RHCSA 9

#who	:show current connected users.
#w	:same as who but with more details.
#chvt 3	:change to the virtual terminal N° 3, or you can press **ctrl+alt+f3**.
#man ls	:to get ls command documentation.
#man –k or **apropos**	:to search for a command in mandb based on keywords, but you need to run the **mandb** command first.
#vim file1	:file editor, you can use some commands while using vim:

yy :copy text,
dd :delete line,
p :paste,
gg :go to the top of the file,
G :go to the end of file,
^ :go to start of line,
$:go to the end of line,
/hi :search for the word hi,
%s/old/new/g :replace word “old” with the word “new”
se number :show lines number,
a :start insert mode,
o :insert line.

#vimtutor	:vim course.

Redirection:

>	:redirect output into file (if the is already exit it will delete the contenet before).
>>	:append output into the end of file.
2>	:redirect errors.
Pipping |	:is to use the output of the first command as an input for the second on (e.g cat/etc/hosts | grep linda).
#history	:print commands history.
~/.bash_history	:history file.
HISTSIZE and **HISTFILESIZE**	:to define entries number.
#history –w	:synchronize history to .bash_history.
#history –c	:clear history.
#history –d nn	:delete command nn from history

Expansion:

#ls *	: show all
#ls a?*	: show all file’s name starts with a and followed at least by 1one character.
#ls [a-e]*	:show all files start with a or e.
#ls [a..e]*	:show all files start with a,b,c..e.
#touch file{1..9}	: create from file1 to file9.
#useradd {linda,bob,rose}	: create users linda, bob and rose.

Variables

#key=value	:local variable e.g: #color=red.
#echo $color	:print variable value
#export color=red	:local variable for bash and subbash

RHCSA 9

#alias dir='ls –ltr' :define costume command.

- ⇨ Also can be configure on .bashrc or .bash_profile, to become persistent.
- ⇨

files

#which passwd :show command file path.

#find / -name "hosts" :prints all file with the name hosts.

#find /etc –type f –size +100M :list all file within a size bigger than 100M.

#find /etc -size +1M –exec grep –l student {} \; : -exec: call for another command
{} use the previous output as input
\; close exec.

#find / -user wassim : show all files owned by a specific user.

#find / -perm g+s or **find / -perm /4000** :show all files that have the a specific permission (in this case suid)

#cut file1 :print file1 content.

#cp -ar /etc/passwd . : copy file to the current folder

#cp -ar /etc/shadow /home/alice : copy file to alice home

#tr '[:lower:]' '[:upper:]' </etc/hosts> ./upperfile : made a copy from hosts file to the current directory uner the name upperfile, and change all lower case to upper case.

#cut -d: f7 /etc/hosts : show the 7th field from each line in the file passwd.
-d: specify delimiter used to separate fields in text line
f7 specify which field to extract.

# mv /etc/login.defs /opt/doc : move login.dfs to /opt/doc

Links

#ln –s /etc/hosts ~/link1 :symbolic link/ soft link.

#ln –p /etc/passwd ~/phylink :physical link/ hard link.

Archive

#tar –cvf arch.tar /etc	**:**create archive (without compression).
#tar –czvf arch.tar.gz /etc	:compress with gzip.
#tar –cjvf arch.tar.bz2 /etc	:compress with bzip2.
#tar –cJvf arch.tar.xy /etc	:compress with xy.
#tar –tvf arch.tar	:print archive content.
#tar –xvf arch.tar.gz /extract_path.	:extract archive file. Can add **–C**
#tar –uvf arch.tar /tmp	:append archive.
#tar –rvf arch.tar /etc	**:**update archive.

grep

grep: is a tool to filter an ouput or to to find a specific information in a text file.

RHCSA 9

#ps aux | grep sshd	:to get all sshd processes.
#grep Wassim /etc/*	:filter all the files contain word "Wassim"(shows the file and the lines).
#grep –I wassim /etc/*	:intensive filter, ignore uppercase and lower case.
#grep –A 5 –B 5 allow /etc/ssh/sshd_config	**:**show 5 lines after and 5 lines before the specific word.
#grep –R root *	:recursive search in the current directory and sub directory.
#grep –l wassim /etc/*	:print only the file that continent the text Wassim in it.
#grep '^w' file1	:grep all lines satarts with w.
#grep '$nova' file1	:grep all lines ends with nova.
#grep 'anna\b' file1	:grep word ends with anna
#grep 'b.*t' file1	:grep words starts with b and ends with t, and whatever the nbr of character in between is 0 or more.
#grep 'b.+t' file1	: 1 or more.
#grep 'b.?t' file1	:0 or 1 caracter in between.
#grep 'bo\{3\]t} file1	:start with b and ends with t and o repeated 4times (e.g boooot).

