

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 **t4.yaml** file in the Vim text editor.

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

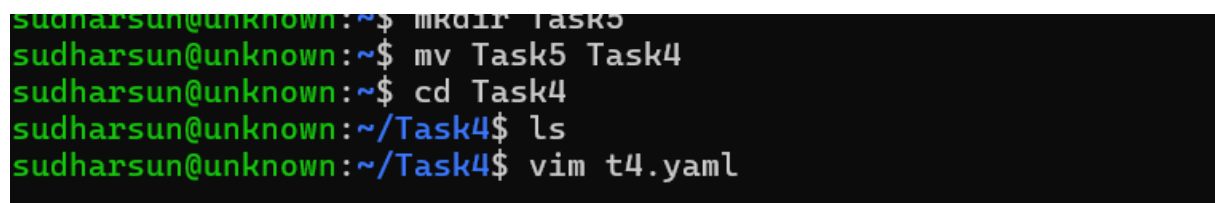
Inside Vim:

- 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 t4.yaml
```

Screenshot:



```
sudharsun@unknown:~$ mkdir Task5
sudharsun@unknown:~$ mv Task5 Task4
sudharsun@unknown:~$ cd Task4
sudharsun@unknown:~/Task4$ ls
sudharsun@unknown:~/Task4$ vim t4.yaml
```

Step 4: Apply the YAML Configuration Using kubectl

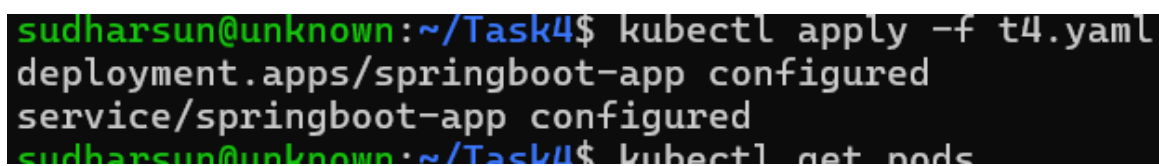
This command deploys resources defined in `t4.yaml` to the Kubernetes cluster.

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

Code:

```
kubectl apply -f t4.yaml
```

Screenshot:



```
sudharsun@unknown:~/Task4$ kubectl apply -f t4.yaml
deployment.apps/springboot-app configured
service/springboot-app configured
sudharsun@unknown:~/Task4$ kubectl get pods
```

Step 5: Check Running Pods

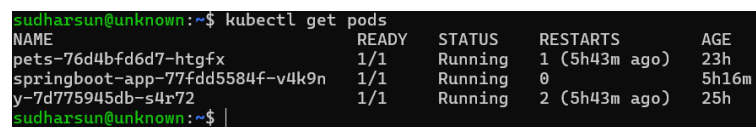
This command lists all running pods in the Kubernetes cluster.

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

Code:

```
kubectl get pods
```

Screenshot:



```
sudharsun@unknown:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
pets-76d4bfd6d7-htgfx              1/1     Running   1 (5h43m ago)   23h
springboot-app-77fdd5584f-v4k9n    1/1     Running   0           5h16m
y-7d775945db-s4r72                 1/1     Running   2 (5h43m ago)   25h
sudharsun@unknown:~$ |
```

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:

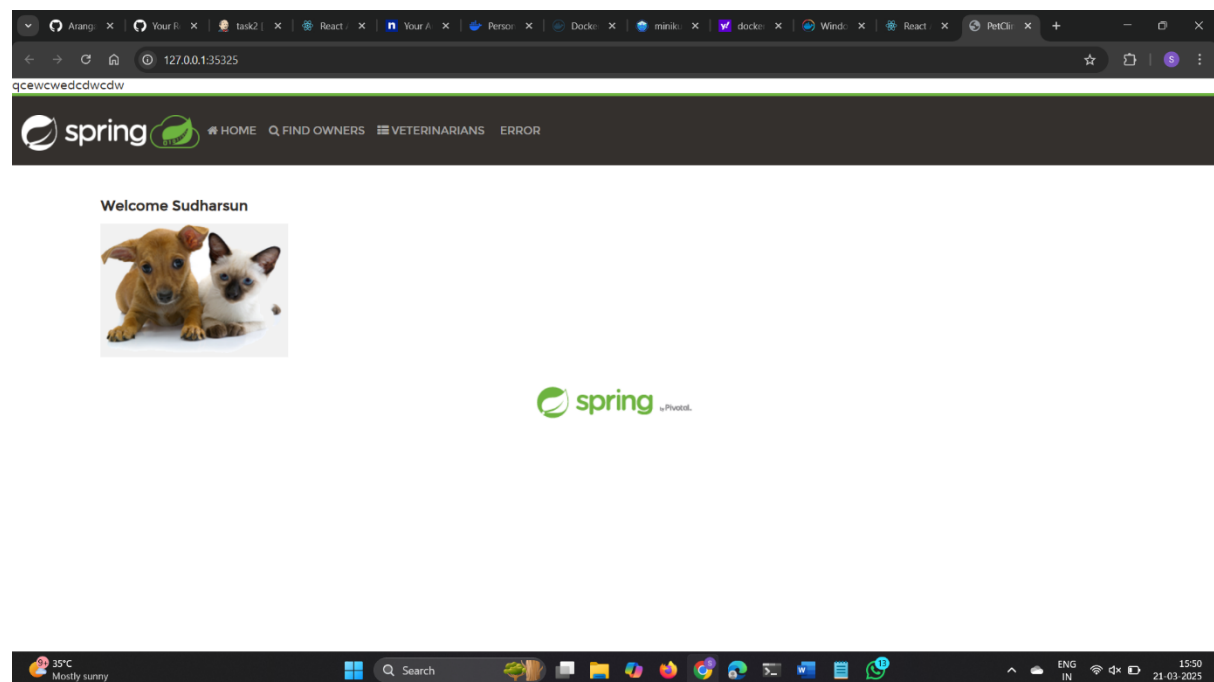
```
sudharsun@unknown:~/Task4$ minikube service springboot-app
```

NAMESPACE	NAME	TARGET PORT	URL
default	springboot-app	http/8080	http://192.168.49.2:32233

★ Starting tunnel for service springboot-app.

NAMESPACE	NAME	TARGET PORT	URL
default	springboot-app		http://127.0.0.1:34205

🌐 Opening service default/springboot-app in default browser...
http://127.0.0.1:34205
👉 Because you are using a Docker driver on linux, the terminal needs to be open to run it.



Prerequisites:

1)Minikube installed and running

Code:

```
minikube start
```

2)kubectl installed and configured

Code:

```
kubectl version --client
```

3)A valid `t4.yaml1` file containing Kubernetes resource definitions (e.g., Deployment, Service).