

*

Basic Architecture of K8S -

K8S supports: (v1.16)

No more than 5000 nodes

150000 total pods

300000 total containers

100 pods per node

Master node -

- responsible for managing the cluster
- Monitors nodes & pods in a cluster
- When a node fails, moves the workload of the failed node to another worker node

4 components of master node

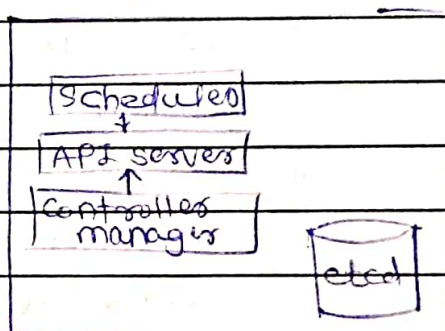
- API server - for all communications

(JSON over HTTP API)

- scheduler - schedules pods on nodes

- Controller managers - runs controllers

- Etcd - open source, distributed key-value database from CoreOS



Master

① API Server

- APIs allow application to communicate with one another
- It is the front-end for the K8S control plane
- Exposes API for almost every operation
- The users, management devices and command line interfaces all talk to the API server to interact with the K8S cluster
- users interact with the API using a tool called kubectl
- kubectl is a command line utility to interact with

K8S API.

- kubectl is a Go language binary.

② Scheduler -

- Schedules pods across multiple nodes
- component on the master that watches newly created pods that have node assigned, and selects a node for them to run on
- The scheduler obtains from etcd, via the API Server, resource usage data for each worker node in the cluster.
- ** - Scheduler gets the info. for hardware configuration from configuration file and schedules the pods on nodes accordingly.

③ Control Manager -

- This is a component on the master that runs controllers
 - kube-controller-manager
 - Node controller
 - Replication controller
 - Endpoints controller
 - service account & token controller
 - cloud-controller-manager
- Responsible for overall health of the cluster
 - Ensures nodes are running all the time
 - correct no of pods are running as per spec file
- * - Basically, each controller is a separate process, but to reduce complexity, they are all compiled into a single binary & run in a single process.

Node controller - Responsible for noticing & responding when nodes go down

Replication controller - Responsible for maintaining the correct no. of pods for every replication controller object in the system

Endpoints controller - populates the endpoints object (i.e. joins services & pods)

Service account & token controller - create default accounts and API access tokens for new namespaces

(4) etcd -

- open source, distributed key-value databases from Core OS

- consistent & highly - available key value store used as K8s backing store for all cluster data

- single source of truth for all components of the K8s cluster.

- * - out of all the master components, only the API server is able to communicate with the etcd data store

- etcd can be part of the K8s master or it can be configured externally.