Root privileges

#su -	:switch to root user.
#usermod –aG wheel Wassim	:memebers of the group wheel are allowed to use sudo.
etc/sudoers	:sudoers config file
etc/sudoers.d	:drop-in files folder for sudoers config.

****drop-in files config:**

#vim /etc/sudoers.d/lisa

Lisa ALL=/usr/bin/passwd ,!/usr/passwd root
=> lisa allowed to changer other users password but not the root password.

#vim /etc/sudoers.d/group_users

%users ALL=/usr/sbin/mount /dev/sdb
=>members of the group users are allowed to mount sdb.

Users and Groups

#useradd wassim :create uses (with default settings).

#useradd supernova –d /home/dir_super –g tekup –G wheel –s /sbin/nologin –u 3004

- -d : to specify the user's home directory
- -g : to specify user's primary group
- -G : supplementary groups
- -u : to specify the user's UID
- -s : to specify user's shell, example of shell:
  - /sbin/nologin : this shell prevents interactive login.
  - /bin/sh : It's less feature-rich than Bash but still provides a shell environment.
  - /sbin/bash : full interactive shell login.

RHCSA 9

#usermod –L linda : the user linda is locked.

#usermod –U linda : unlock the user.

#usermod –e 2023-09-12 linda : change user expiration date.

#usermod –s /bin/bash linda :change user shell.

#userdel linda :delete user.

#newgrp dev linda :temporary primary group.

#groupadd finance : create new group named fianance.

#groupdel dev : delete a group.

#lid –g finance : list all members of the group finance.

/etc/default/useradd : config file of useradd.

/etc/login.defs : default setting of creating a new user.

/etc/skel : files on this folder will be created to the user home directory upon creation.

#passwd linda : change linda password.

# chage –d 0 linda : force linda to change the password next time login
=> chage options:
-d set date of last password change to LAST_DAY.
-E: set account expiration date to EXPIRE_DATE.
-I : set password inactive after expiration to INACTIVE.
-m set minimum number of days before password change to MIN_DAYS.
-M set maximum number of days before password change to MAX_DAYS.
-W : set expiration warning days to WARN_DAYS.

/etc/security/pwquality.conf : password policy config file.

/etc/shadow : where the password hashage is stored.

File owner ship

Permission on folder and files are: **read=r=1, write=w=2, execute=x=1**.

#chown linda file : change the file owner to linda.

#chown :dev file : change the file groupe owner to dev.

#chgrp dev file : change group owner.

#chmod 750 file : change file access permission.

#chmod g+x o-rw u+rwx file : g+x add the execute permission for group.

o-w : restric the permissions of read and write for others.

u+rwx : add the all permissions to the user (owner).

#chmod g+s folder : setgid : every file or folder will be created on this folder, it will be owned by the group ownership of the parent folder, can write also :

chmod 2751 folder

#chmod +T folder : sticky bit : to make only the owner of file/ sub-directory can delete it.

chmod 1751 folder

umask: is a shell setting that subtract the umask value from the default permission on folde/file.

linkedin.com/in/saadaouiwassim

RHCSA 9

Default permission on a folder are 777 and for a file are 666

- e.g umask 022 on a file:
- Owner: 6 (read and write) - 0 (umask) = 6
- Group: 6 (read and write) - 2 (umask) = 4
- Others: 6 (read and write) - 2 (umask) = 4

you can set mask by running the cmd **umask** on the shell, or added it to .bashrc or .bash_profile

e.g: umask 752 or echo "umask 752">> ~/.bashrc

#getfacl file	: show file access list.
#setfacl -m u:alice:rw f1	:add the user alice to f1 ACL and permissions are set to rw.
#setfacl -m g:operations:rw f1	: add the group operations to the acl.
#setfacl -m u:bob:rw,g:tekup:rw f1	: add user and group to the acl.
#setfacl -m d:u:bob:rw folder1	: the option d is only applied to folders, to make sure the acl will be inherited to sub-folders and files.
#setfacl -b f1	: delete all acl for f1
#setfacl -x g:tekup f1	: delete groupe tekup from acl.
#getfacl -R f1 >acl.save	: save acl to a file
#setfacl -b f1	: delete all f1 ACL
#setfacl --restore=acl.save f1	: retore f1 acl

Network

#ip link show	: show current network devices.
#ip addr show	: show network device configuration.
#hostnamectl hostname tekup	:change hostname.
/etc/hosts	: contain hosts and their IP.
/etc/resolv.conf	: contain dns config.
/etc/nsswitch.conf	: hostname resolution.

