**COMMANDS**

**Clear** : To clear the screen

**pwd** : present working directory

**mkdir** : command is used to create a directory/folder

Example: mkdir directoryname

**mkdir -p** : command is used to create a folder/directory if it doesn’t exist

**cd**  : command is used to change directory

Example: cd folder name

**"cd .."** – command is used to come out of the directory/ back to previous folder

**touch**  - command is used to create a file

Example: touch filename

**To create multiple files/directories**

Folder: mkdir foldername1 foldername2 ….

File: touch filename1 filename2 ….

l**s**  - command is used to check the directory and file/ used to display in the long list

if it starts with "d" - directory

if it starts with "-" – file

if it starts with “l” link

**vi**  - command is used to edit the file

Example: vi filename

**wq!** - save and quit the program

**q!** - quit without saving

**cat** - to display the content of the file.

**rm -rf:** Command is used to delete the folder. Example: rm -rf folder name

**rm:** command is used to delete the file. Example: rm filename

**To delete multiple files/directories**

Folder: rm -rf foldername1 foldername2 ….

File: rm filename1 filename2 ….

**Deleting files through cases**

Case 1: by names – rm filename

Case 2: by extensions – rm \*.ext (.html, .xls etc)

**Command to set warning before deleting file/directory.**

rm -i filenames

rm -rf -i foldernames

**rm -rf \*:** to delete the complete directory

**ls -lt**: used to display recently created files at the top order

**ls -lrt**: used to display recently created files at the bottom

**ls -a**: used to display hidden files (hidden files start with ".filename")

**du -sh folder name**: used to display the size of the file/ folder(current file)

**du -sh \*** -used to display the size of all the directories/ folders

**du** - disk usage **s**- size **h**- human-readable format

**df -h**: used to check the size/memory of the system

**df -h .** : used to check the size /memory of the current drive

df - disk free

**ls foldername** - to display the content of the folder

**Copy files/folder:**

**cp**: used to copy a file/folder from source to destination r - recursive

**cp pathofsource pathofdetination** - used to copy the file from source to destination

**cp -r foldername destination**- used to copy the folder from source to destination

**Move/Rename files or folders**

mv - cmd is used to renafilesile or directory and used to move the

file or directories from source to destination path.

mv oldfilenmae newfilename - renaming a file

mv olddirectoryname newdirectoryname - renaming a directory

mv directory pathofdestination/: to move the files or directories to destination

(escape): set nu - used to set the line numbers in a side

(escape)/patternname - search pattern

(escape)dd - deleting the line

ls -l filename: to check username/ownership of file

ls -l foldername: to check username/ownership of directory

**Change ownership**

ex : chown -R root:root foldername

sudo chown -R root:root foldername - chnage username with root permission

chown - used to chnage the usernmae and root name of a file and directory

chown -R usernmae:groupname dircetoryname- used to chnage ownerhsip of folder name

chown usernmae:groupname filename - used to chnage ownerhsip of file name

"./ "-> current directory"

"../" ->previous directory

"../../" ->previous to previous directory

**User** : **Group** : **others**

R-4 W-2 X-1

**chmod : used to change the permission of a file or directory**

chmod permission filename ex: chmod 764 filename

chmod -R permission directoryname ex: chmod -R 761 directoryname

r-read; w-write; x-executable;

history >file

**head**: to display first n lines of a file

ex: head filename (default n-10)

head -5 filename: first 5 lines

head -3 filename: first 3 lines

**tail:** to display last n lines of a file

ex: tail filename

tail -2 filename: last 2 lines

**wc : word count**; used to count no. of lines, no of words, no. of characters

l- lines

w - words

c - characters

wc filename: no of lines, words, characters

wc -l filename: no of lines

wc -w filename: no of words

wc -c filename: no of characters

**>: redirect operator (>)** - used to copy output of one command to a new file.

If the file already exists it will over write.

ex: wc file > filename ; wc file > file1 ; wc -l file > file1

**>>: append symbol Apend (>>)** - used to apend the output of command to the end of a file.

ex: wc file >> filename ; wc file >>file1

not only "wc" any other command like "ls" etc can be used along with above operators

