# Hare Krishna

## Cloud Computing

## Topic

## KUBERNETES

## Roll No.0003

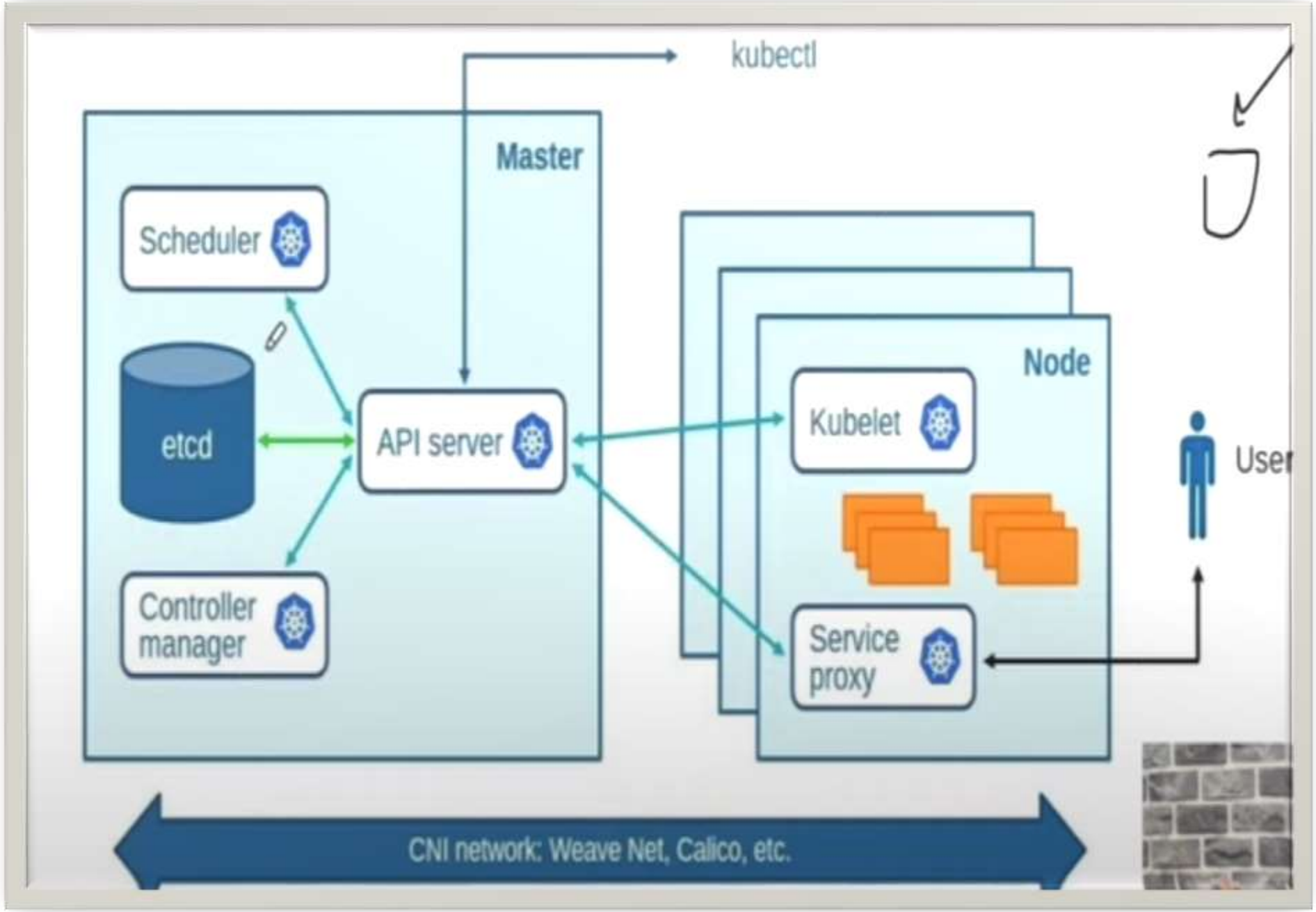
It's Me-Ishu boy

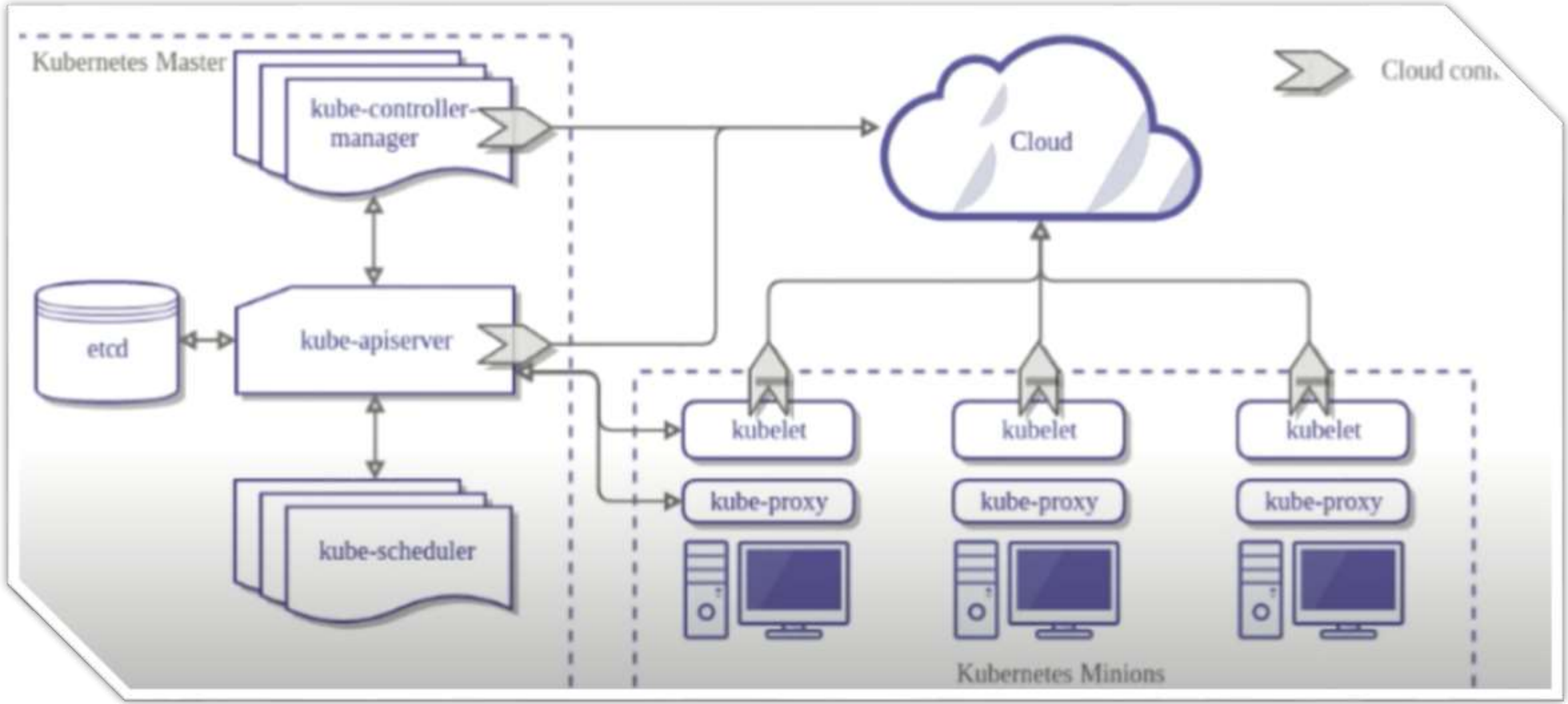
Kubernetes, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications. It's a portable, extensible, and open-source platform that facilitates both declarative configuration and automation.

Kubernetes was originally designed by Google, and is now maintained by the Cloud Native Computing Foundation (CNCF). It's a highly scalable and flexible system that can run on a variety of platforms, including on-premises, in the cloud, or in a hybrid environment.

The name Kubernetes originates from Greek, meaning "helmsman" or "pilot". K8s as an abbreviation results from counting the eight letters between the "K" and the "s".

It was founded by Joe Beda, Brendan Burns, and Craig McLuckie. It was first announced in mid-2014. K8s v1.0 was released on July 21, 2015.


kubectl
Master
Scheduler
etcd
API server
Controller manager
Node
Kubelet
Service proxy
User
CNI network: Weave Net, Calico, etc.


