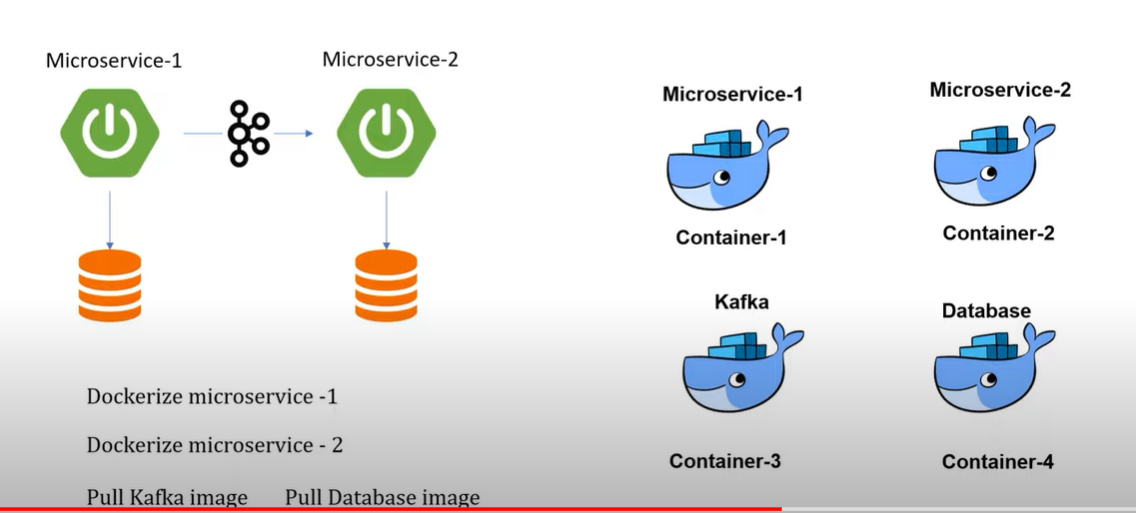
**Kubernetes :**

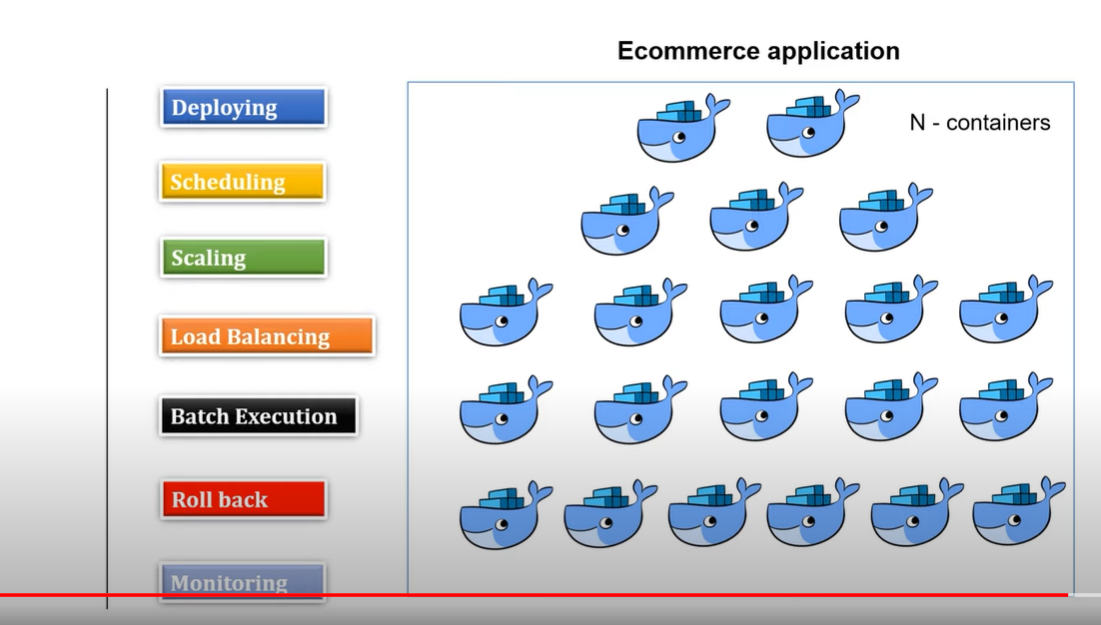
It is an open source container orchestration engine or container management tool, it automates deploying , scaling and managing containerized application.

