Practice Test 1 RHCSA (EX200)

Question 1

1. You've forgotten the root password for ServerA. Reset the root password to "word" to regain access to the system.
    - Reboot and access grub (press any key while the kernel selector appears)
    - Select the Red Hat kernel and hit 'e' to edit
    - After the 'ro' type `rd.break`
    - Then hit Ctrl + X to boot
    - You’ll boot to a 'rescue' prompt that looks like this: `switch_root:/#`.
    - Remount the root partition in read-write mode so that you can run commands. Enter the following: `mount -o remount rw /sysroot` and then hit ENTER.
    - Now type `chroot /sysroot` and hit enter. This will change you into the `sysroot (/)` directory, and make that your path for executing commands.
    - Now you can simply change the password for root using the `passwd` command.
    - Next, before you reboot, you will need to make sure that SELinux allows the file changes. At the prompt ,enter: `touch /.autorelabel`. This will signal SELinux on the next reboot that the filesystem has changed (the changed password) and allow the change to be loaded. This will cause the whole filesystem to be 'relabeled' which might take a while, depending on the size of the filesystem and the speed of the machine, so be aware of this possibility.
    - Type `exit` to leave the chroot environment and enter `reboot`.

!!! Question 2

2. Configure a Local Yum/DNF Repository on ServerA using the RHEL-9 ISO image mounted on the /mnt directory.
    - More on: Create a local repo in RHEL 9
    - On Oracle VirtualBox while running the VM click at the top menu 'Devices', then click 'Optical Drives' then choose and click on the rhel ISO image.
    - Run `df -h` to verify it was added to your device
    - `mkdir /localrepo`
    - `cd /run/media/macc/RHEL-9-0-0-BAseOS-x86_64`
    - `mount /dev/sr0 /localrepo/`
    - `cd /etc/yum.repos.d/`
    - `vi /etc/yum.repos.d/rhel9.repo`

```
# vi /etc/yum.repos.d/rhel9.repo
[BaseOS]
name=BaseOS Packages Red Hat Enterprise Linux 9
metadata_expire=-1
gpgcheck=1
enabled=1
baseurl=file:///mnt/disc/BaseOS/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[AppStream]
name=AppStream Packages Red Hat Enterprise Linux 9
metadata_expire=-1
gpgcheck=1
```

```
enabled=1
baseurl=file:///mnt/disc/AppStream/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

- - Update the package index: `dnf clean all`
  - `dnf repolist`
  - `dnf install nfs-utils`

Question 3

3. Configure a NetworkManager connection profile named "myprofile1" on ServerA for the enp0s3 device with the following static settings:
   - Static IPv4 Address: 192.168.1.2/24
   - Static IPv6 Address: fd01::101/64
   - IPv4 Default Gateway: 192.168.1.1
   - IPv6 Default Gateway: fd01::100
   - IPv4 DNS Servers: 8.8.8.8, 8.8.4.4
   - IPv6 DNS Server: fd01::111
   - DNS Search Domain: example.com
   - `nmcli connection modify enp0s3 ipv4.addresses 192.168.1.2/24`
   - `nmcli connection modify enp0s3 ipv6.addresses fd01::101/64`
   - `nmcli connection modify enp0s3 ipv4.method manual`
   - `nmcli connection modify enp0s3 ipv6.method manual`
   - `nmcli connection modify enp0s3 ipv4.gateway 192.168.1.1`
   - `nmcli connection modify enp0s3 ipv6.gateway fd01::100`
   - `nmcli connection modify enp0s3 ipv4.dns 8.8.8.8`
   - `nmcli connection modify enp0s3 +ipv4.dns 8.8.4.4`
   - `nmcli connection modify enp0s3 ipv6.dns fd01::111`
   - `vi /etc/resolv.conf`
     - Add the line: `search example.com`
   - `nmclic onnection down enp0s3 && nmcli connection up enp0s3`

Question 4

4. On ServerA, add the following secondary IPV4 address statically to the connection profile named "myprofile1". Do this in a way that doesn’t compromise your existing settings:
   IPV4 – 10.0.0.5/24
   - Run `nmtui` to create 'myprofile1'
     - Select edit a connection
     - Select Add and Enter
     - Select Ethernet
     - Now enter the 'myprofile1' as the name of the new profile
     - And enter the same ethernet device as for your normal connection
     - Escape out of it and click OK in the main menu of `nmtui`
   - `nmcli connection show --active`\
   - `nmcli connection show myprofile1`
   - `nmcli connection modify myprofile1 +ipv4.addresses 10.0.0.5/24`

   - `nmcli connection down myprofile1 && nmcli connection down myprofile1`
   - `systemctl restart NetwrokManager`
   - `nmcli connection show myprofile1`

Question 5

5. On ServerA, include the following secondary IPv6 address statically to the connection profile named "myprofile1". Do this in a way that doesn’t compromise your existing settings:
   IPv6 – fd01::121/64

- `nmcli connection modify myprofile1 +ipv6.addresses fd01::121/64`
- `nmcli connection down myprofile1 && nmcli connect ion up myprofile1`
- `nmlci connection show myprofile1`

Question 6