When you try to ping google.com, your system will check /etc/hosts then dns, and then your hostname.

#ip addr add dev ens160 10.0.0.1/24	: add temporary address.
#ip route 2.2.2.2/24 via 10.0.0.1/24	: add route to the route table.
/etc/NetworkManager/system-connection	: folder where you will find the file for you network config.
#nmcli general permission	: to check permission on network manager.
#nmcli con show or nmcli dev status	: show connection.

# nmcli connection add con-name mycon ifname ens160 type ethernet ip4.add 192.168.1.11 ipv4.gatway 192.168.1.2 ipv4.method manual

⇨ This command is used to a new connection

- **con-name**: to set the new connection name
- **ifname**: to set the interface name
- **type**: to set the connection type
- **ipv4.method**: used to specify the method used to configure IPv4 settings for a network connection, **auto** (the coonection will receive ip@ from DHCP, so the address we set may will be change), **manual**(static address, so the address we set won't change).

RHCSA 9

#nmcli con up mycon : to activate the connection.

# nmcli connection modify ens160 ipv4.add 192.168.130.229/24 ipv4.gateway 192.168.130.1 ipv4.dns 192.168.130.254 +ipv4.dns 8.8.8.8

➔ modify my current connection settings (this's what you will be asked for on the RHCSA exam).

# nmcli connection reload ➔to reload all connection files

#nmtui : console where you can set hostname & connections.

#ss : investigate sockets.

Managing software

RPM : software on RHEL is installed using packages in RPM format.

#rpm –qa : show all installed packages.

#rpm –qf gedit : show from which packages gedit was installed.

#rpm –ql gedit : Shows u what files were put on ur computer when u installed a program using RPM

#rpm –q --scripts podman : shows the executed scripts while installing the package.

Setting up Ripository access

#dh -h : check for available space (check / it should have available space >10GiB, or it won't work

#dd if=/dev/sr0 of=/rhel.iso bs=1M :copy files from sr0 to rhel.iso.

#mkdir /repo : mount point.

#echo "/rhel.iso /repo iso9660 defaults 0 0" >> /etc/fstab : persistent mount.

#mount -a

# dnf config-manage --add-repo="file///repo/BaseOS" : add baseos repository to /etc/yum.repos.d

#ls /etc/yum.repos.d : to check for the file.

#dnf repolist : list of repository.

or can be created manually

#vim /etc/yum.repos.d/BaseOS.repo

```
>[repo_BaseOS]
>name=BaseOS
>baseurl=file///repo/BaseOS
>enable=1
>gpgcheck=0
```

➔disable gpg key check.

Managing packages with dnf

#dnf list "selinux" :list selinux installed and available packages.

#dnf seach seinfo :search in packages name and summary.

#dnf search all seinfo :search even in description.

#dnf provides */containerfile :search for packages that provides a specific file.

#dnf info httpd : show package info.

RHCSA 9

#dnf install firewalld	: install package.
#dnf update	: update installed packages.
#dnf group list	: show Available Environment Groups.
#dnf group list hidden	: show all available groups (include not installed groups).
#dnf group info "vitualization host"	: list packages within group.
#dnf group install "ftp server" --with-optional	: install with optional packages.
/var/log/dnf.rpm.lop	: all transaction that dnf perform.
#dnf history	: summary of all installation and removal transaction.
#dnt history undo n	: undo transaction number n.

process and jobs

#command &	: run a command on background.
#jobs	: list all jobs that runs on background.
#fg 1	: run back job 1 to forward ground.
#ps aux	: review of all processes.
# ps -fax	: shows heirarical relation between processes.
#ps -fu wassim	: show all processes owend by wassim.
#ps -f --forest -c sshd	: show a process tree.
#ps L	: show you a list of threads.
#ps -eo pid, ppid, user, cmd	: some specifiers to show a list of processes.
#top	: realtime process monetering.
>f	: shows available display field.
#nice -n 19 dd if=/dev/zero to of=/dev/null	:nice and renice are used to set the priority of a processit value between -20 and 19, -20 highest priority and 19 is the lowest.
#renice -n 19 PID	: renice process using PID.

