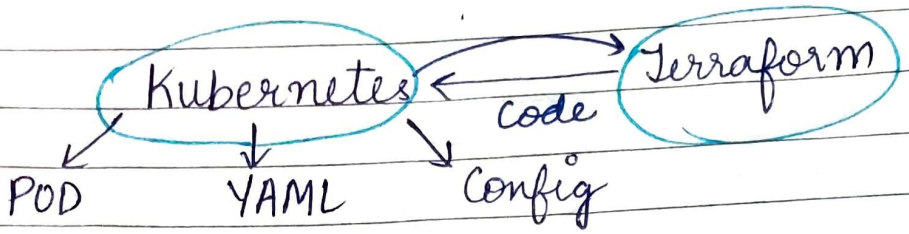
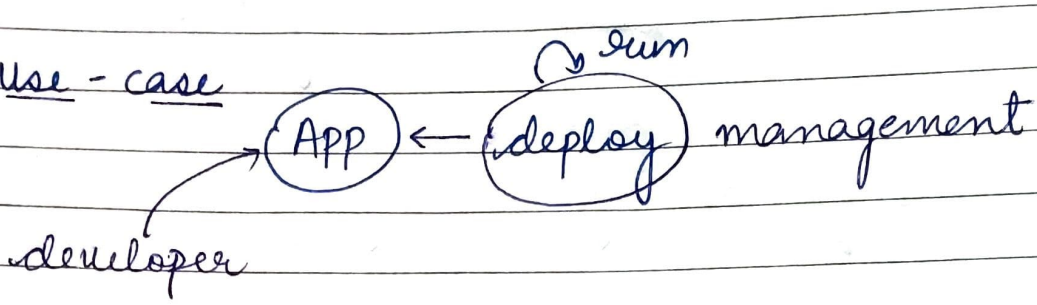
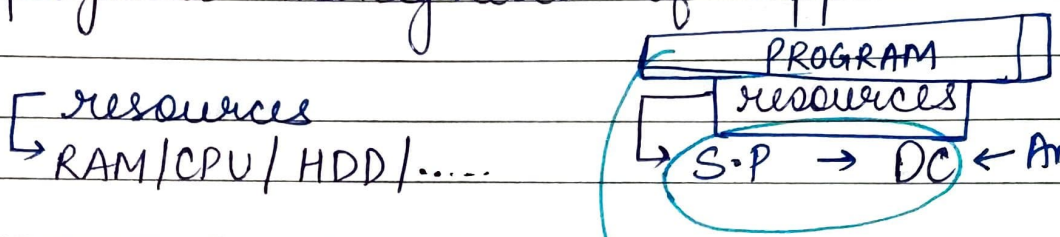
DAY-12

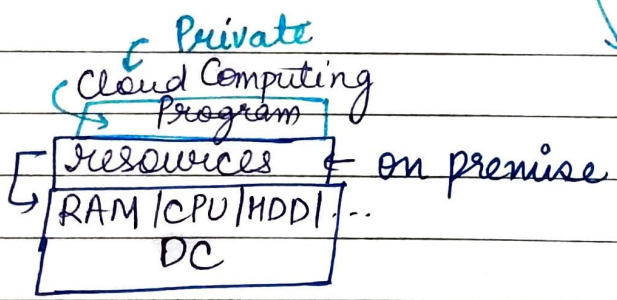


- Use-case



→ We need our app deployed and after deployment managment of app.





