Kubernetes

Prepared by:
Ms. Avita Katal
Assistant Professor (SG)
School of Computer Science
UPES, Dehradun

Objectives

1. Define Kubernetes.
2. Explain what Kubernetes is not.
3. Relate Kubernetes concepts.
4. Describe Kubernetes capabilities.
5. Describe the Kubernetes ecosystem.

What Kubernetes is?

*"An open-source system for **automating deployment, scaling, and management of containerized applications**."*

- Kubernetes is widely available and has positioned itself as the de facto choice for **container orchestration**.
- Kubernetes is an **open-source containerization orchestration platform**, developed as a project by **Google** and currently maintained by the **Cloud Native Computing Foundation.**
- It is **easily portable across clouds and on-premises**
- With its growing ecosystem of projects and products by both member and non-member partners, Kubernetes is recognized as the "**go to" container orchestration solution.**
- **Kubernetes as a Service** grew as an extension of **Containers as a Service (CaaS).**
- Kubernetes facilitates **declarative management** in which it automatically performs the necessary operations towards achieving the called for state.

What Kubernetes is not?

It is not a traditional, all-inclusive platform as a service.

Does not provide continuous integration/ continuous delivery pipelines to build applications or deploy source code.

Does not prescribe logging, monitoring, and alerting solutions

Does not provide built-in middleware, databases, or other services.

Not rigid or opinionated but, more of a flexible model that supports an extremely diverse variety of workloads, including stateless, stateful, and data-processing workloads

Kubernetes Concept

Terms	Meaning
Pods and Workloads	Pods represent the **smallest deployable compute object** and the **higher-level abstractions to run workloads.**
Services	Services **expose applications running on sets of Pods**. Each Pod is assigned a **unique IP address**. **Sets of Pods have a single DNS name.**
Storage	Kubernetes supports both **persistent and temporary storage for Pods**.
Configuration	Configuration, which refers to the **provisioning of resources for configuring Pods**.
Security	Security measures for cloud-native workloads, which enforce **security for Pod and API access.**
Policies	Policies for **groups of resources**, ensuring that Pods match to Nodes so that the **kubelet can find them and run the Pods**.

Kubernetes Concept

Terms	Meaning
Schedule/Eviction	Scheduling and Eviction that **runs and proactively terminates one or more Pods on resource-starved Nodes.**
Preemption	Preemption, which is all about **prioritization.** Preemption terminates lower priority Pods so that Pods with higher priority can schedule and run on Nodes.
Administration	Cluster administration provides the **details necessary to create or administer a cluster.**

Kubernetes Capabilities

UPES
UNIVERSITY OF TOMORROW

Automated rollouts and rollbacks

Automated **rollouts of changes to application or configuration, health monitoring and rolling back changes.**

Storage Orchestration

Storage orchestration that mounts a chosen storage system including **local storage, network storage, or public cloud.**

Horizontal Scaling

Horizontal scaling of workloads based on metrics, or via commands.

Kubernetes Capabilities

Automated Bin Packing

- **Increases utilization and cost savings** using a mix of critical and best-effort workloads.
- Performs **container auto-placement** based on resource requirements and conditions without sacrificing high availability.

Secret and Configuration management

- Kubernetes includes **Secret and configuration management** of sensitive information including **passwords, OAuth tokens, and SSH keys**.

- It handles **deployments and updates to secrets and configuration** without rebuilding images.

Kubernetes Capabilities

IPv4 /IPv6 Dual stack

- Kubernetes assigns both **IPv4 and IPv6 addresses to Pods and Services.**

Batch Execution

- Manages **batch and continuous integration workloads and automatically replaces failed containers, if configured.**

Kubernetes Capabilities

Self-Healing

It **self-heals failing or unresponsive containers**

Self-Discovery and Load Balancing

- It discovers **Pods using IP addresses or a DNS name**
- **Load balances traffic** for better performance and high availability
- Kubernetes **adds features to your cluster without modifying source code**.

Kubernetes Ecosystem

- The Kubernetes ecosystem is a large, rapidly growing ecosystem where its **services, support, and tools are widely available.**

- Running containerized applications **requires separate tools** in addition to the Kubernetes orchestration tool.

