

Linux System Administration for DevOps Engineers



By Ritesh Kumar Singh

Email Address: - riteshkumarsingh9559@gmail.com

LinkedIn: - <https://www.linkedin.com/in/ritesh-kumar-singh-41113128b/>

GitHub: - <https://github.com/singhritesh85>



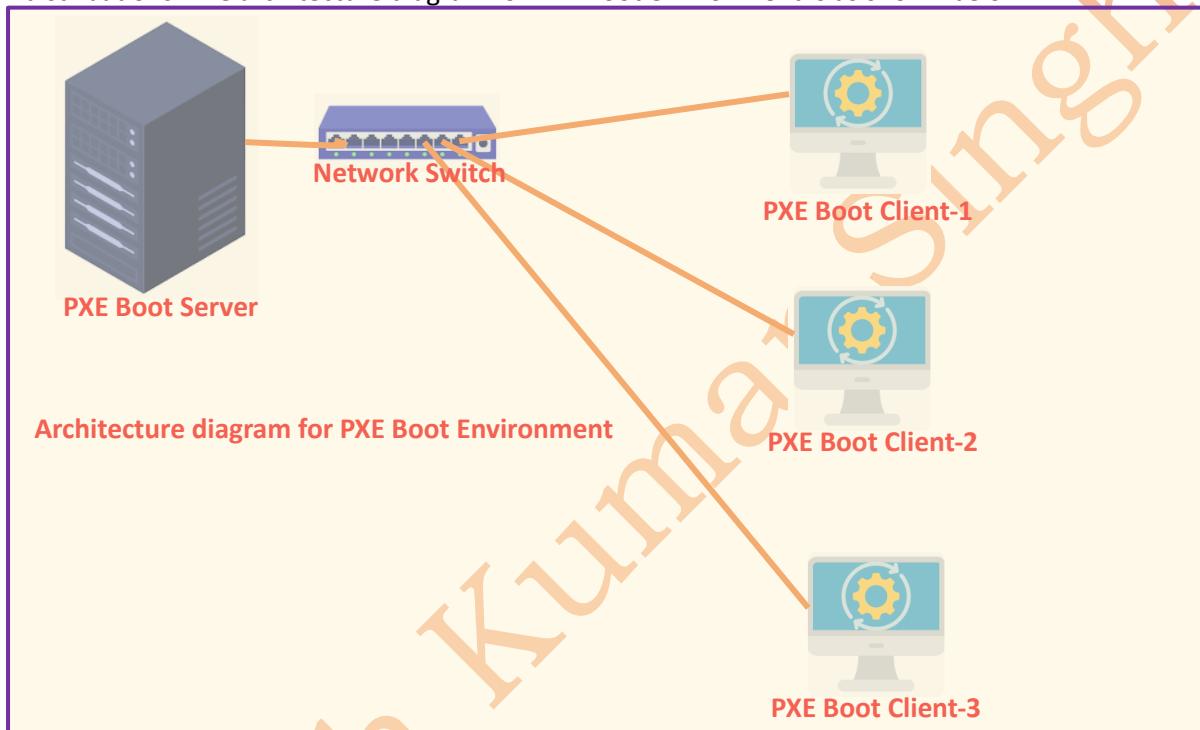
या कुन्देन्दुतुषारहारधवला या शुभ्रवस्त्रावृता
या वीणावरदण्डमण्डितकरा या श्वेतपद्मासना।
या ब्रह्माच्युत शंकरप्रभृतिभिर्देवैः सदा वन्दिता
सा मां पातु सरस्वती भगवती निःशेषजाङ्गापहा ॥

Linux System Administration for DevOps Engineers

To natively install Linux on a machine in Organisations two methods usually followed 1. **PXE (Preboot Execution Environment) Boot** or 2. Boot from a **USB Flash Drive (bootable/multi-boot pen drive)**.

1. PXE (Preboot Execution Environment) Boot

PXE Boot environment uses client-server model, there will be one PXE Boot Server (which assigns IP addresses using DHCP to the Client, TFTP Server transfer initial Boot Files bootloader and kernel and NFS Server to host the Linux Distribution Files). On PXE Client we need to Install the Linux distributions. The architecture diagram for PXE Boot environment is as shown below.



To configure PXE Boot Server I followed the steps as written below.

```
[root@pxe-boot-server ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether [REDACTED] brd [REDACTED]
    altname enp2s1
    inet 192.168.40.128/24 brd 192.168.40.255 scope global dynamic noprefixroute ens33
        valid_lft 1787sec preferred_lft 1787sec
    inet6 [REDACTED] scope link noprefixroute
        valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether [REDACTED] brd [REDACTED]
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever

[root@pxe-boot-server ~]# yum install -y dnsmasq syslinux tftp-server nfs-utils
```

```

Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing : 1/1
  Running scriptlet: syslinux-nolinux-6.04-6.el8.noarch 1/1
  Installing : syslinux-nolinux-6.04-6.el8.noarch 1/1
  Installing : syslinux-6.04-6.el8.x86_64 1/1
  Running scriptlet: dnsmasq-2.79-35.el8_10.x86_64 2/7
  Upgrading  : dnsmasq-2.79-35.el8_10.x86_64 3/7
  Running scriptlet: dnsmasq-2.79-35.el8_10.x86_64 3/7
  Running scriptlet: nfs-utils-1:2.3.3-64.el8_10.x86_64 4/7
  Upgrading  : nfs-utils-1:2.3.3-64.el8_10.x86_64 4/7
  Running scriptlet: nfs-utils-1:2.3.3-64.el8_10.x86_64 4/7
  Installing : tftp-server-5.2-27.el8.x86_64 5/7
  Running scriptlet: tftp-server-5.2-27.el8.x86_64 5/7
  Running scriptlet: dnsmasq-2.79-33.el8_10.x86_64 6/7
  Cleanup    : dnsmasq-2.79-33.el8_10.x86_64 6/7
  Running scriptlet: dnsmasq-2.79-33.el8_10.x86_64 6/7
  Running scriptlet: nfs-utils-1:2.3.3-59.el8.x86_64 7/7
  Cleanup    : nfs-utils-1:2.3.3-59.el8.x86_64 7/7
  Running scriptlet: nfs-utils-1:2.3.3-59.el8.x86_64 7/7
  Verifying   : syslinux-6.04-6.el8.x86_64 1/7
  Verifying   : syslinux-nolinux-6.04-6.el8.noarch 2/7
  Verifying   : tftp-server-5.2-27.el8.x86_64 3/7
  Verifying   : nfs-utils-1:2.3.3-64.el8_10.x86_64 4/7
  Verifying   : nfs-utils-1:2.3.3-59.el8.x86_64 5/7
  Verifying   : dnsmasq-2.79-35.el8_10.x86_64 6/7
  Verifying   : dnsmasq-2.79-33.el8_10.x86_64 7/7

Upgraded: dnsmasq-2.79-35.el8_10.x86_64
Installed: syslinux-6.04-6.el8.x86_64          nfs-utils-1:2.3.3-64.el8_10.x86_64
syslinux-nolinux-6.04-6.el8.noarch           tftp-server-5.2-27.el8.x86_64

Complete!

```

[root@pxe-boot-server ~]# mv /etc/dnsmasq.conf /etc/dnsmasq.conf.bkp

```

[root@pxe-boot-server ~]# cat /etc/dnsmasq.conf
interface=ens33 # e.g., eth0, enp1s0
dhcp-range=192.168.40.150,192.168.40.250,12h
dhcp-option=3,192.168.40.1 # Gateway
dhcp-option=6,8.8.8.8,8.8.4.4 # DNS servers

# TFTP server configuration
enable-tftp
tftp-root=/var/lib/tftpboot

# PXE boot configuration
dhcp-boot=pxelinux.0

```

The **Syslinux** **PXE** bootloaders are installed under the path /usr/share/syslinux as shown in the screenshot attached below.

```

[root@pxe-boot-server ~]# ls /usr/share/syslinux/
altmbn.bin  cpu.c32  dmtest.c32  hdt.c32  isohdpfx_f.bin  lfs.c32  lua.c32  pcitest.c32  sanboot.c32  vesamenu.c32
altmbn_c.bin  cpuid.c32  dosutil  hexdump.c32  isohdppx.bin  libcom32.c32  mboot.c32  pmload.c32  sdi.c32  vpdtest.c32
altmbn_f.bin  cpuidtest.c32  elf.c32  host.c32  isohdppx_c.bin  libgpl.c32  mbr.bin  poweroff.c32  sysdump.c32  whichsys.c32
cat.c32  debug.c32  etherse1.c32  ifcpu.c32  isohdppx_f.bin  liblua.c32  mbr_c.bin  prdhcp.c32  syslinux.c32  zzjson.c32
chain.c32  dhcp.c32  gfbboot.c32  ifcpu64.c32  isolinux-debug.bin  libmenu.c32  mbr_f.bin  pwd.c32  syslinux.com
cmd.c32  diag  gptmbr.bin  ifmemdisk.c32  isolinux.bin  libutil.c32  memdisk  pxchn.c32  syslinux.exe
cmenu.c32  dir.c32  gptmbr_c.bin  iflop.c32  kbdmap.c32  linux.c32  meminfo.c32  pxelinux.0  syslinux64.exe
config.c32  disk.c32  gptmbr_f.bin  isohdpfx.bin  kontron_wdt.c32  lpxelinux.0  menu.c32  reboot.c32  vesa.c32
cptime.c32  dmi.c32  gpxecmd.c32  isohdpfx_c.bin  ldlinux.c32  ls.c32  pci.c32  rosh.c32  vesainfo.c32

```

Copy all Syslinux bootloaders from /usr/share/syslinux/ to /var/lib/tftpboot as shown in the screenshot attached below.

[root@pxe-boot-server ~]# cp -r /usr/share/syslinux/* /var/lib/tftpboot/

The **PXE Server** reads its configuration from file pxelinux.cfg, which is found in the directory described in the tftp-root setting from the DNMSAQ configuration file above.

First, create a pxelinux.cfg directory and create a default file using the commands as written below

```
[root@pxe-boot-server ~]# mkdir /var/lib/tftpboot/pxelinux.cfg
[root@pxe-boot-server ~]# touch /var/lib/tftpboot/pxelinux.cfg/default

[root@pxe-boot-server ~]# cat /var/lib/tftpboot/pxelinux.cfg/default
default menu.c32
prompt 0
timeout 300
ONTIMEOUT local

menu title ##### PXE Boot Menu #####
label 1
menu label ^1) Install Almalinux 8
kernel images/almalinux/vmlinuz
append initrd=images/almalinux/initrd.img inst.repo=nfs:nfsvers=4:192.168.40.128:/srv/nfs/almalinux/

[root@pxe-boot-server ~]# mkdir -p /srv/nfs/almalinux

[root@pxe-boot-server ~]# mount -o loop /root/AlmaLinux-8.10-x86_64-dvd.iso /mnt
mount: /mnt: WARNING: device write-protected, mounted read-only.

[root@pxe-boot-server ~]# rsync -avz /mnt/* /srv/nfs/almalinux/

[root@pxe-boot-server ~]# umount -l /mnt

[root@pxe-boot-server ~]# cat /etc/exports
/srv/nfs/almalinux 192.168.40.0/24(rw,sync,no_root_squash)

[root@pxe-boot-server ~]# exportfs -r

[root@pxe-boot-server ~]# systemctl restart nfs-server
[root@pxe-boot-server ~]# systemctl status nfs-server
● nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; disabled; vendor preset: disabled)
   Active: active (exited) since Sun 2025-10-26 15:28:44 IST; 13s ago
     Process: 40080 ExecStart=/bin/sh -c if systemctl -q is-active gssproxy; then systemctl reload gssproxy ; fi (code=exited, status=0/SUCCESS)
    Process: 40066 ExecStart=/usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)
    Process: 40064 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
   Main PID: 40080 (code=exited, status=0/SUCCESS)

Oct 26 15:28:44 pxe-boot-server systemd[1]: Starting NFS server and services...
Oct 26 15:28:44 pxe-boot-server systemd[1]: Started NFS server and services.

[root@pxe-boot-server ~]# umount -l /mnt

[root@pxe-boot-server ~]# mkdir -p /var/lib/tftpboot/images/almalinux
[root@pxe-boot-server ~]# cp /srv/nfs/almalinux/images/pxeboot/vmlinuz /var/lib/tftpboot/images/almalinux/
[root@pxe-boot-server ~]# cp /srv/nfs/almalinux/images/pxeboot/initrd.img /var/lib/tftpboot/images/almalinux/
[root@pxe-boot-server ~]# systemctl restart dnsmasq tftp
[root@pxe-boot-server ~]# systemctl enable dnsmasq tftp
```

```
[root@pxe-boot-server ~]# systemctl status dnsmasq tftp
● dnsmasq.service - DNS caching server.
  Loaded: loaded (/usr/lib/systemd/system/dnsmasq.service; enabled; vendor preset: disabled)
  Active: active (running) since Sun 2025-10-26 15:34:10 IST; 7s ago
    Main PID: 40199 (dnsmasq)
      Tasks: 1 (limit: 22873)
     Memory: 1.0M
      CGroup: /system.slice/dnsmasq.service
              └─40199 /usr/sbin/dnsmasq -k

Oct 26 15:34:10 pxe-boot-server systemd[1]: Started DNS caching server.
Oct 26 15:34:10 pxe-boot-server dnsmasq[40199]: started, version 2.79 cachesize 150
Oct 26 15:34:10 pxe-boot-server dnsmasq[40199]: compile time options: IPv6 GNU-getopt DBus no-i18n IDN2 DHCP DHCPv6 no-Lua TFTP no-conntrack ipset auth DNSSEC
Oct 26 15:34:10 pxe-boot-server dnsmasq-dhcp[40199]: DHCP, IP range 192.168.40.150 -- 192.168.40.250, lease time 12h
Oct 26 15:34:10 pxe-boot-server dnsmasq-tftp[40199]: TFTP root is /var/lib/tftpboot
Oct 26 15:34:10 pxe-boot-server dnsmasq[40199]: reading /etc/resolv.conf
Oct 26 15:34:10 pxe-boot-server dnsmasq[40199]: using nameserver 192.168.40.2#53
Oct 26 15:34:10 pxe-boot-server dnsmasq[40199]: read /etc/hosts - 2 addresses

● tftp.service - Tftp Server
  Loaded: loaded (/usr/lib/systemd/system/tftp.service; indirect; vendor preset: disabled)
  Active: active (running) since Sun 2025-10-26 15:34:10 IST; 7s ago
    Docs: man:in.tftpd(8)
  Main PID: 40200 (in.tftpd)
    Tasks: 1 (limit: 22873)
   Memory: 184.0K
      CGroup: /system.slice/tftp.service
              └─40200 /usr/sbin/in.tftpd -s /var/lib/tftpboot

Oct 26 15:34:10 pxe-boot-server systemd[1]: Stopping Tftp Server...
Oct 26 15:34:10 pxe-boot-server systemd[1]: tftp.service: Succeeded.
Oct 26 15:34:10 pxe-boot-server systemd[1]: Stopped Tftp Server.
Oct 26 15:34:10 pxe-boot-server systemd[1]: Started Tftp Server.
```

```
[root@pxe-boot-server ~]# firewall-cmd --permanent --add-service=dhcp
success
[root@pxe-boot-server ~]# firewall-cmd --permanent --add-service=tftp
success
[root@pxe-boot-server ~]# firewall-cmd --permanent --add-service=nfs
success
[root@pxe-boot-server ~]# firewall-cmd --reload
success
```

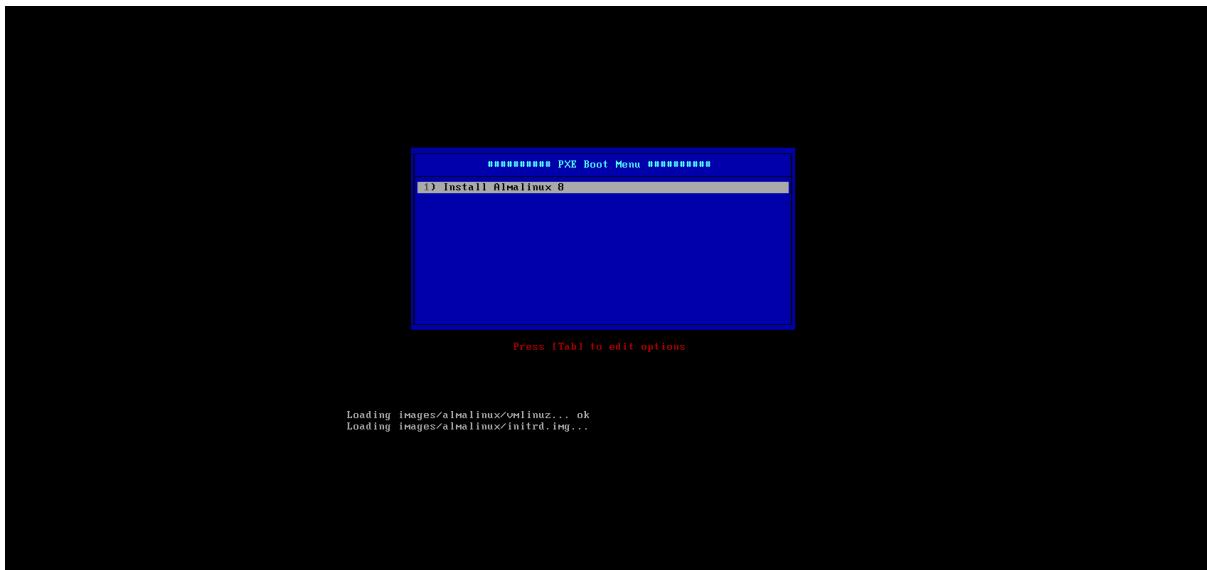
```
[root@pxe-boot-server ~]# firewall-cmd --permanent --add-port=2049/tcp
success
[root@pxe-boot-server ~]# firewall-cmd --permanent --add-port=2049/udp
success
[root@pxe-boot-server ~]# firewall-cmd --permanent --add-port=111/tcp
success
[root@pxe-boot-server ~]# firewall-cmd --permanent --add-port=111/udp
success
[root@pxe-boot-server ~]# firewall-cmd --reload
success
```

```
[root@pxe-boot-server ~]# firewall-cmd --permanent --add-service=rpc-bind
success
[root@pxe-boot-server ~]# firewall-cmd --permanent --add-service=mountd
success
```

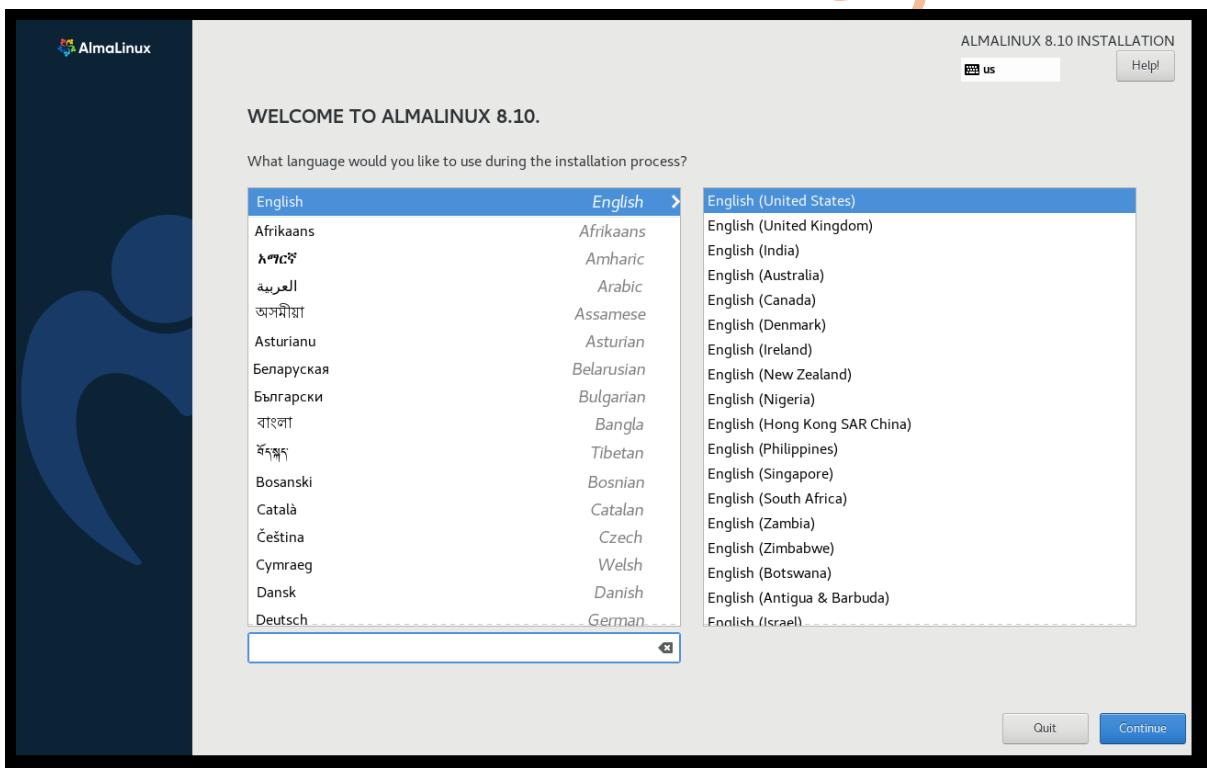
```
[root@pxe-boot-server ~]# firewall-cmd --reload
success
```

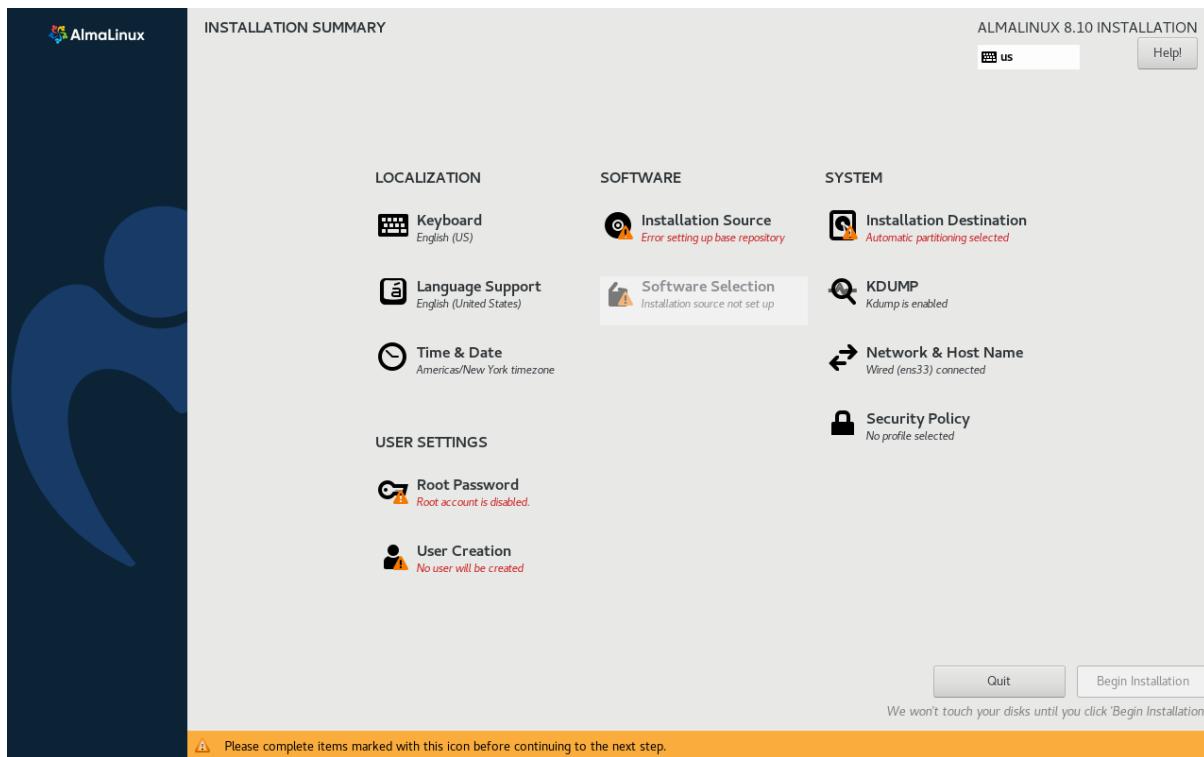
Here I considered the Linux Systems (PXE Boot Server and PXE Clients) are available in the network 192.168.40.0/24. Using below steps I shown how to install Linux Distros on PXE Client.

Power On the PXE Client (Systems where you want to install Linux OS) and press **esc** to bring the startup menu and repeatedly press the key **F12** (for Network Boot). If Network Boot Option is not showing in the startup menu, then you need to Enable it from the BIOS Option. Press F10 key for BIOS Option and enable the Network Boot. Then restart your system and press **F12** Key for Network Boot (PXE Boot).

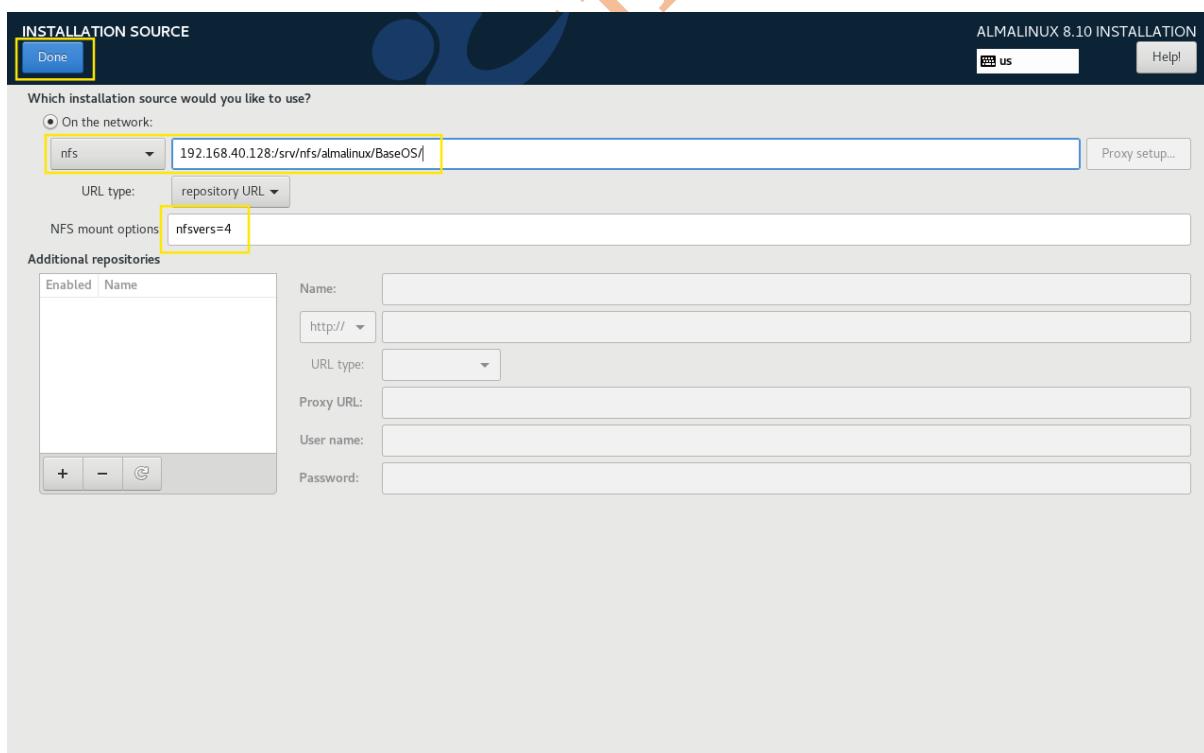


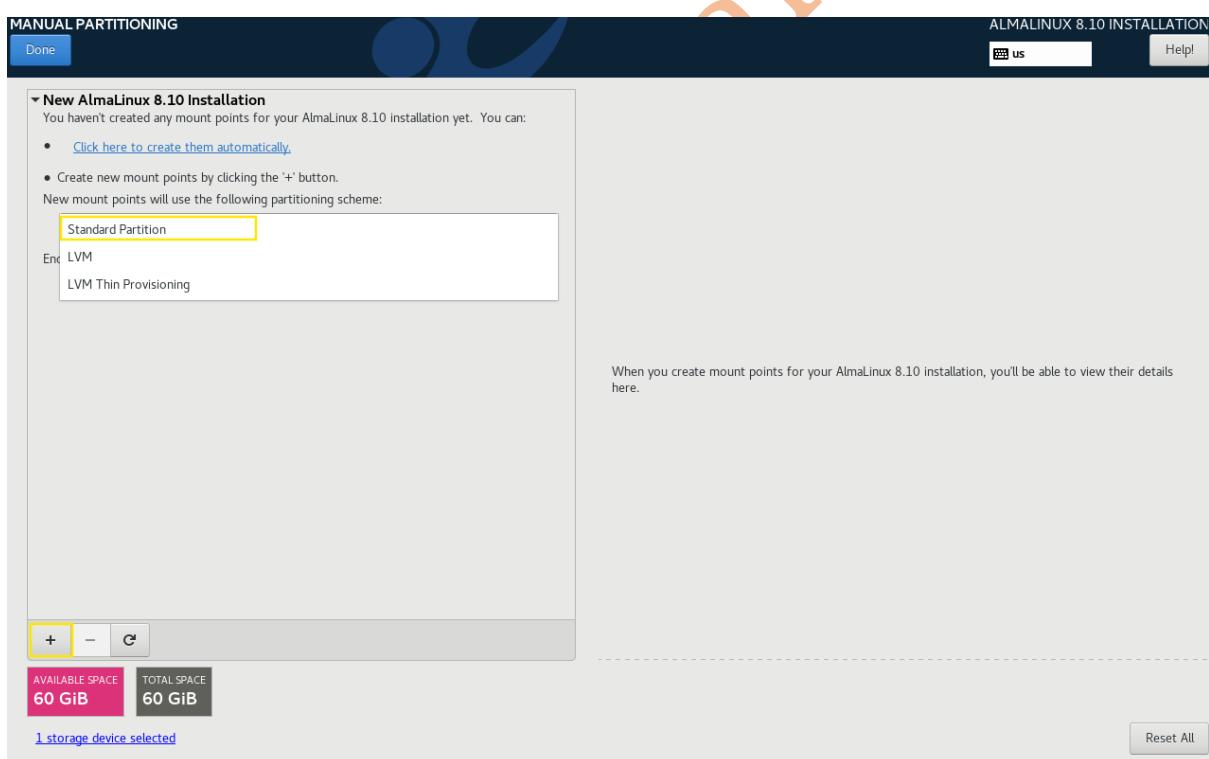
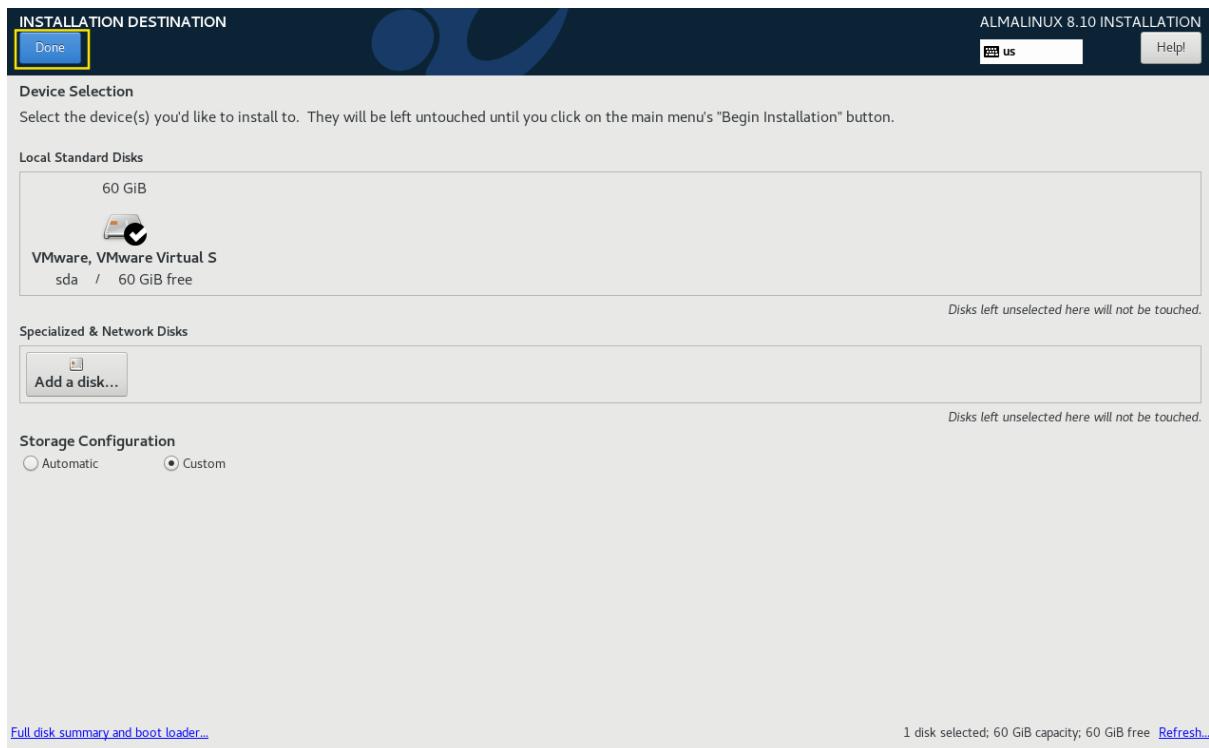
As shown in the screenshot attached below, installer had been started.





I had chosen the NFS URL as **192.168.40.128:/srv/nfs/almalinux/BaseOS/** as can be shown in the screenshot attached below. As I was interested to install Alma Linux 8 using NFS Share for Server Mode here.





MANUAL PARTITIONING

ALMALINUX 8.10 INSTALLATION

Done **Help!**

New AlmaLinux 8.10 Installation

You haven't created any mount points for your AlmaLinux 8.10 installation yet. You can:

- Click here to create them automatically.
- Create new mount points by clicking the '+' button.

New mount points will use the following partitioning scheme:

Standard Partition

Encrypt automatically created mount points by default:

Encrypt my data.

When you create mount points for your AlmaLinux 8.10 installation, you'll be able to view their details here.

+ - C

AVAILABLE SPACE **60 GiB** TOTAL SPACE **60 GiB**

1 storage device selected

Reset All

MANUAL PARTITIONING

ALMALINUX 8.10 INSTALLATION

Done **Help!**

New AlmaLinux 8.10 Installation

You haven't created any mount points for your AlmaLinux 8.10 installation yet. You can:

- Click here to create them automatically.
- Create new mount points by clicking the '+' button.

New mount points will use the following partitioning scheme:

Standard Partition

Encrypt automatically created mount points by default:

Encrypt my data.

ADD A NEW MOUNT POINT

More customization options are available after creating the mount point below.

