i) What is K8S
- Open Source Container orchestration tool
- Developed by google
- Helps manage containerized applications in different deployment environments.

Need for container orchestration tool
- Trend from Monolith to Microservices
- Increased usage of containers
- Demand for a proper way of managing those hundreds of containers.

Features do orchestration tools offer
- High Availability or no downtime
- Scalability or high performance
- Disaster recovery - backup and restore

K8S Architecture
- There o is one master node and multiple worker nodes
- Where each worker node has a kubelet process running on it

  kubelet :- Is actually a kubernetes process that makes it possible for the cluster to communicate each other and execute some tasks on those node.

- Each worker node has containers of different applications deployed on it.

- On masternode actually runs several K8S processes that are absolutely necessary to run & manage the cluster properly.

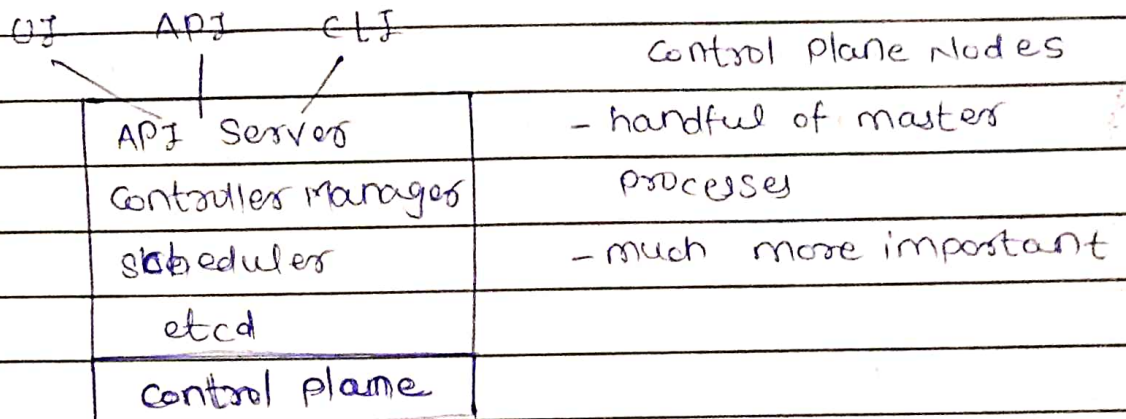 API Server :- Also is a container an api server entrypoint to K8S cluster
              UI, API, CLI → all talk to api server

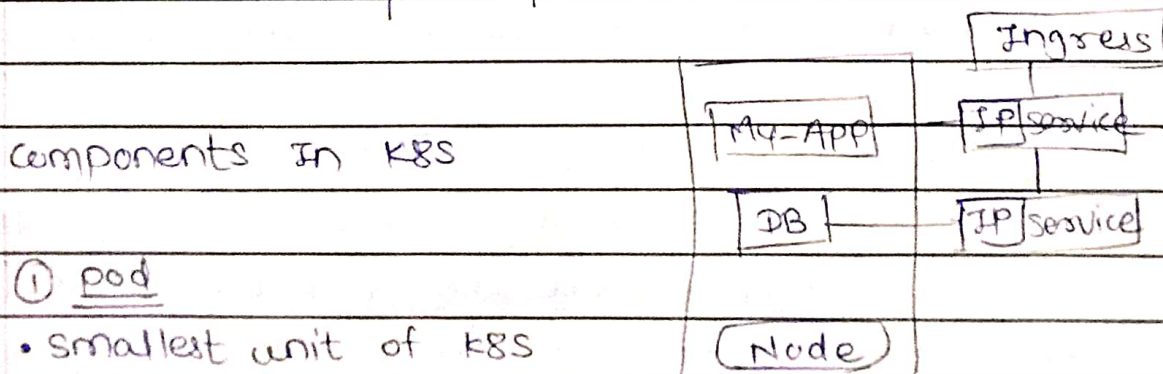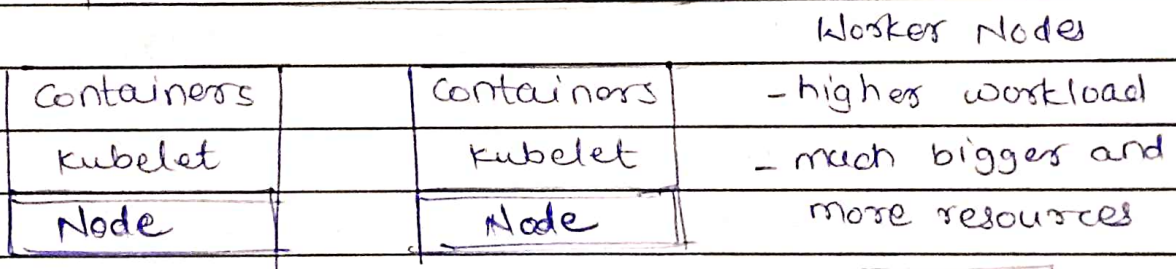**Controller Manager :-** keeps track of whats happening in the cluster

Scheduler :- scheduling containers on different nodes based on the workload

etcd :- K8S backing store

- **virtual Network :-** creates one unified machine

UI API CLI	
	Control Plane Nodes
API Server	- handful of master processes
Controller Manager	
Scheduler	- much more important
etcd	
control plane	

virtual Network

				Worker Nodes
Containers		Containers		- higher workload
kubelet		kubelet		- much bigger and more resources
Node		Node		

Components In K8S

```
                    ┌─────────┐
                    │ Ingress │
         ┌──────────┼─────────┤
         │ My-App   │ IP Service │
         │          │          │
         │ DB ──────│ IP Service │
         │          └─────────┘
         │ (Node)   │
         └──────────┘
```

① **pod**

- smallest unit of K8S
- Abstraction over container
  (You only interact with the K8S layer)
- Usally 1 Application/container per pod
- Each pod contains its own IP address through that they can communicate
- pods can die easily
- New IP address on re-creation
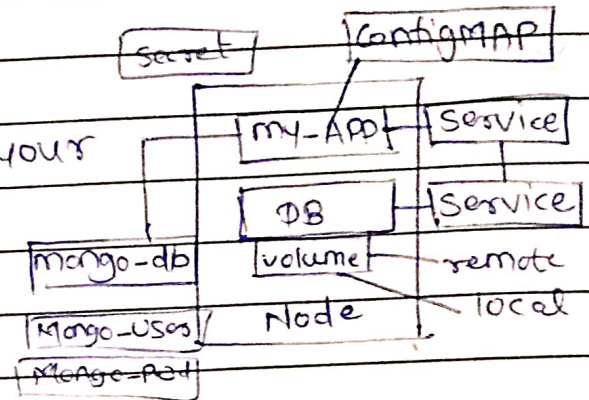
② Service (svc)

- permanent IP address
- Lifecycle of pod & service not connected.
- You specify the type of service on creation
- Internal service is the default type

③ Ingress (ing)

- Instead of service the request goes first to ingress and it does the forwarding then to the service

④ Configmap (cm)

- External configuration of your application

⑤ Secret

- Used to store secret data
- Reference secret in Deployment/pod

⑥ vol

- Storage on local machine
-