- The ecosystem includes services like:
  - ***Building container images.***
  - ***Storing images in a container registry.***
  - ***Application logging and monitoring and CI/CD capabilities.***

Kubernetes Ecosystem

UPES
UNIVERSITY OF TOMORROW

Public cloud
Prisma, IBM, Google, AWS

Open Source Frameworks
Red Hat, SUSE, VMware, Docker, Mesosphere, CloudFoundry

Management
Digital Ocean, CloudSoft, Weaveworks, SUPERGIANT, loodse

Load Balancing
VMWare, AVI networks, NGiNX

Tools
JFrog, Univa, Aspen, Mesh, Bitnami, Cloud 66

Monitoring and Logging
Sumalogic, DATADOG, New Relic, Grafana, sysdig, SignalFX, dynatrace

Security
GUARDCORE, BLACKDUCK, yubico, cilium, aqua, TwistLock, alcide

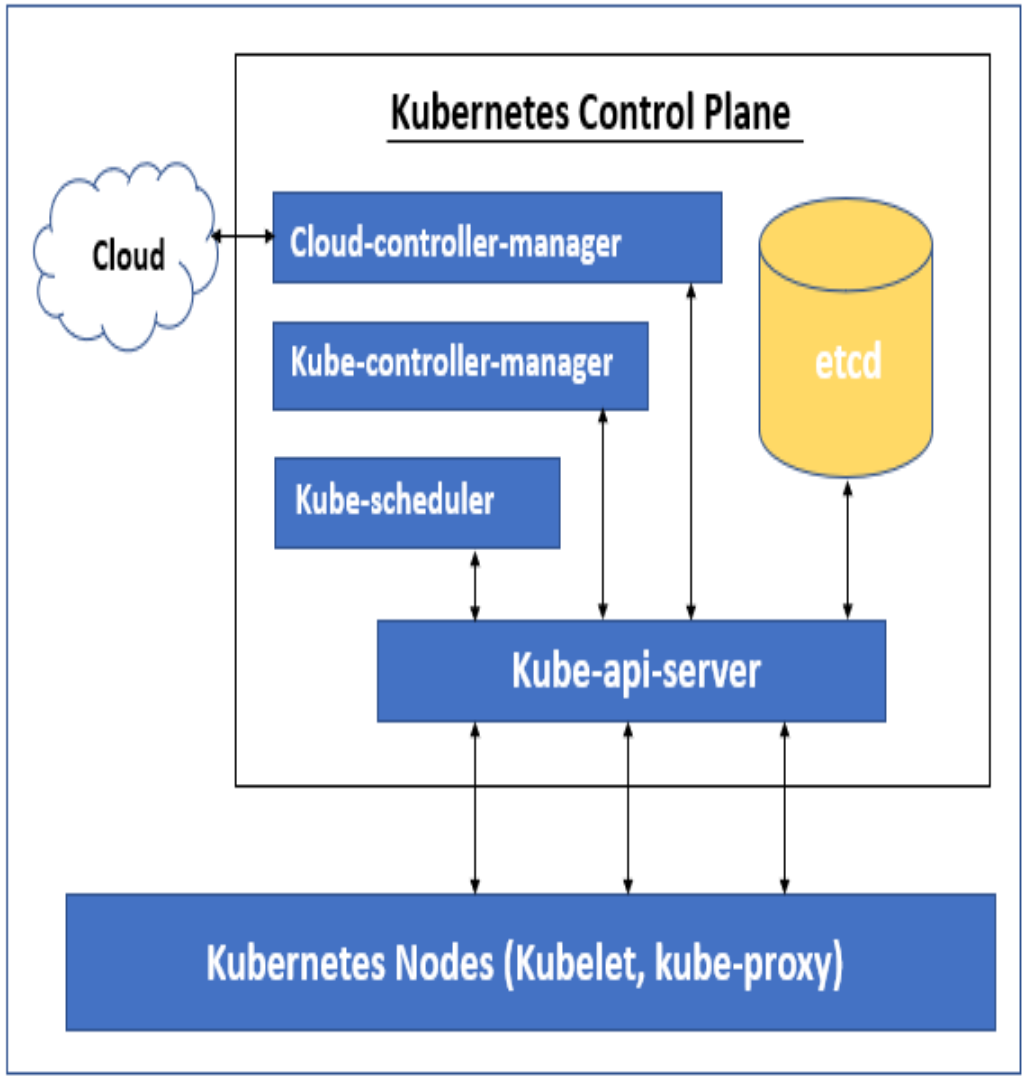
Kubernetes

Prepared by:
Ms. Avita Katal
Assistant Professor (SG)
School of Computer Science
UPES, Dehradun

