

RH134 Report

Ruojia Zhang

## Chapter 1

Figure 1

Guided exercise for Writing Simple Bash Scripts. Also set roger Zhang as ps1.

```
[student@workstation ~]$ PS1='[Roger Zhang@\h \w] >'  
[Roger Zhang@workstation ~] >lab console-write start  
  
Starting lab.  
  
Preparing servera for lab exercise work:  
  
  · Verifying that servera is running: ..... SUCCESS  
  · Verifying that servera has required packages: ..... SUCCESS  
[Roger Zhang@workstation ~] >ssh student@servera  
Activate the web console with: systemctl enable --now cockpit.socket  
  
This system is not registered to Red Hat Insights. See https://cloud.redhat.com/  
to register this system, run: insights-client --register  
  
Last login: Tue Sep  1 08:19:05 2020 from 172.25.250.9  
student@servera ~]$ vim firstscript.sh  
student@servera ~]$ sh firstscript.sh  
student@servera ~]$ cat output.txt  
this is my first bash script  
  
#####  
student@servera ~]$ vim firstscript.sh  
student@servera ~]$ chmod a+x firstscript.sh  
student@servera ~]$ ./firstscript.sh  
student@servera ~]$ rm firstscript.sh output.txt  
student@servera ~]$ exit  
Logout  
Connection to servera closed.  
[Roger Zhang@workstation ~] >lab console-write finish  
Completing the lab on servera:  
  
Lab finished.  
  
[Roger Zhang@workstation ~] >
```

Note: Write and execute a simple Bash script. Redirect the output of a simple Bash script to a file. You can see the lab finished here.

Figure 2

Lab: Improving Command-line Productivity. In this lab, I create a Bash script that can filter and get relevant information from different hosts.

```
student@workstation:~  
File Edit View Search Terminal Help  
· Preparing servera for required packages: ..... SUCCESS  
· Preparing serverb for required packages: ..... SUCCESS  
· Preparing workstation for required packages: ..... SUCCESS  
· Preparing sudo in servera: ..... SUCCESS  
· Preparing sudo in serverb: ..... SUCCESS  
· Prepare /var/log/secure in servera: ..... SUCCESS  
· Prepare /var/log/secure in serverb: ..... SUCCESS  
  
[Roger Zhang@workstation ~] > mkdir -p /home/student/bin  
[Roger Zhang@workstation ~] >vim ~/bin/bash-lab  
[Roger Zhang@workstation ~] >chmod a+x ~/bin/bash-lab  
[Roger Zhang@workstation ~] >vim ~/bin/bash-lab  
[Roger Zhang@workstation ~] >bash-lab  
[Roger Zhang@workstation ~] >lab console-review grade  
  
Grading the lab:  
  
· Verifying that the script exists: ..... PASS  
· Verifying that the script is executable: ..... PASS  
· Verifying that the script is using a loop: ..... PASS  
· Verifying that the output file for servera exists: ..... PASS  
· Verifying that the output file for serverb exists: ..... PASS  
· Verifying hostnames in output file for servera: ..... PASS  
· Verifying hostnames in output file for serverb: ..... PASS  
· Verifying CPU data required in output file for servera: ..... PASS  
· Verifying CPU data required in output file for serverb: ..... PASS  
· Verifying CONFIGFILE filtered content for servera: ..... PASS  
· Verifying CONFIGFILE filtered content for serverb: ..... PASS  
· Verifying LOGFILE filtered content for servera: ..... PASS  
· Verifying LOGFILE filtered content for serverb: ..... PASS  
· Verifying comment lines in output file for servera: ..... PASS  
· Verifying comment lines in output file for serverb: ..... PASS  
  
Overall lab grade..... PASS  
  
[Roger Zhang@workstation ~] >lab console-review finish  
  
Completing the lab on workstation, servera and serverb:  
  
· Cleaning exercise output files in workstation: ..... SUCCESS  
· Restoring sudoers file in servera: ..... SUCCESS  
· Restoring sudoers file in serverb: ..... SUCCESS  
· Clean /var/log/secure in servera: ..... SUCCESS  
· Clean /var/log/secure in serverb: ..... SUCCESS  
  
Lab finished.  
[Roger Zhang@workstation ~] >
```

Note: Create a Bash script and redirect its output to a file. Use loops to simplify your code. Filter the relevant content using grep and regular expressions.

## Chapter Review

In this chapter, I learned: How to create and execute simple Bash scripts. How to use loops to iterate through a list of items from the command-line and in a shell script. How to search for text in log files and configuration files using regular expressions and grep.

## Chapter 2

Figure 3

Guided Exercise: Scheduling a Deferred User Job. You can see in the screenshot that all test are passed

```
}${SHELL:-/bin/sh} << 'marcinDELIMITER04991d8e'
date >> /home/student/myjob.txt
marcinDELIMITER04991d8e
[student@servera ~]$ atq
1      Sun Dec  4 18:43:00 2022 a student
3      Mon Dec  5 16:00:00 2022 g student
4      Mon Dec  5 16:05:00 2022 b student
[student@servera ~]$ atrm 3
[student@servera ~]$ atq
1      Sun Dec  4 18:43:00 2022 a student
4      Mon Dec  5 16:05:00 2022 b student
[student@servera ~]$ logout
Connection to servera closed.
[Roger Zhang@workstation ~] >lab scheduling-at finish

Completing the lab on servera:
· Removing deferred jobs for student on servera..... SUCCESS
· Removing /home/student/myjob.txt from servera..... SUCCESS
· Removing /home/student/tea.txt from servera..... SUCCESS
· Removing /home/student/cookies.txt from servera..... SUCCESS

Lab finished.

[Roger Zhang@workstation ~] >
```

Note: I Schedule a job to run at a specified time in the future. Inspect the commands that a scheduled job runs. Delete the scheduled jobs.

Figure 4

Guided Exercise: Managing Temporary Files. You can see in the screenshot that all tests are passed

```
student@workstation:~  
File Edit View Search Terminal Help  
Starting lab.  
  
Preparing servera for lab exercise work:  
  
. Backing up /usr/lib/tmpfiles.d/tmp.conf on servera..... SUCCESS  
. Ensuring /etc/tmpfiles.d/tmp.conf does not exist on servera. SUCCESS  
. Ensuring custom config file does not exist on servera..... SUCCESS  
. Ensuring /run/momentary does not exist on servera..... SUCCESS  
  
[Roger Zhang@workstation ~] >ssh student@servera  
Activate the web console with: systemctl enable --now cockpit.socket  
  
This system is not registered to Red Hat Insights. See https://cloud.redhat.com/  
To register this system, run: insights-client --register  
  
Last login: Sun Dec  4 18:35:48 2022 from 172.25.250.9  
[student@servera ~]$ sudo -i  
[sudo] password for student:  
[root@servera ~]# cp /usr/lib/tmpfiles.d/tmp.conf \  
>  
cp: missing destination file operand after '/usr/lib/tmpfiles.d/tmp.conf'  
Try 'cp --help' for more information.  
[root@servera ~]# cp /usr/lib/tmpfiles.d/tmp.conf /etc/tmpfiles.d/tmp.conf  
[root@servera ~]# q /tmp 1777 root root 5d^C  
[root@servera ~]# vim /etc/tmpfiles.d/tmp.conf  
[root@servera ~]# systemctl-tmpfiles --clean /etc/tmpfiles.d/tmp.conf  
[root@servera ~]# vim /etc/tmpfiles.d/momentary.conf  
[root@servera ~]# systemd-tmpfiles --create /etc/tmpfiles.d/momentary.conf  
[root@servera ~]# ls -ld /run/momentary  
drwx----- 2 root root 40 Dec  4 18:48 /run/momentary  
[root@servera ~]# touch /run/momentary/testfile  
[root@servera ~]# sleep 30  
[root@servera ~]# systemctl-tmpfiles --create /etc/tmpfiles.d/momentary.conf  
[root@servera ~]# ls -l /run/momentary/testfile  
-rw-r--r-- 1 root root 0 Dec  4 18:48 /run/momentary/testfile  
[root@servera ~]# logout  
[student@servera ~]$ logout  
Connection to servera closed.  
[Roger Zhang@workstation ~] >lab scheduling-tempfiles finish  
  
Completing the lab on servera:  
  
. Restoring original /usr/lib/tmpfiles.d/tmp.conf on servera.. SUCCESS  
. Deleting /etc/tmpfiles.d/tmp.conf from servera..... SUCCESS  
. Deleting custom config file from servera..... SUCCESS  
. Deleting /run/momentary from servera..... SUCCESS  
  
Lab finished.  
  
[Roger Zhang@workstation ~] >
```

Note: I have done: Configure systemd-tmpfiles to remove unused temporary files from /tmp. Configure systemd-tmpfiles to periodically purge files from another directory.

