Pearson Content Week2 Lab guide

Day1:

* 1. Working with Logical Volume Manager

1.1 LVM creation

1.2 Extending/Expanding volume group and logical volume

* 1. Working on VDO in RHEL 8
  2. Working with Stratis in RHEL 8
  3. Checking GRUB2 config files and Modifying boot timer
  4. Resetting root password
  5. Troubleshooting common boot problems
  6. Working with Tuning Profile on server
  7. Changing nice values of processes

1.1 Working with Logical Volume Manger - LVM Creation

1.1.1 This Lab should be run on Server1 as root user

1.1.2 Login to Server1(192.168.4.3) system for this lab

**ssh** [**student@192.168.4.3**](mailto:student@192.168.4.3)

**su -l root**

**(Enter the password as “Passw0rd!” for root user)**

**hostname ; echo ; whoami**

Machine generated alternative text:



1.1.3 For LVM documentation and help on LVM commands try these commands

**pinfo lvm**

**pvcreate --help**

**vgcreate --help**

**lvcreate --help**

**vgextend --help**

**lvextend --help**

Task – Preparing the Disks for LVM creation on Disk vdb1 and vdc1

1.1.4 First Check if the Disk/partition to be used is empty and available, run this command to check on mount point, filesystem type if any exists on the disk's which you are going use for LVM

**lsblk -fp**

Machine generated alternative text:



For this Lab we will be using fdisk on disk's /dev/vdb and /dev/vdc to create two partitions 512MB

*Note: Partition disk using either fdisk, gdisk or parted - make sure disk type is set to LVM*

*In fdisk code for LVM type of disk is 8E*

*In gdisk code for LVM type of disk is 8E00*

**fdisk /dev/vdb**

**N**

**P**

**1**

<Press **Enter-Key** for first sector size>

**+512M**

To Change Partition type

**T**

**1**

**8E** (you may press L key to list partition numbers"

**P** (to print the partition table to confirm if the Partition is set to Type “Linux LVM”)

**W** (to write and quit)

To update the kernel or partition being created enter following command

**udevadm settle**

Machine generated alternative text:



1.1.5 **Repeat above steps on disk /dev/vdc to create a partition**

1.1.6 Run lsblk to check if the partition has been created

**lsblk**

Machine generated alternative text:



1.1.7 Use pvcreate to add the two new partitions created as Physical Volume (PVs)

**pvcreate /dev/vdb1 /dev/vdc1**

**pvs (Run “pvs” command to check on the PV’s created on system)**

**pvs**

Machine generated alternative text:



1.1.8 Use vgcreate to create a new VG named “server1\_lvm\_vg” on the two PV’s

**vgcreate server1\_lvm\_vg /dev/vdb1 /deb/vdc1**

**vgs (Run “vgs” command to check on the VG’s created on system)**

Machine generated alternative text:



*(Pro Tip – If you need to create a different size of PV [default 4MiB] you can do same during vg creation only command for same is as follow’s which creates a PE size of 16MiB*

***vgcreate -s 16 <vg\_name> <PV’s to use>)***

1.1.9 User lvcreate to create a 450MiB LV named “server1\_lvm\_lv1” from the VG “

**lvcreate -n server1\_lvm\_lv1 -L 450M server1\_lvm\_vg**

**lvs ( Run “lvs” command to check on the LV’s created on system)**

Machine generated alternative text:



1.1.10 Run following command to check on PE (Physical Extent) and LE (Logical Extent) created/allocated

**pvdisplay /dev/vdb1 /dev/vdc1**

Machine generated alternative text:



**vgdisplay server1\_lvm\_vg**

Machine generated alternative text:



**lvdisplay**

Machine generated alternative text:



1.1.11 Add a Filesystem xfs on created LV and Mount this logical volume persistently on /mnt/cloudproject

**mkfs.xfs /dev/server1\_lvm\_vg/server1\_lvm\_lv1**

**mkdir -p /mnt/cloudproject**

**mount /dev/server1\_lvm\_vg/server1\_lvm\_lv1 /mnt/cloudproject**

**df -Th**

**umount /mnt/cloudproject**

Machine generated alternative text:



1.1.12 Add following line to /etc/fstab to make the mount persistent

**/dev/server1\_lvm\_vg/server1\_lvm\_lv1 /mnt/cloudproject xfs defaults 0 0**

*(Pro-Tip: Create a backup of the /etc/fstab file in /tmp directory as a precaution)*

Machine generated alternative text:



1.2 Extending or Expanding Volume Group and logical volume

*Note: Backup data of the LV disk in production environment to avoid any data loss can be ignored in lab*

1.2.1. This lab should be run on Server1 as root user

**ssh** [**student@192.168.4.3**](mailto:student@192.168.4.3) **su -l root  
hostname ; echo ; whoami**

Machine generated alternative text:



1.2.2. Verify LVM mount and write sample file to it

**vgs ; echo ; lvs ; echo ; df -Th  
 echo “This is a sample file on LVM disk” >> /mnt/cloudproject/samplefile.txt  
 cat /mnt/cloudproject/samplefile.txt**

Machine generated alternative text:



1.2.3 Unmount the LVM disk mounted on “/mnt/cloudproject”

**umount /mnt/cloudproject  
 mount | grep cloudproject**

Machine generated alternative text:



1.2.4 Verify the Disk space using lsblk command and add new partition of size 512MiB using parted

(Note: You may use fdisk is you are comfortable using same)

**lsblk  
parted /dev/vdb print  
parted -s /dev/vdb mkpart primary 539MiB 1051MiB  
parted -s /dev/vdb set 2 lvm on  
udevadm settle**

Machine generated alternative text:



1.2.5 Add new disk to PV , extend VG server “server1\_lvm\_vg” and extend LV “server1\_lvm\_lv1” by 800MB

**pvs ; echo “--------” ; vgs ; echo “-------” ; lvs   
pvcreate /dev/vdb2  
vgextend server1\_lvm\_vg /dev/vdb2  
pvs ; echo “--------” ; vgs  
lvextend -L +800M /dev/ server1\_lvm\_vg/ server1\_lvm\_lv1  
lvs**

Machine generated alternative text:



***Note***

*The example specifies the exact size to make the final LV, but you could have*

*specified the amount of additional space desired:*

***• -L +800M*** *to add the new space using size in MiB.*

***• -l 312*** *to specify the total number of extents (312 PEs x 4 MiB).*

***• -l +200*** *to add the additional extents needed.*

1.2.6 Mount the disk and grow the filesystem

**mount /dev/server1\_lvm\_vg/server1\_lvm\_lv1 /mnt/cloudproject  
df -Th | grep cloudproject  
xfs\_growfs /mnt/cloudproject  
df -Th | grep cloudproject  
ls /mnt/cloudproject ; echo “ ---- “ ; cat /mnt/cloudproject/samplefile.txt**

Machine generated alternative text:



*Pro-Tip: If using EXT4 filesytem then command to use for extend would be*

*resize2fs /dev/<vgname>/<lvname>*

*Check the disk status to check LVM in action*

***lsblk***

Machine generated alternative text:



 \*\*\* Lab Working with Logical Volume Manager Completed \*\*\*

* 1. Working on VDO in RHEL8

2.1 This lab should be run on Server1 as root user

**ssh** [**student@192.168.4.3**](mailto:student@192.168.4.3) **su -l root  
hostname ; echo ; whoami**

Machine generated alternative text:



For help on vdo look at the help file

**pinfo vdo   
man vdo** ( Look at the example section in manual for sample command)

2.2 Verify VDO software is installed on the system and start the service

**yum list installed vdo   
rpm -qi vdo ; rpm -qa kmod-kvdo  
systemctl is-enabled vdo  
systemctl is-active vdo**

Machine generated alternative text:



*Note : If the package is not getting installed then create this repo file in /etc/yum.repos.d*

*vim /etc/yum.repos.d/os.repo ----- enter following text*

*[dvd\_baseos]*

*name=Oracle Linux 8 DVD BaseOS Repo*

*baseurl=http://192.168.4.1/DVD/BaseOS/*

*enabled=1*

*gpgcheck=0*

*[dvd\_appstream]*

*name=Oracle Linux 8 DVD AppStream Repo*

*baseurl=http://192.168.4.1/DVD/AppStream/*

*enabled=1*

*gpgcheck=0*

**Important Note for Lab**

Shutdown the Server1 VM, in the VM manager Create An Additional Disk of 5GB and attach to the VM before proceeding to next steps

2.3 Create the “vdo1” volume using the “vdo create” command and list using “vdo list” and check if Deduplication + Compression is enabled for the volume

**vdo create --name=vdo1 - -device=/dev/vde --vdoLogicalSize=10G**

**vdo list**

**vdo status --name=vdo1 | grep -E 'Deduplication|Compression'**

Machine generated alternative text:



2.4 Create a Filesystem XFS on the vdo disk and mount at /mnt/vdo1

**mkdir /mnt/vdo1  
mkfs.xfs -K /dev/mapper/vdo1  
mount /dev/mapper/vdo1 /mnt/vdo1  
mount | grep vdo1**

Machine generated alternative text:



2.5 Check VDO disk usage statistics and copy a 500MB file multiple time to check status

**vdostats --human-readable  
dd if=/dev/urandom of=500MBfile bs=1M count=500  
ll -h  
cp -v 500MBfile /mnt/vdo1/1stfile  
vdostats -human-readable  
cp -v 500MBfile /mnt/vdo1/2ndfile  
vdostats --human-readable**

Notice that we tough we have copied two 500MB files the disk is not growing and also Saving% has increased

You can test this copying the same 500MB file 3 more times and check disk status

Machine generated alternative text:



 \*\*\* Lab Working with VDO in RHEL 8 Completed \*\*\*

* 1. **Working with Stratis in RHEL8**

**Note: Before starting the lab – reset the lab snapshot to start of class**

3.1 This lab should be run on Server1 as root user

**ssh** [**student@192.168.4.3**](mailto:student@192.168.4.3)

**su -l root**

**hostname ; echo ; whoami**

Machine generated alternative text:



For help on stratis look at the help file

**man stratis ; pinfo stratis**

3.2 Install the stratis stratis-cli packages , enable the Stratisd daemon and start the service

**yum install stratisd stratis-cli**

**systemctl enable - -now startisd**

**systemctl status stratisd**

Machine generated alternative text:



*Note : If the package is not getting installed then create this repo file in /etc/yum.repos.d , move other repo files which are present in this folder*

*vim /etc/yum.repos.d/os.repo ----- enter following text*

*[dvd\_baseos]*

*name=Oracle Linux 8 DVD BaseOS Repo*

*baseurl=http://192.168.4.1/DVD/BaseOS/*

*enabled=1*

*gpgcheck=0*

*[dvd\_appstream]*

*name=Oracle Linux 8 DVD AppStream Repo*

*baseurl=http://192.168.4.1/DVD/AppStream/*

*enabled=1*

*gpgcheck=0*

3.3 Create a stratis pool on /dev/vdb with name “stratispool1” and list the pool

**stratis pool create stratispool1 /dev/vdb**

**stratis pool list**

Machine generated alternative text:



3.4 Expand the capacity of the “stratispool1” to use disk /dev/vdc and verify the new pool size

**stratis pool add-data stratispool1 /dev/vdc**

**stratis pool list**

Machine generated alternative text:



3.5 List the block devices backing the stratispool1

**stratis blockdev list stratispool1**

Machine generated alternative text:



3.6 Add a Thin-Provisioned filesystem “stratis-fs1” on “stratispool1” and mount at “/mnt/stratisvol”

**stratis filesystem create stratispool1 stratis-fs1**

**stratis filesystem list**

**mkdir -p /mnt/stratis-vol**

**mount /stratis/stratispool/stratis-fs1 /mnt/stratis-vol**

**mount | grep stratis-vol**

**Notice that stratis-fs1 is formatted with xfs filesystem**

Machine generated alternative text:



