📘 Kubernetes Architecture – Beginner Friendly Detailed Notes

🚀 What is Kubernetes (K8s)?

Kubernetes is a container orchestration platform. It automates the deployment, scaling, and management of containerized applications across a cluster of machines.

🧱 High-Level Architecture

Kubernetes has two main layers:

1. **Control Plane (Master Node)** – manages the cluster
2. **Worker Nodes** – runs application workloads

🟩 1. Control Plane Components (Master Node)

The control plane makes global decisions about the cluster (e.g., scheduling), and detects and responds to cluster events (e.g., restarting failed pods).

🧩 Components:

📌 a. kube-apiserver

- The front-end of the Kubernetes control plane.
- Accepts REST calls from CLI (kubectl) or UI.
- Authenticates and validates requests.
- Communicates with etcd.

📌 b. etcd

- A distributed key-value store.
- Stores all cluster data (e.g., nodes, pods, configs).
- Backup of etcd = backup of your whole cluster state.

📌 c. kube-scheduler

- Watches for new pods with no assigned node.
- Selects the best node to run the pod based on:
  - Resource availability
  - Node affinity
  - Taints and tolerations
  - Other constraints

📌 d. kube-controller-manager

- Runs various controllers:
  - **Node Controller** – monitors node status
  - **Replication Controller** – ensures desired pod count
  - **Endpoints Controller** – manages endpoint objects
  - **Service Account & Token Controller**

📌 e. cloud-controller-manager *(optional)*

- Integrates with cloud provider APIs.
- Manages:
  - Load balancers
  - Volumes
  - Node instances

🟦 2. Worker Node Components

Worker nodes run the containers that make up your application.

🧩 Components:

📌 a. kubelet

- Communicates with API Server.
- Ensures the containers described in PodSpecs are running and healthy.

📌 b. kube-proxy

- Manages network rules.
- Enables communication between services (internal & external).
- Uses iptables or IPVS.

📌 c. Container Runtime

- The software responsible for running containers.
- Examples: **Docker**, **containerd**, **CRI-O**

📦 3. Pods and Workloads

🧊 What is a Pod?

- The smallest and simplest unit in Kubernetes.

- Encapsulates one or more containers with shared storage/network.
- Containers in the same pod can communicate via localhost.

🔁 Full Working Flow (End-to-End)

1. **User runs a command**: kubectl apply -f app.yaml
2. **API Server receives the request**
3. **Validates and stores it in etcd**
4. **Scheduler assigns a pod to a node**
5. **Controller manager ensures desired state**
6. **kubelet on selected node creates the pod**
7. **Container runtime pulls and starts the container**
8. **kube-proxy sets up networking for service discovery**

🌐 Networking Model in K8s

- Each Pod gets a **unique IP address**.
- Pods can talk to each other **without NAT**.
- Services (ClusterIP, NodePort, LoadBalancer) expose Pods.
- kube-dns or CoreDNS allows name-based resolution.