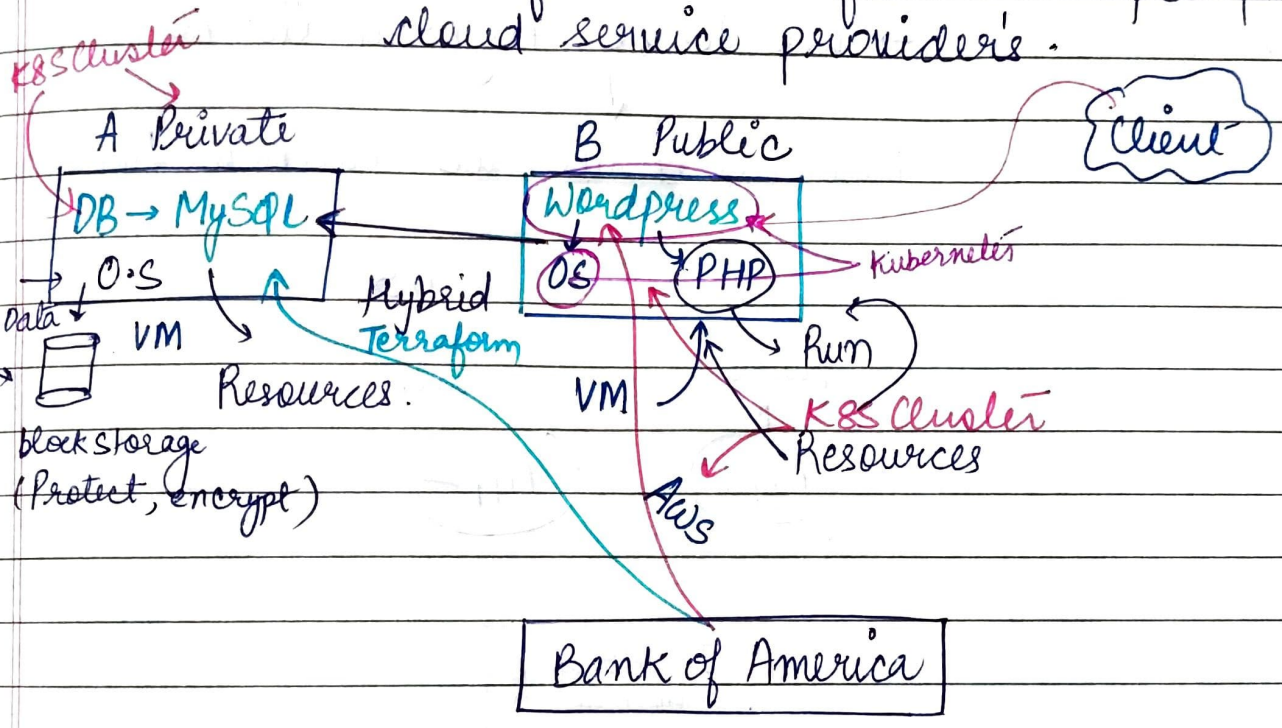
Public
Cloud Computing

Private → Open Nebula, Openstack, Cloud stack, etc
Clouds

- Public Cloud Provides whole Infrastructure as a service

A company on a project may want some services from GCP (Compute), some from AWS (Storage), and some from Azure (Networking).

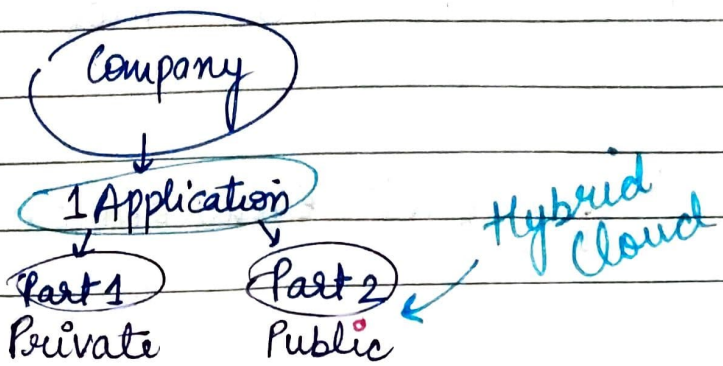
- Multi Cloud - Use of services from multiple public cloud service providers.



→ You have least visibility (least control) over something you have stored on public cloud as you don't have any idea about where your data is stored (the rack).

