

**Role of Master Node:**

* Kubernetes cluster contains Containers running on bare metal/ vm instances/ cloud instances/ all mix.
* Kubernetes designates One or more of these as a master and all others as workers.
* The master is now going to run set of Kubernetes processes. These processes will ensure smooth functioning of cluster, this process are called as “control plane”.
* We can use multi master for high availability. Master run control plane to run cluster smoothly.

**Components of control plane:**

* Kube-API-Server
* Control Manager
* Kube-Scheduler
* Etcd-Cluster

**Kube-API-Server:**

* This API server interacts directly with users. (i.e. We apply .ymal or .json manifest to kube-api-server)
* These kube-api-server is meant to scale automatically as per load.
* It is the front-end of Control Plane.
* It directly communicates with other components of Control Plane and Nodes.
* It is the mediator to communicates all components of Control Plane with each other.

**Controller Manager:**

* Controller manager makes sure that actual state of cluster matches the desired state.
* There are two possibility choices for controller manager
* If Kubernetes on cloud, then it will be cloud controller manager.
* If Kubernetes on non-cloud, then it will be cube controller manager.

**Components of cloud controller manager on master that run controller:**

* **Node-Controller:**
  + For checking the cloud provider to determine if node has been deleted in the cloud after its top responding.
* **Route-Controller:**
  + It is responsible for setting up network routes on our cloud.
* **Service-Controller:**
  + It is responsible for creating, updating and deleting load balancers on our cloud against services of type load balancer.
* **Volume-Controller:**
  + It is used for creating attaching and mounting volumes and interacting with the cloud provider to orchestrate volume.

**Etcd-Cluster:**

* Etcd cluster stores metadata and status of cluster.
* Etcd is consistent and highly available to store key-value.
* It is source of touch for cluster state. (gives info about state of cluster)
* Etcd has following features:

**A Fully Replicated:** The entire state is available on every node in the cluster.

**B Secure:** Implements automatic TLS with Optional Client-Certificate authentication.

**C Fast:** Benchmarked at 10,000 writes per second.

**Kube-Scheduler: (action)**

* When users make request for the creation & Management of pods kube-scheduler is going to take action on these requests.
* It handles POD creation & management.
* Kube-scheduler match/ assign any node to create and run pods.
* A scheduler watches for newly created pods that shave no node assigned. For every pod that the scheduler discovers the scheduler become responsible for finding the best node for that pod to run on.
* Scheduler gets the information for hardware configuration from configuration files and schedule the PODs on node accordingly.

**Nodes:**

Node components run on every node, maintaining running pods and providing the Kubernetes runtime environment.

**Kubelet:**

* It is the agent running on the node.
* It doesn't manage containers which were not created by Kubernetes.
* It listens to Kubernetes master (e.g. -pod creation request)
* It uses Port 10255.
* It sends success/ foil report to master.

**Kube-Proxy:**

* kube-proxy is a network proxy that runs on each node in our cluster, implementing part of the Kubernetes Service concept.
* kube-proxy maintains network rules on nodes. These network rules allow network communication to your Pods from network sessions inside or outside of your cluster.
* It assigns IP to each pod.
* It is required to assign IP addresses to pods (it assign dynamic Ip’s).
* Kube proxy runs on each node & this make sure that each pod will get it's own unique IP address.

**POD:**

* It is the smallest unit in K8s.
* POD is a group of one or more containers that are deployed together on the same host.
* In K8s, the control unit is POD not a container.
* It Consist of one or more tightly coupled containers.
* POD runs on node, which is control by master.
* K8s only knows about PODS, it does not know about individual containers.
* We cannot start container without a POD.
* One POD usually contains one container.