**1.Introduction-Architecture-kubeadm\_Cluster**

--- I, control plane (Master Nodes)

2, worker plane (worker nodes)

**Kubernetes architecture**

--- scenario – I want to deploy ms application with 6 containers. I need to contact the kubernetes admin for this.

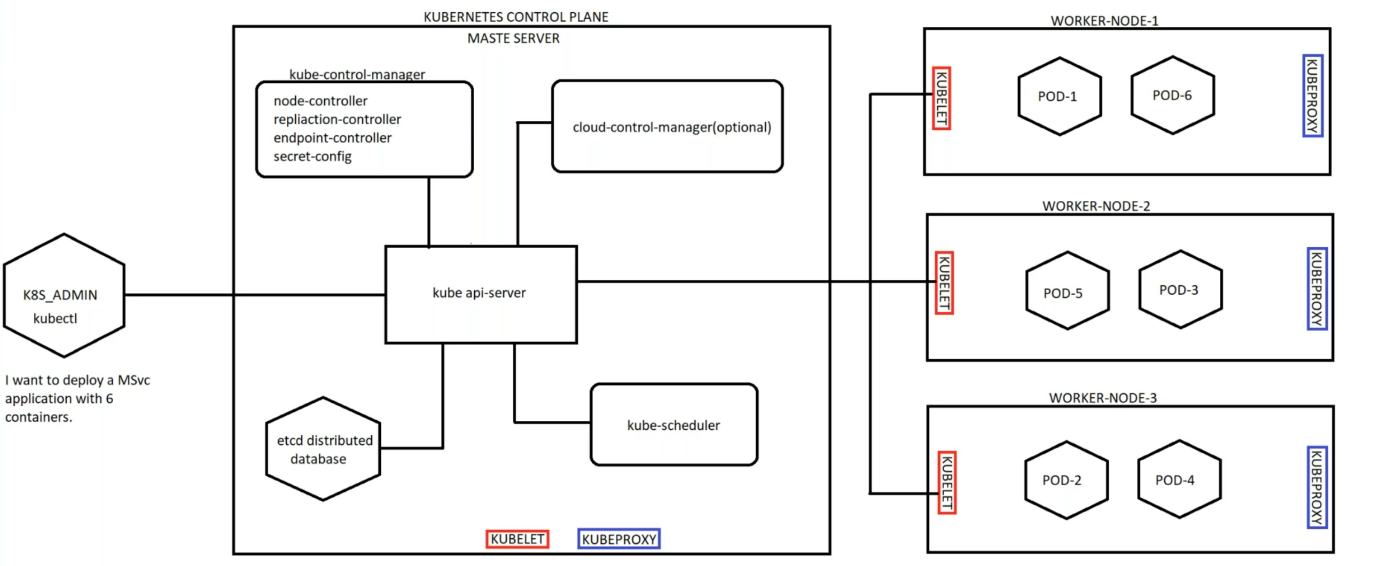
**Kube api server**

--- kubernetes admin will tell the kube api server that I want to create a ms application with 6 containers using kubectl command.

--- the kube api server is management server, after receiving request form admin, kube api server store that information in distributed database,

--- **you have tasked to deploy an application with 6 containers, who is actually deploying those containers or pods…?**

Kubelet.



**Kube schedular**

--- the kube schedular continuously ping the kube api server and it will ask kube-api server that if I have any work.

--- after receiving the request from kube admin, the kube api server will write that configuration to etc database and the kube api server will tell kube schedular that we have to create an application with 6 containers and we have 3 nodes. So, please go and check for the cpu and memory and restrictions of our nodes.

--- the kube schedular will go and check all the nodes for cpu and memory…etc and it will report back to kube api server with information like, this much of resources available in our nodes. So, you can schedule 2 pods in node1, 3 pods in node2 and 1 pod in node3.

**Kubelet**

--- after receiving the required information form kube schedular, kube api server convey the pods scheduling information to Kubelet and the Kubelet deploy containers on respective pods.

**Kube porxy**

--- kube proxy is a network porxy that runs on each node in your cluster, kube proxy maintain network rules on nodes. These network rules allow network communication to your pods from network session inside or outside of your cluster.

--- **note** – all the kubernetes will run as a pod except Kubelet. The Kubelet run as service, service means programme. Kubelet run as a service.

**Creating pod**

**# want to create pod**

--- kubectl run testpod1 --image=nginx:latest

**# expose pod using nodeport service.**

--- kubectl expose pod testpod1 --port=8000 --target-port=80 –type=NodePort

**# list service**

--- kubectl get svc