### Chapter Review

In this chapter, I learned: Jobs that are scheduled to run once in the future are called deferred jobs or tasks. Recurring user jobs execute the user's tasks on a repeating schedule. Recurring system jobs accomplish administrative tasks on a repeating schedule that have system-wide impact. The systemd timer units can execute both the deferred or recurring jobs.

## Chapter 3

Figure 5

Guided Exercise: Adjusting Tuning Profiles. You can see in the screenshot that all test are passed

```
student@workstation:~$  
File Edit View Search Terminal Help  
  
This system is not registered to Red Hat Insights. See https://cloud.redhat.com/  
to register this system, run: insights-client --register  
  
Last login: Sun Dec  4 18:45:36 2022 from 172.25.250.9  
[student@servera ~]$ yum list tuned  
Red Hat Enterprise Linux 8.2 BaseOS (dvd)           151 MB/s | 2.2 MB    00:00  
Red Hat Enterprise Linux 8.2 AppStream (dvd)        208 MB/s | 5.8 MB    00:00  
Last metadata expiration check: 0:00:01 ago on Sun 04 Dec 2022 06:53:41 PM EST.  
Installed Packages  
tuned.noarch                         2.13.0-6.el8          @anaconda  
[student@servera ~]$ systemctl is-enabled tuned  
enabled  
[student@servera ~]$ systemctl is-active tuned  
active  
[student@servera ~]$ sudo tuned-adm list  
[sudo] password for student:  
Available profiles:  
- accelerator-performance      - Throughput performance based tuning with disabled higher latency STOP states  
- balanced                   - General non-specialized tuned profile  
- desktop                     - Optimize for the desktop use-case  
- hpc-compute                - Optimize for HPC compute workloads  
- intel-sst                  - Configure for Intel Speed Select Base Frequency  
- latency-performance        - Optimize for deterministic performance at the cost of increased power consumption  
- network-latency            - Optimize for deterministic performance at the cost of increased power consumption, focused on low latency network performance  
- network-throughput         - Optimize for streaming network throughput, generally only necessary on older CPUs or 40G+ networks  
- powersave                 - Optimize for low power consumption  
- throughput-performance     - Broadly applicable tuning that provides excellent performance across a variety of common server workloads  
- virtual-guest              - Optimize for running inside a virtual guest  
- virtual-host               - Optimize for running KVM guests  
Current active profile: virtual-guest  
[student@servera ~]$ sudo tuned-adm profile powersave  
[student@servera ~]$ sudo tuned-adm active  
Current active profile: powersave  
[student@servera ~]$ logout  
Connection to servera closed.  
[Roger Zhang@workstation ~] >lab tuning-profiles finish  
  
Cleaning up the lab on servera:  
  · Set active tuning profile to virtual-guest..... SUCCESS  
Lab finished.  
[Roger Zhang@workstation ~] >
```

Note: In this exercise, I've done tune a server's performance by activating the tuned service and applying a tuning profile.

Figure 6

Lab: Tuning System Performance. You can see in the screenshot that all tests are passed

```

student@workstation:~$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum;pgrep md5sum)
  PID %CPU NI COMMAND
 1698 97.8 2 sha1sum
 1714 98.1 -2 md5sum
[student@server ~]$ sudo renice -n 10 1698 1714
1698 (process ID) old priority 2, new priority 10
1714 (process ID) old priority -2, new priority 10
[student@server ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum;pgrep md5sum)
  PID %CPU NI COMMAND
 1698 98.6 10 sha1sum
 1714 98.8 10 md5sum
[student@server ~]$ logout
Connection to server closed.
[Roger Zhang@workstation ~] >lab tuning-review grade

Grading the student's work on serverb:
  - Check nice levels..... PASS
  - Check active tuning profile..... PASS

Overall lab grade..... PASS

[Roger Zhang@workstation ~] >lab tuning-review finish

Cleaning up the lab on serverb:
  - Stop processes used during lab..... SUCCESS
  - Set active tuning profile to virtual-guest..... SUCCESS

Lab finished.

```

Note: I have done: Activate a specific tuning profile for a computer system. Adjust the CPU scheduling priority of a process.

### Chapter Review

In this chapter, I learned: The tuned service automatically modifies device settings to meet specific system needs based on a pre-defined selected tuning profile. To revert all changes made to system settings by a selected profile, either switch to another profile or deactivate the tuned service. The system assigns a relative priority to a process to determine its CPU access. This priority is called the nice value of a process. The nice command assigns a priority to a process when it starts. The renice command modifies the priority of a running process.

### Chapter 4

Figure 7

Guided Exercise: Securing Files with ACLs. You can see in the screenshot that all test are passed

```
student@workstation:~
```

```
File Edit View Search Terminal Help
```

```
Password:
```

```
[consultant2@servera ~]$ ./loadavg.sh
[bash: ./loadavg.sh: No such file or directory]
```

```
[consultant2@servera ~]$ cd /shares/content/
[consultant2@servera content]$ ./loadavg.sh
ldavg 0.23, 0.11, 0.04
```

```
[consultant2@servera content]$ mkdir reports
[consultant2@servera content]$ echo "TEST REPORT" > reports/test.txt
[consultant2@servera content]$ logout
[student@servera ~]$ su - consultant1
Password:
```

```
[consultant1@servera ~]$ cat /shares/content/serverb-loadavg.txt
cat: /shares/content/serverb-loadavg.txt: Permission denied
```

```
[consultant1@servera ~]$ logout
[student@servera ~]$ su - sysadmin1
Password:
```

```
[sysadmin1@servera ~]$ getfacl /shares/content
getfacl: Removing leading '/' from absolute path names
# file: shares/content
# owner: root
# group: operators
# flags: -s-
user::rwx
user:consultant1:---
group::rwx
mask::rwx
other::---
default:user::rwx
default:user:consultant1:---
default:group::rwx
default:group:consultants:rwx
default:mask::rwx
default:other:---
```

```
[sysadmin1@servera ~]$ logout
[student@servera ~]$ logout
Connection to servera closed.
[Roger Zhang@workstation ~] >lab acl-secure finish
```

```
Completing the lab in servera:
```

```
· Cleaning exercise files and folders in servera: ..... SUCCESS
· Deleting users in servera: ..... SUCCESS
· Deleting groups in servera: ..... SUCCESS
```

```
Lab finished.
```

```
[Roger Zhang@workstation ~] >
```

Note: In this exercise, I've done Use ACL entries to grant access to a group, and deny access to one of its members. Verify that the existing files and directories reflect the new ACL permissions. Set the default ACL on a directory, and confirm that new files and directories inherit its configuration.

Figure 8

Lab: Controlling Access to Files with ACLs. You can see in the screenshot that all tests are passed

```

File Edit View Search Terminal Help
[root@serverb ~]# logout
[student@serverb ~]$ su - manager1
Password:
[manager1@serverb ~]$ cd /shares/cases
[manager1@serverb cases]$ echo hello > manager1.txt
[manager1@serverb cases]$ mkdir manager1.dir
[manager1@serverb cases]$ echo hello > manager1.dir/test.txt
[manager1@serverb cases]$ logout
[student@serverb ~]$ su - contractor1
Password:
[contractor1@serverb ~]$ cd /shares/cases
[contractor1@serverb cases]$ echo hello > manager1.txt
[contractor1@serverb cases]$ cat shortlist.txt
###Shortlist of Clients to call###TEMPLATE###
[contractor1@serverb cases]$ mkdir contractor1.dir
[contractor1@serverb cases]$ echo hello > contractor1.dir/test.txt
[contractor1@serverb cases]$ logout
[student@serverb ~]$ su - contractor3
Password:
[student@serverb ~]$ su - contractor3
Password:
[contractor3@serverb ~]$ cd /shares/cases
[contractor3@serverb cases]$ cat shortlist.txt
###Shortlist of Clients to call###TEMPLATE###
[contractor3@serverb cases]$ logout
[student@serverb ~]$ logout
Connection to serverb closed.
[Roger Zhang@workstation ~] >lab acl-review grade

Grading the lab:

· Verifying folder permissions in serverb: ..... PASS
· Verifying folder group ownership in serverb: ..... PASS
· Verifying contractors ACL in serverb: ..... PASS
· Verifying contractor3 ACL in serverb: ..... PASS
· Verifying contractors Default ACL in serverb: ..... PASS
· Verifying contractor3 Default ACL in serverb: ..... PASS

Overall lab grade..... PASS

[Roger Zhang@workstation ~] >lab acl-review finish

Completing the lab in serverb:

· Cleaning exercise files and folders in serverb: ..... SUCCESS
· Deleting users in serverb: ..... SUCCESS
· Deleting groups in serverb: ..... SUCCESS

Lab finished.

[Roger Zhang@workstation ~] >

```

Note: I have done: Configure set-GID permission on a folder, to inherit group ownership on files and folders inside. Configure ACL entries to allow or deny read/write/execute permissions to users and groups on files and directories. Configure default ACL to get the right ACL and file permissions automatically, on new files and directories.

