The Heart of Kubernetes: Understanding Its Architecture ⚙

1. Master Node (Control Plane)

The Master Node is the brain of the Kubernetes cluster. It manages the entire cluster and ensures that everything runs smoothly. Here are its key components:

⦿ Cloud Controller Manager (CCM)

Handles interactions with cloud providers (e.g., AWS, Azure, GCP). It ensures that Kubernetes can leverage cloud-specific features like load balancers or storage.

⦿ API Server:

Acts as the gateway to the Kubernetes cluster. All communication with the cluster (e.g., deploying applications, scaling pods) goes through the API Server.

⦿ etcd:

A highly reliable distributed key-value store that holds all the configuration data for the cluster. Think of it as the database for Kubernetes.

⦿ Scheduler:

Decides where to run your application pods by allocating them to worker nodes based on resource availability and other constraints.

⦿ Controller:

Ensures that the desired state of the cluster matches the actual state. For example, if a pod fails, the controller will restart it.

2. Worker Nodes

Worker nodes are where your applications actually run. Each worker node contains the necessary tools to manage and execute containers. Here are its key components:

⦿ Kubelet:

The primary agent on each worker node. It ensures that the pods defined in the cluster specification are running correctly. It communicates with the API Server to report the status of pods and nodes.

⦿ Kube-proxy:

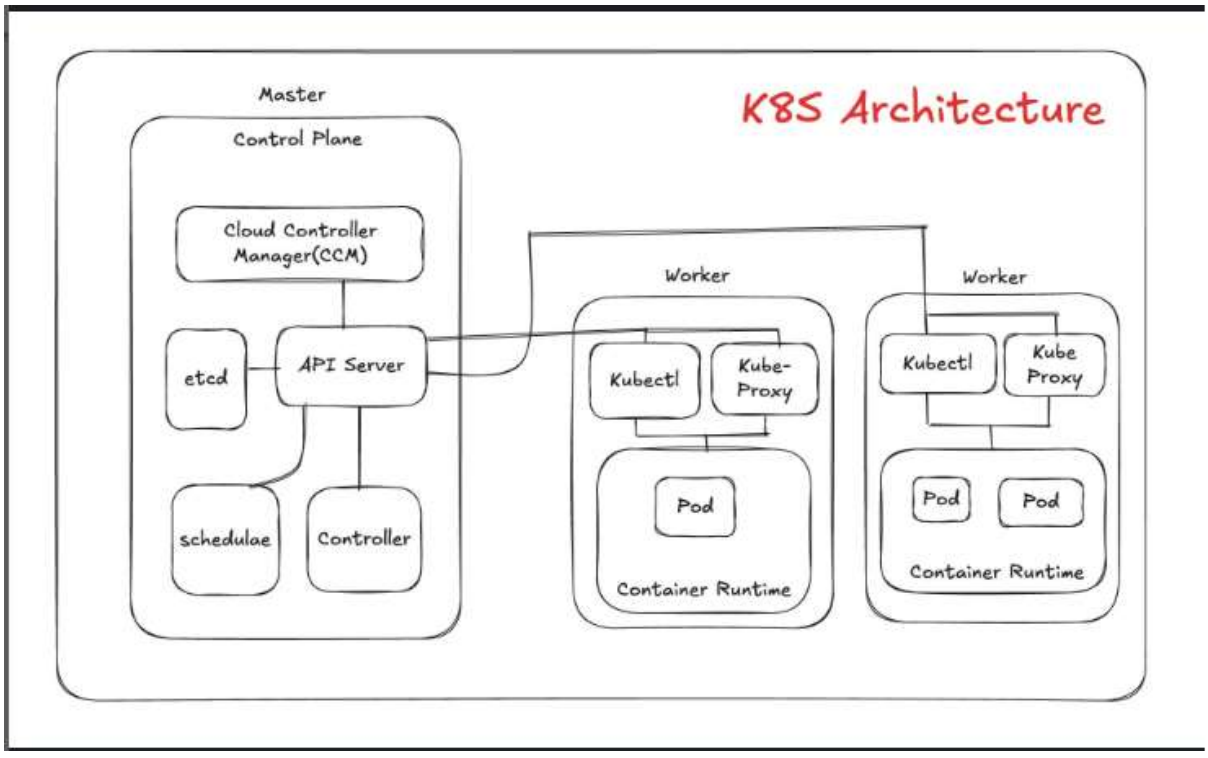
Manages network rules and enables communication between services within the cluster. It ensures that traffic reaches the correct pods.

⦿ Container Runtime:

The software responsible for running containers. Common container runtimes include Docker, containerd, or CRI-O.

⦿ Pods:

The smallest deployable units in Kubernetes. Pods can contain one or more tightly coupled containers. They are the building blocks of your applications.



In Kubernetes, nodes can be added or removed based on workload demands. Adding more worker nodes allows the cluster to run more applications or manage higher workloads, enhancing scalability.