



**RHCSA.GURU**

# **Solutions : Mock Test 4**

**Task 1** - Configure the system to enable IPv6 packet forwarding. Ensure that the configuration persists across system reboots.

**Solution -** 1- Use the `sysctl` command to enable IPv6 forwarding for the current session

```
• bash-5.1# sysctl -w net.ipv6.conf.all.forwarding=1  
net.ipv6.conf.all.forwarding = 1
```

2-Add the configuration to [/etc/sysctl.conf](#) so it remains active after a reboot and reload the sysctl configuration to activate the change without rebooting.

```
● bash-5.1# echo "net.ipv6.conf.all.forwarding = 1" >>  
  /etc/sysctl.conf
```

```
● bash-5.1# sysctl -p  
  net.ipv6.conf.all.forwarding = 1
```

3-Verify that IPv6 forwarding is successfully enabled.

```
● bash-5.1# sysctl net.ipv6.conf.all.forwarding  
  net.ipv6.conf.all.forwarding = 1  
○ bash-5.1#
```

**Task 2** - Create a user named max with a User ID (UID) of 6000, ensuring the user has no access to an interactive shell.

**Solution - 1- Create the user max with a specific UID and no interactive shell:** Use `useradd` with `-u` to specify the UID, and `-s /sbin/nologin` to prevent the user from accessing an interactive shell.

```
● bash-5.1# useradd -u 6000 -s /sbin/nologin max  
○ bash-5.1#
```

2-Use `id` command to display the user's UID and groups, while checking `/etc/passwd` confirms the shell configuration.

```
● bash-5.1# id max
uid=6000(max) gid=6000(max) groups=6000(max)
● bash-5.1# grep max /etc/passwd
max:x:6000:6000::/home/max:/sbin/nologin
○ bash-5.1#
```



### Task 3 - Modify the SELinux type of the /etc/ssh directory to var\_log\_t

**Solution -** 1- Use `ls -ldZ` command displays the SELinux context of the specified directory

- **-l:** Displays detailed information about files or directories, such as permissions, owner, and size.
- **-d:** Shows information about the directory itself, not its contents.
- **-Z:** Displays the SELinux security context of files or directories.

```
● bash-5.1# ls -ldZ /etc/ssh
drwxr-xr-x. 4 root root system_u:object_r:etc_t:s0 4096 Jan
 23 11:25 /etc/ssh
○ bash-5.1#
```



2- Use **semanage fcontext** command with the **-a** option to add a new context mapping, and the **-t** option specifies the type. The **(/.\*)?** ensures all files and subdirectories are included.

```
bash-5.1# semanage fcontext -a -t var_log_t "/etc/ssh(/.*)"?  
bash-5.1#
```

3- Use **restorecon** command to apply the new SELinux context to the directory and its contents, making the changes effective.

```
bash-5.1# restorecon -Rv /etc/ssh  
Relabeled /etc/ssh from system_u:object_r:etc_t:s0 to system_u:object_r:var_log_t:s0  
Relabeled /etc/ssh/ssh_config.d from system_u:object_r:etc_t:s0 to system_u:object_r:var_log_t:s0  
Relabeled /etc/ssh/ssh_config.d/50-redhat.conf from system_u:object_r:etc_t:s0 to system_u:object_r:var_log_t:s0  
Relabeled /etc/ssh/sshd_config.d from system_u:object_r:etc_t:s0 to system_u:object_r:var_log_t:s0
```





4- Verify the updated SELinux context of the /etc/ssh directory using **ls** command

```
• bash-5.1# ls -ldZ /etc/ssh
drwxr-xr-x. 4 root root system_u:object_r:var_log_t:s0 4096 J
an 23 11:25 /etc/ssh
○ bash-5.1#
```





**Task 4** - Use Podman to pull the registry.access.redhat.com/ubi8 image and run a container named 'ubi8-container' from it. Ensure the container runs with a user named harry. If the harry user does not already exist on the system, create it.