Mount Point: /

Desired Capacity: 55GiB

Cancel **Add mount point**

+

AVAILABLE SPACE **60 GiB** TOTAL SPACE **60 GiB**

1 storage device selected

Reset All

MANUAL PARTITIONING

ALMALINUX 8.10 INSTALLATION

sda1

Mount Point: / **Device(s):** VMware, VMware Virtual S (sda)

Desired Capacity: 55 GiB **Modify...**

Device Type: Standard Partition Encrypt

File System: xfs Reformat

Label: **Name:** sda1

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

AVAILABLE SPACE 5 GiB **TOTAL SPACE** 60 GiB

1 storage device selected

Update Settings **Reset All**

MANUAL PARTITIONING

ALMALINUX 8.10 INSTALLATION

sda1

Mount Point: /boot **Device(s):** VMware, VMware Virtual S (sda)

Desired Capacity: 1024 MiB **Modify...**

Device Type: Standard Partition Encrypt

File System: xfs Reformat

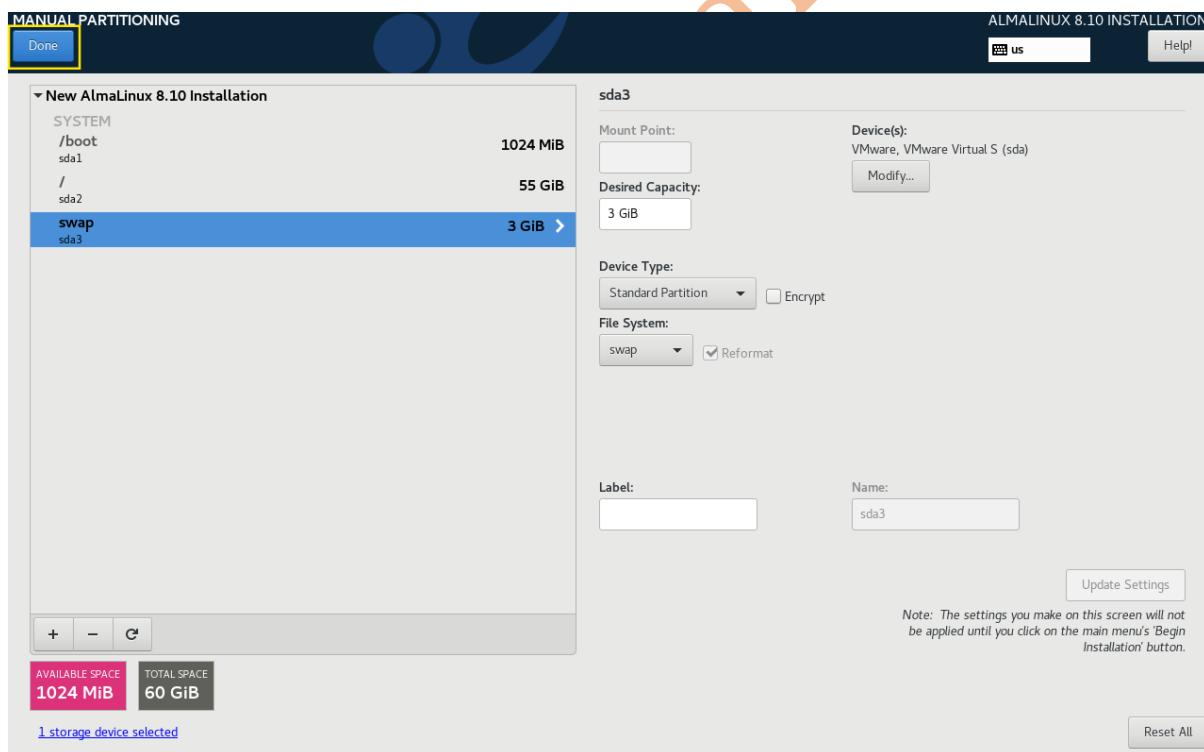
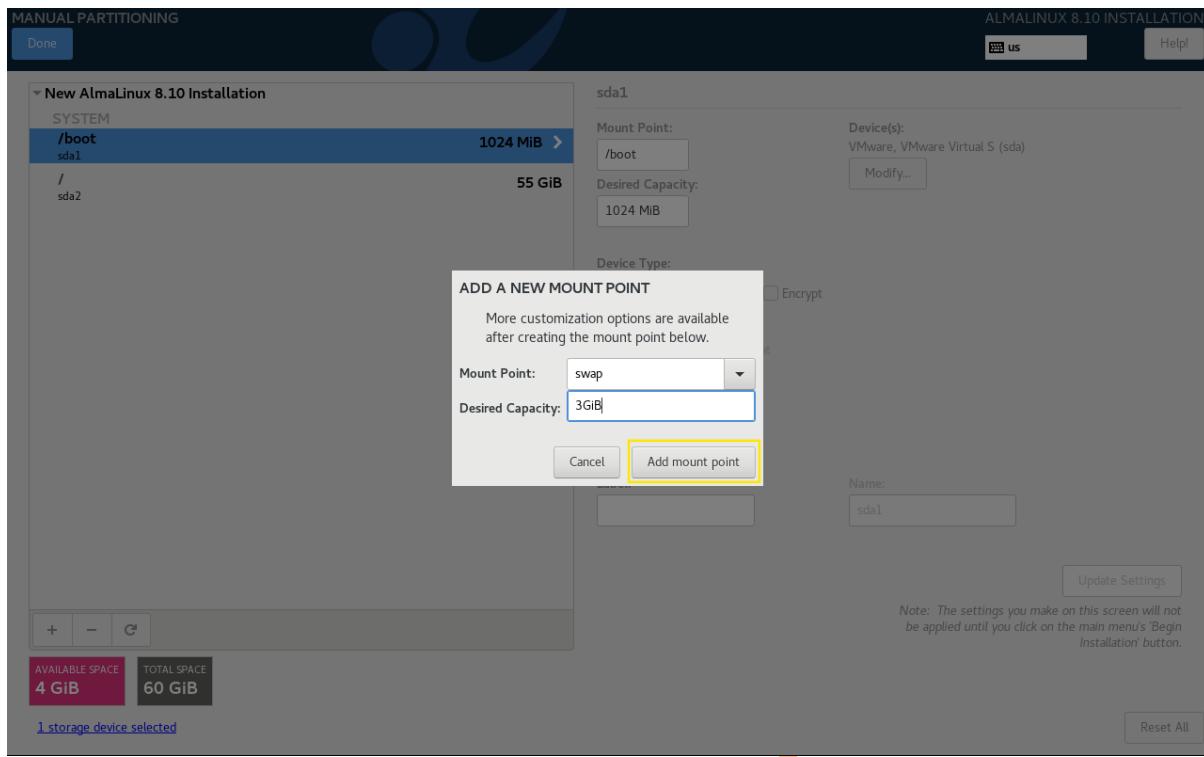
Label: **Name:** sda1

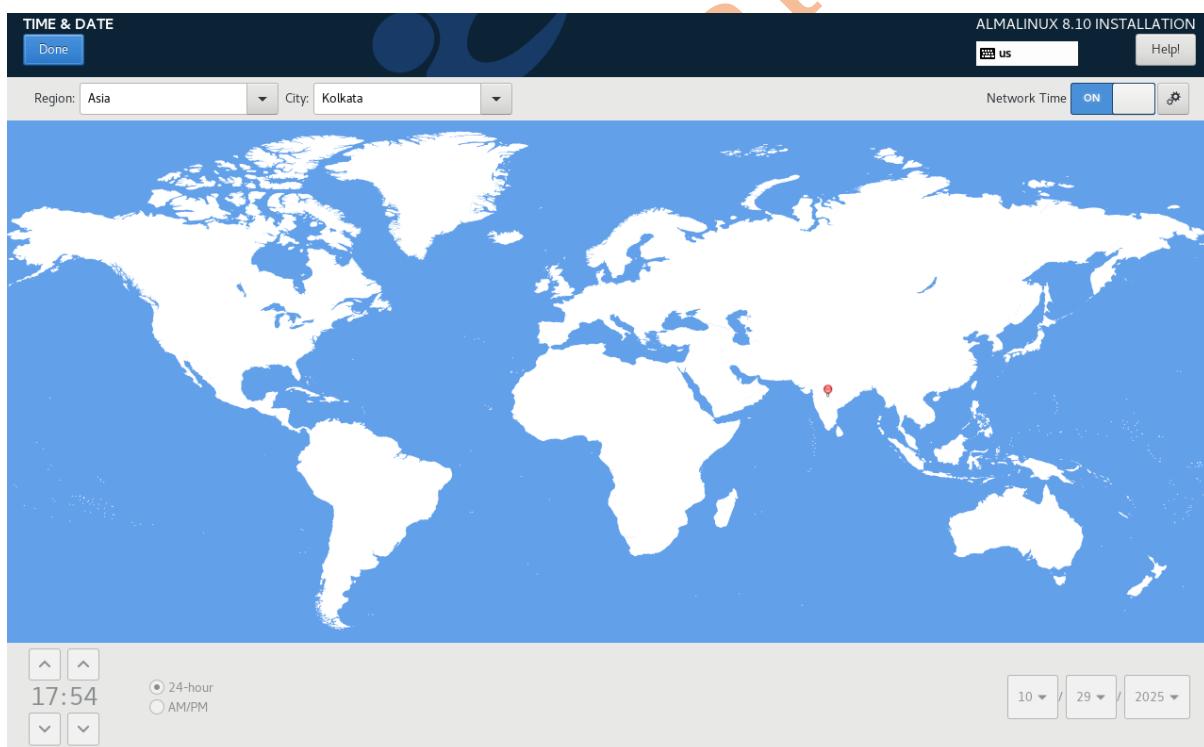
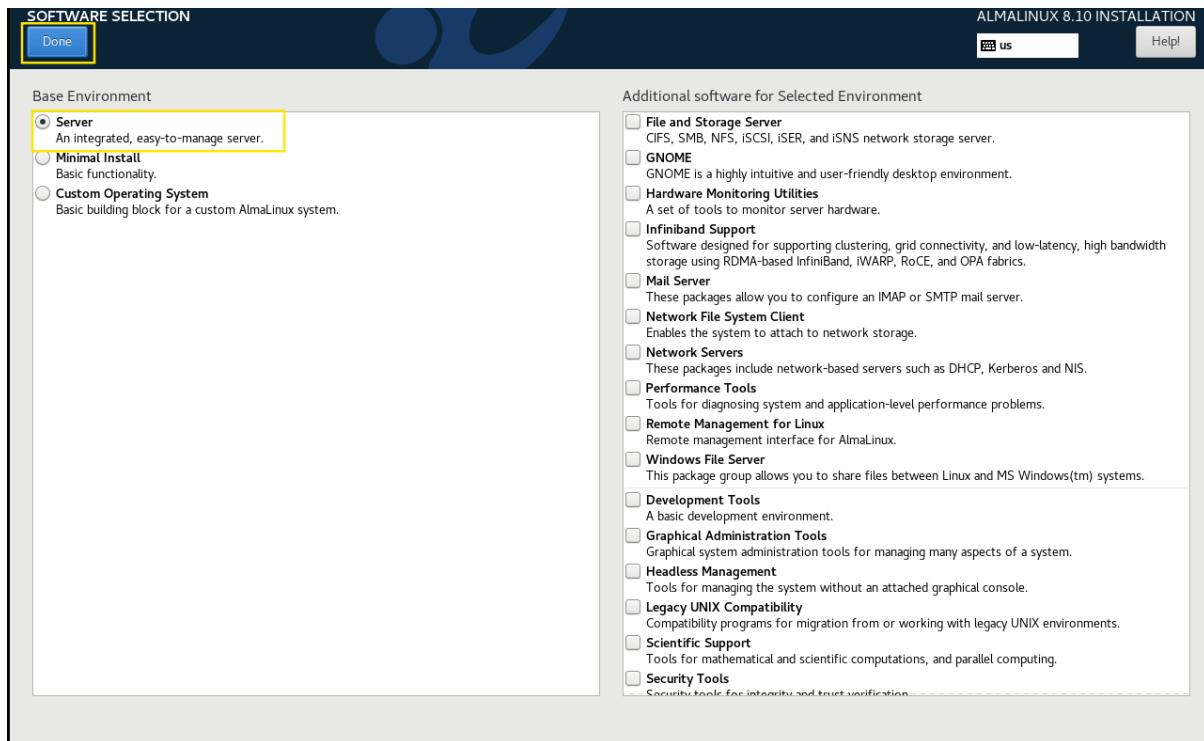
Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

AVAILABLE SPACE 4 GiB **TOTAL SPACE** 60 GiB

1 storage device selected

Update Settings **Reset All**





NETWORK & HOST NAME

ALMALINUX 8.10 INSTALLATION

Ethernet (ens33)
Intel Corporation 82545EM Gigabit Ethernet Controller (Copper) (PRO/1000 MT Single Port Adapter)

Ethernet (ens33)
Connected
Hardware Address 00:0C:29:2F:A6:CB
Speed 1000 Mb/s
IP Address 192.168.40.187/24
Default Route 192.168.40.1
DNS 192.168.40.2

ON

Done

Host Name: localhost.localdomain **Apply**

Configure...

Current host name: localhost.localdomain

ROOT PASSWORD

ALMALINUX 8.10 INSTALLATION

Done

The root account is used for administering the system. Enter a password for the root user.

Root Password: **Too short**

Confirm:

The password is too short. You will have to press Done twice to confirm it.

CREATE USER

Done

ALMALINUX 8.10 INSTALLATION

us Help!

Full name: [REDACTED]

User name: [REDACTED]

Make this user administrator

Require a password to use this account

Password: [REDACTED] Too short

Confirm password: [REDACTED]

Advanced...

⚠ The password is too short. You will have to press Done twice to confirm it.

INSTALLATION SUMMARY

ALMALINUX 8.10 INSTALLATION

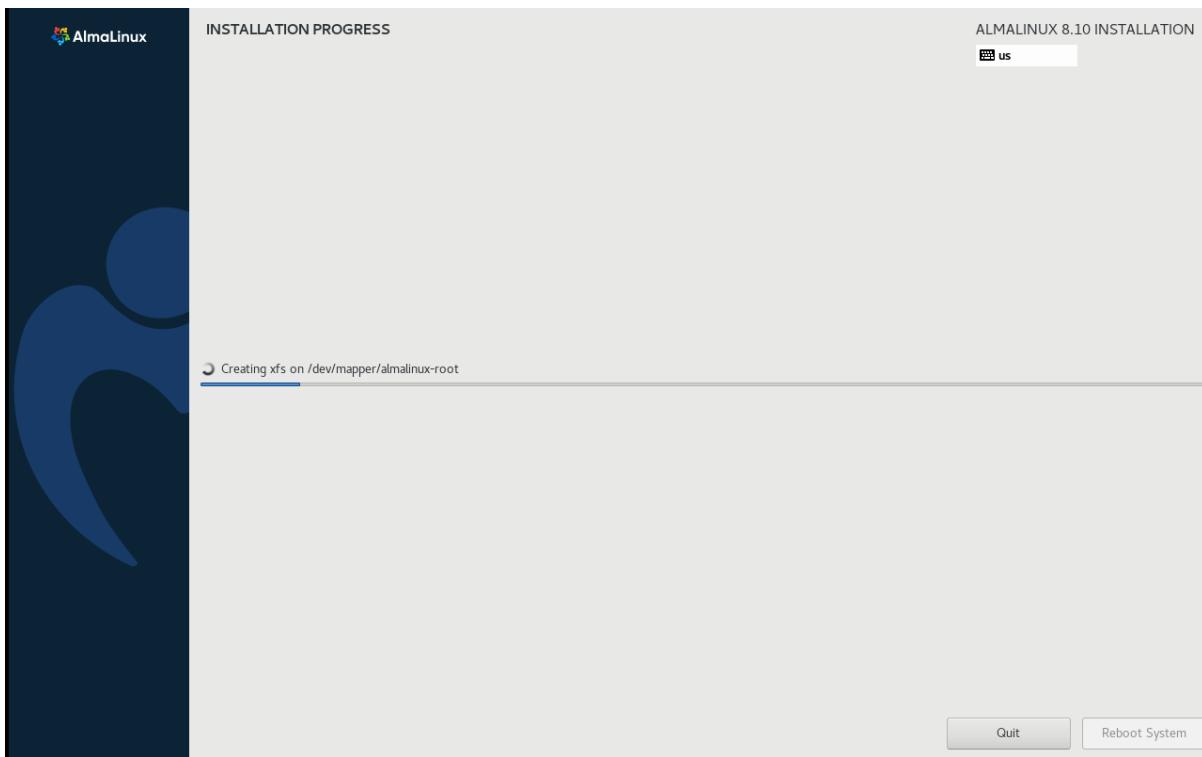
us Help!

LOCALIZATION	SOFTWARE	SYSTEM
<ul style="list-style-type: none"> Keyboard English (US) Language Support English (United States) Time & Date Asia/Kolkata timezone 	<ul style="list-style-type: none"> Installation Source NFS server nfs:nfsvers=4:1...malinuv/BaseOS/ Software Selection Server 	<ul style="list-style-type: none"> Installation Destination Automatic partitioning selected KDUMP Kdump is enabled Network & Host Name Wired (ens3) connected
USER SETTINGS		
<ul style="list-style-type: none"> Root Password Root password is set User Creation Administrator ritesh will be created 		

Begin Installation

We won't touch your disks until you click 'Begin Installation'.

As shown in the screenshot attached below, installation had been started.



Now you need to reboot the system as shown in the screenshot attached below.



```

AlmaLinux 8.10 (Cerulean Leopard)
Kernel 4.18.0-553.el8_10.x86_64 on an x86_64
Activate the web console with: systemctl enable --now cockpit.socket

localhost login: [REDACTED]
Password: [REDACTED]@localhost ~]$ sudo -i

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

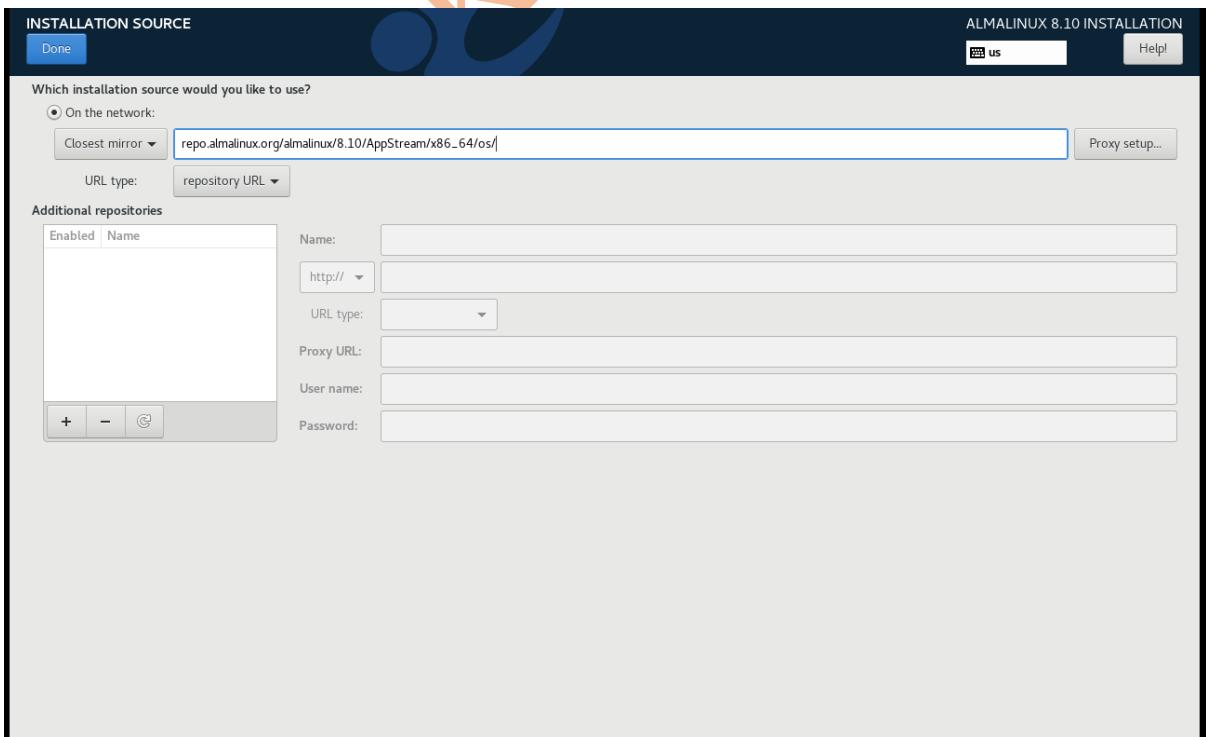
#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for [REDACTED]:
[root@localhost ~]#

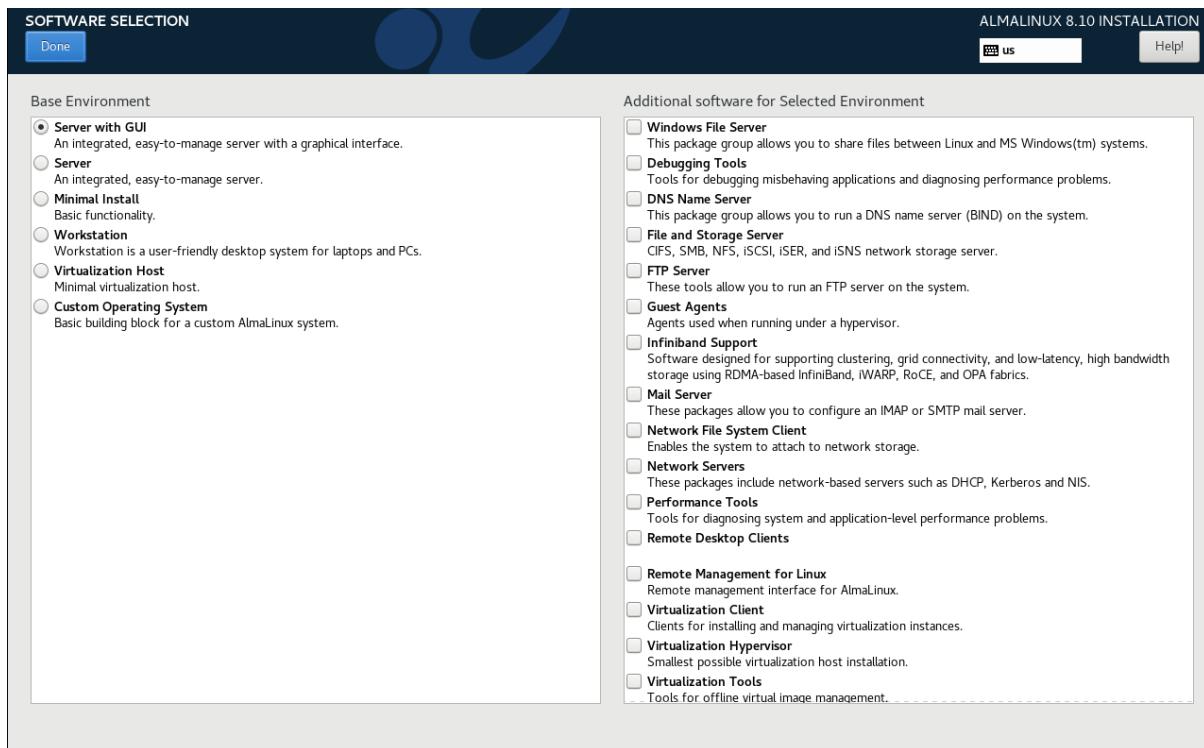
```

Finally, you can also login into this server using PuTTY/MobaXterm or from your Linux Machine using the IP Address of this Linux Server.

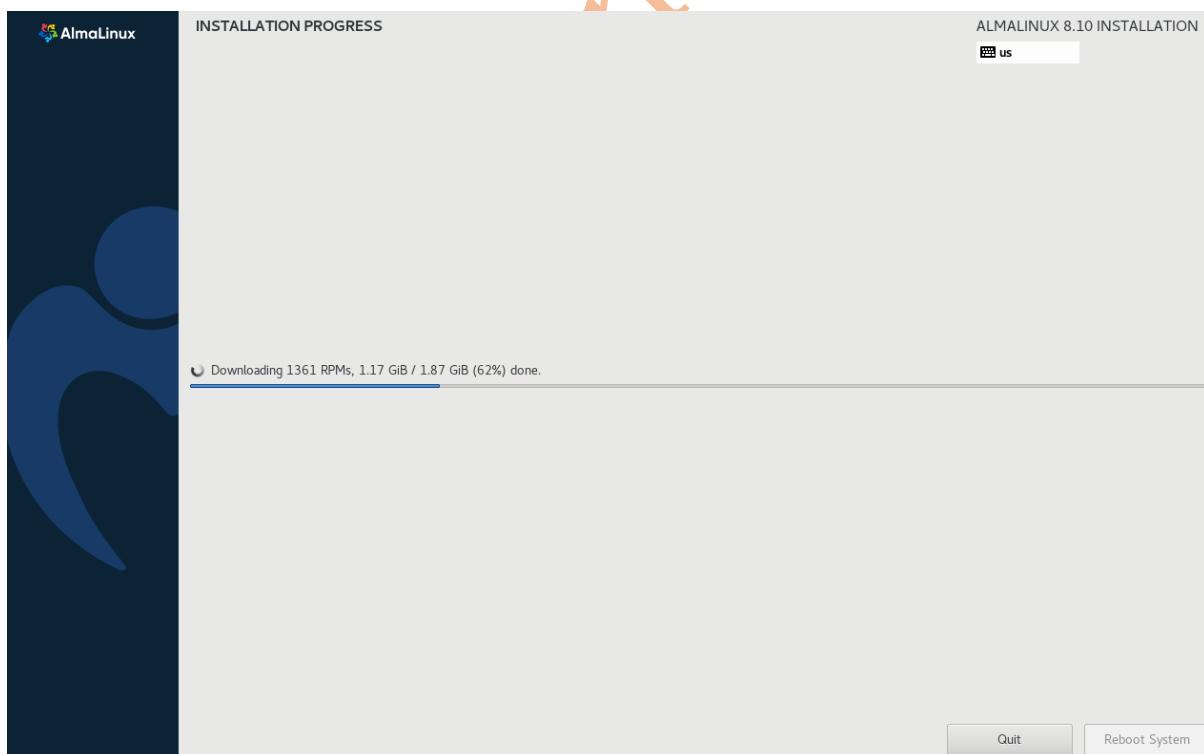
You can use Closest Mirror https://repo.almalinux.org/almalinux/8.10/AppStream/x86_64/os/ while booting a Linux machine as shown in the screenshot attached below. The URL provided here to boot the Almalinux machine is safe to use as it is the official repository for Almalinux. However, it is safe to use Closest Mirror <https://repo.almalinux.org> but it completely depends on your organisation to use it or not.



Before using Closest Mirror make sure Network & Host Ethernet should be in connected mode.



Further steps you can follow as I explained earlier.

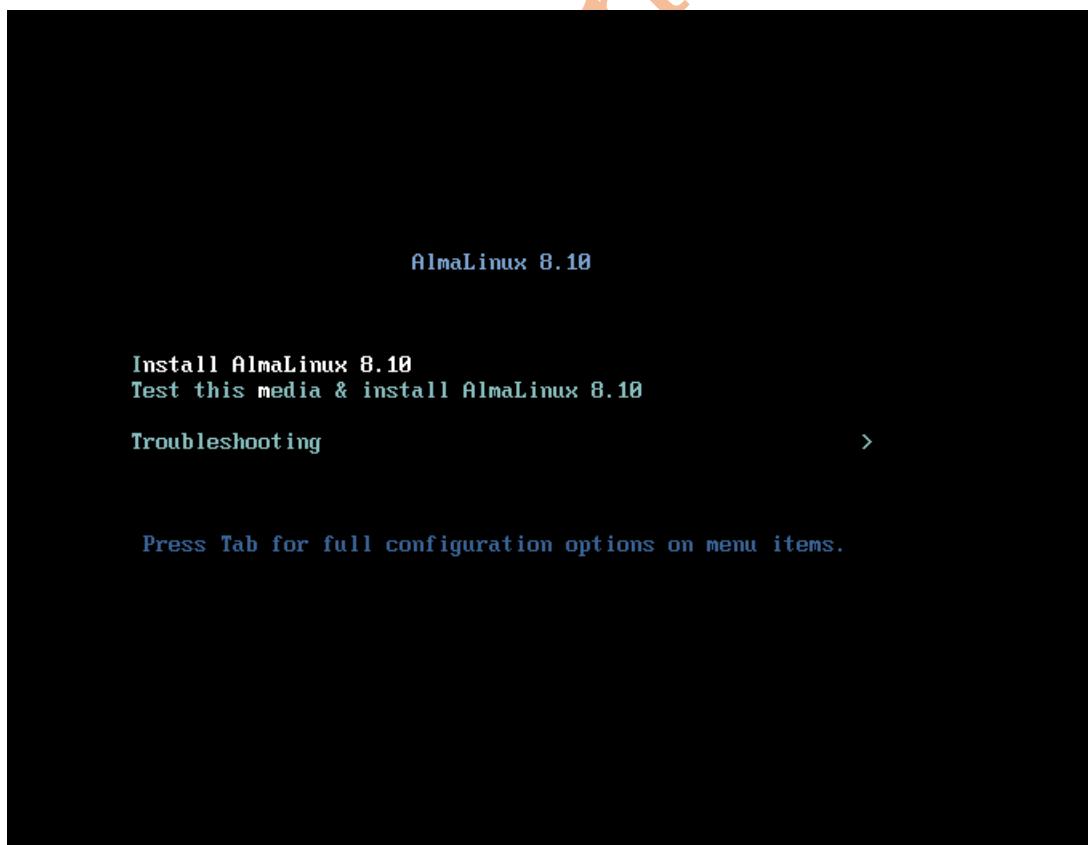


2. Boot from a USB Flash Drive (bootable/multi-boot pen drive)

To boot a Linux machine using a USB Flash Drive (multi boot pen drive). You need first a multi boot pen drive (or bootable pen drive) then plugin-in the pen drive into the slot of your machine then power on your machine and start pressing the **esc** button and a start up menu will come on the screen here it will be mentioned which key to press for Boot Menu or BIOS Menu. To bring the Boot Menu usually F9 Key will be pressed but it varies with different company computers. For present case I consider by pressing F9 key Boot Menu will come and then select the USB Flash Drive Option and the Operating System (Alma Linux 8).



RiteshR



Now Installer will come then follow the procedure as I mentioned above to proceed further with Almalinux OS installation (While installation make sure you select the correct disk and not the Flash Drive). After Installation make sure to enable the Secure Boot Option from BIOS.

I used Ventoy to make the USB Flash Drive (Pen drive) multi boot as shown in the screenshot attached below. You can the link <https://www.ventoy.net/en/download.html> to download Ventoy.

```
[root@localhost ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda        8:0    0   50G  0 disk
└─sda1     8:1    0   1G  0 part /boot
  └─sda2     8:2    0   49G  0 part
    ├─almalinux-root 253:0    0 45.1G  0 lvm  /
    └─almalinux-swap 253:1    0  3.9G  0 lvm  [SWAP]
sdb        8:16   1  57.8G  0 disk
sr0       11:0   1 1024M  0 rom
```

```
[root@localhost ventoy-1.1.07]# ./Ventyo2Disk.sh -i /dev/sdb
*****
Ventoy: 1.1.07 x86_64
longpanda admin@ventoy.net
https://www.ventoy.net
*****

Disk : /dev/sdb
Model: HP USB Flash Drive (scsi)
Size : 57 GB
Style: MBR

Attention:
You will install Ventoy to /dev/sdb.
All the data on the disk /dev/sdb will be lost!!!

Continue? (y/n) y

All the data on the disk /dev/sdb will be lost!!!
Double-check. Continue? (y/n) y

Create partitions on /dev/sdb by parted in MBR style ...
Done
Wait for partitions ...
partition exist OK
create efi fat fs /dev/sdb2 ...
mkfs.fat 4.1 (2017-01-24)
success
Wait for partitions $vPART1 and $vPART2 ...
/dev/sdb1 exist OK
/dev/sdb2 exist OK
/dev/sdb1 NOT exist
writing data to disk ...
sync data ...
esp partition processing ...

Install Ventoy to /dev/sdb successfully finished.

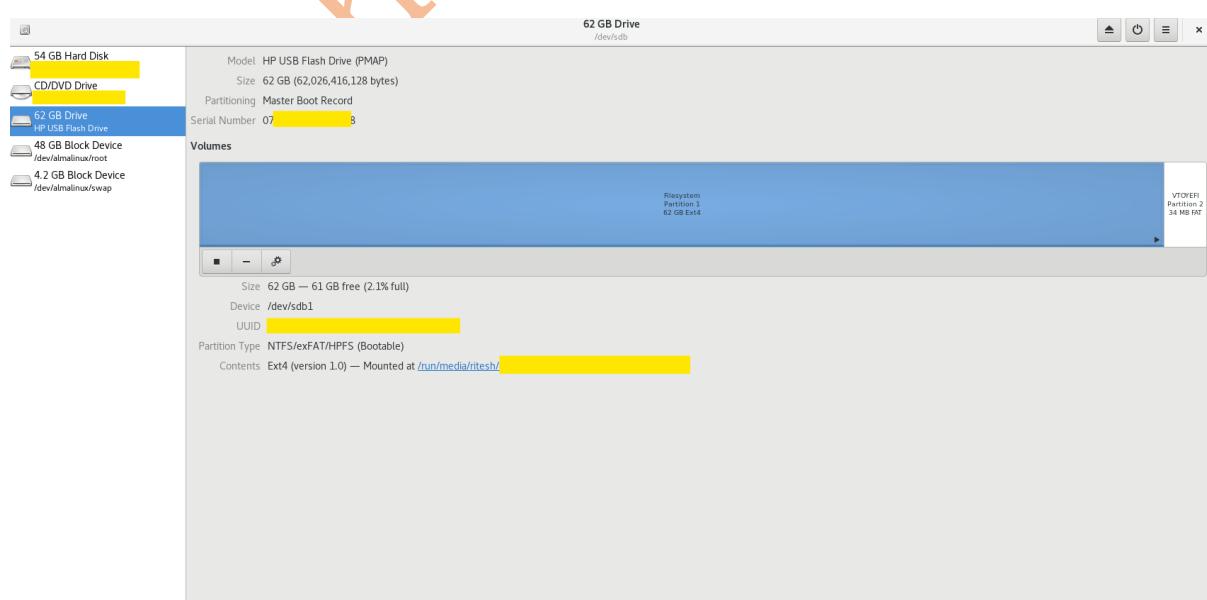
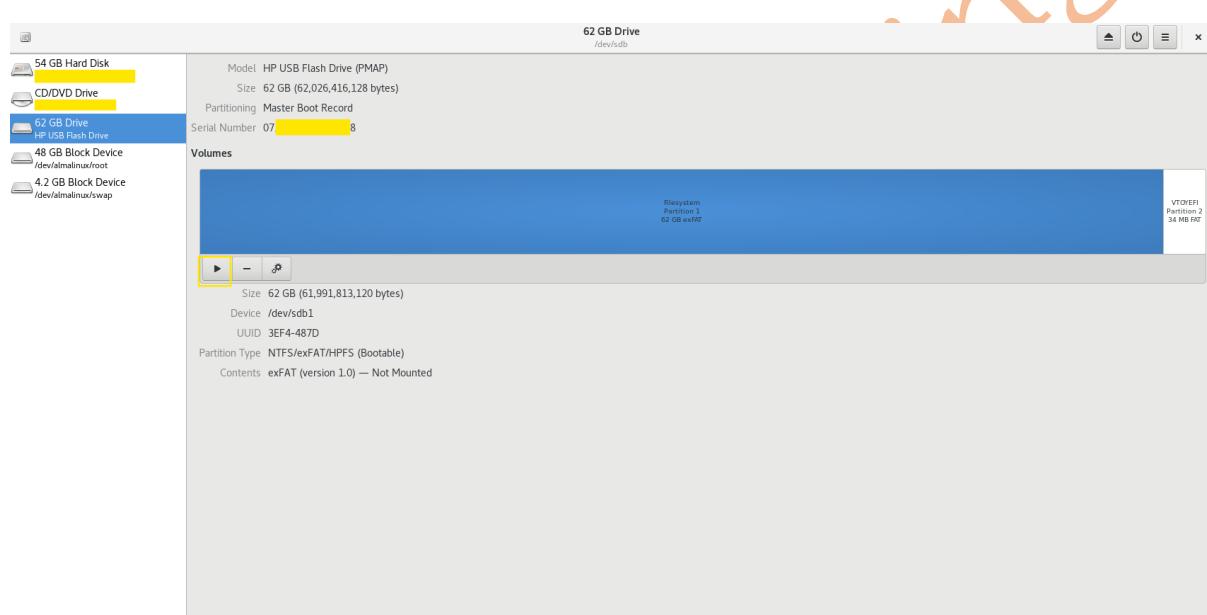
[root@localhost ventoy-1.1.07]# pwd
/home/ritesh/Downloads/ventoy-1.1.07-linux/ventoy-1.1.07
```

Then mount the USB Flash Drive as shown in the screenshot attached below.

