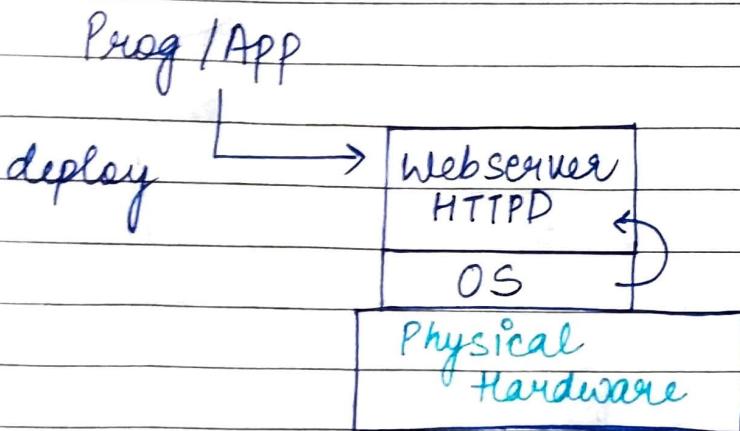


DAY-11

Kubernetes (K8S)

Pre-requisite :> Docker



For launching any environment (OS with server) to deploy Web App / Program, we need containers (Docker)

↓
Provision / destroy OS environment in one second.

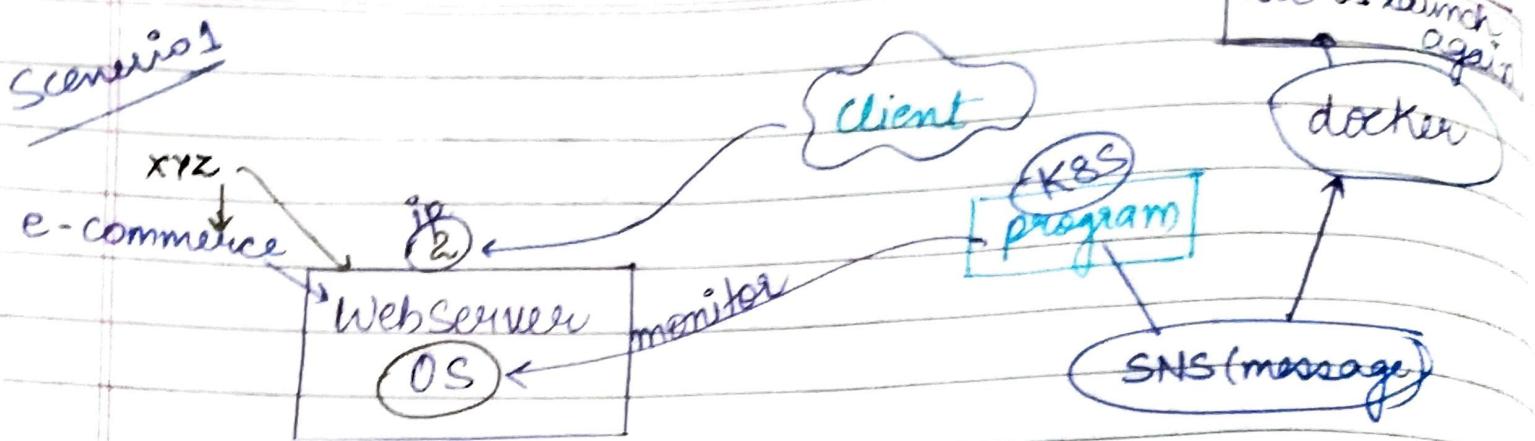
systemctl status docker

We need image, to launch OS in Docker :-

docker run -dit --name myOS1
httpd:latest

docker ps .

curl http://172.13.01.12 (Docker OS IP)



If OS goes down, client goes to fetch your site, he won't find

docker rm -f myOS1 (removes OS)
forcefully

In this world we have program to keep watch on OS, program, service, etc. If anything goes wrong these program send notification to the manager of OS/service program (notification (SNS) - email generally)

Again this manager will ask the docker to launch the same OS again

Hence entire environment comes up within second

But challenge is that it's all manual, and this agile world we need automation

~~manual~~

- Due to automation, client rarely sees a downtime.

Architecture ← fault tolerance ?

