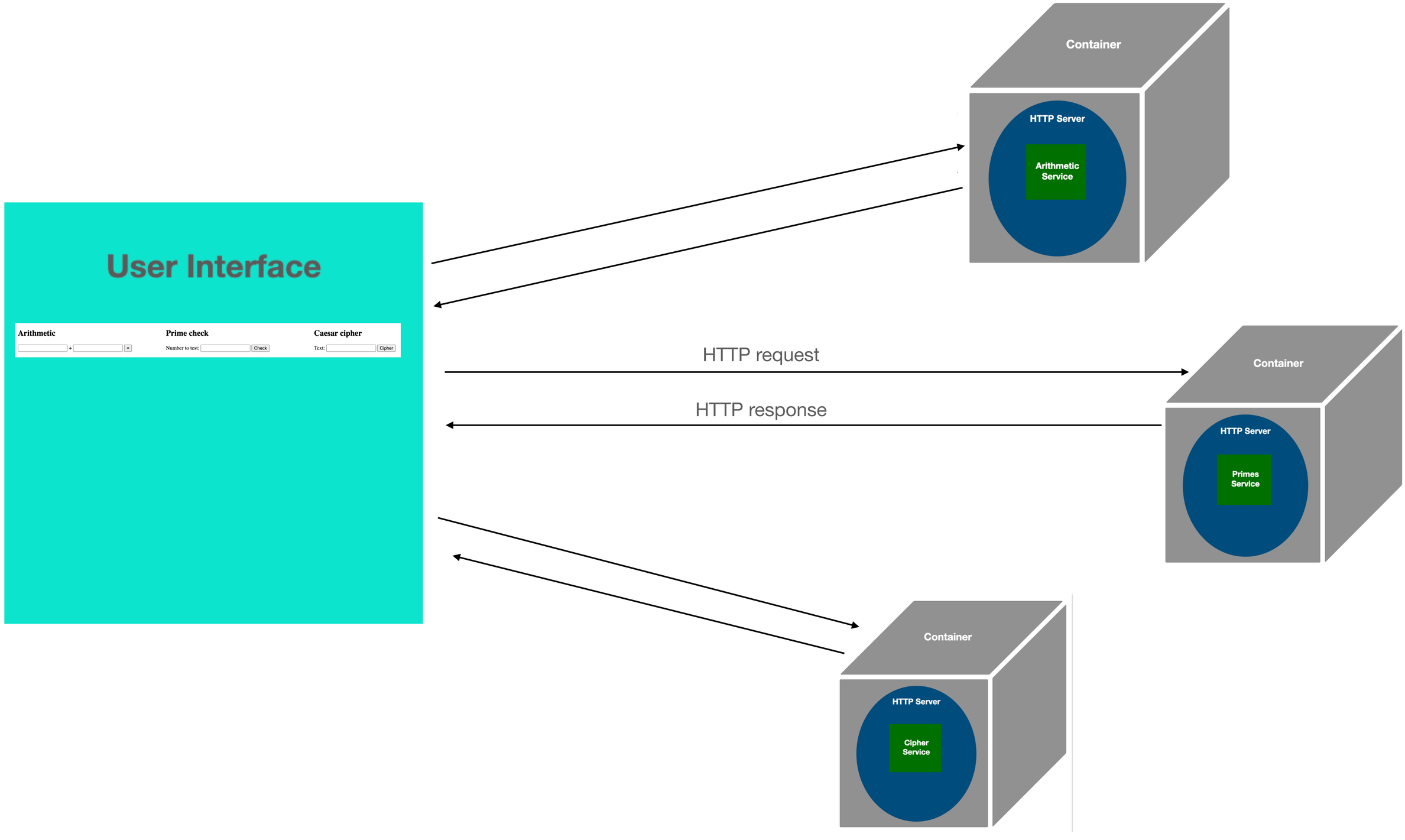
Practicum 10

CSCI-P 465/565 Software Engineering I

Indiana University Bloomington - Spring 2024

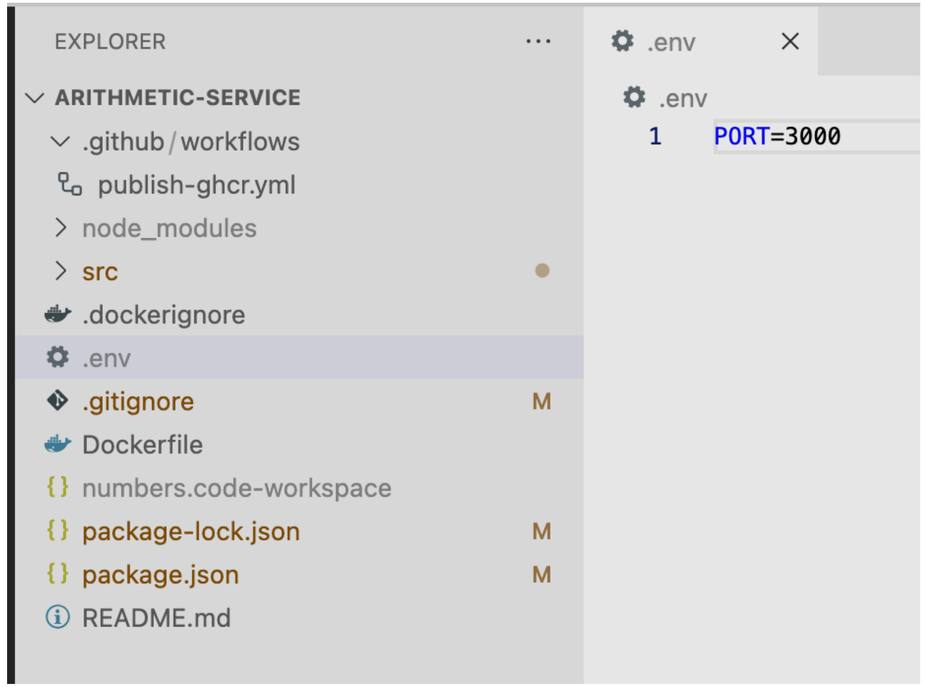

User Interface
Arithmetic
+
=
Prime check
Number to test:
Check
Caesar cipher
Text:
Cipher
Container
HTTP Server
Arithmetic Service
HTTP request
HTTP response
Container
HTTP Server
Primes Service
Container
HTTP Server
Cipher Service

New tools

- Docker Compose
  - Docker Compose is included with Docker and allows us to configure, build, run, and manage multiple containers at the same time. It's useful for development, but we won't use it in production.
- Azure Storage
  - Azure Storage is a service to store files in the cloud. We can manage the assets through the Azure Portal, through the APIs, or from the command line. We'll upload a video through the Azure Portal and then use the Node.js Azure Storage SDK to read it back.
- MongoDB
  - MongoDB is a popular NoSQL type of database. It's lightweight, easy to set up and use, and it's convenient for microservices.