It is also known as k8s because there are 8 characters between k and s in the word Kubernetes.

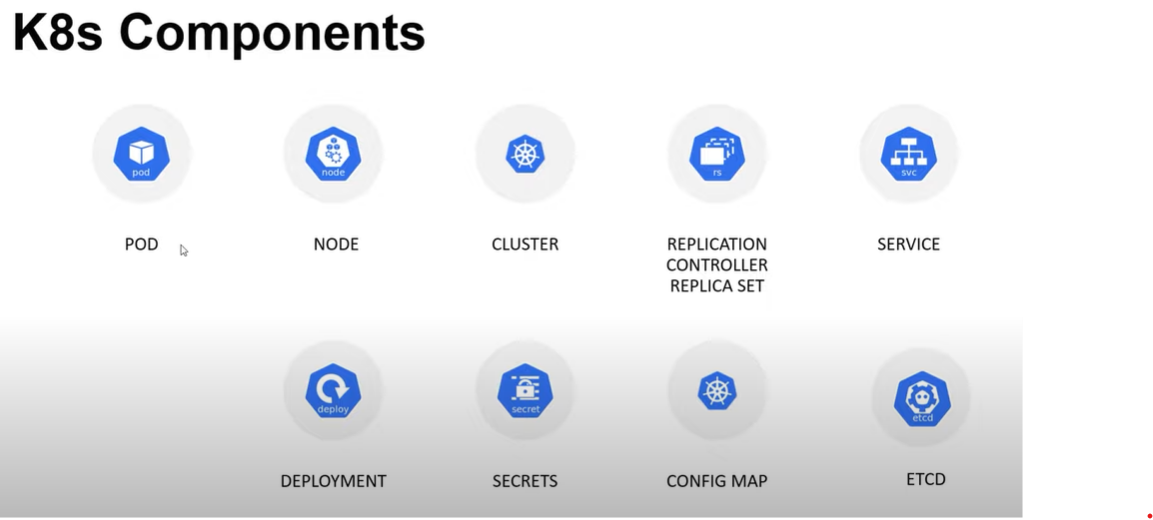
It is maintained by Cloud Native Computing Foundation (CNCF) and it is developed with Go Lang.

Management i.e Deploying, Scheduling, Scaling and Load Balancing





**Components of Kubernetes :**



Pod :

A pod is a collection of containers and its storage inside a node of Kubernetes Cluster. It is possible to create a pod with multiple containers inside it.