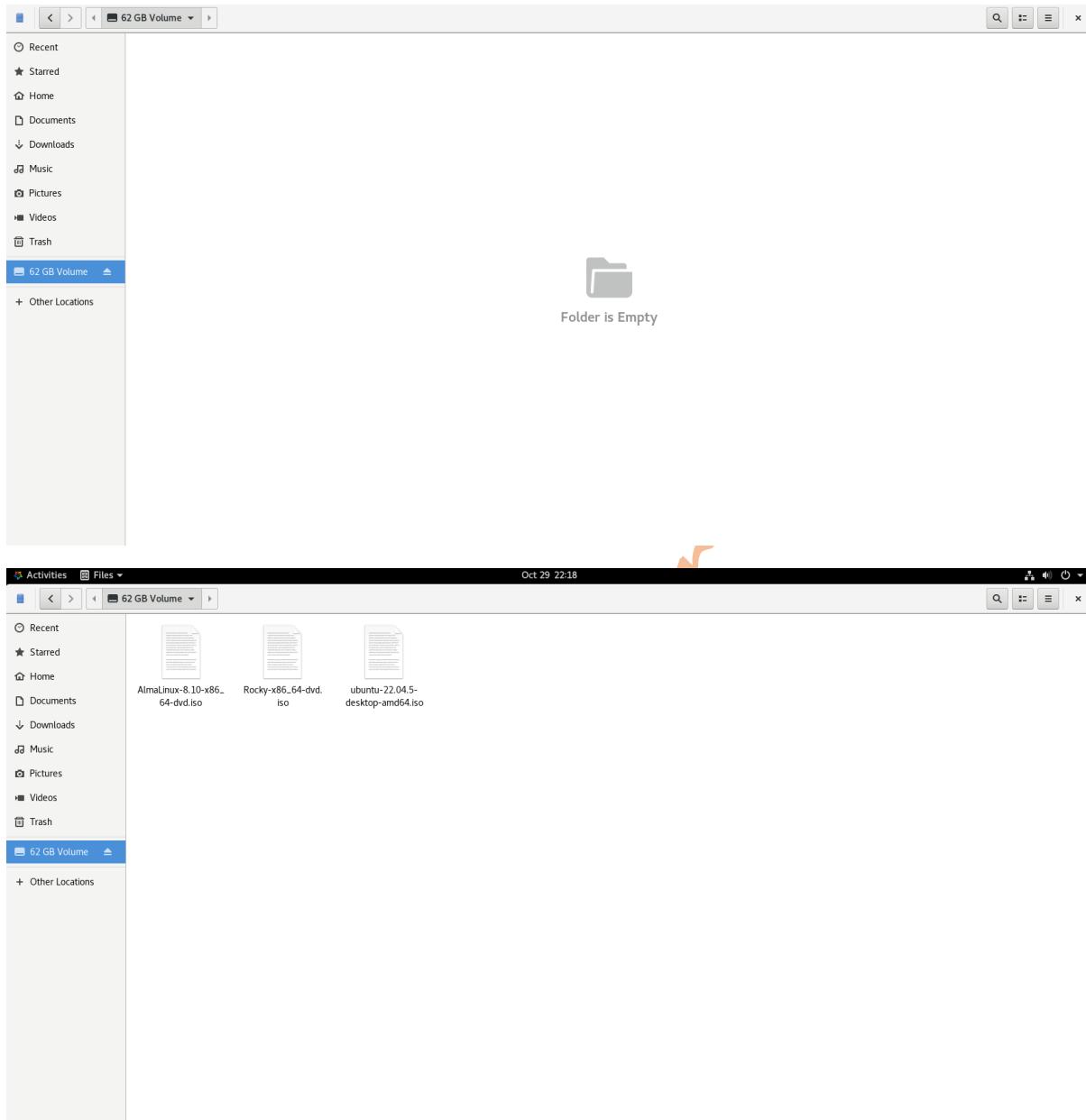
```
[root@localhost ~]# mkfs.ext4 /dev/sdb1
mke2fs 1.45.6 (20-Mar-2020)
/dev/sdb1 contains a exfat file system
Proceed anyway? (y,N) y
Creating filesystem with 15134720 4k blocks and 3784704 inodes
Filesystem UUID: 4d1c78b4-c609-4bbc-9f66-60fc99f57782
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (65536 blocks):

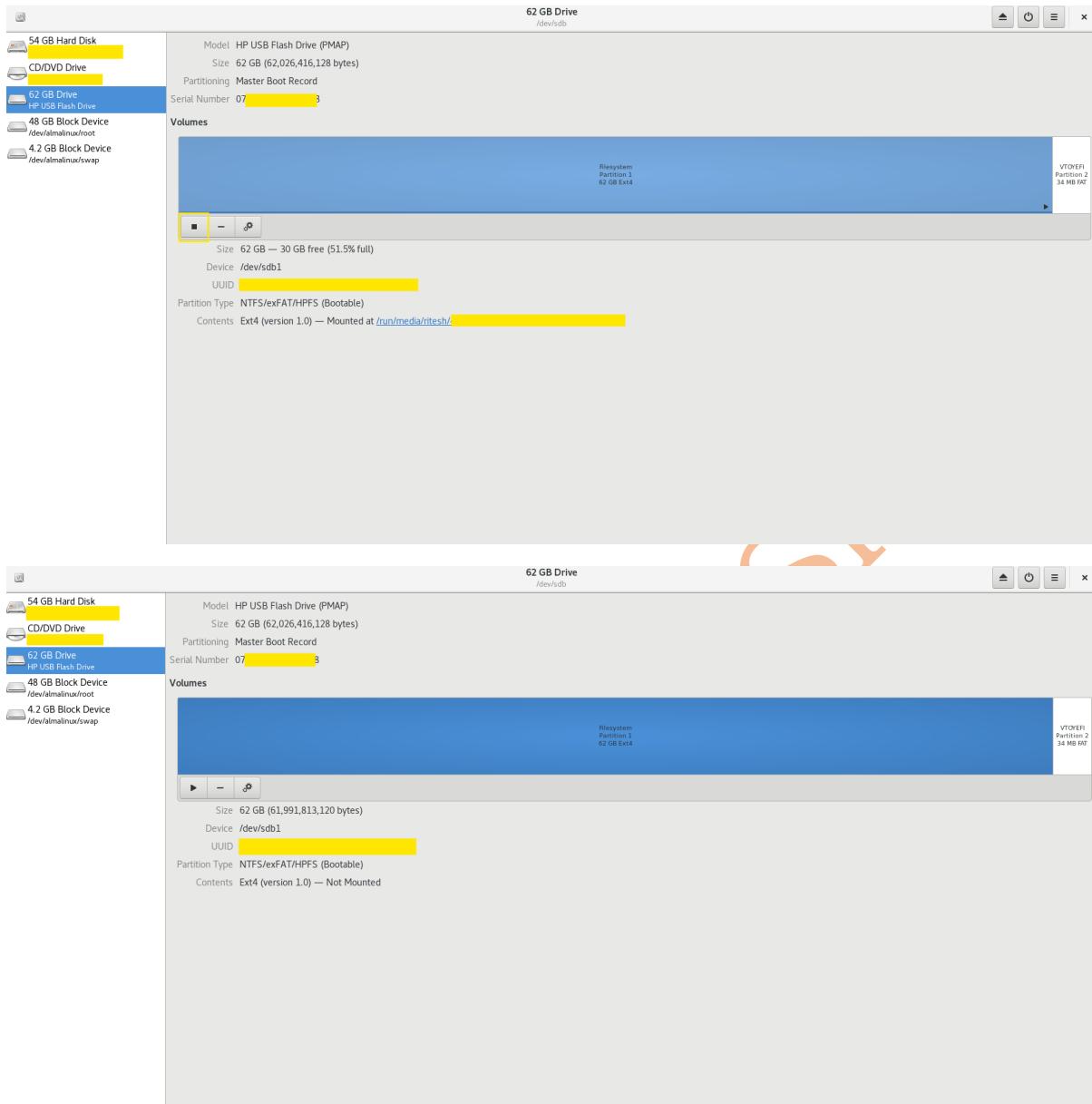
done
Writing superblocks and filesystem accounting information: done
```



Then drag and drop the Operating system ISO files in the USB Flash Directory as shown in the screenshot attached below.



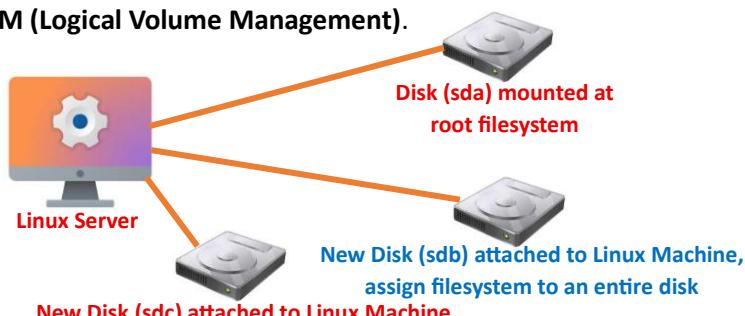
Unmount the /dev/sdb1 partition of the USB Flash Drive



Your multi-boot USB Flash Drive (multi-boot pen drive) is ready; you can use it to boot the Linux System as discussed above.

Disk Partition in Linux

In Linux **disk partition** is a logical section of a physical storage drive (like an HDD or SSD) which is treated as a separate disk. Under this topic I will explain you how to do the disk partition and how to create the **LVM (Logical Volume Management)**.



As shown in the diagram above in this example there are three disks sda, sdb and sdc attached to the Linux machine. Disk **sda** (50GB) contain the Almalinux 9 Operating System and for disk **sdb** (10GB) I will assign ext4 filesystem to this disk and mount it to a directory. For disk **sdc** (10GB) I will create the partition and then LVM, finally mount them to a directory.

```
[root@therema-server ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda        8:0    0   50G  0 disk
└─sda1     8:1    0    1G  0 part /boot
└─sda2     8:2    0   49G  0 part
  ├─almalinux-root 253:0    0   44G  0 lvm  /
  └─almalinux-swap 253:1    0    5G  0 lvm  [SWAP]
sdb        8:16   0   10G  0 disk
sdc        8:32   0   10G  0 disk
sr0       11:0   1  11.4G 0 rom

[root@therema-server ~]# mkfs.ext4 /dev/sdb
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 2621440 4k blocks and 655360 inodes
Filesystem UUID: [REDACTED]
Superblock backups stored on blocks:
[REDACTED]

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

[root@therema-server ~]# mkdir /dexter
[root@therema-server ~]# vim /etc/fstab
[root@therema-server ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Thu Oct 30 10:11:07 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/almalinux-root /          xfs    defaults      0 0
UUID=[REDACTED] /boot              xfs    defaults      0 0
/dev/mapper/almalinux-swap none      swap   defaults      0 0

/dev/sdb /dexter ext4 defaults 0 0
[root@therema-server ~]# mount -a

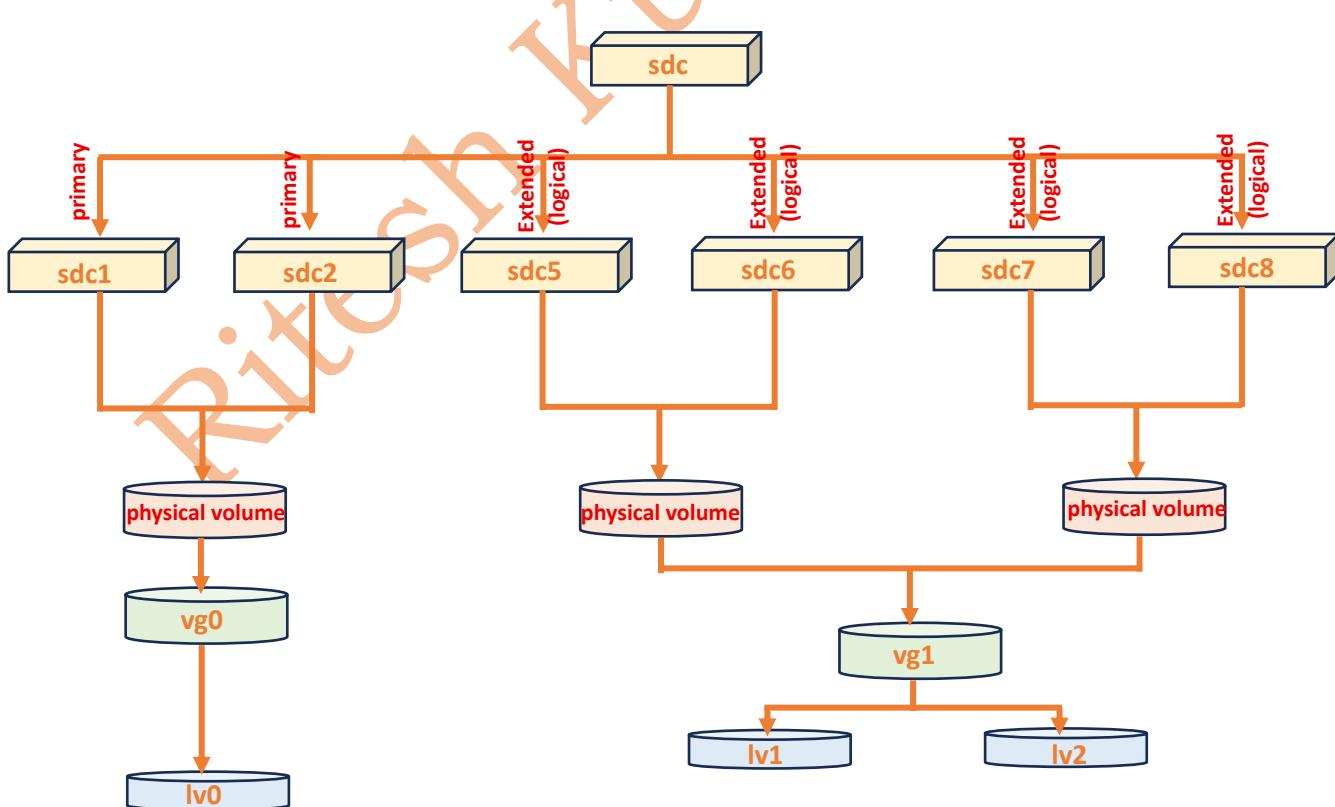
[root@therema-server ~]# df -hT
Filesystem      Type  Size  Used  Avail Use% Mounted on
devtmpfs        devtmpfs 4.0M   0    4.0M  0% /dev
tmpfs           tmpfs   1.8G   0    1.8G  0% /dev/shm
tmpfs           tmpfs   725M  9.6M  716M  2% /run
/dev/mapper/almalinux-root xfs    44G  4.7G  40G  11% /
/dev/sda1        xfs    960M 366M  595M  39% /boot
tmpfs           tmpfs   363M  52K   363M  1% /run/user/42
tmpfs           tmpfs   363M  36K   363M  1% /run/user/1000
/dev/sdb         ext4   9.8G  24K   9.3G  1% /dexter
```

Here disk partitioning scheme was used as MBR (Master Boot Record), and hence you can create **four primary partitions or three primary and one extended partition** (extended partition further can contain several logical partitions).

```
[root@therema-server ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda        8:0    0   50G  0 disk
└─sda1     8:1    0   1G  0 part /boot
└─sda2     8:2    0   49G  0 part
  └─almalinux-root 253:0  0   44G  0 lvm  /
    └─almalinux-swap 253:1  0   5G  0 lvm  [SWAP]
sdb        8:16   0   10G 0 disk /dexter
sdc        8:32   0   10G 0 disk
sr0       11:0    1 11.4G 0 rom

[root@therema-server ~]# fdisk -l /dev/sdc
Disk /dev/sdc: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Here for **sdc** I will create two primary and one extended partition, under extended partition I had created four logical partitions and finally the two LVM (using two logical partitions each in one LVM). Its details are as shown in the architecture diagram drawn below. With two primary partitions I will create one LVM.



```
[root@therema-server ~]# fdisk /dev/sdc

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x1ccad1ba.

Command (m for help): n
Partition type
  p  primary (0 primary, 0 extended, 4 free)
  e  extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-20971519, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-20971519, default 20971519): +2G

Created a new partition 1 of type 'Linux' and of size 2 GiB.

Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): L

00 Empty          24 NEC DOS      81 Minix / old Lin  bf Solaris
01 FAT12         27 Hidden NTFS Win 82 Linux swap / So c1 DRDOS/sec (FAT-
02 XENIX root    39 Plan 9       83 Linux          c4 DRDOS/sec (FAT-
03 XENIX usr     3c PartitionMagic 84 OS/2 hidden or c6 DRDOS/sec (FAT-
04 FAT16 <32M   40 Venix 80286   85 Linux extended c7 Syrinx
05 Extended      41 PPC PReP Boot  86 NTFS volume set da Non-FS data
06 FAT16          42 SFS          87 NTFS volume set db CP/M / CTOS / .
07 HPFS/NTFS/exFAT 4d QNX4.x     88 Linux plaintext de Dell Utility
08 AIX            4e QNX4.x 2nd part 8e Linux LVM      df BootIt
09 AIX bootable   4f QNX4.x 3rd part 93 Amoeba        e1 DOS access
0a OS/2 Boot Manag 50 OnTrack DM   94 Amoeba BBT    e3 DOS R/O
0b W95 FAT32     51 OnTrack DM6 Aux 9f BSD/OS       e4 SpeedStor
0c W95 FAT32 (LBA) 52 CP/M        a0 IBM Thinkpad hi ea Linux extended
0e W95 FAT16 (LBA) 53 OnTrack DM6 Aux a5 FreeBSD      eb BeOS fs
0f W95 Ext'd (LBA) 54 OnTrackDM6   a6 OpenBSD      ee GPT
```

Hex code or alias (type L to list all): 8e
 Changed type of partition 'Linux' to 'Linux LVM'.

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

```
[root@therema-server ~]# fdisk -l /dev/sdc
Disk /dev/sdc: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x1ccad1ba
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdc1		2048	4196351	4194304	2G	8e	Linux LVM

Ritesh Kumar Singh || Email Address: - riteshkumarsingh9559@gmail.com || LinkedIn: - <https://www.linkedin.com/in/ritesh-kumar-singh-41113128b/> || GitHub: - <https://github.com/singhrithesh85>

```
[root@therema-server ~]# fdisk /dev/sdc

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): n
Partition type
  p  primary (1 primary, 0 extended, 3 free)
  e  extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2):
First sector (4196352-20971519, default 4196352):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (4196352-20971519, default 20971519): +2G

Created a new partition 2 of type 'Linux' and of size 2 GiB.

Command (m for help): t
Partition number (1,2, default 2):
Hex code or alias (type L to list all): 8e

Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

```
[root@therema-server ~]# fdisk -l /dev/sdc
Disk /dev/sdc: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x1ccad1ba

Device      Boot   Start     End  Sectors  Size Id Type
/dev/sdc1           2048 4196351 4194304    2G 8e Linux LVM
/dev/sdc2        4196352 8390655 4194304    2G 8e Linux LVM
```

```
[root@therema-server ~]# fdisk /dev/sdc

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): n
Partition type
  p  primary (2 primary, 0 extended, 2 free)
  e  extended (container for logical partitions)
Select (default p): e
Partition number (3,4, default 3):
First sector (8390656-20971519, default 8390656):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (8390656-20971519, default 20971519): +5G

Created a new partition 3 of type 'Extended' and of size 5 GiB.

Command (m for help): n
Partition type
  p  primary (2 primary, 1 extended, 1 free)
  l  logical (numbered from 5)
Select (default p): l

Adding logical partition 5
First sector (8392704-18876415, default 8392704):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (8392704-18876415, default 18876415): +1G

Created a new partition 5 of type 'Linux' and of size 1 GiB.

Command (m for help): n
Partition type
  p  primary (2 primary, 1 extended, 1 free)
  l  logical (numbered from 5)
Select (default p): l

Adding logical partition 6
First sector (10491904-18876415, default 10491904):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (10491904-18876415, default 18876415): +1G

Created a new partition 6 of type 'Linux' and of size 1 GiB.

Command (m for help): n
Partition type
  p  primary (2 primary, 1 extended, 1 free)
  l  logical (numbered from 5)
Select (default p): l

Adding logical partition 7
First sector (12591104-18876415, default 12591104):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (12591104-18876415, default 18876415): +1G

Created a new partition 7 of type 'Linux' and of size 1 GiB.

Command (m for help): n
Partition type
  p  primary (2 primary, 1 extended, 1 free)
  l  logical (numbered from 5)
Select (default p): l

Adding logical partition 8
First sector (14690304-18876415, default 14690304):
```

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (14690304-18876415, default 18876415): +1.5G
Created a new partition 8 of type 'Linux' and of size 1.5 GiB.

Command (m for help): t
Partition number (1-3,5-8, default 8): 5
Hex code or alias (type L to list all): 8e

Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): t
Partition number (1-3,5-8, default 8): 6
Hex code or alias (type L to list all): 8e

Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): t
Partition number (1-3,5-8, default 8): 7
Hex code or alias (type L to list all): 8e

Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): t
Partition number (1-3,5-8, default 8):
Hex code or alias (type L to list all): 8e

Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

```
[root@therema-server ~]# fdisk -l /dev/sdc
Disk /dev/sdc: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x1ccad1ba

Device      Boot   Start     End   Sectors  Size Id Type
/dev/sdc1           2048  4196351  4194304    2G 8e Linux LVM
/dev/sdc2        4196352  8390655  4194304    2G 8e Linux LVM
/dev/sdc3        8390656 18876415 10485760    5G  5 Extended
/dev/sdc5        8392704 10489855  2097152    1G 8e Linux LVM
/dev/sdc6       10491904 12589055  2097152    1G 8e Linux LVM
/dev/sdc7       12591104 14688255  2097152    1G 8e Linux LVM
/dev/sdc8       14690304 17836031  3145728  1.5G 8e Linux LVM
```

```
[root@therema-server ~]# partprobe /dev/sdc
```

```
[root@therema-server ~]# pvcreate /dev/sdc1 /dev/sdc2
Physical volume "/dev/sdc1" successfully created.
Physical volume "/dev/sdc2" successfully created.
[root@therema-server ~]# pvs
PV          VG      Fmt Attr PSize  PFree
/dev/sda2  almalinux lvm2 a--  <49.00g    0
/dev/sdc1            lvm2 ---   2.00g 2.00g
/dev/sdc2            lvm2 ---   2.00g 2.00g
```

```
[root@therema-server ~]# vgcreate vg0 /dev/sdc1 /dev/sdc2
Volume group "vg0" successfully created
```

```
[root@therema-server ~]# vgdisplay
--- Volume group ---
VG Name           vg0
System ID
Format           lvm2
Metadata Areas   2
Metadata Sequence No 1
VG Access        read/write
VG Status        resizable
MAX LV
Cur LV
Open LV
Max PV
Cur PV
Act PV
VG Size         3.99 GiB
PE Size          4.00 MiB
Total PE        1022
Alloc PE / Size 0 / 0
Free PE / Size  1022 / 3.99 GiB
VG UUID
```

```
[root@therema-server ~]# lvcreate -L 1.5G -n lv0 vg0
Logical volume "lv0" created.
```

```
[root@therema-server ~]# lvdisplay
--- Logical volume ---
LV Path          /dev/vg0/lv0
LV Name          lv0
VG Name          vg0
LV UUID          ZN0J7y-FLvs-bt5X-fNLb-f7bg-09A0-Iit0gR
LV Write Access  read/write
LV Creation host, time therema-server, 2025-10-30 21:47:09 +0530
LV Status        available
# open           0
LV Size          1.50 GiB
Current LE       384
Segments         1
Allocation       inherit
Read ahead sectors auto
- currently set to 256
Block device     253:2
```

```
[root@therema-server ~]# mkswap /dev/vg0/lv0
Setting up swap space version 1, size = 1.5 GiB (1610608640 bytes)
no label, UUID=4fb61356-68e1-4c18-a42a-4634006c0a02
```

```
[root@therema-server ~]# mkswap /dev/vg0/lv0
Setting up swap space version 1, size = 1.5 GiB (1610608640 bytes)
no label, UUID=[REDACTED]
[root@therema-server ~]# vim /etc/fstab
[root@therema-server ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Thu [REDACTED] 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/almalinux-root /          xfs    defaults      0 0
UUID=[REDACTED] /boot              xfs    defaults      0 0
/dev/mapper/almalinux-swap none    swap   defaults      0 0

/dev/sdb /dexter ext4 defaults 0 0

/dev/vg0/lv0 swap swap defaults 0 0
[root@therema-server ~]# swapon -a
```

```
[root@therema-server ~]# pvcreate /dev/sdc5 /dev/sdc6
Physical volume "/dev/sdc5" successfully created.
Physical volume "/dev/sdc6" successfully created.
[root@therema-server ~]# pvcreate /dev/sdc7 /dev/sdc8
Physical volume "/dev/sdc7" successfully created.
Physical volume "/dev/sdc8" successfully created.
```

```
[root@therema-server ~]# pvdisplay
```

```
[root@therema-server ~]# pvs
  PV          VG      Fmt Attr PSize  PFree
  /dev/sda2   almalinux lvm2 a--  <49.00g    0
  /dev/sdc1   vg0      lvm2 a--  <2.00g  508.00m
  /dev/sdc2   vg0      lvm2 a--  <2.00g  <2.00g
  /dev/sdc5
  /dev/sdc6
  /dev/sdc7
  /dev/sdc8

[root@therema-server ~]# vgcreate vg1 /dev/sdc5 /dev/sdc6 /dev/sdc7 /dev/sdc8
  Volume group "vg1" successfully created

[root@therema-server ~]# vgs
  VG      #PV #LV #SN Attr   VSize   VFree
  almalinux  1   2   0 wz--n- <49.00g    0
  vg0        2   1   0 wz--n-   3.99g  2.49g
  vg1        4   0   0 wz--n-   4.48g  4.48g

[root@therema-server ~]# lvcreate -L 1.5G -n lv1 vg1
  Logical volume "lv1" created.
[root@therema-server ~]# lvs
  LV   VG      Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  root almalinux -wi-ao---- <44.00g
  swap almalinux -wi-ao----   5.00g
  lv0  vg0      -wi-ao----   1.50g
  lv1  vg1      -wi-a-----   1.50g
[root@therema-server ~]# lvcreate -L 1.5G -n lv2 vg1
  Logical volume "lv2" created.
[root@therema-server ~]# lvs
  LV   VG      Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  root almalinux -wi-ao---- <44.00g
  swap almalinux -wi-ao----   5.00g
  lv0  vg0      -wi-ao----   1.50g
  lv1  vg1      -wi-a-----   1.50g
  lv2  vg1      -wi-a-----   1.50g
```

```
[root@therema-server ~]# mkfs.ext4 /dev/vg1/lv1
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 393216 4k blocks and 98304 inodes
Filesystem UUID: 669f0ece-7933-4a74-ae4f-2dea12d886de
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

[root@therema-server ~]# mkfs.ext4 /dev/vg1/lv2
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 393216 4k blocks and 98304 inodes
Filesystem UUID: 09914630-fd42-4f5a-a7ea-299666625de9
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

[root@therema-server ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Thu Oct 30 10:11:07 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/almalinux-root /          xfs    defaults        0 0
UUID=[REDACTED] /boot                xfs    defaults        0 0
/dev/mapper/almalinux-swap none     swap   defaults        0 0

/dev/sdb /dexter ext4 defaults 0 0
/dev/vg0/lv0 swap swap defaults 0 0
/dev/vg1/lv1 /pablo/therema ext4 defaults 0 0
/dev/vg1/lv2 /pablo/mederma ext4 defaults 0 0
[root@therema-server ~]# mount -a
```

```
[root@therema-server ~]# df -hT
Filesystem      Type   Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs 4.0M    0  4.0M  0% /dev
tmpfs          tmpfs   1.8G    0  1.8G  0% /dev/shm
tmpfs          tmpfs   725M  9.8M  715M  2% /run
/dev/mapper/almalinux-root xfs    44G  4.7G  40G  11% /
/dev/sda1        xfs   960M 366M  595M  39% /boot
tmpfs          tmpfs   363M  96K  363M  1% /run/user/1000
/dev/sdb         ext4   9.8G  24K  9.3G  1% /dexter
/dev/sr0        iso9660 12G    0  100% /run/media/ritesh/AlmaLinux-9-6-x86_64-dvd
/dev/mapper/vg1-lv1  ext4   1.5G  24K  1.4G  1% /pablo/therema
/dev/mapper/vg1-lv2  ext4   1.5G  24K  1.4G  1% /pablo/mederma
```

Till now I had created three partitions in the disk **sdc**, two primary and one extended. Now I will create a new partition named as **/dev/sdc4** and will create physical volume then extend vg1 and finally lv1. After that I will extend the LVM lv0 and will shrink the LVM lv2.

Before creating a new partition in the disk **sdc**, unmount and swapoff all the existed partitions of the disk.

```
[root@therema-server ~]# swapoff /dev/vg0/lv0
[root@therema-server ~]# umount -l /dev/vg1/lv1
[root@therema-server ~]# umount -l /dev/vg1/lv2

[root@therema-server ~]# fdisk /dev/sdc

Command (m for help): n
Partition type
 p  primary (2 primary, 1 extended, 1 free)
 1  logical (numbered from 5)
Select (default p):

Using default response p.
Selected partition 4
First sector (18876416-20971519, default 18876416):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (18876416-20971519, default 20971519): +512M

Created a new partition 4 of type 'Linux' and of size 512 MiB.

Command (m for help): t
Partition number (1-8, default 8): 4
Hex code or alias (type L to list all): 8e

Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): w
The partition table has been altered.
Syncing disks.
```

```
[root@therema-server ~]# fdisk -l /dev/sdc
Disk /dev/sdc: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x18eed553

Device      Boot   Start     End   Sectors   Size Id Type
/dev/sdc1        2048  4196351  4194304    2G 8e Linux LVM
/dev/sdc2      4196352  8390655  4194304    2G 8e Linux LVM
/dev/sdc3      8390656 18876415 10485760    5G  5 Extended
/dev/sdc4      18876416 19924991  1048576 512M 8e Linux LVM
/dev/sdc5      8392704 10489855  2097152    1G 8e Linux LVM
/dev/sdc6      10491904 12589055  2097152    1G 8e Linux LVM
/dev/sdc7      12591104 14688255  2097152    1G 8e Linux LVM
/dev/sdc8      14690304 17836031  3145728 1.5G 8e Linux LVM

Partition table entries are not in disk order.
```

```
[root@therema-server ~]# partprobe /dev/sdc
[root@therema-server ~]# pvcreate /dev/sdc4
Physical volume "/dev/sdc4" successfully created.
[root@therema-server ~]# pvdisplay
```

```
"/dev/sdc4" is a new physical volume of "512.00 MiB"
--- NEW Physical volume ---
PV Name          /dev/sdc4
VG Name
PV Size         512.00 MiB
Allocatable     NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE      0
PV UUID
```

```
[root@therema-server ~]# vgextend vg1 /dev/sdc4
  Volume group "vg1" successfully extended
[root@therema-server ~]# vgdisplay
--- Volume group ---
VG Name           vg1
System ID
Format          lvm2
Metadata Areas    5
Metadata Sequence No 4
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV           2
Open LV           0
Max PV           0
Cur PV           5
Act PV           5
VG Size          4.98 GiB
PE Size          4.00 MiB
Total PE         1275
Alloc PE / Size  768 / 3.00 GiB
Free  PE / Size  507 / 1.98 GiB
VG UUID
```

```
[root@therema-server ~]# lvextend -L +512M /dev/vg1/lv1
  Size of logical volume vg1/lv1 changed from 1.50 GiB (384 extents) to 2.00 GiB (512 extents).
  Logical volume vg1/lv1 successfully resized.
```

```
[root@therema-server ~]# e2fsck -f /dev/vg1/lv1
e2fsck 1.46.5 (30-Dec-2021)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/vg1/lv1: 11/98304 files (0.0% non-contiguous), 15524/393216 blocks
[root@therema-server ~]# resize2fs /dev/vg1/lv1
resize2fs 1.46.5 (30-Dec-2021)
Resizing the filesystem on /dev/vg1/lv1 to 524288 (4k) blocks.
The filesystem on /dev/vg1/lv1 is now 524288 (4k) blocks long.
```

```
[root@therema-server ~]# lvextend -L +512M /dev/vg0/lv0
  Size of logical volume vg0/lv0 changed from 1.50 GiB (384 extents) to 2.00 GiB (512 extents).
  Logical volume vg0/lv0 successfully resized.
[root@therema-server ~]# mkswap /dev/vg0/lv0
mkswap: /dev/vg0/lv0: warning: wiping old swap signature.
Setting up swap space version 1, size = 2 GiB (2147479552 bytes)
no label, UUID=
```

```
[root@therema-server ~]# lvextend -L +512M /dev/vg0/lv0
  Size of logical volume vg0/lv0 changed from 1.50 GiB (384 extents) to 2.00 GiB (512 extents).
  Logical volume vg0/lv0 successfully resized.
[root@therema-server ~]# mkswap /dev/vg0/lv0
mkswap: /dev/vg0/lv0: warning: wiping old swap signature.
Setting up swap space version 1, size = 2 GiB (2147479552 bytes)
no label, UUID=
```

```
[root@therema-server ~]# fsck -f /dev/vg1/lv2
fsck from util-linux 2.37.4
e2fsck 1.46.5 (30-Dec-2021)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/mapper/vg1-lv2: 11/65536 files (0.0% non-contiguous), 13275/262144 blocks
[root@therema-server ~]# resize2fs /dev/vg1/lv2 512M
resize2fs 1.46.5 (30-Dec-2021)
Resizing the filesystem on /dev/vg1/lv2 to 131072 (4k) blocks.
The filesystem on /dev/vg1/lv2 is now 131072 (4k) blocks long.
```

```
[root@therema-server ~]# lvreduce -L 1G /dev/vg1/lv2
  File system ext4 found on vg1/lv2.
  File system size (512.00 MiB) is smaller than the requested size (1.00 GiB).
  File system reduce is not needed, skipping.
  Size of logical volume vg1/lv2 changed from 1.50 GiB (384 extents) to 1.00 GiB (256 extents).
  Logical volume vg1/lv2 successfully resized.
```

Ritesh Kumar

```
[root@therema-server ~]# lvdisplay
--- Logical volume ---
LV Path          /dev/vg1/lv1
LV Name          lv1
VG Name          vg1
LV UUID          TS3ara-29TR-Hsoe-r4uv-ZHc3-bODT-DDtRrO
LV Write Access  read/write
LV Creation host, time therema-server, 2025-10-30 22:13:37 +0530
LV Status        available
# open           1
LV Size          2.00 GiB
Current LE       512
Segments         3
Allocation       inherit
Read ahead sectors auto
- currently set to 256
Block device    253:3

--- Logical volume ---
LV Path          /dev/vg1/lv2
LV Name          lv2
VG Name          vg1
LV UUID          gHXDF4-ejUo-ZRww-bZQx-Am5S-WXeH-M6423c
LV Write Access  read/write
LV Creation host, time therema-server, 2025-10-30 22:14:43 +0530
LV Status        available
# open           1
LV Size          1.00 GiB
Current LE       256
Segments         2
Allocation       inherit
Read ahead sectors auto
- currently set to 256
Block device    253:4

--- Logical volume ---
LV Path          /dev/vg0/lv0
LV Name          lv0
VG Name          vg0
LV UUID          ZN0J7y-FLvs-bt5X-fNLb-f7bg-09A0-Iit0gR
LV Write Access  read/write
LV Creation host, time therema-server, 2025-10-30 21:47:09 +0530
LV Status        available
# open           2
LV Size          2.00 GiB
Current LE       512
Segments         2
Allocation       inherit
Read ahead sectors auto
- currently set to 256
Block device    253:2
```

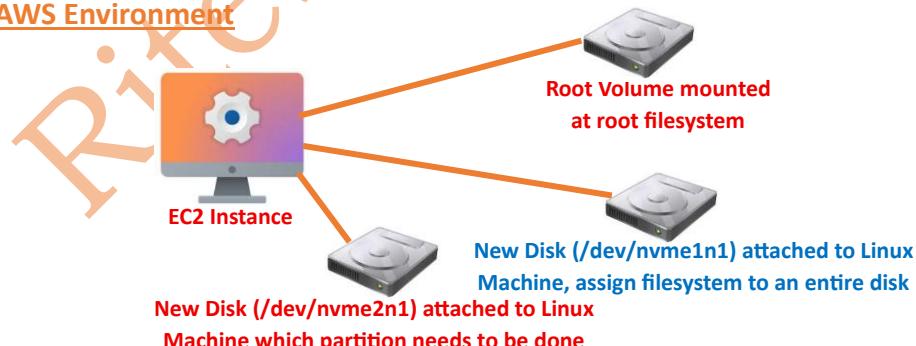
```
[root@therema-server ~]# swapon -a
[root@therema-server ~]# mount -a
[root@therema-server ~]# df -ht
Filesystem           Type      Size  Used Avail Use% Mounted on
devtmpfs             devtmpfs  4.0M    0  4.0M  0% /dev
tmpfs                tmpfs     1.8G    0  1.8G  0% /dev/shm
tmpfs                tmpfs    725M  9.9M  715M  2% /run
/dev/mapper/almalinux-root xfs      44G  5.8G  39G 14% /
/dev/sda1              xfs     960M 366M  595M 39% /boot
/dev/mapper/vg1-lv1   ext4     2.0G  24K  1.9G 1% /pablo/therema
/dev/sdb               ext4     9.8G  24K  9.3G 1% /dexter
tmpfs                tmpfs    363M  52K  363M 1% /run/user/42
tmpfs                tmpfs    363M  92K  363M 1% /run/user/1000
/dev/sr0               iso9660 12G   12G   0 100% /run/media/ritesh/AlmaLinux-9-6-x86_64-dvd
/dev/mapper/vg1-lv2   ext4     974M  24K  907M 1% /pablo/mederma
[root@therema-server ~]# free -mh
              total        used         free       shared  buff/cache   available
Mem:      3.5Gi     1.3Gi     838Mi        19Mi     1.7Gi     2.2Gi
Swap:    7.0Gi        0B      7.0Gi
```

The disk portioning as shown above had been performed on **Almalinux 9.6**.

