Setting up the development environment:

1) Download and install Node.js, which indeed include npm
2) Ensure you are in main branch, then download the code into local machine
3) Extract the zip file
4) Open VS Code and open the downloaded folder
5) Execute the following commands in terminal to initialize npm and to install express:
   - npm init -y #initialize npm
   - npm install express #install express
   - Initialize git using git init #initialize git
   - cd hello-service
   - npm init -y #initialize npm
   - npm install express #install express
   - cd ../world-service
   - npm init -y #initialize npm
   - npm install express #install express

Building docker images:

Execute following commands in terminal to build image of hello microservice and push it to docker hub:

```
cd ..

docker login

cd hello-service
```

#to build the docker images of hello microservice

```
docker build -t helloservice:latest -f Dockerfile.helloservice . #to build the docker images of hello microservice
```

#tag and push the images to Docker hub

docker tag helloservice:latest harshakata/helloservice:latest #harshakata is my dockerhub username. Command is docker tag helloservice:latest your-docker-hub-username/helloservice:latest docker push harshakata/helloservice:latest

Execute following commands in terminal to build image of world microservice and push it to docker hub:

cd ../world-service

#to build the docker images of world microservice

docker build -t worldservice:latest -f Dockerfile.worldservice .

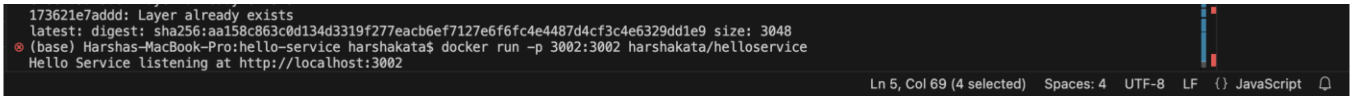
#tag and push the images to Docker hub

docker tag worldservice:latest harshakata/worldservice:latest #harshakata is my dockerhub username.. command is docker tag worldservice:latest your-docker-hub-username/worldservice:latest

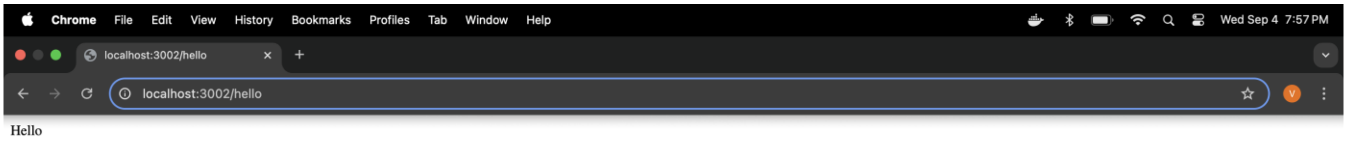
docker push harshakata/worldservice:latest

Run services locally

docker run -p 3002:3002 harshakata/helloservice   #Will return url http://localhost:3002. Add /hello to get the response Hello

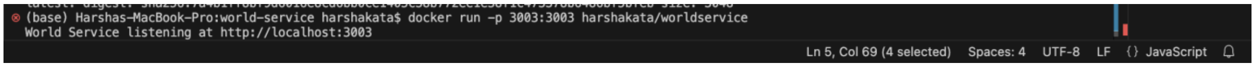


```
173621e7addd: Layer already exists
latest: digest: sha256:aa158c863c0d134d3319f277eacb6ef7127e6f6fc4e4487d4cf3c4e6329dd1e9 size: 3048
(base) Harshas-MacBook-Pro:hello-service harshakata$ docker run -p 3002:3002 harshakata/helloservice
Hello Service listening at http://localhost:3002
```

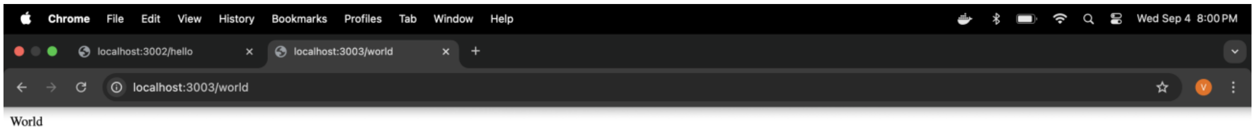


Hello

docker run -p 3003:3003 harshakata/worldservice #Will return url http://localhost:3003. Add /world to get the response World



```
(base) Harshas-MacBook-Pro:world-service harshakata$ docker run -p 3003:3003 harshakata/worldservice
World Service listening at http://localhost:3003
```



World

Deploying application on Kubernetes:

Deploy the hello service using below commands:

- cd ../hello-service
- kubectl apply -f helloservice.yaml

Deploy the world service using below commands:

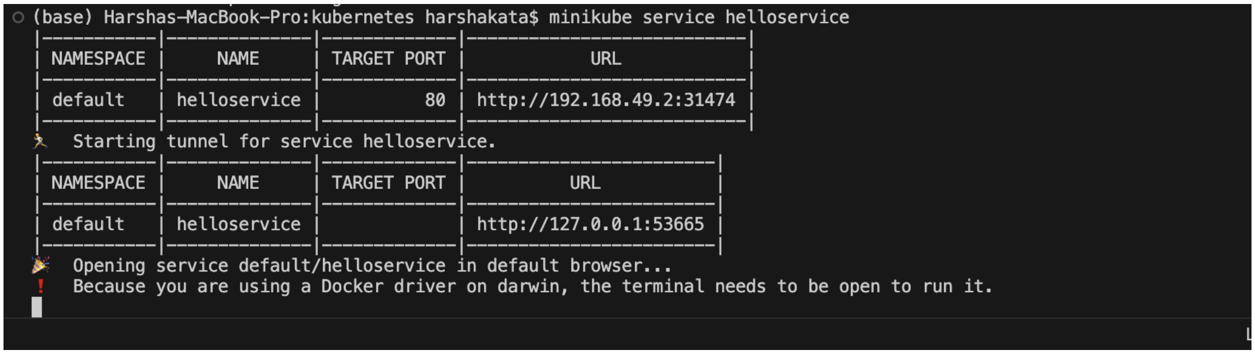
- cd ../world-service
- kubectl apply -f worldservice.yaml

Ensure both services are accessible through Kubernetes services

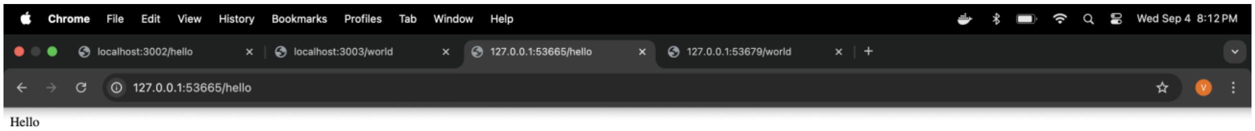
Open two new terminals

In terminal 1 execute:

minikube service helloservice #opens service in web browser.. add /hello to get the response hello



```
○ (base) Harshas-MacBook-Pro:kubernetes harshakata$ minikube service helloservice
  |-----------|--------------|-------------|---------------------------|
  | NAMESPACE |     NAME     | TARGET PORT |            URL            |
  |-----------|--------------|-------------|---------------------------|
  | default   | helloservice |          80 | http://192.168.49.2:31474 |
  |-----------|--------------|-------------|---------------------------|
  🏃  Starting tunnel for service helloservice.
  |-----------|--------------|-------------|------------------------|
  | NAMESPACE |     NAME     | TARGET PORT |          URL           |
  |-----------|--------------|-------------|------------------------|
  | default   | helloservice |             | http://127.0.0.1:53665 |
  |-----------|--------------|-------------|------------------------|
  🎉  Opening service default/helloservice in default browser...
  ❗  Because you are using a Docker driver on darwin, the terminal needs to be open to run it.
  █
```

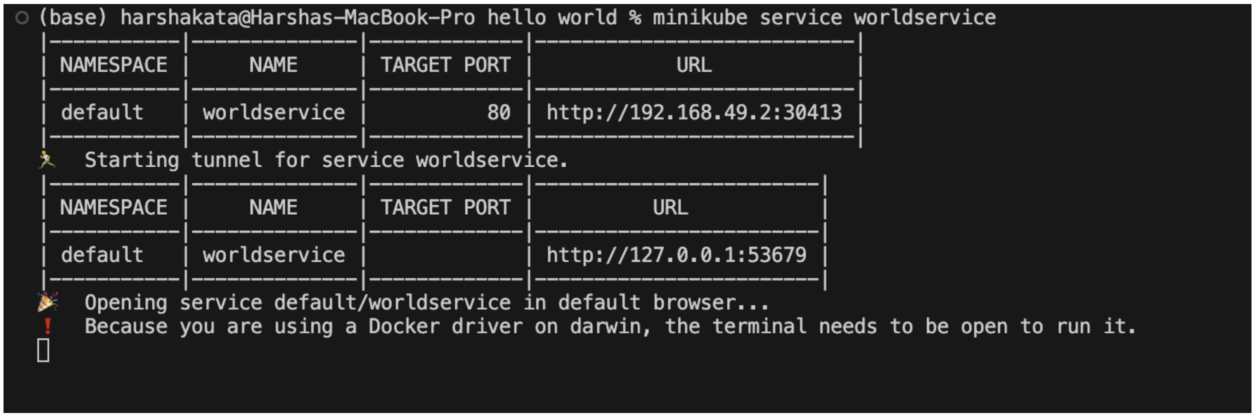


127.0.0.1:53665/hello

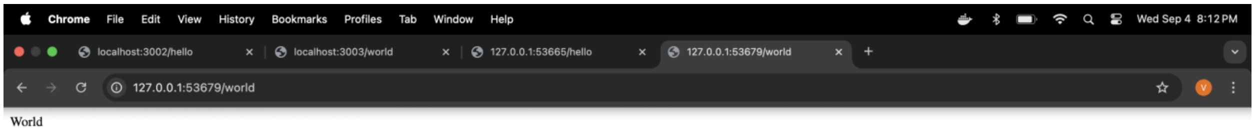
Hello

In terminal 2 execute:

minikube service world   #opens service in web browser.. add /world to get the response world



```
○ (base) harshakata@Harshas-MacBook-Pro hello world % minikube service worldservice
  |-----------|--------------|-------------|---------------------------|
  | NAMESPACE |     NAME     | TARGET PORT |            URL            |
  |-----------|--------------|-------------|---------------------------|
  | default   | worldservice |          80 | http://192.168.49.2:30413 |
  |-----------|--------------|-------------|---------------------------|
  🏃  Starting tunnel for service worldservice.
  |-----------|--------------|-------------|------------------------|
  | NAMESPACE |     NAME     | TARGET PORT |          URL           |
  |-----------|--------------|-------------|------------------------|
  | default   | worldservice |             | http://127.0.0.1:53679 |
  |-----------|--------------|-------------|------------------------|
  🎉  Opening service default/worldservice in default browser...
  ❗  Because you are using a Docker driver on darwin, the terminal needs to be open to run it.
  ▯
```



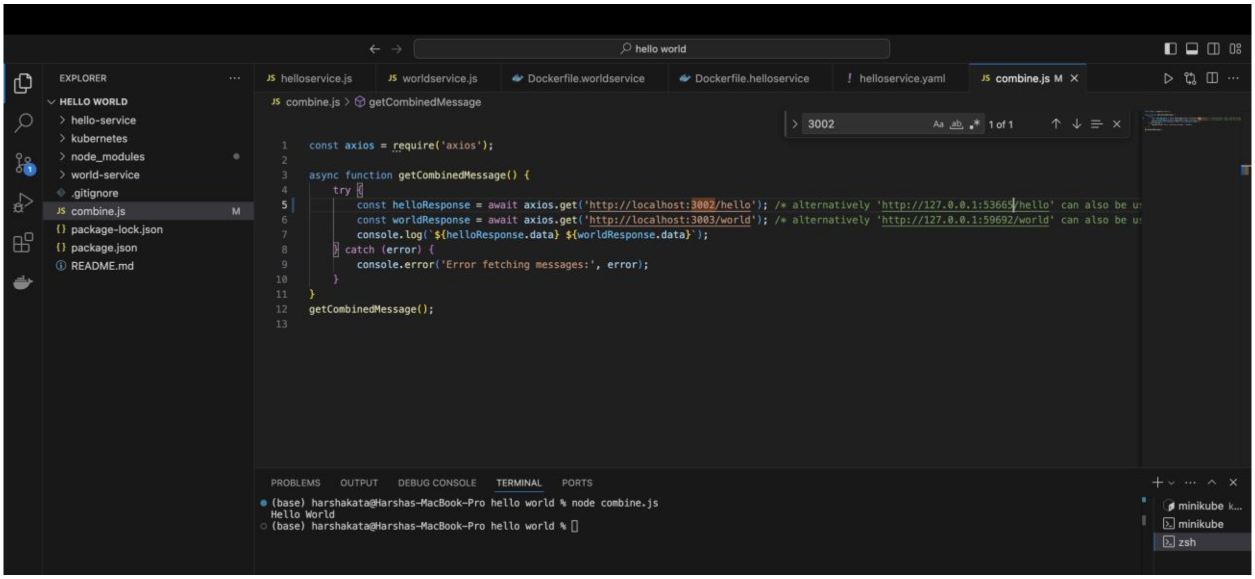
127.0.0.1:53679/world

World

Run script "combine.js" to print "Hello World"

In main terminal execute:

- cd..
- node combine.js   #will print "Hello World" in console



```
const axios = require('axios');

async function getCombinedMessage() {
    try {
        const helloResponse = await axios.get('http://localhost:3002/hello'); /* alternatively 'http://127.0.0.1:53665/hello' can also be us
        const worldResponse = await axios.get('http://localhost:3003/world'); /* alternatively 'http://127.0.0.1:59692/world' can also be us
        console.log(`${helloResponse.data} ${worldResponse.data}`);
    } catch (error) {
        console.error('Error fetching messages:', error);
    }
}
getCombinedMessage();
```

```
(base) harshakata@Harshas-MacBook-Pro hello world % node combine.js
Hello World
(base) harshakata@Harshas-MacBook-Pro hello world % 
```

Docker image links:

world: https://hub.docker.com/layers/harshakata/worldservice/latest/images/sha256:7a4b1ff8bf9d6016e8cd6bb0ce1405c38b772ee1e38f1c473378b6486bf5bfeb?uuid=8f6ea230-7cc4-4730-9a1e-0317ed314bbe%0A

(https://hub.docker.com/repository/docker/harshakata/worldservice/general)

hello:

https://hub.docker.com/layers/harshakata/helloservice/latest/images/sha256:aa158c863c0d134d3319f277eacb6ef7127e6f6fc4e4487d4cf3c4e6329dd1e9?uuid=8f6ea230-7cc4-4730-9a1e-0317ed314bbe%0A

(https://hub.docker.com/repository/docker/harshakata/helloservice/general)