Objectives

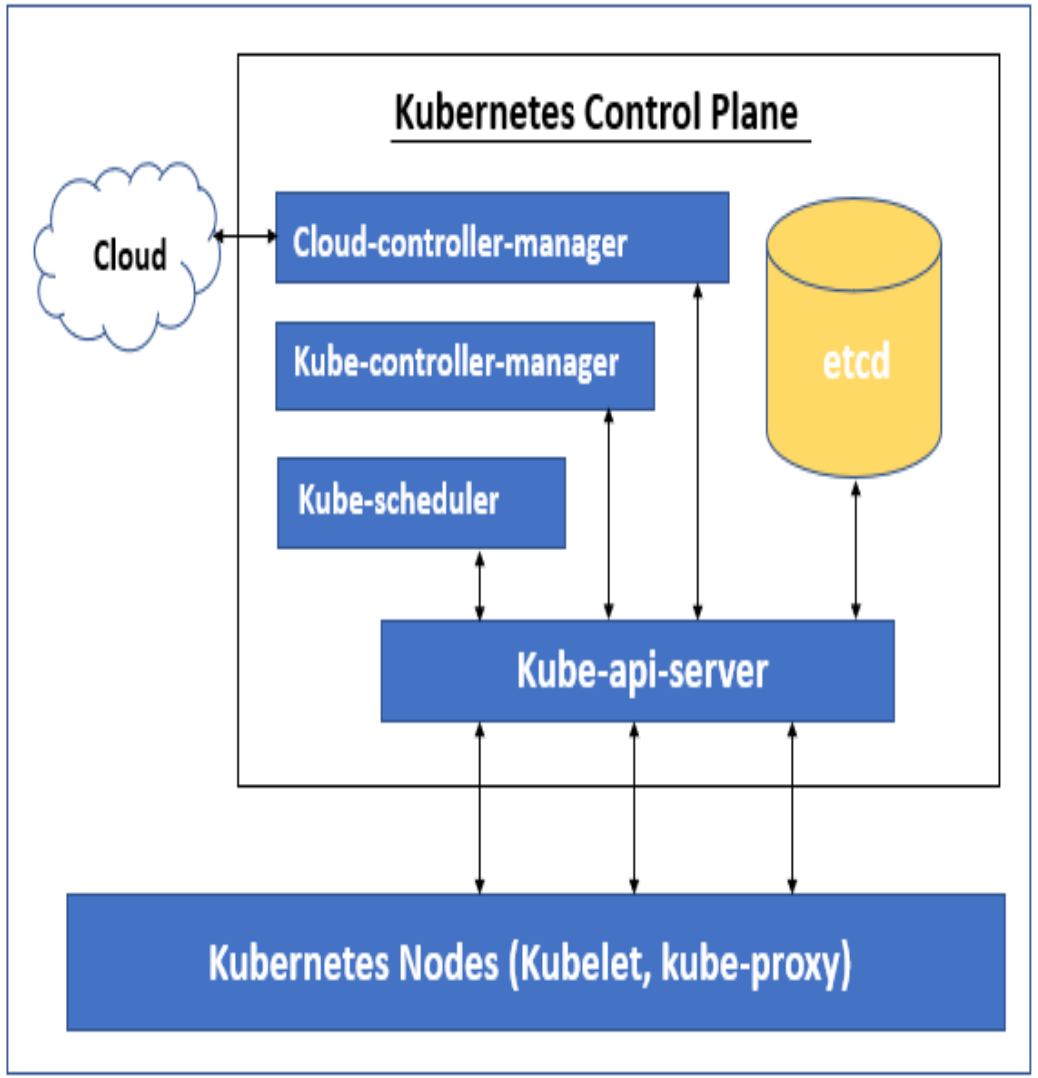
1. Identify the components of a Kubernetes architecture.
2. Identify the components of a control plane.
3. Identify the components of a worker plane.

Kubernetes Architecture



- A deployment of Kubernetes is called a **Kubernetes cluster.** A Kubernetes cluster is a **cluster of nodes that runs containerized applications.**
- Each cluster has **one master node** (**the Kubernetes Control Plane**) and **one or more worker nodes**.
- The **control plane maintains the intended cluster state** by making decisions about the cluster and detecting and responding to events in the cluster.
- Nodes are the worker machines in a Kubernetes cluster. In other words**, user applications are run on nodes.**
- **Nodes are not created by Kubernetes itself, but rather by the cloud provider.** This allows Kubernetes to run on a variety of infrastructures. The nodes are then managed by the control plane.

