RHEL Intensive

Created by: Shujah Ullah
Course: RHEL (DCCS)
Tutor: Kazim Sheikh

Table of Contents

Preface

This document is provided with the understanding that it may contain errors or omissions. While every effort has been made to ensure accuracy, the author accepts no liability for any inaccuracies. The content is provided for informational purposes and is made available on an "as-is" basis. Users are encouraged to independently verify any information and use it at their discretion. It is free to use, share, and distribute.

Connect with me on Linkdein-ShujahUllah

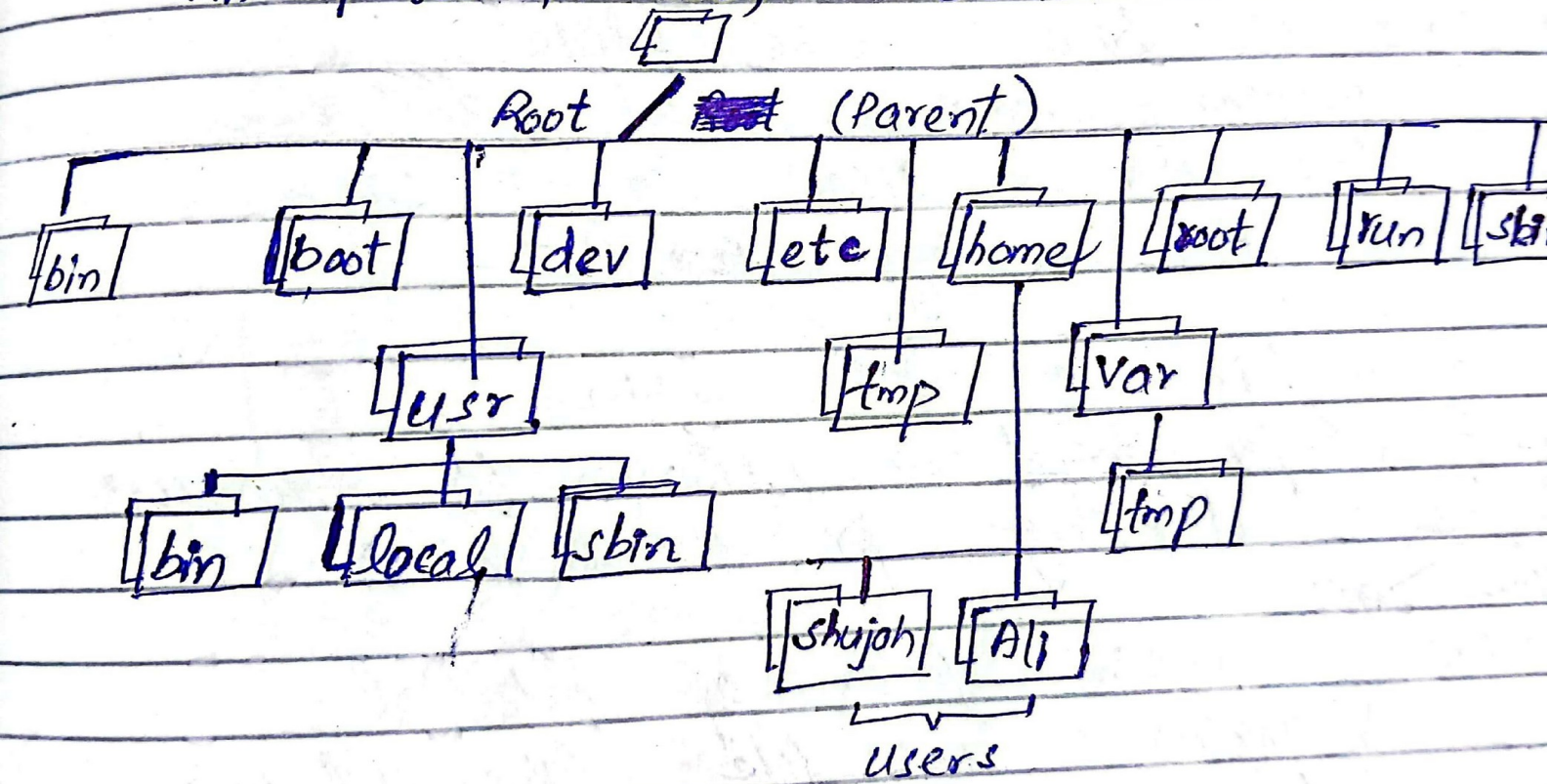
001. Linux Fundamentals

Linux File System:

RHEL Intensive

Lec #1 File system Hierarchy

```
touch test{1..100}    # Creates files test1 ... test1
rm -f test{1..100}    # Deletes  "  "  "  "
```



Absolute path : cd /boot/grub2

Relative path : cd boot
cd grub2

virtual termina

most imp of all directories

/boot : imp kernel files
/dev : device drivers (tty)
/etc : configuration files of apps
/bin : program files of commands ie ls, pwd
binaries
/sbin : super bin i.e. fdisk
contains commands that can't run by Normal user but can only run by Root -

bin
binaries for Normal usr

sbin
binaries for Super use

/home : directory for Normal users
shujah Ali
/usr : contains files of installed programs (.so file like dll in windows
/var : log files (like event viewer in windo

Using VI Editor:

vi Editor :

vi filename.txt

- → press 'i' to Enter Insert Mode.
- → " 'Esc' " Exit " " .
- → :w → save
- → :wq → save + Exit
- → :wq! → Forceful save + Exit
- → :q → Quit without save

works in 'esc' mode

- → "yy" → copy line & "p" → paste
- → 'dw' → Delete word
- → 'x' → delete single letter
- → 'u' → Undo
- → ":%.s/word/replacewith/g" → search & replace
- → :/keyword/ → search keyword in File
- → ":X" → protect file with password

using Vim Editor

* Vim is more powerful than vi

vim test.txt

↓

Esc → :X → encryption key → :wq

This will encrypt all data in the file data can't be read with cat after encryption.

* configuration files changed using vi or vim

Softlinks vs Hard links:

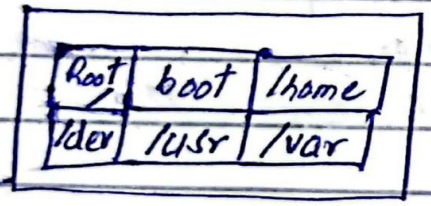
③

Soft links & hard links

partition — logical division of physical storage (Hard disk) into isolated sections

Defined at Disk level

Hard disk



V/S

directory /Folder

located with in partitions is a container & is part of hierarchial file sy

Partition

/home

All → Directory

Each partition is formated with a specific file System

File system	Partitions
tmpfs	/run
/dev/sda1	/boot
/dev/sda2	/usr

*) To check whether is partition or Not

```
df -h
```

it shows all partitions inside Disk

inside that directory -

File usually has '0' link count '1', if any file has link count more than one then it is hard link

Soft link

ln -s <filename> <link file>

example

ln -s /opt/file1 /home/linkfile

⇒ can be created across partitions

⇒ if parent file is deleted child becomes orphan & link is broken

⇒ Soft links are mostly used as compared with HL

⇒ Also called Symlink

⇒ child don't retain data after parent file got deleted

Hard link

ln <filename> <link file>

ln /opt/file1 /opt/filelink

⇒ Hard links only created inside same partition

⇒ child file retains data even if parent is deleted.

⇒ is a kind of backup but Not Recommended due to link in the same partition

⇒ For checking whether there exist Hard link use "ls -l" & check link number if that no is more than 1 it means there exist hard link of that file

-rw-r--r-- 2 root file1

this no (link count)

Inode :- Every file in Linux is indexed with a unique number called its inode no -

Hard Link : Each hard linked file has same NO

Soft Link : Inode NO,s of soft link file are diff

Parent → child

File sys → Super block → Inode Table
→ for inode NO

ls (-i) filename
>> 721602 filename

⇒ To create Hidden file
touch .hidden

⇒ For showing hidden file
ls -a

⇒ For checking inode NO of hidda file
ls -ia

Creating new group & Adding users;

# sudo groupadd group1
sudo groupadd group2

# sudo usermod -aG group1 (Ali) user
sudo usermod -aG group2 usama

groups Ali
» Ali group1

To change/switch the user

su - username

i386 / package
i686.rpm 64bit package
noarch.rpm for both 32 bit & 64 bit Architec

④ Ownership & permissions-

useradd 'username'
groupadd 'group name'

ls -l testfile.txt

-rwx rwx --- shujah Eaglegroup 25 time testfile
u g o user group size bytes

- file
'd' directory
's' socket
'p'

changing owner/user of file
chown <username> <file name>

changing permissions

chmod u+rwx
chmod ugo+rwx
chmod ugo-rwx
user / group / others

chmod 777 file.txt
-rwx rwx rwx

chmod 707 file
-rwx --- rwx

chmod 605
- rw- --- r-x
110 000 101

chmod -Rv 777 directoryname
↓
to change perm of all files under directory

Note: In case of inherit permissions
Directory 0 → rwx still others can

Users, Ownerships and permissions:

Run Levels:

Advance permisssions (ACL):

(5) **Run Levels:**

0: Halt / shutdown.
1: Single user Mode.
2: Multi user Mode.
3: multi user mode with Networking
4: Undefined or custom
5: GUI + multi user + Networking
6: Reboot

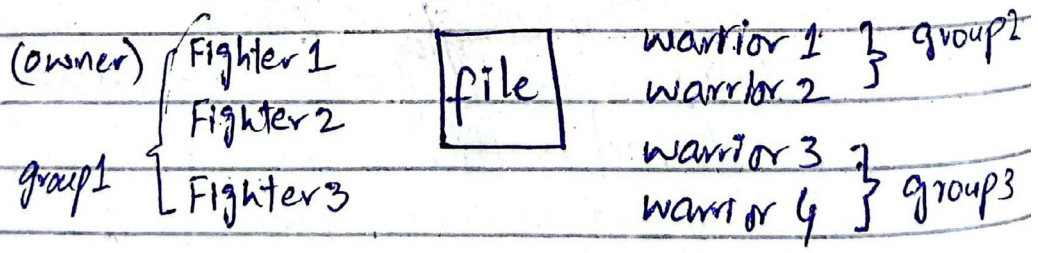
To change run level: runlevel 0,1,2-6

*) chattr +i file.txt
↳ even root can't change or access the file

Advance permissions

Access control list (ACL) :-

ACL v/s chmod the difference?



For "file" give following permissions

Fighter 1 → rwx
Fighter 2 → r-x
Fighter 3 → --x
warrior 1 → rwx
warrior 2 → ---
warrior 3 → -w-
warrior 4 → r--

since Fighter 2, 3, warrior 1, 2, 3, 4 are fall in others so, using 'chmod' its not possible to set individual permi-

using "Access Control List (ACL)"

setfacl: To set individual permissions for Multiple users

```
setfacl -m user:fighter1:rwx file.txt
setfacl -m user:fighter2:rx  file.txt
setfacl -m user:fighter3:x   file.txt
setfacl -m user:warrior1:rwx file.txt
setfacl -m user:warrior2:    file.txt
setfacl -m user:warrior3:w   file.txt
setfacl -m user:warrior4:r   file.txt
```

getfacl : shows ACL for a File
getfacl myfile.txt

Run level:

init [0-6]

Q: How to recover, if you forget Root passwd?
Ans: switch to single user Mode (RL=1)
in Single user Mode sys don't ask for password.

Recovering Root passwd:
In VM Reset Machine, when grub splash window open then press ↑ key & then 'e' to Modify kernel parameters.
set following parameters.
rd.break enforcing=0 rdgb quiet
Press Ctrl+X → mount -o remount, rw /sysroot
chroot /sysroot → passwd root → touch /.autorelabel

Run level 3 (Console Mode) -

init 3

In console Mode (run level 3) user have Max 6 terminals -

To switch Terminal

alt + [F1 - F6]

user can work on 6 terminals maxim

Gnome	v/s	VCS
↓		↓
Pts (driver)		tty (driver)
/dev/pts		/dev/pts

→ In Run level 3 : virtual console VCS is Default terminal, tty is driver that loads VCS -

→ In Run level 5 & 6 : gnome is the default terminal, "pts" is driver which loads gnome -

Total 400 terminals can be opened at a time in gnome -

→ VCS is default terminal for Run level 3 & below -

→ ~~To~~ can we change VCS to gnome in Run level 3?

Yes

start x → Brings gnome "GUI" in RL 3

pkill X → switch back to VCS

→ If Runlevel changes, all commands & processes running will be stopped (swiped), but files will be there in other RL too.

→ putty uses "tty".

---

important note:-

In CentOS 7 & older versions uses Sys-V init system for managing system initialization which include Run levels & inittab file.

"Run levels are obsolete & init Tab file is no longer used"

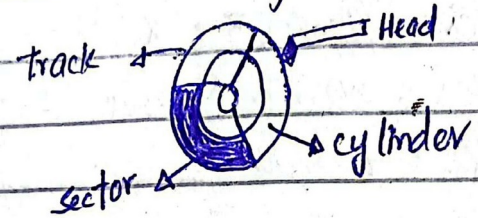
Newer versions have transitioned to "systemd" as default initialization system.

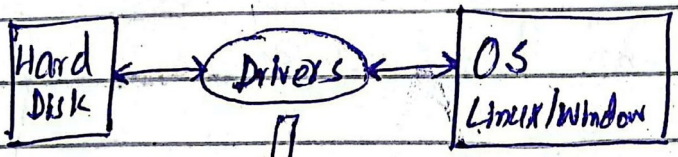
systemd	vs	Run level
multi-user.target		3
graphical.target		5
unit files & targets		init tab

(6) Storage & Partitions :-

Harddisks
SATA, PATA, SSD, SAS, SAN, Pendrive, NAS, HDD.

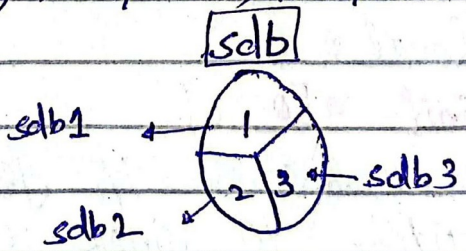
Construction of Hard Disk





- low level programs
- /dev/sda , /dev/sdb , /dev/sdc

Partitioning



- Driver files created Automatically when we create partition in Harddisk
- Drivers exist in /dev Folder.
- For Hard disk with 4 partitions total 5 drivers exist sdb, sdb1, sdb2, sdb3, sdb
- 'Flash' means writing data to HD.

Partitioning Tools in Linux
2TB > fdisk, cfdisk, sfdisk parted

Storage and Mounting Partitions (fdisk):

Mounting New Partition with fdisk

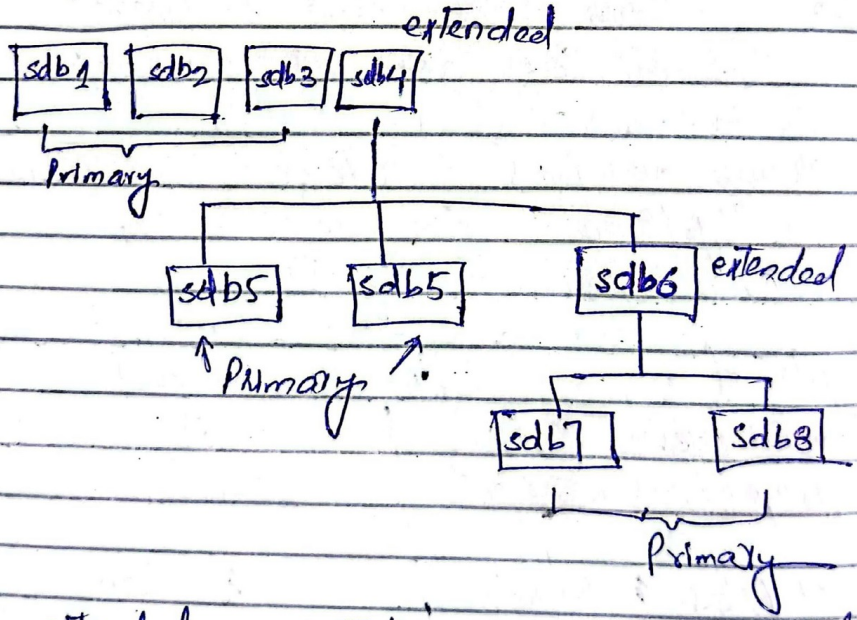
→ Create new Virtual Harddisk
→ Discover your Harddisk in Linux.
→ ls /dev/sd*
  » /sda /dev/sdb /dev/sdc (newly connected disk)
→ If harddisk is connected to online system then we have to Discover it Manually.
  echo "- - -" > /sys/class/scsi_host/host0/scan
  ↳ Cmd to Discover HD connected to online sys
→ fdisk -l /dev/sdc
Creating partition
→ fdisk /dev/sdb
  ↓ press 'n' <new partition>
  ↓ primary
  ↓ set partition Number
  ↓ size +700M
Now enter 'p' to display partition
" " 'w' to save & update to the partition table -
→ update kernel table -
  partx -a /dev/sdc
Creating File system
→ mkfs.ext4 /dev/sdc

Attach to Directory -
→ mkdir /tomcat
  mount <device driver> <directory name>
  ↳ this mount is temprory, after reboot we have to mount. Again
"fstab File" update for permanent Mount
→ At last confirm if Mounted to FS

fdisk /dev/sda

- "To create, manage, change partitions but don't Format the partition" -

⇒ By default 4 partitions are allowed as primary - If we want more partitions than 4, then we have to add one partition as extended partition (Not primary) & then add more primary partitions under that extended



extended partitions never mounted nor used, they are used as name for other extended primary partitions

If an extended partition deleted then all partitions under it

Device Driver or UUID	Mount Point	fs Type	Permissions	Dump	fsck check
UUID /dev/sda1	/media/drive	ext4	defaults	0	1

entries in /etc/fstab file

Mounting storage :

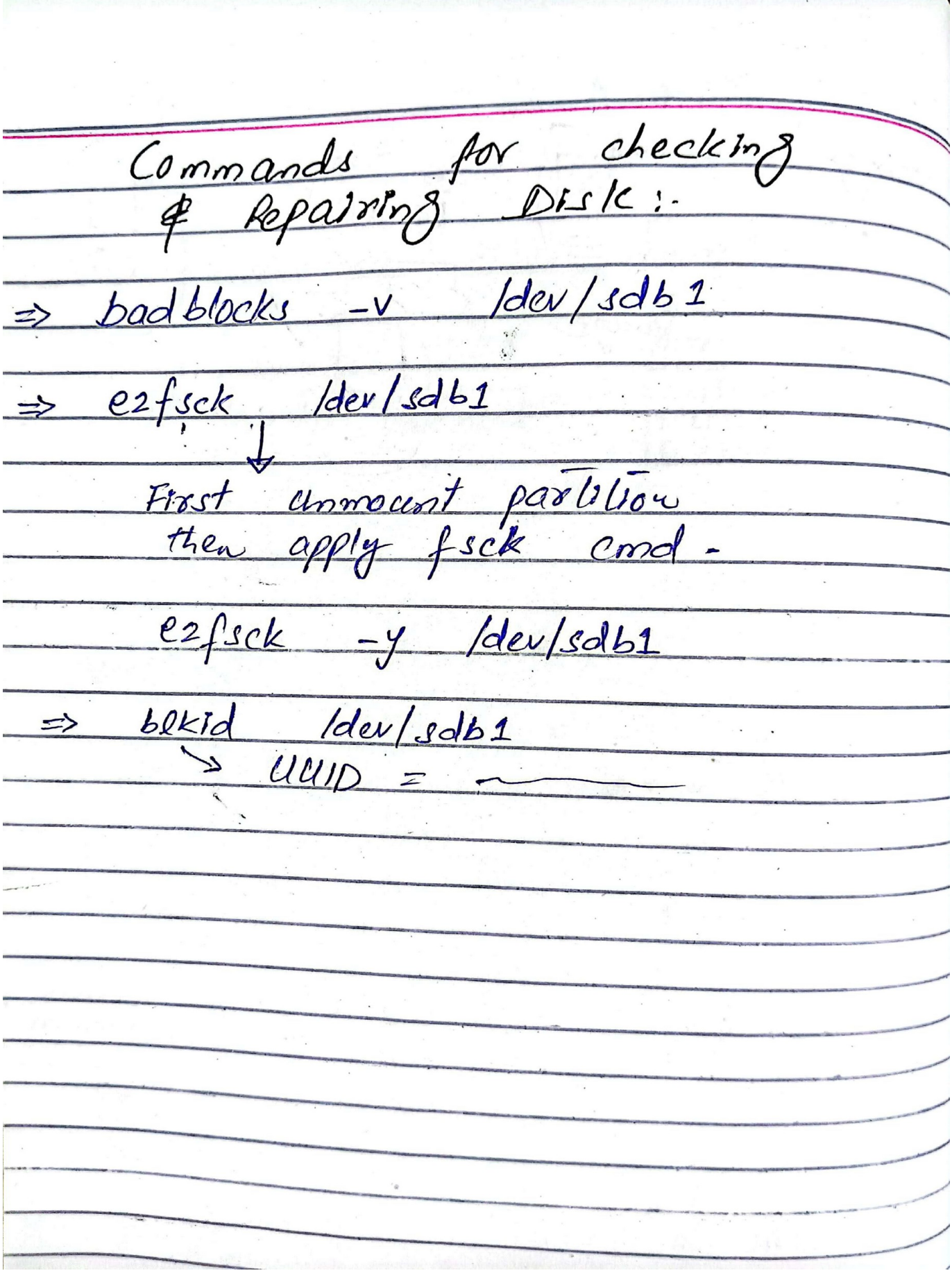
1. lsblk ⟶ Discover Media
2. fdisk /dev/sd* ⟶ Make partitions
3. partx -a /dev/sd* ⟶ update kernel about partition Table
4. mkfs.ext4 /dev/sd* → Format partition with file system
5. mount /dev/sda1 /home/newdir
   ↳ Mount partition over a directory (this will attach driver file of partition to directry)
6. sudo nano /etc/fstab
   ↳ ~~change~~ update fstab file for Persistent Mount, otherwise Mount will Removed after shutdown.

standard for partitioning

1 - 3 Primary
4 ⟶ extended
5 - 7 ⟶ primary /logical
8 ⟶ ~~ext~~ extended

Partition table → msdos/mbr
→ GPT

In GPT we can create 128 primary partitions.



