



Cloud Computing

Linux World Summer Training 2017

DAY 1

OS : Linux : RHEL : RedHat enterprise Linux

Version: RedHat 7.3

How much RAM to give to any virtual machine is decided dynamically.

<i>Available RAM</i>	<i>How much to give</i>
2 GB	1.5 GB
4 GB	3 GB
8 GB	6 GB
12 GB	8 GB
16 GB	8 GB

h/d – real,virtual

physical h/d – Bare Metal Setup

port forwarding -

<https://www.howtogeek.com/66214/how-to-forward-ports-on-your-router/>

ADVANTAGES OF CLI OVER GUI -

1. Remote control or login
2. To close or terminate any program using CLI.

SOME USEFUL COMMANDS –

✚ There is not any file without any folder. Every file is inside a folder.

✚ To find directory of a program –

```
>> which firefox
```

```
>> ls program_name – tells whether the program exists or not.
```

✚ Firefox stores its data in –

- Open firefox in `/usr/bin/` directory.

```
>> gedit firefox
```

- Go to line number 65, there you can find the path.

First go home using `cd` and then Go in directory

```
$HOME/.mozilla/firefox ( This is the same path as found on line number 65.)
```

✚ GUI of Linux is said **gnome**.

✚ **Which command opens terminal ?**

This command can be used in remote access of a terminal.

```
>> gnome-terminal
```

This command opens a terminal.

- ✚ The program which actually runs any command is called **shell**. By default, we are running bash shell.

cmd run : shell : bash

- ✚ We can create our shell in python.

- ✚ Creating a shell loop-

```
>> for ( ( i=1; i<=10; i++ ) )  
> do // for starting the loop.  
> gnome-terminal  
> done // at the end of the loop.
```

- ✚ Program (in hard disk) becomes a process when it loads in RAM. Every process has an id `process_id`.

- ✚ This command gives all the processes running in the system.

```
>> ps -aux
```

Can be used for getting process id of a running process.

- ✚ We can filter any output of a command using pipe "|".

```
>> ps -aux | grep gnome-terminal
```

grep is for searching gnome-terminal in the output of the command `ps`.

- ✚ We can terminate any process using `process_id` which can be found using above command.

```
>> kill process_id
```

will terminate the program corresponding to that `process_id`.

DAY 2 (rpm, yum, wireless connection)

- ✚ Linux user : root(administrator)

Login with root user

- ✚ Either interact with GUI or CLI

gnome is running program for GUI.

bash (shell) is for CLI.

- ✚ CLI is faster than GUI because we're directly running command.

- ✚ S/w is a collection of files. It's an archive. When we install any software it's actually extract all the files from the software and copies them to different locations of computer according to their uses.

- ✚ Finds the software related to file_name or command. Gives the name of software. It can be used to find whether a software is installed on the system or not and the version of the software.

```
>> rpm -qf /usr/bin/firefox
```

OR

```
>> rpm -q -f file_name
```

AND

```
>> rpm -q firefox
```

-q is for doing query.

rpm (RedHat package manager) is software extension in RedHat.

'-' is used for options/switches.

✚ manual for a command.

```
>> man command_name
```

✚ Gives all the softwares installed in the system.

```
>> rpm -qa
```

✚ Removing a program (erase)

```
>> rpm -e firefox
```

Removes a software, not a file.

rm is used for file deletion.

✚ If we want to find all the files that come from the Firefox software when it was installed and in which directories it copied the files, then use this command -

```
>> rpm -ql firefox
```

✚ Redirects the output of the rpm command to less through pipe.

```
>> rpm -ql firefox | less
```

```
>> rpm -ql firefox | more
```

✚ All the commands that you have run after opening the shell.

```
>> history
```

✚ Every command is a program file.

✚ `rpm -ql firefox` command's output should give this `/usr/bin/firefox` path also. Because this `firefox` command is also created by firefox software when it was installed.

✚ If we are running a command, and output gives `command not found`, that doesn't always mean command is wrong, it may be possible reason that software is not install related to that command, so it is saying `command not found`.

✚ Repository is the location where softwares are stored and we can download software from that location.

✚ To install a software from a dvd or iso file-

Mount iso file > open > packages > find software

Then, to install that there are two methods -

1st is to use GUI by double clicking.

2nd method is to use CLI.

Using CLI -

To install software run

```
>> rpm -i firefox.....[Full Name of Software]
```

Use tab-completion.

After installing, verify using `rpm -f firefox`.

- ✚ For detailed information when downloading or installing the package use these options -

```
>> rpm -i -h -v firefox... OR -ivh
```

i – install

v – verbose h – hash (shows hash, downloading progress)

- ✚ ls will give its output to the piped command and that command will run on the output of ls.

```
>> ls | grep python
```

grep is for searching, it searches python in the output of ls command.

- ✚ Limitation of rpm -

- If one s/w is depended on another s/w then it can't do install all of them automatically. We have to install every software manually.
- It doesn't do dependency resolution.

- ✚ To install any software, there are mainly two program or commands. Manage Package(software)-

1st rpm

2nd yum

We mostly use yum command to install software.

- ✚ Gives how many repos and software in the repos are available for yum to download.

```
>> yum repolist
```

✚ At least one time we've to tell to `yum` where the software lies, so that it can go to that path and find software there.

- Let's say we've software available in dvd.
- We have to tell `yum` first time that the software is in dvd drive.
- So, we have to configure `yum` to give path of software.

To do that -

➤ Go to directory

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

➤ Create a new file with `.repo` extension

➤ paste the path as

```
[MyRepo]
```

```
baseurl = file:///..path of softwares ..(for local path)
```

```
gpgcheck = 0
```

`path` is the 'path' where to find software.

- Every repo has a unique name which is written in square brackets.
- Set a option `gpgcheck=0` which doesn't checks the signature of files when download and installed.
- `gpg` is a algorithm to check the signatures of files for security purposes.
- It matches the signatures means matches hash values.

✚ `epel` is a software that gives all the repositories addresses where software can be found and are available to download.

```
epel release rpm
```

After installing `epel`, it automatically adds the path in `/etc/yum.repos.d/` directory.

- ✚ `list` can be used to find all the software that matches with the given name. Write `*` to match anything. It finds the software in available repositories for `yum`.

```
>> yum list python3*
```

- ✚ To find the hash value or checksum or signature of file.

```
>> sha1sum filename
```

- ✚ Local files URL starts with – `file:///...`

- ✚ To view the information about connected hard-disk.

```
>> fdisk -l
```

`/dev/sda` here `sda` is the name of the hard-disk.

- ✚ RedHat Linux doesn't support `NTFS` file system.

- ✚ We can't save any data is hard-disk without any file.

- ✚ Every file is inside a directory or folder.

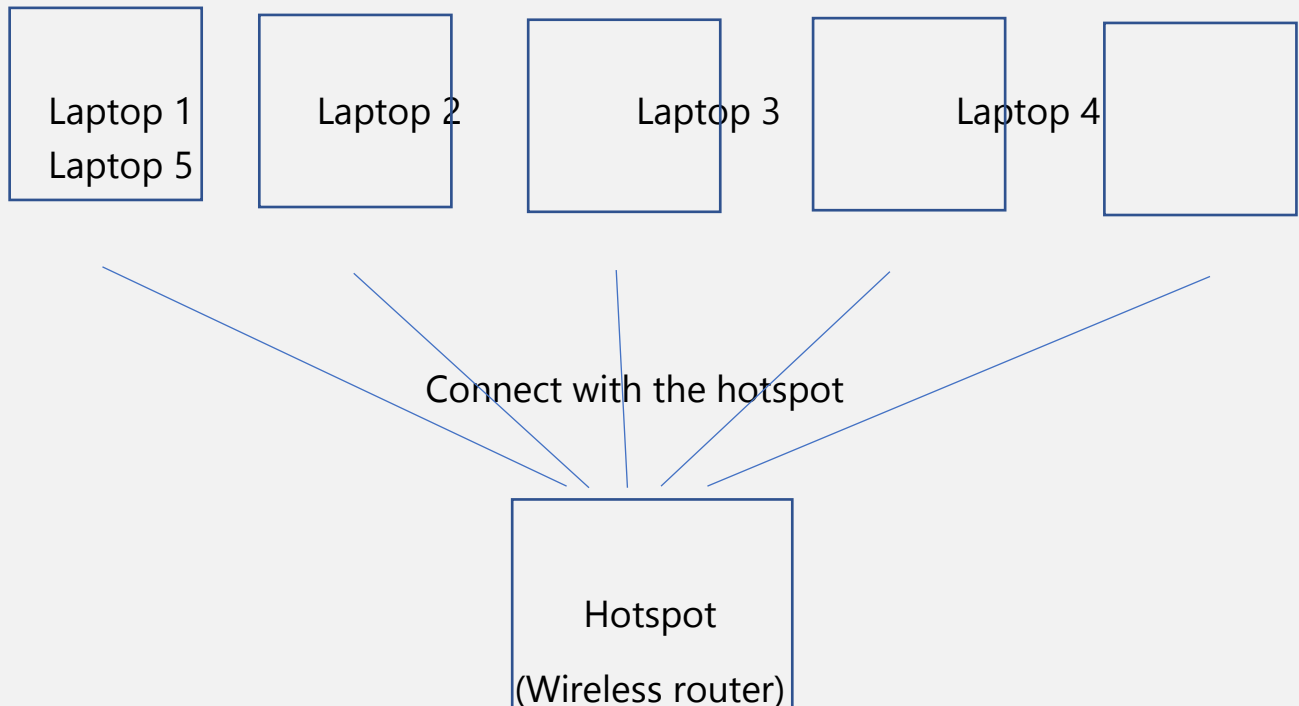
- ✚ There is not any way to remove a file's data. We can overwrite it but can't remove.

Partition → folder → file → data

- ✚ To install NTFS driver

```
>> yum install ntfsprogs
```

✚ To connect laptops wirelessly -



- Make a hotspot and connect all the laptops with that hotspots.
- Change the VM network settings-
 - Change **attached to** from NAT to Bridged Adaptor.
 - Select wireless option in the name.
 - NAT allows to use internet but not between VMs
 - Use bridge to connect to other laptops.
- Disconnect your connection and again connect.
- Now all the laptops are wirelessly connected.

✚ If we have to connect two OS then we required two things-

1st network card

2nd IP address

✚ DHCP server -

Assigns dynamically IP addresses.

✚ Now after connecting laptops, to find the IPs of laptop and to know about connected networks card.

```
>> ifconfig
```

This gives the IPs assigned to all connected laptops.

✚ Now to connect with another laptop use -

```
>> ping ip_address(given by inet in ifconfig)
```

✚ Now to Remote login in another laptop use-

```
>> ssh user_ip_address
```

OR

```
>> ssh username@user_ip_address
```

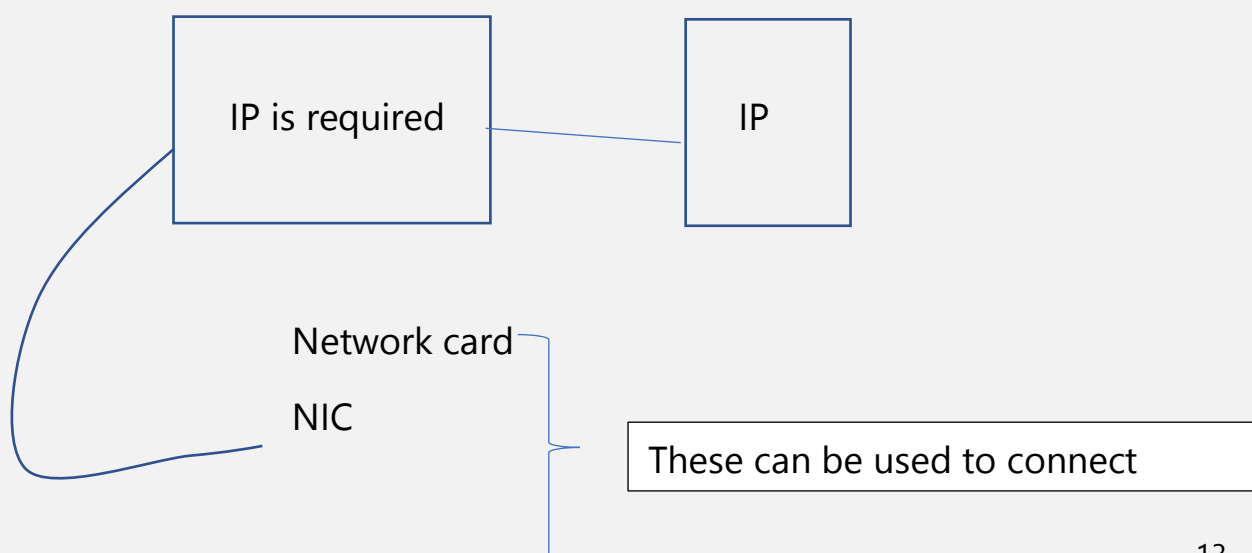
DAY 3 (IP, DNS, SSH, HTTP server)

- ✚ Most secure system in the world is the system which is not connected to internet.
- ✚ A medium is required to connect laptops (physical medium)-
 1. Fiber optical
 2. Wireless
- ✚ Things required to connect systems remotely -
 1. All systems must be connected to same hotspot.
 2. There must be an **IP address** to connect systems.
- ✚ **IP address** is of network card, not of OS like mobile number is the SIM number not the phone's number.
IP address and network cards can be checked by **ifconfig**.

In **ifconfig** output, **enp0s3** is network card name.

```
>> ifconfig enp0s3
```

Shows details about that particular **enp0s3** card.



LAN

Ethernet

- ✚ Spoofing a number. Changes phone number, can use anyone's phone number.

Caller id spoof software

VOIP (Voice Over Internet Protocol)

Two mobile number can be same using some spoofing software.

- ✚ IP can be given manually or auto.
 - Auto IP is given by DHCP server.
 - DHCP → if enabled → provides services

- If we have to change our IP manually –

```
>> ifconfig enp0s3 IP_address
```

Given IP address will be assigned to our network card.

- ✚ Protocol is just the way two systems communicate.
- ✚ Host is our computer.

- ✚ IANA is a community that manages IP address.

Range of every octet is 0-255.

- ✚ Every part of IP is called octet.

Example –

```
192.168.12345 → 192.168.48.57
```

```
192.168 → 192.0.0.168
```

```
1 → 0.0.0.1
```

Every IP given above is a valid IP.

- ✚ Every IP which is in range of 4 bytes is a valid IP.

We think that IP is of 4 octets but that's not true. Instead, every IP is of 4 bytes.

What's the process is that divide 32-bit number in 4 parts and convert it into decimal and show. So, actually we are dividing 32-bit number. So, IP is big 32-bit number that we are dividing just for convenience.

If we provide a 32-bit number in decimal, that's also a valid IP.

IP address is of 1 octet not of 4 octets.

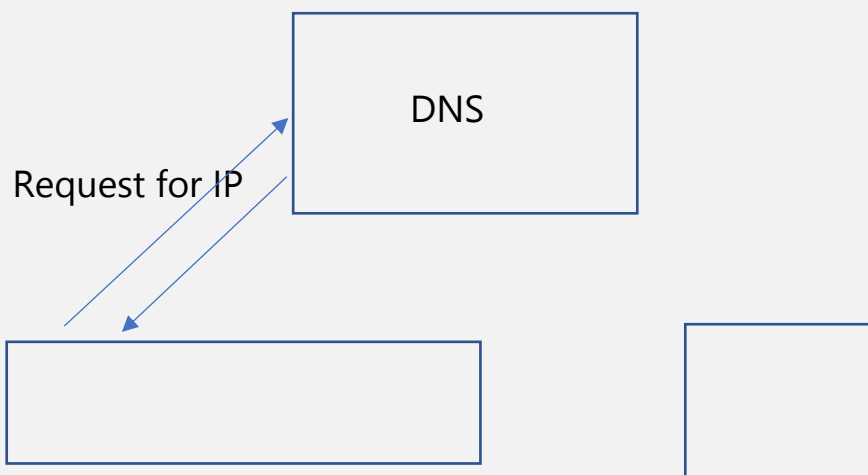
- ✚ To again get back IP from DHCP server if it was changed manually-

```
>> dhclient -v enp0s3
```

Shows two IP, one of your mobile hotspot and other that is assigned to you.

- ✚ DNS server (Domain Name Server or Name Server) –

Gives the IP of website.



we(search for google.com) → google

- ✚ Gives the IP of google.com

```
>> nslookup google.com
```

- ✚ When we try to again run command `dhclient`, it says that it is already running. So to kill the old IP assigned by DHCP server or the running DHCP server, use the number in the output of `dhclient` command.

```
>> kill ...[number]...
```

- ✚ Binary calculator in Linux -

```
>> bc
```

- ✚ How to convert IP into a single octet ?
Suppose we have a IP -

216.58.220.195

$$216 * 2^{24} + 58 * 2^{16} + 220 * 2^8 + 195 * 2^0 = ?$$

This calculated number can be used as an address of website related to that IP.

Can be used to **bypass** the blocked website.

- ✚ Gives the login account name.

```
>> whoami
```

- ✚ Change password for root user.

```
>> passwd root
```

- ✚ Local Host – I am working on my computer
- ✚ Remote Host – I am working on someone's other system from my computer.
- ✚ Same local login and remote login

- ✚ To see the running processes -

```
>> ps -aux | grep firefox
```

- ✚ Information about assigned number of graphics display

```
>> w
```

gdm – graphical display manager

- ✚ To run any command, it requires shell or bash shell.

```
>> echo x
```

```
>> echo $x
```

- ✚ To know about all System Variables -

```
>> set
```

```
>> set | less
```

- ✚ If a system variable has to be use in another program, then it has to be **exported**.

- ✚ To open any graphical program from remote system –
 - When we open any graphical program from the remote user, it says no display specified.
`>> echo $DISPLAY`
 - We have a shell variable `DISPLAY` (system variable)
 - We have to specify display.
 - To set the value of `DISPLAY`
`>> export DISPLAY = :0`
 - Now we can open any graphical program from the remote.

