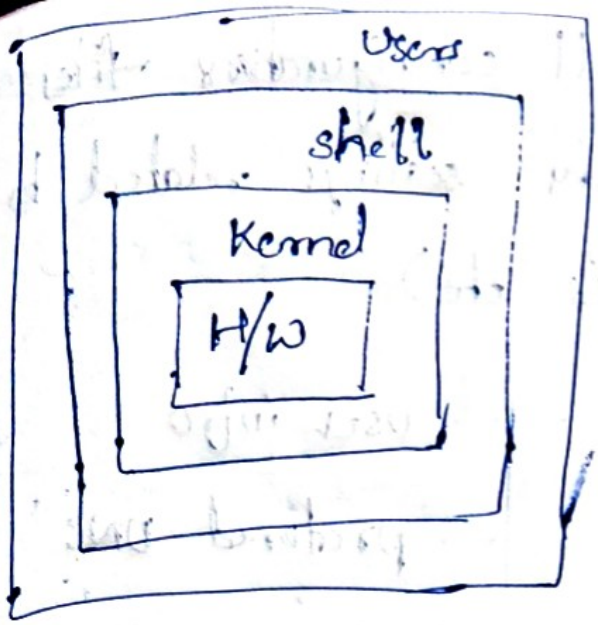


shell :- It translates user defined commands to kernel in such a way that kernel can understand. [human readable ⇄ (convert) binary]

Kernel :- It is the heart of OS, which manages the H/W,

/ → To level directory

/root → It is the home directory for root user (super user)

/home → It the home directory for other users, it provides working environment of other.

/boot → it contains bootable files for Linux

/etc → it contains all configurations files
{ configurations means settings related to users, s/w, disk etc)

like /etc/passwd ---- user info.
/etc/resolv.conf ---- prefered DNS
/etc/dhcpd.conf ---- DHCP server.

/usr → by default all s/w's are installed in the /usr directory.
(usr = Unix sharable Resource)

eg:- It is like program files in windows.

/opt → It is optional directory for /usr
It contains third party s/w's
eg:- c:\ Program files

/bin → It contains commands used by all users
(binary files)

/sbin → It contains comands used by only Super user (root)

/dev → It contains device files
like /dev/hda - - - for hard disk
/dev/cdrom - - - for cd rom
Ily like device manager of windows.

/proc → It contains all process files
eg:- like task manager in windows.

/var → It contains variable data like mails, log files.

/tmp → contains temporary files for small period of time.

/mnt → It is default mount point for any partition, It is empty by default.
like my computer in windows. like c drive, D: drive.

/media → It contains all of removable media like CD-ROM, pen drive.

/lib → it contains all library files which are used by OS.

→ To check kernel version

`uname -r` → for redhat

`lsb_release -a`

or

`cat /etc/os-release` → for ubuntu

→ To check no. of updates

`yum check-update`

→ To check server version

`cat /etc/redhat-release` → redhat

`cat /etc/debian_version` → for ubuntu, fedora, centOS

→ To see the calendar

`cal`

→ To see the calendar of 2024

`cal 2024`

→ To see particular month's calender in 1948

```
cal 8 1948
```

August

→ `touch file{1..5}` → it create 5 files

→ `mkdir dir{1..5}` → " 5 directories

→ shows that you are in home directory

→ To check inode number of file

```
ls -i file_name
```

inode → series of numbers given to a file

Soft link

→ It creates a link between two files, and the same data will be present in both the files → used for shortcuts

Creation

```
ln -s file_name soft
```

Hard link

→ If the original file is deleted then also link file will contain data

→ Used for backups

```
ln original_file destination_file
```

eg:- ln file1 hard

grep :- To display / to search particular word in a huge file.

`grep root /etc/passwd` → it display only the lines which contains root

Search from two files same word

`grep root /etc/passwd /etc/shadow`

→ To see the line numbers in which the root word is present

`grep -n root /etc/passwd`

→ To search more words in that file.

`grep -e root -e adm /etc/passwd`

(or)

`grep -E 'root|adm|sshd|ftp|usca' /etc/passwd`

`rpm -qa` → it shows all the s/w's which are installed.

less → it displays content in page wise

```
less /etc/passwd
```

more → also same as less

head → it displays top 10 lines

tail → " last 10 lines

```
head /etc/passwd
```

```
tail /etc/passwd
```

→ sort :- It arranges the file contents in alphabetical order or numerical order.

```
sort file_name
```

→ `sort -u file_name` → it shows only unique lines

→ cut -d : -f 1 /etc/passwd

cut -d : -f 1,5 /etc/passwd

cut -d : -f 1-5 /etc/passwd

separator (" ", @, $, #) field

sed :- Stream editor → searching something in the file and editing.

```
sed 's/search for/replace with/g' filename
```

eg:-

```
sed 's/Linux/ubuntu/g' file1
```

tee :- It will show the o/p and capture in a file.

```
tail -5 /etc/passwd | tee -a out_file_name
```

find :-

It is used to find any file

```
find / -name file_name
```

```
find / -name *.war
```

```
find / -name "egg*."
```

→ To search directory

find / -type d -name mydir

{l..} → link file

find / -user myusr

find / -inum 1234786

→ default umask value = 022

→ default value of new file = rw, r, r

→ " directory = rwx, rx, rx

777
-022
755 (7 = 4+2+1 = rwx, 5 = 4+1 = rx, 5 = 4+1 = rx)

for file

777
-111
666
-022
644 (6 = 4+2 = rw, 4 = r, 4 = r)

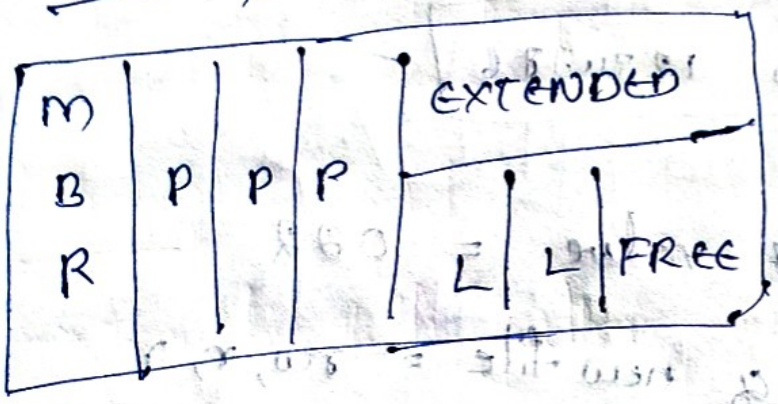
→ We can change the umask value

umask 002 or any permission

Partitions

→ It means to divide a single hard drive into many logical drives.

Disk Partitioning Criteria:-



MBR = MASTER BOOT RECORD

P → Primary Partition

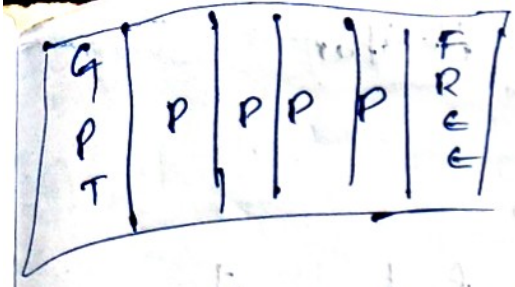
EXTENDED = Extended Partition

L = Logical partition

FREE = Free space

→ We can create 16 Primary Partitions if the Hard disk is 9TB

→ In primary partitions the OS is installed.



GPT = GUID PARTITION TABLE

P = PARTITION

FREE = FREE SPACE

→ You can create 128 partitions if hard disk is 2TB.

"S" letter technology (scsi, sas, sata, ssd)

/dev/sda sdb sdc

/dev/sda1 sdb1 sdc1

virtual disks

/dev/vda vdb vdc

/dev/xvda xvdb xvdc

cd/dvd drive

/dev/sr0

/dev/scd0

File systems supported for Redhat

→ ext2, ext3, ext4, XFS

→ It supports the journaling feature, it means recovers the data, any unsaved data will be saved, it keeps track of saving the data.

Mounting :-

→ Connecting the drive and folder.

Unmounting :-

→ disconnecting the drive and folder.

→ whenever we want to read/write the data on the storage devices.

eg :- partition, pendrive, cd-drive ---> mounted on a directory/folder

files related to mounting

① /etc/mtab :- Right now how partitions are

mounted and where they are mounted.

② /etc/fstab :- The partitions which are permanently mounted.

To view the existing partitions

```
fdisk -l
```

→ To see the more details of the device

```
parted -l
```

Partition administration using fdisk

```
# fdisk <device_name>
```

If Press n to create new partition

+5G ↵

It create 5G space.

To save press w ↵

→ To update the kernel about the partition

```
partprobe /dev/sda
```

```
partprobe <device_name>
```

→ when you create extended logical partition, for the first time, and continue to logical partition reboot it, then the kernel will be updated.

→ To see the updated partitions

```
cat /proc/partitions
```

→ To see the formats of the device.

```
lsblk -f
```

* Extended partition can't be formatted only logical partition is formatted

To format

```
mkfs.<file system type> <partition name>
```

eg:- mkfs.ext4 /dev/sda5

Mount:- Now, connect it to some directory.

```
mkdir /test
```

```
mount /dev/sda5 ./test
```
→ temporary mount.

* → when you mount temporary, the mount point will be lost when you reboot or shut down the machine

→ ext maintains a folder where your lost data is found,

i.e `lost+found` → folder.

Unmount:-

```
umount /directory_name
```

eg:- `umount /test`

→ To see all the mounts

```
mount | grep device_name
```

Permenant mounting

→ whenever you want to do permanent mounting go to fstab

```
vim /etc/fstab
```

→ write the data as shown below in the fstab

devicename	Mountpoint	Type of FS	Mountoptions	Dumping	check Sequence
/dev/sda5	/test	ext4	defaults	0	
/dev/sda6	/dev	xfs	defaults	0	

→ In dumping 1 → take backup of the data when the system is crashed
0 → don't take backup.

→ mainly give 1 for OS related partitions

→ fsck :- (check sequence)
In which order it has to check the partitions

Creation:-

vim ~~/fstab~~

→ vim /etc/fstab

→ write the necessary information

→ :wq

→ mount -a → then only it is mounted, don't reboot the machine always

Note:- In vim type small 'o' it will open a new line to write.

Scenario:-

→ In /etc/fstab, You have

/dev/sda5 ---→ /a

/dev/sda6 ---→ /b

/dev/sda7 ---→ /c

/dev/sda8 ---→ /d

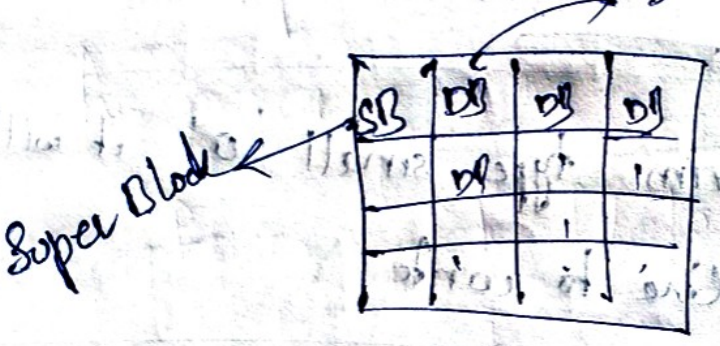
when you delete /dev/sda5 then the benefits

of the /dev/sda5 will be renamed as /dev/sda5

Now /dev/sda6 renamed as /dev/sda5

then it will be mounted /a folder, it make a problem,

→ To overcome this, we have UUID for each block.



→ Super block holds all the information of each data block

→ Each super block has UUID → unique User ID.

→ No two SB has same ID.

→ So that UUID are not changeable.

→ To know UUID

blkid /dev/sda5 | blkid device_name

* Instead of name of device in fstab use UUID, then their won't be conflicts and device name are not changed automatically.

UUID	mountpoint	Type of FS	mount options	dumping	fsck

Trouble shooting :-

→ when you are inside the directory and trying to do umount, then it throws a error, i.e target is busy.

Sol:

→ Come out of that directory and umount.

(*) Now you are not in that directory and doing umount, but it throws error i.e busy, it means some other user is in that directory.

then how to know who is in that directory

```
fuser -cu /test2
```

→ It shows the username who is inside that directory.

