

# LINUX DAY-1

There are different types of commands in linux:

- System commands
- Hardware commands
- Network commands
- File commands
- Search Commands
- User commands
- Permission commands

Whenever you connect with instance, we logging as ec2-user,

Ec2-user is the default user in amazon-linux

But we have to login as **root user**

**Because Root user is the ultimate king of the linux**

to login from ec2-user to root user

**commands:**

sudo -i (or) sudo su -

sudo = super user do

To logout from root user

**command:** exit

**SYSTEM COMMANDS:**

uname : used to get OS of our instance

uname -r : used to get kernel version of OS

uname -a : used to get full information of OS

uptime : used to get since how long our server is in running state

uptime -p : it displays only time

uptime -s : it displays about date & time of our system run time

hostname : used to get hostname of the system

hostname -i : used to private ip address of a system

ip addr : used to private ip address of a system

ip route : used to private ip address of a system

ifconfig : used to private ip address of a system

hostnamectl set-hostname tcs-swiggly-dev : used to change hostname

date : used to display todays date

date +"%d" : it displays only date

date +"%m" : it displays only month

date +"%y" : it displays only year

date +"%H" : it displays only hours

date +"%M" : it displays only Minutes

date +"%S" : it displays only seconds

date +"%D" : it displays date (mm/dd/yyyy)

date +"%F" : it displays date (yyyy-mm-dd)

date +"%A" : it displays only day of the week

date +"%B" : it displays only month of the year

timedatectl : used to get timezone of our system

timedatectl set-timezone Asia/Kolkata : used to set Indian timezone for our system

whoami : used to display in which user we login

who : displays the no of users login info

## **HARDWARE COMMANDS:**

cat /proc/cpuinfo : used to get cpuinfo

cat -----> is a linux command which is used to read the data in a file

/proc ---> it is a directory which contains processor of our system

cpuinfo -----> it is a file which contains cpu info

lscpu : used to get cpuinfo

cat /proc/meminfo : used to get RAM info about our system

free : used to get RAM info in KB

free -m : used to get RAM info in MB

df -h : used to get storage info about our system

fdisk -l : used to get list of volumes attached to our system

# LINUX CLASS-2

## FILE COMMANDS:

To create a file in linux : touch filename

To see list of files : ll (or) ls

ll vs ls

ll : full info about files

ls : it gives only file names

To create multiple files : touch aws azure gcp

To create files in sequential order : touch file{1..7}

To see latest files on top : ll -t

to see the files in reverse order : ll -r

to see all files (including hidden) : ll -a

## TO REMOVE FILES:

To delete files with permissions: rm filename

To delete multiple files with permissions : rm aws azure gcp

To delete a file without permissions : rm -f filename

To delete multiple files without permissions : rm -f aws azure gcp

To delete files in sequential order : rm -f file{1..7}

To delete all files which are started with A name : rm -f a\*

TO delete all text files : rm -f \*.txt

## DIRECTORIES (FOLDERS) :

To create a directory : `mkdir (make directory)`

To create a multiple directories : `mkdir aws azure gcp`

To create sequential directories : `mkdir folder{1..6}`

To remove empty directories : `rmdir foldername`

To remove multiple empty directories : `rmdir aws azure gcp`

To remove empty directories in sequential order : `rmdir folder{1..5}`

To remove all empty directories : `rmdir *`

To remove all files and folders : `rm -rf *`

## INSERT DATA IN A FILE:

To read the data in a file : `cat filename`

To insert some data in a file : `cat>filename`

To append some data in a file : `cat>>filename`

To save the data : `ctrl + d`

To display the data including line numbers : `cat -n filename`

To print top 10 lines of a file : `head filename`

To print last 10 lines of a file : `tail filename`

To print a particular range : `sed -n '5,17p' filename`

To print the data in reverse order : `tac filename`

To read the data from multiple files : `cat file1 file2`

To read the data from multiple files : `more file1 file2`

To get no of lines, words & letters in a file : `wc filename`

24 198 1047 aws

24 = no of lines

198 = no of words

1047 = no of letters

aws = filename

To get only no of lines in a file : `wc -l filename`

To get only no of words in a file : `wc -w filename`

To get only no of characters in a file : `wc -c filename`

# LINUX DAY-3

## DIRECTORIES :

To change directory : `cd foldername`

To get the present working directory : `pwd`

To go to root directory : `cd (or) cd ~`

To go back to previous folder : `cd -`

To go to one step back folder : `cd ../`

To go back to 2 steps back : `cd ../../`

To create a directory inside a directory : `mkdir folder1/folder2`

To check the list of files & folders in folder1 : `ll folder1`

To create parenting directories : `mkdir -p folder1/folder2/folder3/folder4`

To create a file inside a folder : `touch foldername/filename`

## COPY COMMAND: `cp source destination`

command: `cp file1 file2`

explanation: data from file1 copies to file2

If we use this command the data will gets overwrite from file2, To avoid that we can use `cat` command

