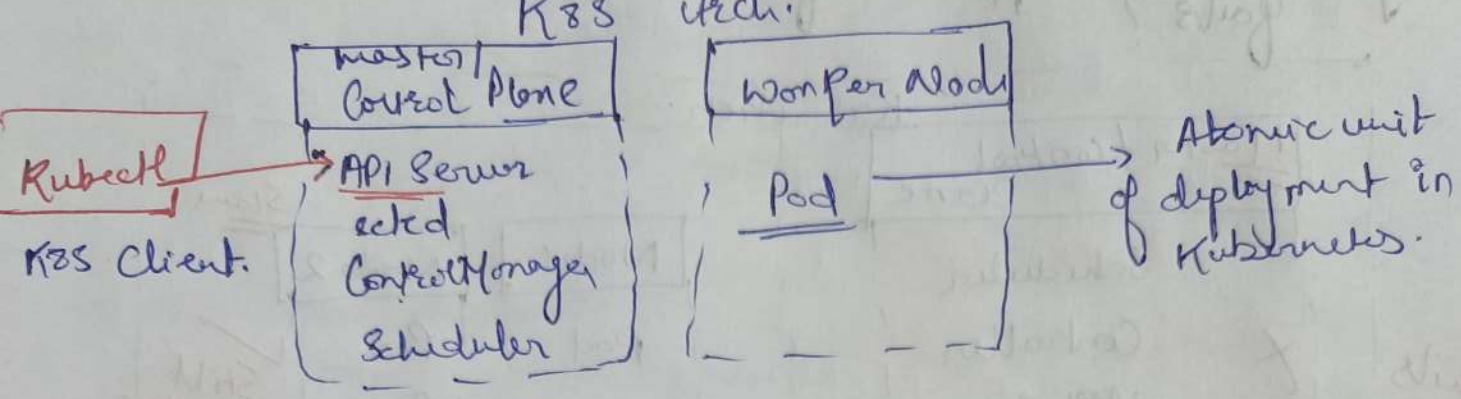
Q2 Explain Kubernetes Architecture



- Master / control plane have some components which takes care of the cluster orchestration.
- Worker node are responsible for working running apps inside the cluster.

⇒ Control Plane components :

① API Server : It exposes API for interacting with the cluster.

Eg we have "kubectl CLI"

To deploy a pod.

Any we run (kubectl) create : cmd. → API Server.

This Info is sent to API Server inside K8S Cluster.

Then API Server receives that request and perform certain validation & work with other components in control plane. .. cont.

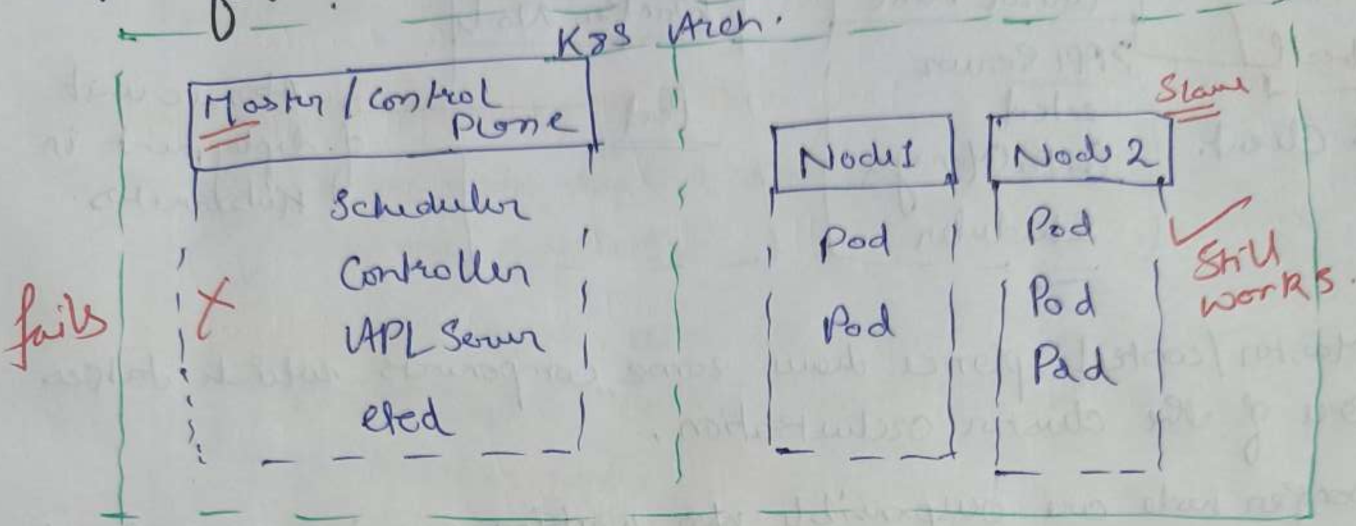
② etcd : dist. key-value store, that store the state of cluster.