Checking and Repairing Disks

Fast
Low Compression

slow
High Compression

File compression / Decompression

gzip <file name> /compression
bzip2 <file names> /compression
↳ suitable for large files

gunzip <file.gz> /decompression

Directory compression / decompression

① zip <dir.zip *> // zip all files // in current Directory

② zip -r <archive_name.zip> <dir_to zip>

above will some times not work
tar is Best for Archiving Directories
tar don't Compress it only Archives -

③ tar -czf <dir.tar.gz> <dir-tobe-Compress>

* bzcat/zcat abc.txt.gz (Read data from uncompressed File)

Archive + Compression -

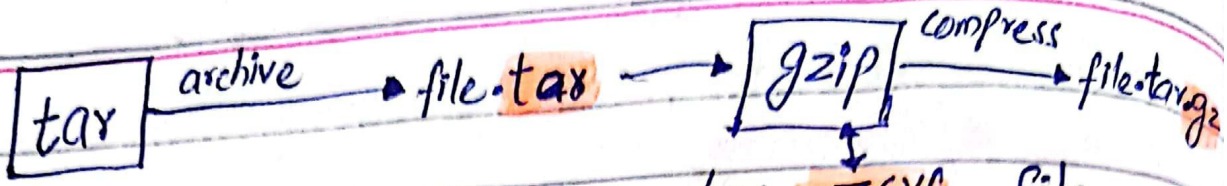
④ zip mydir.zip mydir/*
compress & Archives all files
from "mydir" to "mydir.zip" file

zip <destination Address> <source Addr/*>

unzip <filename.zip>
unzip -d <dir location> <filename.zip>

zip -r archive.zip .
#zip files in current Dir & sub direct
in archive.zip

File Compression (tar, gzip):



tar -cvf file

tar -zcvf file

tar	v/s	gzip
→ Archives only		→ Archives + compress
→ Linux Builtin Tool		→ 3rd party for Linux, D.O.S, Windows, MacOS, Risc

tar -cvf <file.tar> <file to be Archive>

tar -xvf <file.tar> // Un Archive

"tar can be used with gzip, to Archive & Compress"

tar -zcvf <file.tar.gz> <file to be Compr

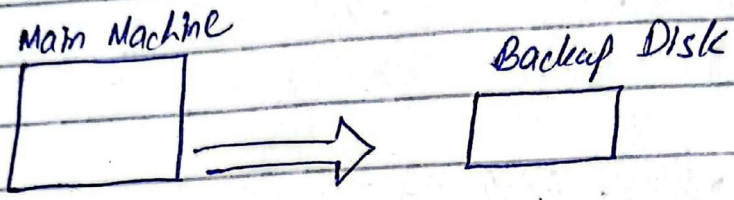
tar -xzvf <file.tar.gz> // Un Compress

- -c = ~~Compress~~ Archive
- -x = expand - Un compress
- -v = verbose
- -f = file
- -z = gzip compress

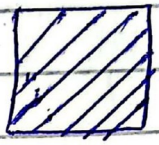
(Remote sync)

Rsync ⇒ For Incremental Backup

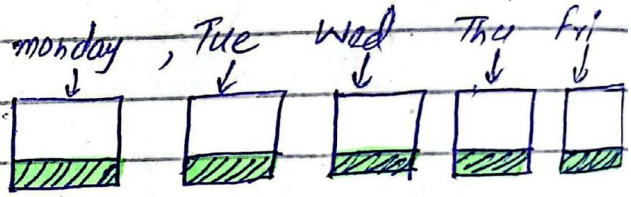
Backup Types
- → Full Backup
- → Incremental Backup
- → Differential Backup



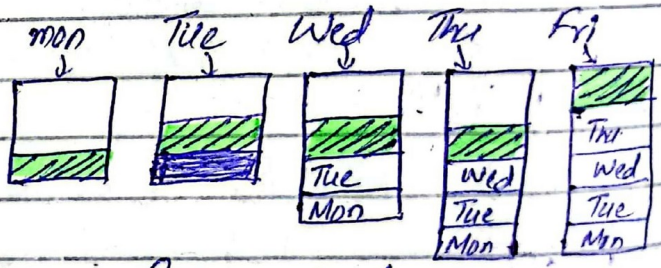
Full Backup



Incremental Backup



Differential Backup



→ Backup all data Since last Full Backup

Backup Devices / Storage Mediums
- → Tape Drives (Magnetic Tapes)
- → Hard Disk clonners (HDD, SSD) Docking Stations
- → SAN / NAS
- → Cloud

Rsync

rsync [options] source destination

-a --archive
-r --recursive
-z --compress
-v --verbose
-h --human readable
--delete delete files on destination that dont exist on the source.

remote connections

rsync [options] source_user@host: source path destination

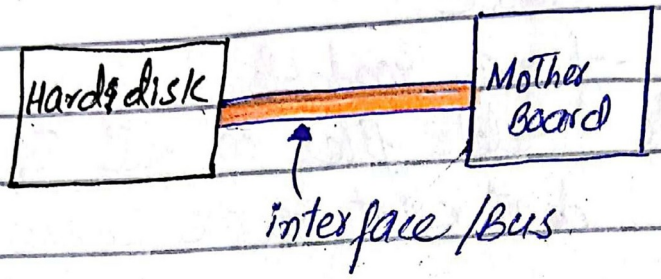
rsync -aAXvr --delete <sourcedir/> <backupdir/>

-a archive Mode
-A Preserve ACLs (permissions)
-X Preserve Attributes
-v verbose
-r recursive (directory inside dir)

STORAGE Technologies Overview -

Hard disks

- HDD
- NVMe
- SSD

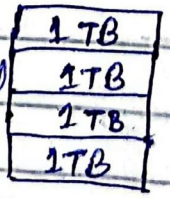


Hard Drive Interface Technologies -

1. SATA (serial ATA) ⇒ old IDE
2. PCIe used with high performance SSD & NVMe
3. SCSI
   - Parallel (obsolete / older)
   - Serial (SAS)

SAS is used with scalable storage systems such as RAID, SAN, NAS.

1TB "SAS" Hard drives connected together to form 4TB RAID storage -



4TB RAID

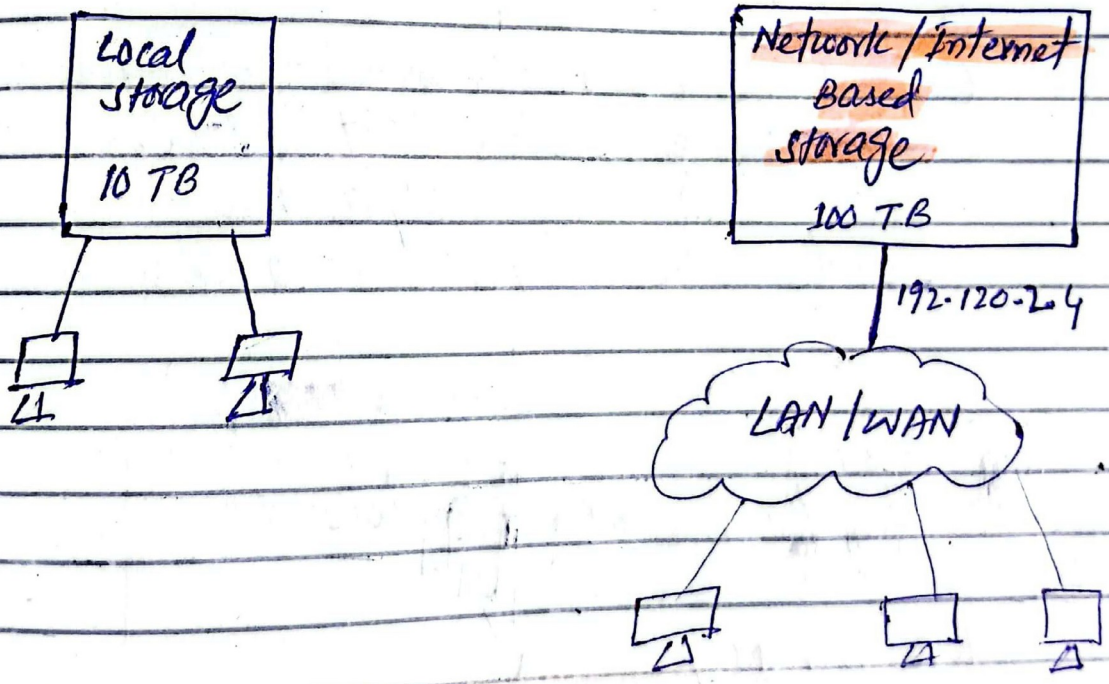
Storage Technologies overview:

SATA	PCIe/NVMe	SAS/SCSI
→ consumer & General purpose	High speed & performance systems	enterprise level storage systems

SCSI

- → SAS (Serial Attached SCSI)
- → iSCSI (internet SCSI)

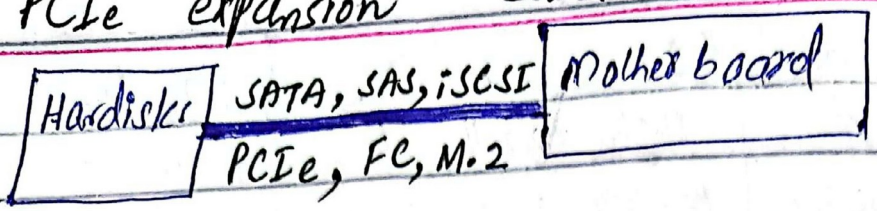
SAS vs iSCSI



(4) Fiber channel FC (very fast & most expensive)

(5) M.2 U.2 (used with laptops b/c is miniature)

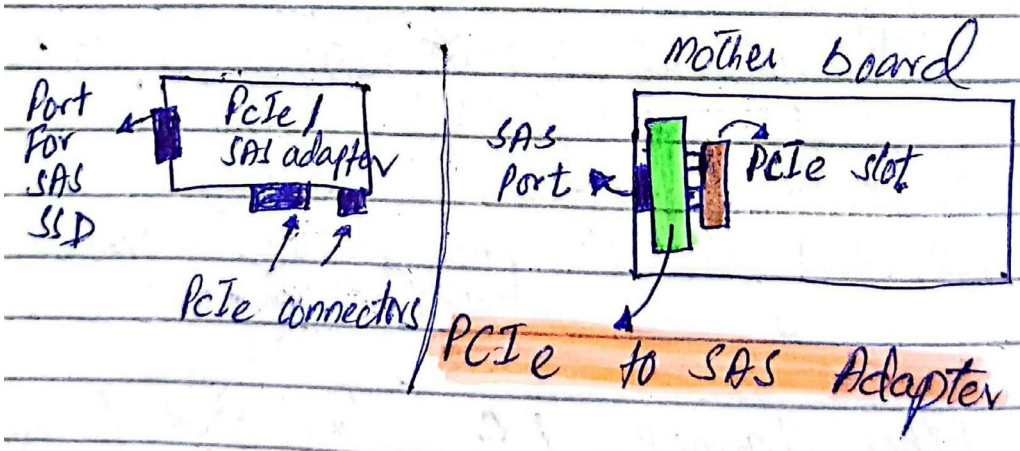
PCIe expansion Cards (HBA cards).

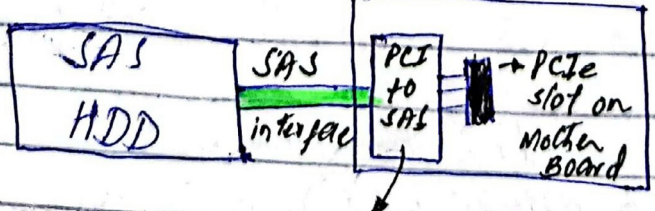


Now that we have known there are various interfaces / technologies available for connecting a Hard drive to Motha board, suppose we have Hard Drive that is supported by SAS interface we have two options

① If our Mother board has dedicated SAS slot we can directly connect our SAS drive to it.

② If Mother board don't have dedicated SAS slot, then we have to use expansion card i.e. PCIe to SAS





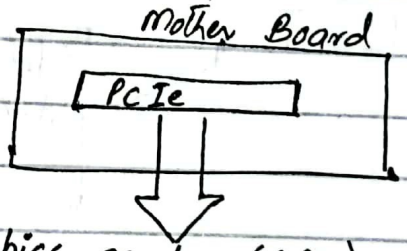
Adaptor or PCIe expansion card

Every Mother board has a Native interface available which is "PCIe" (Peripheral Component Internet Express) its high speed interface. that's why devices connected directly to it (NVMe SSD,s) provide much higher data Transfer speeds.

There are various expansion cards available for machines that doesn't directly support other interfaces. They are called HBA cards.

PCIe expansion cards / Adapters

~~→ PCIe to SAS~~



- Graphics cards (GPU)
- Network interface cards (NIC)
- Sound cards
- Storage Controllers (NVMe SSD Controllers and RAID controllers)
- HBA card (Host Bus Adapter)

HBA card is PCIe expansion card that is used to connect & Manage storage Devices HDD, SSD,s.

Types of HBA cards out interfaces

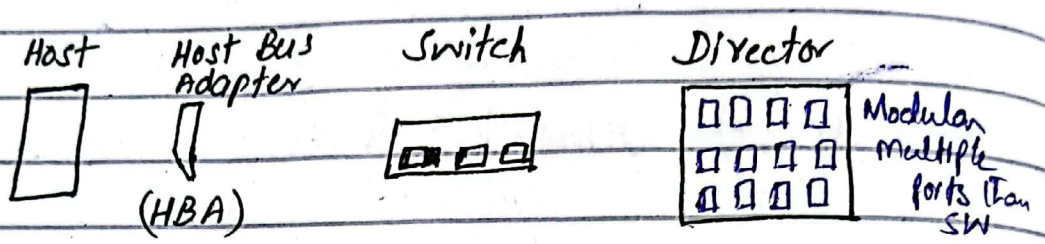
→ SAS to PCIe → iSCSI to PCIe

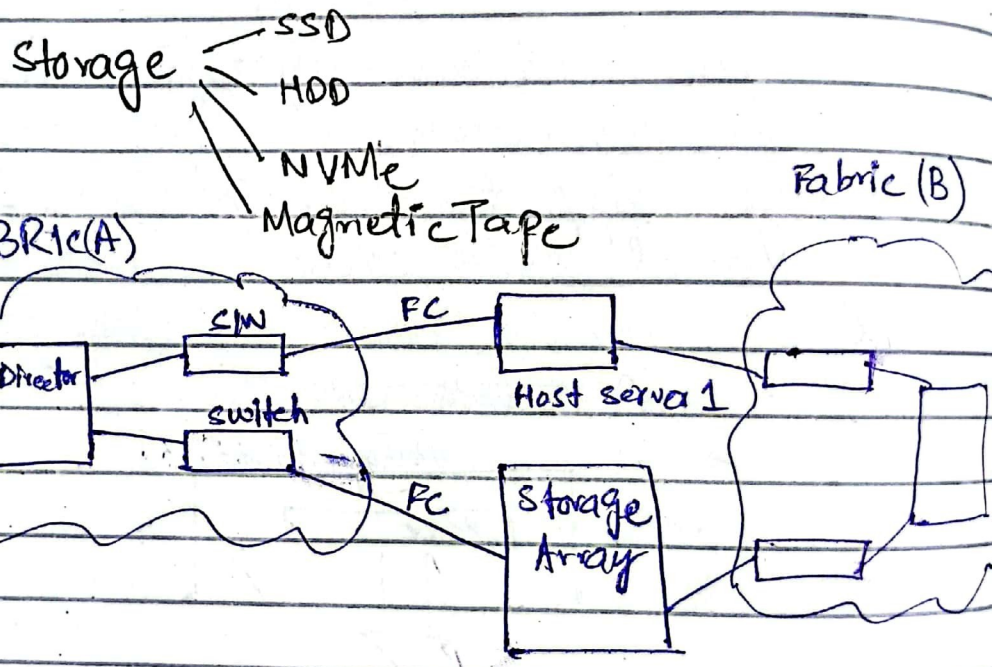
→ Fiber channel to PCIe → SATA to PCIe

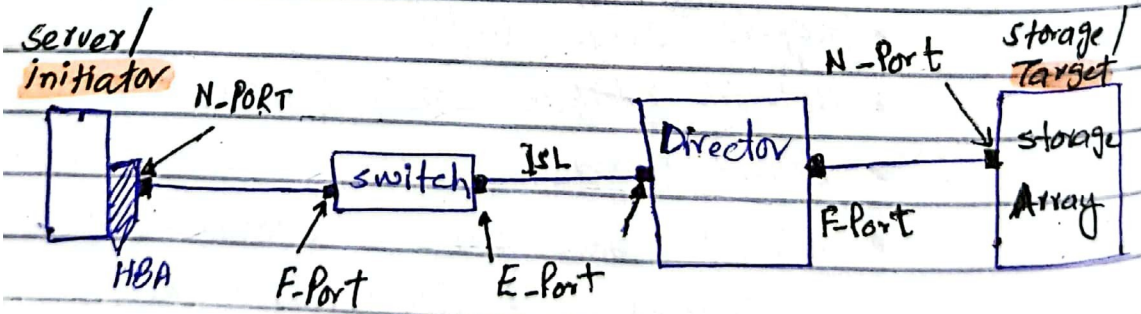
SAN (Storage Area Network)

'Network connected storage'

Components of Fiber channel SAN





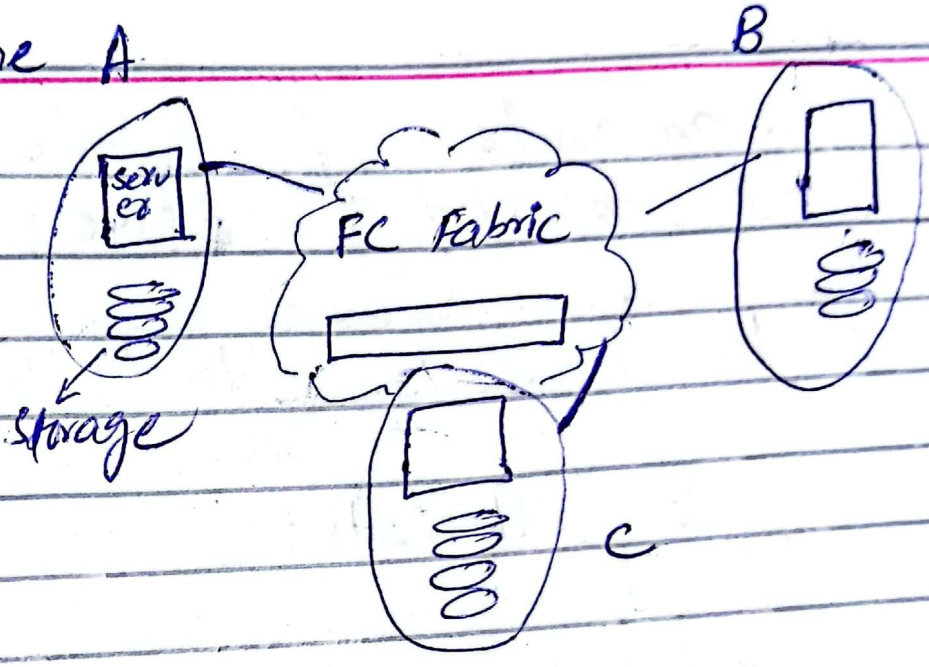


WWN:

each port has unique wordwide port name & Node name assigned by manufacturer.

10:00:00:05:1e:15:27

Zoning :-



STORAGE:

Local
Remote
cluster Based

Local storage :-

partitioning standards
- → Master Boot Record (MBR)
- → GUID partition Table

storage consumption
- → NVDIMM Management
- → Block Storage Manag
- → File Storage
  - ↙ xfs / ext4, NFS & SMB

LVM (logical volume Manager)

Local File Systems
- Rhel → xfs, ext4 → legacy

Remote storage:

connectivity options
- → iSCSI
- → Fiber channel (FC)

NVMe

Network File systems
- NFS, SMB

cluster Based storage:

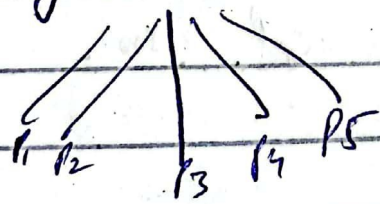
GFS2

Process Daemons :-

ps -el < All running processes >
pgrep <process name> / pidof <proces>
lscpu
ps tree # shows process tree
top # lists top process wrt CPU & memory consumption -

#:() {:|:& };:
↳ Fork Bumb 😟

systemd ⟶ PID = 1 PPID = 0



init process → PID = 1 , PPID = 0

init is first process called by kernel during boot process-

---

ps -aux
kill command signals
kill signal → kernel

Processes Handling

Important Process-Related Commands in RHEL

Viewing Processes

1. **ps**
   - **Description:** Displays information about active processes.
   - **Example:** `ps aux` shows all running processes with detailed information.
   - **Example:** `ps aux | grep chrome` check particular process.

