1. **What is Kubernetes?**

=>

- kubernate is an open source orchestration platform

- orchestration platform means manage docker containers

- Kubernate provides a framework for managing the complex task of deploying, scaling, and operating applications in containers.

1. **What are K8s?**

* K8s is another term for Kubernetes

1. **What is orchestration when it comes to software and DevOps?**

* orchestration platform means manage docker containers
* for example, you have six or seven microservices for an application to run. If you place them in separate containers, this would inevitably create obstacles for communication. Orchestration would help in such a situation by enabling all services in individual containers to work seamlessly to accomplish a single goal.

1. **How are Kubernetes and docker are related?**

* [Docker is an open-source](https://www.simplilearn.com/tutorials/docker-tutorial) platform used to handle software development. Its main benefit is that it packages the settings and dependencies that the software/application needs to run into a container, which allows for portability and several other advantages. Kubernetes allows for the manual linking and orchestration of several containers, running on multiple hosts that have been created using Docker.

1. **What is the difference between deploying applications on hosts and containers?**

* Deploying on \*\*hosts\*\* means running apps directly on the machine's operating system, sharing resources with other apps. In \*\*containers\*\*, apps run in isolated environments with their own dependencies, making them more lightweight and portable. Containers are easier to scale and manage across different environments than host-based deployments.

1. **What are the features of Kubernetes?**

* orchestration
* scalabilty - according to request manage docker containers
* load balancing
* self healing - one of the containers damage / not working it manage by other one

1. **What are the main components of Kubernetes architecture?**

* It is based on Cluster architecture means of group of server.
* Control Plane /Master Node

task give to control plane and manage everything in kubernate

-API server

-Scheduler

-controller - manager

-etcd

* Worker Node /slave

-pod(it is process)

-containers

-docker

-kubelet

-kubeproxy

* KubeCTL - command line utility to communicate with control plane

=>K8S Architecture Components

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

- API server will receive incoming requests and it will store into ETCD

-ETCD is K8S cluster database

-Scheduler check pending task in etcd and then schedule all pending task to worker nodes

-kubelet personal assitance / worker node agent

-scheduler will get available worker nodes info by using kubelet.

-kube proxy provide network for cluster communication

-POD is a smallest building block that we run in kubernates cluster

-POD represents runtime instance of our application

-POD deploy the application

-Container prsent inside POD and docker engine to run docker container inside POD

Note : In k8s, our project will be executed as POD . Inside POD containers will be created

-Controller Manager will monitor all k8s resources functionality.

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

=> K8S Cluster Setup

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

1.Self Managed K8S Cluster

-Mini Kube (Single Node Cluster)

-Kubeadm (Multi Node Cluster)

2. Managed K8S cluster

-AWS EKS(Elastic K8S Service)

-Azure AKS

-GCP GKE (Google)

-IBM IKE

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

=> AWS EKS

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

==> what is pod?

- POD is the smallest building block that we can deploy in k8s cluster

- our application will be deployed in k8s cluster as a pod only

- for one application we can create multiple POD replicas for high availability

-for every POD one IP address will be generated

-if POD got damaged/crashed then k8s will replace it (Self -healing)

- to create PODS we will use MANIFEST YML files

NOte : By default PODS are accessible only with in the cluster (we cant access outside)

-to expose PODS for outside access we need to use K8S services concept.

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

=>What is Service in K8S?

-k8s service is used to expose PODS

-3 types

1.Cluster IP (to access PODS with in the cluster )

2.Node Port (to access PODS using NODE Public IP)

3.Load Balancer (to distribute the traffic to POD replicas)

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

==> K8S Manifest YML syntax

apiVersion: apps/v1

kind: Deployment

metadata:

name: javawebappdeployment

spec:

replicas:2

stategy:

type: Rollingupdate

selector:

matchLabels;

app: javawebap

template:

metadata:

name:javawebapppod

labels:

app: javawebapp

spec:

containers:

name : javawebappcontainer

image : ashokit/javawebapp

ports:

-containerPort:8080

apiVersion: v1

kind: Service

metadata:

name: javawebappsvc

spec:

type: LoadBalancer

selecto:

app:javawebapp

ports:

port:80

containerPort:8080

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

#check pod running

-kubectl get pods

#check pod running in which worker node

-kubectl get pods -o wide

#check services created

-kubectl get svc

#check deployments created

-kubectl get deployment

# Execute k8s manifest yml

kubectl apply -f <yml file name>