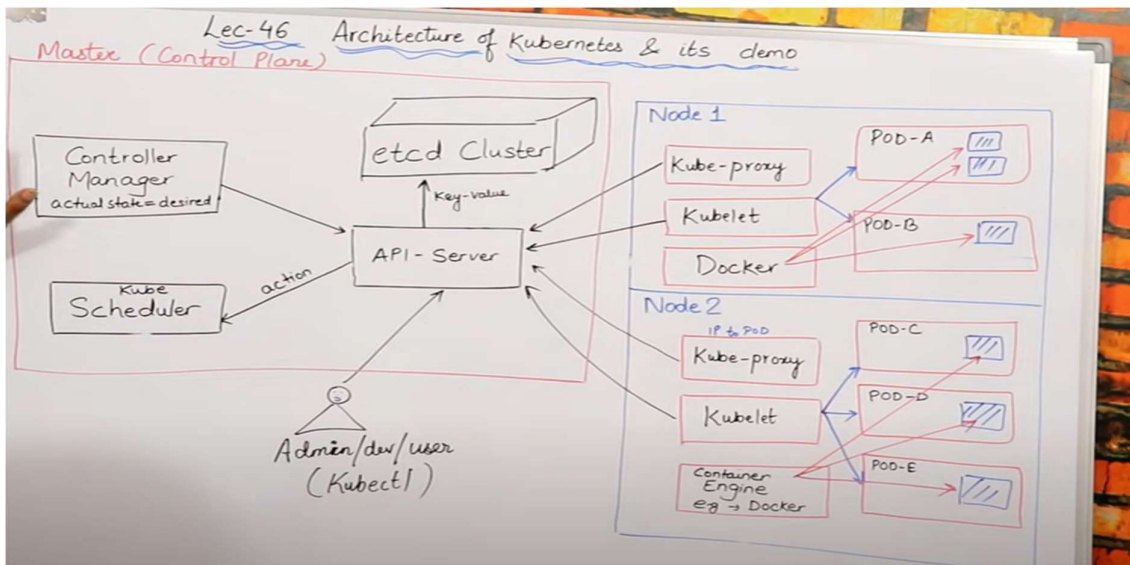
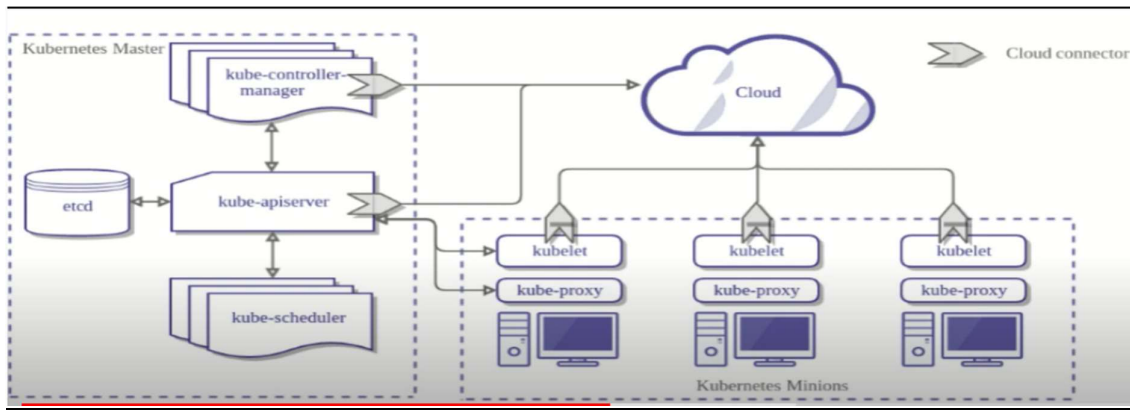
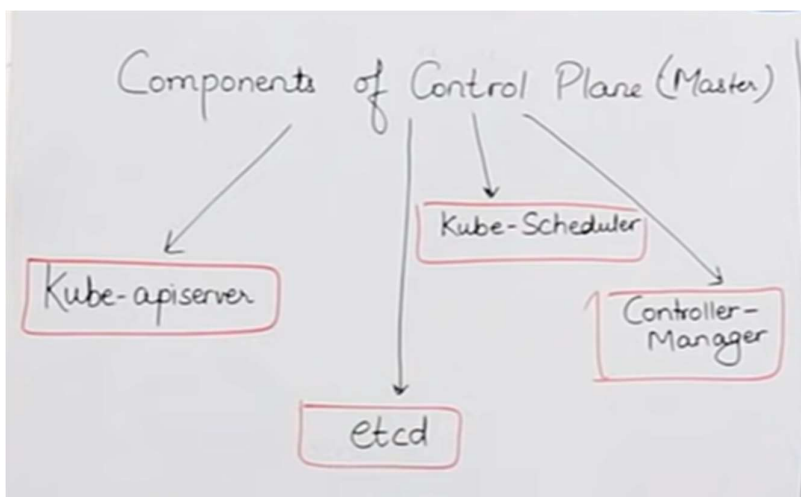