Profiles

#sysctl -a	: show system variables.
#sysctl vm.swappiness=40	: change variable value.

problem: sysctl containes about 1004 variable, which means it's hard to set all those variable.

solution: is to use a profile or to custom profiles.

#tuned-adm list	: shows all available profiles.
#tuned-adm profile my profile	: set a profile.
#mkdir /etc/tuned/myprofile	: create folder form my custom profile.
#vim /etc/tuned/myprofile/tuned.conf	:create my profle

```
>[sysctl]
>vm.swappiness=66
```

#tuned-adm profile myprofile	: change profile to myprofile.
#sysctl -a | grep vm.swappiess	: check changes.

Note: we should change the value **reapply_sysctl** to 0 in the config file /etc/tuned/tuned.main.conf.

#loginctl	: manage users and sessions.

linkedin.com/in/saadaouiwassim

RHCSA 9

Managing system services

#systemctl enable httpd	: enables service.
#systemctl disable httpd	: disable service.
#systemctl status httpd	: show service status.
# systemctl start httpd	: start service.
#systemctl stop httpd	: stop service.
#systemctl reload httpd	: reload service without stopin' it.
#systemctl restart httpd	: restart service.
#systemctl edit httpd	: edit service file config.
#export EDITOR=/usr/bin/vim	: to change system editor.
#systemctl list-dependencies sshd.service	: complete overview of all currently loaded units and their dependency.

Problem: some services cannot run simultaneously with other services on the same system.

Solution: is to use mask which will create a symbolic link to /dev/null the unwanted service to ensure that it cannot be stared

#systemctl mask sshd.service	: mash sshd.service.
#systemctl unmask sshd.service	: unmask the masked service.

**** **sretart a service automatically in failure case.**

```
#systemctl edit httpd.service
        >[service]
        >Restart=always
        >RestartSec=5s
#systemctl restart httpd.service
```

Task scheduling

Timer:

#systemctl list-units -t timer	: show all active timers.
#systemctl list-unit-files *.timer	: show all configured timer file.

#systemctl /etc/system/system/tmp-clean.service : configure service file that will started by timer

```
        >[Unit]
        >Description=clean tmp
        >[Service]
        >type=oneshot
        >ExecStart=/usr/bin/ rm -rf /tmp/*
```

#vim /etc/system/system/tmp-clean.timer : timer file that will run the service file tmp-clean.service.

```
        >[Unit]
        >Description=Run tmp-clean.service weekly
        >[Timer]
```

>OnBootSec= 15min : service will be trigged 15min after the system boot.

>OnUnitActiveSec= 1w : this option will make sure service will be trigged only after one week sence the last time.

RHCSA 9

>**Persistent=true**	: when a timer is missed, persistent will make sure it will be run immediately when the system boot.

other timer's option:

OnActiveSec= 1h : to make sure the service is trigged 1h after booting or after starting the timer.

OnBootSec= 15min : service will be trigged 15min after booting the system. the old option was (OnStartupSec= 15min).

OnUnitSec= 1w : to make sure that timer will wait for 1week to triggers the service again.

PS: you should add Persistent=true , to make the service will be trigged immediately next reboot if the timer is missed.

WakeSystem= true : it will wake the system from {sleep or suspend mode} to run the service.

OnCalendar= 10:00 : trigged the service daily at 10:00.

OnCalendar= Mon 10:00 : every monday at 10:00.

OnCalendar= Sun..Fri 13:00,19:00 : from sunday to friday at 13H and at 19H, and after that the service won't be trigged again.

OnCalenda= Mon, Tue *-*-* 14:00: every Mon & Tue at 14H.

On calendar= 2023-07-12..2023-07-23 19:00 :from 07/12 to 07/23 at 19H

******Crontab**

#crontab -e	: create cron job
/etc/crontab	: a file that shows how to setup a cron job.
/etc/cron.d	**:** directory to drop you cronjob config file.

Logging

#journalctl	: show the entire journal.
#journalctl -p err	: show errors only.
#journalctl -f (synchronization).	: show the last 10lines+ adds new massage
#journalctl -u sshd	: show journal of specific service.
#journalctl --since "-1hour"	: show journal of the last 1hour.
#journalctl --since today	: show today journal.
#journalct -o verbose	: detailed journal.
/etc/logrotate	: logrotate config file
logrotate.timer	: timer to clean log files.

