

TASK 4 – Kubernetes Using Shell Script

Step 1: MiniKube

Start the minikube using minikube start command

```
suganth@suganth-debian: ~$ minikube start
minikube v1.35.0 on Debian 12.10
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! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Step 2: Folder Creation

Create a folder named task4

```
suganth@suganth-debian: ~$ cd Music
suganth@suganth-debian: ~/Music$ mkdir Task-4
```

Step 3: New Yaml File

Create a new vim file named devops.yaml

```
suganth@suganth-debian: ~/Music/Task-4$ nano task4.yaml
```

Step 4: Yaml file

Enter the yaml file code using the insert

```
Activities Terminal Mar 22 10:24 AM
suganth@suganth-debian: ~/Music/Task-4/dev
task4.yaml
GNU nano 7.2
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: suganth07/petclinic
          imagePullPolicy: Always
          ports:
            - containerPort: 80
              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: 80
      protocol: TCP
      targetPort: 80
    type: NodePort
  selector:
    app: springboot-app
Read 42 lines
⌘ Help ⌘ Write Out ⌘ Where Is ⌘ Cut ⌘ Execute ⌘ Location ⌘ Undo ⌘ Set Mark ⌘ To Bracket ⌘ Previous ⌘ Back ⌘ Prev Word
⌘ Exit ⌘ Read File ⌘ Replace ⌘ Paste ⌘ Justify ⌘ Go To Line ⌘ Redo ⌘ Copy ⌘ Where Was ⌘ Next ⌘ Forward ⌘ Next Word
```

Step 5: Apply

Apply the changes made in the devops.yaml file

```
aranganathan@debian: ~$ kubectl apply -f task4.yaml
deployment.apps/springboot-app created
service/springboot-app created
```

Step 6: Get Pods

Get the pods information to check if it is running or not.

```
aranganathan@debian: ~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
pet-7478bdc68-w2j8m                1/1     Running   4 (20m ago)  13h
petclinic-5d5bc7b0dd-fhs8v         1/1     Running   2 (20m ago)  40m
s1-79c9c79599-88fph                1/1     Running   0           19m
s2-8469dc6644-lzjf8                1/1     Running   0           15m
s3-75dd844465-mhs57                1/1     Running   0           13m
springboot-app-588d7df488-k4lwz     1/1     Running   1 (20m ago)  29m
aranganathan@debian: ~$
```

Step 7: Service

Open the service springboot-app in the browser

```
aranganathan@debian: ~$ minikube service springboot-app
-----|-----|-----|-----|
| NAMESPACE | NAME          | TARGET PORT | URL               |
|-----|-----|-----|-----|
| default    | springboot-app | http/80     | http://192.168.49.2:38224 |
|-----|-----|-----|-----|
Starting tunnel for service springboot-app.
-----|-----|-----|-----|
| NAMESPACE | NAME          | TARGET PORT | URL               |
|-----|-----|-----|-----|
| default    | springboot-app | http/80     | http://127.0.0.1:41847 |
|-----|-----|-----|-----|
Opening service default/springboot-app in default browser...
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.
Opening in existing browser session.
.....
```

Step 8: Output

The output is shown in the browser in the localhost url present