↓

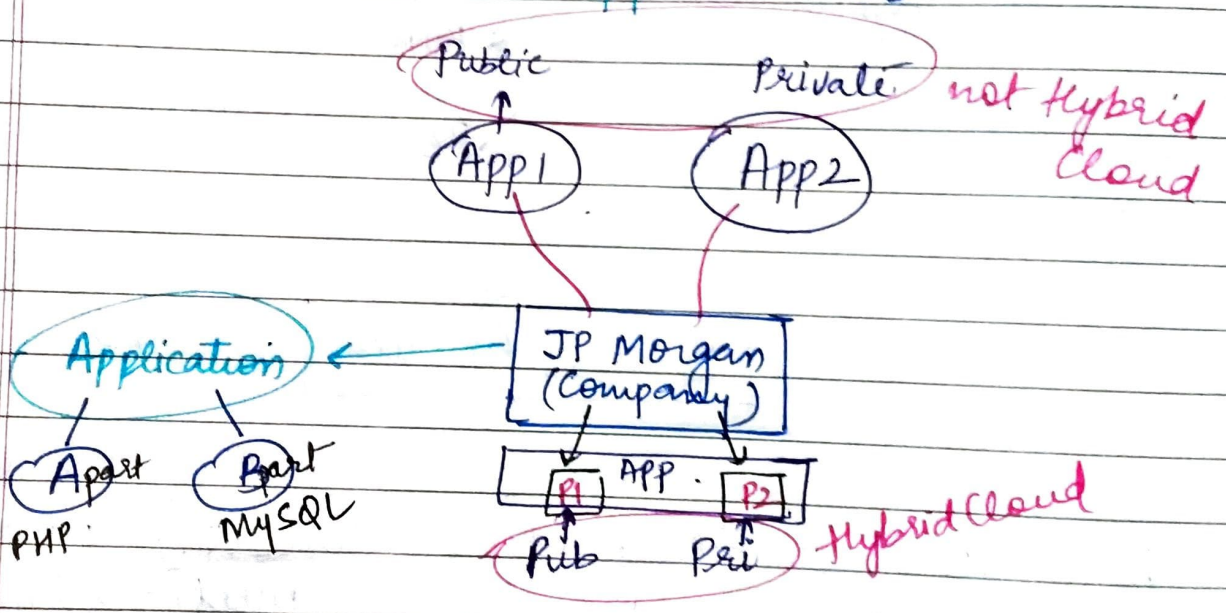
For more control, we use private cloud.



Scenario :-

▷ Suppose a company is running one application, some part of the application is running on Private and some part public. Then it makes up a Hybrid Cloud.

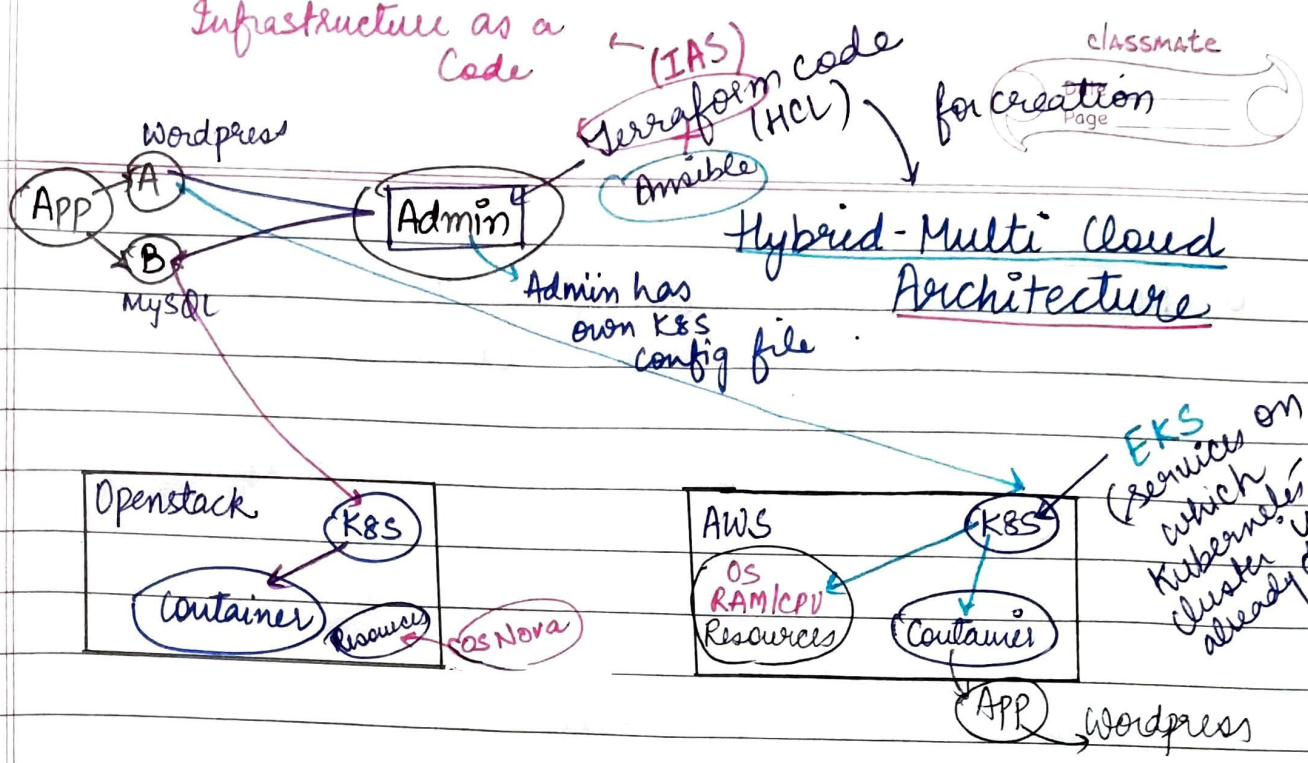
▷ Challenge → we have Hybrid Cloud
↓
[But we have to launch Application
↳ Launch App over OS.



