

What is Kubernetes?

Kubernetes, often abbreviated as **K8s**, is an open-source **container orchestration platform** designed to automate the deployment, scaling, and management of containerized applications. Originally developed by Google, Kubernetes is now maintained by the **Cloud Native Computing Foundation (CNCF)**.

Key Features of Kubernetes

1. **Automated Deployment and Scaling:** Automates container deployment, scaling based on traffic or resource usage, and management.
2. **Self-Healing:** Restarts failed containers, replaces unresponsive nodes, and automatically kills unresponsive containers.
3. **Load Balancing and Service Discovery:** Distributes traffic across containers and provides internal and external IPs for service discovery.
4. **Storage Orchestration:** Automatically mounts storage systems like local storage, cloud providers, or network file systems.
5. **Rolling Updates and Rollbacks:** Updates applications incrementally while ensuring no downtime and provides easy rollback in case of failure.
6. **Configuration Management:** Manages application configurations through ConfigMaps and Secrets.

Core Components of Kubernetes

Component	Description
Cluster	A group of machines running Kubernetes to manage workloads.
Node	A single machine (physical or virtual) in a cluster that runs workloads.
Pod	The smallest deployable unit in Kubernetes; typically contains one or more containers.
Service	Provides stable access (network) to Pods and performs load balancing.
Deployment	Manages the lifecycle of Pods and ensures desired state is maintained.
Ingress	Manages external HTTP and HTTPS access to services within the cluster.
Namespace	Provides logical isolation of resources within a cluster.
ConfigMap	Stores configuration data in key-value pairs.
Secret	Stores sensitive data such as passwords or tokens securely.

Kubernetes Architecture

1. Master Node (Control Plane):

Manages the cluster and coordinates activities. Key components:

- **API Server:** Handles communication with users and other components.
- **Scheduler:** Assigns workloads to worker nodes.
- **Controller Manager:** Maintains cluster state and manages tasks like replication and node health.
- **etcd:** A key-value store to save cluster state data.

2. Worker Node:

Executes workloads assigned by the control plane. Key components:

- **Kubelet:** Ensures containers are running in Pods.
- **Kube-proxy:** Manages networking and forwards traffic to the correct Pod.
- **Container Runtime:** Runs containers (e.g., Docker, containerd, CRI-O).

Basic Kubernetes Commands

Cluster and Node Management

Command	Description
kubectl version	Displays the version of Kubernetes installed.
Kubectl api-resources	Displays resources list.
kubectl get nodes	Lists all nodes in the cluster.

Pod Management

Command	Description
kubectl get pods	Lists all Pods in the cluster.
kubectl describe pod <pod_name>	Displays detailed information about a specific Pod.
kubectl delete pod <pod_name>	Deletes a specific Pod.

Kubernetes Workflow

1. Write a Manifest File:

Define the desired state of your resources (e.g., Pods,) in YAML or JSON.

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
  - name: mycon
    image: httpd
    ports:
    - containerPort: 80
```

1. **Apply the Manifest File:**

Use the command `kubectl apply -f <file.yaml>` to deploy the configuration.

2. **Monitor Resources:**

Check the status using commands like `kubectl get pods`

Common Kubernetes Use Cases

- Microservices architecture.
- Continuous Deployment (CD) and Continuous Delivery (CI/CD) pipelines.
- Dynamic scaling of applications.
- Managing hybrid cloud and multi-cloud environments.