RHCSA Practice Tasks

Environment Details:

Host running: RHEL 9+

A user named: Student with Sudo privileges

2 Additional disks with at least 6GB Storage

A Red Hat Developer Free Subscription

Internet Access

A network interface called: eth0 (preferably)

Users:

1.Create a group called admin

1.1. The group admin should have a GID of 10015

2.Create 3 users named: Andrew, Dan and Natalie

2.1: Andrew should have an UID of 1046

2.2: Natalie should have a non interactive login shell

2.3:Andrew and Dan should be have the group admin as a supplementary group

Run the script as root: User.sh to Grade

Privileges:

1.All members of the admin group should have sudo privileges for all commands

2.Dan should have sudo privileges for all commands

Run the script as root: Privileges.sh to Grade

User-defaults:

1.All newly created users should have to change their passwords every 30 days

2.All newly created users should receive a warning 5 days before expiry

3.All newly created users should have a minimum password age of 10 days

4.All newly created users should have a file called **“All-users**” with the message: **“Created for all homes”** within their home directory.

Run the script as root: User-defaults.sh to Grade

User-passwords:

1.Create 3 users named Pass1,Pass2 and Pass3 with a password of Redhat

1.1: The user Pass1’s password should expire every 10 days

1.2 The users Pass2’s account should expire in 30 days from the current date

1.3 The user Pass3 should change their password upon first login

Run the script as root: User-passwords.sh to Grade

Permissions:

1.Create a directory called /home/perms

2.Ensure the directory is owned by the user Andrew and the group admin

3.Ensure that all newly created content within the directory will inherit the group ownership of the directory

4.Ensure only the owner of the directory or owner of the files within that directory can delete files within that directory

5.Ensure the following privileges are set on the directory: Owner: Read-Write-Execute, Group-Read-Write-Execute, Others-Execute

Run the script as root: Permissions.sh to Grade

Files:

Run the script as root: Files-start.sh to Start

1.Find all files belonging to the user William and save their location with the file /home/Paths

2.Create a symbolic link called /home/Softlink pointing to the file /home/Linkme

3.Create a hard link called /home/Hardlink Pointing to the file /home/Linkme

4.Create an archive with gzip compression called /home/Archiveme.tar.gz with the contents of the directory /home/Archiveme

Run the script as root: Files-finish.sh to Grade

Recurrent Jobs:

1.Create a recuring job called sysjob which will run the script called /home/myscript.sh every Tuesday at 5 PM as root

2.Create a recuring job for the user Andrew which will run the: “echo test” command every 10 minutes everyday in January

Run the script as root: Recurrent-jobs.sh to Grade

Targets

1.Set the default target the system should boot in to multi-user.target

Run the script as root: Default-target.sh to Grade

Time-date:

1.Set the local time-zone to Bucharest

2.Add an NTP server with the address: ntp.server.com

3.Enable NTP

Run the script as root: Time-date.sh to Grade

Repositories:

1.Set up a repository with the following details:

name: BaseOS

baseurl: http://myrepo.com

enabled: true

gpgcheck: false

id: BaseOS.Dvd

Run the script as root: Repositories.sh to Grade

Networking:

1.Set the hostname to Mytesthost.com

2.Reference the IP 10.1.1.1 to the name “private” on the localhost file

3.Create a network connection with the following details:

Name: myconnection

Interface: eth0 (If exists)

IPv4: 192.168.1.1/24 and 192.168.1.2/24

Gateway: 192.168.1.254

Dns: 192.168.1.254

Autoconnect: True

Run the script as root: Hosts-file.sh to Grade

Run the script as root: Network-connection.sh to Grade

Run the script as root: Hostname.sh to Grade

Firewall default zone:

1.Set the default zone to public

Run the script as root: Firewall.sh to Grade

Selinux Mode:

1.Set the enforcement level of SELinux to permissive

2.Configure SELinux to start in permissive mode on boot

Run the script as root: SELinux-mode.sh to Grade

Troubleshooting boot:

Run the script as root: Troubleshooting-boot-start.sh to Start

1.Set the root password to redhat

2.Fix the boot issue

Troubleshooting Service:

Run the script as root: Troubleshooting-service-start.sh to Start

1.Install the httpd service

2.Enable the httpd service

3.Run the troubleshooting-service.sh script

4.Attempt to start the httpd service

5.Make the httpd service work and configure firewalld to make it accessible from outside the localhost as well

Run the script as root: Troubleshooting-service-finish.sh to Grade

Basic Storage:

1.Create 2 partitions with a size of 650 M using the GPT partitioning scheme

Run the script as root: Basic-storage to Grade

Advanced Storage:

1.Create a volume group called vg1 using one of the previously created partitions

2.Create a logical volume called lv01 with a size of 300M

3.Format the logical volume lv01 with an xfs filesystem

4.Create a directory called /mountlvm

5.Mount the logical volume lv01 on /mountlvm

6.Ensure persistent mounting on boot of the logical volume lv01 on /mountlvm by UUID

Run the script as root: Advanced-storage.sh to Grade

Advanced storage 2:

1.Extend the volume group vg1 with the second partition

2.Extend the logical volume lv01 by 200M

3.Grow the filesystem of the logical volume lv01

4.Create a logical volume named lv-swap1 with a size of 300M

5.Format lv-swap1 as swap space

6.Mount the logical volume lv-swap1 persistently on boot using it’s UUID

Run the script as root: Advanced-storage2.sh to Grade

Storage Stack

1.Install the stratisd service and cli

2.Start the stratisd service

3.Enable the stratisd service

4.Create a stratis pool called pool1

5.Create a stratis filesystem within pool1 called fs1

6.Create a directory called /mountstratis

7.Persistently mount the stratis filesystem fs1 on /mountstratis using it's UUID

Run the script as root: Stratis.sh to Grade

Podman

0.Install container-tools

1.As the user student directly

2.Create the directory /home/student/containercontent

3.Create the file /home/student/containercontent/index.html with the message "Hello Container!"

4.Create a container called webserver running in detached mode running and httpd image with the following details:

4.1.Forward traffic from port 8080 of the host to port 8080 of the container

4.2.Mount /home/student/containercontent within /var/www/html within the container

5.Allow traffic on port 8080/tcp within firewalld

Run the script as student: Podman.sh to Grade

Podman Service:

1.As the user student directly

2.Create the following path: /home/student/.config/systemd/user

3.Create a service file based on the previously created webserver container

4.Enable the newly created service and ensure it starts whenever the system boots

Run the script as student: Podman-service.sh to Grade

Task: Tuned

1.Enable the the tuned service

2.Start the tuned service

3.Check the recommended profile by tuned

4.Set the recommend profile by tuned

Run the script as root: Tuned.sh to Grade

Additional Task: Remote Transfer

1.Use SCP or Rsync or SFTP to transfer something