**Kubernetes consists of below components :**

1. **API server** - listen to kubectl command and other requests.
2. **etdc** - distributed database to manage master and worker node.
3. **Scheduler** - responsible for assigning work to master and worker nodes, looks for new container and assign them to worker node.
4. **Controller** - Brain of the kubernetes, when nodes/pods/cluster etc goes down, controller comes into play and start

starts the new one if required.

1. **Container Runtime** - underline software on which container applications run, in most of the cases it Docker.

But there are some other options as well like rkt etc.

1. **Kubelete -** These are the agent running on each container to insure the each node is running as expected.

**Master Node components** : API-Server, ETDC, Scheduler, Controller

**Worker Node Components** : Container Runtime, Kubelete

Some Important kubectl Commands :

1. kubectl get pod <pod-name> -o yaml > pod-definition.yaml

This command extract the yaml definition file from the pod.

1. kubectl edit pod <pod-name>

this command is used to edit the pod when no pod-definition file is provided to edit.

ReplicationController vs ReplicaSet :

Purpose of both is same, but replicaSet is the new recommended way for creating the and maintaining the replicas.

Some important shorthand commands

**Create an NGINX Pod**

kubectl run --generator=run-pod/v1 nginx --image=nginx

**Generate POD Manifest YAML file (-o yaml). Don't create it(--dry-run)**

kubectl run --generator=run-pod/v1 nginx --image=nginx --dry-run -o yaml

**Create a deployment**

kubectl create deployment --image=nginx nginx

**Generate Deployment YAML file (-o yaml). Don't create it(--dry-run)**

kubectl create deployment --image=nginx nginx --dry-run -o yaml

**Generate Deployment YAML file (-o yaml). Don't create it(--dry-run) with 4 Replicas (--replicas=4)**

kubectl create deployment --image=nginx nginx --dry-run -o yaml > nginx-deployment.yaml

**Save it to a file, make necessary changes to the file (for example, adding more replicas) and then create the deployment.**