TASK 4 – KUBERNETES SHELLS SCRIPT

NAME: SINDHU K

ROLL NO: 22CSR196

STEPS:

1. Create a folder and move into that folder.

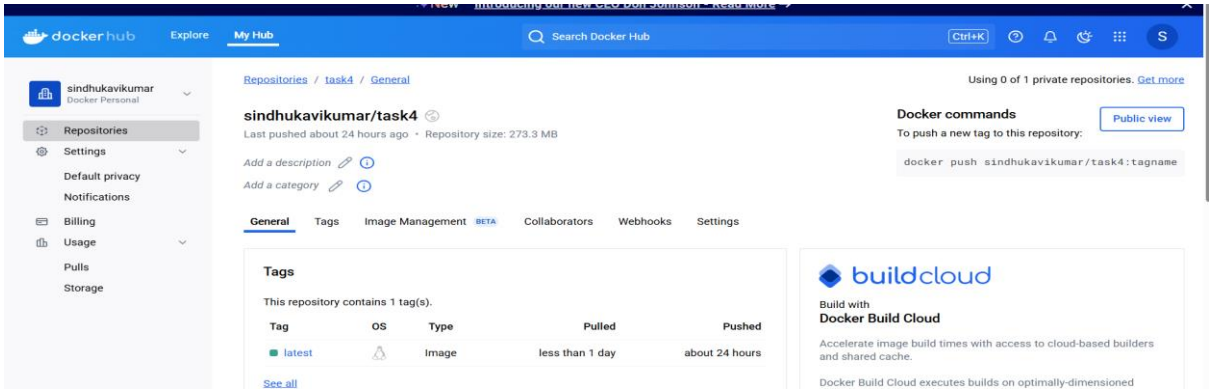
```
sindhukumar@Sindhu:~$ mkdir newtas
sindhukumar@Sindhu:~$ cd newtas
```

2. In that folder, create a file with a .yaml extension.
3. Copy the deployment script into the .yaml file. The script will deploy a Spring Boot application in Kubernetes and expose it externally via a NodePort service on port 80.
4. Apply the script using the following command:

   kubectl apply -f file.yaml

```
sindhukumar@Sindhu:~/newtas$ nano aba.yaml
sindhukumar@Sindhu:~/newtas$ kubectl apply -f aba.yaml
deployment.apps/springboot-app configured
service/springboot-app unchanged
```

```
 GNU nano 7.2                                                    aba.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: sindhukavikumar/task2
        imagePullPolicy: Always
        ports:
        - containerPort: 80
          name: http
          protocol: TCP
          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
```



5. Verify that the pods are running using the command:

   kubectl get pods

```
sindhukumar@Sindhu:~/newtas$ kubectl get pods
NAME                              READY   STATUS    RESTARTS      AGE
dogpic-574d5b796c-txq8l           1/1     Running   1 (27m ago)   18h
r1-5bdf8d8895-hzq56               1/1     Running   7 (27m ago)   21h
r2-6fcf47c689-mmzzs               1/1     Running   6 (27m ago)   20h
r3-5b8ccdd58d-x4lmk               1/1     Running   4 (27m ago)   19h
r5-6c8cb4bd94-rp86c               1/1     Running   3 (27m ago)   19h
springboot-app-7f88bfd67c-9f8tc   1/1     Running   0             45s
```

6. Expose the service using Minikube and obtain the URL:

   minikube service <service-name>

```
sindhukumar@Sindhu:~/newtas$ minikube service  springboot-app
|-----------|----------------|-------------|--------------------------|
| NAMESPACE |      NAME      | TARGET PORT |           URL            |
|-----------|----------------|-------------|--------------------------|
| default   | springboot-app | http/80     | http://192.168.49.2:31802 |
|-----------|----------------|-------------|--------------------------|
🏃  Starting tunnel for service springboot-app.
|-----------|----------------|-------------|------------------------|
| NAMESPACE |      NAME      | TARGET PORT |          URL           |
|-----------|----------------|-------------|------------------------|
| default   | springboot-app |             | http://127.0.0.1:38451 |
|-----------|----------------|-------------|------------------------|
🎉  Opening service default/springboot-app in default browser...
👉  http://127.0.0.1:38451
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```

7. Use the obtained URL to view the output in the browser.