Restarting chronyd to make changes effective
# firewall-cmd –add-service=ntp --permanent	Configuring firewall to accept inbound traffic

4. NFS Server
   To export LDAP user's Home Directories and one more NFS share

Command	**Action/Description**
# dnf install nfs-utils	Installing nfs server package(s)
# systemctl nfs-server --now enabled	Starting and configuring service to start at boot
# vim /etc /exports	Defining exports in exports file
/nfsshare *(rw)	
/home/ldapuser *(rw)	
/ldap /home *(rw)	
:wq	

# exportfs -arv	Exporting NFS exports
# firewall-cmd –add-service={ nfs,rpc-bind,mound } --permanent	Configuring firewall to accept inbound
# firewall-cmd --reload	Reloading firewall to make configs effective

5. Samba Server

Command	**Action/Description**
# dnf install samba cifs-utils	Installing required packages
# systemctl smb --now enabled	Starting/enabling smb.service
# vim /etc / smb.conf	Defining samba share
[samba]	
comment = samba_share	
path = /samba	
writable = yes	
:wq	
# systemctl restart samba	Restarting smb.service to make changes effective
# smbpasswd -a smb1	Creating samba user profile for smb1 user
# firewall-cmd --add-service=samba --permanent	Configuring firewall to accept inbound traffic
# firewall-cmd --reload	
# semanage fcontext -a -t samba_share_t “/samba(/.*)?”	Setting correct SELinux context type
# restorecon -Rv /samba	Restoring context

Importing IPA Server Appliance

IPA Server:
https://drive.google.com/file/d/1aufrKZJq2955HwvhVL4EEEBezK9wbOfE/view?usp=drivesdk

Unzip and import the IPA Server.ova file in VirtualBox. You can do this by opening file and it will be automatically imported. Wait for 3 4 minutes for import to complete and the start VM. Login to IPA Server using password as password Verify different services provided by IPA Server

Verifying Services Configured on IPA Server

Procedure:

1. Login to the ipa server
2. Use cli:

```
# hostname
# ip address show      or      # ifconfig
```

Check dns

```
# host ipaserver.example.com
# host 192.168.99.254
# host 192.168.99.10          ← this will be assigned to our rhel system later on
```

Check for ldap users

```
# id ldap
# id ldap1
# id ldap2
# id smb1
```

Test switch to ldap user

```
# su - ldap
# id
# pwd
```

Verify ntp server

```
# systemctl status chronyd
```

Check if allow ntp for clients

```
# more /etc/chrony.conf
…
# Allow ntp clients to access from network
allow 192.168.99.0/24
…
```

Check for nfs server

```
# systemctl status nfs-server
```

Check nfs exports file

```
# more /etc/exports
```

Check smb server

```
# systemctl status smb
```

Check some samba config

```
# more /etc/samba/smb.conf
…
[samba]
comment = samba_share
path = /samba
writable = yes
…
```

Check selinux Boolean settings

```
# getsebool -a | grep nfs
…
nfs_export_all_rw → on
…
```

```
# getsebool -a | grep samba
…
samba_export_all_rw → off
```

Check SMB user

```
# su - root
# pdbedit -L
```

Check Firewall configuration

```
# su - root
# firewall-cmd --list-all
```

Putting System and IPA Server on Same Internal Network

Procedure:

1. Set ipa server virtualbox network to internal network
2. Set rhel server virtualbox network to internal network
   (ip: 192.168.99.10, hostname: system.example.com)
3. Must be on same virtualbox internal network name

Module 2: Configuring Server Services

See slides for setting up IPA Server

- Installing DNS and LDAP Services
- Configuring Forward and Reverse DNS Zones & Records
- Creating LDAP Users & Home Directories
- Configuring NTP Server to Allow Client(s) to Use Services
- Configuring NFS Server to Export LDAP User's Home Directories
- Configuring SAMBA Share

Setting up samba server for anonymous access

1. Install Samba4 in RHEL 8

   ```
   # dnf install samba samba-client samba-common
   ```

2. Once the installation is complete, start the Sambe service, enable it to auto-start at system boot time and verify that service using the systemctl commands as follows.

   ```
   # systemctl start smb
   # systemctl enable smb
   # systemctl status smb
   ```

3. Next, if you have a firewalld configured, you need to add the Samba service in the firewall configuration to allow access to shared directories and files through system.

   ```
   $ sudo firewall-cmd --permanent --add-service=samba
   $ sudo firewall-cmd --reload
   ```

4. Create a backup copy of default samba configuration file which comes with pre-configuration settings and various configuration directives.

   ```
   # cp /etc/samba/smb.conf /etc/samba/smb.conf.orig
   ```

5. Setting Up Samba4 Anonymous File Sharing on RHEL 8
   Create the shared directory which will store files on the server. Then define the appropriate permissions on the directory as shown.

```
# mkdir -p /srv/samba/anonymous
# chmod -R 0777 /srv/samba/anonymous
# chown -R nobody:nobody /srv/samba/anonymous
```

6. Next, using the chcon utility, change the SELinux security context for the created samba shared directory.

```
# chcon -t samba_share_t /srv/samba/anonymous
```

7. Now open the configuration file using your favorite text-based file editor to configure the anonymous unsecured file sharing on a shared directory.

```
# vim /etc/samba/smb.conf
```

8. Modify the following global parameters and add a section for the Anonymous share. Note that you can set your own values where necessary (read man smb.conf for more information).

```
[global]
    workgroup = WORKGROUP
    netbios name = rhel
    security = user
...
[Anonymous]
    comment = Anonymous File Server Share
    path = /srv/samba/anonymous
    browsable =yes
    writable = yes
    guest ok = yes
    read only = no
    force user = nobody
```

9. Run the following command to verify if the configuration is correct.

```
# testparm
```

10. If the Samba configuration is OK, go ahead and restart the samba service for the recent changes to take effect.

```
# systemctl restart smb
```

11. Finally, test if the Anonymous share is working fine, log into your Windows machine, open the Windows Explorer, click on Network, then click on the RHEL host, or use the server IP address to access it (running ip add command on the server can help you to view the IP address).

Setting Up Samba4 Secure File Sharing on RHEL 8

1. In order to create a securely shared directory, you need to create a Samba system group. All users of the secured share will be added to this group. You can use the groupadd command to create the group as follows.

```
# groupadd smbgrp
```

2. Then use usermod command to add all users, for example, “user1” to the group and set a password for each user as shown.

```
# usermod user1 -aG smbgrp
# smbpasswd -a password123
```

3. Next, create the secure directory which will securely store shared files, then set the appropriate permissions on the directory. Also, change the SELinux security context for the directory as follows.

```
# mkdir -p /srv/samba/secure
# chmod -R 0770 /srv/samba/secure
# chown -R root:smbgrp /srv/samba/secure
# chcon -t samba_share_t /srv/samba/secure
```

4. Next, open the configuration file for editing.

```
# vim /etc/samba/smb.conf
```

And add the following section at the end of the file.

```
…
[Secure]
    comment = Secure File Server Share
    path =  /srv/samba/secure
    valid users = @smbgrp
    guest ok = no
    writable = yes
    browsable = yes
```

Save the changes and close the file.

5. Next, verify the samba configuration again, by running the testparm command.

```
# testparm
```

6. Restart Samba services to apply the changes.

```
# systemctl restart smb.service
# systemctl restart nmb.service
```

7. Lastly, test if the Secure share is working fine. From your Windows machine, open the Windows Explorer, click on Network, then click on the RHEL host, or else try to access the server using its IP address as explained before. You'll be asked to enter your username and password to login the RHEL 8 server.

   Note: User credentials are cached so succeeding access won't ask for re-logins. You may clear the windows cache by restarting the "workstation" service in Windows Services (services.msc)

Module 3: Deploying, Configuring and Maintaining Services

Assigning Hostname

Assign hostname system.example.com to System machine.

Command	**Action/Description**
# hostnamectl or hostname	To display current hostname assigned to System
# hostnamectl set-hostname system.example.com	To assign new hostname to System
# man hostnamectl	To check manual page for hostnamectl

Configuring Ethernet Connection with Static IPv4 Address and DNS IP

Configure IP Address 192.168.99.10 on ethernet interface enp0s3 on system.example.com and set the DNS IP Address as 192.168.99.254. Configure the Default Gateway as 192.168.99.1. IP assigned must be static.

Command	Action/Description
nmcli connection show	To display existing connections configured on different interfaces and their status
ip address	To display existing connections with configured IP Address(s) and status
nmcli connection add con-name system type ethernet ifname enp0s3 ipv4.addresses 192.168.99.10/24 ipv4.gateway 192.168.99.1	To configure ethernet interface with IPv4 Address and Gateway IP Address
nmcli connection modify system ipv4.dns 192.168.99.254 ipv4.method manual	To configure DNS IP Address on connection and make connection static
nmcli connection up system	To activate connection
systemctl restart NetworkManager	To restart network service
more /etc/resolv.conf	To verify DNS IP Address
cd /etc/sysconfig/network-scripts/	Location of Connection profile file **ifcfg-system**
man nmcli and man nmcli-examples	Manual pages for **nmcli** and **nmcli-examples**

Note

NetworkManager is default network manager in RHEL 8 and you will not find network.service pre installed in RHEL 8 which is installed in case of RHEL 7.

You can install network service by installing package with dnf /yum command.

```
# dnf install network-scripts
```

network.service uses network scripts for network management . network.service is deprecated and will be removed probably in next major release.

Scheduling cron job as root user in Other User's crontab

Schedule a script /test.sh as user lisa which should be executed every 15 minutes.

Command	Action/Description
crontab -u lisa -e	Open **lisa's** crontab as root user for editing
*/15 * * * * /test.sh	Make entry in crontab file
:wq	Write and quit
crontab -u lisa -l	List crontab of user **lisa**

```
The time and date fields are:

      field          allowed values
      -----          --------------
      minute         0-59
      hour           0-23
      day of month   1-31
      month          1-12 (or names, see below)
      day of week    0-7 (0 or 7 is Sunday, or use names)
```

Note :
System administrator(root) can restrict access to crontab for different users using /etc/cron.allow and /etc/cron.deny files.

if,

- /etc/cron.allow file exists , user must be listed in this file to use crontab.
- /etc/cron.deny exists , user must not be listed in this file to use crontab.
- Both files does not exist, only root user can use crontab.