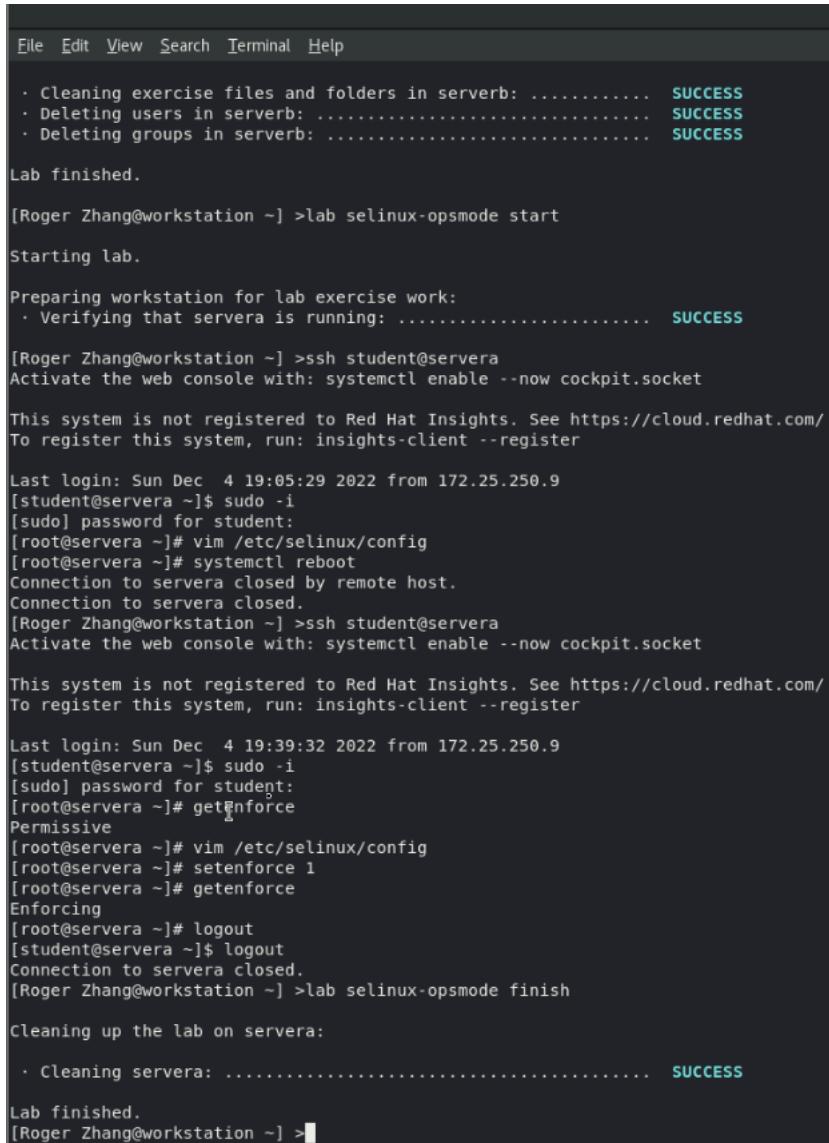
### Chapter Review

In this chapter, I learned: ACLs provide fine-grained access control to files and directories. The getfacl command displays the ACLs on a file or directory. The setfacl command sets, modifies, and removes default and standard ACLs on files and directories. Use default ACLs for controlling new files and directories permissions. Red Hat Enterprise Linux uses systemd and udev to apply predefined ACLs on devices, folders, and files.

## Chapter 5

Figure 9

Guided Exercise: Changing the SELinux Enforcement Mode. You can see in the screenshot that all test are passed



The screenshot shows a terminal window with the following content:

```
File Edit View Search Terminal Help

· Cleaning exercise files and folders in serverb: ..... SUCCESS
· Deleting users in serverb: ..... SUCCESS
· Deleting groups in serverb: ..... SUCCESS

Lab finished.

[Roger Zhang@workstation ~] >lab selinux-opsmode start

Starting lab.

Preparing workstation for lab exercise work:
· Verifying that servera is running: ..... SUCCESS

[Roger Zhang@workstation ~] >ssh student@servera
Activate the web console with: systemctl enable --now cockpit.socket

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

Last login: Sun Dec 4 19:05:29 2022 from 172.25.250.9
[student@servera ~]$ sudo -i
[sudo] password for student:
[root@servera ~]# vim /etc/selinux/config
[root@servera ~]# systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[Roger Zhang@workstation ~] >ssh student@servera
Activate the web console with: systemctl enable --now cockpit.socket

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

Last login: Sun Dec 4 19:39:32 2022 from 172.25.250.9
[student@servera ~]$ sudo -i
[sudo] password for student:
[root@servera ~]# getenforce
Permissive
[root@servera ~]# vim /etc/selinux/config
[root@servera ~]# setenforce 1
[root@servera ~]# getenforce
Enforcing
[root@servera ~]# logout
[student@servera ~]$ logout
Connection to servera closed.
[Roger Zhang@workstation ~] >lab selinux-opsmode finish

Cleaning up the lab on servera:
· Cleaning servera: ..... SUCCESS

Lab finished.

[Roger Zhang@workstation ~] >
```

Note: In this exercise, I've done Use ACL entries to grant access to a group, and deny access to one of its members. Verify that the existing files and directories reflect the new ACL permissions. Set the default ACL on a directory, and confirm that new files and directories inherit its configuration.

Figure 10

Lab: Managing SELinux Security. You can see in the screenshot that all tests are passed

```

student@v

File Edit View Search Terminal Help
pg_helper_exec_t, gpg_pinentry_tmp_t, gpg_pinentry_tmpfs_t, gpm_tmp_t, gpsd_exec_t, groupadd
qlibtmp_t, httpd_cache_t, httpd_config_t, httpd_exec_t, httpd_helper_exec_t, httpd_keytab_t
p_tmp_t, httpd_rotatelogs_exec_t, httpd_squirrelmail_t, httpd_suexec_exec_t, httpd_suexec_tm
t, httpd_sys_script_exec_t, httpd_tmp_t, httpd_tmpfs_t, httpd_unconfined_script_exec_t, httpd_
, httpd_var_lib_t, httpd_var_run_t, hugetlbfs_t, hwclock_exec_t, hwloc_dhwd_exec_t, ibacm_lo
_log_t, inetd_tmp_t, init_tmp_t, initrc_tmp_t, initrc_var_log_t, innd_log_t, install_exec_t,
n_t, ipsec_log_t, ipsec_mgmt_exec_t, ipsec_tmp_t, iptables_exec_t, iptables_tmp_t, irc_exec_
jetty_tmp_t, jetty_unit_file_t, jetty_var_lib_t, jetty_var_run_t, jockey_exec_t, jockey_var
ui_exec_t, kdumpgui_tmp_t, keepalived_tmp_t, keepalived_unconfined_script_exec_t, keystone_c
keystone_cgi_script_exec_t, keystone_log_t, keystone_tmp_t, kismet_exec_t, kismet_log_t, kis
krb5_host_rcache_t, krb5_keytab_t, krb5kdc_conf_t, krb5kdc_log_t, krb5kdc_tmp_t, ksmtuned_lo
c
^C
[root@serverb ~]# http://serverb/lab.html^C
[root@serverb ~]# http://serverb/lab.htmlausearch -m AVC -ts recent^C
[root@serverb ~]# ausearch -m AVC -ts recent
-----
time->Sun Dec 4 19:46:35 2022
type=AVC msg=audit(1670201195.082:2406): avc: denied { getattr } for pid=9108 comm="httpd"
text=unconfined_u:object_r:default_t:s0 tclass=file permissive=0
-----
time->Sun Dec 4 19:46:35 2022
type=AVC msg=audit(1670201195.082:2407): avc: denied { setattr } for pid=9108 comm="httpd"
text=unconfined_u:object_r:default_t:s0 tclass=file permissive=0
[root@serverb ~]# ls -dZ /lab-content /var/www/html
    unconfined_u:object_r:default_t:s0 /lab-content  system_u:object_r:httpd_sys_content_t
[root@serverb ~]# semanage fcontext -a \
> -t httpd_sys_content_t '/lab-content(/.*)?'
[root@serverb ~]# restorecon -R /lab-content/
[root@serverb ~]# logout
[student@serverb ~]$ logout
Connection to serverb closed.
[Roger Zhang@workstation ~] >lab selinux-review grade

Grading the lab on serverb:
· Confirming files are in /lab-content: ..... PASS
· Confirming the Apache DocumentRoot is unchanged: ..... PASS
· Confirming the web content is accessible: ..... PASS

Overall lab grade..... PASS

[Roger Zhang@workstation ~] >lab selinux-review finish

Cleaning up the lab on serverb:
· Cleaning up firewall rules: ..... SUCCESS
· Removing httpd: ..... SUCCESS
· Removing permit access to /lab-content: ..... SUCCESS
· Removing /lab-content: ..... SUCCESS
Lab finished.

[Roger Zhang@workstation ~] >

