

## Experiment No: 10

Aim: To Study and implement container orchestration using Kubernetes.

Theory:

- 1) Explain need of container orchestration tool
  - ⇒ Container orchestration tools are needed to manage deployment scale, and automate containers effectively.
  - They ensure containers run as intended, manage resource allocation, handle networking, storage, and maintain high availability across a cluster of machines.
  - Container orchestration tools automate various tasks involved in managing containers, such as deployment, scaling, load balancing, health monitoring, and recovery.
  - As the number of containers and the size of the infrastructure grows, manual management becomes impractical.

2) What is Kubernetes? Describe its features.

- ⇒ Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications.
- It's features are:-
  - i) Automated rollouts and rollbacks
  - ii) Service discovery and load balancing.

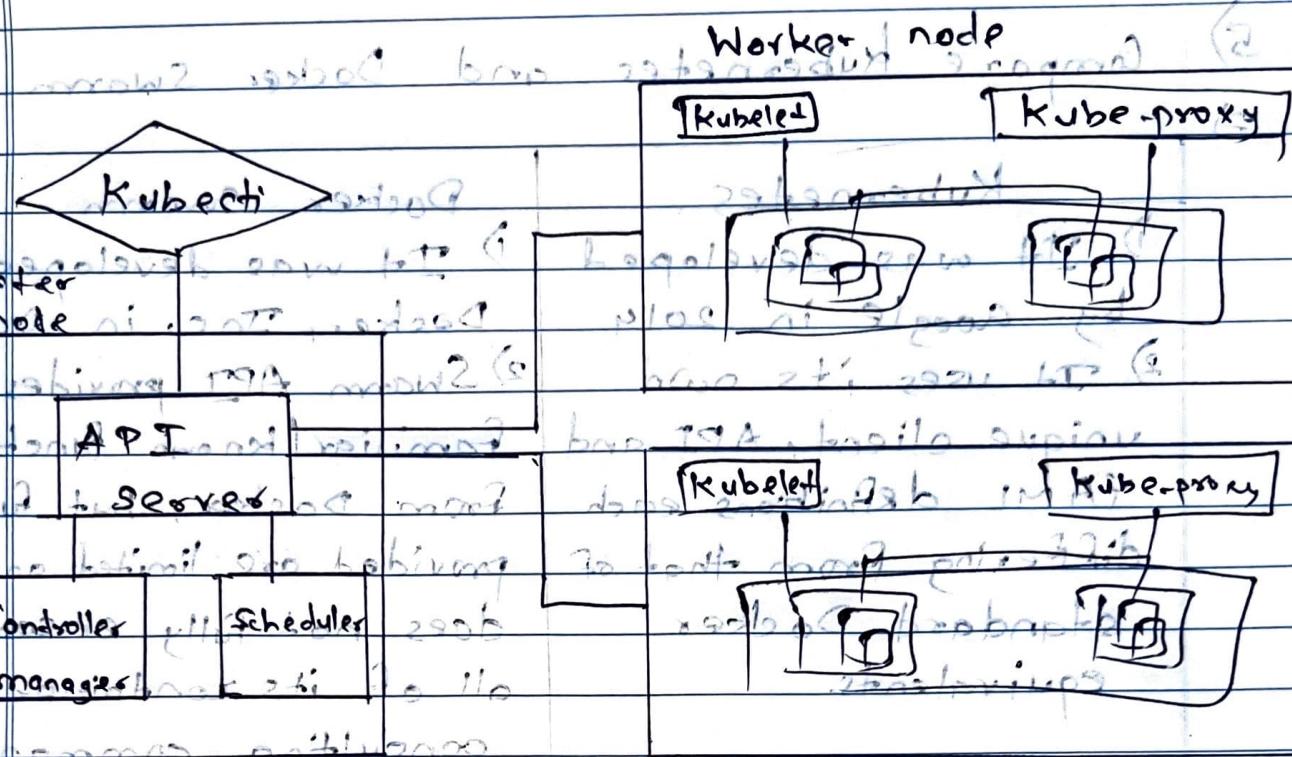
## Cloud Formation

- iii) Storage orchestration:  $\frac{1}{2}$  OT 1 min
- iv) Self-healing:  $\frac{1}{2}$  min
- v) Secret and configuration management
- vi) Automatic bin packing:  $\frac{1}{2}$  min
- vii) Batch execution
- ix) Horizontal scaling:  $\frac{1}{2}$  min
- x) Multi-IP failover:  $\frac{1}{2}$  min
- x) Designed for extensibility:  $\frac{1}{2}$  min

Q) Explain Kubernetes components & its working mechanism and its architecture.

- Ans) Kubernetes components include the Master and Worker nodes to establish a cluster.
- The Master node oversees the cluster and manages its status through various components such as the API server, controller manager, scheduler, and etcd.
  - Worker nodes host the containers and run the necessary Kubernetes components like kubelet, kube-proxy, and container runtime.

Kubernetes has a client-server architecture and has master and worker nodes, with the master being installed on a single Linux system and the nodes on many Linux workstations. It follows a simple bootstrap mechanism where each node has persistent storage.



abnormal, broad ratio no (2) from a stronger LTO (2)

4) Differences between PODs and nodes

↳ nodes are created in groups like auto scaling  
label, as pod spec. in sub podNode

1) A pod is smallest  
deployable unit in  
Kubernetes (2)

2) It represents a single  
instance of a running  
process in your cluster.

3) PODs are ephemeral  
and can be created,  
destroyed or replicated  
dynamically based on  
workload requirement.

2) A node is a physical  
or virtual machine in a  
Kubernetes cluster.

2) It is the underlying  
infrastructure where  
POD runs

3) Nodes are responsible  
running and managing PODs  
providing the necessary  
computing, networking  
and storage resource.

## 5) Compare Kubernetes and Docker Swarm

### Kubernetes

- 1) It was developed by Google in 2014
- 2) It uses its own unique client, API and YAML definitions each differing from that of standard Docker equivalents.

- 3) It supports a more complex, flexible architecture with stronger service guarantees due to which performance slows down in comparison to Docker Swarm.
- 4) It supports auto-scaling.

### Docker Swarm

- 1) It was developed by Docker, Inc. in 2013
- 2) Swarm API provides many familiar/known functionalities from Docker, but functionalities provided are limited and it does not fully encompass all of its containers' consulting commands.

- 3) On other hand, supports simple architecture, so in terms of sheer speed, it always has an added advantage.
- 4) It cannot do auto-scaling.

~~some Q&A about k8s in passing~~

~~Q1: What is the difference between Docker Swarm and Kubernetes?~~

~~A: Docker Swarm is a distributed system for managing multiple Docker hosts. It provides a way to run Docker containers across multiple hosts and manage them as a single unit. Kubernetes is a more advanced system that provides a way to run Docker containers across multiple hosts and manage them as a single unit, but it also provides a way to run other types of containers, such as Java, Python, and Node.js. Kubernetes is designed to be more scalable and easier to manage than Docker Swarm.~~