* Kubernetes is open source container-orchestration management tool
* Kubernetes is an open source system started by Google to help orchestrate (deploy, scale and manage) containerized applications.
* Kubernetes simplify container deployment, scaling of containers & container load balancing.
* Kubernetes can run on-premises, public clouds(Google,Azure,AWS,etc)
* Kubernetes for automate deployment, scaling and management of containerized applications.
* Kubernetes is not a containerization platform. It is a multi-container management solution.
* It was originally designed by Google and is now maintained by the Cloud Native Computing Foundation.
* Kubernetes aims, to provide a “platform for automating deployment, scaling and operations of application containers across the clusters of the hosts.
* Kubernetes, is a powerful container management tool
* Kubernetes work with a range of container tools include Docker
* Kubernetes automates the deployment of your Docker containers
* Docker is used to isolate, pack, and ship your application into containers; Kubernetes is like the container scheduler for deploying and scaling the application. The two technologies are designed to work together and make app deployment a breeze.

# Why use Kubernetes?

* Kubernetes can run on-premises bare metal, OpenStack, public clouds Google, Azure, AWS, etc.

# Features of Kubernetes

* Automated Scheduling
* Self-Healing Capabilities
* Automated rollouts & rollback
* Horizontal Scaling & Load Balancing

# Kubernetes Disadvantages

* Kubernetes can be an overkill for simple applications
* Kubernetes is very complex and can reduce the productivity

**Kubernetes Basics**

### Cluster:

It is a collection of hosts(servers) that helps you to aggregate their available resources. That includes ram, CPU disk, and their devices into a usable pool.

### Master:

The master is a collection of components which make up the control panel of Kubernetes. These components are used for all cluster decisions. It includes both scheduling and responding to cluster events.

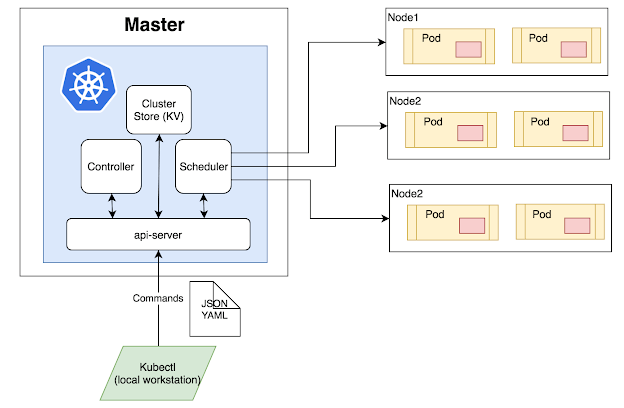
### Node:

It is a single host which is capable of running on a physical or virtual machine. A node should run both kube-proxy, minikube, and kubelet which are considered as a part of the cluster.

### Namespace:

It is a logical cluster or environment. It is a widely used method which is used for scoping access or dividing a cluster.

## Kubernetes Architecture



### Master Node

The master node is the first and most vital component which is responsible for the management of Kubernetes cluster. It is the entry point for all kind of administrative tasks. There might be more than one master node in the cluster to check for fault tolerance.

The master node has various components like API Server, Controller Manager, Scheduler, and ETCD. Let see all of them.

API Server**:**The API server acts as an entry point for all the REST commands used for controlling the cluster.

### Scheduler

The scheduler schedules the tasks to the slave node. It stores the resource usage information for every slave node. It is responsible for distributing the workload.

It also helps you to track how the working load is used on cluster nodes. It helps you to place the workload on resources which are available and accept the workload.

### Etcd

etcd components store configuration detail and wright values. It communicates with the most component to receive commands and work. It also manages network rules and port forwarding activity.

### Worker/Slave nodes

Worker nodes are another essential component which contains all the required services to manage the networking between the containers, communicate with the master node, which allows you to assign resources to the scheduled containers.

* Kubelet: gets the configuration of a Pod from the API server and ensures that the described containers are up and running.
* Docker Container: Docker container runs on each of the worker nodes, which runs the configured pods
* Kube-proxy:Kube-proxy acts as a load balancer and network proxy to perform service on a single worker node
* Pods: A pod is a combination of single or multiple containers that logically run together on nodes

## Other Key Terminologies

### Replication Controllers

A replication controller is an object which defines a pod template. It also controls parameters to scale identical replicas of Pod horizontally by increasing or decreasing the number of running copies.

### Replication sets

Replication sets are an interaction on the replication controller design with flexibility in how the controller recognizes the pods it is meant to manage. It replaces replication controllers because of their higher replicate selection capability.

### Deployments

Deployment is a common workload which can be directly created and manage. Deployment use replication set as a building block which adds the feature of life cycle management.

### Stateful Sets

It is a specialized pod control which offers ordering and uniqueness. It is mainly used to have fine-grained control, which you have a particular need regarding deployment order, stable networking, and persistent data.

### Daemon Sets

Daemon setsare another specialized form of pod controller that runs a copy of a pod on every node in the cluster. This type of pod controller is an effective method for deploying pods that allows you to perform maintenance and offers services for the nodes themselves.