**pipe (|)** - used to give or pass output of one command as an input for next comment.

**xargs:** the output of the previous command is passed as a sequence (line by line) to the input for next command

xargs rm -rf

ex: file = filename

to display 7th line of a file

head -7 file | tail -1

to display 5th line of a file

head -5 file | tail -1

to display 5th and 6th line of a file

head -6 file | tail -2

to count no. of characters in a 7th line of a file

head -7 file | tail -1 | wc -c

to count no. of words in 5th &6th line of a file

head -6 file | tail -2 | wc -w

to count no. of characters in a 8th line of a file and copy to a new file

head -8 file | tail -1 | wc -c >file1

If an empty line occurs in between file, what would be the output? Answer: => 1

**hostname:** used to display server name

uname -v: check linux version

uname -a: to dispay all the details of linux

**To print the statements:**

echo \* : this command does the same job as "ls" command do

echo : used to print the statements :syntax : echo "welcome" ; output = welcome

echo -e: used to print the multiple statements. Example echo -e "good morning \n weclome"

\n: will display the statements in two lines: outpt will be displayed as

good morning

welcome

ex: echo -e "good morning \t welcome"

\t : t => tab - will display the statements in with space(tab) output: good morning welcome

**grep: is used to search a pattern in a file**

syntax: grep pattern filename or grep "pattern" filename

recommended to use pattern within "" ie, 2nd syntax

search specific pattern from file (case sensitive)

ex: grep "pattern" filename

grep -e "pattername1" -e "patternname2" filename - used to search multiple patterns

example for case insensitive => grep -i "pattername" filename

example to search specific word in a file=> grep -i -w "pattern" filename

the pattern in above syntax has to be a word from file (case insensitive)

grep -i -c "pattern" filename => used to count no. of lines which contain the patterns

grep -i "patternname" \* => used to search pattern in all the files which is in current directory (directory will nt be considered)

grep -i -R "patternname" \* => used to search all the patterns in directory and in its sub directory (case insensitive)

grep -i -R -l "patternname" \* => same as ( "grep -R -l "patternname" \* ) but case insensitive

grep -l "patternname" \* => it will list file name which contain the pattern in current directory.

grep -R -l "patternname" \* => it will list file name in the present directory as well as sub directory only if the file contains the pattern

grep -i "^patternname" \* => it will print the line which starts with the pattern

grep -i "patternname$" \* => to print the lines which ends with letters/patterns

grep -i "^$" \* => to print the empty lines in a file

grep -i -c "^$" filename => to count the empty lines of a file

grep -v "pattern" filename => to print all the lines which does not contain a pattern

grep -n "pattern" filename => to print lines along with line number which contain a pattern

**SED command (stream editor)**

It is used to replace a string by another string in a file

Sed ‘s/pattern/pattern2/g’ filename

S indicates substitute - replace a string only in a console not in the file

Sed -i ‘s/pattern/pattern2/g’ filename

i indicates insert- it will replace a file in a string

Sed -i ‘s/pattern/pattern2/Ig’ filename

Capital I indicates case insensitive

Sed -i ‘$s/pattern/pattern2/Ig’ filename – to change the pattern at the last line of the file

Sed -i ‘2s/pattern/pattern2/Ig’ filename - 2S : it will replace only in a second line

Sed -i ‘2, 5s/pattern/pattern2/Ig’ filename - 2, 5S means it will replace in 2 thru 5 lines

Sed -i ‘1s/pattern/pattern2/2g’ filename: It will replace from second occurrence of a file in the 1st line

Sed -i ‘1s/pattern/pattern2/2’ filename: It will replace only the second occurrence of a file in the 1st line

**How to delete a line using sed commands**

Sed -i ‘4d’ file name : it will delete 4th line of a file

sed -i ‘2,5d’ file name : it will delete lines from 2 to 5 of a file

sed -i ‘2d;5d’ filename : it will delete 2nd and 5th line of a file

sed -i ‘/^$/d’ file name : it will delete a empty line in a file

sed -I ‘/patternname/d’ file name : it will delete a line which contains a pattern