`cat file1 >> file2 ----->` this command will only copies the data, it will not overwrite the data

Copy the file to directory : `cp filename directory`

## MOVE COMMAND: `mv source destination`

command : `mv file1 file2`

We will use this command to rename the file as well

Move the file to directory : mv filename directory

To compare multiple files at a time : cmp file1 file2

To get the difference from multiple files : diff file1 file2

ANALYSIS ABOUT THE FILE:

-rw-r--r-- 1 root root 41 Sep 19 15:46 aws

TYPE OF THE FILE:

Regular files : - (hyphen)

Directory files : d

Character files : c

Blocked files : b

PERMISSIONS OF A FILE:

r : read ----> 4

w : write ----> 2

x : execute -----> 1

hyphen (-) : nothing -----> 0

user permissions : rw- : 4 + 2 + 0 = 6

group permissions : r-- : 4 + 0 + 0 = 4

others permissions : r-- : 4 + 0 + 0 = 4

-----



overall file permissions : 644

-----

example:

-w- : 0 + 2 + 0 = 2

rw- : 4 + 2 + 0 = 6

--x : 0 + 0 + 1 = 1

**ACCESS CONTROL LIST:**

Here ACL should be 1 for the file

ACL should be 2 for the folder

**OWNERS OF A FILE (root root)**

root : user

root : group

**NO OF CHARACTERS IN A FILE (41)**

**FILE CREATION DETAILS :**

Sep 19 15:46

**FILE NAME :** (aws)

# LINUX CLASS-4

VIM EDITOR:

IT IS USED TO MODIFY THE DATA IN A FILE. IT HAS 3 MODES

1. COMMAND MODE
2. INSERT MODE
3. SAVE & QUIT MODE

**COMMAND MODE:** This is the default mode in vim editor. This is used to perform some actions like copy the data, delete the data, and undo, redo, search for a word and also we can move to a particular line.

gg : used to go to 1st line of a file

G : used to go to last line of a file

5gg : used to go to 5th line of a file

:19 : we can move to 19th line of a file

:set number : used to set a numbers of a file

yy : copy a line

3yy : used to copy 3 lines from our cursor

p : used to paste the data

5p : used to paste 5 times

dd : delete the entire line

5dd: used to delete 5 lines at a time

u : used for undo

ctrl + r : used for redo

/word : used to search for a word

?word : used to search for a word in a file

:%s/oldword/new-word/ : used to replace a single occurrence in a line

ex: my name is **mustafa**, **mustafa** is a devops engineer, **mustafa** is from hyd.

:%s/mustafa/flm

if we use this command the the o/p: my name is **flm**, **mustafa** is a devops engineer, **mustafa** is from hyd.

:%s/mustafa/flm/g

if we use this command the the o/p: my name is **flm**, **flm** is a devops engineer, **flm** is from hyd.

because i used g (global)

**INSERT MODE:** It is used to insert the data.

But by default we are in command mode, if you wish to go to command mode to insert mod you can use small i (i)

To go back to command mode : esc

A : used to go to the end of the line

I : used to go to the starting of the line.

O : used to create a new line (up)

o : used to create a new line (down)

**SAVE & QUIT MODE:** This mode is save the data and quit from vim editor

:w : used to save the data in a file

:q : used to quit from vim editor

:wq : used to save & quit at a time

:q! : used to quit from vim editor forcefully

:wq! : used to save & quit forcefully

vim -o file1 file2 : used to view 2 files (one by one view)

vim -O file1 file2 : used to view 2 files (side by side)

