Kubernetes

1. Kubernetes is Open-source Orchestration tool and it is developed by Google.
2. It helps to manage containerized applications in different deployment environments like physical, virtual and cloud.

**Features of Kubernetes:**

1. High Availability-(it means No-down time)
2. Scalability--(it means high performance)
3. Disaster Recovery - Back and Restore

**Node**: Which is a simple server a physical (or) virtual machine basic component (or ) smallest unit of kuberenetes is pod.

1. Pod is basically an abstraction over a container.
2. Pod gets it’s own IP address
3. If one pod is goes down (or) crashed that time a new pod will get created in it’s place but ip won’t change because of “Service” and volumes also remains constant.

**Service:** Service is basically static Ip (or) permanent Ip address that can be attached to the each pod.

Service has Two functionalities:

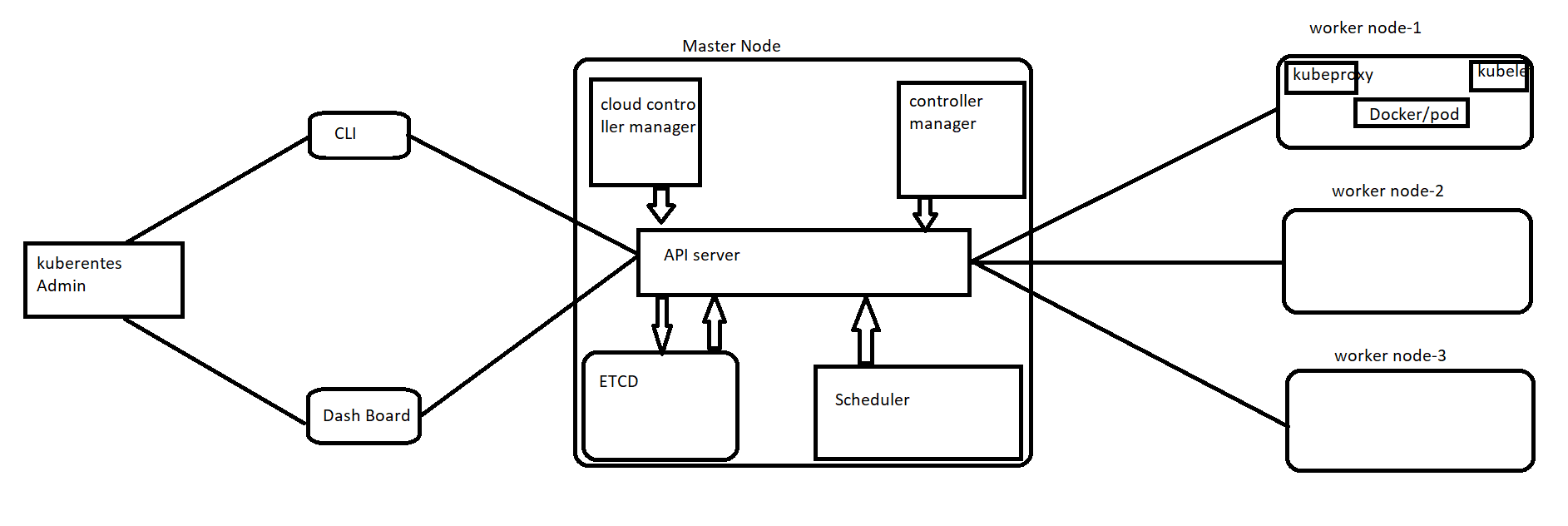
1. Permanent IP
2. Load Balancer

**Cluster**: It is a group of servers (or) group of nodes will be part of single system.

**Volumes:** It works is that is basically attaches a physical storage on a hard drive to your pod and that storage cloud be either an a local machine (or) the same server node where the pod is running.

**Kubernetes Architecture**

1. Master node is main node where you will be installing kubernetes software.(where you install kubernetes that is called master node)
2. Using master node we will be able to control multiple nodes.Technically this nodes are called as “worker nodes”
3. Kubernetes architecture you will be able to intract with kubernetes using different channels. Those channels are technically called as “CLI” and “Dash Board”



**API Server:** This is the main component in the kubernetes. Where all the integrations and communication across the kubernets will be happening through API server.

**Scheduler:** When you performing any deployment on kubernetes cluster it is a component which will be deciding where your replicas will be created

Example: If you want to create 6 replicas

1. Scheduler check the first worker node and understanding the configuration for that configuration how many replicas it suppose to created.
2. Scheduler keeps on intract with the API server
3. API server will intract with the woker nodes.

**ETCD:** It stores the entire configuration and state of the cluster.

1. All worker node status will be storing into “ETCD”
2. It is act as a Database.

**Controller Manager:** For any reason if one replica is down immediately it should bring up alternate replica because of kubernetes maintain the “Desired state” by default.

1. This instruction passed by the controller manager.
2. Controller manager intract with the API server.

**Worker Node:**

1. Every worker have kubelet and kubeproxy components

**Kubelet:** It is responsible for managing the deployment of pods to kubernetes nodes.

1. It recives commands from the API server and instructs the container runtime to start (or) stop containers.
2. Kubelet intract with both container and node.

**Kube-proxy:** Which will help to expose the service which running on worker node.

1. Kube-proxy like a Load balancer

**POD:** smallest unit of kuberenetes. Without pod a container cannot be part of a cluster.

**Difference between kubelet and kubectl?**

1. Kubelet is the technology that applies, creates, updates and destroys containers on a kubernetes node.
2. Kubectl is command line interface tool for working with a kubernetes cluster.

#kubectl get nodes -----> it shows the basic information about the nodes in a cluster

#kubectl apply -f filename -----> if you want to create a kubernetes resources such as a service.