================

K8s Architecture

================

K8s will follow the cluster architecture

cluster means group of servers will be available

In k8s cluster we will have a Master node and worker node

Master node is also called control plane

master node will manage the entire cluster and worker node will create our containers

========================

k8s cluster components

=========================

Control plane

- api server

- schedular

- controller manager

- ETCD

Worker node

- kubelet

- kube proxy

- Docker runtime

- POD

- container

=> To deploy our application using k8s we need to communicate with control node.

=> We will use KUBECTL (CLI) to communicate with control plane.

=> API Server will receive the request given by kubectl and it will store that request in ETCD with pending status.

=> ETCD is an internal database of k8s cluster.

=> Schedular will identify pending requests available in ETCD and it will identify worker node to schedule the task.

Note: Schedular will identify worker node using kubelet.

=> Kubelet is called as Node Agent. It will maintain all the worker node information.

=> Kube Proxy will provide network for the cluster communication.

=> Controller Manager will verify all the tasks are working as expected or not.

=> In Worker Node, Docker Engine will be available to run docker container.

=> In K8s, container will be created inside POD.

=> POD is a smallest building block that we can create in k8s cluster.

=> Inside POD, docker container will be created.

Note: In K8s, everything will be represented as POD only.

If i want to create pod inside the worker node we are going to use .yml file

============

What is pod

============

Pod is the smallest building block in a K8s cluster

Application will be deployed as pod in K8s

we can create multiple pods for one application

To create a POD we will use yml file (manifest.yml)

In pod manifest.yml we will configure our docker image

If a pod is damaged/crashed/deleted the K8s will create a new pod. This concept is called self-healing

If our application is running in multiple pods then k8s will distribute the load to all the running pods. This is called load balancer

Pods count can be increased or decreased based on the load, which is called scalability

=================

k8s cluster setup

=================

1) Mini Kube => Single Node cluster => Only for practice purpose

2) Kubeadm Cluster => Self-Managed Cluster => We are responsible for everything

3) Provider Managed Cluster => Ready Made Cluster => Provider will take care of everything

Ex : AWS EKS, Azure AKS, GCP GKE etc...

Note: Provider Managed Clusters are chargeable.