# Devops

Devops : Chef, Ansible, Docker, Kubernetes, puppet(older) : all are open source.

AWS Devops : Code commit, Code star, code deploy, code pipeline

Azure Devops : Azure repos, Artifacts

we can creates 100 of server with one configuration.

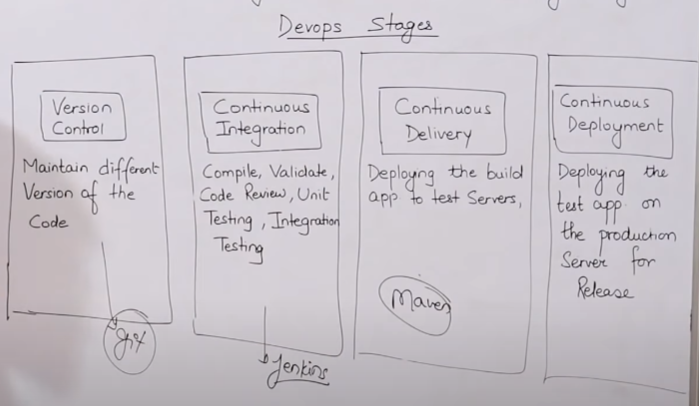
# Devops stages

Version control

Continous integration

Continuous delivery

Continuous depolyment : Chef, Ansible, Docker, Kubernetes



# Networking concepts

Router, Switch, Gateway

Cloud provide three service : Iaas, Paas, Saas

# Linux

## Sudo su

become admin

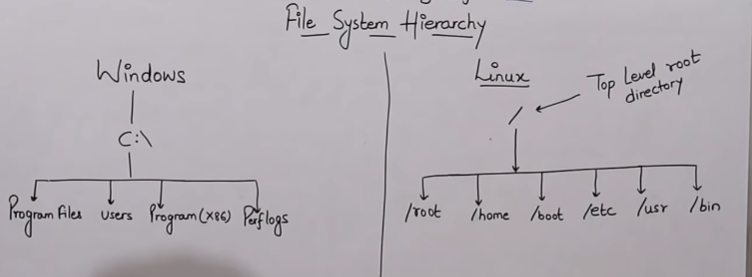


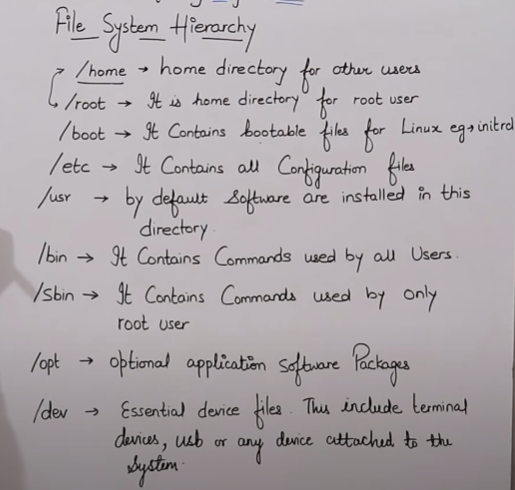
Linux is kernal

* Rhel(Red hat 🡪 free + paid)
* Fedora
* Debian
* Others
* Ubuntu(3rd most popular os), centOS(community os, fastest), amazon linux
* kali linux(famous for hacking)

cmd : yum updata chef

## File system hierarchy

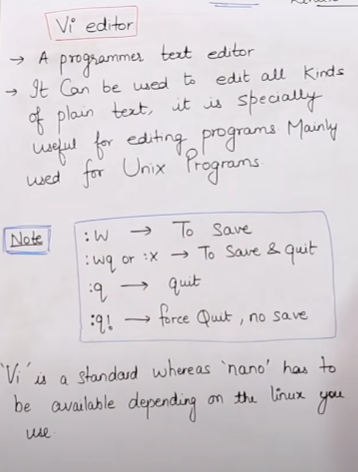




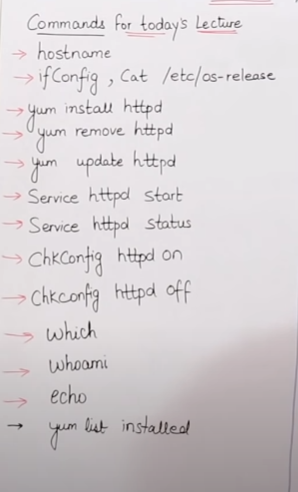
## Create file

Vi is standard and available in all linux.