```

Note: I have done: Identify issues in system log files. Adjust the SELinux configuration..

## Chapter Review

In this chapter, I learned: The getenforce and setenforce commands are used to manage the SELinux mode of a system. The semanage command is used to manage SELinux policy rules. The restorecon command applies the context defined by the policy. Booleans are switches that change the behavior of the SELinux policy. They can be enabled or disabled and are used to tune the policy. The sealert displays useful information to help with SELinux troubleshooting.

## Chapter 6

Figure 11

Guided Exercise: Adding Partitions, File Systems, and Persistent Mounts. You can see in the screenshot that all test are passed

```
File Edit View Search Terminal Help
End? 1001MB
(parted) quit
Information: You may need to update /etc/fstab.

[root@servera ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system  Flags
 1       1049kB  1001MB  1000MB  primary

[root@servera ~]# udevadm settle
[root@servera ~]# mkfs.xfs /dev/vdb1
meta-data=/dev/vdb1              isize=512    agcount=4, agsize=61056 blks
                                =           sectsz=512  attr=2, projid32bit=1
                                =           crc=1     finobt=1, sparse=1, rmapbt=0
data     =           reflink=1
          =           bsize=4096   blocks=244224, imaxpct=25
          =           sunit=0    swidth=0 blks
naming   =version 2             bsize=4096   ascii-ci=0, ftype=1
log      =internal log          bsize=4096   blocks=1566, version=2
          =           sectsz=512  sunit=0 blks, lazy-count=1
realtime =none                 extsz=4096   blocks=0, rtextents=0
[root@servera ~]# mkdir /archive
[root@servera ~]# lsblk --fs /dev/vdb
NAME   FSTYPE LABEL UUID                                     MOUNTPOINT
vdb
└─vdb1 xfs      6cab9389-fd79-4530-96f0-7b3f8544b61c
[root@servera ~]# ^C
[root@servera ~]# ^C
[root@servera ~]# vim /etc/fstab
[root@servera ~]# systemctl daemon-reload
[root@servera ~]# mount /archive
[root@servera ~]# mount | grep /archive
/dev/vdb1 on /archive type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
[root@servera ~]# logout
[student@servera ~]$ lab storage-partitions finish
-bash: lab: command not found
[student@servera ~]$ logout
Connection to servera closed.
[Roger Zhang@workstation ~] >lab storage-partitions finish

Cleaning up the lab on servera:
  - Cleaning up the second disk on servera..... SUCCESS
  - Restoring /etc/fstab on servera..... SUCCESS

Lab finished.

[Roger Zhang@workstation ~] >
```

Note: In this exercise, I've done use parted, mkfs.xfs, and other commands to create a partition on a new disk, format it, and persistently mount it..

Figure 12

Lab: Managing Basic Storage. You can see in the screenshot that all tests are passed

```

File Edit View Search Terminal Help
[root@serverb ~]# mkswap /dev/vdb2
Setting up swap space version 1, size = 489 MiB (512749568 bytes)
no label, UUID=52dd6256-9380-46ff-be38-271cb9e1e7e0
[root@serverb ~]# vim /etc/fstab
[root@serverb ~]# mkswap /dev/vdb3
Setting up swap space version 1, size = 488 MiB (511700992 bytes)
no label, UUID=f5793b8f-2cab-44dd-abed-d2331fd1b455
[root@serverb ~]# vim /etc/fstab
[root@serverb ~]# vim /etc/fstab
[root@serverb ~]# systemctl daemon-reload
[root@serverb ~]# swapon -a
[root@serverb ~]# swapon --show
NAME      TYPE   SIZE USED PRIOS
/dev/vdb2 partition 489M   0B   10
/dev/vdb3 partition 488M   0B   20
[root@serverb ~]# logout
[student@serverb ~]$ lab storage-review grade^C
[student@serverb ~]$ logout
Connection to serverb closed.
[Roger Zhang@workstation ~] >lab storage-review grade

Grading the student's work on serverb:

· vdb uses the GPT partitioning scheme..... PASS
· The XFS backup partition exists on vdb..... PASS
· The backup partition size is 2 GB (1.8 to 2.2 GB)..... PASS
· The XFS file system exists on the backup partition..... PASS
· The XFS file system is mounted at /backup..... PASS
· The XFS file system is defined in /etc/fstab..... PASS
· The swap1 partition exists on vdb..... PASS
· The swap1 partition size is 512 MB (460 to 564 MB)..... PASS
· The swap2 partition exists on vdb..... PASS
· The swap2 partition size is 512 MB (460 to 564 MB)..... PASS
· The swap1 partition is initialized as swap..... PASS
· The swap1 swap space is activated..... PASS
· The swap1 swap space is defined in /etc/fstab..... PASS
· The swap2 partition is initialized as swap..... PASS
· The swap2 swap space is activated..... PASS
· The swap2 swap space is defined in /etc/fstab..... PASS
· The swap2 swap space is preferred over swap1..... PASS

Overall lab grade..... PASS

[Roger Zhang@workstation ~] >lab storage-review finish

Completing the lab on serverb:

· Cleaning up the second disk on serverb..... SUCCESS
· Restoring /etc/fstab on serverb..... SUCCESS

Lab finished.

[Roger Zhang@workstation ~] >

```

Note: I have done: Display and create partitions using the parted command. Create new file systems on partitions and persistently mount them. Create swap spaces and activate them at boot.

### Chapter Review

In this chapter, I learned You can use the parted command to add, modify, and remove partitions on disks with the MBR or the GPT partitioning scheme. You use the mkfs.xfs command to create XFS file systems on disk partitions. You need to add file-system mount commands to /etc/fstab to make those mounts persistent. You use the mkswap command to initialize swap spaces.

## Chapter 7

Figure 13

Guided Exercise: Creating Logical Volumes. You can see in the screenshot that all test are passed

```
File Edit View Search Terminal Help
proc on /psoc type proc (rw,nosuid,nodev,noexec,relatime)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=907700k,nr_inodes=226925,mode=755)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,mode=755)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,seclabel,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel)
pstree on /sys/fs/pstree type pstree (rw,nosuid,nodev,noexec,relatime,seclabel)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel)
cgroup on /sys/fs/cgroup/net_cls/net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel)
configs on /sys/kernel/config type configs (rw,relatime)
/dev/vda3 on / type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autosys (rw,relatime,fd=27,pgrp=1,timeout=0,mqueue=0)
mqueue on /dev/mqueue type mqueue (rw,relatime,seclabel)
debugfs on /sys/kernel/debug type debugfs (rw,relatime,seclabel)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
/dev/vda2 on /boot/efi type vfat (rw,relatime,fmask=0077,dmask=0077,codepage=437,inode64)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=187056k)
/dev/mapper/servera_01_vg-servera_01_lv on /data type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
[root@servera ~]# df -h /data
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/servera_01_vg-servera_01_lv  395M   24M  372M   6% /data
[root@servera ~]# logout
[student@servera ~]$ logout
Connection to servera closed.
[Roger Zhang@workstation ~] >lab lvm-creating finish

Completing the exercise on servera:
· Restoring original /etc/fstab on servera..... SUCCESS
· Ensuring /data is not busy ..... SUCCESS
· Unmounting /data filesystem ..... SUCCESS
· Removing /data directory ..... SUCCESS
· Removing servera_01_lv logical volume ..... SUCCESS
· Removing servera_01_vg volume group ..... SUCCESS
· Removing physical volume /dev/vdb1 ..... SUCCESS
· Removing physical volume /dev/vdb2 ..... SUCCESS
· Cleaning up the second disk on servera ..... SUCCESS

[Roger Zhang@workstation ~] >
```

Note: In this exercise, I've done Create physical volumes, volume groups, and logical volumes with LVM tools. Create new file systems on logical volumes and persistently mount them.