[OS environment]



- We need a program for monitoring the environment and as soon as environment goes down relaunch it .

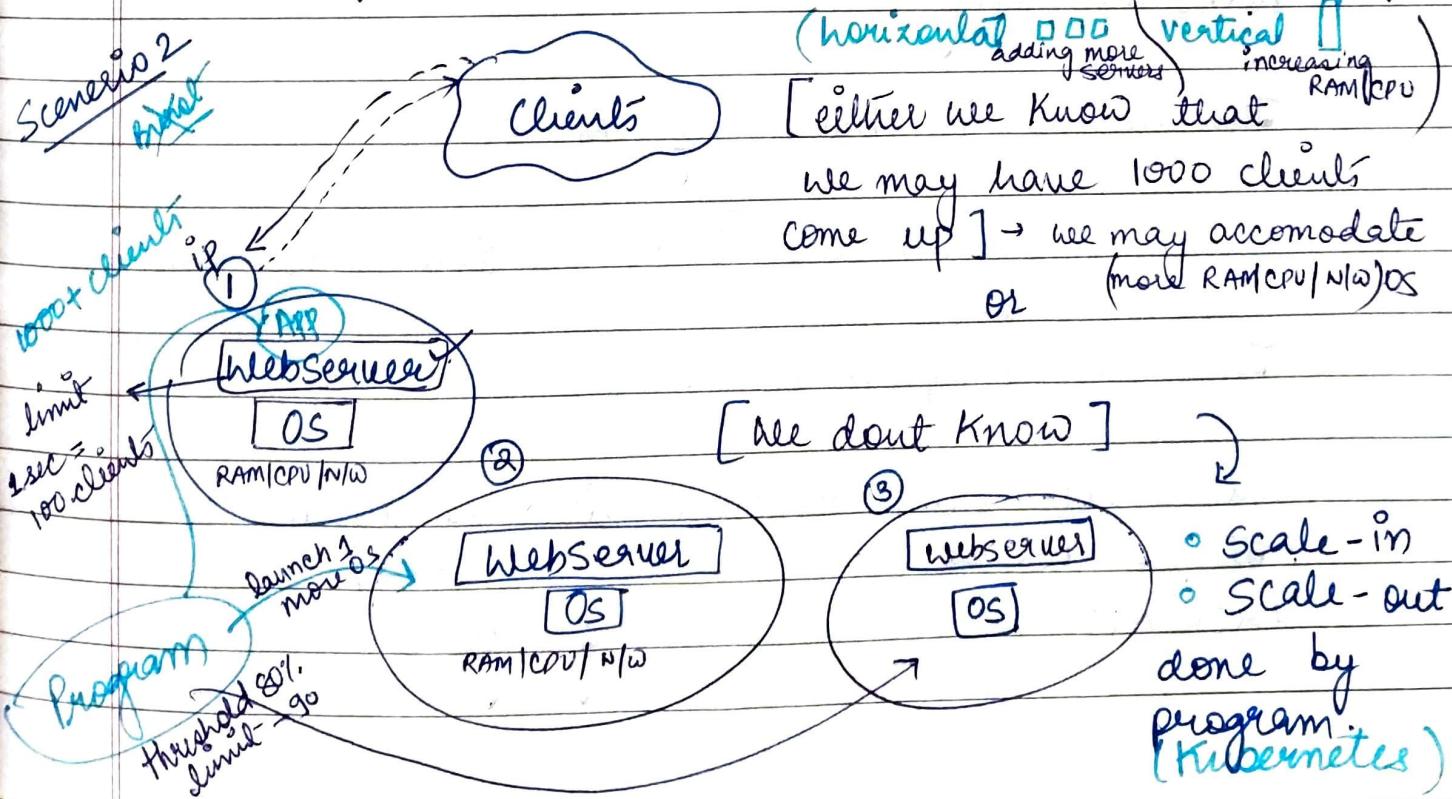


This program is **Kubernetes / docker swarm**

Kubernetes

- monitors multiple of containers
- powerful tool

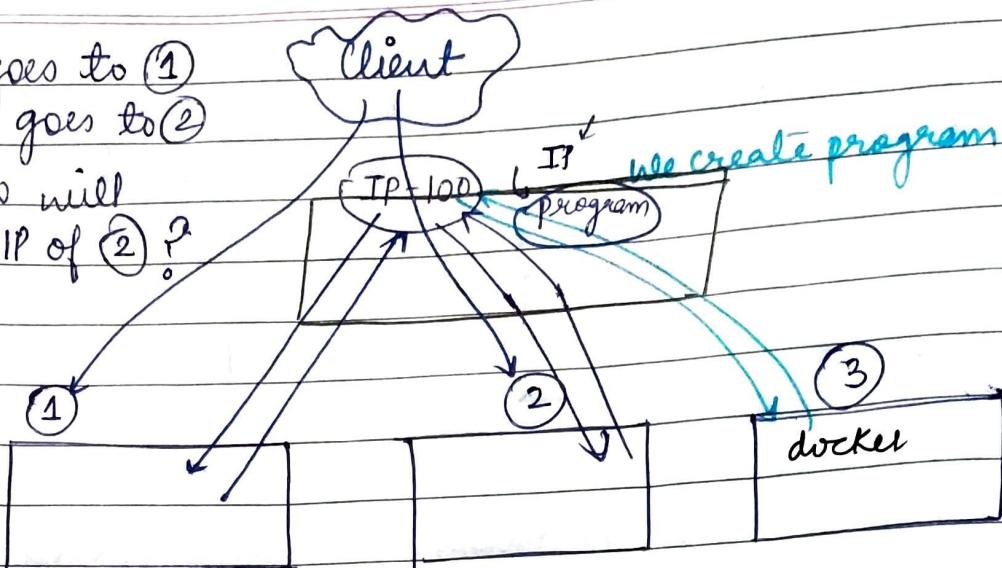
As soon as your environment goes down, Kubernetes relaunches another and hence we achieve fault-tolerance and high availability



Horizontal Scaling

→ Scale In
Scale Out

- ▷ Client goes to ① if busy goes to ② but how will he know IP of ②?



-x-

- These three OS are different.
- All these have different IP.
- This environment is dynamic as when demand increases more OS / servers are added and as demand decreases it removes OS / servers.

IP 100 knows that, it's not the webserver it extracts the information from IP ① OS / webserver only.

This results in balancing of loads, **Load Balancing**.

Client should know **1 ip**, the program does the load balancing.
This IP 100 program is **Kubernetes**

Kubernetes will remove the server IP if not in use. (OS)

Provides **load Balancer** for you.

- Program works as Inverse proxy

vertical scaling

scale-up
scale-down.

CLASSMATE

Date _____
Page _____

- Service discovery → new webserver IP launches and IP100 program recognizes it.
- Service discovery, Inverse proxy, load balancer)
Kubernetes. ↵

* For launching an OS we have 4 methods :-

- 1) Cloud Computing
- 2) Virtualization
- 3) Bare-metal
- 4) Containerization

Why prefer container P

- One click, Container launches OS/ env within 1 sec.

Launch OS → Container



Who will manage containers?

(eg:- Docker containers)

(Monitor these containers)

If something fails, who will provide load balancer part or who will provide scaling part.

(both horizontal/vertical)

Management & Orchestration
(Container Orchestration tools)

- K8S

- Docker Swarm

COE (Container Orchestration Enginee)