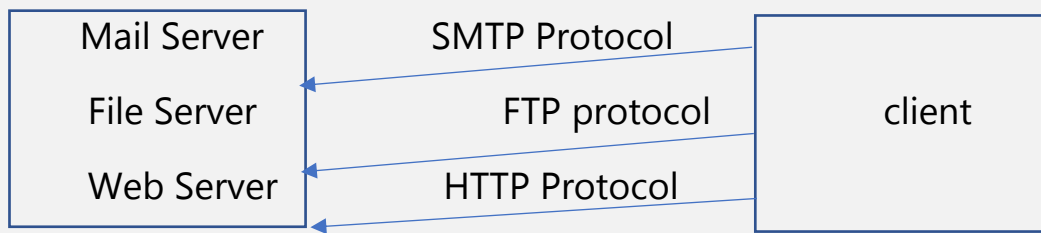
- ✚ To find another IPs connected to our network, we do network scanning.

nmap is a good tool for network scanning.

```
>> yum install nmap  
>> nmap -sP 192.168.1.0-255
```

SERVER AND CLIENT –

- ✚ Configuring any server -
 1. Install software (rpm, use yum)
 2. Configure according to our requirement
 3. Execute/ start services



Server	Protocol
Mail server	SMTP
file server	FTP
web server	HTTP
name server	DNS
Auto IP assign server	DHCP
file/folder sharing server	NFS
remote login server	SSH

✚ How to configure/setup our own Apache web server –

1. Install Software - httpd

```
>> yum -q httpd (check if it is installed or not)
>> yum install httpd
```

2. Configure, already done when installing software.
copy / deployment of website. (index.html)

```
>> cd /var/www/html/
>> gedit index.html
```

3. Execute (starting the services)

```
>> systemctl start httpd
```

```
>> systemctl status httpd
```

To stop **firewall**

```
>> systemctl stop firewalld
```

At client side go to this **URL** -

http://server_ip/webpage.html

index.html is the home page or the root page for the website.

HOW TO SSH BETWEEN WINDOWS AND LINUX -



✚ Required things at client side-

1. Client must have web browser
2. Web server address

✚ SSH is a program to connect client from server.

SSH → SSH → SSH server

1st program name (client program, in mac also SSH, in windows putty)

2nd SSH protocol (for web, HTTP protocol)

3rd server

windows

Linux



putty is used in windows for remote login in Linux servers.

For android-

JuiceSSH - SSH client

Identity: new

Username: root

Click save

Connection: root / ip

DAY 4 (Multi-user, interpolation, backwaking, tty, startx, w command)

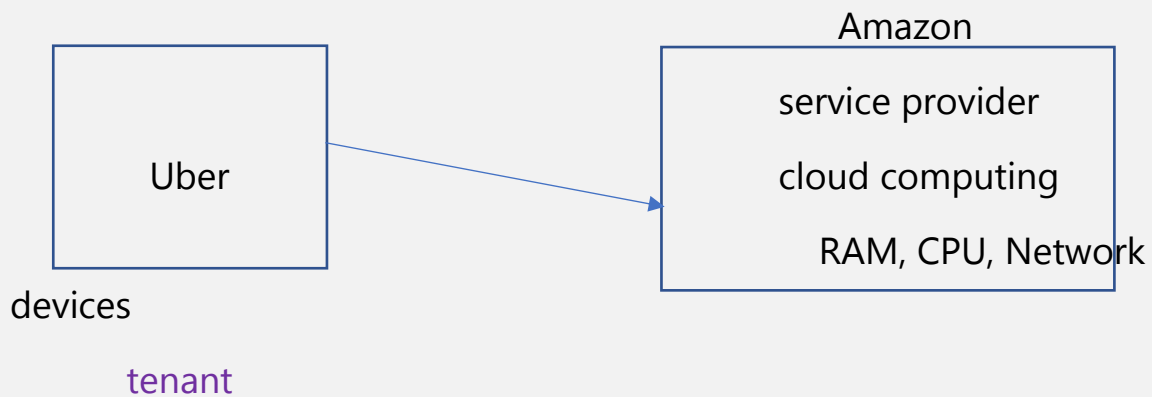
- ✚ Cloud computing is the managing IT resources of a company.

Resources that are managed by other company on your behalf, is called cloud computing.

Amazon (Amazon Web Services, AWS) is one of the best company in the world that manages resources of other companies.

- ✚ RAM and CPU processes things not hard-disk.

- ✚ Difference b/w **cloud computing** and **web hosting**-
you don't have any control in web hosting.



- ✚ **tenant** is the one who is taking services of cloud.

- ✚ To start or run a OS hard-disk is not required, we required only **compute units** (RAM and CPU).

Live OS

Network boot

CD boot

+ If we have to store data which is persistent (permanent), use **storage unit** (hard-disk).

+ Virtualization is a part of cloud.

Example – We're connecting pen-drive in cloud and it is detecting on our laptop.

+ Services available in cloud computing –

1. Infrastructure as a service (IAS)
2. Software as a service (SAAS)
3. Storage as a Service (StAAS)
4. Network as a service (NAAS)
5. Platform as a service
6. Security as a service
7. Container as a service

One of the best service is IAS (Infrastructure as a service) of cloud.

LINUX COMMANDS-

```
>> date +%r
```

+ Creating a personal user -

```
>> useradd adarsh
```



```
>> passwd adarsh // for creating password
```

+ Switch user

```
>> su - root
```

```
>> su - username
```

+ logout

```
>> exit
```

+ \$ represents that you are login with some guest user.

+ # represents root user.

+ history is individual for every user.

+ Calendar

```
>> cal 2016
```

+ Linux never stores file creation time. It stores last modification time.

For example, ls shows always last modification time.

```
>> ls
```

+ / represents a directory.

Example - `cd /var/www/html/`

+ Creates a empty file -

```
>> touch file.txt
```

- ✚ To write some data in created empty file, use `cat` command.

Removes the old data, `replace`.

Dangerous command.

Useful when `overwriting`.

```
$ cat > file.txt
```

Takes the input data from the command line and saves in the file `file.txt`

Use `Ctrl + d` for saving and exit.

- ✚ To append data in the existing file -

```
$ cat >> file.txt
```

HOW TO LOGIN WITH MULTI-USER SIMULTANEOUSLY –

- ✚ We know that Linux is `multi-user` system.

Where we `login` that is terminal.

Where we `run` command that is shell.

- ✚ One terminal is required for one user.

Linux gives `multi-terminals`.

So, Linux is `multi-user` operating system.

- ✚ There are total of 6 terminals available in Linux -

1 GUI (F1 or F7)

5 CLI (F2 to F6)

✚ Can be opened by pressing keys -

Alt + Ctrl + Function Key + F(1-6)

✚ To know about terminal name, use

>> tty

Every terminal is a device and is managed in dev folder.

✚ Where our graphical programs run, that display is :0

✚ This command can tell how many users are login or SSH login from which IP address.

>> w

✚ How to convert CLI into GUI of those 6 terminals -

➤ Login with CLI.

➤ GUI is a program and can be started with this command.

>> startx -- :1 (1 means 2nd GUI)

➤ When we run w command, it shows xinit in the description of corresponding login user.

➤ You can login from putty in windows, can start new session for multiple SSHs from windows to Linux.

- ✚ This command automatically runs `w` command in every 2 seconds and shows changes in the output.

```
>> watch w
```

To change the interval time, use `-n` option

```
>> watch -n 0.1 w
```

- ✚ To delete any user -

```
>> userdel -r username
```

- ✚ Now, to come out from `startx` GUI that was started in the terminal -
First `logout` in that `GUI` and then `exit` from terminals F2 or F3

- ✚ You can login from any interface – GUI, CLI, SSH. You will have the same power, doesn't matter which interface you are using for login.

PYTHON –

Interpreted

Compiled

Scripted

- ✚ Difference between interpreted and scripted and compiled language?

✚ X=1;

People says that this variable will be loaded into RAM and it will clear after your program terminates.

How will you explain that variable is loaded in RAM or not? How will you verify this thing .

Have you ever think about that?

Is data really stores in RAM and it removes after terminating program?

If we know how to read RAM, then everything can be solved.

When program terminates, data will be removed that's not perfect true.

Memory dumping → reading RAM

We can get password.

✚ Python can be use in interactive mode.

✚ JSS can be used in Java language for interactive mode.

✚ Live interpreter of python is known as REPL (Read Evaluation Print Loop).

✚ Single quote doesn't process special character in a string.

Double quote process these things.

This process is called **Interpolation** or processing special characters.

```
>> print r'adarsh \n jain'
```

Due to `r_` it will not interpolate the string.

`r` stands for raw.

✚ Backwacking - `\'`

Escaping - `\n`

```
>> type(variable)
```

✚ python `int` data type size is 4 Byte but if it's more than 4 bytes, then it's dynamically allocates more bytes.

```
>> x[1:15:2]
```

```
>> x[:]
```

```
>> x[::]
```

```
>> x[::-1]
```

```
>> x[::-1] # reverse string
```

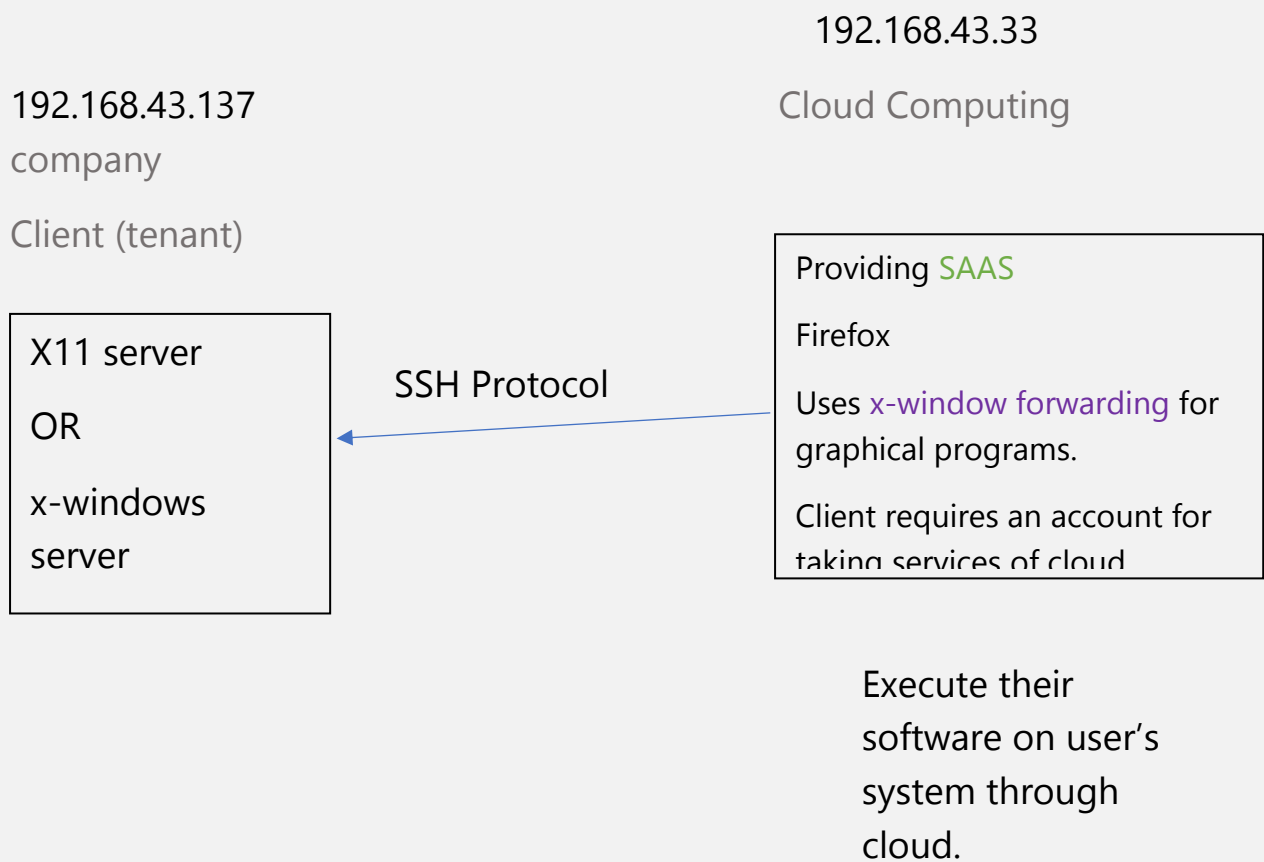
```
>> x,y = y,x # swapping #how it's possible?
```

```
>> dir(x) # gives all functions that can be used on x variable
```

DAY 5 (x-window forwarding, SSH, SCP)

SAAS (SOFTWARE AS A SERVICE)-

Example transferring Firefox from cloud as a service to its client.



🚦 Ping 2 only times and then quits -

```
>> ping -c 2 192.168.0.163
```

c stands for count.

✚ Required an account in cloud company for accessing services of cloud and we will login from that account on server side to provide services.

✚ SSH is used for -

Running remote program.

Remote login.

✚ Suppose Firefox is required on IP 192.168.43.137 (tenant)

Protocol used for transferring service is SSH between cloud and tenant.

Firefox will run on 192.168.43.33 system's RAM. All resources used will be of that system.

For displaying GUI, there is running a graphical program in background. There is a program running in background that is responsible for GUI.

x-window server program is required to run GUI program.

For example,

When we discussed above about multi-user, then we see there that only CLI is available for that users but we can start GUI program for them.

Press Ctrl + Alt + F(2-6)

>> gedit

Only running CLI so can't display GUI program. We have to first run GUI Program only the GUI program can be launched.

- ✚ To run Firefox, on system **192.168.43.137**, we request through SSH, and cloud computing company have to share their graphical display using x-window forwarding.

Forward x-window to client through SSH protocol.

- ✚ At server side -

- Enable this feature in SSH.
- Change configuration file of SSH where this is to be enabled.
- Cloud computing have to be forwarded.

```
>> gedit /etc/ssh/sshd_config
```

Line number 115 – **X11Forwarding yes**

Now, Server side is complete.

- So, now -
 - Firefox is installed at server.
 - User Account is created on cloud service provider.
 - x-forwarding is enabled.
 - Networking is good.

- ✚ Finally, to connect and run Firefox on user tom, use -

```
>> ssh -X -l tom 192.168.43.33 firefox
```

Use SSH, l for login user tom (user account), IP address will provide services and run Firefox program.

- ✚ If there is only one Firefox installed on cloud system then only one user can access.

This is the limitation of Firefox. It's not multi-user.

Only one Firefox can run at a time, it doesn't mean any other user can see other's data. Every user's data will be separate.

All history, bookmarks, downloads will be save on cloud and we can access them at any time.

This all mean that now Firefox is centralized.

- ✚ Internet used in running Firefox is of server not of client.

You can check it by turning off internet on client side (without turning off network or disconnecting from network).

- To disconnect only internet -

```
>> route del -net 0.0.0.0
```

This disconnects from ISP. Now, internet will not work but still you can ping from other's system that means you're connected from network.

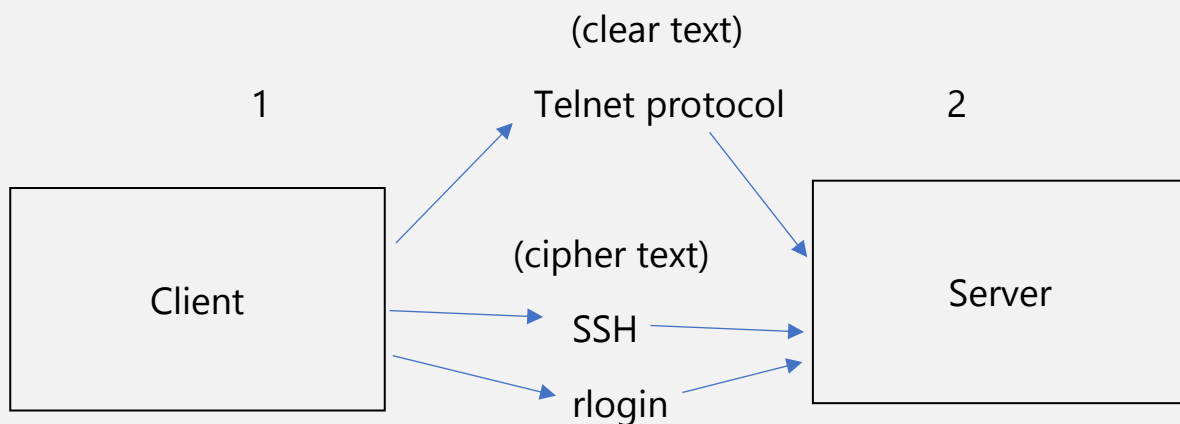
- To again connect, from internet -

Disconnect and again connect connection.

- ✚ Resources is using of cloud but client is running the program, it gives feel of virtualization.

LINUX –

✚ Remote login server –



➤ telnet is not secured, not encrypted.

➤ SSH is encrypted.

✚ Alias can be used in shortening command.

```
>> alias ping = "ping -c 4"
```

```
>> alias d = date
```

✚ To see Public Key stored that was used in SSH (Secure Shell) while connecting with someone's IP.

```
>> cat /root/.ssh/known_hosts
```

✚ System that supports SSH protocol, is also SSH server.

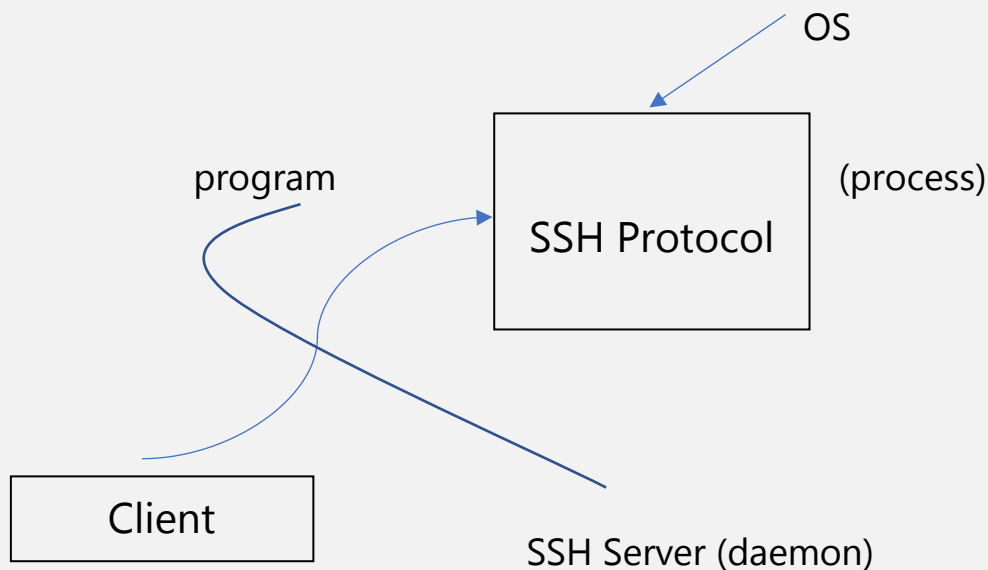
✚ How to confirm that system is using SSH protocol or not-

```
>> systemctl status sshd
```

```
>> systemctl stop sshd
```

- ✚ When we say that a system is connected, it connects with program running in OS, not with OS or IP. It doesn't connect with IP, it is just the way of connect, only address.

