

## 1. Setting up the development environment:

### **Choose a programming language and framework:**

I chose Node.js with Express.js to create endpoints /hello and /world.

### **Setting up development environment:**

1. Downloaded and installed Node.js, which indeed includes npm.
2. Created a new directory (hello-world)
3. Opened the directory using VS Code
4. Initialized npm using `npm init -y`
5. Installed express using `npm install express`
6. Initialized git using `git init`

## 2. Creating the Microservices:

- Created hello.js in the root directory with endpoint /hello and set it to port 3000
- Created world.js in the root directory with endpoint /world and set it to port 3001

## 3. Containerizing the Microservices with Docker

### **Create Dockerfiles:**

- Created Dockerfile.hello in the root directory to create a docker image of hello.js microservice
- Created Dockerfile.world in the root directory to create a docker image of world.js micro service

### **Built Docker images:**

- Made sure that the Docker desktop is running and login into docker using “Docker login” command in the terminal

Executed following command to build the docker images of both micro services

- `docker build -t hello:latest -f Dockerfile.hello .`
- `docker build -t world:latest -f Dockerfile.world .`

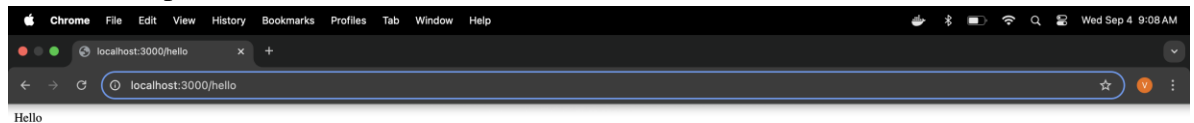
### **Tagged and pushed the images to Docker hub**

- `docker tag hello:latest harshakata/hello:latest`
- `docker tag world:latest harshakata/world:latest`
  
- `docker push harshakata/hello:latest`
- `docker push harshakata/world:latest`

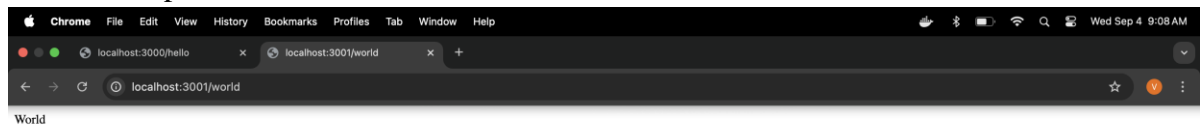
Note: harshakata is my docker hub username

Run the services locally:

- Checked if the services are working independently by executing following commands:
  - `docker run -p 3000:3000 harshakata/hello`



- `docker run -p 3001:3001 harshakata/world`



#### 4. Deploying the application on Kubernetes:

**Set up a Kubernetes cluster:**

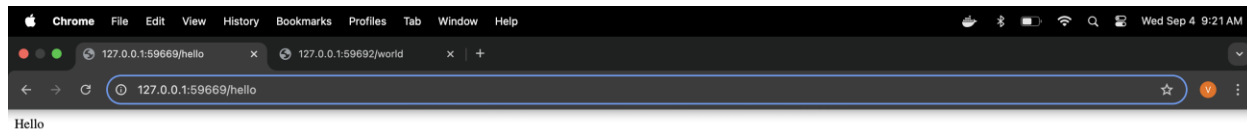
- Started Minikube using `minikube start`

**Create Kubernetes Manifests:**

- Created `hello.yaml` and `world.yaml` file defining deployments and services for each micro service
- Deployed the services using below commands:
  - `kubectl apply -f hello.yaml`
  - `kubectl apply -f world.yaml`

**Ensured both services are accessible through Kubernetes services by executing following commands.**

- `minikube service hello`

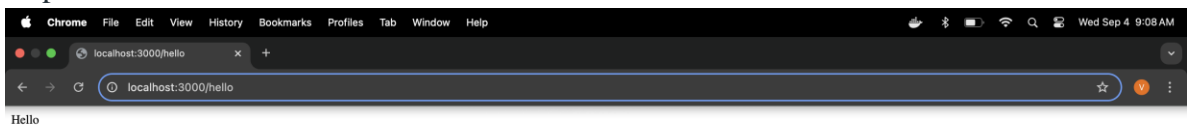


- minikube service world



## 5. Testing and Integration

Tested the **Hello** and **World** services individually to confirm they return the correct responses.



### Create a simple script or service:

- Created a simple script “combine.js” to read response from individual services and to print “Hello World”

- Run both minikube services in separate terminals
- Executed “node combine.js” in another terminal to print the output in console

The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a project named 'HELLO-WORLD' with files like 'combine.js', 'hello.js', 'world.js', 'Dockerfile.hello', 'Dockerfile.world', 'hello.yaml', 'world.yaml', 'package-lock.json', and 'package.json'. The main editor window displays the content of 'combine.js'.

```

1  const axios = require('axios');
2
3  async function getCombinedMessage() {
4      try {
5          const helloResponse = await axios.get('http://127.0.0.1:59669/hello'); /* alternatively 'http://localhost:3000/hello' can also be u
6          const worldResponse = await axios.get('http://127.0.0.1:59692/world'); /* alternatively 'http://localhost:3001/world' can also be u
7          console.log(`${helloResponse.data} ${worldResponse.data}`);
8      } catch (error) {
9          console.error('Error fetching messages:', error);
10     }
11 }
12
13 getCombinedMessage();
14

```

The Terminal panel at the bottom shows the output of a git push command:

```

Compressing objects: 100% (165/165), done.
Writing objects: 100% (174/174), 423.07 KiB | 8.83 MiB/s, done.
Total 174 (delta 20), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (20/20), completed with 4 local objects.
To https://github.com/Harshakata/hello-world.git
7c96614..37be08a master -> master
branch 'master' set up to track 'origin/master'.
(base) harshakata@Harshas-MBP hello-world % node combine.js
Hello World
(base) harshakata@Harshas-MBP hello-world %

```

On the right side of the terminal, there is a list of open terminals: 'bash', 'zsh', 'minikube', and 'zsh'. The 'zsh' terminal is currently active.

Used following git commands to timely push the code into git repository:

Git add .

Git commit -m “commit message”

Git push -u origin master