3.7 Verify that thin volume grows when new data is added

**stratis filesystem list**

**echo “This is a Stratis Fileystem – Thin Volume” >> /mnt/stratis-vol/samplefile.txt**

**dd if=/dev/urandom of=/mnt/stratis-vol/1GbFile bs=1M count=1024**

**stratis filesystem list**

Machine generated alternative text:



3.8 Creating a Snapshot on the stratis-fs1 with name “snap-of-fs1”, deleting a file from source and restoring from snap

**stratis filesystem snapshot stratispool1 stratis-fs1 snap-of-fs1**

**stratis filesystem list**

**rm /mnt/stratis-vol/samplefile.txt**

**mkdir -p /mnt/snap-stratis**

**mount /stratis/stratispool1/snap-of-fs1 /mnt/snap-stratis**

**cat /mnt/snap-stratis/samplefile.txt**

Machine generated alternative text:



3.9 Destroy the Stratis-fs1 , snap-of-fs1 and Remove the Stratis pool stratispool1

**umount /mnt/stratis-vol**

**umount /mnt/snap-stratis**

**stratis filesystem destroy stratispool1 stratis-fs1**

**stratis filesystem destroy stratispool1 snap-of-fs1**

**stratis filesystem list**

**stratis pool destroy stratispool1**

**stratis pool list**

Machine generated alternative text:



\*\*\*\* Lab Working with Stratis in RHEL 8 Completed \*\*\*\*

* 1. Checking GRUB2 Config file and modifyingDo boot timer

4.1 This Lab should be run on Server1 as root user

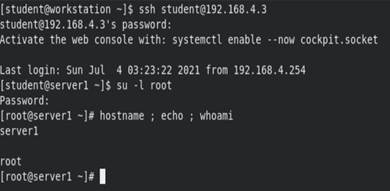
Login to Server1(192.168.4.3) system for this lab

**ssh** [**student@192.168.4.3**](mailto:student@192.168.4.3)

**su -l root**

**(Enter the password as “Passw0rd!” for root user)**

**hostname ; echo ; whoami**



4.2 . Launch the console of Server1 from Virtual Machine Manager (Keep this open on the background we would need to see this)

serverl on QEMU/KVM 
File 
Virtual Machine 
View Send Key 
Oracle Linux Sera—er 8.3 
Kernel 4.18.Ø-zqe.e18.xß6_64 on an x86_64 
the console with: systerætl enable 
seB.er1 login: 
- -nou cockpit. socket 

4.3 Run these commands on the ssh console taken on workstation system

cat /etc/default/grub

Edit the GRUB\_TIMEOUT value to 10 seconds and remove the rhgb and Quite options on GRUB\_CMDLINE\_LINUX

vim /etc/default/grub

cat /etc/default/grub | grep -E "TIMEOUT|CMDLINE"

grub2-mkconfig > /boot/grub2/grub.cfg

reboot

(root@serverl cat /etc/defautt/grub 
GRUB 
GRUB 
GRUB 
GRUB 
GRUB 
GRUB 
GRUB 
GRUB 
TIMEOUT=20 
•s, release . •S, ' 
DEFAULT-saved 
DISABLE SUBMENU=true 
TERMINAL OUTPUT-econsoteø 
/etc/system- release) " 
CMDLINE rd.lvm. rd. rhgb qui 
DISABLE RECOVERY-"true" 
ENABLE BLSCFG=true 
(root@serverl vim /etc/defautt/grub 
(root@serverl cat /etc/defautt/grub I grep -E "TIMEOUTICMDLINE" 
GRUB 
TIMEOUT=IØ 
CMDLINE rd.lvm. rd. Ivm.FFol_server1/swap " 
GRUB 
[ root@serverl grub2•mkconfig > /b00t/grub2/grub. cfg 
Generating grub configuration file . 
done 
[root@serverl reboot 
Connection to 192.168.4.3 closed by remote host. 
Connection to 192 .168.4.3 closed. 
1 
(student@workstation —IS 

Check the console of server1 taken from Virtual Machine Manager to verify on the boot-timer has changed and boot messages displayed - if need restart again if you missed it.

* 1. **Resetting Root password (Chapter 18 - Page 416 - Revision 2 Book )**

This lab needs to be run on console of Server2 - Take console of server2 from Virtual Machine Manager

5.1 On system boot, press **E** when the GRUB 2 boot menu is shown. (Press Spacebar key to stop the timer)

File Virtual Machine View Send Key 
Oracle Linux Server 
Oracle Linux Server 
Oracle Linux Server 
Oracle Linux Server 
serverz on QEMU/KVM 
d (5.4.17-Z011.7.4.e18uek.x86 64) 
(4.18.O-Z40.10.1.e18_3 . x86 _64) 8.3 
(4.18.o-zqo.e18.x86_64) 8.3 
(o-rescue-4a6a563Z7Z5Z417db8eeeb8bdd6Z69c5) 8.3 
Use the t and keys to change the selection. 
Press 'e' to edit the selected item, or 'c' for a command prompt. 

5.2. Enter **rd.break console=tty1** as a boot argument to the line that loads the kernel and press **Ctrl-X** to boot with this option.

load video 
et gfx_pay -keep 
i nsmod Zio 
I inux ( root )/vml inuz—5 .4.17—2011.7.4.e18uek -x86 _ 64 
I—root ro crashkernel —auto serverl—sua rd . Ivm. 
rl/root rd. Ivm. rhgb quiet rd . break console=ttyl 
in i trd ($root in i tramf s—5.4 .17—2011. ? .4.e18ue . x 
n i trd 
mg 
une 
Press Ctrl—x to start, Ctrl—c for a command prompt or Escape to 
d i scar e 
s and return to the menu. Pressing Tab I ists 
possible completions . 

5.3. You’ll now be dropped at the end of the boot stage where initramfs is loaded, just before a mount of the root file system on the directory /. Run this command to check sysroot mount as Read-only (RO)

**mount | grep sysroot**

5.4. Type **mount -o remount,rw /sysroot** to get read/write access to the system image.

**mount | grep sysroot**

5.5 At this point, make the contents of the /sysimage directory your new root directory by typing

**chroot /sysroot**

5.6. Now you can enter the new password for the user root using following command

**echo "Passw0rd!" | passwd root --stdin**

5.7. Because at this very early boot stage SELinux has not been activated yet, the context type on /etc/shadow will be messed up. If you reboot at this point, no one will be able to log in. So you must make sure that the context type is set correctly.

**touch /.autorelabel**

**exit**

**exi**t

Exit I to 
You might •ant to saw 
after it 
: grep 
on xf (ro. 
swi tch : grep 
on 
sh: rot 
tCh_r-OOt : 
sh-4 echo I ——Stdin 
for user 
: a ion 
sh-4.4t trn•ch .autorelabel 
exit 
suitch_root exit 
to d tßB stick or 
lat ir .attr2 ru•quota 

5.8 Upon reboot enter the New password for root user , system might take a minute or two to bootup

\*\*\* Lab Resetting Root password Complete "

* 1. Troubleshooting common boot problems

6.1 Exploring Troubleshooting Targets

This lab needs to be run on console of Server2 - Take console of server2 from Virtual Machine Manager

6.1.1. (Re)start your computer Server2. When the GRUB menu shows, select the first line in the menu and **press e**.

File Virtual Machine View Send Key 
Oracle Linux Server 
Oracle Linux Server 
Oracle Linux Server 
Oracle Linux Server 
serverz on QEMU/KVM 
d (5.4.17-Z011.7.4.e18uek.x86 64) 
(4.18.O-Z40.10.1.e18_3 . x86 _64) 8.3 
(4.18.o-zqo.e18.x86_64) 8.3 
(o-rescue-4a6a563Z7Z5Z417db8eeeb8bdd6Z69c5) 8.3 
Use the t and keys to change the selection. 
Press 'e' to edit the selected item, or 'c' for a command prompt. 

6.1.2. Scroll down to the line that starts with linux $(root)/vmlinuz. At the end of this line, type

**systemd.unit=rescue.target console=tty1**

Also remove the options **rhgb quite** from this line. Press Ctrl-X to boot with these modifications.

load video 
set gfx_pay load—keep 
i nsmod z io 
linux ( root)zvml 
I—root ro crashkernel -auto resume=/devzma erzol serverl—sua rd . Ivm. Iv=ol ser\ 
verlzroot rd. Ivm. systemd .unit=rescue . target console—ttyl 
i ni trd ($root ) / in i tramf s—5.4 .17—201 
Press Ctrl—x to start, Ctr for a command prompt or Escape to 
discar e s and return to the menu. Pressing Tab I ists 
possible complet ions . 

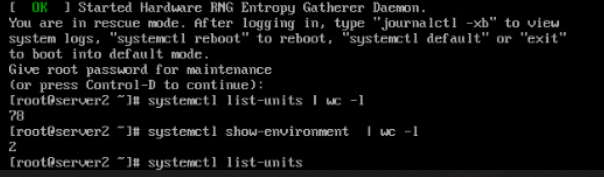
6.1.3. Enter the root password when you are prompted for it.

6.1.4. Type **systemctl list-units | wc -l** << This shows all unit files that are currently loaded number of lines .

6.1.5. Type **systemctl show-environment | wc -l** << This shows current shell environment variables number of lines.

Type **systemctl list-units** << This will list the unit files loaded make note of few

6.1.6. Type **systemctl reboot** to reboot your machine.



6.1.7. When the GRUB menu appears, **press e** again to enter the editor mode. At the

end of the line that loads the kernel, type **systemd.unit=emergency.target console=tty1**

Press **Ctrl-X** to boot with this option.

load_video 
set gfx_payload -keep 
insmod gzio 
I inux ($root)/vml inuz—5 .4 .17—2011.7.4 . e18uek .x86_64 
I—root ro crashkernel=auto rd. Ivm. 
verl/root rd . Ivm. systemd .unit=emergency. target console=ttyl 
initrd ($root ? .4.e18uek . x86 _ 64 . i mg $ tuned_initrd 
Press Ctrl—x to start, Ctrl—c for a command prompt or Escape to 
discard edits and return to the menu. Pressing Tab lists 
possible complet ions . 

6.1.8. When prompted for it, enter the root password to log in.

6.1.9. After successful login, type **systemctl list-units | wc -l** . Notice that the number of unit files loaded is reduced to a bare minimum.

susten logs. "sustentl reboot" to reboot, "sustentl default" or 
to b«jt into default mde. 
(or press Control-D to cont it"") : 
: f i for i 
troot9semærZ sust—tl list-units -I 
"exit" 

6.2 Repairing File System issues at boot

This lab needs to be run on console of Server2 - Take console of server2 from Virtual Machine Manager

6.2.1 Login as root user, Add this entry to the end of /etc/fstab << to simulate boot error

**cp -v /etc/fstab /tmp/fstab.bak**

**echo "/dev/vdxx /REMOVE-ME REMOVE-ME" >> /etc/fstab**

**tail /etc/fstab**

**systemctl reboot**

Oracle Sener .3 
•eb console with: syste«tl enable ctrkpit.sKEket 
set".er2 1 (»in: root 
Pas : 
Last Jul 
I rootesem.-er•Z 
• 't.•fstab.bak• 
trootPscner2 tail -I 'ctc'fstab 
REnn.Æ-æ 

6.2.2 When the GRUB menu appears, **press e** to enter the editor mode. At the

end of the line that loads the kernel, type **systemd.unit=emergency.target console=tty1**

Press **Ctrl-X** to boot with this option.

load_video 
set gfx_payload -keep 
insmod gzio 
I inux ($root)/vml inuz—5 .4 .17—2011.7.4 . e18uek .x86_64 
I—root ro crashkernel=auto rd. Ivm. 
verl/root rd . Ivm. systemd .unit=emergency. target console=ttyl 
initrd ($root ? .4.e18uek . x86 _ 64 . i mg $ tuned_initrd 
Press Ctrl—x to start, Ctrl—c for a command prompt or Escape to 
discard edits and return to the menu. Pressing Tab lists 
possible complet ions . 