Scheduling cron job as Normal User(non-root)

Schedule a script /test1.sh as user bob which should be executed 12:15 every Monday. User bob should be allowed to use crontab.

Command	Action/Description
vim /etc/cron.allow	Open **cron.allow** file in editing mode
bob	List **bob** in cr**on.allow** file
:wq	Write and quit **cron.allow** file
su - bob	Switch to user **bob**
crontab -l	List bob's crontab
crontab -e	Open crontab in editing mode
15 12 * * 1 /test1.sh	List this entry in crontab
:wq	Write and quit
crontab -l	List bob's crontab to verify cron job is scheduled

Scheduling jobs using at command

Schedule below command using at to execute 30 minutes from now

ps ef > process.txt

Check the queue of at jobs to verify.

Command	Action/Description
dnf install at	Install package for **atd** if not installed already
systemctl start atd and systemctl enable atd	To start and enable **atd** service
systemctl status atd.service	To check status of **atd**
at now + 30 minutes	Execute at to schedule job
at> ps -ef > process.txt	To save Ctrl+d
atq	Display **at** jobs queue (Pending jobs)
atrm	To delete jobs
man at	Manual page for at

at Jobs Scheduling Examples	Description
at now + 30 minutes	To schedule job to execute after 30 minutes from now
at now + 1 hour	To schedule job to execute after 1 hour from now
at now + 2 days	To schedule job to execute 2 days after from now
at 4pm + 2 days	To schedule job to execute 2 days after from 4PM
at now + 3 weeks	To schedule job to execute 3 weeks after from now
at HH:MM	To schedule job at some specific time
at midnight	To schedule job to execute at midnight
at noon	To schedule job to execute at noon
at teatime	To schedule job to execute at teatime (4pm)

Configure Service to Start at System boot

Example:

autofs is a program for automatically mounting directories on an as-needed basis. Auto-mounts are mounted only as they are accessed, and are unmounted after a period of inactivity. Because of this, automounting NFS/Samba shares conserves bandwidth and offers better overall performance compared to static mounts via fstab.

To start autofs service at system boot:

Command	Action/Description
dnf install autofs	To install **autofs** service
systemctl start autofs	To start **autofs** service
systemctl enable autofs	To start **autofs** service at boot
systemctl status autofs	To check status of **autofs** service

Configuring System to Use Time Services

Configure System to use NTP server configured on IPA Server (ipaserver.example.com)

• Configure iburst option to make the initial synchronisation faster.

Command	Action/Description
systemctl status chronyd	To check status of **chronyd** service
timedatectl set-timezone TIME_ZONE	To set time zone
timedatectl set-time HH:MM:SS	To set time
hwclock -w	To sync hardware clock to system time if needed
vim /etc/chrony.conf server ipaserver.example.com iburst :wq	To configure System to use NTP server **(IPA Server)**
systemctl restart chronyd	Restart **chronyd** to make changes effective
vim /etc/chrony.conf local stratum 10 :wq	To enable **IPA Server** to serve as NTP source with stratum 10
systemctl restart chronyd	Restart service on **IPA Server** to make changes effective
chronyc sources	Display **chronyc sources** on **System**

Working With Module Streams-dnf

List the AppStream modules available in repository.

- List the different Streams available for perl module.
- Install Stream 5.26 of perl module.

Command	Action/Description
dnf module list or yum module list	To list available module in AppStream repository
dnf module info perl or yum module info perl	To list different streams for perl module
dnf module info --profile perl:5.26	To list specific perl stream
dnf install @perl:5.26	To install stream 5.26 of perl module

Modifying Bootloader (GRUB2) Settings

Modify GRUB2 (Bootloader) not to boot the System with GUI Mode booting screen(rhgb).

- Also all boot messages should be shown on the screen.
- Make sure interface names(eth*) should be used.

Command	Action/Description
vim /etc/default/grub Add biosdevname=0 and net.ifnames=0 to variable GRUB_CMDLINE_LINUX Remove rhgb and quiet :wq	Open GRUB settings file in editing mode
grub2-mkconfig -o /boot/grub2/grub.cfg	Rebuild GRUB config file, System on **BIOS** firmware
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg	Rebuild GRUB config file, System on **UEFI** firmware
systemctl reboot	To reboot the System

rhgb (redhat graphical boot) This is a GUI mode booting screen with most of the information hidden while the user sees a rotating activity icon spinning and brief information as to what the computer is doing.

quiet - hides most boot messages before rhgb starts. These are supposed to make the common user more comfortable.

Module 4: Using RedHat Essential Tools

Review: Local User and Group Management

Users and Group Management

a. Using Terminal

Groups

Groups allow multiple users with similar security and access levels to be linked, making management of those users easier. A local group is created with the groupadd command.

```
# groupadd dba
```

The group information is visible in the "/etc/group" file. Each group has a GID. If this is not assigned explicitly, the next largest number is used. We can see group we just defined has been assigned the GID of 500.

```
# cat /etc/group | grep dba
dba:x:500:
```

If you have the same groups across multiple servers it makes sense to set the GID explicitly to make sure it is the same across all servers.

```
# groupadd -g 1000 dba
```

Existing groups are modified using the groupmod command.

```
# groupmod -g 2000 dba
# groupmod -n new_dba dba
```

Groups are deleted using the groupdel command.

```
# groupdel new_dba
```

Users

The useradd command creates new local users.

```
# useradd oracle
```

The user details are visible in the "/etc/passwd" file. If no UID is specified, the next largest UID is assigned. A new group with a group name matching the user name is also created. By default, the users home directory is created under the "/home" directory and the shell is "/bin/bash".

```
# cat /etc/passwd | grep oracle
oracle:x:500:500::/home/oracle:/bin/bash
```

```
# cat /etc/group | grep oracle
oracle:x:500:
```

As with groups, if you have the same user across several servers it makes sense to explicitly define a UID so it matches on all servers. If the users should be assigned to an existing group, this can be done while creating the user also.

```
# groupadd -g 1000 dba
# useradd -G dba -u 2000 tim_hall
# cat /etc/passwd | grep tim_hall
tim_hall:x:2000:2000::/home/tim_hall:/bin/bash

# cat /etc/group | grep tim_hall
dba:x:1000:tim_hall
tim_hall:x:2000:
```

There are flags to alter the default shell (-s) and default home directory (-d), but for the most part these should be unnecessary.

Most of the user details can be modified using the usermod command.

```
# usermod -s /bin/ksh tim_hall
# usermod -a -G oinstall tim_hall
```

The passwd command is used to set the password for a specified user, or the current user if no user name is specified.

```
# passwd tim_hall
Changing password for user tim_hall.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

```
# passwd
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

The userdel command removes a user. The "-f" option deletes the user even if the user is currently logged in. The "-r" flag removes the user's home directory.

```
# userdel -r tim_hall
```

When logged in as the "root" user, the command prompt will display a "#" symbol. For ordinary users, the "$" symbol is displayed.

Password expiry

Password expiry (ageing) is controlled using the chage command. To check the current password expiry information use the "-l" option.

```
# useradd tim_hall
# chage -l tim_hall
Last password change                                    : Mar 01, 2012
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

There are a number of options available, but the most commonly used ones are shown below.

```
# # Set the days before change required (-M) and the number of days warning (-W)
# chage -M 30 -W 5 tim_hall
```

```
# # Immediates expire a password.
# chage -d 0 tim_hall
```

Changes are visible using the "-l" list option again.

```
# chage -l tim_hall
Last password change                                    : password must be changed
Password expires                                        : password must be changed
Password inactive                                       : password must be changed
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 30
Number of days of warning before password expires       : 5
```

Others:

To lock a user's account, use the passwd command with the -l option:

```
# passwd -l username
```

To unlock the account, specify the -u option:

```
# passwd -u username
```

To change how long a user's account can be inactive before it is locked, use the usermod command. For example, to set the inactivity period to 30 days:

```
# usermod -f 30 username
```

To change the default inactivity period for new user accounts, use the useradd command:

```
# useradd -D -f 30
```

A value of -1 specifies that user accounts are not locked due to inactivity.

If you want to grant certain users authority to be able to perform specific administrative tasks via sudo, use the visudo command to modify the **/etc/sudoers** file.

For example, the following entry grants the user erin the same privileges as root when using sudo, but defines a limited set of privileges to frank so that he can run commands such as systemctl, rpm, and dnf:

```
erin ALL=(ALL) ALL
frank ALL= SERVICES, SOFTWARE
```

Making SSH Connection to Remote Host

Establish SSH connection to ipaserver.example.com from system.example.com

• Use user ldap to make this connection.

Command	Action/Description
systemctl status sshd	To check status of **sshd** service on **IPA Server**
ssh ldap@ipaserver.example.com	To establish ssh connection as user(username) **ldap**
Enter password for ldap :******	Enter password for user **ldap**
hostnamectl	Verify user **ldap** is connected to IPA Server
man ssh	Manual page for **openSSH SSH client**
man sshd	Manual page for **openSSH daemon**

Securely Copy files Across Hosts Using scp

Securely copy /root/process.txt file from system.example.com to /tmp directory on ipaserver.example.com

• Use the password password for this task.

Command	Action/Description
scp /root/process.txt ipaserver.example.com:/tmp/	To transfer files securely with scp
password : ******	Enter password
man scp	Manual page for SCP

Using grep to Match Lines Starting With Pattern(s)

Copy all lines starting with word Sed or sed from /file.txt and copy to file /root/file

Command	Action/Description
man grep	To display manual page for grep
egrep '^Sed|^sed' /file.txt > /root/file	Copy lines starting with **Sed** or **sed**
more /root/file	To verify if lines are copied

Using grep for non-matching patterns (Invert-match)

Copy all lines not containing sEd or SeD from file /file.txt to /root/invert file.

Command	Action/Description
egrep -v 'sEd|SeD' /file.txt > /root/invert	To copy required lines
more /root/invert	To verify if lines are copied
man grep	To check manual page for grep

Using find Command to List Files Owned by User

Save all the files(regular files) owned by user lisa in / lisa_files

Command	Action/Description
find / -user lisa -type f > /root/lisa_files	To find **lisa's** files and copy
more /root/lisa_files	To verify results
man find	To check manual page for **find**

Using find Command to Locate File Using Name of File

Locate the file smb.conf searching through and save the output in /conf file.

• STDERR should not be saved.

Command	Action/Description
find / -name smb.conf -type f > /conf	To find and save file on indicated location
more /conf	To verify results

Using find to Search Files with extension .txt

Locate the files with extension .txt searching through / and save the output in /text file.

• STDERR should also be copied.

Command	Action/Description
find / -name "*.txt" -type f > /text 2>&1	To find files with extension **.txt** and copy files
more /text	To verify results

Using find Command to Save all Directories Owned by User

Find all the directories owned by user bob and save the output to bob_dir

Command	Action/Description
find / -user bob -type d > /bob_dir	To find directories owned by bob and copy them
more /bob_dir	To verify results

Using find Command to List all Files & Directories based on UID

Find all the directories and files owned by user with userid 1002 and save the output to /uid1002

• STDERR should not be copied.

Command	Action/Description
find / -uid 1002 > /uid1002	To find required files and copy them
more /uid1002	To verify results
man find	To check manual page for **find**

Using tar Command to Archive and Compress Contents of Directory

Use tar command to archive all contents of /home directory in /root/home.tar file.

• Compress the archived files using bzip2.

Command	Action/Description
tar -cjvf home.tar.bz2 /home	To archive and compress contents of /home
ls -l --block-size=MB	To verify gain after compression
man tar	To check manual page for tar

Using tar command to Extract the Data from Archive

Use tar command to extract the contents of home.tar.bz2 to / directory.

•Delete contents of /home directory before extracting the data.

Command	Action/Description
tar -xvf home.tar.bz2 -C /	To extract contents of home.tar.bz2 in **/**
cd /home	Navigate to /home directory
ls -l	List the contents to verify

Using tar to Archive Contents with gzip Compression

Use tar command to archive all contents of /etc directory in /root/etc.tar file.

• Compress the archived files using gzip

Command	Action/Description
tar -czvf etc.tar.gz /etc	To archive and compress using **gzip**
ls -lrt	To verify results

Decompressing Files using gunzip and bunzip2

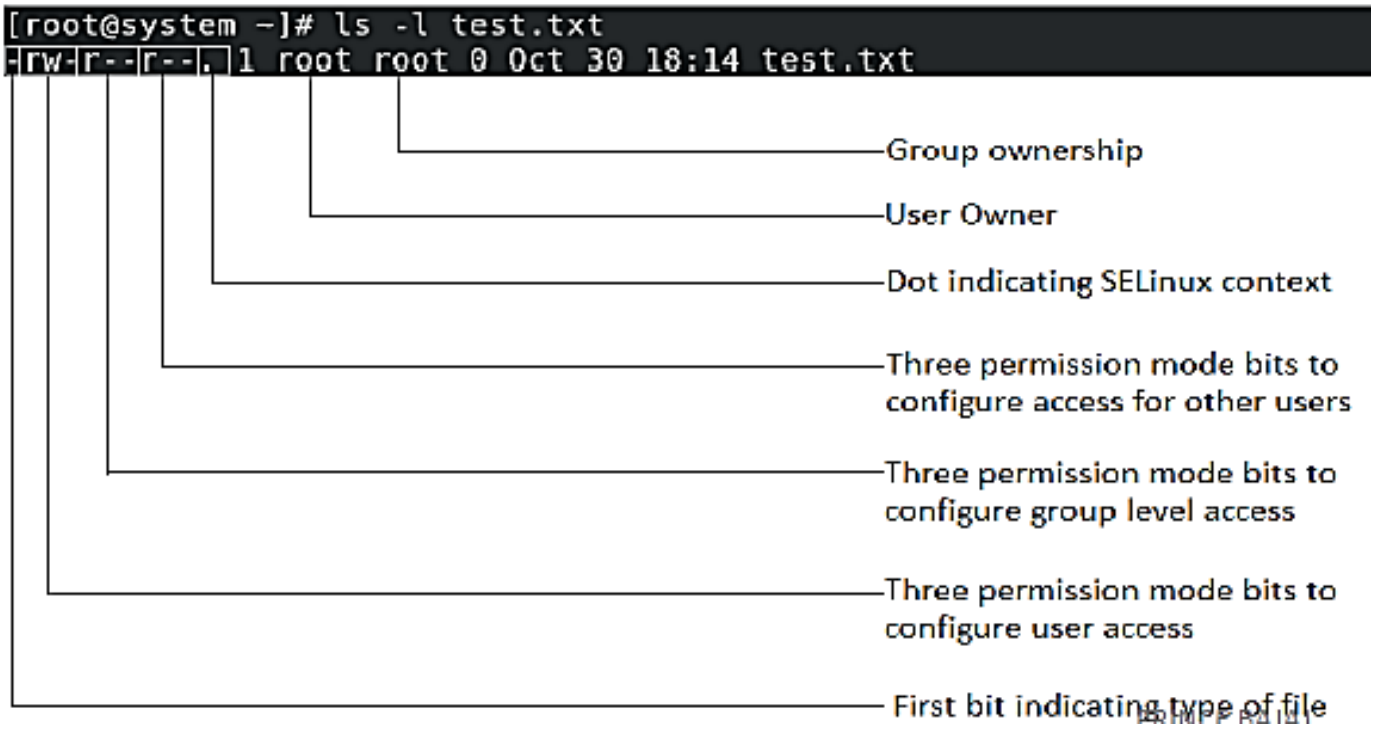
Use gunzip command to decompress contents of /root/etc.tar.

• Use bunzip2 command to decompress contents of /root/home.tar.bz2

Command	Action/Description
gunzip /root/etc.tar.gz	To decompress contents of **/root/etc.tar.gz**
bunzip2 /root/home.tar.bz2	To decompress contents of **/root/home.tar.bz2**
man gunzip	To check manual page for **gunzip**
man bunzip2	To check manual page for **bunzip2**

Introducing Filesystem Permissions and Commands

Permission Type	File	Directory
read	Allows a user only to read the contents of file	Allows a user to list files under directory
write	Allows a user to read and write(modify) to file	Allows a user to create new files under directory and list all files
execute	Allows a user to execute binary code in the file	Allows a user to navigate or move to directory



Setting User & Group Ownership and Configuring Permissions

Command	Command Usage
chown	To set user owner and Group ownership of filesystem
chmod	To configure permissions using permission mode bits
setfacl	To configure ACL's (Access control lists) for additional user Access

Symbolic Representation Method

Note : Absolute paths or Relatives paths of files and directories should be used with all commands

chmod command Usage Examples	Action
chmod u+rwx FILE_NAME	Provides read/write/execute permissions to user owner on file
chmod g-w FILE_NAME	Removes write permission for file at group level without changing other existing permissions
chmod o+r DIR_NAME	Provides read permission to others on directory without affecting other permissions
chmod a+r DIR_NAME	Provides all system users with read right on directory without affecting other permissions
chmod ug-x FILE_NAME	Removes execute permissions on file for user and at group level
chmod ugo+x FILE_NAME	Provides execution permissions on file to user, group members and others

Numeric Mode Representation Method

Note : Absolute paths or Relatives paths of files and directories should be used with all commands

Permission Set	read (4)	write (2)	Execute (1)	Octal Number
read only	4	0	0	4+0+0=4
write only	0	2	0	0+2+0=2
execution only	0	0	1	0+0+1=1
read, write	4	2	0	4+2+0=6
read, execution	4	0	1	4+0+1=5
write, execution	0	2	1	0+2+1=3
read, write, execution	4	2	1	4+2+1=7

chmod Examples	Action
chmod 0600 FILE_NAME	Provides read/write permissions on file to user owner and removes all other permissions for group and others if they exist.
chmod 0001 DIR_NAME	Provides read permission to others on directory and removes all other permissions for group and user owners if they exist.
chmod 0766 DIR_NAME	Provides all permissions to user owner and read/write permissions to group and others.

Creating Symbolic Links

Create symbolic link for file /test/sym/link/sym_link in /root directory.

Command	Action/Description
mkdir -p /test/sym/link	To create directory path
touch /test/sym/link/sym_link	To create empty file **file** on path
cd /root	Navigate to root directory
ln -s /test/sym/link/sym_link	To create symlink

Module 5: Operate Running Systems and Processes

Adjusting Priority of Process With renice

Run CPU intensive process on System with nice value of -5 to give more CPU attention than default.

• Adjust the niceness value to 5 so that CPU pays less attention to this process

Command	Action/Description
nice -n -5 dd if=/dev/zero of=/dev/null	To start process with **nice** value **-5**
renice -n 5 -p PID	To adjust **nice** value
top	To verify changes

Note:

- Nice value can be between -20 to 19 . Lesser the NICE value, more CPU resources will be used. Higher the nice value, less CPU attention will be given.
- Never run process with nice value of -20 ,CPU will give highest priority and no other jobs will be able to

Running Job in background with Predefined nice Value

Example: Run below command in back ground with nice value of 10

sleep 3600

Command	Action/Description
nice -n 10 sleep 3600 &	To start a process with pre-defined nice value
top	To verify results

Killing Processes Forcefully

Kill all running dd processes to stop forcefully.

Command	Action/Description
top	To check running dd processes and PID's
kill -9 PID	To kill process forcefully
man kill	Manual page for **kill** command
kill --help	To display help for **kill** command

Tuning kernel Parameter vm.swappiness Persistently

Modify kernel parameter vm.swappiness to set the value to 10.
• Changes done should persist after reboot.

Definition:
it represents the percentage of the free memory before activating swap. The lower the value, the less swapping is used, and the more memory pages are kept in physical memory.

Changing the value directly influences the performance of the Linux system. These values are defined:
0 : swap is disable
1 : minimum amount of swapping without disabling it entirely
10 : recommended value to improve performance when sufficient memory exists in a system
100 : aggressive swapping

Command	Action/Description
sysctl -a | grep swappiness	To display existing value of parameter
sysctl -w vm.swappiness=10	To change value of parameter in run time
vim /etc/sysctl.conf vm.swappiness=10 :wq	To make changes persistently
systemctl reboot	To reboot System
sysctl -a | grep swappiness	To verify results PRINCE BA

Introducing Tuned & Tuned Profiles

Tuned is service which monitors the system and optimizes the performance of system for different use cases.

- There are pre-defined tuned profiles which are present on path /usr/lib/tuned.
- Tuned profiles are designed keeping in mind three parameters linked closely to performance of system.
  - High throughput
  - Low latency
  - Saving power
- Predefined profiles are divided into two categories :
  - Power-saving profiles
  - Performance-boosting profiles
- You can customise a tuned profile based on standard profile or can create a completely new profile. Such profiles are always created under directory /etc/tuned. In case you want to create a new profile by adding different settings to pre-defined profile, copy the profile from /usr/lib/tuned to /etc/tuned and add/modify different settings.
- If two profiles have same name under /usr/lib/tuned and /etc/tuned , profile under /etc/tuned takes precedence.

Tuned Profile Configuration File- tuned.conf

tuned.conf –Tuned profile definition

- ✓ Tuned profile is defined in tuned.conf file on directory path /usr/lib/tuned/<profile>/tuned.conf or on path /etc/tuned/<profile>/tuned.conf.
- ✓ Profile on path /etc/tuned takes precedence.

Tuned profiles distributed with RHEL 8
Balanced, powersave, throughput performance, latency performance, network latency, network throughput, virtual guest, virtual host

Set recommended tuned profile on System and verify same after changes are done

Command	Action/Description
dnf install tuned	Installing **tuned** service
systemctl enable --now tuned	Starting and configuring **tuned** to start at boot
tuned-adm list	Listing available **tuned** profiles
tuned-adm recommend	Listing **tuned** recommend profile
tuned-adm active	Listing active profile on **System**
tuned-adm profile recommended_profile	Activating recommended **tuned** profile
tuned-adm active	Verifying active profile after changes
man tuned-adm	To check manual page for **tuned-adm**

Create a custom tuned profile with name myprof based on base profile vitual guest and below mentioned different settings.

- vm.swappiness =40
- CPU governor = powersave
- Make sure sysctl settings should not be overridden by tuned profile.

Command	Action/Description
dnf install tuned	Installing **tuned** service
systemctl enable --now tuned	Starting/enabling **tuned** servcie
tuned-adm active	Listing active tuned profile
mkdir /etc/tuned/myprof	Creating directory under **/etc/tuned** to create new profile with name **myprof**
vim /etc/tuned/myprof/tuned.conf [main] include = virtual-guest [cpu] type = cpu governor = powersave [my_sysctl] type = sysctl vm.swappiness = 40 :wq	Creating tuned configuration file for profile **myprof**
tuned-adm profile myprof	Activating customized tuned profile **myprof**
systemctl restart tuned	Restarting **tuned** service to make changes effective

Activate additional tuned profile powersave to merge with existing profile.
• powersave profile must have high priority under conflicting conditions.

Command	Action/Description
dnf install tuned	Installing **tuned** service
systemctl enable --now tuned	Starting/enabling **tuned** service
tuned-adm list	Listing available **tuned** profile
tuned-adm active	Listing active **tuned** profile
tuned-adm profile myprof powersave	Merging existing active profile with additional profile **powersave**
tuned-adm list	Verifying active profiles

Note:Tuned service try to optimize the system in best way possible by merging profiles under load conditions but in case of conflicting settings , last profile is given precedence.

Module 6: Users and Groups Management

Default User Settings

When we create user without specifying or using any options with useraddcommand default user settings are applied to user account.

Default user settings are applied from below two files:
•/etc/default/useradd -Default values for user account creation
•/etc/login.defs -Defaults about shadow-utils components ,password aging controls, UIDs ,GIDs etc.

Verifying Different Settings of User Account

We will create test user (username -test) with default user settings and we will set password as password.

We will verify default user settings from below three files :
•/etc/passwd -User account information
•/etc/shadow -Secure user account information , Information about password aging and account controls.
•/etc/group -Group account information

Overriding Default User Settings

To override default user settings we will use useradd , passwd and chage with different options.

Command	Action/Description
useradd	To create user account
passwd	To set **password** for user account
chage	To display/configure password aging and account controls
man useradd	To check manual page for **useradd**
man passwd	To check manual page for **passwd**
man chage	To check manual page for **chage**

Modifying Defaults in login.defs

Configure System (system.example.com) to use UID_MIN and GID_MIN as 5000 by default.

- Create one test user (username) test1 and verify that default selected UID is 5000=+.
- Make user account password less .

Command	Action/Description
vim /etc/login.defs UID_MIN 5000 GID_MIN 5000 :wq	Configure **System** to use Minimum UserID and GroupID as 5000
useradd test1	Create **test1** user with default settings
more /etc/passwd | grep test1	Verify default **UID** assigned is 5000=+
passwd -u -f test1	To make user account password less
su - test1	Try user login as non-root(normal) user, user should be able to login without password

Modifying Defaults in useradd file

Modify default password inactivity period (INACTIVE variable) on System to 5 days.

Command	Action/Description
useradd -D	To display **useradd** defaults
useradd -D -f 5	To modify default password inactivity period **(INACTIVE=5)** to 5
useradd –D	To verify results

Creating User with Specific UID and Non-interactive Shell

Create a user with name lara with password access

- Use UID 6000 for this user.
- User should have non interactive shell

Command	Action/Description
useradd -u 6000 -s /sbin/nologin lara	To create user with **UID 6000** and **non-interactive** login shell
passwd lara	To set the password for user **lara**
more /etc/passwd | grep lara	To verify user account information
more /etc/group | grep lara	To verify Group information

Creating User with Non-Default Home Directory and Password Aging Controls

Create a user riya with home directory riya /private and set password access

- At first login, she should be forced to change her password.
- Password should be set to expire every month.

Command	Action/Description
mkdir -p /riya/private	To create non-default home directory
useradd -d /riya/private riya	To create user with non-default home directory **/riya/private**
chown riya:riya /riya/private	To set User and Group Owner as **riya** on this directory
chmod 700 /riya/private	To restrict the access to **riya** for her home directory
passwd riya	To set the password for user **riya**
chage riya	To set the maximum password age and password change after first login
more /etc/passwd	To verifying user account information
more /etc/shadow	To verify configured password aging controls, account controls etc
more /etc/group	To verify group(s) related information

Assigning Supplementary (Secondary) Group with Specific GID to User

Create a group named rhcsa and assign this group to user lara as secondary group.

- GID 5555 should be used

Command	Action/Description
groupadd -g 5555 rhcsa	To Create group sys with **GID 5555**
usermod -aG rhcsa lara	To assign **rhcsa** group to user **lara** as supplementary group
more /etc/group	To verify group related information

Configuring Password Aging and Account Expiration for User Account

Create user harry and set password as access

- Account should expire on 31 st Dec current year
- Password should expire every 7 days
- Set password expiry warning to 2 days

Command	Action/Description
useradd harry	To create user **harry** with default settings
passwd harry	To set the password for user **harry**
chage harry	To set the **maximum password age** ,**password expiry warning** and **account expiration** date
more /etc/passwd | grep harry	Verifying user account information
more /etc/shadow | grep harry	Verifying configured password aging controls and account controls

Setting User Owner and Group Owner on Directory

Create the directory dir and set the group and user ownership to rhcsa and lara respectively.

• Give read only access to group rhcsa and rwx access to lara

Command	Action/Description
mkdir /dir	To create directory **/dir**
chown lara:rhcsa /dir	To set user and group ownership
ls -ld /dir	To verify permissions

Configuring System as IPA Client to Use LDAP Users

Configure System (system.example.com) as IPA client to use LDAP services configured on IPA Server with Free IPA Server solution.

• Use user ***admin*** and password ***password*** to enroll System into our IPAServer.

• ldap users should be able to login on System

Command	Action/Description
dnf install ipa-client	To install package for ipa client
ipa-client-install	Configuring System as **IPA Client**
su - ldap	Verify **ldap** user login

Configuring autofs to Mount Home Directory of LDAP user

Configure system.example.com to automount home directory of LDAP user ldap when logged in.

• Home directory of LDAP user is / ldapuser ldap

• Home directory is shared by ipaserver.example.com through NFS export

• LDAP user should get his home directory when logged in.

Command	Action/Description
dnf install autofs	To install packages required for **autofs**
systemctl start autofs	To start **autofs** service
systemctl enable autofs	Configuring to start service at boot
vim /etc/auto.master /home/ldapuser /etc/auto.ldap :wq	To define master automounter map
vim /etc/auto.ldap ldap ipaserver.example.com:/home/ldapuser/ldap :wq	
systemctl restart autofs	Restart **autofs** service
su - ldap	Switch user to **ldap**
pwd	Current directory should be ldap user's home directory

Configuring Autofs to Mount Home Directories of Multiple Users Using Wild Cards

Configure system.example.com to automount home directories of LDAP users ldap1 and ldap2

- Home directories of LDAP users ldap1 and ldap2 are ldap /home/ldap1 and ldap /home/ldap2 respectively.
- Home directories are shared by ipaserver.example.com through NFS export.
- LDAP user should get his home directory when logged in.

Command	Action/Description
dnf install autofs	Installing **autofs**
systemctl start autofs	Starting **autofs** service
systemctl enable autofs	Configuring **autofs** to start at boot
vim /etc/auto.master /ldap/home /etc/auto.ldap12 :wq	To define base location for home directory
vim /etc/auto.ldap12 * ipaserver.example.com:/ldap/home/& :wq	
systemctl restart autofs	Restarting **autofs** service
su - ldap1	Switch user to **ldap1**
pwd	Current directory should be home directory of **/ldap/home/ldap1**

Deleting User's Account

Create user (username) maria on System with default user settings.

- Delete user maria from system.example.com
- User's home directory and mailbox should also be deleted.

Command	Action/Description
useradd maria	To create user **maria** with default user settings
userdel -r maria	To delete user **maria** ,also home directory and mail spool
userdel --help	To check help for **userdel**

