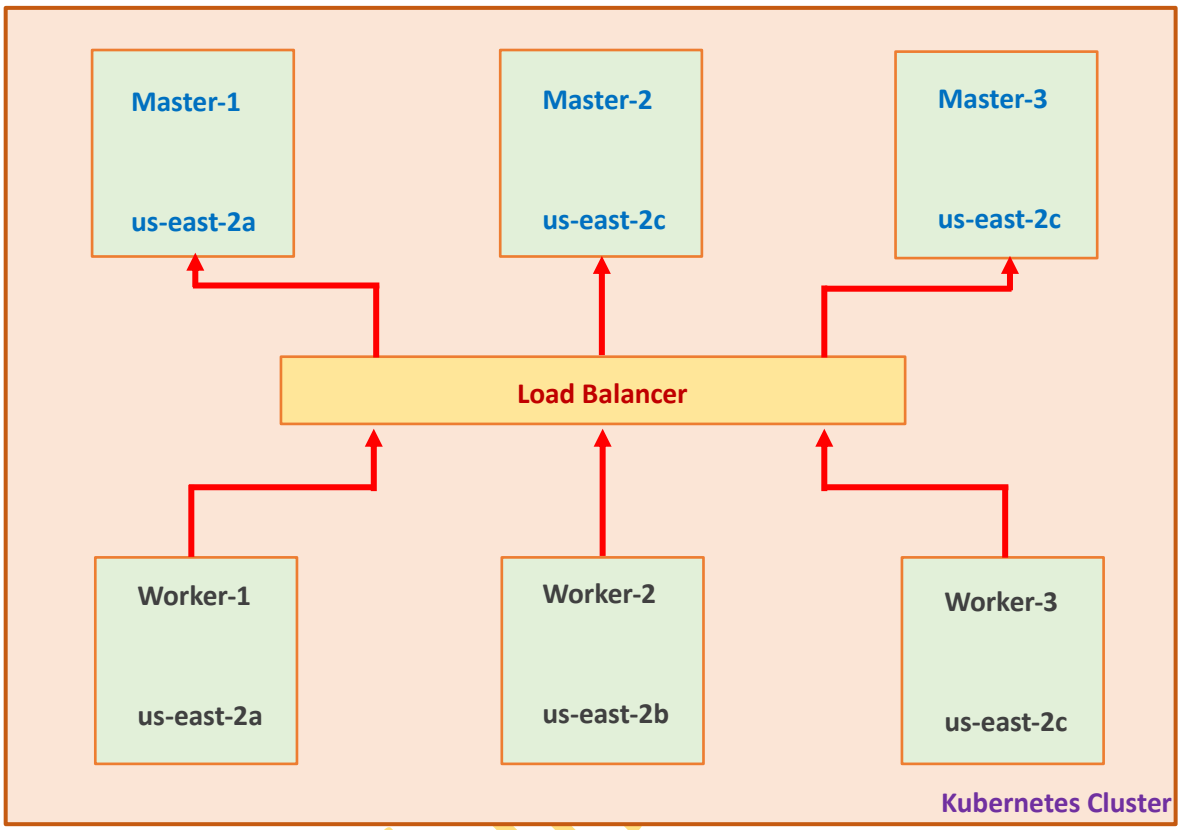
Kubernetes Architecture



In the above diagram I have shown a generalised diagram for HA Kubernetes cluster which has three master nodes and three worker nodes (each one of them in us-east-2a, us-east-2b and us-east-2c regions). HA Kubernetes cluster must contain multiple nodes of Master and Worker along with multiple replicas of the Applications which will run on the worker nodes. So that if any node or pod fails another will take its place.

Each of the worker nodes will connect with Master (API Server) using the Load Balancer as shown in the diagram drawn above.

Components of Kubernetes Cluster is mentioned above. For better understanding I had divided the components in two sides, Master side components and Worker Side components.

Master Side Components

1. API Server: - Worker will communicates to Master through the API Server. It acts as a first point of contact to the Master.
2. Etcd: - Etcd stores the cluster data. As for example, whenever a user runs the command kubectl get nodes to get the information about the nodes present in the Kubernetes Cluster then API Server will connect with the etcd and get the information about it and will present the same information on the console to user.
3. Controller Manager: - Controller Manager is responsible for maintaining the desired number of pods or nodes. As for example if a pod gets terminated the controller manager is responsible to spin up a new pod to maintain the desired number of pods.

4. Scheduler: - Scheduler or Kube-Scheduler decides on which node the pod is going to be scheduled.

Worker Side components

1. Kubelet: - Kubelet is responsible for the creation of the pod. Kubelet communicates with the container runtime.
2. Container Runtime: - Container runtime (docker) is responsible for running the application container inside the pod.
3. Kube-Proxy: - Kube-Proxy is responsible for networking inside the Kubernetes cluster. Kube-Proxy runs as Daemonset in the Kubernetes Cluster.

Whenever a request comes to API Server to schedule the pod then API Server will communicate with Scheduler and ask scheduler to create the pod. Then Scheduler will communicate with the kubelet which runs as a service on Worker node. Kubelet creates the pod and after creation of pod the kubelet will inform to API Server that pod has been created on the Worker node. The API Server will make its entry into the Etcd, so that next time a request will come to show the pods then API Server will communicate with the Etcd and show the pods.

Ritesh Kumar Singh