③ Control Manager : Manage & execute various controls.

Eg. Deployment Controller

Imp

Q. What happens if Kubernetes Control-Plane fails?



-> But certain operations will be restricted.

we can't

× ① Deploy new app -> bcoz we deploy the new app with help of sending reqn to API Server.

or

× ② can't delete, update, create new pod

× ③ Can't auto heal a pod

× ④ Horizontal pod autoscale (HPA) : can't inc. pod.

-> All the functionalities won't work if control plane is down. But existing pods over worker/slave node still works.

-> Apps running inside the cluster is not disturbed, cont. to run.

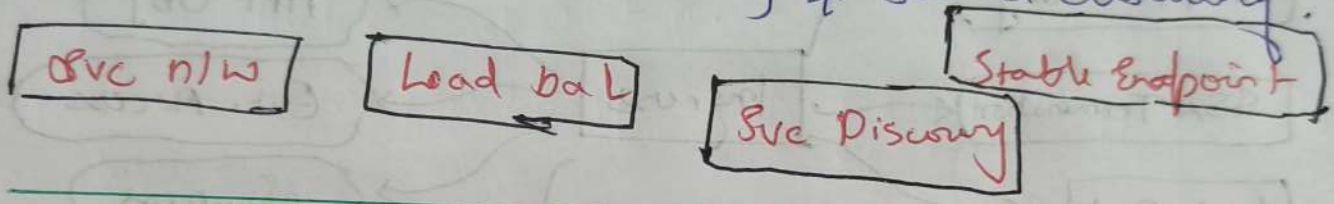
Q14 How does K8S handles container networking?

→ K8s assign each pod on unique IP address, and pods can comm^n with each other across nodes using this IP Addr^n.

→ K8S also support service n/w, which provide a stable endpoint for accessing pods and allows load balancing & service discovery.

Pod → unique IP

Pod → unique IP

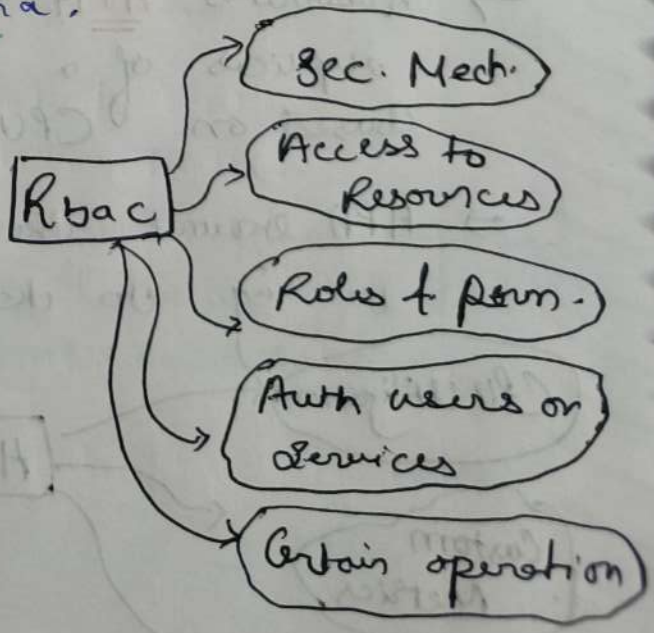


Q15 Explain RBAC "Role based access control" & imp?

→ K8S Rbac is a security mechanism that regulates access to K8S resources based on roles & permission.

→ Rbac ensures that only auth. users or services can perform certain operation within cluster, ensures security & governance.