6.2.3. When prompted for it, enter the root password to log in.

**mount | grep ro**

**mount -o remout,rw /**

**mount | grep root**

- grep r00 ~ 
丨 ~ ~ 8 ㄧ , f t ¢ r ¢ t at , ' supsnrts tiest-.—s until (Bx?fffffff) 
一 川 t grep 'Ot 
丨 「 ootOscr , x: 「 ~ 

**vim /etc/fstab**

Comment the line "/dev/vdxx /REMOVE-me …."

**tail -1 /etc/fstab**

**systemctl reboot**

Created bu on Tue Feb 9 
See •n pages fstab(5). r iMfs(B). at-Wor blkid(B) •re info. 
After this run 
units this f 
' -root 
—sump 
def au I ts 
def au I ts 
def I 

6.2.4 System would boot into the OS - login as root user to confirm on work

\*\*\*\* This concludes the lab on troubleshooting filesystem issue on boot \*\*\*\*

* 1. Working with Tuning Profile on server

7.1 This Lab should be run on Server1 as root user

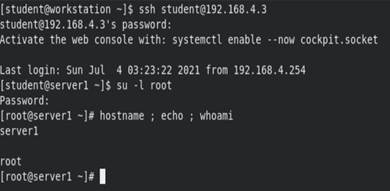
Login to Server1(192.168.4.3) system for this lab

**ssh** [**student@192.168.4.3**](mailto:student@192.168.4.3)

**su -l root**

**(Enter the password as “Passw0rd!” for root user)**

**hostname ; echo ; whoami**



For info on Tuned run

**man tuned-adm**

**tuned-adm --help**

7.2 Check if Tuned is installed on server (It probably already installed.)

**yum -y install tuned**

**yum info installed tuned**

7.3 Type systemctl status tuned to check whether tuned currently is running. If it is not, use systemctl enable --now tuned.

**systemctl status tuned.service**

**systemctl enable --now tuned.service**

7.4 To see which profile currently is used, type

**tuned-adm active**

7.5 To see which tuned profile is recommended, type

**tuned-adm recommend**

7.6. To select and activate the throughput-performance profile, type

**tuned-adm profile throughput-performance**.

**tuned-adm active**

**tuned-adm recommend**

[root@serverl yum -Y 
install tuned 
Last metadata expiration 
Check: ago on sun 04 Jul 2021 AM CDT. 
Package tuned-2. 14 .0-3.0.1.e18 3.1.noarch is already installed. 
Dependencies resolved . 
tuned-adm active 
tuned-adm active 
Nothing to do. 
Complete! 
[ root@serverl 
active 
[ root@serverl 
enabled 
( root@serverl 
Current active 
[ root@serverl 
vi rtual-guest 
[ root@serverl 
( root@serverl 
Current active 
[ root@serverl 
virtual-guest 
[root@serverl - 
systemctl is-active tuned .service 
systemctl is-enabled tuned. service 
profile: 
vi rtual-guest 
tuned-adm recommend 
tuned-adm profile throughput-performance 
profile: 
th roughput- perfo rmance 
tuned-adm recommend 

* 1. Changing nice values of processes

8.1 This Lab should be run on Server1 as root user

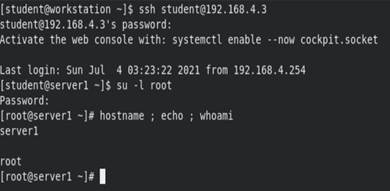
Login to Server1(192.168.4.3) system for this lab

**ssh** [**student@192.168.4.3**](mailto:student@192.168.4.3)

**su -l root**

**(Enter the password as “Passw0rd!” for root user)**

**hostname ; echo ; whoami**



8.2 Start a few background process which start using CPU resources

Note: Don't run these commands on a production server as they shoot-up CPU Usage

**dd if=/dev/urandom of=/dev/null &**

**md5sum /dev/urandom &**

**cat /dev/urandom >> /dev/null &**

**jobs**

**ps aux | grep -E "md5sum|cat|dd"**

Notice the 3rd Column which shows the CPU usage for the Processes

Machine generated alternative text:
root 
root 
dbus 
file 
root 
root 
root 
root 
[root@serverl 
dd if=/dev/urandom of=/dev/null 
[1] 1960 
[root@serverl 
md5sum /dev/urandom & 
[2] 1963 
[root@serverl 
cat /dev/urandom /dev/null & 
[3] 1968 
[root@serverl 
[1] 
Running 
Running 
[314 Running 
[ root@serverl 
11:11 
11:11 
11:11 
14:28 
1:45 
14:28 
1:32 
14:28 
1:18 
14:30 
2 
172 
872 
j obs 
ps aux I 
0.0 0.0 
0.0 0.0 
0.0 0.3 
dd if=/dev/urandom of=/dev/null & 
md5sum /dev/urandom & 
cat /dev/urandom >> /dev/null & 
grep -E 
"md5sum I cat I dd" 
o 
o 
85020 
6144 
- -sys log -only 
7324 
904 
18692 
1800 
7420 
2008 
12108 
1100 
o: 
o. 
o. 
o. 
00 
pts/O 
pts/O 
pts/O 
pts/O 
s 
SSI 
R 
R 
R 
R+ 
•00 
- -systemd- activation 
1960 71.8 
1963 68.8 
1968 65.5 
2000 0.0 
0.0 
0.1 
0.1 
0.0 
[kthreaddl 
[ipv6_add rconf] 
(us r/bin/dbus - daemon 
--system - -address= 
dd 
if=/dev/urandom of=/dev/null 
rnd5sum /dev/urandom 
/dev/urandom 
cat 
- -color=auto -E md5surnlcatldd 
g rep 
1 
[root@serverl 

8.3 To start a process with a higher nice value - which would assign better CPU resources to this new process

**nice -n -5 md5sum /dev/urandom&**

**ps aux | grep md5sum**

Notice that new process has a better CPU allocation according to 3rd column

Machine generated alternative text:
root 
root 
root 
[root@serverl nice -n 
-5 md5sum /dev/urandom 
0.1 
4:54 
0.1 
14:35 
0:10 
0.0 
14:35 
[4] 2005 
[root@serverl ps aux I 
grep md5sum 
1963 66.1 
2005 97.6 
2007 0.0 
18692 
18692 
12108 
1800 
1756 
976 
pts,'O 
pts,'O 
pts/0 
14. 
• 28 
o. 
md5sum 
md5sum 
grep - 
/dev/urandom 
/dev/urandom 
-color—auto 
md5sum 
1 
[root@serverl 

8.4 To change the nice value of a existing process - we need to renice the process ID

**pgrep md5sum**

**renice -n 0 2005**

**ps aux | grep md5sum**

Machine generated alternative text:
[root@serverl renice -n 0 2005 
2005 (process ID) old priority -5, new priority 
14 : 28 
5 44 
14 : 35 
2:35 
14 : 38 
[root@serverl ps aux 
grep md5sum 
root 
root 
root 
1963 57.8 0.1 
2005 96.7 0.1 
2011 0.0 0.0 
18692 
18692 
12108 
1800 ptszo 
1756 ptszo 
944 pts/O 
R 
R 
o. 
•00 
md5sum 
/dev/urandom 
md5sum 
/dev/urandom 
- -color=auto 
md5sum 
g rep 

**8.5 To change the nice values for all the Processes of md5sum**

**renice -n -5 $(pgrep md5sum)**

**ps aux | grep md5sum**

Machine generated alternative text:
[root@serverl 
1963 (process 
2005 (process 
[root@serverl 
0.1 
14:28 
7:34 
0.1 
14 : 35 
4:25 
0.0 
14:42 
renice -n -5 $(pgrep md5sum) 
ID) old priority 6, new priority 
ID) old priority 6, new priority 
-5 
-5 
ps aux 
55.8 
70.1 
0.0 
I grep md5sum 
root 
root 
root 
1963 
2005 
2026 
18692 
18692 
12168 
1800 pts,'0 
1756 pts,'0 
996 ptszo 
o. 
md5sum 
md5sum 
grep - 
/dev/urandom 
/dev/urandom 
-color—auto 
md5sum 

Pro-Tip - You may use the top command to check the running process and change nice value of process from there as well

\*\*\*\* This Concludes the Day 1 Lab Activities \*\*\*\*

Day2

* 1. BASH Shell Scripts

1.1 Write a SHELL Script to accept name as input on running it and display "Hi <name entered> , Welcome to RHCSA Training" along with display

current date new line

1.2 Write a SHELL Script to accept a number as argument and run for loop to display numbers from 1 to argument entered

1.3 Write a SHELL Script to accept a number and run a For loop to display 1 to the entered number

* 1. Task Scheduling

2.1 Create cron job for user tux to run at every 1 hour to append file /tmp/users.txt with number of users logged into server

2.2 Working with Timer Units and Temporary files

* 1. SELinux

3.1 Verifying and Modifying SELinux Mode

3.2 Setting a Context Label on a Nondefault Apache Document-Root

3.3 Checking SELinux Booleans and working with Violations on SELinux and Troubleshooting

* 1. BASH SHELL Scripts

This Lab should be run on Server1 as root user

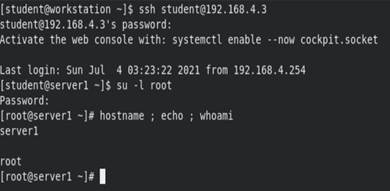
Login to Server1(192.168.4.3) system for this lab

**ssh** [**student@192.168.4.3**](mailto:student@192.168.4.3)

**su -l root**

**(Enter the password as “Passw0rd!” for root user)**

**hostname ; echo ; whoami**



1.1 Write a SHELL Script to accept name as input on running it and display "Hi <name entered> , Welcome to RHCSA Training" along with display

current date new line

1.1.1 Create SHELL Script file with filename "script1.sh" and enter following script text into it

vim script1.sh

#!/bin/bash

TODAY=$(date +%d-%m-%y)

if [ -z $1 ]; then

Echo Enter Your Name

read NAME

else

NAME=$1

fi

echo "Hi $NAME, Welcome to RHCSA Training"

echo

echo Today is $Today

exit 0

1.1.2 Change the permission for script1.sh to be executable by everyone

chmod +x script1.sh

[root@serverl vim scriptl.sh 
[root@serverl cat scriptl.sh 
/bin/bash 
TODAY-$(date 
if ( -z $1 1; then 
echo Enter Your Name 
read name 
else 
fi 
echo "HI $name, Welcome 
echo 
echo Today is 
$TODAY 
exit O 
( root@serverl 
1 root root 
-rw-r--r--. 
[ root@serverl 
chmod 
[ root@serverl 
s: cannot access : 
[root@serverl Is -t 
-rwxr-xr-x. 1 root root 
1 
[ root@serverl 
RHCSA Training" 
scriptl.sh 
178 Jul 7 07:15 scr!ptl.sh 
•x scriptl.sh 
scre 
NO such file or directory 
scriptl.sh 
178 7 07:15 scriptl.sh 

1.1.3 Execute the script using any of the following options

./script1.sh OmLokesh

Or

./script1.sh

Or

source script1.sh

[root@serverl ./scriptl .sh 
Enter Your Name 
OmL0kesh 
HI OmLokesh, Welcome to RHCSA Training 
Today is 07-07-21 
[root@serverl ./scriptl .sh OMLOKESH 
HI OMLOKESH, Welcome to RHCSA Training 
Today is 07-07-21 
(root@serverl source scriptl.sh 
Enter Your Name 
OMLOKESH 
HI OMLOKESH, welcome to RHCSA Training 
Today is 07-07-21 
(student@serverl —IS 

