kubernetes

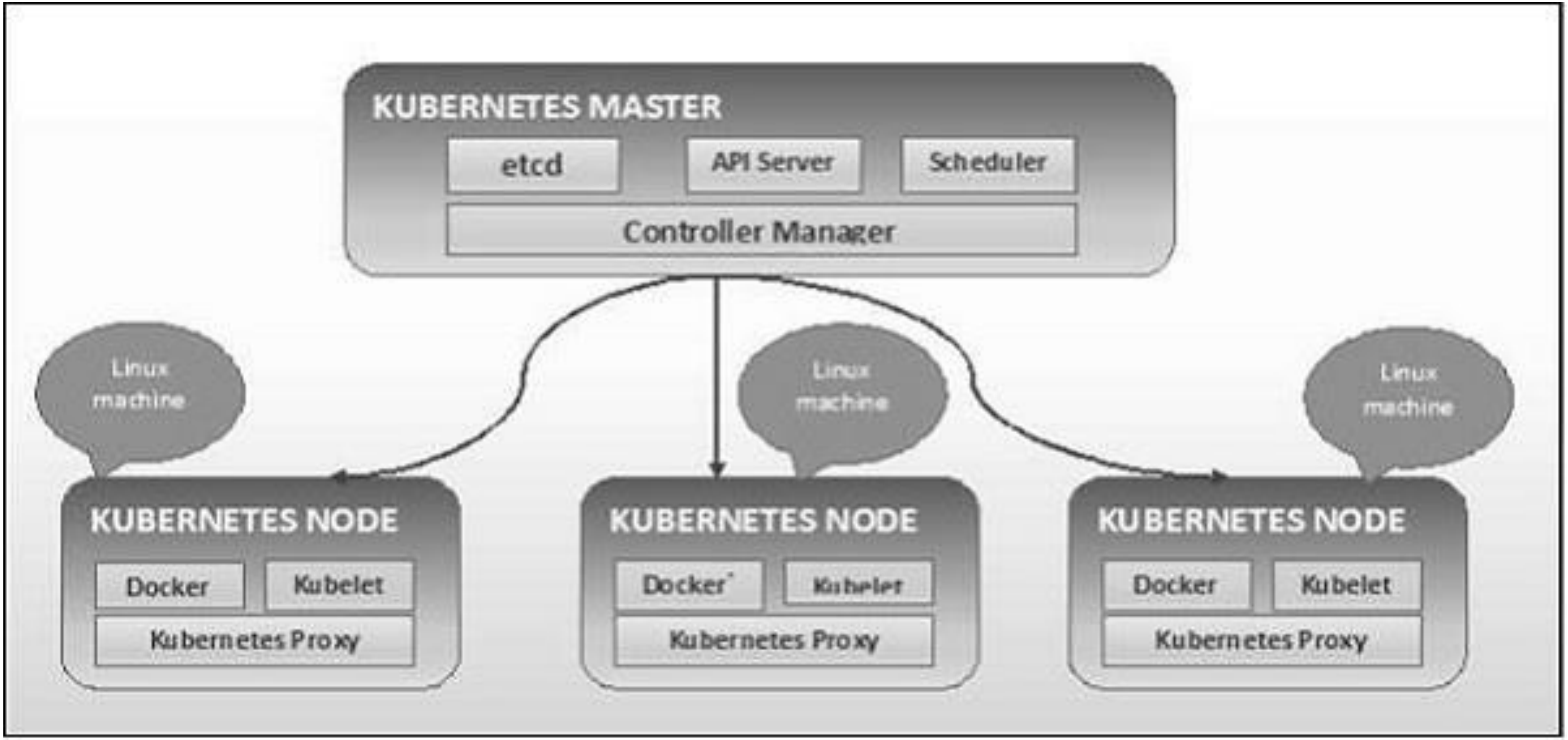
What is K8S?

- Kubernetes is a container management technology developed in Google lab to manage containerized applications in different kind of environments such as physical, virtual, and cloud infrastructure.
- It is an open source system which helps in creating and managing containerization of application.
- This is now hosted by hosted by Cloud Native Computing Foundation (CNCF)
- This is also known as the enhanced version of Borg which was developed at Google to manage both long running processes and batch jobs, which was earlier handled by separate systems.

Features of Kubernetes

- Continues development, integration and deployment
- Containerized infrastructure
- Application-centric management
- Auto-scalable infrastructure
- Environment consistency across development testing and production
- Loosely coupled infrastructure, where each component can act as a separate unit
- Higher density of resource utilization
- Predictable infrastructure which is going to be created