Managing storage

#df -h	**:**shows available space.
#lsblk	: print block devices.
#blkid	: shows block's UUID.
/proc/partition	: partition and disk statistics.
/etc/fstab	: persistent mount file.
/run/systemd/generator/	: folder for drop-in mount files (not encluded on RHCSA exam)

to create a new partion you can use: **fdisk**, **gdisk** or **parted.**

*****create and mount a new partion:**

#fdisk /dev/sda	**:**create a new partition from sda device
>m	:for help.

RHCSA 9

>n	: create new partition
>w	: write to disk table and exit.
>q	: exist without saving changes.

#mkfs.xfs /dev/sda1 : create xfs file system on sda1 partition.

#mkdir /xfs1 : folder which will the mount point for the new partion

# mount /dev/sda1 /xfs1 : for temporary mounting (will be discard after reboot).

#echo " /dev/sda1 /xfs1 xfs default 0 0">> /etc/fstab : persistent mount

# mount -a : to make sure to mount all unmounted devices.

#findmnt --verify : you can always use this command to verify if there's any syntax errors in /etc/fstab.

****** mounting usig UUID and LABEL**

#blkid	: to get block UUID.
#tune2fs -L	: set a label on ext file system.
#xfs_admin -L	: set a label on xfs file system.
#mkfs.* -L	: set a label while creating a file syetem.

in case of cloning device two device will be have the same UUID, so u need to use the command:

#xfs_admin -U generate /dev/sda1 : to get a new UUID.

****UUID**

#blkid

#echo "UUID=….. /xfs1 xfs defaults 0 0" >> /etc/fstab

#mount -a

*****LABLE**

#xfs_admin -L super /dev/sda1 : set label named "super" to /dev/sda1 (xfs file system).

#echo "LABLE=super /xfs1 defaults 0 0" >>/etc/fstab

#mount -a

*****Swap**

#fdisk /dev/sda	: to create a new partition for swap
>n	: new partition
>t	: to change partition type
>8200	: linux swap hexa code
>w	: save changes.
#mkswap /dev/sda2	: create swap file system on /dev/sda2.
#swapon /dev/sda2	: activate the new swap partition.
#swapoff /dev/sda2	:disactivate the swap partition

#echo "/dev/sda2 none swap defaults 0 0" >>/etc/fstab

*****LVM creation**

#pvcreat /dev/sda1 :create physical volume

#vgcreate vgdata /dev/sda1 : create a volumle groupe named vgdata from the physical volume /dev/sda1.

#lvcreate -n lvdata -L 1G vgdata : create logical volume named lvdata with a size of 1GiB from vgdata.

#mkfs.xfs /dev/vgdata/lvdata : create file system on lvdata.

#echo "/dev/vgdata/lvdata /lvfolder xfs defaults 0 0" >>/etc/fstab

RHCSA 9

#mount -a

*****extent (which mean to set volume based on block size)**

#vgcreate -s 8M vgdata /sda1 :set the physical extent volume (echa block size is 8MiB)

#lvcreate -l 2 -n lv1 vgdata : create a logical volume within a size of 2 blocks,each block's sized of 8MiB.

*****extend LVM size**

#vgextend vgdata /dev/sda2 :extend vg volume.

#lvextend -r -l +50%FREE /dev/vgdata/lv1 : add 50% of the free space on vgdata to lv1

#lvextend -r -l +2 /dev/vgdata/lv1 : extend lv1 with two blocks.

#lvextend -r -L +1G /dev/vgdata/lv1 : add 1GiB to lv1.

#lvextend -L +1G /dev/vgdata/lv1 /dev/sda2 : specify from which physical volume you will add more volume to lv1.

*****reduce volume**

#pvmove -v /dev/sda2 /dev/sda1 : move all the contenant of sda2 extents(blocks) to sda1.

#vgreduce vgdata /dev/sda2 : reduce vgdata volume.

*****startis volume**

#dnf install stratis-cli startisd

#systemctl enable --now stratisd

#stratis pool create mypool /dev/sda1 : create pool named mypool with the volume sda1.

#stratis pool --add-data mypool /dev/sda2 : add more volume to mypool.

#stratis pool list :list all created pools.

#stratis blockdev list : list all pool blockdevices.

#stratis fs create mypool myfs1 : create file system on mypool.

#echo "UUID=… /myfs1 xfs defaults,x_systemd.requires=stratisd.service 0 0">>/etc/fsrab

➔mount stratis fs.