→ But we want to know what he is doing inside that directory.

```
lsof /test2
```

→ lsof shows all the system process

→ To stop the activity of the user or stop his process.

```
fuser -ck /test2
```

-c → current process
-k → kill

→ To know the disk space

```
df -h
```

→ To know the size of partitions.

```
df -h /dev/sda5
```

Swap Space Management!

→ Swap space in Linux is used when the amount of physical memory (RAM) is full.

→ If the system needs more memory resources and the RAM is full, inactive pages in memory are moved to the swap space.

⭐ Swap is a waiting area, for the process are waiting in Swap memory but it can't run any process.

Swap out! — whenever the process goes out from swap memory.

Swap in! — whenever the process come into swap memory.

→ To check free space is Ram, swap, memory

```
free -h
```

→ To check which partition is used in swap

`swapon` (or) `swapon -s`

Scenario:-

Suppose your swap memory is not sufficient then create new swap and add it.

→ fdisk /dev/sda

(or)

/device_name from which you have to create swap

→ create partition → it will create normal partition

→ To change it to swap, we must change the ID of the partition

	ID
Swap	82
Linux	83
LVM	8e

→ type t ↵

→ partition no ↵

→ type 82 ↵

→ To print the table : p ↵ and w ↵

Now format the swap partition after creating.

```
mkswap <partition_name>
```

→ To activate the swap space use

```
swapon <partition_name>
```

→ To deactivate the swap space use

```
swapoff <partition_name>
```

Now to make it permanent take the UUID and paste in /etc/fstab

UUID swap swap defaults 0 0

→ To activate all swap entries from fstab

```
swapon -a
```

→ To remove the swap partition from the system.

step-1: first deactivate it

```
swapoff /device_name
```

step-2: Remove from fstab

*For emergency purpose we can use file as swap for some time, create 1GB of swap file.

Creation:-

→
```
dd if=/dev/urandom of=/swapfile_name
                bs=1M count=1024
```

dd = disk dump

if = i/p file

of = o/p file

bs = blocks per second (1M) to generate 1GB it has to run 1024 times)

count = 1024

/dev/urandom → it creates random data.

→ find out in partition you create the file partition. and use that path

```
df -h
```

→ Now change permissions as rw to user

```
chmod 600 /swapfile
```

→ Now format it as

```
mkswap -f /file_name_of_swap
```

→ Activate swap file.

```
swapon /swap_file
```

→ To make it permanent

```
vim /etc/fstab
```

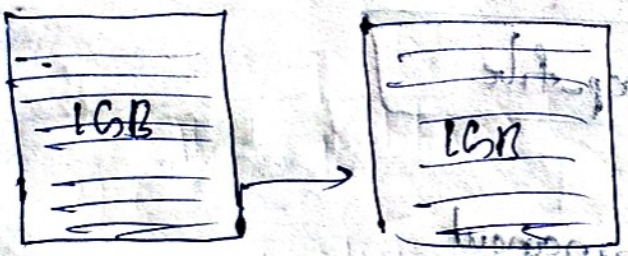
UUID swap swap defaults 0 0

Note! - This swap file is not recomanded because it creates slowness in the process, in emergency basis use file swap

→ We call swap in
- Linux : swap space
- Solaris : swap space
- IBM AIX : paging space
- hp-ux : paging space
- windows : virtual memory.

LVM

Logical Volume Management



Suppose I have 1GB of space for storage for an application, if the users are working in it and my 1GB of space is running out and I need to create more space and to store the data, and one more 1GB is created and add, if this is also running out

How far you will be migrating file from one disk to another so on. and so forth?

→ It is not possible in real time, so to avoid these circumstance we implement LVM and store data in LV's : whose size can be easily increased whenever required without downtime.

→ Virtual Volume (i.e which is not real) it means I can increase the size and decrease the size.

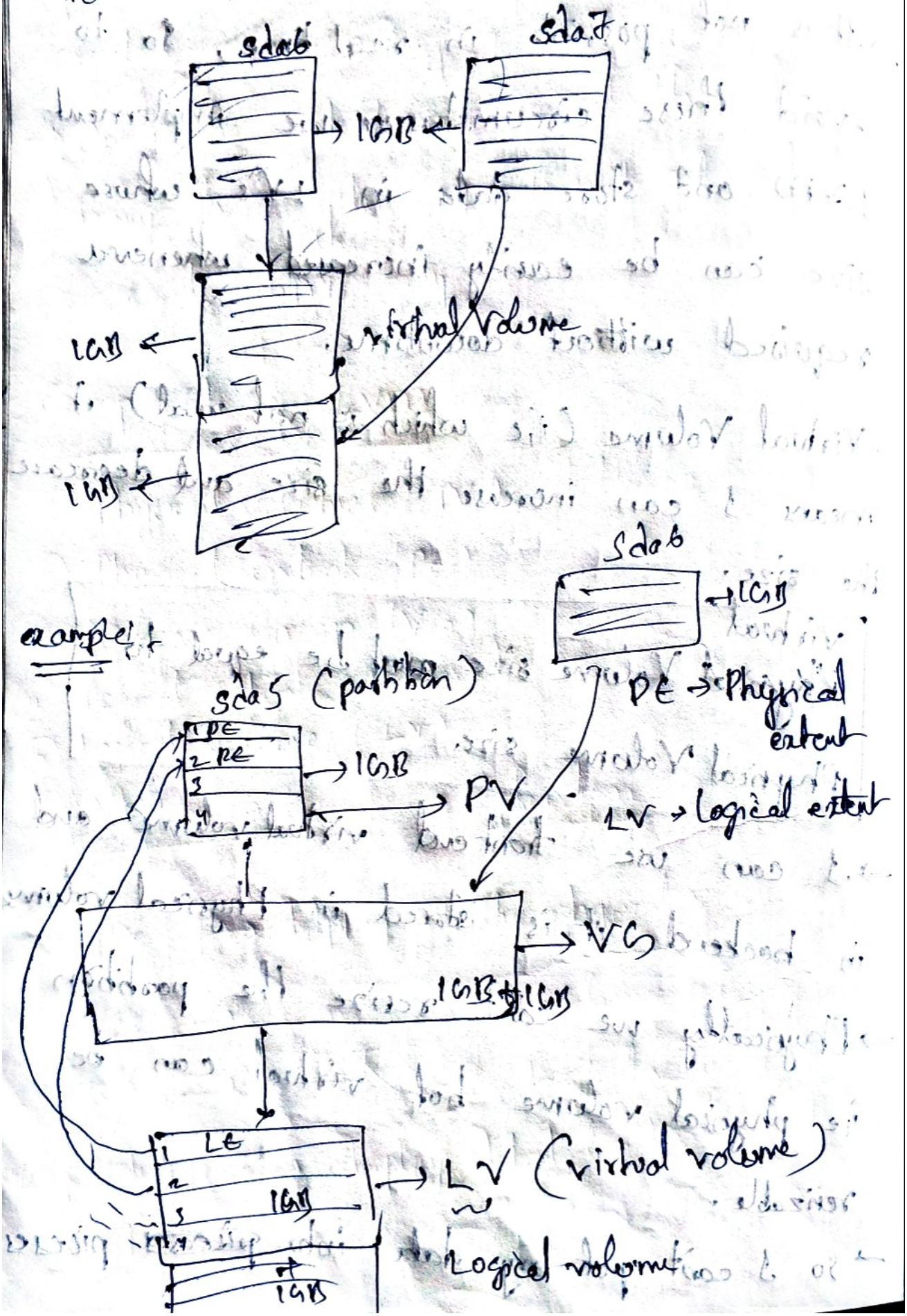
> Virtual Volume size must be equal to Physical Volume size

→ I can use front end virtual volume and in backend it is stored in Physical volume

→ Physically we can't resize the partition ie physical volume but virtual can be resizable.

→ So I can't put my data into ~~pieces~~ pieces

→ create one more partition and add it to virtual volume.



→ The moment you added PV into VG, the PV is divided into small partitions automatically. are called Physical Extents and Logical volume too is divided and called as logical Extents.

→ There is no rule that is 1st LE can map to 1st PE but it can map to any other also depends on which extents is free.

→ The resizing of LV's, maping of PE to LE, maintaing the PV is done by VG.

→ You can have any no. of VG's, one volume for one application as per your requirements.

→ There will be sizes for PE and LE the default size is 4MB, sizes are : the extent sizes 1MB, 2MB, 4MB, 8MB, 16MB, 32MB ... 128MB,

→ If the extent size is 4MB for VG. then in PE, LE the extent sizes are created of 4MB.

→ Once the extent size is set, then how many PV's you add, it will get extent size of 4MB only.

Components of LVM

① Physical volumes (PV)
② Physical extent (PE)
③ Volume Group (VG)
④ Logical extent (LE)
⑤ Logical volume (LV)

LVM Command functions

① pvs → display all the physical volumes.
② vgs → " " " volume groups
③ Lvs → " " logical volumes
④ pvdisplay → display detailed info about PVs
⑤ vgdisplay → " " VG

6. lvdisplay → displays detailed info about lv's
7. pvcreate → Create a new physical volume
8. vgcreate → " " volume group
9. lvcreate → " " logical volume.
10. vgextend → Add a new physical disk to a volume group.
11. vgreduce → Reduces a volume group by removing a PV from it.
12. lvextend → Extends the size of logical volume
13. lvreduce → Reduces the size of logical volume.
14. lvresize → Resizes a logical volume, i.e. increases as well as decrease the size.
15. pvmove → Moves the contents of a PV from one PV to another
16. lvremove → Removes/deletes a logical volume
17. vgremove → " " volume group
18. pvremove → " " physical volume.

Lvm creation:-

step-I:- first create a partition

```
fdisk /dev/sdaf
```

step-II:- change this Lvm ID i.e 8e

: t ↵

: L ↵

: 8e ↵

: p → to see the table of partition

: w ↵ save

cat /proc/partitions → to see partitions.

step-III:- Now create a PV from the partition.

```
pvcreate <partition_name>
```

eg:- pvcreate /dev/sdaf

→ To check pv's,

```
pvs or pvdisplay
```

Allocatable : No i.e it is not added in any volume group.

step-IV: Create volume group

`vgcreate <give_some_name_for_vg> <PV name>`

q:- vgcreate myvg /dev/sda7

→ To check, `vgs` or `vgdisplay <vgname>`

Now,

Allocatable : yes
PG Size : 4.00 MB
Total PE : 127
Free PE : 127
PV size : 512.00 MB / not usable 4.00 MB

it is used to store metadata of PV's etc

(512 - not usable 4.00MB) / 4.00 MB (extent size) = 127

step-V: Now create LV

lvcreate -L <size of Lv> -n <name for Lv>
<vgs name>

eg:- lvcreate -L 300M -n mylv myvg
(to create a LV of 300MB.)

→ lv display

LV Path : /dev/myvg/mylv

open : 0 {it changes to 1 when it is mounted}

→ Path is very important for formating and mounting

step-VI :- format

```
mkfs.ext4 /dev/myvg/mylv
```

step-VII :- Mount

mkdir /mydirectory

```
mount /dev/myvg/mylv /mydirectory
```

mount -a

for permanent mounting edit /etc/fstab

→ To extend lv

```
lvextend -L +1GB /dev/myvg/mylv
```

→ To remove lv

```
lvremove /dev/myvg/mylv
```

→ To increase the default extent size in vg

```
vgcreate -s 16 myvg /dev/sda
```

→ To add pv to prosed group:

```
vgextend vg_name device_name
```

→ To remove pv from the vg

```
vgreduce vg_name device_name
```

USER ADMINISTRATION

→ In Linux/Uni's user is the one who uses the system.

→ Users on a systems are identified by username and userid (uid).

→ Every user of the system is assigned a unique user ID (the UID).

→ Users name and UID are stored in /etc/passwd

→ User's password is stored in /etc/shadow in encrypted form.

→ Users are assigned a home directory and a program that is run when they login (usually a shell).

→ Users can't read, write, execute each other files without permission.