Kubernetes Architecture

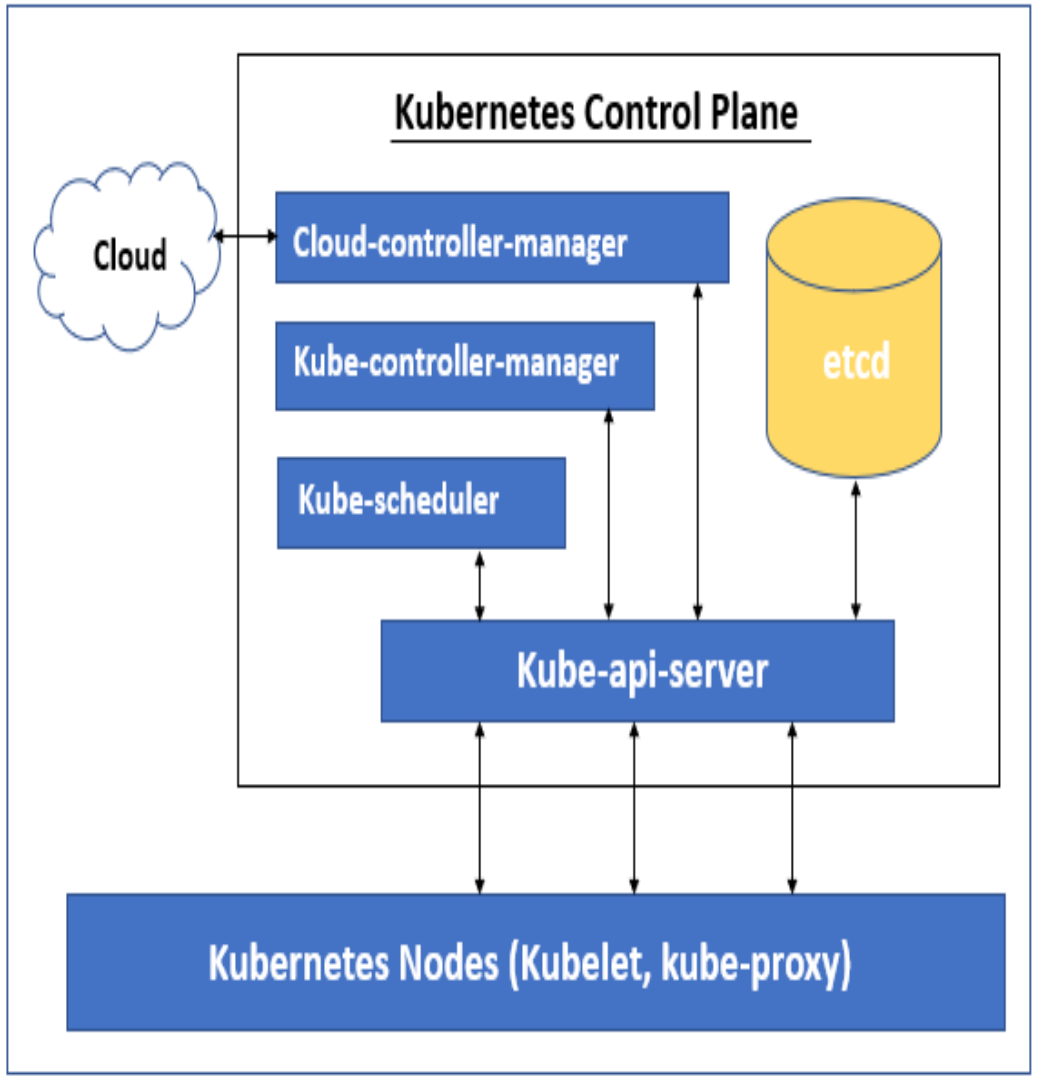


A control plane consists of controllers, API server, scheduler and an etcd.

KUBERNETES API

- The Kubernetes API **server exposes the Kubernetes API.**
- The API server **serves as the front-end for the control plane.** All communication in the cluster utilizes this API.
- The main implementation of a Kubernetes API server is **kube-apiserver** which is designed to **scale horizontally—by deploying more instances.**
- You can **run several instances of kube-api-server and balance traffic between those instances.**