```
[root@therema-server ~]# cat /etc/os-release
NAME="AlmaLinux"
VERSION="9.6 (Sage Margay)"
ID="almalinux"
ID_LIKE="rhel centos fedora"
VERSION_ID="9.6"
PLATFORM_ID="platform:el9"
PRETTY_NAME="AlmaLinux 9.6 (Sage Margay)"
ANSI_COLOR="0;34"
LOGO="fedora-logo-icon"
CPE_NAME="cpe:/o:almalinux:almalinux:9::baseos"
HOME_URL="https://almalinux.org/"
DOCUMENTATION_URL="https://wiki.almalinux.org/"
BUG_REPORT_URL="https://bugs.almalinux.org/"

ALMALINUX_MANTISBT_PROJECT="AlmaLinux-9"
ALMALINUX_MANTISBT_PROJECT_VERSION="9.6"
REDHAT_SUPPORT_PRODUCT="AlmaLinux"
REDHAT_SUPPORT_PRODUCT_VERSION="9.6"
SUPPORT_END=2032-06-01
```

AWS Environment



In industry you may face the problem that on a Running EC2 Instance the attached volume may full, in that case you need to increase the size of attached EBS Volume. As per the project requirement you may need to create a new EBS volume and attach it to the Running EC2 Instance. I had used Amazon Linux 2023 as the Operating System.

```
[root@ip-172-31-18-218 ~]# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  4.0M   0  4.0M  0% /dev
tmpfs          tmpfs    453M   0  453M  0% /dev/shm
tmpfs          tmpfs   181M  424K 181M  1% /run
/dev/nvme0n1p1  xfs     8.0G  1.6G 6.4G 20% /
tmpfs          tmpfs   453M   0  453M  0% /tmp
/dev/nvme0n1p128 vfat    10M  1.3M  8.7M 13% /boot/efi
tmpfs          tmpfs   91M   0   91M  0% /run/user/1001
```

Let us consider that the attached EBS Volume had been full and we need to increase the size of EBS Volume. First in the terraform script for EC2 Instance provide the increased size for parameter **volume_size** to the desired new size in GB then run **terraform plan** proceed by **terraform apply**. For this project earlier the root volume size was 8GB and later it was increase by 7GB (targeted size=15GB). Which has started reflecting in the AWS Console as well after running the terraform apply command. Now check using the command **lsblk**, you find that volume size had been increase but not reflected in the mounted partition size.

```
[root@ip-172-31-18-218 ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
nvme0n1    259:0   0  15G  0 disk
└─nvme0n1p1 259:1   0   8G  0 part /
└─nvme0n1p127 259:2   0   1M  0 part
└─nvme0n1p128 259:3   0  10M  0 part /boot/efi
[root@ip-172-31-18-218 ~]# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  4.0M   0  4.0M  0% /dev
tmpfs          tmpfs    453M   0  453M  0% /dev/shm
tmpfs          tmpfs   181M  424K 181M  1% /run
/dev/nvme0n1p1  xfs     8.0G  1.6G 6.4G 20% /
tmpfs          tmpfs   453M   0  453M  0% /tmp
/dev/nvme0n1p128 vfat    10M  1.3M  8.7M 13% /boot/efi
tmpfs          tmpfs   91M   0   91M  0% /run/user/1001
```

Run the below command and then check you will find that mounted partition size will be increased.

```
[root@ip-172-31-18-218 ~]# growpart /dev/nvme0n1 1
CHANGED: partition=1 start=24576 old: size=16752607 end=16777183 new: size=31432671 end=31457247
[root@ip-172-31-18-218 ~]# xfs_growfs /dev/nvme0n1p1
meta-data=/dev/nvme0n1p1      isize=512    agcount=2, agsize=1047040 blks
                               =           sectsz=4096  attr=2, projid32bit=1
                               =           crc=1    finobt=1, sparse=1, rmapbt=0
                               =           reflink=1 bigtime=1 inobtcount=1 nnext64=0
                               =           exchange=0
data      =           bsize=4096   blocks=2094075, imaxpct=25
                               =           sunit=128  swidth=128 blks
naming    =version 2          bsize=16384  ascii-ci=0, ftype=1, parent=0
log       =internal log       bsize=4096   blocks=16384, version=2
                               =           sectsz=4096 sunit=4 blks, lazy-count=1
realtime  =none              extsz=4096   blocks=0, rtextents=0
data blocks changed from 2094075 to 3929083
[root@ip-172-31-18-218 ~]# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  4.0M   0  4.0M  0% /dev
tmpfs           tmpfs     453M   0  453M  0% /dev/shm
tmpfs           tmpfs     181M  424K  181M  1% /run
/dev/nvme0n1p1  xfs      15G  1.6G  14G  11% /
tmpfs           tmpfs     453M   0  453M  0% /tmp
/dev/nvme0n1p128 vfat     10M  1.3M  8.7M  13% /boot/efi
tmpfs           tmpfs     91M   0  91M  0% /run/user/1001
```

Using Terraform script I created two EBS volumes and attached it to the same Running EC2 Instance. Then I checked using the command **lsblk** my findings are as shown in the screenshot attached below.

```
[root@ip-172-31-18-218 ~]# lsblk
NAME      MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
nvme0n1    259:0    0  15G  0 disk
└─nvme0n1p1 259:1    0  15G  0 part /
└─nvme0n1p127 259:2    0   1M  0 part
└─nvme0n1p128 259:3    0  10M  0 part /boot/efi
nvme1n1    259:4    0   5G  0 disk
nvme2n1    259:5    0  10G  0 disk

[root@ip-172-31-18-218 ~]# mkfs.xfs /dev/nvme1n1
meta-data=/dev/nvme1n1      isize=512    agcount=8, agsize=163840 blks
                               =           sectsz=512  attr=2, projid32bit=1
                               =           crc=1    finobt=1, sparse=1, rmapbt=0
                               =           reflink=1 bigtime=1 inobtcount=1 nnext64=0
                               =           exchange=0
data      =           bsize=4096   blocks=1310720, imaxpct=25
                               =           sunit=1  swidth=1 blks
naming    =version 2          bsize=4096  ascii-ci=0, ftype=1, parent=0
log       =internal log       bsize=4096   blocks=16384, version=2
                               =           sectsz=512 sunit=1 blks, lazy-count=1
realtime  =none              extsz=4096   blocks=0, rtextents=0
[root@ip-172-31-18-218 ~]# mkdir -p /thaplex/zikomo
```

```
[root@ip-172-31-18-218 ~]# cat /etc/fstab
#
UUID= [REDACTED]           /          xfs    defaults,noatime 1  1
UUID=161A-5EA2      /boot/efi      vfat   defaults,noatime,uid=0,gid=0,umask=0077,shortname=winnt,x-systemd.automount 0 2

/dev/nvme1n1 /thaplex/zikomo xfs defaults 0 0
[root@ip-172-31-18-218 ~]# mount -a
[root@ip-172-31-18-218 ~]# df -hT
Filesystem      Type  Size  Used  Avail Use% Mounted on
devtmpfs        devtmpfs 4.0M   0  4.0M  0% /dev
tmpfs          tmpfs   453M   0  453M  0% /dev/shm
tmpfs          tmpfs   181M  440K 181M  1% /run
/dev/nvme0n1p1   xfs   15G  1.6G 14G  11% /
tmpfs          tmpfs   453M   0  453M  0% /tmp
/dev/nvme0n1p128 vfat   10M  1.3M  8.7M 13% /boot/efi
tmpfs          tmpfs   91M   0  91M  0% /run/user/1001
/dev/nvme1n1     xfs   5.0G  68M  4.9G  2% /thaplex/zikomo

[root@ip-172-31-18-218 ~]# fdisk /dev/nvme2n1

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x8fd10c7b.

Command (m for help): n
Partition type
  p  primary (0 primary, 0 extended, 4 free)
  e  extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-20971519, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-20971519, default 20971519): +2G

Created a new partition 1 of type 'Linux' and of size 2 GiB.

Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): L

00 Empty              24 NEC DOS      81 Minix / old Lin  bf Solaris
01 FAT12             27 Hidden NTFS Win 82 Linux swap / So c1 DRDOS/sec (FAT-
02 XENIX root        39 Plan 9       83 Linux [REDACTED] c4 DRDOS/sec (FAT-
03 XENIX usr         3c PartitionMagic 84 OS/2 hidden or c6 DRDOS/sec (FAT-
04 FAT16 <32M        40 Venix 80286  85 Linux extended c7 Syrinx
05 Extended          41 PPC PReP Boot 86 NTFS volume set da Non-FS data
06 FAT16              42 SFS          87 NTFS volume set db CP/M / CTOS / .
07 HPFS/NTFS/exFAT  4d QNX4.x      88 Linux plaintext de Dell Utility
08 AIX                4e QNX4.x 2nd part 8e Linux LVM df BootIt
09 AIX bootable       4f QNX4.x 3rd part 93 Amoeba e1 DOS access
0a OS/2 Boot Manag   50 OnTrack DM   94 Amoeba BBT e3 DOS R/O
0b W95 FAT32          51 OnTrack DM6 Aux 9f BSD/OS e4 SpeedStor
0c W95 FAT32 (LBA)   52 CP/M        a0 IBM Thinkpad hi ea Linux extended
```

```

Aliases:
    linux          - 83
    swap           - 82
    extended       - 05
    uefi           - EF
    raid            - FD
    lvm             - 8E
    linuxex        - 85
Hex code or alias (type L to list all): 83
Changed type of partition 'Linux' to 'Linux'.

```

```

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

```



```

[root@ip-172-31-18-218 ~]# fdisk /dev/nvme2n1

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): n
Partition type
  p   primary (1 primary, 0 extended, 3 free)
  e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (2-4, default 2):
First sector (4196352-20971519, default 4196352):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (4196352-20971519, default 20971519): +2G

Created a new partition 2 of type 'Linux' and of size 2 GiB.

Command (m for help): t
Partition number (1,2, default 2):
Hex code or alias (type L to list all): 83

Changed type of partition 'Linux' to 'Linux'.

Command (m for help): p
Disk /dev/nvme2n1: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: Amazon Elastic Block Store
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0x8fd10c7b

Device      Boot   Start     End Sectors Size Id Type
/dev/nvme2n1p1        2048 4196351 4194304    2G 83 Linux
/dev/nvme2n1p2        4196352 8390655 4194304    2G 83 Linux

```

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

```
[root@ip-172-31-18-218 ~]# mkdir /thaplex/jonghou
[root@ip-172-31-18-218 ~]# mkdir /thaplex/dexter
```

```
[root@ip-172-31-18-218 ~]# partprobe /dev/nvme2n1
[root@ip-172-31-18-218 ~]# mkfs.ext4 /dev/nvme2n1p1
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: dd5dd21e-5e2f-4788-a07a-7b53bffc0d07
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912
```

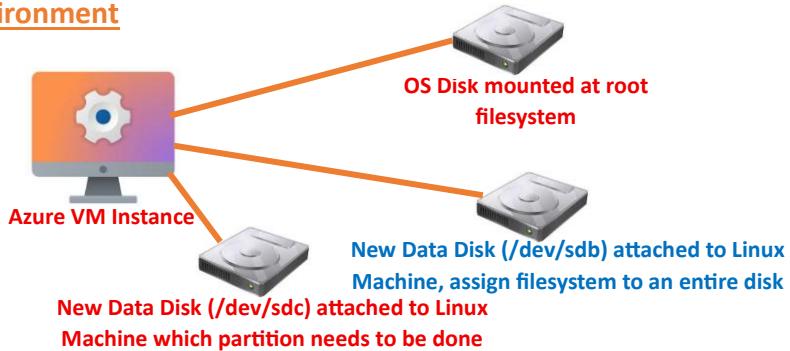
```
Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
[root@ip-172-31-18-218 ~]# mkfs.ext4 /dev/nvme2n1p2
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: 3043ccb2-b479-4d94-a149-f1a0555332ba
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
[root@ip-172-31-18-218 ~]# cat /etc/fstab
#
UUID=XXXXXXXXXXXXXX        /          xfs    defaults,noatime 1 1
UUID=XXXXXXXXXXXXXX        /boot/efi    vfat   defaults,noatime,uid=0,gid=0,umask=0077,shortname=winnt,x-systemd.automount 0 2
/dev/nvme1n1 /thaplex/zikomo xfs defaults 0 0
/dev/nvme2n1p1 /thaplex/jonghou ext4 defaults 0 0
/dev/nvme2n1p2 /thaplex/dexter ext4 defaults 0 0
[root@ip-172-31-18-218 ~]# mount -a
[root@ip-172-31-18-218 ~]# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  4.0M   0    4.0M  0%  /dev
tmpfs           tmpfs     453M   0   453M  0%  /dev/shm
tmpfs           tmpfs     181M  448K  181M  1%  /run
/dev/nvme0n1p1  xfs      15G  1.6G  14G  11%  /
tmpfs           tmpfs     453M   0   453M  0%  /tmp
/dev/nvme0n1p128 vfat     10M  1.3M  8.7M  13%  /boot/efi
tmpfs           tmpfs     91M   0   91M  0%  /run/user/1001
/dev/nvme1n1   xfs      5.0G  68M  4.9G  2%  /thaplex/zikomo
/dev/nvme2n1p1  ext4     2.0G  24K  1.8G  1%  /thaplex/jonghou
/dev/nvme2n1p2  ext4     2.0G  24K  1.8G  1%  /thaplex/dexter
```

Azure Environment



In industry you may face the problem that on a Running Azure VM the attached OS Disk may full, in that case you need to increase the size of attached OS Disk. For this project I had created the Azure VM using terraform script and as per the project requirement I need to increase the size of OS Disk from 30GB to 35GB. In terraform script of azure vm I increased the size using the parameter **os_disk = 35** (targeted size in GB), increasing the size of Azure VM OS Disk will restart the Azure VM So plan this activity during non-production hours. As per the project requirement you may need to create new Data Disks and attach it to the Running Azure VM. I created two data disks of size 5 and 10GB and attached to the running Azure VM using the terraform script. I had used Almalinux 8.10 as the Operating System.

```
[root@Dexter-Server ~]# cat /etc/os-release
NAME="AlmaLinux"
VERSION="8.10 (Cerulean Leopard)"
ID="almalinux"
ID_LIKE="rhel centos fedora"
VERSION_ID="8.10"
PLATFORM_ID="platform:el8"
PRETTY_NAME="AlmaLinux 8.10 (Cerulean Leopard)"
ANSI_COLOR="0;34"
LOGO="fedora-logo-icon"
CPE_NAME="cpe:/o:almalinux:almalinux:8::baseos"
HOME_URL="https://almalinux.org/"
DOCUMENTATION_URL="https://wiki.almalinux.org/"
BUG_REPORT_URL="https://bugs.almalinux.org/"

ALMALINUX_MANTISBT_PROJECT="AlmaLinux-8"
ALMALINUX_MANTISBT_PROJECT_VERSION="8.10"
REDHAT_SUPPORT_PRODUCT="AlmaLinux"
REDHAT_SUPPORT_PRODUCT_VERSION="8.10"
SUPPORT_END=2029-06-01
```

```
[root@Dexter-Server ~]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda     8:0    0   30G  0 disk
└─sda1  8:1    0    1M  0 part
└─sda2  8:2    0  200M  0 part /boot/efi
└─sda3  8:3    0    1G  0 part /boot
└─sda4  8:4    0 28.8G  0 part /
sdb     8:16   0    4G  0 disk
└─sdb1  8:17   0    4G  0 part /mnt
sdc     8:32   0   10G  0 disk
sdd     8:48   0    5G  0 disk
sr0     11:0   1  634K 0 rom
```

The screenshot after increasing the size is as shown below.

```
[root@Dexter-Server ~]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda     8:0    0   35G  0 disk
└─sda1  8:1    0    1M  0 part
└─sda2  8:2    0  200M  0 part /boot/efi
└─sda3  8:3    0    1G  0 part /boot
└─sda4  8:4    0 33.8G  0 part /
sdb     8:16   0    4G  0 disk
└─sdb1  8:17   0    4G  0 part /mnt
sdc     8:32   0   10G  0 disk
sdd     8:48   0    5G  0 disk
```

After increasing the size of OS Disk, it will be automatically reflected in the partition as shown in the screenshot attached below.

```
[root@Dexter-Server ~]# df -hT
Filesystem  Type      Size  Used Avail Use% Mounted on
devtmpfs    devtmpfs  406M   0  406M  0% /dev
tmpfs       tmpfs     449M   0  449M  0% /dev/shm
tmpfs       tmpfs     449M  12M  438M  3% /run
tmpfs       tmpfs     449M   0  449M  0% /sys/fs/cgroup
/dev/sda4   xfs       34G  2.7G  32G  8% /
/dev/sda3   xfs      1014M 225M  790M 23% /boot
/dev/sda2   vfat      200M  5.9M  194M  3% /boot/efi
/dev/sdb1   ext4      3.9G  28K  3.7G  1% /mnt
tmpfs       tmpfs     90M   0   90M  0% /run/user/1000
```

```
[root@Dexter-Server ~]# mkfs.xfs /dev/sdd
meta-data=/dev/sdd
      isize=512    agcount=4, agsize=327680 blks
      =         sectsz=4096  attr=2, projid32bit=1
      =         crc=1    finobt=1, sparse=1, rmapbt=0
data     =         bsize=4096  bigtime=0 inobtcount=0
      =         sunit=0    blocks=1310720, imaxpct=25
naming   =version 2    swidth=0 blks
log      =internal log  bsize=4096  ascii-ci=0, ftype=1
      =         sectsz=4096  blocks=2560, version=2
realtime =none        extsz=4096  sunit=1 blks, lazy-count=1
Discarding blocks...Done.
[root@Dexter-Server ~]# mkdir -p /zaxv/hanzo
```

```
[root@Dexter-Server ~]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Fri Sep  5 09:17:25 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=          /          xfs    defaults      0 0
UUID=          /boot      xfs    defaults      0 0
UUID=          /boot/efi  vfat   defaults,uid=0,gid=0,umask=077,shortname=winnt 0 2
/dev/disk/cloud/azure_resource-part1  /mnt    auto  defaults,nofail,x-systemd.requires=cloud-init.service,_netdev,comment=cloudconfig      0      2
/dev/sdd /zaxv/hanzo xfs defaults 0 0
[root@Dexter-Server ~]# mount -a
```

```
[root@Dexter-Server ~]# df -hT
Filesystem  Type      Size  Used  Avail Use% Mounted on
devtmpfs    devtmpfs  406M   0    406M  0%  /dev
tmpfs       tmpfs     449M   0    449M  0%  /dev/shm
tmpfs       tmpfs     449M   12M  438M  3%  /run
tmpfs       tmpfs     449M   0    449M  0%  /sys/fs/cgroup
/dev/sda4   xfs      34G   2.9G  31G   9%  /
/dev/sda3   xfs      1014M  225M  790M  23%  /boot
/dev/sda2   vfat     200M   5.9M  194M  3%  /boot/efi
/dev/sdb1   ext4     3.9G   28K   3.7G  1%  /mnt
tmpfs       tmpfs     90M   0    90M   0%  /run/user/1000
/dev/sdd   xfs      5.0G   68M   5.0G  2%  /zaxv/hanzo
```

```
[root@Dexter-Server ~]# fdisk /dev/sdc

Welcome to fdisk (util-linux 2.32.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xbdbbbf64.

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-20971519, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-20971519, default 20971519): +2G

Created a new partition 1 of type 'Linux' and of size 2 GiB.

Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): 83
Changed type of partition 'Linux' to 'Linux'.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

Ritesh K

```
[root@Dexter-Server ~]# fdisk /dev/sdc

Welcome to fdisk (util-linux 2.32.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): n
Partition type
  p  primary (1 primary, 0 extended, 3 free)
  e  extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (2-4, default 2):
First sector (4196352-20971519, default 4196352):
Last sector, +sectors or +size{K,M,G,T,P} (4196352-20971519, default 20971519): +2G

Created a new partition 2 of type 'Linux' and of size 2 GiB.

Command (m for help): t
Partition number (1,2, default 2):
Hex code (type L to list all codes): 83

Changed type of partition 'Linux' to 'Linux'.

Command (m for help): p
Disk /dev/sdc: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0xbdbbbf64

Device      Boot   Start     End Sectors Size Id Type
/dev/sdc1        2048 4196351 4194304    2G 83 Linux
/dev/sdc2    4196352 8390655 4194304    2G 83 Linux
```

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

```
[root@Dexter-Server ~]# partprobe /dev/sdc
```

```
[root@Dexter-Server ~]# mkfs.xfs /dev/sdc1
meta-data=/dev/sdc1
      isize=512    agcount=4, agsize=131072 blks
      =                      sectsz=4096  attr=2, projid32bit=1
      =                      crc=1     finobt=1, sparse=1, rmapbt=0
data   =                      reflink=1  bigtime=0 inobtcount=0
      =                      bsize=4096 blocks=524288, imaxpct=25
      =                      sunit=0    swidth=0 blks
naming =version 2            bsize=4096 ascii-ci=0, ftype=1
log    =internal log         bsize=4096 blocks=2560, version=2
      =                      sectsz=4096 sunit=1 blks, lazy-count=1
realtime =none               extsz=4096 blocks=0, rtextents=0
Discarding blocks...Done.
[root@Dexter-Server ~]# mkfs.ext4 /dev/sdc2
mke2fs 1.45.6 (20-Mar-2020)
Discarding device blocks: done
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: 910842a7-4748-4217-81fa-155f803af7a6
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912

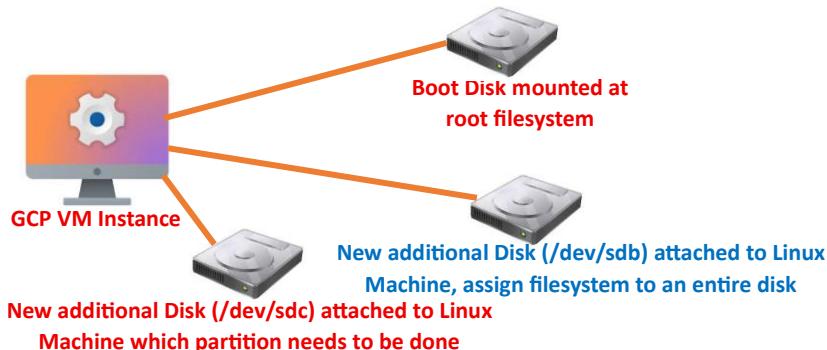
Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

[root@Dexter-Server ~]# mkdir /zaxv/mederma
[root@Dexter-Server ~]# mkdir /zaxv/dexter
```

```
[root@Dexter-Server ~]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Fri Sep  5 09:17:25 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID= [REDACTED] /          xfs    defaults        0 0
UUID= [REDACTED] /boot      xfs    defaults        0 0
UUID= [REDACTED] /boot/efi vfat   defaults,uid=0,gid=0,umask=077,shortname=winnt 0 2
/dev/disk/cloud/azure_resource-part1 /mnt    auto   defaults,nofail,x-systemd.requires=cloud-init.service,_netdev,comment=cloudconfig      0      2
/dev/sdd /zaxv/hanzo xfs defaults 0 0
/dev/sdc1 /zaxv/mederma xfs defaults 0 0
/dev/sdc2 /zaxv/dexter ext4 defaults 0 0
[root@Dexter-Server ~]# mount -a
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
devtmpfs	devtmpfs	406M	0	406M	0%	/dev
tmpfs	tmpfs	449M	0	449M	0%	/dev/shm
tmpfs	tmpfs	449M	12M	437M	3%	/run
tmpfs	tmpfs	449M	0	449M	0%	/sys/fs/cgroup
/dev/sda4	xfs	34G	2.9G	31G	9%	/
/dev/sda3	xfs	1014M	225M	790M	23%	/boot
/dev/sda2	vfat	200M	5.9M	194M	3%	/boot/efi
/dev/sdb1	ext4	3.9G	28K	3.7G	1%	/mnt
tmpfs	tmpfs	90M	0	90M	0%	/run/user/1000
/dev/sdd	xfs	5.0G	68M	5.0G	2%	/zaxv/hanzo
/dev/sdc1	xfs	2.0G	47M	2.0G	3%	/zaxv/mederma
/dev/sdc2	ext4	2.0G	24K	1.8G	1%	/zaxv/dexter

Google Cloud Platform (GCP) Environment



In industry you may face the problem that on a Running GCP VM Instance the attached boot disk may full, in that case you need to increase the size of boot disk. For this project I had created the GCP VM Instance using terraform script and as per the project requirement I need to increase the size of boot disk from 20GB to 25GB. In terraform script of azure vm I increased the size using the parameter **size = 25** (targeted size in GB) under the block **boot_disk** in terraform script. As per the project requirement you may need to create new Data Disks and attach it to the Running Azure VM. I created two data disks of size 10GB each and attached to the running Azure VM using the terraform script. I had used Rocky Linux 8.10 as the Operating System.

```
[root@mederma-server ~]# cat /etc/os-release
NAME="Rocky Linux"
VERSION="8.10 (Green Obsidian)"
ID="rocky"
ID_LIKE="rhel centos fedora"
VERSION_ID="8.10"
PLATFORM_ID="platform:el8"
PRETTY_NAME="Rocky Linux 8.10 (Green Obsidian)"
ANSI_COLOR="0;32"
LOGO="fedora-logo-icon"
CPE_NAME="cpe:/o:rocky:rocky:8:GA"
HOME_URL="https://rockylinux.org/"
BUG_REPORT_URL="https://bugs.rockylinux.org/"
SUPPORT_END="2029-05-31"
ROCKY_SUPPORT_PRODUCT="Rocky-Linux-8"
ROCKY_SUPPORT_PRODUCT_VERSION="8.10"
REDHAT_SUPPORT_PRODUCT="Rocky Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="8.10"
```

```
[root@mederma-server ~]# lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0   20G  0 disk
└─sda1   8:1    0  200M  0 part /boot/efi
└─sda2   8:2    0 19.8G  0 part /
[root@mederma-server ~]# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  1.8G   0  1.8G  0% /dev
tmpfs          tmpfs     1.8G   0  1.8G  0% /dev/shm
tmpfs          tmpfs     1.8G  17M  1.8G  1% /run
tmpfs          tmpfs     1.8G   0  1.8G  0% /sys/fs/cgroup
/dev/sda2       xfs      20G  4.3G  16G  22% /
/dev/sda1       vfat     200M  5.8M  194M  3% /boot/efi
tmpfs          tmpfs   367M   0  367M  0% /run/user/1000
```

After running the terraform script, size of boot disk had been increased and attached two extra disks to the GCP VM Instance as shown in the screenshot attached below.

```
[root@mederma-server ~]# lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0   25G  0 disk
└─sda1   8:1    0  200M  0 part /boot/efi
└─sda2   8:2    0 19.8G  0 part /
sdb      8:16   0   10G  0 disk
sdc      8:32   0   10G  0 disk
[root@mederma-server ~]# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  1.8G   0  1.8G  0% /dev
tmpfs          tmpfs     1.8G   0  1.8G  0% /dev/shm
tmpfs          tmpfs     1.8G  17M  1.8G  1% /run
tmpfs          tmpfs     1.8G   0  1.8G  0% /sys/fs/cgroup
/dev/sda2       xfs      20G  5.0G  15G  26% /
/dev/sda1       vfat     200M  5.8M  194M  3% /boot/efi
tmpfs          tmpfs   367M   0  367M  0% /run/user/1000
```

For extra disk **/dev/sdb** I assigned xfs filesystem to it and then mounted it to a directory. For extra disk **/dev/sdc** I created two primary partitions **/dev/sdc1** and **/dev/sdc2** then assigned filesystem xfs and ext4 then mounted it to the directory to make it usable.

For boot disk you can see from above attached screenshot that size had been increased but not reflecting in the mounted partition, which can be achieved as discussed below.

In GCP Rocky Linux 8.10 VM Instance **growpart** command is not available to make it available you need to install the package **cloud-utils-growpart**, you can install it using the command **yum install cloud-utils-growpart -y**.

```
[root@mederma-server ~]# yum install cloud-utils-growpart -y
```

```
[root@mederma-server ~]# growpart /dev/sda 2
CHANGED: partition=2 start=411648 old: size=41529344 end=41940991 new: size=52017119 end=52428766
[root@mederma-server ~]# xfs_growfs /dev/sda2
meta-data=/dev/sda2              isize=512    agcount=4, agsize=1297792 blks
                                =          sectsz=4096 attr=2, projid32bit=1
                                =          crc=1   finobt=1, sparse=1, rmapbt=0
data     =           bsize=4096   blocks=5191168, imaxpct=25
                                =          sunit=0  swidth=0 blks
naming   =version 2             bsize=4096   ascii-ci=0, ftype=1
log      =internal log          bsize=4096   blocks=2560, version=2
                                =          sectsz=4096 sunit=1 blks, lazy-count=1
realtime =none                 extsz=4096   blocks=0, rtextents=0
data blocks changed from 5191168 to 6502139
[root@mederma-server ~]# df -hT
Filesystem  Type      Size  Used Avail Use% Mounted on
devtmpfs    devtmpfs  1.8G   0   1.8G  0% /dev
tmpfs       tmpfs     1.8G   0   1.8G  0% /dev/shm
tmpfs       tmpfs     1.8G  17M  1.8G  1% /run
tmpfs       tmpfs     1.8G   0   1.8G  0% /sys/fs/cgroup
/dev/sda2   xfs       25G  5.1G  20G  21% /
/dev/sda1   vfat      200M  5.8M  194M  3% /boot/efi
tmpfs       tmpfs     367M   0  367M  0% /run/user/1000
```

```
[root@mederma-server ~]# mkdir -p /pablo/pather
```

```
[root@mederma-server ~]# mkfs.xfs /dev/sdb
meta-data=/dev/sdb              isize=512    agcount=4, agsize=655360 blks
                                =          sectsz=4096 attr=2, projid32bit=1
                                =          crc=1   finobt=1, sparse=1, rmapbt=0
data     =           bsize=4096   blocks=2621440, imaxpct=25
                                =          sunit=0  swidth=0 blks
naming   =version 2             bsize=4096   ascii-ci=0, ftype=1
log      =internal log          bsize=4096   blocks=2560, version=2
                                =          sectsz=4096 sunit=1 blks, lazy-count=1
realtime =none                 extsz=4096   blocks=0, rtextents=0
Discarding blocks...Done.
```

```
[root@mederma-server ~]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Tue Oct 14 09:28:19 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=[REDACTED]          /          xfs    defaults        0 0
UUID=[REDACTED]          /boot/efi    vfat   defaults,uid=0,gid=0,umask=077,shortname=winnt 0 2
/dev/sdb /pablo/pather xfs defaults 0 0
[root@mederma-server ~]# mount -a
```

```
[root@mederma-server ~]# df -hT
Filesystem      Type      Size  Used  Avail Use% Mounted on
devtmpfs        devtmpfs  1.8G   0    1.8G  0% /dev
tmpfs          tmpfs     1.8G   0    1.8G  0% /dev/shm
tmpfs          tmpfs     1.8G  17M  1.8G  1% /run
tmpfs          tmpfs     1.8G   0    1.8G  0% /sys/fs/cgroup
/dev/sda2       xfs       25G  5.1G  20G  21% /
/dev/sda1       vfat      200M  5.8M  194M  3% /boot/efi
tmpfs          tmpfs    367M   0   367M  0% /run/user/1000
/dev/sdb        xfs       10G  104M  9.9G  2% /pablo/pather
```

```
[root@mederma-server ~]# fdisk /dev/sdc

Welcome to fdisk (util-linux 2.32.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x3b60d6a9.

Command (m for help): n
Partition type
  p  primary (0 primary, 0 extended, 4 free)
  e  extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-20971519, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-20971519, default 20971519): +2G

Created a new partition 1 of type 'Linux' and of size 2 GiB.

Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): 83
Changed type of partition 'Linux' to 'Linux'.

Command (m for help): n
Partition type
  p  primary (1 primary, 0 extended, 3 free)
  e  extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (2-4, default 2):
First sector (4196352-20971519, default 4196352):
Last sector, +sectors or +size{K,M,G,T,P} (4196352-20971519, default 20971519): +2G

Created a new partition 2 of type 'Linux' and of size 2 GiB.
```

```

Command (m for help): t
Partition number (1,2, default 2):
Hex code (type L to list all codes): 83

Changed type of partition 'Linux' to 'Linux'.

