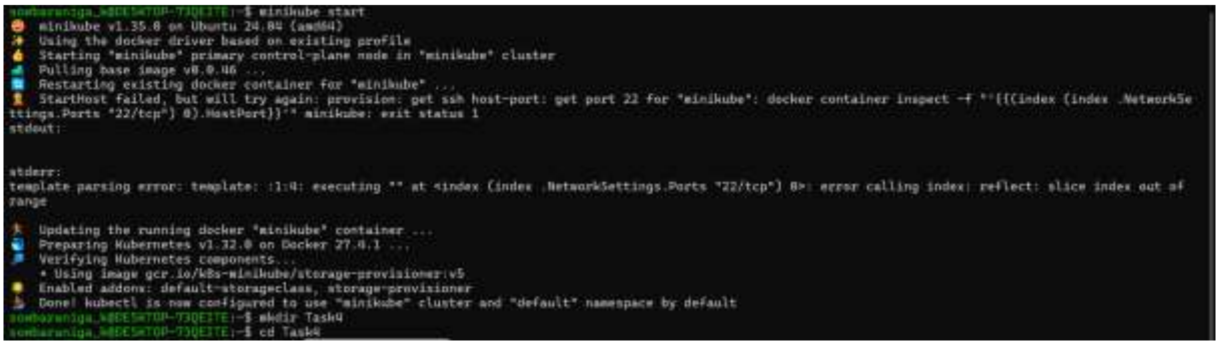
TASK 4 – KUBERNETES SHELLS SCRIPT

NAME: SOWBARANIGA K

ROLL NO: 22CSR202

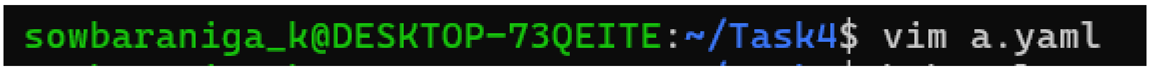
STEPS:

1. Create a folder and move into that folder.



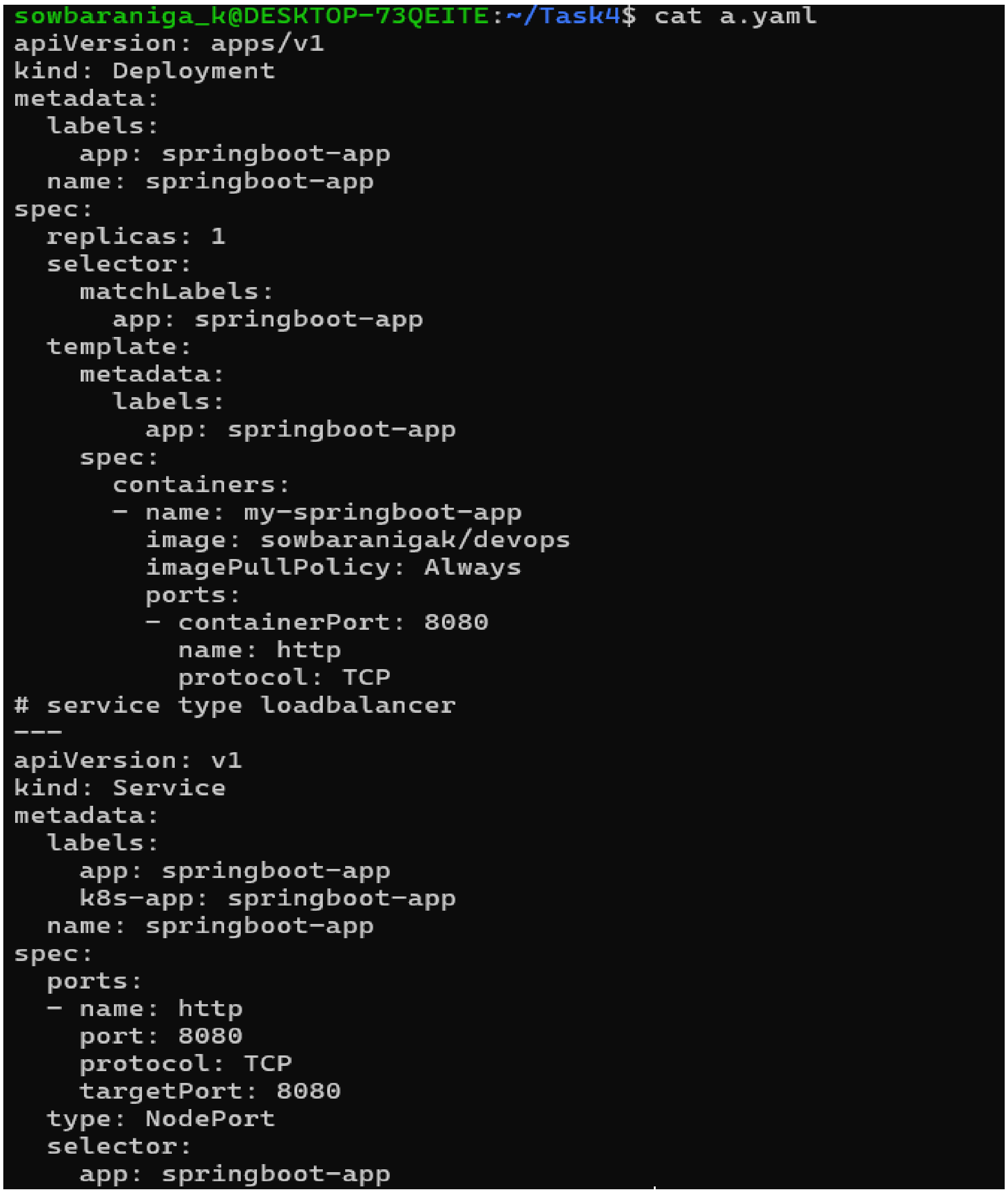
```
sowbaraniga_k@DESKTOP-73QEITE:~$ minikube start
😄  minikube v1.35.0 on Ubuntu 24.04 (amd64)
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.46 ...
🔄  Restarting existing docker container for "minikube" ...
🤦  StartHost failed, but will try again: provision: get ssh host-port: get port 22 for "minikube": docker container inspect -f "'{{(index (index .NetworkSe
ttings.Ports "22/tcp") 0).HostPort}}'" minikube: exit status 1
stdout:


stderr:
template parsing error: template: :1:4: executing "" at <index (index .NetworkSettings.Ports "22/tcp") 0>: error calling index: reflect: slice index out of
range

🏃  Updating the running docker "minikube" container ...
🐳  Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
sowbaraniga_k@DESKTOP-73QEITE:~$ mkdir Task4
sowbaraniga_k@DESKTOP-73QEITE:~$ cd Task4
```