Configuring Superuser Access

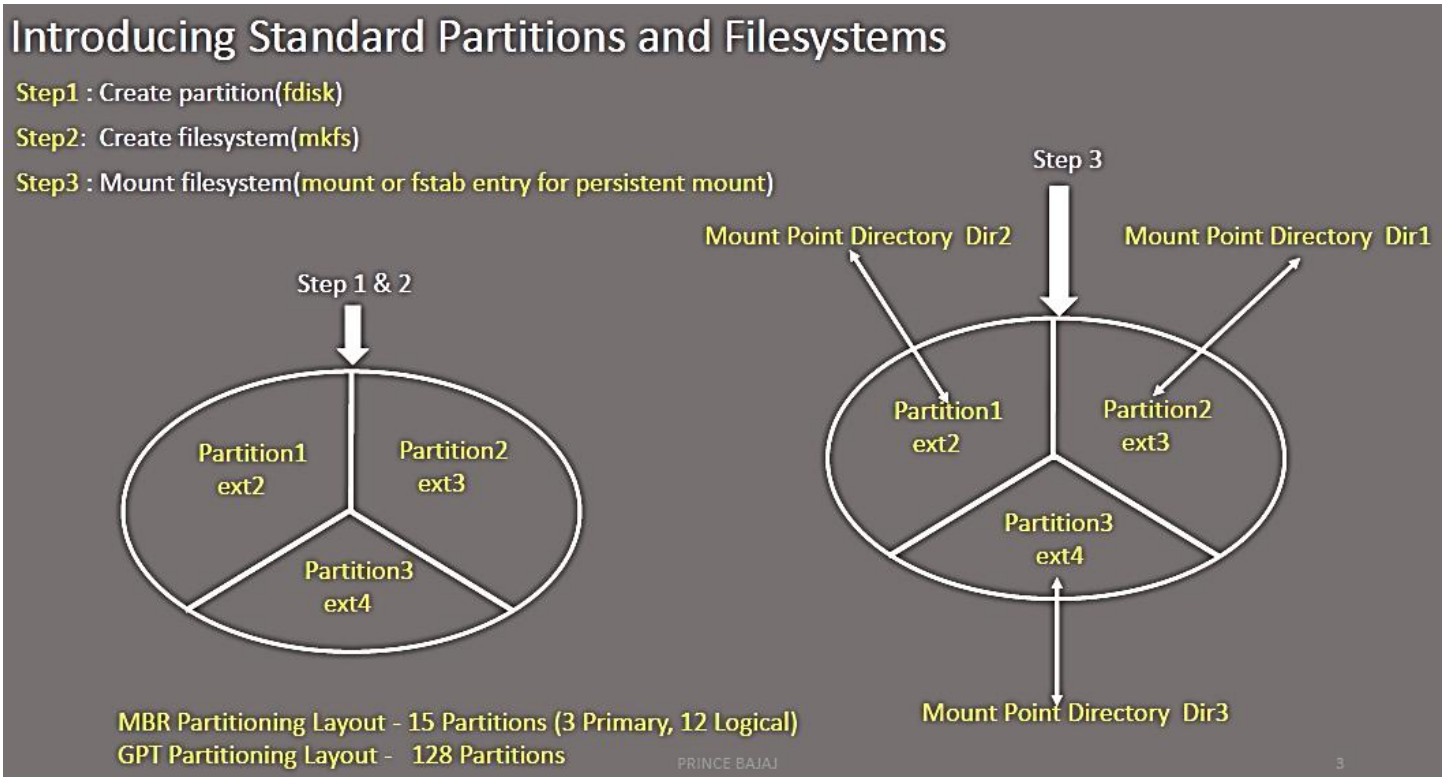
Configure superuser access for user harry to enable him to use root privileges with sudo

• Create a user with username testuser using sudo

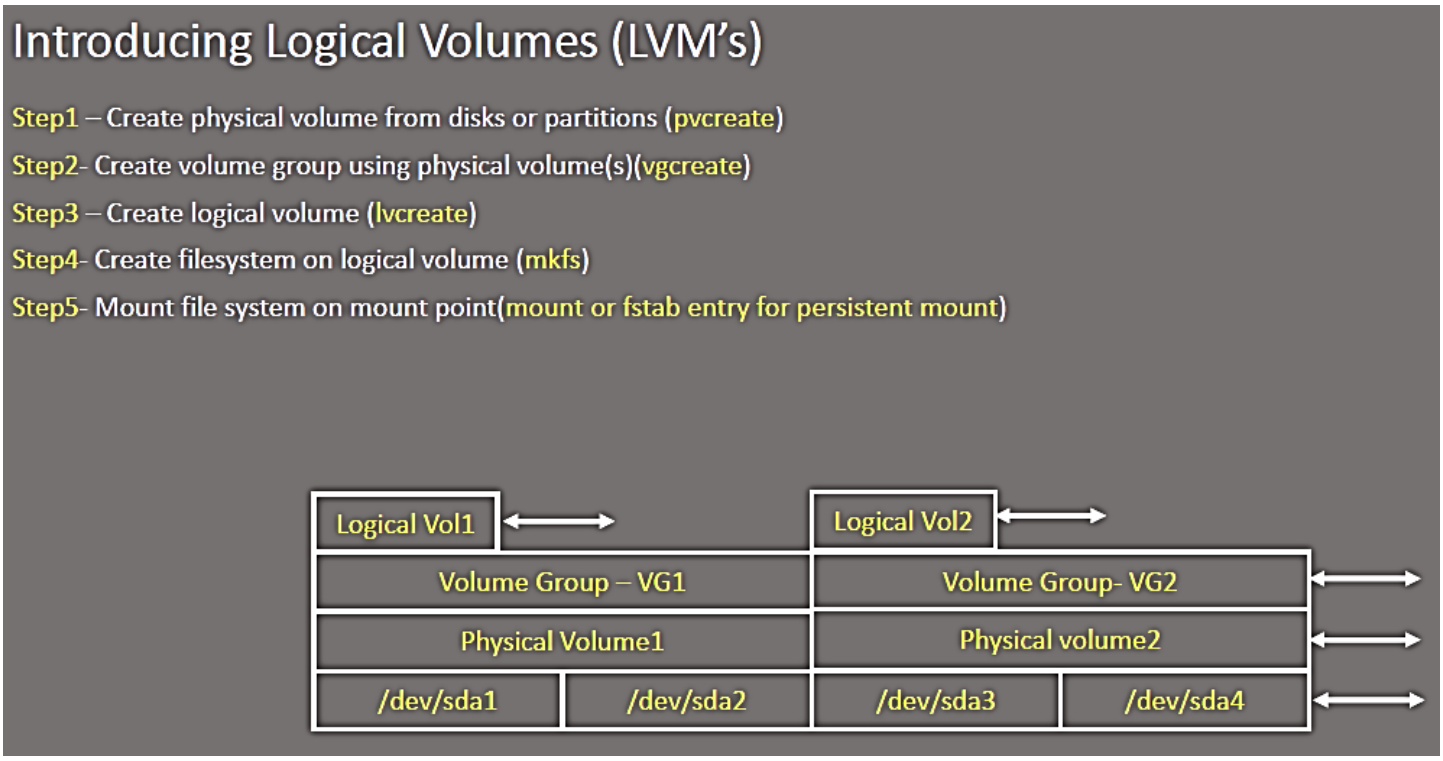
Command	Action/Description
more /etc/sudoers	To verify line **%wheel ALL=(ALL) ALL** is not commented
usermod -aG wheel harry	To add user **harry** to wheel group
sudo whoami	To verify access
sudo useradd testuser	To create user using **sudo**

Module 7: Configuring Local Storage and File Systems

Introducing Standard Disk Partitions and Filesystems



Introducing Logical Volumes



Creating Extended and Logical Partitions

Create a standard disk partition of 1 GiB size and mount this on mnt /partition directory.

- Partition should use xfs file system.
- Mount should be persistent.

Command	Action/Description
fdisk /dev/sda First input : **n** , Second input : **e** , Two times Enter(To assign remaining space for Logical partitions) ,Third input : **n** , Enter (Default First sector),Fourth input: **+1G** ,**wq** (to save and quit)	To create container for extended partition (Container) and to create logical partition of **size 1 Gib**
partprobe	To inform kernel about this partition
mkdir /mnt/partition	To create mount directory
mkfs.xfs /dev/sda5	To create xfs filesystem on partition
mount /dev/sda5 /mnt/partition	To test mounting filesystem temporarily
mount	To verify mounted filesystem
vim /etc/fstab /dev/sda5 /mnt/partition xfs defaults 0 0 :wq	To mount filesystem persistently (to persist across reboot)
mount -a	To mount through fstab
lsblk	To list block devices

Mounting Filesystem with Read Only Permission Through fstab

Create a disk partition of size 1 GiB and mount this for read only access on mnt /fat directory.

- Use vfat file system for the partition.
- Mount should be persistent.

Command	Action/Description
fdisk /dev/sda First input : **n** ,Enter(Default First sector), Second input : **+1G,** Enter ,**w** (to save and quit)	To create logical partition of **1 GiB** size
partprobe	To inform kernel about this partition
mkdir /mnt/fat	To create mount point directory
mkfs.vfat /dev/sda6	To create **vfat** filesystem on partition
mount -o ro /dev/sda6 /mnt/fat	To test mounting filesystem
mount	To verify mounted filesystem
vim /etc/fstab /dev/sda6 /mnt/fat vfat ro 0 0 :wq	To mount filesystem persistently
lsblk	To list block devices

Configuring & Adding SWAP to System

Configure and add 1 GiB swap space to your System

Command	Action/Description
fdisk /dev/sda First input : **n** ,Enter, Enter (Default First sector), Second input : **+1G** , Enter, Third input : **t** ,Enter ,Enter(for default Partition),Fourth input : **82** ,Enter, **w** (to save and quit),Enter	To create logical partition of **1 GiB** size type **Linux swap**
partprobe	To inform kernel about these changes
mkswap /dev/sda7	To configure partition as swap
vim /etc/fstab /dev/sda7 swap swap defaults 0 0 :wq	Make entry in **fstab** file
swapon -a	To activate swap as per entry in **fstab**
free -m	To verify added swap memory

Configuring & Mounting Logical Volume

Configure logical volume with name lv_volumewhich should use 200 MiBfrom volume group vg_groupof size 300 MiB.

- ext4file system should be used
- Mount this on /mnt/log_voldirectory and mount should be persistent.

Command	Action/Description
fdisk /dev/sda First input : **n** ,Enter,Enter (Default first sector) , Second input :**+300M** ,Enter, Third input : **t**,Enter,Enter(for default Partition),Fourth input : **8e** ,Enter,**w** (to save and quit),Enter	To create logical partition of size **300 MiB** with type **LVM**
partprobe	To inform kernel about this partition
pvcreate /dev/sda8	To create physical volume
vgcreate vg_group /dev/sda8	To create volume group from physical volume
lvcreate -n lv_volume -L 200M vg_group	To create logical volume on volume group
mkdir /mnt/log_vol	To create mount directory
mkfs -t ext4 /dev/vg_group/lv_volume	To create **ext4** filesystem on logical volume
mount /dev/vg_group/lv_volume /mnt/log_vol	To mount filesystem temporarily using mount
mount	To verify mounted filesystem
vim /etc/fstab /dev/vg_group/lv_volume /mnt/log_vol ext4 defaults 0 0 :wq	Make entry in **fstab** to mount filesystem persistentl
mount -a	To mount through **fstab**

Configuring Logical Volume (LVMs) Using Physical Extents of Non-default Size

Configure logical volume with name volumewhich should use 20 PE's from volume group named group of size 400 MiB.

- Size of PE should be 16 MiBand file system used must be ext4file system.
- Mount this on /mnt/volume directory and mount should be persistent.
- Use UUID to mount this.

Command	Action/Description
fdisk /dev/sda First input : **n** ,Enter,Enter (Default first sector), Second input : **+400M** ,Enter, Third input : **t**, Enter, Enter(for default Partition),Fourth input : **8e** ,Enter, **w** (to save and quit),Enter	To create logical partition
partprobe	To inform kernel about changes
pvcreate /dev/sda9	To create physical volume
vgcreate -s 16M group /dev/sda9	To create volume group with **PE size of 16 MiB**
lvcreate -n volume -l 20 group	To create logical volume using **20 PE's** from volu
mkdir /mnt/volume	To create mount point
mkfs.ext4 /dev/group/volume	To create **ext4** filesystem on logical volume
mount /dev/group/volume /mnt/volume	To mount filesystem in runtime
mount	To verify mounted filesystem
vim /etc/fstab UUID =? /mnt/volume ext4 defaults 0 0 :wq	To mount persistently through **fstab**
mount -a	To verify mounted filesystem

Configuring Logical Volume using 100% FREE PE's on Volume Group

Configure logical volume with name lvmfrom volume group vgroupof size 512 MiB.

- Logical volume should use complete free space on volume group.
- Create ext4file system on this volume.

Command	Action/Description
fdisk /dev/sda First input : **n** Enter,Enter(Default first sector), Second input : **+512M** ,Enter, Third input : **t** ,Enter,Enter (Default partition), Fourth input : **8e**,Enter ,**w** (to save and quit),Enter	To create logical partition of ty
pvcreate /dev/sda10	To create physical volume
vgcreate vgroup /dev/sda10	To create volume group
lvcreate -n lvm -l 100%FREE vgroup	To create logical volume using volume group
mkfs.ext4 /dev/vgroup/lvm	To create ext4 filesystem
lvdisplay	To display logical volume(s)
vgdisplay	To display volume group(s)
pvdisplay	To display physical volume(s)

Extending Logical Volume (LVMs) and Resizing Filesystem

Resize the lvmlv_volumeso that after reboot size should be in between 230MiBto 260 MiB.

•Make sure complete logical volume should be usable.

Command	Action/Description
lvdisplay	To display logical volumes
lvextend -r -L +45M /dev/vg_group/lv_volume	To extent logical volume and resize the filesystem

Extending Volume Group Size to extend Logical Volume & Filesystem

Extend size of LVM with name lvmby 256 MiB.

•Create new partition to increase the size of volume group.

•Format the complete volume with ext4 file system.

Command	Action/Description
fdisk /dev/sda First input : **n** ,Enter, Enter (Default first sector),Second input : **+300M** , Enter,Third input : **t** ,Enter ,Enter(Default partition) Fourth input : **8e** ,Enter, **w** (to save and quit),Enter	To create logical partition of type Linux LVM
pvcreate /dev/sda11	To create physical volume
vgextend vgroup /dev/sda11	To extend volume group
lvextend -r -L +256M /dev/vgroup/lvm	To extend logical volume
lvdisplay	To display logical volume(s)

Overriding Existing Filesystem Type

Create a standard partition of size 100 MiB and format this with ext4file system.

•Change the file system to xfs and verify same.

Command	Action/Description
fdisk /dev/sda First input : **n** ,Enter,Enter(To select default first sector), Second input :**+100MiB** ,Enter, **w** (to save and quit),Enter	To create partition (Logical partition) of **100 MiB** size
partprobe	To inform kernel about partition table changes
mkfs.ext4 /dev/sda12	To create **ext4** filesystem on partition
mkfs.xfs -f /dev/sda12	To change the filesystem type to **xfs** or to create **xfs** filesystem
blkid	To verify changes

Configuring Directory for Group Collaboration

Create a directory /private/home.

- Set the user ownership to lisa and group ownership to group.
- Give full permissions to group group on this directory.
- User riya should have no access on this directory.
- Add user's bob and harry to group group.
- Files created by user bob and harry under this directory should have group ownership set to group.

Command	Action/Description
mkdir -p /private/home	Creating directory path **/private/home**
chown lisa:group /private/home	Changing user and group ownership
chmod g+rwx /private/home	Configuring full permissions at group level
setfacl -m u:riya:- /private/home	Removing all permissions for user **riya**
usermod -aG group bob & usermod -aG group harry	Adding users to group **group**
chmod g+s /private/home	To set GID bit
getfacl /home/private	To display configured access control lists

Mounting NFS Share through fstab

Discover the NFS share exported by NFS server ipaserver.example.com.

- Mount the share /nfsshareon directory /nfs/share and mount should be persistent.
- NFS version 3 should be used.

Command	Action Description
dnf group install "Network File System Client"	Installing NFS client
showmount -e ipaserver.example.com	Discovering NFS exports
mkdir -p /nfs/share	Creating mount point directory
mount -o nfsvers=3 ipaserver.example.com:/nfshare /nfs/share	Mounting NFS share in run time
umount /nfs/share	Unmounting NFS share
vim /etc/fstab ipaserver.example.com:/nfsshare /nfs/share nfs _netdev,nfsvers=3 0 0 :wq	Making entry in **fstab** file for persistent mount
mount -a	Mounting through **fstab**
mount	Verifying the mounted filesystem and **version**

Mounting Samba Share through fstab

Discover the samba share and mount share sambaon /samba/smb1 directory with smb1user.

•Use the passwordpasswordto mount this share.

Command	Action/Description
dnf install samba-client cifs-utils	Installing **Samba** client
smbclient -L ipaserver.example.com	Listing **Samba** shares
mkdir -p /samba/smb1	Creating mount point directory
mount -o username=smb1 //ipaserver.example.com/samba /samba/smb1 Enter the Samba user password : *********	Mounting share in runtime
umount /samba/smb1	Unmounting share
vim /etc/fstab //ipaserver.example.com/samba /samba/smb1 cifs _netdev,username=smb1,password=password 0 0 :wq	Making entry in **fstab** file for persistent mount
mount -a	Mounting through **fstab**
mount	Verifying mounted share

Module 8: Networking

Configuring IPv6 Address & DNS IP Address

Configure eth0interface with ipv6 address 2020::1/64 and set DNS address as 2020::2

•Already existing IPv4 network configurations should not be impacted.

Command	Action/Description
nmcli connection modify system ipv6.addresses 2020::1/64 ipv6.dns 2020::2 ipv6.method manual	Configuring ipv6 on ethernet interface
nmcli connection up system	To restart/activate connection
ip address show	To display IP Address configurations
nmcli connection show system	To display connection information
more /etc/resolv.conf	To verify configured DNS IP address
man nmcli	To display Manual page for **nmcli**
man nmcli-examples	To display Manual page for **nmcli-examples**

Configuring Hostname Resolution Using Hosts File

Configure hostname resolution for host system1.example.com using hosts file.

•Set the hosts file as priority for hostname resolution in nsswitch.conffile.

•Test if hostname resolution is working fine.

Command	Action/Description
vim /etc/hosts 192.168.99.20 system1.example.com :wq	To add entry in hosts file
getent hosts system1.example.com	To verify hostname resolution is working fine

Configuring Static Route

Configure static route on system.example.com for destination 10.1.1.0/24via 192.168.99.30.

•Route configuration must be persistent after reboot.

•eth0 should be used as exit interface.

Command	Action/Description
ip route add 10.1.1.0/24 via 192.168.99.30	Adding static route in runtime
ip route show or route -n	To display route(s)
nmcli connection modify system ipv4.routes "10.1.1.0/24 192.168.99.30"	To add persistent route using command line
vim /etc/sysconfig/network-scripts/route-system 10.1.1.0/24 via 192.168.99.30 dev eth0 :wq	To add persistent route using config file
nmcli connection up system	To restart/activate connection

Restricting Specific service to Specific Network using firewall-cmd/firewalld

Configure system.example.com machine to restrict sshaccess to 192.168.99.0/24network.

Command	Action/Description
firewall-cmd --list-all	Displaying firewall configurations
firewall-cmd --add-rich-rule 'rule family="ipv4" source address="192.168.99.0/24" service name="ssh" accept' --permanent	Adding firewalld rich rule to accept traffic form **192.168.99.0/24** network
firewall-cmd --remove-service=ssh --permanent	Removing **ssh** service from services list
firewall-cmd --reload	Reloading firewall to make changes effective
firewall-cmd --list-all	To verify firewall configs after making changes

Note:

Remove ssh service from services list, if you don't remove ssh service, then rich rule configured to accept ssh traffic from 192.168.99.0/24 network only will not be effective. This is due to order in which firewall devaluates the different definitions on firewall. If firewalld will find ssh service on services list, it will allow access irrespective of accessing network and rich rule will be ignored**.**

To Test This:

We have only one network, so it is not possible to test this. To test this working of rule , you just add this rule to allow access for some host not on 192.168.99.0/24network and then test sshconnection from ipaserver.example.com, it must be denied.

Module 9: Managing Security

Introducing SELinux

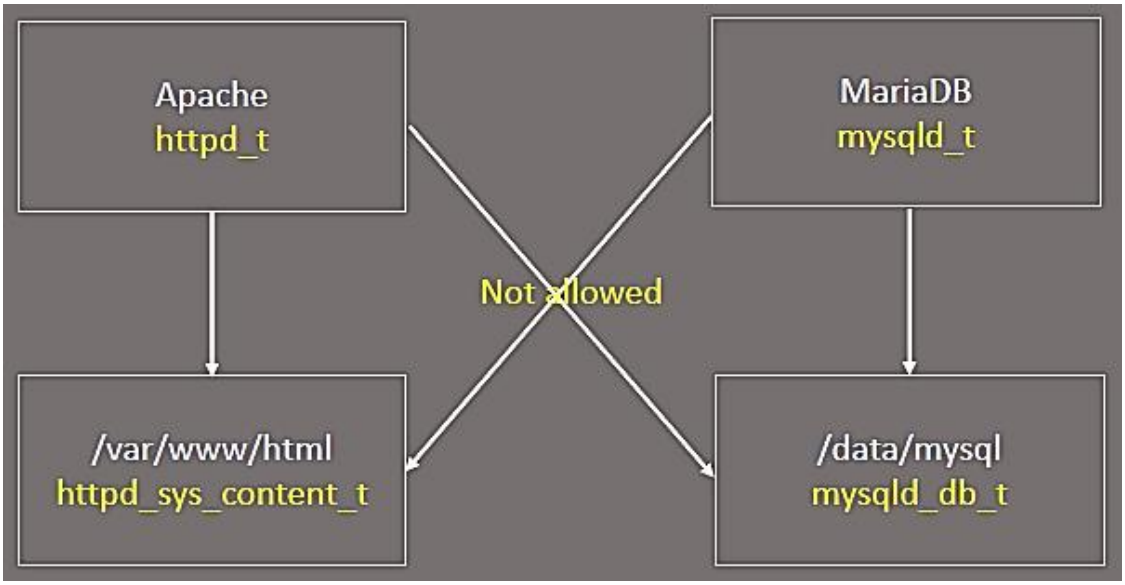
- SELinux implements Mandatory Access Control (MAC). Every process and system resource has a special security label called a SELinux context.

- The SELinux policy uses these contexts in a series of rules which define how processes can interact with each other and the -various system resources. By default, the policy does not allow any interaction unless a rule explicitly grants access.

- SELinux contexts have several fields: user, role, type, and security level. The SELinux type information is the most important when it comes to the SELinux policy, as the most common policy rule which defines the allowed interactions between processes and system resources uses SELinux types and not the full SELinux context.
- For example, the SELinuxtype name for the web server process is httpd_t.
- The type context for files and directories normally found in /var/www/html/ is httpd_sys_content_t

Commands :
ps -eZ -To list SELinuxcontext for Processes
ls -ldZ -To list SELinuxcontext for Directories

Setting SELinux Context Type Persistently



Web server (httpd) needs to access the files in /web directory. Set the correct SELinuxcontext type on /web directory to make this possible.
•Restore the SELinuxContext.
•Changes done should be persistent.

Command	Action/Description
chcon -t httpd_sys_content_t /web	Configuring SELinux context type in run time
semanage fcontext -a -t httpd_sys_content_t "/web(/.*)?"	Configuring SELinux context type persistently
restorecon -R -v /web	Restoring SELinux context
ls -ldZ /web	Displaying SELinux context
man semanage fcontext	To display man page for **semanage fcontext**
man restorecon	To display man page for **restorecon**

Configuring Firewall Using firewall-cmd

Configure the firewall on system.example.com to allow inbound http traffic.

• Changes done should be persistent.

Command	Action/Description
systemctl status firewalld	To check status of **firewalld** service
firewall-cmd --get-services	To get list of **firewalld** services
firewall-cmd --add-service=http	Adding http service on firewall in runtime
firewall-cmd --add-service=http --permanent	Adding http service on firewall persistently
firewall-cmd --list-all	Displaying firewall configs

Configuring Firewall Using firewall-config (Graphical Interface)

Configure the firewall to accept inbound traffic on 443/tcpport.

•Changes done should be persistent.

•Use firewall-configfor this task.

Command	Action/Description
dnf install firewall-config	Install package firewall-config
firewall-config (startx needed)	Launching firewall-config (**startx** needed)
firewall-cmd --list-all	Checking firewall configs

Configuring Key Based Authentication For SSH

Configure password-less sshlogin for system.example.com to establish connection to ipaserver.example.com.

•Use passphrase access to protect the private key.

Command	Action/Description
ssh ipaserver.example.com	Testing SSH connection
ssh-keygen -t rsa	Generating SSH key-pair
ssh-copy-id ipaserver.example.com	To copy public key to SSH Server
ssh ipaserver.example.com	Testing password less connection
cd /root/.ssh	Path where private/public key-pair is stored on client
more /root/.ssh/authorized_keys	File where public key is stored on server side

Setting SELinux Boolean Persistently

List all SELinuxbooleansand set the SELinuxbooleansamba_export_all_rwto 1 to allow Samba server to share exports with r/w permissions.

•Changes should be persistent.

Command	Action/Description
getsebool -a	To list all SELinux Booleans
setsebool -P samba_export_all_rw 1	To set the Boolean persistently

Setting SELinux Context Type on Non-Default Port for SSH Service

Configure correct Selinux context type for ssh service to listen on non-default TCP port 555.

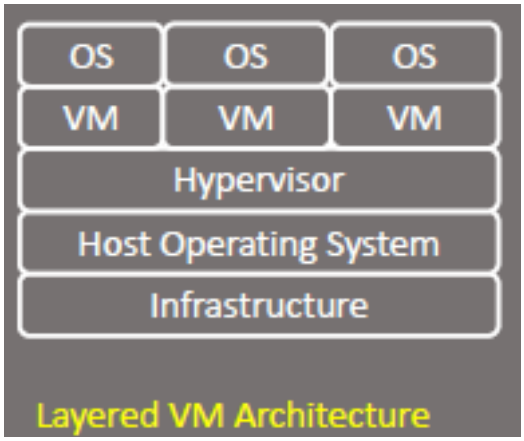
•Changes should be persistent.

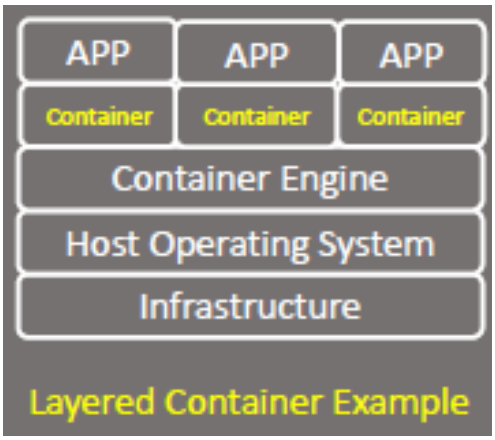
Command	Action/Description
semanage port -l	To list all SELinux context types set on Default ports
semanage port -a -t ssh_port_t -p tcp 555	To set the correct selinux context type on non-default port

Module 10: Containerization

Introducing Containers and Container Images

- Container is nothing but running code in its own isolated environment. All files , settings, tools and resources needed to run container are provided through image called container image.
- Container is launched from container image and can not exist without image.
- Virtual machines are example of virtualization at Infra level, but containers are example of virtualization at OS level. Different containers running on machine share the same operating system but VM's don't.





- Docker Engine is example of popular container engine and containers launched by Docker Engine are usually called Docker containers.
- RedHat does not support Docker in RHEL-8 and built its own daemon less tools to manage individual containers, usually called Linux containers(LXC).
- Also RedHat developed OpenShift for multi node deployments to run multiple containers in different groups known as pods.
- Red Hat Enterprise Linux implements Linux Containers using core technologies such as Control Groups (Cgroups) for Resource Management, Namespaces for Process Isolation.
- For RHCSA exam, You are expected to know how to manage individual Linux containers using set of tools and you don't need to know about Docker Engine and OpenShift Platform.
- podman , buildah, skopeo and runc are tools developed by RedHat to manage individual containers. We will mainly discuss about podman and skopeo which is expected at RHCSA level.
    - podman : Pod Manager for managing pods and container images.
    - skopeo : For copying, inspecting, deleting and signing images.

Searching , Retrieving Images and registries.conf file

To search container images in different container registries, podman search command line is used.

Now what are container registries?

- A container registry is a collection of repositories made to store container images which is normally located on some remote server. You can search image in different registries and pull the image to your machine using podman pull command line.
- You can find list of registries in /etc/containers/registries.conf config file.
- There are there sections in this file :
    - [registries.search] - By default podman search searches registries defined in this section in given order.
    - [registries.insecure] - Insecure registries which does not use TLS are added in this section.
    - [registries.block] - Registries defined in this section are not allowed access from your local system.
- As root user, you can edit the file /etc/containers/registries.conf to change system-wide search settings but as regular user you can create your own registries.conf file in your home directory $HOME/.config/containers/registries.conf to override system wide settings.

Installing Container Tools and Walk Through podman Help

To work with individual containers, We need to install container tools.

Install container tools with following command:

