Instalación necesaria

kubectl apply -f <https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.10.1/deploy/static/provider/cloud/deploy.yaml>  
  
kubectl get pods –n ingress-nginx



Comandos importantes si o si   
  
1. Pruebas jcoco . integración etc

#ejecutar pruebas

.\mvnw test jacoco:report

#ver jacoco

target/site/jacoco/index.html

2. script all

Se encarga de ejecuta los deployment, services , pvc , pcv

.\deploy-kubernetes.ps1

3.

#en el explorador de archivos

C:\Windows\System32\drivers\etc\hosts

#en el archivo

127.0.0.1 car-api.local



# este borra el pod que contiene la base de datos s

kubectl delete pod mariadb-car-776b754949-szzh7

si todo esta bien autonmaticamente se cre mi pod de esta y si la persistencia esta los datos que estaban antes deben estar ahí

Comandos durante El desarrollo

# correr mi aplicacion en local para eso es necesario modificar el propite spring.datasource.url=jdbc:mariadb://localhost:3306/carsdb

\mvnw spring-boot:run

#url para crear

Invoke-WebRequest -Uri "http://localhost:8080/api/cars-create" -Method POST -Headers @{"Content-Type"="application/json"} -Body '{"brand":"Toyota","model":"Camry","year":2023,"color":"Blue","price":30000.0}'

# Construir la imagen

docker build -t car-microservice:latest .

# Crear una red Docker para que los contenedores se comuniquen

docker network create car-network

# Ejecutar MariaDB en la red

docker run --name mariadb-car --network car-network -e MYSQL\_ROOT\_PASSWORD=root -e MYSQL\_DATABASE=carsdb -d mariadb:latest

# Ejecutar la aplicación en la red (asegúrate de que application.properties use mariadb como host)

docker run --name car-app --network car-network -p 8080:8080 -d car-microservice:latest