Documentation (Setup Process and How to Run the Application)

Prerequisites :
1.Tools needed: JDK, Maven, Docker, Kubernetes (Minikube),Intellij.
2.IntelliJ IDE recommended
3.Specific versions of each tool (e.g., Java 17, Maven 3.x, Docker version, Minikube version).
4.Use java with Spring Boot.

Verify installations :
Verify Java using the command: `java -version`
Verify Maven using the command: `mvn -version`
Verify Docker using the command: `docker --version`
Verify Kubernetes using the command: `kubectl version --client`
Verify Minikube status using the commands: `minikube start` and `minikube status`

Building and running the microservices:
Navigate to each microservice directory and run:
mvn clean package
This will generate a JAR file in the target folder.

Containerizing the Microservices with Docker:
Create a file named Dockerfile in the root directory of the hello-service project and world-service- project.
Navigate to each microservice directory and build the Docker images using:
Run the following:
eval $(minikube docker-env)
docker build -t world-service .
docker build -t hello-service .

For specific images:
docker build -t harsha9505/world-service .
docker build -t harsha9505/hello-service .
Then
docker tag hello-service harsha9505/hello-service
docker tag world-service harsha9505/world-service

Push the images to Docker Hub:
docker push harsha9505/hello-service
docker push harsha9505/world-service

Run and pull locally:
docker run -p 8081:8081 harsha9505/hello-service
docker run -p 8081:8081 harsha9505/world-service

Deploying the Application on Kubernetes:
Start the Minikube cluster:
minikube start
Navigate to the directory containing the Kubernetes YAML files and apply them using the following commands:
kubectl apply -f hello-deployment.yaml
kubectl apply -f hello-service.yaml
kubectl apply -f world-deployment.yaml
kubectl apply -f world-service.yaml
Monitor Pods and Services:
Wait for the pods to be ready. Check their status with:
`kubectl get all`

Obtain Service URLs:
Get the URL for the Hello service by running:
`minikube service hello-service -n default --url`
Get the URL for the World service by running:
`minikube service world-service -n default --url`

After getting the URLs, copy and open them in a browser.
For Hello service, add `/hello` to the URL for hello-service-v2.
Example:127.0.0.1:55399/hello
For World service, add `/world` to the URL for world-service-v2 to view the outputs.
Example:127.0.0.1:55399/world

Script that calls both endpoints and prints "Hello World":
After getting the URLs, replace the HELLO_URL and WORLD_URL variables in the test.sh script with the URLs obtained. Ensure you modify the HELLO_URL and WORLD_URL variables accordingly in the script.
Save the updated script as test.sh.
Make sure both Hello and World services are running by checking their status:
kubectl get all

To execute the test.sh file, follow these steps:
1.Make the script executable: chmod +x test.sh
2.Run the script:./test.sh

Docker images to Docker Hub
https://hub.docker.com/repositories/harsha9505

Another method for Hello world that calls both end points.
After deploying manifest files,
Run
minikube service hello-service
Minikube service world-service

Open browser:
http://127.0.0.1:55399/hello-outputs :“hello”
http://127.0.0.1:55412/world-outputs:”world
http://127.0.0.1:55399/hello-world - outputs:”hello-world”