Type	example	USER ID UID	GROUP ID GID	Home Directory	shell
Super User	- root	- 0	- 0	- /root	- /bin/bash
System User	- ftp, ssh, apache	- 1 to 999	- 1 to 999	- /var/ftp, /var/www/html, /var/named, etc	- sbin/nologin
Normal User	- Visitor, myuser, etc	- 1000 to 60,000	- 1000 to 60,000	- /home/user name	- /bin/bash

* → Every user has a home directory as (/home/username) and mail box is created (/var/spool/mail/username), and a unique UID and GID.

→ In /etc/passwd

root:x:0:0:root::/bin/bash

root → name

x → link to password file ie /etc/shadow

0 or 1 → UID

0 or 1 → GID

root or bin → comment (brief info about the user)

/root or /bin → home directory of user.

bin/bash or /sbin/nologin → shell.

Creating User

```
useradd <user_name> <options>
```

```
Useradd <options> <user_name
```

options :-

-u user id

-G Secondary group id

-g primary group id

-d home directory

-c comment

-s shell

eg:- Create user sudarshan

```
useradd sudarshan
```

→ It will create user and allocate default uid, gid and shell.

eg:- create a user sudarshan, add uid as 1506, gid as 1506, shell as /bin/sh.

```
useradd sudarshan -u 1506 -d /home/sudarshan
 -c TeamLead -s /bin/sh
```

To modify something in user

```
usermod sudarshan -u 1507 -d /home/sudh
 -c Manager -s /bin/bash
```

Assign Password to a User

→ To change the password root has powe to do that. but a user can't change the password of other user.

As a root → passwd sudarshan

As a sudarshan → passwd

To check the which user is :

```
whoami
```

To get detailed info

```
who am i
```

To display no. of users

```
who
```

To check the status of password for user

eg:- passwd -S sudarshan

```
passwd -S user_name
```

→ It shows password is set, or encryption info.

To lock the user account / to disable the account

```
usermod <options> <user_name>
```

options:- -l to change login name

-L to LOCK account

-U to UNLOCK account.

eg:- To change user name sudarshan to chakram

```
usermod -l chakram sudarshan
```

```
usermod -l <new_name> <old_name>
```

eg:- To LOCK sudarshan account

```
usermod -L sudarshan
```

To check the status

```
passwd -S sudarshan
```

result:-

sudarshan : ! $. ~~AB78~~

it means the account is locked.

→ To unlock the account

```
usermod -U sudarshan
```

→ I want user to work with application and I don't want user to login;

stop user login

→ For that change the shell of the user as nologin shell.

→ If you give /sbin/nologin, the user logins will be blocked but the user is active.

```
usermod -s /sbin/nologin sudarshan.
```

→ To change password of user sudarshan

→ Switch to sudarshan user

```
su - sudarshan
```

→ Just type

```
passwd
```

Scenario:- As of my organisation rules, password must expire after 7 days, change of password must after 24 hours from creation of password, I want to send notification to change password before 1 day of expiration. I want to keep the user active even the password is expired.

Parameters of password, like min and max password age, password expiration warnings and a/c expiration date etc

`chage -l <user_name>` → To view the details.

eg:- chage -l sudarshan

Now to change the parameters

`chage <user_name>`

eg:- chage sudarshan

→ It prompts line by line, give details as per your requirement.

Min Password Age [0] : 1 {You can change after 1 day of your creation}

0 → means you can change anytime.

Max Password Age [99999] : 7 { it will expire after 7 days }

Last Password change (YYYY-MM-DD) [2024-07-30]: don't disturb it, just type ↵

Password Expiration Warning [7] : 2 (2 day before expiration).

Account Expiration date (YYYY-MM-DD) (-1) : 2024-07-31.

Password Inactive [-1] : 1 { 1, you can change password and continue even after expiration }

[-1] :- It means, your password expired but you didn't do login and changed before expiration

After 15 days or some day of expiration, You loged in, but it accepts the old password and asks you to change the password to continue

To delete the user.

```
userdel <user_name>
```

but it won't delete the home directory and mail box for that user

To delete everything for that user

```
userdel -r <user_name>
```

GROUP ADMINISTRATION

→ Users are assigned to groups with groupID

→ The group name and GID are stored in /etc/group.

→ Each user is given their own private group.

→ They can be added to their groups to gain additional access.

→ All users in a group can share files that belong to the group.

→ Primary groups are created automatically at the time of user creation.

Create a group

```
groupadd <name for the group>
```

eg:- groupadd mygroup

Create a group with user specified group id:

```
groupadd <ophion> <name for the group>
```

eg:- groupadd -g 1050 mygroup2

modifying the group properties

```
groupmod <ophion> <arguments> <group name>
```

ophions:-

-g → to change the GID
-o → to override the previous GID
-n → to change group name

eg:- change the group id from 1050 to 1060

```
groupmod -g 1060 mygroup2
```

eg:- change the group name from mygroup2 to mygroup3.

```
groupmod -n mygroup3 mygroup2
```

```
groopmod -n new_group_name old_name
```

To make the primary group to sudarshan

user

```
usermod -g mygroup3 sudarshan
```

→ Now sudarshan belongs to mygroup3

To change the sudarshan group ID to previous GID

```
usermod -g sudarshan sudarshan
```

```
usermod -g group_name user
```

Adding multiple Users to a group

`gpasswd <options> <arguments> <group_name>`

options :-

-M : for adding multiple users to a group

-a : " " single " "

-A : for adding a group Administrator

-d : removing a user from a group;

eg :-

`gpasswd -M <user>, <user>, <user> <group_name>`

gpasswd -M u1, u2, u3 mygroup

Adding a single user

`gpasswd -a u4 mygroup`

→ Admin info. is present in a file

`/etc/gshadow`

eg: mygroup:x:!::u1,u2

↓

it is empty, ie no admin is there

→ Now add a admin to the group,

eg:- `gpasswd -A chakram mygroup.`

o/p in /etc/gshadow

mygroup : x : ! : chakram : U1, U2

To delete a group

`groupdel <group_name>;`

eg:- groupdel mygroup

CONTROLLING ACCESS TO FILES

① Special permissions or Advanced Permissions

② Access Control list (ACL)

Special Permissions or Advanced Permissions

→ There are three special permissions that can be assigned to file or directory apart from basic file permissions (rwx), they

(i) SUID :- SET USER ID

(ii) SGID :- SET GROUP ID

(iii) STICKY BIT (others)

Permissions	Symbolic form	Numeric form	Syntax
SETUID	s or S	4	#chmod u+s (or) chmod 4766
SETGID	s or S	2	#chmod g+s (or) chmod 2766
STICKY BIT	t or T	1	#chmod o+t (or) chmod 1766

4 766
↓ ↑
special permission. normal permissions

Note :- where s = setuid + execute permission, and S = setuid only, Same for SGID and for sticky bit.

Scenario :-

for example /etc/shadow file, only root has permissions to update it, only root can create password for other users or change password, A normal user can't update

other users password.

i.e. root (0) ID myuser (1001) ID

myuser (0) ← SUID

SUID will convert non-owners ID into owners ID temporarily.

→ To see the file permissions location

which passwd | which /etc/passwd

which ls | which vgs etc.

→ which passwd → it gives the location of that file.

ls -l /usr/bin/passwd → shows the permissions

as -rws r-x r-x 1 root root /usr/bin/passwd

user group other

↓

it indicates the SUID

user

-rws → u = suid + x

group

-rws → g = sgid + x

other

rwt → 0 1 sbit + x

To remove suid for user

```
chmod u-s /usr/bin/passwd
```

ls -l /usr/bin/passwd

Now login using another user and try to change password, it shows error, it means user's ~~root~~ permissions are removed.

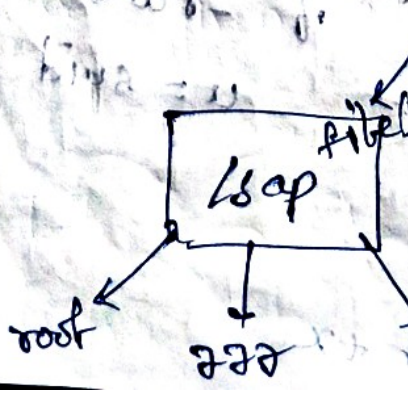
To add permissions again

```
chmod u+s /usr/bin/passwd
```

For SGID :-

Now group members ID is changed to the file or directory's group ID, so that he will become the group owner to execute that file/directory.

For example



Let say I have a folder /sap, it's owner is root and given full permissions and it has a group called sapgrp and a user u1 is the user creates a file in the group he will be the owner of the file because he became member of the sapgrp with SGID.

All the files belong to this directory should belong to sapgrp group.

→ mkdir /sap

→ chmod 777 /sap

→ ls -ld /sap
(ld ↓ directory)

→ groupadd sapgrp

→ chgrp sapgrp /sap

→ ls -ld /sap

Now login as user u1 and create files as touch f1 f2 f3

→ ll

→ It shows the owner and group owner is u1.

→ Now change it sapgrp for that & apply SETGID.

```
chmod g+s /sap
```

→ ls -ld /sap

→ Now create files as the user u1

touch f1 f2 f3

→ ll

o/p:-

```
-rw-rw-r-- 1 u1 u1  0 ... } before SETGID
-rw-rw-r-- 1 u1 u1  0 ... }
-rw-rw-r-- 1 u1 sapgrp 0 ... } after SETGID
-rw-rw-r-- 1 u1 sapgrp 0 ... }
```

→ Let's say I have given full permissions to the /sap directory any one can come inside and create and delete files of other users, it means there is no security.

want to make secure, other user can create files, but he can't delete other users data for the we use STICKY BIT

```
chmod o+t /sap
```

ls -ld /sap

→ Now login as other user and create files and do delete other users files, it won't allow to delete.

→ rule;

	FILE	DIR
SUID	✓	✗
SGID	✓	✓
SBIT	✗	✓

Network Configuration & Troubleshooting

→ It is a connection between two or more machines to communicate with each other

① NIC (n/w Interface Controller or Card)

A n/w interface controller (also known as n/w interface card, n/w adapter, LAN adapter and by similar terms) is a computer H/w component that connects a computer to computer n/w.

In window

LAN

In rhel6, lower versions

eth0, eth1 ..

above rhel6, rhel7

the names are not fixed, it depends on which which company adapter you are using.

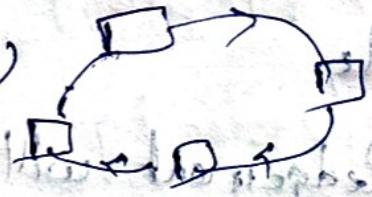
like eno
ens0, ens1
enp0s8
en0112324 etc.

② Topology
It is a scheme/design in which the computers are connected in the n/w.

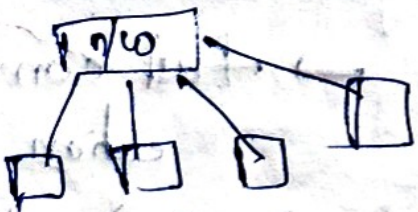
(i) Bus n/w topology.
(ii) Ring
(iii) Star
(iv) Mesh
(v) Tree

(i)

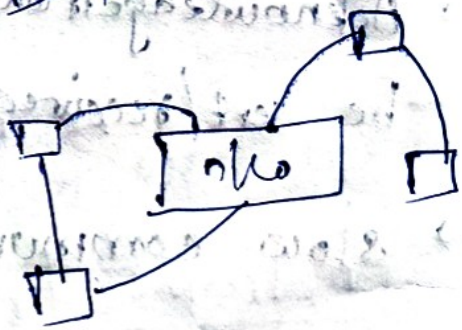
(ii)



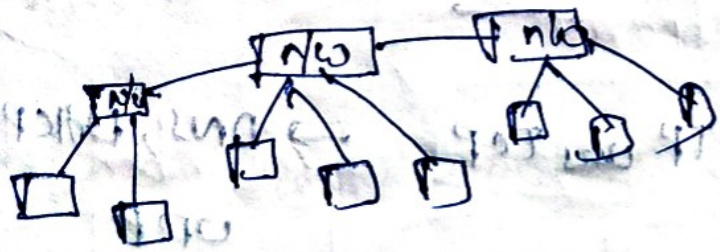
(iii)



(iv)



(v)



③ Protocol :-

→ These are the rules to follow in the n/w.