## Architecture of kubernetes & Demo



### Master Control Plane



## → API SERVER

- Agar kisi ne kuch bhi request kiya hai toh pehle vo api server kein pass jayega api server unki request accept karta hai, aur unki request leke vo aage concern person kein pass forward karta hai, hum diagram meien dekhenge sab api server par hi request kar rahe hai.
- Node humesha api server sein hi baat karta hai kyu ki vo front facing rehata hai, matlab node kein andar jo kubelet hota hai vo hardum api sein contact karta hai.
- Agar kisi user ko bhi baad karna haia toh vo api server sein hi baat karega aur kisi sein direct jaa ke baat nahi karega. Chahe vo controller ho chahe scheduler ho

① Kube-api-server (For all Communications)  
→ This api-server interacts directly with user (i.e we apply .yaml or json manifest to Kube-apiserver)  
→ This Kube-apiserver is meant to scale automatically as per load.  
→ Kube api-server is front-end of Control-plane

## → Control Manager

- Control manager ka kaam hai ki agar aapne 4 container ki demand ki hai apane pod kein andar toh 4 aapko 4 container milani chahiye yein control manager
- Control manager kaam karta hai agar mene kisi bhi tareeke sein koi bhi request kar raha toh vo sai tareeke sein humare pass pahuch rahi hai ya nahi cec karta hai aur uspar kaam karta hai.
- **ACTUAL STATE = DESIRED STATE** ye kaam karta hai humari actual state k desired state banane mein.
- Agar hum apana kubernetes pura cloud par chala rahe toh vaha par **cloud-control manager** naam sein eak package hoga ,
- Agar hum kubernetes cloud par nahi hai kisi physical server par hai toh vaha par humein **kube-controller manager** kein naam sein eak package dikh jayega.
- **Actual state ka** matlab hai ki present mein humare pass kitane container hai ya kitane pod hai
- **Desired state ka** matlab hai ki agar humko kitane container aur pod ki jarurat hai vo requirement humare desired state mein rehata hai.

Controller- Manager  
→ Make sure actual state of cluster matches to desired state  
Two possible Choices for Controller Manager  
① If K8s on cloud, then it will be Cloud-controller-manager

