**Inside frontend create the react js application**

npx create-react-app react-user-app

cd react-user-app

npm install axios

open the project in VS code

react JS : 3000

browser

spring boot : 9090

CORS policy : Cross Origin resource sharing

**Container Management tool (Orchestration tool)**

Container management tool is responsible to manage more than one container life like deploying, scaling, backup etc.

1. Docker swarm
2. Kubernetes
3. Open shift
4. AWS EKS (Elastic Kubernetes Service)
5. Azure Container apps

Docker compose is a part of docker which take the help of yml file to run more than one container.

Docker compose doesn’t provide scale up and down on demand.

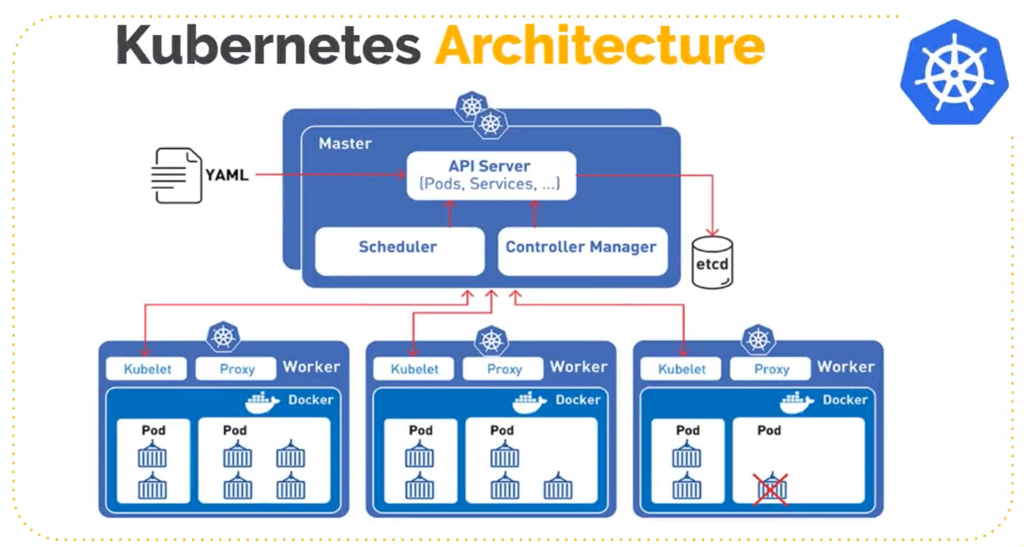
All container must be running in same **machine** or **node**.

Kubernetes : it is an open source platform use to maintain more than one container. It help to deploying more than one container on same machine or different machine (nodes). Kubernetes is a part google.

In short we call Kubernetes K8s

Need for orchestration tool.

1. Complexity of deploying and managing more than one container in same machine or different machine.
2. Resource allocation.
3. Scale up and down on demand.
4. Security
5. Monitoring
6. Logs files



Application 🡪 nginx server 🡪 running react js application

Image -🡪 to this react js application using container.

Pods : in Kubernetes environment we can’t run container directly. Pods contains one or more than one container. In one node or machine we can run more than one pods.

Below tools provide cluster environment for Kubernetes

1. Docker desktop
2. **Minikube** : it is an open source tool which provide features as single node Kubernetes cluster environment.

Node : Machine or device.

Cluster : we need to combine more than one machine to share the resource through network environment.

Master Node : Actual Kubernetes cluster environment.

Worker node : machine going to deploy the application on cluster environment.

**Install the minikube with below URL**

[**https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download**](https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download)

**check the minikube version**

**minikube version**

**then start the minikube**

**minikube start**

**using below command enable few minikube service**

**minikube addons enable metrics-server** **docker context use default**

**to communicate Kubernetes cluster we need to kubectl tool or command.**

**Kubernetes cluster ie(kubectl command line tool) help to communicate Kubernetes cluster environment.**

**https://kubernetes.io/releases/download/#binaries**

**kubectl cluster-info**

**kubectl get nodes it provide all machine details.**

**kubectl get pods**

**kubectl get services**

**Pods : it contains one or more than one containers. Each containers responsible to run the application.**

**Pods can’t expose container outside.**

**With help of service we need to expose container outside pods. So we can access that application.**

**Kubernetes deployment : A deployment is a controller that manage the life cycle of pods. Using deployment we can scale up and down more than one pods. With help of deployment we can create replica of more than one pods.**

**We need to create any application like spring boot or react js.**

**That application we need to publish in docker hub account.**

**That application we run using deployment concept using Kubernetes environment.**