



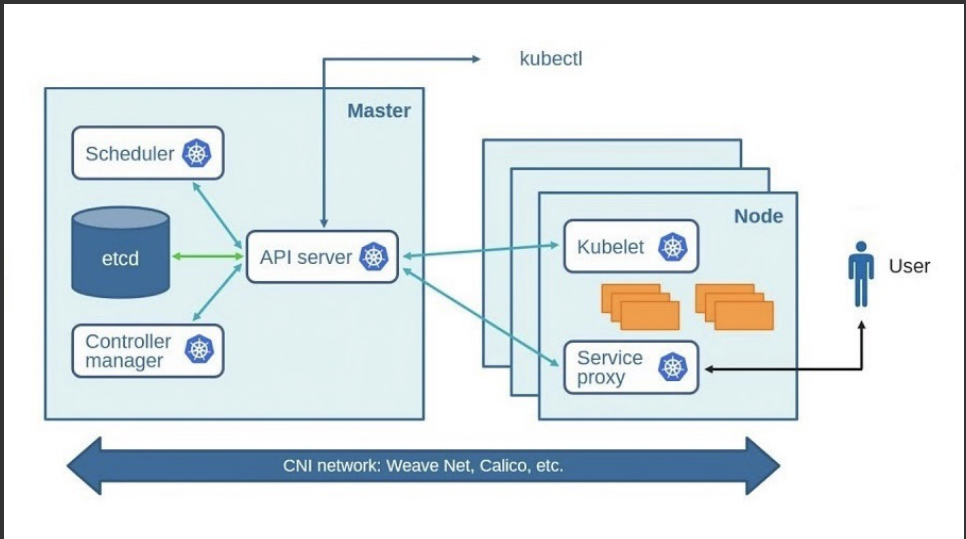
***TRAIN WITH***  
***SHUBHAM***

# Kubernetes (CKA)

- ★ Cluster Architecture
- ★ API Primitives
- ★ Services



## ★ Cluster Architecture



## ★ Container Runtime Interface (CRI)

→ You can use crio / rkt as container services for Kubernetes

→ docker.shim is the service used to support docker in Kubernetes (Not used Now)

Docker → Tool to run containers

containerd → Docker engine uses containerd which pulls images from registries, manages them and gives it to lower-level runtime (host)

CRI-O → Container Runtime interface implements K8s 2 alternative to containerd.

## ★ ETCD

- ↳ contains all the information related to Nodes, Pods, configs, secrets, Accounts, Roles, Bindings, etc.
- ↳ Key value datastore
- ↳ ETCDCTL is a cli tool used to interact with ETCD server

## ★ Kube-apiserver

- ↳ It is used to authenticate user, validate requests, Retrieve data, update ETCD, communicate with other components of cluster.

## ★ Kube-controller manager

↳ Continuously monitor various components of the cluster and works towards managing / restoring to the desired state.

## ★ Node controller

↳ communicates with Kubeapi server and manages nodes. (Every 5 seconds)

↳ Checks again for 40 seconds then marks as "Unreachable"

↳ After 5 minutes it replaces

## ★ Replication controller

- ↳ Responsible for monitoring status for replica set.
- ↳ Ensures that desired no. of pods are available at the required time.

★ Cronjob, deployment controller, Persistent Volume Protection, Binder, etc. [ALL included in Kube controller Manager]

## ★ Kube scheduler

- ↳ responsible for scheduling the pods on the nodes
- ↳ it just decides which pod to place on which node based on the CPU, RAM, resources on the Node
- ↳ Kubelet places the nodes after scheduler decides.
- ↳ Right container / Pod is sent to right ship / Node.

## ★ Kubelet

- ↳ is on the worker node and registers the Node with the Pod.

↳ Monitors the status of Pods and reports to the Kube api server.

↳ Need to install Kubelet on worker nodes.

## ★ Kube Proxy

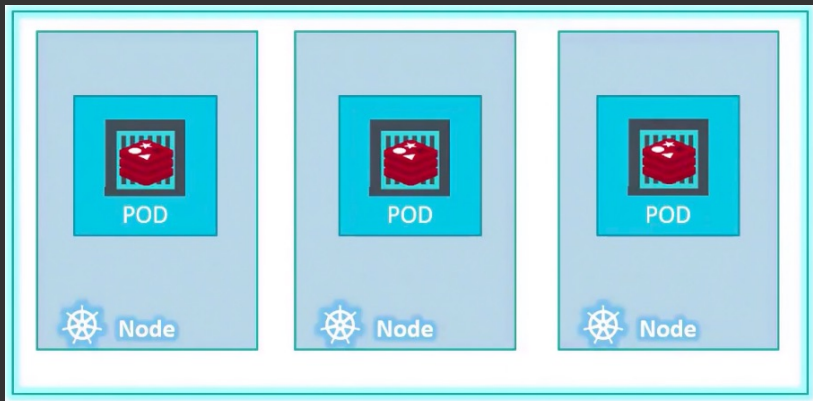
↳ Pod network allows to connect / communicate Pods for each other by Pod IP.

