



RHCSA.GURU

Solutions : Mock Test 3

Task 1 - Create a user named Max and a group named sysadmin. Grant the user and group sudo privileges so that they can run all commands as root without being prompted for a password. Additionally, set the account expiration for the user Max to December 21, 2025.

Solution - 1-Create the Max user, sysadmin group and add user to group

- Use the **groupadd** command to create the group
- Use **useradd** command to create user
 - **-m** creates the user's home dir if it does not exists
 - **-G** option to add user in group
 - **-e** option specifies an expiration date for the user's account

```
bash-5.1# groupadd sysadmin
bash-5.1# useradd -m -G sysadmin -e 2025-12-21 Max
bash-5.1#
```

2-Edit the sudoers file for the user Max and group sysadmin to allow them to run all commands without being prompted for a password:

```
• bash-5.1# echo "Max ALL=(ALL) NOPASSWD:ALL" | tee /etc/sudoers.d/Max
Max ALL=(ALL) NOPASSWD:ALL
• bash-5.1# echo "%sysadmin ALL=(ALL) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/sysadmin
%sysadmin ALL=(ALL) NOPASSWD:ALL
○ bash-5.1#
```



Task 2 - Ensure that every new user account created automatically contains a file named `Todo.txt` in their home directory. Additionally, create a cron job for the user `Max` (create if it doesn't exist) that runs daily every 1 minute and executes the command `Ex200 Testing with logger`.

Solution - 1-Place a `Todo.txt` file in the `/etc/skel` directory (which contains default files copied to new users' home directories) to ensure every new user has it.

```
● bash-5.1# echo "Sample File" | sudo tee /etc/skel/To
do.txt
Sample File
● bash-5.1# ls /etc/skel/
Todo.txt
○ bash-5.1#
```

2-Opens the crontab editor for the user Max.

```
● bash-5.1# crontab -e -u Max
  crontab: installing new crontab
○ bash-5.1#
```

3-Setup the cron job

- **Enter insert mode:** Add the **cron job** `* * * * * /bin/echo "Ex200 Testing" | /usr/bin/logger`.
- **Cron timing (* * * * *):** Runs the job every minute of every hour, day, month, and week.
- **Save and exit:** Press Esc, then type :wq! and press Enter.

```
* * * * * /bin/echo "Ex200 Testing" | /usr/bin/logger
~
~
```



Task 3 - Start the firewalld service to manage firewall rules. Then, configure the firewall to allow inbound HTTP traffic by adding the HTTP service.

Solution - 1-Install, Start and Enable Firewalld Service:

```
● bash-5.1# dnf install firewalld -y
Updating Subscription Management repositories
Unable to read consumer identity
```

```
● bash-5.1# systemctl start firewalld
● bash-5.1# systemctl enable firewalld
○ bash-5.1# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; vendor preset: enabled)
   Active: active (running) since Sat 2025-01-
```



2- Add the HTTP service to the firewall to allow inbound traffic and reload the firewall to apply changes

```
● bash-5.1# firewall-cmd --permanent --add-service=http  
success
```

```
● bash-5.1# firewall-cmd --reload  
success
```

```
○ bash-5.1#
```

```
● bash-5.1# firewall-cmd --permanent --list-services  
cockpit dhcpv6-client http ssh
```

```
○ bash-5.1#
```



Task 4 - Create a bash script named **symlink.sh** in the **/home** directory that performs the following actions:

- The script should create a symbolic link named **jerry** in the **/home** directory, which points to the **/etc/passwd** file.
- If already exists, the script should output **"Already existed"**.
- If the symbolic link **jerry** is newly created, the script should output **"Created"**.