Without starting required program(process) to connect system, we can't connect.



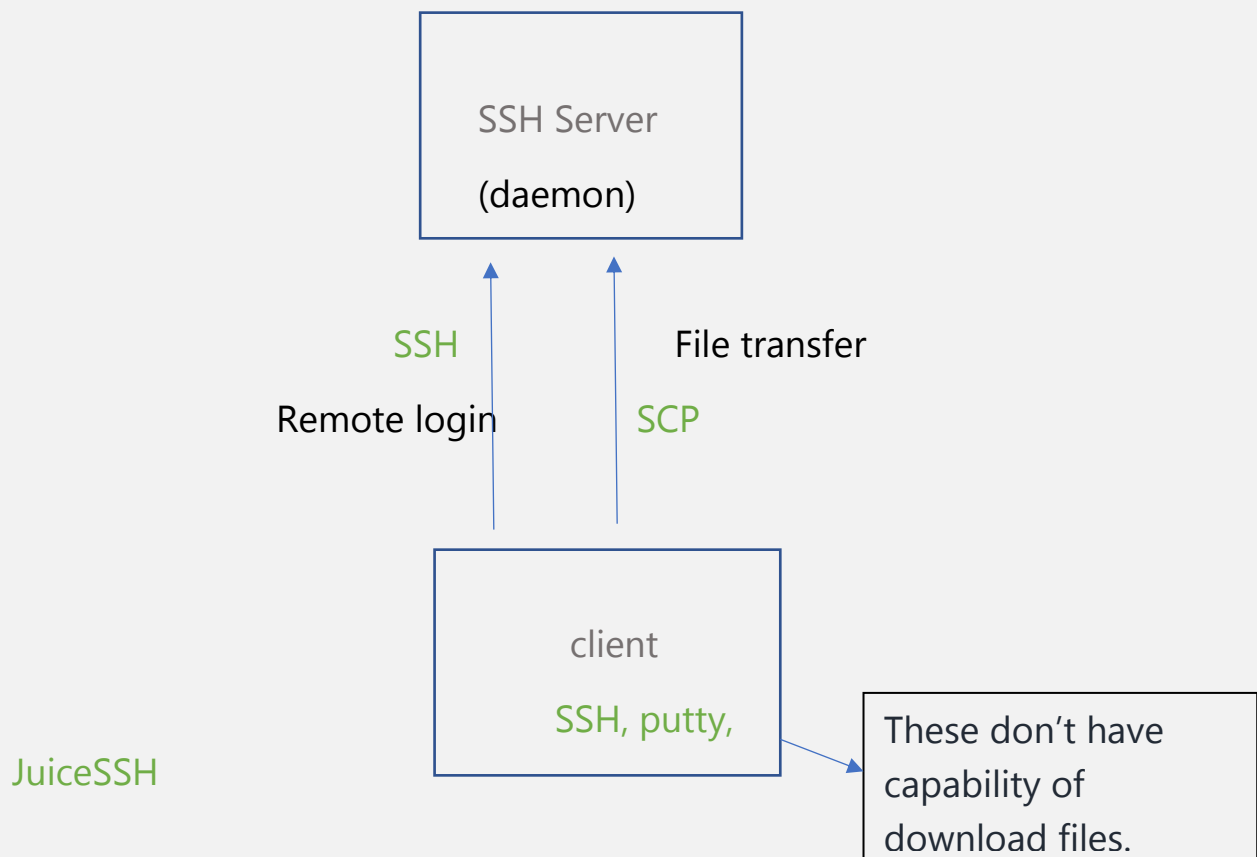
- ✚ When we login in someone's system, we're in user's home directory by default.

```
>> echo $HOME  
>> cd $HOME  
>> cd
```

- ✚ Services which are running in background are called daemon.

For example, servers running in background are daemons.

sshd, httpd → d - daemon



- ✚ To know details about a user -

```
>> id username
```

- ✚ We can do everything with SSH in a file (edit, copy, remove, create) but can't download or upload in our system via SSH.

These are the limitations of client program (SSH, putty etc)

They cannot transfer file.

- ✚ For file transferring, scp is useful.

➤ Downloading -



```
>> scp 192.168.43.137:/root/Desktop/b.txt /root
```

IP

Path



Source

Destination

➤ Uploading-

```
>> scp /root/Desktop/a.txt 192.168.43.137:/root
```

➤ Folder uploading -

```
>> scp -r /root/Adarsh/ 192.168.43.137:/root/
```

✚ In windows, we can use WinSCP for file transferring.

✚ To see the information about RAM, run -

```
>> free
```

```
>> free -m
```

Shows in MB.

✚ We can get data from RAM by reading RAM.

PYTHON-

✚ This command gives system's commands output and returns exit status (0 or something else).

```
>> import os
```

```
>> x = os.system("date")
```

x stores return exit code, not the output of the running command.

✚ How to check that a command is successfully run or not-

We can't decide from command's o/p that it is successfully run or not.

We can check this from exit status or code.

After executing each command, Linux stores exit code in variable ?

```
>> echo $?
```

0 – success

Otherwise – fail

We can check the `exit status` for checking a successful command.

✚ A command's output can't be simultaneously `print` and `assign`.

If a function prints something then it can't be assigned.

If it returns a value, then it can be assigned.

✚ `os.system ('date')` print's date's output and returns 0, so we can store only 0 and can't store output.

That's the limitation of OS command.

✚ Another function that can be used for running system's command is -

```
>> import commands
```

```
>> s = commands.getoutput('date')
```

```
>> s = commands.getstatusoutput('date')
>> s
>> (0, 'Thu...2017')
```

Return the output of command and stores in s

✚ To change the colour of terminal (foreground) -

```
>> commands.getoutput("tput setaf 1")
>> '\x1b[31m'
>> os.system("tput setaf 1")
>> changes colour
```

setaf – foreground

setab – background

✚ To reset terminal again-

```
>>reset
```

Creating executable python file –

1. Make the file executable.

```
>> chmod +x test.py
```

2. We have to include

```
#! → shebang
```

Specifies environment

```
#!/usr/bin/python2
```

Specifies path of interpreter (when GUI running, double-click)

Now you can also run file by

```
>> ./filename
```

- ✚ Graphical program which runs when double clicked on folder to open folder in GUI -

```
>> nautilus foldername/
```

Nautilus is the file manager for GNOME desktop.


- ✚ For printing multiple lines, use triple quotes.

DAY 6 (StAAS, PATH, Password Cracking, Partitions)

Example of SAAS -

- Gmail
- Online Microsoft office 365
- Salesforce
- online gaming

SECURITY IN SAAS -

 A user can access anything from cloud including firefox. But we want to provide only firefox service.

So, we want that user can only access firefox, to do that –

- One way for doing this is restrict every program permission for that user.

On cloud system-

- Restrictions are on file not on command.
- Use ACL (Access Control List) for applying restrictions.

```
>> setfacl -m u:yash:- /usr/bin/gnome-terminal
```

u – user (yash)

'-' for giving no permission to user yash

m – we're modifying permissions

then give the path of program file.

But the limitation of this method is that we have to restrict every program on our system for security purpose.

- Another method is to create an application specific user. User will only use specified application and can't access anything else.

We normally create OS specific user. To create application specific user, run

```
>> useradd -s /usr/bin/firefox jack
```

Now user jack can open only firefox, even if it is trying to open some other application.

If we have to give some other services to this user like gnome-terminal then we can unblock those services using this command -

```
>> setfacl -m u:jack:rx /usr/bin/gnome-terminal
```

- ✚ To provide windows application on Linux through cloud we have to install windows software on cloud, only then we can provide those services or give users those software.

Cloud is running on Linux. So, it can provide only those s/w that it has.

- ✚ How to install Windows software on Linux –

To install windows software, we have to create environment of windows in Linux that is Windows Virtual Environment

Wine software can be used to install windows software in Linux.

- ✚ To check your internet connection, try to ping with google.com-

```
>> ping google.com
```


LINUX –

- ✚ When we run any command on Linux, in background it asks for path of command to *which*, *which* then query with PATH variable to find the path of command that has to be run.

For example, when we run `date` command, then *which* finds the path of `date` command from PATH variable and then runs it.

```
>> date
```

First runs *which* to find path

Then runs `/usr/bin/date`

- ✚ How *which* find location of commands -
which contacts with PATH variable (system variable)

It checks in every path specified by PATH.

Shell → *which* → path

which is not a command that's a shell function.

- ✚ To prove that every command, uses PATH variable then run the following command.

This command empties path variable -

```
>> PATH
```

Now no command will run

- ✚ Every user has its separate path file.

✚ If a command is not running, then there might be following reasons –

1. Every command is a program file. So, may be that software is not installed.
To solve this, install required software.
2. Path is not set for the current user.
Use which to check that it's path set or not
3. User doesn't have permissions to run that command.
Ex – `useradd` can't run by other users because they don't have permissions to run that command.

✚ If we're `root` user and command is not running then there might be following reasons -

1. Software is not installed
2. Path is not set.

✚ To know about a File that it is from which software (file query).
Provide file path in argument.

```
>> rpm -qf /etc/passwd
```

✚ Command and program file -

Command can be run from any location

To run Program file, we have to go in the folder.

✚ To convert program file in command, append the path of program file in `PATH` variable -

```
>> PATH=/root/Desktop/python:$PATH (No space)
```

- ✚ Another program can't use this variable. So, it should be exported if we want that another program can use this variable.

- ✚ All the variables that we set in a terminal session, are temporary.
To permanent add these changes, we have to write these in a file.

When we open terminal, `bash` shell runs in background and it runs some programs whenever shell is started.

So, if we want that when we're opening terminal and some program should run, then we can write those commands in file.
That file is -

```
>> gedit /root/.bashrc
```

Add any command in this file that you want to run whenever terminal starts.

- ✚ We can use name instead of IP while doing networking-

Create a database like structure.

Every system is a host.

```
>> gedit /etc/hosts
```

Add IP and name in this file. After this you can use specified name to connect with IPs.

- ✚ To see Host name of your system-

```
>> hostname
```

- ✚ To set host name

```
>> hostnamectl set-hostname adarsh
```

✚ `cat` cannot be used for editing the documents at CLI anyhow. To edit any file in CLI, use `vim`.

`w` – save

`q` - quit

`wq` – save and exit

`q!` – force quit

`esc` - command/control mode

When using **SSH**, we can't use `gedit`, we should use `vim` to edit file remotely.

✚ Copy and move -

`>> cp source destination`

`>> mv a.txt Desktop/b.txt`

Rename also uses cut paste at same location.

PASSWORD BREAKING IN REDHAT 7-

If you forgot root's password then to break that follow these steps -

1. Restart your OS.
2. Use arrow keys (at boot screen) where two options are there.
3. Press '`e`' for edit.
4. Go to a line starting with `linux16`
5. Write this at the end of line `rd.break enforcing=0`

6. Press ctrl + x.
7. You are now in **emergency mode**.
8. Run the following command


```
>> mount -o remount,rw /sysroot/
>> chroot /sysroot/
```

Now we have entered in system without root's password. Set the root's password -

- ```
>> passwd root
```
9. Set new password
  10. Run >> exit
  11. Run >> exit

```
insmod xfs
set root='hd0,msdos1'
if [x${feature_platform_search_hint} = xy]; then
 search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' ff4815ee-b636-4b4b-9191-0599ff5dac59
else
 search --no-floppy --fs-uuid --set=root ff4815ee-b636-4b4b-9191-0599ff5dac59
fi
linux16 /vmlinuz-3.10.0-514.el7.x86_64 root=/dev/mapper/rhel-root ro c\
rashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet LANG=en_IN.\
UTF-8 rd.break enforcing=0
initrd16 /initramfs-3.10.0-514.el7.x86_64.img

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.
```

Activate Windows  
Go to Settings to activate Windows.

```
switch_root:/#
switch_root:/#
switch_root:/#
switch_root:/#
switch_root:/#
switch_root:/# mount -o remount,rw /sysroot/
switch_root:/#
switch_root:/#
switch_root:/# chroot /sysroot/
sh-4.2#
sh-4.2#
sh-4.2# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
sh-4.2# exit
exit
switch_root:/# exit_
```

After login in the system with `root` user -

### **Set SELinux (Security Enhance Linux)**

If you break root password, then you can't again login after shutdown the system.

So, run this command to avoid that –

```
>> restorecon /etc/shadow
```

🔗 *How to find shell function and shell command?*

## **PARTITIONS (FILE SYSTEM) –**

✚ Without creating partitions in hard-disk, we can't create folder.

✚ Partitions in Linux-

```
>> df
```

```
>> df -h (human readable)
```

/ drive in Linux

C drive in windows

✚ **1 GB = 1000 MB**

These are not the units of computer. Computer uses power of 2, not the power of 10.

Computer units are GiB, MiB (Mebi Bit) which is 1024.

$10^3$  MB = 1 GB (for humans)

$2^{10}$  MB = 1 GiB (for computer)

System works on 2 power and manufacturer works on 10 powers.

✚ Shows size in MiB, GiB

```
>> lvsdisplay
```

✚ Shows information about partition of disk -

```
>> fdisk -l
```

```
>> fdisk -l /dev/sda (sda is the name of hard-disk)
```

✚ For practicing partition, insert virtual hard-disk in Linux using virtual-box.

✚ Some points to remember –

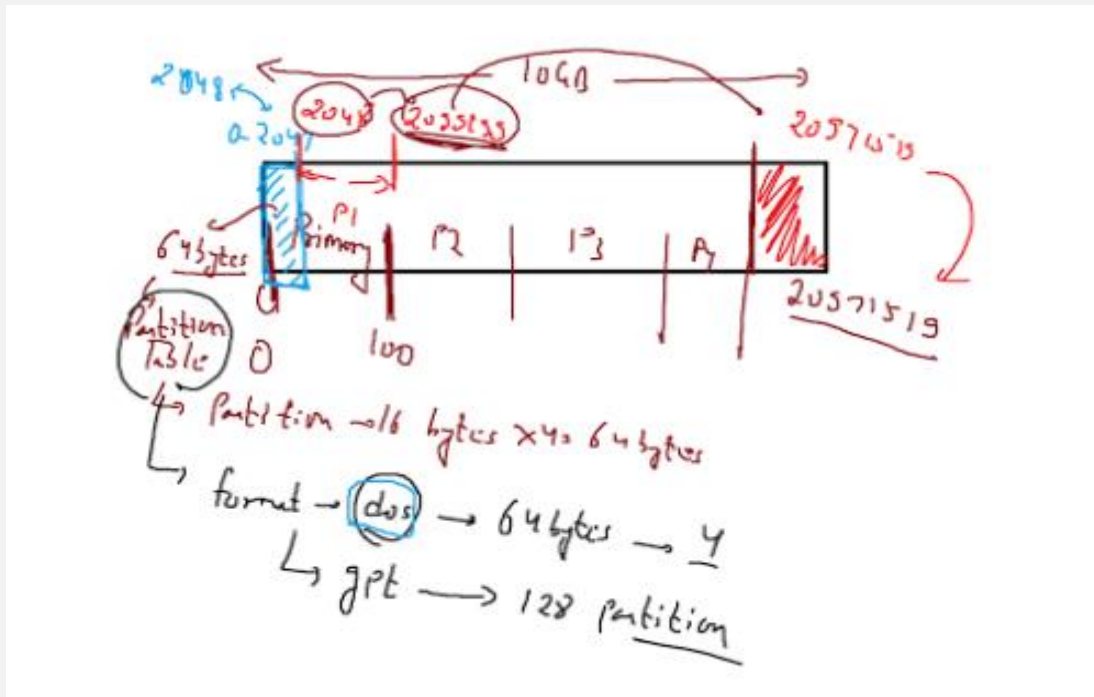
1. Hard-disk doesn't understand GB or MB, it's unit is sector.

1 sector = 512 bytes

To find actual size of hard-disk, find number of sectors \* 512

Allocates space in hard-disk in sectors.

2. Only 4 partitions can be created in a single hard-disk.



## MAKING PARTITIONS IN HARD-DISK

➤ Opens hard-disk prompt.

```
>> fdisk /dev/sdb
```

➤ Print partitions information of hard-disk.

```
>> p
```

➤ Create a new partition.

```
>> n
```

➤ Choose primary partition.

➤ Press 1 (1<sup>st</sup> partition)

- Initial some sectors (0-2047 = 2048 sectors = 1 MB) are **reserved** in hard-disk
- Actual space starts with 2048<sup>th</sup> sector.
- +1G
- Create 4 partitions like this.
- After 4<sup>th</sup> partition you can't create more partitions.

Finally -

- Press **w** for save
- Press **q** for quit without saving any partition.

This will Remove all partitions made because that was temporary.

- For deleting partition -

>> d

## WHY THERE IS LIMIT OF 4 ?

Because where we store the information of partitions, metadata of partitions, is fixed and of 64 bytes. This information is stored in partition table.

1 partition required 16 bytes so only 4 partitions can be created.

In 1 MB (2048 sectors) reserved in hard-disk, 64 bytes are reserved for storing this information.

To see partition table-

```
>> fdisk -l
```

Manufacturer of hard-disk never tell about partitions.

When we first time use hard-disk, it is initialized or formatted. That format decides number of partitions in hard-disk.

OS creates a format of hard-disk when it was first initialized and that format decides number of partitions.