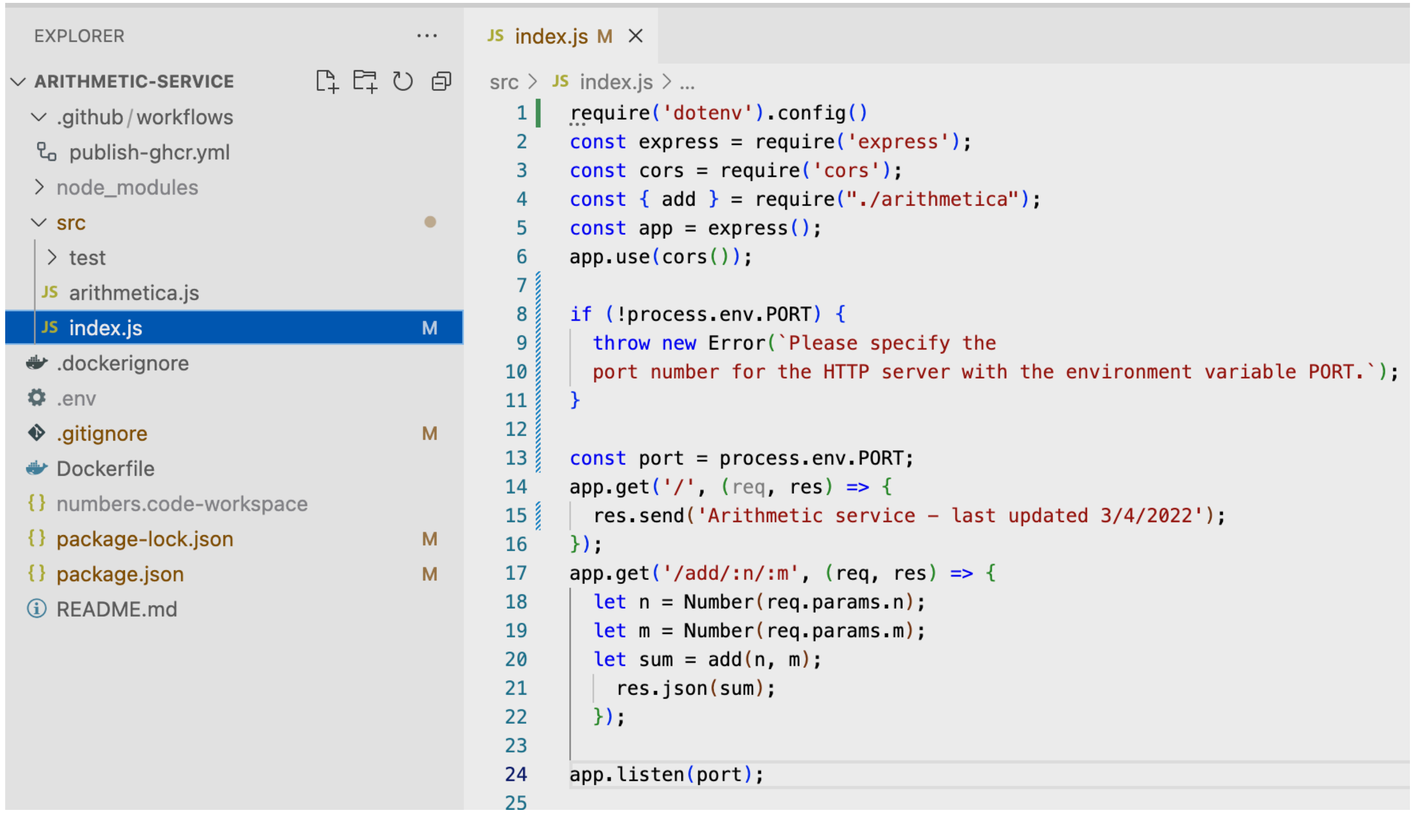
Steps

- Add a file called .env to your arithmetic-service folder.
- The .env file will store sensitive configuration data that you don’t want to push to Github. Make sure you add the .env file to your .gitignore.



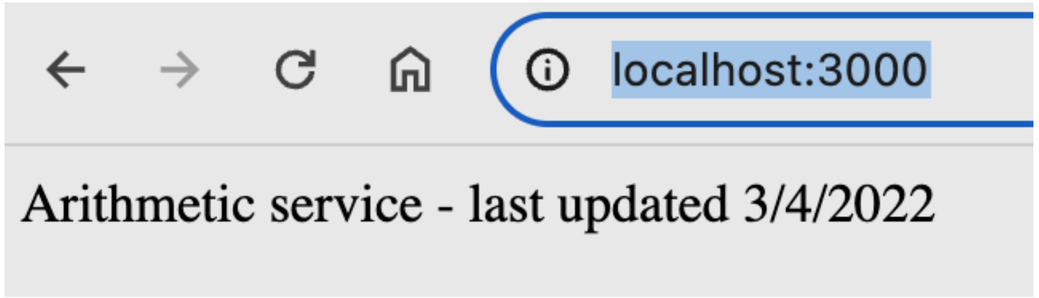
Steps

- Modify the index.js file to read the port number from the .env file.