2. In that folder, create a file with .yaml extension.



```
sowbaraniga_k@DESKTOP-73QEITE:~/Task4$ vim a.yaml
```

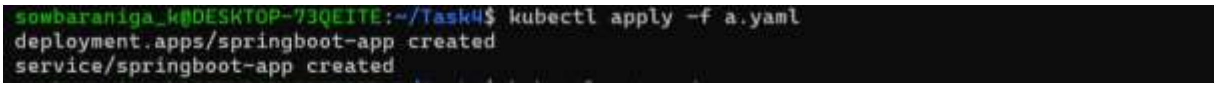
3. Copy the deployment script into the .yaml file. The script will deploy a Spring Boot application in Kubernetes and expose it externally via a NodePort service on port 8080.



```
sowbaraniga_k@DESKTOP-73QEITE:~/Task4$ cat a.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: sowbaranigak/devops
        imagePullPolicy: Always
        ports:
        - containerPort: 8080
          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: 8080
    protocol: TCP
    targetPort: 8080
  type: NodePort
  selector:
    app: springboot-app
```

4. Apply the script using the following command:

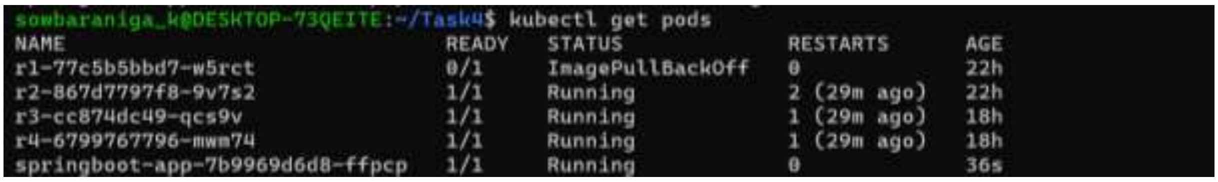
   kubectl apply -f file.yaml



```
sowbaraniga_k@DESKTOP-73QEITE:~/Task4$ kubectl apply -f a.yaml
deployment.apps/springboot-app created
service/springboot-app created
```