Format of partition that we use is DOS format → 64 bytes

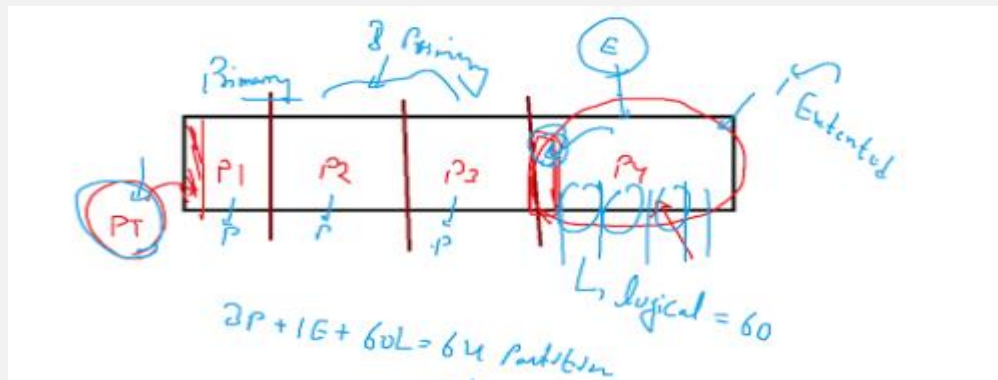
GPT format → 128 partitions can be created.

Partition table → format → DOS → 4 partitions

Partition table → format → GPT → 128 partitions

## ✚ Increasing number of partitions –

- Create a new partition table in hard-disk.



- P4 partition will be created in such a way that it's a separate hard-disk.
- This partition is **extended partition** in which we can create more partitions.
- **Logical partition** takes space in extended partition. Information or the partition table will be stored in extended partition.
- Create 3 primary and last 1 extended partition.
- Total 64 partitions can be made now.  
**3 primary + 60 logical + 1 extended**
- But 63 partitions can be used for data storage (remove 1 extended partition)
- There is no difference between primary and logical partition except that no one controls primary but logical is controlled by extended. So if we remove extended partition then all logical partitions will be removed.

## ✚ To see partitions information -

>>lsblk (list of block devices)

Hard-disk is also known as block devices

✚ Extended partition can't be used for data storage only logical and primary can be used.

✚ If partition has to be used for storage then follow these 3 steps-

1. Create physical partition
2. Format it
3. Activate/mount

## **Format partition -**

- ❖ Partition must create an index for every new file for faster processing.
- ❖ Whenever a file is to be opened, find that file in index.
- ❖ This index is formed in partition when its formatted first time. It's also called **file system**.
- ❖ This index table is known as **inode (index node) table**.
- ❖ So, every partition has to be formatted.
- ❖ OS reads only inode table to show sizes of folders, files, drives etc.
- ❖ This inode table can be changed, then OS will show different sizes then the actual size.
- ❖ When we remove a file, it only removes inode entry of that file.
- ❖ For example, remove a file of size 1GB and 100 GB, time will be same.
- ❖ When we format a partition, it only removes index page, data will not be removed. So, we can recover data from that partition.
- ❖ File system creates inode table to manage files.

❖ To format partition -

```
>> mkfs.ext4 /dev/sdb1
```



Example - NTFS, ext2, ext3, ext4, xfs etc.

- ❖ Time slicing is used in fibre optic cable.

## **Mounting or Activating -**

- ❖ Only two kinds of thing can be used in a OS -

File and folder

- ❖ We cannot go directly in a device.

So, the created device has to be converted in a folder Or link with a folder Or mount with a folder in order to use this.

```
>> mkdir /data
```

```
>> mount /dev/sdb1 /data (This data is like a pen drive
mount and unmount)
```

```
>> cd /data/
```

```
>> cat > linux.txt
```

```
>> umount /dev/sdb1
```

```
>> cd /data/
```

```
>> ls
```

Mount again

- ❖ To know about which partition is mounted on which folder.

```
>> df -h
```

- ❖ Shows inode table.

```
>> ls -l
```

❖ Shows inode number.

```
>> ls -il
```

---

## *DAY 7 (Object StAAS, pen-drive working, partitions in pen-drive)*

---

### **STORAGE AS A SERVICE (STAAS) –**

- When we use the storage of cloud, that is the **StAAS**. In this, cloud system has storage and the user use that storage on his system and can use that to store data.
- Using storage of cloud.
- Attach storage device on cloud and access by some other PC.

Example – pen-drive detection

Google drive and drop box

- Cloud system has the capability of **Storage Scaling**.
- We can integrate all storage.  
For example, if we have multiple pen-drives then we can integrate all pen-drive storages in a single one.

✚ There are mainly two types of storage –

1. Object
2. Block

**Object Storage** - We can not a make partition in google drive storage. We can only store files and folders there. This is called object storage. We can't install OS there as there are no partitions.

Example –

- Google drive (google compute engine, GCE), OneDrive, drop box, Microsoft azure

- Amazon has its own cloud service **AWS. S3**(Simple Storage Service, SSS) is the product of amazon that provides **StAAS**. It's a **public cloud**. Anyone can use their services.
- OpenStack is the biggest private cloud.
- OpenStack has product **swift** (Object Storage).

**Block Storage** - If we're able to make partitions in provided storage, then we can install OS. We have a hard-disk and we can make partitions in them, this type of storage is known as **block storage**.

Example -

- Block storage service of **AWS – EBS (Elastic Block Storage)**  
Cloud provides facility of scaling storage that's the elastic property of cloud.
- Block storage of OpenStack – **Cinder**

## **OBJECT STORAGE (OBJECT STAAS) –**

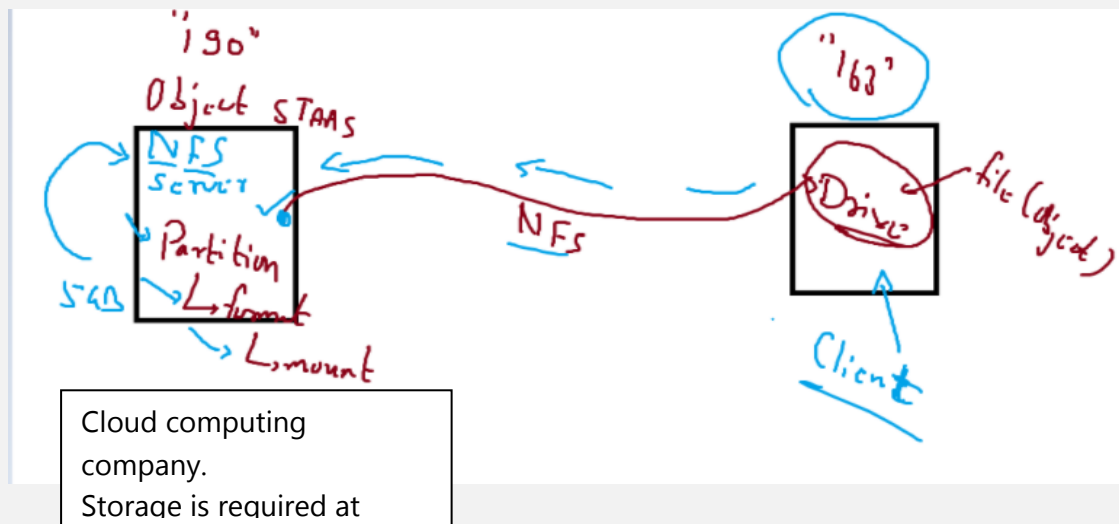
- All data is centralized on cloud in Object StASS.
- Proper networking is prerequisite in cloud computing, not a limitation.
- For transferring data, we require a protocol – **NFS (Network File System)**

Other examples for protocols that we can use are –  
CIFS, SSHFS, HTTPFS, GlusterFS

- We will use NFS, SSHFS, HTTPFS protocol.

**FS** – File System

- Storage is not the file system but partitions are file system.  
A blank hard-disk is a storage device.  
After creating partition it's a file system.
- Cloud computing company have to format their partitions for give them as storage to their users.



## Setting up Cloud System -

- Cloud have to make partition according to requirement of user.
- Mount that partition to a folder. This folder will be used for user storage.
- Suppose we mounted partition sdb1 to a folder /cloud.

```
>> mkdir /cloud
```

```
>> mount /dev/sdb1 /cloud
```

- Now, we've to convert cloud computing system in a NFS Server.

Firstly, Open following file for NFS

```
>> vim /etc/exports
```

Write in this file -

```
/cloud *(rw,no_root_squash)
```

- \* means anyone (any IP) can access this folder.
- We can give IP name in place of \* for a specific user.
- NFS shares read-only by default. So, we have to share drive in rw (read and write) mode.
- **No\_root\_squash** will give unlimited or all powers to user to access drive or the storage space.

Secondly, start the service of NFS (Start NFS daemon)–

```
>> systemctl restart nfs
```

```
>> systemctl status nfs
```

- When client comes to cloud for accessing storage, firewall will stop client to connect. So, stop firewall.

This command will stop firewall temporary

```
>> Systemctl stop firewalld
```

To permanent stop firewall service use -

```
>> Systemctl stop firewalld
```

## Setting up Client System –

- Create a directory in /media folder.

```
>> mkdir /media/mydrive
```

Name of folder will be the drive name. Make this in media folder, so it will show on desktop.

- Now mount cloud's storage on your created folder.

```
>> mount 192.168.43.33:/cloud /media/mydrive
```

IP is the cloud's IP and /cloud is the shared folder by cloud system.

- Now you can access cloud's storage, can put data there, can do everything with that storage.

# LINUX –

## / PARTITION IN LINUX -

- / is a partition in Linux, root folder of Linux.
- Everything is inside / directory.
- We can't make anything outside /.
- Absolute path, starts with /
- You can check / size by `df -h` command, mounted on /
- You can't fill more data than its size.
  
- Suppose we make a directory `db` in /.
  - ```
>> mkdir /db
```
 - ❖ This new directory `db` will take space from /.
 - ❖ But to give it more space, mount some other partition on this folder.
 - ❖ Then that folder will take space from mounted partition, not from root, even if folder is inside /.
 - ❖ That's how we can create more space in only root without using the space of root. This is useful for cloud system to provide storage for their users.

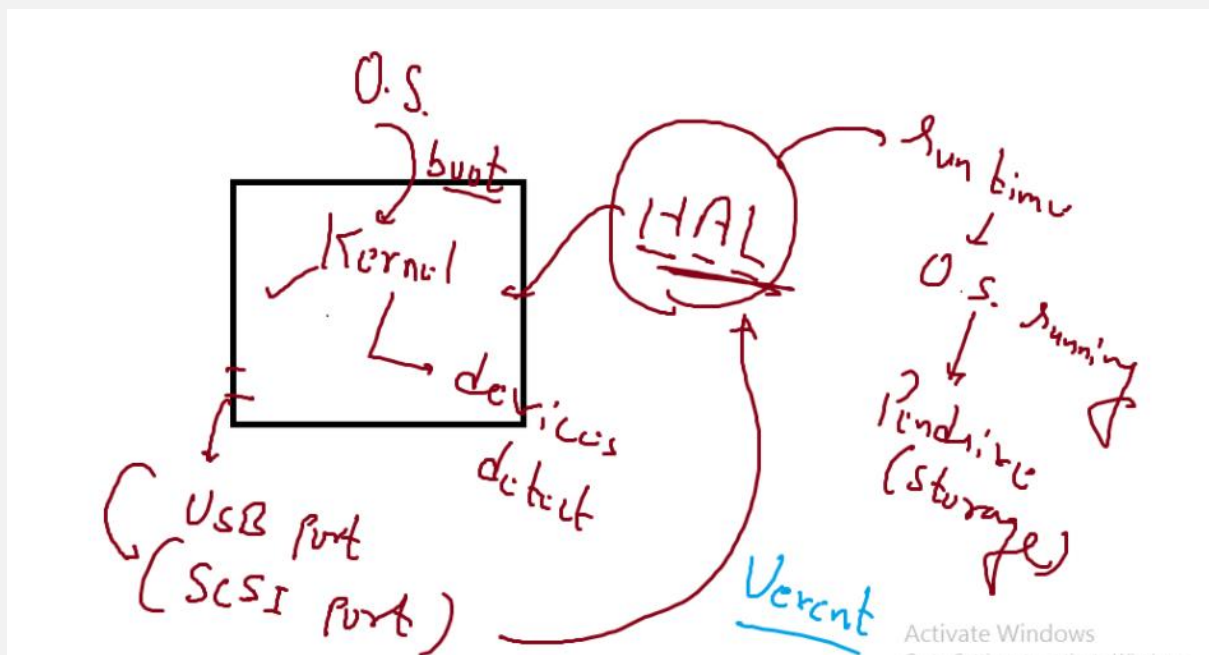
- This is the method so that folder will not take space from / and take it from some other partition.

- ✚ Difference between different file system is only in the index table or inode table.

- ✚ Driver is required for every device to work on a system.

Pen-drive detection and working –

- Normally drivers or devices are detected while booting up system.
- First program that loads while booting is kernel, which detect all devices and runs drivers.
- Drivers are loaded by kernel.
- When we attach a pen-drive, its detected-on system immediately, why?
- How to detect drivers while running OS but not at the boot time?



- Kernel only loads once, can't be reloaded again without rebooting the system.
- But now in modern OS, they have the capability to detect drivers at run time.
- In Linux, there is a program *HAL (Hardware Abstraction Layer)* which detects pen-drive or storage system while running OS.

- Pen-drive is attached to **USB-port** that is technically called **SCSI port**.
- This **SCSI port** sends a signal to **HAL** program, this signal flow or event is called **uevent**.

✚ We can create USB as a password.

When we attach pen-drive, there is a service udev.

Deactivate root account.

udev device management

<http://opensourceforu.com/2012/06/some-nifty-udev-rules-and-examples/>

✚ If you want to see signals that comes when attaching pen-drive

>> **udevadm monitor**

✚ When we attach a pen-drive and if that's already formatted then the system automatically mounts that drive to a folder, this is done by GUI program. It creates a folder and mounts pen-drive in that folder.

If GUI is not running then it will not be mounted and we have to mount pen-drive to some folder.

✚ If partition is mounted in **media** folder, only then GUI program will show that drive on desktop and that's only a link to mounted drive. To see the actual mounted folder, use **df -h**.

✚ To see the label of drive -

```
>> e2label /dev/sdb1
```

- ✚ To set the label of drive -

```
>> e2label /dev/sdb1 label_name(C:)
```

- ✚ New devices made on every partition and driver is loaded separately for every partition.

Using Pen-Drive for Making Partitions –

- Delete partition in pen-drive which is already there.
- Whenever remove partition –
Unmount partition from the folder and then remove
- This command can be used to check whether partition is mounted or not.

```
>> df -h
```

And then, unmount

```
>> umount /dev/sdc1 (unmount from  
run/media/root/adarsh)
```

- Now go into the disk and delete the partition.

```
>> d
```
- Now save before making new partition. Then make new partitions.

```
>> w
```

- This removed information is saved in partition table.
- But kernel may think that partition is still there

So, to update kernel

```
>> partprobe /dev/sdc (hard-disk name, not the partition name)
```

- Now create partition
- Update kernel using partprobe
- Then Format

```
>> mkfs.ext4 /dev/sdc1
```

- Finally mount to folder in media.
- Windows will not understand the format ext4.
- Many pen-drives are in FAT32 format because almost all OS supports that format.

To format partition in FAT format use

```
>> mkfs.vfat /dev/sdc1
```

- Shows the format of drives.

```
>> df -hT
```

- Shows many information about partitions.

```
>> blkid
```

✚ So, there are 4 steps -

1. Umount partition.
2. Delete partition using fdisk.
3. Save with "w" in fdisk.
4. Run `partprobe hard_disk_name`

✚ Detailed info of a software-

```
>> rpm -qi software_name
```

PYTHON –

✚ Python3 is object oriented, python 2 is not.

✚ In python2 we can't print a line without `\n` using `print`. So, there is a function which prints line without `\n`.

```
>> import sys
>> sys.stdout.write("hi")
```

✚ Python2 `input()` function is like `eval()` function of python3. Using `input()` function in python2 is dangerous because it takes everything as an expression and evaluates that.

✚ To change the prompt of python shell –

```
>> import sys
>> sys.ps1 = "adarsh >>>"
```

✚ If command prompt takes input, pass that input through pipe.

```
>> echo "redhat" | passwd jack --stdin
```

Gives input password for user jack using pipe.

✚ When we type something, we see the typed characters on screen because program do "echo back".

Password input is a separate program and doesn't echo back so we don't see any character while inputting password.

When we take input using `raw_input`, it echo back the input.

✚ To take password input in python -

```
>> import getpass
```

```
>> getpass.getpass()
```

`getpass()` in python takes input but doesn't echo back input means doesn't show input value.

DAY 8 (StAAS program, TCP, UDP, LVM)

- ✚ To see all the folders that we've shared or the shared folders which someone other on their system can use -

```
>> exportfs -v
```

- ✚ Client can know which folder cloud have shared.

To see at client side which folders cloud is sharing, run this command at client side -

```
>> showmount -e 192.168.43.92
```

- ✚ Unmounting drive -

```
>> umount /media/drive_name(drive full path)
```

- ✚ Following program mounts a drive from cloud to client in inputted drive name.

```
driveName=raw_input("Enter ur drive name : ")
mountPointCreate=commands.getstatusoutput("mkdir /media/{}".format(driveName))

if mountPointCreate[0] == 0:
    print "Mount Point create successfully"
else:
    print "Mount point not create .."

shareMount=commands.getstatusoutput("mount 192.168.0.190:/vimal /media/{}".format(driveName))

if shareMount[0] == 0:
    print "Drive Created .."
    removeDrive=raw_input("press 'q' to remove the drive : ")
    if removeDrive == "q":
        commands.getstatusoutput("umount /media/{}".format(driveName))
        raw_input("successfully removed. press enter to close ....")
    else:
        print "option not supported"
else:
    print "Drive not created .."
```

LINUX –

- ✚ If someone ping us then we don't know that someone is pinging me.

Even if someone is sending traffics or doing SSH or connected with us or transferring data with us, then we don't know about all this.

Attackers will be through wire or wireless.

If we know how to read wireless traffics then we can know everything about our internet connection.

This is called Capturing packets.