Solution - 1-Create the script in /home directory

```
bash-5.1# vi /home/symlink.sh
```



```
#!/bin/bash

if [ -L /home/jerry ]; then
    echo "Already existed"
else
    ln -s /etc/passwd /home/jerry
    echo "Created"
fi
```

- **Check symbolic link:** The script checks if the link `/home/jerry` exists using `-L`.
- **Output if exists:** Prints "Already existed" if the link exists.
- **Create link if not:** Creates the link `/home/jerry` pointing to `/home` and prints "Created" if it doesn't exist

2-Make the Script Executable: Change the file permissions to make the script executable:

```
bash-5.1# chmod a+x /home/symlink.sh
bash-5.1# ls -lh /home/symlink.sh
-rwxr-xr-x. 1 root root 130 Jan 18 10:55 /home/symlink.sh
```

3-Run the script to verify

```
● bash-5.1# /home/symlink.sh  
Created  
● bash-5.1# /home/symlink.sh  
Already existed  
○ bash-5.1#
```

Task 5 - Create a Stratis pool using 'nvme1n1' as the disk, and then extend the pool by adding 'nvme2n1' as an additional disk.

Solution - 1- Install the stratis-cli package, start the stratisd service, and then create a Stratis pool named mypool using the /dev/nvme1n1 device with the stratis command.

```
● bash-5.1# dnf install stratis-cli
Updating Subscription Management repositories.
Unable to read consumer identity
```

```
● bash-5.1# systemctl start stratisd
● bash-5.1# stratis pool create mypool /dev/nvme1n1
```



2-Extend the Pool by Adding nvme2n1 and Check the status of the pool to ensure that nvme2n1 has been successfully added

```
• bash-5.1# stratis pool add-data mypool /dev/nvme2n1
• bash-5.1# stratis pool list
```

Name	Total	Used	Free	Properties
Alerts				
mypool	10 GiB	530 MiB	9.48 GiB	~Ca,~Cr, Op 9f25de9c-1e67-4

```
• bash-5.1# stratis blockdev list
```

Pool Name	Device Node	Physical Size	Tier	UUID
mypool	/dev/nvme1n1	5 GiB	DATA	d245328
mypool	/dev/nvme2n1	5 GiB	DATA	7bdf98a

```
○ bash-5.1#
```



Task 6 - Create a Container as a System Startup Service

- (a) Create a container named logserver using the rsyslog image, which is stored in the registry on Docker Hub account of rhcsaguru/rsyslog.
- (b) Configure the container to start automatically as a system service during system startup.

Solution - 1- Install podman and pulls the **rsyslog** image from the **rhcsaguru** user Docker Hub registry

```
○ bash-5.1# dnf install podman -y
Updating Subscription Management repositories.
○ bash-5.1# podman pull rhcsaguru/rsyslog
? Please select an image:
  registry.access.redhat.com/rhcsaguru/rsyslog:latest
  registry.redhat.io/rhcsaguru/rsyslog:latest
  ▶ docker.io/rhcsaguru/rsyslog:latest
```



2- Creates the **logserver** container using the rsyslog image.

```
bash-5.1# podman create --name logserver rhcsaguru/rsyslog  
e8484208d424e7944bbe33a42a699ea57086c4ebf81bd29413b0a36b3bf7dde7  
bash-5.1#
```

3- Generates a systemd service file for the logserver container.

```
● bash-5.1# podman generate systemd --name logserver >  
/etc/systemd/system/logserver.service
```



4- Enables the logserver service to start on boot .

```
bash-5.1# systemctl enable logserver.service
Created symlink /etc/systemd/system/default.target.wants/logserver.service → /etc/systemd/system/logserver.service.
bash-5.1#
```

5-Verifies that the logserver service is running after reboot.

```
bash-5.1# reboot
bash-5.1# podman ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED
PORTS         NAMES
e8484208d424   docker.io/rhcsaguru/rsyslog:latest  rsyslogd -n 10 min
514/tcp       logserver
bash-5.1# systemctl status logserver.service
● logserver.service - Podman container-logserver.service
   Loaded: loaded (/etc/systemd/system/logserver.service; enabled;
   Active: active (running) since Mon 2025-01-20 07:06:04 UTC; 42s
```



Task 7 - Configure a local repository

- Pull the Docker image rhcsaguru/local-repo from Docker Hub.
- Run the image as a container, exposing it on port 80.
- Configure a local YUM repository named local using http://localhost:80/.
- Use the local repository to install the ngrep package.