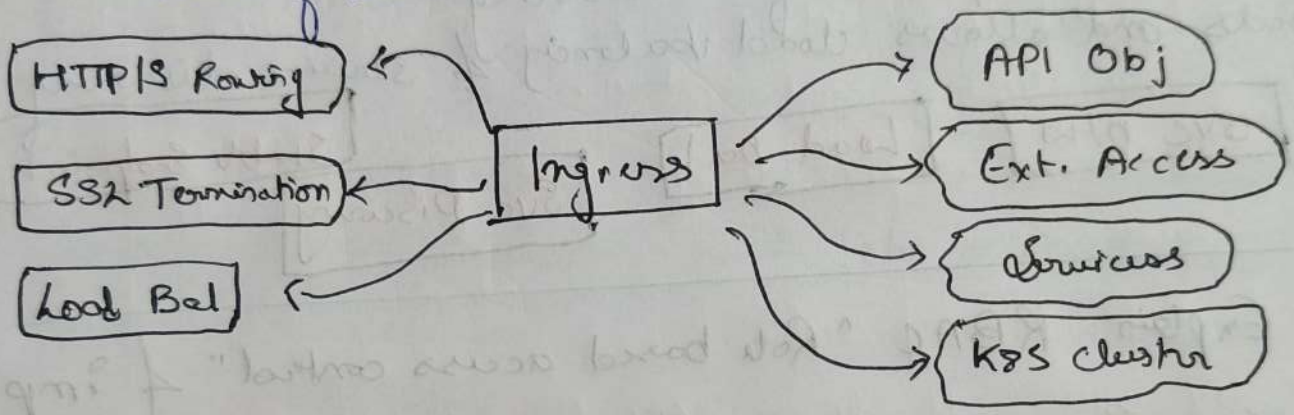
→ It allows admins to define fine-grained access controls and enforce the principle of least privilege.



Q12 What is the purpose of Kubernetes Ingress?

→ Kubernetes Ingress is an API Object that manages external access to services within a K8S cluster.

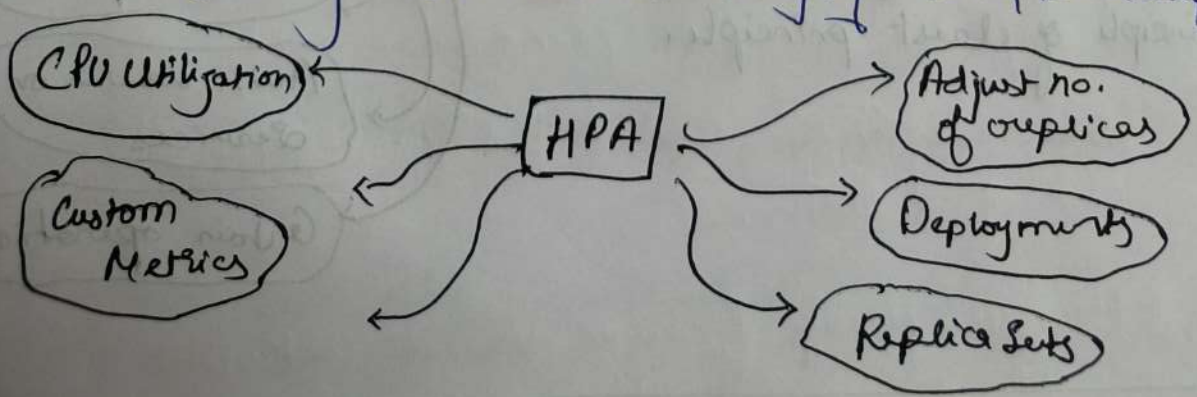
- Provide HTTP and HTTPS routing, load balancing, SSL termination, and name based virtual hosting.
- Ingress allows external traffic to reach services running within cluster.



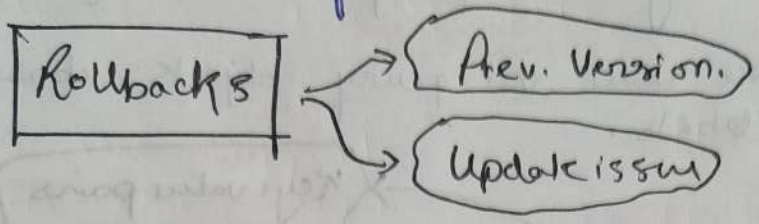
Q13 Explain the concept of K8S Horizontal Pod Autoscaler (HPA).

