KUBERNETES

- It is an open source container orchestration tool
  - ↳ coordination & management of multiple computer systems
- Developed by Google
- Helps us manage containerized applications in different deployment environments

Functionalities

- High availability (Always accessible by the users) → No downtime
- Scalability or high performance (It loads fast & users have a very high response rate from the application)
- Disaster recovery (Faces problems like data is lost, servers explode the server should have mechanism to pick up the data & to restore it to the latest state)
- The containerized application can run from the latest stayed after the recovery

Kubernetes Basic Architecture

- Made up of atleast 1 master node then connected to have couple of worker node (slave node)
- Each node had a kublet process running on it
- Kubelet is actually a kubernetes process that makes it possible for the cluster to communicate to each other & execute some tasks on nodes
- Each worker node has docker containers of diff applications deployed on it depends on how the work load is distributed by the master node.
- Worker node where actual work is happening, Application is running.

Master node : Runs & manages the cluster properly

API server — Entry point to the Kubernetes cluster

↓

A container.

- ✓ UI — Kubernetes dashboard
- ✓ API — Using scripts & Automating tech
- ✓ CLI — command-line

Controller Manager — ✓ Overview of whats happening in the cluster

/something needs to repair or may be container died or restarted

Scheduler — Basically responsible for scheduling containers on diff nodes

etcd — Basically holds at any time the current state of the Kubernetes cluster. All configuration data and all the status data of each node, each container

- Backups is actually made from etcd snapshots because you can recover the whole cluster state using that etcd snapshot
- If we loose a master node we will not be able to access the cluster anymore so maintain atleast two master inside your Kubernetes

master node is down the cluster continues function smoothly because you have other master available

Pod

- ✓ Pod is abstraction layer over container
- ✓ Pod is the smallest unit user will configure
- ✓ Each worker node we will have multiple pods.
- ✓ Each pod has multiple containers
- ✓ Each pod has its own IP addresses, own server
- ✓ Communication is between internal IP addresses
- ✓ pods are ephemeral (dies frequently)

✓ send the configuration request to the server to API — Main entry point to the cluster — deployment

✓ The request has to be in yellow format or JSON format.

~~send~~ ✓ template, Blueprint for creating pods in specific configuration

✓ configuration request in Kubernetes is declarative