Command (m for help): p
Disk /dev/sdc: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0x3b60d6a9

Device      Boot   Start     End Sectors Size Id Type
/dev/sdc1        2048 4196351 4194304   2G 83 Linux
/dev/sdc2      4196352 8390655 4194304   2G 83 Linux

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

```

```
[root@mederma-server ~]# partprobe /dev/sdc
```

```
[root@mederma-server ~]# mkdir /pablo/hexa
[root@mederma-server ~]# mkdir /pablo/maxo
```

```
[root@mederma-server ~]# mkfs.xfs /dev/sdc1
meta-data=/dev/sdc1              isize=512    agcount=4, agsize=131072 blks
                                =          sectsz=4096  attr=2, projid32bit=1
                                =          crc=1       finobt=1, sparse=1, rmapbt=0
data     =          bsize=4096   reflink=1  bigtime=0 inobtcount=0
                                =          sunit=0     swidth=0 blks
naming   =version 2             bsize=4096  ascii-ci=0, ftype=1
log      =internal log          bsize=4096  blocks=2560, version=2
                                =          sectsz=4096 sunit=1 blks, lazy-count=1
realtime =none                  extsz=4096  blocks=0, rtextents=0
Discarding blocks...Done.
```

```
[root@mederma-server ~]# mkfs.ext4 /dev/sdc2
mke2fs 1.45.6 (20-Mar-2020)
Discarding device blocks: done
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: [REDACTED]
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

[root@mederma-server ~]# mount -a
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
devtmpfs	devtmpfs	1.8G	0	1.8G	0%	/dev
tmpfs	tmpfs	1.8G	0	1.8G	0%	/dev/shm
tmpfs	tmpfs	1.8G	17M	1.8G	1%	/run
tmpfs	tmpfs	1.8G	0	1.8G	0%	/sys/fs/cgroup
/dev/sda2	xfs	25G	5.1G	20G	21%	/
/dev/sda1	vfat	200M	5.8M	194M	3%	/boot/efi
/dev/sdb	xfs	10G	104M	9.9G	2%	/pablo/pather
tmpfs	tmpfs	367M	0	367M	0%	/run/user/1000
/dev/sdc1	xfs	2.0G	47M	2.0G	3%	/pablo/hexa
/dev/sdc2	ext4	2.0G	24K	1.8G	1%	/pablo/maxo

Types of Linux File Systems

File system is nothing but way to manage data on storage device. There are several types of native File system used in Linux as for example ext, ext2, ext3, ext4, XFS and Btrfs (used in SUSE Linux) some of them are explained below.

ext3 (extended version 3 filesystem) supports maximum filesystem size of 32TB and it also supports Journaling.

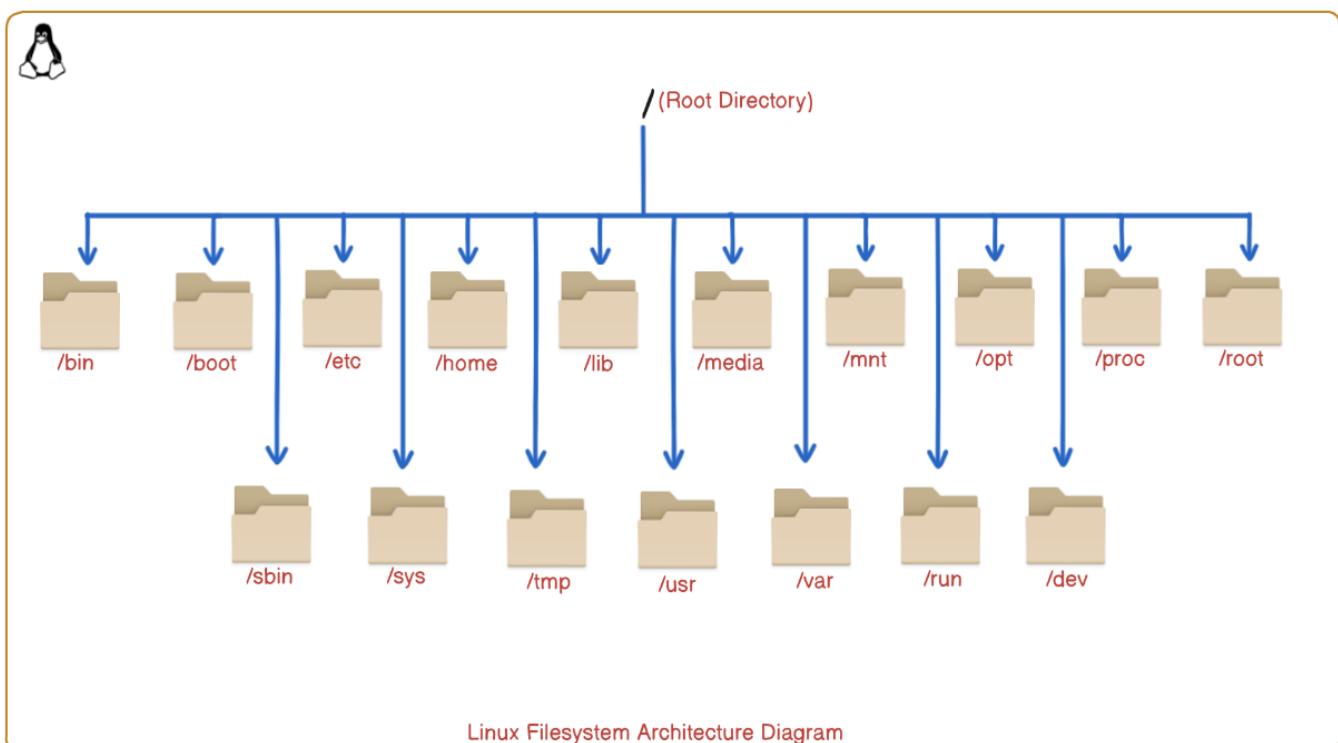
ext4 (extended version 4 filesystem) supports maximum filesystem size of 1EB and it supports Journaling.

XFS is a high-performance filesystem. It is used for large files and volumes. It supports Journaling.

These three are most frequently used native filesystems in Linux.

Journaling in Linux Filesystem means metadata and logs will be stored to special area called journal before writing them permanently to disk. Whenever there is a system crash using Journaling, it will restore the data. Journaling concept was introduced in ext3 filesystem.

Linux Filesystem Hierarchy



Linux Filesystem Architecture Diagram

/ (Root Level Directory or Top-Level Directory) in Linux is the highest point in the Linux filesystem hierarchy. It contains all other directory.

/bin contains system binaries, these are executable files. Each file corresponds to the Linux commands which can be used by all the users.

/boot contains files and directory which will be used to boot the Linux machine. This directory contains the Linux Kernel (vmlinuz file) and bootloader (grub2 directory).

/etc containers configuration files for Linux System.

/home is the home directory for Linux normal users. This directory contains personal directories for Linux normal users.

/lib directory contains libraries which will be utilized by the commands present in the directory /bin and /sbin.

/media is automatic mount point for removable devices, like CD-ROM, USB.

/mnt is the temporary mount point for removable devices. It is used by system administrators.

/opt is optional directory which is used to install third party software.

/proc is virtual filesystem and contains the information about kernel and linux processes.

/root is the home directory for root user (or superuser).

/sbin directory contains system binaries, which are the linux commands executable by Linux Administrator.

/sys is virtual filesystem and provides an interface to kernel and hardware.

/tmp directory contains temporary files which will be removed on system reboot.

/usr contains user-level applications, their binaries and documentation.

/var contains files and directories which size changes when Linux system is up.

/run directory lost the files and directories which it contains on system reboot. Contains information about currently running daemons or processes.

/dev directory contains device files for hardware interface.

Linux File Permission

Linux file permissions controls who can access and manipulate files and directories.

1. Permission Categories:

- Owner (u): The user who owns the file or directory.
- Group (g): Users who are members of the group assigned to the file or directory.
- Others (o): All other users on the system.

2. Permission Types:

- Read (r or 4):
 - For files: Allows viewing the file's contents.
 - For directories: Allows listing the contents of the directory (e.g., using ls).
- Write (w or 2):
 - For files: Allows modifying or deleting the file.
 - For directories: Allows creating, deleting, or renaming files within the directory, and modifying files within the directory if the user also has write permissions to those specific files.
- Execute (x or 1):
 - For files: Allows running the file as a program or script.
 - For directories: Allows accessing or traversing into the directory (e.g., using cd) and accessing metadata about its contents.

Permissions are typically viewed using the ls -l command. It is represented as a 10-character string:

-rwxrwxrwx

```
[root@Mederma-Server medoko]# chmod 755 myfile.sh
```

Grant read, write, and execute permissions to the owner, and read and execute to group and others as shown in the screenshot attached above.

```
[root@Mederma-Server medoko]# ls -l
total 0
-rwxr-xr-x. 1 root root 0 Nov  3 17:15 myfile.sh
```

Ownership Management

The **chown** command changes the owner of a file or directory, and **chgrp** changes the group ownership, -R is used to assign permission recursively.

```
[root@Mederma-Server ~]# chown -R john bali
[root@Mederma-Server ~]# chgrp -R maxo bali/
[root@Mederma-Server ~]# ls -ld bali/
drwxr-xr-x. 2 john maxo 4096 Nov  3 17:23 bali/
[root@Mederma-Server ~]# ls -ltr bali/
total 0
-rw-r--r--. 1 john maxo 0 Nov  3 17:23 file12.txt
-rw-r--r--. 1 john maxo 0 Nov  3 17:23 file11.txt
-rw-r--r--. 1 john maxo 0 Nov  3 17:23 file10.txt
-rw-r--r--. 1 john maxo 0 Nov  3 17:23 file20.txt
-rw-r--r--. 1 john maxo 0 Nov  3 17:23 file19.txt
-rw-r--r--. 1 john maxo 0 Nov  3 17:23 file18.txt
-rw-r--r--. 1 john maxo 0 Nov  3 17:23 file17.txt
-rw-r--r--. 1 john maxo 0 Nov  3 17:23 file16.txt
-rw-r--r--. 1 john maxo 0 Nov  3 17:23 file15.txt
-rw-r--r--. 1 john maxo 0 Nov  3 17:23 file14.txt
-rw-r--r--. 1 john maxo 0 Nov  3 17:23 file13.txt
```

Instead of using **chown** and **chgrp** command the same ownership could be achieved by using **chown** command as shown below.

```
[root@Mederma-Server ~]# chown -R yokimov:nobellauraete nanofibers/
[root@Mederma-Server ~]# ls -ld nanofibers/
drwxr-xr-x. 2 yokimov nobellauraete 177 Nov  3 17:31 nanofibers/
[root@Mederma-Server ~]# ls -ltr nanofibers/
total 0
-rw-r--r--. 1 yokimov nobellauraete 0 Nov  3 17:31 tech9.txt
-rw-r--r--. 1 yokimov nobellauraete 0 Nov  3 17:31 tech8.txt
-rw-r--r--. 1 yokimov nobellauraete 0 Nov  3 17:31 tech7.txt
-rw-r--r--. 1 yokimov nobellauraete 0 Nov  3 17:31 tech6.txt
-rw-r--r--. 1 yokimov nobellauraete 0 Nov  3 17:31 tech5.txt
-rw-r--r--. 1 yokimov nobellauraete 0 Nov  3 17:31 tech4.txt
-rw-r--r--. 1 yokimov nobellauraete 0 Nov  3 17:31 tech3.txt
-rw-r--r--. 1 yokimov nobellauraete 0 Nov  3 17:31 tech2.txt
-rw-r--r--. 1 yokimov nobellauraete 0 Nov  3 17:31 tech10.txt
-rw-r--r--. 1 yokimov nobellauraete 0 Nov  3 17:31 tech1.txt
```

Manage Access Control List in Linux for files and directories

The **setfacl** command in Linux is used to manage Access Control Lists (ACLs) for files and directories, it provides more granular permission than standard chmod permissions.

-m (modify) modifies or adds an ACL entry

setfacl -m u:username:<permissions> <filename>

setfacl -m u:john:rw dexter.txt, gives user john read and write permission for the file dexter.txt.

```
[root@Mederma-Server ~]# setfacl -m u:john:rw dexter.txt
[root@Mederma-Server ~]# getfacl dexter.txt
# file: dexter.txt
# owner: root
# group: root
user::rw-
user:john:rw-
group::r--
mask::rw-
other::r--
```

-x (remove) Removes a specific ACL Permission.

setfacl -x u:username <filename>

setfacl -x u:john dexter.txt, removes john's ACL permission.

```
[root@Mederma-Server ~]# setfacl -x u:john dexter.txt
[root@Mederma-Server ~]# getfacl dexter.txt
# file: dexter.txt
# owner: root
# group: root
user::rw-
group::r--
mask::r--
other::r--
```

User management in Linux

A new user can be added using the command as shown below. Where **-m** flag will create the home directory if already does not exist and **-s** flag will be used for login shell. Passwd command is used to provide the password for the user.

```
[root@Mederma-Server ~]# useradd -m -s /bin/bash xikonov
[root@Mederma-Server ~]# passwd xikonov
Changing password for user xikonov.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

```
[john@Mederma-Server ~]$ su - xikonov
Password:
[xikonov@Mederma-Server ~]$
```

User details are present in /etc/passwd and its encrypted passwd is present in /etc/shadow file. The file /etc/shadow can only be open by root (superuser) user.

Using **usermod** I had added the existing user xikonov to the group nobellauraete as shown in the screenshot attached below.

```
[root@Mederma-Server ~]# usermod -aG nobellauraete xikonov
[root@Mederma-Server ~]# getent group nobellauraete
nobellauraete:x:1004:xikonov
```

To delete a user with its home directory you can use the command **userdel -r <username>**.

```
[root@Mederma-Server ~]# userdel -r xikonov
```

To delete the group, you can use the command **groupdel <groupname>**

```
[root@Mederma-Server ~]# groupdel nobellauraete
```

If Linux Administrator wants to lock the user account, then the **usermod john --lock** command should be used as shown below. Now the user will not be able to login using his password.

```
[root@Mederma-Server ~]# usermod john --lock
```

```
[ritesh@Mederma-Server ~]$ su - john
Password:
su: Authentication failure
```

After unlock the user account the user can login again as shown in the screenshot attached below.

```
[root@Mederma-Server ~]# usermod john --unlock
```

```
[ritesh@Mederma-Server ~]$ su - john
Password:
[john@Mederma-Server ~]$
```

Here, I am showing you the use case when all the users are only allowed with the SSH Keys and password is disabled to login on the server.

In such a situation you can generate the SSH Keys (public and private keys) using the command as shown below.

```
[root@herema-server ~]# ssh-keygen
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: [REDACTED] A root@herema-server
The key's randomart image is:
+--[RSA 3072]----+
| [REDACTED] |
+---[SHA256]-----+
```

The user who run **ssh-keygen** command, in the home directory of that user in the **.ssh** directory private key (**id_rsa**) and public key (**id_rsa.pub**) had been generated.

Then using **ssh-copy-id** command copy the public key to remote server which remote session you are interested into using SSH.

```
[root@herema-server ~]# ssh-copy-id john@192.168.40.227
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
john@192.168.40.227's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'john@192.168.40.227'"
and check to make sure that only the key(s) you wanted were added.

[root@Mederma-Server ~]# cat /home/john/.ssh/authorized_keys
ssh-rsa A
```

Then disable password authentication in the file **sshd_config** present at the absolute path **/etc/ssh/sshd_config** which remote session you are interested into.

```
[root@Mederma-Server ~]# vim /etc/ssh/sshd_config
```

Change **PasswordAuthentication** from **yes** to **no**.

```
    .
    .
    .
PasswordAuthentication no
```

```
[root@Mederma-Server ~]# systemctl reload sshd
```

The user who wants to login into the server must have the private key.

```
ritesh@DESKTOP-0 F:~$ chmod 600 dexter.pem
ritesh@DESKTOP-0 F:~$ ssh -i dexter.pem john@192.168.40.227
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Mon Nov  3 20:24:22 2025 from 192.168.40.1
[john@Mederma-Server ~]$
```

Process Management in Linux

Linux **Processes** are programs which is running commands and utilizing resources. Each process is identified by a unique Process ID. To list the processes, you can run the command **ps -ef**. If you want to search any process using specific string then use the command **ps -ef | grep <string-name>**.

```
[root@Mederma-Server ~]# ps -ef
UID      PID  PPID  C STIME TTY          TIME CMD
root      1      0  0 11:22 ?
root      2      0  0 11:22 ?
root      3      2  0 11:22 ?
root      4      2  0 11:22 ?
root      5      2  0 11:22 ?
root     10      2  0 11:22 ?
root     11      2  0 11:22 ?
root     12      2  0 11:22 ?
root     13      2  0 11:22 ?
root     14      2  0 11:22 ?
root     15      2  0 11:22 ?
root     16      2  0 11:22 ?
root     17      2  0 11:22 ?
root     18      2  0 11:22 ?
root     19      2  0 11:22 ?
root     20      2  0 11:22 ?
root     21      2  0 11:22 ?
root     26      2  0 11:22 ?
root     27      2  0 11:22 ?
root     28      2  0 11:22 ?
root     30      2  0 11:22 ?
root     31      2  0 11:22 ?
root     32      2  0 11:22 ?
root     33      2  0 11:22 ?
root     34      2  0 11:22 ?
root     35      2  0 11:22 ?
root     36      2  0 11:22 ?
root     37      2  0 11:22 ?
root     38      2  0 11:22 ?
root     39      2  0 11:22 ?
root     40      2  0 11:22 ?
root     41      2  0 11:22 ?
root     42      2  0 11:22 ?
root     43      2  0 11:22 ?
root     44      2  0 11:22 ?
root     46      2  0 11:22 ?
root     63      2  0 11:22 ?

[root@Mederma-Server ~]# ps -ef|grep -i "ssh"
root    1144      1  0 11:23 ?
          0:00:00 /usr/sbin/sshd -D -oCiphers=aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-ctr,aes256-cbc,aes128-gcm@openssh.com,aes128-ctr,aes128-cbc -oMACs=hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512-etm@openssh.co
m,hmac-sha2-256,hmac-sha1,umac-128@openssh.com,hmac-sha2-512 -oGSSAPIKeyAlgorithms=gsx-curve25519-sha256-,gsx-nistp256-sha256-,gsx-group14-sha256+,gsx-group16-sha512-,gsx-gex-sha1-gss-group14-sha1- -oKexAlgorithms=curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group14-sha256,diffie-hellman-group18-sha512,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1 -oHostKeyAlgorithms=ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384,ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp521,ecdsa-sha2-nistp251-cert-v01@openssh.com,ssh-ed25519,ssh-ed25519-cert-v01@openssh.com,rsa-sha2-256,rsa-sha2-256-crt-v0
1@openssh.com,rsa-sha2-512,rsa-sha2-512-cert-v01@openssh.com,ssh-rsa,ssh-rsa-cert-v01@openssh.com -oPubKeyAcceptedKeyTypes=ecdsa-sha2-nistp256,ecdsa-sha2-nistp521,ecdsa-sha2-nistp251-cert-v01@openssh.com,ssh-ed25519,rsa-sha2-256,rsa-sha2-512,rsa-sha2-512-cert-v01@openssh.com,ssh-rsa,ssh-rsa-cert-v01@open
sh.com -oCASignatureAlgorithms=ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-ed25519,rsa-sha2-256,rsa-sha2-512,ssh-rsa
root    4143   1144  0 12:26 ?
          0:00:00 sshd: ritesh [priv]
ritesh   4147   4143  0 12:26 ?
          0:00:00 sshd: ritesh@pts/1
root    4257   4189  0 12:30 pts/1
          0:00:00 grep --color=auto -i ssh
```

You can see in the above command I had used **-i** with grep which means perform search with case-insensitive.

To list all running processes on Linux system you can run the command **ps aux** as shown below.

```
[root@Mederma-Server ~]# ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      1  0.0  0.3 175848 14064 ?        Ss 11:22  0:04 /usr/lib/systemd/systemd --switched-root --system --deserialize 18
root      2  0.0  0.0      0   0 ?        S 11:22  0:00 [kthreadd]
root      3  0.0  0.0      0   0 ?        I< 11:22  0:00 [rcu_gp]
root      4  0.0  0.0      0   0 ?        I< 11:22  0:00 [rcu_par_gp]
root      5  0.0  0.0      0   0 ?        I< 11:22  0:00 [slub_flushwq]
root     10  0.0  0.0      0   0 ?        I< 11:22  0:00 [mm_percpu_wq]
root     11  0.0  0.0      0   0 ?        S 11:22  0:00 [rcu_tasks_rude_]
root     12  0.0  0.0      0   0 ?        S 11:22  0:00 [rcu_tasks_trace]
root     13  0.0  0.0      0   0 ?        S 11:22  0:00 [ksoftirqd/0]
root     14  0.0  0.0      0   0 ?        I 11:22  0:00 [rcu_sched]
root     15  0.0  0.0      0   0 ?        S 11:22  0:00 [migration/0]
root     16  0.0  0.0      0   0 ?        S 11:22  0:00 [watchdog/0]
root     17  0.0  0.0      0   0 ?        S 11:22  0:00 [cpuhp/0]
root     18  0.0  0.0      0   0 ?        S 11:22  0:00 [cpuhp/1]
root     19  0.0  0.0      0   0 ?        S 11:22  0:00 [watchdog/1]
root     20  0.0  0.0      0   0 ?        S 11:22  0:00 [migration/1]
root     21  0.0  0.0      0   0 ?        S 11:22  0:00 [ksoftirqd/1]
root     26  0.0  0.0      0   0 ?        S 11:22  0:00 [kdevtmpfs]
root     27  0.0  0.0      0   0 ?        I< 11:22  0:00 [netns]
root     28  0.0  0.0      0   0 ?        S 11:22  0:00 [kauditfd]
root     30  0.0  0.0      0   0 ?        S 11:22  0:00 [khungtaskd]
root     31  0.0  0.0      0   0 ?        S 11:22  0:00 [oom_reaper]
root     32  0.0  0.0      0   0 ?        I< 11:22  0:00 [writeback]
root     33  0.0  0.0      0   0 ?        S 11:22  0:00 [kcompactd0]
root     34  0.0  0.0      0   0 ?        SN 11:22  0:00 [ksmd]
root     35  0.0  0.0      0   0 ?        SN 11:22  0:02 [khugepaged]
root     36  0.0  0.0      0   0 ?        I< 11:22  0:00 [crypto]
root     37  0.0  0.0      0   0 ?        I< 11:22  0:00 [kintegrityd]
root     38  0.0  0.0      0   0 ?        I< 11:22  0:00 [kblockd]
```

To list processes and resource utilization you can use the command **top** or **htop** but htop provides a feature-rich experience. In Almalinux by-default htop command is not installed you need to install it as shown below. It is a useful command for **monitoring** or **Root Cause Analysis (RCA)**. You can find out which process is utilizing maximum resource and can kill the same using **kill** command if the process is of no use at present point of time. If that process is important to run and no resource is free then scale up (vertical scaling) the resources of your Linux machine. This activity should be done during non-production hours. However, if you are using a cloud environment like **AWS, Azure, or GCP (Google Cloud Platform)** then it is suggested to perform Autoscale (Scale out or horizontal scaling) your EC2 Instances or VM Instances.

```
[root@Mederma-Server ~]# yum install -y htop
```

```
[root@Mederma-Server ~]# htop
```

Main [I/O]									
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%[MEM%
4922	root	20	0	225M	4612	3196	R	3.2	0.1
1132	root	20	0	592M	22324	17412	S	0.6	0.6
1146	root	20	0	606M	33980	15476	S	0.6	0.9
1 root	root	20	0	171M	14084	8944	S	0.0	0.4
769	root	20	0	193M	12880	9732	S	0.0	0.3
810	root	20	0	99M	13240	7820	S	0.0	0.4
811	root	20	0	184M	668	12	S	0.0	0.0
812	root	20	0	184M	668	12	S	0.0	0.0
813	root	20	0	184M	668	12	S	0.0	0.0
989	rpc	20	0	67332	5416	4688	S	0.0	0.1
910	root	16	-4	127M	2292	1652	S	0.0	0.0
911	root	16	-4	127M	2292	1652	S	0.0	0.1
916	root	16	-4	48728	3280	2920	S	0.0	0.1
917	root	16	-4	127M	2292	1652	S	0.0	0.1
943	root	20	0	79284	6844	6064	S	0.0	0.2
945	root	20	0	122M	4700	4132	S	0.0	0.49
946	root	20	0	6760	1952	1788	S	0.0	0.1
949	root	20	0	122M	4700	4132	S	0.0	0.0
951	libstorage	20	0	8692	1712	1560	S	0.0	0.0
952	polkitd	20	0	1574M	28236	18756	S	0.0	0.8
953	root	20	0	278M	11376	9696	S	0.0	0.3
954	root	20	0	549M	11384	9620	S	0.0	0.3
956	root	20	0	50488	5332	4268	S	0.0	0.1
957	rktit	21	1	187M	3416	3076	S	0.0	0.1
959	root	20	0	532M	15280	12952	S	0.0	0.4
960	avahi	20	0	58360	4556	4112	S	0.0	0.1
962	chrony	20	0	136M	4488	3764	S	0.0	0.12
963	root	20	0	509M	11684	8304	S	0.0	0.3
964	root	39	19	17636	1684	1456	S	0.0	0.0
965	root	20	0	83532	9640	7484	S	0.0	0.3

Managing priority in Linux

In Linux when we execute a command a priority getting assigned to that (when we execute a command in Linux then kernel assigns a priority to that) a command or process with higher priority will be executed first. You can priorities the priority of a command execution using the nice command.

```
nice -n <nice_value> <command>
```

Lower is the nice_value and higher will be priority, among two command **nice -n 10 bash test.sh** and **nice -n 5 bash demo.sh**. Second command priority will be more than first command. For its execution I opened three terminals and on one terminal I ran **nice -n 10 bash test.sh** and on another terminal, I ran **nice -n 5 bash demo.sh**. On third terminal I ran **ps -lef | grep sleep** as shown below.

```
[root@Mederma-Server ~]# nice -n 10 bash test.sh
```

```
[root@Mederma-Server ~]# nice -n 5 bash demo.sh
```

```
[root@Mederma-Server ~]# ps -efl | grep bash
```



```
0 R root      6255  5749 26 90 10 - 57441 -      15:24 pts/4    00:00:17 bash test.sh
0 S root      6256  5670 35 85 5 - 57441 -      15:24 pts/3    00:00:22 bash demo.sh
```

As shown in the screenshot attached above a process with **PRI** value **85** has higher priority than a process with **PRI** value **90**. Which means kernel will listen first **bash demo.sh** and then **bash test.sh**.

It is possible to change the priority of a running process or command using **renice** command as shown in the screenshot attached below.

renice <priority> <PID>

```
[root@Mederma-Server ~]# bash test.sh
```

```
[root@Mederma-Server ~]# bash demo.sh
```

```
[root@Mederma-Server ~]# ps -efl | grep bash
0 S root      6353  5749 62 80  0 - 57441 -          15:54 pts/4    00:00:04 bash test.sh
0 R root      6354  5670 35 80  0 - 57441 -          15:54 pts/3    00:00:01 bash demo.sh
0 R root      6356  6120  0 80  0 - 57294 -          15:54 pts/5    00:00:00 grep --color=auto bash
[root@Mederma-Server ~]# renice 5 6354
6354 (process ID) old priority 0, new priority 5
[root@Mederma-Server ~]# renice 2 6353
6353 (process ID) old priority 0, new priority 2
[root@Mederma-Server ~]# ps -efl | grep bash
0 S root      6353  5749 46 82  2 - 57441 -          15:54 pts/4    00:00:26 bash test.sh
0 R root      6354  5670 33 85  5 - 57441 -          15:54 pts/3    00:00:18 bash demo.sh
0 R root      6360  6120  0 80  0 - 57294 -          15:55 pts/5    00:00:00 grep --color=auto bash
```

As explain in the attached screenshot I executed two shell scripts **bash test.sh** and **bash demo.sh** and changed its priority using renice command and finally found that its priority had been changed as expected.

There is command **sleep 60** (Sleep for 60 Seconds) is running and you want to kill this process, the same can be achieved with the command as shown below.

kill -9 <PID>

```
[root@Mederma-Server ~]# sleep 60
[root@Mederma-Server ~]# ps -ef|grep sleep
root      6381  5749  0 16:06 pts/4    00:00:00 sleep 60
root      6383  6120  0 16:06 pts/5    00:00:00 grep --color=auto sleep
[root@Mederma-Server ~]# kill -9 6381
[root@Mederma-Server ~]# ps -ef|grep sleep
root      6385  6120  0 16:06 pts/5    00:00:00 grep --color=auto sleep
```

Cronjob in Linux

Cron Job is used in Linux to schedule the periodic task. While scheduling task using cronjob we use five fields **minute, hour, day, month, and day-of-a-week** (left to right in crontab -e).

<u>Fields</u>	<u>Range of Value</u>
---------------	-----------------------

minute:	0-59
hour:	0-23
day:	1-31
month:	1-12

day-of-a-week: 0-7 where 0 and 7 used for Sunday.

I am explaining cronjob using a use case, consider that in your project you are using an application which is generating shared files and libraries at the path **/mederma** which will be of no use after one month and there is no need to take its backup and you can simply delete these files and libraries on 28th of each month at 12:00 PM, this can be achieved using crontab as shown below.

```
[root@Mederma-Server ~]# crontab -l
0 12 28 * * sh dexter.sh

[root@Mederma-Server ~]# cat dexter.sh
#!/bin/bash

df -hT | grep "Filesystem"           Type      Size   Used  Avail Use% Mounted on" > therema.txt
df -hT | grep "/dev/sd" >> therema.txt
df -hT | grep "/dev/mapper" >> therema.txt
cat therema.txt

rm -rf /mederma/* ### Removes all the files and directories present in the directory /mederma/
[root@Mederma-Server ~]# cat therema.txt
Filesystem            Type      Size   Used  Avail Use% Mounted on
/dev/sda1              xfs     1014M  282M  733M  28% /boot
/dev/sdb               xfs      10G   104M  9.9G   2% /mederma
/dev/mapper/almalinux-root xfs      46G   36G  9.9G  79% /
```

You can check the logs for cron as shown below.

```
[root@Mederma-Server ~]# vim /var/log/cron
```

```

Nov  4 17:21:01 Mederma-Server CROND[3314]: (root) CMD (sh dexter.sh)
Nov  4 17:21:01 Mederma-Server CROND[3295]: (root) CMDOUT (Filesystem)
Nov  4 17:21:01 Mederma-Server CROND[3295]: (root) CMDOUT (/dev/sda1)          Type      Size   Used  Avail Use% Mounted on
Nov  4 17:21:01 Mederma-Server CROND[3295]: (root) CMDOUT (/dev/sdb)          xfs       1014M  282M  733M  28% /boot)
Nov  4 17:21:01 Mederma-Server CROND[3295]: (root) CMDOUT (/dev/mapper/almalinux-root xfs       10G    104M  9.9G  2% /mederma)
Nov  4 17:21:02 Mederma-Server crontab[3324]: (root) BEGIN EDIT (root)
Nov  4 17:21:09 Mederma-Server crontab[3324]: (root) REPLACE (root)
Nov  4 17:21:09 Mederma-Server crontab[3324]: (root) END EDIT (root)
Nov  4 17:21:13 Mederma-Server crontab[3331]: (root) LIST (root)

```

After successfully running the cronjob I checked the /mederma directory and found no files or directory was present as shown in the screenshot attached below.

```
[root@Mederma-Server ~]# ll /mederma/
total 0
```

Boot Process in Linux

Linux boot process is nothing but loading the Operating system and making the Linux machine On. Below are the steps involved in booting process.

1. Initialization of BIOS (Basic Input/Output System).
2. Load the bootloader (in RHEL7 or later versions GRUB2 is the bootloader).
3. bootloader loads the Linux kernel and initializes initramfs.
4. Initialize kernel and mount the root filesystem.
5. Execute init process, init process is the first process in Linux with PID 1.
6. Finally, runlevels will be executed. There are seven runlevels in Linux system **Runlevel 0, Runlevel 1, Runlevel 2, Runlevel 3, Runlevel 4, Runlevel 5 and Runlevel 6**.

Runlevel 0: system halt or shutdown

Runlevel 1: Single-user mode without networking.

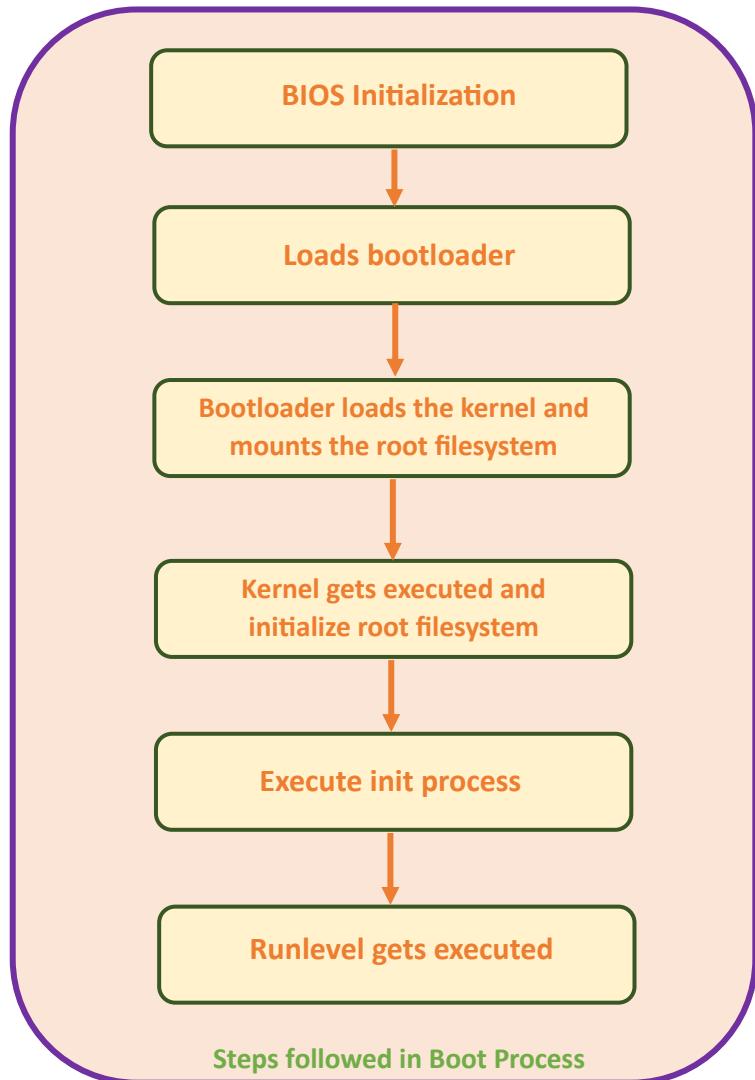
Runlevel 2: Multi-user mode without networking.

Runlevel 3: Multi-user mode with networking

Runlevel 4: User-definable or unused

Runlevel 5: GUI mode

Runlevel 6: Reboot the system.



File attributes in Linux

File attributes are metadata properties that provide additional information about the file or directories.

chattr command: - **chattr** command in linux is used to change the attribute of a file or directory in Linux and hence allow who can edit the file or delete the file or directory. With chattr command it is possible to use two flags, **i** and **a** flag. When **i** flag is used no one can edit or delete the file or directory and when **a** flag is used then no one can delete the file, only append the file.

To list the attributes, you can use the command **lsattr**. You can use the flags -a (all files and directories including hidden files and directories), -d (at the directory level), -R (recursively) with **lsattr** command.