1.2 Write a SHELL Script to accept few arguments and run for loop to display the arguments entered

1.2.1 Create a SHEEL Script with filename "script2.sh" and enter following script into it

vim script2.sh

#!/bin/bash

# run this script with a few arguments

echo you have entered $# arguments

for i in $@

do

echo $i

done

exit 0

1.2.2 Change the permission for script1.sh to be executable by everyone

chmod +x script2.sh

1.2.3 Execute the script using any of the following option

./script2.sh One Two Three

(root@serverl vim 
Iroot@serverl cat 
#! /bin/bash 
echo You have entered 
for i in 
do 
echo $i 
done 
exit O 
script2. sh 
script2. sh 
$$ arugments 
root@serverl chmod script2. Sh 
I root@serverl Is •I script2. sh 
-rwxr-xr-x. 1 root root 83 Jut 7 07:32 script2.sh 
Iroot@serverl ./script2 .sh one two three 
You have entered 3 arugments 
one 
two 
th ree 
root@serverl 
1 

1.3 Write a SHELL Script to accept a number and run a For loop to display 1 to the entered number

1.3.1 Create a Script file with file name script3.sh and enter the following script into it

vim script3.sh

#!/bin/bash

echo Enter a Number less than 10

read number

for (( i=1 ; i <= $number ; i++ )); do

echo Number now is $i

done

exit 0

1.3.2 Change the permission for script1.sh to be executable by everyone

chmod +x script3.sh

1.3.3 Execute the script using any of the following option and given any number less than 10 as input when prompted

./script3.sh

[root@serverl vim script3.sh 
[root@serverl cat script3. sh 
# ! /bin/bash 
echo Enter a Number less then 
10 
read number 
for (( $number; i++)); 
do 
echo Number now is Si 
done 
exit O 
[root@serverl chmod ex script3.sh 
[root@serverl Is -I script3.sh 
-rwxr-xr-x. 1 root root 133 Jul 7 07:59 script3.sh 
[root@serverl sh 
Enter a Number less then 10 
3 
Number now is 
Number now is 
Number now is 
[ root@serverl 
2 
3 

* 1. Task Scheduling

This Lab should be run on Server1 as root user

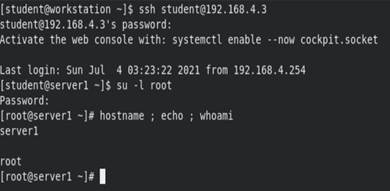
Login to Server1(192.168.4.3) system for this lab

**ssh** [**student@192.168.4.3**](mailto:student@192.168.4.3)

**su -l root**

**(Enter the password as “Passw0rd!” for root user)**

**hostname ; echo ; whoami**



Help command for cron would be as follows

**>> pinfo cron**

**>> crontab --help**

**>> cat /etc/crontab**

2.1 Create cron job for user tux to run at every 1 hour to append file /tmp/users.txt with number of users logged into server

2.1.1 Check status of the crond daemon

**systemctl status crond -l**

[ root@serverl systemctl status crond.service -I 
• crond . service 
- Command Scheduler 
03:01. 
03:01. 
03:01. 
03:29. 
03:29. 
04:01. 
05 :01. 
• active (running) since Sun 2021-07-04 
main PID: 1138 (crond) 
: 1 (limit: 10617) 
: 2.7M 
CGroup: /system. slice/crond. service 
L—1138 /usr/sbin/crond -n 
Loaded : 
Active . 
Tasks 
Memory 
loaded (/usr/lib/systemd/system/crond . service; enabled; vendor preset: enabled) 
CDT; 3 days ago 
Jul 
Jul 
Jul 
Jul 
Jul 
Jul 
08 
08 
08 
08 
08 
08 
08 
08 
08 
08 
03:29:01 
06:01:01 
serverl 
serve rl 
serve rl 
serverl 
serverl 
serverl 
serverl 
serve rl 
serve rl 
serverl 
anacron [46461: Anacron started on 2021-07-08 
anacron [4646]: Witt run job in 28 min. 
anacron 146461: Jobs wilt be executed sequentially 
anacron (46461: Job • cron . daily • started 
run-parts(46671: 
(/etc/cron . daily) finished logrotate 
anacron [46461: Job cron .daily• terminated 
anacron [46461: Normat exit (1 job run) 
CROND14674J: ( root) CMD (run-parts /etc/cron . hourly) 
CROND[46981: (root) CMD (run-parts /etc/cron .hourly) 
CROND[47211 : 
(root) CMD (run-parts /etc/cron .hourty) 
1 
[root@serverl 

To list the crontab format

**cat /etc/crontab**

[root@serverl tmpJ# cat /etc/crontab 
SHELL=/bin/bash 
PATH=/sbin : 'bin : /usr/sbin : /usr/bin 
MAILTO=root 
# For details see man 4 crontabs 
# Example of job definition: 
# . - minute (O - 59) 
- hour (O 
- 23) 
- day of month (1 - 
31) 
- month (1 - 
12) 
OR jan, feb,mar, apr . 
- day of week (e - 
6) (Sundare or 7) OR , thu, fri, sat 
user- name command 
to be executed 

2.1.2 Check if user "tux" is present, if not then create one

**id tux ; echo ; getent passwd tux**

**useradd tux**

**id tux ; echo ; getent passwd tux**

[ root@serverl 
id: •tux': 
no 
( root@serverl 
[ root@serverl 
uidz1001(tux) 
id tux; echo ; 
such user 
useradd tux 
id tux; echo ; 
tux) 
getent passwd tux 
getent passwd tux 
tux: x: : 1001: : /home/tux : /bin/bash 
( root@serverl 

2.1.3 Create a script file "usercount.sh" on /tmp directory to find-out number of users logged in and write to file /tmp/users.txt and Make the script file executable

**vim /tmp/usercount.sh**

#!/bin/bash

USERCOUNT=$(w -h | wc -l)

TODAY=$(date "+Date: %D Time: %T"

echo "There are ${USERCOUNT} Users logged into system at ${TODAY} …." >> /tmp/users.txt

echo " " >> /tmp/users.txt

**cd /tmp ; chmod a+rx usercount.sh ; ll usercount.sh**

Machine generated alternative text:
[root@serverl 
[root@serverl 
#! /bin/bash 
vim /tmp/usercount. sh 
cat /tmp/usercount. sh 
date "*Date: Time: 
echo "Thier are ${USERCOUNT} users logged into system at ${TODAY} 
echo " 
• /tmp/users. txt 
• /tmp/users. txt 
[root@serverl 
[root@serverl 
-rwxr-xr-x. 1 
[root@serverl 
chmod +x /tmp/usercount.sh 
/tmp/usercount.sh 
root root 196 Jul 9 02:10 /tmp/usercount. sh 
cd /tmp 

Test the script by running it once

**cd /tmp ; ./usercount.sh ; cat users.txt**

Machine generated alternative text:
[root@serverl tmpJ# ./usercount .sh 
Thier are 1 users logged into system 
[root@serverl tmpJ# ./usercount .sh 
Thier are 1 users logged into system 
Thier are 1 users logged into system 
cat users. txt 
at Date: 07/69/21 
cat users. txt 
at Date: 
at Date: 
07/09/21 
07/09/21 
Time: 
Time : 
Time : 

2.1.4 Create a cron job for user "tux" to execute this script

To list cron jobs of any user use command

**crontab -l -u tux**

Machine generated alternative text:
[root@serverl tmpJ# crontab -l -u tux 
no crontab for tux 
1 
[root@serverl tmpJ# 

Create a cron job for user tux use command and mention the text given to execute the script which created

*NOTE: We have created the cron job to run every minute to test the Job being run*

**crontab -e -u tux**

**0-59 \* \* \* \* /tmp/usercount.sh**

**crontab -l -u tux**

Machine generated alternative text:
[root@serverl crontab -e -u tux 
crontab: installing new crontab 
[root@serverl crontab -I -u tux 
0-59 * * * * /tmp/usercount.sh 
1 
[root@serverl 

2.1.5 Check if the cron job has been run by check the /var/log/cron and display the /tmp/users.txt

**tail /var/log/cron ; cat /tmp/users.txt**

*Note: Change the permissions for /tmp/users.txt so all users can write to it using command* ***chmod a+rwx /tmp/users.txt***

Note: Few Cron jobs will be stored at /var/spool/cron - location with user name as the filename

Machine generated alternative text:
Jul 
02:43. 
Jul 
Jul 
02 : 45. 
Jul 
Thier 
Thier 
Thier 
Thier 
Thier 
•42:01 
Thier 
•43:01 
Thier 
•44: 01 
Thier 
•45 : 02 
Thier 
•46:01 
9 
9 
9 
9 
02. 
02 . 
•44. 
1 
1 
1 
1 
1 
1 
1 
1 
1 
•01 
•01 
•02 
01 
serverl 
serverl 
serverl 
serverl 
CROND[5841] : 
CROND[5866] : 
CROND[5892] : 
CROND[5917] : 
(tux) 
(tux) 
(tux) 
(tux) 
CMD 
CMD 
CMD 
CMD 
(/tmp/usercount . sh) 
(/tmp/usercount . sh) 
(/tmp/usercount . sh) 
(/tmp/usercount . sh) 
a re 
a re 
a re 
a re 
a re 
a re 
are 
are 
are 
users 
users 
users 
users 
users 
users 
users 
users 
users 
logged 
logged 
logged 
logged 
logged 
logged 
logged 
logged 
logged 
into 
into 
into 
into 
into 
into 
into 
into 
into 
system 
system 
system 
system 
system 
system 
system 
system 
system 
at 
at 
at 
at 
at 
at 
at 
at 
at 
Date : 
Date : 
Date : 
Date : 
Date : 
Date: 
Date: 
Date: 
Date: 
07/09/21 
07/09/21 
07/09/21 
07/09/21 
07/09/21 
07/09/21 
07/09/21 
07/09/21 
07/09/21 
Time : 
Time : 
Time : 
Time : 
Time : 
Time : 
Time : 
Time : 
Time: 
02. 
02. 
02. 
02. 
02. 

2.2 Working with Timer Units and Temporary files

2.2.1 Install the sysstat application , this application manages the systemd timer unit called sysstat-collect.timer

**man 5 systemd.timer**

**yum install sysstat**

**cat /usr/lib/systemd/system/sysstat-collect.timer | grep OnCalender**

This timer unit collects system statistics every 10 minutes

[root@serverl rpm -qa I grep SYSStat 
sysstat-11.7.3-5.et8.x86 64 
(root@serverl cat /usr/lib/systemd/system/sysstat-cotlect . timer I grep OnCalendar 
On Ca enda : 00/ 
1 
[ root@serverl 

Note: If you need to modify the sysstat to collect statistics at different interval then you need to copy timer unit to **/etc/systemd/system**

Location and modify the ONCalender entry and reload the timer units using

**systemctl daemon-reload ; systemctl enable --now sysstat-collect.timer**

2.2.2 Temporary files are managed by tool systemd-tmpfiles which provides an easy way to manage temporary directors and files

**man 8 systemd-tmpfiles**

**cat /etc/tmpfiles.d/tmp.conf**

**cat /usr/lib/tmpfiles.d/tmp.conf**

To check the status of the Temporary Files Timer run this command

**systemctl cat systemd-tmpfiles-clean.timer**

2.2.3 Modify the /tmp directory clean up to 5 days

Copy the tmp.conf file from /usr to /etc

**cp -v /usr/lib/tmpfiles.d/tmp.conf /etc/tmpfiles.d/tmp.conf**

**Vim /etc/tmpfiles.d/tmp.conf <<<< Edit following line to reflect 5 days save and quit file**

**q /tmp 1777 root root 5d**

1777 

**systemd-tmpfiles --clean /etc/tmpfiles.d/tmp.conf**

root@serverl — # man 8 syste 
- trnp 
( root@serverl cp -v /usr/lib/tmpfiles.d/tmp.conf /etc/tmpfiles.d/tmp. conf 
conf' /etc/tmpfites.d/tmp. conf • 
root server 1 — vim etc t files. d t 
conf 
• •clean /etc/tmpfites.d/tmp.conf 
root@serverl -Je systemd•tmptiles 
[rootaserverl systemctl daemon-reload 