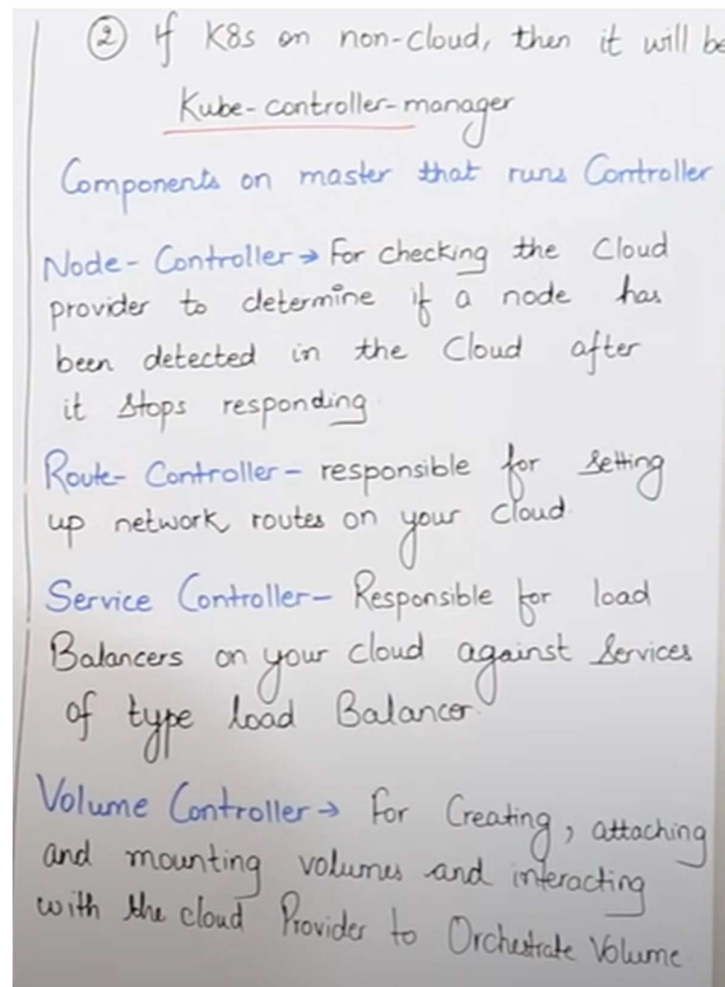
**Components of Control manager :-** control manager ke in andar kuch components hote hai jo ki control rakhta hai.

(1): **Node controller** = Node ke baare mein sari jankari rakhta hai Node controller

(2) **Route Controller** = node se pod tak jane ka ya jitana bhi routing ka kaam hoga vo humara route controller handle karta hai.

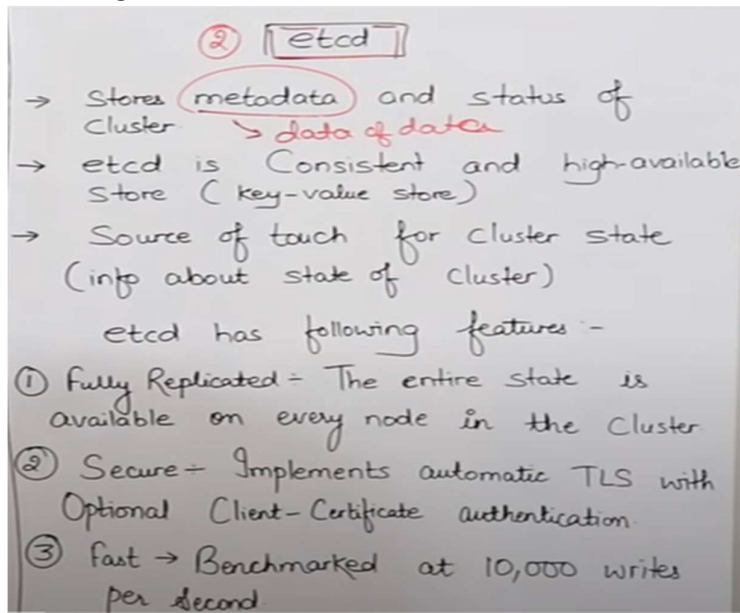
(3) **service controller** = load balancer ke related jitana bhi kaam hota hai vo humara service controller dekhata hai.

(4) **volume controller** = Volume ke related humara jitana kaam rahega vo volume controller handle karta hai.



## → ETCD CLUSTER

- ETCD ek tareeke ka database hai storage hai, etcd batata ha ki humare pod mein kitane container hai aur agar jitane container ki requirement hai usase kaam container agar pod mein hai toh **ETCD** control manager ko batayega ki is pod mein itane kam container hai toh ye problem solve karo.
- ETCD kein andar humare pure pod ki information padi hoti hai ki kitane container hai, kya kya hai vo sab information humare **ETCD** kein pass hoti hai.
- ETCD ko kewal **API SERVER** access kar sakta baki koi direct ETCD ko access nahi kar sakta. Agar kisi ko karna hai toh Api server par request karega then vo **Api server** jaake uski request ko leke jayega etcd kein baad aur jo control manager chahata hai vo **api server** vaha sein uthayega aur laake control manager ko dedega.