2. **top**
   - **Description:** Provides a dynamic, real-time view of running processes.
   - **Example:** Simply run `top` and it will show an interactive list of processes.

3. **htop**
   - **Description:** An interactive process viewer similar to `top`, but with a more user-friendly interface.
   - **Example:** Run `htop` (requires installation).

4. **pgrep**
   - **Description:** Searches for processes by name or other attributes.
- **Example:** `pgrep ssh` lists the PIDs of all SSH processes.

Managing Processes

5. **kill**
   - **Description:** Sends a signal to a process, usually to terminate it.
   - **Example:** `kill 1234` sends the default `TERM` signal to process 1234.

6. **killall**
   - **Description:** Sends a signal to all processes running a specified command.
   - **Example:** `killall firefox` terminates all instances of Firefox.

7. **pkill**
   - **Description:** Sends a signal to processes based on name and other attributes.
   - **Example:** `pkill -9 apache` forcibly kills all Apache processes.

8. **nice**
   - **Description:** Starts a process with a specified scheduling priority.
   - **Example:** `nice -n 10 myscript.sh` runs `myscript.sh` with a lower priority.

9. **renice**
   - **Description:** Changes the priority of an already running process.
   - **Example:** `renice 10 -p 1234` changes the priority of process 1234.

Monitoring and Debugging

10. **strace**

   - **Description:** Traces system calls and signals.
   - **Example:** `strace -p 1234` traces the system calls of process 1234.

11. **lsof**
   - **Description:** Lists open files and the processes that opened them.
   - **Example:** `lsof -i :80` lists processes using port 80.

12. **pidstat**
   - **Description:** Reports statistics for Linux tasks.
   - **Example:** `pidstat -u` reports CPU usage by process.

13. **vmstat**
   - **Description:** Reports virtual memory statistics.
   - **Example:** `vmstat 1` updates virtual memory statistics every second.

14. **iostat**
   - **Description:** Reports CPU and I/O statistics.
   - **Example:** `iostat -x 2` reports extended statistics every 2 seconds.

Starting and Stopping Services

15. **systemctl**
   - **Description:** Controls the systemd system and service manager.
   - **Example:** `systemctl restart httpd` restarts the Apache service.

16. **service**
   - **Description:** Controls services in RHEL 6 and earlier.
   - **Example:** `service httpd restart` restarts the Apache service.

Process Control

17. **bg**
   - **Description:** Resumes a suspended job in the background.
   - **Example:** `bg %1` resumes job number 1 in the background.

18. **fg**
   - **Description:** Brings a background job to the foreground.
   - **Example:** `fg %1` brings job number 1 to the foreground.

19. **jobs**
   - **Description:** Lists background jobs.
   - **Example:** `jobs` shows all jobs started in the current terminal session.

20. **nohup**
   - **Description:** Runs a command immune to hangups, with output to a non-tty.
- **Example:** `nohup myscript.sh &` runs `myscript.sh` even after logging out.

Miscellaneous

21. **pskill**
    - **Description:** Windows equivalent tool available via `epel-release`.
    - **Example:** `pskill -u user_name` kills all processes owned by a specified user.

22. **pmap**
    - **Description:** Reports memory map of a process.
    - **Example:** `pmap 1234` reports the memory map of process 1234.

002. Network Administration

RHEL Intensive - Network User administration

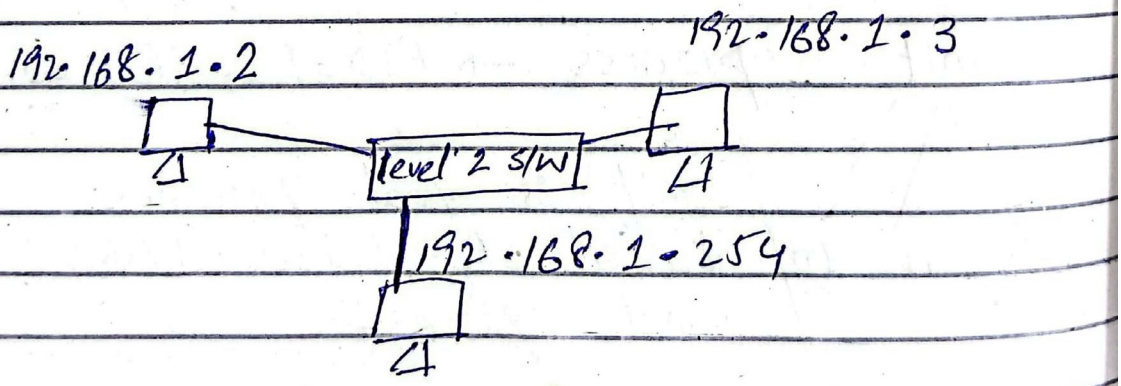
Network configuration

⇒ ifconfig

⇒ In legacy Rhel sys there exist file 'ifcfg' having Network settings
/etc/sysconfig/network-scripts/ifcfg

⇒ In Recent Rhel versions Network Manager is used instead of ifcfg
nmcli connection show

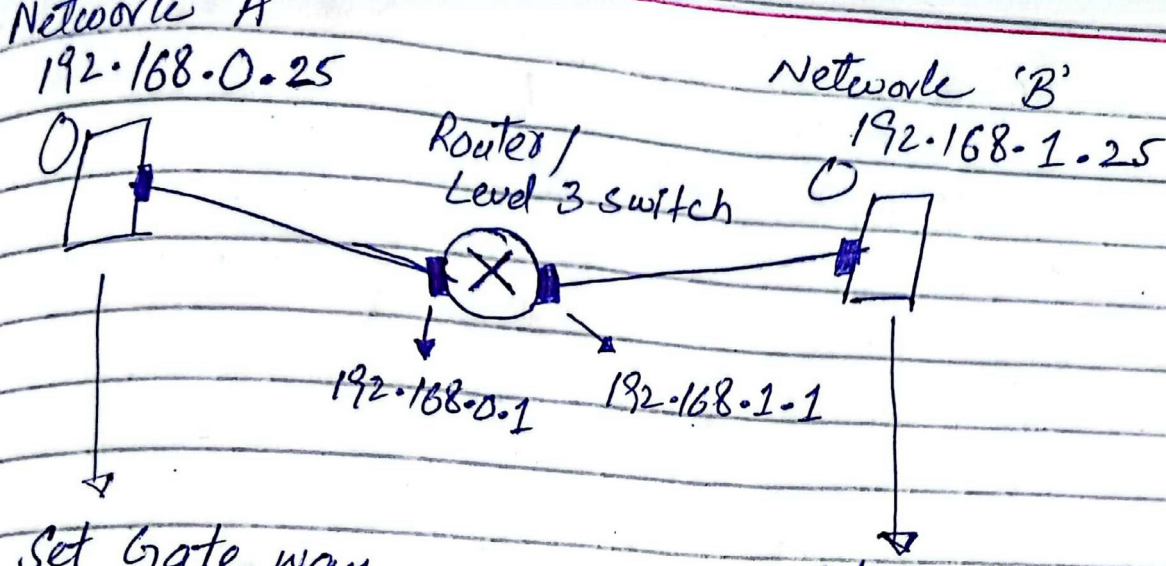
nmtui "Gui based nm"



192 . 168 . (1) . (1)
Network Address
Host IP Address

192.168.1.(1) → Network Adress-
192.168.1.(255) → ~~loop back Adress~~ / Broad cast IP

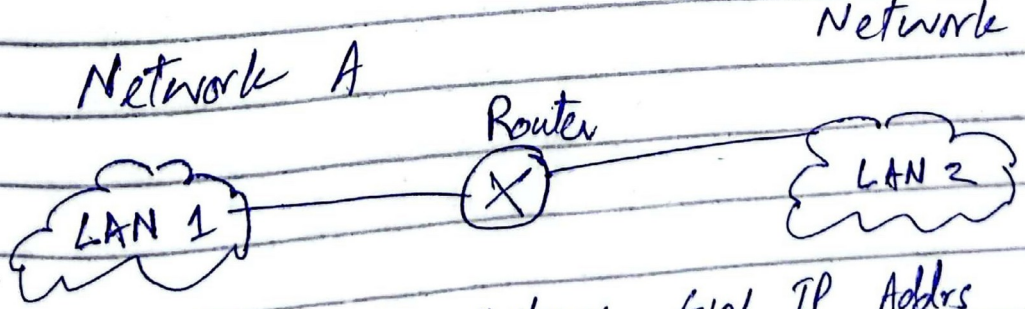
127.0.0.1 → loop Back Addrs, used for Machine to communicate with itself.



#. Note Router IP is used as Gateway Address to host Network - oneside

Note Router Port ~~State~~ which is Connected to Network is assigned Network Address of that Network as IP -

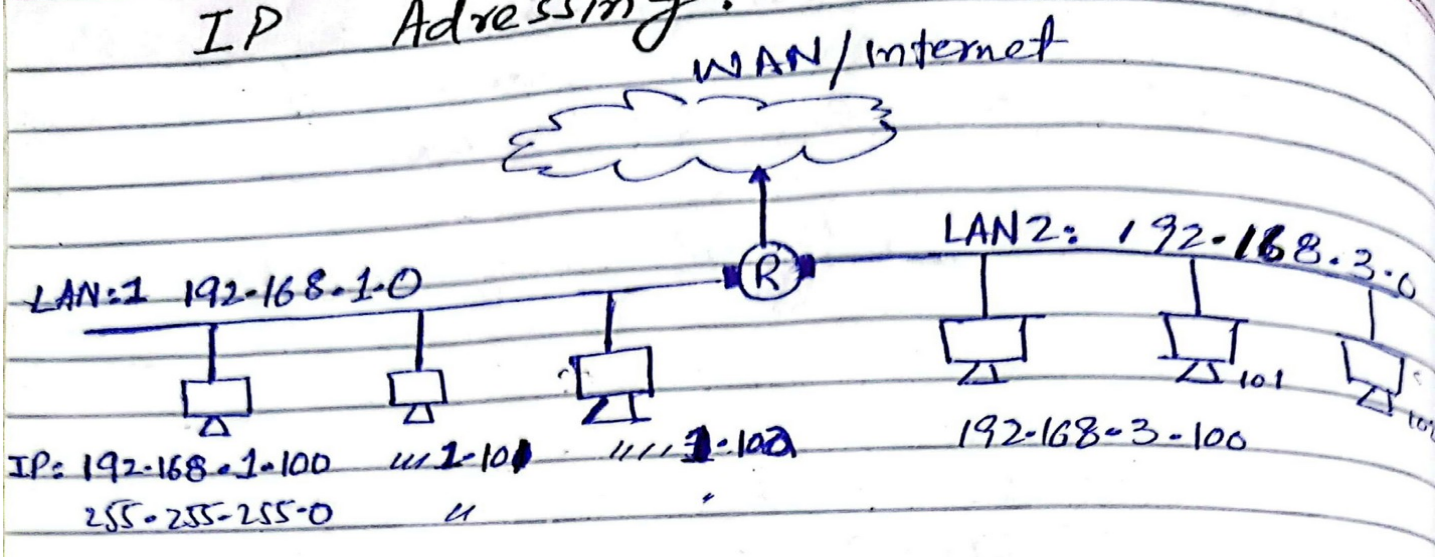
"Routers are used to connect two different physical Networks - helps to jump from one Network to other."



route -n #shows GW IP Addrs

route add default gw 192.168.1.1

IP Adressing :-



Classfull IP Addressing (IpV4) -

Decimal → 128 . 11 . 3 . 31

Binary → 10000000 00001011 00000011 00011111

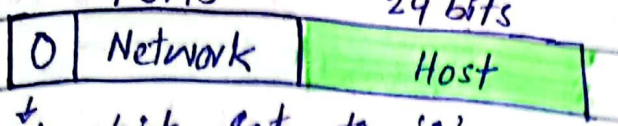
8bit 8bit 8bit 8bit

⇒ 32 bit or 4 byte Number

IP Address | Network Portion | Host Portion

	Byte 1	Byte 2	Byte 3	Byte 4	Net Mask
class A	Network ID	HOST ID			255.0.0.0
class B	Network ID		HOST ID		255.255.0.0
class C	Network ID			Host ID	255.255.255.0
class D	Multicast Address				
class E	Reserved				

Class A IP Addressing:



network Mask
255.0.0.0

First bit set to '0' to represent class 'A'.

Possible No of Networks : 2^7 = 128 [1.0.0.0 – 127.000]

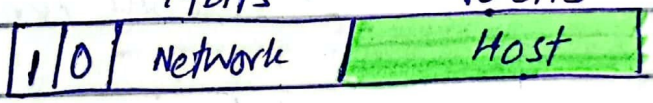
Possible NO of Hosts per Network : $2^{24} - 2$ = 16,777,214

start Address = 1.0.0.1 – 127.255.255.254

Network Adress ⇒ Always start Address of IP range 0.0.0.0

Broadcast Adress ⇒ Always Last Addrs of IP range (127.255.255.255).

class B IP Addressing

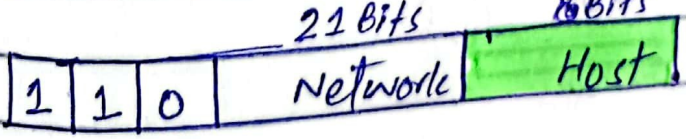


net mask
255.255.0.0

Possible No of Networks : 2^{14} = 16,384 [128.0.0.0 – 191.255.0.0]

Possible No of IP address/network = 2^{16} = 65536 [128.0.0.1 – 191.255.255.254]

class C IP Adressing –



net mask
255.255.255.0

No of possible Networks : 2^{21} = 2,097,152 [192.0.0.0 – 223.255.255.0]

No of IP Addrs / Network : 2^8 = 256 [192.0.0.1 – 223.255.255.254]

	Network	Hosts
Class A	2^7 = 128 networks	$2^{24} - 2$ = 16,777,214 Hosts
	1.0.0.0 – 127.0.0.0	1.0.0.1 – 127.255.255.254
class B	2^{14} = 16,384	2^{16} = 65,536
	128.0.0.0 – 191.255.0.0	128.0.0.1 – 191.255.255.254
class C	2^{21} = 20,97,152	2^8 = 256
	192.0.0.0 – 223.255.255.0	192.0.0.1 – 223.255.255.254
class D		

Problem with classfull Addressing –

Suppose we have requirement to have '2^{10}' hosts IPs ?
we can choose class B and will get 2^{16} host IPS
But we will only use '2^{10}' IPs
$2^{16} - 2^{10} = 2^6$ IP will unused

So we need flexible Solution
IANA has made Solution
and Solution is
CIDR (classless inter domain Routing)
in which There are No fixed classes, instead user can get only No of IPs he Needs
Not more Not less –

Classless IP Addressing -

notation : a.b.d.c /(n) → prefix / mask

→ There is no FIX class

→ we can get only as much IPs as we need.

example:- We need 1000 IP addresses

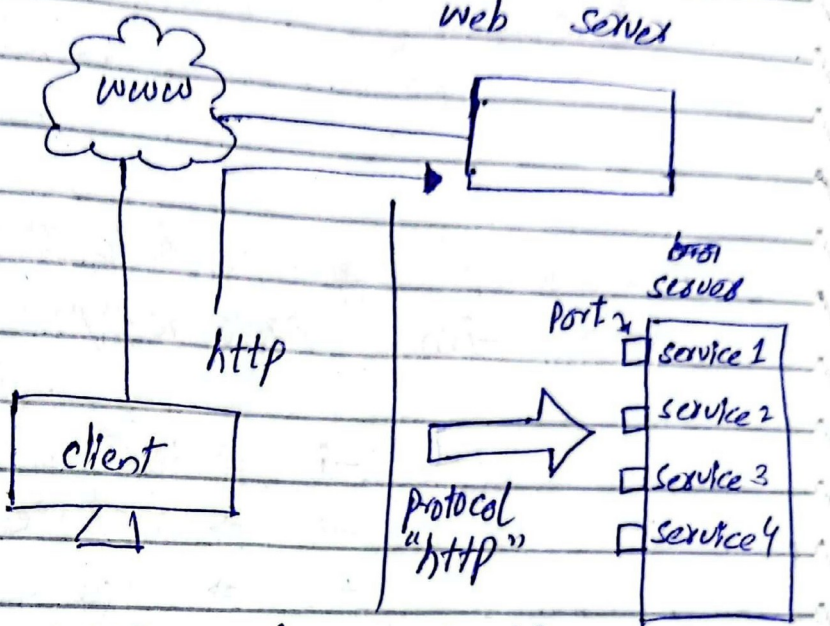
lets calculate min No of Host Bits

$$\log 2(1000) = 10$$

Total no of bits In IPv4 = 32

32 - 10 = 22 bits for Networ or Blocke

Protocols, ports, services



protocol : http, ftp, smtp, ssh

ports : at application layer each protocol is associated with a port.

0 – 1023 privileged ports

* nmap : used to check opened port status of remote computer without logging in -

* netstat
  - ping
  - Trace route
  - wireshark
  - nmon
  - tcp dump
  - netstat
  - iptraf
  - iproc

Installing packages

Red hat package Manager — debian
.rpm — .deb

dpkg

rpm is manual way, we have to install dependencies first

① rpm -ivh package.rpm

-i ⇒ install
-v ⇒ verbose
-h ⇒ hash

② rpm -qa | grep -i zsh

(To check whether package installed)

② Yum (yellow dog update Manager)

SWAP: virtual Memory
works like RAM but
is in Hard disk
swap stores process -

→ Virtual file sys (VFS) -

→ swapping or paging -

free -m (to check swap)

swap -s (to check swap partition)

/dev/sda3

fdisk -l

Extending swap space
① fdisk
② file way

To check which process is using swap

pidof firefox
>> 2405
cd /proc/2405
less status # status of 2405 process
↓
Vm swap ~~fied~~

Performance Monitoring tools (Interview prep)

Q: Server is in high utilization mode, how to diagnose?

#1 top command
pinpoint high consuming process
by check 'id' 100% low 0% high

#2 df -h
check if Harddisk/partition is Full

iostat
to check Read/write stats of Harddisk

#3 check Network load

- tcpdump
- nmon
- wireshark
- ip def
- sar

} network Monitoring Tools

#4 check Buffer and cache

Buffer → In Ram
cache → In Swap
(temp data in Hard disk)

~~top~~

Due to memory leakage problem (application didnot free memory) ~~swap~~ & cache may increase significantly -

- valgrind tool (to check for memory leakage)

To clear the buffer & cache
synch; echo 3 > /proc/sys/vm/drop_caches

top, ps, iostat, sar, vmstat, pidstat, strace, ipcs, @lsof, htop, iometer nmon, tcpdump

iostat -x -d -p
sda1 r/s w/s rsec/s wsec/s
sda2

strace → system call trace of a process / command.

top -flags **

top -h -help
top -u - filter process of user
top -c - high util process on top
top -m - big memory consuming

Linux Process States

→ Running R
→ sleeping S
→ Zombie Z

Zombie is Dead process, still present in the memory (RAM) & is waiting for Parent process to kill it.