○ Kubernetes
→ Great tool for management.

Containers, are used most of the time.
↓
Docker
↳ for fast environment setup

Infrastructure as a Code ← (IAS)

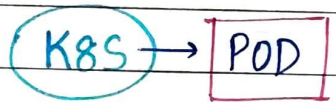


- All use case I want containers to be used.
- Application over Containers

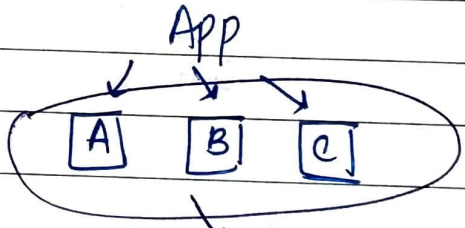
Integration for Terraform and Kubernetes

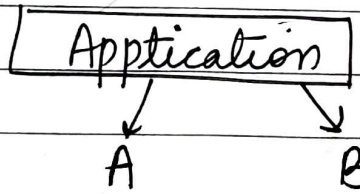
↓

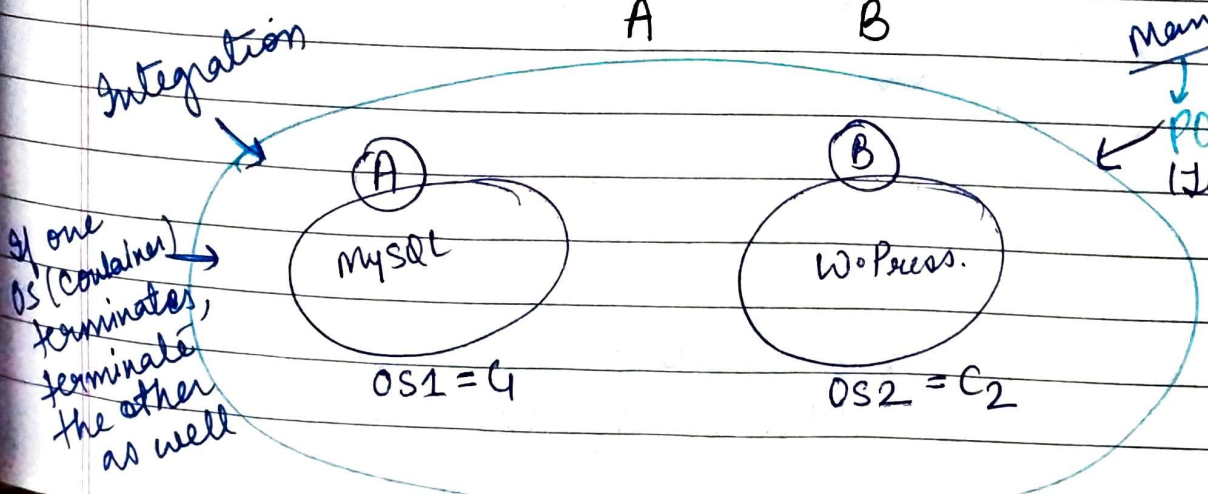
For creation of Kubernetes cluster over AWS.