Figure 14

Lab: Managing Logical Volumes. You can see in the screenshot that all tests are passed

```
Grading the student's work on serverb:  
  . serverb_01_lv is 768MiB in size ..... PASS  
  . serverb_02_lv is 128MiB in size ..... PASS  
  . serverb_02_lv is mounted at /storage/data2 ..... PASS  
  . serverb_02_lv is formatted with an XFS filesystem ..... PASS  
  . serverb_02_lv belongs to the serverb_01_vg volume group ..... PASS  
  . serverb_01_vg has the correct size ..... PASS  
  
overall lab grade..... PASS  
  
[Roger Zhang@workstation ~] >lab lvm-review finish  
  
Completing the lab on serverb:  
  . Unmounting /storage/data1 filesystem ..... SUCCESS  
  . Removing /storage/data1 directory ..... SUCCESS  
  . Unmounting /storage/data2 filesystem ..... SUCCESS  
  . Removing /storage/data2 directory ..... SUCCESS  
  . Restoring /etc/fstab on serverb..... SUCCESS  
  . Removing serverb_01_lv logical volume ..... SUCCESS  
  . Removing serverb_02_lv logical volume ..... SUCCESS  
  . Removing serverb_01_vg volume group ..... SUCCESS  
  . Removing physical volume /dev/vdb1 ..... SUCCESS  
  . Removing physical volume /dev/vdb2 ..... SUCCESS  
  . Cleaning up the second disk on serverb ..... SUCCESS  
  
[Roger Zhang@workstation ~] >
```

Note: I have done: Resize the serverb\_01\_lv logical volume to 768 MiB. Create a new 128 MiB logical volume called serverb\_02\_lv with an XFS file system, persistently mounted at /storage/data2.

### Chapter Review

In this chapter, I learned LVM allows you to create flexible storage by allocating space on multiple storage devices. Physical volumes, volume groups, and logical volumes are managed by a variety of tools such as pvcreate, vgreduce, and lvextend. Logical volumes can be formatted with a file system or swap space, and they can be mounted persistently. Additional storage can be added to volume groups and logical volumes can be extended dynamically.

## Chapter 8

Figure 15

Guided Exercise: Managing Layered Storage with Stratis. You can see in the screenshot that all test are passed

```

student@workstation:~>

File Edit View Search Terminal Help
[roger@servera ~]# stratis pool create stratispool1 /dev/vdb
[roger@servera ~]# stratis pool list
Name          Total Physical
stratispool1  5 GiB / 37.63 MiB / 4.96 GiB
[roger@servera ~]# stratis pool add-data stratispool1 /dev/vdc
[roger@servera ~]# stratis filesystem create stratispool1 stratis-filesystem1
[roger@servera ~]# mkdir /stratisvol
[roger@servera ~]# mount /stratis,stratispool1,stratis-filesystem1 /stratisvol
[roger@servera ~]# echo "Hello World!" > /stratisvol/file1
[roger@servera ~]# dd if=/dev/urandom of=/stratisvol/file2 bs=1M count=2048
stratis filesystem list
2048+0 records in
2048+0 records out
2147483648 bytes (2.1 GB, 2.0 GiB) copied, 12.0966 s, 178 MB/s
[roger@servera ~]# stratis filesystem list
Pool Name      Name           Used       Created      Device
        UUID
stratispool1   stratis-filesystem1  2.39 GiB Dec 04 2022 20:31  /stratis,strati
filesystem1    eff841ffbc1541a49be73fddb3209c3c
[roger@servera ~]# stratis filesystem snapshot stratispool1 stratis-filesystem1
file1-snap
[roger@servera ~]# rm /stratisvol/file1
rm: remove regular file '/stratisvol/file1'? y
[roger@servera ~]# mkdir /stratisvol-snap
[roger@servera ~]# mount /stratis,stratispool1,stratis-filesystem1-snap /stratisvol-snap
[roger@servera ~]# cat /stratisvol-snap/file1
Hello World!
[roger@servera ~]# umount /stratisvol-snap
[roger@servera ~]# umount /stratisvol
[roger@servera ~]# stratis filesystem destroy stratispool1 stratis-filesystem1-snap
[roger@servera ~]# stratis filesystem destroy stratispool1 stratis-filesystem1
[roger@servera ~]# logout
[student@servera ~]$ logout
Connection to servera closed.
[roger.zhang@workstation ~] >lab advstorage-stratis finish

Completing the lab on servera:
· Removing stratis-filesystem1-snap from on servera..... SUCCESS
· Removing stratis-filesystem1 from servera..... SUCCESS
· Removing stratispool1 from servera..... SUCCESS
· Removing /stratisvol from servera..... SUCCESS
· Removing /stratisvol-snap from servera..... SUCCESS
· Ensuring clean additional disks on servera..... SUCCESS

Lab finished.

[roger.zhang@workstation ~] >

```

Note: In this exercise, I've done Create a thin-provisioned file system using Stratis storage management solution. Verify that the Stratis volumes grow dynamically to support real-time data growth. Access data from the snapshot of a thin-provisioned file system.

Figure 16

Lab: Implementing Advanced Storage Features. You can see in the screenshot that all tests are passed

```
student@workstation:~  
File Edit View Search Terminal Help  
Roger Zhang@workstation ~] >ssh student@serverb  
sh: connect to host serverb port 22: Connection refused  
Roger Zhang@workstation ~] >ssh student@serverb  
sh: connect to host serverb port 22: Connection refused  
Roger Zhang@workstation ~] >ssh student@serverb  
[[A  
ctivate the web console with: systemctl enable --now cockpit.socket  
  
his system is not registered to Red Hat Insights. See https://cloud.redhat.com/  
to register this system, run: insights-client --register  
  
ast login: Sun Dec  4 20:34:32 2022 from 172.25.250.9  
[[A  
student@serverb ~]$ sudo -i  
sudo] password for student:  
root@serverb ~]# logout  
student@serverb ~]$ logout  
onnection to serverb closed.  
Roger Zhang@workstation ~] >lab advstorage-review grade  
  
grading the student's work on serverb:  
  
· Verifying the stratis file system on serverb..... PASS  
· Verifying the stratis snapshot on serverb..... PASS  
· Verifying the files in the stratis file system on serverb... PASS  
· Verifying the files in stratis snapshot on serverb..... PASS  
· Verifying the VDO volume on serverb..... PASS  
· Verifying the files in VDO volume on serverb..... PASS  
  
verall lab grade..... PASS  
  
Roger Zhang@workstation ~] >lab advstorage-review finish  
  
ompleting the lab on serverb:  
  
· Removing labfs-snap from on serverb..... SUCCESS  
· Removing labfs from serverb..... SUCCESS  
· Removing labpool from serverb..... SUCCESS  
· Removing /labstratisvol from serverb..... SUCCESS  
· Removing /labstratisvol-snap from serverb..... SUCCESS  
· Removing install.img from serverb..... SUCCESS  
· Removing labvdo volume from serverb..... SUCCESS  
· Ensuring clean additional disks on serverb..... SUCCESS  
· Restoring original /etc/fstab on serverb..... SUCCESS  
  
ab finished.  
Roger Zhang@workstation ~] >
```

Note: I have done: Create a thinly provisioned file system using Stratis storage management solution. Verify that the Stratis volumes grow dynamically to support real-time data growth. Access data from the snapshot of a thinly provisioned file system. Create a volume using Virtual Data Optimizer and mount it on a file system. Investigate the impact of data deduplication and compression on a Virtual Data Optimizer volume.

### Chapter Review

In this chapter, I learned The Stratis storage management solution implements flexible file systems that grow dynamically with data. The Stratis storage management solution supports thin provisioning, snapshotting, and monitoring. The Virtual Data Optimizer (VDO) aims to reduce the cost of data storage. The Virtual Data Optimizer applies zero-block elimination, data deduplication, and data compression to optimize disk space efficiency.