* Cat
* create file > file1 🡪 write 🡪 ctrl + d(save)
* cat > file : create new file
* cat file : view file
* cat >> file : append data in next line not edit old
* Copy file
* Cat file1 file2 > file3 : new file with name file3
* tac: to see the content of bottom to top
* Can not edit with this
* Concatenate
* Touch : empty file( create 3 time stamp access time, modify, change time)
* touch file : if exist, update all time
* touch -a file : update only access time
* stat file : all access time
* Vi/vim : programer editor
* :w save
* :wq or :x save or quit
* :q quit
* :q! force quit
* Don’t use for view
* H J K L : navigation left right up down
* nano : editor
* create file > file1 🡪 write 🡪 ctrl + x(save)
* Don’t use for view



## Yum and httpd command:



2. cat /etc/os-release : which version of linux is used

3. hostname -I : ip of server

4. nohup httpd & : automatically run in background

5. which chef : whether chef is installed or not

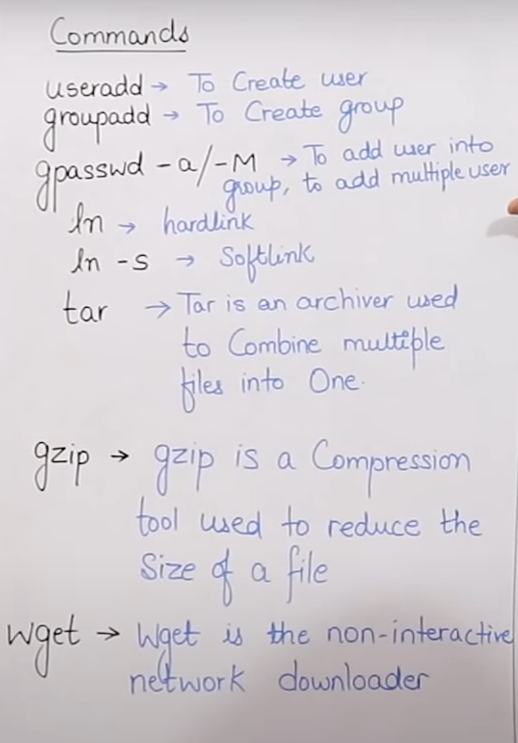
6. whoami : root or name of user

7. grep root in etc/passwd : search root in in passwd folder

8. less, more, tail, head

9. rmdir :empty dir and rm -rf: remove non-empty file/dir, -r : remove empty dir

## User add and group add



### Add and check user

Useradd rikki : add rikki user

cat /etc/passwd/ : check user(automatically create group as well)

### Add and check group

groupadd tech : added group tech

Cat /etc/group/ : check group tech

### Add user to group tech

gpasswd -a vikki, rakesh tech : it will vikki, rakesh in group

### ln : create soft and hard shortcut link

ln -s file1 soft1 : create soft link with name soft1 of file1

ln file1 hard1 : create backup also replicate the data

### tar

tar -cvf dix.tar dix

gzip dix.tar

### untar and unzip:

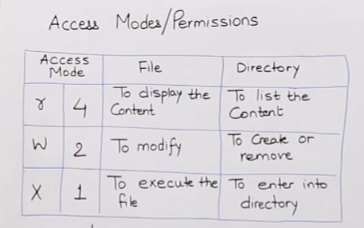
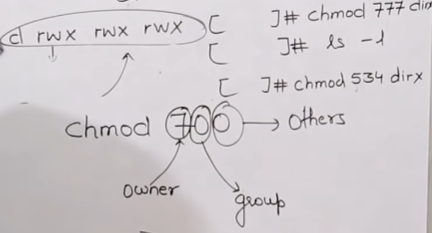
gunzip dix.tar.zip

tar -xvf dix.tar

### Access type

* chmod
* chown
* chgrp

rwx – 4 2 1

Chmod 77 dir

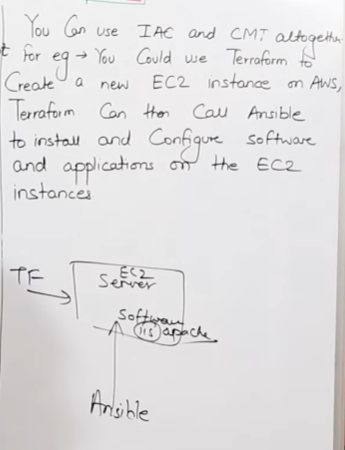
Chown rikki file1 : change owner

Chgrp tech file1 : change group

# CMT(Configuration management tools)

Ansible, chef, puppet are used for install and manage software on existing server, not for installing the server.

