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

NAME : SANDEEP P S
ROLLNUM : 22CSL261

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.

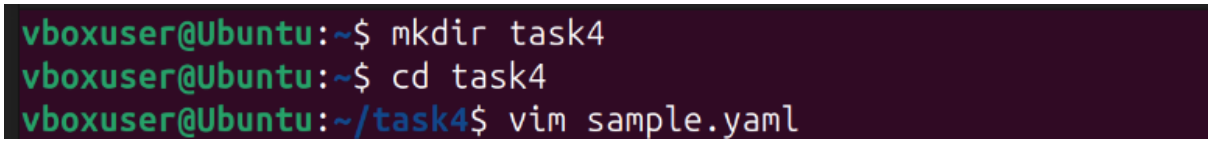
Inside Vim:

- 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
```

Screenshot:



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

Screenshot:

```
vboxuser@Ubuntu:~/task4$ kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
springboot-app-6fbbffbfcc-svf9x   1/1     Running   0          21s
```

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

Screenshot:


```
vboxuser@Ubuntu:~/task4$ kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
springboot-app-6fbbffbfcc-svf9x   1/1     Running   0          21s
```

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

Screenshot:

Prerequisites:

1)Minikube installed and running

Code:

minikube start

2)kubectl installed and configured

Code:

kubectl version --client

3)A valid `sample.yaml` file containing Kubernetes resource definitions (e.g., Deployment, Service).