**Solution** - 1- Use `id` command checks if the user exists, and if not, `useradd` creates the user. The `||` operator runs the second command only if the first command fails.

```
bash-5.1# id harry || useradd harry
uid=6001(harry) gid=6001(harry) groups=6001(harry)
bash-5.1#
```

2-Install **podman** using dnf package manager.

```
○ bash-5.1# dnf install podman -y
Updating Subscription Management repositories.
Unable to read consumer identity
```

3-Use **podman pull** command downloads the specified image to the local system for use in containers.

```
● bash-5.1# podman pull registry.access.redhat.com/ubi8
Trying to pull registry.access.redhat.com/ubi8:latest.
Getting image source signatures
Checking if image destination supports signatures
Copying blob c9235833a899 done |
Copying config a54f16c811 done |
```



4- The **podman run** command creates a container with **--user** to specify the user ID (UID) and group ID (GID) of **harry**. The container runs in the background with **-d** and executes **sleep infinity** to keep it running.

**Command :** `podman run --name ubi8-container --user $(id -u harry):$(id -g harry) -d registry.access.redhat.com/ubi8 sleep infinity`

```
bash-5.1# podman run --name ubi8-container --user $(id -u harry):$(id -g harry) -d registry.access.redhat.com/ubi8 sleep infinity
7ced0492bd9dc5316981aeebb174932449f49ab8d4e16a9c6fc778ce7bcd8c33
bash-5.1#
```

5- Use **podman exec** command runs the **id** command inside the container to confirm it is running as the **harry** user.

```
• bash-5.1# podman exec -it ubi8-container id
uid=1002(harry) gid=1002(1002) groups=1002(1002)
○ bash-5.1#
```



**Task 5** - Write a script named `users.sh` in workspace directory to add multiple users based on a comma-separated list of usernames provided as input. Set the password for each newly created user to `student@rhel`.

**Solution -** 1- Use a text editor like `vi` to create the `users.sh` script in the specified directory.

```
● bash-5.1# vi /home/coder/workspace/users.sh
○ bash-5.1#
```



## 2-Add the script logic.

```
$ users.sh > ...
1  #!/bin/bash
2
3  # Prompt for comma-separated usernames
4  read -p "Enter usernames (comma-separated): " usernames
5
6  # Split usernames into an array
7  IFS=',' read -r -a user_array <<< "$usernames"
8
9  # Loop through each username
10 for user in "${user_array[@]}"; do
11     # Trim whitespace
12     user=$(echo "$user" | xargs)
13
14     # Check if the user exists, if not, add the user
15     if id "$user" &>/dev/null; then
16         echo "User $user already exists."
17     else
18         useradd "$user"
19         echo "student@rhel" | passwd --stdin "$user" &>/dev/null
20         echo "User $user created and password set to student@rhel."
21     fi
22 done
23
```



3-Use the **chmod** command to grant execute permissions to the script.

```
● bash-5.1# chmod +x /home/coder/workspace/users.sh  
○ bash-5.1#
```

4-Execute the script and provide usernames as input.

```
● bash-5.1# /home/coder/workspace/users.sh  
Enter usernames (comma-separated): alex,bob,charlie  
User alex created and password set to student@rhel.  
User bob created and password set to student@rhel.  
User charlie created and password set to student@rhel.  
1.  
○ bash-5.1#
```

**Task 6** - Create a partition table of type Linux LVM on the disk /dev/nvme1n1.

**Solution** - 1-Creates a new partition table of type GPT on the disk /dev/nvme1n1 using parted cli utility to manage partition

```
● bash-5.1# parted /dev/nvme1n1 mklabel gpt  
Information: You may need to update /etc/fstab.
```

2-Creates a primary partition on /dev/nvme1n1 from 1MiB to the end of the disk.

```
● bash-5.1# parted /dev/nvme1n1 mkpart primary ext4 1M  
iB 100%  
Information: You may need to update /etc/fstab.
```