Kubernetes Master
kube-controller-manager
Cloud
Cloud con
etcd
kube-apiserver
kubelet
kubelet
kubelet
kube-proxy
kube-proxy
kube-proxy
kube-scheduler
Kubernetes Minions

**Master Node:**

•**API Server:** The brain of Kubernetes. It handles all communication with the cluster, manages the cluster state, and ensures all resources are properly allocated and managed.

•**etcd:** A highly available and consistent distributed key-value store. It's used to store all configuration data, including pod definitions, service definitions, and cluster state.

•**Scheduler:** Responsible for deciding on which worker node to place newly created pods based on resource availability and other factors.

•**Controller Manager:** Monitors the state of the cluster and automatically takes corrective actions to ensure the desired state is maintained. This includes things like restarting failed pods, scaling up/down deployments, and deleting completed pods.

**Worker Node:**

•**Kubelet:** An agent running on each worker node that is responsible for managing pods and containers. It communicates with the API Server to receive pod instructions and ensures the pods are running as intended.

•**Kube Proxy:** Handles network routing and service discovery. It directs traffic to the appropriate pods based on service definitions.

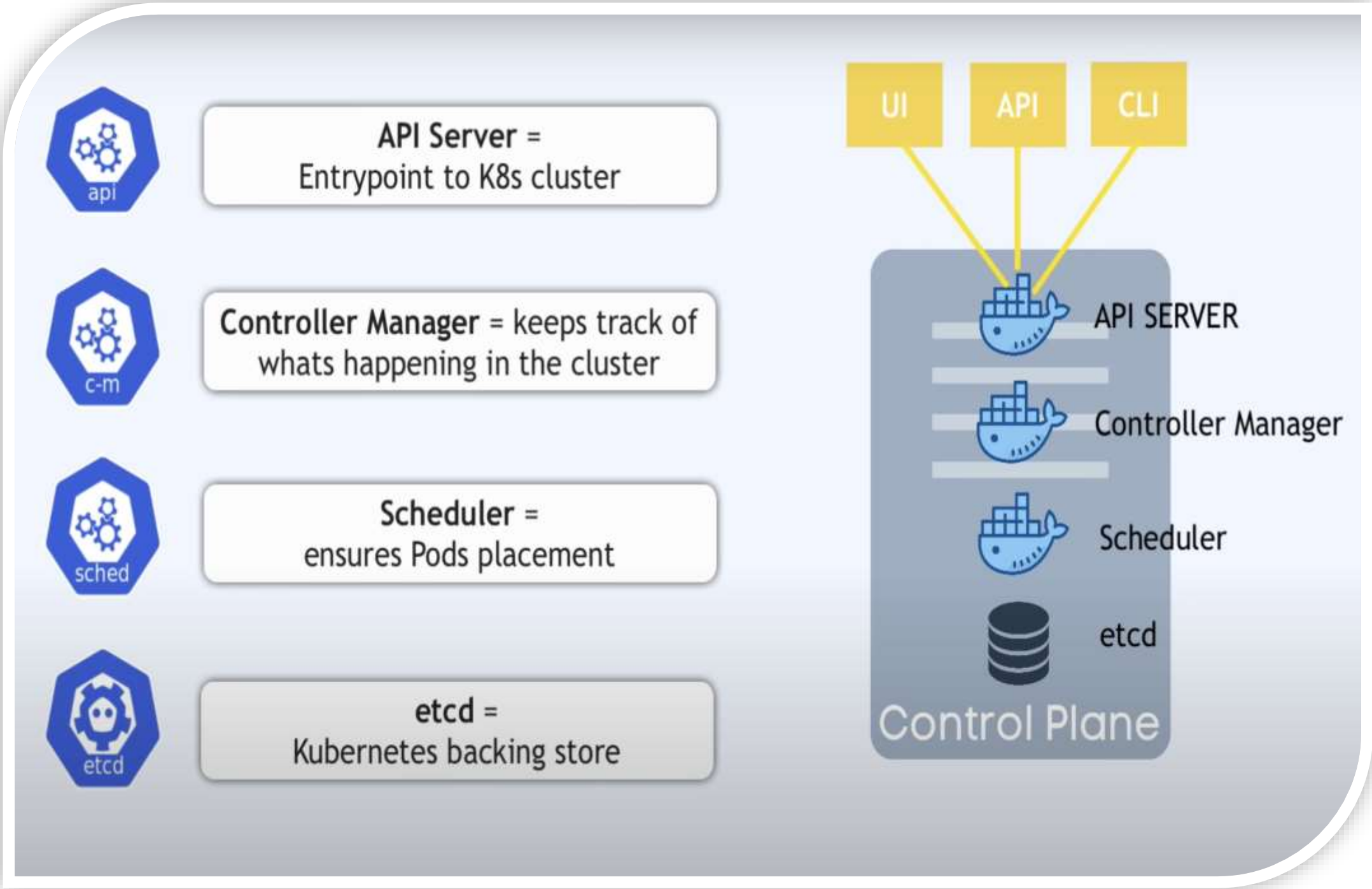
•**Pods:** The smallest unit of deployment in Kubernetes. They contain one or more containers that share resources and networking.

