

# Step-by-Step Guide to Deploying a Spring Boot App on Minikube using Kubernetes

## **Step 1: Create a New Directory**

This command creates a new directory named task4 in the current working directory.

**Code:**

```
mkdir task4
```

## **Step 2: Navigate to the Directory**

This command moves you into the task4 directory.

**Code:**

```
cd task4
```

## **Step 3: Create a YAML Configuration File**

This command opens the **sample.yaml** file in the Vim text editor.

If the file does not exist, Vim will create it.

Inside Vim:

- Press **i** to enter insert mode.
- Write your Kubernetes YAML configuration (e.g., a deployment or service).
- Press **ESC**, type **:wq**, and press **Enter** to save and exit.

**Code:**

```
vim sample.yaml
```

#### **Step 4: Apply the YAML Configuration Using kubectl**

This command deploys resources defined in sample.yaml to the Kubernetes cluster.

Ensure that **Minikube** or another Kubernetes cluster is running before executing this.

**Code:**

```
kubectl apply -f sample.yaml
```

#### **Step 5: Check Running Pods**

This command lists all running pods in the Kubernetes cluster.

It provides details such as pod name, status, restarts, and age.

**Code:**

```
kubectl get pods
```

#### **Step 6: Expose the Spring Boot Application via Minikube**

This command exposes the springboot-app service in Minikube.

It opens the application in a web browser by forwarding traffic to a local machine-accessible URL.

## Code:

```
minikube service springboot-app
```

## Screenshots:

```
GNU nano 7.2
sivadeep@sivadeep-Vivobook:~/task4
```

sample.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: springboot-app
  name: springboot-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: springboot-app
  template:
    metadata:
      labels:
        app: springboot-app
    spec:
      containers:
        - name: my-springboot-app
          image: sivadeep30/springboot
          imagePullPolicy: Always
          ports:
            - containerPort: 8080
              name: http
              protocol: TCP
  # service type loadBalancer
...
apiVersion: v1
kind: Service
metadata:
  labels:
    app: springboot-app
    k8s-app: springboot-app
  name: springboot-app
spec:
  ports:
    - name: http
      port: 8080
      protocol: TCP
      targetPort: 8080
  type: NodePort
  selector:
```

```
sivadeep@sivadeep-Vivobook:~/task4$ vim sample.yaml
sivadeep@sivadeep-Vivobook:~/task4$ kubectl apply -f sample.yaml
deployment.apps/springboot-app created
service/springboot-app created
```

```
sivadeep@sivadeep-Vivobook:~/task4$ kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
myapp-757c876dcf-d5kj9  1/1    Running   1 (23h ago)  23h
petapp-66678659c4-n56vp  1/1    Running   0          31m
petapp1-856fb8895c-km25t 1/1    Running   0          28m
springboot-app-6d457ccf99-xxltk 1/1    Running   0          11s
sivadeep@sivadeep-Vivobook:~/task4$ minikube service springboot-app

|-----|-----|-----|-----|
| NAMESPACE | NAME      | TARGET PORT | URL           |
|-----|-----|-----|-----|
| default   | springboot-app | http://8080  | http://192.168.49.2:31102 |
|-----|-----|-----|-----|
💡 Opening service default/springboot-app in default browser...
```