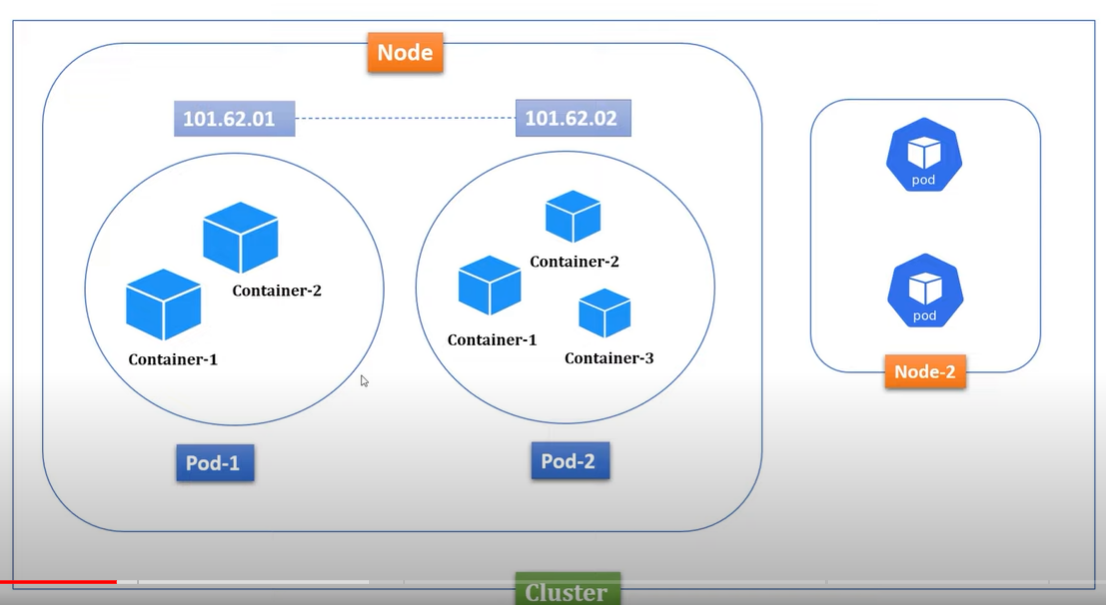
Node :

A node is a working machine in a Kubernetes Cluster which is also known as a minion. They are working units, can be physical, VM or a cloud instance.

Each node has all the required configuration required to run a pod such as the proxy service and kubelet service along with Docker, which is used to run the docker containers on the pod created on the node.

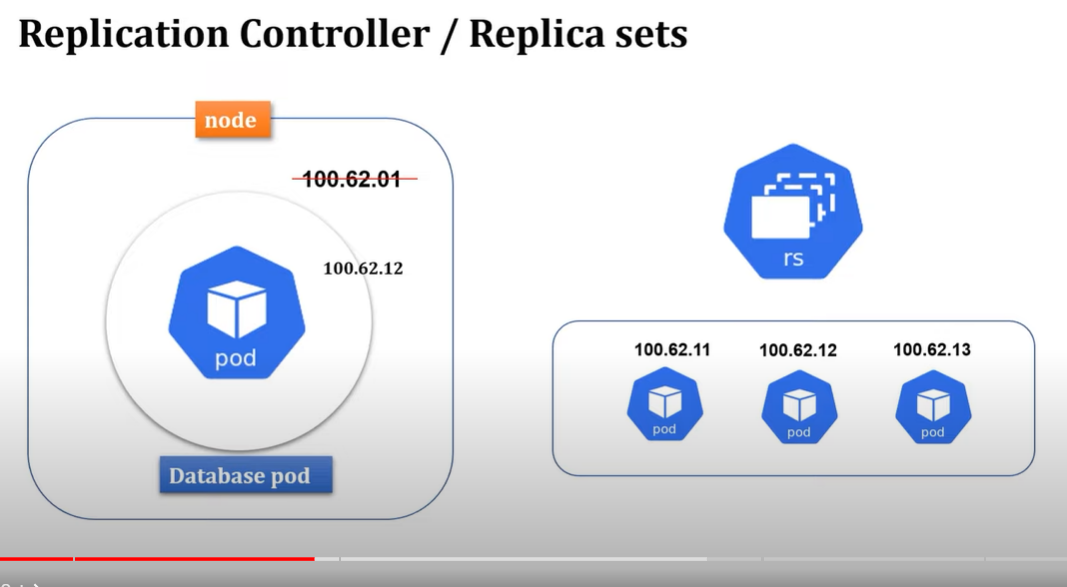
Cluster :

A cluster is a combination of pods and nods.



