Appendix B

Solutions to RHCSA Sample Exam 1

The questions in the RHCSA Sample Exam 1 will help measure your understanding of the material covered in this book. As indicated in the introduction, you should be capable of completing the RHCSA exam in 3 hours. However, we have capped the time for this lab at 2.5 hours to accustom you to working under time constraints.

The RHCSA exam follows a “closed book” format. Nevertheless, you are permitted to refer to any documentation available on the Red Hat Enterprise Linux computer. Although test centers permit note-taking, these notes cannot be taken out of the examination room.

In the majority of cases, there isn't a single solution or method to resolve a problem or install a service. With the plethora of options available in Linux, it is impossible to cover every potential scenario.

For the forthcoming exercises, avoid using a production computer. Even a minor error in any of these exercises could render Linux unbootable. If you cannot recover using the steps provided in these exercises, you might need to reinstall Red Hat Enterprise Linux. Consequently, you may not be able to recover any data saved on the local system.

Red Hat conducts its exams electronically, which is why the exams in this book can be accessed from the McGraw Hill companion website. This exam, named RHCSAsampleexam1, is available in PDF format. For instructions on setting up RHEL 9 as a suitable system for a practice exam, please refer to Appendix A.

RHCSA Sample Exam 1 Discussion

In this discussion, we'll describe briefly one way to meet the requirements listed for the RHCSA Sample Exam 1.

1. To complete this task, review Exercise 5-2, "Recover the Root Password," in Chapter 5.
2. Assuming that the network interface is named eth0, that the NAT subnet address of your VMware Workstation Player setup is 192.168.0.100, and that the IP of the default gateway/DNS is 192.168.0.2, execute the following commands:

   ```
   # nmcli con mod eth0 ipv4.addresses "192.168.0.100/24"
   # nmcli con mod eth0 ipv4.gateway "192.168.0.2"
   # nmcli con mod eth0 ipv4.method manual
   # nmcli con mod eth0 ipv4.dns "192.168.0.2"
   # nmcli con down eth0
   # nmcli con up eth0
   # nmcli con show eth0
   # hostnamectl set-hostname server1.example.com
   ```

3. To complete this task, review Exercise 4-2 in Chapter 4.
4. Create a file named /etc/yum.repos.d/epel.repo and add the following content:

   ```
   [epel]
   name = epel
   baseurl = http://linuxsoft.cern.ch/epel/9/Everything/x86_64/
   gpgcheck = no
   ```

   Then, run the following command to install the htop RPM:

   ```
   # dnf install htop
   ```

5. Use the **parted** command to create a new partition. Assume that your hard drive is /dev/vda, the new partition is number 3, and there is some free space starting at about 19GB. Start the **parted** utility with **parted /dev/vda** and type

   ```
   (parted) unit mib
   (parted) print
   (parted) mkpart primary 19000MiB 19500MiB
   (parted) set 3 lvm on
   (parted) quit
   ```

   Then, run the following commands:

   ```
   # pvcreate /dev/vda3
   # vgcreate -s 8M vg01 /dev/vda3
   # lvcreate -l 32 -n lv_project vg01
   ```

6. Format the volume and create the mount point:

```
# mkfs.xfs /dev/vg01/lv_project
# mkdir /project
```

   To make sure that the volume is automatically mounted the next time the system is booted, configure it in /etc/fstab to the appropriate format, with the UUID associated with the volume, as provided by the **blkid** command:

```
# blkid /dev/vg01/lv_project
```

   Then, add the following line to /etc/fstab:

```
UUID=<substitute with UUID value> /project xfs defaults 0 0
```

7. Run the following command to complete this task:

```
# find /etc -type f -name "*.conf" >/root/configfiles.txt
```

8. To complete this task, run:

```
# tar -cjf /tmp/etc.tar.bz2 /etc
```

9. The /home/engineers directory should be owned by the group engineers. As long as users donna and mike are not part of that group, and other users don't have permissions (or ACLs) on that directory, access should be limited to members of the engineers group. The directory should also have SGID permissions:

```
# useradd nancy
# echo "changeme!" | passwd --stdin nancy
# useradd randy
# echo "changeme!" | passwd --stdin randy
# useradd donna
# echo "changeme!" | passwd --stdin donna
# useradd mike
# echo "changeme!" | passwd --stdin mike
# groupadd -g 2000 engineers
# usermod -aG engineers nancy
# usermod -aG engineers randy
# mkdir /home/engineers
# chgrp engineers /home/engineers
# chmod 2770 /home/engineers
```

10. If you've modified user mike's account to make his account expire in seven days, the right expiration date should appear in the output to the **chage -l mike** command. To complete the task, run

```
# chage -E $(date -d "+7 days" +"%Y-%m-%d") mike
```

11. There are a number of ways to set up a cron job; it could be configured in the /etc/crontab file or as a cron job for the user root or mike with the **crontab -u <user> -e** command. To complete the exercise, you can add the following line to /etc/crontab:

    ```
    50 3 2 * * root /bin/find /home/mike/tmp -type f -exec /bin/rm {} \;
    ```

12. One effective way to share a file between users is to create a group, add the relevant users to that group, and then change the file's group ownership to that group. Create a new group, for example, "projectgroup":

    ```
    # groupadd projectgroup
    ```

    Add users mike and donna to projectgroup:

    ```
    # usermod -aG projectgroup mike
    # usermod -aG projectgroup donna
    ```

    Next, as user mike, create the file, change its group ownership, and modify the file permissions to enable read access for the group:

    ```
    $ touch /opt/project.test
    $ chown mike /opt/project.test
    $ chgrp projectgroup /opt/project.test
    $ chmod 640 /opt/project.test
    ```

13. Create a file /usr/local/bin/backup.sh with the following content and make it executable with **chmod +x /usr/local/bin/backup.sh**:

    ```
    #!/bin/bash

    # Check if an argument is provided
    if [ "$#" -ne 1 ]; then
        echo "Usage: backup.sh <DIRECTORY>"
        exit 1
    fi
    # Check if the provided argument is a directory
    if [ ! -d "$1" ]; then
        echo "Error: $1 is not a directory."
        exit 1
    fi
    # Create a tar.gz archive
    tar -czf backup_$(date +%Y%m%d%H%M%S).tar.gz -C $1 .
    ```

    Next, schedule the script to run automatically at 2:00 A.M. daily to create a backup of the /home directory in the /tmp filesystem. To do so, add the following line to /etc/crontab:

    ```
    0 2 * * * root /usr/local/bin/backup.sh /home
    ```

14. First, create the user sam:

    ```
    # useradd -u 1234 -d /nethome/sam sam
    ```

    Then, configure NFS automounting. Install the autofs package:

    ```
    # dnf install autofs nfs-utils
    ```

    In the /etc/auto.master file, add the following line:

    ```
    /nethome /etc/auto.nethome
    ```

    Now, you need to create and configure the map file (/etc/auto.nethome) you just referenced in the auto.master file. This file will contain the following line (substitute for the IP address of tester1.example.com):

    ```
    sam -rw <ip_address_of_tester1.example.com>:/exports/sam
    ```

    After the configuration files are set up, start and enable the autofs service:

    ```
    # systemctl enable autofs --now
    ```

    Now, if you open a shell as user sam (**su – sam**), the NFS share from tester1.example.com:/sam will be automatically mounted as user sam's home directory. The **-rw** option ensures that the NFS share is mounted in read-write mode.

15. On tester1.example.com, run

    ```
    # semanage port -a -t http_port_t -p tcp 8234
    # systemctl restart httpd
    ```

    After running this command, Apache should be able to bind to port 8234. You can verify this with the following command on tester1.example.com:

    ```
    # ss -tlpn | grep 8234
    ```

16. Run the following commands on tester1.example.com to allow other hosts to connect to the web server on port 8234:

    ```
    # firewall-cmd --add-port=8234/tcp --permanent
    # firewall-cmd --reload
    ```

    Change the SELinux context of the /html directory to the type httpd_sys_content_t. This type is used for static web content that should be accessible by **httpd**:

    ```
    # semanage fcontext -a -t httpd_sys_content_t "/html(/.*)?"
    ```

    The argument **"/html(/.*)?"** is a regular expression that matches the /html directory and all its subdirectories and files. The **-a** option is used to add a record. After that, use the **restorecon** command to apply this context mapping to the running filesystem:

    ```
    # restorecon -R -v /html
    ```

Then, the following command on server1.example.com should display the “Success!” test page (substitute for the IP address of tester1):

```
# curl http://<ip_address_of_tester1>:8234
```

17. Run the following commands as user mike to start the container:

```
$ mkdir ~/html
$ podman run -d --name httpd \
 -v /home/mike/html:/var/www/html:Z \
 -p 8081:8080 \
 registry.access.redhat.com/ubi9/httpd-24
```

Execute the following command to generate a systemd configuration file for the container service:

```
$ podman generate systemd --name httpd --new --files
```

This command results in creating a container-httpd.service. Next, copy this file into the /home/mike/.config/systemd/user directory. If this directory doesn’t already exist, you’ll have to create it:

```
$ mkdir -p /home/mike/.config/systemd/user
$ cp container-httpd.service ~/.config/systemd/user
```

Afterwards, you’ll need to stop and remove the current httpd container:

```
$ podman stop httpd
$ podman rm httpd
```

Now, it’s time to refresh systemd and set up the service to automatically start when your system boots:

```
$ systemctl --user daemon-reload
$ systemctl --user enable container-httpd.service
$ loginctl enable-linger
```

To verify that everything is set up correctly, reboot your system and check if the container is running:

```
$ systemctl --user status container-httpd.service
$ podman ps
```

18. Just like with all Red Hat exams, it’s essential for your modifications to endure a system reboot. Therefore, reboot the system and verify that your configurations remain fully functional.

This page is intentionally left blank to match the printed book.