→ It will decide that how one machine will communicate with other.

→ We will exchange information in the form of packets.

TCP/IP	UDP
→ Transmission Control Protocol	→ User Datagram Protocol.
→ It is connection oriented	→ Connectionless
→ Reliable	→ Non-Reliable
→ Acknowledgement will be sent/received	→ No acknowledge.
→ slow communication	→ fast communication.
→ Protocol no. for TCP is 6	→ no is 17.
→ HTTP, FTP, SMTP uses TCP	→ DNS, DHCP use UDP.

→ It is secure → It is not secure.

→ In this password are encrypted → In this not

③ IP addrew :

→ It is a unique identity given to any machine, like phone number.

ipv4 ⟶ 4 octec value
0.0.0.0 to 255.255.255.255(1-254)

LAN → In one building only.
WAN → In two buildings communication
MAN → Mehopolian Area n/w ie, communication between two cities / coontries : eg:(internet)

CLASS-A 1 - 126 } → used in WAN, MAN
CLASS-B 128 - 191
CLASS-C 192 - 223 } → used in LAN

CIDR :- Classless Inter-Domain Routing.

→ 127.0.0.0 to 127.255.255.255 is reserved for loopback address

Loopback :-

→ A special IP no (127.0.0.1), it is used to test whether your computer is sending and receiving information. i.e. it sends information to his machine.

→ To know the ip of your laptop

`ifconfig` (or) `ip a`

→ The adapter info is kept

`/etc/sysconfig/network-scripts/`

→ To see the name of your machine

`hostname`

→ To change hostname

`hostname <name>` and reboot etc.

it will change temporarily.

→ To change it permanent.

`cd /etc/hostname` and reboot it / just exit and relogin.

→ To know the detailed info about your machine

`hostnamectl`

→ to change hostname permanently.

`hostnamectl set-hostname <name>`

→ To check other machine is reachable

`ping ip-address`

eg:- `ping 192.168.10.0` → it goes on counting for pinging.

→ I want ping it just ~~for~~ 3 times

`ping -c3 192.168.10.0`

→ The DNS server info is kept in

/etc/resolv.conf

ie vim /etc/resolv.conf

eg:- nameserver 192.168.10.10

Add above line in that file,

→ If you don't have DNS server then go into /etc/hosts file and add ip address and name of your machine

eg:- www.sudarshan.com 192.168.10.0

```
vim /etc/hosts
192.168.10.0  www.sudarshan.com
then ping www.sudarshan.com.
```

It means hosts is resolving ip into hostname.

Connections! -

→ It is used to change the ip-address of the machine.

→ To see the connections

`nmcli` network Manager cli

→ To check the status of your adapter.

`nmcli dev status`

→ To do with nmcli just type nmcli and shines Tab

`nmcli a Tab` it shows the options.

→ To see the connection details

`nmcli con show`

→ To see in detailed about connection

`nmcli con show <connection_name>`

Scenario:

* Make a new connection "office"

office
enp0s3
type → ethernet
ipv4 → 192.168.10.81/24
gwv4 → 192.168.10.1
autoconnect → yes/no

→ Type nmcli 2 times Tab, it will show options and follow thats it.

```
nmcli con add con-name office type
ethernet ipv4.addresses 192.168.10.81/24
ipv4.gateway 192.168.10.1 autoconnect
no
```

→ Now make it activate

```
nmcli con up office
```

Use "nmtui" tool (or)

→ To check the speed of your adapter

eg: ethtool enp0s3

```
ethtool <adapter_name>
```

(or)

```
mii-tool <adapter_name>
```

→ To check the port no.

```
netstat -ntlp  → tcp ports
```

```
netstat -nulp  → for udp ports
```

(2) Access Control List (ACL)

→ If you want to give rights to a file or directory between users, groups and others. We create ACLs.

→ To check the acl permissions:

```
getfacl <option> <dir/file_name>
```

options:

-d : Displays the default ACL

-R : Recurses into subdirectories

eg:- To see the ACL on myfile

```
getfacl myfile
```

Scenario:

(i) I have a file → myfile

(ii) And users u1, u2, u3

(iii) And groups g1, g2, g3

(iv) Assign permissions as below

u1 : rwx	g1 : rx
u2 : rw	g2 : r
u3 : r	g3 : NIL

→ touch myfile

→ Now add ACL to the file

To assign acl to particular user

```
setfacl -m u:<user_name>:permission> <file/dir
```

eg:- setfacl -m u:u1:rwx myfile

```
setfacl <opton> <argument> <file/dir_name>
```

option:-

-m : modifies an ACL

-x : Removes an usr/group from ACL

-R : Recurse into subdirectories

-b : remove ACL from a file/dir

arguments:

u: user, g: group

→ groupadd g1

groupadd g2

groupadd g3

→ setfacl -m u:u2:rw, u:u3:r myfile

→ setfacl -m g:g1:rx, g:g2:r, g:g3:0 myfile.

→ getfacl myfile..

→ ls -l myfile

-rw-rwx--+ ↳ it indicates ACL is present.

→ To remove u3 from the ACL

```
setfacl -x u3 myfile
```

→ To remove group g3 from ACL.

```
setfacl -x g:g3 myfile
```

→ To remove ACL for a file

```
setfacl -b myfile
```

Scenario:

(i) Create ACL for directory mydir

(ii) Create files file1, file2, file3

→ `mkdir mydir`

→ `touch mydir/file{1,2,3}`

`setfacl -Rm u:u1:rwx,u:u2:rw,u:u3:r,g:g1:rx,g:g2:r,g:g3:0 mydir`

→ To remove u3 from the mydir

`setfacl -Rx u3 mydir`

→ To remove group g3 from mydir

`setfacl -Rx g:g3 mydir`

→ To remove ACL for a directory

`setfacl -Rb mydir`

SUDO

(Substitute user do / super user do)

→ The file /etc/sudoers has all the roles that users follow.

→ Don't edit the /etc/sudoers directly, instead use "visudo"

→ The logs of all users are stored in

`/var/log/secure`

→ We can customize the log file, i.e go to /etc/sudoers file and add a line at the bottom of the file as "Defaults logfile = /var/log/sudo.log"

→ To make a normal user as admin go to /etc/sudoers file as

`visudo`

add a line myusr ALL=(ALL) ALL

Alloco root to ron anycomands anyushere

root ALL = (ALL) ALL

myusea ALL = (ALL) ALL → add a usea like this in that file

→ All general logs of sudo usea, Logins, logout are stored in

```
/var/log/secure
```

→ Now create custom log-file for sudo

→ visudo

→ Now add as

```
Default logfile = /var/log/sudo.log
```

:wq

→ Do some comands with sudo usea

→ Now see the logs in

```
cat /var/log/sudo.log
```

(*) I want to make 10 useas are admins.

→ Create a group of 10 usea and give

admin powers.

→ Create a group called mygroup

→ `groupadd mygroup`

→ Now add users u1, u2, u3 in that group.

→ gpasswd -M u1, u2, u3 mygroup

→ Now go into visudo. and add the line as

Allows people in group wheel to run all commands

%wheel ALL=(ALL) ALL

`%mygroup ALL=(ALL) ALL`

→ To copy a line, keep cursor on the line and type `yy`

(*) I don't want to give all comands to execute but only some comands.

→ first find the path of the comand and paste in visudo:

eg:- ~~fdisk~~ -l. find path of it as

which fdisk | which parted

o/p : /usr/sbin/fdisk /usr/sbin/parted.

visudo

%mygroup ALL = (ALL) /usr/sbin/fdisk, /usr/sbin/parted

(* space)

what are the comands you want to give the users to execute, paste the path and comma (,) and space and next comand.

→ To know what are the comands permissions given to a particular user

login as user and

sudo -l

→ We can assign bunch of commands to a user/group as

→ Let me give all NETWORKING commands

visudo

→ Now uncomment NETWORKING

→ Add in the ~~group~~ place of group as

```
%mygrup ALL = (ALL) NETWORKING
```

```
sudo -l
```

* I can make my custom alias and give that custom commands to group

```
## customs
Cmnd_Alias MYCUSTOM = add paths of the
                      commands,
```

→ Now add the MYCUSTOM in the group as

```
% mygroup ALL = (ALL) MYCUSTOM
```

I want to block some comands to a user,

→ Take the path of that comands.

→ visudo

→ myuser ALL=(ALL) ALL, !path of that comand,

Scenario:

I want to restrict to edit visudo by other user and I want to restrict other user to login into another user account

→ find path of visudo and su

which visudo → path : /usr/sbin/visudo

which su → path : /usr/bin/su

→ Now as root user

visudo

myuser ALL=(ALL) ALL, !/usr/sbin/visudo, !/usr/bin/su.

BOOTING PROCEDURE

→ Press the power button on your system and after few moments you see the Linux login prompt.

→ when you power on the system and until you get login page, the process which happens between these stages is called booting process.

RHEL-6	RHEL 7/8
BIOS → Basic i/o system executes MBR	BIOS Perform POST
MBR → Master Boot Record executes GRUB	MBR loads GRUB2
GRUB → Grand Unified Bootloader executes kernel	GRUB2 (i) loads Vmlinuz kernel image (ii) executes the content of initramfs image
Kernel → Kernel executes /sbin/init	Kernel (i) loads necessary drive modules from initrd img (ii) starts systems first proce systemd
Init → Init executes runlevel programs	~~systemD~~
Runlevel → Runlevel programs are executed from /etc/rc.d/rc*.d/	

SYSTEMD

(i) Reads configuration files from the /etc/systemd directory

(ii) Reads files linked by /etc/systemd/system/default.target

(iii) Brings the system to the state defined by the system target.

TARGETS

Target has replaced runlevels in Rhel 7/8

POST! → Power On Self Test

→ If everything is going fine it continues but everything is not going fine it gives beep sound

↓

BIOS! → Now BIOS will start to check the boot order (example if you kept HDD in the first, it will load the HDD first)

→ while power on, press f2 or f10 or f12 it will take you to BIOS and set order for boot device

MBR will now activate Bootloader (GRUB)

↓

GRUB is responsible for loading the system by running certain scripts, the scripts which are responsible for booting.

↓

Next kernel will activate remaing all the devices

↓

Next is the init is the first process activated after kernel, it PID is 1.

↓

Next is the runlevel is the working environment, where you can work with command line or graphical ui. or emergency mode to repair something.

↓

last systemd will login screen.

→ In this way booting process goes on.

SYSTEMD :-

→ It activates the process parallely and boots the system faster.

→ The GRUB2 configuration file is located at `/boot/grub2/grub.cfg`

→ Don't edit the above file instead edit `/etc/default/grub`

→ After editing the file you have to run the comand to apply the changes

```
grub2-mkconfig
```

Scenario!

Suppose You are upgrading the kernel from (4.18 to 4.22), while after doing this you are facing some compatable issues, and once you upgraded you can't degrade it, this is the problem, and if you want to degrade take backup

of all data, applications etc and remove the kernel and installing from starting older version,

→ To over come this, GRUB given a option ie install multiple kernels and make your older version as default, If the system is not compatable with 4.22 then go back and reboot it, it loads older version.

eg:- 4.18 → default.
4.20
4.22
4.85

→ The max no. of kernels you can install is 16 versions.

→ You can make which kernel to be default, in grub2 file.

rpm -qa kernel → no. of kernels

Timout value" :-

It is the time to choose to continue booting or stop booting and choose other kernel.

```
set timeout = 5 seconds
```

```
grep timeout /boot/grub2/grub.cfg
```

To edit that value

```
vim /etc/default/grub
```

set timeout = 10 secs

→ To make changes, run the below:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

→ kernel will create a temporary ram file system, called as initramfs to store some kernel information to activate some of the drivers.

Systemd :-

→ It is the first process

→ It reads the file /etc/systemd/system/default.

→ It brings the system to the state target by system initializing tasks such as -

1) setting the hostname
2) initializing the n/w
3) Initializing SELinux based configurations.
4) Printing a welcome banner
5) Cleaning up directories in /var
6) starting swap
7) Mounting file system.

Run level / Targets

→ We have 7 runlevels, out of which we use Runlevel 3 and Runlevel 6.

