CI/CD Project – Microservice with Jenkins, Docker, Kubernetes, AWS

1. Prerequisites

- Java + Maven installed
- Docker installed
- Minikube + kubectl + Helm installed
- Jenkins installed
- DockerHub account
- (Future) AWS account for EKS

2. Microservice Setup

Spring Boot app with endpoints:

- `/api/hello` → returns "Hello from Microservice!"
- `/api/health` → returns "OK"

Run locally:

```
./mvnw spring-boot:run
```

3. Dockerization

Dockerfile:

```
FROM openjdk:17-jdk-slim
WORKDIR /app
COPY target/microservice-0.0.1-SNAPSHOT.jar app.jar
ENTRYPOINT ["java","-jar","/app/app.jar"]
```

Build & Push:

```
./mvnw clean package -DskipTests
docker build -t sivaraj7/microservice:latest .
docker push sivaraj7/microservice:latest
```

4. Kubernetes Deployment (Helm)

values.yaml:

```
image:
repository: docker.io/sivaraj7/microservice
```

```
tag: latest
service:
type: LoadBalancer
port: 80
```

Deploy locally:

```
helm install microservice ./charts/microservice
kubectl get pods
minikube service microservice --url
```

5. Jenkins Pipeline

Jenkinsfile automates:

- Checkout from GitHub
- Build JAR
- Build & Push Docker image
- Deploy with Helm to Minikube

... (Jenkinsfile script included here) ...

6. AWS EKS (Future)

```
eksctl create cluster --name microservice-cluster --region us-east-1 --nodes 2
aws eks --region us-east-1 update-kubeconfig --name microservice-cluster
kubectl get nodes
helm upgrade --install microservice ./charts/microservice --set image.repository=docker.io/sivaraj7/microservice --set image.tag=latest
```

7. Resume Highlight

"Built a CI/CD pipeline using Jenkins, Docker, Kubernetes, and Helm, deploying a Spring Boot microservice to Minikube locally and prepared configuration for AWS EKS."