## Chapter 9

Figure 17

Guided Exercise: Managing Network-Attached Storage with NFS. You can see in the screenshot that all test are passed

```
student@workstation:~
```

```
File Edit View Search Terminal Help
· Configuring SELinux Contexts in serverb ..... SUCCESS
· Making /etc/nfs.conf backup in serverb ..... SUCCESS
· Configuring NFSv4 Server in serverb ..... SUCCESS
· Configuring Firewall in serverb ..... SUCCESS
· Making /etc/fstab backup in servera ..... SUCCESS

[Roger Zhang@workstation ~] >ssh student@servera
Activate the web console with: systemctl enable --now cockpit.socket

This system is not registered to Red Hat Insights. See https://cloud.redhat.com
To register this system, run: insights-client --register

Last login: Sun Dec  4 20:30:41 2022 from 172.25.250.9
[student@servera ~]$ sudo -i
[sudo] password for student:
[root@servera ~]# mkdir /public
[root@servera ~]# mount -t nfs serverb.lab.example.com:/shares/public /public
[root@servera ~]# umount /public
[root@servera ~]# vim /etc/fstab
[root@servera ~]# mount /public
[root@servera ~]# cat /public/NOTES.txt
###In this file you can log all your notes###
[root@servera ~]# echo "This is a test" > /public/Test.txt
[root@servera ~]# logout
[student@servera ~]$ logout
Connection to servera closed.
[Roger Zhang@workstation ~] >lab netstorage-nfs finish

Completing the lab in servera and serverb:
· Restoring /etc/fstab in servera:..... SUCCESS
· Rebooting servera:..... SUCCESS
· Clearing NFSv4 Server configuration in serverb ..... SUCCESS
· Removing shared directories and files in serverb ..... SUCCESS
· Restoring SELinux configuration in serverb ..... SUCCESS
· Restoring firewall in serverb ..... SUCCESS
· Deleting admin1 user on serverb ..... SUCCESS
· Deleting sysmanager1 user on serverb ..... SUCCESS
· Deleting admin group on serverb ..... SUCCESS
· Waiting for servera to become available ..... SUCCESS
· Removing directories in servera ..... SUCCESS
· Deleting admin1 user on servera ..... SUCCESS
· Deleting sysmanager1 user on servera ..... SUCCESS
· Deleting admin group on servera ..... SUCCESS

Lab finished.

[Roger Zhang@workstation ~] >
```

Note: In this exercise, I've done Test an NFS Server using the mount command. Configure NFS Shares in the /etc/fstab configuration file to save changes even after a system reboots.

Figure 18

Lab: Accessing Network-Attached Storage. You can see in the screenshot that all tests are passed

```
student@workstation:~  
File Edit View Search Terminal Help  
Grading the lab:  
  .  
  . Verifying autofs is installed/started/enabled in servera: ... PASS  
  . Verifying autofs files in servera: ..... PASS  
  . Verifying autofs master map configuration in servera: ... PASS  
  . Verifying autofs indirect map configuration in servera: ... PASS  
  . Verifying autofs operation in servera: ..... PASS  
overall lab grade..... PASS  
[Roger Zhang@workstation ~] >lab netstorage-review finish  
completing the lab in servera and serverb:  
  . Disabling autofs from boot in servera:..... SUCCESS  
  . Rebooting servera:..... SUCCESS  
  . Clearing NFSv4 Server configuration in serverb: ..... SUCCESS  
  . Removing shared directories and files in serverb: ..... SUCCESS  
  . Restoring SELinux configuration in serverb: ..... SUCCESS  
  . Restoring firewall in serverb: ..... SUCCESS  
  . Deleting manager1 user on serverb ..... SUCCESS  
  . Deleting manager2 user on serverb ..... SUCCESS  
  . Deleting dbuser1 user on serverb ..... SUCCESS  
  . Deleting sysadmin1 user on serverb ..... SUCCESS  
  . Deleting contractor1 user on serverb ..... SUCCESS  
  . Deleting consultant1 user on serverb ..... SUCCESS  
  . Deleting managers group on serverb ..... SUCCESS  
  . Deleting production group on serverb ..... SUCCESS  
  . Deleting operators group on serverb ..... SUCCESS  
  . Waiting for servera to become available: ..... SUCCESS  
  . Clearing NFSv4 Client configuration in servera: ..... SUCCESS  
  . Uninstalling packages in servera:..... SUCCESS  
  . Removing exercise files in servera: ..... SUCCESS  
  . Deleting manager1 user on servera ..... SUCCESS  
  . Deleting manager2 user on servera ..... SUCCESS  
  . Deleting dbuser1 user on servera ..... SUCCESS  
  . Deleting sysadmin1 user on servera ..... SUCCESS  
  . Deleting contractor1 user on servera ..... SUCCESS  
  . Deleting consultant1 user on servera ..... SUCCESS  
  . Deleting managers group on servera ..... SUCCESS  
  . Deleting production group on servera ..... SUCCESS  
  . Deleting operators group on servera ..... SUCCESS  
Lab finished.  
[Roger Zhang@workstation ~] >
```

Note: I have done: Install required packages needed to set up the automounter. Configure an automounter indirect map, getting resources from a preconfigured NFSv4 server.

### Chapter Review

In this chapter, I learned Mount and unmount an NFS export from the command line. Configure an NFS export to automatically mount at startup. Configure the automounter with direct and indirect maps, and describe their differences.

## Chapter 10

Figure 19

Guided Exercise: Selecting the Boot Target. You can see in the screenshot that all test are passed

```
Roger Zhang@workstation ~] >lab boot-selecting finish
cleaning up the lab on workstation:
  o
  - Restoring configuration files on workstation..... SUCCESS
  - Setting the default target to graphical on workstation..... SUCCESS

Lab finished.

Roger Zhang@workstation ~] >
```

Note: In this exercise, I've been able to update the system default target and use a temporary target from the boot loader.

Figure 20

Lab: Controlling the Boot Process. You can see in the screenshot that all tests are passed

```
student@workstation:~ 
File Edit View Search Terminal Help
[Roger Zhang@workstation ~] >boot-review grade
bash: boot-review: command not found...
[Roger Zhang@workstation ~] >lab boot-review grade

Grading the student's work on serverb:
  . root password is redhat..... PASS
  . The boot issue is fixed..... PASS
  . The default target is graphical..... PASS

Overall lab grade..... PASS

[Roger Zhang@workstation ~] >lab boot-review finish

Cleaning up the lab on serverb:
  . Resetting root password on serverb..... SUCCESS
  . Setting the default systemd target on serverb..... SUCCESS
  . Restoring configuration files on serverb..... SUCCESS
  . Setting SELinux in Enforcing mode on serverb..... SUCCESS

Lab finished.

[Roger Zhang@workstation ~] >
```

Note: I have done: Reset a lost root password. Diagnose and fix boot issues. Set the default systemd target.

### Chapter Review

In this chapter, I learned systemctl reboot and systemctl poweroff reboot and power down a system, respectively. systemctl isolate target-name.target switches to a new target at runtime. systemctl get-default and systemctl set-default can be used to query and set the default target. Use rd.break on the kernel command line to interrupt the boot process before control is handed over from the initramfs. The root file system is mounted read-only under /sysroot. The emergency target can be used to diagnose and fix file-system issues.

## Chapter 11

Figure 21

Guided Exercise: Managing Server Firewalls. You can see in the screenshot that all test are passed

```
student@servera: ~]$ sudo bash -c "echo 'I am servera.' > /var/www/html/index.html"
student@servera: ~]$ sudo systemctl enable --now httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
student@servera: ~]$ logout
Connection to servera closed.
Roger Zhang@workstation ~] >ssh student@servera
Activate the web console [with: systemctl enable --now cockpit.socket

his system is not registered to Red Hat Insights. See https://cloud.redhat.com/
o register this system, run: insights-client --register

ast login: Sun Dec  4 21:15:51 2022 from 172.25.250.9
student@servera: ~]$ sudo systemctl status nftables
sudo: password for student:
nftables.service - Netfilter Tables
  Loaded: loaded (/usr/lib/systemd/system/nftables.service; disabled; vendor preset: disabled)
  Active: inactive (dead)
    Docs: man:nft(8)
student@servera: ~]$ sudo systemctl mask nftables
Created symlink /etc/systemd/system/nftables.service → /dev/null.
student@servera: ~]$ curl -k https://serera.lab.example.com
url: (6) Could not resolve host: serera.lab.example.com
student@servera: ~]$ logout
Connection to servera closed.
Roger Zhang@workstation ~] >curl -k https://servera.lab.example.com
I am servera.
Roger Zhang@workstation ~] >lab netsecurity-firewalls finish
leaning up the lab on servera:

· Remove httpd from servera..... SUCCESS
· Reset firewall to defaults..... SUCCESS

ab finished.

Roger Zhang@workstation ~] >
```

Note: In this exercise, I've been able to configure firewall rules to control access to services.

