

# TASK 4

## Step 1: Folder Creation

Create a folder named task4 and start minikube

```
sudharshana@sudharshana:~$ cd task4
sudharshana@sudharshana:~/task4$ ls
task4.yaml
sudharshana@sudharshana:~/task4$ 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 ...
🔄 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: storage-provisioner, default-storageclass
👉 Done! kubectrl is now configured to use "minikube" cluster and "default" namespace by default
sudharshana@sudharshana:~/task4$ kubectrl apply -f devops.yaml
error: the path "devops.yaml" does not exist
sudharshana@sudharshana:~/task4$ kubectrl apply -f task4.yaml
deployment.apps/springboot-app created
service/springboot-app created
sudharshana@sudharshana:~/task4$ kubectrl get pods
NAME                                READY   STATUS    RESTARTS   AGE
petclinic-597f4f7554-tc584          1/1     Running   3 (87s ago)  17h
springboot-app-d549b45f8-bq7vc      0/1     ContainerCreating   0           10s
```

## Step 2: Create tas.yaml

Create a yaml file and write the api script in it and apply the script

```
sudharshana@sudharshana:~/task4$ vim tas.yaml
sudharshana@sudharshana:~/task4$ kubectrl apply -f tas.yaml
```

```
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: sudharshanab/petclinic
          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
"tas.yaml" 42L, 760B
```

## Step 3: Minikube service

After applying the YAML, check if the service is running

```
sudharshana@sudharshana:~/task4$ kubectl apply -f tas.yaml
deployment.apps/springboot-app configured
service/springboot-app configured
sudharshana@sudharshana:~/task4$ minikube service springboot-app
```

| NAMESPACE | NAME           | TARGET PORT | URL                       |
|-----------|----------------|-------------|---------------------------|
| default   | springboot-app | http/8080   | http://192.168.49.2:32074 |

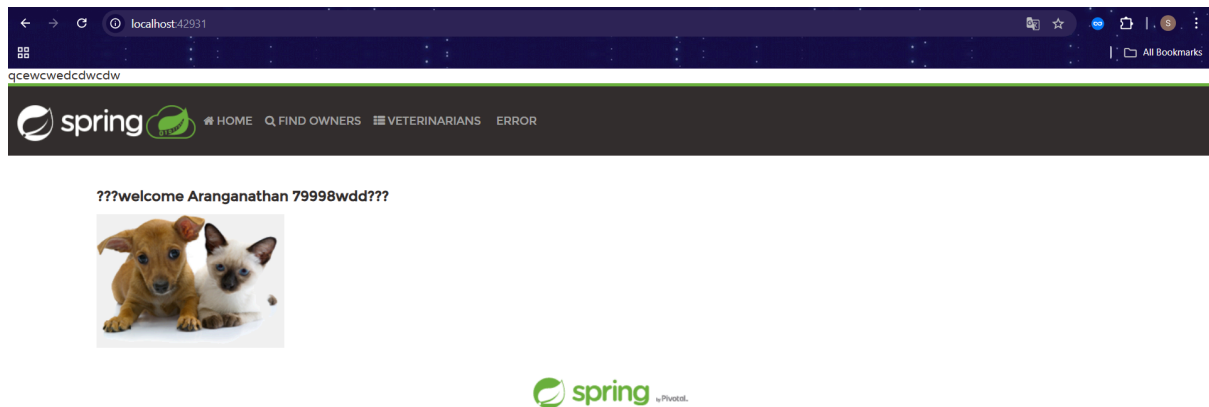
Starting tunnel for service springboot-app.

| NAMESPACE | NAME           | TARGET PORT | URL                    |
|-----------|----------------|-------------|------------------------|
| default   | springboot-app |             | http://127.0.0.1:42931 |

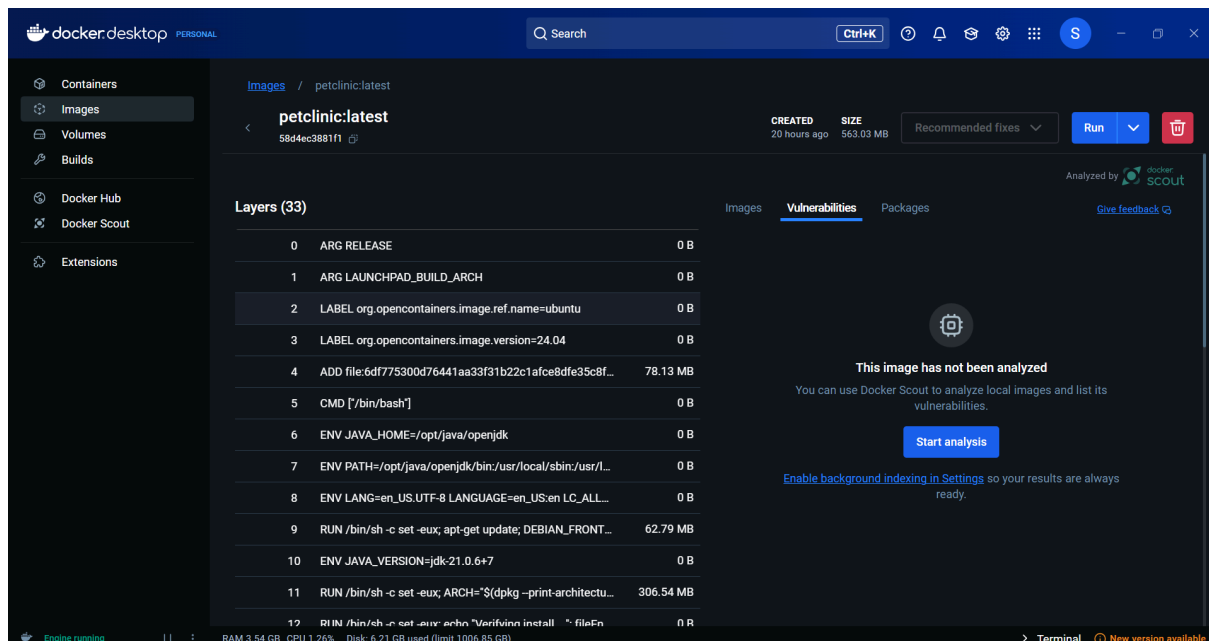
Opening service default/springboot-app in default browser...  
http://127.0.0.1:42931  
Because you are using a Docker driver on linux, the terminal needs to be open to run it.

## Step 5: Check for the Website

Open the docker image in the localhost and check if it is working



## Docker Image (Petclinic):



## Docker File (in Github):