## → Kube scheduler

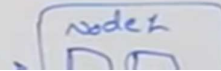
- kube scheduler humari desired state ko pura karane ka kaam karta hai matlab agar mere pass 8 container ki requirement but mere pass 6 hi container hi present mein hai toh actual state mein hai toh kube scheduler new 2 container banayega aur requirement fullfill karega,
- kube scheduler ko control manager information deta hai ki ye karo aur kube scheduler usko perform karta hai, matlab control manager api server ko batata hai aur phir api server kube scheduler ko batata hai.
- Agar koi user manifest file mein ye likhata hai ki mujhe pod bana kar do toh vo kaam umara kube scheduler karta hai.
- Pod creation sein jitani request hoti hai vo ab kube scheduler handle karta hai, aur unko manage bhi karta hai .
- Scheduler khud decide karta hai ki kis node par usko pod banana hai , kis node par kitani jagah khalli hai us hisab sein vo us node par jaakar pod bana deta hai.

### ③ Kube-scheduler (action)

- When users make request for the Creation & Management of Pods, kube-scheduler is going to take action on these requests
- Handles Pod Creation and Management
- Kube-scheduler match/assign any node to Create and run pods.

→ A Scheduler watches for newly Created Pods that have no node assigned. For every Pod that the Scheduler discovers, the Scheduler becomes Responsible for finding best node for that pod to run on.

→ Scheduler gets the information for hardware Configuration from Configuration files and Schedules the Pods on nodes accordingly.

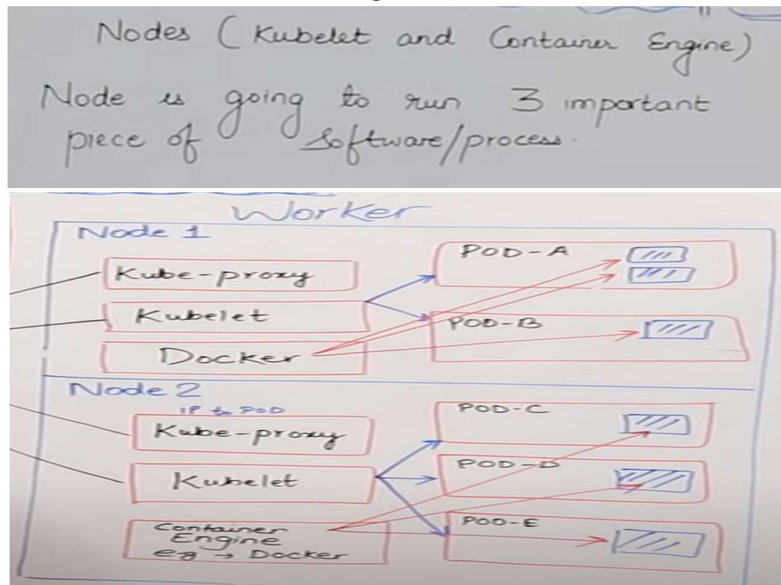


#### → Admin/dev/user

- Hum jo code execute karne kein likhate hai usko **manifest file** kehate hai, aur manifest file kein andar hum **json** format ya phir **yaml** format mein code likhate hai
- Hum jo bhi cheej chahate hai usko yaml file ya json file mein likh de, aur us file ko aur code ko kehate hai manifest
- Manifest file user likhata hai aur usko bhejata hai api server kein pass or api server read karta hai aur user ki demand dekhata hai aur phir api server **control manager** ko batata hai aur **control manager** phir uspar kaam karate hai