▷ Kubernetes:-

Container orchestration system

Container management system

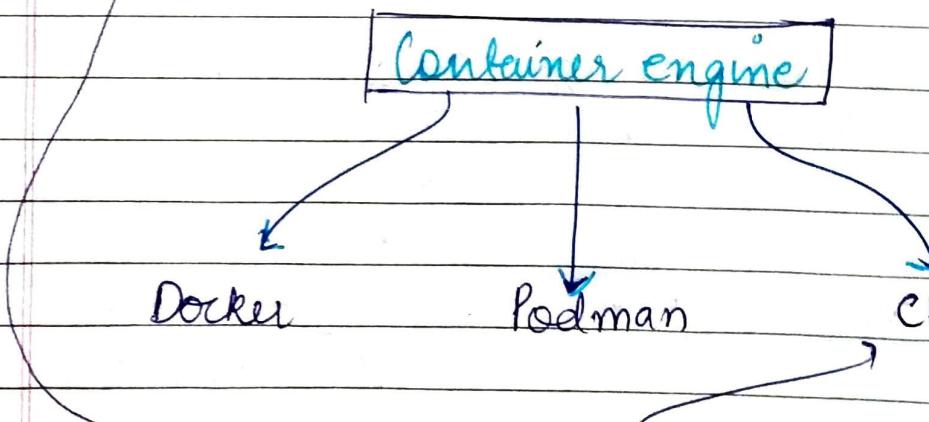
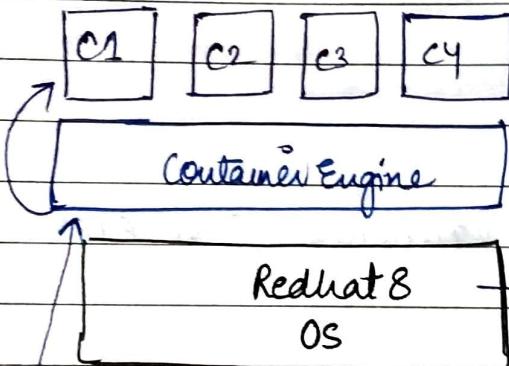
Kubernetes are not the one who are launching OS.

↓
Kubernetes just monitors/manages the containers

▷ Docker - Container engine [Docker Engine]

- launches OS

- Kubernetes instructs docker engine to launch OS with its own commands.



Instructed docker to launch/remove OS and keeps on watching/monitoring

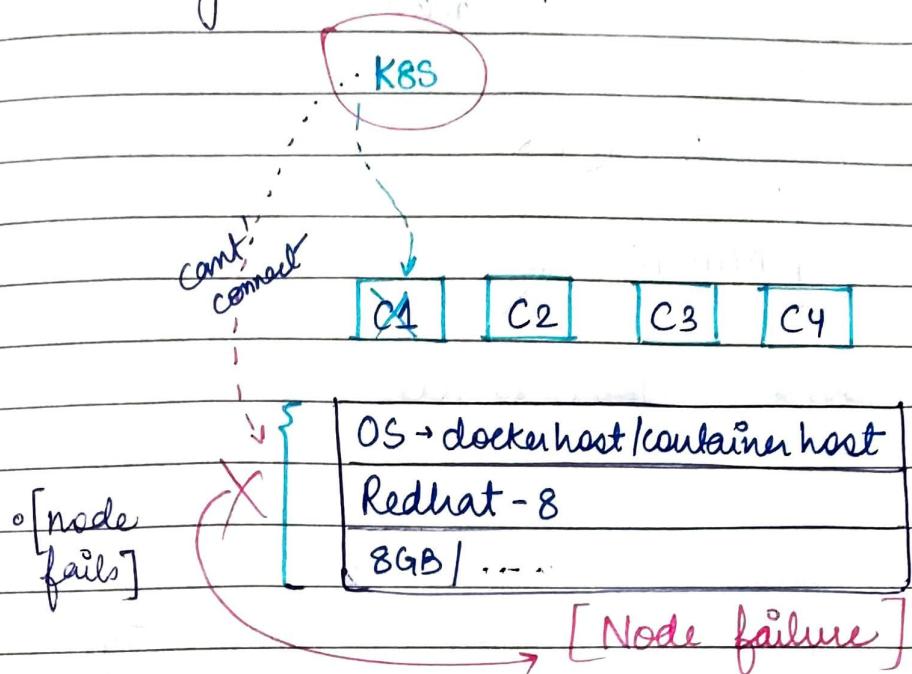
Suppose the container is now running on docker but in future it runs on CRI-O,

- Multi - Node Cluster
- Kubernetes scheduler
- We can have more than one master node as one master node fails the second comes in command