Once terraform is created the server then it call api to ansible for the installing the software



# Chef((IaC tool)

* It is administrative job. Like it update and manage all of your server. It replaced system administrator job.
* It is not used to install server.
* It is used to install software on server.
* No need to do any thing mannully.

Dev + ops(configuration management tool)

## CMT type

* Push based(Ansible, salt stack ) : : os get update from server to update the os or app

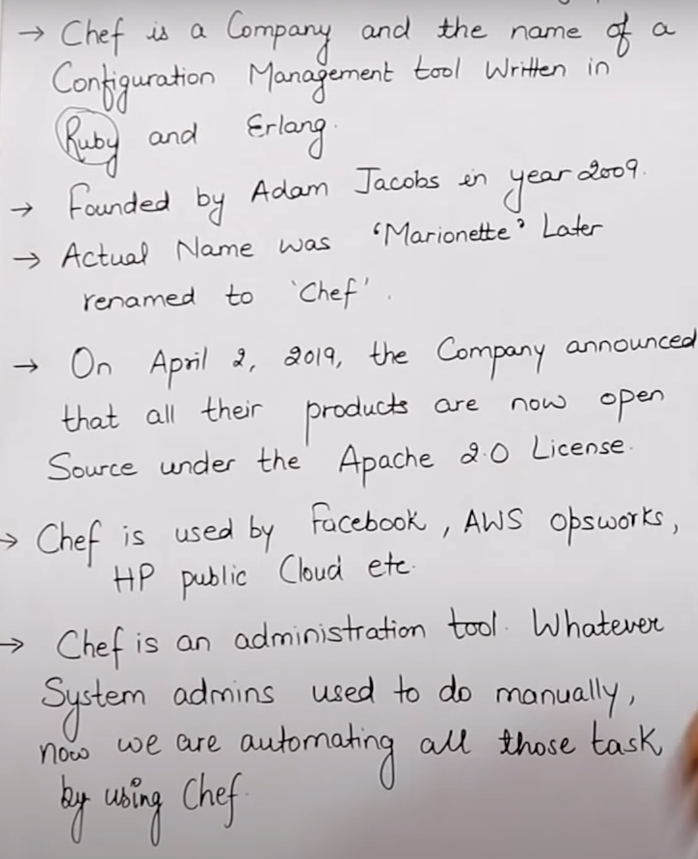
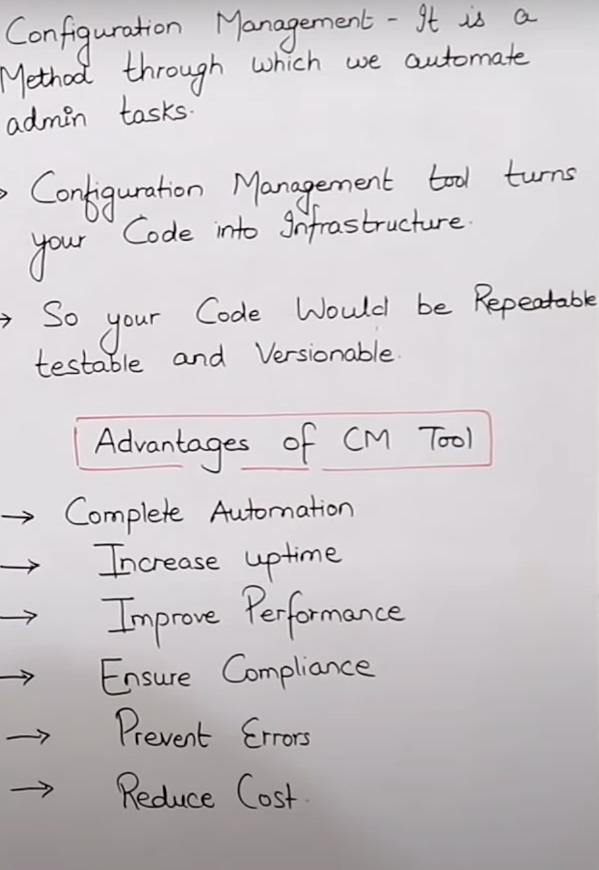
1. eg like update to whatsapp updated version request
2. configuration is easy
3. server control is admin hand

* Pull based(CHEF, Puppet) : os check from server on regular basis for update

1. Adding new machine are easy
2. Once new machine are added, Automatically periodically check from server for update.
3. Suppose one machine is down, once it is update it automatically check for the update

Infrastructure as code(IFC)

Chef supermarket provides ready made code. No need to write, all code is already avaiable.