**How to print the lines using sed command**

sed -n ‘4p’ filename : to print the 4th line of a file

sed -n ‘2,5p’ file name : to print the 2 to 5th line of a file

sed -n ‘3p;5p’ filename : to print 3rd and 5th line of a file

sed -n ‘$p’ file name to print the last line of a file

**Cut is used to cut the files (display) in column wise**

cut -d “ “ -f1 filename : d : display the first column of a file

delimiter f :column 1: first column

cut -d “ “ -f1-3 filename : display from 1st through 3rd column of a file

cut -d “ “ -f1,3 filename ; display 1st and 3rd column of a file

**Disadvantages,**

1. we can’t display in row wise
2. If we leave more space between the columns, then we will not get desired output.

**awk : this command is used to cut (display) the files in both column and row wise**

awk -F “ “ ‘{print $2}’ filename : To cut & display the 2nd column of a file

awk -F “ “ ‘{print $NF}’ filename : : To cut & display the last column of a file

awk ‘{print $NF}’ filename : To cut & display the last column of a file

awk ‘{print $(NF -1)}’ filename : To cut & display the last but one column of a file

awk ‘{print $2, $4}’ filename: To cut & display the 2nd and 4th column of a file

awk ‘ (NR>1) {print $2, $5}’ filename : display from 2nd row and print 2nd & 5th column in a file

awk ‘ NR==2, NR==3 {print $NF}’ filename : it will display rows 2 to 3 of last column

awk ‘ NR==2; NR==4 {print $0}’ filename: it will display only 2nd and 4th row of a file

**Free**: it is used to check the system memory (RAM)

Free -m mega bites free -g giga bites

**Find : it is used to find the location of a file or directory**

Find / -iname filename / indicates searches in root directory -I case insensitive

Find . . (dot) indicates the current directory

Find only the files with the file name Devops in the current directory

Find . -type f -iname Devops

Find only the directories with day1 in root directory

Find / -type d -iname day1: use sudo

How to list the files which are created or modified 3 months ago

Find . -type f -mtime +90

How to list the files which are created or modified within 3 months

Find . -type f -mtime -90

How to list the files which are created or modified within 60days and 30 days ago

Find . -type f -mtime -60 ! -mtime -30

List the files which are modified within 10 min

Find . -type f -mmin -10

List the files which are modified 10 min ago

Find . -type f -mmin +10

List the files which have permission 777

Find . -type f -perm 777

List the files which are empty

Find . -type f -empty

List the files which have a size is greater than 4k

Find . -type f -size 4k

To restrict the automatic recursive, we use maxdepth

Max depth (number) indicates it will search the number of the levels you gave.

Find . -maxdepth 2 -type f -iname filename

**Sudo apt install tree**

Su (superuser or switch user): it is used to switch (login) as another user

Example: su -username

Sudo: it is used to run the command with root permissions

Example: to switch to root user sudo su –

Exit: to logout

**Link: it is a shortcut to the file.**

Meanwhile, if we make any changes in the link also it will reflect in the file

**There will be two types of links**

1. Soft link (symbolic link): it is a shortcut to a file if we make any changes to the original file it will reflect in the link, if the file is deleted then the link doesn’t work

Syntax: **ln -s filename linkname**

1. Hard link: it is a short cut to a file if we make any changes to the original file it will reflect in the link, if the file is deleted then the link will work

Syntax: **ln filename linkname**

**Diff b/n Soft link and hard link**

1. If I delete the actual file, the soft link will not work
2. If I delete the actual file, the hard link will work, because it points to the inode of a file. Inode is the memory address (unique identification number for file)

**Ps (process status):** ps is a command used to check the current process running in the background on a system.

How do you check whether a specific process is running or not.

