=> docker build -t spring-boot-app .

Service - a service exposes our application outside the Kubernetes cluster. It acts as the load balancer that distributes requests made to our application to various instances of the application running in the cluster.

Deployment - a deployment is a blueprint that is used to create instances of our application in the cluster.

=> create deployment.yaml file==========>

apiVersion: v1 # Kubernetes API version

kind: Service # Kubernetes resource kind we are creating

metadata: # Metadata of the resource kind we are creating

name: spring-boot-service

spec:

selector:

app: spring-boot-app

ports:

- protocol: "TCP"

port: 8080 # The port that the service is running on in the cluster

targetPort: 8080 # The port exposed by the service

type: LoadBalancer # type of the service. LoadBalancer indicates that our service will be external.

---

apiVersion: apps/v1

kind: Deployment # Kubernetes resource kind we are creating

metadata:

name: spring-boot-app

spec:

selector:

matchLabels:

app: spring-boot-app

replicas: 2 # Number of replicas that will be created for this deployment

template:

metadata:

labels:

app: spring-boot-app

spec:

containers:

- name: spring-boot-app

image: spring-boot-app # Image that will be used to containers in the cluster

imagePullPolicy: IfNotPresent

ports:

- containerPort: 8087 # The port that the container is running on in the cluster

=>kubectl apply -f deployment.yaml

minikube dashboard

minikube get services

minikube service service-name