Kubernetes

Kubernetes is a container management technology developed in Google lab to manage containerized applications in different kind of environments. It is an open source system which helps in creating and managing containerization of application.

I also learned about flowing terms of Kubernetes:

* Master- It’s the main manager of cluster.
* Node- The actual work is performed by nodes on which we deploy pods.
* Pods- Atomic Unit of Scheduling.
* Services- Provides Stable Network/IP.
* Deployments- RC+, rolling updates, roll backs, etc.
* Controller- Various controller like Node Cont. , Endpoint’s Cont. , Namespace Cont., etc.
* API Server- Contains Information like API Version, Kind, Meta Data, Speci, Replica.
* Scheduler- Responsible for distributing the workload.

Master

It has following components:

* Cluster Store- Cluster state and configuration
* Scheduler- It schedules the tasks in the cluster
* API Server- Contains Information like API Version, Kind, Meta Data, Speci, Replica
* Controller- There are various controller like Node controller, End points controller, Namespace controller, etc.

Nodes

It contains Kubelets, Container Engine , Kube Proxy

Kubelet

* It watch API Server.
* Initiate Task.

Container Engine

* It pulls Images.
* It Start/Stop Container.

Kube Proxy

* Manages 1 I/P per POD.

Actual & Desired State

* Let’s get this by an eg: If there are n Desired states and 1 node fails then Actual node are n-1.So, Kubernetes shifts pod on another surviving node.
* Kubernetes constantly checks actual and desired nodes.

Pods

* Containers runs inside Pods.
* It can have n no. of containers.
* It also may contains Network Stack , Kernel namespaces.
* All containers in pod share the pod environment.
* All containers share same IP.
* Generally we should put 1 container in 1 pod.
* If there is tight coupling then we may put multiple container in same pod.
* Lifecycle- Pending-Running-Succeeded/Failed
* If a pod fails then another pod takes it’s place.

Replication Controller:

* Replica of same pod configuration.

Services

* IP change every time we push an update of pods.
* So, In an example Front End & Back End Pods.
* If we have 4 FE & 2 BE Pods. Then there would be connection between each FE pod with that of each BE pods. So, Service helps us by providing Stable IP & DNS name which connects both FE & BE pods. As all are just linked with Service & not with each other which helps us when we loose any pod and if new pod arrivers with new IP , Service update links that new one.

Labels

* Helps to tag Info. to an entity.
* E.g.: Pod, Back End, Front End, Version, etc.