3- Sets the partition type to Linux LVM for the first partition on `/dev/nvme1n1`.

```
● bash-5.1# parted /dev/nvme1n1 set 1 lvm on  
Information: You may need to update /etc/fstab.
```

4- Updates the kernel to detect partition changes. Verify it by listing block device details like name, size, type, and file system.

```
● bash-5.1# partprobe /dev/nvme1n1  
● bash-5.1# lsblk -o NAME,SIZE,TYPE,FSTYPE /dev/nvme1n1  
1  
NAME          SIZE TYPE FSTYPE  
nvme1n1       5G disk  
└─nvme1n1p1    5G part  
○ bash-5.1#
```

**Task 7** - Locate all files owned by the coder user and copy them to the /root/Backup directory.

**Solution** - 1-Create the /root/Backup directory if it doesn't already exist

```
● bash-5.1# mkdir /root/Backup  
○ bash-5.1#
```

2-Use **find** command to locate files owned by the **coder** user (**-user coder**) of type file (**-type f**). The **--parents** option in **cp** preserves the directory structure relative to the root when copying files. Errors like inaccessible directories are redirected to **/dev/null**.

**Command** : `find / -user coder -type f -exec cp --parents {} /root/Backup \;`  
`2>/dev/null`

```
bash-5.1# find / -user coder -type f -exec cp --par  
ents {} /root/Backup \; 2>/dev/null
```

3- Verify the files copied to /root/Backup

```
● bash-5.1# ls /root/Backup/  
home var  
○ bash-5.1#
```

## Task 8 - Create a 512MB swap partition on the disk /dev/nvme2n1

**Solution -** 1- Use the **fdisk** utility to create a partition of size 512MB and set its type to **Linux swap** (code **82**). This prepares the disk for use as swap space.

```
bash-5.1# fdisk /dev/nvme2n1

Welcome to fdisk (util-linux 2.37.4).
```

2- Inside **fdisk**, perform the following steps:

- Press **m** for help.
- Press **n** to create a new partition.
- Choose **p** for primary.
- Select partition number **1**.

```
Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partition)
Select (default p): p
Partition number (1-4, default 1): 1
```



- Press **Enter** to accept the default first sector.
- Type **+512M** for the last sector.
- Press **p** to print the partition table and verify.
- Press **t** to change the partition type.
- Type **82** for Linux swap.
- Press **w** to write changes and exit.

```
First sector (2048-10485759, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-10485759, default 10485759): +512M
```

```
Created a new partition 1 of type 'Linux' and of size 512 MiB.
```

```
Command (m for help): p
Disk /dev/nvme2n1: 5 GiB, 5368709120 bytes, 10485760 sectors
Disk model: Amazon Elastic Block Store
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0x453c014a
```

| Device         | Boot | Start | End     | Sectors | Size | Id | Type |
|----------------|------|-------|---------|---------|------|----|------|
| /dev/nvme2n1p1 |      | 2048  | 1050623 | 1048576 | 512M | 83 | Linu |

```
Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): 82
Changed type of partition 'Linux' to 'Linux swap / Solaris'.
```

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```





3- Use **partprobe** command to update the kernel about changes made to the partition table without requiring a reboot.

```
bash-5.1# partprobe
bash-5.1#
```

4- Verify the new partition

```
bash-5.1# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
nvme1n1              259:0    0   5G  0 disk
nvme2n1              259:1    0   5G  0 disk
└─nvme2n1p1          259:7    0 512M  0 part
```

5- Use **mkswap** command to initialize the partition for use as swap space.

```
bash-5.1# mkswap /dev/nvme2n1p1
Setting up swapspace version 1, size = 512 MiB (536866816 bytes)
no label, UUID=c2a3c0ca-56fa-461a-9f6c-205b569a0793
bash-5.1#
```