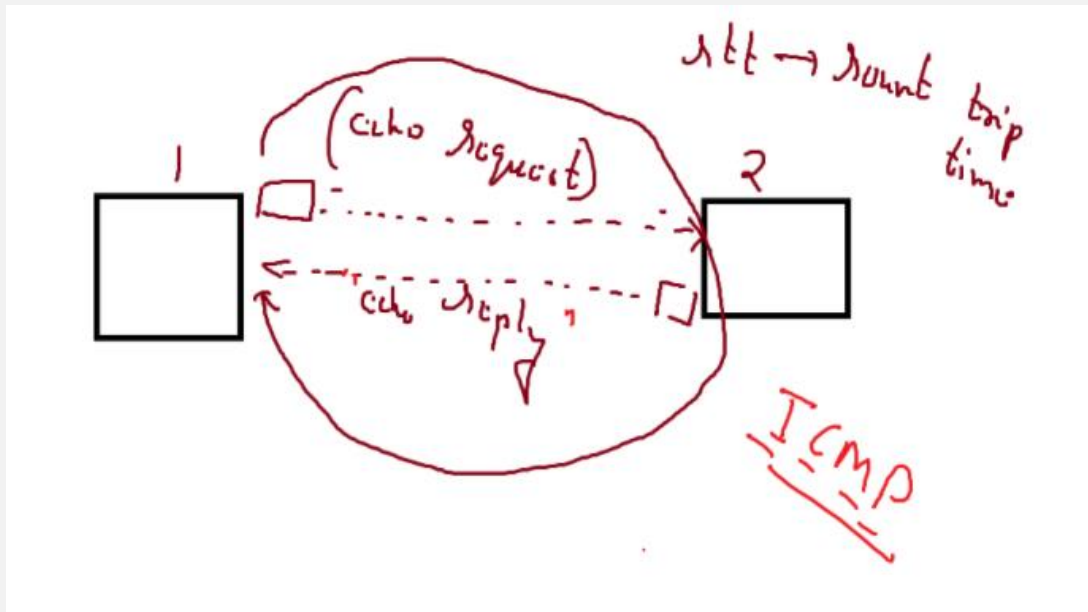
- ✚ Data comes on network card, not on OS.

- ✚ We can read packets of network card using –

```
>> tcpdump -i enp0s3 -n
```

i - interface

- ✚ When we use tcpdump and someone pings our system, then tcpdump shows echo request and echo reply.
- ✚ Ping is a command that uses ICMP protocol in background.



TCP and UDP protocol -

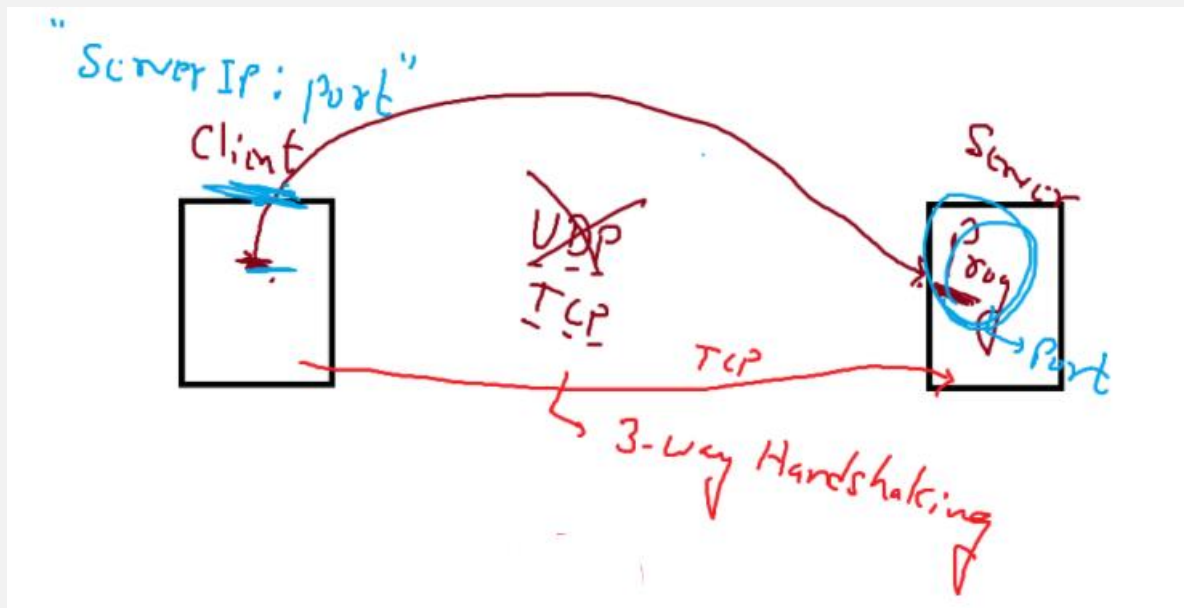
✚ TCP protocol controls packets transmission.

Connection oriented, first establishes connection between systems and then transfers data.

For example – A phone call first establish and then we talk with each other.

Reliable protocol, gives acknowledgement of packets transmitted.

- ✚ UDP protocol – Unreliable protocol, connection less.



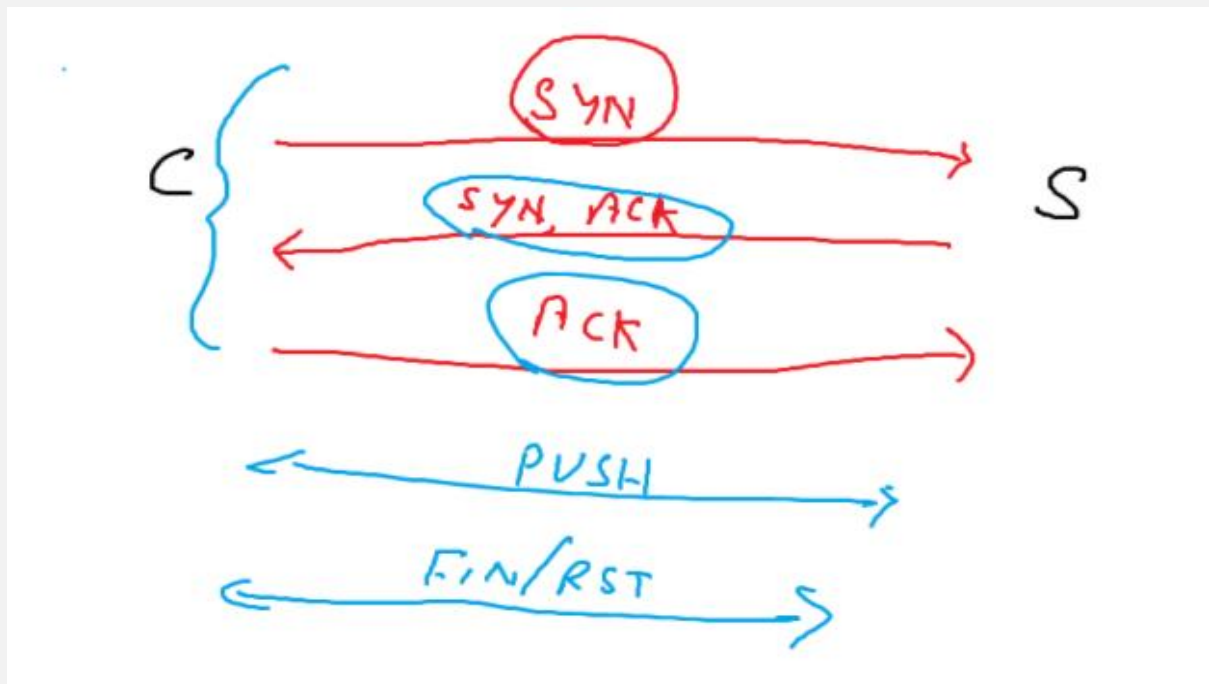
- ✚ SSH server runs on TCP concept.
- ✚ DNS, DHCP servers runs on UDP.

- ✚ If you want packets transmission info use TCP.

- ✚ TCP uses *3-way hand shaking* for connecting with system.

3-way hand shaking means that first 3 packets are transmitted for connection.

`tcpdump` shows these initial 3 packets with length 0 that means no data is transferred in those packets. Those packets are mainly for connection.



- ✚ P (PUSH) flag when connecting.
FIN/RST flag when disconnecting.
- ✚ IP address and port number are required to connect two systems.
OS or IP doesn't connect.
SSH server uses port number 22 by default.
- ✚ Only those programs which can connect to a network have a port number.
To see all program which uses some port -

```
>> netstat -tnlp
```
- ✚ Whenever client connects to a system – IP and port numbers are required.
Client sends port number whenever it connects to another system.
Packets arrived first checks IP and then see port number.

✚ SSH automatically uses port 22. This is written in a file.

To see all the programs and port number used by them, open following file -

```
>> vim /etc/services
```

Change port number in this file for a program and try to connect, it will not.

For example, change the port number for SSH and try to SSH with another system. It will not SSH because SSH program uses this file for port number and the other host will not be listening on that port.

✚ tcpdump shows all packets transmitted.

- To see only those packets which are coming or going from/to a particular IP.

```
>> tcpdump -i enp0s3 host 192.168.43.92
```

- To see only those packets which are coming or going from/to a particular IP at a particular port number -

```
>> tcpdump -i enp0s3 host 192.168.43.92 and port 80 -n
```

LVM –

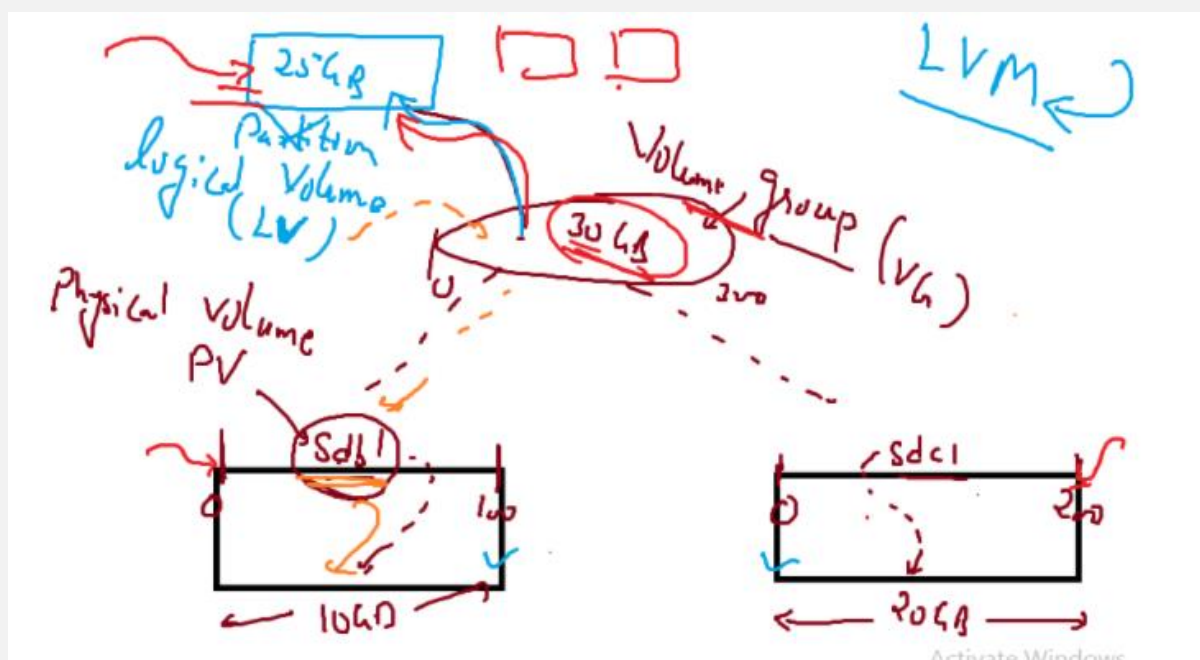
Example of LVM -

Suppose we've 2 pen drive of 10 GB and 8 GB, so there is total of 18 GB. We've a file of 12 GB. We can't copy this file in pen drive even if we've total of 18 GB storage.

So, some technology is there using which we can integrate storage. This technology or concept is known as LVM.

Suppose we have 200 and 100 sectors in two pen-drives then LVM will integrate those sectors into one –

0-100 sector + 0-200 sector = 0-300 sector



✚ Partition of physical(real) hard-disk is called → *physical volume(PV)*

✚ When multiple PVs are connected or grouped then it forms → *volume group (VG)*

- ✚ 10GB + 20 GB → make a partition of size 25 GB
This will be **logical volume**.
LVM (logical volume) will be partition, then format and mount.

Process for creating **LVM** -

1. Create **Physical Volume (PV)** -
We've two hard-disk **sdb** and **sd**c. Decide how much storage to give from hard-disks to **Volume Group (VG)**.
Make partitions in those hard-disk according to requirement.

sdb – 10GB → partition of 10 GB → sdb1

sd

c – 8GB → partition of 4 GB → sdc1

Total storage will be 14 GB.

2. Convert partitions in **PV**. Without converting into **PV**, **VG** will not understand the format. **VG** understands only **PV**.

To see already created PV -

```
>> pvdisplay /dev/sdb1
```

To create new PV -

```
>> pvcreate /dev/sdb1
```

Now, partitions are converted in PV. Do **pvcreate** for both partitions – sdb and sd

c.

3. Now to make VG -

```
>> vgdisplay vg1
```

```
>> vgcreate vg1 /dev/sdb1 /dev/sdc1
```

vg1 is the name of VG.

Then specify partitions that has to be grouped (any number).

Now vg1 has size 14 GB.

4. Now make partitions, these all will be logical partitions because these are virtual partitions. (LV, logical volume)-

```
>> lvcreate --size 12G --name mylv1 vg1
>> lvdisplay
```

Path of LV mylv1 is - /dev/vg1/mylv1

5. Format and mount -

```
>> mkfs.ext4 /dev/vg1/mylv1
>> mount /dev/vg1/mylv1 /media/LVM
```

Now we've a partition of size 12 GB.

- ✚ To increase this storage in real time without turning off OS -

```
>> lvextend --size +1G /dev/vg1/mylv1
>> lvdisplay
```

- But `vgdisplay` still shows 12 GB size and `lvdisplay` shows 13 GB.
- Because 12 GB is formatted but newly created 1 GB is not formatted.
- We can use only formatted partition so 1 GB which is not formatted is not added in mylv1 for storage.
- Now if we format then complete partition will format and data will be lost.

- But a command will only format those part which is not formatted. So, data will be saved. This will be online without disconnecting hard-disk.
- This format is called online resizing.

✚ Without formatting complete partition even if user is still working we can format the rest 1 GB partition –

```
>> resize2fs /dev/vg1/mylv1
```

✚ But, we can increase size of LV up to size of VG.

✚ To increase the size of VG, add new disk and add it to VG (vg1)

```
>> vgextend vg1 /dev/sdc2
```

✚ Every vg command follows same syntax.

```
>> vgcreate VolumeGroupName [PhysicalVolumePath...]
```

✚ xfs, NTFS, vfat format LVM doesn't support reducing size. We can extend the size but can't reduce again.


```
[root@master ~]# pvdisplay /dev/sdd1
Failed to find physical volume "/dev/sdd1".
[root@master ~]#
[root@master ~]#
[root@master ~]# pvcreate /dev/sdd1
Physical volume "/dev/sdd1" successfully created.
[root@master ~]# pvdisplay /dev/sdd1
"/dev/sdd1" is a new physical volume of "10.00 GiB"
--- NEW Physical volume ---
PV Name           /dev/sdd1
VG Name
PV Size           10.00 GiB
Allocatable       NO
PE Size           0
Total PE          0
Free PE           0
Allocated PE      0
PV UUID           Zcx4cd-z8bQ-Zasj-Y0yX-LDce-3LJP-hMYJXL
```

```
[root@master ~]# pvdisplay /dev/sde1
Failed to find physical volume "/dev/sde1".
[root@master ~]# pvcreate /dev/sde1
Physical volume "/dev/sde1" successfully created.
[root@master ~]# pvdisplay /dev/sde1
"/dev/sde1" is a new physical volume of "4.00 GiB"
--- NEW Physical volume ---
PV Name           /dev/sde1
VG Name
PV Size           4.00 GiB
Allocatable       NO
PE Size           0
Total PE          0
Free PE           0
Allocated PE      0
PV UUID           XKd0cx-Yb0f-n7f8-wqiX-3gJN-hygy-VIeX3I
```

```

[root@master ~]# vgdisplay vg1
  Volume group "vg1" not found
  Cannot process volume group vg1
[root@master ~]# vgcreate vg1 /dev/sdd1 /dev/sde1
  Volume group "vg1" successfully created
[root@master ~]# vgdisplay vg1
  --- Volume group ---
  VG Name                vg1
  System ID
  Format                  lvm2
  Metadata Areas          2
  Metadata Sequence No    1
  VG Access               read/write
  VG Status                resizable
  MAX LV                  0
  Cur LV                  0
  Open LV                 0
  Activate Windows
  Go to Settings to activate Windows

[root@master ~]# lvcreate --size 12G --name mylv1 vg1
  Logical volume "mylv1" created.
[root@master ~]#
[root@master ~]#
[root@master ~]# lvdisplay

```

Then format using mkfs.ext4

```

[root@master ~]# mkdir /media/mydata
[root@master ~]# mount /dev/vg1/mylv1 /media/mydata
[root@master ~]#

```

```

[root@master ~]# lvextend --size +1G /dev/vg1/mylv1
  Size of logical volume vg1/mylv1 changed from 12.00 GiB (3072 extents) to 13.00 GiB (3328 extents).
  Logical volume vg1/mylv1 successfully resized.
[root@master ~]#

```

```

[root@master ~]# resize2fs /dev/vg1/mylv1
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/vg1/mylv1 is mounted on /media/mydata; on-line
resizing required
old_desc_blocks = 2, new_desc_blocks = 2
The filesystem on /dev/vg1/mylv1 is now 3407872 blocks long.

```

Now to remove again LVM –

```
[root@master ~]# umount /media/mydata/
[root@master ~]# lvre
lvreduce lvremove lvrename lvresize
[root@master ~]# lvre
lvreduce lvremove lvrename lvresize
[root@master ~]# lvremove /dev/vg1/mylv1
Do you really want to remove active logical volume vg1/mylv1? [y/n]
: y
Logical volume "mylv1" successfully removed
[root@master ~]# vgremove vg1
Volume group "vg1" successfully removed
[root@master ~]# pvremove /dev/sdd1
Labels on physical volume "/dev/sdd1" successfully wiped.
[root@master ~]# pvremove /dev/sde1
Labels on physical volume "/dev/sde1" successfully wiped.
[root@master ~]#
```

DAY 9 (A Program, nmap, permanently mounting mount point)

✚ Use LVM to make partitions for cloud services.

