RHCSA_v8.2

==========

*Important_Instructions: Please read carefully.

hostname: servera.lab.example.com (172.25.250.10)
hostname: serverb.lab.example.com (172.25.250.11)

- You will be given by 2 VMs
- Total number of Questions will be around 22
- In one system root password is already set (no need to reset) but in second system password need to be recovered.
- In your both system root passwd is "trootent"
- In one system Network configuration is required but in another one networking is already done
- NTP need to be configured in only one system (not in both)
- YUM Repo need to configured in both systems.
- There is not any Q to configure LDAP Client (it is already configured).
- You just need to configure automounting for LDAP user's Home DIR in one system. (follow same steps as RHEL-7)
- Firewall and SELinux both will be pre-enabled.

==========

Do this in Server-a:

#Q1. Configure network and set the static hostname.

IP ADDRESS = 172.25.250.10
NETMASK = 255.255.255.0
GATEWAY = 172.25.250.254
DNS = 172.25.250.254
Domain name = lab.example.com
hostname = servera.lab.example.com

==========

#Q2. Configure YUM repos with the given link (2 repos: 1st is Base and 2nd is AppStream)

- Base_url= http://content.example.com/rhel8.0/x86_64/dvd/BaseOS
- AppSterm_url= http://content.example.com/rhel8.0/x86_64/dvd/AppStream

==========

#Q3. Debug SELinux:

- A web server running on non standard port 82 is having issues serving content. Debug and fix the issues.

- The web server on your system can server all the existing HTML files from /var/www/html (NOTE: Do not make any changes to these files)
- Web service should automatically start at boot time.

==========

#Q4. Create User accounts with supplementary group.

- Create the group a named "**sysadms**".
- Create users as named "**natasha**" and "**harry**", will be the supplementary group "**sysadms**".
- Create a user as named "**sarah**", should have non-interactive shell and it should be not the member of "**sysadms**".
- Password for all users should be "**trootent**"

==========

#Q5. Configure a cron job that runs every 1 minutes and executes:

l**ogger "EX200 in progress"** as the user **natasha**.

==========

#Q6. Create a collaborative Directory.

- Create the Directory "**/home/manager**" with the following characteristics.
- Group ownership of "/home/manager" should go to "**sysadms**" group.
- The directory should have full permission for all members of "**sysadms**" group but not to the other users except "**root**".
- Files created in future under "**/home/manage**r" should get the same group ownership .

==========

#Q7. Configure NTP

- Synchronize time of your system with the server **classroom.example.com**.

==========

#Q8. Configure AutoFS

- All **Ldapuser2** home directory is exported via **NFS**, which is available on **classroom.example.com (172.25.254.254)** and your NFS-exports directory is **/home/guests** for **Ldapuser2**,
- **Ldapuser2's** home directory is **classroom.example.com:/home/guests/ldapuse2**
- **Ldapuser2's** home directory should be automount autofs service.
- Home directories must be **writable** by their users.
- while you are able to log in as any of the user **ldapuser1** through **ldapuser20**, the only home directory that is accessible from your system is **ldapsuser2**

==========

#Q9. ACL.

- Copy the file /etc/fstab to /var/tmp/ and configure the "ACL" as mentioned following.
- The file /var/tmp/fstab should be owned by the "root".
- The file /var/tmp/fstab should belong to the group "root".
- The file /var/tmp/fstab should not be executable by any one.
- The user "sarah" should be able to read and write to the file.
- The user "harry" can neither read nor write to the file.
- Other users (future and current) should be able to read /var/tmp/fstab.

==========

#Q10. Create user **'bob'** with **2112** uid and set the password **'trootent'**

==========

#Q11. Locate all files owned by user "**harry**" and copy it under **/root/harry-files**

==========

==========

#Q12. Find a string '**ich'** from "**/usr/share/dict/words**" and put it into **/root/lines** file.