*****stratis snapshot**

#stratis fs snapshot mypool myfs mysnap : create a snapshot of myfs

#mkdir /mysnap :create mount point

#mount /dev/stratis/mypool/mysnap /mysnap : mount mysnap on /mysnap

#ls -l /mysnap :check /mysnap continent.

#stratis fs destroy mypool myfs : destroy myfs.

Boot procedure

/etc/default/grub : to edit persistently Grub2 parameters.

#grub2-mkconfig -o /boot/grub2/grub.cfg : to compile changes to grub.cfg on xfs file system.

#grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg : to compule changes on EFI file system.

Boot procedure

/etc/default/grub : to edit persistently Grub2 parameters.

#grub2-mkconfig -o /boot/grub2/grub.cfg : to compile changes to grub.cfg on xfs file system.

RHCSA 9

#grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg : to compule changes on EFI file system.

....

System targets

multi-users.target	:multi users without graphical environment.
graphical.target	:graphical desktop environment.
rescue.target & emergency.target	: for troubleshooting and system recovery.
#systemctl get-default xxx.target	: get the current default target
#systemctl set-default xxx.target	: set the default target.
#systemctl isolatre xxx.target	:change target on the running time.

also you can boot into specific, from boot screen press “e” and add system.unit=xxx.target to the end of line linux.

*******root password recovery**

step1: from boot screen press “e”.

step2: add init=/bin/bash to end of line linux

#mount -o remount, rw /	: to change into read-write mode on /
#passwd root	:change password.
#touch /.autorelabel	: to resolve problem related to selinux.
# exec /usr/lib/systemd/systemd	:to restart the system on a normal way.

....

Shell bash scripts

*****if….else**

#vim test.sh

```
>#!/bin/bash
>if test -z “$1”
>then
>       echo you have to provice argument
>       exit 3
>fi
>if test “$1”=”hello”
>then
>       echo you tayped hello
>else
>       echo you typed something else
>fi
>if [ -f $1 ]            ➔ same as test -f $1
>then
>       echo $1 is a file
>else
>       echo $1 is not a file
>fi
```

#chmon u+x test.sh	: to make the file executable.
#. test.sh	: run the file

linkedin.com/in/saadaouiwassim

RHCSA 9

****** for**

```
#vim n-bonj.sh
        >#!/bin/bash
        >read num
        >for((i=1; i<= $num; i++))
        >do
        >echo bonjour n: $i
        >done
```

#chmon u+x n-bonj.sh : to make the file executable.

#. n-bonj.sh : run the file

******while**

```
#vim break-counter.sh
        >#!/bin/bash
        >if [ -z $1]
        >then
        >       echo provide beak duration in minute
        >       read COUNTER
        >else COUNTER= $1
        >counteur=$5((COUNTER*60
        >while [ $counter -gt 0
        >do
        >       echo $counter seconds remaining
        >       counter=$(( counter -1))
        >       sleep 1
        >echo break is over
```

#chmon u+x break-counter.sh : to make the file executable.

#. break-counter.sh

<u>Note: you can use</u> bash -x to to see in details what's the script doing while running

#bash -x break-counter.sh

you can ckeck man test for test options

SSH

#ssh-keygen :distribute ssh key

#echo "192.168.133.229 sever" >>/etc/hosts : add server ip to hosts

#ssh-copy-id server : send key to server.

➔ in case you create key protected with passphrase ➔everytime you try to to run a command on the remote server, you will be asked to confirm the passphrase

<u>Solution:</u>

#ssh-agent /bin/bash

#ssh-add

RHCSA 9

#ssh -X server gedit : the optin -X is used to run application on the remote server.

/etc/ssh/ssh_config : client-side configuration file for OpenSSH.

/etc/ssh/sshd_config : server-side config file.

#scp file1 file2 student@server:/home/student : copy file securely.

#rsync -a * server:/home/student/ : synchronize files between current directory and remote server path.

http server

#dnf install httpd

/etc/httpd/conf/httpd.conf	:config file
/etc/httpd/conf.d/	:drop-in files storing folder.
/var/www/htdocs	: defaults document root.

Selinux

#getenforce	: show current selinux state
#setenforce	: change between enforcing & permissive mode.
/etc/sysconfig/selinux	: config file.

selinux also can be context from the boot menu, in the end of line "linux".

enforcing=0	: permissive mode
enforcing=1	: enforcing mode
selinux=0	: disable selinux
selinux=1	: enable selinux

#semanage fcontext :to manage selinux context

#restorecon -Rv /var/www/html : to inherite selinux policy of the parent folder.

