**Kubernetes Basics and Architecture.**

* What is Kubernetes?
* Kubernetes also know as k8s is an open-source container management tool which automates container deployment, container scaling, and load balancing. It schedules, run and manages isolated container which are running on virtual/physical or cloud machines.
* History of Kubernetes:
* Google developed an internal system called borg (latest called omega) to deploy and manage thousands of google applications and services on their cluster. In 2014, google introduced Kubernetes an open-source platform written in Golang and later donated to CNCF (cloud native computing foundation).
* Features of Kubernetes
* Orchestration (clustering of any number of containers running on different network)
* Autoscaling
* Auto healing
* Load balancing
* Platform independent
* Fault tolerance (node/ pod failure)
* Rollback
* Health monitoring of Pod
* Batch Execution
* What is cluster?
* A cluster is a set of physical or virtual machines that are connected together to run containerized application.
* What is Node?
* Node refers to a physical or virtual machine that forms a part of the cluster. There are two main types of Nodes

1. Master Node (Control Plane)
2. Worker Node

* Master Node:
* The master node is the control plane of the Kubernetes cluster. It manages the cluster’s state, schedules workloads onto worker nodes and handles scaling and failover.
* Worker Node:
* Worker Nodes are where the actual workloads run. They host the pods, which are the basic unit of deployment in the Kubernetes. Each worker node runs a container runtime like docker for managing containers, as well as other necessary Kubernetes components like kube-let and kube-proxy.
* Architecture of Kubernetes

Cluster

Kube-let

Worker Node

Container

Pod

Container engine

Kube-Proxy

ETCD

Kube Controller-Manager

Kube Scheduler

API Server

Control Plane

(Master Node

* Components of Master Node :
* API Server : It is used to communicate between components. It interacts directly with user. It is meant to scale automatically according to load or request load. It is also called the front-end of the control plane.
* ETCD : It stores the cluster’s configuration data and state. It stores the data in the key value form.
* Kube Scheduler : It is the component which is responsible for scheduling pods onto worker nodes.
* Kube controller-manager : It manages various controllers responsible for maintaining the desired state of the cluster.
* Components of Worker Node:
* Kube-let : It is responsible for communication with the API server and managing pods on the node. It also ensures that the container described in the manifest are running and healthy, and reports node status back to the control plane. It provides the pod information to etcd through API server. It runs on port 10255.
* Kube-Proxy : Manages network connectivity to pods in the cluster. It also allocates the IP address to the Pods.
* Container Engine : The software responsible for running containers withing pods such as Docker and containerd.
* Pod : The smallest deployable unit in Kubernetes, consisting of one or more containers that share networking and storage resources. One pod can contain multiple containers but it is recommended to use only one container per pod. Pods have their own IP address but the containers which are stored in the pods don’t have IP address.
* Types of Architecture.
* There are 3 types of Kubernetes architecture.