6- Use the **blkid** command to retrieve the UUID of the partition, which is needed for persistent configuration

```
bash-5.1# blkid /dev/nvme2n1p1
/dev/nvme2n1p1: UUID="c2a3c0ca-56fa-461a-9f6c-205b569a0793"
UUID="453c014a-01"
bash-5.1#
```

7- Edit **/etc/fstab** and add the following line, replacing **<UUID>** with the actual UUID from the previous step:

UUID=<UUID> swap swap defaults 0 0

```
bash-5.1# vi /etc/fstab
UUID=7B77-95E7 /boot/efi vfat defaults,uid=0,gid=0,umask=077,shortname=winnt 0 2

UUID=c2a3c0ca-56fa-461a-9f6c-205b569a0793 swap swap
defaults 0 0
```





8- **mount -a** command applies all configurations in **/etc/fstab**, and **swapon -a** activates all swap entries.

```
● bash-5.1# mount -a
● bash-5.1# swapon -a
○ bash-5.1#
```

9- Use **swapon -s** lists all active swap spaces.

```
● bash-5.1# swapon -s
Filename                                Type      S
size          Used          Priority
/dev/nvme2n1p1      partition  5
24284           0           -2
○ bash-5.1#
```



**Task 9** - Copy the file `/etc/fstab` to `/var/tmp`. Set permissions and ACLs on `/var/tmp/fstab` to meet the following requirements:

- Owned by user `root` and group `root`.
- User `harry` has read and write permissions.
- User `natasha` (create the user if it doesn't exist) cannot read or write.
- All other users (current and future) can read the file.

**Solution -**

1- Use `cp` command to copy the file `/etc/fstab` to `/var/tmp` directory.

```
bash-5.1# cp /etc/fstab /var/tmp
```

2- Use **chown** command ensures the file is owned by the **root** user and group.

```
• bash-5.1# chown root:root /var/tmp/fstab
• bash-5.1# ls -l /var/tmp/fstab
-rw-r--r--. 1 root root 274 Jan 23 13:26 /var/tmp/fstab
```

3- **setfacl** command explicitly gives **harry** read and write permissions

```
• bash-5.1# useradd harry
• bash-5.1# setfacl -m u:harry:rw /var/tmp/fstab
○ bash-5.1#
```

4- Create the user **natasha** if not present

```
● bash-5.1# useradd natasha  
○ bash-5.1#
```

5- Deny all permissions for **natasha** and grant read-only permissions to all other users using **setfacl** command

```
● bash-5.1# setfacl -m u:natasha:--- /var/tmp/fstab  
● bash-5.1# setfacl -m o:r /var/tmp/fstab  
○ bash-5.1#
```

## 6-Verify the ACL settings.

```
bash-5.1# getfacl /var/tmp/fstab
getfacl: Removing leading '/' from absolute path names
# file: var/tmp/fstab
# owner: root
# group: root
user::rw-
user:harry:rw-
user:natasha:---
group::r--
mask::rw-
other::r--

bash-5.1#
```



**Task 10** - Configure the system to enable persistent Journald log storage with a maximum size of 100 M.

**Solution** - 1-Configure Journald configuration file for storage settings. Set **Storage=persistent** for logs to persist after a reboot and **SystemMaxUse=100M** to cap disk usage for logs using vi editor.

```
bash-5.1# vi /etc/systemd/journald.conf
```

```
[Journal]
Storage=persistent
#Compress=yes
#Seal=yes
#SplitMode=uid
#SyncIntervalSec=5m
#RateLimitIntervalSec=30s
#RateLimitBurst=10000
SystemMaxUse=100M
```



2-Restart the Journald service to apply changes. Use `journalctl --disk-usage` command to verify that the current log size on disk, ensuring the new limit is active.

```
● bash-5.1# journalctl --disk-usage
  Archived and active journals take up 16.0M in the fi
  le system.
● bash-5.1# systemctl restart systemd-journald
○ bash-5.1#
```