```
[root@Mederma-Server ~]# mkdir dexter
[root@Mederma-Server ~]# touch dexter/file{1..10}.txt
[root@Mederma-Server ~]# chattr -R +i dexter
[root@Mederma-Server ~]# lsattr -d dexter
----i----- dexter
[root@Mederma-Server ~]# lsattr -R dexter
----i----- dexter/file1.txt
----i----- dexter/file2.txt
----i----- dexter/file3.txt
----i----- dexter/file4.txt
----i----- dexter/file5.txt
----i----- dexter/file6.txt
----i----- dexter/file7.txt
----i----- dexter/file8.txt
----i----- dexter/file9.txt
----i----- dexter/file10.txt
[root@Mederma-Server ~]# rm -f dexter/file10.txt
rm: cannot remove 'dexter/file10.txt': Operation not permitted
[root@Mederma-Server ~]# rm -rf dexter
rm: cannot remove 'dexter/file1.txt': Operation not permitted
rm: cannot remove 'dexter/file2.txt': Operation not permitted
rm: cannot remove 'dexter/file3.txt': Operation not permitted
rm: cannot remove 'dexter/file4.txt': Operation not permitted
rm: cannot remove 'dexter/file5.txt': Operation not permitted
rm: cannot remove 'dexter/file6.txt': Operation not permitted
rm: cannot remove 'dexter/file7.txt': Operation not permitted
rm: cannot remove 'dexter/file8.txt': Operation not permitted
rm: cannot remove 'dexter/file9.txt': Operation not permitted
rm: cannot remove 'dexter/file10.txt': Operation not permitted
```

When I removed the **i** attribute, I was able to delete the directory as shown in the screenshot attached below.

```
[root@Mederma-Server ~]# chattr -R -i dexter
[root@Mederma-Server ~]# lsattr -d dexter
----- dexter
[root@Mederma-Server ~]# lsattr -R dexter
----- dexter/file1.txt
----- dexter/file2.txt
----- dexter/file3.txt
----- dexter/file4.txt
----- dexter/file5.txt
----- dexter/file6.txt
----- dexter/file7.txt
----- dexter/file8.txt
----- dexter/file9.txt
----- dexter/file10.txt
[root@Mederma-Server ~]# rm -rf dexter
```

```
[root@Mederma-Server ~]# mkdir mederma
[root@Mederma-Server ~]# touch mederma/dexo{1..5}.txt
[root@Mederma-Server ~]# chattr +a mederma/*
[root@Mederma-Server ~]# rm -rf mederma
rm: cannot remove 'mederma/dexo1.txt': Operation not permitted
rm: cannot remove 'mederma/dexo2.txt': Operation not permitted
rm: cannot remove 'mederma/dexo3.txt': Operation not permitted
rm: cannot remove 'mederma/dexo4.txt': Operation not permitted
rm: cannot remove 'mederma/dexo5.txt': Operation not permitted

[root@Mederma-Server ~]# echo "Hi, Hello" > mederma/dexo1.txt
-bash: mederma/dexo1.txt: Operation not permitted
[root@Mederma-Server ~]# echo "Hi, Hello" >> mederma/dexo1.txt
[root@Mederma-Server ~]# cat mederma/dexo1.txt
Hi, Hello
```

As shown above after applying **a** attribute to the files it was allowed to append into the files but not to delete the files.

```
[root@Mederma-Server ~]# chattr -a mederma/*
[root@Mederma-Server ~]# rm -rf mederma
[root@Mederma-Server ~]#
```

Linux inode

Inode in Linux also called as **index node** will give the address where actual data block is stored. Every inode is associated with inode number. When you assign a filesystem to a partition in Linux and mount it on a directory then an **inode table** will be created. When you create a file or directory where that partition is mounted then an inode number is associated with that and inode number will be kept in the inode table. To check the details of inode you can use the command as shown below.

```
[root@Mederma-Server ~]# df -ihT
Filesystem           Type      Inodes  IUsed   IFree  IUse% Mounted on
devtmpfs             devtmpfs    447K    416   447K    1% /dev
tmpfs                tmpfs       455K     1   455K    1% /dev/shm
tmpfs                tmpfs       455K    926   454K    1% /run
tmpfs                tmpfs       455K     17   455K    1% /sys/fs/cgroup
/dev/mapper/almalinux-root xfs        20M   128K   20M    1% /
/dev/sdb              xfs        5.0M     3   5.0M    1% /mederma
/dev/sda1             xfs        512K   311   512K    1% /boot
tmpfs                tmpfs       455K     27   455K    1% /run/user/1000
```

```
[root@Mederma-Server ~]# ls -li /mederma/
total 0
131 -rw-r--r--. 1 root root 0 Nov  6 13:53 therema1.txt
132 -rw-r--r--. 1 root root 0 Nov  6 13:53 therema2.txt
133 -rw-r--r--. 1 root root 0 Nov  6 13:53 therema3.txt
134 -rw-r--r--. 1 root root 0 Nov  6 13:53 therema4.txt
135 -rw-r--r--. 1 root root 0 Nov  6 13:53 therema5.txt
```

As shown in the screenshot attached above the inode of the files inside the mederma directory is listed with the numbers 131, 132, 133, 134 and 135.

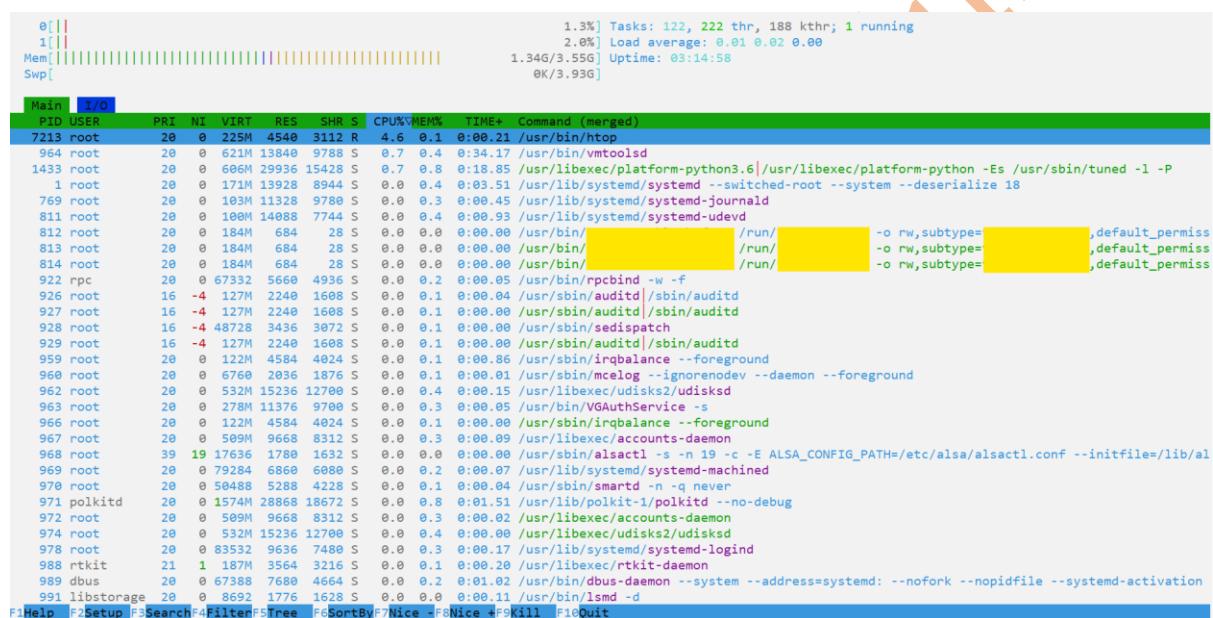
Performance Monitoring

Linux provides a wide variety of commands for performance monitoring, some of the commands are as written below.

htop (to use htop you must need to install the package htop), **mpstat** (to use mpstat you must need to install sysstat package), **iftop** (to install iftop you must need to install the package iftop), **free**, **vmstat**, **uptime**, **du** and **lsof**.

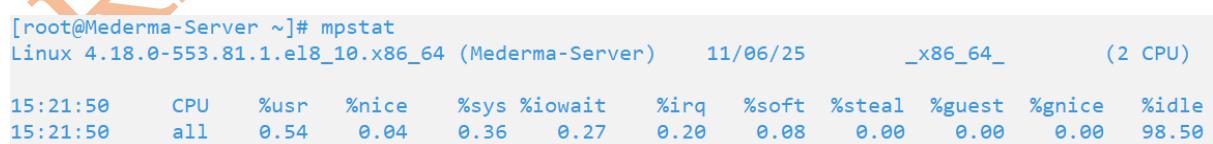
Using htop command you can continuously monitor system resources as shown in the screenshot attached below.

[root@Mederma-Server ~]# htop



To monitor Linux system performance and resource usage you can use mpstat command. To use mpstat command you need to install the package **sysstat**. The mpstat command is used for processor related statistics, individual core usage, and CPU metrics.

[root@Mederma-Server ~]# yum install -y sysstat



To display network traffic, you can use the command **iftop**.

[root@Mederma-Server ~]# yum install -y iftop

[root@Mederma-Server ~]# iftop

```

 12.5Kb          25.0Kb          37.5Kb          50.0Kb          62.5Kb
Mederma-Server      => 192.168.40.1
Mederma-Server      <=
Mederma-Server      => _gateway
Mederma-Server      <=


TX: [REDACTED] cum: 1.55KB peak: 3.31Kb
RX: [REDACTED] 630B 2.10Kb
TOTAL: 2.16KB 4.97Kb

rates: 2.87Kb 3.09Kb 3.09Kb
       2.10Kb 1.23Kb 1.23Kb
       4.97Kb 4.32Kb 4.32Kb

```

```
[root@Mederma-Server ~]# iftop
interface: ens33
IP address is: 192.168.40.227
MAC address is: [REDACTED]
```

The **free -mh** command is used in Linux to see the **memory** and **swap** usage as shown in the screenshot attached below.

```
[root@Mederma-Server ~]# free -mh
              total        used        free      shared  buff/cache   available
Mem:      3.5Gi       1.3Gi       1.1Gi      21Mi       1.1Gi       1.9Gi
Swap:    3.9Gi        0B       3.9Gi
```

To see the **swap** and **memory** statistics you can use the command **vmstat** as shown in the screenshot attached below.

```
[root@Mederma-Server ~]# vmstat
procs -----memory----- --swap-- -----io---- -system-- -----cpu-----
r b  swpd  free  buff  cache  si  so  bi  bo  in  cs us sy id wa st
1 0     0 1107652  3352 1196388   0   0   37   12   73  110  1  1 98  0  0
```

The **uptime** command in linux is used to show from how much time the server is up as shown in the screenshot attached below.

```
[root@Mederma-Server ~]# uptime
15:48:55 up 3:47, 4 users, load average: 0.08, 0.08, 0.03
```

The **du** command is used to show the disk usage (how much space is occupied) as shown in the screenshot attached below.

```
[root@Mederma-Server ~]# du -sh /root
29G  /root
```

As shown in the screenshot attached above it shows how much space is occupied for /root directory.

The **lsof** command in linux is used to check which process has opened the file as shown in the screenshot attached below. It is specially used when you want to delete any file and it will throw the error that file is in use then you can use the command to check which process is utilizing it. Kill those processes and then delete the file.

```
[root@Mederma-Server ~]# lsof /mederma/therema1.txt
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1000/gvfs
      Output information may be incomplete.
COMMAND  PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
sh    8700 root    1w   REG  8,16     4152  131 /mederma/therema1.txt
sh    8769 root    1w   REG  8,16    4458  131 /mederma/therema1.txt
sh    8820 root    1w   REG  8,16    4716  131 /mederma/therema1.txt
```

The **iostat** command provides the I/O statistics for block devices and CPUs as shown in the screenshot attached below.

```
[root@Mederma-Server ~]# iostat
Linux 4.18.0-553.81.1.el8_10.x86_64 (Mederma-Server) 11/06/25 _x86_64_ (2 CPU)

avg-cpu: %user %nice %system %iowait %steal %idle
          5.32    0.03   7.56    0.56    0.00  86.52

Device      tps   kB_read/s   kB_wrtn/s   kB_read   kB_wrtn
sda       1.41     60.31      20.35    990172    334047
sdb      68.68      0.20     259.03     3247   4253013
scd0      0.00      0.00      0.00        2        0
dm-0      1.36     56.99      20.22   935694    331913
dm-1      0.01      0.14      0.00     2220        0
```

To install **logrotate** the required package is **logrotate**, I used Almalinux 8.10 and it is by default installed on it. as shown in the screenshot attached below.

```
[root@Mederma-Server ~]# rpm -qa|grep -i "logrotate*"
logrotate-3.14.0-6.el8.x86_64
```

Applications generate the logs in logfiles, there will be time when size of these logfiles may be very large and hence affect the performance of the application. So, you can use **Logrotate** to rotate the logfiles and drop them periodically as explained below.

```
[root@Mederma-Server ~]# mkdir /var/log/httpd/newlog
```

Here rotate is 3 which means last three rotated logs will be kept and others are dropped. For rotated logfiles the extension used will be in the format of day, month, year, hour, minute and second, notifempty is used which means do not rotate the logfiles if logfile is empty. Compress the logfiles after rotation and store at the absolute path provided by olddir as shown below. Execute the commands as provided below under postrotate.

```
[root@Mederma-Server ~]# cat /etc/logrotate.d/httpd.conf
/var/log/httpd/* {
    rotate 3
    dateext
    dateformat -%d%m%Y-%H%M%S
   notifempty
    compress
    olddir /var/log/httpd/newlog/
    postrotate
        echo "A rotation just took place."
        systemctl reload httpd
    endscript
}
```

```
[root@Mederma-Server ~]# crontab -e
crontab: installing new crontab
[root@Mederma-Server ~]# crontab -l
0 12 14,28 * * /sbin/logrotate -f /etc/logrotate.d/httpd.conf
```

Forcefully execute the logrotate using crontab as shown in the screenshot attached above. It will be executed on 14th and 28th of every month at 12:00 PM.

How to create a Service in Linux

Here I had created a service in Linux to execute a shell script as shown in screenshot attached below.

```
[root@Mederma-Server ~]# cat /opt/dexter.sh
echo "My custom service is running!" >> /var/log/dexter_service.log

[root@Mederma-Server ~]# cat /etc/systemd/system/dexter.service
[Unit]
Description=Dexter Service
After=network.target

[Service]
ExecStart=/bin/bash /opt/dexter.sh
Type=simple
#User=root ### Run as a specific user
#Group=root ### Run as a specific group
Restart=on-failure
RestartSec=60

[Install]
WantedBy=multi-user.target

[root@Mederma-Server ~]# systemctl daemon-reload

[root@Mederma-Server ~]# systemctl start dexter.service
```

```
[root@Mederma-Server ~]# systemctl status dexter.service
● dexter.service - Dexter Service
   Loaded: loaded (/etc/systemd/system/dexter.service; disabled; vendor preset: disabled)
     Active: inactive (dead)

Nov 06 19:21:15 Mederma-Server systemd[1]: dexter.service: Failed with result 'exit-code'.
Nov 06 19:22:15 Mederma-Server systemd[1]: dexter.service: Service RestartSec=1min expired, scheduling restart.
Nov 06 19:22:15 Mederma-Server systemd[1]: dexter.service: Scheduled restart job, restart counter is at 2.
Nov 06 19:22:15 Mederma-Server systemd[1]: Stopped Dexter Service.
Nov 06 19:22:15 Mederma-Server systemd[1]: Started Dexter Service.
Nov 06 19:22:15 Mederma-Server systemd[1]: dexter.service: Main process exited, code=exited, status=217/USER
Nov 06 19:22:15 Mederma-Server systemd[1]: dexter.service: Failed with result 'exit-code'.
Nov 06 19:22:57 Mederma-Server systemd[1]: Stopped Dexter Service.
Nov 06 19:22:57 Mederma-Server systemd[1]: Started Dexter Service.
Nov 06 19:22:58 Mederma-Server systemd[1]: dexter.service: Succeeded.
```

```
[root@Mederma-Server ~]# cat /var/log/dexter_service.log
My custom service is running!
```

Different component used in the file **/etc/systemd/system/dexter.service** is as explained below.

Description: - Description about the service

After=network.target Start the network management stack

ExecStart will talk about the shell script to be executed

Type will say about how service has started or stopped; type of the service is simple which means Systemd considers the service started immediately after the main service process is forked.

WantedBy=multi-user.target Start the service when the system boots into a normal, multi-user state.

This how you can create the service in Linux and start the same when needed.

Auditd

In Linux if you want to monitor system activity then you can monitor the same using audit as explained below.

First check audit package is installed or not if not then install it using the command **yum install -y audit**.

```
[root@Mederma-Server ~]# rpm -qa|grep "audit"
python3-audit-3.1.2-1.el8_10.1.x86_64
audit-3.1.2-1.el8_10.1.x86_64
audit-libs-3.1.2-1.el8_10.1.x86_64
```

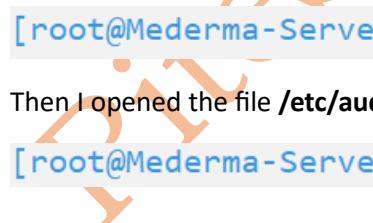
I am using **Almalinux 8.10** and audit package is already installed on that Linux machine as shown in the screenshot attached above. In the service configuration file for auditd present at the absolute path **/usr/lib/systemd/system/auditd.service**, **RefuseManualStop** was kept as **yes**. So, it was not possible to restart the **auditd** service. I changed it to **RefuseManualStop=no** reload the daemon using the command **systemctl daemon-reload** as shown in the screenshot attached below.

```
[Unit]
Description=Security Auditing Service
DefaultDependencies=no
## If auditd is sending or receiving remote logging, copy this file to
## /etc/systemd/system/auditd.service and comment out the first After and
## uncomment the second so that network-online.target is part of After.
## then comment the first Before and uncomment the second Before to remove
## sysinit.target from "Before".
After=local-fs.target systemd-tmpfiles-setup.service
##After=network-online.target local-fs.target systemd-tmpfiles-setup.service
Before=sysinit.target shutdown.target
##Before=shutdown.target
Conflicts=shutdown.target
RefuseManualStop=no
ConditionKernelCommandLine=!audit=0
ConditionKernelCommandLine=!audit=off

Documentation=man:auditd(8) https://github.com/linux-audit/audit-documentation

[Service]
Type=forking
PIDFile=/run/auditd.pid
ExecStart=/sbin/auditd
## To not use augenrules, copy this file to /etc/systemd/system/auditd.service
## and comment/delete the next line and uncomment the auditctl line.
## NOTE: augenrules expect any rules to be added to /etc/audit/rules.d/
ExecStartPost=-/sbin/augenrules --load
#ExecStartPost=-/sbin/auditctl -R /etc/audit/audit.rules
# By default we don't clear the rules on exit. To enable this, uncomment
# the next line after copying the file to /etc/systemd/system/auditd.service
#ExecStopPost=/sbin/auditctl -R /etc/audit/audit-stop.rules
Restart=on-failure
# Do not restart for intentional exits. See EXIT CODES section in auditd(8).
RestartPreventExitStatus=2 4 6

### Security Settings ###
MemoryDenyWriteExecute=true
LockPersonality=true
"/usr/lib/systemd/system/auditd.service" 47L, 1851C
```

 [root@Mederma-Server ~]# systemctl daemon-reload

Then I opened the file **/etc/audit/rules.d/audit.rules** and wrote the audit rule as shown below.

[root@Mederma-Server ~]# vim /etc/audit/rules.d/audit.rules

```
## First rule - delete all
-D

## Increase the buffers to survive stress events.
## Make this bigger for busy systems
-b 8192

## This determine how long to wait in burst of events
--backlog_wait_time 60000

## Set failure mode to syslog
-f 1

-w /etc/ssh/sshd_config -p warx -k sshd_config_changes
```

[root@Mederma-Server ~]# systemctl restart auditd

```
[root@Mederma-Server ~]# systemctl status auditd
● auditd.service - Security Auditing Service
  Loaded: loaded (/usr/lib/systemd/system/auditd.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2025-01-12 10:43:56 IST; 43min ago
    Docs: man:auditd(8)
          https://github.com/linux-audit/audit-documentation
  Process: 4074 ExecStartPost=/sbin/augenrules --load (code=exited, status=0/SUCCESS)
  Process: 4067 ExecStart=/sbin/auditd (code=exited, status=0/SUCCESS)
 Main PID: 4069 (auditd)
   Tasks: 4 (limit: 22875)
  Memory: 1.6M
   CGroup: /system.slice/auditd.service
           └─4069 /sbin/auditd
                 ├─4071 /usr/sbin/sedispatch
```

Log files for audit are kept at the absolute path **/var/log/audit/audit.log**. User **john** tried to change the log I check the same using tail -f as shown in the screenshot attached below.

[root@Mederma-Server ~]# tail -f /var/log/audit/audit.log

```
type=SYSCALL msg=audit(1680100000.000:13 aef=1001 a1=1001 a2=0 a3=0 items=1 pid=4492 pid=4596 auid=1000 uid=1001 gid=1001 euid=1001 suid=1001 fsuid=1001 egid=1001 sgid=1001 tty pts1 ses=4 comm="cat" exe="/usr/bin/cat" subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c.1823 key="sshd_config_changes" ARCH=x86_64 SYSCALL=openat AUID="ritesh" UID="john" GID="john" EUID="john" FSUID="john" EGID="john" SUID="john" FSGID="john"
type=CWD msg=audit(1680100000.000:14 aef=1001 a1=1001 a2=0 a3=0 items=1 pid=4492 pid=4596 auid=1000 uid=1001 gid=1001 euid=1001 suid=1001 fsuid=1001 egid=1001 sgid=1001 tty pts1 ses=4 comm="cat" exe="/usr/bin/cat" subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c.1823 key="sshd_config_changes" ARCH=x86_64 SYSCALL=openat AUID="ritesh" UID="john" GID="john" EUID="john" FSUID="john" EGID="john" SUID="john" FSGID="john"
type=PATH msg=audit(1680100000.000:15 aef=1001 a1=1001 a2=0 a3=0 items=1 name="/etc/ssh/sshd_config" inode=35674990 dev=fd:00 mode=01006000 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:etc_t:s0 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0UID="root" Ogid="root"
type=PROCTITLE msg=audit(1680100000.000:16 aef=1001 a1=1001 a2=0 a3=0 items=1 pid=4492 pid=4596 auid=1000 uid=1001 gid=1001 euid=1001 suid=1001 fsuid=1001 egid=1001 sgid=1001 tty pts1 ses=4 comm="cat" exe="/usr/bin/cat" subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c.1823 key="sshd_config_changes" ARCH=x86_64 SYSCALL=openat AUID="ritesh" UID="john" GID="john" EUID="john" FSUID="john" EGID="john" SUID="john" FSGID="john"
```

As we know that **/etc** directory kept the configuration files and hence as a Linux Administrator I want to monitor which normal user tried to open that file. For that I applied the audit rule in the file **/etc/audit/rules.d/audit.rules** and then restarted the auditd service and then checked the audit logs as shown in the screenshot attached below.

[root@Mederma-Server ~]# vim /etc/audit/rules.d/audit.rules

```
-w /etc -p warx -k etc_config_changes
```

[root@Mederma-Server ~]# systemctl restart auditd

```
[root@Mederma-Server ~]# systemctl status auditd
● auditd.service - Security Auditing Service
  Loaded: loaded (/usr/lib/systemd/system/auditd.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2025-11-07 11:59:18 IST; 14min ago
    Docs: man:auditd(8)
          https://github.com/linux-audit/audit-documentation
 Process: 4652 ExecStartPost=/sbin/augenrules --load (code=exited, status=0/SUCCESS)
 Process: 4645 ExecStart=/sbin/auditd (code=exited, status=0/SUCCESS)
 Main PID: 4647 (auditd)
   Tasks: 4 (limit: 22875)
  Memory: 1.8M
 CGroup: /system.slice/auditd.service
         └─4647 /sbin/auditd
             ├─4649 /usr/sbin/sedispatch
```

The user **john** opened the file **/etc/passwd** and found that it was reported in the audit log as shown below.

```
type=SYSCALL msg=audit(1694011500.000:1): arch=c000003e syscall=257 success=yes exit=3 a0=fffff9c a1=7f6bf2cd1e44 a2=80000 a3=0 items=1 ppid=4492 pid=4841 auid=1000 uid=1001 gid=1001 euid=1001 suid=1001 egid=1001 sgid=1001 tty pts1 ses=4 comm="cat" exe="/usr/bin/cat" subj=unconfined_u:unconfined_r:unconfined_t:s0:c0.c1023 key="etc_config_changes" ARCH=x86_64 SYSCALL=openat AUID="ritesh" UID="john" GID="john" EUID="john" SUID="john" EGID="john" FSGID="john"
type=CWD msg=audit(1694011500.000:2): cwd="/home/john"
type=PATH msg=audit(1694011500.000:3): item=0 name="/etc/ld.so.cache" inode=37150208 dev=fd:00 mode=0100644 ouid=0 ogid=0 rdev=00:00 obj=unconfined_u:object_r:ld_so_cache_t:s0:c0.c1023 key="etc_config_changes" ARCH=x86_64 SYSCALL=openat AUID="ritesh" UID="john" GID="john" EUID="john" SUID="john" EGID="john" FSGID="john"
type=PROCTITLE msg=audit(1694011500.000:4): proctitle="cat"
type=SYSCALL msg=audit(1694011500.000:5): arch=c000003e syscall=257 success=yes exit=3 a0=fffff9c a1=7ffe05420627 a2=0 a3=0 items=1 ppid=4492 pid=4841 auid=1000 uid=1001 gid=1001 euid=1001 suid=1001 egid=1001 sgid=1001 tty pts1 ses=4 comm="cat" exe="/usr/bin/cat" subj=unconfined_u:unconfined_r:unconfined_t:s0:c0.c1023 key="etc_config_changes" ARCH=x86_64 SYSCALL=openat AUID="ritesh" UID="john" GID="john" EUID="john" SUID="john" EGID="john" FSGID="john"
type=CWD msg=audit(1694011500.000:6): cwd="/home/john"
type=PATH msg=audit(1694011500.000:7): item=0 name="/etc/passwd" inode=37150355 dev=fd:00 mode=0100644 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:passwd_file_t:s0 nametype=NORMAL cap_fperm=0 cap_fmode=0 cap_fvalue=0 cap_frootid=0 OUID="root" OGID="root"
type=PROCTITLE msg=audit(1694011500.000:8): proctitle="cat"
```

Lock grub bootloader with a password

On a Linux machine it is safe to lock grub bootloader with a password, below is the procedure how do we lock the grub boot loader with a password. Login to your Linux machine and then run the command as shown below.

```
[root@Mederma-Server ~]# grub2-setpassword
Enter password:
Confirm password:
```

As shown in the above screenshot I had provided the password and confirmed with the same password. The hashed value of the password will show in the file present at the absolute path **/boot/grub2/user.cfg**.

```
[root@Mederma-Server ~]# cat /boot/grub2/user.cfg
GRUB2_PASSWORD=
```

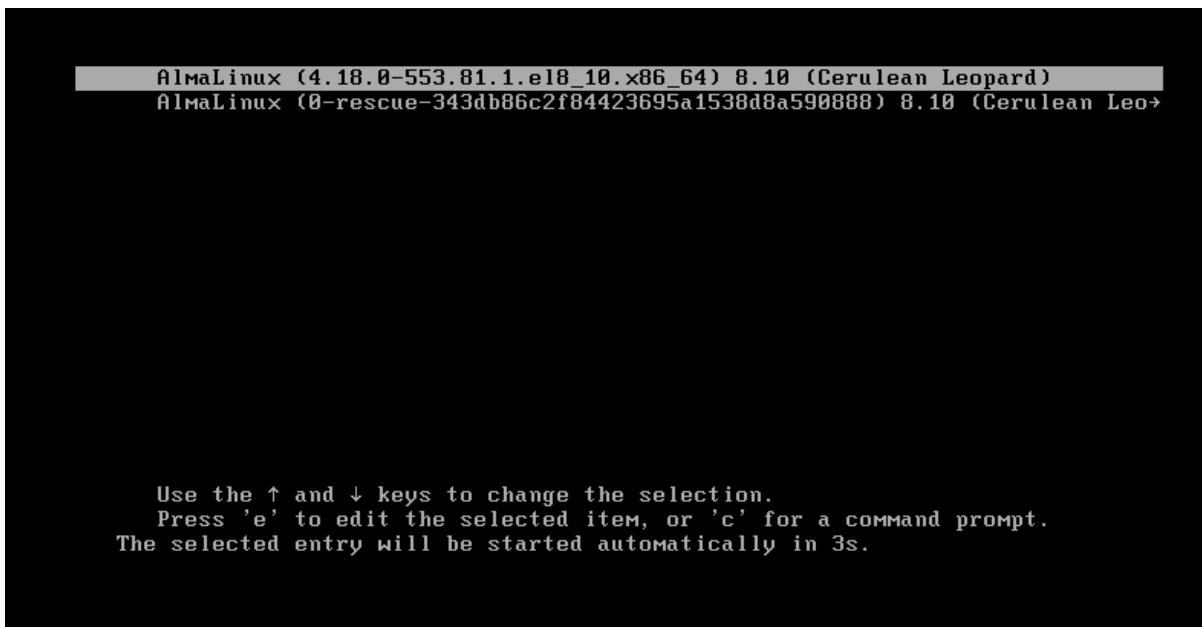
Then open the file **40_custom** present at the absolute path **/etc/grub.d/40_custom** and edit it with username and hashed password value as obtained from the previous step.

```
[root@Mederma-Server ~]# cat /etc/grub.d/40_custom
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries. Simply type the
# menu entries you want to add after this comment. Be careful not to change
# the 'exec tail' line above.
set superusers="sysuser"
password_pbkdf2 sysuser
```

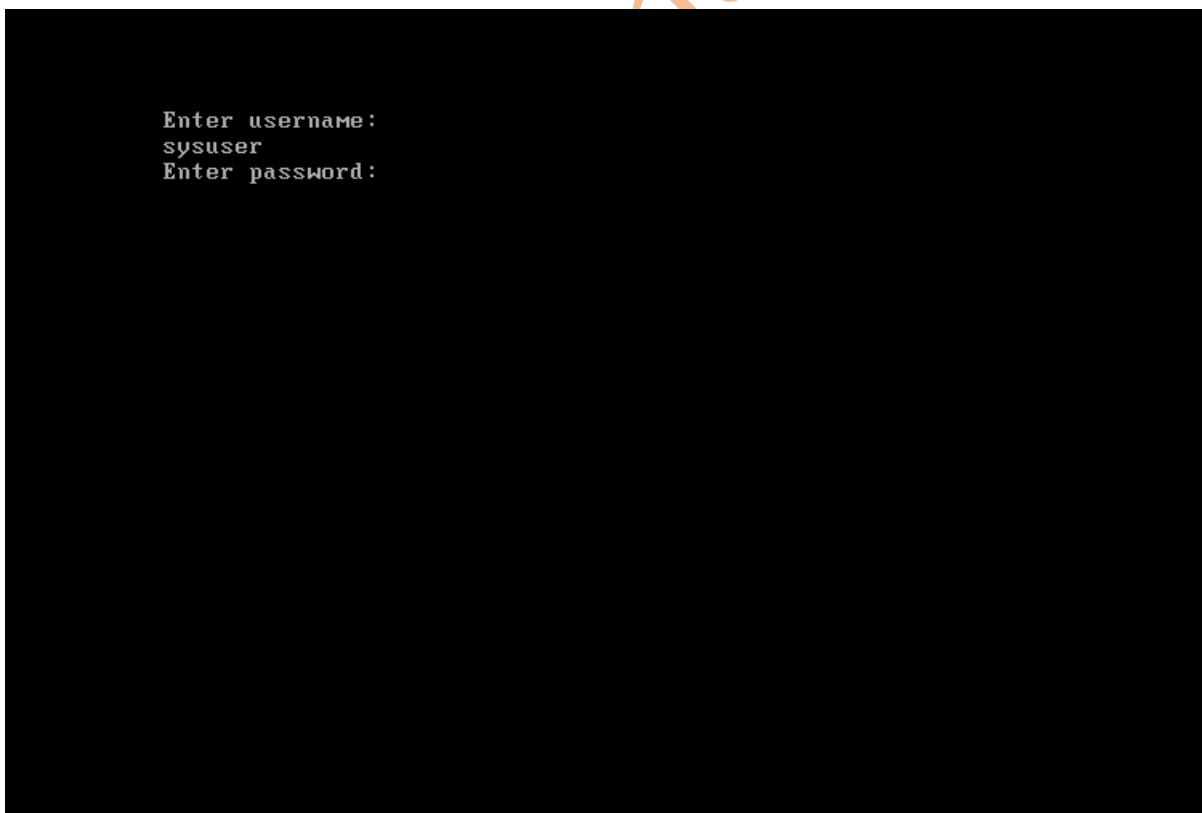
Regenerate the file **grub.cfg** as shown in the screenshot attached below.

```
[root@Mederma-Server ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
done
```

Now, reboot your Linux machine and select the key **e** to break the normal boot process as shown in the screenshot attached below.



Provide the username and password then it will redirect to edit the Grub configuration as shown in the screenshot attached below.



```

load_video
set gfx_payload=keep
insmod gzio
linux ($root)/vmlinuz-4.18.0-553.81.1.el8_10.x86_64 root=/dev/mapper/almalinux\
-root ro crashkernel=auto resume=/dev/mapper/almalinux-swap rd.lvm.lv=almalinu\
x/root rd.lvm.lv=almalinux/swap rhgb quiet
initrd ($root)/initramfs-4.18.0-553.81.1.el8_10.x86_64.img $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 completions.

For security reasons it is suggested to lock your grub bootloader.

SELinux and AppArmor

SELinux stands for **security enhanced linux** it is a security system which is built-in into Linux Kernel. It is possible to configure SELINUX temporarily (on reboot the system will ineffective) or permanently (need to configure at absolute path **/etc/selinux/config**).

In SELinux there will be three modes **1. enforcing**, **2. permissive** and **3. disabled**.

Enforcing mode is the default and most secure mode.

Permissive mode SELinux logs actions that would have been denied but does not actually deny them.

Disabled mode SELinux is completely turned off, and no security policies are loaded or enforced.

If you need to configure selinux with permissive mode then you can use the command **setenforce 0**, you can check the selinux status using the command **sestatus** as shown in the screenshot attached below.

```

[root@Mederma-Server ~]# setenforce 0
[root@Mederma-Server ~]# sestatus
SELinux status:                      enabled
SELinuxfs mount:                     /sys/fs/selinux
SELinux root directory:              /etc/selinux
Loaded policy name:                  targeted
Current mode:                        permissive
Mode from config file:              enforcing
Policy MLS status:                  enabled
Policy deny_unknown status:         allowed
Memory protection checking:         actual (secure)
Max kernel policy version:          33

```

```
[root@Mederma-Server ~]# setenforce 1
[root@Mederma-Server ~]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:              targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:    actual (secure)
Max kernel policy version:      33
```

But when you will reboot the system this temporary configuration will no longer exist. So, you need to configure it from the file **/etc/selinux/config** as shown below.

```
# 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=enforcing
# SELINUXTYPE= can take one of these three values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes are protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Then **reboot** the system and check the selinux status using the command **sestatus** as shown below.

```
[root@Mederma-Server ~]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:              targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:    actual (secure)
Max kernel policy version:      33
```

By applying appropriate selinux stamping a process will be able to access the files. As for example check the selinux stamping for file **/etc/passwd**.

```
[root@Mederma-Server ~]# ls -Z /etc/passwd
system_u:object_r:passwd_file_t:s0 /etc/passwd
```

I changed the stamping from **passwd_file_t** to **admin_home_t** as shown below.

```
[root@Mederma-Server ~]# chcon -t admin_home_t /etc/passwd
[root@Mederma-Server ~]# ls -Z /etc/passwd
system_u:object_r:admin_home_t:s0 /etc/passwd
```

Then I tried to change the password of user **john** using the command as shown below and my finds are as shown below.

```
[root@Mederma-Server ~]# passwd john
passwd: Unknown user name 'john'.
```

Now, recharge the stamping as shown below.

```
[root@Mederma-Server ~]# restorecon /etc/passwd
[root@Mederma-Server ~]# ls -Z /etc/passwd
system_u:object_r:passwd_file_t:s0 /etc/passwd
```

Here you will find that after appropriate stamping, you will be able to change the password as shown below.