•**Containers:** Lightweight, isolated processes that encapsulate everything needed to run an application, including libraries, dependencies, and configuration.

**Communication:**

•**Kubectl:** A command-line tool used by users to interact with the Kubernetes API server.

•**CNI Network:** Container Network Interface, a standard interface for managing network connectivity for containers. The image shows "Weave Net, Calico, etc" as examples of commonly used CNI implementations.


api
API Server =
Entrypoint to K8s cluster
c-m
Controller Manager = keeps track of
whats happening in the cluster
sched
Scheduler =
ensures Pods placement
etcd
etcd =
Kubernetes backing store
UI
API
CLI
API SERVER
Controller Manager
Scheduler
etcd
Control Plane

**Control Plane Components**

- **API Server:** The entry point to the cluster, receiving requests from users and tools.
- **Controller Manager:** Ensures that the desired state of the cluster is maintained by continuously monitoring and adjusting.
- **Scheduler:** Determines which worker node each new pod will be placed on.
- **etcd:** Stores the cluster state, such as pod configurations and resources.

**Worker Nodes:** Run applications in the form of Pods

**Interactions:**

- Users and tools interact with the cluster through the **API Server**.
- The **Controller Manager** ensures that the desired state of the cluster is maintained.
- The **Scheduler** assigns pods to worker nodes based on resource availability.
- The **etcd** database is used to store the cluster state.

This structure allows for a highly scalable and automated Kubernetes cluster management.

- Node is a server or virtual Machine
- Pod is the smallest unit in Kubernetes which gives an abstraction over docker container

**Development of Kubernetes over time**

- **2014**: Kubernetes was first announced by Google
- **2015**: Kubernetes was open-sourced and donated to the Cloud Native Computing Foundation (CNCF)
- **2016**: Kubernetes 1.0 was released, marking the first stable version
- **2017**: Kubernetes 1.6 was released, introducing features like rolling updates and self-healing
- **2018**: Kubernetes 1.10 was released, introducing features like automated deployment and scaling
- **2019**: Kubernetes 1.15 was released, introducing features like improved security and networking
- **2020**: Kubernetes 1.18 was released, introducing features like improved scalability and performance

**Advantages of Kubernetes**

- **Scalability**: Automatically adjust the size of a cluster required to run a service
- **High Availability**: Ensure no downtime by automatically restarting containers that fail
- **Flexibility**: Run on a variety of platforms, including on-premises, in the cloud, or in a hybrid environment
- **Automation**: Automate day-to-day operations, reducing manual effort
- **Extensibility**: Easily extend and customize to meet specific needs

**Disadvantages of Kubernetes**

- **Complexity**: Can be complex to manage and troubleshoot
- **Resource Intensive**: Requires significant resources, including CPU, memory, and storage
- **Security Risks**: If not properly configured, can introduce security risks

# Azure Kubernetes Service (AKS)

Deploy and scale containers on managed Kubernetes

Simplify Kubernetes operations, build cloud-native apps, and innovate with AI and open-source technology.

Try Azure for free

Create a pay-as-you-go account