**Step to deploy simple web application created using React JS**

1. **Create react js project**
2. **Build the react js application**
3. **Then create docker file**
4. **Publish the image.**
5. **Then start minikube (one the of the cluster environment)**
6. **minikube start (start minikube cluster)**
7. **minikube dashboard (open Minikube GUI base dashboard)**
8. **create the deployment file with anyname.yml : this file contains pods configuration details. One or many pods link with pre defined or user defined image**

**deployment.yml**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: my-server**

**labels:**

**app: my-server**

**spec:**

**replicas: 3**

**selector:**

**matchLabels:**

**app: my-server**

**template:**

**metadata:**

**labels:**

**app: my-server**

**spec:**

**containers:**

**- name: web-server-container**

**image: akashkale/my-reactjs:pro1**

1. **kubectl apply -f deployment.yml**
2. **kubectl get deployments**
3. **kubectl get pods**
4. **if we want access container part of pods we need to expose as service. Then only we can access container.**
5. **Now we need to create the service.yml. which contains deployment details and expose as service.**

**service.yml**

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: simple-app-service**

**labels:**

**app: my-server**

**spec:**

**type: LoadBalancer**

**selector:**

**app: my-server**

**ports:**

**- port: 80**

**nodePort: 30001**

**targetPort: 80**

**protocol: TCP**

1. **kubectl apply -f service.yml**
2. **kubectl get service**
3. **now to view application output we need to take the help of minikube command**
4. **minikube service servicename**
5. **you can view the application on browser.**