* Kubernetes is a devops development tool
* It is mainly used to automate the deployments of container based applications across the cluster of nodes
* It also manages services of containerized applications using various methods like scalability, predictability, and high availability.

**Kubernetes terminology:**

1. **Pod**: it is the smallest and basic unit of Kubernetes application. This object indicates the processes which are running in the cluster.
2. **Node**:

* it is nothing but a single host which is used to run virtual machine or a physical machine
* A node in the Kubernetes cluster is known as minion.

1. **Service**: A service in Kubernetes is a logical set of pods which work together.

With the help of services users can manage node managing configurations.

1. **Replicaset**:

* A Replicaset in Kubernetes is used to identify a particular no.of pod replicas running at a given time.
* Replica set replaces replication controller because it is more powerful and allows a user to use the set based labels selector

1. **Namespace**: Kubernetes supports various virtual clusters which are known as namespaces. It is way of dividing the cluster resources between two or more users.

**Kubernetes architecture:**

**Kublet:**

* It is a primary node agent which communicates with the master node and that runs on each node within cluster
* It is responsible for managing containers on that node by ensuring they are running according to the provided pod specifications
* It acts as a intermediary between Kubernetes control plane and individual worker nodes to execute containerized applications
* It receives instructions from the control plane and communicates with the container run time to start, to stop, monitor and check container operations on the node

**Kube-proxy:**

* It is a network proxy that runs on each node in a Kubernetes cluster.
* It is responsible for routing traffic between services and pods and for implementing virtual Ip for services.
* It is a Kubernetes agent installed on every node in the cluster.
* It monitors service objects, and the changes made to their end points and translates them into actual network tools inside the node.

Some important Kubernetes interfaces:

1. **Kubectl** - it is a command line tool or interface used to interact with cluster
2. **Kubadm** - it is a tool to initialise and configure Kubernetes cluster including setting up control plane components and connecting worker nodes.
3. **Kublet** – it is an agent running on each node within the cluster responsible for managing pods and containers on that node

Essentially, kubectl is for user interaction, kubadm is for cluster creation and kublet is for managing workloads on individual nodes within the cluster.

**Using Kubectl to create a deployment:**

1. Once you have running Kubernetes cluster you can deploy yours containerized applications on top of it. To do so we create a Kubernetes deployment.
2. The deployment instructs Kubernetes how to create an update instances of your application.
3. Once you have created deployment the Kubernetes control plane schedules the application instances included in that deployment to run on individual nodes in the cluster.
4. Once application instances are created Kubernetes deployment controller continuously monitors those instances.
5. If the node hosting an instance goes down or gets deleted the deployment controller replaces the instance with an instance on another node in the cluster.
6. This provides a self healing mechanism to address machine failure or maintenance

**Kubernetes architecture:**

The architecture of Kubernetes follows the client-server architecture. It consists of two main components

1. Master node (control plane)
2. Worker node (slave node)

**API Server**

The Kubernetes API server receives the REST commands which are sent by the user. After receiving, it validates the REST requests, process, and then executes them. After the execution REST commands, the resulting state of a cluster is saved in ‘etcd’ as a distributed key-value store

**Scheduler**

The scheduler in a master node schedules the tasks to the worker nodes. And, for every worker node, it is used to store the resource usage information. In other words, it is a process that is responsible for assigning pods to the available worker nodes.

**Controller Manager**

The controller manager is also known as controller. It is demon that executes in the non-terminating control loops. The controllers in a master node perform a task and manage the state of the cluster. In the Kubernetes, the controller manager executes the various types of controllers for handling the nodes, endpoints, etc.

**ETCD**

It is an open-source, simple, distributed key-value storage which is used to store the cluster data. It is a part of a master node which is written in a GO programming language.

Now, we have learned about the functioning and components of a master node; lets see what is the function of a slave/worker node and what are its components.

**Worker/Slave node**

The worker node in a Kubernetes is also known as minions. A worker node is a physical

**Kubernetes commands**

kubectl get pods

kubectl get pods –all-namespaces

kubectl get all

kubectl create deployment <name> --image=<image>

kubectl delete pod <pod name>

kubectl describe pod <pod name>

kubectl logs <pod name>

kubectl exec -it <pod name> -- /bin/bash

kubectl scale deployment <deployment name> --replicas=<number>

kubectl port-forward <pod name> <local port>:<pod port>

https://laas.makemylabs.in/wsm/USTDOCKSWARM25022520250225044422?invitation\_id=721dbfbe-9f4c-48fa-930b-398d9e044b82&context=True

ustdockswarm06@mml.local

X5BjEzsKq