How to kill zombie process
Reboot Machine

To check zombie process
>> top → zombie 0

CPU utilization info in 'top' cmd

Cpu(s): us , sy , ni , wa , hi , si , st

us : utilization by user in %age
sy : " " system " ,
ni : niece value
id : 100% (low) , 0% (High)
wa : weight
hi : Hardware interrupt
si : Software interrupt
st : Software Trace

changing priority of a process -

Priority = -19 (Highest) , +20 (lowest)

process with niece value '-19' will have highest priority which means that it will execute first in CPU cycle -

To change niece value -

1) top
2) Enter key 'r'
3) Enter pid value of process
4) Enter renice value fo ex -10

-19 — 20
highest ↙ ↘ lowest

If process takes 2hrs to complete then

after increasing priority, it could finish in 1.5 hrs, so by changing NICE value we can make a process slow or fast -

FSH

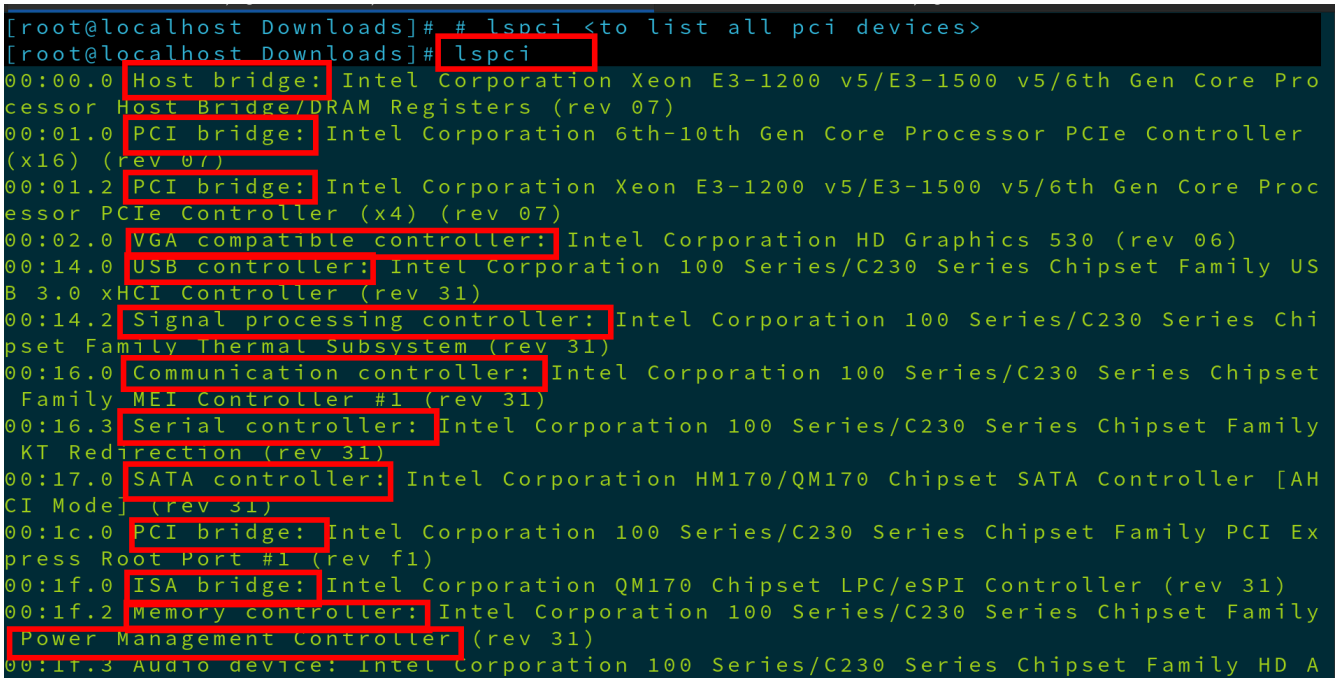
/boot → contains kernel File

vmlinuz-5.14.0-427.13.1.el9_4.x86_64 (13MB)

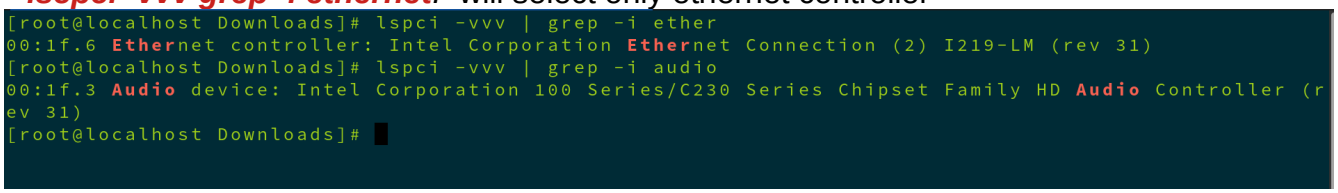
/boot → contains boot loader grub

Network Configuration

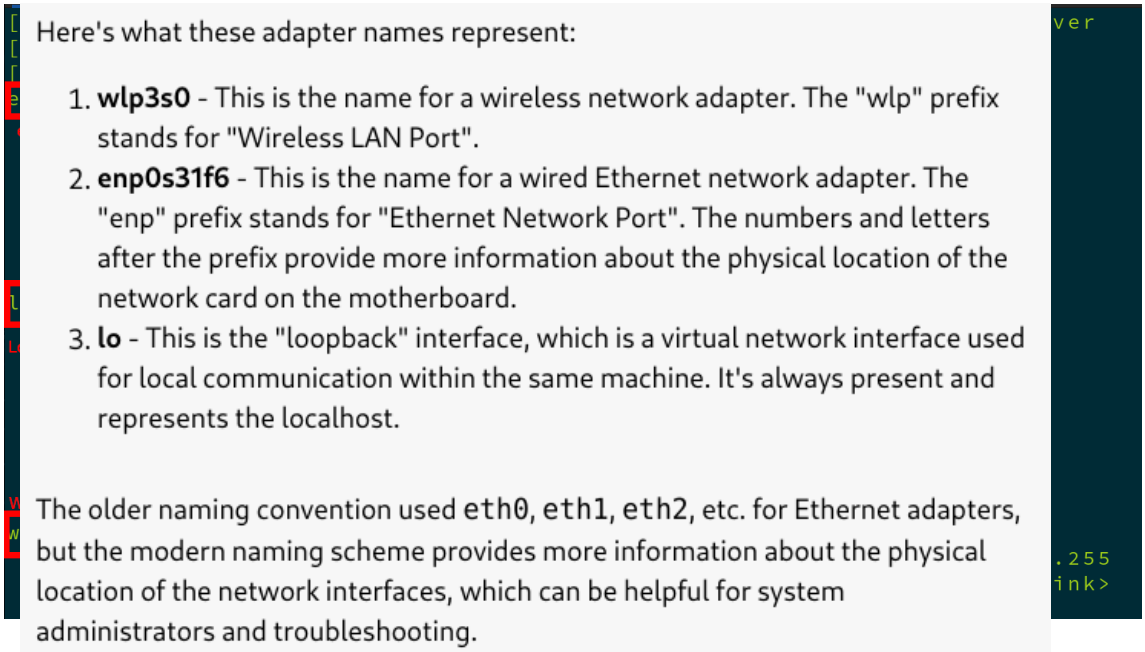
lscpci: to list all pci devices connect with system



lscpci -vvv grep -i ethernet: will select only ethernet controller



ifconfig -a: will show ethernet connections along with their adapter names, in older versions they are named as eth0 eth1 eth2. In Rhel9 naming scheme has changed.



More about Naming schemes for network interfaces:

Network interface naming schemes or policies refer to the conventions used by operating systems to automatically assign names to network interfaces on a system. The naming conventions have evolved over time to provide more consistent and informative names.

The most common network interface naming schemes are:

1. **Traditional naming scheme**:
   - This was the original naming scheme used in older Linux distributions.
   - Network interfaces were typically named `eth0`, `eth1`, `eth2`, etc., based on the order in which they were detected by the system.
   - This scheme could be problematic when interfaces were added or removed, as the names might change unexpectedly.
2. **Predictable network interface names (PNIS)**:
   - This naming scheme was introduced in systemd, a system and service manager for Linux.
   - The goal was to provide more predictable and persistent network interface names,
3. **Consistent Network Device Naming (CNDN)**:
   - This is an alternative naming scheme developed by Dell, and it is used in some Linux distributions, such as Red Hat Enterprise Linux (RHEL) and CentOS.
   - CNDN names follow a similar pattern to PNIS, but the names are based on the physical location of the network interface on the motherboard.
   - Examples of CNDN names include `eno1`, `ens1`, `enx78e7d1234567`, etc.
4. **Netplan-based naming**:
   - This is a newer naming scheme introduced with the Netplan network configuration tool in Ubuntu 17.10 and later.
   - Netplan-based names follow a similar pattern to PNIS and CNDN, but they can also include user-friendly names, such as `ethernet` or `wifi`.
   - Examples of Netplan-based names include `enp0s31f6`, `wlan0`, `ethernet-enp0s31f6`, etc.

The choice of network interface naming scheme can depend on the Linux distribution, the system hardware, and the preferences of the system administrator. The predictable and consistent naming schemes (PNIS, CNDN, and Netplan-based) are generally preferred over the traditional `ethX` naming, as they provide more stability and easier identification of network interfaces.

Ethtool <adaptername>: to list settings for adapter

```
[root@localhost Downloads]# ethtool enp0s31f6
Settings for enp0s31f6:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: Symmetric Receive-only
        Supports auto-negotiation: Yes
        Supported FEC modes: Not reported
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: Symmetric Receive-only
        Advertised auto-negotiation: Yes
        Advertised FEC modes: Not reported
        Speed: Unknown!
        Duplex: Unknown! (255)
        Auto-negotiation: on
        Port: Twisted Pair
        PHYAD: 2
        Transceiver: internal
        MDI-X: Unknown (auto)
        Supports Wake-on: pumbg
        Wake-on: g
        Current message level: 0x00000007 (7)
                               drv probe link
        Link detected: no
```

Change IP address and netmask by using ifconfig command:

```
[root@localhost Downloads]# ifconfig enp0s31f6 192.168.1.103 netmask 255.255.255.0
[root@localhost Downloads]# #ifconfig
[root@localhost Downloads]# ifconfig
enp0s31f6: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.1.103  netmask 255.255.255.0  broadcast 192.168.1.255
        ether c8:5b:76:67:13:23  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  memory 0xf2100000-f2120000
```

In the previous RHEL versions IP settings were saved in ifcfg file <etc/sysconfig/network-scripts/ifcfg> However in recent versions ifcfg format is ***deprecated***

```
[root@localhost Downloads]# cd /etc/sysconfig/network-scripts/
[root@localhost network-scripts]# ls
readme-ifcfg-rh.txt
[root@localhost network-scripts]# ^C
[root@localhost network-scripts]# cat readme-ifcfg-rh.txt
NetworkManager stores new network profiles in keyfile format in the
/etc/NetworkManager/system-connections/ directory.

Previously, NetworkManager stored network profiles in ifcfg format
in this directory (/etc/sysconfig/network-scripts/). However, the ifcfg
format is deprecated. By default, NetworkManager no longer creates
new profiles in this format.
```

Im using Rhel 9 and here network configurations are saved in **/etc/NetworkManager/system-connections.**

```
[root@Eagle shujah]# ls /etc/NetworkManager/system-connections
 ethernet1.nmconnection  'The Falcon.nmconnection'
[root@Eagle shujah]#
```

Making Persistent IP changes: above ip remains changed until system doesn't restart. To make persistent IP we have to modify setting in network configuration file.Change autoconnect = false to auto connect = true and save.

```
root@Eagle shujah]# cat /etc/NetworkManager/system-connections/ethernet1.nmconnection
connection]
id=ethernet1
uuid=8010aa57-fb78-42f6-8f50-aa13cc83fcc3
ype=ethernet
utoconnect=false
nterface-name=enp0s31f6
timestamp=1716595382

ethernet]
ac-address=C8:5B:76:67:13:23

ipv4]
ddress1=192.168.1.120/24,192.168.1.1
ethod=manual

ipv6]
ddr-gen-mode=default
ethod=ignore

proxy]
root@Eagle shujah]#
```

Make connection Active:
*After that type **ifconfig** command in terminal, if your connection isnt showing it means its down check all connections by **ifconfig -a***
*make connection up: **ifconfing enp0s31f6 up***
Check again ifconfig your connection should show if its active.

Restart Network Manager:
to update changes made in config file restart service.
systemctl restart NetworkManager

Changing Network Settings using Network Manager nmcli / nmtui

Using DHCP : Automatic IP Assignment.

nmcli coonection modify <connectionname> ipv4.method auto

```
[root@localhost /]# #Using nmcli for NW config
[root@localhost /]# nmcli connection show
NAME         UUID                                  TYPE      DEVICE
The Falcon   e722fd4a-3418-4bf6-9ddc-0f02d80a5542  wifi      wlp3s0
lo           43625f70-e496-4087-a4e5-d2a51a4fd97c  loopback  lo
ethernet1    8010aa57-fb78-42f6-8f50-aa13cc83fcc3  ethernet  --
[root@localhost /]# # to use DHCP
[root@localhost /]# nmcli connection modify ethernet1 ipv4.method auto
```

Using Static : Manual IP Assignment.

nmcli coonection modify <connectionname> ipv4.method manual ipv4.addresses <ipaddrss/netmask> ipv4.gateway <gate way ip address>

```
[root@Eagle shujah]# ls /etc/NetworkManager/system-connections
 ethernet1.nmconnection  'The Falcon.nmconnection'
[root@Eagle shujah]#
```

Making Persistent IP changes: above ip remains changed until system doesn't restart. To make persistent IP we have to modify setting in network configuration file. Change autoconnect = false to auto connect = true and save.

```
[root@Eagle shujah]# cat /etc/NetworkManager/system-connections/ethernet1.nmconnection
[connection]
id=ethernet1
uuid=8010aa57-fb78-42f6-8f50-aa13cc83fcc3
type=ethernet
autoconnect=false
interface-name=enp0s31f6
timestamp=1716595382

[ethernet]
mac-address=C8:5B:76:67:13:23

[ipv4]
address1=192.168.1.120/24,192.168.1.1
method=manual

[ipv6]
addr-gen-mode=default
method=ignore

[proxy]
[root@Eagle shujah]#
```

Activate your connection:

nmcli connection up <connection name>

```
[root@localhost /]# # activate the profile
[root@localhost /]# nmcli connection show
NAME         UUID                                  TYPE      DEVICE
The Falcon   e722fd4a-3418-4bf6-9ddc-0f02d80a5542  wifi      wlp3s0
lo           43625f70-e496-4087-a4e5-d2a51a4fd97c  loopback  lo
ethernet1    8010aa57-fb78-42f6-8f50-aa13cc83fcc3  ethernet  --
[root@localhost /]# nmcli connection up ethernet1
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/10)
[root@localhost /]# ifconfig
enp0s31f6: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.1.120  netmask 255.255.255.0  broadcast 192.168.1.255
        ether c8:5b:76:67:13:23  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  memory 0xf2100000-f2120000
```

Restart Network Manager:

to update changes made in config file restart service.

systemctl restart NetworkManager

ip address show <connection name>

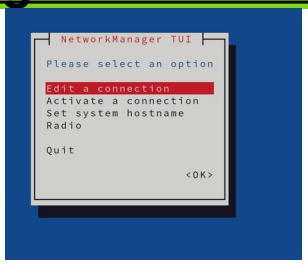
```
[root@localhost /]# #verify ip settings of NIC in my case ethernet port isnt connected with router
[root@localhost /]# # so it will show down state
[root@localhost /]# ip address show enp0s31f6
2: enp0s31f6: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default q
len 1000
    link/ether c8:5b:76:67:13:23 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.120/24 brd 192.168.1.255 scope global noprefixroute enp0s31f6
       valid_lft forever preferred_lft forever
```

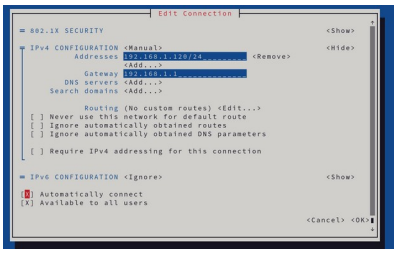
Show Gateway IP:

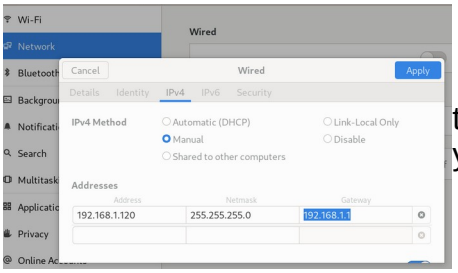
ip route show default

```
[root@localhost /]# ip route show default
default via 192.168.1.1 dev enp0s31f6 proto static metric 100 linkdown
default via 192.168.1.1 dev wlp3s0 proto dhcp src 192.168.1.36 metric 600
[root@localhost /]#
```

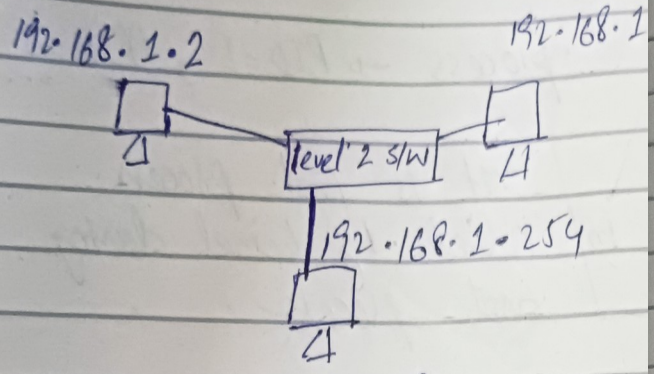
using Network Manger GUI interface nmtui







type ***nmtui*** command in terminal and GUI window will open you can also change you network settings from there:



192 . 168 . (1) . (1) → Host IP

Network Address

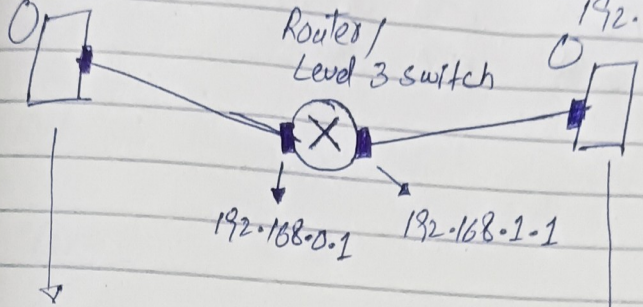
192.168.1.(1) → Network Adress

192.168.1.(255) → ~~Loop back~~

Broad cast IP

Network 'A'
192.168.0.25

Network 'B'
192.168.1.25



Set Gate way
192.168.0.1

Set Gate way
192.168.1.1

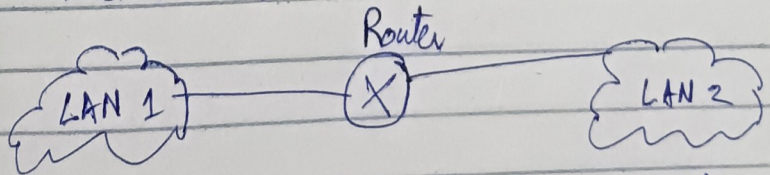
Note Router IP is used as Gateway Address to host Network oneside

Note Router Port ~~side~~ which is Connected to Network is assigned Network Address of that Network as IP.

"Routers are used to connect two different physical Networks. helps to jump from one Network to other."

Network A

Network B



route -n #shows GW IP Addrs

route add default gw 192.168.1.1

DNS / Resolv.conf

Setting up Default GW
method 1: ***route add default gw 192.168.1.1***
method2: ***sudo nmcli connection modify Ethernet ipv4.gateway 192.168.1.1***

To show default Gateway IP
route -n
ip route show default

DNS name server
etc/resolv.conf can have multiple Name Server IPs
#adding name server
1) ***vi /etc/resolve.conf***
2) press i for insert mode
3) add line nameserver <ip of nameserver>
Google DNS IP 8.8.8.8
Open DNS IP: 208.67.222.123 / 208.67.220.123

When we type facebook.com in our browser it first reads file resolv.conf. Then it goes to DNS server given in the file for example 8.8.8.8 google DNS, after that it get the IP of facebook.com from DNS List. Then it sends http GET/POST request at the IP of Facebook.

package management

RHEL and CentOS (RPM-based systems)

1. Using `rpm`
- **Install a package**:
  rpm -i package_name.rpm

- **Update a package**:
  rpm -U package_name.rpm

- **Remove a package**:
  rpm -e package_name

- **Verify a package**:
  rpm -V package_name

- **List all installed packages**:
  rpm -qa
  rpm -qa | grep -i <package-name> *(shows if package is installed)*

- **Show package information**:
  rpm -qi package_name

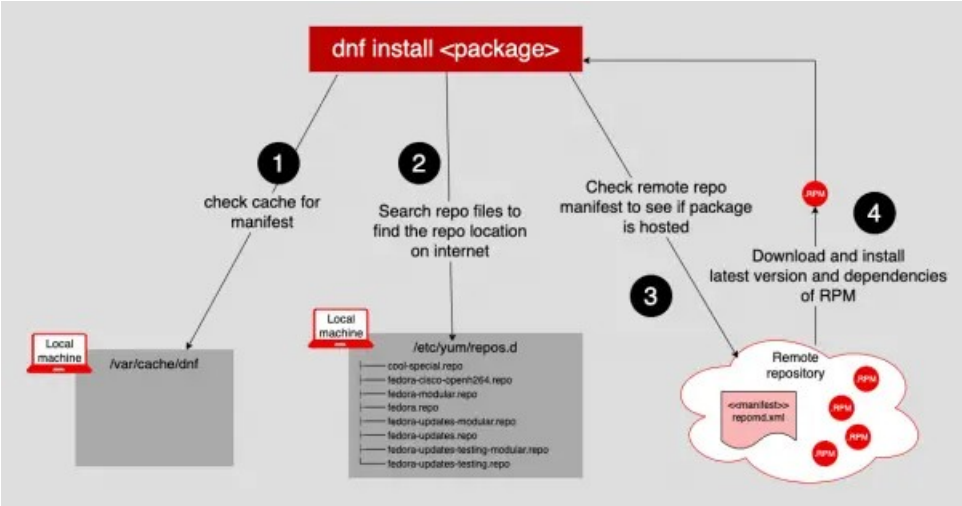
- **List files in a package**:

rpm -ql package_name (shows all files written by a package and their location)
- **List configuration files of a package**:
rpm -qc package_name

- **List help Documents of a package**:
rpm -qd package_name

- **Query which package owns a file**:
rpm -qf /path/to/file

2. Using `yum` (Older RHEL and CentOS versions)
How Yum Works:



When we type yum install <package> it first check for package in the enabled repos
/etc/yum/repos.d directory conatins files for repos like in my case:
[root@server1 yum.repos.d]# ls
epel-cisco-openh264.repo epel.repo epel-testing.repo google-chrome.repo redhat.repo
then yum goes to mirror url (url of public repo on internet) given in the repo and download latest package form there using wget command.
After downloading in local machine yum install package using rpm. Meanwhile it also performs integrity checking of downloaded package if gpgcheck = 1 in the repo file.