6. On ServerA, configure the system time to the "America/New_York" timezone.
   - `timedatectl status`
   - `timedatectl set-timezone "America/New_York"`

Question 7

7. On ServerA, ensure NTP synchronization is configured for accurate timekeeping.
   - `vi /etc/chrony.conf`
     - Add the line: `server 8.8.8.8` (replace IP for any NTP server IP)
   - `timedatectl set-ntp true`
   - `systemctl restart chronyd`
   - `chronyc`
   - `sources`

Question 8

8. On ServerA, use /dev/sdb to do the following:
   1. Create a 2GiB volume group named “myvg”.
   2. Create a 500MiB logical volume named “mylv” inside the “myvg” volume group.
   3. The “mylv” logical volume should be formatted with the ext4 filesystem and mounted persistently on the /mylv directory.
   4. Extend the ext4 filesystem on “mylv” by 500M.
   - `fdisk -l`
   - `fdisk /dev/sdb`
     - `m` `p` `n` `p` `1` `Enter` `+2GiB` `p` `l` `t` `8e` `p` `w`
   - `pvcreate /dev/sdb1`
   - `pvdislpay`
   - `vgcreate myvg /dev/sdb1`
   - `vgdisplay`
   - `lvcreate -n mylv --size 500MiB myvg`

- `lvdisplay`
- `mkfs.ext4 /dev/myvg/mylv`
- `mkdir /mylv`
- `mount /dev/myvg/mylv /mylv`
- `vi /etc/fstab`
    - Add the line: `/dev/mapper/myvg-mylv /mylv ext4 defaults 0 0`
- `lvextend -L +500MiB /dev/myvg/mylv`
- `lvs`

Question 9

9. Set up a basic web server on ServerA to display the message "Welcome to the webserver!" upon connection, while ensuring that the firewall allows http/https services
    - `firewall-cmd --list-all`
    - `dnf install httpd`
    - `systemctl start httpd`
    - `vi /var/www/html/index.html`
        - Add the line: `"Welcome to the webserver!"`
    - `firewall-cmd --add-service=http`
    - `firewall-cmd --add-service=https`
    - curl http://localhost

Question 10

10. Locate and copy all files larger than 3MB within the "/etc" directory on ServerA to a new directory "/find/3mfiles".
    - `mkdir /find/3mfiles`
    - `find /etc/ -size +3M -exec cp {} /find/3mfiles \;`

Question 11

11. On ServerA, ensure that boot messages are displayed, not silenced, for troubleshooting purposes.
    - `vi /etc/defaults/grub`
        - Remove: `rhgb quiet` from the `GRUB_CMDLINE_LINUX` line
    - `grub2-mkconfig -o /boot/grub2/grub.cfg`

Question 12

12. On ServerA, create a Bash script named /script.sh that outputs the second argument followed by the first argument when executed with two arguments (e.g., "test2 test1" for ./script.sh test1 test2).
    - `vi /script.sh`
        - Add the line: `echo $2 $1`
    - `chmod a+x /script.sh`
    - `/script.sh test1 test2`

Question 13

13. Ensure that a file named "Congrats" is automatically added to the home folders of all new users created on ServerA.
    - `touch /etc/skel/Congrats`

Question 14

14. Enforce password expiration after 90 days and a minimum length of 8 characters for all user passwords on ServerA.
    - `vi /etc/login.defs`
        - Change line to: `PASS_MAX_DAYS 90`
    - `vi /etc/security/pwquality.conf`
        - Change line to: `minlen = 8`

Question 15

15. Create the following users and groups on ServerA, configure their permissions for specific directories, and ensure appropriate file ownership for newly created files.