Solution - 1-Pull the required image from Docker Hub:

```
bash-5.1# podman pull rhcsaguru/local-repo
✓ docker.io/rhcsaguru/local-repo:latest
Trying to pull docker.io/rhcsaguru/local-repo:latest
```

2- Start a container using the image, exposing it on port 80

```
bash-5.1# podman run -d -p 80:80 rhcsaguru/local-repo  
dcc601795335dd03b098be6fb18c2b7f1f176d1540fab49e2ea104888a108e6c  
bash-5.1#
```

3- Ensure that the container is running and accessible.

```
bash-5.1# podman ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
e8484208d424	docker.io/rhcsaguru/rsyslog:latest	rsyslogd -n	14 minutes ago
p 5 minutes	514/tcp	logserver	
dcc601795335	docker.io/rhcsaguru/local-repo:latest	/usr/sbin/nginx -...	24 seconds ago
p 25 seconds	0.0.0.0:80->80/tcp	80/tcp inspiring_ritchie	

```
bash-5.1# curl http://localhost:80/  
<html>  
<head><title>Index of /</title></head>  
<body bgcolor="white">
```

4- Create a repository configuration file in /etc/yum.repos.d . This configures the local repository to use the container's exposed URL.

```
bash-5.1# vi /etc/yum.repos.d/local.repo
```

```
[local]
name=Local Repository
baseurl=http://localhost:80/
enabled=1
gpgcheck=0
```

```
~
```

```
~
```

```
~
```

```
:wq!
```



5-Lists the available packages in the local repository to confirm it is correctly configured.

```
bash-5.1# dnf repoquery --repo=local
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server.
o register.

Local Repository
curl-0:7.61.1-22.el8.x86_64
ngrep-0:1.47-3.1.20180101git9b59468.el8.x86_64
tree-0:1.7.0-15.el8.x86_64
```

6-Installs the ngrep package from the local repository and Confirms that the **egrep** package is installed.

```
bash-5.1# dnf install ngrep

Arch      Version                               Repo      Size
=====
Installing:
ngrep x86_64 1.47-3.1.20180101git9b59468.el8
local 46 k
```

Task 8 - Create a custom network interface named **dummy**. Assign the following configuration to the interface:

Hostname: **dev.example.com** **IP Address:** **192.168.1.42**

Netmask: **255.255.255.0** **Gateway:** **192.168.1.1** **NameServer:** **8.8.8.8**

Solution - 1- Creates a dummy network interface named dummy0. Assigns the IP address 192.168.1.42 with netmask /24 and Sets the gateway to 192.168.1.1.

Command : nmcli connection add type dummy ifname dummy con-name dummy ip4 192.168.1.42/24 gw4 192.168.1.1

```
bash-5.1# nmcli connection add type dummy ifname dummy con-name dummy ip4 192.168.1.42/24 gw4 192.168.1.1
Connection 'dummy' (ebd78d4e-7ede-4008-8fed-7215b5acbf05) successfully added.
```



2- Configures 8.8.8.8 as the DNS server for the dummy0 connection and Sets the system hostname to dev.example.com.

```
• bash-5.1# nmcli connection modify dummy ipv4.dns 8.8  
.8.8  
• bash-5.1# hostnamectl set-hostname dev.example.com
```

4- Brings the dummy interface online.

```
• bash-5.1# nmcli connection up dummy  
Connection successfully activated (D-Bus active path  
: /org/freedesktop/NetworkManager/ActiveConnection/5  
)
```



Task 9 - Create a volume group named guruvg and a logical volume named gurulv using a 5100 MB physical volume located at /dev/nvme1n1, and then mount it to /mymount.

Solution - 1- Install lvm2 and then initializes /dev/nvme1n1 as a physical volume for LVM.