Runlevel 3 → will opens the command line interface when the kernel loads

Runlevel 5 → it will give you GUI.

→ To find out which run level are you working

```
who -r
```

→ To view current default target

```
systemctl get-default
```

→ To set a default target

```
systemctl set-default <Target.target>
```

multi-user.target
or
graphical.target

→ The monitoring command is `watch`

→ The `startx` command loads the kernel into graphics

Scenario:

→ If you forgot the root password, You can recover by rebooting and going into emergency mode

→ while rebooting when the count down starts press any key like up arrow or down arrow etc

→ Keep your cursor on line number one, and press e. to enter into edit mode.

→ Now take your cursor to line no 4 where linux/vmlinuz is written, and type ctrl e at the end of the line write as

rd.break it will take you into emergency mode.

→ ctrl x

→ mount | grep sysroot

→ mount -o remount /sysroot -o rw

→ chroot /sysroot

→ Now change the password

→ passwd

→ touch /.autorelabel

→ exit
→ exit

Reparing the corrupted boot loader and recovering it

→ There might be a situation where your boot loader i.e GRUB might got corrupted and you want to recover it.

→ Repairing the GRUB means installing a new grub on the existing one from RHEL 7/8 installation media/ DVD.

step-II: Boot the system with DVD

→ Go to troubleshooting and boot into
- Rescue installed system mode

step-I: Click on the vm and right click on it → settings → system → optical

Next click on storage → click on empty → click on cd and choose a disk file from the system. and power on the machine

step-IV :-

→ It says that our os will be kept under

/mnt/sysimage

→ Press 1 ↵

→ chroot /mnt/sysimage

→ find in which device your os is found

→ grub2 -install /dev/sda

grub2-install <device_name>

→ exit

→ exit/poweroff

→ Now click on the machine → settings → make HDD as first boot device. and power on the machine.

Managing installed Services

→ let's say httpd is installed, we have to manage it

RHEL 6

service : status, start, stop, reload, restart

chkconfig : on, off.

RHEL 7/above!

systemd :-

systemctl : status, start, stop, reload, restart

systemctl : enable, disable.

reload → means refreshing

restart → " starting the service again

(it will stop the service and sta

* systemctl status httpd (detailed info)

or

systemctl is-active httpd

or

systemctl is-enabled httpd

FIREWALLD

Port :- It is door/plug, the people from outside can connect your application over the n/w.

ssh/22, telnet/23

http/80

https/443

ftp/21

smtp/25

→ We have 65,536 ports

→ To enable security, who can connect for your system, we use firewalld

→ Firewalld protects you system

→ To check the firewall is active

```
systemctl status firewalld.service
```

(or)

```
firewall-cmd --state
```

→ when firewall is running all the port numbers are blocked

→ To see no. of service and ports are opened

```
firewall-cmd --list-all
```

→ To open a port no for ftp

```
firewall-cmd --add-service=ftp --permanent
```

→ Now restart the service

```
firewall-cmd --reload
```

→ To remove the ftp service port

```
firewall-cmd --remove-service=ftp --permanent
```

```
firewall-cmd --reload
```

→ (0 - 1024) ports are reserved and above ports are used for customization

→ To add a port number 8080

```
firewall-cmd --add-port=8080/tcp --permanent
```

check in which family the port belongs to by `netstat -ntulp`

→ `firewall-cmd --reload`

→ To remove the port no.

```
firewall-cmd --remove-port=8080/tcp --permanent

firewall-cmd --reload
```

→ Now a days hardware firewalls are using by organisation to prevent from hacking

Introduction to Cockpit in RHEL/8

→ The cockpit is a free and open source web-based server management.

eg:- AWS,

→ It is pre installed s/w, we need to enable it.

→ To enable cockpit

```
systemctl enable --now cockpit.socket
```

→ Next time when you login connect the server it shows a url link, then with that link access your server from the browser.

Redhat credentials

ID sudarshan.sws@gmail.com

pass: Sudarshan@455

Backup and Restore

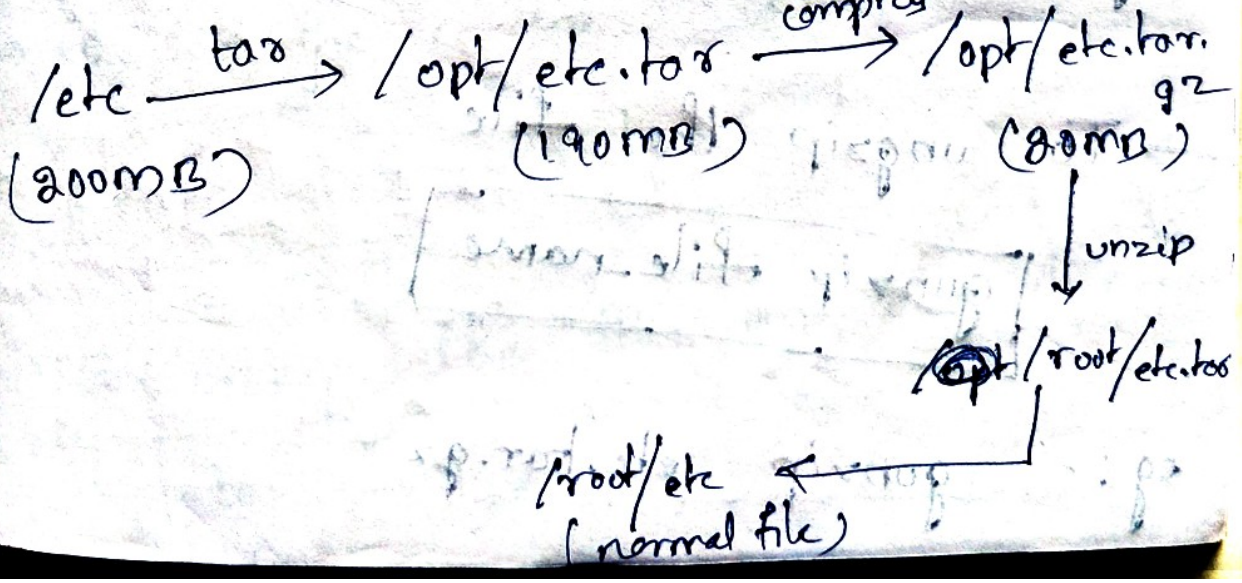
→ A backup or the process of backing up is making copies of data which may be used to restore the original after a data loss event.

→ To take the backup, we have a tool called tar

→ To compress the file we have a tool called gzip.

Scenario:

* Take the backup of file /etc and compress it and revert back to a normal file.



→ To take backup file using tar

```
tar -cvf <destination and name to be> <source file>
```

eg:- tar -cvf /opt/etc.tar /etc

→ Now compress the file using gzip:

```
gzip <file_name>
```

eg:- gzip etc.tar

→ To see the contents in gzip file

```
tar -tvf <file_name>
```

eg:- tar -tvf etc.tar.gz

→ To ungzip the file

```
gunzip file_name
```

eg:- gunzip etc.tar.gz

→ To make it normal file.

```
tar -xvf file_name
```

eg:- tar -xvf etc.tar

Shortcut:

(*) To do all these at a time, i.e backup and compress.

```
tar -zcvf <destination and name to be> <source file>
```

eg:- tar -zcvf /opt/etc.tar.gz /etc

(or)

tar -zcvf /opt/etc.tgz /etc

→ To see the content inside it.

```
tar -tvf /etc.tgz
```

→ To unzip the file and make it normal file

```
tar -zxvf etc.tar.gz
```

Cron Jobs

→ It is possible to create jobs that you want to reoccur. This process is known as job scheduling.

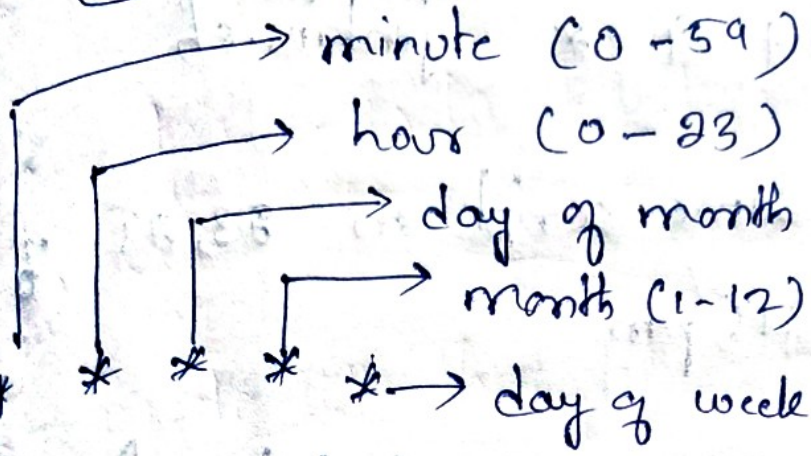
→ For Redhat or other linux this process is handled by the cron service or a daemon called crond.

files related to cron and at

* /etc/crontab → which contains cron job forma
* /var/spool/cron/UserName → contains jobs scheduled by users.
* /etc/cron.deny → used to restrict the users from using cron jobs.
* /etc/cron.allow → used to allow only users who names are mentioned in this file to use cron jobs
* /var/log/cron → log file for cron jobs

→ To see the format of cron jobs

`cat /etc/crontab`



Scenario:

* Reboot the machine at 23 sep 2020 @ 7:30 am

30 7 23 9 4 reboot

* Create a dir in /opt with name crondir @ 23 sep 11:45 am

45 11 23 9 4 mkdir /opt/crondir

or → any day on sep 23

45 11 23 9 * mkdir /opt/crondir

(*) run a script jenkins.sh on 24th sep @ 2:30 pm

30 14 24 9 * jenkins.sh

(*) run the same script on 23, 25 of sep @ 3:15 pm

15 15 23,25 9 * jenkins.sh

(*) run the same script on 23 to 27 sep @ 4:20 pm

20 16 23-27 9 * jenkins.sh.

(*) run same script on 22 to 25 and 27 to 30 sep @ 5:30 pm

30 17 22-25, 27-30 9 * jenkins.sh

(*) Backup script /bkp.sh should be executed only on saturdays @ 11:30 pm

30 23 * * 6 /bkp.sh

(*) Execute the /bkp.sh every 15 mins

*/15 * * * * /bkp.sh

→ Edit your crontab file, or create one if it doesn't already exist.

```
crontab -e
```

→ To display your crontab file

```
crontab -l
```

→ Remove your crontab file

```
crontab -r
```

→ To see a particular user doing crontab

```
crontab -u
```

if combined with -e, edit a particular user crontab file

if -l display particular user crontab file

if -r ddelete a particular user's crontab file.

→ tty is used to display your screen address

(*) To display welcome on your screen at 7:44 pm on 22 sep

→ tty

o/p: /dev/pts/0

→ crontab -e

44 19 22 9 * echo "welcome" > /dev/pts/0

☆ → To execute a cronjob crond.service must be in active.

Scenario:

Run the backup script of /etc file on 23 sep @ 11:30 am.

vim /bkp.sh

#!/bin/bash

echo "backup of /etc is starting"

sleep 2 /* it will not execute the next command for 2 seconds */

tar -zcvf /opt/etc.tgz /etc

echo "Backup of /etc is completed"

:wq

→ Now crontab -e

30 11 23 9 * /bkp.sh

→ Before that file must have executable permissions for all users

```
chmod +x /bkp.sh
```

→ To check myuser crontab as a root user

```
crontab -lu myuser
```

→ To edit users crontab file

```
crontab -eu myuser
```

→ To see who has edited the crontab file

```
tail /var/log/cron
```

(*) Restrict myuser not to have cron job to run/execute/edit.

`vim /etc/cron`

→ Search for the file cron.deny

→ vim cron.deny

→ Now just add user_name

→ :wq.

(*) I have 100 users, I want to allow only 4 users to run cron job and block 96 users.

→ create a `/etc/cron.allow` → this won't be present in /etc/cron.

→ vim /etc/cron.allow

u97

u98

u99

u100

: wq

Note :- If cron.allow file is created and is empty, no user can be allowed to do cron job except it must contain any name, because of this cron.allow is not created by default.

Let's say, You have added users in both ; cron.allow and cron.deny.. then cron.allow will have more precidence or power to execute, cron.deny won't be executed.