    Users:
    - amr and biko (members of the "admins" group)
    - carlos and david (members of the "developers" group)
      Directories:
    - /admins (accessible only to owner and admins group members, owned by biko)
    - /developers (accessible only to developers group members, owned by carlos)
      File Ownership:
    - New files in /developers or /admins should be owned by the respective group owner.
    - Only file creators should be allowed to delete their files.
    - `groupadd admins && groupadd developers`
    - `useradd -G admins amr && useradd -G admins biko`
    - `useradd -G developers carlos && useradd -G developers david`
    - `mkdir /admins`
    - `mkdir /developers`
    - `chgrp admins /admins`
    - `chown biko /admins`
    - `chgrp developers /developers`
    - `chown carlos /developers`
    - `chmod o-x /admins`
    - `chmod o-x /developers`
    - `chmod g+s /admins`
    - `chmod g+s /developers`
    - `chmod +t /admins`
    - `chmod +t /developers`

Question 16

16. On ServerA, configure a cron job that writes the message "Get Ready!" to the system log file /var/log/messages at noon (12 PM) on weekdays only. Ensure the job is executed with appropriate permissions and logging for troubleshooting.
    - `crontab -e`

    - Add the line: `00 12 * * 1-5 logger "Get Ready!`

Question 17

17. On ServerA, create a compressed tar archive file named "/root/local.tgz" that contains the directory "/usr/local/" and its contents, ensuring appropriate permissions and verification of the archive.
    - `tar cvzf /root/local.tgz /usr/local`

Question 18

18. On ServerA, create a 200MB swap partition using /dev/sdb that automatically activates at boot.
    - `free -m`
    - `fdisk /dev/sdb`
        - `p` `n` `p` `2` `Enter` `+200M` `l` `t` `2` `82` `p` `w`
    - `mkswap /dev/sdb2`
    - `blkid`
    - `vi /etc/fstab`
        - Add the line: `UUID=... swap swap defaults 0 0`
    - `mount -a`
    - `swapon /dev/sdb2`
    - `free -m`
    - `lsblk`

Question 19

19. Set up an SSH Passwordless root remote login from ServerA to ServerB
    - `vi /etc/ssh/sshd_config` On server B
        - Uncomment and change line to: `PermitRootLogin yes`
    - `ssh-keygen` On serverA
    - `ssh-copy-id root@192.168.1.12` On serverA
    - `ssh root@192.168.1.12`

Question 20

20. Set the maximum number of SSH login attempts to 2 on ServerA.
    - `vi /etc/ssh/sshd_config`
        - Uncomment and change line to: `MaxAuthTries 2`
    - `systemctl restart sshd`

Question 21

21. Configure ServerA to perform the following tasks:
    1. Install container-tools.
    2. Use podman to search for the official Redis container.

3. Inspect the Redis image using skopeo.
4. Pull the Redis image using podman.
5. Add the tag "myredis" to the "docker.io/library/redis" image.
6. Set SELinux to "permissive" mode.
7. Set the SELinux Boolean value of "container_manage_cgroup" to "on" and ensure persistence across reboots.
8. Run the Redis container in detached mode with the name "redis" on port 6379:6379 using the "myredis" image.
9. Use systemd to control the startup of the Redis podman container.
10. Remove the Redis container.

- `dnf install container-tools`
- `podman search redis --filter=is_official`
- `podman pull docker.io/library/redis` (image name)
- `podman images`
- `podman tag docker.io/library/redis myredis`
- `getenforce`
- `vi /etc/selinux/config`
  - Change line to: `SELINUX=permissive`
- `setsebool -P container_manage_cgroup on`
- `podman run -d --name redis -p 6379:6379 localhost/myredis`
- `podman ps`
- `podman generate systemd --name redis --new --files`
- `cat /root/container-redis.service`
- `cp /root/container-redis.service /etc/systemd/system/`
- `systemctl enable container_redis.service`
- `systemctl disable container_redis.service`
- `systemctl stop container.redis.service`
- `podman ps`
- `podman rm redis`

Question 22

22. On ServerA, create a concise and efficient shell script named "/names.sh" that extracts and prints a clear list of usernames and their corresponding primary groups from the "/etc/passwd" file. Ensure the script is well-formatted, adheres to best practices, and includes informative comments for clarity.

- `vi /names.sh`
  - Add the line: `#!/bin/bash`
  - Add the line: `cut -d : -f 1,4 /etc/passwd`
- `chmod a+x /names.sh`
- `/names.sh`

Question 23

23. On ServerA, write a script named “/users_shells.sh” that generates a list of usernames from /etc/passwd along with their login shell.