↳ Kube Proxy runs on each node, using IP Tables rules so that any service can connect to Pods from outside.



## ★ Pods

↳ Kubernetes doesn't deploy containers directly, it is encapsulated in Pods.



↳ Smallest object you can create in Kubernetes.

↳ New Pod is created in case we need to scale our application.

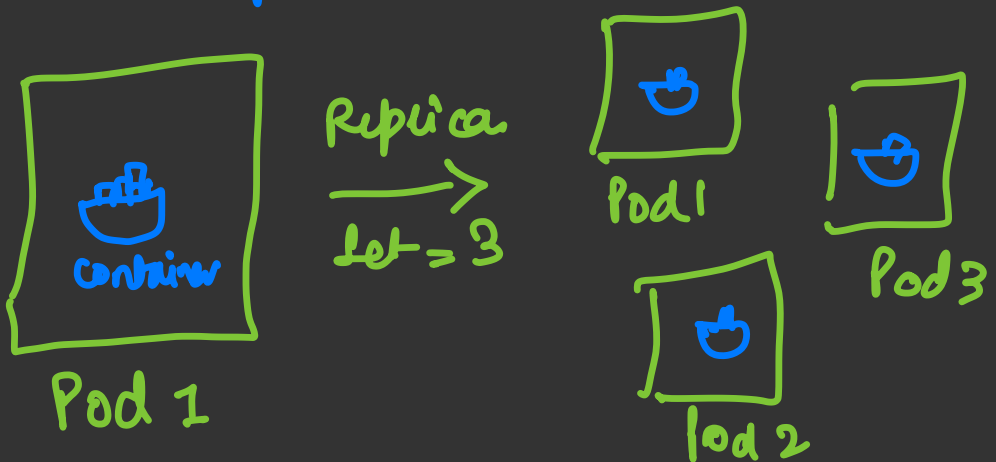
↳ New Node is added to cluster in case further scaling is needed.

↳ We can have multi-container Pods as well as a side-car container or Helper container.

## ★ Replica Set.

↳ To prevent users from losing access to the app, replication controller gives high availability.

↳ Helps in load Balancing and Scaling.

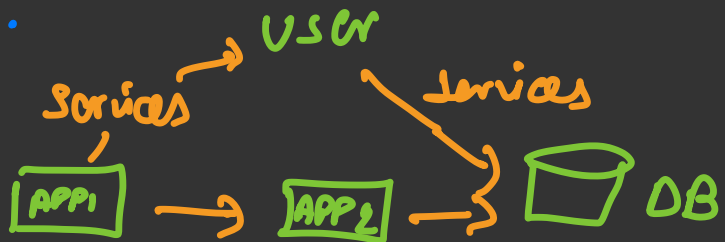


## ★ Deployment

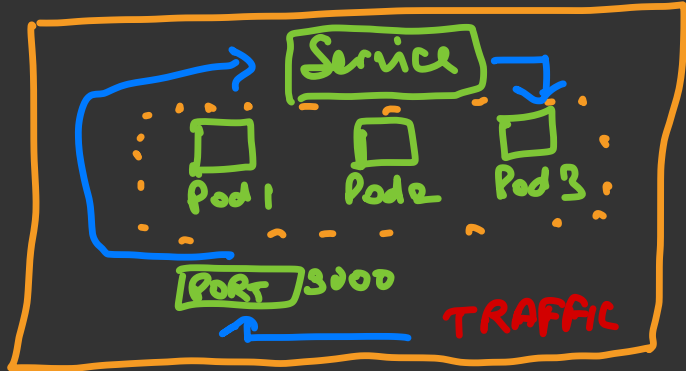
- ↳ Pods deploy single instances of an application
- ↳ Deployment allows to update the Pods infrastructure with Replica, rolling updates, etc.

## ★ Services

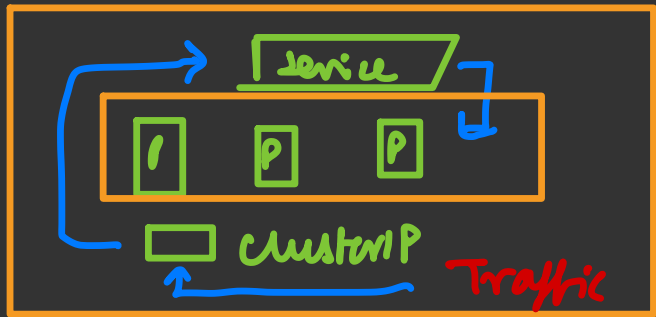
- ↳ Helps us connect our applications with other applications / databases etc.



