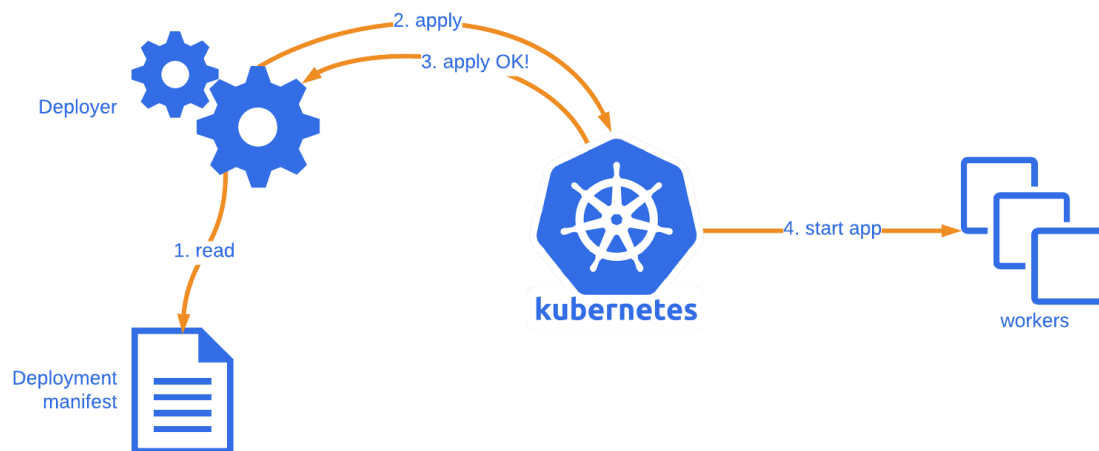


Kubernetes (K8s) Basic Documentation



What is Kubernetes?

Kubernetes (K8s) is an **open-source container orchestration platform** that automates the deployment, scaling, and management of containerized applications. Developed by Google and now maintained by the **Cloud Native Computing Foundation (CNCF)**, Kubernetes helps ensure high availability and fault tolerance of your apps in production environments.

History of Kubernetes

- **2014:** Google open-sourced Kubernetes, based on its internal container orchestration system called **Borg**.
 - **2015:** Donated to CNCF (Cloud Native Computing Foundation).
 - **Now:** It is the de facto standard for container orchestration across cloud-native infrastructures.
-

How Kubernetes Works

Kubernetes runs applications in **containers** across a **cluster** of machines. It provides:

- **Self-healing:** Replaces and reschedules containers when nodes die.
- **Load balancing:** Distributes traffic evenly across apps.

- **Automated rollouts/rollbacks:** Smooth version upgrades.
 - **Storage orchestration:** Automatically mounts persistent storage.
-

Kubernetes Cluster

A Kubernetes cluster is made up of:

- **Control Plane:** The brain of the cluster that manages the system.
- **Worker Nodes:** Where actual application workloads (containers) run.

Each cluster represents a **desired state** for applications, and K8s ensures this state is maintained continuously.

Control Plane Components

These components manage the cluster:

1. API Server

- Acts as the **front-end** for Kubernetes.
- All **kubectl commands** interact with it.
- It validates and configures the data.

2. etcd

- A **key-value store** that stores all cluster data (configuration, state, secrets).
- Think of it as the **database of Kubernetes**.

3. Controller Manager

- Watches the desired state and takes actions to ensure the current state matches.
- Examples:
 - Node controller (checks node health).
 - ReplicaSet controller (ensures pod count).

4. Scheduler

- Assigns pods to worker nodes based on resource availability, policies, etc.
-

Worker Node Components

Each node runs the components necessary to manage individual containers.

1. kubelet

- Agent running on each node.
- Ensures containers are running as expected.

2. kube-proxy

- Manages **networking** on the node.
- Handles routing and load balancing.

3. Container Runtime

- Responsible for **running containers** (Docker, containerd, CRI-O, etc.).
-

Pods

- A **pod** is the smallest deployable unit in K8s.
 - Each pod contains one or more containers.
 - Pods in the same deployment share **network and storage**.
-

Thumb Rules (Logical Concepts)

These are important conceptual truths every Kubernetes user should remember:

1. Self-Healing (ReplicaSet/Deployment Behavior)

"If a pod is deleted manually, Kubernetes will recreate it to match the desired state."

Why?

- A **ReplicaSet** always ensures the number of running pods matches its configuration.
- A **Deployment** manages ReplicaSets and handles rollbacks and updates.

2. Stateless vs. Stateful

Stateless apps (like web servers) are easier to scale than stateful apps (like databases).

- Use **Deployments** for stateless apps.
- Use **StatefulSets** for apps requiring persistent identity/storage.

3. Never edit pods directly

Always edit higher-level objects like **Deployments**, **ReplicaSets**, or **StatefulSets**.

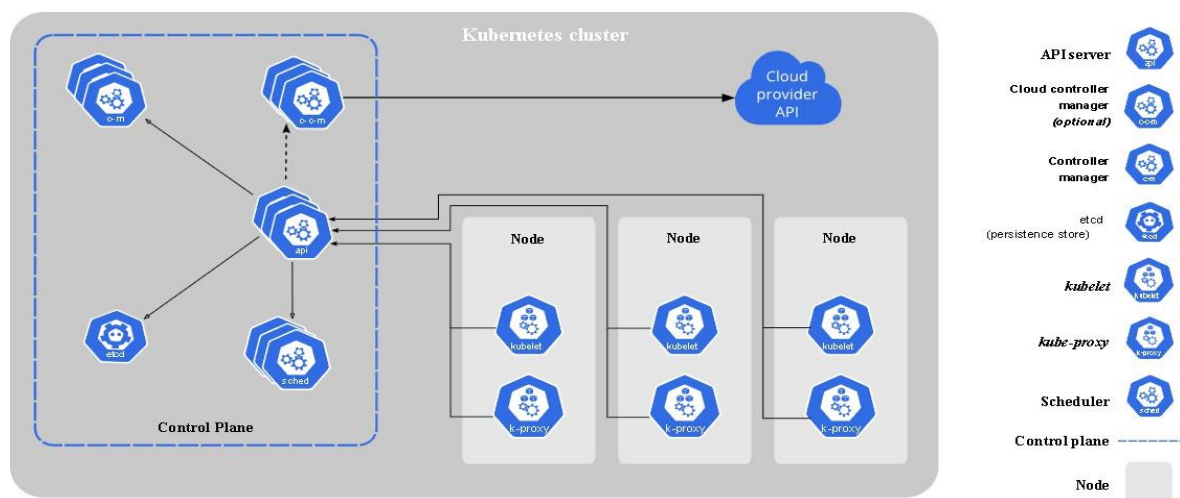
- Pods are ephemeral; changes to them directly will be lost upon restart or recreation.

4. Desired State vs. Current State

The entire architecture of K8s revolves around this principle.

- K8s continuously compares what is running vs. what *should* be running.
- Controllers take actions to close any gap.

Kubernetes Architecture (Diagram Description)



Summary

Component	Role
API Server	Entry point to cluster via kubectl
etcd	Key-value storage for state
Scheduler	Assigns pods to nodes
Controller Manager	Ensures current state = desired state
kubelet	Runs containers and monitors pods
kube-proxy	Manages networking & service routing
Container Runtime	Executes containers
Pod	Smallest deployable unit (contains containers)

Thank you.....