DEVOPS COMMANDS

What is devops?

- devops engineer works on servers
- Cloud Computing means any business service which is used in network as you pay as you go model.

They are total 3 types in cloud computing:

- 1. AWS
- 2. AZURE
- 3. GCP(Google Cloud Platform)

How many ways to login AWS/Azure/GCP ?

- There are two types:

1. **CONSOLE (UI—USERINTERFACE):**

In console we have 2 ways to login

Root User Login –admin user

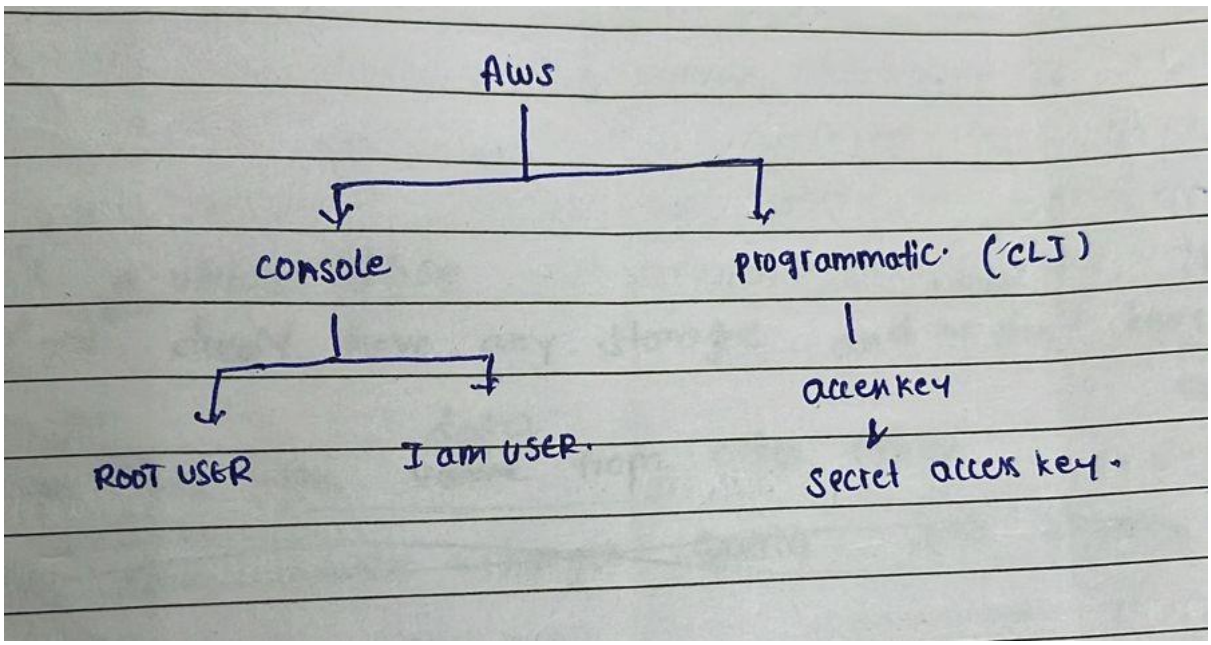
- We will login into root user through email & password

I am user – restricts & limited access will be provided

- We will login into I am user through username and password

2. **PROGRAMMATIC—CLI (COMMAND LINE INTERFACE):**

- If we want to access aws we will access through argumets(in this we will give command and we will access)
- In this access key & security access key (we can establish connection)



How to launch/create the ec2 servers in aws?

1.**Name** :- Identification of server/application

2. **Os** :- distribution of linux

- Ubuntu
- Amazon linux
- Redhat
- Mac-os
- Windows
- Debaian

3. **Instance type** :- size of the server(ram and cpu we can configure)

4. **Keypair** :- security credentials for login into the server. It contains public and private keys

- Data is encrypted (only system can understand)
- .PEM :- to connect from windows to linux through public
- .PPK :- to connect from windows to linux through private

5. **Networking** :- to provide public ip and private ip

Public ip : to access the application over the internet

Private ip : to connect from one server to another server

6. **Volume:** to store the application (or) user's sensitive data

- by default we will get 8gb

After connecting to server we need to connect to linux and run the linux commands

What is linux and why we will use?