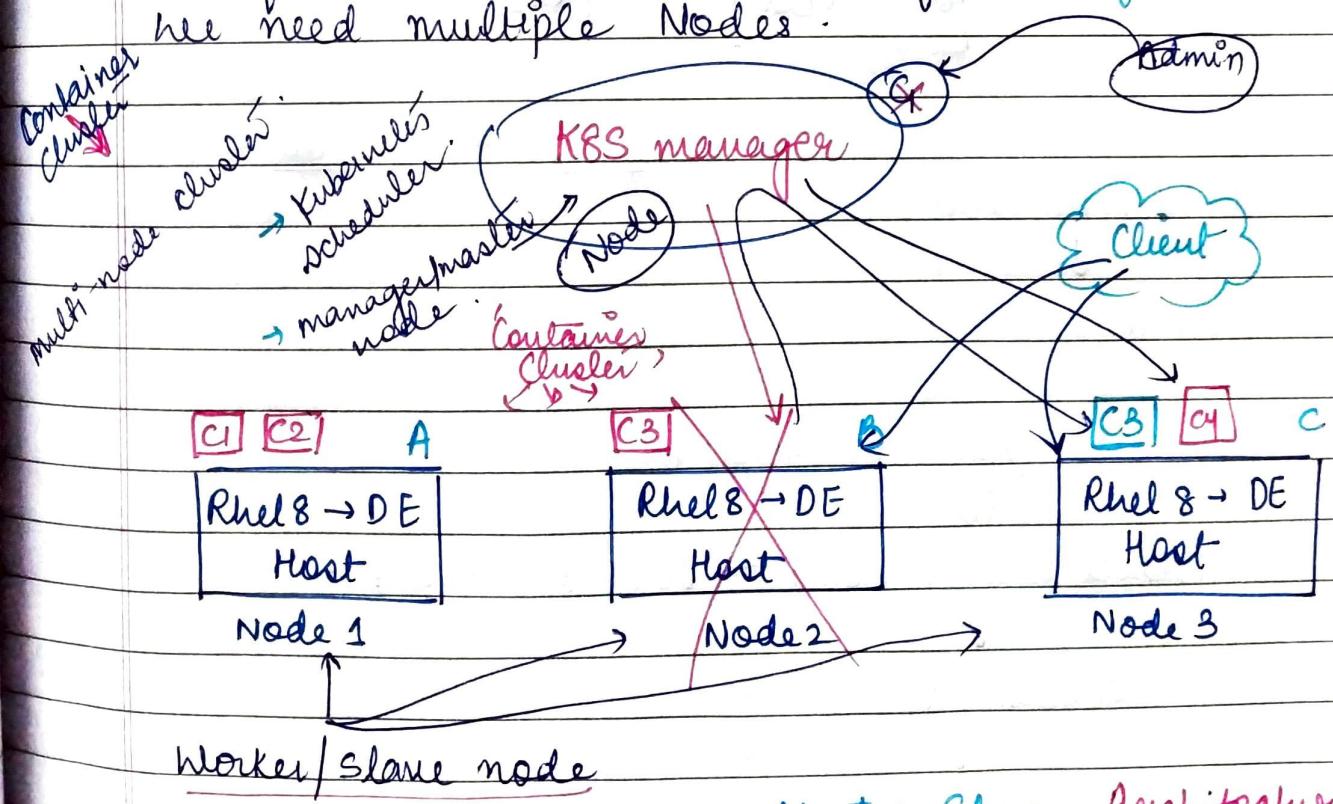
CLASSMATE

Date _____
Page _____

- We just need to Known Kubernetes Command, to manage docker, podman, CRIOS, engine's.