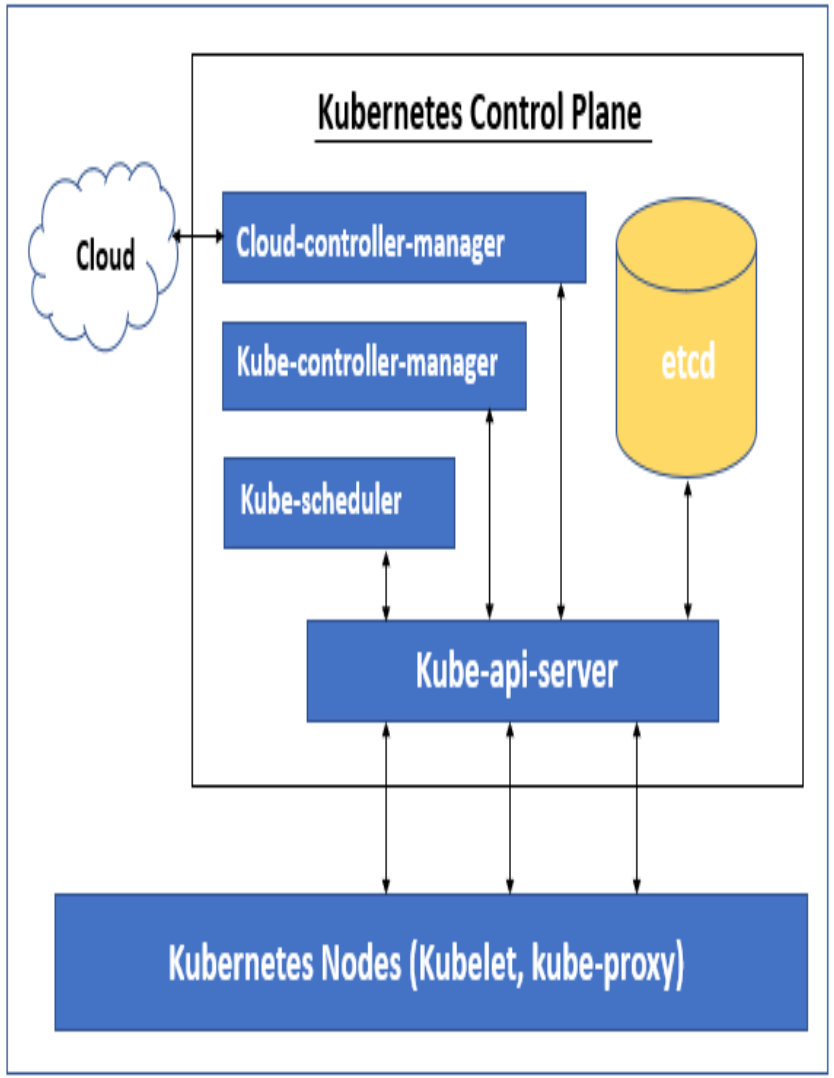
Kubernetes Architecture



Etcd

- Etcd is a **highly available, distributed key value store** that contains all the cluster data.
- When you tell **Kubernetes to deploy your application, that deployment configuration is stored in etcd.**
- Etcd **defines the state in a Kubernetes cluster, and the system works to bring the actual state to match the desired state.**

Kubernetes Architecture



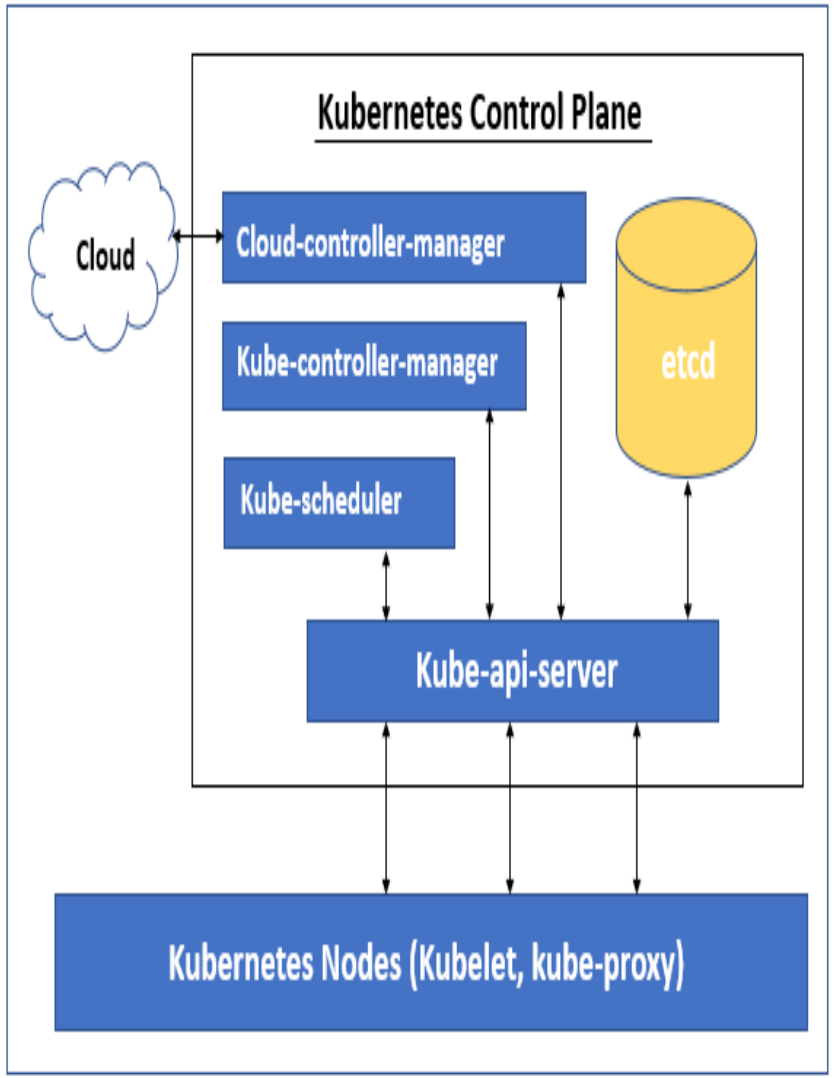
Kubernetes-scheduler

- The Kubernetes scheduler **assigns newly created Pods to nodes**. This basically means that the kube-scheduler determines **where your workloads should run within the cluster.**
- The scheduler **selects the most optimal node** according to Kubernetes **scheduling principles, configuration options, and available resources**.