## Worker control plane

➔ Worker node ko bahut sein log minion bhi bolate hai

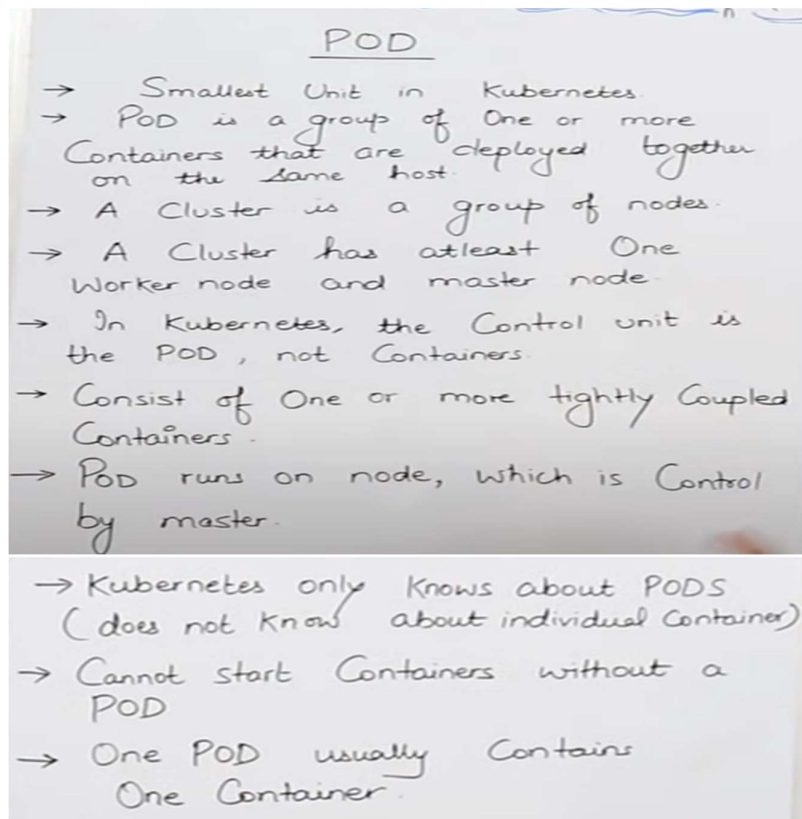


### Pod: -

- Kubernetes kabhi bhi container sein baat nahi karta vo hardum pod sein baat karta hai
- Smallest unit of the kubernetes is pod
- Pod hum node kein andar banate hai
- Koi bhi 2 pod aapas mein baat nahi kar sakte
- Hum pod kein andar bahut sein container rakh sakte hai but achaa ye hota hai ki hum kewal eak hi container rakhe pod mein, aisa isliye kyu ki agar pod mein eak bhi container fail huva toh automatically baki us pod mein jitane bhi container honge sab fail ho jayenge.
- Kubernetes mein kabhi bhi container kein pass Ip address nahi hota hardum pod ko assign hota hai ip address.
- Container ka khud ka koi ip address nahi hota kubernetes mein pod ka ip address hota hai, pod kein kitane bhi containers ho but unki koi ip address nahi hogi hardum ip address pod ko assign hoti hai .
- Agar humara pod fail hogaya hai toh humare pod ki ip bhi band ho jayegi aur jab naya pod banega toh humare naye pod kein sath new ip bhi generate ho jayegi. Humara kaam pura hoga jaise pehale thaa bas ip change ho jayegi.
- Pod kein andar hum containers ko rakhate hai