✚ Program to create and mount a drive of specified size requested by a user –

```
#!/usr/bin/python2

import commands

userName = raw_input("Enter User Name: ")
driveSize = raw_input("Enter Drive Size (in GB) : ")

createPart = "lvcreate --size {0}G --name {1} vgcloud".format(driveSize, userName)

driveCreateStatus = commands.getstatusoutput(createPart)

if(driveCreateStatus[0] == 0):
    print("Drive Created Successfully.")

    formatDrive = "mkfs.ext4 /dev/vgcloud/{0}".format(userName)
    commands.getstatusoutput(formatDrive)

    makeFolder = "mkdir -p /cloud/{0}".format(userName)
    commands.getstatusoutput(makeFolder)

    mountDrive = "mount /dev/vgcloud/{0} /cloud/{0}".format(
                                                                username)
    commands.getstatusoutput(mountDrive)

    shareNFS = "echo '/cloud/{0} *(rw,no_root_squash)' >>
                /etc/exports".format(userName)

    commands.getstatusoutput(shareNFS)

    restartNFS = "systemctl restart nfs"
    commands.getstatusoutput(restartNFS)

else:
    print("Drive Not Created.")
```

LINUX –

- ✚ For shutting down Linux system

```
>> init 0
```

- ✚ For rebooting Linux system

```
>> init 6
```

- ✚ / partition in Linux is LVM.

- ✚ To see other's port using info (remote system)–

```
>> yum install nmap --disablerepo=rpel
```

```
>> nmap -sP 192.168.43.33
```

s – scan

P – port

```
>> nmap -sT IP
```

T – TCP

To check which ports remote system is using -

```
>> nmap -sT IP -p 1234(port)
```

To find the program running on port (versioning scan) –

```
>> nmap -sT IP -p 1234 -sV
```

- ✚ **Mount point** is where partitions are mounted. It's temporary. After rebooting the system, mounted file systems are automatically become unmount.

Permanently mounting mount point –

- ✚ There is a file which have all the information of mounting which is read at the boot time and mounts file system at specified locations in file. / partition is also mentioned in that file. That file is **fstab**.

```
>> vim /etc/fstab
```

- ✚ To see, which partitions are mounted on which folder in detail (there is also **df -h** command which shows mount information)

```
>> mount
```

rw – rewritable, ro – read only

- ✚ Write this in fstab file –

➤ Syntax -

device name	mount point	format	options	1
2				

➤ Write this -

/dev/sdb7	/media/newdata	xfs	defaults	1	2
-----------	----------------	-----	----------	---	---

1. Device name – partition, LVM, CDROM, pen-drive, nfs

2. Mount point – directory name
3. Format (file system) – NTFS, XFS, NFS, ext4, VFAT
4. Options – ro, rw, defaults

Device	mountpoint	fs	Options	Disk sync	Disk checking
↳ Part	↳ dir	↳ ntfs	↳ ro	0	0
↳ Lvm		↳ vfat	↳ rw		
↳ cdrom		↳ ext4	↳ defaults	1	1
↳ Pen port		↳ xfs			
↳ <u>NFS</u>		↳ nfs			2

5. Disk sync – 0 or 1
 - Whenever we write some data in file and save it frequently then that written data doesn't actually save in hard-disk. Rather it saves or stores in RAM and finally OS writes that simultaneously in hard-disk. Otherwise operation will be too slow.
 - This is disk sync.
 - This chunk of RAM memory where data is stored, is called **dirty memory**.
 - To know about every information about RAM, it's memory, open following file -


```
>> vim /proc/meminfo
```

```
>> free -m // to see RAM
```
 - During power-off time, disk sync option either sync data of dirty memory or doesn't.
 - 0 – Don't sync with hard-disk

1 – Sync with hard-disk

6. Disk checking – 0 or 1 or 2

- In initial OS, whenever we read a file, data loads on RAM and remove from hard-disk. If power goes then that data may lose.
- But now in modern OS, they create a swap file. Whenever we open a file, OS creates a swap file (.swp) of the opened file which have all the data of opened file temporary stored. This is also called creating a snapshot of file.
- You can check this by opening a file and running `ls -al` command.
- Open one file two times, check `ls -a`, there will be two files with `.swp` and `.swo` extension.

0 – Doesn't check disk, don't restore swap file.

1 – Check disk, clean swap file. (more priority)

2 – Check disk, clean swap file. (less priority)

So / partition is 1.

- ✚ After writing in `fstab` file, verify this by checking for any error otherwise system may not start again -

```
>> mount -a
```

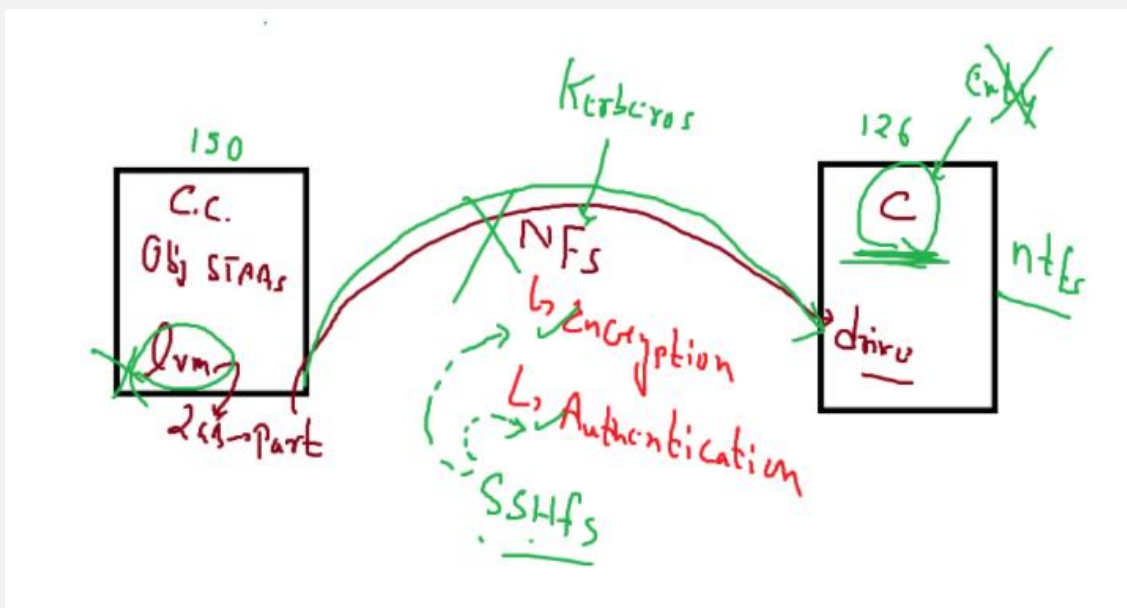
If no output, everything is right.

DAY 10 (SSHFS)

✚ NFS protocol –

- Works on clear text, no security, no encryption.
- Doesn't support authentication.
- Using **Kerberos** (for security) we can secure NFS protocol.

So, we'll use SSH protocol which is secure.



✚ SSH supports both encryption and authentication.

For transferring file system there is **SSHFS (Secure Shell File System)** protocol which transfers file using SSH protocol.

- ✚ Drive's format at cloud side doesn't matter whether it is ext4 or xfs. Client should support NFS or SSH protocol. So, if our client is windows then the file system doesn't matter, that is transferred via NFS server. So, type is NFS format which both supports.

StAAS using SSHFS protocol –

✚ Mount a drive in */cloud/adarsh* folder.

✚ *Setting up cloud -*

- SSHFS by default shares every folder in the system.
- Only adarsh(user) can access his folder, so made him owner –

```
>> chown adarsh /cloud/adarsh
```

chown - Change owner

adarsh - username

/cloud/adarsh - folder to be shared (folder path)

- None other can use this folder, so change the permissions -

```
>> chmod 700 /cloud/adarsh
```

✚ *Setting up client –*

```
>> mkdir /cloud/newst
```

```
>> mount adarsh@192.168.43.92:/cloud/adarsh /media/newst
```

✚ *mount* command with IP by default uses NFS protocol. So, sharing type has to be mentioned.

```
>> mount -t sshfs adarsh@192.168.43.92:/cloud/adarsh /media/newst
```

- Install SSHFS driver at client side

```
>> yum install fuse-sshfs -y
```

- But, mount can't access some drivers. So directly use -

```
>> sshfs adarsh@192.168.43.92:/cloud/adarsh /media/newst
```

PYTHON –

✚ tuple, string is immutable and list is mutable data type.

✚ *Why two different data type tuple and list?*

✚ Difference in for and while loop?

for loop initializes itself but while loop doesn't initialize. For loop always initializes with a statement if given and after completing an iteration, it executes a statement signing as successful completion of iteration.

while is almost same as if statement -

```
i=1
while(i>5):
    print("ok")
else:
    print("not ok")
```

✚ Doesn't prints new line.

```
for x in 'adarsh':
    print x,
```

o/p → a d a r s h

✚ Following function can take multiple arguments.

```
def multi(*x):
```

✚ Function can be assigned in another variable.

✚ To remove any variable –

```
del name(variable, function, module)
```



```
def lw():  
    print "hi"
```

```
x = lw()  
print(x)  
>> hi  
>> None
```

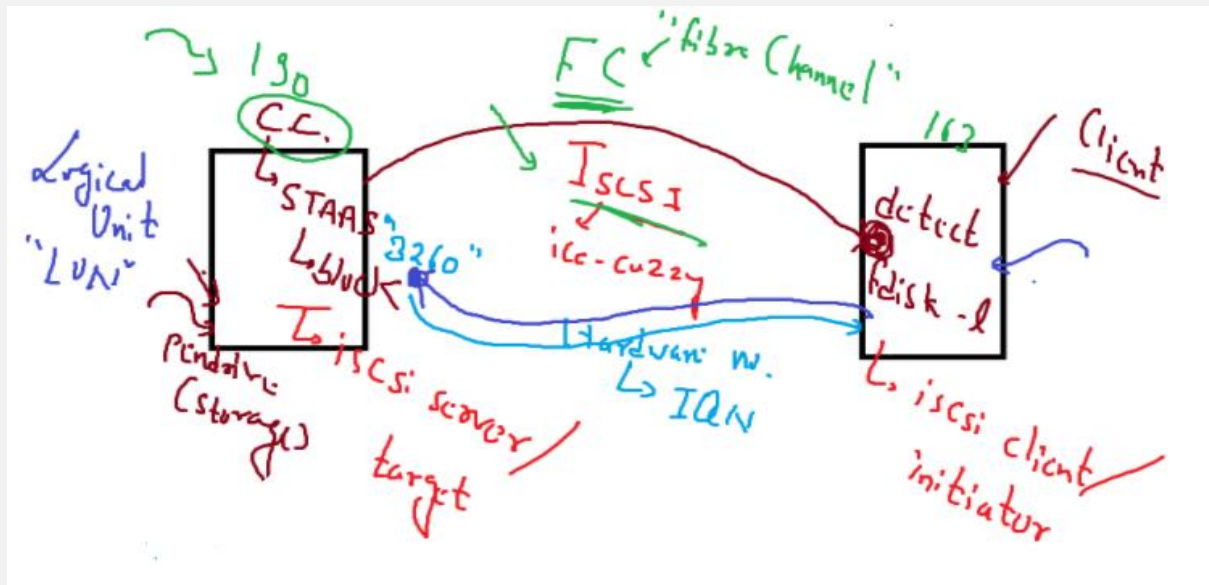
DAY 11 (Block Storage, iSCSI, SSH terminal input)

iSCSI (Internet Small Computer Systems Interface) an Internet Protocol (IP)-based storage networking standard for linking data storage facilities. It provides block-level access to storage devices by carrying SCSI commands over a TCP/IP network. iSCSI is used to facilitate data transfers over intranets and to manage storage over long distances. It can be used to transmit data over local area networks (LANs), wide area networks (WANs), or the Internet and can enable location-independent data storage and retrieval.

The protocol allows clients (called initiators) to send SCSI commands (CDBs) to storage devices (targets) on remote servers. It is a storage area network (SAN) protocol, allowing organizations to consolidate storage into storage arrays while providing clients (such as database and web servers) with the illusion of locally attached SCSI disks. It mainly competes with Fibre Channel, but unlike traditional Fibre Channel which usually requires dedicated cabling,

- ✚ Hard-disk is also known as block devices. So, in block storage we'll share hard-disk (a block of storage) with client.

- ✚ Pen-drive detection using block storage. Driver will load on client



side.

- ✚ iSCSI (ice-cuzzy) protocol is used for providing SCSI service which will give block storage to user.
- ✚ Server and client are also known as –
iSCSI sever (target) and iSCSI client (initiator)

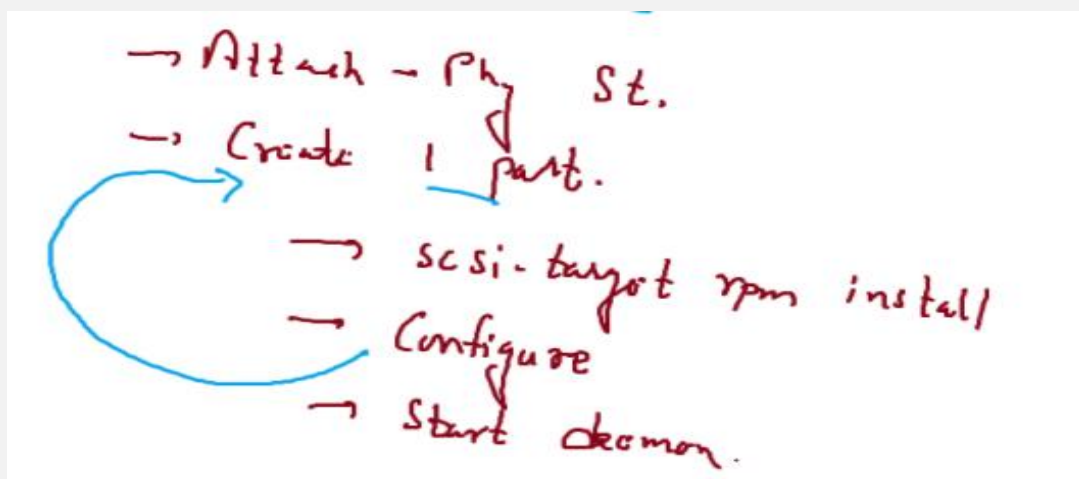
Making iSCSI server -

1. Install software for iSCSI server
Use one s/w for iSCSI server → targetd or scsi-target.rpm

```
>> yum install *scsi-target*
```

```
root@master:~  
File Edit View Search Terminal Help  
[root@master ~]# yum list *scsi-target*  
Loaded plugins: langpacks, product-id, search-disabled-repos,  
                : subscription-manager  
This system is not registered to Red Hat Subscription Management. You  
can use subscription-manager to register.  
Repository 'myrepo' is missing name in configuration, using id  
Available Packages  
scsi-target-utils.x86_64                1.0.55-4.el7      epel  
scsi-target-utils-gluster.x86_64        1.0.55-4.el7      epel  
[root@master ~]# yum install scsi-target-utils -y
```

2. Attach physical storage (pen-drive).
3. Create a partition in pen-drive which client will use.
4. Formatting of partition is not required.



5. Configure server.
6. Start services of server.

Configuring iSCSI Server–

- ✚ Open configuration file for iSCSI server -

```
>> vim /etc/tgt/targets.conf
```

- ✚ Write following lines in configuration file targets.conf-

```
<target mycloud>
```

```
        backing-store /dev/sdc1
    </target>
```

mycloud → IQN number

```
File Edit View Search Terminal Help
# This is a sample config file for tgt-admin.
#
# The "#" symbol disables the processing of a line.

# Set the driver. If not specified, defaults to "iscsi".
default-driver iscsi

<target mycloud>
    backing-store /dev/sde1
</target>

# Set iSNS parameters, if needed
#iSNSServerIP 192.168.111.222
#iSNSServerPort 3205
#iSNSAccessControl On
#iSNS On
```

- ✚ System first finds h/d number then loads driver.

We've to send h/d number of pen drive from cloud through internet.

This h/d number is IQN number.

- ✚ Client will give the IQN number to cloud to connect that hard-disk.

- ✚ After writing in file, restart service -

```
>> systemctl restart tgttd
```

- ✚ ISCSI server uses port number 3260

- ✚ ISCSI server's command to check sharing info of written IQN number in file. If everything is right then it will show size of hard-disk and much more.

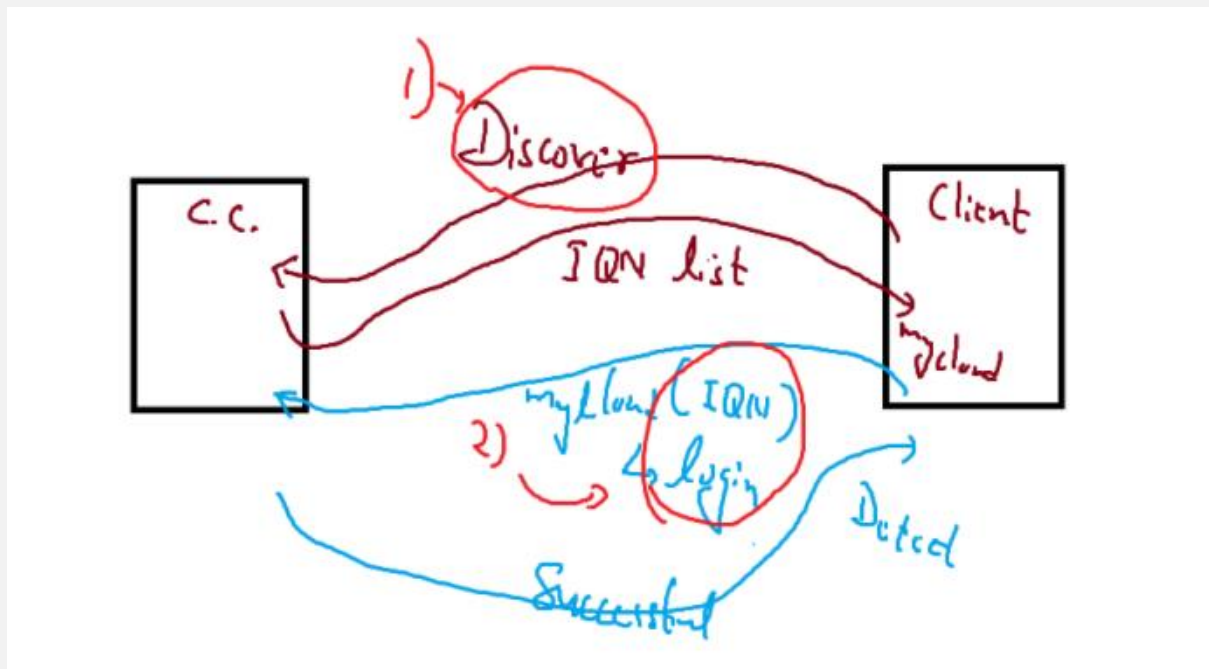
```
>> tgt-admin -show
```

Logical Unit Number (LUN) -

In SCSI terminology, LUN stands for logical unit, which are specified by unique logical unit numbers. A LUN represents an individually addressable (logical) SCSI device that is part of a physical SCSI device (target). In an iSCSI environment, LUNs are essentially numbered disk drives. An initiator negotiates with a target to establish connectivity to a LUN; the result is an iSCSI connection that emulates a connection to a SCSI hard disk. Initiators treat iSCSI LUNs the same way as they would a raw SCSI or IDE hard drive; for instance, rather than mounting remote directories as would be done in NFS or CIFS environments, iSCSI systems format and directly manage filesystems on iSCSI LUNs.