-**Check list of all repos**:
sudo yum repolist all

-**Check list of enabled repos**:
sudo yum repolist

-**Check brief info for repos**:
sudo yum repoinfo

-**clear old cache so that yum updates repolists**:
sudo yum clean all

- **Install a package**:
yum install package_name
yum install httpd -y

- **Update a package**:

yum update package_name

- **Update whole system**:
yum update (warning use very carefully)
it updates all installed *packages*, which includes *core system packages* and *kernel* updates.

- **To get report **:
yum check-update
shows that at what versions packages will be updated.

- **update all excluding some**:
yum update -x mysql php
update all excluding mysql and php

- **Remove a package**:
yum remove package_name

- **Search for a package**:
yum search package_name

- **List installed packages**:
yum list installed

- **Clean the cache**:
yum clean all

-** *search repo name which contains package**:*
yum list | grep -i vsftpd
-**brief info of package**:
yum info vstfpd

3. Using `wget`
wget is used to download package, if package is not in our default repo list (checked by *yum search package name*) then we have to downoad it from its url:
wget https://dl.google.com/linux/direct/google-chrome-stable_current_x86_64.rpm
sudo yum localinstall google-chrome-stable_current_x86_64.rpm

3. Using `dnf` (RHEL 8 and later, CentOS 8 and later)
- **Install a package**:
dnf install package_name

- **Update a package**:
dnf update package_name

- **Remove a package**:
dnf remove package_name

- **Search for a package**:
dnf search package_name

- **List installed packages**:
dnf list installed

- **Show package information**:
  dnf info package_name

- **Clean the cache**:
  dnf clean all

Debian-based Systems (Debian, Ubuntu)

1. Using `dpkg`
- **Install a package**:
  dpkg -i package_name.deb

- **Remove a package**:
  dpkg -r package_name

- **List installed packages**:
  dpkg -l

- **Show package information**:
  dpkg -s package_name

- **List files in a package**:
  dpkg -L package_name

- **Query which package owns a file**:
  dpkg -S /path/to/file

2. Using `apt` (Advanced Package Tool)
- **Update package list**:
  sudo apt update

- **Upgrade all packages**:
  sudo apt upgrade

- **Install a package**:
  sudo apt install package_name

- **Remove a package**:
  sudo apt remove package_name

- **Search for a package**:
  apt search package_name

- **List installed packages**:
  apt list --installed

- **Show package information**:
  apt show package_name

- **Clean the local repository (remove cached packages)**:

sudo apt clean

3. Using `apt-get` (Older command, still widely used)
- **Update package list**:
  sudo apt-get update

- **Upgrade all packages**:
  sudo apt-get upgrade

- **Install a package**:
  sudo apt-get install package_name

- **Remove a package**:
  sudo apt-get remove package_name

- **Search for a package**:
  apt-cache search package_name

- **List installed packages**:
  dpkg --get-selections | grep -v deinstall

- **Show package information**
  apt-cache show package_name

- **Clean the local repository (remove cached packages)**:
  sudo apt-get clean

Patching

Patching is the process of updating a package/kernel/OS to a latest available version.
- **Take Approval for patch**
- **stop service**
- **Take Backup of DB**
- **inform**
- **update**

commands for update

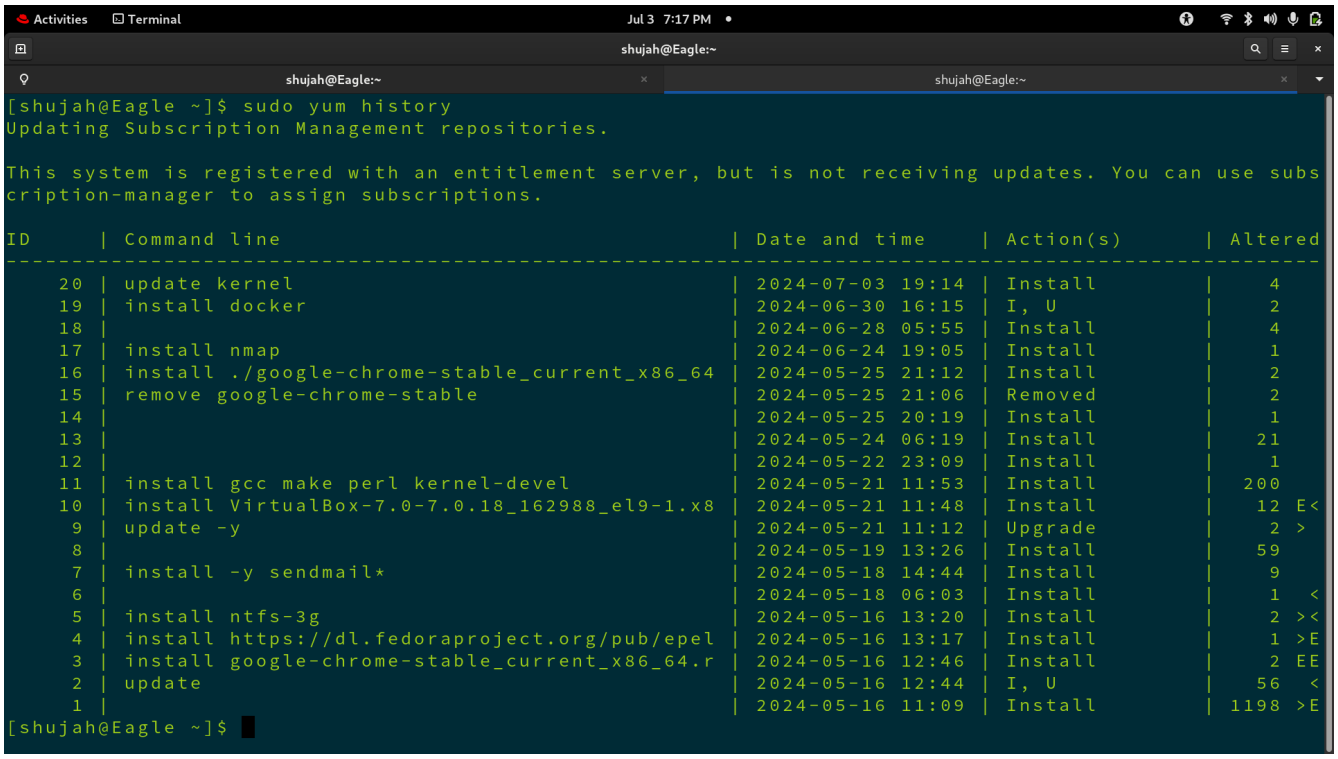
mysql –version
yum check-update mysql
If higher version is available then;
yum update mysql

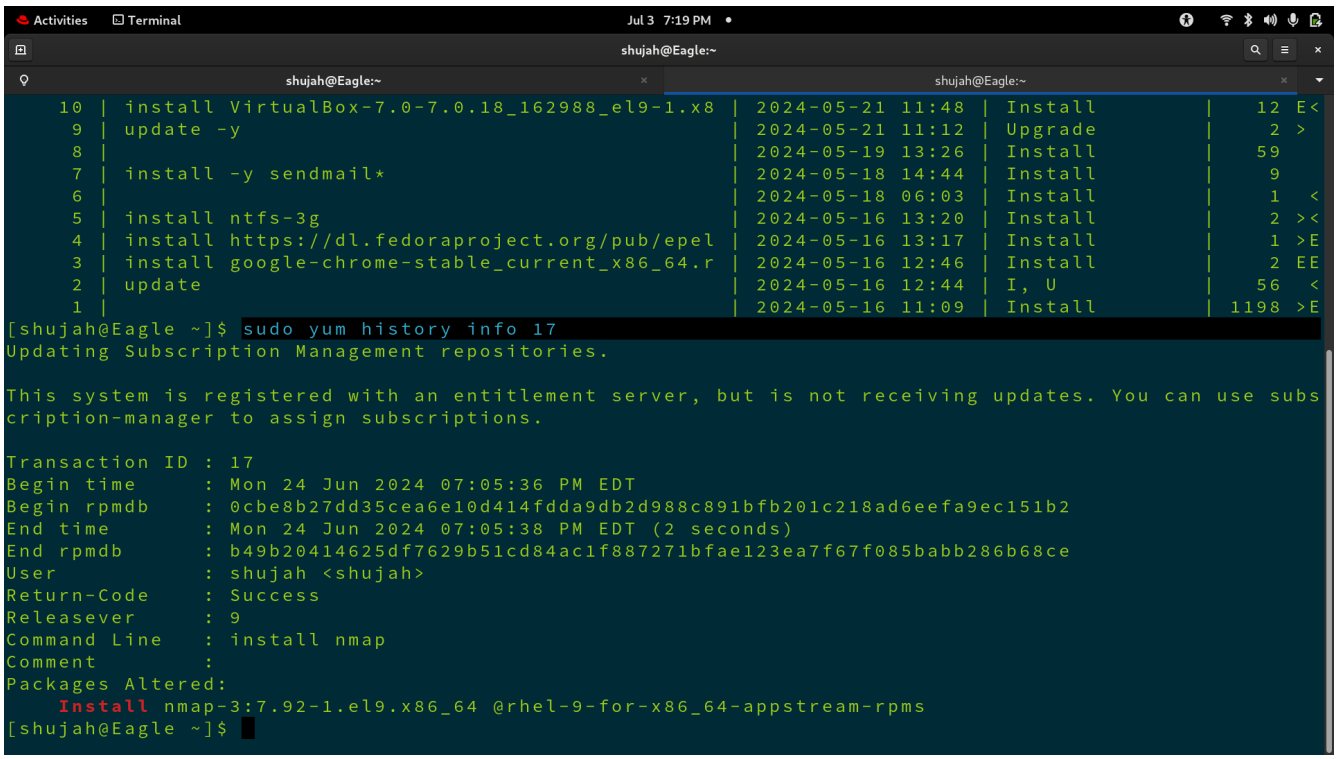
other ways for patching are GUI based -Managengine - space walk(red hat satelite server)

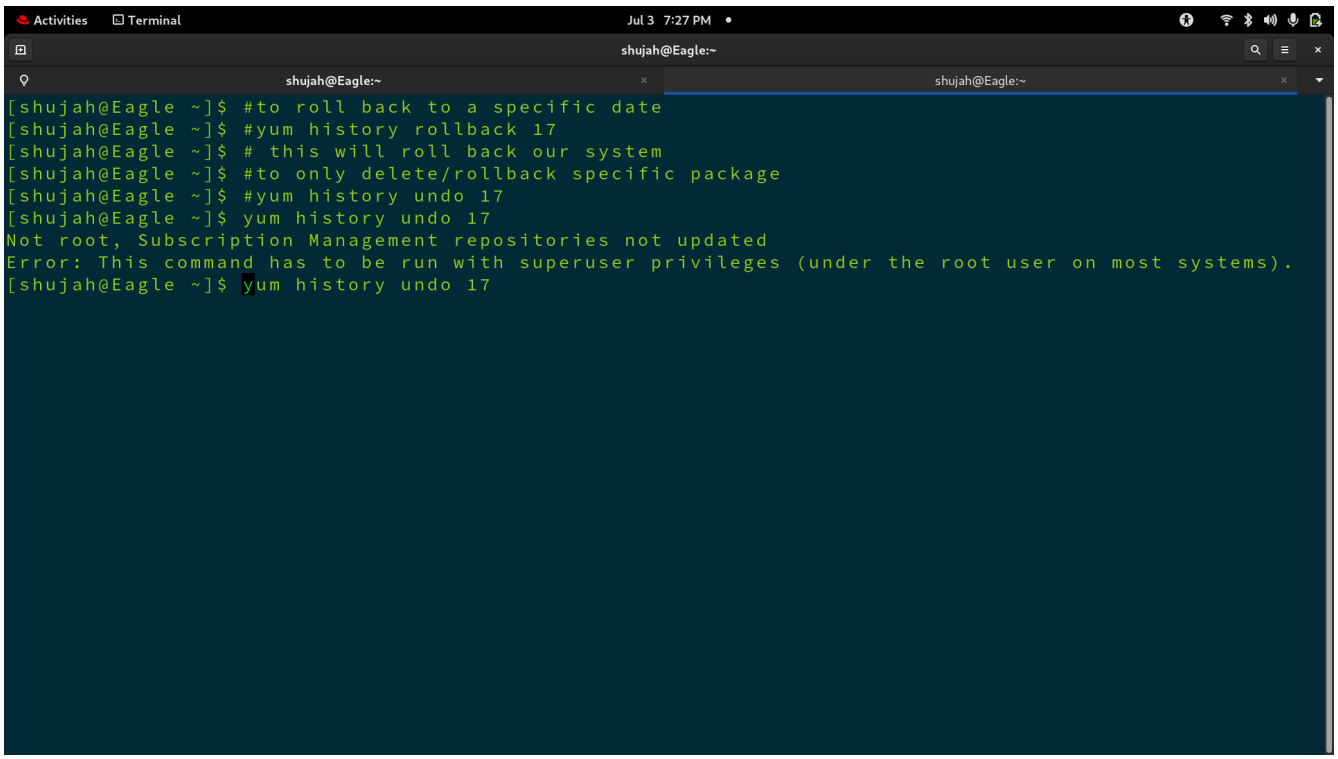
Rollback

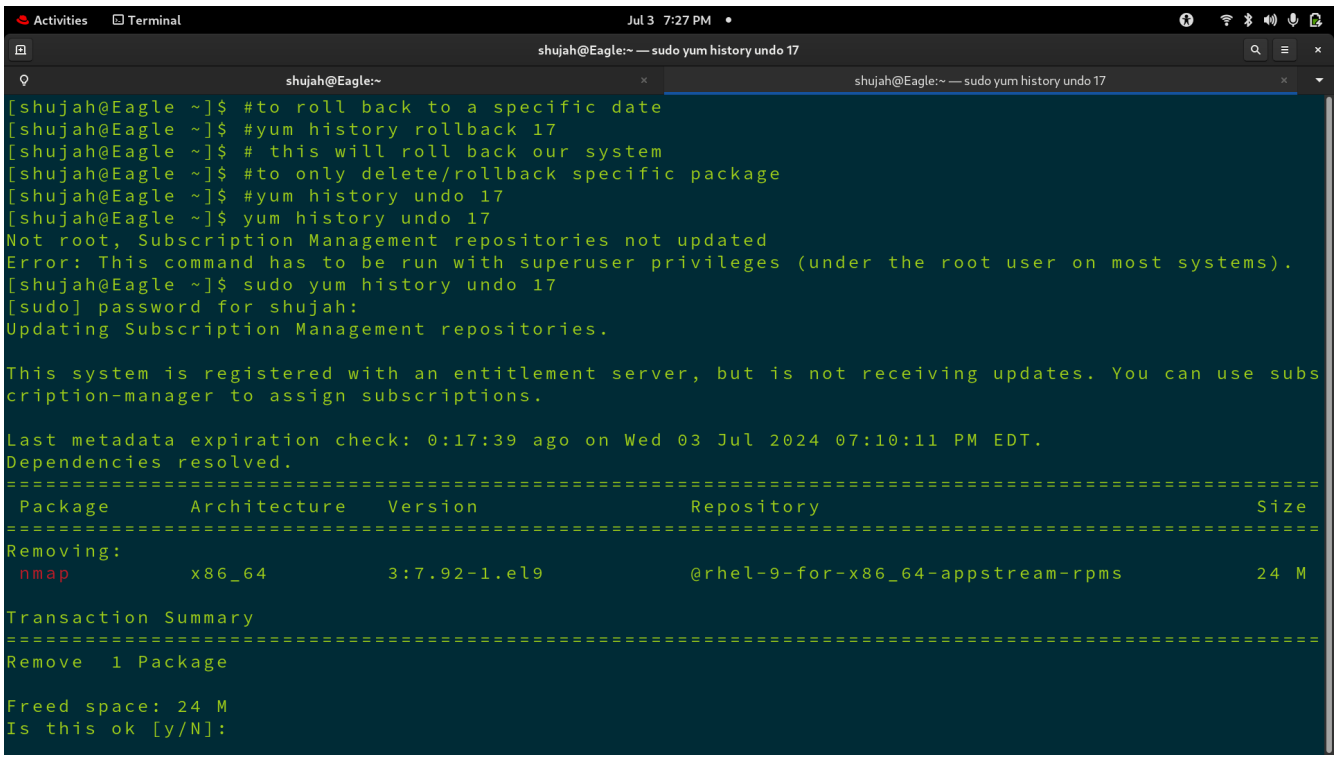
yum history >>> check history of yum.

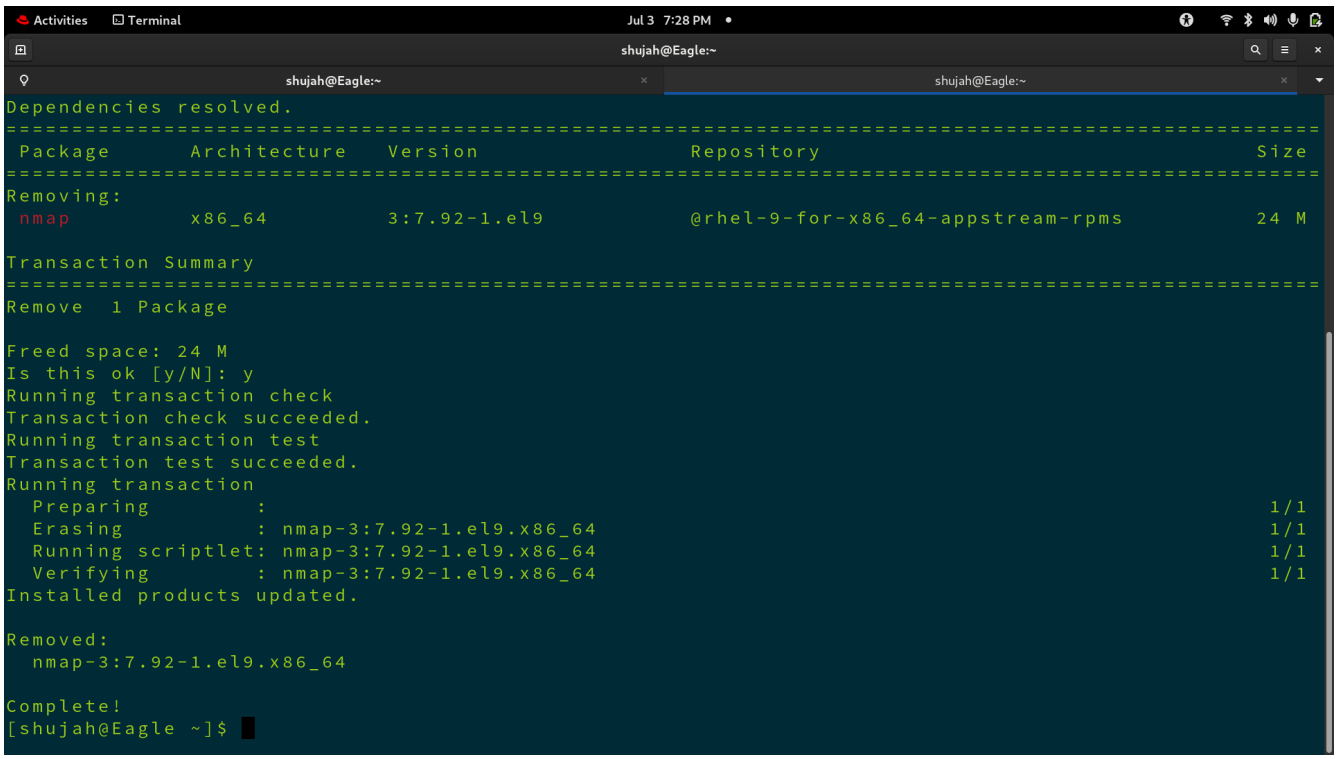
yum history info id-no >>detailed info of package installed.
yum history rollback id-no >>rollback to a specific time/date.
yum history undo id-no >>rollback/delete only given package id.

