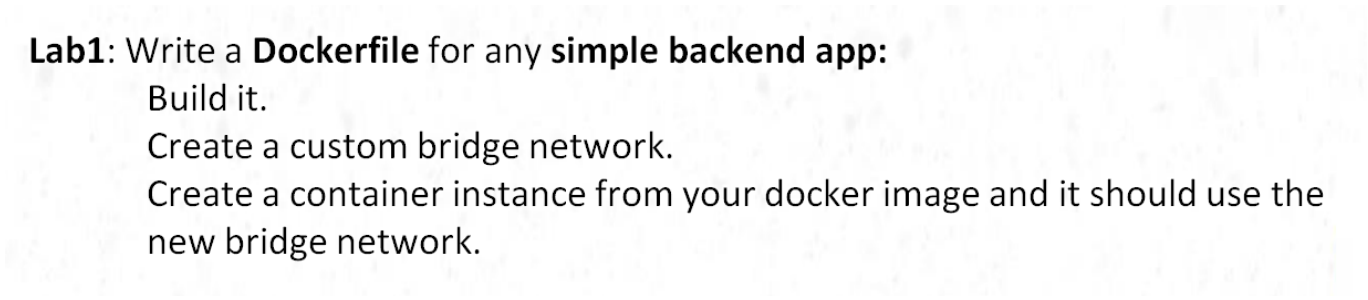
Lab1: Simple BackEnd App with Docker

Architecture



Step 1: Build the Backend Application

Using Node.js

```
# Initialize a new Node.js project
npm init -y

# Install Express framework
npm install express

# Run the application locally
node app.js
```

Step 2: Prepare the Docker Image

```
# Build a Docker image from the Dockerfile
docker build -t zerosploit .
```

Step 3: Create a Custom Bridge Network

```
# Create a custom Docker network
docker network create network-zerosploit --driver bridge
```

```
D:\Projects\NTI-ZeroSploit-main\NTI-ZeroSploit-main\Docker\Lab1 (lab1@1.0.0)
λ docker network create network-zerosploit --driver bridge
e9185856377328a851c7a75a27e44bca6cf719d7e80c5db282b3f0e3bdf0aa45

D:\Projects\NTI-ZeroSploit-main\NTI-ZeroSploit-main\Docker\Lab1 (lab1@1.0.0)
λ docker network ls
NETWORK ID     NAME                DRIVER    SCOPE
1a570fb2507f   bridge              bridge    local
2013fb3eccff   host                host      local
e91858563773   network-zerosploit   bridge    local
340df45ee96e   none                null      local
```

Step 4: Run the Container using the Custom Network

```
# Run the container with the custom network and port mapping
docker run -d --name task1 -p 5000:30000 --network e91858563773 zerosploit
```

```
D:\Projects\NTI-ZeroSploit-main\NTI-ZeroSploit-main\Docker\Lab1 (lab1@1.0.0)
λ docker run -d --name task1 -p 5000:30000 --network e91858563773 zerosploit
29c8223981d70c901e45e5729f75b9c6dc087ac601bfb67ded8d2e11f33bcc02
```

Final Output

We can now access the app through port 5000 and the container can communicate with other containers on the zerosploit network.