

What is Kubernetes?

Kubernetes (K8s) is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. It helps in managing containers (such as Docker containers) across clusters of machines.

How CI/CD and Docker Fit Together:

1. **Code Development:**
 - Developers write code and store it in a source code repository like GitHub or GitLab.
 2. **CI/CD Pipeline:**
 - The code is integrated into a CI/CD pipeline (e.g., Jenkins).
 - Maven is used to build the code and generate artifacts.
 3. **Deployment:**
 - The artifact is deployed into a testing or lab environment.
 - Errors may occur due to dependencies or environmental differences.
 4. **Docker:**
 - Docker is used to bundle the code and all its dependencies into a Docker image.
 - The image is stored in a Docker registry (e.g., Docker Hub).
 - This allows you to pull and deploy the containerized application anywhere.
-

Why Kubernetes?

While Docker solves the problem of packaging and running applications, Kubernetes provides solutions for managing containers at scale:

- **Limitations of Docker:**
 - Docker alone does not provide features like auto-scaling, self-healing, or handling the complexities of running containers at scale in production environments.
 - **Why Kubernetes (K8s):**
 - Kubernetes provides auto-healing and auto-scaling based on application load.
 - It ensures zero downtime during deployment, which is critical for most companies.
 - Kubernetes uses **Pods**: A Pod is the smallest deployable unit in Kubernetes and can hold one or more containers.
-

Kubernetes Architecture:

1. **Nodes:**
 - Kubernetes clusters consist of two types of nodes:
 - **Master Node:** Controls and manages the Kubernetes cluster.
 - **Worker Node:** Executes the tasks assigned by the master node (runs containers).
 2. **Components of Kubernetes:**
 - **Kubelet:** Responsible for ensuring that containers are running in the Pods.
 - **Kube-proxy:** Manages networking and communication between Pods and services.
 - **API Server:** The heart of Kubernetes that interacts with the Kubernetes cluster via `kubectl` commands.
 - **Scheduler:** Assigns Pods to nodes based on resource requirements.
 - **Controller Manager:** Manages controllers like ReplicaSets, ensuring that the desired state of the system is maintained.
 - **ETCD:** A distributed key-value store that stores cluster configuration data and state information.
 3. **Scaling:**
 - Kubernetes automatically manages scaling using **ReplicaSets**. You can specify the desired number of replicas (containers), and Kubernetes ensures they are running as required.
 - **Auto-scaling** adjusts the number of Pods based on resource usage or custom metrics.
-

Kubernetes Services in Cloud:

- Kubernetes can be managed through cloud services like **AWS EKS** (Elastic Kubernetes Service), which simplifies the setup and management of Kubernetes clusters.
 - To interact with Kubernetes, you need certain tools installed:
 - **AWS CLI:** AWS Command Line Interface for managing AWS resources.
 - **kubectl:** Command-line tool to interact with the Kubernetes cluster.
 - **eksctl:** A tool for creating and managing AWS EKS clusters.
-

Tools Needed for Kubernetes Setup:

1. **Visual Studio Code** (for code writing and editing).
2. **AWS CLI** (for interacting with AWS resources).
3. **kubectl** (for managing Kubernetes clusters).
4. **eksctl** (for managing AWS EKS).

Before setting up Kubernetes, ensure these tools are installed on your system.