Kubernetes - Cluster Architecture



Master Node Components

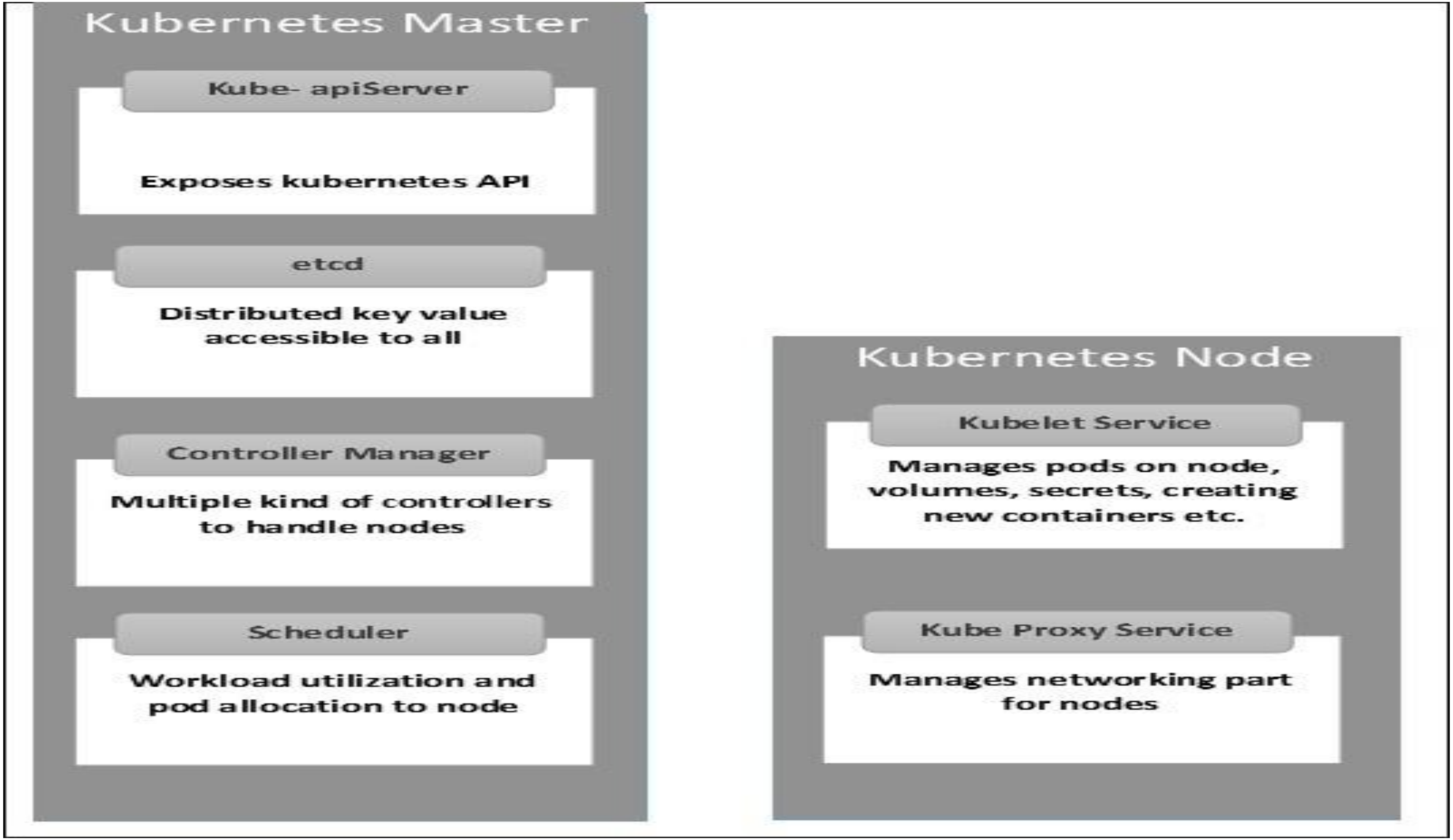
- **Kube API-server** performs all the administrative tasks on the master node. A user sends the rest commands as YAML/JSON format to the API server, then it processes and executes them. The Kube API-server is the front end of the Kubernetes control plane.
- **etcd** is a distributed key-value store that is used to store the cluster state. Kubernetes stores the file in a database called the **etcd**. Besides storing the cluster state, etcd is also used to store the configuration details such as the subnets and the config maps.
- **Kube-scheduler** is used to schedule the work to different worker nodes. It also manages the new requests coming from the API Server and assigns them to healthy nodes.
- **Kube Controller Manager** task is to obtain the desired state from the API Server. If the desired state does not meet the current state of the object, then the corrective steps are taken by the control loop to bring the current state the same as the desired state.

Control manager in Kubernetes architecture

- **Node Manager,** it manages the nodes. It creates new nodes if any node is unavailable or destroyed.
- **Replication Controller,** it manages if the desired number of containers is running in the replication group.
- **Endpoints controller,** it populates the endpoints object that is, joins Services & Pods.

Node Components

- **Kubelet** is an agent that runs on each worker node and communicates with the master node. It also makes sure that the containers which are part of the pods are always healthy. It watches for tasks sent from the API Server, executes the task like deploy or destroy the container, and then reports back to the Master.
- **Kube-proxy** is used to communicate between the multiple worker nodes. It maintains network rules on nodes and also make sure there are necessary rules define on the worker node so the container can communicate to each in different nodes.
- **Kubernetes pod** is a group of one or more containers that are deployed together on the same host. Pod is deployed with a shared storage/network, and a specification for how to run the containers. Containers can easily communicate with other containers in the same pod as though they were on the same machine.
- **Container Runtime** is the software that is responsible for running containers. Kubernetes supports several container runtimes: Docker, containers.