POD ≠ Container







▷ PODS

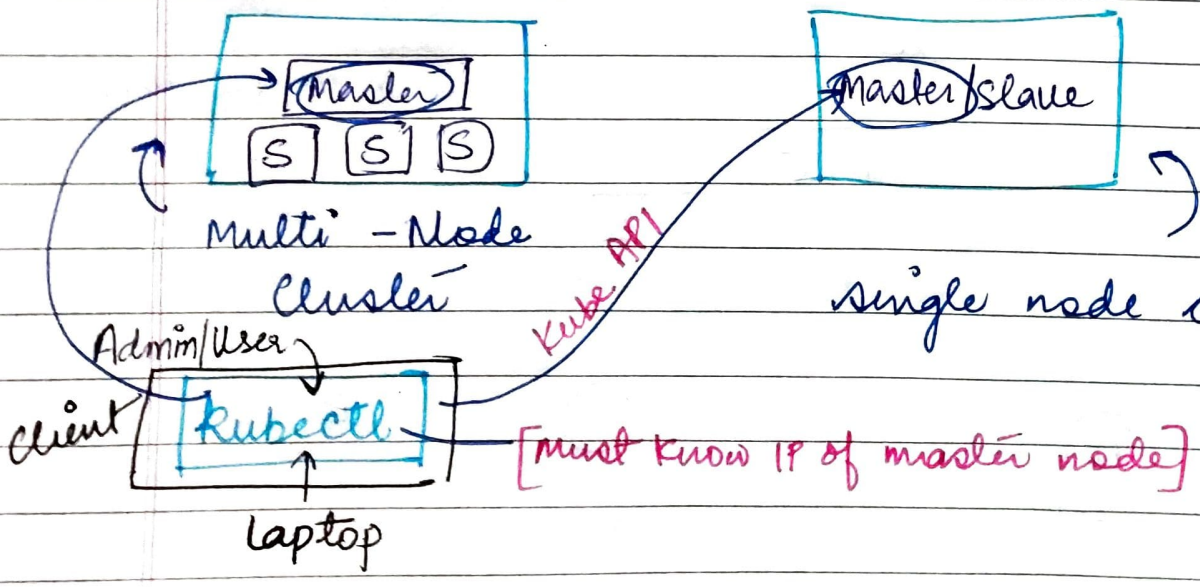
- Main power of Kubernetes is POD'S.
  Inside pods we have containers.
  - ↓ Single Container PODS
  - ↘ Multi Container PODS

```
> minikube start
> Kubectl delete all --all
> Kubectl get pods.
> Kubectl create deployment myweb --image=
                    Vimal13/apache-webserver-php.
> Kubectl get pods
> Kubectl run -h
> kubectl run help.
```

(to launch pods)

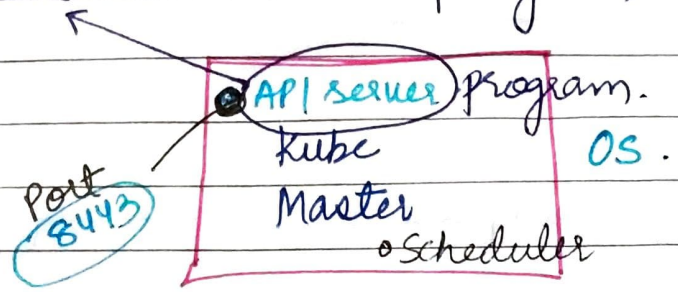
→ (Pod) (container engine (docker)) → to launch container

(contact)



→ Kubectl command always connect to the 'Master Node' (IP of master Node)

listens to kubectl ~~programs~~ command. (Client Interaction)