```
# dnf module install -y container-tools
```

We can work with containers as root user as well as regular users(rootless users),functionality available from RHEL 8.1. By setting rootless users, system admin limit potential damage to containers from regular users.

After installing container tools, we will walk through podman man page and help.

Pulling Container Image from Registry & Inspecting Image

Download Apache Web Sever Container image (httpd 2.4) on your System and inspect the container image.

- Check the Exposed ports in container image configurations.
- Authenticate using your RedHat credentials to download image if needed.

Command	Action/Description
podman search httpd	To search Apache image **httpd 2.4** in configured Registries
podman login REGISTRY_PATH	Authenticating to download image
podman pull IMAGE_PATH	Downloading Container Image
podman inspect --format '{{ .Config.ExposedPorts }}' IMAGE_PATH	Displaying Configurations and checking Exposed ports
man podman	To display man page for **podman**
podman pull --help	Displaying help for **podman pull**

Running/Stopping Container and Deleting Image

Run the httpd Container in background from image downloaded in previous task.

- Assign name myweb to this container.
- Verify using podman ps if container is running.
- Stop the container and verify again if container is stopped.
- Delete the container and container image.

Command	Action/Description
podman run --name myweb -d IMAGE_PATH	To run webserver container in **background**
podman ps	Listing running containers
podman stop myweb	Stopping **container**
podman rm myweb	Deleting **container**
podman rmi IMAGE_PATH	Deleting **container image**

Running Apache Service inside Container

Pull Apache Web server image (httpd 2.4) and run the container with name webserver.

- Configure webserver to display content “Welcome to container-based web server”.
- Port 3333 should be used on host machine to receive http requests.
- Start bash shell in container to verify configurations.

Command	Action/Description
podman search httpd	Searching **httpd 2.4** container image
podman pull IMAGE_PATH	Pulling/downloading image to local System
podman inspect IMAGE_PATH	Displaying **config info** contained in Image
podman run --name webserver -d -p 3333:EXPOSED_PORT -v /var/www/html:ORIG_DATA_PATH IMAGE_PATH	Running container and publishing port(s)
vim /var/www/html/index.html Welcome to container-based web server :wq	Creating **index.html** on host and adding me
podman restart webserver	Restarting container **webserver**
curl http://system.example.com:3333	Verifying if webserver serves **index.html**
podman run --help	Getting help for **podman** command line

Configuring System to Start Container as Systemd Service at boot

Configure System to start webserver container at boot as SYSTEMD service.

- Start/enable SYSTEMD service to make sure container will start at boot.
- Reboot System and verify if container is running as expected.

Command	Action/Description
podman generate systemd webserver	Generating **systemd unit file** for container
vim /etc/systemd/system/httpd-container.service #Paste generated systemdd unit file contents here# :wq	Creating service unit file with name **httpd-container.service**
systemctl daemon-reload	**Reloading systemd** to make changes effective
systemctl start httpd-container.service	Starting **httpd-container.service** service
systemctl enable httpd-container.service	Enabling **httpd-container.service** service
systemctl status httpd-container.service	Displaying status of **httpd-container.service** service
systemctl reboot	Reboot System
systemctl status httpd-container.service	Displaying status of **httpd-container.service** service again to service is **Active(running).**
man systemd.service	To display man page for **systemd service units**

Running Mariadb Service inside Container

Pull Mariadb Server image to your System.

- Run container mariadb from the image and publish Exposed port.
- Set root password for mariadb service as mysql.
- Verify if you can login as root from local host.

Command	Action/Description
podman search mariadb	Searching **mariadb** image
podman login REGISTRY_PATH	Authenticating to pull image
podman pull REGISTRY_PATH/IMAGE_NAME	Downloading image to **System**
podman inspect IMAGE_PATH	Displaying configs inside image e.g. Exposed
podman run --name mariadb -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=mysql -v /var/lib/mysql:PREFIX/var/lib/mysql IMAGE_PATH	Running **mariadb** Container and publishing Exposed port.
dnf install mysql	Installing mysql on **System** Host
podman inspect --format '{{ NetworkSettings.IPAddress }}' mariadb	Listing **IP address** assigned
mysql -h IPAddress -u root -p	Connecting to **mariadb** database

Extending Privileges to Containers

There is possibility to run privileged or non-privileged containers using podman. They behave and act differently on the Host.

Below are features that can be opened from a container to host.

- Privileges : A privileged container (--privileged) runs applications on host as root user. So, you can delete files/directories mounted from host to container even if they are owned by root.
- Process Tables: A non-privileged containers can see only processes running inside container but can not see processes running on host. Privileged container (--pid=host) on the other hand can see all processes (use ps -e) running on host as well.
- Network Interfaces : By default, container has one external network interface and one loop back interface, but privileged container (--net=host) allows to access all network interfaces directly from container.
- Inter-process communications: A privileged container can use IPC facility to see information about active messages queues, shared memory segments etc.
- To explore different options, check podman run --help or man podman-run.

Understanding Runlabels Within Image

Some Special RedHat container images come with labels which provide pre-defined command lines to work with those images.

To run the pre-set command line in IMAGE, we need to use below command:

