**Java project deployment**

**Step 1 : Fork the details into your git**

Use fork button on the public git

**Step 2: set up locally**

Commands used on command prompt

Git clone <https://github.com/Shivani-siri/skillupjava.git>

**Step 3: CI/CD pipeline**

CI pipeline: created a Github YAML actions file (.github/workflows/ci.yml)

name: Java CI Pipeline

on:

push:

branches:

- main

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v3

- name: Set up JDK 11

uses: actions/setup-java@v3

with:

java-version: '11'

- name: Build with Maven

run: mvn clean install

**Step 4: Containerization**

Created a Dockerfile

FROM openjdk:11-jdk-slim

WORKDIR /app

COPY target/skillupjava.jar /app/skillupjava.jar

ENTRYPOINT ["java", "-jar", "/app/skillupjava.jar"]

And I have built and tested locally using below commands:

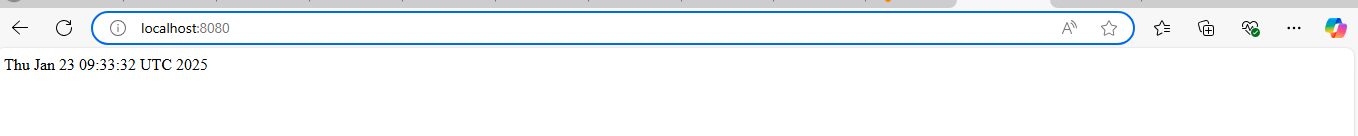
docker build -t skillupjava .

docker run -p 8080:8786 skillupjava

This maps port **8080** on your host to port **8786** inside the container.

**Step 5: Verifiation**

Verify [localhost:8080](http://localhost:8080/)



**Step 6 : Orchestration**

Create a deployment.yaml and service.yaml.

I created deploymemt.yaml and service.yaml files with the help of chatgpt

Deployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

  name: skillupjava-deployment

spec:

  replicas: 3

  selector:

    matchLabels:

      app: skillupjava

  template:

    metadata:

      labels:

        app: skillupjava

    spec:

      containers:

      - name: skillupjava

        image: shivani9430/skillupjava:latest

        ports:

        - containerPort: 8080

        env:

        - name: SPRING\_PROFILES\_ACTIVE

          value: "production"

service.yaml

apiVersion: v1

kind: Service

metadata:

  name: skillupjava-service

spec:

  selector:

    app: skillupjava

  ports:

  - protocol: TCP

    port: 8080

    targetPort: 8080

  type: NodePort

then start minikube using

command : minikube start

check the existing pods using

command : kubectl get pods

deploy to kubernetes:

kubectl apply -f deployment.yaml

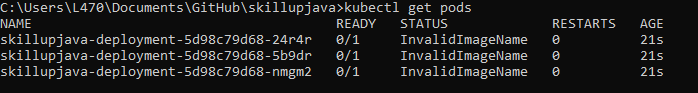
kubectl apply -f service.yaml

then run

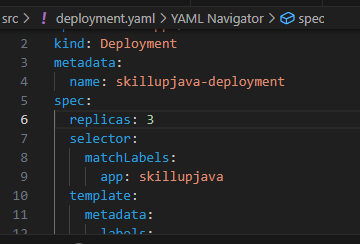
kubectl get deploymemts



The run kubectl get pods again,



We are seeing three pods because , we have give replicas in the deployment.yaml file as 3



Your java application is now successfully deployed.

You can check the logs using the container id

You can see the application description also using

Command: kubectl describe deployment skillupjava