In enterprise deployments, LUNs usually represent subsets of large RAID disk arrays, often allocated one per client. iSCSI imposes no rules or restrictions on multiple computers sharing individual LUNs; it leaves shared access to a single underlying filesystem as a task for the operating system.

- ✚ Created partition is a partition of 2 GB for cloud but it will be a hard-disk for client system.



✚ Now to establish connection between cloud and client system following request are made -

1. Client sends a discovery request to cloud.
2. Cloud sends IQN list.
3. Then client send IQN number (login request) and detects storage.

At client side -

1. Install software -

```
>> yum install iscsi-initiator-utils
```

2. Use >> `man iscsiadm` for discovery and login command.

➤ Discovery command -

```
>> iscsiadm --mode discoverydb --type sendtargets
--portal 192.168.43.226 --discover
```

➤ Login command -

```
iscsiadm --mode node --targetname mycloud --  
portal 192.168.43.226:3260 --login
```

Example of iSCSI –

Let cloud have shared a partition /dev/vg1/mylv2 of size 40MB.

Client created 3 partition of size 10MB in his system.

Now if we run fdisk -l in cloud's system, then –

```
Disk /dev/mapper/vg1-mylv2: 41 MB, 41943040 bytes, 81920 sectors  
Units = sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disk label type: dos  
Disk identifier: 0xa38a68e5  
  
    Device Boot      Start         End      Blocks   Id  System  
/dev/mapper/vg1-mylv2p1    2048        22527       10240    83   Linux  
/dev/mapper/vg1-mylv2p2   22528        43007       10240    83   Linux  
/dev/mapper/vg1-mylv2p3   43008        63487       10240    83   Linux  
[root@adarsh tgt]#
```

Passing password to SSH using command line -

✚ SSH can also be used in the following way. It's useful while programming.

```
>> ssh IP command
```

✚ SSH will not check public key -

```
>> ssh -o StrictHostKeyChecking=no 192.168.43.33
```

✚ Using **sshpass** we can give cmd input in SSH command.

- `sshpass` is a utility designed for running ssh using the mode referred to as "keyboard-interactive" password authentication, but in non-interactive mode.
- ssh uses direct TTY access to make sure that the password is indeed issued by an interactive keyboard user. Sshpass runs ssh in a dedicated tty, fooling it into thinking it is getting the password from an interactive user.

```
>> sshpass -p redhat ssh -o stricthostkeychecking=no -l  
root 192.169.43.33
```

```
>> sshpass -p redhat scp j.php root@192.168.43.33:/root/
```

DAY 12 (Storage Snapshot)

Storage Snapshot -

A storage snapshot is a set of reference markers for data at a particular **point in time**. A snapshot acts like a detailed table of contents, providing the user with accessible copies of data that they can roll back to.

Copy-on-write snapshots store metadata about the location of the original data without copying it when the snapshot is created. This enables rapid recovery of data in case of a disk write error, corrupted file or program malfunction. Data in a copy-on-write snapshot is consistent with the exact time the snapshot was taken, hence the name copy-on-write. However, all previous snapshots must be available if complete archiving or recovery of all the data on a network or storage medium is required. Every copy-on-write process requires one read and two writes; data needs to be read and written to a different location before it is overwritten.

Snapshot vs. backup -

There are a number of benefits to using storage snapshots as part of a larger backup strategy. Snapshots provide quicker and easier access to data and can be leveraged by backup applications to enable features like instant recovery. But while storage snapshot technology is a helpful supplement to a backup plan, it is not considered a full replacement for a traditional backup.

Relying on stored snapshots for backups can take up storage space and seriously impact performance, and a storage snapshot is an instance, not a full copy of the data. Snapshots are dependent on source data, so if that data is lost, the snapshot is gone as well. Because of these

vulnerabilities, it is not recommended to use snapshots in lieu of a full backup.

- ✚ Snapshot can be used in changing the state of a storage from current state to any past state.
- ✚ We can take snapshot of any storage.
- ✚ We can go in any previous state using snapshots.

- ✚ To take any website snapshot use *wayback machine*, a site.

- ✚ Our cloud technology should support these snapshots technology.

- ✚ Snapshot is only supported in LVM partition, not in normal partition.

Creating Snapshot –

- ✚ Cloud have a LVM named `/dev/vgcloud/raju`
Client have mounted above LVM in his `/media/test` folder.

- ✚ Now to take snapshot –

```
>> lvcreate --size 100M --name mysnap1 -s /dev/vgcloud/raju
```


In above command, we're creating a LVM from LVM.

s- taking a snapshot of LVM.

>> `lvdisplay` shows a new LVM mysnap1 as a snapshot.

Shows size 3GB (lvsize) in `lvdisplay` but we give only 100MB.

✚ Snapshot takes less size to store everything. This is major difference in backup and snapshot.

✚ How it is technically possible –

➤ Snapshot has size in MB but have the data in GB.

100MB → 50GB

- In 100MB there is a link to actual file storage.
- Snapshot simply doesn't show any other file, show only that time's file, link to those files.
- If we remove a file from main partition, then snapshot copies data in snapshot storage.
- So sometimes it fails if deleted file has more size then snapshot's size.
- So, it's recommended that create snapshot of size equal to main LVM.
- Size of snapshot also depends on how frequently user is removing file.
- 100MB (snapshot size) is used in storage of meta data, changes in files, related links, removed data.
- This storage is known as **COW (Copy On Write)**
- If size is more, snapshot partition will be fail and can't be used again.

```
--- Logical volume ---
LV Path                /dev/vgcloud/mysnap1
LV Name                 mysnap1
VG Name                 vgcloud
LV UUID                 ZxzKb2-6A8D-p5fb-Lzot-9ePZ-lBs4-Kv5vBh
LV Write Access         read/write
LV Creation host, time  adarsh, 2017-06-17 11:46:58 +0530
LV snapshot status      active destination for raju
LV Status                available
# open                  0
LV Size                  3.00 GiB
Current LE               768
COW-table size           100.00 MiB
COW-table LE             25
Allocated to snapshot    0.01%
Snapshot chunk size      4.00 KiB
Segments                 1
Allocation                inherit
Read ahead sectors       auto
- currently set to      8192
Block device             253:5
```

✚ Snapshot partition doesn't require formatting.

✚ Now suppose client removed some of its file but took a snapshot before deleting.

To show client snapshot

```
>> mkdir /snap1
```

```
>> mount /dev/vgcloud/mysnap1 /snap1
```

Client will mount /snap1 in his system using SSHFS and he can see all those files at that point of time.

✚ We can use this technology in iSCSI

Suppose client have 3 partitions and these partitions are LVM for cloud. If any partition is deleted (may be OS partition), we can restore them using snapshot.

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPPOINT
sda	8:0	0	40G	0	disk	
└sda1	8:1	0	1G	0	part	/boot
└sda2	8:2	0	39G	0	part	
└└rhel-root	253:0	0	35G	0	lvm	/
└└rhel-swap	253:1	0	4G	0	lvm	[SWAP]
sdb	8:16	0	10G	0	disk	
└sdb1	8:17	0	2G	0	part	
└└vgcloud-raju-real	253:3	0	3G	0	lvm	
└└└vgcloud-raju	253:2	0	3G	0	lvm	/cloud/raju
└└└vgcloud-mysnap1	253:5	0	3G	0	lvm	/sanp
└sdb2	8:18	0	2G	0	part	
└└vgcloud-raju-real	253:3	0	3G	0	lvm	
└└└vgcloud-raju	253:2	0	3G	0	lvm	/cloud/raju
└└└vgcloud-mysnap1	253:5	0	3G	0	lvm	/sanp
└└vgcloud-mysnap1-cow	253:4	0	100M	0	lvm	
└└└vgcloud-mysnap1	253:5	0	3G	0	lvm	/sanp
└sdb3	8:19	0	2G	0	part	
sr0	11:0	1	3.5G	0	rom	/run/media/root/RHEL-7.3 Server.x86_64

Sound transfer in SaaS while playing video in client system –

Pulse-audio can be used for this purpose.

Song is playing in one system and sound is in another system.

PulseAudio is a networked low-latency sound server for Linux, POSIX and Windows systems.

LINUX -

- ✚ For using escape sequence in echo -

```
>> echo -e "adarsh \n jain"
```

- ✚ cp in background uses cp -i which uses interactive mode. It's because cp is a alias of cp -i in .bashrc file.

- ✚ Alias is concept of shell.

```
>> alias
```

Shows all alias in system.

- ✚ To remove any alias -

```
>> unalias cp
```

- ✚ Alias is a feature of shell. Python uses its own shell to run command, not the bash shell, so unalias is not required while using cp in programming in python because cp will be cp not cp -i.

- ✚ Shell is for human being, to run a command shell is not always required. A program can directly run command without any shell, any GUI. It directly interacts with OS to run command.

PYTHON –

- ✚ What is a process id? – a number, a file or a directory or something else

It's a folder with name process id and all the related data to program is stored in that folder. When we kill the process, it removes associated folder from RAM. So, the program closes.

- ✚ All data related to RAM is stored in /proc folder.

In this folder, we can find different folders related to every process id.

- ✚ We can check python program status in /proc/process id/status file.

Listen for ctrl+c in program -

Some program stores file in system and if someone terminate program then that file lives in system and that is not good for security. So, listening for this signal is good.

```
import signal
def mystop(x,y):
    print("bye")
    exit()
signal.signal(2, mystop)
```

2 for ctrl+c (KeyboardInterrupt signal)

DAY 14

- ✚ Kickstart is used for automation of installation of OS.

VIRTUALIZATION –

- ✚ Os uses hardware, used for running application, helps in human interaction with hardware.
- ✚ Use only one server on one system, it is recommended for security.
- ✚ If one server is hacked then another services may be hacked.
- ✚ Security-Enhanced Linux (SELinux) is a Linux kernel security module that provides a mechanism for supporting access control security policies, including United States Department of Defense–style **mandatory access controls (MAC)**.

SELinux ensures if more than one program are running on a system, than they will be isolated. So one program does not affect another program.

Still running more than one server on a same system is not recommended.

- ✚ OS is not optimized for full use of hardware.

Only 25% resources of total hardware is used.

In companies this can be worse where they have too much power but is not using that.

50% of total investing is on resources but only 10% of resources is using.

So, they come up with virtualization technique.

- ✚ When an OS runs on a hardware, it control on that complete hardware. None other OS can use that hardware at that time. So we cannot run (but can install) two OS on one hardware.

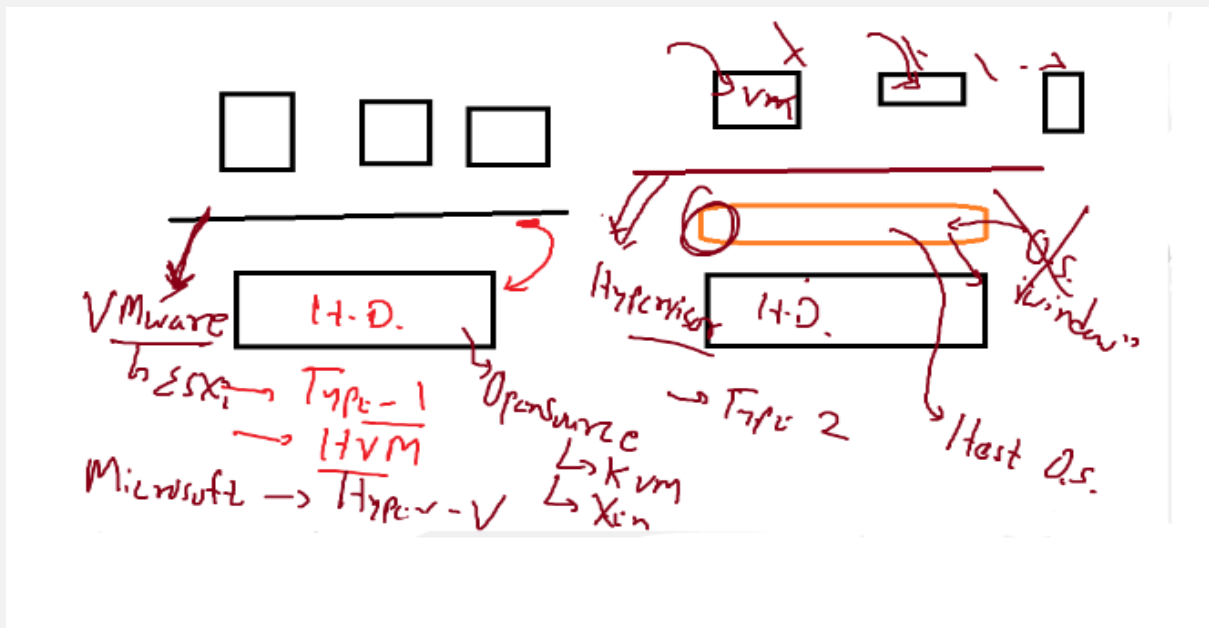
However, we cannot change complete OS to change use of hardware by OS, so the **virtualization** technique is created.

Now whenever you install OS on any hardware, it will not install OS directly on hardware. They create a **program**, which will install first on hardware. Then install OS on that program.

When OS will install, this program will tell about underlying hardware information. This program will tell OS about hardware resources. Then OS will only use that hardware which is told by that program.

This layer of program is hardware for OS, but it is not actual hardware. This layer is **virtual hardware**. It is known as **hypervisor**.

Today every company uses hypervisor.



✚ A **hypervisor** or **virtual machine monitor (VMM)** is computer software, firmware or hardware that creates and runs virtual machines. A computer on which a hypervisor runs one or more virtual machines is called a **host machine**, and each virtual machine is called a **guest machine**. The hypervisor presents the guest operating systems with a virtual operating platform and manages the execution of the guest operating systems. Multiple instances of a variety of operating systems may share the virtualized hardware resources: for example, Linux, Windows, and mac OS instances can all run on a single physical x86 machine.

✚ To remove some issues of **hypervisor**, containers are made and docker is example of that.

✚ There are two type of virtualization –

- **Type-1 virtualization (HVM, Hardware Virtual Machine)** → Install layer program on actual hardware and then install OS. Most companies use this.
- **Type-2 virtualization** → Install a layer on OS, now hypervisor is a program (virtualbox, vmware). This is not used in companies because if base system corrupts then everything can break.

✚ VMware has **ESXi hypervisor**.

✚ Microsoft has **Hyper-V**.

✚ Amazon uses **Xen**.

✚ Redhat uses **Kernel-based Virtual Machine (KVM)**.

DOCKERS –

- ✚ Install docker-engine-

Setup yum repo

[docker]

baseurl = <https://yum.dockerproject.org/repo/main/centos/7>

gpgcheck=0

```
>> yum install docker-engine
```

- ✚ Start the services of docker –

```
>> systemctl restart docker
```

- ✚ To see all the images available in docker –

```
>> docker images
```

- ✚ To load an image in docker –

```
>> docker load -i ubuntu-14.04.tar
```

- ✚ To see all the running OSs –

```
>> docker ps
```

- ✚ To run or start a OS –

```
>> docker run -it ubuntu:14.04
```

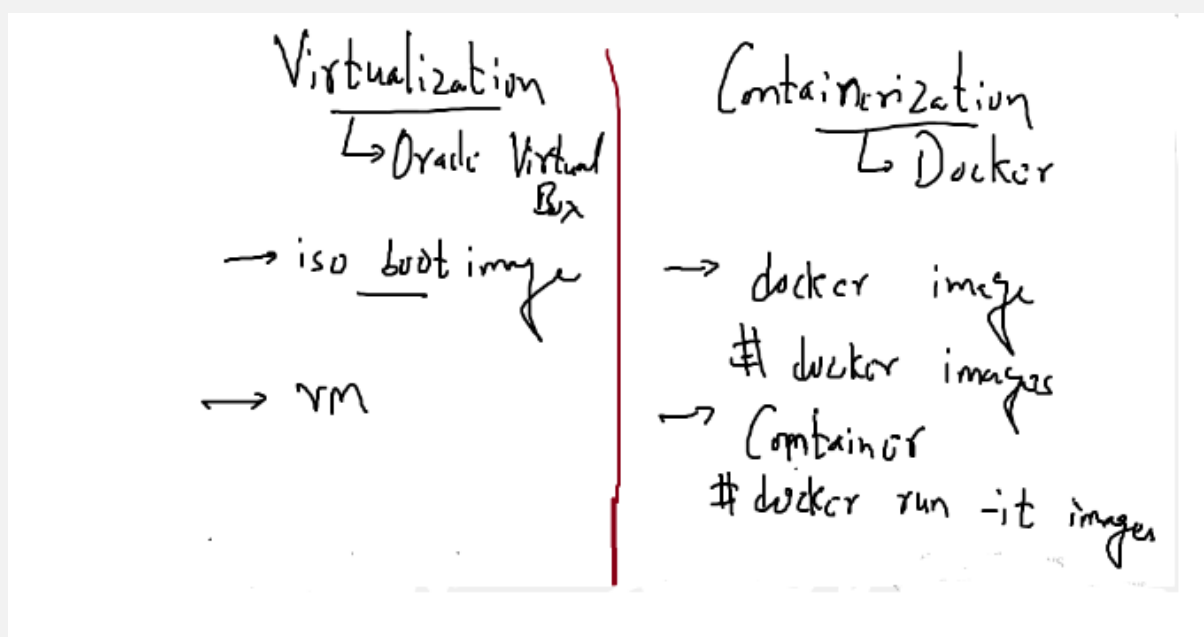
- ✚ To come out from OS console –

>> exit

```
# yum install docker-engine
# systemctl restart docker
# docker load -i ubuntu-14.04.tar
# docker images
# docker run -it ubuntu:14.04
```

```
open new terminal
# docker ps
```

✚ Difference between virtualization and containerisation –



→ poweroff	→ docker cmit
→ list	→ docker ps docker ^{or} ps -a } all
→ Start	→ docker start <u>ContainerID</u>
→ console	→ docker attach <u>ContainerID</u>

→ list	→ docker ps docker ^{or} ps -a } all
→ Start	→ docker start <u>ContainerID</u>
→ console	→ docker attach <u>ContainerID</u>

- ✚ To run any process in background in linux, for example when we run firefox or gedit it waits on terminal, to remove that use –


>> firefox &

- ✚ To see all the processes running in background –

>> jobs

- ✚ To start process again in terminal –

>> **fg** (and then type the name of service)

 Run docker using this command –

>> **docker run -dit ubuntu:14.04**

i - interactive

t – terminal

DAY 15 (20 June)

DOCKERS –

- ✚ Every software has use cases, docker also have.