- `vi /users_shells.sh`
  - Add the line: `#!/bin/bash`
  - Add the line: `cut -d : -f 1,7 /etc/passwd`
- `chmod a+x /users_shells.sh`

    - `/users_shells.sh`

Question 24

24. What is the default nice level assigned to a process when it's started using the nice command without specifying any additional niceness parameters?
    - 10

Question 25

25. On ServerA, craft a robust and informative shell script named "/find.sh" that efficiently counts the number of regular files matching a specified pattern (provided as the first argument) within the "/home" directory and its subdirectories. Incorporate error handling, informative output, and best practices to enhance its usability and reliability.
    - `vi /find.sh`

File editor:

```
#!/bin/bash
if test $# == 1
then
        find /home/* -type f -name $1 | wc -l
else
        echo "No key pattern specified"
fi
```

- - `chmod a+x /find.sh`
  - `/find.sh uwu`

Question 26

26. On ServerA, create a versatile and informative shell script named "/trim.sh" that effectively removes any occurrences of the vowels "a", "i", "e", "o", and "u" from all provided arguments, regardless of their order. Ensure the script is well-formatted, incorporates error handling, provides clear output, and adheres to best practices.
    - `vi /trim.sh`

```
#!/bin/bash
echo $@ | tr -d aeiou
```

- - `chmod a+x /trim.sh`
  - `/trim.sh hola como estas`

Question 27

27. Which specific character, when strategically appended to a command, initiates its execution in the background, allowing you to continue interacting with the current shell while the command runs concurrently?
    - `&`

- `echo uwu &` (testing)

Question 28

28. Predict the precise output generated by the following command, carefully considering the seq command's syntax and behavior:
$ seq 1 5 20
    - `1 6 11 16`
    - Usage: `seq [FIRST] [INCREMETN] [LAST]`

Question 29

29. Using appropriate commands, create a backup of the Master Boot Record (MBR) located on the device "/dev/sda" of ServerA. Store the backup in the file "/backup/mbr.img". Ensure the backup process adheres to the following requirements:
    - Block Size: 512 bytes
    - Number of Blocks Copied: 1
    - Verification: Confirm successful backup creation
    - `mkdir /backup`
    - `dd if=/dev/sda of=/backup/mbr.img bs=512 count=1`
    - `ll /backup`

Question 30

30. Identify the redirection operator that enables reading input from the current source until a specified separator string, located on a separate line without trailing spaces, is encountered.
    - `<<`

!!! Question 31

31. Configure ServerA to automatically mount the home directories of users tom and sam from ServerB using NFS. The home directories on ServerB are located at "/home/tom" and "/home/sam," with user IDs 1010 and 1020, respectively. The mount should be established in the local "/home" directory on ServerA, ensuring read and write permissions, efficient resource usage, and seamless user experience.

- (on both server and client)
    - `dnf update -y`
    - `dnf install nfs-utils -y`
    - `dnf install nfs -y`
    - `dnf install autofs -y`
    - `systemctl enable --now autofs.service`
- (on server ServerB)
    - `useradd -u 1010 tom`
    - `useradd -u 1020 sam`
    - `chmod a+x /home/to
    - `chmod a+x /home/sam`
    - `systemctl enable nfs-server`

    - `systemctl stop firewalld`
    - `vi /etc/exports`
        - Add the line: `/home/tom *(rw,sync,no_root_squash)`
        - Add the line: `/home/sam *(rw,sync,no_root_squash)`
    - `exportfs -rv`
- (on client ServerA)
    - `showmount -e 192.168.12`
    - `mkdir /home/sam`
    - `mkdir /home/tom`
    - `mount 192.168.1.12:/home/tom /home/tom`
    - `mount 192.168.1.12:/home/sam /home/sam`

OR

- On ServerB
    - `dnf update -y`
    - `dnf install nfs-utils -y`
    - `useradd tom`
    - `useradd sam`
    - `chmod a+x /home/tom`
    - `chmod a+x /home/sam`
    - `systemctl enable nfs-server`
    - `vi /etc/exports`
        - Add the line: `/home/tom *(rw,sync,no_root_squash)`
        - Add the line: `/home/sam *(rw,sync,no_root_squash)`
    - `exportfs -rv`
    - `firewall-cmd --add-service=nfs --permanent`
    - `firewall-cmd --reload`
    - `firewall-cmd --list-all`
- On ServerA
    - `dnf update -y`
    - `dnf install nfs-utils -y`
    - `dnf install autofs -y`
    - `systemctl enable autofs --now
    - `showmount -e ServerB` (probably wont work)
    - `useradd -M -u 1010 tom` // The “-M” option is used to not create the user’s home directory.
