\*\*KUBE AI SERVER:

Receives requests (e.g., to deploy an application, update resources) from users or internal components.

Validates and processes these requests

Interacts with the etcd database to store or retrieve the cluster's state.

Ensures the desired state of the cluster is achieved by working with other components (like the scheduler and controllers).

\*\*CONTROL MANAGER:

The **Kube Controller Manager** in Kubernetes is a key component responsible for managing various controllers that ensure the Kubernetes cluster operates as expected

 **Node Controller**: Monitors the health of nodes and takes action if nodes go down.

 **Replication Controller**: Ensures that the correct number of replicas (pods) are running at all times.

 **Endpoint Controller**: Manages endpoint objects, keeping them in sync with services and pods.

\*\*ETCD:

**etcd** is a distributed key-value store used by Kubernetes to store all cluster data.

It acts as the **central database** for Kubernetes, holding the entire cluster's state.

\*\*SHEDULAR:

The **Kube Scheduler** is responsible for deciding **which node** in the cluster should run a newly created pod.

**Workload Distribution**: Ensures that the workload is evenly distributed across nodes.

\*\*KUBELET:

The **Kubelet** is an agent that runs on every node (worker or master) in the Kubernetes cluster

**Node Status**: Reports the status of the node back to the API serve

\*\*KUBEPROXY:

**Service Networking**: It ensures that network traffic for a service is routed to the appropriate pod.

The **Kube Proxy** is responsible for **networking** within a Kubernetes cluster

**Load Balancing**: Distributes incoming traffic across the multiple pods that are part of a service.