→ Kubernetes HPA automatically adjust the no. of replicas of a deployment or replica set. of based on CPU utilization or custom metrics.

→ HPA ensures the desired no. of pods are running to handle varying workloads efficiently.

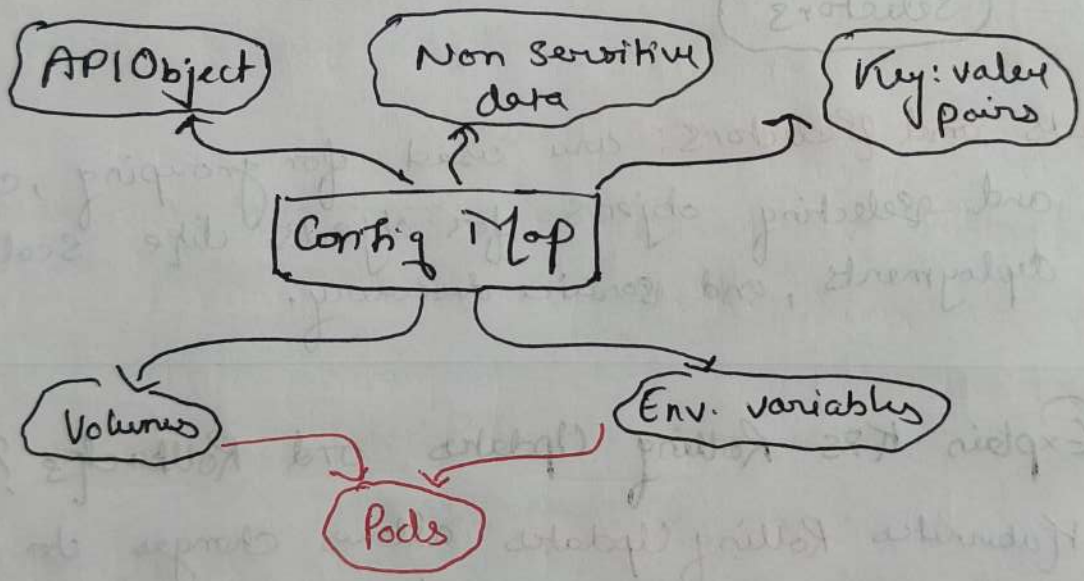


→ Rollbacks : revert to a previous version of deployment in case of issues or failures during an update.



Q11 What is K8S config maps, and how is it used?

→ A Config Map is an API Object help to store non sensitive config data in key: value pairs.

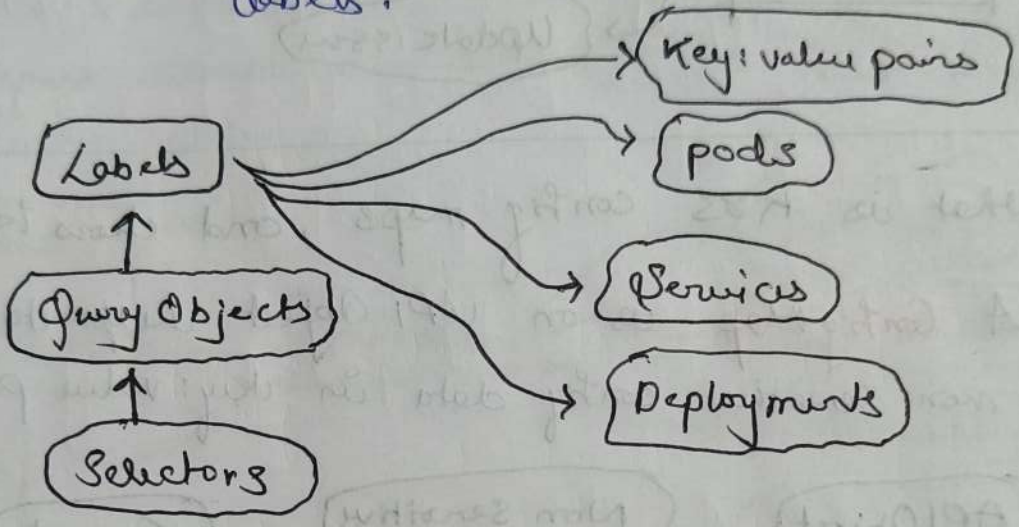


→ Config maps can be mounted as volumes or exposed as env. variables in pods, allow apps to consume config data without hardcoding it.