#semanage -a -t httpd_sys_content_t "/web(/.*)? to change selinux policy of the flolder /web.

#restore -Rv /web :apply the new policy.

*****how to get the right selinux policy**

#dnf install selinux-policy-doc : selinux policy documentation.

#man -k _selinux |grep httpd : show all selinux policy related to httpd.

*****changing port for ssh to 2022**

#semanage port -a -t ssh_port -t -p tcp 2022

*****Boolean**

#getsebool -a |grep ftpd : show all selinux Boolean related to ftpd.

#set -P ftpd_use_nfs on : activate a Boolean

#semanage boolean -l -c : shows all boolean that have non default settings.

#journalctl | grep sealer : to get all selinux alerts.

#grep AVC /var/log/audit/audit.log: used to search for entries related to AVC (Access Vector Cache) denials in the audit log file on a system with SELinux enabled.

RHCSA 9

Firewalld

#ss :shows all sockets

#ss -tu :show connected tcp and udp sockets

#ss -tua :show sockets that are in listening state.

#firewall-cmd --list-all : list complete config of the firewall

#firewall-cmd --get-services : show all managed services by the firewall

#firewall-cmd --add-service http : allow the http service temporary

#firewall-cmd --add-service http --permanent : allow the http service permanently.

#firewall-cmd reload : reload firemwall config.

Mannaging time

#hwclock --hctosys : set time from hardware clock to system time.

#hwclock --systohc :set time from system to hardware clock.

#date : show date and time.

#timedatectl status :show current time settings.

#timedatectl set-time : set system time.

#timedatectl set-timezone : set system time zone.

#timedatectl set-ntp : enable/disable network time synchronization.

#chronyd : used for time synchronization and clock management.

#chronyc sources : verify proper synchronization.

/etc/chrony.conf : chrony config file.

Remote file system and automation

*****configure nfs server**

#dnf install nfs-utils

#mkdir -p /nfsdata /home/ldap/ldapuser{1..9}

#echo "/nfsdata *(rw, (no_root_squash))" >> /etc/exports

#echo "/home/ldap *"(rw, (no_root_squash))" >>etc/exports

#systemctl enable --now nfs-server

#for i in nfs mountd rpc-bind; do firwall-cmd --add-service $i --permanent; done

#firewall-cmd reload

#show mount -e nfsserver :to check the nfs server is accessible.

*****client side**

#dnf install nfs-utils

#mount server:/nfsdata /mnt

*****automont (client side)**

#dnf install autofs

#echo "/nfsdata /etc/auto.nfsdata">>/etc/auto.master

#echo "files -rw nfsserver:/nfsdata">>/etc/auto.nfsdata

#systemctl enable --now autofs

*****automount for home directory**

#echo "/homes /etc/auto.homes">>/etc/auto.master

RHCSA 9

#echo "* -rw nfsserver:/home/ldap">>/etc/auto.homes

#systemctl restart autofs

Containers

#dnf install container-tools

#podman login registry.redhat.io	: login to a registry.
#podman login registr.redhat.io	: to get your current login credentials.
#podman search	: search for images from registries
#podman build	: build an image from containe image.
#podman run	: run a container
#podman stop	: stop a container
#podman rm	: remove a container
#podman images	: list you images
#podman inspect	: show container or image datails
#podman pull	: pull (download) image from registries
#podman exec	: run a command in a running container
#podman ps	: list info abut active containers.

#podman run -d --name sleepy docker.io/library/httpd sleep 3600

➔-d: run a container in detached mode, which means container will run in background, sleep 3600 : container will run a specific service and then will sleeps for 1hour.

#podman run -it --name mycontainer docker.io/library/httpd

➔-it : start a container with interactive terminal

#podman logs mycontainer :explore logs created by the container.

#podman run -d --name mydb quay.io/centos7/mariadb-103-centos7

➔this command will exit with error

#skopeo inspect docker://quay.io/centos7/mariadb-103-centos7

➔display metadata about container image

➔don't forget to add **docker://** or it won't work

#podman run -d --name mydb -e MYSQL_ROOt_PASSWORD=wasssim -p 8080:80 quay.io/centos7/mariadb-103-centos7

➔ -e : set container envirement variable, -p : publish a container's port, or range of ports. #firewall-cmd --add-port 8080/tcp : allow port access on firewall.

