==============================**KUBERNETES**============================

Containerization --------→ Docker

Container Orchestration ----------→ Kubernetes, Docker swarm

What is Kubernetes?

\* Kubernetes is an Container Orchestration Engine and Open source Tool for managing Containers.

\* Responsiblities to deploying a containers, scaling, descaling the containers and load blancing .

\* Kubernets not an replacement of Docker but it is an replacement of Docker-swarm.

\* kubernetes is introduced by goolgle but donated to CNCF (Cloud Native Computing Foundation). It is developed using Go / Golang in 2014.

\* Kubernetes released v1.0 in 2015

\* Current stable release v1.32

Features of Kubernetes:

-----------------------------

1) automated Scheduling:

Kubernetes sheduler to launch the containers on cluster nodes based on their resource requirements.

2) Self Healing Capabilities:

kubernetes have a healing capacity if the container or cluster node is dead. To doesn’t advertise to serve the load to an other worker.

3) Automated rollout and rollback :

If any mis configuratin issues occuer in updated version without downtime kubernets can rollout the pass version.

4) Horizondal scalling and load blancing:

The pod resouces request and limit can increase is vertical scalling .

But the pod replica is scale to another worker is called Horizondal Scalling.

5) service discovery and load blancing:

Kubernetes automatically generates the ip address and DNS names to the pod because of the communication didn’t affet with in the cluster.

6) storage Orchestration:

Kubernetes have local storage or you can mount the system storage or use public cloud such as AWS, Azure and GCP or use NFS network file systems ,iscsi storages.

**Kubernetes Architecture:**

---------------------------------

Kubernetes implements a cluster computing Background.Every this working insides a cluster.in that cluster one or more nodes act as ‘master‘ rest of nodes working as ‘nodes’

Cluster ---→group of node servers.

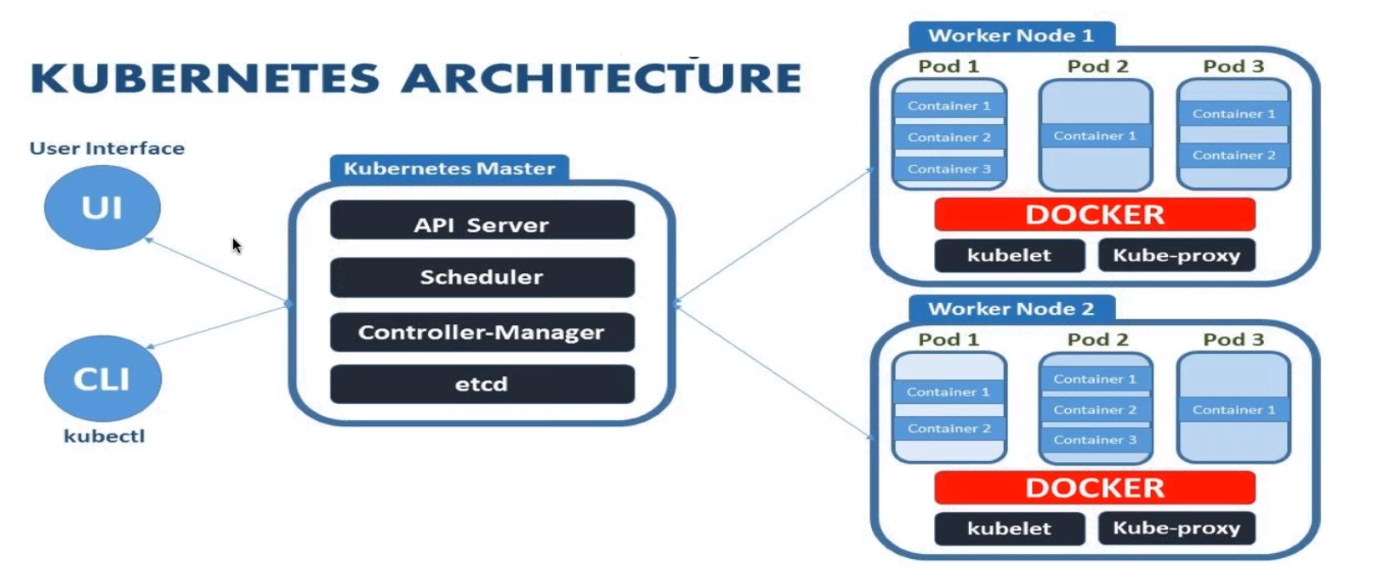


Fig Kubernetes Architecture.

Kubernetes Components:

-------------------------------

1) Mater\_Node:

\* the Mater node called it as Control-Plane also.

\*Mater node to manage the worker nodes and pod amanage ment if the the pod Is down to scale up the pod or assign thr pod to another node and etc…

1.1) Kubectl(CLI) :

\* Kubectl is the command configuration tool it is intract with api server.

\* it present only in control-plane server.

1.2) API Server:

\* api server to validate the request and authenticate the request.

\* ex (kubectl run nginx-test –image=nginx –port=80) to validate the command is valid or not valid and have to authenticate this node have permission to execute this command and else.

1.3) ETCD :

\* ETCD is the simple key value store.

\* ETCD is Stores the data about kubernetes clusters . It means to store the how many pods are running and where its running and etc…

1.4) Scheduler:

\* Scheduler to schedule the pods to the unscheduled specfic node with help of Kubelet.

1.5) control Manager:

\*control manager runs the controllers in background .which run different tasks in kubernetes.

1.5.1) Node controler:

\*it is the responsible for moniter and notify the node is downtime.

1.5.2) Replication controler:

\* It maintaines the number of pods . It control homant pods in running somwhere in the cluster.

1.5.3) Endpoint controler:

\* it jons the services and pods together

1.5.4) Repilcaset controller:

\*It ensure the number of replication pods runinng alltime

1.5.5) Deployment Controler:

\* It provides a declarative update and replicasets

1.5.6) Daemonset controller:

\* it ensure the node run a copy of specfic pods

1.5.7) Job controller:

\* it is the supervisor process for pods carrying out of jobs.

2) Worker\_Node:

The worker node where acctually to runs application pods in that cluster.

2.1) Kubelet:

\* kubelet is primary node agent it present all nodes in the cluster.

2.2) kube proxy:

\* kube proxy maintain the network rules inside the clusters and outside the cluster for an effective network communication. It also present in all nodes in the cluster.

3) container runtime (container-D) :

\* the container Runtime means to to create a images run as the container using container runtime . “Kubernetes Deprecated the Docker” but docker files or docker can ude by the container-d becaue of “all containerazation software have some standars (OCI --→Open Container Initiative) .

\* container runtime(software) also peresent in all nodes in that cluster.

**Installation OF Kubernetes:**

**------------------------------------**