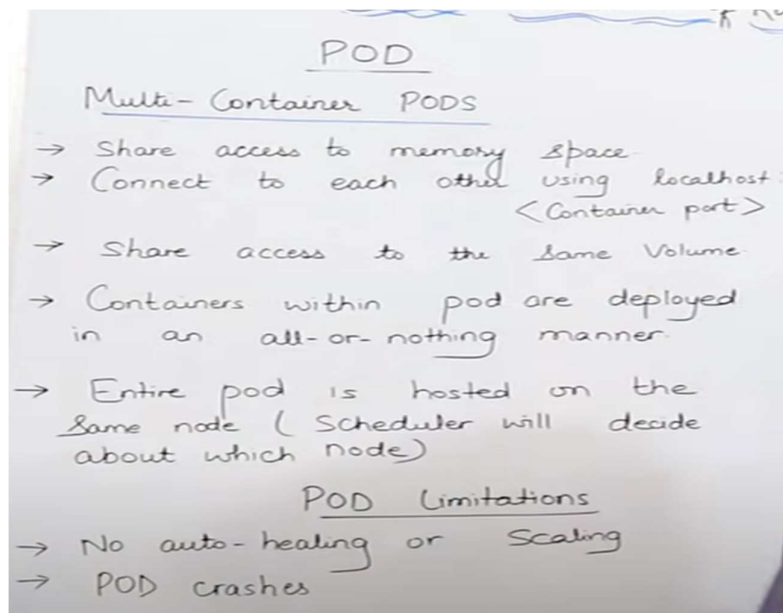


- Bina pod banaye hum container create nahi kar sakte .
- Pod eak prakar ka baksha hai kubernetes kein andar jiske andar hum apane containers ko rakhate hai ya create karte hai.



## ➔ Multi-container pods

- Agar mere pods mein multiple container hai toh un sab container ki volume aapas mein shared hogi.
- Pod kein andar jo multiple container honge vo aapas mein locally eak dusare se in connect honge matlab localhostkein through.



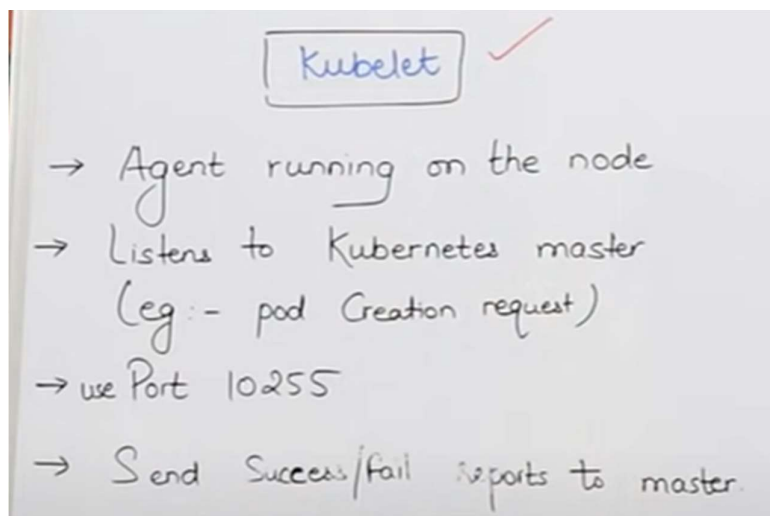
- ➔ By default pod mein auto scalling nahi hoti but agar kubernetes ki higher level objects ko add karke hum uski fuctionality ko badha sakte hai aur scalling vale feature add kar sakte hai.
- ➔ Agar hum replication set karte hai toh usase hum auto-scaling aur auto-healing kar payenge.
- ➔ **Versioning** ka matab hai ki hum version create kar rahe har baaar kuch jab naya change karte hai toh uska new version bana dete hai, hum deployment naam ka object use karte hai iske liye
- ➔ **Rollback** sein hota hai ki current version sein peechale version par jana , hum deployment naam ka object use karte hai iske liye.
- ➔ **Service** naam ka object add karne sein hum apane **pod** ko eak static Ip dekh sakte phir vo ip huar change nahi hogi.
- ➔ **Volume** naam ka object hum use karte hai data store karne kein liye jaise agar in the case mere container kharab hogaya aur mere pod bhi kharab hogaya toh mera data bhi chala jayega but hum eak **non-ephemeral storage** banate hai jiske andar hum apana data store karte aur vo storage humare node kein bahar rehata hai aur iske andar humara data store rehata hai long term tak bhale hi humare container aur pod kharab kyu na ho jaye.