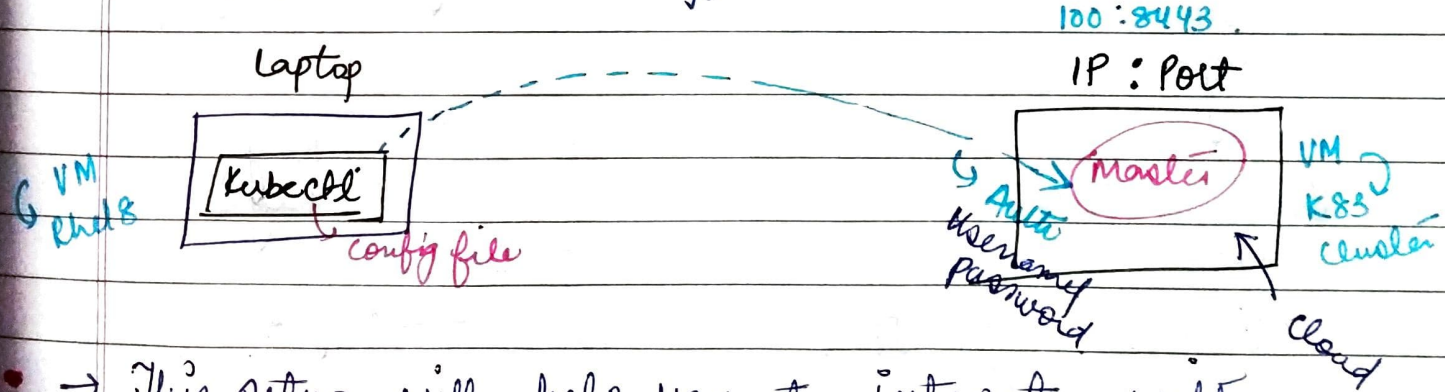
- Kubectl must know IP of Master Nodes.
- Port No. of API must also be known

> minikube ip.

command line → cd .kube (1)

→ notepad config (2)

→ When one program connects to another they use http protocol (mostly)



→ This setup will help you to integrate with Terraform.

Redhat [Rhel 8] command line

> kubernetes version

when this command runs for first time → config incomplete, don't have config file

> kubectl get pods --server https://192.168.99.100:8443

--key my.key

We want program for authentication

Key]

[Minikube provides key]

any file in windows → transfer file to VM. → winscp → Transfer files to VM. (Just need VM IP) (Rhel8 IP)

config folder

C → user → minikube (unhide if not visible)

→ config

→ client.key (private key)

→ client (security certificate)

CA → https.

firefox

webserver ← key

→ Connection VIA HTTPS → Certificate

↑ issued by certificate authority

- When a program wants to connect to https. We need certificate.

[Certificate Authority]

kubectl → AWS CLI

info → JSON / YAML

AWS Cloud Computing

AWS → CLI / Code → JSON

K8S → CLI / Code → YAML

Kubectl

This clusters have cluster information.

we dont write 1 2 3

clusters:

1. Cluster: (-cluster)

file:
- Server :
- client-key : —
- client-certificate : —
- certificate-authority : —

→ One single kubectl should have capability to connect to any master cluster

→ I can tell my kubernetes, I may have multiple cluster

↓

one cluster info I have.

2. Cluster :
server : —
certificate-authority : —

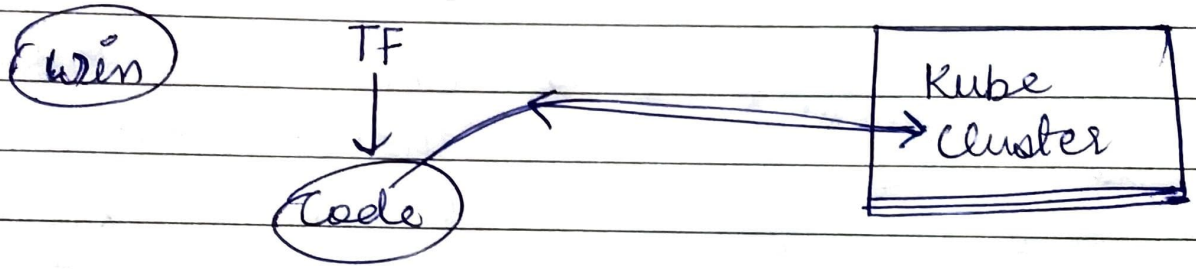
YAML syntax → vim file.yaml

syntax:

apiVersion : v1
kind : config
clusters :
- cluster :
server :
certificate-authority :
name : mylocal cloud

contexts :
- context :
cluster 1
user 1

> mkdir .kube
> .kube
> notepad config.yml
> kubectl get pods.
> Kubectl delete pods all.
> cd..
> Desktop
> cd terraform
> mkdir Kubepod
> cd Kubepod
> notepad pod.tf



pod.tf

context : minikube
provider: "kubernetes"