

Roll no : 22ISR001

Introduction

Steps to Deploy

- Create deployment.yaml:

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: sample-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: sample-app
  template:
    metadata:
      labels:
        app: sample-app
    spec:
      containers:
        - name: sample-app
          image: abinayabalusamy/bookbarter
          ports:
            - containerPort: 80

```

2. Create Kubernetes Service

```
abinaya@ABINAYA:~$ vim service.yaml
```

```

apiVersion: v1
kind: Service
metadata:
  name: sample-app-service
spec:
  selector:
    app: sample-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: NodePort

```

3. Deploy the Application in Kubernetes

```
abinaya@ABINAYA: ~$ minikube start
minikube v1.35.0 on Ubuntu 24.04 (amd64)
Using the docker driver based on existing profile
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.46 ...
Restarting existing docker container for "minikube" ...
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
Verifying Kubernetes components...
  Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: storage-provisioner, default-storageclass
Done! kubectrl is now configured to use "minikube" cluster and "default" namespace by default
abinaya@ABINAYA:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

```
abinaya@ABINAYA:~$ kubectl apply -f deployment.yaml
deployment.apps/sample-app configured
```

```
abinaya@ABINAYA:~$ kubectl apply -f service.yaml
service/sample-app-service created
```

4. Access the Application

```
abinaya@ABINAYA:~$ minikube service sample-app-service
```

NAMESPACE	NAME	TARGET PORT	URL
default	sample-app-service	80	http://192.168.49.2:32541

```
Starting tunnel for service sample-app-service.
```

NAMESPACE	NAME	TARGET PORT	URL
default	sample-app-service		http://127.0.0.1:44303

```
Opening service default/sample-app-service in default browser...
http://127.0.0.1:44303
Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```

5. Output