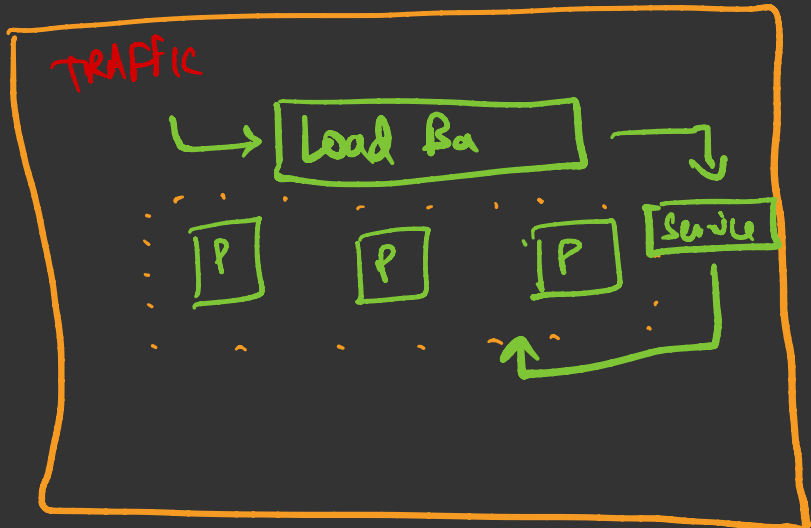
## ① Node Port



## ② Cluster IP



## ③ Load Balancer

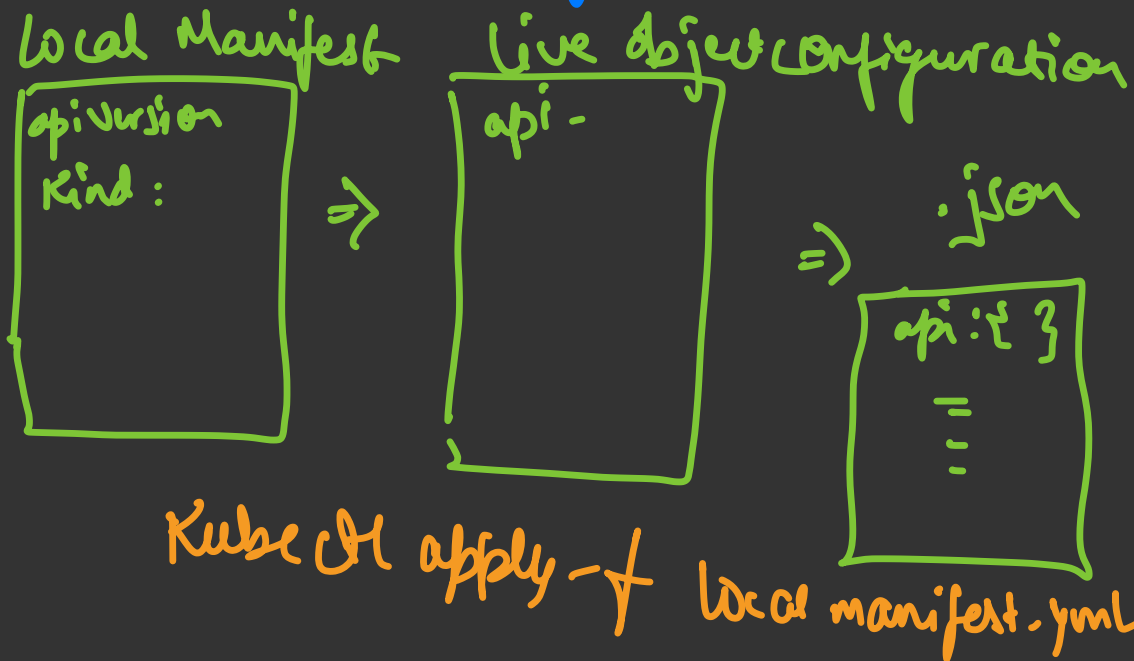


# \* Kubectl

↳ A command line tool used to communicate with a Kubernetes cluster's control plane.

↳ kubectl apply

↳ creates the live object for the configuration



# ★ Namespaces

↳ used to organise your K8s cluster into smaller cluster

↳ Pods, deployments can run in their own namespaces in an isolated way

① Default

② Kube-system

③ Kube-public

④ Kube-node-lease.

`kubectl create namespace <name>`