Higher Level Kubernetes Objects-  
Replication Set → autoScaling and autohealing  
Deployment → Versioning and Rollback  
Service → Static (Non-ephemeral) IP and Networking.  
Volume → Non-ephemeral storage



## Kubelet:-

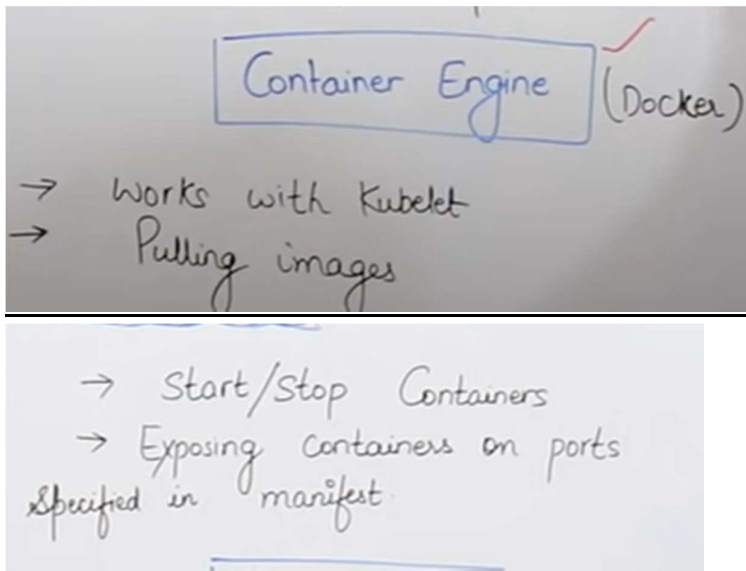
- KUBELET eak agent hai jo ki humare master or worker node ko aapas mein interact karte hai, Kubelet hi responsible hota hai humare containers ki monitoring karne ke liye jo humare containers chal rahe usko monitoring karne ke liye
- Kubelet ka kaam hota hai pod ko control karna , control ka matlab ki pod mein kitane container hone chahiye aur kya-kya humare pod mein hona chahiye ye sab kaam humara kubelet karta hai.
- Kubelet request karta hai → api server ko api server request karta hai → ETCD cluster ko ETCD cluster ke baad vo khabaar jayegi humare → Control manager ke liye pass , phir control manager dekhega ki kis pod ke liye andar kaun sa container banana hai kaun sa hatana hai toh uske liye phir vo eak baar → ETCD cluster par jayega aur phir ETCD cluster se jab vo confirm ho jayega toh uske baad vo → api server ke liye request karega → kube scheduler ke liye pass kube scheduler jaa ke phir container bana dega agar banane ki jarurat padi agar hatane ki pad toh hata bhi dega
- Hum jo bhi command issue kar rahe hai apne master node par usko receive karna master node se liye worker node par, vo sara ka sara kaam humara kubelet sambhalata hai.
- Kubelet ne bataya apne container run time ko, aur container run time humare case mein hai docker ki humara jo agala container aayega ya pod aayega usko humare is node mein hi launch karna hai
- Kubelet by default port 10255 par kaam karta hai but hum isko change bhi kar sakte hai.