Kubernetes Master

Kube- apiServer

Exposes kubernetes API

etcd

Distributed key value accessible to all

Controller Manager

Multiple kind of controllers to handle nodes

Scheduler

Workload utilization and pod allocation to node

Kubernetes Node

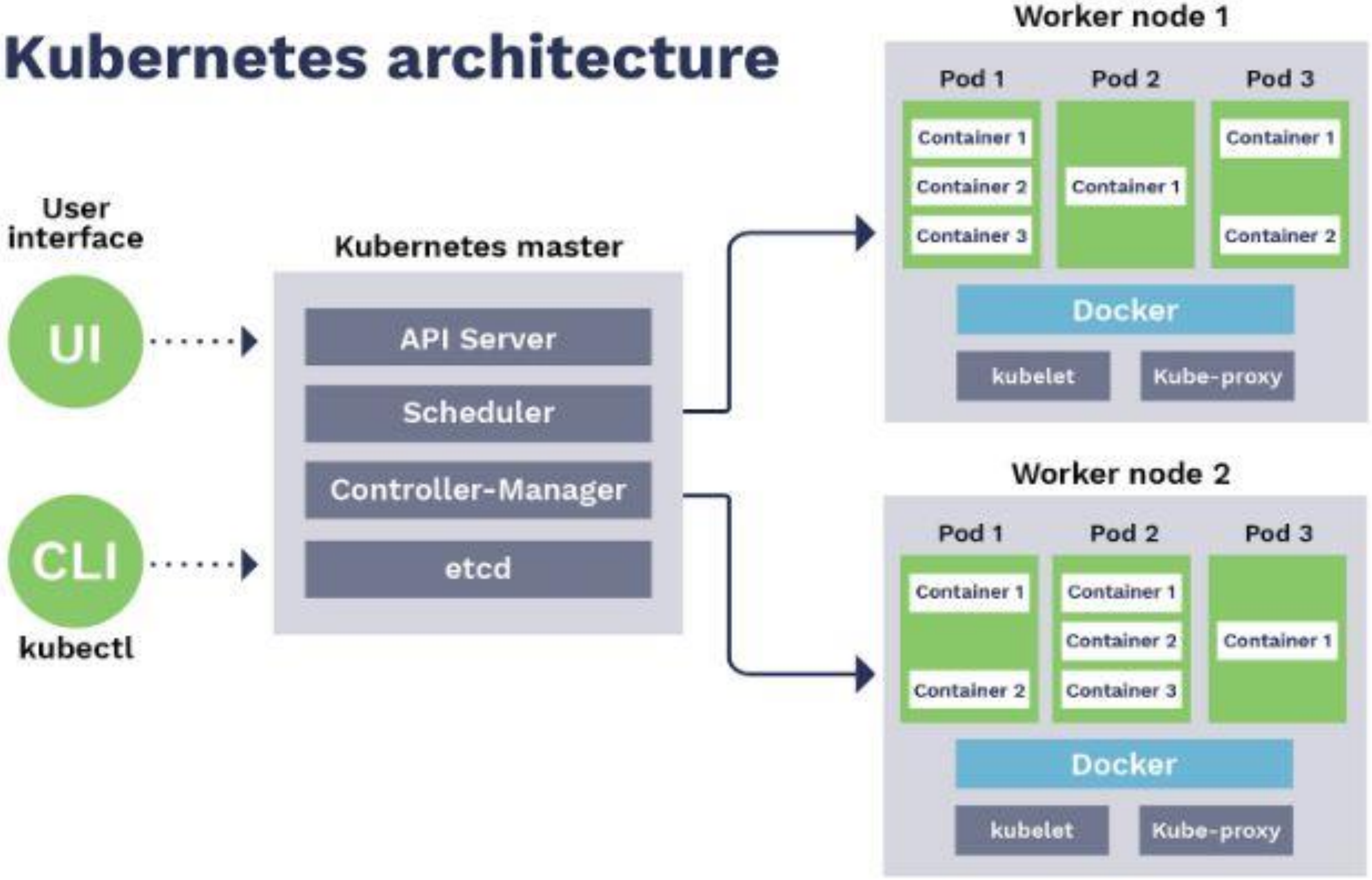
Kubelet Service

Manages pods on node, volumes, secrets, creating new containers etc.

Kube Proxy Service

Manages networking part for nodes

Kubernetes architecture



User interface

UI

CLI

kubectl

Kubernetes master

API Server

Scheduler

Controller-Manager

etcd

Worker node 1

Pod 1

Container 1

Container 2

Container 3

Pod 2

Container 1

Pod 3

Container 1

Container 2

Docker

kubelet

Kube-proxy

Worker node 2

Pod 1

Container 1

Container 2

Pod 2

Container 1

Container 2

Container 3

Pod 3

Container 1

Docker

kubelet

Kube-proxy

Kubernetes Service