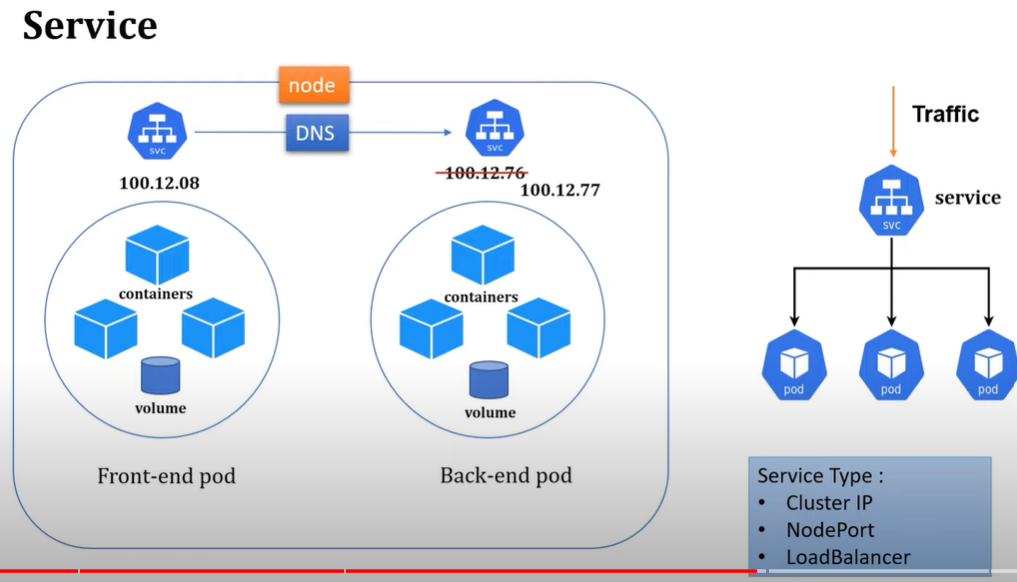
Replication Controller :

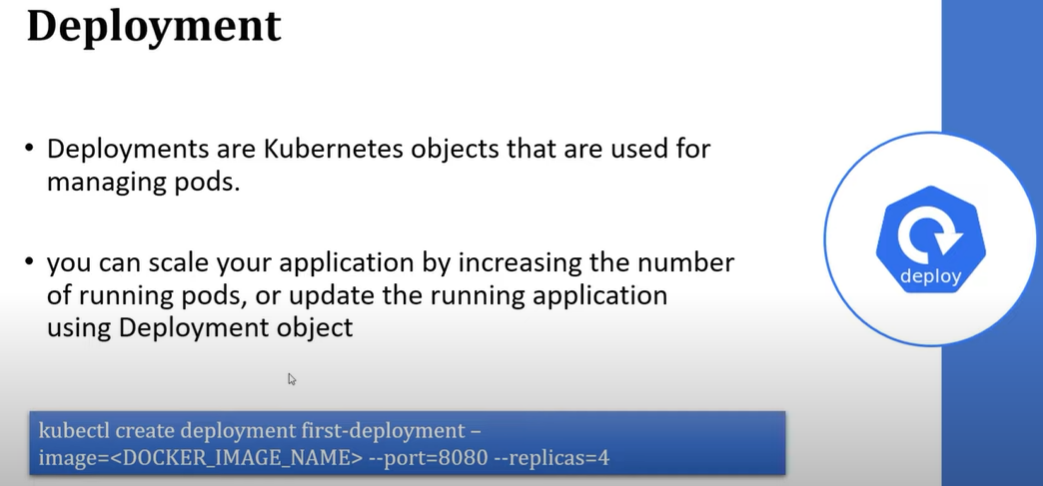
It is one of the key feature of the Kubernetes, which is responsible for managing the pod life cycle. It is responsible for making sure that the specified no of pod replicas are running at any point of time. It has the capacity to bring up or down the specified no of pods. It is the best practice to use the replication controller to manage the pod life cycle rather than creating a pod again and again.