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

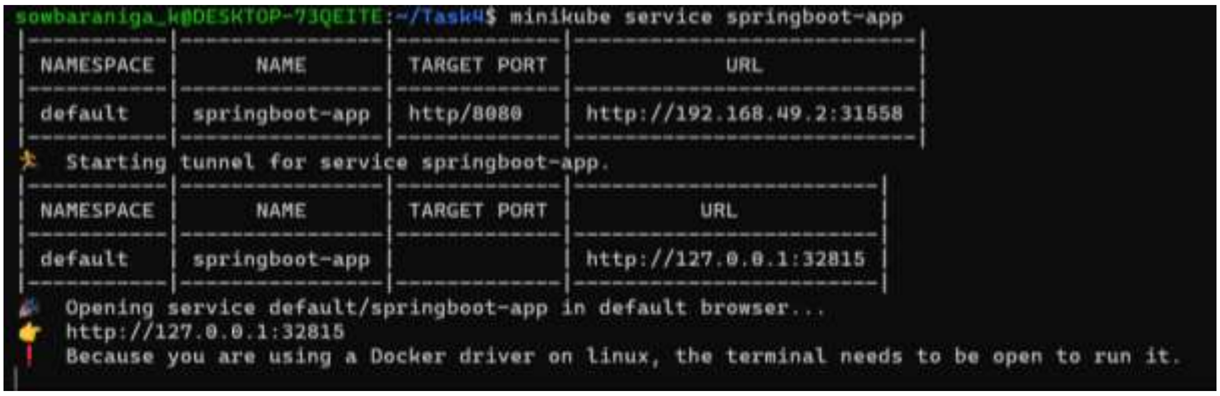
   kubectl get pods



```
sowbaraniga_k@DESKTOP-73QEITE:~/Task4$ kubectl get pods
NAME                                READY   STATUS             RESTARTS       AGE
r1-77c5b5bbd7-w5rct                 0/1     ImagePullBackOff   0              22h
r2-867d7797f8-9v7s2                 1/1     Running            2 (29m ago)    22h
r3-cc874dc49-qcs9v                  1/1     Running            1 (29m ago)    18h
r4-6799767796-mwm74                 1/1     Running            1 (29m ago)    18h
springboot-app-7b9969d6d8-ffpcp     1/1     Running            0              36s
```

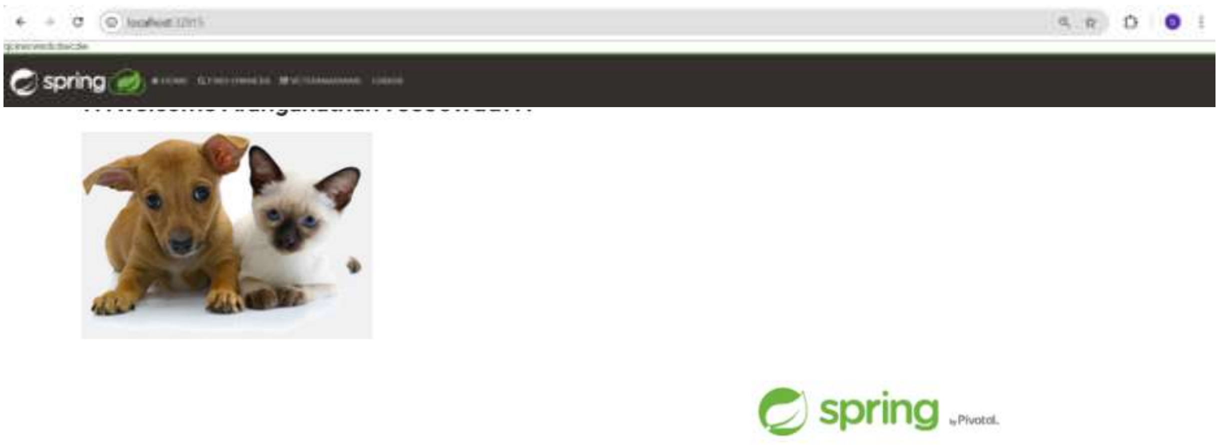
6. Expose the service using Minikube and obtain the URL:

   minikube service <service-name>



```
sowbaraniga_k@DESKTOP-73QEITE:~/Task4$ minikube service springboot-app
|-----------|----------------|-------------|---------------------------|
| NAMESPACE |      NAME      | TARGET PORT |            URL            |
|-----------|----------------|-------------|---------------------------|
| default   | springboot-app | http/8080   | http://192.168.49.2:31558 |
|-----------|----------------|-------------|---------------------------|
🏃  Starting tunnel for service springboot-app.
|-----------|----------------|-------------|------------------------|
| NAMESPACE |      NAME      | TARGET PORT |          URL           |
|-----------|----------------|-------------|------------------------|
| default   | springboot-app |             | http://127.0.0.1:32815 |
|-----------|----------------|-------------|------------------------|
🎉  Opening service default/springboot-app in default browser...
👉  http://127.0.0.1:32815
❗  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.


spring
spring by Pivotal.