```
# podman container runlabel <label> IMAGE
```

Different Labels include:

- install : Sets up host system to run the container. It creates different files and directories on host that container will user later .
- run : Run the container with different command line options. Normally it opens privileges to host and mount the filesystems from host to container.
- uninstall: Cleans up the host system when container has been run.

*Example of RedHat image which includes runlabels is rsyslog. In next task, we will run rsyslog container using runlabels.

Running rsyslog Container Service Using Runlabels

Pull rsyslog image (registry.redhat.io/rhel8/rsyslog) to your System.

• Run container as root user using runlabels defined within the container image.

• Verify working of container and create systemd unit file to start container at boot.

Command	Action/Description
podman login registry.redhat.io	Authenticating to registry
podman pull registry.redhat.io/rhel8/rsyslog	Pulling image to System
podman container runlabel install --display registry.redhat.io/rhel8/rsyslog	Displaying install runlabel
podman container runlabel install registry.redhat.io/rhel8/rsyslog	Running install runlabel
podman container runlabel run --display registry.redhat.io/rhel8/rsyslog	Displaying run runlabel
podman container runlabel run registry.redhat.io/rhel8/rsyslog	Running run runlabel
podman generate systemd -f rsyslog	Execute this command at path /etc/systemd/system to create unit file
systemctl start *.service and systemctl enable *.service	Starting and enabling service to start container at boot

Introducing Rootless Containers and Pre-requisites

podman allows us to run rootless containers (as normal user) which is good feature and provides more security.

To run rootless containers, execute below steps as root user.

- Install podman and slirp4netns packages (If not installed already):