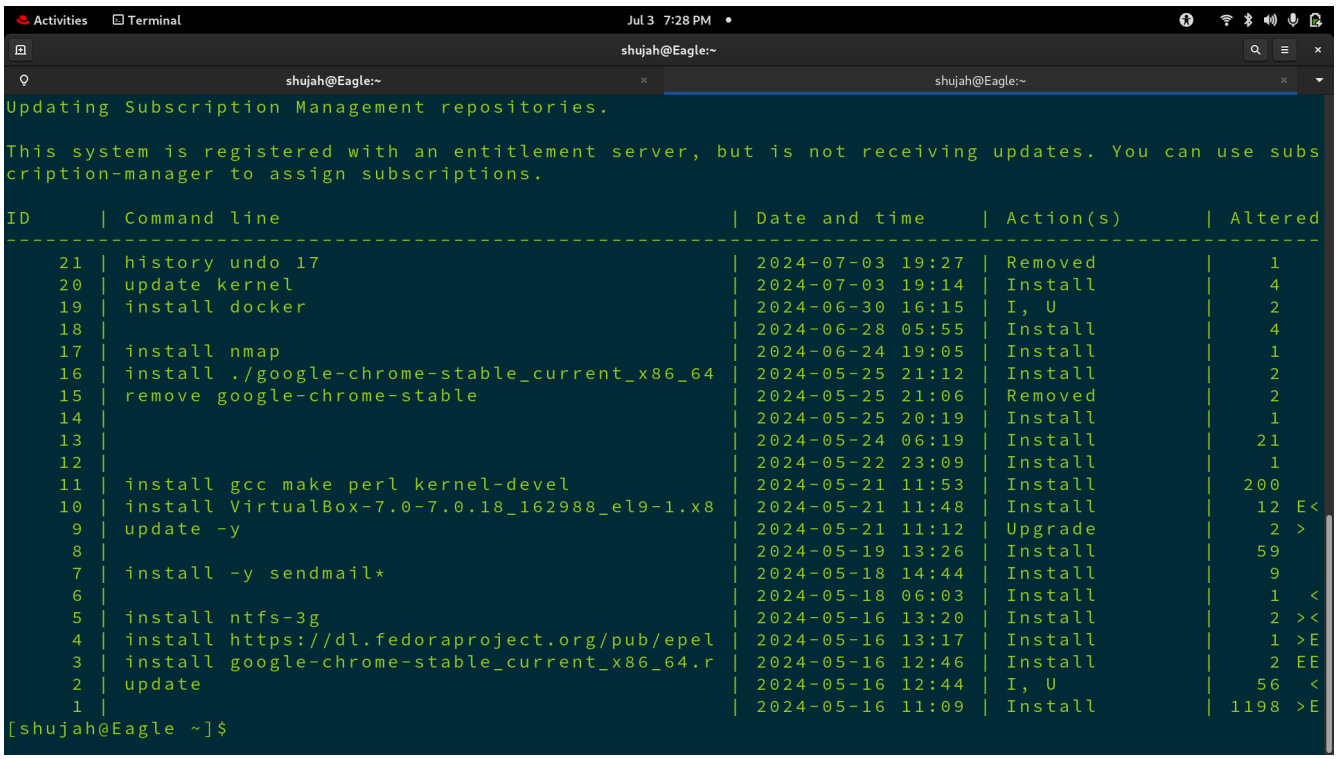



Activities
Terminal
Jul 3 7:19 PM
shujah@Eagle:~
shujah@Eagle:~
shujah@Eagle:~
10 | install VirtualBox-7.0-7.0.18_162988_el9-1.x8 | 2024-05-21 11:48 | Install | 12 E<
9 | update -y | 2024-05-21 11:12 | Upgrade | 2 >
8 | | 2024-05-19 13:26 | Install | 59
7 | install -y sendmail* | 2024-05-18 14:44 | Install | 9
6 | | 2024-05-18 06:03 | Install | 1 <
5 | install ntfs-3g | 2024-05-16 13:20 | Install | 2 ><
4 | install https://dl.fedoraproject.org/pub/epel | 2024-05-16 13:17 | Install | 1 >E
3 | install google-chrome-stable_current_x86_64.r | 2024-05-16 12:46 | Install | 2 EE
2 | update | 2024-05-16 12:44 | I, U | 56 <
1 | | 2024-05-16 11:09 | Install | 1198 >E
[shujah@Eagle ~]$ sudo yum history info 17
Updating Subscription Management repositories.
This system is registered with an entitlement server, but is not receiving updates. You can use subscription-manager to assign subscriptions.
Transaction ID : 17
Begin time : Mon 24 Jun 2024 07:05:36 PM EDT
Begin rpmdb : 0cbe8b27dd35cea6e10d414fdda9db2d988c891bfb201c218ad6eefa9ec151b2
End time : Mon 24 Jun 2024 07:05:38 PM EDT (2 seconds)
End rpmdb : b49b20414625df7629b51cd84ac1f887271bfae123ea7f67f085babb286b68ce
User : shujah <shujah>
Return-Code : Success
Releasever : 9
Command Line : install nmap
Comment :
Packages Altered:
Install nmap-3:7.92-1.el9.x86_64 @rhel-9-for-x86_64-appstream-rpms
[shujah@Eagle ~]$


Activities
Terminal
Jul 3 7:27 PM
shujah@Eagle:~
shujah@Eagle:~
shujah@Eagle:~
[shujah@Eagle ~]$ #to roll back to a specific date
[shujah@Eagle ~]$ #yum history rollback 17
[shujah@Eagle ~]$ # this will roll back our system
[shujah@Eagle ~]$ #to only delete/rollback specific package
[shujah@Eagle ~]$ #yum history undo 17
[shujah@Eagle ~]$ yum history undo 17
Not root, Subscription Management repositories not updated
Error: This command has to be run with superuser privileges (under the root user on most systems).
[shujah@Eagle ~]$ yum history undo 17


Activities
Terminal
Jul 3 7:27 PM
shujah@Eagle:~ — sudo yum history undo 17
shujah@Eagle:~
shujah@Eagle:~ — sudo yum history undo 17
[shujah@Eagle ~]$ #to roll back to a specific date
[shujah@Eagle ~]$ #yum history rollback 17
[shujah@Eagle ~]$ # this will roll back our system
[shujah@Eagle ~]$ #to only delete/rollback specific package
[shujah@Eagle ~]$ #yum history undo 17
[shujah@Eagle ~]$ yum history undo 17
Not root, Subscription Management repositories not updated
Error: This command has to be run with superuser privileges (under the root user on most systems).
[shujah@Eagle ~]$ sudo yum history undo 17
[sudo] password for shujah:
Updating Subscription Management repositories.
This system is registered with an entitlement server, but is not receiving updates. You can use subs
cription-manager to assign subscriptions.
Last metadata expiration check: 0:17:39 ago on Wed 03 Jul 2024 07:10:11 PM EDT.
Dependencies resolved.
Package Architecture Version Repository Size
Removing:
nmap x86_64 3:7.92-1.el9 @rhel-9-for-x86_64-appstream-rpms 24 M
Transaction Summary
Remove 1 Package
Freed space: 24 M
Is this ok [y/N]:


Activities
Terminal
Jul 3 7:28 PM
shujah@Eagle:~
shujah@Eagle:~
shujah@Eagle:~
Dependencies resolved.
Package Architecture Version Repository Size
Removing:
nmap x86_64 3:7.92-1.el9 @rhel-9-for-x86_64-appstream-rpms 24 M
Transaction Summary
Remove 1 Package
Freed space: 24 M
Is this ok [y/N]: y
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Erasing : nmap-3:7.92-1.el9.x86_64 1/1
Running scriptlet: nmap-3:7.92-1.el9.x86_64 1/1
Verifying : nmap-3:7.92-1.el9.x86_64 1/1
Installed products updated.
Removed:
nmap-3:7.92-1.el9.x86_64
Complete!
[shujah@Eagle ~]$



•

--

Installing a package from Source Code

Source code install provides customization. To install a package from its source code follow step below.

1. Download tar package from repo (example: https://archive.apache.org/dist/httpd/httpd-2.4.7.tar.bz2)
2. to download command is wget https://archive.apache.org/dist/httpd/httpd-2.4.7.tar.bz2
3. after downloading go to directory and expand compressed file tar -xzvf httpd-2.4.7.tar.bz2
4. package conatins following files INSTALLATION, Make.
5. read installation instructions less INSTALLATION
6. ./configuration
7. make
8. make install
9. it may ask for additional dependencies while make also gcc compiler should installed in system to make the package.
10. to run package ./package-name -k run

Logical Volume Manager (LVM)

Implementing Logical Volume Manager (LVM) in Red Hat Enterprise Linux (RHEL) involves several steps, from creating new partitions to configuring LVM. Below is a step-by-step guide to help you through the process:

Step 1: Create New Partitions

1.Identify Available Disks**:

lsblk

2. Create a New Partition**:
- Use the `fdisk` tool to create a new partition on the desired disk (e.g., /dev/sdb).

fdisk /dev/sdb
```
- Inside `fdisk`, follow these steps:
  - Press `n` to create a new partition.
  - Select `p` for primary partition or `e` for extended partition.
  - Choose the partition number (usually 1 if it's the first partition on this disk).
  - Press `Enter` to select the default starting sector.
  - Enter the size of the partition (e.g., `+10G` for a 10GB partition).
  - Press `t` to change the partition type.
  - Enter `8e` to set the type to Linux LVM.
- Press `w` to write the changes and exit.

**Step 2: Create Physical Volume (PV)**

1. Create PV on the New Partition**:

***pvcreate /dev/sdb1***

2. Verify PV Creation**:

***pvdisplay***

**Step 3: Create Volume Group (VG)**

1. Create VG Using the PV**:

***vgcreate myvg /dev/sdb1***

2. Verify VG Creation**:

***vgdisplay***

**Step 4: Create Logical Volume (LV)**
```

1. Create LV from the VG**:
   lvcreate -n mylv -L 5G myvg
   -n mylv: specifies the name of the logical volume.
   -L 5G: specifies the size of the logical volume.

2. Verify LV Creation**:

   lvdisplay

Step 5: Create Filesystem on LV

1. Create Filesystem**:

   mkfs.ext4 /dev/myvg/mylv
caution filesystem should be created after creating LVM not before.

Step 6: Mount the Filesystem

1. Create Mount Point**:

   mkdir /mnt/mydata

2. Mount the Filesystem**:

   mount /dev/myvg/mylv /mnt/mydata

3. Verify Mounting**:

   df -h

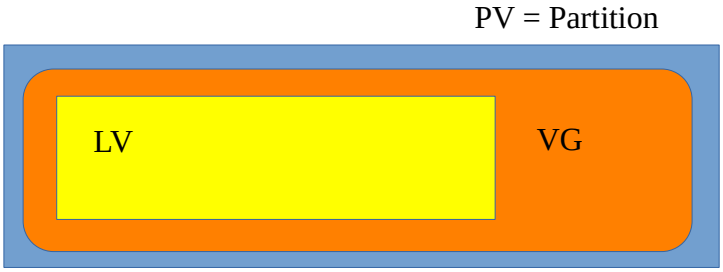
Step 7: Make the Mount Permanent

1. Edit `/etc/fstab`**:

   nano /etc/fstab

2. Add the Following Line**:
  /dev/myvg/mylv /mnt/mydata ext4 defaults 0 2

3. Verify `/etc/fstab` Entry**:
   umount /mnt/mydata
   mount -a
   df -h



STEPS TO REMOVE LVM

- ***umount /mylvm***
- ***lvremove /dev/myvg/mylvm***
- ***vgremove myvg***
- ***pvremove dev/sda1***
-

Resizing a Logical Volume Manager (LVM)

Resizing a Logical Volume Manager (LVM) involves changing the size of logical volumes, which can be increased or decreased as needed. Here are the steps to accomplish both tasks:

Steps to Increase the Size of an LVM

Step 1: Identify the Logical Volume to Resize

1. **List Logical Volumes**:

   lvdisplay

Step 2: Increase the Size of the Logical Volume

1. **Extend the Logical Volume**:

   lvextend -L +5G /dev/myvg/mylv OR lvextend -l +100%free /dev/myvg/mylv

   - `-L +5G` specifies that you are adding 5GB to the existing logical volume.

2. **Verify the Logical Volume Size**:

   lvdisplay

Step 3: Resize the Filesystem

1. **Resize the Filesystem**:
   - For ext4 filesystem:

     resize2fs /dev/myvg/mylv

   - For xfs filesystem:

     xfs_growfs /mnt/mydata

2. **Verify the Filesystem Size**:

   df -h

Steps to Decrease the Size of an LVM

Note: Reducing the size of a logical volume can be risky and may result in data loss if not done correctly. Ensure you have a complete backup before proceeding.

Step 1: Identify the Logical Volume to Resize

1. **List Logical Volumes**:

   lvdisplay

Step 2: Unmount the Filesystem

1. **Unmount the Filesystem**:

   umount /mnt/mydata

Step 3: Check and Reduce the Filesystem

1. **Check the Filesystem**:
- For ext4 filesystem:

   e2fsck -f /dev/myvg/mylv

   - For xfs filesystem:

   xfs_repair /dev/myvg/mylv

2. **Resize the Filesystem**:
- For ext4 filesystem:

   resize2fs /dev/myvg/mylv 5G

   - Reduces the filesystem to 5GB.
- **XFS filesystem cannot be reduced directly. You will need to back up the data, recreate the filesystem with the desired size, and then restore the data.**

Step 4: Reduce the Size of the Logical Volume

1. **Reduce the Logical Volume**:

   lvreduce -L 5G /dev/myvg/mylv

   - Reduces the logical volume to 5GB.

2. **Verify the Logical Volume Size**:

   lvdisplay

Step 5: Remount the Filesystem

1. **Mount the Filesystem**:

mount /dev/myvg/mylv /mnt/mydata

2. **Verify the Filesystem**:

df -h

CRON JOB SCHEDULER

The cron scheduler is a time-based job scheduling system in Unix-like operating systems. It enables users to schedule scripts, commands, or programs to run automatically at specified times and intervals. The `cron` daemon runs in the background and checks the `/etc/crontab` file, the `/etc/cron.d/` directory, and the individual user crontab files located in `/var/spool/cron/crontabs/`.

Syntax:
* * * * * command_to_execute
- - - - -
| | | | |
| | | | +---- Day of the week (0 - 7) (Sunday = 0 or 7)
| | | +------ Month (1 - 12)
| | +-------- Day of the month (1 - 31)
| +---------- Hour (0 - 23)
+------------ Minute (0 – 59)

Run a script every day at 2:30 AM:
30 2 * * * /path/to/script.sh
Run a cleanup script every hour:
0 * * * * /path/to/cleanup.sh
Run a monitoring script every 5 minutes:
*/5 * * * * /path/to/monitor.sh

crontab -e (to Edit current users crontab)
crontab -l (to list entries/jobs in current users crontab)
crontab -r (to remove current users crontab file)

Example : Run backup script at specific time.
- Make shell script file

touch backup.sh
- Write following script and save

vi backup.sh
#!/bin/bash
rsync -avz /opt/fileserver/ /mnt/
echo backup done
- open crontab file for current user

crontab -e
- enter following jobs/entries and save

*27 17 * * * /opt/backup.sh*

Restrict a user to make cron job scheduling:
enter user name in */etc/cron.deny* file
if *cron.allow* exist then it will deny all users by default except root(Part of OS Hardening steps).

System job Scheduling
OS runs schedule tasks; following cron directories exist in */etc*
[root@Eagle opt]# *ls /etc/ | grep -e cron*
anacrontab
cron.d
cron.daily
cron.deny
cron.hourly
cron.monthly
crontab
cron.weekly

USER MANAGEMENT

Imp Commands:
To add user: useradd <username>
To delete user: userdel <username>
To delete user and its home dir also: userdel -r <username>
To set/change user passwd: passwd <username>
To check user id: id <username>
To modify a user: usermod -u
To disable a user in linux: various ways
a) put comments before user entry in /etc /passwd file
b) remove 'x' from second field in passwd file (put it blank or enter '*')
c) create file touch /etc/nologin
d) modify entry in etc/shadow file
To LOCK user from modifying passwd file: passwd -l <usr name> , passwd -u <usr name> to unlock.

to add new group: groupadd <grp name>
change group of a user: usermod -G Eagle shujah (adds shujah to group Eagle as secondary group)
usermod -g Eagle shujah (adds shujah to group Eagle as primary group)

files related to user management:
/ etc/passwd
/ etc/group
/ etc/shadow
/etc /gshadow
/etc /skel

/ etc/login.defs
/ etc/default/useradd

How user add command works at backend:
When new user is created with useradd following changes are made at backend:
1. create a new entry in / etc/passwd file

2. reference / etc/login.defs and / etc/default/useradd files and update user data base in / *etc/passwd, /etc /shadow, /etc /group, /etc /gshadow files.*
3. Create new home directory for user.
4. copies content of /etc /skel directory to the users home directory.
[shujah@Eagle ~]$ cd /etc/skel
[shujah@Eagle skel]$ l.
. .. .bash_logout .bash_profile .bashrc .mozilla

Details about files:

*1. **etc/login.defs***
provides default values for various parameters for new user (useradd)
UID_MIN 1000
UID_MAX 60000
GID_MIN 1000
GID_MAX 60000
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_WARN_AGE 7
CREATE_HOME yes
UMASK 077
ENCRYPT_METHOD SHA512

2. / etc/default/useradd
 # Default values for useradd(8)
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no

3. / etc/passwd
john:x:1001:1001:John Doe:/home/john:/bin/bash

- **Username**: `john`
- **Password Placeholder**: `x` (indicating the encrypted password is in `/etc/shadow`)
- **User ID (UID)**: `1001`
- **Group ID (GID)**: `1001`
- **GECOS**: `John Doe`
- **Home Directory**: `/home/john`
- **Login Shell**: `/bin/bash`

4. */ etc/shadow*
this files contains passwd policies for a user.
shujah:6Bmu5PsWu.IAgC4/7$Hy4ejZHRWIelDkwWoJe4A8nPm/
vakvwIvOk3P6C0ahEugF6f5wX8j4NWK7TAzBXKeX3mEun54z/
D5jvDwmARb1::0:99999:7:::

there are total 9 entries seperated by : details of each field can be found using **man shadow** command.

5. ***/ etc/passwd***

wheel:x:10:root,john

- **Group Name**: `wheel`
- **Password Placeholder**: `x` (indicating the password is stored in `/etc/gshadow`)
- **Group ID (GID)**: `10`
- **Group Members**: `root, john`

Advance Permissions(SUID, SGID, STICKY BIT)

There are few commands like fdisk, mount, password, shed etc which require root privilege to run. Normal user without root privilege cant run these cmds.
Example:
---
ls -l /sbin/fdisk
-rwxr-xr-x. 1 root root 114920 Feb 8 12:57 /sbin/fdisk
fdisk -l
fdisk: cannot open /dev/sda: Permission denied
---
even we have execute permission for the command for others but normal user couldn't execute it. In order to make these command to be executed by normal users we have to set SUID on the command files.
---
chmod u+s /sbin/fdisk
ls -l /sbin/fdisk
-rwsr-xr-x. 1 root root 114920 Feb 8 12:57 /sbin/fdisk
fdisk -l
Disk /dev/sda: 119.24 GiB, 128035676160 bytes, 250069680 sectors
Disk model: SAMSUNG SSD PM85

If SUID is set on a command file then normal user can execute privileged commands.

set SUID for user
chmod u+s /*opt* /abc
-rwSr--r--. 1 root root 0 Jul 5 17:22 /opt/abc

set GUID for user group
chmod g+s /*opt* /abc
-rwSr--Sr--. 1 root root 0 Jul 5 17:22 /opt/abc

set sticky bit for user others
chmod o+t /*opt* /abc
-rwSr--Sr-T-. 1 root root 0 Jul 5 17:22 /opt/abc

Note: if there is execute permission on file then adv permission letter will be small.
After giving execute permission
sudo chmod 777 /opt/abc
ls -l /opt/abc
-rwxrwxrwx. 1 root root 0 Jul 5 17:22 /opt/abc

sudo chmod u+s /opt/abc
sudo chmod g+s /opt/abc
sudo chmod o+t /opt/abc
ls -l /opt/abc
-rwsrwsrwt. 1 root root 0 Jul 5 17:22 /opt/abc

S,T = Adv permission without execute permission.
s ,t = Adv permission with execute permission.

Setting Adv permissions in octal way.
Chmod *755 /*opt*/abc/
* = 4 for SUID
* = 2 for SGID
* = 1 for sticky bit
* = 7 for SUID+SGID+Sticky Bit
* = 6 for SUID+SGID
Example:
sudo chmod 4777 /opt/abc
ls -l /opt/abc
-rwsrwxrwx. 1 root root 0 Jul 5 17:22 /opt/abc

SUDOERS

SUID and Sticky Bits are not practical they have drawback that when it is set on some command file, then every user has privileged permission to execute the command.
Suppose we want only few users to run fdsik -l not all users.
Solution is to add users and command we want to allow in the sudoers file.

- *visudoers*
- *Ali ALL=/sbin/fdisk -l* (add entry in file and save it)
- su Ali
- sudo fdisk -l (note after adding in sudoers we must put sudo before command, so that it will read sudoers file before command execution)
  for granting password less sudo access add below entry in sudoers file.
- *john ALL=(ALL) NOPASSWD: /sbin/fdisk -l*

Note: we can user templates in sudoers

Giving root access to user
Allow user to run any commands anywhere
1. put this entry in sudoers file
<user name> ALL=(ALL) ALL

Who is Root user in linux
A user with uid = 0 is root user no matter its name is what.
[root@Eagle skel]# id
uid=0(root) gid=0(root) groups=0(root)

2. change uid of user to zero
usermod -u 0 <user name>

or
in /etc/passwd file
thetest:x:1001:1002::/home/thetest:/bin/bash
change uid and gid field to ‘0’
thetest:x:0:0::/home/thetest:/bin/bash

$PATH for user.

In Linux, the $PATH variable is an environment variable that specifies a list of directories where the shell looks for executable files. When you type a command in the terminal, the shell searches through these directories in the order they are listed to find the executable file for that command. PATH is user specific.

- echo $PATH

/home/thetest/.local/bin:/home/thetest/bin:/root/.local/bin:/root/bin:/home/shujah/.local/bin:/home/shujah/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin

- which fdisk

/usr/sbin/fdisk

NOTE: only files that exist in the location specified in PATH variable can be executed directly. Otherwise we have to run it with absolute path.
Example: if we remove /usr/sbin entry from PATH, then in order to execute fdisk command we have to enter /usr/sbin/fdisk in terminal.

UMASK

umask is to set default permissions to directories and files.
[thetest@Eagle ~]$ umask
0022 (default umask)

- *to change umask*
  umask 00077
- *to persist umask add in /etc/bashrc*

calculating default permission of a directory
777- umask
777 – 0022 = 755 (drwxr-xr-x)
777 – 0007 = 770 (drwxrwx---)

calculating default permission of a file
666 – umask
666 – 0022 = 644 (rw-r--r--)

KERNEL MANAGEMENT

Linux Kernel: The core component of the Linux operating system, responsible for managing system resources and facilitating communication between hardware and software.
Linux Kernel Drivers: Specialized software modules that enable the kernel to interact with hardware devices, providing a standardized interface for hardware communication.