GUI programs does not work in docker.

Main target of docker is servers.

- ✚ Use case of docker -

- Docker is mainly used for run programs on server side.
- Docker gives os but we can't make partitions in OS.
- It is not a bad side because. It's not made for this.

- ✚ Companies customises their OS before use. They don't require many things like GUI.

Less program → less RAM used, more security.

More features means more chances to hack, more vulnerabilities.

- ✚ Therefore, docker is made for some specialized purpose.

Docker only gives you computing unit (RAM and CPU).

Example –

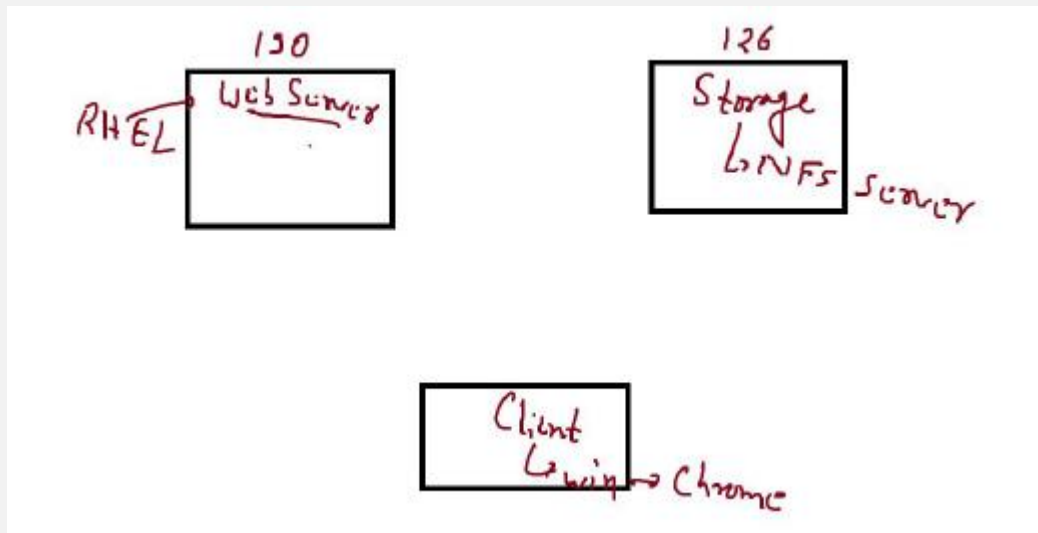
- Firefox uses storage but not manage storage.
- Same is with docker, uses storage but doesn't manage storage.

- ✚ We use OS to run program. Docker gives us an environment to run our program.

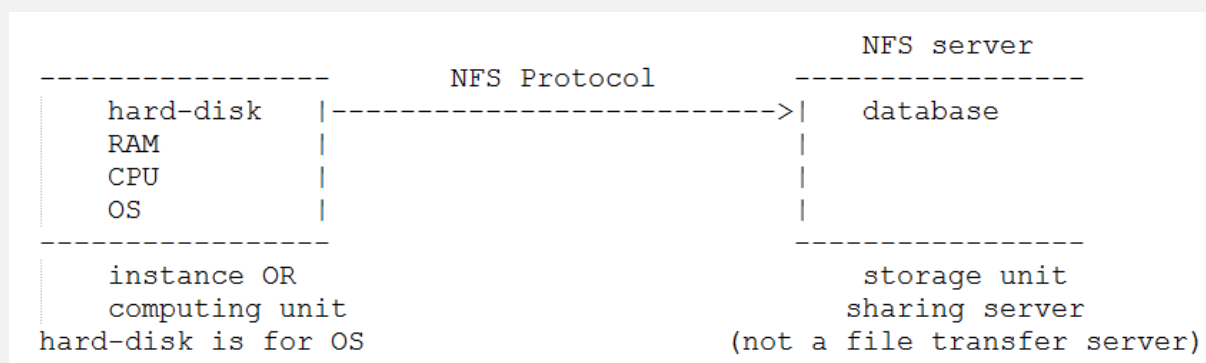
- Companies have their storage separately for their websites.

Their system run only servers and their storage units are different.

It's a good practice. Suppose if there is some problem in servers then data may loss if data is stored on same system.



- Our project will also not store data on server system. We'll use separate storage for that.



- Most important resource is our website data, which is in /var/www/html.

- NFS server will be file sharing server not file transfer server.

✚ Difference in file transfer and file sharing -

- File transfer copies data firstly in hard disk and stores permanently in client system and then we use that data.
- File sharing → We have data in our hard-disk, our client will get data but didn't copy on his hard-disk
- For example - We have some video in pen-drive. When we play video that plays from pen-drive, this is a type of file sharing.
- This is difference b/w file sharing and file transfer.
- Pen-drive is an example for file sharing, FTP is used for file transfer.

✚ LFS (Local File System) - Space is locally available.

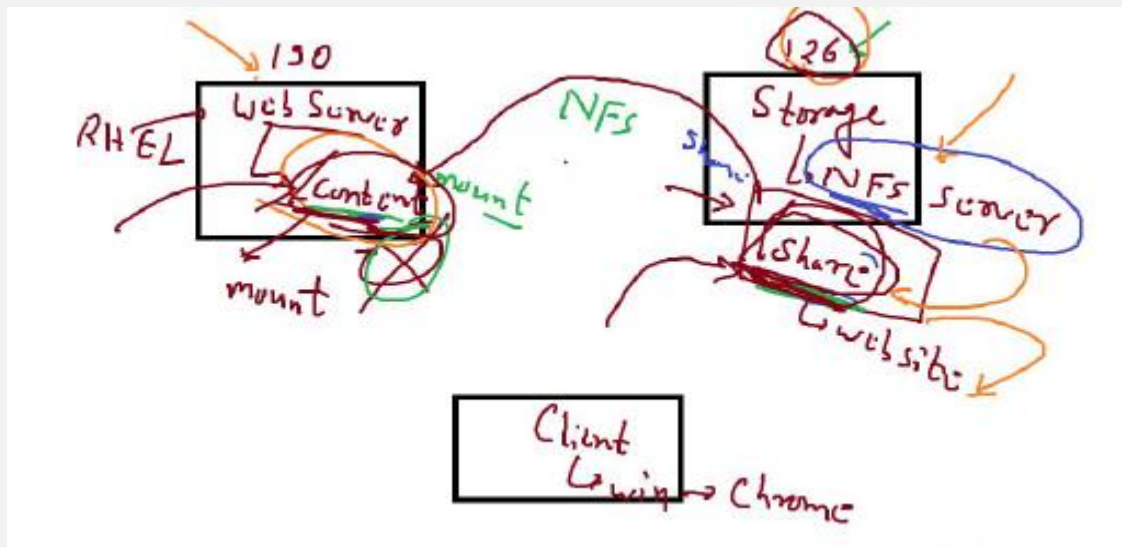
NFS (Network File System) - Space is coming from a network file system.

✚ So using NFS in web server is a good design.

Data will load on web server RAM.

✚ We'll have this kind of setup -

- web server → IP 192.168.43.44 (docker in future)
NFS server → storage unit (may be pen-drive), other OS
- First system (web server) have web server but not storage.
Second system (NFS) have website but not have web server.



Procedure -

✚ Set up a web server on IP 192.168.43.44 -

1. Make new .conf file

```
>> /etc/httpd/conf.d
```

2. Write this -

```
DocumentRoot /content
```

```
<directory /content>
```

```
    require all granted
```

```
</directory>
```

3. Run these commands -

```
>> systemctl stop firewalld
```

```
>> setenforce 0
```

```
>> service httpd restart
```

✚ Set up NFS server on IP 192.168.43.114 -

1. make a new page noserver.html in following directory

```
>> mkdir /share
```

```
>> cd share
```

2. Install software -

s/w - nfs utils

3. configure file -

conf file- /etc/exports

Write folder name in this file

folder name	to whom you want to share
-------------	---------------------------

/share	192.168.43.44
--------	---------------

4. Start services -

```
>> systemctl restart nfs
```

✚ On server side, IP 192.168.43.44 -

Make a folder named content. Out from this folder before mounting or unmounting

```
>> mount 192.168.43.114:/share /content/
```

✚ Now storage unit data or second system's data is in content folder of first system.

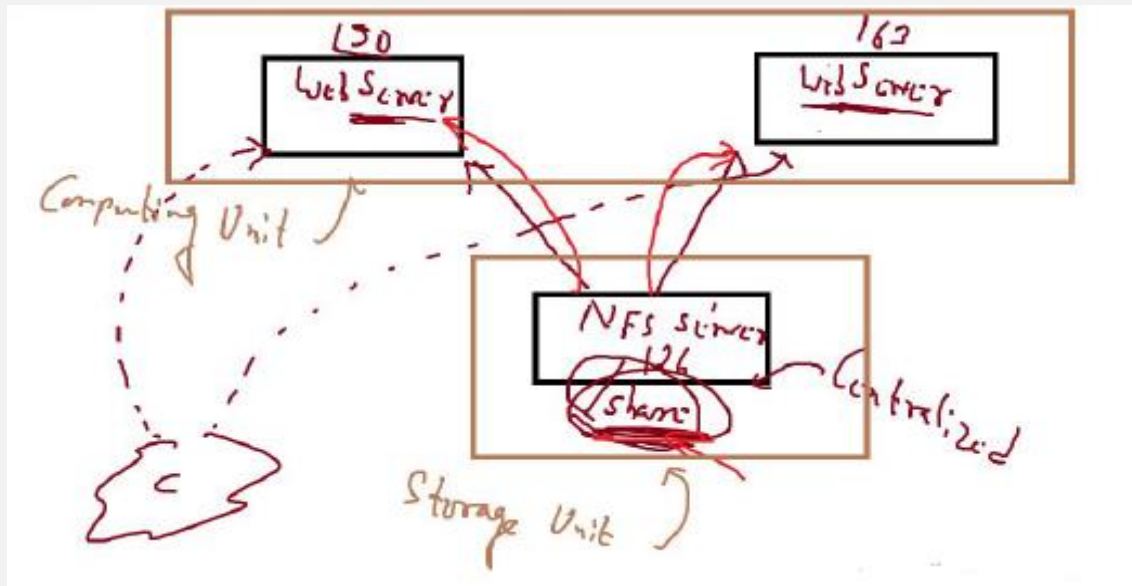
Now client will be able to see this data when he open a web site.

✚ Every server have some limit to connect clients. Therefore, we make a new server and mount again storage unit to serve more client.

✚ Suppose we have new web server on IP 192.168.43.33

Now change NFS server config file and write this -

/share 192.168.43.44 /192.168.43.33



✚ If we change now our website, content will be same on every server. We don't have to copy web pages on every server system. Otherwise, it may be possible that different client will see different website on that point of time.

So, any change will reflect on every web server.

If we're not using storage centralisation, then we have to copy data in every server.

How to configure server, using docker -

✚ Start docker and it's OS -

```
>> systemctl restart docker  
>> systemctl enable docker  
>> docker run -it centos:latest
```

✚ Prints the output of date command

```
>> echo $(date)
```

✚ To stop all running OS -

```
>> docker ps -q //shows id of every running OS  
>> docker stop $(docker ps -q)
```

✚ To permanent remove -

```
>> docker rm id
```

✚ To terminate all containers -

```
>> docker rm $(docker ps -a -q)
```

✚ To remove container while running -

```
>> docker rm -f $(docker ps -a -q)
```

✚ Docker gives unique name to every container with a unique id.

- ✚ We can give docker OS a name when starting-

```
>> docker run -it --name adarsh centos:latest
```

Running server in docker -

- ✚ Command prompt is docker host.

- ✚ yum is already configured in centos docker image.

- ✚ Install httpd software -

```
>> yum install httpd -y
```

- ✚ Create a webpage -

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

```
>> vi index.html
```

- ✚ Starting services -

- If we use systemctl to start the services, this will not work and gives an error.
- systemctl doesn't work in docker.
- In actual redhat system, when we start a service it runs a script in background. That script start daemons.
- To find the path of that script, check status of service

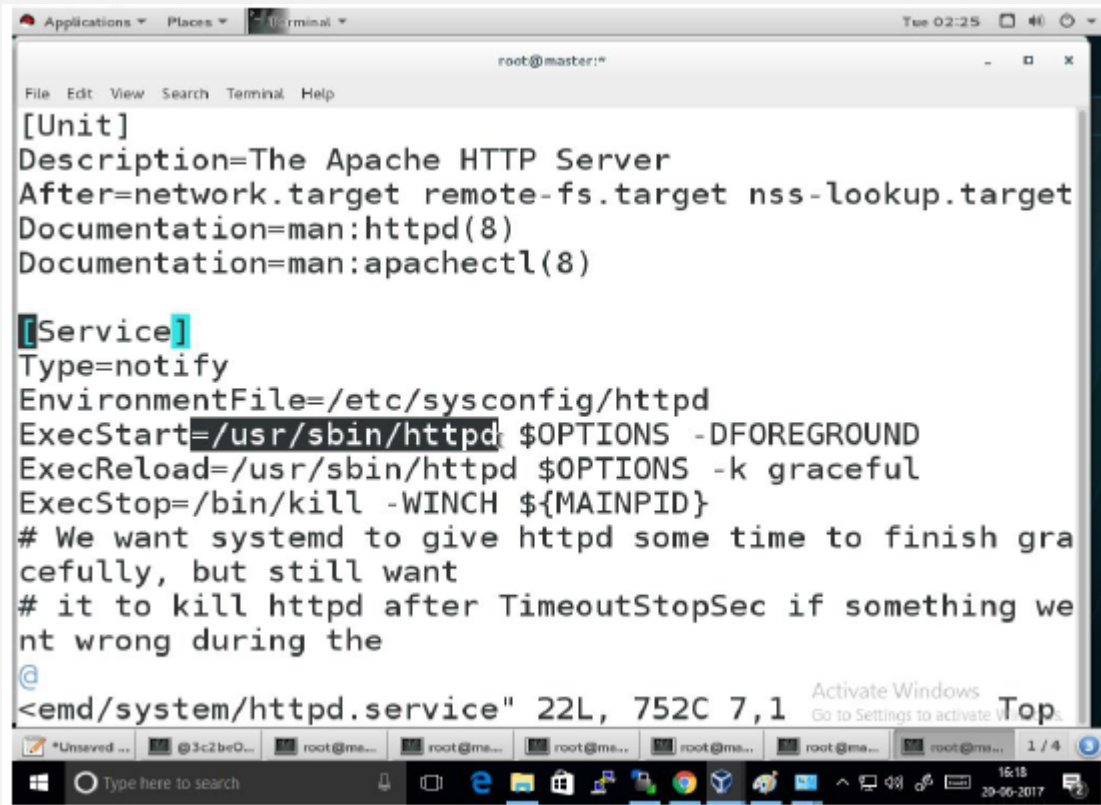
```
>> systemctl status httpd
```

Loaded option shows scripts path.

- In that file, we have a line which starts service -

```
ExecStart = /usr/sbin/httpd.....
```

This command actually starts httpd server.



```
root@master:~# cat /etc/systemd/system/httpd.service
[Unit]
Description=The Apache HTTP Server
After=network.target remote-fs.target nss-lookup.target
Documentation=man:httpd(8)
Documentation=man:apachectl(8)

[Service]
Type=notify
EnvironmentFile=/etc/sysconfig/httpd
ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
ExecReload=/usr/sbin/httpd $OPTIONS -k graceful
ExecStop=/bin/kill -WINCH ${MAINPID}
# We want systemd to give httpd some time to finish gracefully, but still want
# it to kill httpd after TimeoutStopSec if something went wrong during the
@
<emd/system/httpd.service" 22L, 752C 7,1
```

✚ Service is running or not, can be checked by -

```
>> ps -aux | grep httpd
```

✚ So we don't require systemctl we can directly start our web server -

```
>> /usr/sbin/httpd
```

This will start the web server.

✚ **ifconfig** doesn't work in docker.

So install software, which gives ifconfig command

It can be checked in actual red hat system -

```
>> rpm -qf /usr/sbin/ifconfig
```

This comes from net-tools

Install net-tools in docker os.

- ✚ Making image of created web server -

```
>> docker commit webserver apacheimg:v1
```

Name of OS is webserver.

This image can be share with exact setup.

```
# docker run -it --name webserver centos:latest

# yum install httpd -y
# rpm -q httpd
# cd /var/www/html
# vi index.html
web content

# /usr/sbin/httpd

# yum install net-tools
# ifconfig
```

- ✚ To start httpd service when container starts-

```
>> vim /root/.bashrc
```

Write /usr/sbin/httpd in this file.

- ✚ To save created image-

```
>> docker save apacheimg:v1 -o mywebserver.tar
```

- ✚ To copy a file in container -

```
>> docker cp /root/form.txt mycontainer:/
```

- ✚ Program run on docker os takes same RAM as it takes. It will be different from os RAM memory.

DAY 16 (21 June)