```
# dnf install podman slirp4netns -y
```

- Increase Username spaces in Kernel : User namespaces isolate security related identifiers e.g., User IDs and Group IDs.

```
# echo “user.max_user_namespaces=28633” > /etc/sysctl.d/userns.conf
# sysctl -p /etc/sysctl.d/userns.conf
```

For more information, check manual page, man user_namespaces

Running httpd Container as Rootless User

Run rootless httpd container with name web as user rhcsa using image (registry.redhat.io/rhel8/httpd-24).

- Pull image to system and inspect the image.
- Use same index.html file we used earlier and port 3400 should be used on host to receive http requests
- Create user rhcsa and set password as password.

Command	Action/Description
useradd rhcsa	Creating user
passwd rhcsa	Setting password
podman login registry.redhat.io	Authenticating login to Registry (execute command as user rhcsa)
podman pull registry.redhat.io/rhel8/httpd-24	Pulling image (execute command as user rhcsa)
podman run -d --name web -p 3400:EXPOSED_PORT -v /var/www:HTTPD_DATA_ORIG_PATH IMAGE	Running container (execute command as user rhcsa)
podman ps	Verify if container is running
curl http://localhost:3400	Verifying working of httpd service

Configuring System to Start Systemd Service as Specific User

Configure System to start container web at boot.

• Create system unit file with name web-container.service for same purpose.

• You must perform this action as user rhcsca.

Command	Action/Description
mkdir -p /home/rhcsa/.config/systemd/user	Creating Directory Path
podman generate systemd web	Generate systemd unit file and put the contents in /home/rhcsa/.config/systemd/user/web-container.service file
loginctl enable-linger rhcsa	Enabling lingering capability for user to start service at boot with systemd user service manager.
systemctl --user daemon-reload systemctl --user start web-container.service	Starting service using user instance of systemd
systemctl --user start web-container.service	Enabling service using user instance of systemd
systemctl --user status web-container.service	Checking status of containerized service
systemctl reboot	Rebooting system to verify if containerized service is started at boot (as root user)

Module 11: Shell Scripting

Introducing if Statement and Syntax of if statement

Syntax of if Statement :

Example 1: Using Command as condition

