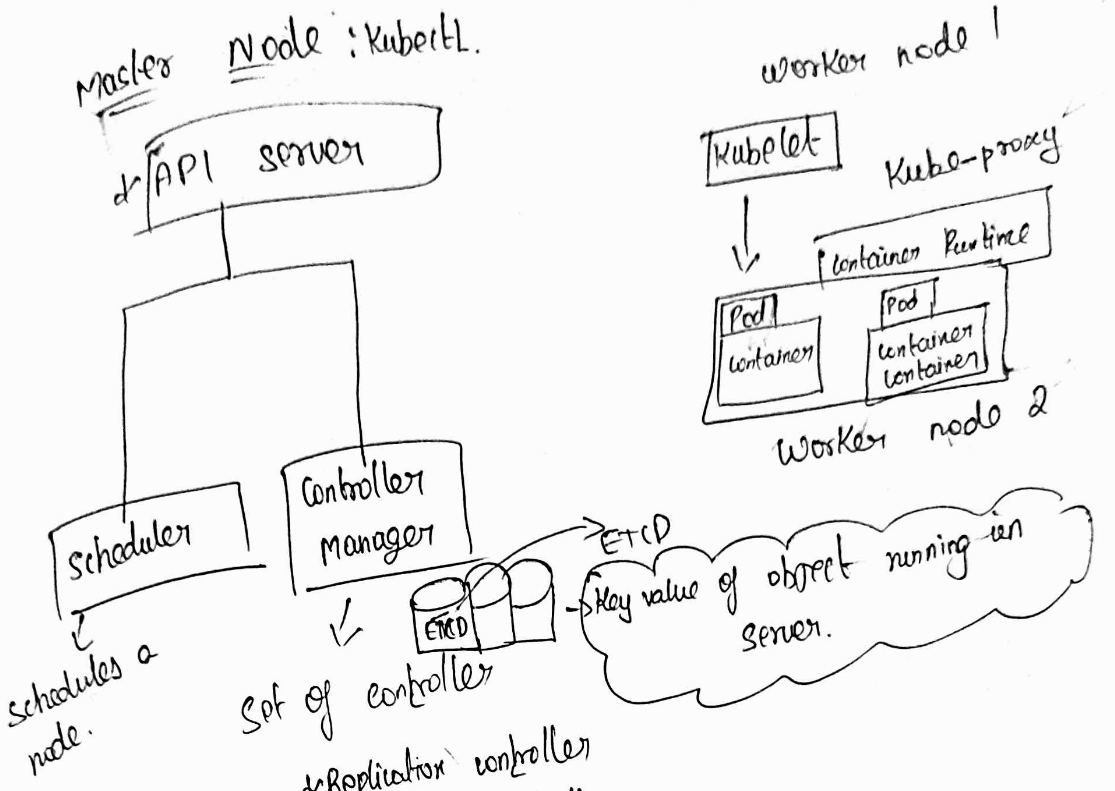
Kubernetes Architecture

→ process of creating, scaling, monitoring & managing.

Kubernetes is a container orchestration Engine.



* Replication controller
* Deamonset controller

Kubernetes object

Creating objects using two ways

* umparative way
* declarative way

demo ~ Kubectl create deploy nginx ==umage=nginx ==port=80. (creating an object).

demo ~ Kubectl create -f deploy-yaml

it convert Yaml ~~file~~ file to Json file and send it to the API server.

Kubernetes namespaces.

In Kubernetes, Name space is used to create a usolated environment to different teams within the same cluster

- Kubectl get ns -> view namespace
- Kubectl create ns dev

namespace/dev created.

```
Kubectl run nginx ==image=nginx
```

↳ Here we didn't mention the namespace so a pod will be create in default location

pod is one ore more containers with shared storage and network resource.

```
Kubectl run nginx ==image=nginx -n dev.
```

↳ creating pod in namespace dev.

We can change (or) set the context to another namespace

```
Kubectl config set-context --current --namespace =dev.
```

deleting an namespace

```
Kubectl delete ns dev.
```

Kubectl -> kubernetes command line tool. allows you to run commands against Kubernetes clusters.

You can use Kubectl to deploy applications, inspect and manage cluster resources and view logs.

Labels and selectors

" Cat command in linux is used to print the content of a file. "

Labels are a set of Key-value pairs attached to Kubernetes obj.

-> Kubectl get pods --show -labels.

label can be added before and after the pod creation

-> Kubectl label pod nginx demo=true.

* Labels are defined in the metadata section of the Kubernetes object.

* When u have a service which has to get attached to a particular ~~Node~~ pod whenever traffic comes to that particular pod it has to route →giving path it to specific number of pods this happens by labels and selectors if selector and label have the same key the service get attached to the number of pods.