#firewall-cmd reload

*****persistent storage**

with root user you can run this command with out problem:

# podman run -d --name mydb -e MYSQL_ROOt_PASSWORD=wasssim -p 8080:80 -v /root/mydb:/var/lib/mysql quay.io/centos7/mariadb-103-centos7

but with a normal user you will face many problems related to files owner ship and selinux

solution:

RHCSA 9

#podman run -d --name mydb -e MYSQL_ROOt_PASSWORD=wasssim -p 8080:80 quay.io/centos7/mariadb-103-centos7

➔run containe with out storage

#podman exec mydb grep mysql /etc/passwd :to get mysql uid:uid

#podman unshare chown 27:27 mydb :chande the folder mydb to become owned by the containers.

#podman run -d --name mydb -e MYSQL_ROOt_PASSWORD=wasssim -p 8080:80 -v /root/mydb:/var/lib/mysql:Z quay.io/centos7/mariadb-103-centos7

➔now you can run the command with out any problems PS: don't forget to add :Z which will take care of selinux context.

*****auto starting container**

#useradd linda

#passwd linda

#loginctl enable-linger linda :to make sure the container service will run even the user linda isn't connected.

#ssh linda@localhost

#mkdir ~/.config/systemd/user

#cd ~/.config/systemd/user

#podman run -d --name myngnix -p 8080:80 ngnix

#podman generate systemd--name mynginx --files --new

#echo "WantedBy=default.target" >>container-myngnix.service

➔wantedby should set only to default.target, otherwise, it won't work.

#systemctl --user daemon-reload

#systemctl --user enable container-myngnix.service

#sudo reboot

#jounalctl | grep containee-myngnix : to check if the container is working

---

*****pdfconvert**

#loginctl enable-linger Wassim

#ssh wassim@localhost

$cd ~

$ mkdir -p ~/data/in ~/data/out

$mkdir -p .config/systemd/user

$cd .config/systemd/user

$git clone https://github.com/sachinyadav3496/Text-To-PDF.git

$podman build -t pdfconvert -f ./Text-To-PDF/Dockerfile

➔to build container image using docker file

$podman run -d --name myapp1 pdfconvert

$podman exec myapp1 cat /etc/passwd

$podman unshare chown 65534:65534 ~/data/in

$podman unshare chown 65534:65534 ~/data/out

$podman stop myapp1 && podman rm myapp1

$podman run -d --name myapp1 -v ~/data/in:/data/input:Z -v ~/data/out:/data/output:Z pdfcon

RHCSA 9

$podman generate systemd--name myapp1 --files --new

$vim container-myapp1.service

➔ check for the line **WantedBy=default.target**

$systemctl --user daemon-reload

$systemctl --user enable container-myapp1.service

********** syslog**

#useradd bob

#passwd bob ➔ set password to tekup

#loginctl enable-linger bob

#mkdir /var/log/containerlog

#chown bob:bob /var/log/containerlog

#ssh bob@localhost

$podman login registry.access.redhat.com

$mkdir container_logserver

$ mkdir -p .config/systemd/user

$ cd .config/systemd/user

$ git clone https://github.com/aheimsbakk/container-syslog-example.git

$ podman run -d --name container-logserver -v /var/log/containerlog/:/var/log/:Z syslog:latest

$podman ps ➔check for the container is working

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

$ systemctl --user daemon-reload

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

---

***********rsyslogpodman build**

1. Create a container logserver from an image rsyslog
2. Configure the container with systemd services by an existing user "linda",
3. Service name should be container-logserver, and configure it to start automatically across reboot.
4. Configure your host journal to store all journal across reboot
5. Copy all *.journal from /var/log/journal and all subdirectories to /home/linda/container_logserver
6. Configure automount /var/log/journal from logserver (container) to /home/linda/container_logserver when container starts.

#vim /etc/systemd/journal.conf

➔make the following changes

```
Storage=persistent
SystemKeepFree=100M
RuntimeKeepFree=100M
```

#systemctl restart systemd-journald

#useradd linda && passwd linda

#loginctl enable-linger linda

#ssh linda@localhost

#podman login registry.access.redhat.com

#mkdir -p .config/systemd/user

#cd .config/systemd/user

RHCSA 9

```
#mkdir container_logserver

# cp -r /var/log/journal/* ~/container_logserver/

# podman search rsyslog

# podman pull docker.io/lendingworks/rsyslog

# podman run -d --name container-logserver -v
/home/linda/container_logserver/:/var/log/journal/:Z
docker.io/lendingworks/rsyslog:latest

#podman ps

#podman generate systemd --name container-logserver --files --new

#systemctl --user daemon-reload

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