1. Single master multiple nodes

Node 1

Master

Node 2

1. Single master single node.

Master

Node

1. Multiple master multiple nodes.

Node 1

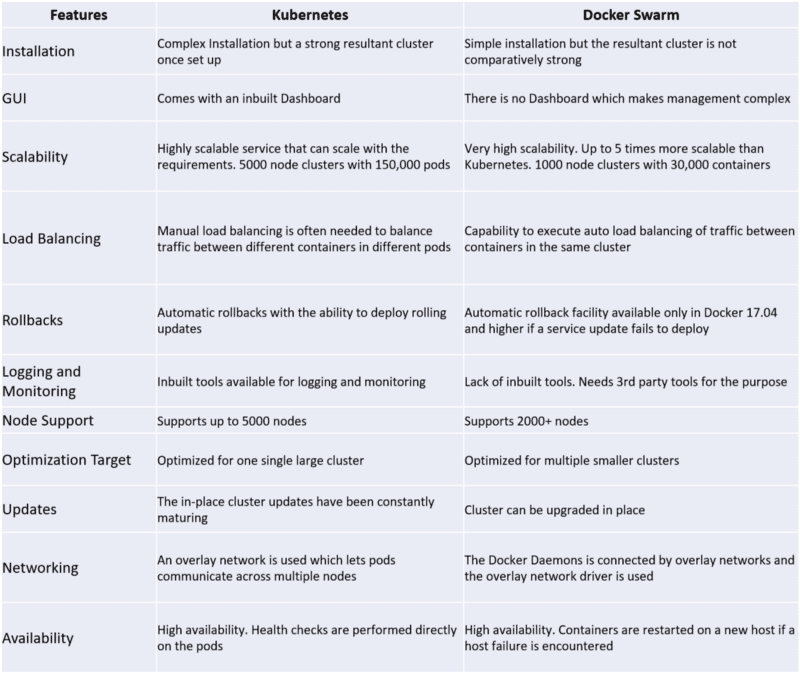
Master 1

Node 2

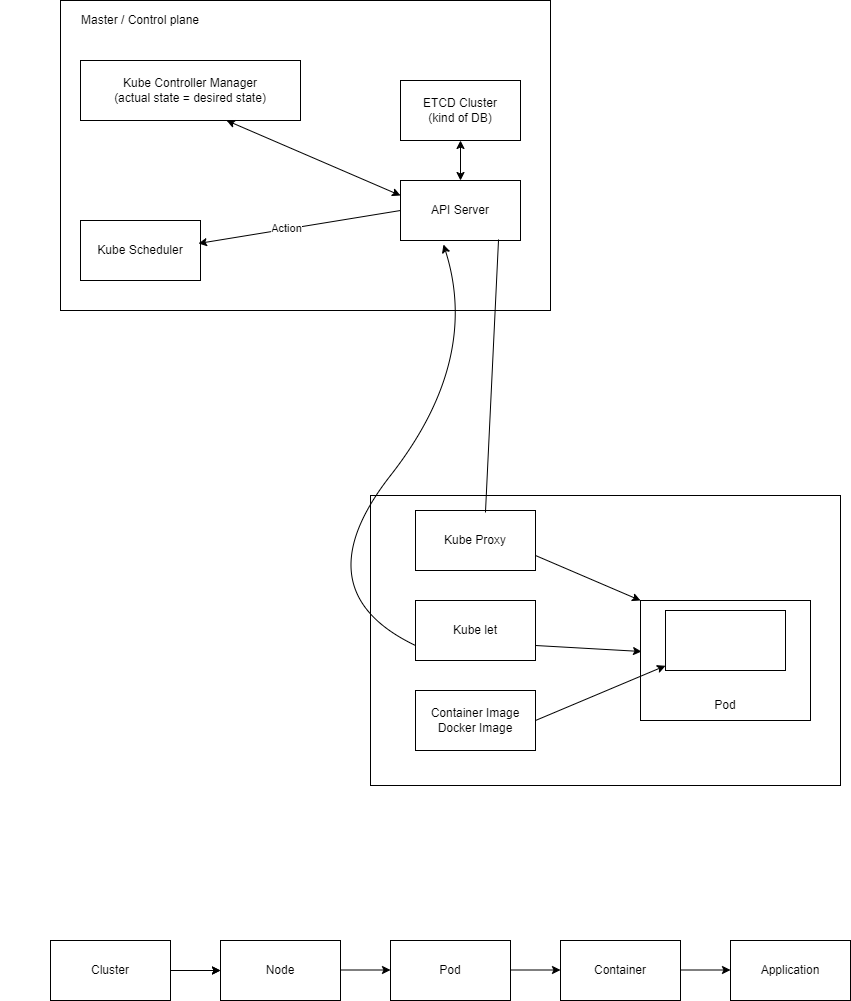
Master 2

Node 3

* Difference between k8s and docker swarm.



* Terms to know :
* Monolithic application : single stone application.
* Microservices : each task is deployed in different container. 2 services connect with each other through API.
* Orchestration tool : container management tool.



* Problems with scaling up the container.
* Can’t communicate with each other.
* Autoscaling
* Load balancing
* Container had to be manage carefully.