```
[root@Mederma-Server ~]# passwd john
Changing password for user john.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

Here I am showing you how to change the SELinux Boolean, first I checked the SELinux status and then listed the SELinux Boolean using the command as shown below.

```
[root@Mederma-Server ~]# getenforce
Enforcing
[root@Mederma-Server ~]#
[root@Mederma-Server ~]# getsebool -a|grep "httpd_enable_homedirs"
httpd_enable_homedirs --> off
```

If SELinux if the Boolean **httpd_enable_homedirs** is **on** then Apache httpd can use the users home directory which is not good for security point of view. But for readers to demonstrate I had on it permanently as shown in the screenshot attached below.

```
[root@Mederma-Server ~]# setsebool -P httpd_enable_homedirs on
[root@Mederma-Server ~]# getsebool -a|grep "httpd_enable_homedirs"
httpd_enable_homedirs --> on
[root@Mederma-Server ~]# setsebool -P httpd_enable_homedirs off
[root@Mederma-Server ~]# getsebool -a|grep "httpd_enable_homedirs"
httpd_enable_homedirs --> off
```

In the same way you can on or off other SELinux Booleans.

AppArmor

Apparmor is mandatory access control (MAC) for Linux where access to resources is centrally managed by the system. In Apparmor a user needs to create the profiles (which is nothing but security policy defining its access rules and restrictions). Apparmor can be used with OpenSUSE or Ubuntu. For present case I am using Ubuntu 20.04 in which Apparmor was by default installed.

Types of Apparmor profile modes – **Complain mode**, **Enforcing mode** and **Unconfined mode**.

Complain mode: - In complain mode violations are blocked and Apparmor logs the attempts.

Enforcing mode: - In enforcing mode if a program performs a forbidden action, then action is denied. Violations will be logged.

Unconfined mode: - Violations are not blocked and Apparmor not logged the restrictions.

To install Apparmor in Ubuntu you can use the command **apt-get install -y apparmor**. You can check the status Apparmor using the command **systemctl status apparmor** and enable it from the boot time using the command **systemctl enable apparmor**.

If Apparmor is installed then you use the command **aa-status** to know what profiles are loaded and in which states they are in. As shown below there are 33 profiles are loaded and they are in enforcing modes.

```
root@ubuntu-2204:~# aa-status
apparmor module is loaded.
33 profiles are loaded.
33 profiles are in enforce mode.
```

To use the command **aa-genprof** you need to install the package **apparmor-utils** using the command **apt-get install -y apparmor-utils**. The command **aa-genprof** is used to create the apparmor profiles.

```
root@ubuntu-2204:~# apt-get install -y apparmor-utils
```

I had created a directory **/mederma** as shown below and I need to create the file and delete the created file in this directory using the shell script as shown below.

```
root@ubuntu-2204:~# mkdir /mederma

root@ubuntu-2204:~# mv example.sh thamu.sh
root@ubuntu-2204:~# chmod +x thamu.sh
root@ubuntu-2204:~# cat thamu.sh
#!/bin/bash

echo "This is an apparmor example."

touch /mederma/sample.txt
echo "File created"

rm /mederma/sample.txt
echo "File deleted"
```

Now, create the apparmor profile with aa-genprof then enforce the apparmor profile as shown below.

```
root@ubuntu-2204:~# aa-genprof /root/thamu.sh
```

Before you begin, you may wish to check if a profile already exists for the application you wish to confine. See the following wiki page for more information:

<https://gitlab.com/apparmor/apparmor/wikis/Profiles>

Profiling: /root/thamu.sh

Please start the application to be profiled in another window and exercise its functionality now.

Once completed, select the "Scan" option below in order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the opportunity to choose whether the access should be allowed or denied.

```
[(S)can system log for AppArmor events] / (F)inish  
Reading log entries from /var/log/audit/audit.log.  
Updating AppArmor profiles in /etc/apparmor.d.
```

Profiling: /root/thamu.sh

Please start the application to be profiled in another window and exercise its functionality now.

Once completed, select the "Scan" option below in order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the opportunity to choose whether the access should be allowed or denied.

```
[(S)can system log for AppArmor events] / (F)inish  
Reading log entries from /var/log/audit/audit.log.
```

```
Profile: /root/thamu.sh
Execute: /usr/bin/touch
Severity: 3

(I)nherit / (C)hild / (N)amed / (X) ix On / (D)eny / Abo(r)t / (F)inish
Are you specifying a transition to a local profile?

(Y)es / [(N)o]

Enter profile name to transition to: /usr/bin/touch

Should AppArmor sanitise the environment when
switching profiles?

Sanitising environment is more secure,
but some applications depend on the presence
of LD_PRELOAD or LD_LIBRARY_PATH.

[(Y)es] / (N)o

Profile: /root/thamu.sh
Execute: /usr/bin/rm
Severity: unknown

(I)nherit / (C)hild / (N)amed / (X) ix On / (D)eny / Abo(r)t / (F)inish
Are you specifying a transition to a local profile?

(Y)es / [(N)o]

Enter profile name to transition to: /usr/bin/rm

Should AppArmor sanitise the environment when
switching profiles?

Sanitising environment is more secure,
but some applications depend on the presence
```

RIT

```

of LD_PRELOAD or LD_LIBRARY_PATH.

[(Y)es] / (N)o
Complain-mode changes:

Profile: /root/thamu.sh
Path: /dev/tty
New Mode: owner rw
Severity: 9

[1 - #include <abstractions/consoles>]
2 - owner /dev/tty rw,
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / (O)wner permissions off / Abo(r)t / (F)inish
Adding #include <abstractions/consoles> to profile.

Profile: /root/thamu.sh^/usr/bin/rm
Path: /mederma/sample.txt
New Mode: owner w
Severity: unknown

[1 - owner /mederma/sample.txt w,]
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / (O)wner permissions off / Abo(r)t / (F)inish
Adding owner /mederma/sample.txt w, to profile.

Profile: /root/thamu.sh^/usr/bin/touch
Path: /mederma/sample.txt
New Mode: owner w
Severity: unknown

[1 - owner /mederma/sample.txt w,]
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / (O)wner permissions off / Abo(r)t / (F)inish
Adding owner /mederma/sample.txt w, to profile.

= Changed Local Profiles =

The following local profiles were changed. Would you like to save them?

[1 - /root/thamu.sh]
(S)ave Changes / Save Selec(t)ed Profile / [(V)iew Changes] / View Changes b/w (C)lean profiles / Abo(r)t

```

Writing updated profile for /root/thamu.sh.

Profiling: /root/thamu.sh

Please start the application to be profiled in another window and exercise its functionality now.

Once completed, select the "Scan" option below in order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the opportunity to choose whether the access should be allowed or denied.

[(S)can system log for AppArmor events] / (F)inish

Reloaded AppArmor profiles in enforce mode.

Please consider contributing your new profile! See the following wiki page for more information: <https://gitlab.com/apparmor/apparmor/wikis/Profiles>

Finished generating profile for /root/thamu.sh.
root@ubuntu-2204:~# aa-enforce /etc/apparmor.d/root.thamu.sh
Setting /etc/apparmor.d/root.thamu.sh to enforce mode.
root@ubuntu-2204:~# ./thamu.sh
This is an apparmor example.
File created
File deleted

It is possible to keep your apparmor profile in complain mode as shown below.

```
root@ubuntu-2204:~# aa-complain /etc/apparmor.d/root.thamu.sh
Setting /etc/apparmor.d/root.thamu.sh to complain mode.
```

Apparmor uses security profiles to restrict application access, here I installed nginx and restricting its access using apparmor profile as shown below.

```
root@dexter-server:~# apt-get install -y nginx
```

```
root@dexter-server:~# systemctl start nginx && systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nginx
root@dexter-server:~# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
    Active: active (running) since Thu 2025-11-13 11:33:39 IST; 19s ago
      Docs: man:nginx(7)

root@dexter-server:~# ls /etc/apparmor.d/
abi          lsb_release   ubuntu_pro_apt_news  usr.bin.tcpdump           usr.lib.libreoffice.program.xpdfimport  usr.sbin.rsyslogd
abstractions nvidia_modprobe  ubuntu_pro_esm_cache  usr.lib.libreoffice.program.oosplash  usr.lib.snapd.snap-confine.real
disable      sbin.dhclient  usr.bin.evince       usr.lib.libreoffice.program.senddoc  usr.sbin.cups-browsed
local        tunables      usr.bin.man         usr.lib.libreoffice.program.soffice.bin  usr.sbin.cupsd
```

After running the **aa-autodep nginx** command minimum profile for apparmor will be generated as shown below.

```
root@dexter-server:~# aa-autodep nginx
Writing updated profile for /usr/sbin/nginx.
root@dexter-server:~# ls /etc/apparmor.d/
abi          lsb_release   ubuntu_pro_apt_news  usr.bin.tcpdump           usr.lib.libreoffice.program.xpdfimport  usr.sbin.nginx
abstractions nvidia_modprobe  ubuntu_pro_esm_cache  usr.lib.libreoffice.program.oosplash  usr.lib.snapd.snap-confine.real
disable      sbin.dhclient  usr.bin.evince       usr.lib.libreoffice.program.senddoc  usr.sbin.cups-browsed
local        tunables      usr.bin.man         usr.lib.libreoffice.program.soffice.bin  usr.sbin.cupsd
root@dexter-server:~# cat /etc/apparmor.d/usr.sbin.nginx
# Last Modified: Thu Nov 13 11:35:31 2025
abi <abi/3.0>,

include <tunables/global>

/usr/sbin/nginx flags=(complain) {
  include <abstractions/base>
  /usr/sbin/nginx mr,
}
```

Now, I had activated complain mode so that it does not enforce policy but generates log when it detects violations of security policy. To make this effective I must restart the nginx service as shown below.

```
root@dexter-server:~# aa-complain /usr/sbin/nginx
Setting /usr/sbin/nginx to complain mode.
root@dexter-server:~# systemctl restart nginx
```

I edited the file **/etc/apparmor.d/usr.sbin.nginx** and restarted the nginx service and then ran the command **aa-logprof** to interactively review AppArmor log events and update security profiles, making the process of building or modifying profiles much more efficient.

```
root@dexter-server:~# mkdir -p /var/www/html/forbidden
```

```
root@dexter-server:~# vim /etc/apparmor.d/usr.sbin.nginx
root@dexter-server:~# apparmor_parser -r /etc/apparmor.d/usr.sbin.nginx
```

Here I restarted the nginx service using the command **systemctl restart nginx** but unable to restart then I ran the command **aa-logprof** to get insights about issues and the ability to add necessary lines in the custom profile.

```

root@dexter-server:~# aa-logprof
Updating AppArmor profiles in /etc/apparmor.d.
Reading log entries from /var/log/syslog.
Complain-mode changes:

Profile: /usr/sbin/nginx
Path: /usr/share/nginx/modules-available/mod-http-geoip2.conf
New Mode: owner r
Severity: unknown

[1 - include <abstractions/evince>]
 2 - owner /usr/share/nginx/modules-available/mod-http-geoip2.conf r,
(A)llow / [(D)eny] / (I)gnore / (Glob with (E)xtension / (N)ew / Audi(t) / (O)wner permissions off / Abo(r)t / (F)inish
Adding include <abstractions/evince> to profile.

Profile: /usr/sbin/nginx
Path: /etc/nginx/nginx.conf
New Mode: owner r
Severity: unknown

[1 - owner /etc/nginx/nginx.conf r,]
(A)llow / [(D)eny] / (I)gnore / (Glob with (E)xtension / (N)ew / Audi(t) / (O)wner permissions off / Abo(r)t / (F)inish
Adding owner /etc/nginx/nginx.conf r, to profile.
Enforce-mode changes:

Profile: /usr/lib/snapd/snap-confine
Capability: net_admin
Severity: 8

[1 - capability net_admin,]
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish
Adding capability net_admin, to profile.

Profile: /usr/lib/snapd/snap-confine
Capability: perfmon
Severity: 7

[1 - capability perfmon,]
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish

(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish
Adding capability perfmon, to profile.

Profile: /usr/sbin/cupsd
Capability: net_admin
Severity: 8

[1 - capability net_admin,]
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish
Adding capability net_admin, to profile.

= Changed Local Profiles =

The following local profiles were changed. Would you like to save them?

[1 - /usr/lib/snapd/snap-confine]
 2 - /usr/sbin/cupsd
 3 - /usr/sbin/nginx
(S)ave Changes / Save Selec(t)ed Profile / [(V)iew Changes] / View Changes b/w (C)lean profiles / Abo(r)t
Writing updated profile for /usr/lib/snapd/snap-confine.
Writing updated profile for /usr/sbin/cupsd.
Writing updated profile for /usr/sbin/nginx.

```

Then I enforced the nginx service, restarted it and checked its status as shown below.

```

root@dexter-server:~# aa-enforce /usr/sbin/nginx
Setting /usr/sbin/nginx to enforce mode.
root@dexter-server:~# systemctl restart nginx
root@dexter-server:~# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2025-11-13 11:53:40 IST; 2s ago

```

```
root@dexter-server:~# cat /etc/apparmor.d/usr.sbin.nginx
# Last Modified: Thu Nov 13 11:52:36 2025
include <tunables/global>

/usr/sbin/nginx {
    include <abstractions/apache2-common>
    include <abstractions/base>
    include <abstractions/evince>
    include <abstractions/nis>
    include <abstractions/web-data>

    capability dac_override,
    capability dac_read_search,
    capability net_bind_service,
    capability setgid,
    capability setuid,

    deny /var/www/html/forbidden/* r,

    /etc/group r,
    /etc/nginx/conf.d/ r,
    /etc/nginx/mime.types r,
    /etc/nsswitch.conf r,
    /etc/passwd r,
    /etc/ssl/openssl.cnf r,
    /run/nginx.pid rw,
    /usr/sbin/nginx mr,
    /var/log/nginx/access.log w,
    /var/log/nginx/error.log w,
    owner /etc/nginx/modules-enabled/ r,
    owner /etc/nginx/nginx.conf r,
    owner /etc/nginx/sites-available/default r,
    owner /etc/nginx/sites-enabled/ r,
    owner /usr/share/nginx/modules-available/mod-http-geoip.conf r,
    owner /usr/share/nginx/modules-available/mod-http-image-filter.conf r,
    owner /usr/share/nginx/modules-available/mod-http-xslt-filter.conf r,
    owner /usr/share/nginx/modules-available/mod-mail.conf r,
    owner /usr/share/nginx/modules-available/mod-stream-geoip.conf r,
    owner /usr/share/nginx/modules-available/mod-stream.conf r,
}

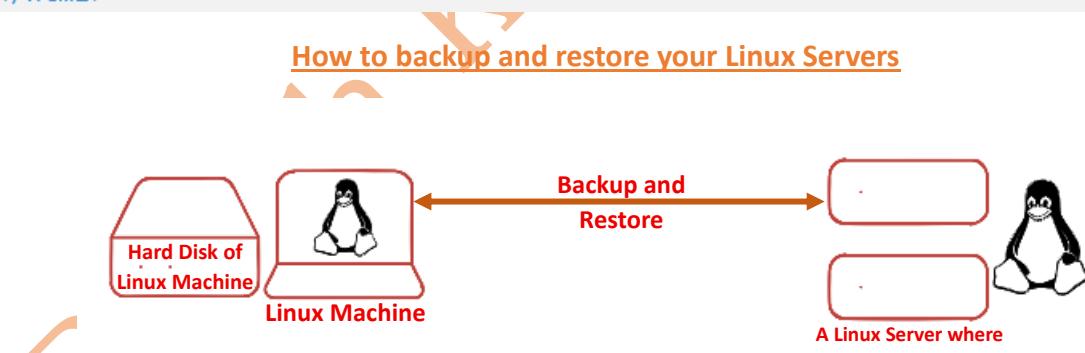
root@dexter-server:~# curl http://localhost/forbidden/index.html
<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx/1.18.0 (Ubuntu)</center>
</body>
</html>
```

Each time you are in complain or enforce mode for AppArmor and receive any errors with the service, you can use the logging utility `aa-logprof` that will give you more insights about issues and the ability to add necessary lines in your custom profile automatically.

```
root@dexter-server:~# curl http://localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
}
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```



In the current scenario I am backing up the Linux Server using rsync on local server and on remote server both. However, you can take the backup only on remote server.

```
[root@Dexter-Server ~]# cat dexter.sh
#!/bin/bash

rsync -avz --delete --exclude={"/dev/*","/proc/*","/sys/*","/tmp/*","/run/*","/mnt/*","/media/*","/backup/*","/lost+found"} / /backup/
rsync -avz --delete /backup/ root@192.168.40.135:/zack/backup/

echo "Backup complete..."
echo "Backup complete..."
echo "Backup complete..."

[root@Dexter-Server ~]# chmod +x dexter.sh ^C
[root@Dexter-Server ~]# crontab -e
crontab: installing new crontab
[root@Dexter-Server ~]# crontab -l
15 13,22 * * * sh dexter.sh
```

```
cat dexter.sh
#!/bin/bash

rsync -avz --delete --
exclude={"/dev/*","/proc/*","/sys/*","/tmp/*","/run/*","/mnt/*","/media/*","/backup/*","/lost+found"}
/ /backup/

rsync -avz --delete /backup/ root@192.168.40.135:/zack/backup/

echo "Backup complete..."
echo "Backup complete..."
echo "Backup complete..."
```

You can restore the backup from local using the command **rsync -avz /backup/ /.**

As shown in the screenshot attached above the crontab will be executed at 1:15 PM and 10:15 PM. You can restore it as per your need. You can see in the screenshot attached above with rsync I had used --delete which is used in case of incremental backup.

Perform Security Checks and ensure compliance using OSCAP

In AWS Environment

OSCAP uses **CIS benchmark** for checking and improving system security compliance. For the current project I launched four EC2 Instances and an application LoadBalancer using Terraform on one EC2 Instance I installed httpd webserver on second EC2 Instance I installed ansible which I will use as an Ansible Controller node. On rest of the two EC2 Instances (One with RHEL 9 as Operating System and another with Amazon Linux 2023 as Operating System) I will check and improve the system security compliance using OSCAP with the help of Ansible.

I had generated the ansible playbook **playbook.yaml**, using bootstrap script as shown in the screenshot attached below.

```
#####
# Install oscap #####
yum install -y openscap-scanner scap-security-guide
oscap xccdf eval --oval-results --profile cis --results-arf /tmp/arf.xml --report /tmp/report.html /usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml
oscap xccdf generate fix --fix-type ansible --result-id "" --fetch-remote-resources /tmp/arf.xml > /opt/playbook-rhel9.yml

#####
# Install OSCAP #####
yum install -y openscap-scanner scap-security-guide
oscap xccdf eval --oval-results --profile xccdf_org.ssgproject.content_profile_standard --results-arf /tmp/arf.xml --report /tmp/report.html /usr/share/xml/scap/ssg/content/ssg-al2023-ds.xml
oscap xccdf generate fix --fix-type ansible --result-id "" --fetch-remote-resources /tmp/arf.xml > /opt/playbook-almalinux2023.yml
```

Then send this generated playbook and generated report to the Ansible Controller Node and httpd webserver respectively using the command as shown below.

```
[root@yellow ~]# scp -rv /opt/playbook-rhel9.yml root@yellow:~/
[root@yellow ~]# scp -rv /opt/playbook-almalinux2023.yml root@yellow:~/
```

```
[root@... ~]# scp -rv /tmp/report.html root@...:/var/www/html/report-rhel9-old.html
```

```
[root@... ~]# scp -rv /tmp/report.html root@...:/var/www/html/report-amazonlinux2023-old.html
```

On Ansible controller the inventory and the executed command output is as shown below.

```
[root@ansible-controller ~]# cat /home/ec2-user/inventory/hosts
[rhel9]
10.10.4.131
[amazonlinux2023]
10.10.4.148

[root@ansible-controller ~]# ansible-playbook -i /home/ec2-user/inventory/hosts --limit rhel9 playbook-rhel9.yml

TASK [Record Attempts to Alter the localtime File - Check if watch rule for /etc/localtime already exists in /etc/audit/audit.rules] *****
ok: [10.10.4.131]

TASK [Record Attempts to Alter the localtime File - Add watch rule for /etc/localtime in /etc/audit/audit.rules] *****
changed: [10.10.4.131]

TASK [Gather the package facts] *****
ok: [10.10.4.131]

TASK [Configure auditd Disk Error Action on Disk Error] *****
changed: [10.10.4.131]

TASK [Gather the package facts] *****
ok: [10.10.4.131]

TASK [Configure auditd Disk Full Action when Disk Space Is Full] *****
changed: [10.10.4.131]

TASK [Gather the package facts] *****
ok: [10.10.4.131]

TASK [Configure auditd admin_space_left Action on Low Disk Space] *****
changed: [10.10.4.131]

TASK [Gather the package facts] *****
ok: [10.10.4.131]

TASK [Configure auditd max_log_file_action Upon Reaching Maximum Log Size] *****
changed: [10.10.4.131]

TASK [Gather the package facts] *****
ok: [10.10.4.131]

TASK [Configure auditd space_left Action on Low Disk Space] *****
changed: [10.10.4.131]

PLAY RECAP *****
10.10.4.131 : ok=2165 changed=346 unreachable=0 failed=0 skipped=635 rescued=0 ignored=0

[root@ansible-controller ~]# ansible-playbook -i /home/ec2-user/inventory/hosts --limit amazonlinux2023 playbook-almalinux2023.yml

TASK [Enable syslog plugin] *****
changed: [10.10.4.148]

TASK [Find /lib/ file(s) recursively] *****
ok: [10.10.4.148]

TASK [Set permissions for /lib/ file(s)] *****
changed: [10.10.4.148] => (item=/lib/python3.9/site-packages/urllib3/packages/ssl_match_hostname.py)

TASK [Find /lib64/ file(s) recursively] *****
ok: [10.10.4.148]

TASK [Set permissions for /lib64/ file(s)] *****
skipping: [10.10.4.148]

TASK [Find /usr/lib/ file(s) recursively] *****
ok: [10.10.4.148]

TASK [Set permissions for /usr/lib/ file(s)] *****
skipping: [10.10.4.148]

TASK [Find /usr/lib64/ file(s) recursively] *****
ok: [10.10.4.148]

TASK [Set permissions for /usr/lib64/ file(s)] *****
skipping: [10.10.4.148]

TASK [Check for duplicate values] *****
ok: [10.10.4.148]

TASK [Deduplicate values from /etc/ssh/sshd_config] *****
skipping: [10.10.4.148]

TASK [Insert correct line to /etc/ssh/sshd_config] *****
changed: [10.10.4.148]

PLAY RECAP *****
10.10.4.148 : ok=152 changed=29 unreachable=0 failed=0 skipped=36 rescued=0 ignored=0
```

Now, generate the report by executing the ansible playbook as shown below.

Go to Ansible Controller Node and run the playbook as shown below.

```
[root@ansible-controller ~]# cat playbook-report.yaml
---
- name: Install packages for OSCAP and generate report
  hosts: all
  become: true
  tasks:
    - name: On RHEL 9
      block:
        - name: Install packages for OSCAP
          yum:
            name:
              - openscap-scanner
              - scap-security-guide
            state: present
        - name: Create the Report
          ansible.builtin.shell: oscap xccdf eval --oval-results --profile cis --results-arf /tmp/arf.xml --report /tmp/report-rhel9.html /usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml
          args:
            chdir: /tmp
            register: oscap_scan_output
            failed_when: oscap_scan_output.rc != 0 and "fail" in oscap_scan_output.stderr | lower
          when: ansible_os_family == "RedHat" and ansible_distribution_major_version == "9"
    - name: On AmazonLinux2023
      block:
        - name: Install packages for OSCAP
          yum:
            name:
              - openscap-scanner
              - scap-security-guide
            state: present
        - name: Create the Report
          ansible.builtin.shell: oscap xccdf eval --oval-results --profile xccdf_org.ssgproject.content_profile_standard --results-arf /tmp/arf.xml --report /tmp/report-amazonlinux2023.html /usr/share/xml/scap/ssg/content/ssg-al2023-ds.xml
          args:
            chdir: /tmp
            register: oscap_scan_output
            failed_when: oscap_scan_output.rc != 0 and "fail" in oscap_scan_output.stderr | lower
          when: ansible_distribution == 'Amazon' and ansible_distribution_major_version == '2023'

[root@ansible-controller ~]# ansible-playbook -i /home/ec2-user/inventory/hosts playbook-report.yaml
PLAY [Install packages for OSCAP and generate report] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host 10.10.4.148 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [10.10.4.148]
ok: [10.10.4.131]

TASK [Install packages for OSCAP] ****
skipping: [10.10.4.148]
ok: [10.10.4.131]

TASK [Create the Report] ****
skipping: [10.10.4.148]
changed: [10.10.4.131]

TASK [Install packages for OSCAP] ****
skipping: [10.10.4.131]
ok: [10.10.4.148]

TASK [Create the Report] ****
skipping: [10.10.4.131]
changed: [10.10.4.148]

PLAY RECAP ****
10.10.4.131      : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
10.10.4.148      : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Ritesh

```

cat playbook-report.yaml

---

- name: Install packages for OSCAP and generate report
  hosts: all
  become: true
  tasks:
    - name: On RHEL 9
      block:
        - name: Install packages for OSCAP
          yum:
            name:
              - openscap-scanner
              - scap-security-guide
            state: present
        - name: Create the Report
          ansible.builtin.shell: oscap xccdf eval --oval-results --profile cis --results-arf /tmp/arfx.xml --report /tmp/report-rhel9.html
          /usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml
          args:
            chdir: /tmp
          register: oscap_scan_output
          failed_when: oscap_scan_output.rc != 0 and "fail" in oscap_scan_output.stderr | lower
          when: ansible_os_family == "RedHat" and ansible_distribution_major_version == "9"
    - name: On AmazonLinux2023
      block:
        - name: Install packages for OSCAP
          yum:
            name:
              - openscap-scanner
              - scap-security-guide
            state: present
        - name: Create the Report
          ansible.builtin.shell: oscap xccdf eval --oval-results --profile xccdf_org.ssgproject.content_profile_standard --results-arf /tmp/arfx.xml --
          report /tmp/report-amazonlinux2023.html /usr/share/xml/scap/ssg/content/ssg-al2023-ds.xml
          args:
            chdir: /tmp
          register: oscap_scan_output
          failed_when: oscap_scan_output.rc != 0 and "fail" in oscap_scan_output.stderr | lower
          when: ansible_distribution == 'Amazon' and ansible_distribution_major_version == '2023'

```

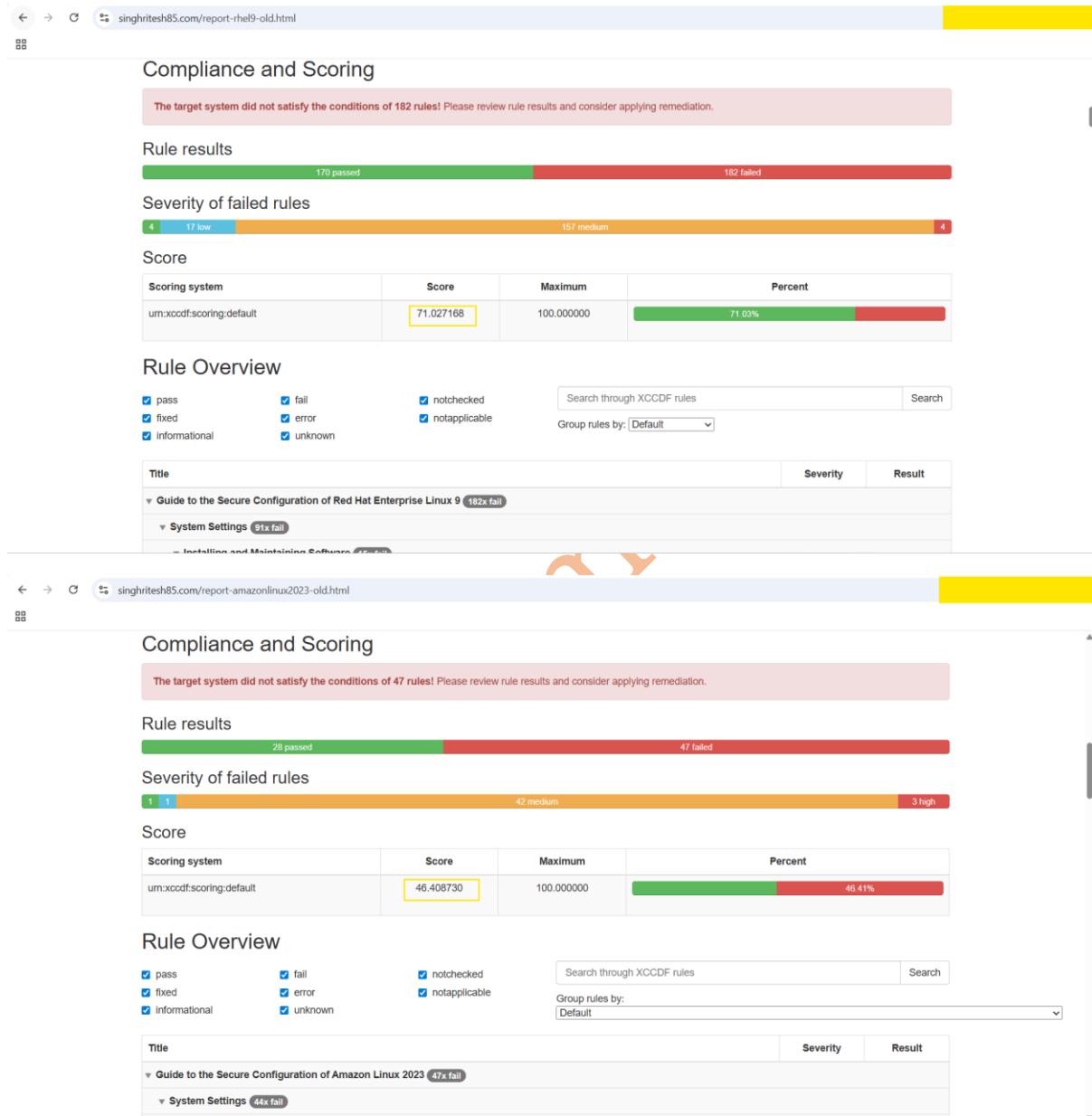
Then send the report.html to web server and open to verify as shown below.

From 1st Server (RHEL9)

```
[root@192.168.1.101 ~]# scp -rv /tmp/report-rhel9.html root@192.168.1.102:/var/www/html/
```

From 2nd Server (AmazonLinux2023)

```
[root@192.168.1.102 ~]# scp -rv /tmp/report-amazonlinux2023.html root@192.168.1.101:/var/www/html/
```



Below screenshot shows the report After running the Ansible Playbook. You can see the score had been improved to 97% and 61% for RHEL9 and AmazonLinux2023 servers respectively.

singhritesh85.com/report-rhel9.html

Compliance and Scoring

The target system did not satisfy the conditions of 12 rules! Please review rule results and consider applying remediation.

Rule results

342 passed 12 failed

Severity of failed rules

1 other 5 low 5 medium 1 high

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	97.370056	100.000000	97.37%

Rule Overview

checkboxes: pass, fail, error, unknown, notchecked, notapplicable

Search through XCCDF rules Search
Group rules by: Default

Title	Severity	Result
Guide to the Secure Configuration of Red Hat Enterprise Linux 9 12x fail		
System Settings 11x fail		

singhritesh85.com/report-amazonlinux2023.html

Compliance and Scoring

The target system did not satisfy the conditions of 35 rules! Please review rule results and consider applying remediation.

Rule results

40 passed 35 failed

Severity of failed rules

1 low 31 medium 2 high

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	61.118549	100.000000	61.12%

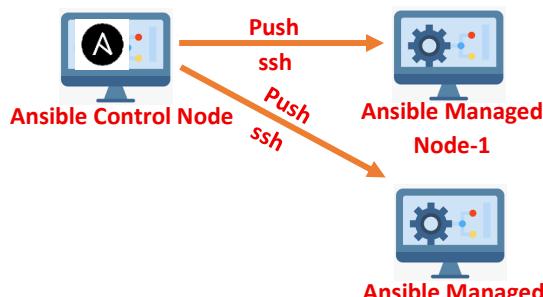
Rule Overview

checkboxes: pass, fail, error, unknown, notchecked, notapplicable

Search through XCCDF rules Search
Group rules by: Default

Title	Severity	Result
Guide to the Secure Configuration of Amazon Linux 2023 35x fail		
System Settings 33x fail		

You can create an AMI of this EC2 instance and use as the Golden AMI. As I used Ansible here so I am explaining about it in brief. Ansible is a configuration management tool which works on push mechanism (**Ansible control node sends configurations, commands, and software to the managed nodes**).



For your reference I kept all the Ansible playbooks and Terraform script in the GitHub Repo <https://github.com/singhritesh85/Linux-Administration-for-DevOps-Engineers.git>.

In Azure Environment

Here I had created four Azure VMs one for Ansible Controller, second for httpd Web Server (and Application Gateway) and two more Azure VMs (one with Almalinux8 and another with Ubuntu 22.04 Operating System which I will scan using OSCAP for configuration and vulnerability scan) using Terraform.

Either you already have the playbooks or generate it using the procedure as shown below. For this I installed OSCAP on both Almalinux8 and ubuntu 22.04 as shown below.