==========

#Q13. create an archive '**/root/backup.tar.bz2**' of **/usr/local** directory and compress it with bzip2.

==========

Server-2:

`````````

NOTE: In this Server 3 Disks will be given.

1. /dev/vda : for ROOT filesystem ( don't do anything under this Disk )
2. /dev/vdb : You need to use Swap and LVM Partition.
3. /dev/vdc : Will be used for Stratis.

==========

#Q14. Reset **root** user password and make it '**trootent**'

==========

#Q15. Configure YUM Repos

- Base_url= "http://content.example.com/rhel8.0/x86_64/dvd/BaseOS"
- AppStrem_url= "http://content.example.com/rhel8.0/x86_64/dvd/AppStream"

==========

#Q16. Resize a logical Volume

- Resize the logical volume "**mylv**" so that after reboot the size should be in between **200MB** to **300MB**.

==========

#Q17. Add a **swap** partition of **512MB** and mount it permanently.

==========
`````````

#Q18. Create a logical Volume and mount it permanently.

- Create the logical volume with the name "**wshare**" by using **50PE's** from the volume group "**wgroup**".
- Consider each **PE size** of the volume group as "**8 MB"**.
- Mount it on **/mnt/wshare** with file system **vfat**.

==========

#Q19. Create a new STRATIS volume according to following requirements:

- Use the unpartitioned disk
- The volume is named **'stratisfs'** belongs to **'stratispool'**
- The volume must be mounted permanent under '/**stratisvolume'**
- Place a copy of the file "http://classroom.example.com/content/Rhcsa-v8/rhel-8_Ex200_Q_1" under '/**stratisvolume'**
- Take a **snapshot** of **stratisfs** named **stratissnap**.

(OR)

Create a new VDO partition using to following requirements:

- Use the unpartitioned disk
- Vdo name "**Vdo1**" and logical size should be **50GB**
- Mount it on **/vdomount** permanently with file system **xfs**.

==========

#Q20. Configure System Tuning:

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

==========

#Q21.

- Create a container **logserver** from an image **rsyslog** in node1 From **registry.lab.example.com**
- Configure the container with systemd services by an existing user "**Walhalla**",
- Service name should be **container-logserver**, and configure it to start automatically across reboot.

Node1: root steps

```
# useradd user1
# passwd user1
# yum module install container* -y
# ll /var/log/
# vim /etc/systemd/journald.conf
[Journal]
Storage=persistent
```

:wq!
/run/log
systemctl restart systemd-journald
ll /run/log
ll /var/log/
su - user1
mkdir /home/user1/container-logserver/
exit
cp -r /var/log/journal/ /home/user1/container-logserver/
chown -R user1:user1 /home/user1/container-logserver/
systemctl restart systemd-journald
reboot

after bootup do ssh to user1 and continue:
ssh user1@servera.lab.example.com

#22

- Configure your host journal to store all journal across reboot
- Copy all *.journal from /var/log/journal and all subdirectories to **/home/Walhalla/container_logserver**
- Configure automount **/var/log/journal** from **logserver** (container) to **/home/walhalla/container_logserver** when container starts.

node1:user1 steps
---
podman login regisrty.redhat.io
username:
password:
podman search rsyslog
podman pull registry.redhat.io/rhel8/rsyslog
podman image list
podman run -d --name logserver -v /home/user1/container-logserver:/var/log/journal:Z registry.redhat.io/rhel8/rsyslog
podman container list
podman ps
mkdir -p ~/.config/systemd/user
cd .config/systemd/user/
loginctl enable-linger
loginctl show-user user1
podman generate systemd --name logserver -f -n
systemctl --user daemon-reload

'in this step, we went to host and one time we rebooted it'

```
# systemctl --user enable --now container-logserver.service
# systemctl --user start --now container-logserver.service
# systemctl --user status --now container-logserver.service
# podman exec -it logserver /bin/bash
# ls /var/log/
# exit
```

---------- E N D ----------