

KUBERNETES

Workmate

Date

Page No

classmate

Date

Page

Configuration Management

- maintain all change in your infrastructure using softwares.
- immutability

Managing Containers

⇒ Monolithic Application

- \$ application where components (like backend, frontend, chat message, database, etc) are bundled together i.e. deployed as one application, so this is known as monolithic application.

So to make any change in any component you have to deploy everything again.

To solve this problem we use microservices.

Microservices

So here you can deploy each component separately

Ex:

Frontend → S1 Servers

Backend → S2

Database → S3

Messages → S2

Networking → S1

Everything is running in its own container

Fault isolation

- teams divided on frontend, backend, etc.

How to manage?

How to communicate?

Service Mesh

Later.

One application for one virtual machine
This is hell for scaling hence Docker

Dev ops (Development & Operations)

App
Docker
image
containers

\$ docker run hello-world → image name
→ run an image to create a container

How docker image works?

→ docker images contains OS files and app dependencies.

Images are ~~not~~ built in layers

→ each layer immutable file
→ collection of files and directories

How to create your own docker images?
Dockerfile

Orchestration

- helps deploying and managing applications dynamically.
- deploy
 - zero-downtime
 - scale
 - heal containers

cloud native application

- ~~not~~ meets modern business demands and follows all the above principles & also runs Kubernetes. Not same as cloud computing

Kubernetes

- It is much more than a container orchestrator
- u can run it in own cloud on machines on cloud providers
 - migrate from to other providers
 - replicate, ~~scale~~ scale services
 - zero downtime, fault tolerance, etc
 - can use volumes (external storage)
 - provides load balancing
 - can store secret info (passwords...)
 - & much more

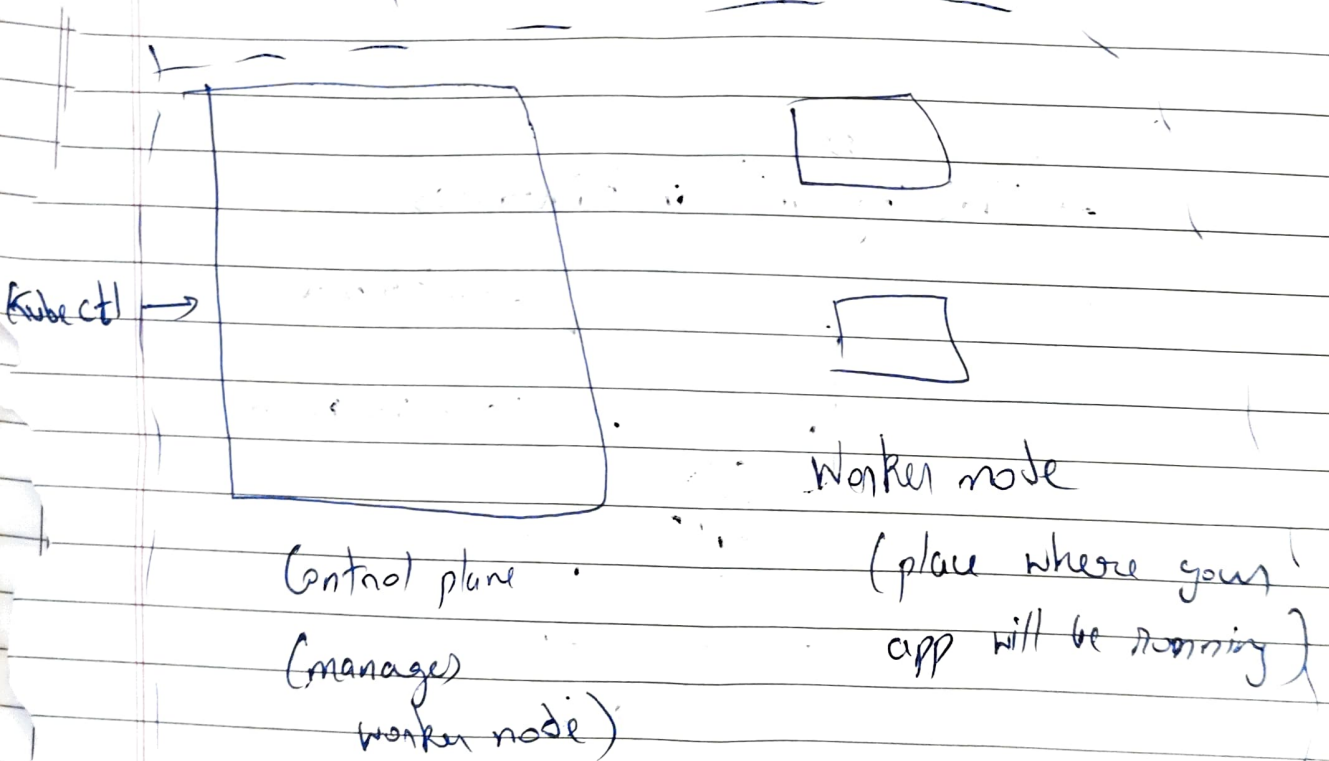
Google made K8s (Kubernetes) & donated to CNCF

Kubernetes Cluster

Cluster

→ control plane + nodes

→ VM



Kubernetes cluster

(collection of all the worker nodes & control plane)

Kubectl

→ Kubernetes cli (command line tool)

likes \$ docker on git

Ex → tells control plane start 10 app, control plane will check for empty worker nodes.

2 ways to interact:

Better

i) Declarative:

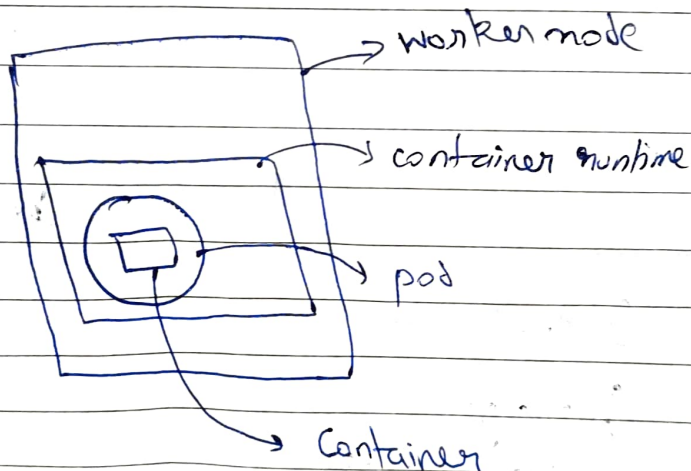
- creates manifest files (YAML files).

ii) Imperative:

- telling cluster for specific commands

Kubernetes Pods

→ Scheduling unit in Kubernetes



- ① Create microservice
- ② ~~Att~~ Containerise it
- ③ Put container in ~~pod~~
- ④ Deploy these pods to controller