- It is a shell (we can execute multiple commands using linux)
- It is an open source
- Fresh implementation of API'S
- Many networking protocols

Protocol:

- Set of rules , which are connected between the user and server

HTTP : HYPER TEXT TRANSFER PROTOCAL

- Process of data between the application and user

TCP : TRANSMISSION CONTROL PROTOCOL

- It connects server through ip

UDP: USER DATAGRAM PROTOCOL

- It is used for live sessions

FTP : FILE TRANSFERRING PROTOCOL

- It is used for file transferring

RTMP : REAL TIME MESSAGING PROTOCOL

- It is used to transfer audio and video in realtime.

ICMP : INTERNET CONTROL MESSAGING PROTOCOL

- It increses the cooldown session

How many users in linux?

They are total 3 users in linux:

1**. SuperUser :-** admin of the server(default user)

2. **SystemUser** :- if we install any software in servers then it is called system user

3. **NormalUser** :- if we want to give access to users for providing specific permissions

Super user, system user, normal user --> default location is --> /etc/passwd

After connecting to server what do and what are things we need to configure

- ~$-->sudo permission-->if we want to execute any command we need to give sudo
- **sudo –i** --> from sudo it is changing to root user
- **pwd** --> present working directory
- **ls** --> list of directories
- / --> home directory

- If we want to read the data inside the file, then we will use **Cat passwd**

How to create normal users?

There are 2 ways to create:

1. **useradd username** --> it will directly create an user

2. adduser username --> it asks for set password, we need to set the password

- If we want to identify whether the password is assigned or not , we will use **cat shadow**
- If we want to assign a new password/set, we have to use **passwd username**

User operations:

Create user	--> **useradd username**
Create user with id	--> **useradd -u id username**
Create user with passwd	--> **adduser username**
Create user with passwd	--> **passwd username**
Reset user passwd	--> **passwd username**
Change user name	--> **usermod -l newname oldname**
Change id	--> **usermod -u newid name**
Lock user	--> **usermod -L username**
Unlock user	--> **usermod -U username**
Check if user has passwd	--> **cat shadow**
Check if user is locked	--> **cat shadow**
Delete user	--> **userdel username**

Groups: multiple users we can add in a single group
Normal user: single users will be created/added
How to create group?

There are 2 ways to create:

1. **groupadd groupname**

2. **addgroup groupname**

Group Operations:

Create group	--> **groupadd name**
Create group with id	--> **groupadd -g id name**
Group name change	--> **groupmod -n newname oldname**
Group id change	--> **groupmod -g newID name**
Group delete	--> **groupdel groupname**

To add users to groups, we have some commands(gpasswd)

adding single user	--> **gpasswd –a username groupname**
adding multiple users	--> **gpasswd –M username groupname**
adding admin among the users	--> **gpasswd –A username groupname**
to delete a user from group	--> **gpasswd –d username groupname**

If you want to check whether the admin is created, you have to check in **gshadow**

SYSTEM COMMANDS:

to check our system os	**--> cat/etc/os-release**
to print hostname private	**--> hostname –i**
to change hostname	**--> hostnamectl set-hostname sunil**
init 6	**--> to restart the server**
to print public ip	**--> curl ifconfig.io**
ram related information	**--> free –h(human understandle language)**
to identify disc space	--> **df –h**
we can monitor entire server	--> **htop**
cpu full information	--> **cat/proc/cpuinfo**
memory information	--> **cat/proc/meminfo**
system procedure	--> **ps**
kill process	**--> kill process id**
to check work status of entire system	--> **ps aux**
to find active users	--> **w**
directory size	--> **du –sh <dir>**
to check open files	**--> lsof**
to find details of all files in current directory	--> **ls -lrth**

How to create a file?