Service :

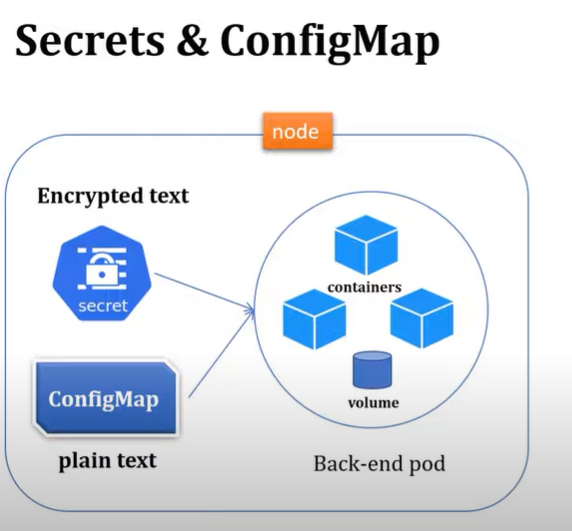
A service can be defined as a logical set of pods. It can be defined as an abstraction on the top of the pod which provides a single IP address and DNS Name by which pods can be accessed. With service, it is very easy to manage load balancing configuration. It helps pods to scale very easily.





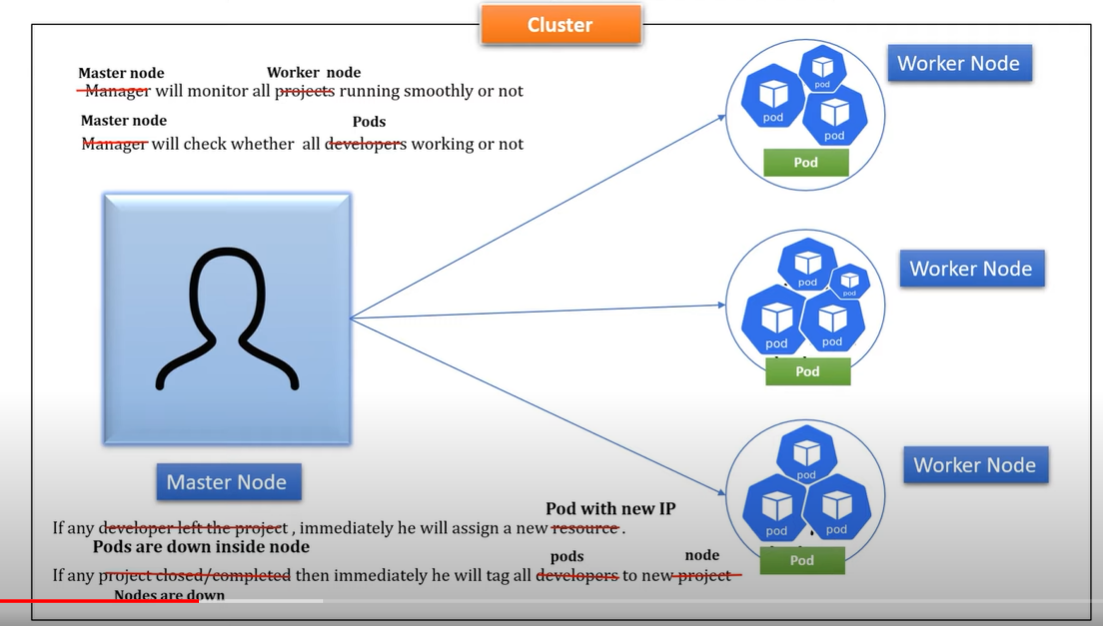
Secrets and ConfigMap :

It is basically used to store confidential details outside of the containers. Secrets store details in encrypted format whereas configMap in plain text.





