**Kubernetes –**

\* Kubernetes is an open-source Container Management tool that automates container deployment,

container scaling, descaling, and container load balancing (also called a container orchestration tool).

\* Kubernetes is an open-source platform that manages Docker containers in the form of a cluster.

**Features of Kubernetes:**

Automated Scheduling– Kubernetes provides an advanced scheduler to launch containers on cluster nodes. It performs resource optimization.

Self-Healing Capabilities– It provides rescheduling, replacing, and restarting the containers which are dead.

Automated Rollouts and Rollbacks– It supports rollouts and rollbacks for the desired state of the containerized application.

Horizontal Scaling and Load Balancing– Kubernetes can scale up and scale down the application as per the requirements.

**Docker Swarm** **Kubernetes**

No Auto-Scaling Auto-Scaling

Does Auto Load-Balancing Manually configure your Load-Balancing settings

It performs rolling updates to containers straightaway K8S performs rolling updates to Pods as a whole

Share storage volumes with any other containers Share storage volumes between multiple containers inside the same pods

It uses 3rd party tools like ELK K8S provides in-built tools for logging and monitoring

Architecture of Kubernetes

Architecture of kubernates

Kubernetes follows the client-server architecture where we have the master installed on one machine and the node on separate Linux machines

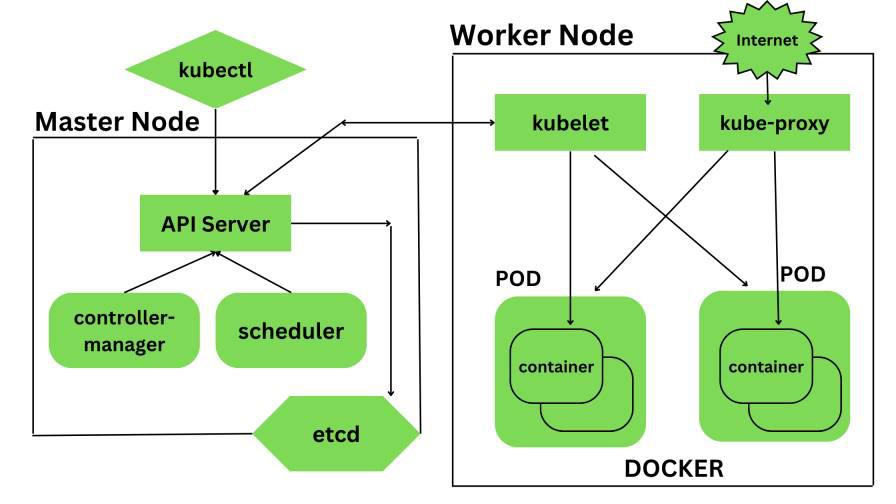
A master and its controlled nodes(worker nodes) constitute a “Kubernetes cluster”.

A developer can deploy an application in the docker containers with the assistance of the Kubernetes master.

there are two nodes in kubernates

1. master node or controller node

2. worker node or controlled node slave node



1. Kubernetes- Master Node Components –

Kubernetes master is responsible for managing the entire cluster, coordinates all activities inside the cluster,

and communicates with the worker nodes to keep the Kubernetes and your application running.

kubernates master node consist of four componants

1. api server

2. controller manager

3. scheduler

4. etcd

1. api server :-

The API server is the entry point for all the REST commands used to control the cluster.

All the administrative tasks are done by the API server within the master node.

We can interact with these APIs using a tool called kubectl.

‘kubectl’ is a very tiny go language binary that basically talks to the API server to perform any operations that we issue from the command line.

It is a command-line interface for running commands against Kubernetes clusters

2. controller manager :-

Also known as controllers.

It is a daemon that runs in a non terminating loop and is responsible for collecting and sending information to the API server.

It regulates the Kubernetes cluster by performing lifestyle functions such as namespace creation and lifecycle event

garbage collections, terminated pod garbage collection, cascading deleted garbage collection, node garbage collection, and many more.

controllers are responsible for the overall health of the entire cluster by ensuring that nodes are up and running all

the time and correct pods are running as mentioned in the specs file.

3. scheduler :-

It is a service in the master responsible for distributing the workload.

It is responsible for tracking the utilization of the working load of each worker node and then placing the workload on which resources are available and can accept the workload.

it schedules these pods accordingly. The scheduler is responsible for workload utilization and allocating the pod to the new node.

4. etcd–

It is a distributed key-value lightweight database.

In Kubernetes, it is a central database for storing the current cluster state at any point in time

and is also used to store the configuration details such as subnets, config maps, etc.

It is written in the Go programming language.

kubernates worker node consist of four componanats

1. kubelet

2. kube proxy

3. pod

4. docker

Kubernetes Worker node contains all the necessary services to manage the networking between the containers,

communicate with the master node, and assign resources to the containers scheduled.

1. kubelet:-

It is a primary node agent which communicates with the master node and executes on each worker node inside the cluster.

It gets the pod specifications through the API server and executes the container associated with the pods

and ensures that the containers described in the pods are running and healthy.

If kubelet notices any issues with the pods running on the worker nodes then it tries to restart the pod on the same node.

If the issue is with the worker node itself then the Kubernetes master node detects the node failure and decides to recreate the pods on the other healthy node.

2.kube proxy :-

It is the core networking component inside the Kubernetes cluster.

It is responsible for maintaining the entire network configuration.

Kube-Proxy maintains the distributed network across all the nodes, pods, and containers and exposes the services across the outside world.

It acts as a network proxy and load balancer for a service on a single worker node and manages the network routing for TCP and UDP packets.

3. pod

A pod is a group of containers that are deployed together on the same host.

With the help of pods, we can deploy multiple dependent containers together so it acts as a wrapper around these containers

so we can interact and manage these containers primarily through pods

4. docker

Docker is the containerization platform that is used to package your application and all its dependencies together in the form of containers

to make sure that your application works seamlessly in any environment which can be development or test or production.

Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.