They are 2 types to create a file :

- **1. Empty file(touch)**
- **2. Non-empty file(cat)**

Empty file:

We can create an empty file using touch command

creating file	--> **touch filename**
creating multiple files	--> **touch username{1..100}**
removing empty file	--> **rm filename**
removing multiple file	--> **rm username{1..100}**

Non-Empty file :

To create non-empty file we have 2 ways

1. **Using vi (structured format)**
2. **Using cat (unstructured format)**

2 . **using cat command:**

- **Cat > f1** ---> it will create a file and we have to insert a data
- **Ctrl+d** ---> to save the data which we have entered
- **Ctrl+z** ---> if we don't want to save till that particular line use this
- **Cat >>f1** ---> to add data to an existing file we will use this >>
- **Cat > f1** ----> whenever we want to create a new file then the data will be deleted with >
- **Cat f1 >>f2** ---> to append data/copy one file data into another file

1 . using vi (visual editor):

There are total 3 types of modes

1. **Command mode**
2. **Insert mode**
3. **Extend command mode**

1. Command mode :

- Command mode is a default mode when we create a file using vi we enter into command mode by default.
- In this mode we can do multiple operations.

Commands in command mode:

gg	-- Cursor moves to start of the file.
n gg	-- Cursor moves to nth line of file.
G	-- Cursor moves to end of the file.
x	-- Deletes a letter.
dw	-- Delete a word.
dd	-- Delete a line.
n dd	-- Delete n lines after cursor.
yy	-- Copy a line.
p	-- Paste.
/	-- For searching any letter or word.
n	-- Next of search in forward direction.
N	-- Next of search in backward direction

2. Insert mode :

- Insert mode is used to insert data into the file.
- In insert mode ,we can edit ,add and remove
- From command mode, if we want to enter into insert mode, we need to give letter '**I'**

3. Extend command mode :

- With the help of extend command mode we can save and quit
- To enter into extend command mode, we have to use **esc shift + :**

w	--> to save the data.
q	--> to quit.
q!	--> Forcefull quit without saving.
wq	--> Save and quit.
wq!	--> Forcefull save and quit.
x	--> Forcefull save and quit.
se nu	--> Shows the numbers for lines.

File and Folder Operations:

mkdir filename	--> creates an empty folder/directory.
mkdir -p folderstruct	--> creates a nested folder/directory.
ls -R foldername	--> shows the tree structure of a folder/directory.
cp source destination	--> copy file from source to destination with or without rename
cp -r source destination	--> copy one folder to another folder.
cp -rf source destination	--> forcefull copy of source folder to destination folder.
mv source destination	--> move folder or file from source to destination.
mv oldname newname	--> renames old name of folder to new name of folder.
rmdir folder	--> removes an empty folder.
rm -rf	--> removes a non-empty folder.
rm -r	--> removes a non-empty folder.

Filtering Commands:

These commands are used to filter and get the data according to our requirements.

GREP: (Global Regular expression print):

grep letter/word file	--> search and filter the given expression from required file.
grep -c	--> gives the count of lines in which the given expression exist.
grep -i	--> search and filter the given expression from required file without case-sensitive.
grep -e	--> helps to search and filter multiple expressions.
grep -v	--> gives the entire file by removing the given expression.
grep -n	--> I want a particular letter /word to find out in which line it is
grep -o	--> it will expose only what is the data required

grep -h --> search and filter the given expression from required file with case- sensitive.

grep -l --> whether the given data is present in directory or not

Sort command:

sort filename --> sort in ascending order only the starting number.

sort -n --> sort ascending order in the form of digits.

sort -r --> sort in reverse order only the starting number.

sort -nr --> sort in reverse order in the form of digits.

Sed command:

- I wanted to change data/modify into lower to upper (or) replace data
- sed s(sub)/e(data which we want to replace)/E(data which we want to add)/g(globally)

Cut command:

helps to print required columns.

cut coln file

cut -b 1 f1 --> Prints 1st column of f1.

cut –b 1,3 f1 --> Prints 1st,3rd column of f1.

cut –b 1-3 f1 --> Prints 1st to 3rd column of f1.

Awk:

Helps to print nth word of every row present in the file.

awk '{print$n}' f1

'**$n**'-represents nth word.

Paste command:

- if we want to combine one or more file we will use paste command

paste file1 file2

More:

- more filename
- it will fetch the data according to the stages
- when it reaches the end of file, it quits automatically

Less:

- It doesn't show the percentage of the file. When it reaches the end of file it doesn't quits automatically, we have to use 'q' letter to quit.
- less filename

Hidden files:

Hidden files starts with ' **.**'

ls -a -- To check hidden files.

ll -- Gives the fill info of the files and folders present in present directory.

ls -al -- Same as ll.

la -- To check hidden files.