    - `useradd -M -u 1020 sam`
    - `vi /etc/auto.master`
        - Add the line: `/home /etc/auto.home_sam` // `[mountpoint] [share]`
        - Add the line: `/home /etc/auto.home_tom`
    - `vi /etc/auto.home_sam`
        - Add the line: `* -fstype=nfs,rw,sync ServerB:/home/sam`
    - `vi /etc/auto.home_tom`
        - Add the line: `* -fstype=nfs,rw,sync ServerB:/home/tom`
    - `systemctl restart autofs`
    - `su - tom`
    - `ll /home`
    - `pwd`

!!! Question 32

32. On ServerA, as user sam, create a persistent, rootless Apache HTTP web server container using the "registry.redhat.io" registry, ensuring it adheres to best practices for security and efficiency. Apply the following specifications:
    - Container tag: "httpd-24"
    - Container name: "httpd"
    - Registry credentials: username "admin", password "administrator"
    - Mount a persistent storage volume from "~/www-data/" to "/var/www/html/" within the container
    - Create an "index.html" file containing "Hello World!" in the "~/www-data/" directory
    - Map host port 8080 to container port 8080
    - Set environment variables: HTTPD_USER=test, HTTPD_PASSWORD=test
    - Manage the container using systemd for persistence and automatic startup
    - Remember: The tilde (~) is a Linux "shortcut" to denote a user's home directory. Thus tilde slash (~/) is **the beginning of a path to a file or directory below the user's home directory**.

- Run as root:
    - `adduser sam`
    - `passwd uwu`
    - `loginctl enable-linger sam`
    - `loginctl show-user sam` // Verify that linger is enabled for the user sam
    - `ssh sam@localhost`
- Run as user sam:
    - `podman login registry.redhat.io`
        - Enter redhat username and password to log in
    - `podman search httpd-24`
    - `podman pull registry.access.redhat.com/ubi9/httpd-24`
    - `podman images`
    - `podman tag registry.access.redhat.com/ubi9/httpd-24 httpd-24`
    - `mkdir ~/www-data`
    - `echo "Hello World" > ~/www-data/index.html`
    - `podman run -d --name httpd -p 127.0.0.1:8080:8080 -e HTTPD_USER=test -e HTTPD_PASSWORD=test -v ~/www-data:/var/www/html/:Z localhost/httpd-24`
    - `podman ps`
    - `curl http://localhost:8080`
    - `mkdir -p ~/.config/systemd/user`
    - `cd ~/.config/systemd/user`
    - `podman generate systemd -f -n httpd --new`
    - `ls` // Verifying that the "container-httpd.service" systemd unit file is generated
    - `podman stop httpd`
    - `podman ps`
    - `podman rm httpd`
    - `podman ps -a`
    - `systemctl --user daemon-reload`
    - `systemctl --user enable --now container-httpd.service`
    - `podman ps` //Verify the container has been created
    - `systemctl --user stop container-httpd.service`

Question 33

33. On ServerA, a web server running on port 88 is unable to serve content correctly. Troubleshoot and rectify the issue to ensure:
    - **Web Server Functionality:** All HTML files within /var/www/html are served successfully.
    - **Non-Standard Port Usage:** The web server operates on port 88.
    - **Automatic Startup:** The web server initiates automatically at system startup.
    - `systemctl status httpd`
    - `ls -Z /var/www/html/` // To view the security context of the HTML file
    - `semanage fcontext -m -t httpd_sys_content_t /var/www/html/page1`
    - `restorecon -R -v /var/www/html/page1`
    - `ls -Z /var/www/html/*`
    - `semanage port -a -t http_port_t -p tcp 88` // add port 88
    - `semanage port -l | grep http` // To check whether port 88 is allowed
    - `vi /etc/httpd/conf/httpd.conf`
        - Add the line: `Listen 88` // This will allow port 88 to be used by http
    - `systemctl restart httpd`
    - `systemctl enable httpd`
    - `curl http://192.168.1.11:88/page{1..3}`

Question 34

34. On ServerA, configure the atd service to specifically grant access to the user Adam while denying access to the user Tom. Ensure the configuration adheres to Red Hat best practices for security and clarity.