Kubernetes-controller manager

- The Kubernetes controller manager runs all the **controller processes that monitor the cluster state, and ensure the actual state of a cluster matches the desired state.**

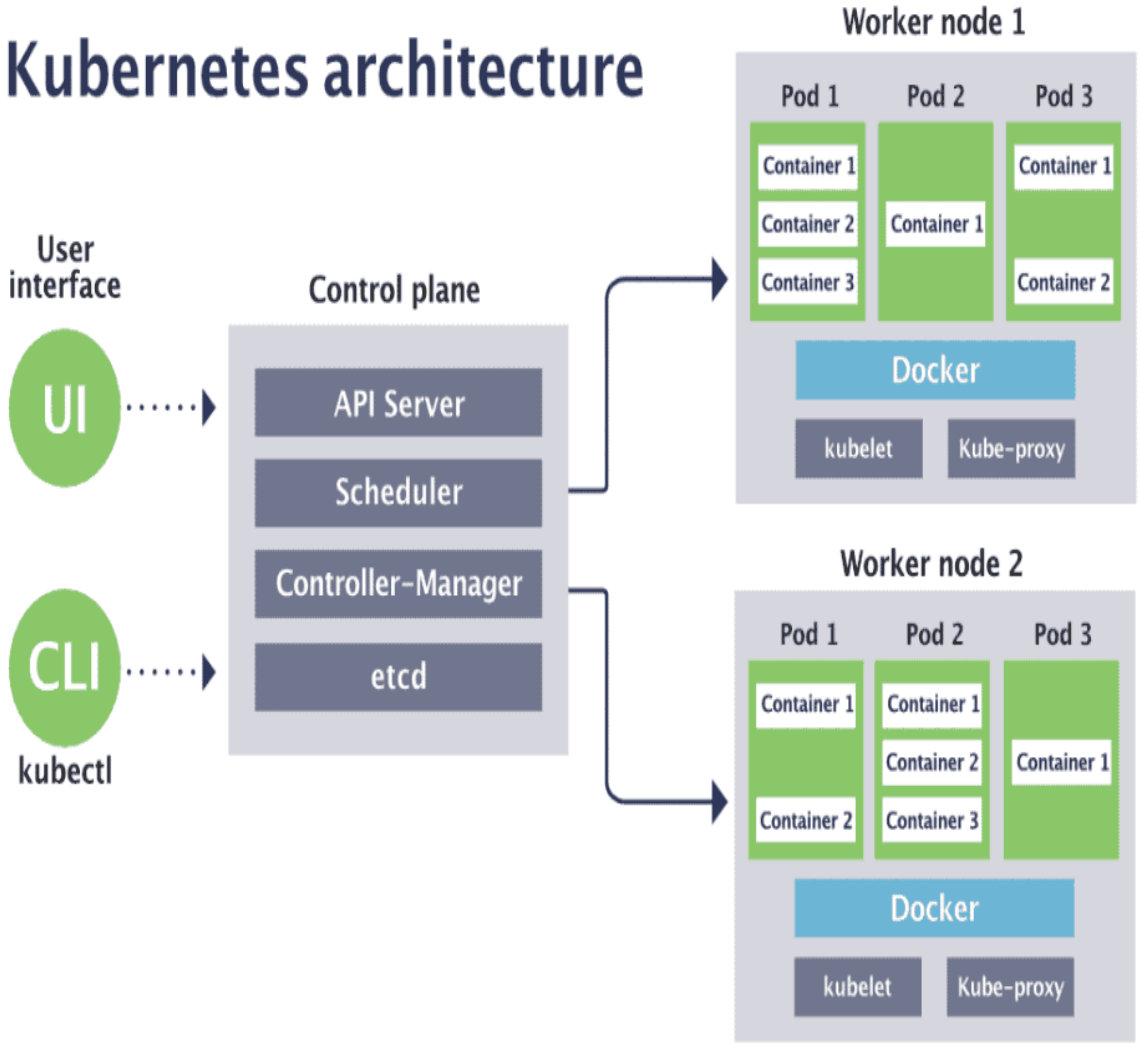
Kubernetes Architecture



Cloud-controller manager

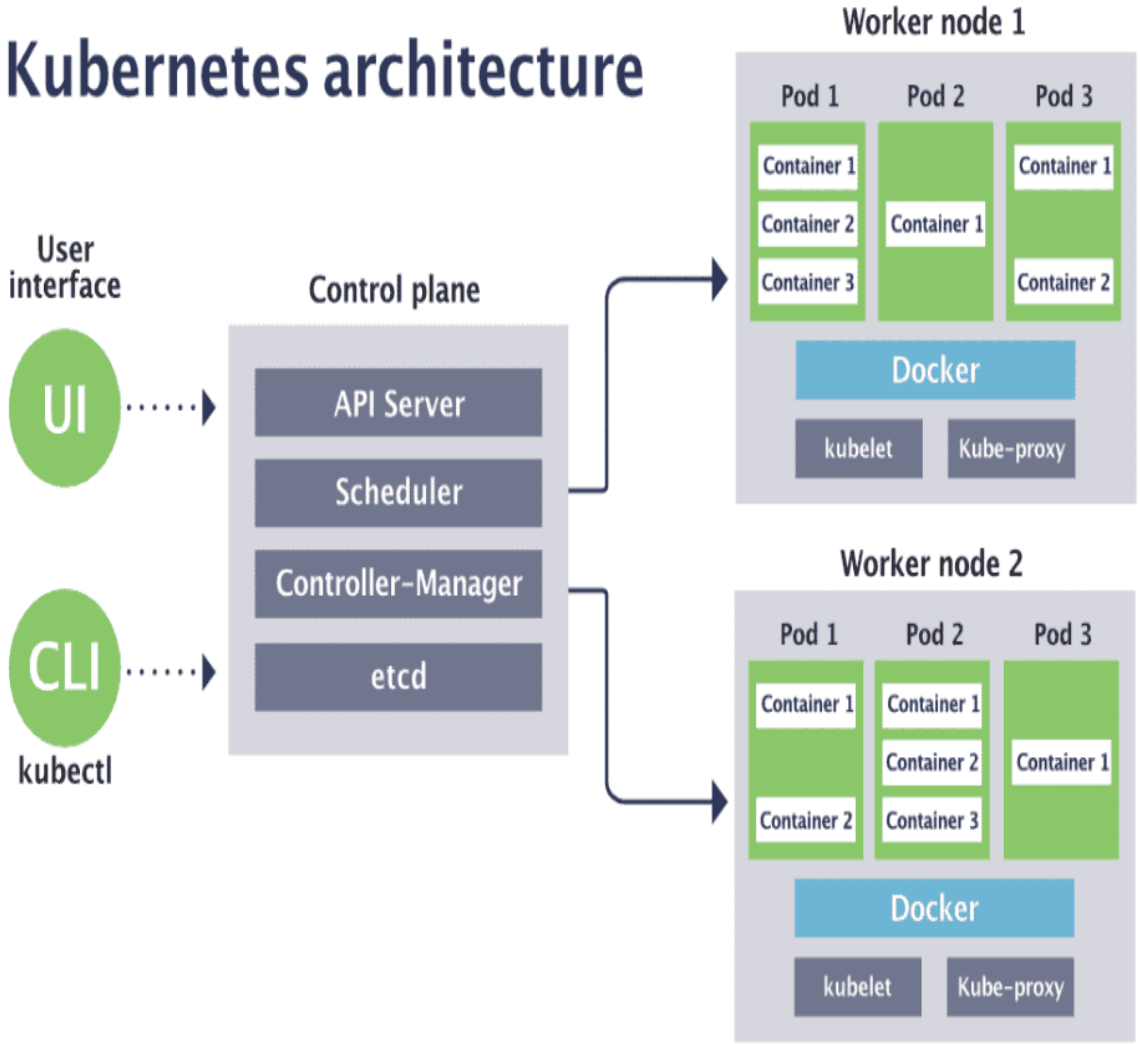
- The cloud controller manager **runs controllers that interact with the underlying cloud providers**.
- These controllers effectively **link clusters into a cloud provider's API.**
- Since Kubernetes is open source and would ideally be adopted by a variety of cloud providers and organizations,
- Kubernetes strives to be as **cloud agnostic** as possible.
- The cloud-controller-manager allows both Kubernetes and the cloud providers **to evolve freely without introducing dependencies on the other.**