On Almalinux8

```
yum install -y openscap-scanner scap-security-guide

oscap xccdf eval --oval-results --profile cis --results-arf /tmp/arf.xml --report /tmp/report.html
/usr/share/xml/scap/ssg/content/ssg-almalinux8-ds.xml

oscap xccdf generate fix --fix-type ansible --result-id "" --fetch-remote-resources /tmp/arf.xml >
playbook-almalinux8.yaml
```

On Ubuntu 22.04

```
apt-get update

apt-get install libopenscap8 -y

wget https://github.com/ComplianceAsCode/content/releases/download/v0.1.78/scap-security-guide-0.1.78.zip

apt-get install -y unzip

unzip scap-security-guide-0.1.78.zip
```

Securely copied these playbooks to ansible controller node as shown below.

```
[root@oscap-vm ~]# scp -rv playbook-almalinux8.yaml ritesh@172.203.81.125:~/scap-security-guide-0.1.78/ansible/ubuntu2404-playbook-cis_level2_server.yml ritesh@172.203.81.125:~/scap-security-guide-0.1.78/ansible/ubuntu2204-playbook-cis_level2_server.yml

[root@ansible-controller ~]# cp /home/ritesh/playbook-almalinux8.yaml .

[root@ansible-controller ~]# cp /home/ritesh/ubuntu2204-playbook-cis_level2_server.yml .

[root@ansible-controller ~]# ls
playbook-almalinux8.yaml  ubuntu2204-playbook-cis_level2_server.yml
```

```
[root@ansible-controller ~]# cat /home/ritesh/inventory/hosts  
[almalinux8]  
20.████.77  
[ubuntu22_04]  
135.████.120
```

Before running the ansible playbook, I established the password-less authentication between Ansible Controller and Ansible Managed Nodes as shown below.

```
[root@ansible-controller ~]# ssh-copy-id root@20.████.77
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '20.████.77 (20.████.77)' can't be established.
ECDSA key fingerprint is SHA256:████████████████████████████████████████.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@20.████.77's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@20.████.77'"
and check to make sure that only the key(s) you wanted were added.

[root@ansible-controller ~]# ssh-copy-id root@135.████.120
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '135.████.120 (135.████.120)' can't be established.
ECDSA key fingerprint is SHA256:████████████████████████████████████████.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@135.████.120's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@135.████.120'"
and check to make sure that only the key(s) you wanted were added.

[root@ansible-controller ~]#
```

Finally, Run the Playbook as shown in the screenshot attached below.

```
[root@ansible-controller ~]# ansible-playbook -i /home/ritesh/inventory/hosts --limit almalinux8 playbook-almalinux8.yaml

TASK [Record Attempts to Alter the localtime File - Add watch rule for /etc/localtime in /etc/audit/audit.rules] ****
changed: [20.████.77]

TASK [Gather the package facts] ****
ok: [20.████.77]

TASK [Configure auditd Disk Error Action on Disk Error] ****
changed: [20.████.77]

TASK [Gather the package facts] ****
ok: [20.████.77]

TASK [Configure auditd Disk Full Action when Disk Space Is Full] ****
changed: [20.████.77]

TASK [Gather the package facts] ****
ok: [20.████.77]

TASK [Configure auditd admin_space_left Action on Low Disk Space] ****
changed: [20.████.77]

TASK [Gather the package facts] ****
ok: [20.████.77]

TASK [Configure auditd max_log_file_action Upon Reaching Maximum Log Size] ****
changed: [20.████.77]

TASK [Gather the package facts] ****
ok: [20.████.77]

TASK [Configure auditd space_left Action on Low Disk Space] ****
changed: [20.████.77]

PLAY RECAP ****
20.████.77 : ok=1940 changed=342 unreachable=0 failed=0 skipped=584 rescued=0 ignored=0
```

In Ansible playbook for ubuntu 22.04 I did below changes.

```
- name: Install cron and Network-Manager
  ansible.builtin.package:
    name:
      - cron
      - network-manager
    state: present
  when: '"linux-base" in ansible_facts.packages'
  tags:
    - CJIS-5.10.1.3
    - DISA-STIG-UBTU-22-651025
    - NIST-800-53-CM-6(a)
    - NIST-800-53-SI-7
    - NIST-800-53-SI-7(1)
    - PCI-DSS-Req-11.5
    - PCI-DSSv4-11.5.2
    - aide_periodic_cron_checking
    - low_complexity
    - low_disruption
    - medium_severity
    - no_reboot_needed
    - restrict_strategy
```

```

- name: Ensure NetworkManager is installed
  ansible.builtin.package:
    name: '{{ item }}'
    state: present
  with_items:
    - network-manager    #####NetworkManager
  when: ( not ( ansible_virtualization_type in ["docker", "lxc", "openvz", "podman",
    "container"] ) )
  tags:
    - DISA-STIG-UBTU-22-291015
    - NIST-800-171-3.1.16
    - NIST-800-53-AC-18(3)
    - NIST-800-53-AC-18(a)
    - NIST-800-53-CM-6(a)
    - NIST-800-53-CM-7(a)
    - NIST-800-53-CM-7(b)
    - NIST-800-53-MP-7
    - PCI-DSS-Req-1.3.3
    - PCI-DSSv4-1.3
    - PCI-DSSv4-1.3.3
    - low_complexity
    - medium_disruption
    - medium_severity
    - no_reboot_needed
    - unknown_strategy
    - wireless_disable_interfaces

```



```

[root@ansible-controller ~]# ansible-playbook -i /home/ritesh/inventory/hosts --limit ubuntu22_04 ubuntu2204-playbook-cis_level2_server.yml

TASK [Ensure permission u-s,g-ws,o-wt on /sbin/autrace] ****
ok: [135.***.***.128]

TASK [Test for existence /sbin/auditd] ****
ok: [135.***.***.128]

TASK [Ensure permission u-s,g-ws,o-wt on /sbin/auditd] ****
ok: [135.***.***.128]

TASK [Test for existence /sbin/audispd] ****
ok: [135.***.***.128]

TASK [Ensure permission u-s,g-ws,o-wt on /sbin/audispd] ****
skipping: [135.***.***.128]

TASK [Test for existence /sbin/augenrules] ****
ok: [135.***.***.128]

TASK [Ensure permission u-s,g-ws,o-wt on /sbin/augenrules] ****
ok: [135.***.***.128]

TASK [Test for existence /etc/audit/auditd.conf] ****
ok: [135.***.***.128]

TASK [Ensure permission u-xs,g-xws,o-xwrt on /etc/audit/auditd.conf] ****
ok: [135.***.***.128]

TASK [Find /etc/audit/rules.d/ file(s)] ****
ok: [135.***.***.128]

TASK [Set permissions for /etc/audit/rules.d/ file(s)] ****
changed: [135.***.***.128] => (item=/etc/audit/rules.d/audit.rules)

PLAY RECAP ****
135.***.***.128 : ok=2199 changed=381 unreachable=0 failed=0 skipped=629 rescued=0 ignored=0

```

Now, after scanning the Almalinux 8 and Ubuntu 22.04 Azure VMs generate the report by running the Ansible Playbook as shown in the screenshot attached below.

```

cat playbook-report.yaml

---

- name: Install packages for OSCAP and generate report
  hosts: all
  become: true
  tasks:
    - name: On Almalinux 8
      block:
        - name: Install packages for OSCAP in AlmaLinux 8
          yum:
            name:
              - openscap-scanner
              - scap-security-guide
            state: present
        - name: Create the Report in AlmaLinux 8
          ansible.builtin.shell: oscap xccdf eval --oval-results --profile cis --results-arf /tmp/arfxml --report /tmp/report-almalinux8.html
          /usr/share/xml/scap/ssg/content/ssg-almalinux8-ds.xml
          args:
            chdir: /tmp
          register: oscap_scan_output
          failed_when: oscap_scan_output.rc != 0 and "fail" in oscap_scan_output.stderr | lower
          when: ansible_distribution == "AlmaLinux" and ansible_distribution_major_version == "8"
    - name: On Ubuntu 22.04
      block:
        - name: Install packages for OSCAP in Ubuntu 22.04
          apt:
            name:
              - libopenscap8
            state: present
        - name: Create the Report
          ansible.builtin.shell: oscap xccdf eval --oval-results --profile xccdf_org.ssgproject.content_profile_cis_level2_server --results-arf
          /tmp/arfxml --report /tmp/report-ubuntu2204.html /root/scap-security-guide-0.1.78/ssg-ubuntu2204-ds.xml
          args:
            chdir: /tmp
          register: oscap_scan_output
          failed_when: oscap_scan_output.rc != 0 and "fail" in oscap_scan_output.stderr | lower
          when: ansible_os_family == 'Debian' and ansible_distribution_version == '22.04'

```

```
[root@ansible-controller ~]# cat playbook-report.yaml
---
- name: Install packages for OSCAP and generate report
  hosts: all
  become: true
  tasks:
    - name: On Almalinux 8
      block:
        - name: Install packages for OSCAP in Almalinux 8
          yum:
            name:
              - openscap-scanner
              - scap-security-guide
            state: present
        - name: Create the Report in Almalinux 8
          ansible.builtin.shell: oscapxccdf eval --oval-results --profile cis --results-arf /tmp/arfxml --report /tmp/report-almalinux8.html /usr/share/xml/scap/ssg/content/ssg-almalinux8-ds.xml
          args:
            chdir: /tmp
            register: oscap_scan_output
            failed_when: oscap_scan_output.rc != 0 and "fail" in oscap_scan_output.stderr | lower
            when: ansible_distribution == "AlmaLinux" and ansible_distribution_major_version == "8"
    - name: On Ubuntu 22.04
      block:
        - name: Install packages for OSCAP in Ubuntu 22.04
          apt:
            name:
              - libopenscap8
            state: present
        - name: Create the Report
          ansible.builtin.shell: oscapxccdf eval --oval-results --profile xccdf_org.ssgproject.content_profile_cis_level2_server --results-arf /tmp/arfxml --report /tmp/report-ubuntu2204.html /root/scap-security-guide-0.1.78/ssg-ubuntu2204-ds.xml
          args:
            chdir: /tmp
            register: oscap_scan_output
            failed_when: oscap_scan_output.rc != 0 and "fail" in oscap_scan_output.stderr | lower
            when: ansible_os_family == 'Debian' and ansible_distribution_version == '22.04'
```

[root@ansible-controller ~]# ansible-playbook -i /home/ritesh/inventory/hosts playbook-report.yaml

```
[root@ansible-controller ~]# ansible-playbook -i /home/ritesh/inventory/hosts playbook-report.yaml
PLAY [Install packages for OSCAP and generate report] ****
TASK [Gathering Facts] ****
ok: [20.***.77]
ok: [135.***.120]

TASK [Install packages for OSCAP in Almalinux 8] ****
skipping: [135.***.120]
ok: [20.***.77]

TASK [Create the Report in Almalinux 8] ****
skipping: [135.***.120]
changed: [20.***.77]

TASK [Install packages for OSCAP in Ubuntu 22.04] ****
skipping: [20.***.77]
ok: [135.***.120]

TASK [Create the Report] ****
skipping: [20.***.77]
changed: [135.***.120]

PLAY RECAP ****
135.***.120 : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
20.***.77   : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Securely copy the generated report to webserver as shown below.

```
[root@oscap-vm ~]# scp /tmp/report-almalinux8.html root@20.***.114:/var/www/html/
root@20.***.114's password:
report-almalinux8.html                                              100% 3596KB  57.7MB/s  00:00

root@oscap-vm:~# scp /tmp/report-ubuntu2204.html root@20.***.114:/var/www/html/
The authenticity of host '20.***.114 (20.***.114)' can't be established.
ED25519 key fingerprint is SHA256:[REDACTED]
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '20.***.114' (ED25519) to the list of known hosts.
root@20.***.114's password:
report-ubuntu2204.html                                              100% 4293KB  59.0MB/s  00:00
```

Finally, check the report on webserver as shown in screenshot attached below.

singhritesh85.com/report-almalinux8.html

Compliance and Scoring

The target system did not satisfy the conditions of 10 rules! Please review rule results and consider applying remediation.

Rule results

346 passed 10 failed

Severity of failed rules

1 other	5 low	3 medium	1 high
---------	-------	----------	--------

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	97.682289	100.000000	97.68%

Rule Overview

checkboxes: pass, fail, fixed, informational, notchecked, error, unknown, notapplicable

Search through XCCDF rules: Search
Group rules by: Default

Title	Severity	Result
Guide to the Secure Configuration of AlmaLinux OS 8 (10x fail)		
System Settings (6x fail)		
Installing and Maintaining Software (4x fail)		

singhritesh85.com/report-ubuntu2204.html

Compliance and Scoring

The target system did not satisfy the conditions of 38 rules! Please review rule results and consider applying remediation.

Rule results

303 passed 38 failed

Severity of failed rules

1 low	28 medium	2 high
-------	-----------	--------

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	77.351540	100.000000	77.35%

Rule Overview

checkboxes: pass, fail, fixed, informational, notchecked, error, unknown, notapplicable

Search through XCCDF rules: Search
Group rules by: Default

Title	Severity	Result
Guide to the Secure Configuration of Ubuntu 22.04 (45x fail)		

As you can see in the above attached screenshot the score is 77% which you can increase as for the current scenario the Grub2 Boot Loader password was not set, which I set by updating the playbook as shown below.

Set Boot Loader Password in grub2

high fail

```
#####
- name: Ansible Playbook for xccdf_org.ssgproject.content_profile_cis_level2_server
hosts: all
vars:
  grub_superuser: "admin"
  grub_password_hash: "grub.pbkdf2.sha512."
  var_sudo_logfile: /var/log/sudo.log
  var_sudo_timestamp_timeout: '15'
  remote_login_banner_text: "Authorized[\s\n]+uses[\s\n]+only\.[\s\n]+All[\s\n]+activity[\s\n]+may[\s\n]+be[\s\n]+monitored[\s\n]+and[\s\n]+reported\..$"
  var_password_pam_unix_remember: '5'
  var_password_pam_dcredit: '-1'
  var_password_pam_lcredit: '-1'
  var_password_pam_minclass: '4'
  var_password_pam_minlen: '14'
```

38,1 0%

```

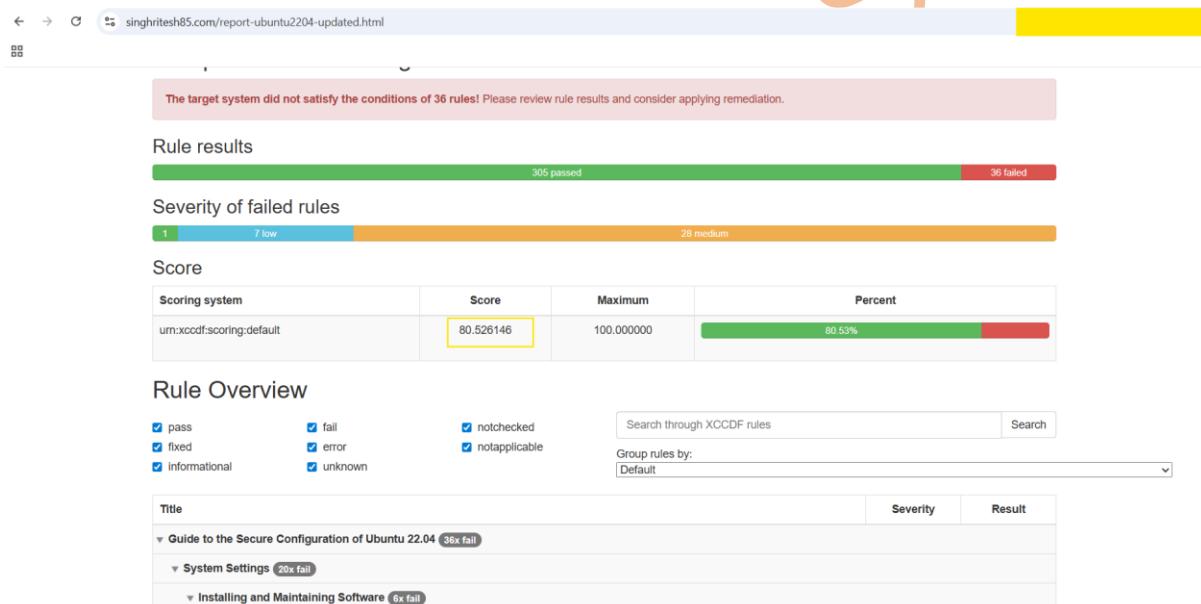
- name: Create or modify 40_custom file with GRUB password
blockinfile:
  path: /etc/grub.d/40_custom
  marker: "# {mark} ANSIBLE MANAGED BLOCK - GRUB PASSWORD"
  block: |
    set superusers="{{ grub_superuser }}"
    password_pbkdf2 {{ grub_superuser }} {{ grub_password_hash }}
    export grub_users
    export grub_password
- name: Update GRUB configuration
  command: update-grub

```

Then ran the playbook for ubuntu 22.04 as shown below followed by generation of report.

```
[root@ansible-controller ~]# ansible-playbook -i /home/ritesht/inventory/hosts --limit ubuntu22_04 ubuntu2204-playbook-cis_level2_server.yml
```

As shown in the screenshot attached below, the update report for ubuntu 22.04 and score had been increased from 77% to 80% as shown in screenshot attached below.



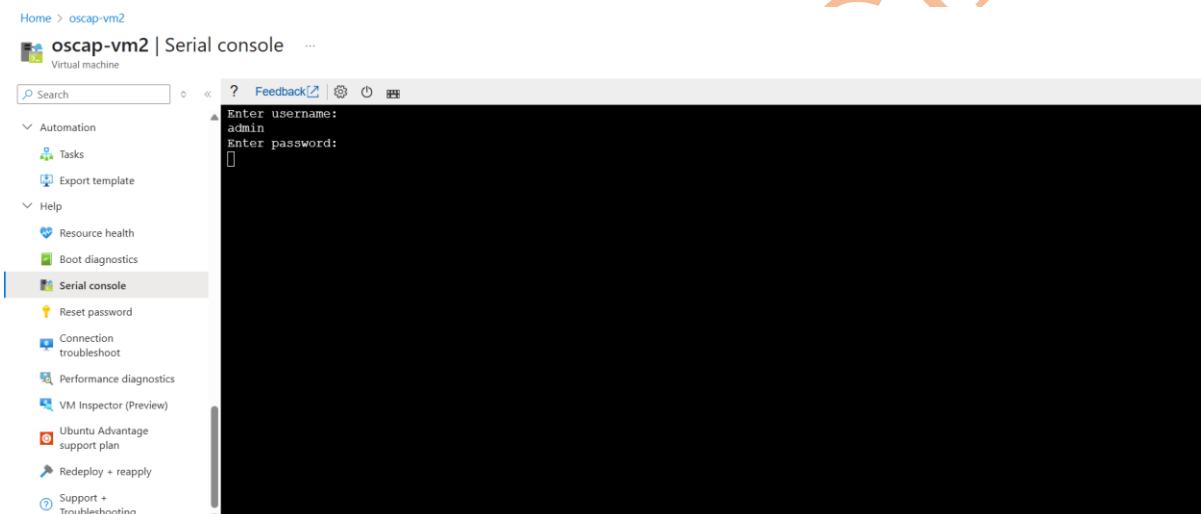
For your reference I kept the Ansible playbooks in the GitHub Repo

<https://github.com/singhritesh85/Linux-Administration-for-DevOps-Engineers.git>.

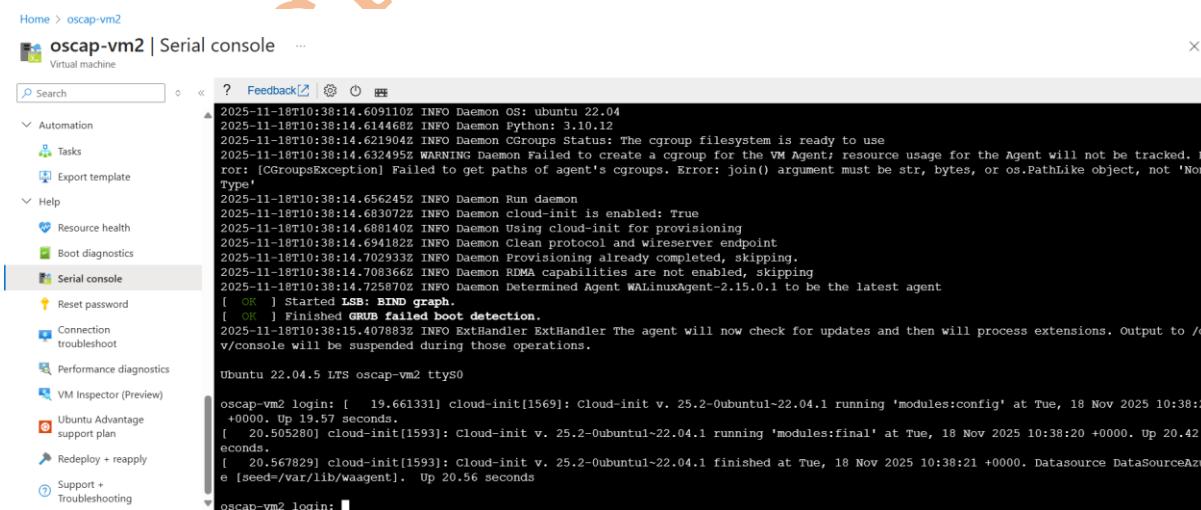
I checked the Grub2 Bootloader password (In Azure Console Go to Serial Console under the Help and restart the Azure VM) as shown in the screenshot attached below.



Provide the Grub2 Bootloader username and password as shown in screenshot attached below.



Then Azure VM will start booting.



In Google Cloud Platform (GCP) Environment

In GCP Environment I had created four GCP VM Instances and an application loadbalancer, on first VM Instance I installed httpd webserver, on second VM Instance I installed Ansible (will be treated as Ansible Controller) and two other VM Instances (one with Rocky Linux 8 and another with RHEL 9). For RHEL 9 I had already explained in AWS EC2 Instance.

Here I installed OSCAP and created the initial report and ansible playbook to scan for security and vulnerability using OSCAP using bootstrap script.

```
#####
Required configuration and Packages #####
yum install -y openscap-scanner scap-security-guide
oscapxccdf eval --oval-results --profile cis --results-arf /tmp/arfxml --report /tmp/report.html /usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml
oscapxccdf generate fix --fix-type ansible --result-id "" --fetch-remote-resources /tmp/arfxml > /opt/playbook-rhel9.yml

#####
Required configuration and Packages #####
yum install -y openscap-scanner scap-security-guide
oscapxccdf eval --oval-results --profile cis --results-arf /tmp/arfxml --report /tmp/report.html /usr/share/xml/scap/ssg/content/ssg-r18-ds.xml
oscapxccdf generate fix --fix-type ansible --result-id "" --fetch-remote-resources /tmp/arfxml > /opt/playbook-rockeylinux8.yml
```

After VM Instances came into Running state and bootstrap script executed successfully (you can check this using the log file /var/log/messages with the help of the command **tail -f /var/log/messages**) I securely copied the generated playbook and old report (without scanning for security and vulnerability) to ansible controller node and httpd webserver respectively as shown in the screenshot attached below.

```
[root@oscap-vm-instance-1 ~]# scp -rv /opt/playbook-rhel9.yml root@34.████████.201:~
[root@oscap-vm-instance-2 ~]# scp -rv /opt/playbook-rockeylinux8.yml root@34.████████.201:~
[root@oscap-vm-instance-1 ~]# scp -rv /tmp/report.html root@34.████████.28:/var/www/html/report-rhel9-old.html
[root@oscap-vm-instance-2 ~]# scp -rv /tmp/report.html root@34.████████.28:/var/www/html/report-rockeylinux8-old.html
```

Now, to scan the Rocky Linux 8 and RHEL 9 servers using OSCAP I executed the ansible playbook as shown in the screenshot attached below.

```
[root@ansible-controller ~]# cat /home/ritesh/inventory/hosts
[rhel9]
172.20.0.100
[rockeylinux8]
172.20.0.101

[root@ansible-controller ~]# ansible-playbook -i /home/ritesh/inventory/hosts --limit rhel9 playbook-rhel9.yml
```

```

TASK [Record Attempts to Alter the localtime File - Check if watch rule for /etc/localtime already exists in /etc/audit/audit.rules] ****
ok: [172.20.0.100]

TASK [Record Attempts to Alter the localtime File - Add watch rule for /etc/localtime in /etc/audit/audit.rules] ****
changed: [172.20.0.100]

TASK [Gather the package facts] ****
ok: [172.20.0.100]

TASK [Configure auditd Disk Error Action on Disk Error] ****
changed: [172.20.0.100]

TASK [Gather the package facts] ****
ok: [172.20.0.100]

TASK [Configure auditd Disk Full Action when Disk Space Is Full] ****
changed: [172.20.0.100]

TASK [Gather the package facts] ****
ok: [172.20.0.100]

TASK [Configure auditd admin_space_left Action on Low Disk Space] ****
changed: [172.20.0.100]

TASK [Gather the package facts] ****
ok: [172.20.0.100]

TASK [Configure auditd max_log_file_action Upon Reaching Maximum Log Size] ****
changed: [172.20.0.100]

TASK [Gather the package facts] ****
ok: [172.20.0.100]

TASK [Configure auditd space_left Action on Low Disk Space] ****
changed: [172.20.0.100]

PLAY RECAP ****
172.20.0.100 : ok=2121 changed=377 unreachable=0 failed=0 skipped=623 rescued=0 ignored=0

```




```

[root@ansible-controller ~]# ansible-playbook -i /home/ritesh/inventory/hosts --limit rockeylinux8 playbook-rockeylinux8.yml

TASK [Record Attempts to Alter the localtime File - Check if watch rule for /etc/localtime already exists in /etc/audit/audit.rules] ****
ok: [172.20.0.101]

TASK [Record Attempts to Alter the localtime File - Add watch rule for /etc/localtime in /etc/audit/audit.rules] ****
changed: [172.20.0.101]

TASK [Gather the package facts] ****
ok: [172.20.0.101]

TASK [Configure auditd Disk Error Action on Disk Error] ****
changed: [172.20.0.101]

TASK [Gather the package facts] ****
ok: [172.20.0.101]

TASK [Configure auditd Disk Full Action when Disk Space Is Full] ****
changed: [172.20.0.101]

TASK [Gather the package facts] ****
ok: [172.20.0.101]

TASK [Configure auditd admin_space_left Action on Low Disk Space] ****
changed: [172.20.0.101]

TASK [Gather the package facts] ****
ok: [172.20.0.101]

TASK [Configure auditd max_log_file_action Upon Reaching Maximum Log Size] ****
changed: [172.20.0.101]

TASK [Gather the package facts] ****
ok: [172.20.0.101]

TASK [Configure auditd space_left Action on Low Disk Space] ****
changed: [172.20.0.101]

PLAY RECAP ****
172.20.0.101 : ok=2076 changed=357 unreachable=0 failed=0 skipped=627 rescued=0 ignored=0

```

After execution of the playbooks for scanning the Linux machines I created the report for RHEL 9 Server and Rocky Linux 8 Server using Ansible Playbook as shown in the screenshot attached below.

```
[root@ansible-controller ~]# cat playbook-report.yaml
---
- name: Install packages for OSCAP and generate report
  hosts: all
  become: true
  tasks:
    - name: On RockyLinux 8
      block:
        - name: Install packages for OSCAP in RockyLinux 8
          yum:
            name:
              - openscap-scanner
              - scap-security-guide
            state: present
        - name: Create the Report in RockyLinux 8
          ansible.builtin.shell: oscap xccdf eval --oval-results --profile cis --results-arf /tmp/arf.xml --report /tmp/report-rockylinux8.html /usr/share/xml/ssg/content/ssg-rhel8-ds.xml
          args:
            chdir: /tmp
          register: oscap_scan_output
          failed_when: oscap_scan_output.rc != 0 and "fail" in oscap_scan_output.stderr | lower
          when: ansible_distribution == 'Rocky' and ansible_distribution_major_version == '8'
    - name: On RHEL 9
      block:
        - name: Install packages for OSCAP in RHEL 9
          yum:
            name:
              - openscap-scanner
              - scap-security-guide
            state: present
        - name: Create the Report in RHEL 9
          ansible.builtin.shell: oscap xccdf eval --oval-results --profile cis --results-arf /tmp/arf.xml --report /tmp/report-rhel9.html /usr/share/xml/ssg/content/ssg-rhel9-ds.xml
          args:
            chdir: /tmp
          register: oscap_scan_output
          failed_when: oscap_scan_output.rc != 0 and "fail" in oscap_scan_output.stderr | lower
          when: ansible_os_family == "RedHat" and ansible_distribution_major_version == "9"
```

```
[root@ansible-controller ~]# ansible-playbook -i /home/ritesh/inventory/hosts playbook-report.yaml
PLAY [Install packages for OSCAP and generate report] ****
TASK [Gathering Facts] ****
ok: [172.20.0.101]
ok: [172.20.0.100]

TASK [Install packages for OSCAP in RockyLinux 8] ****
skipping: [172.20.0.100]
ok: [172.20.0.101]

TASK [Create the Report in RockyLinux 8] ****
skipping: [172.20.0.100]
changed: [172.20.0.101]

TASK [Install packages for OSCAP in RHEL 9] ****
skipping: [172.20.0.101]
ok: [172.20.0.100]

TASK [Create the Report in RHEL 9] ****
skipping: [172.20.0.101]
changed: [172.20.0.100]

PLAY RECAP ****
172.20.0.100 : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
172.20.0.101 : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

```

cat playbook-report.yaml

---

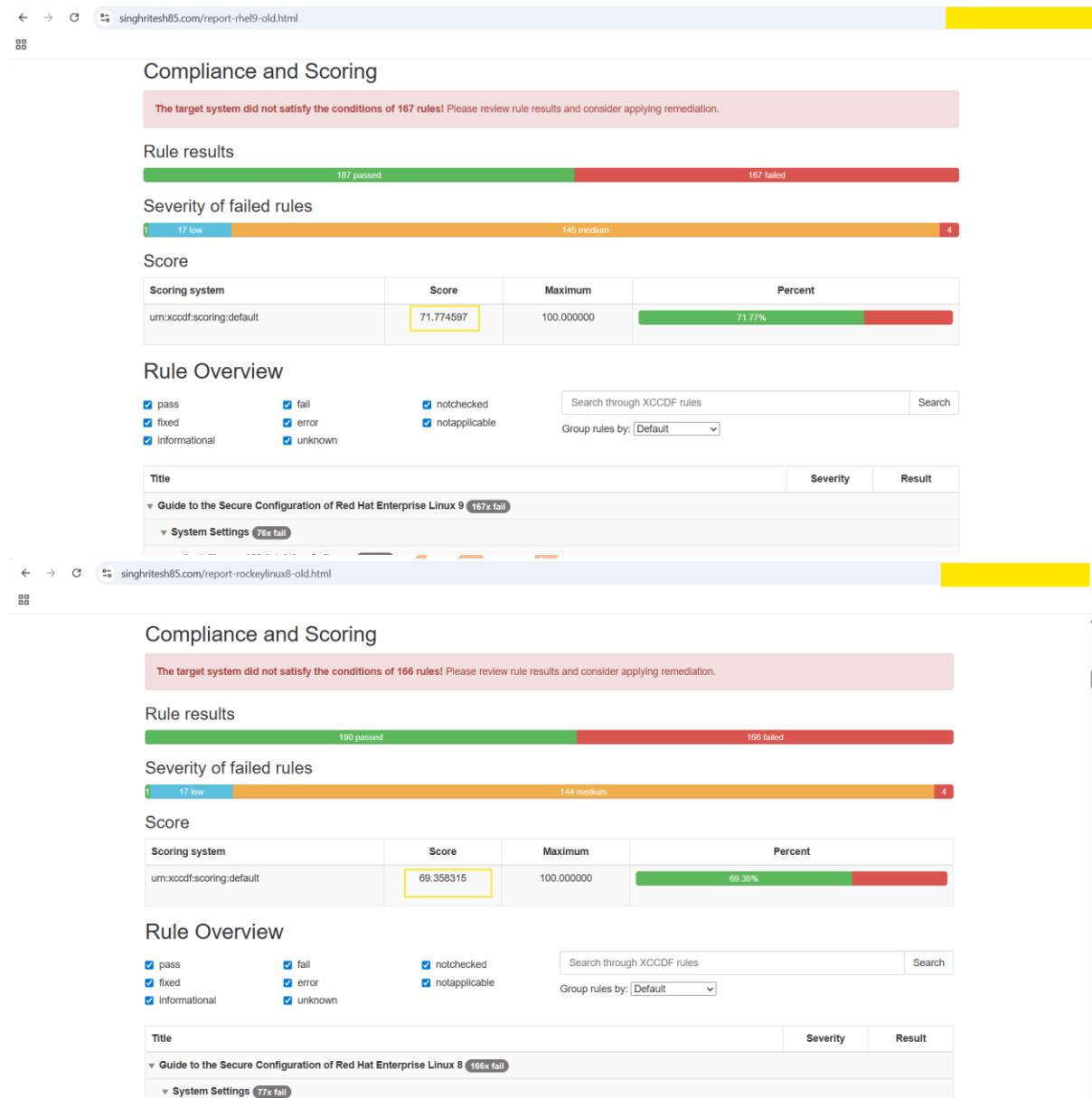
- name: Install packages for OSCAP and generate report
  hosts: all
  become: true
  tasks:
    - name: On RockyLinux 8
      block:
        - name: Install packages for OSCAP in RockyLinux 8
          yum:
            name:
              - openscap-scanner
              - scap-security-guide
            state: present
        - name: Create the Report in RockyLinux 8
          ansible.builtin.shell: oscapxccdf eval --oval-results --profile cis --results-arf /tmp/arf.xml --report /tmp/report-rockylinux8.html
          /usr/share/xml/scap/ssg/content/ssg-rl8-ds.xml
          args:
            chdir: /tmp
          register: oscap_scan_output
          failed_when: oscap_scan_output.rc != 0 and "fail" in oscap_scan_output.stderr | lower
          when: ansible_distribution == 'Rocky' and ansible_distribution_major_version == '8'
    - name: On RHEL 9
      block:
        - name: Install packages for OSCAP in RHEL 9
          yum:
            name:
              - openscap-scanner
              - scap-security-guide
            state: present
        - name: Create the Report in RHEL 9
          ansible.builtin.shell: oscapxccdf eval --oval-results --profile cis --results-arf /tmp/arf.xml --report /tmp/report-rhel9.html
          /usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml
          args:
            chdir: /tmp
          register: oscap_scan_output
          failed_when: oscap_scan_output.rc != 0 and "fail" in oscap_scan_output.stderr | lower
          when: ansible_os_family == "RedHat" and ansible_distribution_major_version == "9"

```

After generating the report, I securely copied the generated report from RHEL 9 Server and Rocky Linux 8 Server to httpd webserver as shown in the screen shot attached below.

```
[root@oscap-vm-instance-1 ~]# scp -rv /tmp/report-rhel9.html root@34.195.200.28:/var/www/html/
[root@oscap-vm-instance-2 ~]# scp -rv /tmp/report-rockeylinux8.html root@34.195.200.28:/var/www/html/
```

I found before scanning the compliance score was 71% and 69% for RHEL9 and Rocky Linux 8 Servers respectively and after scanning these scores had been improved to 97% for both RHEL9 and Rocky Linux 8 Servers as shown in the screenshot attached below.



singhritesh85.com/report-rockeylinux8.html

Compliance and Scoring

The target system did not satisfy the conditions of 11 rules! Please review rule results and consider applying remediation.

Rule results

345 passed | 11 failed

Severity of failed rules

1 other	5 low	4 medium	1 high
---------	-------	----------	--------

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	97.655090	100.000000	97.66%

Rule Overview

Filter: pass fail notchecked notapplicable fixed error unknown informational Group rules by: Default

Title	Severity	Result
Guide to the Secure Configuration of Red Hat Enterprise Linux 8 (11x fail)		
System Settings (11x fail)		

singhritesh85.com/report-rhel9.html

Compliance and Scoring

The target system did not satisfy the conditions of 13 rules! Please review rule results and consider applying remediation.

Rule results

341 passed | 13 failed

Severity of failed rules

1 other	5 low	6 medium	1 high
---------	-------	----------	--------

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	97.640251	100.000000	97.64%

Rule Overview

Filter: pass fail notchecked notapplicable fixed error unknown informational Group rules by: Default

Title	Severity	Result
Guide to the Secure Configuration of Red Hat Enterprise Linux 9 (13x fail)		
System Settings (10x fail)		

For your reference I kept these playbooks and terraform script to provision the infrastructure in GitHub Repo <https://github.com/singhritesh85/Linux-Administration-for-DevOps-Engineers.git>.

Hard Link and Soft Link in Linux

Both hard links and soft links are method to create references for files. Hard link is direct link to the data on disk. If original file is deleted then data is still accessible using the hard link. You can create the hard link using the command `ln /root/file1.txt /mederma/therema`.

```
[root@therema-server ~]# cat file1.txt
Hello Dexter
[root@therema-server ~]# ln /root/file1.txt /mederma/therema
[root@therema-server ~]# cat /mederma/therema
Hello Dexter
[root@therema-server ~]# rm -f file1.txt
[root@therema-server ~]# cat /mederma/therema
Hello Dexter
```

Soft link is pointer to another file. If original file is deleted then soft link will be broken. You can create the soft link using the command `ln -s /root/file2.txt /therema/bingo`.

```
[root@therema-server ~]# ln -s /root/file2.txt /therema/bingo
[root@therema-server ~]# cat /root/file2.txt
Mederma
[root@therema-server ~]# ls -l /therema/bingo
lrwxrwxrwx. 1 root root 15 Nov 12 19:51 /therema/bingo -> /root/file2.txt
[root@therema-server ~]# cat /therema/bingo
Mederma
[root@therema-server ~]# rm -f file2.txt
[root@therema-server ~]# cat /therema/bingo
cat: /therema/bingo: No such file or directory
```

Ritesh

[GitHub Repo: - https://github.com/singhrites85/Linux-Administration-for-DevOps-Engineers.git](https://github.com/singhrites85/Linux-Administration-for-DevOps-Engineers.git)