* 1. SELinux

This Lab should be run on Server1 as root user

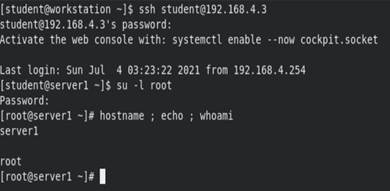
Login to Server1(192.168.4.3) system for this lab

**ssh** [**student@192.168.4.3**](mailto:student@192.168.4.3)

**su -l root**

**(Enter the password as “Passw0rd!” for root user)**

**hostname ; echo ; whoami**



3.1 Verifying and Modifying SELinux Mode

Documentation on SELinux can be found at

**man selinux\_config**

**man setenforce**

3.1.1 Verify the SELinux mode and change it temporarily

**getenforce**

**setenforce permissive**

**getenforce**

[ root@serverl 
-]# 
Enforcing 
[ root@serverl 
[ root@serverl 
-]# 
Permissive 
[root@serverl - 
getenforce 
setenforce permissive 
getenforce 

3.1.2 Modify SELinux mode to be permissive - persistent across boot of system

**vim /etc/sysconfig/selinux**

**systemctl reboot**

After reboot - connect back to server1 and run to check on same

**ssh student@192.168.4.3**

**su -l root**

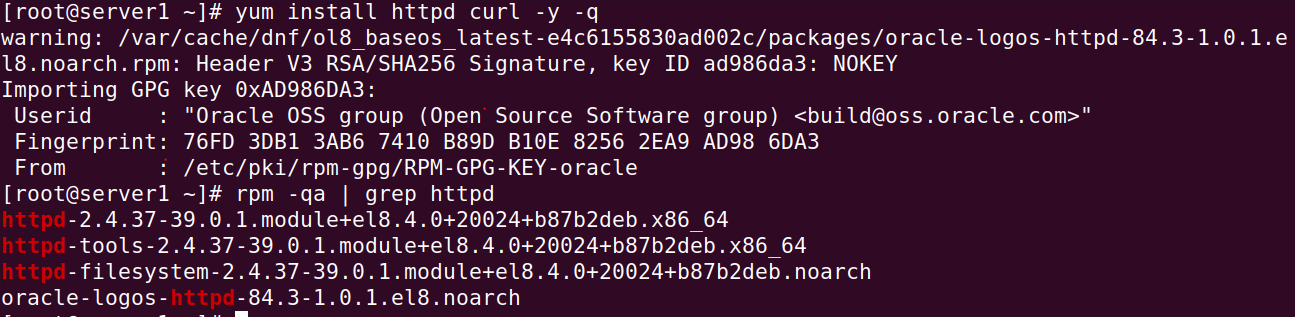
**getenforce ; echo ----- ; sestatus -v**

[root@serverl vim /etc/sysconfig/selinux 
(root@serverl cat /etc/sysconfig/selinux 
# This file controls the state of SELinux on the system. 
# SELINUX= can take one of these three values: 
enforcing 
SELinux security policy is enforced. 
permissive • SELinux prints warnings instead Of enforcing. 
disabled - No SELinux policy is loaded. 
SELINUX=pe rmissive 
# SELINUXTYPE= can take one of these three values: 
targeted - Targeted processes are protected, 
minimum - Modification Of targeted policy. Only selected processes are protected. 
mts - Multi Level Security protection. 
SELINUXTYPE=targeted 
root@serverl systemctl reboot 
Connection to 192 .168.4.3 closed by remote host. 
Connection to 192. 168.4.3 closed. 
(student&/orkstation -1$ ssh student@192.168.4.3 
student@192.168.4.3's password: 
Activate the web console with: systemctl enable - -now cockpit. socket 
Last login: wed Jut 7 2021 from 192.168.4.254 
(student@serverl —1$ su -t 
Password : 
I root@serverl 

( root@serverl 
permissive 
f root@serverl — 
getenforce 

3.2 Setting a SELINUX Context Label on a Nondefault Apache Document-Root on Server1

3.2.1. Open a root shell and type **yum install httpd curl -y**



3.2.2. Still from the root shell, type **mkdir /web**.

3.2.3. Type **vim /web/index.html** and put the following contents in the file: "Welcome to my WebServer - OM"

Machine generated alternative text:
[ root@serverl 
mkdir: created 
[ root@serverl 
[ root@serverl 
-]# mkdir -pv /web 
directory '/web' 
-]# echo "Welcome To My WebServer - 
-]# cat /web/index. html 
/web/index. html 
Welcome To My WebServer - 
0m 

3.2.4. Type **vim /etc/httpd/conf/httpd.conf** to open the Apache configuration file and find the DocumentRoot parameter. Change it so that it reads

**DocumentRoot "/web".**

3.2.5. In the same httpd.conf configuration file, add the following section, as without this section it will be Apache and not SELinux blocking access to the new DocumentRoot:

**<Directory "/web">**

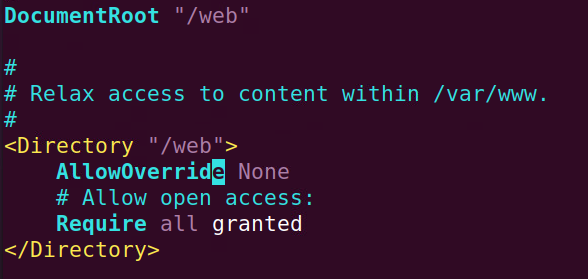
**AllowOverride None**

**Require all granted**

**</Directory>**

Machine generated alternative text:





3.2.6. Type **systemctl enable --now httpd** to start and enable the httpd service.

Note that if the httpd service was already running, you’ll need to use **systemctl restart httpd** to restart it so that it can pick up the changes you’ve made to the

httpd.conf configuration file.

3.2.7. Type **curl** [**http://localhost**](http://localhost) You’ll see the default Red Hat web page and not the contents of the index.html file you have just created.

3.2.8. Type **setenforce 0 ; getenforce** to switch SELinux to permissive mode.

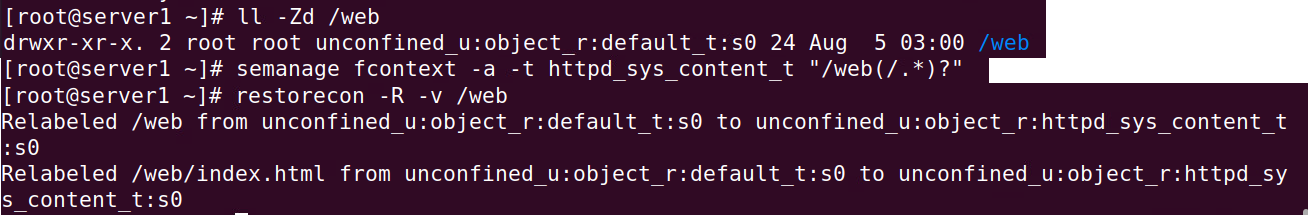
[ root@serverl 
Enforcing 
[ root@se rverl 
Permissive 
-]# 
getenforce 
setenforce 0 , 
getenforce 

3.2.9. Repeat step 7. You’ll now get access to your custom web page, which proves that SELinux was doing something to block access.

3.2.10. To apply the new context label to /web. Type **semanage fcontext -a -t httpd\_sys\_content\_t "/web(/.\*)?"**

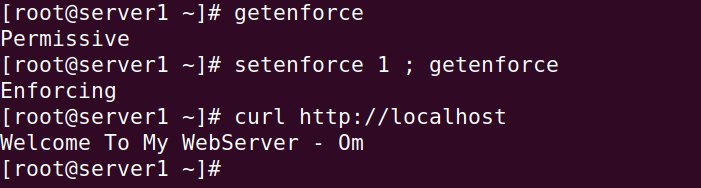
3.2.11. Type **restorecon -R -v /web**.

The -v (verbose) option ensures that you see what is happening and that you will see the new context being applied to /web.



3.2.12. Set SELinux back in enforcing mode, using **setenforce 1; getenforce**.

3.2.13. Type **curl** [**http://localhost**](http://localhost). You’ll get access to your custom web page because SELinux now allows access to it.



3.3 Checking SELinux Booleans and working with Violations on SELinux and Troubleshooting

3.3.1. From a root shell, type **getsebool -a | grep ftp**. You’ll see the ftpd\_anon\_write Boolean, with its current value set to off.

3.3.2. Type **setsebool ftpd\_anon\_write on**. This changes the value in the runtime.

3.3.3. Type **getsebool ftpd\_anon\_write**. It shows the value of the Boolean as on.

3.3.4. Type **semanage boolean -l | grep ftpd\_anon**. Notice that this command shows the runtime configuration set to on, but the permanent setting is still

set to off.

3.3.5. Use **setsebool -P ftpd\_anon\_write** on to switch the runtime and the default setting for the Boolean to on.

3.3.6. Repeat **semanage boolean -l | grep ftpd\_anon**. Notice that it is now set to on, on.

[root@serverl 
ftDd anon 
write - 
[root@serverl 
write 
ttpa anon 
[root@serverl 
[root@serverl 
[root@serverl 
write 
[root@serverl 
getsebool -a I grep 
off 
semanage boolean -1 
(off 
setsebool ftpd_anon 
ftpd anon 
I grep ftpd anon 
off) Allow ftpd to anon write 
write on 
setsebool -P ftpd anon write on 
semanage boolean -l 
(on 
grep ftpd anon 
Allow ftpd to anon write 

Checking SELINUX violations (Note: Make sure that previous lab activity on SELINUX file context has been completed before proceeding)

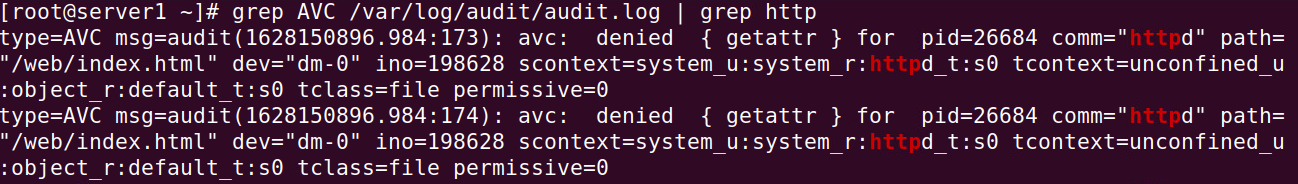
3.3.7 Type **ausearch -m AVC -ts recent**

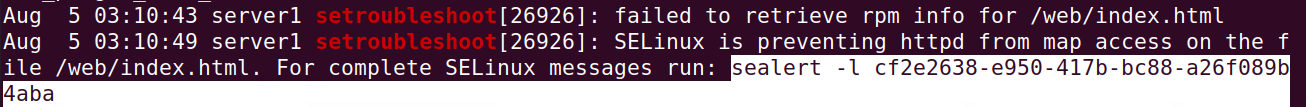
3.3.8 Type **grep AVC /var/log/audit/audit.log | grep http**

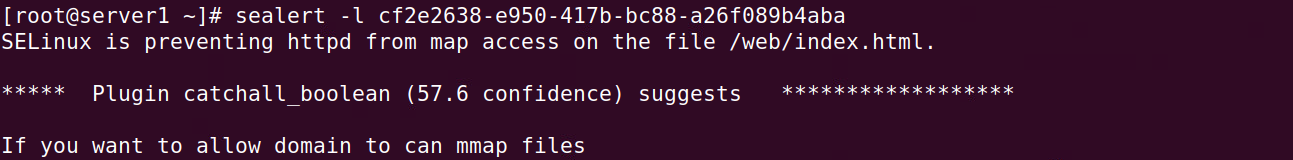
3.3.9 Type **cat /var/log/messages | grep setroubleshoot**