Kubernetes Architecture



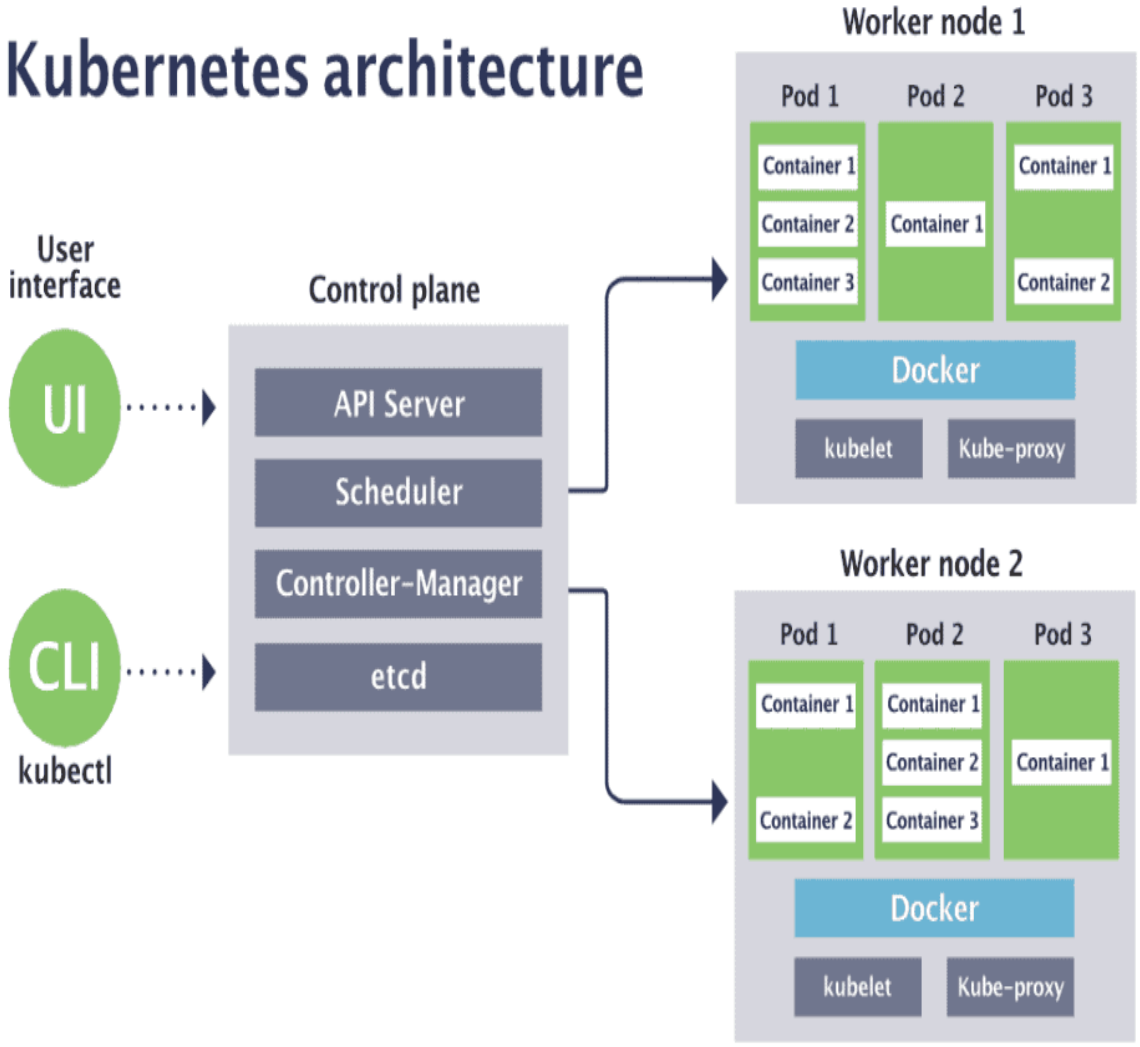
- **A worker plane is made of nodes, kubelet, container runtime and kube proxy.**
- Nodes are the worker machines in a Kubernetes cluster. Nodes can be **virtual or physical machines.**
- Each node is managed by the control plane and **contain the services necessary to run applications.**
- **Nodes include pods**, which are the smallest deployment entity in Kubernetes.
- **Pods include one or more containers**. Containers share **all the resources of the node and can communicate among themselves.**

Kubernetes Architecture



- The **kubelet** is the most important component of a worker node.
- This **controller communicates with the kube-apiserver** to receive new and modified pod specifications and ensure that the pods and their associated containers are running as desired.
- The **kubelet also reports to the control plane on the pods' health and status.**
- In order to start a pod, the **kubelet uses the container runtime**. The **container runtime is responsible for downloading images and running containers.** Rather than providing a single container runtime, **Kubernetes implements a Container Runtime Interface that permits pluggability of the container runtime**.
- While Docker is likely the best-known runtime, **Podman and Cri-o** are two other commonly used container runtimes..

Kubernetes Architecture



- The **Kubernetes proxy is a network proxy** that **runs on each node** in a cluster.

- This **proxy maintains network rules that allow communication to Pods running on nodes**. In other words, communication to workloads running on your cluster. **This communication can come from within or outside of the cluster**

Thank You