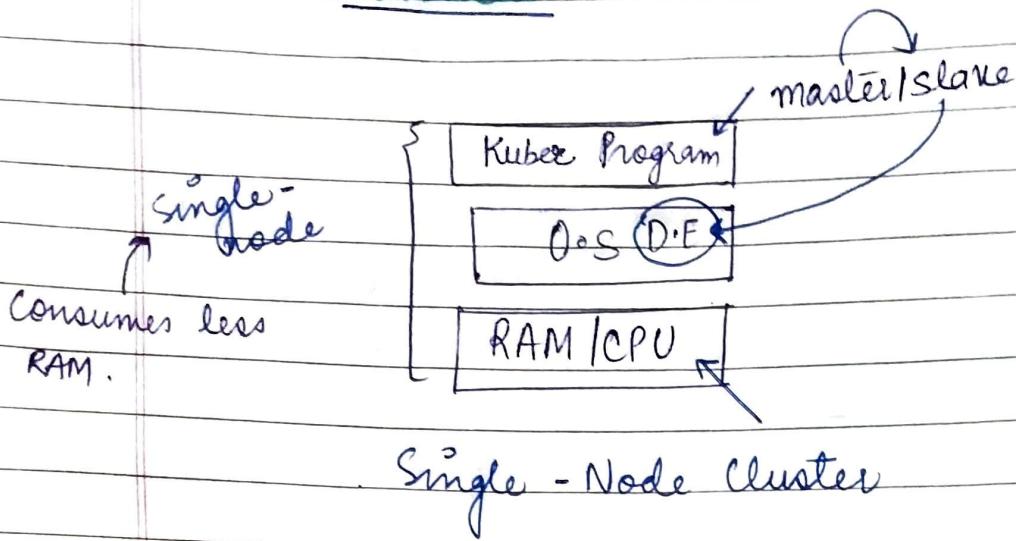


Therefore, to solve this issue of **Node failure**, we need multiple Nodes.

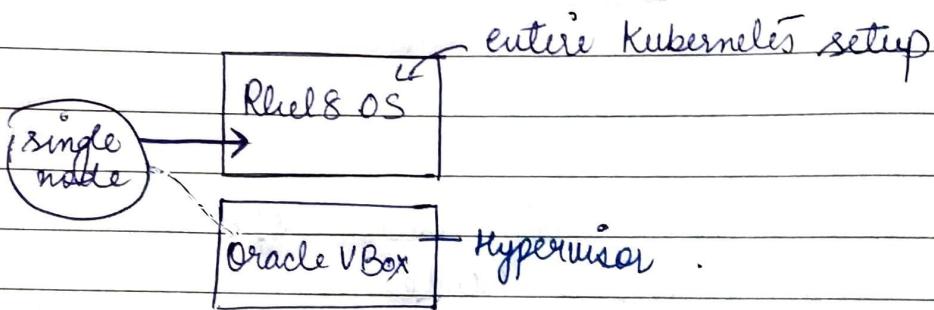


Master-Slave Architecture
Container clusters can be only created by K8S.

▷ Kubernetes Practical



①



① Minikube install

program → Minikube

run on windows

Setup's master-slave
automate .

② windows

③ Supports VBox or Hyper V.
Virtual Box

(Select Virtual Box) → Download exe file



Install all things

go to C program files → program files →
kubernetes .

(now set the environment variables and save the minikube path)



go to Command prompt

Simple and fast

minikube



[single node cluster]

→ minikube version

→ minikube start (downloads iso)
then start

→ minikube start --vm-driver = virtualbox

→ minikube start (download and setup for you)



→ Go to Virtual-box and start OS.

→ dockerhub → vimal13 / ---

- (docker hub) →
- o docker pull vimal13/apache-webserver-php
 - o docker run -dit --name OS3 vimal13/apache-webserver-php

o firefox

on browser

o 172.17.0.4

request

Kubectl

command

Admin/User

Kubectl
(used by admin)

docker

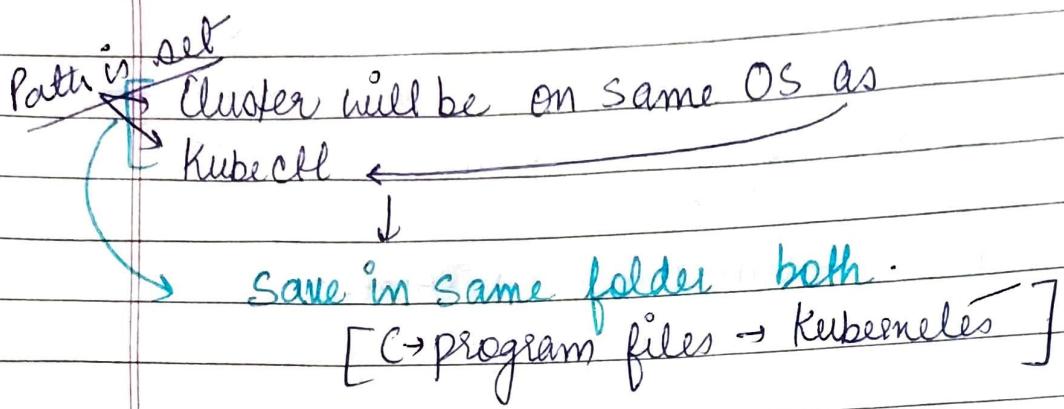
pull the OS
give IP address
monitoring OS