**Make note of the sealert -l <alert-id> and type same at the prompt for more information**

3.3.10 Type **sealert -l cf2e2638-e950-417b-bc88-a26f089b4aba**







\*\*\*\* Day 2 Lab Activities Completed \*\*\*

Day 3

1. Network Security

1.1 Managing the firewall with firewall-cmd.

1.2 Add http service to firewall zone permanently using command line

1.3 Checking firewall config from Cockpit

1.4 Connecting to a Remote Server with Public/Private Keys

1.5 Using SFTP and rsync to transfer file securely

1. File Access Control

2.1 Verify, change and Modify the attribute of file and directory

2.2 Managing Advanced Permissions using ACL's

1. Network Security

This Lab should be run on Server1 as root user

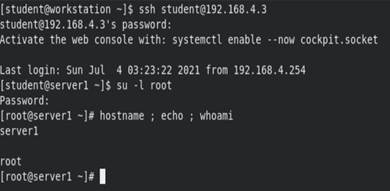
Login to Server1(192.168.4.3) system for this lab

**ssh** [**student@192.168.4.3**](mailto:student@192.168.4.3)

**su -l root**

**(Enter the password as “Passw0rd!” for root user)**

**hostname ; echo ; whoami**



1.1 Managing the firewall with firewall-cmd.

1.1.1 Type **rpm -qa | grep firewall** , to check if the firewalld is installed

1.1.2 Type **systemctl status firewalld.service** to check if Firewalld service is active

Or

Type **systemctl is-active firewalld.service**

[root@serverl rpm -qa I grep firewall 
f IrewaLL 
d-O.8.2-2.O.1.e18.noarch 
firewall 
PYthon3- 
-0.8.2-2.0.1.e18.noarch 
d-filesystem-O.8.2-2.0.1.e18. noarch 
[root@serverl systemctl is-active firewalld. service 
active 

1.1.3 Type **firewall-cmd --get-default-zone** to get the current zone set on system

1.1.4 Type **firewall-cmd --get-zones** to display the zones which available

[ root@serverl 
block dmz drop 
[ root@serve rl 
public 
firewall-cmd 
- -get-zones 
external home internal nm-shared public trusted work 
firewall-cmd 
- -get-default-zone 

1.1.4 Type **firewall-cmd --get-services** to display the services which are managed by the firewall

1.1.5 Type **firewall-cmd --list--all** , to display the current zone services and ports which are active

1.2. Add http service to firewall zone permanently using command line

1.2.1 Check on Workstation system if **curl** [**http://192.168.4.3**](http://192.168.4.3) display's the website - it should fail



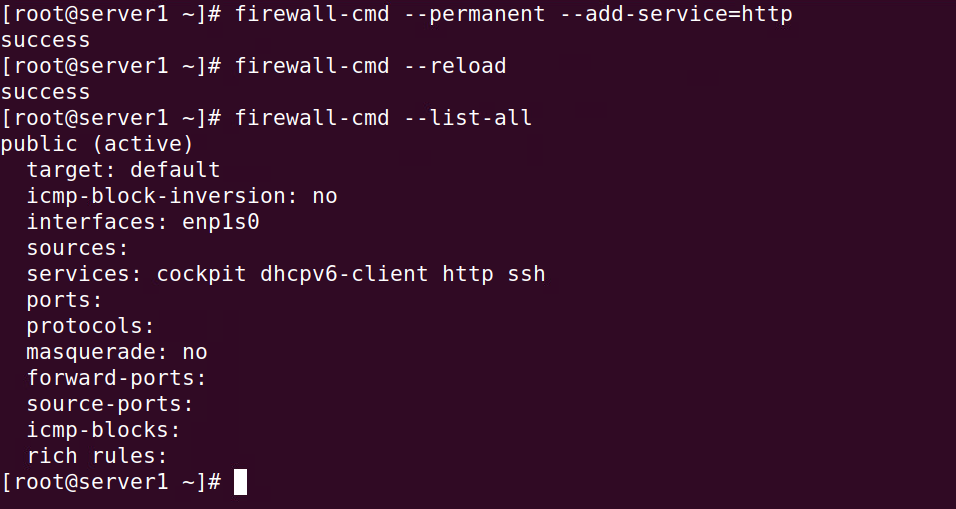
***Note: Make sure you have run the steps to create a website on Day2 to get this to work***

1.2.2 Type following commands in Server1 as root user

**firewall-cmd --permanent --add-service=http**

**firewall-cmd --reload**

**firewall-cmd --list-all**



1.2.3 Check on Workstation system if website open's, type

**curl** [**http://192.168.4.3**](http://192.168.4.3)



1.3 Checking firewall config from Cockpit

1.3.1 Check if cockpit software is installed on system,

**rpm -qa | grep cockpit**

1.3.2 Enable cockpit socket, type

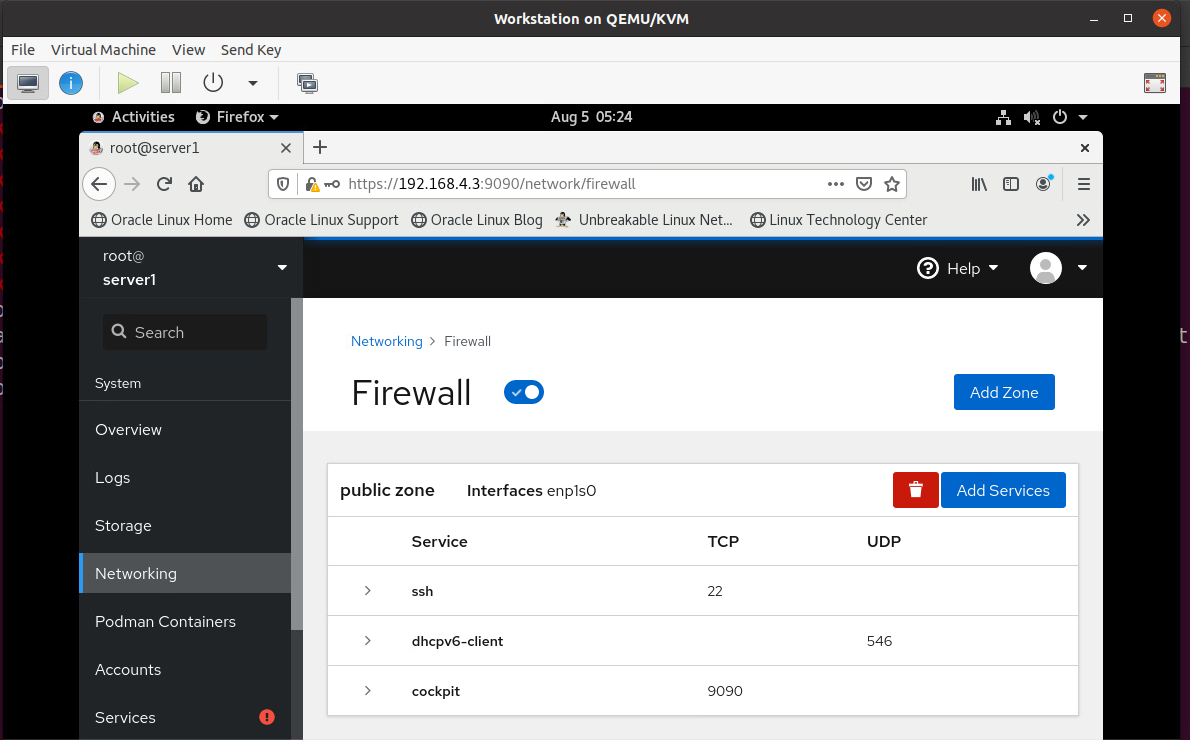
**systemctl enable --now cockpit.socket**

[root@serverl rpm -qa I grep cockpit 
- podman -18.1-2 . module+e18.3.0+7866+f387f528. noa rch 
COCKPII 
cockpit-storaged e18. noarch 
cockpit 
-224.2-1.0.1.e18.x86 64 
cockpit 
-bridge-224.2-1.O.1.e18.x86 64 
cockpj€-packagekit- 224.2-1.0.1. e18 . noarch 
cockpit 
-ws-224.2-1.O.1.e18.x86 64 
cockpit-system-224.2-1.O.1.e18. noarch 
[root@serverl systemctl enable - -now cockpit. socket 
Created symlink /etc/systemd/system/sockets. target . wants/cockpit.socket /usr/lib/systemd/system/c 
ockpit . socket . 
[ root@serverl 

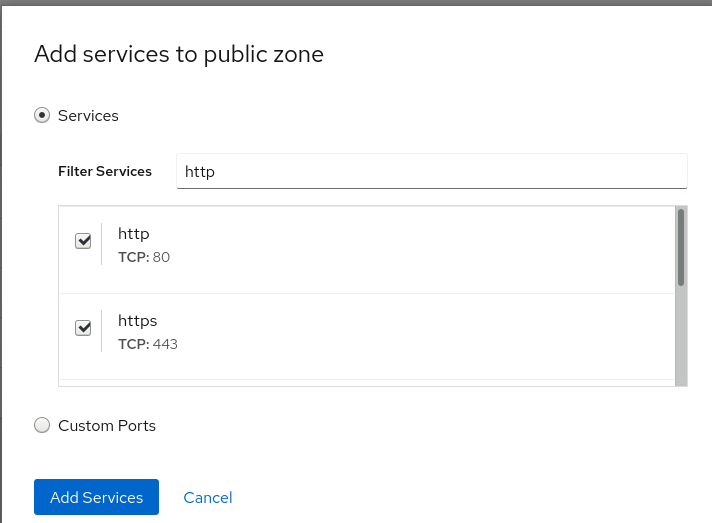
1.3.3 Launch console to Workstation system and open Browser (Firefox), type address <http://192.168.4.3:9090>

On Security warning accept risk and continue, login as user "root" password "Passw0rd!"

1.3.4 Select Networking on left panel and select firewall on right side to open firewall config



1.3.5 Click Add Services to add HTTP and HTTPS service to the current zone.



1.4 Connecting to a Remote Server with Public/Private Keys

1.4.1. On **server1**, open a root shell.

1.4.2. Type **ssh-keygen**. When asked whether you want to use a passphrase, leave blank Press Enter to use the passphrase-less setup.

1.4.3. When asked for the filename in which to store the (private) key, accept the default filename **~/.ssh/id\_rsa**.

1.4.4. When asked to enter a passphrase, press Enter twice. This step created private and public key in the **~/.ssh/id\_rsa**

1.4.5. Type **ll ~/.ssh** to confirm the key's being created

Private Key file is: ~/.ssh/id\_rsa

Public Key file is: ~/.ssh/id\_rsa.pub

Machine generated alternative text:
[root@serverl -]# Is 
-a 
anaconda-ks . c fg 
. bash logout 
. bash history 
. bash profile 
[root@serverl -]# ssh-keygen 
. bashrc 
. config 
.cshrc 
. history . bak 
. tcshrc 
. viminfo 
Generating public/private rsa key pair. 
Enter file in which to save the key (/root/.ssh/id rsa): 
Created directory ' / root/ .ssh' 
Enter passphrase (empty for no passphrase) : 
Enter same passphrase again: 
Your identification has been saved in /root/ .ssh/id rsa. 
Your public key has been saved in / root/ .ssh/id rsa.pub. 
The key fingerprint is: 
SHA256: p8+wFadoHr7+rf4Xa208zKS6Yyd5n1BMXVcoFhCOnnY root@serverl 

[ root@serve rl 
total 
8 
1 
Il —/.ssh 
2602 Aug 
root root 
root root 
566 Aug 
5 08:48 id rsa 
5 08:48 id rsa . pub 

1.4.6. Use **ssh-copy-id 192.168.4.4** to copy to server2 the public key you have just created. You are then asked for the password on the remote server one last time.

1.4.7. After copying the public key, verify that it can actually be used for authentication.

To do this, type **ssh 192.168.4.4** . You should now authenticate without having to enter the password for the remote user account.