## Chef architecture and process:

Work station, chef server, nodes(server)

Chef is installed at nodes and workstation.

## Workstation

* Personal computer or virtual computer where all configuration code is created , tested or changed
* Devops engineer works here on recope
* Workstation work with knife(cmd tool) .

### Recipe: IFC

* Recipe-1 : window config code
* Recipe-2 : unix config code
* --- many more
* Write in Ruby

### Cookbook

Cookbook contains or collections of all recipes

## Chef server or server

* It used to store cookbook
* Middle men between node and workstation.
* All cookbooks are stored here
* Server may be hosted locally or remote

### Knife(CLI tool)

Cookbook transferred from work station to server by knife(CLI tool)

### Bootstrap(Connection)

Also transfer chef server to nodes for connection(bootstrap)

## Node

It has two part:

Each node can have different configuration

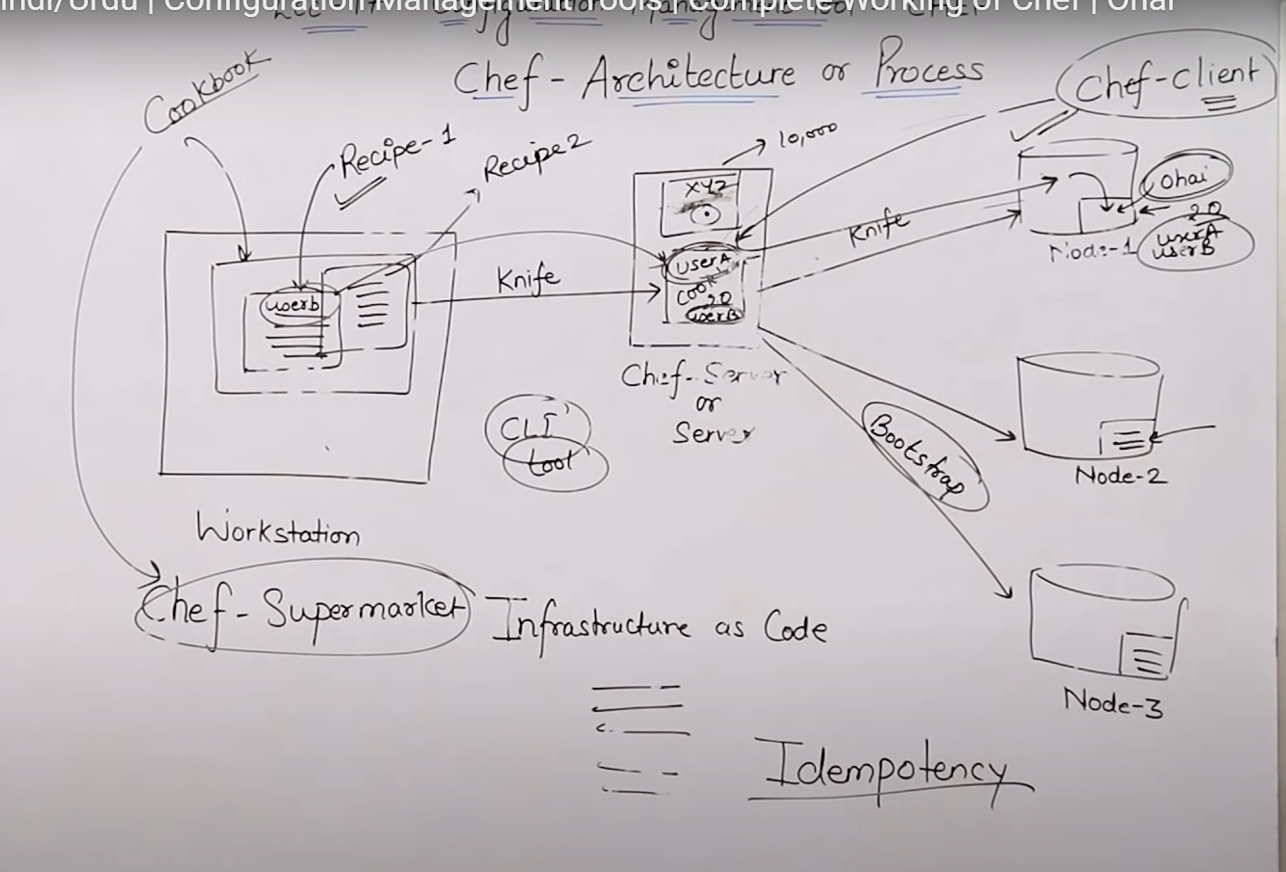
### Ohai

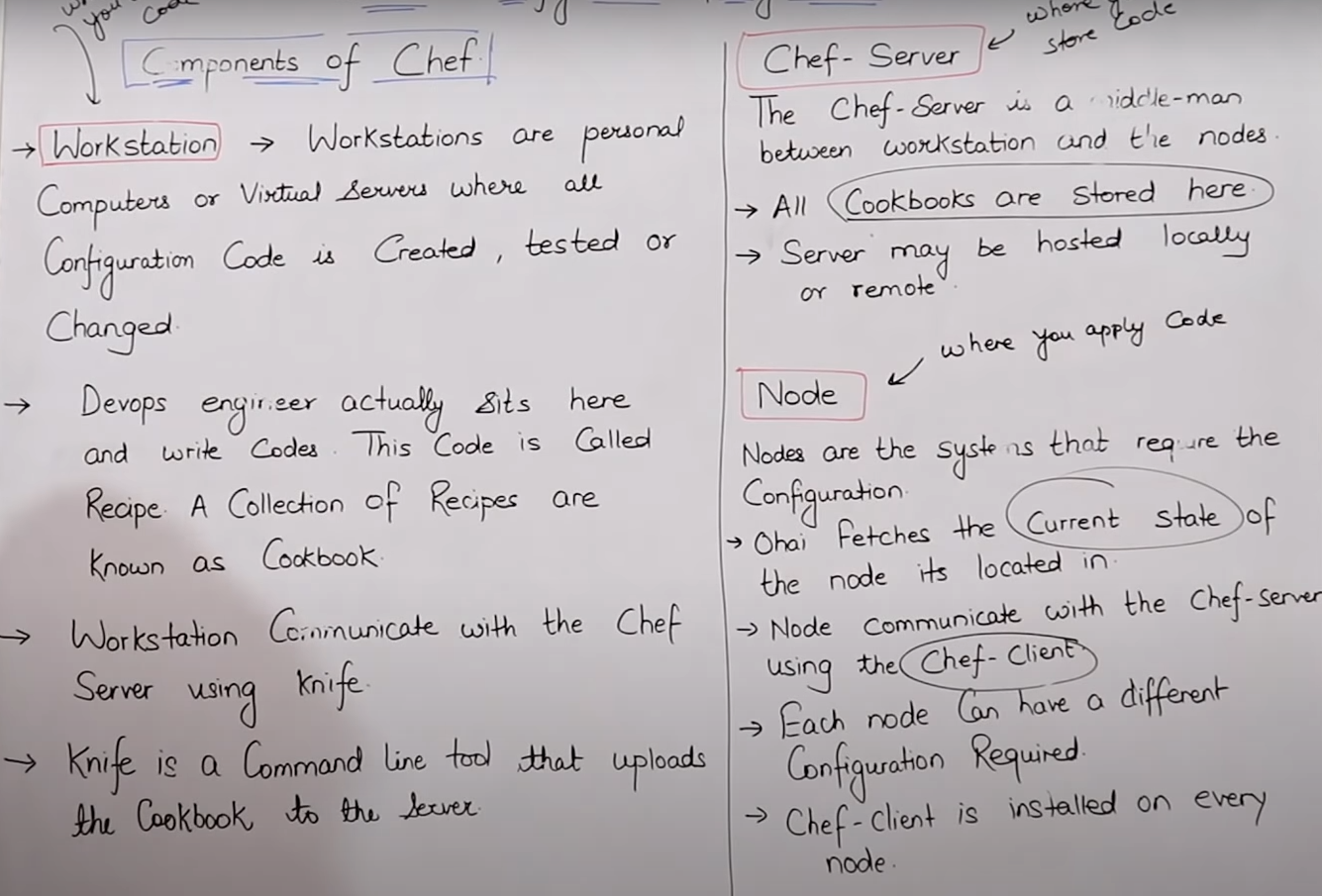
It is kind of database of node like user, password, IP and all configuration of machine is stored inside this.