IP address

K8S Cluster

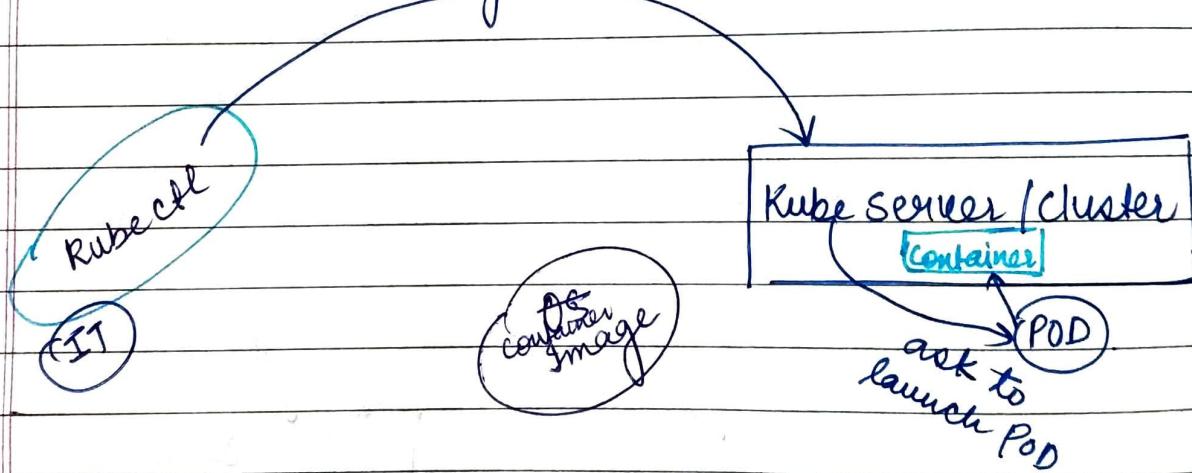
Single Node/
Multi Node

- Download the kubectl command from
`minikube install` (browser)

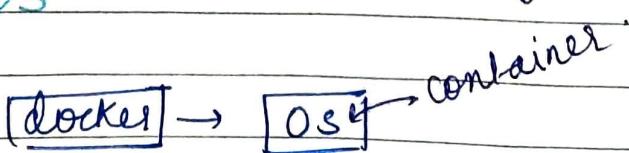


- Minikube - start VM
stop VM and basic

* Kubectl → all major things will be done by this command



- In docker when we launch OS we call it container
- But if we launch containers using Kubernetes its called PODS



- In Kubernetes, if we want to see how many PODS (containers are running)
 - > kubectl get pods (how many pods running)
 - > kubectl deployment myweb --image = vimal13/apache - webserver-php

(download image and launch PODS)

↳ (Container) type

> kubectl get pods

* If one pod fails or we delete then kubectl launches another pod as soon as one fails in its place

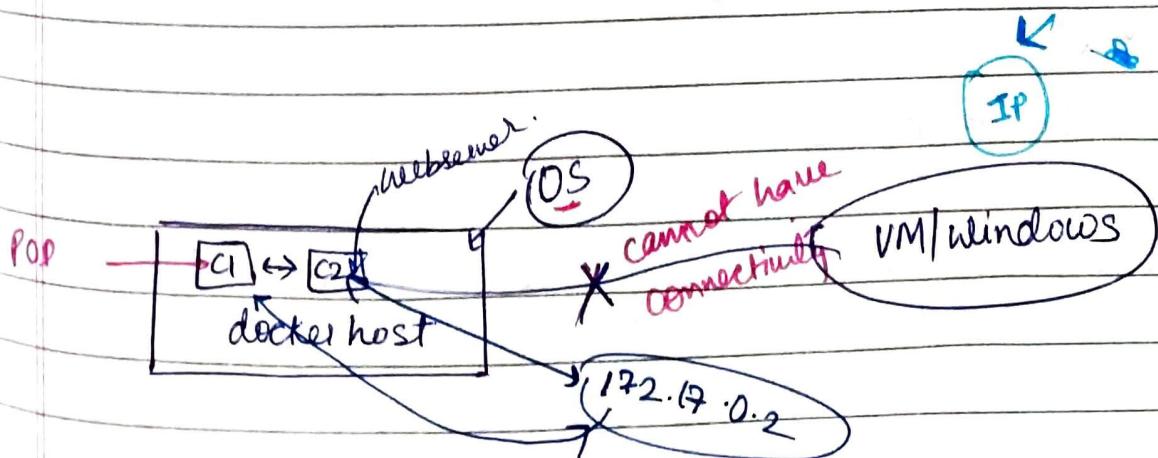
> kubectl pods delete pods podname
(delete pods)

> kubectl get pods

↓ - - -

(Guarantee to restart OS, if it fails OS)

- POD or container running inside cluster



→ Rubectl describe pods(`podname`) (detail about PODS).

