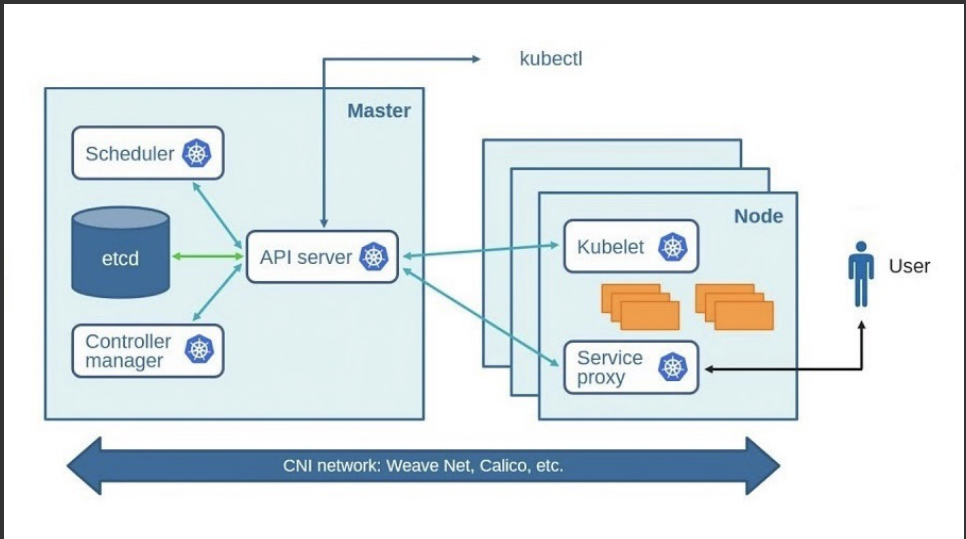TRAIN WITH SHUBHAM

Kubernetes (CKA)

* Cluster Architecture
* API Primitives
* Services

* Cluster Architecture

* Container Runtime Interface (CRI)

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

→ docker shim is the services
  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 &
  alternative to containerd.

⭐ ETCD

↳ contains all the information related to Nodes, Pods, configs, secrets, accounts, Roles, Bindings, etc.

↳ Key value datastore

↳ ETCD CTL is a cli tool used to interact with ETCD server

⭐ Kube-apiserver

↳ It is used to authenticate user, validate request, 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 Kube api server and manages nodes. (Every 5 seconds)

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

↳ After 5 minutes it repleaces

* 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
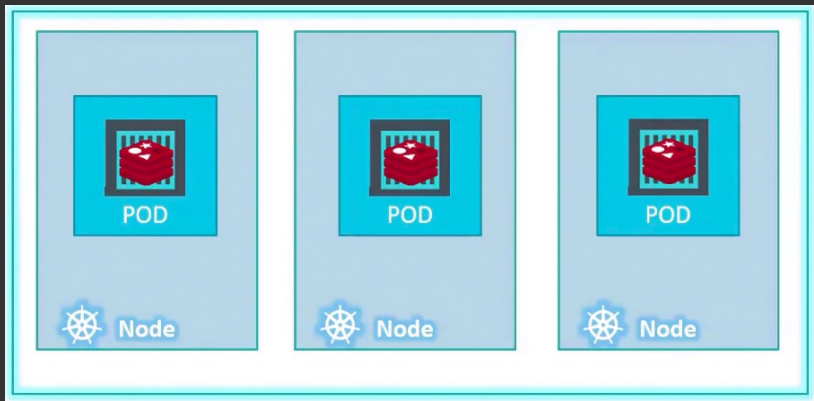on worker Nodes.

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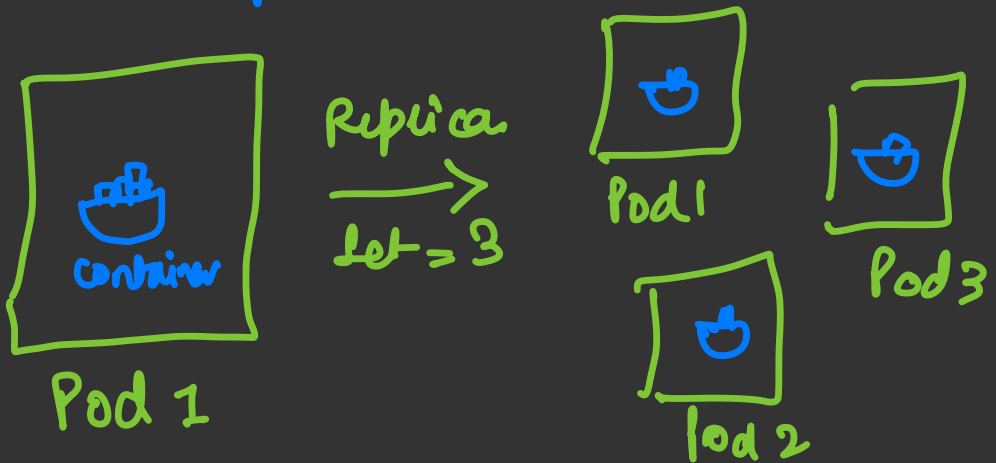
↳ New Node is added to cluster
   in case further scaling is needed

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

⭐ Replica Sets.

