

## =====

### Kubernetes

## =====

=> Open source orchestration platform

=> K8S is used to manage our containers

=> K8S provided a framework to handle containers related tasks  
(deployment, scaling, load balancing etc....)

=> K8S developed by Google and donated to CNCF

## =====

### K8S Advantages

## =====

1) Container Orchestration

2) Auto Scaling

3) Self Healing

4) Load Balancing

## =====

### K8S Architecture

## =====

=> K8S will follow cluster architecture (group of servers)

=> Master Node / Control Plane

=> Worker Nodes

=> K8S control plane will contain below components

- 1) API Server
- 2) Scheduler
- 3) Controller Manager
- 4) ETCD

=> K8S worker node will contain below components

- 1) POD
- 2) Containers
- 3) Docker Engine
- 4) Kubelet
- 5) Kube Proxy

=> To communicate with K8S control plane we have 2 options

- 1) UI (Web Dashboard)
- 2) Kubectl (CLI)

## =====

### K8S Architecture Components

## =====

=> API Server will receive incoming requests and it will store into ETCD.

=> ETCD is k8s cluster database

=> Scheduler will check pending tasks in ETCD and it will schedule those task in worker nodes.

=> Scheduler will get available workers nodes information by using kubelet.

=> Kubelet is called as Worker node agent.

=> Kube-Proxy provides network for cluster communication

=> POD is a smallet building block that we run in kubernetes cluster. POD represents runtime instance of our application.

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

=> Controller-Manager will monitor all k8s resources functionality.

### EKS Cluster Setup DOC : <https://github.com/ashokitschool/DevOps-Documents/blob/main/EKS-Setup.md>

=====  
Kubernete Cluster Core Components  
=====

#### 1) Control Plane

- 1.1) API Server
- 1.2) Scheduler
- 1.3) Controller - Manager
- 1.4) ETCD

#### 2) Worker Node

- 2.1) Kubelet
- 2.2) Kube-Proxy
- 2.3) Docker Engine
- 2.4) Container
- 2.5) POD

=====  
What is POD  
=====

=> POD is a 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 can't 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

=> We have 3 types of services

- 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
  strategy:
    type: RollingUpdate
  selector:
    matchLabels:
      app: javawebapp
  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
  selector:
    app: javawebapp
  ports:
    - port: 80
      containerPort: 8080
```

...

# check pods running

\$ kubectl get pods

# check pods 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>

# Check pod logs

```
$ kubectl logs <pod-name>
```

```
# Delete pod
```

```
$ kubectl delete pod <pod-name>
```

```
# delete resources we have created
```

```
$ kubectl delete all --all
```

```
=====
```

Docker Playlist : <https://www.youtube.com/watch?v=8dccz7ca4FM&list=PLpLBSl8eY8jQ0lp6FeOt0bvkf0ddJW6YM>

EFK Stack Setup in K8S : <https://www.youtube.com/watch?v=8MLcbbfEL1U>

HPA in K8S : <https://www.youtube.com/watch?v=c-tsJrcB50I>

```
=====
```