Figure 22

Lab: Managing Network Security. You can see in the screenshot that all tests are passed

The screenshot shows a terminal window with the following content:

```
Activities Terminal ▾

File Edit View Search Terminal Help
sources:
services: cockpit dhcpcv6-client http ssh
ports:
protocols:
masquerade: no
forward-ports:
source-ports:
icmp-blocks:
rich rules:

[student@serverb ~]$ sudo firewall-cmd --permanent --zone=public --add-port=1001/tcp
success
[student@serverb ~]$ sudo firewall-cmd --reload
success
[student@serverb ~]$ sudo firewall-cmd --permanent --zone=public --list-all
public
    target: default
    icmp-block-inversion: no
    interfaces:
    sources:
    services: cockpit dhcpcv6-client http ssh
    ports: 1001/tcp
    protocols:
    masquerade: no
    forward-ports:
    source-ports:
    icmp-blocks:
    rich rules:

[student@serverb ~]$ logout
Connection to serverb closed.
[Roger Zhang@workstation ~] >lab netsecurity-review grade

Grading the student's work on serverb:
· Checking access to default web server..... PASS
· Checking access to virtual host..... PASS

Overall lab grade..... PASS

[Roger Zhang@workstation ~] >lab netsecurity-review finish

Cleaning up the lab on serverb:
· Remove httpd from serverb..... SUCCESS
· Remove virtual host document root from serverb..... SUCCESS
· Remove non-standard port from http_port_t type on serverb... SUCCESS
· Remove non-standard port from firewall on serverb..... SUCCESS
· Remove http service from firewall on serverb..... SUCCESS

Lab finished.
[Roger Zhang@workstation ~] >
```

Note: I have done: configure firewall and SELinux settings on a web server host.

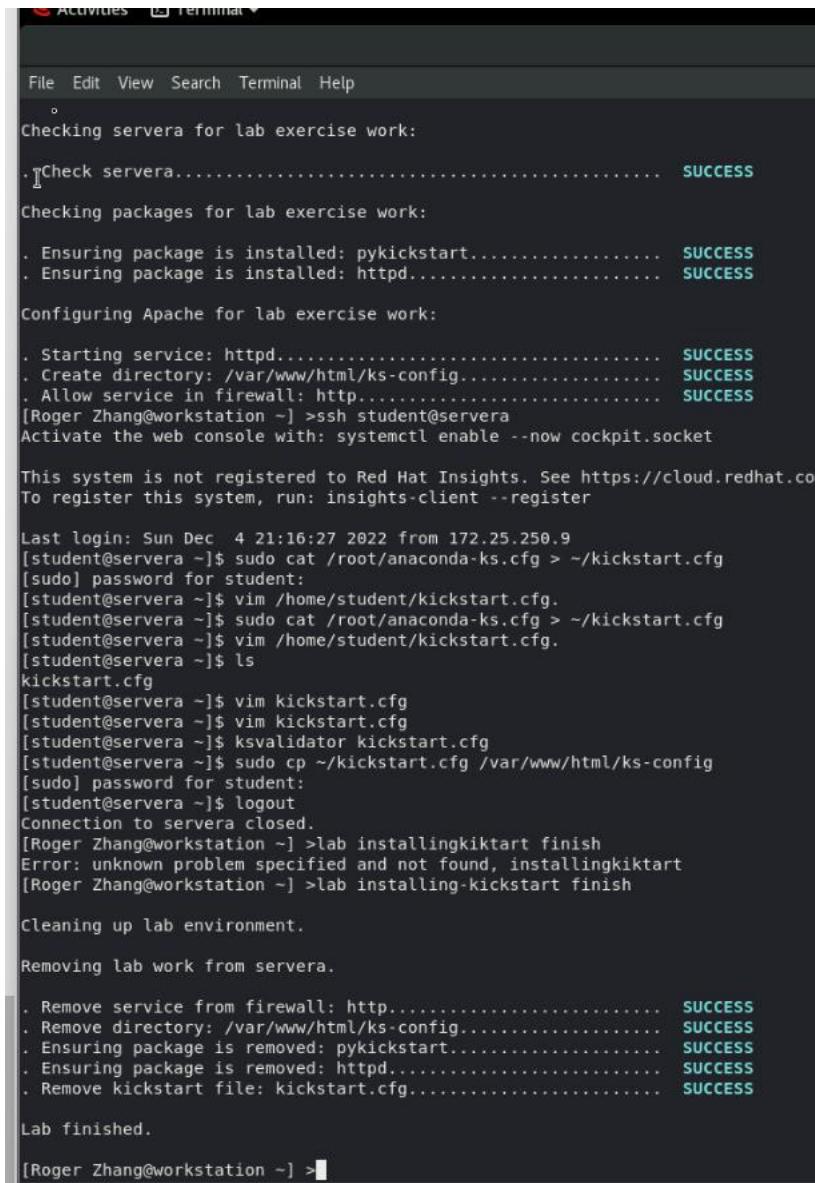
## Chapter Review

In this chapter, I learned The netfilter subsystem allows kernel modules to inspect every packet traversing the system. All incoming, outgoing or forwarded network packets are inspected. The use of firewalld has simplified management by classifying all network traffic into zones. Each zone has its own list of ports and services. The public zone is set as the default zone. The firewalld service ships with a number of pre-defined services. They can be listed using the firewall-cmd --get-services command. Network traffic is tightly controlled by the SELinux policy. Network ports are labeled. For example, port 22/TCP has the label ssh\_port\_t associated with it. When a process wants to listen on a port, SELinux checks to see whether the label associated with it is allowed to bind that port label. The semanage command is used to add, delete, and modify labels.

## Chapter 12

Figure23

Guided Exercise: Automating Installation with Kickstart. You can see in the screenshot that all test are passed



```
Activities Terminal

File Edit View Search Terminal Help

.
Checking servera for lab exercise work:
· Check servera..... SUCCESS
Checking packages for lab exercise work:
· Ensuring package is installed: pykickstart..... SUCCESS
· Ensuring package is installed: httpd..... SUCCESS
Configuring Apache for lab exercise work:
· Starting service: httpd..... SUCCESS
· Create directory: /var/www/html/ks-config..... SUCCESS
· Allow service in firewall: http..... SUCCESS
[Roger Zhang@workstation ~] >ssh student@servera
Activate the web console with: systemctl enable --now cockpit.socket

This system is not registered to Red Hat Insights. See https://cloud.redhat.com
To register this system, run: insights-client --register

Last login: Sun Dec 4 21:16:27 2022 from 172.25.250.9
[student@servera ~]$ sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg
[sudo] password for student:
[student@servera ~]$ vim /home/student/kickstart.cfg.
[student@servera ~]$ sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg
[student@servera ~]$ vim /home/student/kickstart.cfg.
[student@servera ~]$ ls
kickstart.cfg
[student@servera ~]$ vim kickstart.cfg
[student@servera ~]$ vim kickstart.cfg
[student@servera ~]$ ksvalidator kickstart.cfg
[student@servera ~]$ sudo cp ~/kickstart.cfg /var/www/html/ks-config
[sudo] password for student:
[student@servera ~]$ logout
Connection to servera closed.
[Roger Zhang@workstation ~] >lab installingkiktart finish
Error: unknown problem specified and not found, installingkiktart
[Roger Zhang@workstation ~] >lab installing-kickstart finish

Cleaning up lab environment.

Removing lab work from servera.

· Remove service from firewall: http..... SUCCESS
· Remove directory: /var/www/html/ks-config..... SUCCESS
· Ensuring package is removed: pykickstart..... SUCCESS
· Ensuring package is removed: httpd..... SUCCESS
· Remove kickstart file: kickstart.cfg..... SUCCESS

Lab finished.

[Roger Zhang@workstation ~] >
```

Note: In this exercise, I've been able to Create a kickstart file. Use ksvalidator to validate the kickstart file's syntax..