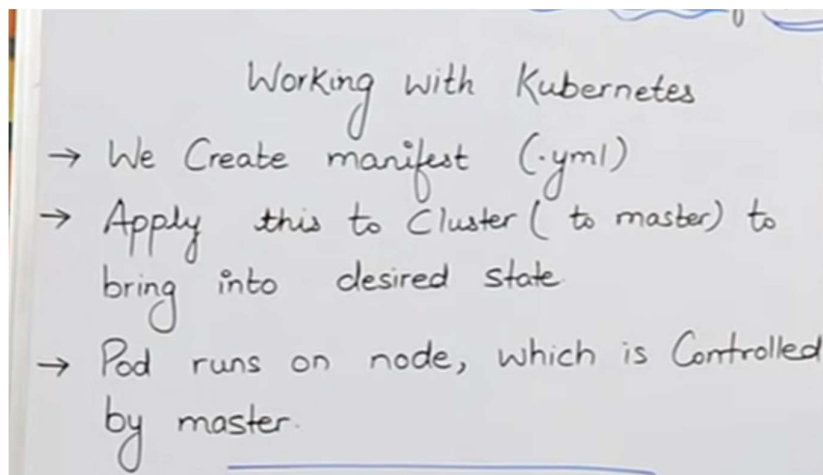
## **Kubeproxy:-**

- Networking sein related jo kaam hota hai vo humara **KubeProxy** kaam karata hai.
- Agar hum pod ko stop karke start karenge toh meri ip address bhi change ho jayegi.
- Iska main kaam hai ki har eak pod kein pass ip address assign ho .
- Kubeproxy basically DNS manager hai kubernetes ka , joki ye manage karta hai ki humare pods kya ip- address assign karna hai , pods ko service sein attract kaise karte hai .
- Agar koi request aa arahi hai mere cluster par toh usko sahi jagah route karna , matlab sahi pod par.

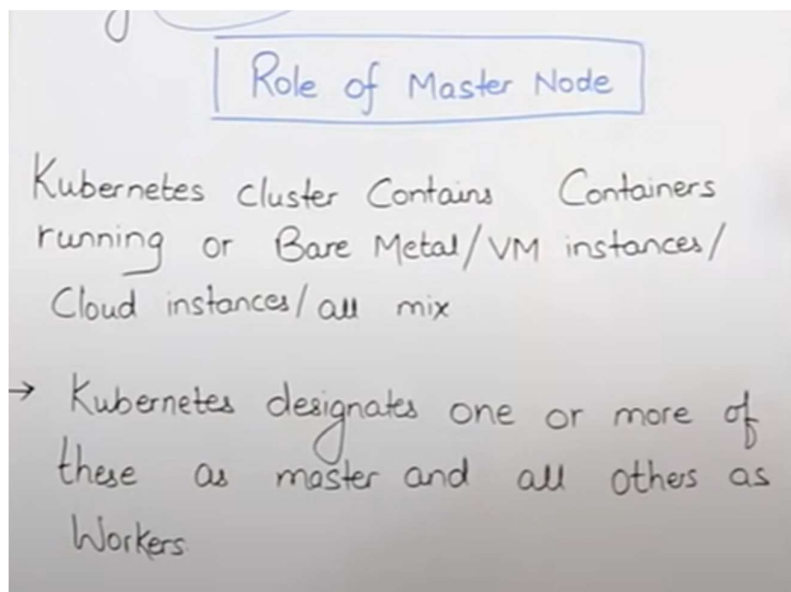
## **Container engine:-**



## **Working with kubernetes**



- ➔ Kubernetes mein humko sabse pehale manifest file create karna hota hai, manifest file hum yaml aur json dono mein likh sakte hai.
- ➔ Ye manifest cluster par applied hota hai matlab group of pods par applied hota hai, cluster humara node kein andar bana hoga, aur manifest file ko master apply karvayega ki iss kaam ko kaise karna hai.



- ➔ Kubernetes mein humara pod agar maan lijiye virtual cloud hai ya toh phir humae on-premises ya phir Bare-metal par hai un sabko hum eak sath kar sakte hai.
- ➔ Jaha par hum master ka package install kar denge vo as a master node jaise kaam karega, aur baki jaha master node package nahi install karunga vo humara worker node hi rahega.

- The master is now going to run Set of K8s processes. These processes will ensure smooth functioning of Cluster. These processes are called "Control Plane".
- Can be Multi-master for high availability.
- Master runs Control Plane to run cluster smoothly.

- ➔ Hm eak sein jada bhi master bana sakte hai ki jissase agar eak master fail ho toh humara dusara master active ho jaye aur control karta rahe.
  - ➔ Master ka main kaam hi yahi hai ki humare jo cluster hai unko dekhata rahe ki vo sahi sein kaam kar rahe hai ya nahi.
- 

Important

Kubectl → Single Cloud  
Kubeadm → on premise  
Kubefed → Federated

- ➔ Agar hum **cloud** par kaam kar rahe toh vaha hum Kubectl command mostly use karte hai.
- ➔ Agar hum on premise hardware par kubernetes install karke kaam kar rahe toh vaha hum **kubeadm** command use karenge.
- ➔ Agar humare pass hybrid cloud, virtual cloud on premise hardware sab hai toh vaha hum **kubefed** command use karenge

➔