ADD SOME TIPS AS A LINUX USER:

ctrl + w : used to delete a single word in a command

ctrl + u : used to delete entire command

ctrl + a : used to go to starting of the command

ctrl + e : used to go to ending of the command

history : used to get history of our commands

!history\_number : used to perform same command from history

ctrl + k : used to cut the command until the cursor

# LINUX CLASS-5

## USER COMMANDS:

To see list of users : `cat /etc/passwd`

To create a user : `useradd username`

To switch to another user : `su - username`

To set a password to the user : `passwd username`

**Note: Whenever we create any user, then one folder will gets created in /home directory**

To create a user without directory : `useradd -M username`

To set expiry date to an user : `useradd -e yyyy-mm-dd username`

To get user expiry details : `chage -l username`

To delete a user : `userdel username`

To delete a user with directory : `userdel -r username`

## GROUP COMMANDS:

To see list of groups in a system : `cat /etc/group`

**Note: whenever we create any user, then group will gets created automatically**

**and when we delete a user, then group also will gets deleted automatically**

if you want to create only group by your own the the command is `groupadd groupname`

To create a group : `groupadd groupname`

To delete a group : `groupdel groupname`

To add a user in a group : `usermod -a -G groupname username`

## **PERMISSION COMMANDS:**

### **CHANGING OWNERS OF A FILE**

To change user of a file : `chown username filename`

To change group of a file : `chgrp groupname filename`

To change user & group at a time : `chown username:groupname filename`

To change user & group at a time to multiple files : `chown user:group file1 file2 file3`

To change user & group at a time to all files : `chown user:group *`

To change user& group of a folder : `chown user:group foldername`

To change user & group of a folder along with files : `chown -R user:group foldername`

To change user & group of a files which are present in folder : `chown user:group foldername/*`

### **TO CHANGE PERMISSIONS OF A FILE :**

r ----> read = 4

w ----> write = 2

x ----> execute = 1

hyphen (-) -----> nothing = 0

To change permissions of a file : `chmod 777 filename`

To change permissions of a multiple files : `chmod 542 f1 f2 f3`

To change permissions of a all files : `chmod 123 *`

To change permissions of a folder : `chmod 561 folder`

To change permissions of a folders along with files : `chmod -R 777 foldername`

To change permissions of a files inside the folder : `chmod 345 foldername/*`

## **SEARCH COMMANDS:**

**GREP command:** Global Regular Expression Print

This command is used to search for a word in a file.

Syntax: grep “word” filename

To search for a word in a file : grep “word” filename

To get the word along with line numbers : grep -n “word” filename

To search with case-sensitive : grep -i -n “word” filename

To get no of occurrences : grep -i -c “word” filename

To search for multiple words in a single file :grep -i -n -e “word1” -e “word2” -e “word3” filename

To search for single word in a multiple files :grep -i -n -e “word1” file1 file2

To search for multiple words in a multiple files :grep -i -n -e “word1” -e “word2” -e “word3” file1 file2 file3

To search for a file in linux machine we can two commands in multiple ways

- 1. Find command:
- 2. Locate command:

**FIND COMMAND : (find path -type ):**

To search for a file using name : find . -name filename

find . -name file	used to find a file in current directory
find /proc/ -name filename	used to find a file in proc directory
find . -type d -name folder	used to find a folder in current directory
find . -type f -name <file1.txt>	used to find a file in current directory
find . -type f -perm 777	Finds all the files whose permissions are 777 in the current directory

<code>find . -type f ! -perm 777</code>	Finds all the files whose permissions are NOT 777 in the current directory
<code>find . -perm /u=r</code>	Finds all Read-Only files in the current directory
<code>find . -perm /a=x</code>	Finds all executables files in the current directory
<code>find . -perm /a=w</code>	Finds all writable files in the current directory
<code>find . -type f -empty</code>	Find all Empty Files in the current directory
<code>find . -type d -empty</code>	Find all Empty directories in the current directory
<code>find / -user &lt;username&gt;</code>	Finds all the files specific user owned in / directory
<code>find / -group groupname</code>	Finds all the files specific group owned in / directory
<code>find . -mtime 10</code>	Finds all the files which are modified 10 days back in current folder
<code>find / -atime 100</code>	Finds all the files which are accessed 10 days back in current folder
<code>find . -cmin -60</code>	Finds all the files which are changed in the last 1 hour in current directory
<code>find . -mmin -60</code>	Finds all the files which are modified in the last 1 hour in current directory
<code>find . -amin -60</code>	Finds all the files which are accessed in the last 1 hour in current directory
<code>find . -size 1k</code>	Finds all 1KB files in current directory
<code>find / -size +50M -size -100M</code>	Finds all the files which are greater than 50MB and less than 100MB in / directory



locate filename	Used to locate a word in linux (by default it will not locate, we need update db every time)
sudo updatedb	used to update linux db
locate -i filename	used to search for a file in case sensitive
locate -n 5 "*.txt"	used to search top 5 text files
locate -c aws*	used to count no of aws files present in server