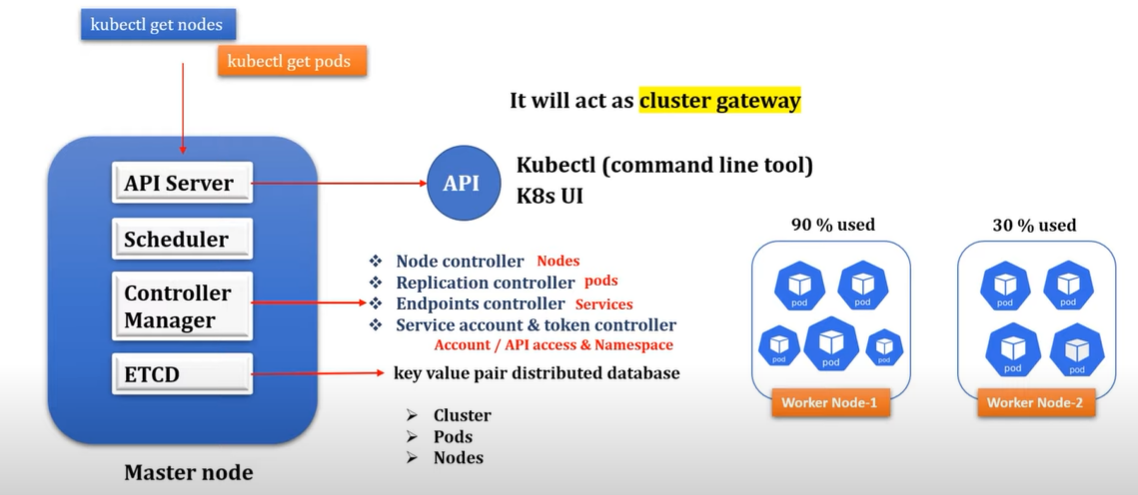
Master Node :

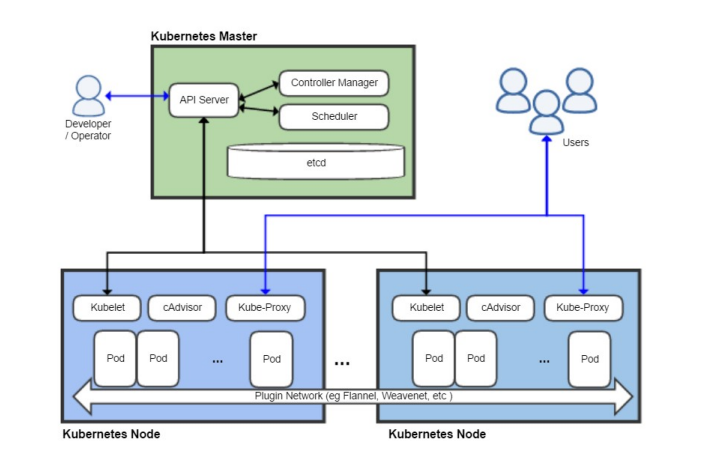


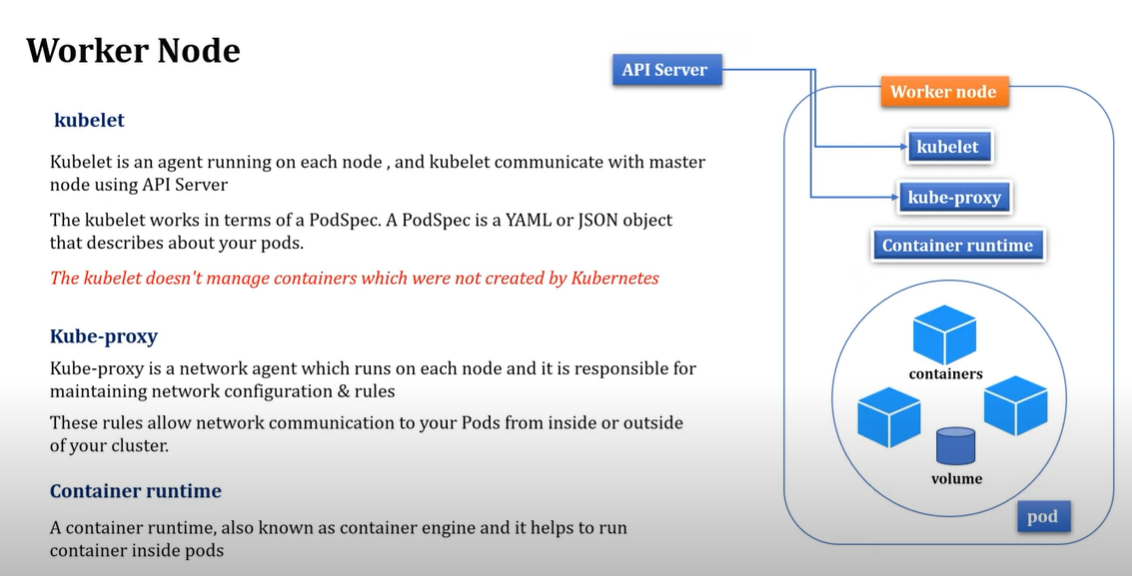
Kubernetes is really a master slave type of architecture with certain components (master) calling the shots in the cluster and the other components (node components) executing application workloads as decided in the master components.