Q9 What are Kubernetes Labels and Selectors, and how are they used?

→ Kubernetes Labels: are key-value pairs attached to objects like pod, service & deployments.

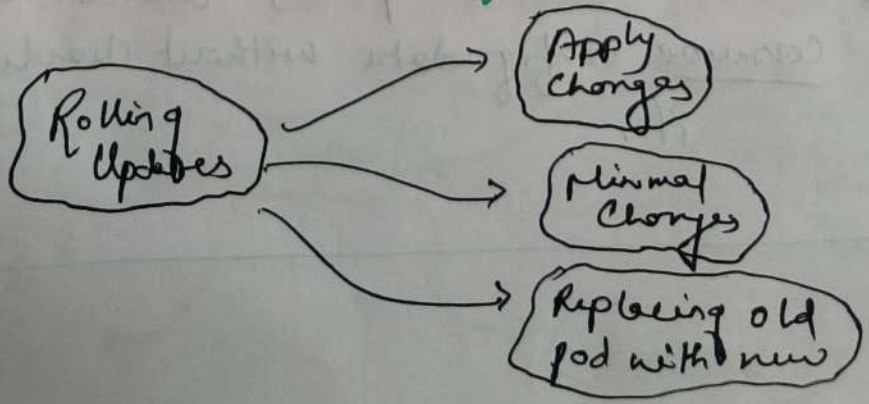
→ Selectors: are used to query objects based on labels.



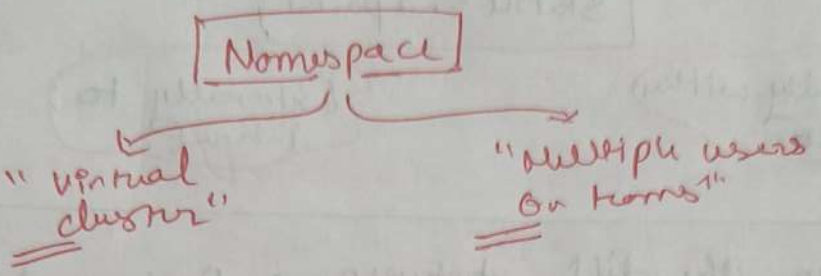
- Labels and Selectors: are used for grouping, organising and selecting objects for objects like scaling, deployments, and service discovery.

Q10 Explain K8s Rolling Updates and Rollbacks?

→ Kubernetes Rolling Updates allows changes to be applied to a deployments with minimal downtime by gradually replacing old pods with new ones.

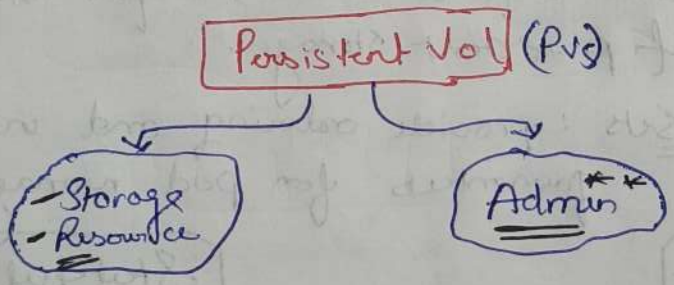


→ Namespace are useful for organising & isolating resources, managing access control, and avoid naming collisions.

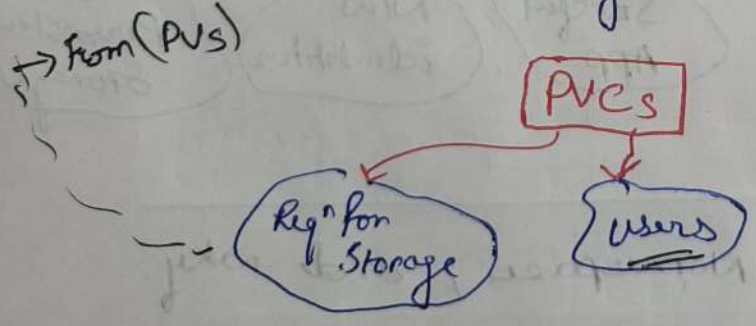


Q8 Explain the concept of Kubernetes Persistent Vol (PVs) and Persistent Vol Claims (PVCs)?