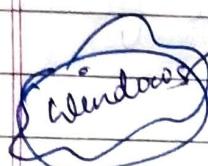
docker

- If I want to have the container to be connected from outside host (outside os) (expose program) ↓ (expose container)

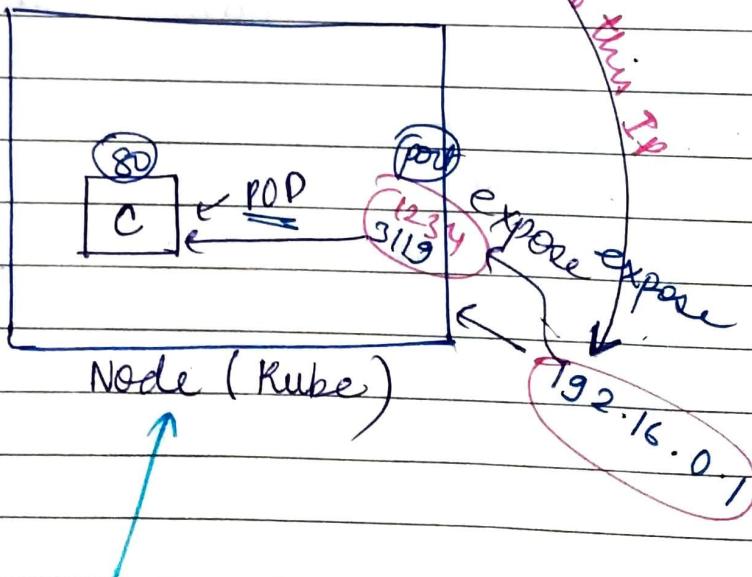
We need concept of (-P) PATing, [-P (PAT)]

- * launch 1 more pod/container (-P)

use = docker



first have to come to this IP



→ You will have connectivity from windows

→ ifconfig

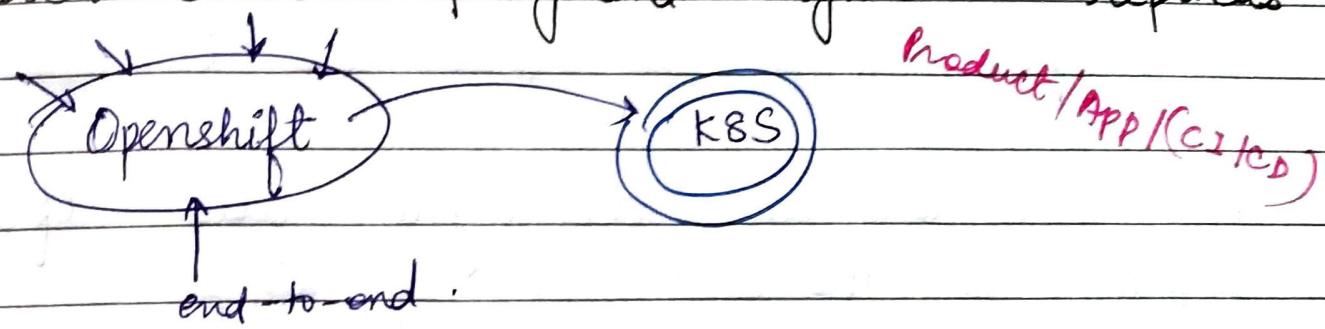
Concept of expose

→ Rubectl expose deployment myweb --type=NodePort
--port=80

► Kubernetes automatically load balances.

→ 172.17.0.3] client can get connected to
→ 172.17.0.2 any pod
→ 172.17.0.5 It keeps on changing due
to load balancing.

> kubectl scale deployment myweb --replicas



> minikube ip. (gives ip)