### Chief-client(part of node)

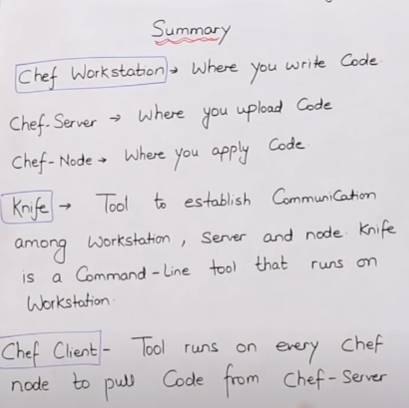
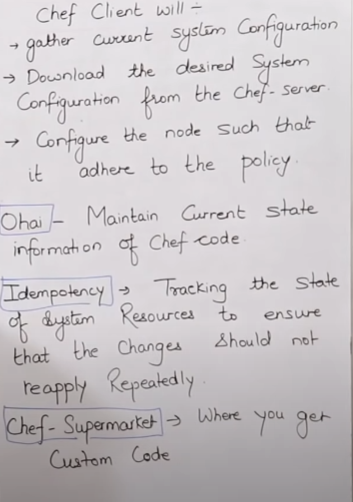
* It bring latest code from server
* Chief always check in the ohai database for the config then it check with chef server or server
* If there is latest changes in chef server then chef client will update in the node and ohai both.
* Chef client check periodically. It will bring only new changes(Idempotency)
* Chef client is installed on every node.

## Architecture Diagram





## Summary

# IaC tools type

### Imperative

* Specific commands or config to achieve desired configuration and must be in order
* Aws eg: VPC, subnet, IgW, Routing table :- all point should be in sequence
* cloudformation

### Decalerative

* No need to mention about sequence
* Terraform is declarative approach

# Terraform((IaC tool):

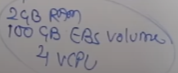
* It is kind of allu, it can fit for any cloud or servers(aws, gcp, azure, openstack, Vmware)
* It is used to create infrasturcture like server not for install software like office, or team.
* It makes your planet just like earth.
* Develped by hashicorp
* Configuration files are written in hashicorp configuration language(human readable)
* Support cloud services, on-premises resources, and third-party services
* Allows version control of the infrasturcture
* Terraform initial support only two cloud form : AWS, Digital ocean
* 2017 microsoft tie-up with terraform
* It is decalerative approach
* Easy to integrate with CMT
* It is easily extensible with plugins
* Terraform keep tracks of your infrastructure in a state file.

## IaC tools

It is used to manage and provision of infrastucture through code.

Manage : create, update , delete server

Provision : allocation of resource



## Automation tool for instructure

All these are vendors specific:

### Cloud based

* Cloudformation : aws infrastructure(ops only) 🡪 create manage update infra like vpc, subnetting, ec2.
* ARM template
* Gcp deployment manager
* Open stack : heat

