Day 1 Linux Deep Dive

- **File system structure**
  /etc - configure files. contains configuration required by all programs.

  /var - Variable files. Var stands for variable files.

  /home – Home directories. All users use home directories to store their personal files.

  /opt – Optional add-on Application. Contains add-on applications from individual vendors.

- **Permissions**
  File permission
  0- none
  1-execute only
  2-write only
  3-write and execute
  4-read only
  5-read and execute
  6-read and write
  7-set all permission

Letters	Definition
`r`	Read permission
`w`	Write permission
`x`	Execute permission

chmod – The chmod command allows you to change access rights to files and directories.

Syntax: chmod[option][MODE]Filename

Option available chmod commend in Linux

Options	Description
`-R`	Apply the permission change recursively to all the files and directories within the specified directory.
`-v`	It will display a message for each file that is processed. while indicating the permission change that was made.
`-c`	It works same as `-v` but in this case it only displays messages for files whose permission is changed.
`-f`	It helps in avoiding display of error messages.
`-h`	Change the permissions of symbolic links instead of the files they point to.

Modes in chmod commends in Linux

Symbolic modes

Operators	Definition
`+`	Add permissions
`-`	Remove permissions
`=`	Set the permissions to the specified values

Reference	Class
u	Owner
g	Group
o	Others
a	All (owner,groups,others)

chown – the chown command is used to change the owner or user of a file or directory. This is an admin command; only the root user can change the owner of a file.

Syntax: chmod[option]new_owner[:new_group]filename

Options in chown commend

-R - change the permission on files that are in the subdirectories of the directory that you are currently in.

-c - change the Permission for each file.

-f - prevent chown from displaying error messages when it is unable to the ownership of a file.

umask- the umask is a system variable that encodes a mask for file permission to be used when a file is created.

The value is a three-digit octal value. Each digit is the result of an ANDing value from 1,2 or 4

Syntax: $umask 543

Permissions	Octal Value	Binary Value	Description
—	0	000	No permission
–x	1	001	only permission to execute
-w-	2	010	only permission to write
-wx	3	011	permission to write and execute
r–	4	100	only permission to read
r-x	5	101	permission to read and execute
rw-	6	110	permission to read and write
rwx	7	111	permission to do all three, i.e. read, write and execute

- **Process Management**

ps command – ps command is used to report the process status. Ps is the name for process status.

Syntax: ps[option]

Ps commend option

-a – list information about all processes most frequently requested. all those except process group leader and process not associated with a terminal.

-A - list of information for all processes.

-d – list information about all processes except session leader.

-e – list information about every process running now.

-f – generates a full listing.

-j – print session ID and process group ID.

-l – generate a long listing.

Kill commend-kill commend is used to kill the background process.

Syntax: kill[-s][-l]%pid

Options

-s -specify the signal to send. the signal maybe given as a signal name or number.

-l – write all value od signal supported by the implementation, if no operant is given.

-pid - processed or job id.