

Kubernetes Deployments & Cluster Architecture

What is Kubernetes?

Kubernetes (also called K8s) is an open-source platform used to automate the deployment, scaling, and management of containerized applications.

Part 1: Kubernetes Deployments

A Deployment in Kubernetes is a resource used to manage a set of identical Pods. It ensures the desired number of Pods are running at all times and allows for updates, rollbacks, and scaling.

Sample Deployment YAML

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
        - name: my-app
          image: nginx
          ports:
            - containerPort: 80
```

Key Commands

```
kubectl apply -f deployment.yaml      # Create deployment
kubectl get deployments                # View deployments
kubectl scale deployment my-app --replicas=5  # Scale
kubectl rollout undo deployment/my-app      # Rollback
```

Part 2: Kubernetes Cluster Architecture

A Kubernetes cluster is a set of machines (nodes) that run containerized applications. It consists of a Master Node (Control Plane) and multiple Worker Nodes.

Kubernetes Deployments & Cluster Architecture

Master Node Components

- kube-apiserver: Frontend for K8s cluster
- etcd: Key-value store
- kube-scheduler: Schedules Pods on nodes
- kube-controller-manager: Manages controllers like ReplicaSets

Worker Node Components

- kubelet: Talks to API server, runs containers
- kube-proxy: Manages network rules
- Container Runtime: Runs containers (e.g., containerd, Docker)

Summary for Students

- Deployment: Helps run and manage multiple instances of a pod
- Cluster: Group of nodes where applications are deployed
- Master Node: Brain of the cluster
- Worker Node: Where actual application pods run