SSH

→ In older ways, we used telnet to connect remote server, On this telnet we used to send and receive everything in plain text, So any one can sniff.

→ SSH (Secure shell) was designed to provide security when accessing another computer.

→ The configuration file of ssh is

`/etc/ssh/sshd_config`

→ SSH deamon or service is sshd

→ To connect other machine with root user.

`ssh pvt_ip_of_other machine`

→ To connect with normal user

`ssh -l myuser pvt_ip_of_othermachine`

→ To check the uptime of your machine

```
uptime
```

→ without logging into other machine, you can check uptime of that machine.

```
ssh pvt-ip-othermachine uptime
```

→ You can execute any comand, without loging in other machine and see the information.

eg:

```
ssh pvt-ip tail /etc/passwd
```

→ To transfer a file from one machine to other machine

eg:- Transfer newfile in /opt into other machine

```
scp newfile 192.168.10.90:/opt
```

```
scp sourcefile pvt-ip:/destination
```

→ To copy a directory

```
scp -r mydir 192.168.10.90:/opt
```

→ To download a file/directory from other machine.

```
scp 192.168.10.90:/opt/newfile /opt
```

```
scp -r 192.168.10.90:/opt/mydir /opt
```

We are downloading from 192.168.10.90 to our machine

ssh passwordless connections

server-1

```
/root/.ssh
id_rsa
id_rsa.pub
authorized_keys
```

server-2

```
/root/.ssh
id_rsa
id_rsa.pub
authorized_keys
```

root

ssh-keygen

→ To generate keys as root, login as root and

```
ssh-keygen
```

→ It must be copied into authorized_keys.

→ copy id_rsa.pub key only, don't share private key.

→ change ~~Challenge Response Authentication~~ KbdInteractive Authentication yes

To copy pub key into other machine

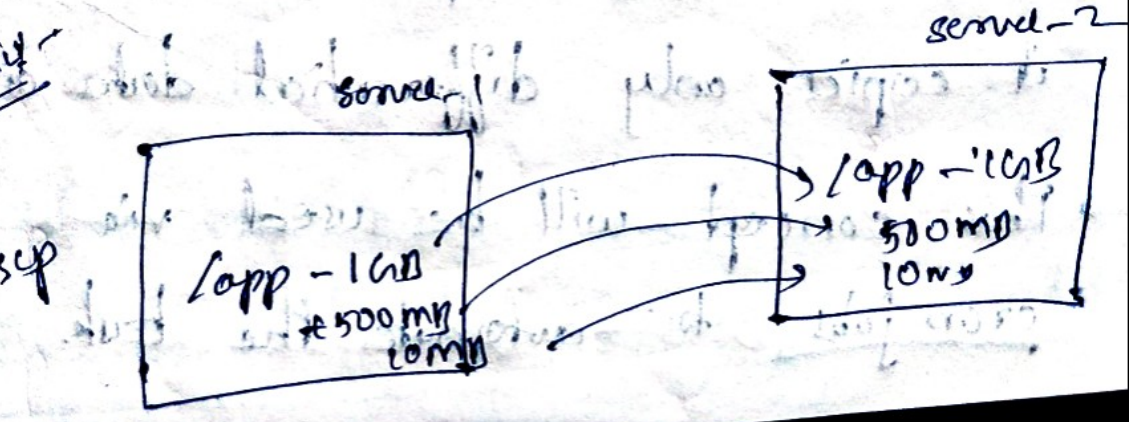
```
ssh-copy-id -i id_rsa.pub 192.168.10.90
```

```
ssh-copy-id -i key-name prt-ip-server2
```

(or)

```
ssh-cop-id prt-ip-server2
```

Scenario



→ I have a production server-1, and I have to copy a /app directory for backup into another server-2,

→ I copied 10GB on first day, and next day I have to add more 500MB, for that I have to do from starting, this is wasting resources and time,

→ I just want to add only 500MB of data into the same directory, for this problem we have Rsync (Remote Synchronization)

Rsync :-

→ First time it copy copies the /app directory,

→ Next time it compares the data and it copies only differential data only.

→ This concept will be used via cron jobs to automate the task.

step-1:
Make ssh passwordless connection.

step-2:
Create a directory /app

step-3:

```
rsync -avz -e ssh /app pvt-ip:/opt
```

z → zip
v → verbose
a → archive
-e → encrypt

In this way data is zipped and tranfered and unzipped other side.

```
--dry-run
```
→ it will just show what it is following.

To delete extra files:

```
rsync -avz -e ssh /app pvt-ip:/opt --delete
```

To copy ssh keys, in the server you want to copy. in that server edit

vim /etc/ssh/sshd_config

→ KbdInteractiveAuthentication no → yes

→ restart the ssh service

→ ~~Navigate~~

→ Now connect to the main server navigate to .ssh folder and do

ssh-copy-id -i id_rsa.pub user@pvt_ip

SOFTWARE MANAGEMENT

→ To manage the s/w in linux, we have two utilities

① RPM → RedHat Package Manager.

② Yum → Yellowdog Updated Modified (or) DNR → DANDIFIED YUM

RPM :-

→ It is a s/w management tool.

→ It is used for installing, uninstalling, verifying, quering and updating s/w packages

→ The s/w packages mostly available in .rpm format.

(i) i386 (32 bit)

(ii) x86_64 (64 bit)

(iii) We have neutral type of package that can be installed on both the 32bit & 64 bit. ie. noarch

noarch → no specific architecture

→ From RedHAT -7 onwards 32bit is depricated, we only have 64bit.

→ To check architecture

`uname -m` (or) `arch`

→ To check all the s/w's installed

```
rpm -qa
```

→ to check only the package/s/w by specific name

```
rpm -q <package_name>
```

eg:- `rpm -q tomcat9`

(or)

`rpm -qa tom*`

Installing any package

```
rpm -ivh openjdk-11-jdk
```

-i → install

v → verbose

h → hash progress ###

For dry-run

```
rpm -ivh openjdk-11-jdk --test
```

To update the package

```
rpm -Uvh openjdk-11-jdk
```

```
rpm -Uvh package_name
```

U → update

→ Patching is nothing but updating.

Uninstalling the package

```
rpm -evh openjdk-11-jdk
```

e → erase

How to fix the corrupted package

→ To find the package of particular command.

```
which date
```

, it shows the location of the date command script

/usr/bin/date

to find date package

```
rpm -qf /usr/bin/date
```

Scenario:- lets corrupt mount command and fix it

step-1:- which date which mount

o/p :- /usr/bin/date o/p :- /usr/bin/mount

step-2:-

cp /usr/bin/date /usr/bin/mount.

Then date and mount command will give same o/p as date.

To fix it:-

→ find the package of mount

```
which mount
rpm -qf /usr/bin/mount
```

→ It shows the package, with that package reinstall

```
rpm -ivh package_name --force
```

→ To understand the role of any package which is installed in your server

`rpm -qi package_name`

To find the configuration files of each package

`rpm -qlc sshd`

`rpm -qlc package_name`

→ To find the documents related to that package.

`rpm -qld package_name`

eg:- `rpm -qld sshd`

YUM :- Yellowdog Update Modified.

→ Now yum is redirected to dnf

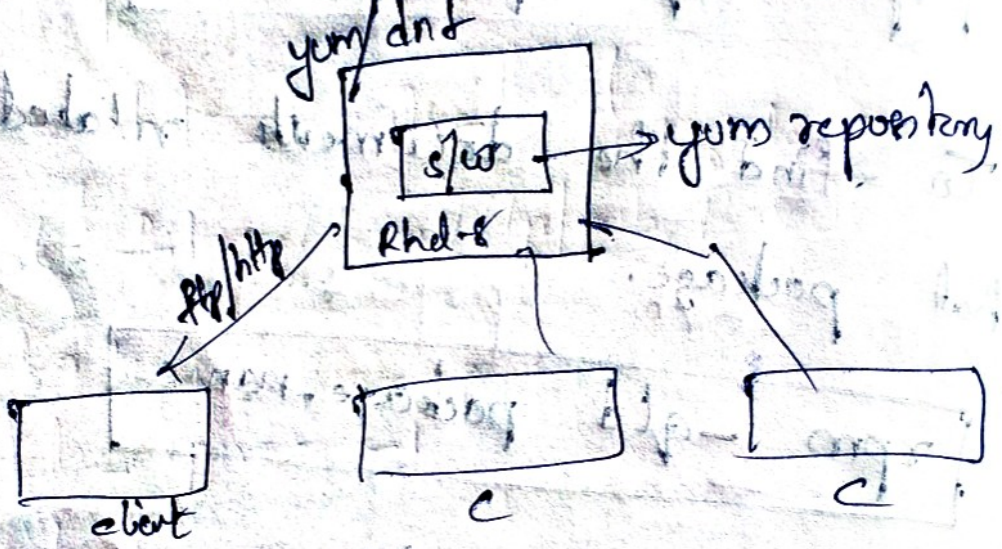
→ Create a server, on that create a folder and keep all your packages inside it, it is called yum Repository.

→ It acts as a centralized repository.

→ Any client can download the package by connecting to it.

→ It uses ftp/http to transfer packages to client. by just typing

```
yum package_name
```



Configuring Yum server in RHEL 6/7

step-1: Here we are using ftp to transfer packages to client.

→ first install vsftpd

```
rpm -ivh vsftpd
```

step-1:

Now go into pub folder in vsftpd

```
cd /var/ftp/pub
```

dump all the packages in the pub folder to transfer publicly ; any name.

mkdir rhel in /pub

Now copy all the packages from DVD to rhel.

```
cp -rvf /media /var/ftp/pub/rhel
```

↓

here my dvd is mounted

step-2:

Now create configuration file, it should in /etc/yum.repos.d

```
cd /etc/yum.repos.d
```

→ `vim <any_name>.repo`

eg:- vim my.repo

In that you have to write BaseOs and Appstream. , name, baseurl of your file

metadata = -1 (it won't change the location when you restart the machine, gpgkey(for the verification of the package by redhat.

[BaseOs]

name = rhel8_baseOs_local

baseurl = file:///var/ftp/pub/rhel8/BaseOs

enabled = 1

metadata_expire = -1

gpgcheck = 1

gpgkey = file:///var/ftp/pub/BaseOs/RPM-GPG-KEY-redhat-release

[Appstream]

name = rhel8_Appstream_local

baseurl = file:///var/ftp/pub/rhel8/Appstream

enabled = 1

metadata_expire = -1

gpgcheck = 1

gpgkey = file:///var/ftp/pub/AppStream/RPM-GPG-KEY-redhat-release

:wq!

step-IV:

```
yum clean all
```

→ To find the repositories

```
yum repo list
```

step-V: ~~make~~

In order to share files with ftp, make following changes:

```
vim /etc/vsftpd/vsftpd.conf
```

anonymous_enable = ~~Yes~~ no → Yes

step-VI: start the ftp service

```
systemctl start vsftpd
systemctl enable vsftpd
```

step-five:

firewall-cmd --add-service=ftp --permanent

firewall-cmd --reload

Now access it from browser as

ftp:// public ip

Your gpgkey will be available in the location

/etc/pki/rpm-gpg

step-six:

In client machine create repo. as copy and paste from yum server

cd /etc/yum.repos.d

vim client.repo

[Base Os]

name = rhel8_baseOs_ftp

baseurl = ftp://public_ip_yum_server/pub/rhel8/BaseOs (base from browser for BaseOs)

enabled = 1

metadata_expire = -1

gpgcheck = 1

gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[AppStream]

name = rhel_bappstream_ftp

baseurl = ftp://public_ip_yum_server/pub/rhel8/AppStream

enable = 1

metadata_expire = -1

gpgcheck = 1

gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

:wq!

→ yum clean all

→ yum repo list.

→ To see all the packages

```
dnf list
```

→ To reinstall a package

```
dnf reinstall package_name
```

→ To remove a package

```
dnf remove package_name*
```

→ To see the package information

```
dnf info package_name
```

PROCESS MANAGEMENT

→ A linux process is a program running in the linux system. Depending on linux distributions, it is also known as service. In Linux community however, a linux

process is called daemon.

→ Every process will consume some resources of your system.

→ Each process has a PID.