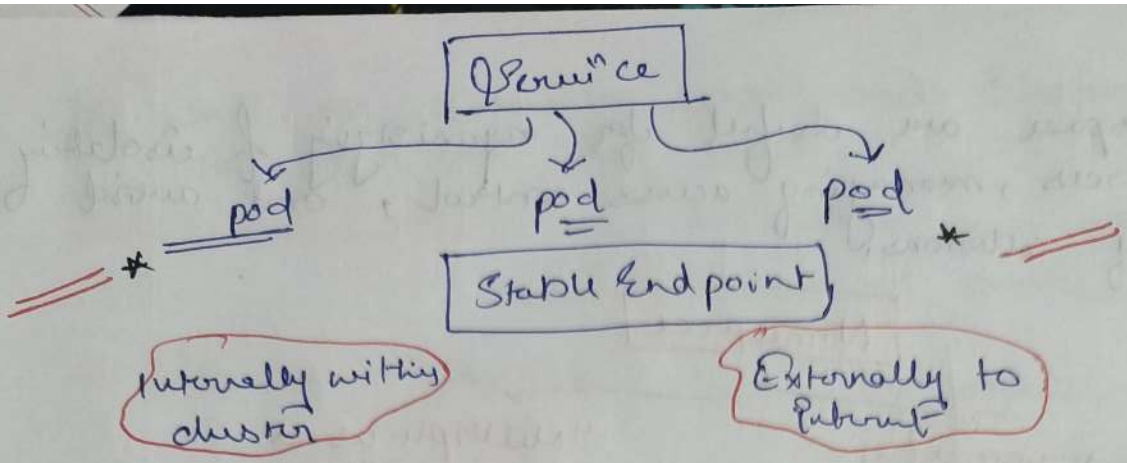
→ Persistent Volume: are storage resources provisioned by an admin in the cluster.



→ Persistent Vol Claims (PVCs): are requests for storage made by users.



→ (PVCs) consume (PVs), providing a way to abstract storage requirement from underlying storage details.

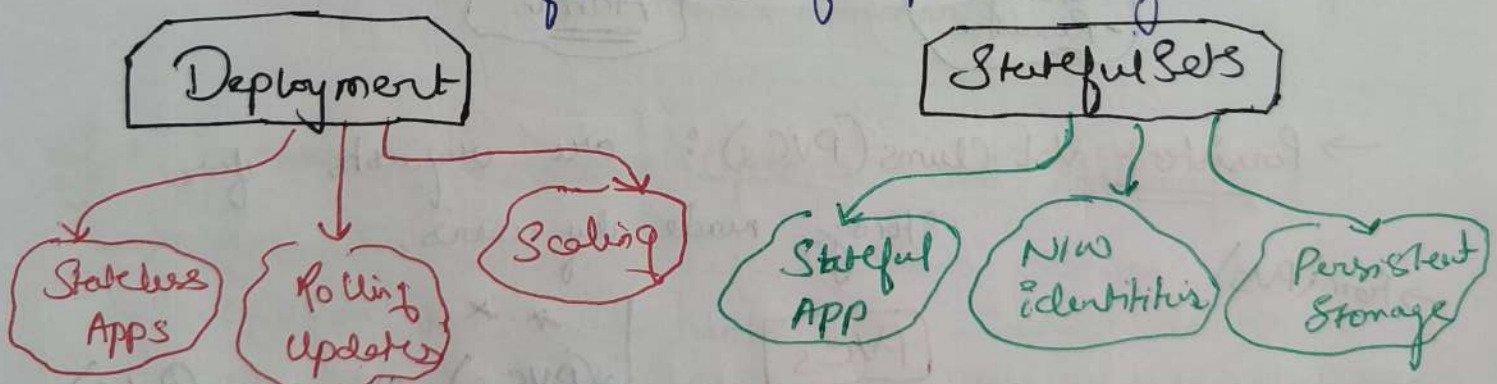


Note *

Q6 Explain the diff between a Deployment and a StatefulSet in Kubernetes?