For confirmation you can run command "**hostname**" and list the contents of folder to find autorized\_keys file ll ~/.ssh

[root@serverl ssh-copy-id 192.168.4.4 
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: 
"/root/. ssh/id_rsa . pub" 
The authenticity of host '192.168.4.4 (192.168.4.4)' can't be established. 
ECDSA key fingerprint is SHA256:b10F+StUYqPgMtX+GDIJLF3fkBcG4jdOBNkrHEeKpv4. 
Are you sure you want to continue connecting (yes/no/ [fingerprint])? yes 
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are al 
ready installed 
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed if you are prompted now it is to ins 
tall the new keys 
root@192.168.4.4' s password: 
Number of key(s) added: 1 
Now try logging into the machine, with: 
"ssh '192.168.4.41 " 
and check to make sure that only the key(s) you wanted were added. 
[root@serverl ssh root@192.168.4.4 
Activate the web console with: systemctl enable 
- -now cockpit. socket 
Last login: Thu Aug 5 2021 
[ root@server2 hostname 
se rver2 
1 
[ root@server2 

[ root@server2 
total 4 
[ root@server2 
11 —/.ssh 
root root 566 Aug 5 08:57 authorized keys 
cat —/ .ssh/authorized keys 
AAAAB3NzaC1yc2EAAAADAQABAAABgQDcKd6UhC12jRGCyaQBLmui4f4SS41CvD13MFMVVTaRCuEHhgLFNWrtedJV3y1 
TILAvYwGMDMqVOWbpFWuMhSuKiPdnaTvE1zB3kQuf6C6dy63DB012uqikmrWAFa/+7gr/KOQLEqWPnOBvZdNdaEwphhTvhDHZi5 
YssDzLIW21qkp6Sf35EgQP17PFINKZ3PiTNeMb6uerb9xEf8FgZm17YjxUanuaVCqw1yZ1aRPdsvMKgEUp/+yai/eG5D+t1VqXk 
JTpQJufHs/nS/01D1uqfh914vx+ScZPbogT070FVSvnJn1grLtNS1fY17n1yVbDZ1y9A8StAeZSV7kcMIJsgscUdRTrTJN59q1d 
LVs1FaeD6g58BHQeBF+3gh4DVGS5SgxTSAcSf 
nY4wpCKKkLXoj50pzkNe7xE011G45qX8sZnkn96EpeK/UwXbK9QRG32M= root@serverl 
1 
[root@server2 — 

1.5 Using SFTP and rsync to transfer files securely

1.5.1 On Server2 Create a directory by name "remotedir" under /tmp and touch /tmp/remotedir/fileonserver2.txt

[ root@server2 -]# mkdir -pv /tmp/remotedir 
mkdir: created directory /tmp/ remotedir' 
[ root@server2 touch /tmp/ remotedir/fileonserver2.xt 
[ root@server2 echo "This file is on server2 " >> /tmp/remotedir/fileonserver2 . Xt 

1.5.2. From Server1 terminal, type **sftp root@192.168.4.4**. This gives you access to an SFTP prompt that is opened on server2.

1.5.3. Type **ls**. You’ll see files in the current working directory on the remote server.

1.5.4. Type **pwd**. This shows the current directory on the remote server.

1.5.5. Type **lpwd**. This shows your local current directory.

1.5.6. Type **lcd /tmp**. This changes the local current directory to /tmp.

1.5.7. Now type **get /tmp/remotedir/fileonserver2.txt** , This file will be downloaded to the local /tmp directory.

1.5.8. Type exit to close your SFTP session.

[root@serverl sftp root@192.168.4.4 
Connected to root@192.168.4.4. 
sftp> Is 
anaconda -ks . cfg 
sftp> Is /tmp/ remotedir/ 
/tmp/ remotedi r/fileonserver2. Xt 
s ftp> pwd 
Remote working directory: /root 
sftp> Ipwd 
Local working directory: / root 
sftp> lcd /tmp 
sftp> get /tmp/ remotedir/fileonserver2 .xt 
Fetching /tmp/ remotedir/fileonserver2. Xt to fileonserver2. Xt 
/tmp/ remotedi r/fileonserver2. Xt 
sftp> exit 
[root@serverl Is -l /tmp/fi1eonserver2. Xt 
1 root root 25 Aug 5 13:33 /tmp/fileonserver2. Xt 
-rw-r--r--. 
100% 
25 
14.8KB/s 

On Server2 create one additional file by name file2onserver2.txt

Machine generated alternative text:



1.5.9. Type **rsync -a 192.168.4.4:/tmp/remotedir/ /tmp**. This synchronizes the contents of the remote directory /tmp in the local directory /tmp.

1.5.10. Type **ls -d /tmp/fileonserver2.txt /tmp/file2onserver2.txt** to verify that the remote directory has been synchronized to the local server.

[root@serverl rsync -av 192.168.4.4: /tmp/ remotedir/ /tmp 
receiving incremental file list 
file20nserver2. Xt 
fileonserver2 . Xt 
sent 71 bytes 
received 225 bytes 592.00 bytes/ sec 
total size is 
53 speedup is 0.18 
[ root@serverl 
—]# Is -l /tmp/fileonserver2. Xt /tmp/file20nserver2. Xt 
1 
root root 28 Aug 5 13:34 /tmp/file20nserver2. Xt 
-rw-r--r--. 
rwr 1 
root root 25 Aug 5 13:31 /tmp/fileonserver2. Xt 
1 
[ root@serverl 

1. File Access Control

Run these steps as the root user on the Server1

2.1 Verify, change and Modify the attribute of file and directory

2.1.1 Create demodir under /tmp/project and file /tmp/project/demoattr.txt

Type **mkdir -pv /tmp/project/demodir**

Type **touch /tmp/project/demoattr.txt**

Type **echo "This file will be locked with attribute of Immutable" >> /tmp/project/demoattr.txt**

[ root@se rverl 
mkdir: created 
mkdir: created 
[ root@serverl 
[ root@serverl 
emoattr . txt 
-]# mkdir -pv /tmp/project/demodir 
directory ' /tmp/project ' 
directory ' /tmp/project/demodir' 
touch /tmp/project/demoattr.txt 
echo "This file will be set locked with attribute of Immutable" 
/tmp/project/d 

2.1.2 List the attributes of the /tmp/project/demodir and /tmp/rpject/demoattr.txt

Type **ls -l /tmp/project**

Type **lsattr /tmp/project**

[ root@serverl 
total 4 
-rw-r--r-- 
drwxr-xr-x. 
2 
[ root@serverl 
Is -l /tmp/project 
root root 57 Aug 5 12:03 demoattr.txt 
root root 6 Aug 5 12:00 
demodir 
Isattr /tmp/project 
- /tmp/project/demodir 
/tmp/project/demoattr . txt 

2.1.3 Change the attribute of demodir and demoattr.txt to "i" immutable

Type **chattr +i /tmp/project/demodir**

Type **chattr +i /tmp/project/demoattr.txt**

Type **lsattr /tmp/project** to list the attributes set

[ root@serve rl 
chattr +1 /tmp/project/demodir/ 
[ root@serve rl 
chattr +i /tmp/project/demoattr.txt 
[ root@serverl 
Isattr /tmp/project/ 
— /tmp/project/demodir 
- /tmp/project/demoattr. txt 

2.1.4 Now try to modify the file demoattr.txt - it would fail

Type **echo "Trying to modify file demoattr….. " >> /tmp/project/demoattr.txt**

Now try to remove the file demoattr.txt - it would fail, also try modify the permission (chmod) to file even that would fail

Type **rm -rf /tmp/project/demoattr.txt**

Type **chmod 777 /tmp/project/demoattr.txt**

[root@serverl echo "Trying to modify file demoattr 
" /tmp/project/demoattr. txt 
-bash: /tmp/project/demoattr. txt: Operation not permitted 
[root@serverl —]# rm -rf /tmp/project/demodir/ 
rm: cannot remove ' /tmp/project/demodir/' : Operation not permitted 
[ root@serverl chmod 777 /tmp/project/demodir/ 
chmod: changing permissions of '/tmp/project/demodir/' : 
Operation not permitted 

2.1.5 To remove the attribute of a file and try to modify the file now - it will work.

Type **chattr -i /tmp/project/demoattr.txt**

Type **lsattr /tmp/project**

Type **echo "This file can be edited now …. " >> /tmp/project/demoattr.txt**

[ root@se rverl 
[ root@se rverl 
----i-- 
[ root@se rverl 
[ root@se rverl 
-]# 
-]# 
-]# 
-]# 
chattr -i /tmp/project/demoattr.txt 
Isattr /tmp/project 
- /tmp/proj ect/demodir 
/tmp/project/demoattr. txt 
echo "This file can be edited no " 
cat /tmp/project/demoattr.txt 
set locked with attribute of Immutable 
>> /tmp/project/demoattr. txt 
This file will be 
This file can be edited no . 
[ root@serverl 

2.2 Managing Advanced Permissions using ACL's

Configure advanced access control for folder to allow users of group "accounts" to write and modify the contents of the folder /tmp/project/ and user tim should only read the folder /tmp/project/

2.2.1 Create user ted , tux add them to group accounts, Create user tim under group sales, note password for all users should be "Passw0rd"

Login to Server1 as Root user and run following commands

Type **for L in tim tux ted ; do useradd -c $L $L ; echo Passw0rd | passwd --stdin $L ; done**

Type **groupadd accounts ; groupadd sales**

Type **usermod -G accounts ted ; usermod -G accounts tux ; usermod -G sales tim**

Type **id tim; id tux ; id ted**

[root@serverl for L in 
done 
Changing password for user 
passwd: all authentication 
Changing password for user 
passwd: all authentication 
Changing password for user ted 
passwd: all authentication 
tim tux ted 
do useradd -c 
tim. 
tokens updated successfully. 
tux. 
tokens updated successfully. 
tokens updated successfully. 
groupadd sales 
accounts 
, usermod -G 
-G accounts ted 
id tux , 
id ted 
$L 
echo PasswOrd I 
pas swd 
--stdin $L , 
[ root@serverl 
[ root@serverl 
[ root@serverl 
—]# 
groupadd 
usermod 
id tim , 
accounts tux 
usermod 
-G sales tim 
, 1605(sales) 
gid=1002(tux) groups=1002(tux) , 1004 (accounts) 
, 1004(accounts) 

2.2.2 List the ACL of the folder /tmp/project, modify the ACL to allow users of group "accounts" to modify the contents

Type **getfacl /tmp/project**

Type **setfacl -m g:accounts:rwx,u:tim:r-- /tmp/project**

Type **getfacl /tmp/project**

Type **ls -ld /tmp/project** to check that there is a plus sign to indicate ACL has been set

[ root@serverl getfacl 
get fact: Removing leading 
# file: tmp/project 
# owner: root 
# group: root 
user: : rwx 
group: :r-x 
other: :r-x 
[root@serverl setfacl 
[root@serverl getfacl 
get fact: Removing leading 
# file: tmp/project 
# owner: root 
# group: root 
user: : rwx 
user:tim: r-- 
group: : r-x 
group: accounts : rwx 
/tmp/project 
'/' from absolute path names 
-m g:accounts:rwx,u:tim:r-- 
/tmp/project/ 
/tmp/project 
'/' from absolute path names 
mask: : rwx 
other::r-x 
[ root@serverl 
drwxrwxr-x+ 3 
Is -Id /tmp/project 
root root 41 Aug 5 12:01 

2.2.3 Verify that users of accounts department can modify the contents of the folder /tmp/project and user tim can only list the contents

Type **su -l ted**

Type **echo "User Ted created this file" >> /tmp/project/newfile.txt**

Type **su -l tux** (enter Password as "Passw0rd" if prompted)

Type **echo "User Tux tried to modify this file " >> /tmp/project/newfile.txt**

This would fail as the newfile.txt is not owned by accounts group

Type **su -l root** (enter Password as "Passw0rd!" on prompt)

Type **chgrp account /tmp/project/newfile.txt**

Now try to edit file as tux user it should work

Type **su -l tim**  ( enter Password as "Passw0rd" if prompted)