Types of process

① Interactive Process :-

→ These are the process that are invoked by a user and can interact with the process.

eg:- vim is a process and we are typing i, dd, yy, :wq etc.

② System process or Daemon :-

→ These are the process belonging to system.

eg:- reload, restart, start, stop etc.

③ Automatic or batch :-

→ These are the process which can start on its own. eg:- cron jobs.

Parent and child Process :-

→ The process which starts or creates another process is called Parent process and the one which got created is known as child process.

→ Every process will be having a parent process except init/systemd process,

→ The PID of init/systemd is 1.

eg:- ls -l | grep

parent → child

→ whenever parent is killed, child will also be killed.

→ In some cases parents are killed but childs are not killed such processes are called Zombie process

→ whenever you find zombie process kill them, or else it creates performance issue

→ To find the process

```
ps
```

→ All the process which are running in foreground and background

```
ps -a
```

→ To see what others user is doing

```
ps -U myuser
```

PID	TTY	TIME	CMD
2129	?	00:00:00	systemd

? → important process, that system may don't kill them

→ To find the status of process

```
ps -elf
```

S → means sleeping / stand by

R → running

Z → zombie

I → internal kernel process

→ To know each process is consuming how much memory and cpu

```
ps -aux | more
```

→ To list all the kill signals.

```
kill -l
```

→ To kill any process

```
kill -signal-no PID
```

eg:- kill -9 PID

9 is used to kill the process

Setting up the priority to a process

→ whenever you give high priority it is getting high cpu, memory, time etc and performance will be high.

→ The priority depends on nice value, range -20 to 20.

-20 → highest priority

+20 → lowest priority.

→ The default value is 0.

→ To setup nice value

```
nice -n -20 to 20 process_name
```

eg:- nice -n 5 cat

→ To change the nice value of cat

```
renice nice_value PID
```

eg:- renice -5 2398

(or)

renice -5 cat

How to monitor cpu, memory, i/o, n/w

→ To monitor CPU

```
ps, sar, lscpu, /proc/cpuinfo
```

continous monitor of cpu

```
sar 1
```

To stop your sar after 3 counts

```
sar 1 3
```

To see the details of your cpu

```
lscpu
```
(or)
```
cat /proc/cpuinfo
```

To see info about memory

```
free, swapon -s, vmstat
```

To see continous info about vmstat

```
vmstat 1
```

upto 3 counts

```
vmstat 1 3
```

To see how much memory is

```
vmstat -s
```

To monitor i/p, o/p

fdisk, parted, df -h, iostat, lsblk, lsusb, lspci

To see a specific device i/p, o/p info

iostat -d sda 1 3

To monitor network

ifconfig, ethtool, mii-tool, ping, netstat, route

To see n/w gateways

netstat -rn

To see the interfaces

netstat -i

top :- It shows the process, cpu utilization and user.

Enter

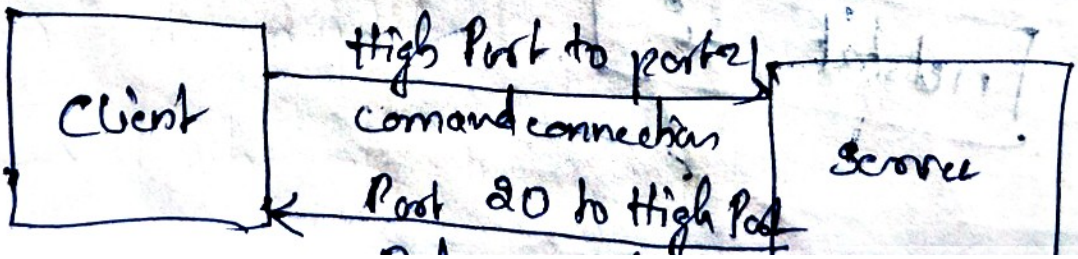
uptime , user , load average

FTP - Server

→ File Transfer Protocol is a n/w protocol used to transfer files from one host to another host over a TCP-based n/w, such as internet.

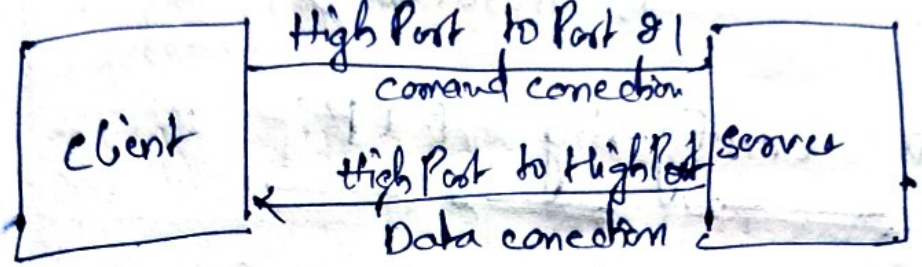
Active FTP :-

Active FTP connection mode is where comand connection is initiated by the Client, and the Data connection is initiated by the server.



→ Passive FTP :-

the server acts entirely passively as the comand connections and the Data connections are both initiated by the client.



→ tcp connection is secure where as udp is not secure.

→ Ftp uses 21 port.

★ Profile of ftp Server

→ Usage : Ftp is used to uploading and download the files

→ Limitations : Directory can't be uploaded or downloaded.

→ Package : vsftpd

→ Port no : 21

→ Config files : /etc/vsftpd/vsftpd.conf

/etc/vsftpd/user_list

/etc/vsftpd/ftpusers

→ Home directory :- /var/ftp (which will be created only when the package is installed)

steps to configuring FTP :-

step-1 :- Install the package

```
yum install -y vsftpd
```

step-II :- Navigate to pub directory and keep your data

```
cd /var/ftp/pub
```

keep some data, for eg :-

→ mkdir mydata

→ cd mydata

→ touch file{1,2,3,4}

step-iii) → check the service vsftpd is running

```
systemctl status vsftpd
```

```
systemctl start vsftpd
systemctl enable vsftpd
```

step-iv) → Navigate to vsftpd.conf and

change anonymous_enable = Yes

```
vim /etc/vsftpd/vsftpd.conf
```

:wq

step-v) → Allow firewall

```
firewall-cmd --add-service=ftp --permanent
firewall-cmd --reload
```

→ Now access it through browser

ftp://public-ip

Connect the Ftp server from another linux machine

step-I:- install ftp

```
yum install -y ftp
```

step-II:- To connect the ftp server:

```
ftp pub/pvt-ip of server
```

enter username : ftp
password : just give ↵ (or) any

step-III:- Then go to pub directory and download the files you want as below

```
get file_name
```

Note:- From which location you are connecting the ftp server, in that location only the file downloaded.

→ To download multiple files

```
mget files_names
```

or

```
mget *
```

If it is prompting yes/no, to disable/enable just type

```
prompt
```

→ To exit from the server just type

```
bye
```

Make public to upload the data into ftp server

step-1: In ftp server

go to pub directory

create any directory as for uploads

```
mkdir uploads
```

step-2: Give write permissions to uthey

```
chmod 777 uploads
```

step-iii: Now change the group of directory to ftp group.

```
chgrp ftp uploads
```

step-iv: Now go to vsftpd.conf

```
vim /etc/vsftp/vsftp.conf
```

change `anon_upload_enable = yes`

→ Restart the service

→ Now anybody can upload the files in ftp server

→ To upload the file

```
put file_name
```

→ If it gives error, "could not create file" it means SELinux is not allowing that

→ SELinux is used to protect the server by outsiders to write.

→ To ~~enable~~ disable selinux in ftp server

→ To see the selinux is enforced

```
getenforce
```

→ To remove enforce

```
setenforce 0
```

Note:- Removing SELinux is a bad process

To give permissions to public without removing selinux

→ See the content of directory

```
ls -ldz uploads
```

→ change the public_content_t to public_content_rw_t for the directory

```
chcon -t public_content_rw_t uploads
```

→ Now change ftpd_anon_write =on
ftpd_anon_full_access =on

```
getsebool -a | grep ftp
```

```
setsebool -P ftpd_anon_write on
setsebool -P ftpd_anon_full_access on
```

→ Now you can upload the files into ftp server

→ To upload all the files

```
prompt
mput *
```

→ You can connect ftp server with any user.

→ To unlock root user, go to ftp server

```
vim /etc/vsftpd/ftpusers
```

put # before the root as #root,

→ go to vim /etc/vsftpd/user_list

#root,

→ restart the service

SAMBA SERVER

→ This server is used when the client is windows,

→ If you want transfer data between window and linux.

Profile :-

Usage : used for sharing file & directories in the n/w between different platforms like Linux - Window

Package : SAMBA, SAMBA-common, SAMBA-client

Daemons : smbd, nmbd.

Port no : 137 (net bios-ns {name server}),
138 (net bios-dgm {Datagram}),
139 (net bios-ssn {session service}).
445 (Microsoft-ds {dist sys}).

★ Kernel File system : CIFS (Common Internet File System)

Config file : /etc/samba/smb.conf.

Sample file : /etc/samba/smb.conf.example

Steps to configure SAMBA Server

step-I :- Install all samba related packages

```
dnf install -y samba*
```

step-II :- Create a director to share with windows clients

```
mkdir /samba
```

```
chmod 777 /samba
```

step-III :- Give SELinux permission to the directory

```
chcon -t samba_share_t /samba
```

step-IV :- Create a user and password and attach to the samba server and

that user shouldn't have other logins

```
useradd U1
```

```
usermod -s /sbin/nologin
```

(or)

```
usermod -s /sbin/nologin
```

→ Now attach to samba server as

```
smbpasswd -a U1
```

→ `pdbedit -L` → it is used to see the samba users.

step-V:- Now copy some line from /etc/samba/smb.conf.example to /etc/samba/smb.conf

→ copy last lines from that file and paste at bottom in another file.

These lines are below :-

[public]

comment = Public Stuff

path = /home/samba

public = yes

writable = no

printable = no

write list = +staff

edit as below

[Myshare]

comment = It's my stuff.

path = /samba : give path of your directory.

public = no

valid users = u1, u2 u3 myuser

writable = no yes

printable =

write list = +u1

hosts allow = 192.168.10.

(or)

give particular ip_address

:wq

→ To test the configuration file

```
testparm
```

```
systemctl enable smb nmb --now
```

step-VI)

```
firewall-cmd --add-service=samba --permanent
firewall-cmd --reload
```

Now connect Samba server from your laptop

→ Right Click My computer

→ Select "Map n/w drive"

→ Folder: `\\pub-ip\Myshare`

share name from config file

→ Username: U1

→ password:

(or)

from run terminal → \\ip-address

To connect samba service from Linux

step-I:- Install samba-client cifs-utils

```
yum install -y samba-client cifs-utils
```

step-II:- Creat one directory eg:

```
mkdir smbcl
```

Now mount it with cifs file system

```
mount -t cifs //pvb-ip/Myshare /smbcl
      -o user=u1
```

Permanent mounting

```
//ip-add/Myshare /smbcl cifs username=u1,password=Redhat,u 0
```

To verify

To connect samba server

```
smbclient -L 192.168.1.175 -U username
```

To connect from Linux machine

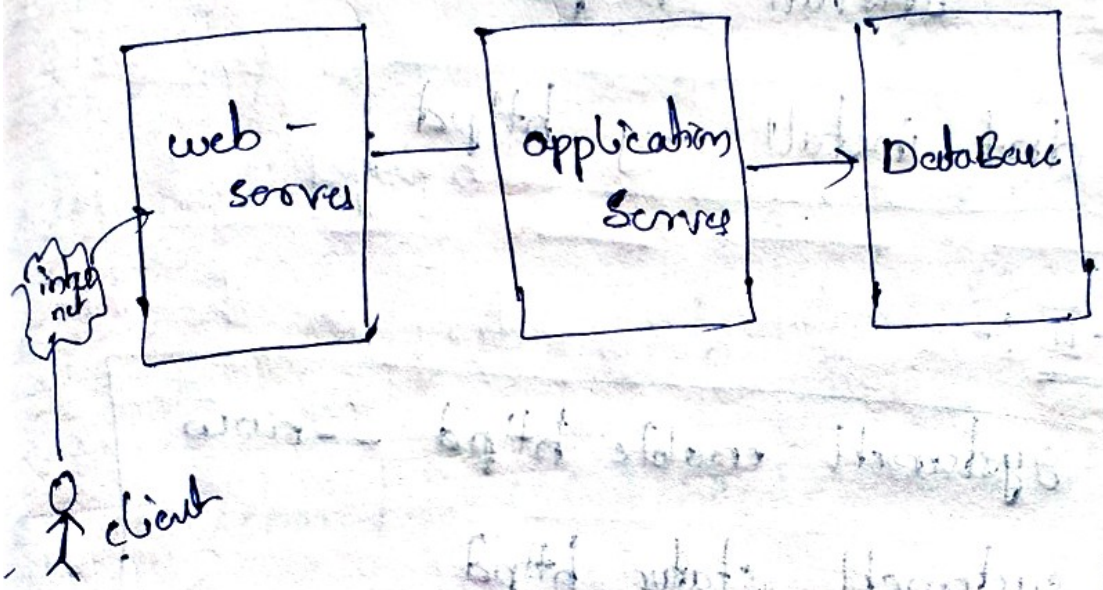
```
smbclient //192.168.10.80/smbcl -U username
```

smbcl → directory name

WEB SERVER

→ It is a physical machine or also a s/w (apache, nginx) acts as a webserver.