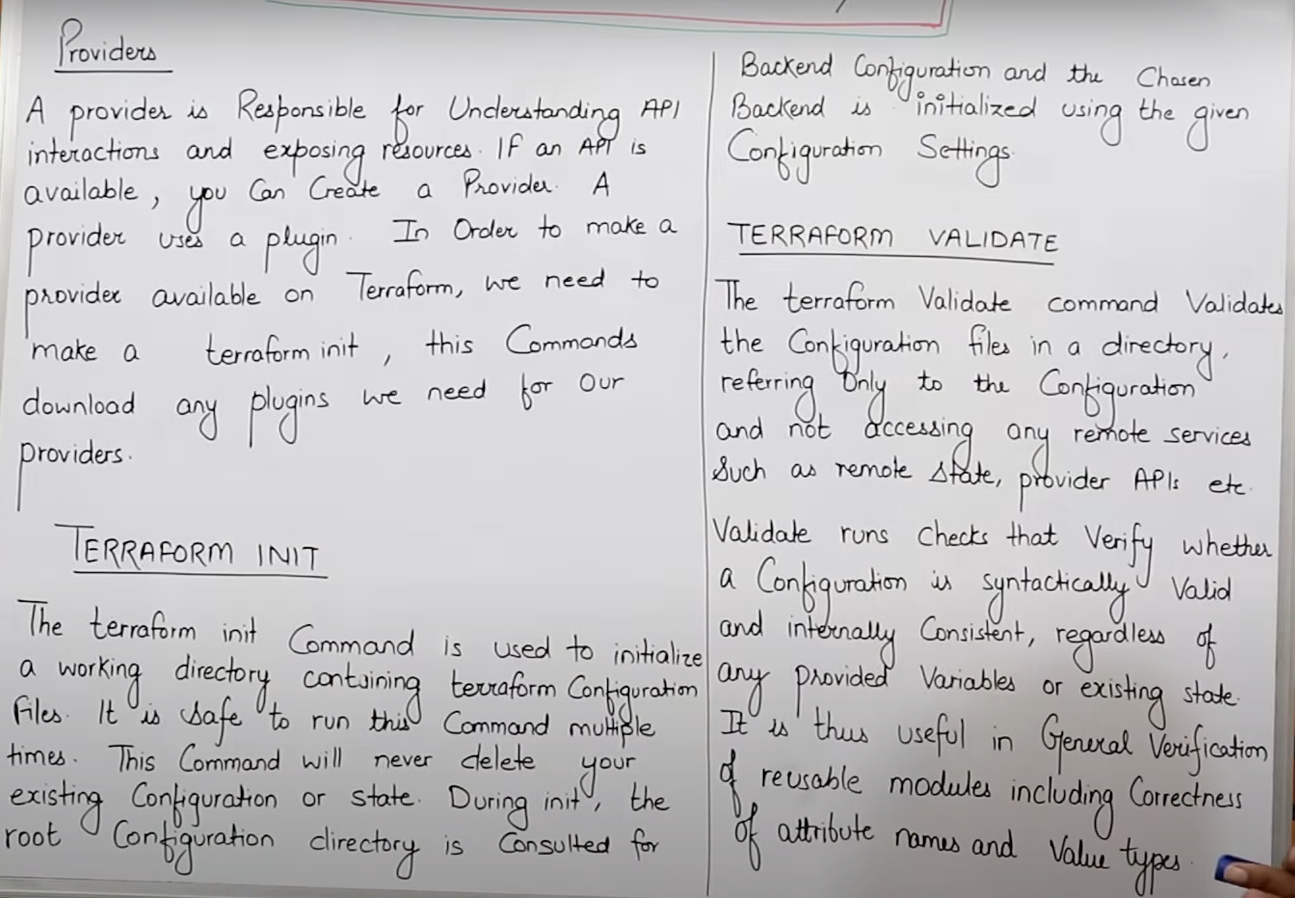
### Open source

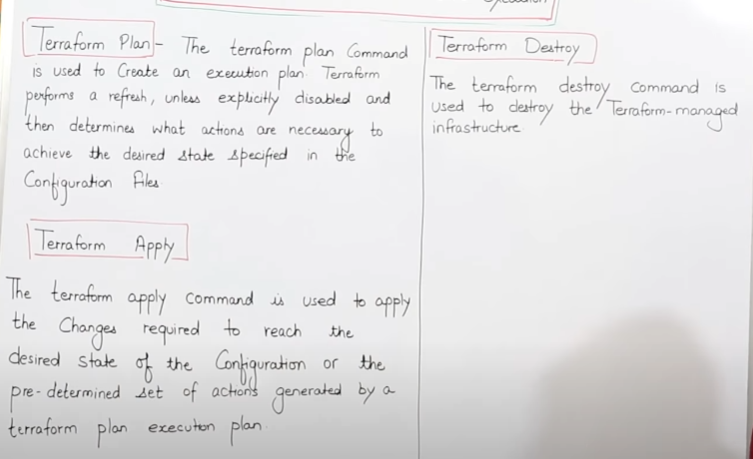
Terraform, Ansible(not pure IaC, but we can make it as IaC)

## Benefits

* It Is not vendor specific.
* It is useful in the case of multicloud and hybird(like a company using three cloud aws, gcp, azure, openstack, Vmware)
* It’s language, HCL, look like the JSON

## Terraform basic commands and HCL Syntax





### Terraform init

* It will take all required plugin
* $ cd .terraform/providers/registry.terraform.io/hashicorp/aws/4.48.0/windows\_386/terraform-provider-aws\_v4.48.0\_x5.exe

### Terraform validate

* Verify syntax

### Terraform plan

* It will show what can be achieved by showing + sign.

### Terraform apply

* It will create all points planned in the plan

### Terraform destory

* Can specify what we want to destroy

Linux dryrun?