```
bash-5.1# dnf install lvm2
Updating Subscription Management repositories.
Unable to read consumer identity
```

```
bash-5.1# pvcreate /dev/nvme1n1
File descriptor 18 (/dev/urandom) leaked on pvcreate invocation
File descriptor 20 (/dev/urandom) leaked on pvcreate invocation
```


2- Creates a volume group named gurvvg using the physical volume /dev/nvme1n1.

```
• bash-5.1# vgcreate gurvvg /dev/nvme1n1
File descriptor 18 (/dev/urandom) leaked on vgcreate invocation.
File descriptor 20 (/dev/urandom) leaked on vgcreate invocation.
File descriptor 21 (/root/.local/share/code-server/logs/20250120T
on vgcreate invocation. Parent PID 1611: /bin/bash
```

3- Creates a logical volume named gurulv with a size of 5100 MB in the volume group gurvvg.

```
• bash-5.1# lvcreate -L 5100M -n gurulv gurvvg
File descriptor 18 (/dev/urandom) leaked on lvcreate invocation.
File descriptor 20 (/dev/urandom) leaked on lvcreate invocation.
File descriptor 21 (/root/.local/share/code-server/logs/20250120T
```



4- Formats the logical volume with the ext4 file system.

```
● bash-5.1# mkfs.ext4 /dev/guruvg/gurulv
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 1305600 4k blocks and
Filesystem UUID: 4f87241b-5967-4533-b3a1-0219a8
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 8
```

5- Creates the directory `/mymount` to serve as the mount point. Mounts the logical volume to `/mymount`.

```
● bash-5.1# mkdir -p /mymount
○ bash-5.1#
```

```
● bash-5.1# mount /dev/guruvg/gurulv /mymount
mount: (hint) your fstab has been modified,
```



6- Adds an entry to `/etc/fstab` to ensure the volume is mounted at `/mymount` after a reboot.

Command: `echo '/dev/guruvg/gurulv /mymount ext4 defaults 0 0' | sudo tee -a /etc/fstab`

```
● bash-5.1# echo '/dev/guruvg/gurulv /mymount ext4 defaults 0 0' | sudo tee -a /etc/fstab
/dev/guruvg/gurulv /mymount ext4 defaults 0 0
```



Task 10 - Extend the logical volume `/dev/mapper/guruvg-gurulv` to a total size of 8 GiB by adding the physical volume `/dev/nvme2n1` to the existing volume group `guruvg`, and then resizing the logical volume accordingly.

Solution - 1- Initializes `/dev/nvme2n1` as a physical volume.

```
● bash-5.1# pvcreate /dev/nvme2n1
File descriptor 18 (/dev/urandom) leaked on pvcreate
File descriptor 20 (/dev/urandom) leaked on pvcreate
File descriptor 21 (/root/.local/share/code-server/lo
on pvcreate invocation. Parent PID 1611: /bin/bash
```



2- Adds `/dev/nvme2n1` to the volume group `guruvg`.

```
● bash-5.1# vgextend guruvg /dev/nvme2n1
File descriptor 18 (/dev/urandom) leaked on vgextend
File descriptor 20 (/dev/urandom) leaked on vgextend
File descriptor 21 (/root/.local/share/code-server/log
on vgextend invocation. Parent PID 1611: /bin/bash
```

3- Resizes the logical volume `gurulv` to 8 GiB. Expands the file system to use the newly allocated space in the logical volume.

```
● bash-5.1# lvextend -L 8G /dev/mapper/guruvg-gurulv
File descriptor 18 (/dev/urandom) leaked on lvextend
invocation. Parent PID 1649: /bin/bash
File descriptor 20 (/dev/urandom) leaked on lvextend
invocation. Parent PID 1649: /bin/bash
```



4- Verify the change .

```
● bash-5.1# sudo lvsdisplay /dev/mapper/guruvg-gurulv |  
  grep 'LV Size'  
    LV Size                8.00 GiB  
○ bash-5.1#
```