→ We have 3-tier architecture of a website



Profile for Apache Server

Usage : Hosting a website.

Package : httpd

Port : 80/http

Config file : /etc/httpd/conf/httpd.conf

Sample Config File : /usr/share/doc/httpd/httpd-vhosts.conf

Configuration directory : /etc/httpd/conf.d

Document root : /var/www/html

Daemon : httpd.

Steps to configure a simple web server

step-I :- Install httpd

```
dnf install -y httpd
```

step-II :-

```
systemctl enable httpd --now
systemctl status httpd
```

step-III :- Allow firewall

```
firewall-cmd --add-service=httpd --permanent
firewall-cmd --reload
```

firewall-cmd --add-port=80/tcp --permanent

step-IV :- Add website source code in html.

```
cd /var/www/html
```

```
vim index.html
```

```
<html>
<h1> MY WEBSITE </h1>
</html>
```

step-V :- Copy the configuration file.

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

step-VI :- Edit the configuration file.

```
vim /etc/httpd/conf.d/chakram.conf
```

```
<VirtualHost pub-ip-of your-server:80>
    ServerAdmin root@rhlinux8.vpts.com
    DocumentRoot "/var/www/html"
    ServerName "www.websitename.com
                or rhlinux8.vpts.com
```

ErrorLog "/var/log/httpd/chaleram-error_log"

CustomLog "/var/log/httpd/chaleram-access_log" common

</VirtualHost>

:wq

```
systemctl reload httpd
```

→ To host multiple web pages

step-2:- Create a directory in the html location and copy index.html page.

```
cd /var/www/html
```

→ mkdir webpage2
→ cd webpage2
→ vim index.html

<html>

<h1> MY WEBSITE PAGE-2 </h1>

</html>

:wq

step-1: vim /etc/httpd/conf.d/chakram.conf

for pages we give "Alias /2" means second page.

→ for third page "Alias /3"

<VirtualHost @pub-ip : 80 >

ServerAdmin root@rhlinux8.vpls.com

DocumentRoot "/var/www/html"

Alias /2 "/var/www/html/webpage2"

ServerName rhlinux8.vpls.com

(or)

www.website.com

ErrorLog "/var/log/httpd/chakram-error_log"

CustomLog "/var/log/httpd/chakram-access-log" common.

</VirtualHost>

systemctl reload httpd

→ From browser access ip_address:80/2 → for second page

Create a local DNS Server

vim /etc/resolv.conf

nameserver 192.168.10.80

:wq.

If you want to connect website you need graphics, command line won't display website for that, install tigervnc in client server

```
dnf install -y tigervnc-server
```

→ firewall-cmd --add-service=vnc-server --permanent

→ firewall-cmd --reload

→ set password for vnc

```
vncpasswd
```

→ start vnc server

```
vncserver
```

It gives a number to connect graphically.

Note:- If you want to give access to

graphical interface then run one more time `vncserver`

It gives number 2.

→ Now go to browser and download "VNC viewer"

→ In VNC, give ip_address:1

give password

→ Click on activities → click on browser firefox and type website_name.ai

mobinux8.vpls.com/2

Redirecting to another website from my website

→ People are connecting to your website and you are directing to another website just add one line in configuration file.

vim /etc/httpd/conf.d/chakram.conf

Add below Alias line as

```
Redirect /3 "http://www.sudarshan.com"
```

:wq.

systemctl reload httpd

To acces pub_dpt/3

Virtual Web Hosting

→ On a single ip_addrss we can host multiple websites

Port based :- webhosting

192.168.10.80:80 → 1st website

192.168.10.80:8080 → 2nd "

192.168.10.80:8081 → 3rd "

You can host max 4 websites on a single ip_addrss

→ You must create different index.html file in new folder and new conf file.

step-1:-

Navigate to /var/www/

→ cd /var/www

→ mkdir second-website

→ cd second-website

→ create index.html

→ Now go to /etc/httpd/conf.d and create another conf file for this second website

vim second-website.conf

Listen 8080.

<VirtualHost 192.168.10.80:8080>

ServerAdmin → same

DocumentRoot "/var/www/second-website"

ServerName → same

ErrorLog "/var/log/httpd/second-website-error_log"

CustomLog "/var/log/httpd/second-website-error_log" common

</VirtualHost>

→ Pass a Listen parameter in conf file as above.

```
systemctl reload httpd
```

firewall-cmd --add-port=8080/tcp --permanent

firewall-cmd --reload

Name base web hosting

→ This is purely dependent on DNS.

192.168.10.80 : 80 → chaleram.com

192.168.10.80 : 80 → sudarshan.com

192.168.10.80 : 80 → hpsy.com

step-2:-

cd /var/www

mkdir /namebased

cd /namebased

create index.html

step-2: Create a conf. file

cd /etc/httpd/conf.d/

vim namebased.conf

NameVirtualHost 192.168.10.80:80

<VirtualHost 192.168.10.80:80>

ServerAdmin root@topsy.vpk.com

DocumentRoot "/var/www/namebased"

ServerName topsy.vpk.com

ErrorLog "/var/log/httpd/namebased-error_log"

CustomLog "/var/log/httpd/namebased-access_log" common

</VirtualHost>

→ reload httpd

step-3:

Configure reverse local zone in DNS and forward local zone in DNS

cd /var/named/

vim my.fz

→ Add a line as

topsy A 192.168.10.80

and vim my.rlz → add a line as

80 PTR topsy.vpts.com.

DNS

Domain Name System/service

Setup Nameserver:-

Package : bind-utils, bind

Service Name : named

Port : 53/tcp, 53/udp

Configuration file : /etc/named/named.conf (or) /etc/named.conf

related config : /etc/named

log file : /var/log/messages

step-2 :- Install the package

```
yum/dnf install -y bind bind-utils
```

step-II:- Edit configuration file

vim /etc/named.conf

options {
listen-on port 53 {127.0.0.0; cp-add; };
allow-query {localhost; 192.168.1.0/24; };
(or) any
n/w address

:wq.

step-III: Create zone database

(i) forward look up

(ii) Reverse look up

vim /etc/named.conf

at bottom there are zone keyword.

edit as below

zone "chakram.com" IN {
type master;
file "fwd.chakram.db";
allow-update {none;};
};

zone "1.168.192.in-addr.arpa" IN {

type master;

file "rev.chakram.db";

allow-update {none;};

};

write ip_addr in reverse

your domain_name

step-iv) - Now create forward lookup zone and reverse lookup zone.

```
cd /var/named
```

#vim fwd.chakram.db

domain_name

email address

```
$TTL    86400
@  IN  SOA  master.chakram.com.  root.chakram.com.
        2021020801  ; Serial
        3600        ; Refresh
        1800        ; Retry
        604000      ; Expire
        86400       ; Minimum TTL
)
@    IN   NS   master.chakram.com.
@    IN   NS   slave.chakram.com.
@    IN   A    192.168.1.175
@    IN   A    192.168.1.176
master    IN   A    192.168.1.175
```

slave IN A 192.168.1.176

:wq

forword look up zone :- It resolves name into ip-address

Reverse look up zone :- It resolves ip into name.

vim rev.chakram.db

```
$TTL 86400
@ IN SOA master.chakram.com. root.chakram.com. (
        2008080001  ; Serial
        3600        ; Refresh
        1800        ; Retry
        604800      ; Expire
        86400       ; Minimum TTL
)
@ IN NS master.chakram.com.
@ IN NS slave.chakram.com.
@ IN PTR chakram.com.
master IN A 192.168.1.175
slave IN A 192.168.1.176
175 IN PTR master.chakram.com.
176 IN PTR slave.chakram.com.
```

step-5:- check the conf file.

```
named-checkconf /etc/named.conf
```

```
named-checkzone chakram.com /var/named
/fwd.chakram.db
```

and

```
" " /rev.chakram.db
```

step-6:- Restart the service.

```
systemctl enable named --now
```

step-7:- Add nameserver address in DNS server via nmtui

DNS servers 192.168.1.175

192.168.1.1

DNS debugging tools

nslookup

dig

host

To connect another machine to your name server machine

→ login into another machine.

→ nmtui

→ Address 192.168.1.5/24 → address of same machine

→ Gateway 192.168.1.1

→ DNS servers 192.168.1.125
192.168.1.1

→ Search domains → chakram.com

→ Name server address into all the machines, instead of doing this add into DHCP

DNS zone :- A DNS zone is a collection of dns records/resource records.

They are :-

(i) forward lookup

(ii) reverse lookup

Resource records :- These records provide info about a specific object

A → address mapping record; it maps a domain name into ipv4address

AAAA → it mapps a domain name into ipv6 address

MX → mail exchanger record.
→ This info is used by SMTP to route emails to proper hosts

CNAME → Canonical Name Record.
→ It is used when you want alias our domain name to an external domain name.
eg :- www.chakram.com
chakram.com

PTR :- Reverse Lookup Pointer Records
→ used to lookup domain name based on ip address

NS :- Name Server Record.

→ This record is responsible for answering DNS queries.

→ It specifies an authoritative name server for given host.

SOA :- Start Of Authority.

→ It specifies core info about DNS zone i.e including primary name server, email address of the domain admin, domain serial no, and several times relating to refreshing the zone

TTL :- Time to live.

→ It is based on time, dns server will refresh the DNS record.

→ It is the amount of time the record is allowed to be cached by resolver

private nameserver :- It will resolve in your domain only, not on internet.

master dns :- stores authoritative records for your domain

slave dns :- relies on master dns server for data

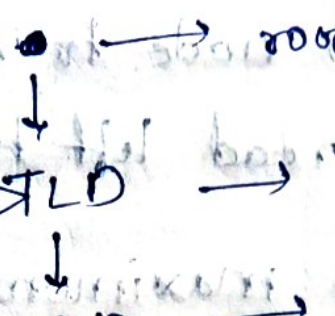
caching only dns :- stores recent requests (or) dns queries like a proxy server.

forwarding only dns :- refers all your dns requests to other dns servers

dns namespace :-

→ It is a structure of dns database with an inverted tree with root node at top.

→ Each node has a label and the root node has a null label, written as " "



→ Top level domain

→ second level domain

→ sub domain / Hostname

gTLD → generic TLD

ccTLD → Country Code TLD.

→ root nameserver, TLD, SLD, Hostname are separated by a dot ".".

TLD :- .com, .in, .info, .org, .edu, .gov, .uk, .au, .io

GTLD :- .com, .org, .gov, .edu, .info, .net

ccTLD :- .in, .uk, .us, .au, etc

→ www.chakram.com

Hostname . SLD . GTLD

→ domain name :- A domain name is a

sequence of labels from node to the root seperated by dots (.) read left to right.

→ The name space has maximum depth of 127 levels; domain names are limited to 255 char in length.

→ The authority for the root domain and gTLD lies with ICANN (Internet Corp. for Assigned Names & Numbers).

ccTLD's are delegated to individual countries for administration purpose.

→ All DNS servers fall into one of 4 categories, they are Recursive Resolver, Root Nameservers, TLD name servers and authoritative nameservers.