* Deployments: are used to manage stateless app & provide features like rolling updates & scaling
* StatefulSets: are used to manage stateful app that require stable, unique n/w identifiers & persistent storage.

→ StatefulSets: provide ordering and uniqueness guarantees for pod management.



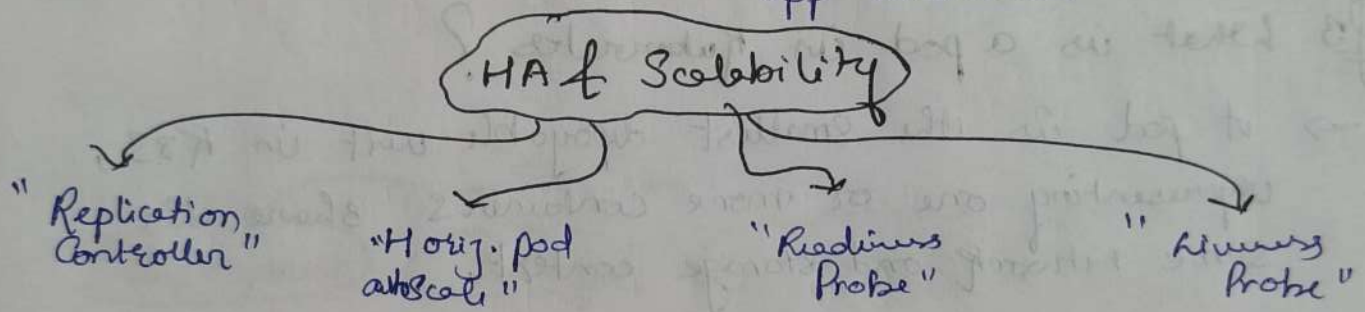
Q7 What is Kubernetes Namespace, and why is it useful?

→ A Kubernetes Namespace is a virtual cluster within a K8S cluster.

→ It provides a way to logically divide cluster resources between multiple users or teams.

Q4 How does K8S ensure high avail (HA) and Scalability of apps?

→ K8S ensures HA & scalability through K8S features like:
- replication controller → maintain specific no. of identical pods.
- horizontal pod autoscale: which automatically adjust the no. of pods in deployment based on resource usage.
- readiness & liveness probes: the health of app instances.



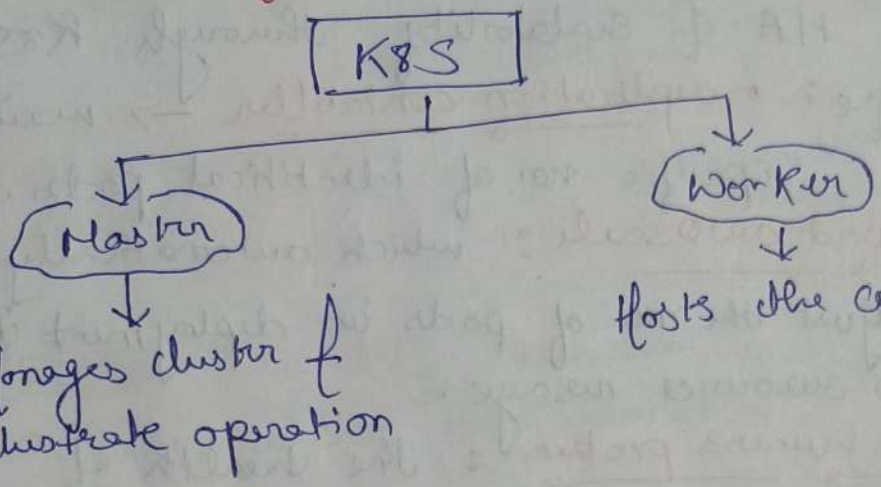
Q5 What is K8S Service & how does it work?

→ A K8S Service is an abstraction that defines a logical set of pods and a policy by which to access them.

→ A Service provides a "stable endpoint" for accessing a set of pods, abstracting the underlying infra.