    - `useradd Adam`
    - `useradd Tom`
    - `echo "Tom" >> /etc/at.deny`
    - `echo "Adam" >> /etc/at.allow`
    - `systemctl restart atd`

Question 35

35. On ServerA, locate all lines within the "/etc/passwd" file that include the string "test". Create a file named "/root/test" containing exact copies of these lines in their original order, excluding any empty lines.
    - `grep test /etc/passwd | grep -v ^$ > /root/test`
        - Note the second `grep` command is to eliminate non matching lines, in this case empty lines

Question 36

36. On ServerA, create a script named "/home/XSam.sh" that grants Sam passwordless sudo access, adhering to security best practices and providing clear validation of the configuration.
    - `vi /home/XSam.sh`

File editor:

```
#!/bin/bash
echo "Sam ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/Sam   # Create file and add line
sudo -u Sam sudo cat /etc/sudoers > /dev/null            # To test that sudo works
if [ $? -eq 0 ]        # If no errors found while runnning this script
```

```
then
        echo Sam has sudo priviliges without a password
else
        echo error
fi
```

- Note the the `$?` is the exit status number, if the exit status number is 0 the script was successful
  - `useradd Sam`
  - `passwd Sam`
  - `chmod a+x /home/XSam.sh`
  - `/home/XSam.sh`

Question 37

37. On ServerA, scan and analyze the audit.log file for SELinux denials and attempts, and save the results to the “/audit_log.txt” file. Ensure the analysis provides clear explanations and actionable recommendations for resolving identified issues.
    - `sealert -a /var/log/audit/audit.log > /audit_log.txt` // sealert is an SELinux troubleshoot client tool (the `a` option is to scan a file)

Question 38

38. On ServerA, create a compressed archive of the "/usr/local/bin/" directory using tar and bzip2. Store the archive under "/home" with the filename "local-bin.tar.bz2". Verify the contents of the archive.
    - `tar cvf /home/local-bin.tar /usr/local/bin/`
    - `bzip2 /home/local-bin.tar`
    - `ls`

Question 39

39. On ServerA, append the message "Ended on $(date) by $LOGNAME" to the "/var/log/messages" file with root privileges. Use regular expressions to precisely verify the message's inclusion in the log file.
    - `ehco "Ended on $(date) by $LOGNAME" >> /var/log/messages`
    - `grep Ended /var/log/messages`

Question 40

40. Configure ServerA to automatically boot into the multi-user.target, ensuring a non-graphical, multi-user environment for command-line administration.
    - `systemctl get-default`
    - `systemctl set-default multi-user.target`

Question 41

41. On ServerA, create a new user named "Samir", and grant him the ability to execute commands with root privileges using sudo. Ensure clarity, conciseness, accuracy, and consider alternative approaches.

- `useradd -aG wheel Samir`
- `passwd Samir`
- `sudo fdisk -l` // While being Samir

Question 42

42. As user Samir, transfer the sensitive file /etc/hosts from ServerA to the /home/ directory on ServerB, ensuring confidentiality and integrity during transit. Choose a secure transfer method appropriate for the Red Hat Enterprise Linux (RHEL) 9 environment.
    - `scp /etc/hosts root@192.168.1.12:/home/`

Question 43

43. Manage the NetworkManager service on ServerA effectively using systemctl commands.
    - `systemctl status NetworkManager`
    - `systemctl enable NetworkManager`
    - `systemctl start NetworkManager`
    - `systemctl start NetworkManager`
    - `systemctl is-active NetworkManager`

Question 44

44. You are the administrator for two Red Hat Enterprise Linux (RHEL) 9 servers, ServerA and ServerB. ServerB runs the Apache HTTP Server and needs to access files in the directory */var/www/html/mydirectory*. However, SELinux is currently preventing this access.

    Task: Modify the SELinux policy on ServerB to grant Apache HTTP Server access to files in */var/www/html/mydirectory* securely, following best practices.

    Additional Considerations:
    - This modification should persist after a server reboot.
    - Minimize the impact on other applications or directories.

- (On ServerB)
    - `dnf install httpd -y`
    - `systemctl enable --now httpd`
    - `firewall-cmd --add-service=http --permanent`
    - `mkdir /var/www/html/mydirectory`
    - `setsebool -P httpd_read_user_content on` //The -P option makes the change persistent across reboots
    - `getsebool httpd_read_user_content`