**KUBERNATES**

1. **What is Kubernates –**  Kubernetes is an open-source platform that automates the deployment, scaling, and management of containerized applications—apps that run inside lightweight, portable environments called containers
2. **Advantages –**

**1. Scalability**  
Kubernetes can automatically scale applications based on traffic or resource usage, making it ideal for dynamic workloads.

**2. High Availability**  
It ensures your application stays online by restarting failed containers and distributing workloads across healthy nodes.

**3. Portability**  
You can run Kubernetes on any infrastructure—cloud, on-premises, or hybrid—without being locked into a specific vendor.

**4. Efficient Resource Management**  
Kubernetes schedules containers intelligently to optimize CPU and memory usage across your cluster.

1. **Disadvantages –**

**1.Complexity**  
Kubernetes has a steep learning curve. Understanding its architecture and configuring it properly takes time and expertise.

**2. Operational Overhead**  
Managing a Kubernetes cluster requires ongoing monitoring, upgrades, and troubleshooting, which can be resource-intensive.

**3. Cost**  
While Kubernetes itself is free, the infrastructure, tooling, and skilled personnel needed to run it effectively can be expensive.

1. **Debugging Challenges**  
   Diagnosing issues in a distributed system can be difficult, especially when multiple containers and services interact.
2. **Architecture of kubernate –**

Kubernetes Cluster

A Kubernetes cluster is made up of two main parts: the **Control Plane** and the **Worker Nodes**.

* The **Control Plane** is responsible for managing the cluster. It makes decisions about scheduling, scaling, and responding to cluster events.
* The **Worker Nodes** are the machines (physical or virtual) that run your application containers.

Control Plane Components

* **kube-apiserver**: This is the front-end of the Kubernetes control plane. It exposes the Kubernetes API and is the main point of interaction for users and components.
* **etcd**: A distributed key-value store that holds all cluster data, including configuration and state.
* **kube-scheduler**: Assigns newly created pods to suitable nodes based on resource availability and other constraints.
* **kube-controller-manager**: Runs various controllers that handle routine tasks like replicating pods, monitoring node health, and managing endpoints.

1. **What is POD –**

In Kubernetes, a **Pod** is the smallest and most basic deployable unit. Think of it as a wrapper around one or more containers that share the same environment.

Here’s what makes a Pod unique:

* **Single or Multiple Containers**: Most Pods contain just one container, but they can host multiple containers that are tightly coupled and need to share resources.
* **Shared Resources**: Containers in a Pod share the same network IP, storage volumes, and configuration. This allows them to communicate easily and coordinate tasks.
* **Ephemeral Nature**: Pods are designed to be temporary. If a Pod fails, Kubernetes can automatically replace it with a new one to maintain the desired state.
* **Logical Host**: A Pod acts like a logical host for its containers, similar to how multiple apps might run on the same physical machine.