

Task 4 – Kubernetes Using Shell Script

Name: Swetha M

RollNo: 22CSR217

Step 1: MiniKube

Start the minikube using minikube start command

```
$ 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" ...
🔥 StartHost failed, but will try again: driver start: start: docker start minikube: exit status 1
stdout:

stderr:
Error response from daemon: failed to create task for container: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: error during container init: error setting cgroup config for procHooks process: failed to write "a *:* rwm": write /sys/fs/cgroup/devices/docker/b7f72b58eccebcc4a8b2e09d0d0166806476f506e751a4caef9a5b1bffbcb04/devices.allow: invalid argument: unknown
Error: failed to start containers: minikube

🔄 Restarting existing docker container for "minikube" ...
mini 🐳 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 and change the directory to task4.
- Create a new vim file named a.yaml
- Enter the yaml file code using the insert
- Apply the changes made in the devops.yaml file

```
$ mkdir Task4
$ cd Task4
$ vim a.yaml
$ kubectl apply -f a.yaml
deployment.apps/springboot-app configured
service/springboot-app unchanged
# kubectl get pods
```

Step 3: Get the pods information to check if it is running or not using the following command and Open the service springboot-app in the browser

Kubectl get pods

Minikube service springboot-app

```
$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
pet-5c5dd74f55-btybj               1/1     Running   3 (4m4s ago)  18h
springboot-app-59d4d4f67d-px9x7    1/1     Running   0           9s
springboot-app-9df9d6d8b-2b1s7     1/1     Terminating 1 (4m4s ago)  9m10s
$ minikube service springboot-app
NAMESPACE | NAME       | TARGET PORT | URL
-----|-----|-----|-----
default   | springboot-app | http/8080    | http://192.168.49.2:32720
* Starting tunnel for service springboot-app.
NAMESPACE | NAME       | TARGET PORT | URL
-----|-----|-----|-----
default   | springboot-app | http/8080    | http://127.0.0.1:45103
Opening service default/springboot-app in default browser...
http://127.0.0.1:45103
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```

Step 4: The output is shown in the browser in the localhost