Figure 24

Lab: Installing Red Hat Enterprise Linux. You can see in the screenshot One section can't pass. Although the file is exactly the same as the lab suggests.

```
Activities Terminal ▾

File Edit View Search Terminal Help

overall lab grade..... FAIL

[Roger Zhang@workstation ~] >lab installing-review grade --help

Grading the student's work on serverb:

· Kickstart file available via http..... PASS
· reboot command commented..... PASS
· BaseOS repo command commented..... PASS
· AppStream repo command modified..... PASS
· url line modified..... PASS
· network command commented..... PASS
· rootpw command modified..... PASS
· authselect command added..... PASS
· services command modified..... PASS
· reqpart command commented..... PASS
· part command commented..... PASS
· autopart command added..... PASS
· Post installation section modified..... FAIL
· Packages section modified..... PASS

Overall lab grade..... FAIL

[Roger Zhang@workstation ~] >lab installing-review --help
This script controls the setup and grading of this lab.
Usage: lab installing-review COMMAND
      lab installing-review -h|--help
COMMAND is one of: start grade finish

[Roger Zhang@workstation ~] >lab installing-review grade

Grading the student's work on serverb:

· Kickstart file available via http..... PASS
· reboot command commented..... PASS
· BaseOS repo command commented..... PASS
· AppStream repo command modified..... PASS
· url line modified..... PASS
· network command commented..... PASS
· rootpw command modified..... PASS
· authselect command added..... PASS
· services command modified..... PASS
· reqpart command commented..... PASS
· part command commented..... PASS
· autopart command added..... PASS
· Post installation section modified..... FAIL
· Packages section modified..... PASS

Overall lab grade..... FAIL

[Roger Zhang@workstation ~] >
```

Note: I have done: Create a kickstart file. Make the kickstart file available to the installer. Perform a kickstart installation.

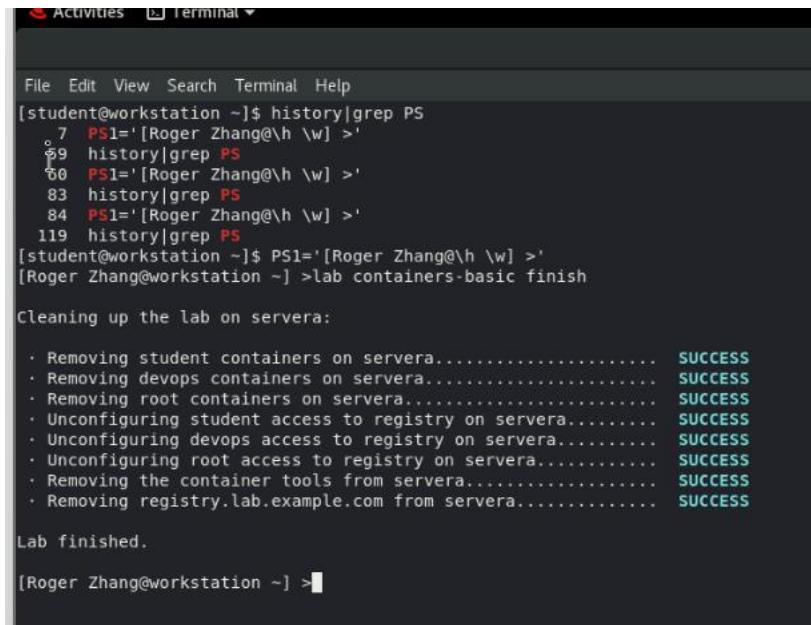
### Chapter Review

In this chapter, I learned The RHEL 8 binary DVD includes Anaconda and all repositories required for installation. The RHEL 8 boot ISO includes the Anaconda installer, accessing repositories over the network during installation. The Kickstart system performs unattended installations. Kickstart files can be created using the Kickstart Generator website or by copying and editing /root/anaconda-ks.cfg. The virt Yum module provides the packages for a RHEL system to become a virtualization host. The cockpit-machines package adds the Virtual Machines menu to Cockpit.

## Chapter 13

Figure25

Guided Exercise: Automating Installation with Kickstart. You can see in the screenshot that all test are passed



The screenshot shows a terminal window with the following content:

```
Activities Terminal ▾
File Edit View Search Terminal Help
[student@workstation ~]$ history|grep PS
 7 PS1='[Roger Zhang@\h \w] >'
 9 history|grep PS
 60 PS1='[Roger Zhang@\h \w] >'
 83 history|grep PS
 84 PS1='[Roger Zhang@\h \w] >'
119 history|grep PS
[student@workstation ~]$ PS1='[Roger Zhang@\h \w] >'
[Roger Zhang@workstation ~] >lab containers-basic finish

Cleaning up the lab on servera:
· Removing student containers on servera..... SUCCESS
· Removing devops containers on servera..... SUCCESS
· Removing root containers on servera..... SUCCESS
· Unconfiguring student access to registry on servera..... SUCCESS
· Unconfiguring devops access to registry on servera..... SUCCESS
· Unconfiguring root access to registry on servera..... SUCCESS
· Removing the container tools from servera..... SUCCESS
· Removing registry.lab.example.com from servera..... SUCCESS

Lab finished.
[Roger Zhang@workstation ~] >
```

Note: In this exercise, I've been able to install container management tools and use them to run a container.

Figure 26

Lab: Lab: Running Containers You can see in the screenshot that all test are passed



Red Hat

Activities Terminal ▾

```
File Edit View Search Terminal Help
Storing signatures
5bbd470ec4c555f03372552e5ca1dbba952392bc29e3b42bcc9bd2dd51d1efeb
[podsvc@serverb ~]$ ~/containers-review/testdb.sh
Testing the access to the database...
SUCCESS
[podsvc@serverb ~]$ mkdir -p ~/.config/systemd/user/
[podsvc@serverb ~]$ cd ~/.config/systemd/user/
[podsvc@serverb user]$ podman generate systemd --name inventorydb --files --n
/home/podsvc/.config/systemd/user/container-inventorydb.service
[podsvc@serverb user]$ podman stop inventorydb
5bbd470ec4c555f03372552e5ca1dbba952392bc29e3b42bcc9bd2dd51d1efeb
[podsvc@serverb user]$ podman rm inventorydb
5bbd470ec4c555f03372552e5ca1dbba952392bc29e3b42bcc9bd2dd51d1efeb
[podsvc@serverb user]$ systemctl --user daemon-reload
[podsvc@serverb user]$ systemctl --user enable --now container-inventorydb.se
Created symlink /home/podsvc/.config/systemd/user/multi-user.target.wants/con
Created symlink /home/podsvc/.config/systemd/user/default.target.wants/con
[podsvc@serverb user]$ ~/containers-review/testdb.sh
Testing the access to the database...
SUCCESS
[podsvc@serverb user]$ podman ps
CONTAINER ID IMAGE COMMAND CR
c7e476d80888 registry.lab.example.com/rhel8/mariadb-103:1-86 run-mysqld 10
[podsvc@serverb user]$ logindctl enable-linger
[podsvc@serverb user]$ logout
Connection to serverb closed.
[Roger Zhang@workstation ~] >lab containers-review grade

Grading the student's work on serverb:
· Required container packages exist..... PASS
· The inventorydb container exists..... PASS
· The container is using the correct image.... PASS
· The container is using the correct image tag..... PASS
· The container host port is 13306..... PASS
· The database user is operator1..... PASS
· The database password is redhat..... PASS
· The database name is inventory..... PASS
· The database root password is redhat..... PASS
· The database uses db_data for storage..... PASS
· The storage is mounted in /var/lib/mysql/data..... PASS
· SELinux context is set for db_data..... PASS
· The systemd unit file exists for the container..... PASS
· The systemd service for the container is enabled..... PASS
· The services for podsvc start at boot..... PASS

Overall lab grade..... PASS

[Roger Zhang@workstation ~] >lab containers-review finish

Cleaning up the lab on serverb:
|
```

Note: I have done: Create detached containers. Configure port redirection and persistent storage.

Configure systemd for containers to start when the host machine starts.

## Chapter Review

In this chapter, I learned Containers provide a lightweight way to distribute and run an application and its dependencies that may conflict with software installed on the host. Containers run from container images that you can download from a container registry or create yourself. Podman, provided by Red Hat Enterprise Linux, directly runs and manages containers and container images on a single host. Containers can be run as root, or as non-privileged rootless containers for increased security. You can map network ports on the container host to pass traffic to services running in its containers. You can also use environment variables to configure the software in containers.



# CERTIFICATE OF ATTENDANCE

ruojia zhang

NAME

George Washington University

SCHOOL/UNIVERSITY

Red Hat System Administration II (RH134)

COURSE

2022年12月5日

DATE

A handwritten signature in black ink that reads "Ken A".

KEN GOETZ

Vice president, Global Training Services at Red Hat



**Red Hat**  
Training and  
Certification