Type **echo "User Tim is trying to modify the file " >> /tmp/project/newfile.txt**

Above command would fail as User Tim has no permission to modify the file

[tux@serverl —IS echo "User 
TUX 
-bash: /tmp/project/newfile. txt: 
[tux@serverl su -l root 
Password : 
su: Authentication failure 
tried to modify this file " 
Permission denied 
[tux@serverl 
Password : 
[ root@se rverl 
[ root@se rverl 
logout 
[tux@serverl 
[tux@serverl 
—IS su -l root 
chgrp accounts /tmp/project/newfile. txt 
exit 
—IS echo "User TUX tried to modify this file " 
—IS cat /tmp/project/newfile.txt 
/tmp/project/newfile.txt 
/tmp/project/newfile.txt 
User TED created this file 
User TUX tried to modify this file . 
[tux@serverl su -l tim 
Passwo rd : 
[tim@serverl —l $ Il /tmp/project/ 
Is: cannot access '/tmp/project/demodir' : 
Permission denied 
Is: cannot access '/tmp/project/demoattr.txt' : 
Permission denied 
Is: cannot access '/tmp/project/newfile. txt' : 
Permission denied 
total 0 
-777777777 7 
[tim@serverl -IS 
7 
demoattr. txt 
demodir 
newfile. txt 

\*\*\*\* This Concludes Day3 Lab \*\*\*\*

Day4

1. Network Storage access

1.1 Mount /shares folder from Server2 at /external folder on Server1

1.2 Auto mount NFS file shares

2. Using simple regular expressions

To get More info on nfs and autofs mount use following try commands

**pinfo nfs**

**man exportfs**

**pinfo showmount**

**man 5 autofs**

1.1 Mount /shares folder from Server2 at /mnt folder on Server1

Task1 Creating a NFS share on Server2 (192.168.4.4)

These Steps will be done on Server2 as Root user to create nfs shares

1.1.1 Connect to Server2 as root user

Type **ssh root@192.168.4.4**

(Enter password as "Passw0rd!" when prompted)

[root@workstation / ssh root@192.168.4.4 
root@192.168.4.4's password: 
Activate the web console with: systemctl enable 
- -now cockpit. socket 
Last login: Thu Aug 5 2021 from 192.168.4.1 
[ root@server2 
server2 
root 
[ root@server2 
hostname 
, echo ; whoami 
1 

1.1.2 Create Some Local folders on Serer2 which would be shared create files under them

Type **mkdir -p /shares/ /users/user{1,2}**Type **echo "This file is on NFS Server2." >> /shares/fileonserver2.txt**

Type **echo "This file is of User1 on NFS Server2." >> /users/user1/user1file.txt**

Type **echo "This file is of User2 on NFS Server2." >> /users/user2/user2file.txt**

[ root@server2 
mkdir: created 
mkdir: created 
mkdir: created 
mkdir: created 
[ root@server2 
[ root@server2 
[ root@server2 
[ root@server2 
[shares 
mkdir 
di rectory 
di recto ry 
di recto ry 
di recto ry 
-pv /shares /users/user{1,2} 
' / shares ' 
'/users' 
' /users/userl' 
' /users/user2 ' 
echo "This file is on NFS SERVER2" /shares/fileonserver2. txt 
echo "This is Userl file on NFS server2" >> / users/userl/userlfile.txt 
—]# echo "This is User2 file on NFS server2" >> / users/user2/user2file.txt 
tree / shares /users 
fileonserver2. txt 
/users 
userl 
user If ile . txt 
user2 
user2file . txt 
2 directories, 3 files 
[root@server2 

1.1.3. Type **yum install -y nfs-utils** to install the required packages.

Machine generated alternative text:



1.1.4. Use vim to create the /etc/exports file and give it the following contents:

**/shares \*(rw,no\_root\_squash)**

**/users \*(rw,no\_root\_squash)**

[root@server2 vim /etc/exports 
[root@server2 cat /etc/exports 
/shares *(rw,no root squash) 
/users *(rw,no root squash) 

1.1.5. Type **systemctl enable --now nfs-server** to start and enable the NFS server.

1.1.6 Add NFS, RPC-Bind and mountd services to firewall

Type **firewall-cmd --add-service nfs --permanent**

Type **firewall-cmd --add-service rpc-bind --permanent**

Type **firewall-cmd --add-service mountd --permanent**

Make the newly added services effective at this point,

Type **firewall-cmd --reload**

[ root@serve r2 
[ root@serve r2 
success 
[ root@serve r2 
success 
[ root@serve r2 
success 
[ root@serve r2 
success 
-]# 
systemctl enable - -now nfs-server.service 
- -add-service=rpc-bind 
- -add - servxce=mountd 
firewall-cmd - -add-service=nfs 
fi rewall- cmd 
fi rewall - cmd 
firewall- cmd - - reload 
- - permanent 
- - permanent 
- -pe rmanent 

1.1.7 Type  **exportfs -arv** to list the exported folders

[ root@server2 exportfs -arv 
exporting * : / users 
exporting *:/shares 

Task2

1.1.8 Now To Mount an NFS Shares created on Server1 Client system , Connect to server1 as root user

Type **ssh root@192.168.4.3** (Enter password as "Passw0rd!" when prompted)

Type **yum install -y nfs-utils** to install the RPM package that contains the showmount utility.

Type **showmount -e 192.168.4.4** to see all exports available from server2.

[ student@serverl 
Passwo rd : 
[ root@serverl 
showmount -e 192.168.4.4 
-bash: showmount: command not found 
[ root@serverl 
yum install -y nfs-utils 

Type **mount 192.168.4.4:/ /mnt**

(Note the space between the slashes in the command.) This performs an NFSv4 pseudo root mount of all NFS shares.

1.1.9 Type **mount | grep nfs** to verify the mount has succeeded.

Type **ls /mnt**. This shows the subdirectories shares and users, which correspond to the mounts offered by the NFS server.

[root@serverl mount 192 .168.4.4:/ /mnt 
[root@serverl mount I grep nfs 
sunrpc on /var/lib/l 
/rpc pipefs type rpc_pipefs (rw, relatime) 
192.168.4.4:/ on /mnt type 
f 34 (rw, relatime , vers=4.2, rsize=262144,wsize=262144, namlen=255 , hard, pro 
to=tcp , timeo=600 , ret rans=2 , sec=sys , clientadd r=192 . 168.4.3, local _ lock-none , add r=192 . 168.4.4) 
[root@serverl Is /mnt 
'shares users 
[root@serverl tree /mnt 
/mnt 
sha res 
fileonserver2. txt 
users 
userl 
userlfile. txt 
user2 
user2file. txt 
4 directories, 3 files 

Note: *If you need to make NFS as a persistent mount then need to create an entry in /etc/fstab file*

*<Server-FQDN-or-IP-Address>:/<share-name> <local-mount-point> nfs rw,sync 0 0*

*Eg. 192.168.4.4:/shares /mnt nfs rw,sync 0 0*

1.2 Auto mount NFS file shares

1.2.1 Configuring Direct and Indirect Maps to Mount NFS Shares

This exercise is performed on server1. It uses the NFS shares provided by server2 that you created in previous Exercise.

Connect to Server1 as root user

Type **ssh root@192.168.4.3** (Enter password as "Passw0rd!" when prompted)

Type **yum install -y autofs** to install the autofs package.

1.2.2. Type **showmount -e 192.168.4.4** , which shows you NFS exports offered by server2.

[student@serverl —IS su -l 
Password: 
[root@serverl yum install -y -q autofs 
[root@serverl showmount -e 192 .168.4.4 
Export list for 192.168.4.4: 

1.2.3. Open the file **vim /etc/auto.master** and add the following line:

**/external /etc/auto.nfsdata**

1.2.4. Type **vim /etc/auto.nfsdata** and add the following line:

**files -rw,sync,fstype=nfs4 192.168.4.4:/shares**

1.2.5. Type **systemctl enable --now autofs** to start and enable the autofs service.

Type **ls /external** . Notice that there is no files directory under /external.

Type **cd /external/files** to get access to the /shares directory.

Type **mount** and notice the last three lines in the mount output, created by the autofs service.

[root@serverl vim /etc/auto.master 
[root@serverl cat /etc/auto.master I grep external 
/etc/auto. nfsdata 
external 
[root@serverl vim /etc/auto.nfsdata 
[root@serverl —]# systemctl enable --now autofs.service 
Created symlink /etc/systemd/system/multi-user.target .wants/autofs.service /usr/lib/systemd/syste 
m/autofs . service. 
[root@serverl Is 
anaconda-ks . cfg 
[root@serverl —]# Is / 
bin 
dev external lib 
media 
ml sc 
mnt 
net 
opt 
proc 
root 
run 
sbin 
s rv 
sys 
tmp 
usr 
var 
web 
boot etc home 
lib64 
[root@serverl tree /external/ 
/external/ 
O directories, O files 
[root@serverl tree /external/files 
/external/files 
I— fileonserver2.txt 
O directories, 1 file 
1 
[root@serverl 

1.2.6 Configuring Wildcard Auto Mounts

This exercise is performed on server1. It uses the NFS shares that are provided by server2, which you created in Exercise 24-1. On server2, the directory /users is exported, which simulates an NFS server that exports home directories. You are going to configure a wildcard mount, such that when /users/user1 is accessed, that exact directory is mounted, and when /users/user2 is accessed, that directory is mounted.

This activity is performed on Server1 as root user

1.2.6.1. Open the file /etc/auto.master and make sure it includes the following line:

**/users /etc/auto.users**

1.2.6.2. Create the file /etc/auto.users and give it the following contents:

**\* -rw,sync,fstype=nfs4 192.168.4.4:/users/&**

1.2.6.3. Type **systemctl restart autofs** to restart the autofs service.

Type **cd /users/user1** to get access to the NFS export /users/user1 on the server2 server

[ root@serverl 
[ root@serverl 
/etc/auto 
[ root@serverl 
[ root@serverl 
* -rw,sync,fstype 
vum 
cat 
vim 
cat 
/etc/auto . master 
/etc/auto .master I grep users 
/etc/auto 
. users 
/etc/auto . 
users 
[ root@serve rl 
[ root@serve rl 
[ root@serve rl 
[ root@serve rl 
[ root@serve rl 
userlfile. txt 
[ root@serve rl 
-]# 
=nfs4 
systemctl restart autofs.service 
cd /users 
users]# 
userl]# 
userll# 
Is 
cd userl 
1 

2. Using simple regular expressions

Run this on Server1 as student user

2.1 Type **ssh student@192.168.4.3**

Read about regular expressions from **man 7 regex** and **man grep**

2.2 Check if the words file is installed if not install it

Type **rpm -qa words**

Note: If above command returns blank then install it using **yum -y install words**

2.2 Copy words file from /usr/share/dict to home directory and find number of words in it

Type **cp -v /usr/share/dict/words /home/student**

Type **wc /home/student/words**

2.3 List words with "robot" in words file and List words starting with "anti"

Type **grep -i robot /home/student/words**

Type **grep '^anti' /home/student/words**

2.5 List number of users with /sboin/nologin shell from /etc/passwd file

Type **grep 'nologin' /etc/passwd | wc -l**

[student@serverl —1$ grep -i '/sbin/nologin' /etc/passwd I wc -l 
28 

2.6 Check if user tux is present on the system using grep

Type **cat /etc/passwd | grep -i tux**

2.7 List Words that **begin** with 'anti' and end with 'ism' that are **exactly** 13 characters long

Type **grep '^anti......ism$' /home/student/words**

Type **grep '^anti.\{6\}ism$' /home/student/words**

**2.8** Words that **begin** with either 'f' or 'F' and **end** in 'z' or 'Z' of any length

Type **grep '^[fF].\*[zZ]$' /home/student/words**

\*\*\*\* This Concludes Day 4 Lab \*\*\*\*