↳ To prevent users from losing access to the app, replication controller gives high availab--ilities

↳ Helps in load Balancing and Scaling.

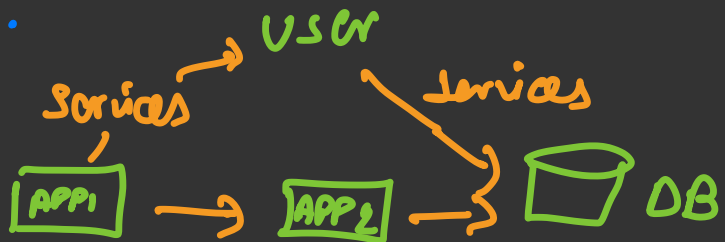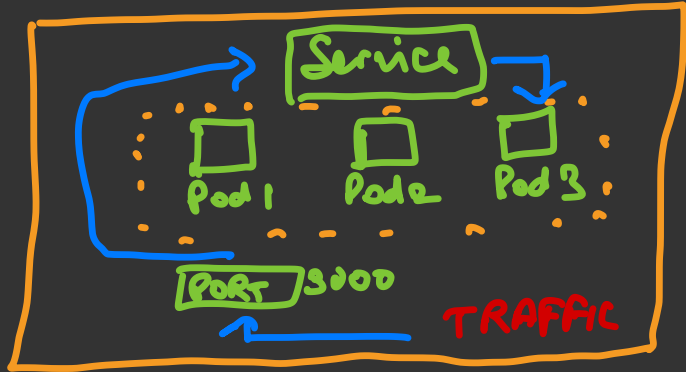Pod 1 → Replica Set = 3 → Pod1 Pod3 Pod 2

container

Pod 1

✱ Deployment

↳ Pods deploy single instance
of an application

↳ Deployment allows to
update the Pods infrastruc-
-ture with Replicas, rolling
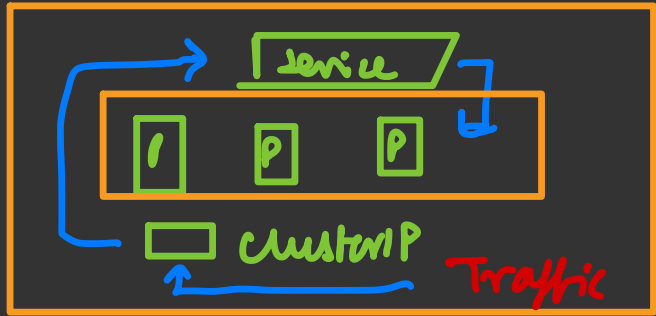updates, etc.

✱ Services

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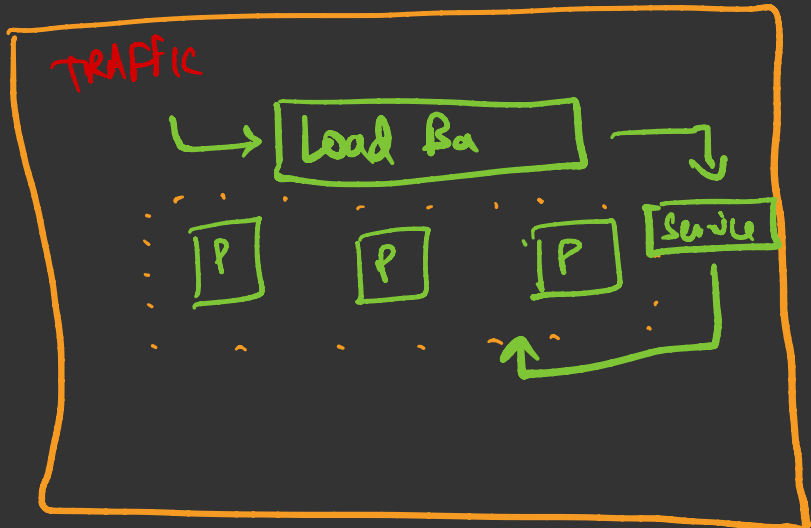applications / databases
etc.

User

Services   Services

[APP1] → [APP2] → ⊐ DB

① Node Port

Service

Pod 1   Pod 2   Pod 3

PORT 3000

TRAFFIC

② Cluster IP

Service

1   P   P

ClusterIP

Traffic

③ Load Balancer

TRAFFIC

Load Ba

P   P   P   Service

Kubectl

⤷ A command line tool used
to communicate with a
Kubernets cluster's control
plane.

⤷ kubectl apply

⤷ creates the live object for
the configuration

Local Manifest   Live object configuration

```
apiversion
Kind :
```
⇒
```
api -
```
⇒ .json
```
api : { }

≡
≡
≡
```

Kubectl apply -f local manifest.yml