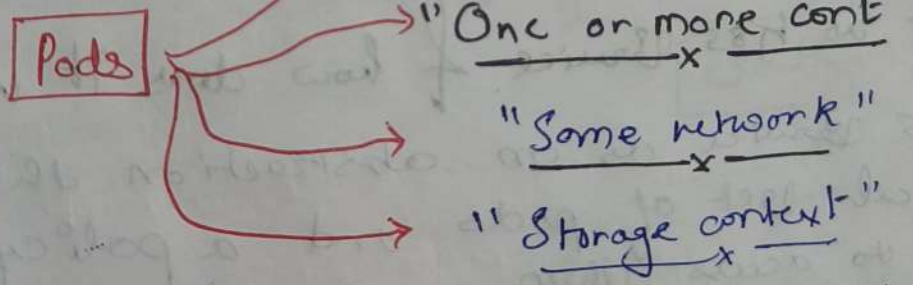
→ Services → can be exposed internally within the cluster or externally to the internet.

→ The key components include the API Server, Scheduler, Controller manager, etcd, Kubelet, and Kube-proxy.



Q3 What is a pod in Kubernetes?

→ A pod is the smallest deployable unit in K8S, representing one or more containers share the same network and storage context.

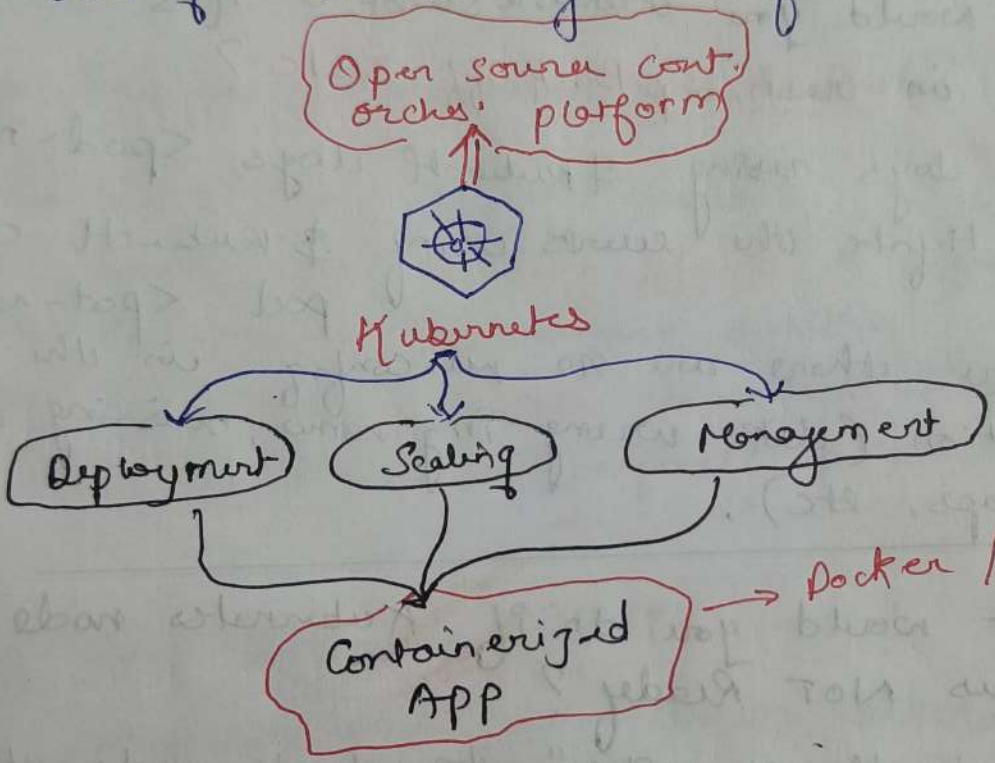


→ Pods are the basic building blocks of apps deployed on Kubernetes, and they can be scaled horizontally.

Top 15 Kubernetes Interview Quest.

Q1 What is Kubernetes, and what problem does it solve?

→ K8S is an open source container orchestration platform designed to automate the deployment, scaling and management of containerized apps.



→ It simplifies management of containerized apps across cluster of hosts, addressing challenges related to deployment, scaling and maintenance.

Q2 Explain the architecture of Kubernetes?

→ Kubernetes follows a master-worker arch.

→ Master node manages the cluster & orchestrates operations, while the worker node host containers.