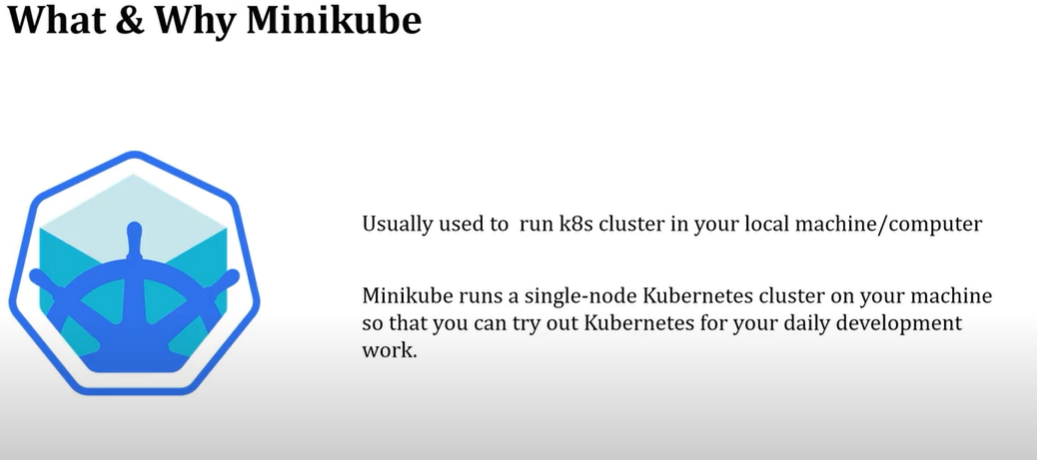
The master components manages the state of the cluster.

* API Server :- A REST Api supporting basic CRUD operations on API Objects such as (pods , deployments and services.) The API Server in itself is stateless. Instead it uses a distributed key value storage system (etcd) as its backend for storing all cluster state. This is the endpoint that a cluster administrator communicates with, for example kubectl.
* Controller Manager : It runs the control loops to watch the desired state in the api server and attempt to move the actual state towards the desired state.
* Scheduler : It takes care of the pod placement across the set of available nodes, striving to balance the resource consumption to not place excessive load on any cluster node.









Kubectl : It is a command line tool. With the help of this, we can connect with the k8s cluster for the daily development from our local machine.

\*\* Kubernetes pulls the docker image when the deployment is created and pods are created , which starts a single container and executes the particular image.

Commands :

To start minikube with docker : minikube start --driver=docker

To check status of minikube : minikube status

To get cluster info : kubectl cluster-info

To get node details: kubectl get node

To allow Kubernetes to read your docker : minikube docker-env

To create deployment : kubectl create deployment springboot-k8s-demo --image=springboot-k8s-demo --port=8081

To verify deployment : kubectl get deployment

Description of deployment : kubectl describe deployment springboot-k8s-demo

To getpods details : kubectl get pods

To delete deployment : kubectl delete deploy springboot-k8s-demo

To check pod details : kubectl describe pods <pod-name>