```
if some command then
Command(s) to be executed
else
Command(s) to be executed
fi
```

If statement decides to execute code based on exit status($?) of command . Zero exit status means SUCCESS and non-zero exit status means FAILURE.

Example 2:

```
if [ some test ] or if test some test
then
Command(s) to be executed
elif [ some test ]
then
Command(s) to be executed
else
Command(s) to be executed
fi
```

For different tests, We can check man page for test, man test

Using if test to Compare Integers

Create a script compare.sh to find largest number out of three numbers.

• On execution, It should ask to enter numbers.

```
#!/bin/bash
read -p "Enter first number:" num1
read -p "Enter second number:" num2
read -p "Enter third number:" num3
if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]; then
      echo "$num1 is largest number"
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]; then
      echo "$num2 is largest number"
else
     echo "$num3 is largest number"
fi
```

Introducing for Statement and Syntax of for Statement

Syntax of for Statement:

```
for VAR in VALUE1 VALUE2 …. VALUEn
do
Some Command(s) using $VAR
done
```

Practical example 1:

```
for username in ex200 ex294
do
useradd $username
done
```

Practical example 2:

```
for username in `cat userlist`
do
useradd $username
done
```

Adding Users Using for Statement with Input file

Create a script create_user.sh under root directory to create users.

- It should read usernames from file /root/userlist and display on STDOUT "User 'username' already exists" if user already exists.
- Otherwise, It should prompt for password for user.
- Set password password for each user.

```
#!/bin/bash
for user in `cat /root/userlist`
do
if grep $user /etc/passwd > /dev/null
then
        echo "User '$user' already exists"
else
        useradd $user
        passwd --stdin $user
fi
done
```

Understanding Command Line Arguments for Script

Processing Script Inputs Using $1,$2….

We can create script to accept variable values as command line arguments.
$0 - Script Name
$1 - First Argument
$2 - Second Argument and so on
$# - Total number of arguments provided
#@ - List of all arguments provided

We will understand this with the help of simple example.
Create a script users.sh under root directory to create users.
• It should accept username(s) as command line arguments.
• It should display "No username provided" if executed without any argument.
• It should display "User already exists, or something wrong happened" if user is not created.

```
#!/bin/bash
if [ "$#" -eq 0 ]; then
    echo "No username provided"
else
    for user in "$@"
    do
        useradd $user 2> /dev/null
          if [ "$?" = "0" ]; then
                echo "User $user has been created"
           else
                echo "User $user already exists, or something wrong happened"
           fi
      done
fi
```