Lecture 6 - 18th Oct 2025

Kubernetes Intro

- Container Orchestration: Kubernetes schedules and manages containers, ensuring they run in a reliable and scalable manner.
- Automated Scaling: It can automatically scale applications up or down based on demand.
- Load Balancing: Kubernetes can distribute traffic across multiple containers for high availability.
- Self-Healing: It detects and replaces failed containers to maintain application health.
- Rollouts and Rollbacks: Kubernetes facilitates controlled updates and rollbacks of application versions.
- Resource Management: It efficiently allocates computing resources to containers.

Architecture

→ master - slave architecture

Detailed

- Pod is encapsulation of one or more containers.
- IP addresses NOT used for referencing pods → "Labels" are used. & "Annotations".
- Self-Creation → min_pods → replica set controller

= 5

→ rollbacks

- kube-yaml-

→ Types of kind : → Service
→ Deployment

Volumes

"Define the size of storage using

"kind : Peersistent Volume"

↓

and claim the piece of it using

"kind : Peersistent Volume Claim"

For upcoming kubernetes assgn

- use gcp.
- setup local using "minikube"
- more advanced → "kind".

→ install "kubectl".

Demo.

1. Creating namespace → similar to IAM
   → a team has a "ns" and only works with that.

(1) - Service → permanent IP address of Pod.
↓
Service is abstrac^n for internal IP address.

(2) External Service.
→ uses ingress → BCOZ instead of IP of pod → we have a domain name.

also called as Service

(3). DB containers can't be replicated because we have images but NOT states.

Then what to do often DB fail.

ConfigMap -

has URL of external

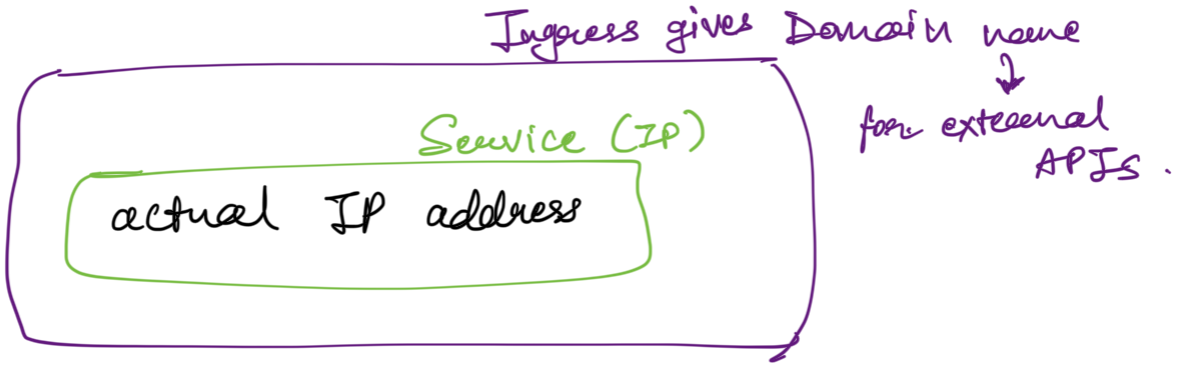
Secret

ConfigMap - config details external to cluster.

Ingress

Service

Volume → local
→ remote



(10) Secret → DB for user credentials.