Types of Linux Kernel Drivers:

1. **Character Drivers**: Manage devices that handle data as a stream of characters, such as keyboards and serial ports.
2. **Block Drivers**: Handle devices that store data in fixed-size blocks, such as hard drives and SSDs.
3. **Network Drivers**: Manage network interface cards (NICs) and handle data transmission over networks.
4. **USB Drivers**: Manage USB devices, such as mice, keyboards, and storage devices.
5. **Graphics Drivers**: Control graphics hardware to render images and handle display functions.

Interaction Between the Linux Kernel and Drivers

- **Loading Drivers**: Drivers can be built into the kernel or loaded as modules at runtime. The kernel detects hardware devices and loads the appropriate drivers during the boot process or when a new device is connected.
- **Syscalls and IOCTLs**: Applications communicate with drivers through system calls (syscalls) and I/O control operations (IOCTLs), which the kernel forwards to the appropriate drivers.
- **Interrupts**: Hardware devices use interrupts to signal the kernel about events, such as data arrival. Drivers handle these interrupts and perform the necessary actions.
- **Direct Memory Access (DMA)**: Some drivers use DMA to transfer data between the device and memory without involving the CPU, improving efficiency.

Where is Linux Kernel?

In the /boot directory
boot contains following files
config-5.14.0-427.22.1.el9_4.x86_64 >> Kernel config file
efi
grub2
initramfs-5.14.0-427.22.1.el9_4.x86_64.img
initramfs-5.14.0-427.22.1.el9_4.x86_64kdump.img
loader
symvers-5.14.0-427.22.1.el9_4.x86_64.gz
System.map-5.14.0-427.22.1.el9_4.x86_64
vmlinuz-5.14.0-427.22.1.el9_4.x86_64 >> Kernel

Where are Kernel Drivers in linux?

[shujah@Eagle /]$ *ls /lib/modules/5.14.0-427.22.1.el9_4.x86_64/kernel*
>>**arch block crypto drivers fs kernel lib mm net samples sound virt**
[shujah@Eagle /]$ ls /lib/modules/5.14.0-427.22.1.el9_4.x86_64/kernel/fs/ext4
>> ext4.ko.xz
kernel modules/drivers also called as klm or lkm are stored as kernel object file with .ko extension.
[shujah@Eagle kernel]$ *ls -R | grep -i .ko | wc -l*
2246 (total no of kernel drivers)

Driver Management commands

insmod, modprobe, rmmode, depmode, modinfo, lsmode

- To check drivers loaded in RAM

  [shujah@Eagle kernel]$ lsmod

Module	Size	Used by
binfmt_misc	28672	1
tls	159744	0

- how to remove driver loaded in RAM

  sudo rmmod <driver name>

- Permanently remove driver so that it wont loaded into RAM after reboot

  rm -rf <path of driver file>

- how to load driver in RAM

  sudo modprobe <driver name>

- for information about driver

  modinfo ext4

Installing new Drivers

Step 1: Identify the Driver

First, you need to identify the specific driver required for your hardware. You can typically find this information from the hardware manufacturer's website or documentation.

lscpi -vvv | grep audio 'or' dmesg | grep -i audio (to check the chipset of driver)

Step 2: Update the System

Ensure your system is up to date:

```
sudo yum update -y
```

Step 3: Install Development Tools

Install necessary development tools and kernel headers if you need to compile the driver from source:

```
sudo yum groupinstall "Development Tools" -y
sudo yum install kernel-devel kernel-headers -y
```

Step 4: Download the Driver

Download the driver package from the hardware manufacturer's website. This could be a tarball (`.tar.gz`), a zip file, or an RPM package.

Example for a tarball:

```
wget http://example.com/driver.tar.gz
tar -xzf driver.tar.gz
cd driver-directory
```

Step 5: Install the Driver

If the driver is provided as an RPM package, you can install it using:

```
sudo rpm -ivh driver.rpm
```

If you need to compile and install the driver from source, follow these steps:

```
./configure
make
sudo make install
```

Step 6: Load the Driver

After installing the driver, load it into the kernel using `modprobe`:

```
sudo modprobe driver_name
```

Step 7: Verify the Installation

Check if the driver is loaded:

```
lsmod | grep driver_name
```

Also, verify that the hardware is recognized:

```
dmesg | grep -i driver_name
```

Step 8: Persistent Loading (Optional)

To ensure the driver is loaded at boot time, you can add it to `/etc/modules-load.d/`:

```
echo driver_name | sudo tee /etc/modules-load.d/driver_name.conf
```

Blacklisting Drivers

Step 1: Create a configuration file in the `/etc/modprobe.d/` directory to blacklist the driver. For example, to blacklist the `r8169` driver:

```
echo "blacklist r8169" | sudo tee /etc/modprobe.d/blacklist-r8169.conf
```

Step 2: Update the Initial RAM Filesystem

After creating the blacklist configuration, update the initial RAM filesystem so that the changes take effect on the next boot.

```
sudo dracut --force
```

Step 3: Reboot the System

```
sudo reboot
```

Updating Kernel:

yum update kernel or

rpm -U <kernel name> or

from source code

When we update kernel it will it will install new kernel in /boot, so a new vmlinuz, initramfs and config file created in addition to old files.

When we update a package then old files are overwritten with newones.

Checking Running kernel:

uname -r

5.14.0-427.22.1.el9_4.x86_64

/proc vs Kernel

/proc contains kernel modules loaded in RAM, all files in /proc are loaded in RAM and running not in the Harddisk. /proc provides a window into the kernel's inner workings. It allows users and applications to read system and process information and modify kernel parameters.

Virtual Filesystem: `/proc` is created by the kernel in memory and does not occupy disk space. Its contents are generated dynamically by the kernel.

System Information Access: `/proc` provides a way to access detailed information about the system's state, including CPU, memory, and process details.

Kernel Interaction: Users and applications can read from and write to certain files in `/proc` to retrieve information and modify kernel parameters at runtime.

Monolithic Kernel / modular: Used by Linux, Drivers can be loaded in realtime.

Micro Kernel: used by Solaris AIX HP, Drivers can NOT be loaded in realtime.

Kernel Tuning

Problem: Error too many files opened by application; solve it by kernel tuning.

checking which files are opened by a Process in RAM.

lsof

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
systemd 1 root mem REG 253,0 44784 4156 /usr/lib64/libffi.so.8.1.0

lsof | wc -l
99726 >> total no of files opened by all loaded processes.

cat /proc/sys/fs/file-max
600000 >> max no of files that can be loaded in RAM are

increasing max no of files that could be opened by loaded processes
echo 900000 > /proc/sys/fs/file-max (not cant do vi bc files are loaded)
[root@Eagle boot]# *cat /proc/sys/fs/file-max*
900000

Tools for Kernel Tuning

1. **sysctl**:
   - A command-line utility used to modify kernel parameters at runtime.
   - Example: `sudo sysctl -w net.ipv4.ip_forward=1`
2. **/proc/sys**:
   - The directory where many kernel parameters can be read and modified.
   - Example: `echo 1 > /proc/sys/net/ipv4/ip_forward`
3. **/etc/sysctl.conf**:
   - A configuration file for setting kernel parameters that should be applied at boot time.
   - Example entry: `net.ipv4.ip_forward = 1`
4. **/etc/sysctl.d/**:
   - Directory containing configuration files for setting kernel parameters. Files in this directory are processed by `sysctl` at boot time

003. RHEL INTENSIVE SERVICES:

PART 1: AWS CLOUD COMPUTING

What is AWS?
AWS is a cloud service provider that enables you to:
Compute: Run virtual servers and containers.
Storage: Store files and data.
Databases: Use managed database services.
Networking: Set up and manage networks.
Analytics: Process data and generate insights.
Machine Learning: Build and train machine learning models.
Security: Secure your infrastructure and applications.

Key AWS Services

- **Compute Services**
  - EC2 (Elastic Compute Cloud): Provides scalable virtual servers.
  - Lambda: Run code without provisioning servers (serverless).
  - ECS (Elastic Container Service): Run and manage Docker containers.
- **Storage Services**
  - s3 Bucket: Like google drive, drop box, one drive, Object storage for any type of data.
  - EBS: ELastic Block Storage, Block storage Volume Attached with ec2 instance.
  - Glacier: Low-cost storage for data archiving and backup.
- **Database Services**
  - RDS (Relational Database Service): Managed relational database service.
  - DynamoDB: Managed NoSQL database service.
  - Aurora: High-performance, scalable relational database service.
- **Networking Services**
  - VPC (Virtual Private Cloud): Isolated network for AWS resources.
  - Route 53: Scalable DNS and domain name registration.
  - CloudFront: Content delivery network (CDN).
- **Management and Monitoring**
  - CloudWatch: Monitoring for AWS resources and applications.
  - CloudTrail: Track user activity and API usage.

AMI: Preconfigured EC2 instances with (operating system, application server, and applications).
Elastic IP: A Public IP that is persistent and doesnt change after reboot. In production systems Elastic IP is used with EC2 Instances.
AZ: Availability ZOnes

EBS(elastic block storage)
1. EBS Volumes

2. EBS Snapshots

EBS Volumes

General Purpose		Provisiones IOPS		HDD VOLUME(SATA)		Previous Generation
ssd		ssd		SATA		HDD
gp3	gp2	iO02	io01	throughput optimized	cold	Standard
16000 IOPS		256000	64000iops	500iops	250iops	40 -200

PART 2: SERVICES

Setting up Client Server:

Server side:
1. set host name: *hostnamectl set-hostname server1.example.com –static*
2. check: *cat etc/hostname*
3. details of host: *hostnamectl*
4. Setup PING:
- ensure that ICMP inbound or outbound is allowed by firewall.
- Setup DNS to ping by the name instead of IP.
- FLAT DNS: *etc/hosts* file act as flat DNS, whenever any internet service like ping, ssh, ftp is run then it will first check etc/hosts file if domain name is found there its ok, if not then it goes towards DNS server for name resolution.
- *vi etc/hosts* Add following entry *<IP of server> server1.example.com*
- to ping client by its domain name, also add entry for client in *etc/*hosts/ file *<IP of client> client1.example.com*
- ping client1.example.com

Client side:
1. set host name: *hostnamectl set-hostname client1.example.com –static*

2. check: *cat etc/hostname*
3. Setup PING: (same steps as for server, ensure to add IP of client
   <IP of client> client1.example.com.
4. ping server1.example.com.

General steps to start any service:

1. check whether service is already installed or not
   systemctl list-unit-files (list of all installed services with status)
   systemctl list-unit-files | grep -i <servicename>
   or rpm -qa | grep -i <servicename>
2. install service:
   yum search <service name>
   yum info <service name>
   yum install -y <service name>
3. Start and enable the service
   sudo systemctl start <service name>
   sudo systemctl enable <service name>
   sudo systemctl is-active <service name>
   sudo systemctl is-enabled <service name>
   or sudo systemctl status<service name>

Alternative way to start service:

Every service has its binary in the /usr/sbin directory. Whenever we start service its binary get loads into RAM from */usr/sbin/servicebinary.*

Alternatively we can start service by executing its binary.

cd /usr/sbin

./servicebinary example ./vsftpd

4. check from server and client whether relevent tcp port is opened .
   sudo ss -tulnp | grep -i <port no>
   sudo telnet <server ip> <port no> (check server port status from client machine)
5. Make necessary changes in configuration files of service
   rpm -qc <service name> (this will list configuration files of service)
   rpm -qd <service name> (this will list documentation related to service)
6. If required add service to firewall rules.

NFS:

port = 2049
protocol = UDP
Purpose: File Sharing bw Linux to Linux only.
doesnt ask for user name & passwd.
Cmd to check for whether NFS port is opened:
netstat -tulnp | grep 2049
ss -tulnp | grep 2049

Samba:

port = 137/139/335
protocol = TCP
Purpose: File Sharing bw Linux to Linux or linux to windows

ask for user name and passwd

FTP:

Port: 21 for connection, 20 for data transfer
services: wuftpd, vsftpd, sftp, proftp
Purpose:

- File Transfer and download
- Sharing and distributing files between computers and servers
- Uploading website content to web servers
- Backing up and archiving data
- Downloading software, updates, and other files from remote repositories

NFS VS FTP the difference:
In the NFS, server shares files and client access those files utilizing server resourses (if 10 GB of data is shared using NFS then space is allocated on server not on client system). if server fails then we cant access data.
On the other hand in FTP data is accessed and downloaded on client computers, its advantage is that downloaded can be accessed even after server crashed. Resources of client computer are used as compared with NFS.

Here are some of the most important TCP port numbers:

- 1. **HTTP (Web)**: Port 80
- 2. **HTTPS (Secure Web)**: Port 443
- 3. **FTP (File Transfer Protocol)**: Port 21
- 4. **SSH (Secure Shell)**: Port 22
- 5. **SMTP (Simple Mail Transfer Protocol)**: Port 25
- 6. **POP3 (Post Office Protocol v3)**: Port 110
- 7. **IMAP (Internet Message Access Protocol)**: Port 143
- 8. **RDP (Remote Desktop Protocol)**: Port 3389
- 9. **DNS (Domain Name System)**: Port 53
- 10. **DHCP (Dynamic Host Configuration Protocol)**: Port 67 (server) and Port 68 (client)
- 12. **SQL (Structured Query Language)**: Port 1433 (Microsoft SQL Server), Port 3306 (MySQL)
- 13. **VNC (Virtual Network Computing)**: Port 5900
- 14. **NTP (Network Time Protocol)**: Port 123
- 15. **SNMP (Simple Network Management Protocol)**: Port 161

16. **Telnet**: Port 23

NFS Service for file sharing bw Client Server

Server Side:

1. In Server Machine Make Directory for sharing with clients:

    mkdir /opt/fileserver/

2. Install and Activate nfs service in server

    rpm -qa | grep -i nfs

if nfs utils is not installed then;

yum install nfs-utils -y
systemctl start rpcbind
systemctl start nfs-server.service
systemctl enable rpcbind (to run after restart)
systemctl enable nfs-server.service

To confirm for whether port 2049 default port for NFS service is opened type following cmd

netstat -tulnp | grep 2049
ss -tulnp | grep 2049

3. Configure NFS exports:

Add directories you want to share over NFS, along with client access permisssion.

sudo vi /etc/exports

add entry and save: */opt/fileserver *(rw,sync,no_root_squash)*

This line will share the `/shared/folder` directory with all clients, allowing read-write access, synchronous operation, and no root squashing.

Reload NFS exports: *sudo exportfs -ra*

Check export is loaded: *exportfs -rv , exportfs -v*

4. Enable nfs service in the firewall configuration:

sudo firewall-config

open the required firewall ports

sudo firewall-cmd --permanent --add-service=nfs
sudo firewall-cmd --permanent --add-service=mountd
sudo firewall-cmd --permanent --add-service=rpc-bind
sudo firewall-cmd –reload

Client Side:

1. Start Services

systemctl start rpcbind
systemctl start nfs-client.service
systemctl enable rpcbind (to run after restart)
systemctl enable nfs-client.service

2. On the client systems, mount the shared NFS directory:

sudo mount -t nfs <nfs-server-ip>:/opt/fileserver /mnt

for persistent mount add entry in /etc/fstab:

<nfs-server-ip>:/opt/fileserver /mnt

reload fstab entry cmd: *mount -a*

* fileserver = shared folder by server */opt/fileserver/*
* /mnt = local mount point on client
* *<nfs-server-ip> = server1.example.com or IP address*

3. Verify the mount

ls /local/mount/point

FTP Service for file transfer bw Client Server

1. Install ftp package:

rpm -qa | grep -i ftp
yum install vsftpd -y

2. Start and enable ftp service:
sudo systemctl start vsftpd
sudo systemctl enable vsftpd
sudo systemctl is-active vsftpd
sudo systemctl is-enabled vsftpd
sudo ss -tulnp | grep -i 21 (to check whether relevent tcp port is opened)

3. In server Create separate folder for ftp and add ftp users
sudo mkdir /var/ftp/
sudo useradd -d /var/ftp/ftpuser1 -s /sbin/nologin username

4. Connect from client using ftp
Accessing from ftp client
- Filezilla is one of many ftp client apps we can use it to connect to server
- or we can use terminal based ftp

ftp://<server-IP or host-name>
Note: default path for ftp is /var/ftp/ . So client will access this path of server.
Rectifications during the process:
- disable Firewall in the server machine or add rule to allow ftp
- You may need to stop selinux in some cases.
- Edit ftp configuration
  check configuration files: *rpm -qc vsftpd*
  >> */etc/vsftpd/vsftpd.conf* (configuration file)
  there are 3 users for ftp

a) anonymous --no identification (public archives and websites use this to downloads files)
b) local user --/etc/passwd
c) Virtual User – not in *etc/*passwd/

vsftpd.conf Important Tags
anonymous_enable=YES/NO
local_enable=YES/NO
write_enable=YES/NO
chroot_local_enable=YES/NO (When Enabled users home directory becomes root directory. If not enabled than user can traverse back to Root dir and can see whole OS. So this tag is v imp)

Difference bw Active and Passive FTP:
Key differences:
1.Connection Initiation:
- Active FTP: Server initiates the data connection.
- Passive FTP: Client initiates the data connection.

2.Firewall Compatibility:
- Active FTP: Requires the client to accept incoming connections, which can be problematic if the client is behind a firewall.
- Passive FTP: Requires the client to only initiate outbound connections, which is generally easier to configure through firewalls.

3.Port Usage:

- Active FTP: Uses port 21 for the control connection and port 20 for the data connection.
- Passive FTP: Uses port 21 for both the control and data connections, with the server providing the client with a random port number for the data connection.

Apache Webserver

1. Install Apache package:
for Rhel package for Apache is httpd
for debian bases sys apache2
yum install httpd -y

2. Start and enable ftp service:
sudo systemctl start httpd
sudo systemctl enable httpd

3. Create index.html file
Home directory of httpd is */etc/*httpd/
ls /etc/httpd
>> conf conf.d conf.modules.d logs modules run state
configuartion file is: httpd.conf
ls /etc/httpd/conf
httpd.conf magic
default path for webpage is:
/var/www/html
cd *var*/www/html
touch index.html

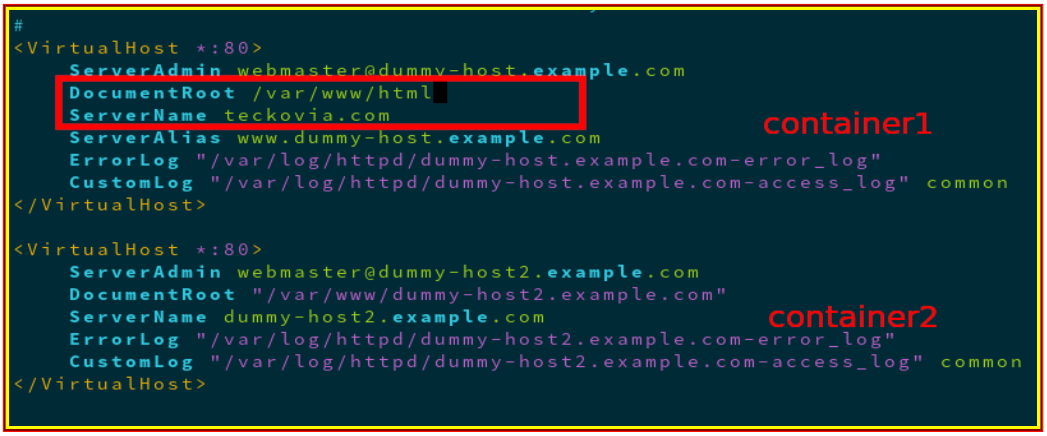
4. Configuration settings:
Main configuration file httpd.conf
Child configuration files inside conf.d directory
ls /etc/httpd (here conf dir contains main config file conf.modules.d contains child config files)
>>conf conf.modules.d modules state
>>conf.d logs run
Main Apache HTTP server configuration file is:
/etc/httpd/conf/httpd.conf
Child config files are:
ls /etc/httpd/conf.modules.d
>>00-base.conf 00-lua.conf 00-proxy.conf 10-h2.conf
>>00-brotli.conf 00-mpm.conf 00-systemd.conf 10-proxy_h2.conf
>>00-dav.conf 00-optional.conf 01-cgi.conf README
The main configuration file httpd.conf has an entry *Include conf.modules.d/*.conf*
which includes child configuration files autoindex.conf userdir.conf welcome.conf at start.

Virtual Host Configuration:

copy sample configuration file.

cp /usr/share/doc/httpd-core/httpd-vhosts.conf /etc/httpd/conf.d/

Edit File: *sudo vi httpd-vhosts.conf*



To test httpd configuration file: *httpd -t*

After that restart and enable httpd service.

Put host name in etc/hosts

vi /*etc*/hosts/

<your ip address> teckovia.com

after that open webbrowser and type teckovia.com

A webpage i.e. index.html file in /var/www/html will open.

ls /var/www/html

>>index.html

Hosting Multiple websites on one IP:

Edit File: *sudo vi httpd-vhosts.conf*

Add new virtual host.