Networking commands:

Ping --> server is getting internet internet or not

hostname –i --> to find the private ip

ping privateip --> we will get speed of internet,ip,icmp,time

ping –c n privateip --> i wanted to count the details at the required number

ip r --> it will find which are the routes connected to the server

traceroute url --> it will fetch full information and who are accessing and from where they are accessing

dig url --> servers which are connected to our domain

nslookup url --> it works same as dig command

--> it will fetch address and name

Ifconfig --> it will fetch internal system information

Inet --> internal ip

Curl url --> it will work for ip's well compared to domain

Listening ports:

Netstat –tnulp --> to check inside the servers how many ports are listening
telnet ifconfig.io --> to verify whether the ports are connected in server or not, if the ports are not connected, we have to add in securitygroups
journalctl –xe --> application errors, logs, persons, servers(information of all)
journalctl –xe || grep failed --> to fetch the failed operations
wget url --> it will download software (or)folder in provided link

Permissions of files and directories:

- we have total 3 users, we need to find which user is assigning for the file
- we have to check whether we can customly add (or) not

After connecting into server:

- create an empty file using **touch filename**
- after creating we need to find out file permissions using **ls –l**
- **-rw-r--r—1**
- **r >>indicates read>>we can only read**
- **w>>indicates write>>we can edit,update,write**
- **x>>indicates execute>>we can remove and we can run scripts**

we will have 3 users :

- **userlevel** (first 3 indicates)
- **grouplevel** (next 3 indicates)
- **others level** (next 3 indicates)

commands :

touch f1 --> empty file

ls –l --> to check the fie permissions

-rw-r—r-- --> it will display this for empty file

Now we will check for non-empty file:

Cat >f2 --> we need to insert some data in it

ls –l --> to check the fie permissions

-rw-r—r-- --> for both empty and non-empty file permissions are same

Now I want to create empty directory and check permissions:

mkdir d1 --> it creates a empty directory

ls –l --> to check the fie permissions

drwx r-x r-x --> it will display what are permissions for empty directory

Now I want to create non empty directory and check permisssions:

mkdir –p d2/d3 --> it creates an nested non-empty directory

ls –l --> to check the permissions

drwx r-x r-x --> for both empty directory and non-empty directory permisssions are same

How to give custom permission for a file (or) folder:

- **chmod --> to change/modify file permissions**

commands:

chmod u+x f1(u indicates userlevel) --> to add permissions to userlevel

chmod u-rwx f1 --> to remove permissions to userlevel

Like wise if you want to add for group level you need to indicate with g and if you want to add to other level indicate it with o

chmod u-rwx,g+rwx f1 **-->** to add multiple permissions to different levels, first one we are removing and second one we are adding

chmod u+rwx,g-rx f1 **--> adding multiple permissions to difdferent levels, this time first one adding and second one removing**

Like wise if you want to add all three groups you can add by denoting ‘,’ and if you want to remove all at a time you can give

Example:

Chmod u-rwx,g-wxr,o-wrx f1 ---> every permission will be deleted

To denote it in numerically instead of alphabetical :

- **r ---> 4**
- **w --->2**
- **o --->1**

Now if you want to add permissions:

Chmod 712 f1 **--->** like when we are giving in numerically we have to add and give the number according to each level like userlevel, grouplevel, otherlevel

Example:

rwx - -x –w-

chmod 000 f1 ---> if we want to give null permissions for every levels we can denote like this

How to assign the owners to the file:

- first we have to create a user using **useradd sunil**
- before assigning owner the original owner to the file will be **root**
- now I am assigning user sunil as file owner
- to assign we have a command
- **chown –R sunil f1 ---> chown is used for giving ownership to the user, and now in this command we are assigning sunil as owner to the file f1**
- sunil will be added to the user as owner
- now if you want to make sunil as group level owner, we will use
- **chown –R sunil:sunil f1**
- **hence this is the process for adding owner to the user in the file**