Ps -ef : it will display running process

**Ps -ef | grep -I “processname”** or ps -C processname

**How do you list process started by a specific user**

Ps -u username

**Kill: kill is used to stop the process forcefully**

Kill -9 pid

Pid – process id

**How your going to stop the process gracefully**

Sudo service service/processname stop

Sudo service service/processname start

Sudo service service/processname restart

Sudo service service/processname status

**List the files which are started with “t”**

1. Ls t\*
2. ls |grep -i “^t”

**How to you login (ssh) or login to remote server**

Ssh secure shell

ssh username (root/ubuntu/any)@server

**port number** :

telnet 23,

http 80,

https 443,

dns 53,

ssh 22

scp 22

tcp 25

**scp :** it is used to copy a file or directory from one server to another server

**scp (secure copy) :** scp filename username@servername:/home/ubuntu/

to copy a directory scp -r directory name user@servername:/home/ubuntu

**rsync:** it is also used to copy a file/directory from one server to another server

**Diff b/w rsync and scp**

While copying data from one server to another server if copying stopped in between due to some network issue once the system is back if I use the scp command it will start copying from the beginning.

If I use the rsync command, it will start copying from where it was stopped

**Uptime:**

From how long the system/server is up and running: **uptime**\_ is used to check the load average and also to check the server is up & running

**How to check other /remote servers/systems are up & running**

Ping **ip address of server**

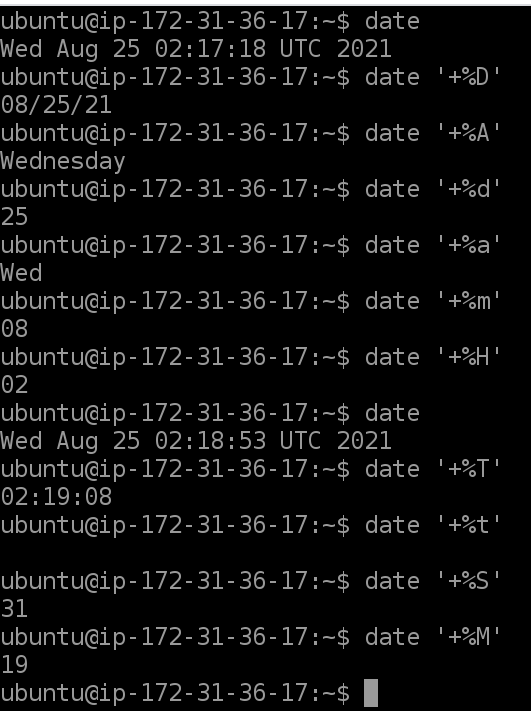
**Load average:** it is the average number of processes that are either in runnable (using CPU, waiting to use CPU) or uninterruptible (waiting for IO access)

IO: Input output

**Zero (0) : is no load and 1 : is fully loaded**

Load average displays average for the past 1, 5 and 15 min respectively.

**Date: it is used to display the current system date and time**



**Nohup (Nohang up)** ; while running a script if execution is stopped due to some network issue once network is back it resumes the execution from where it was stopped.

nohup command name &

how do you run the script at the background: use and symbole (&) at the end of the command

**example:** ls &, cat &

**htop/ top:** **how you’re going to check which process is taking more memory**

Sudo apt(ubuntu)/yum(redhat) install packagename

How to install the packages for ubuntu/debian, apt install package name

For redhat/centos, yum install packagename

**Netstat: it is used to see the available ports on the system and listening(used) ports**

net -tools => netstat -a : listening(used) and open ports

netstat -at : display only tcp ports

netsat -au : to check only udp ports

netstat -l : only listening ports

netstat -lt : listening tcp ports

How do you check whether a specific port is open or not

Netstat -ntulp | grep “portnumber”

**telnet**: command is used to check if the remote server port is open or not

Synatx: telnet servername port value

**whoami:** display the user in which you login

how do you list open files?

**lsof:**

find the process running on a specific port

Syntaxt:lsof -I portnumber

Example: lsof -i tcp:22

List the files used by a specific process

Lsof -p pid(process id)

List the open files for a specific user

Lsof -i -u username

**How do you set up a password-less connection between the server**

**Server 1:** create a public key and private key using **the ssh-keygen** command

Copy the public key

**Server 2:** go to server 2 and copy the public key to the authorized key under the .ssh folder

Next time we log in it will log in without a password

**How to sort lines and numbers:**

Sort, Sort -r, Sort -n, Sort -nr