Add entry to etc/hosts and restart httpd services

Security container: Added with each host to grant access.

```
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot /var/www/html
    ServerName teckovia.com
    ServerAlias www.teckovia.com
    ErrorLog "/var/log/httpd/dummy-host.example.com-error_log"
    CustomLog "/var/log/httpd/dummy-host.example.com-access_log" common
</VirtualHost>

<Directory /var/www/html>
Require all granted
</Directory>
```

Security Conatiners

```
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host2.example.com
    DocumentRoot /opt/shujahweb
    ServerName shujah.com
    ServerAlias www.shujah.com
    ErrorLog "/var/log/httpd/dummy-host2.example.com-error_log"
    CustomLog "/var/log/httpd/dummy-host2.example.com-access_log" common
</VirtualHost>

<Directory /opt/shujahweb>
Require all granted
</Directory>
```

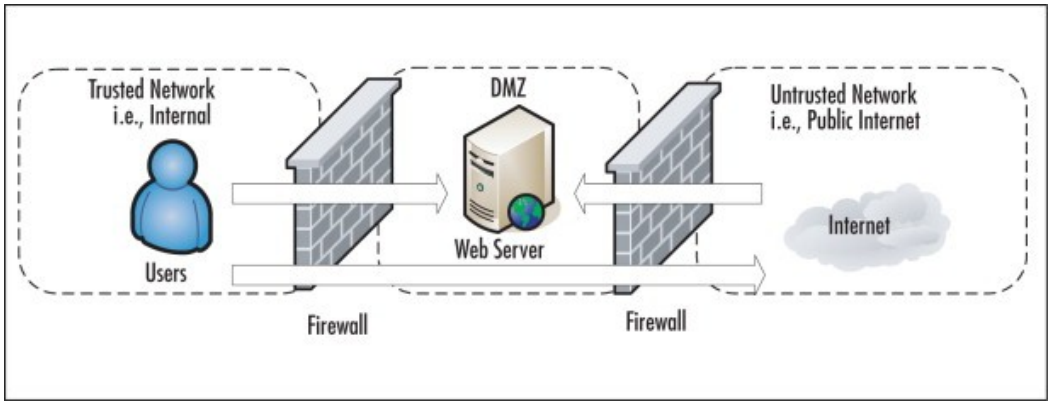
FireWalls:

Types:

1. Hardware based Firewalls

2. Software based firewalls

a. Application level Firewalls >> TCP Wrapper
b. Kernel level Firewalls >>IP tables (Rhel 6), Firewalld (Rhel 8, 9)
c. Firewalls softwares like pfsense.



TCP Wrappers:

Is an app firewall.

sudo dnf install tcp_wrappers

`/etc/hosts.allow`

add entry sshd: 192.168.1.0/24 (allow sshd from given ip addrs)

`/etc/hosts.deny`
add entry vsftpd: 192.168.1.0/24 (block access of vdftpd to given ip addrs)

Applicaton firewall isnt reliable bc external IP can access to server even if its is blocked by hosts.deny

IP Tables: (rhel 6)

By default there is ACCEPT policy for all traffic:
[root@server1 shujahweb]# iptables -L
Chain INPUT *(policy ACCEPT)*
target prot opt source destination

Chain FORWARD *(policy ACCEPT)*
target prot opt source destination

Chain OUTPUT *(policy ACCEPT)*
target prot opt source destination

Changing default policy Block all traffic INBOUND and OUTBOUND:
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
[root@server1 shujahweb]# iptables -L
Chain INPUT *(policy DROP)*
target prot opt source destination

Chain FORWARD *(policy DROP)*
target prot opt source destination

Chain OUTPUT *(policy DROP)*
target prot opt source destination
Now as we have blocked all incomming and outgoing traffic, our configuration is most secure.

To open Specific port for external IP address:
iptables -A INPUT -s 0/0 -p tcp --dport 80 -j ACCEPT

	target	**prot**	**opt**	**source**	**destination**	
iptables -L:	ACCEPT	tcp	--	anywhere	anywhere	tcp dpt:http

iptables -A INPUT -s 192.168.1.39/32 -p tcp --dport 22 -j ACCEPT

iptables -L:	ACCEPT	tcp	--	192.168.1.39	anywhere	tcp dpt:ssh

Note: IN order to make tcp connection to work properly we have to allow that connection at OUTPUT also (3 way handshake).
iptables -A OUTPUT -d 0/0 -m state --state ESTABLISHED -j ACCEPT
All Established incoming connections are allowed to go out.

There are 4 types of tables in iptables 1. Filter table 2. raw table 3. NAT table 4.mangle table.

Our policies are written in filter table by default.
NAT Table: Converts private IP to Public IP.
PAT: port address translation.

Firewalld (rhel 7 and above):

How to open port in firewalld

firewall-cmd –list-services

>>cockpit dhcpv6-client ssh

allow http service:

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

firewall-cmd --reload (must reload after adding service)

>>cockpit dhcpv6-client http ssh

There are various **Zones** in firewalld

firewall-cmd –list-all-zones

changing default zone to Block will block all traffic:

firewall-cmd –set-default-zone=block

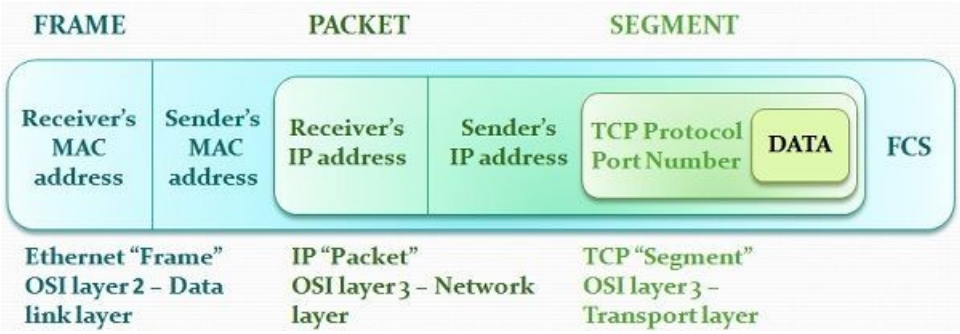
Security groups SG:

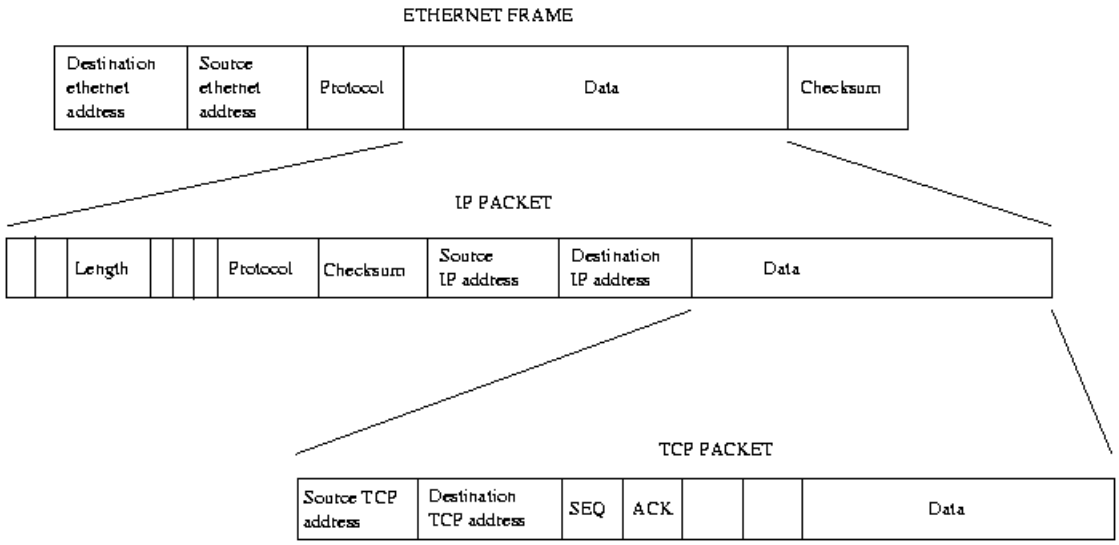
In AWS firewall is SG

Statefull vs stateless firewall

In statefull firewall outgoing connection is just enabled for each eastablished port, np need to enable output port for each connection.

IP Packet vs Frame





SE Linux(security enhanced linux);

Provides OS level security to services and files

Modes

1. Disabled
2. Enabled
    enforcing (full blocked)
    permissive (shows warning message but allow to enter)

To change from disbale to enable or vice versa Reboot is required

sestatus (to check whether SE is enabled or disabled)

setenforce 0 (sets to permissive mode0

setenforce 1 (sets SE to enforcing mod)

getenforce (shows state permissive or enfocing)

To Disable SE linux;

sudo vi /etc/selinux/config

change entry SELINUX=disabled

reboot

Working of SE linux:

Boolean value 0/1: is set to apply SE on services like ftp, samba, nfs.

Context value: is used to set context on Directory and files for additional security.

To check bool values set by SE linux on ftp service for example:

getsebool -a | grep -i ftp

Now there is configuration file for vftpd vsftpd.conf
let say we have enbaled anonymous write in vfdtpd.conf even then it is disbaled by SE linux.
So there are two layers of security configuration one is at configuration of service and other is at SE linux which is at OS level and its priority is higher than former.

Change anonymous write to on
setsebool -P ftpd_anon_write on
Setting context values on files and dir:

DNS (Domain Name Service):

steps to implement caching recursive dns server

install DNS service
yum install bind-* -y
Default home directory for DNS service is
ls /var/named/chroot
dev etc proc run usr var
check status of service and start:
ps -el | grep -i named or
systemctl status named-chroot
systemctl start named-chroot
systemctl enable named-chroot
check for port open
netstat -tulnp | grep -i 53
add service to firewalld
firewall-cmd --permanent –add-service=dns
Rootserver>>TLD(.com.org)>>Domain>>subdomain
portal.teckovia.com.

Now login to client machine and add ip address of domain server in *resolv.conf* file.
dig@<dns server ip>

SSH(Secure Shell):

- SSH is secure remore login protocol
- SSH uses encryption for remote session (public + private key)
- On the other hand telnet is insecure protocol as data transfer is in the form of plain text.
- SSH uses port 22 for tcp connection.

Steps to setup SSH service:
usuually SSH comes as default package with linux OS
systemctl status sshd
systemctl start sshd
systemctl enable sshd
telnet <server ip > 22 or netstat -tulnp | grep -i 22
Configuration of SSH:
SSH has its home directory */etc/ssh/*
configuration file for ssh is *sshd_config*

We can change default port number, configuare password less login etc through this file.
After changing *sshd_config* service needed to be restarted.

How to login from client through ssh:
first sshd should be active in server
then go to client and type ssh < server ip address>
then give user name ad password and you will be given remote terminal access of server machine.

Xinetd:

This is a parent service and has child services under it (telnet etc).
Why xinetd?
NFS, SSH etc are standalone services but there are some services which are small and when started individually are loaded to RAM and occupy space. Now with xinetd it contains services under it and those services are only loaded when client invokes/call them.

SAMBA (File sharing):

- Samba is used for file sharing bw Windows to Linux and Linux to Linux.
- Samba requires user name and password while NFS don't require it.

1) Setting up Samba server

rpm -qa | grep -i samba
yum install samba- -y*
systemctl start smb
systemctl start nmb
systemctl enable smb
netstat -tulnp | grep -i 137 netstat -tulnp | grep -i 139

2) Add smb user

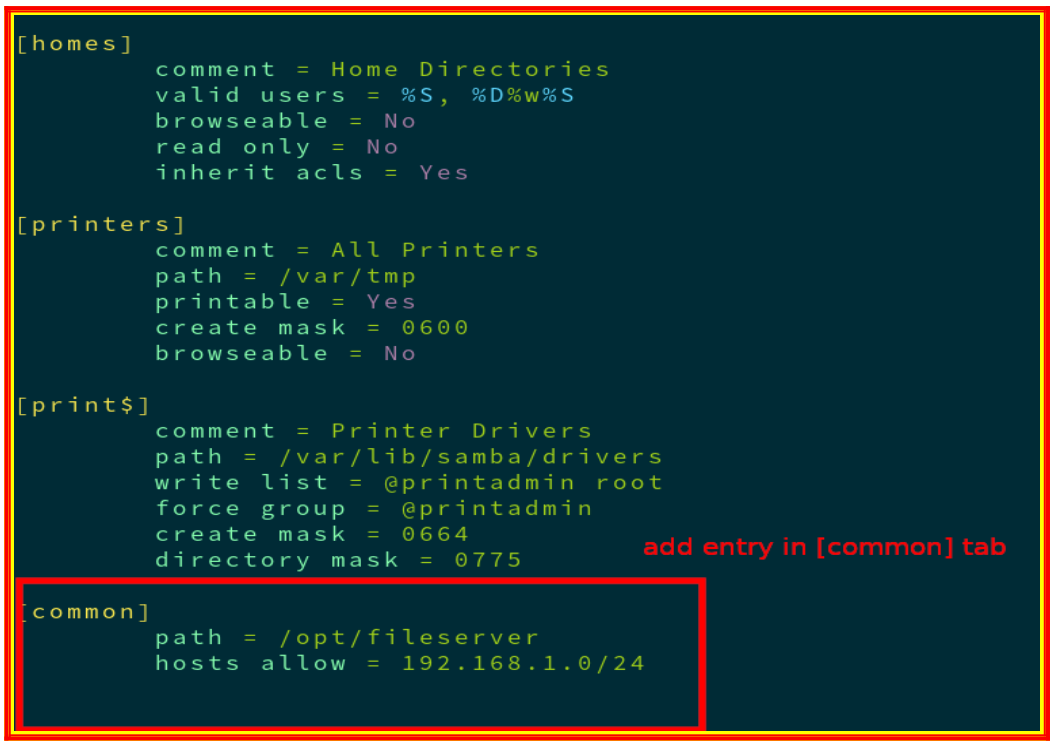
Normal users added with useradd command cant use smb, for smb we have to add new or existing user.

Smb passwd -a ali

3) configuration file of samba

ls */etc/*samba/
lmhosts smb.conf smb.conf.example usershares.conf

vi smb.conf
add folder to be shared *etc*/fileserver/ and client IP in [common] tab.



After changing conf file type **testparm** to load changes to smb service and then restart smb.
Client side:
check for open ports from client machine
telnet <server ip> 137 (samba works on tcp port 137 and 139)
if port is not connecting from client then do following:
1. check connectivity bw server and client
2. add smb service to firewall rule if not added
firewall-cmd –list-services
 >>cockpit dhcpv6-client ssh
allow smb service:
firewall-cmd --permanent --add-service=smb
firewall-cmd --reload (must reload after adding service)

after setting up smb service go to the client machine i.e. windows type server ip \\192.168.2.10 (use forward slash bot backslash) in windows search bar, you will asked to enter user name and passwd. Type smb user name Ali in this example and his passwd and you will get access to shared folder by server machine.

smb.conf tags

The `smb.conf` file is the main configuration file for the Samba suite, which allows file and print sharing between Unix/Linux and Windows systems. The file contains various sections and parameters (tags) that control the behavior of Samba services. Below is a breakdown of common sections and key parameters in `smb.conf` along with their descriptions.

1.Global Section
The `[global]` section contains settings that apply to the overall Samba server and affect all shares.

- **workgroup**: Specifies the Windows workgroup or domain the Samba server belongs to.
  workgroup = WORKGROUP

2. Share Definitions
These sections define the shared resources (e.g., directories or printers) available to clients.

- **[homes]**: A special section that makes each user's home directory available as a share. Each user will see only their own home directory.
  [homes]
  comment = Home Directories
  browseable = yes
  writable = yes

- **[printers]**: A section that allows clients to print to printers connected to the Samba server.
  [printers]
  comment = All Printers
  path = /var/spool/samba
  browseable = no

guest ok = no
writable = no
printable = yes

- **[sharename]**: A custom share definition. Replace `sharename` with your desired share name.
[sharename]
path = /path/to/directory
browseable = yes
writable = yes
valid users = user1 user2

Key parameters within a share definition include:

- **path**: The path to the directory being shared.
path = /path/to/directory
- **browseable**: Determines whether the share is visible in the network neighborhood. `yes` means it is visible.
browseable = yes
-**read only**: Controls whether the share is read-only. `no` means the share is writable.
read only = no
- **guest ok**: Allows guest access to the share. `yes` means no username/password is required.
guest ok = yes
- **writable**: Alias for `read only = no`. Indicates that the share is writable.
writable = yes
- **valid users**: Specifies which users or groups are allowed to access the share.
valid users = user1 user2
- **write list**: Defines users or groups that have write access, even if the share is marked as read-only.
```ini
write list = user1, @group
- **create mask**: Defines the permissions for newly created files. The value is in octal.
create mask = 0644
- **directory mask**: Defines the permissions for newly created directories. The value is in octal.
directory mask = 0755

3. **Advanced Parameters**
These are additional parameters for more specific configurations.

- **force user**: Forces all file operations to be performed as a specific user, regardless of who is actually logged in.
force user = someuser
- **force group**: Similar to `force user`, but forces all file operations to use a specific group.
force group = somegroup
- **available**: Controls whether a share is available. If set to `no`, the share is not accessible.
available = yes
```

How samba works:
Samba is a free software suite that enables file and print sharing between Unix/Linux systems and Windows systems. It implements the Server Message Block (SMB) protocol (also known as Common Internet File System or CIFS), which is the standard protocol used by Windows for file and print services. Samba allows Unix/Linux servers to communicate with Windows clients as if they were native Windows servers.

How Samba Works: Key Components and Concepts

1. **SMB/CIFS Protocol**
- **SMB (Server Message Block)**: A network file sharing protocol that allows applications on a computer to read and write to files and request services from server programs in a network.
- **CIFS (Common Internet File System)**: An extension of the SMB protocol, commonly used by Windows operating systems.

Samba acts as a server that implements SMB/CIFS, allowing Unix/Linux machines to share files and printers with Windows machines.

2. **Samba Daemons**
Samba relies on several background processes (daemons) to handle different aspects of its operation:

- **smbd**: The main daemon that provides file and print services to SMB/CIFS clients. It handles authentication, file sharing, and printing. When a Windows client connects to a Samba server, it communicates with `smbd` to access shared resources.
- **nmbd**: Responsible for NetBIOS name resolution and browsing. It allows Samba to participate in the Windows Network Neighborhood and acts as a WINS (Windows Internet Name Service) server. This daemon makes the Samba server visible to Windows clients.
- **winbindd**: Integrates Unix/Linux systems with Windows Active Directory by providing a way to use Windows accounts and groups on Unix/Linux systems. It allows domain users to authenticate on Unix/Linux systems.

3. **Authentication and Access Control**
Samba provides multiple ways to authenticate users and control access to shared resources:

- **User-Level Security (security = user)**: The most common mode, where each user must provide a valid username and password to access shared resources. Samba verifies the credentials against its own database, a Unix/Linux user database, or an external domain controller.
- **Share-Level Security (security = share)**: A deprecated mode where access is controlled per share, without requiring a username. Each share can have its own password.
- **Domain Security (security = domain)**: Samba acts as a member of a Windows domain, delegating authentication to a domain controller.
- **Active Directory Integration**: Samba can join an Active Directory domain and authenticate users against it.

4. **File Sharing**

Samba allows Unix/Linux directories to be shared with Windows clients. These shared directories can be accessed by users over the network as if they were local folders on the Windows machine. The configuration of these shares is done in the `smb.conf` file.

- **Share Definitions**: In `smb.conf`, each shared directory is defined in its own section. Parameters like `path`, `browseable`, `writable`, and `valid users` control the behavior and access permissions of each share.

5. **Printing Services**
Samba can also share printers connected to a Unix/Linux system with Windows clients. Windows users can send print jobs to a Samba-shared printer just like they would to a printer connected to a Windows server.

6. **Name Resolution and Browsing**
- **NetBIOS**: Samba uses NetBIOS for name resolution, allowing Windows clients to find the Samba server by name on the network. The `nmbd` daemon handles this by broadcasting the server's NetBIOS name.
- **WINS**: Samba can also act as a WINS server, which helps in resolving NetBIOS names across subnets.

7. **Domain and Active Directory Integration**
- **Domain Controller**: Samba can act as a Primary Domain Controller (PDC) or a Backup Domain Controller (BDC), managing authentication and resources for a domain.
- **Active Directory Member**: Samba can join a Windows Active Directory domain, allowing users to log in with their domain credentials and access shared resources on the Samba server.

8. **Permissions and Security**
- **File Permissions**: Samba respects Unix/Linux file permissions, and you can also set Samba-specific permissions using parameters like `create mask` and `directory mask`.
- **Access Control Lists (ACLs)**: Samba can use ACLs for more granular permission management, allowing you to control access to files and directories at a more detailed level.
- **Encryption**: Samba supports SMB3, which includes encryption for secure communication between clients and the server.

How a Typical Samba Interaction Works

1. **Discovery**: A Windows client looks for shared resources on the network, either by browsing the Network Neighborhood or connecting directly to a known server.

2. **Connection**: The client connects to the Samba server using SMB/CIFS, providing a username and password if required.

3. **Authentication**: `smbd` authenticates the user using the specified security mode. If the credentials are valid, the client is granted access to the specified shares.

4. **File/Printer Access**: The client can now read, write, and manipulate files in the shared directories or send print jobs to shared printers.

5. **Session Maintenance**: The connection remains open for as long as the client needs access. The session is maintained by `smbd`, which handles all requests from the client.

6. **Disconnection**: When the client no longer needs access, it disconnects from the Samba server, closing the session.

The End

Connect with me on Linkdein-ShujahUllah