Steps

- Install the dotenv module:
  npm install dotenv
- Run the service:
  node ./src/index.js
- Verify that the server is working:
  http://localhost:3000/

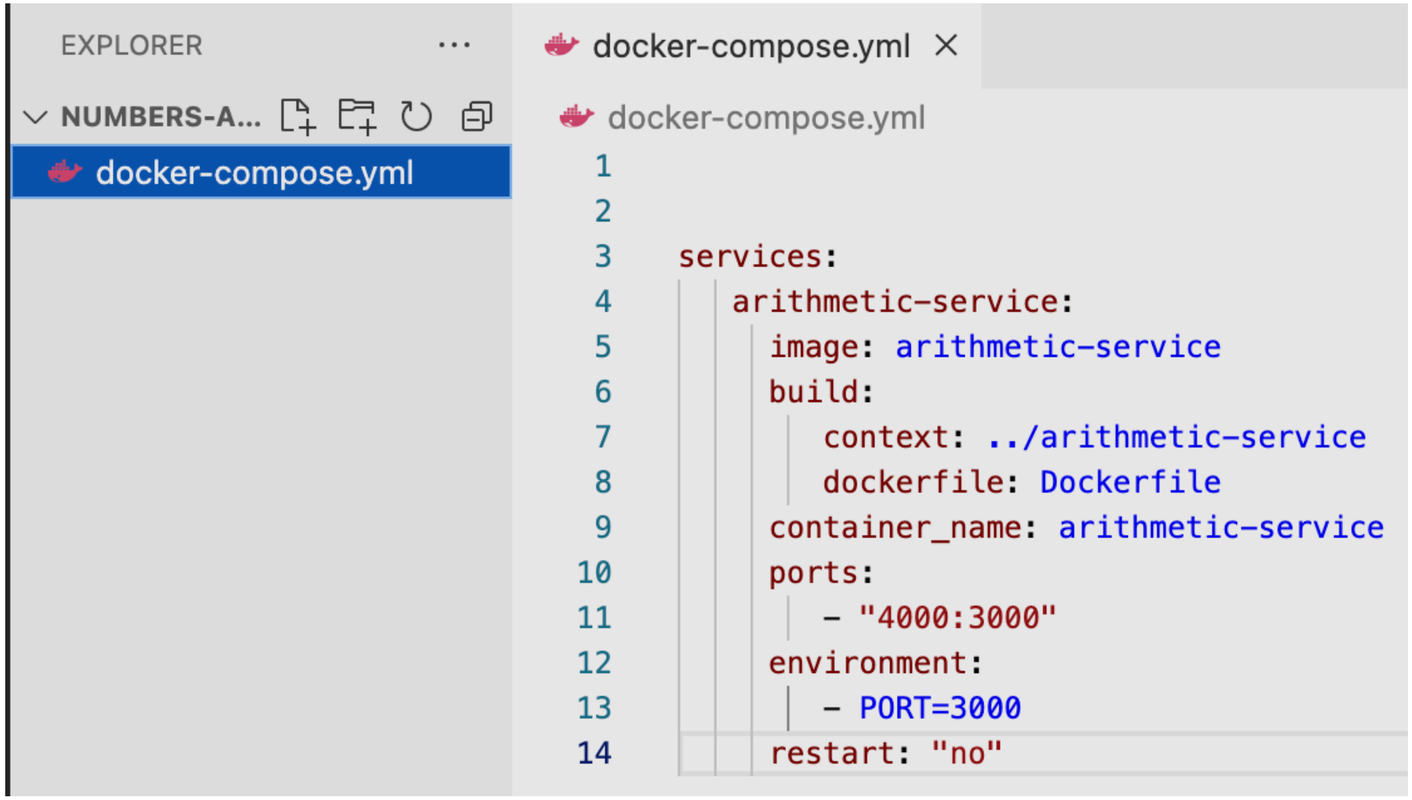


Steps

- Add a folder at the same level as your other backend folders and add a yaml file called docker-compose.yml to it.

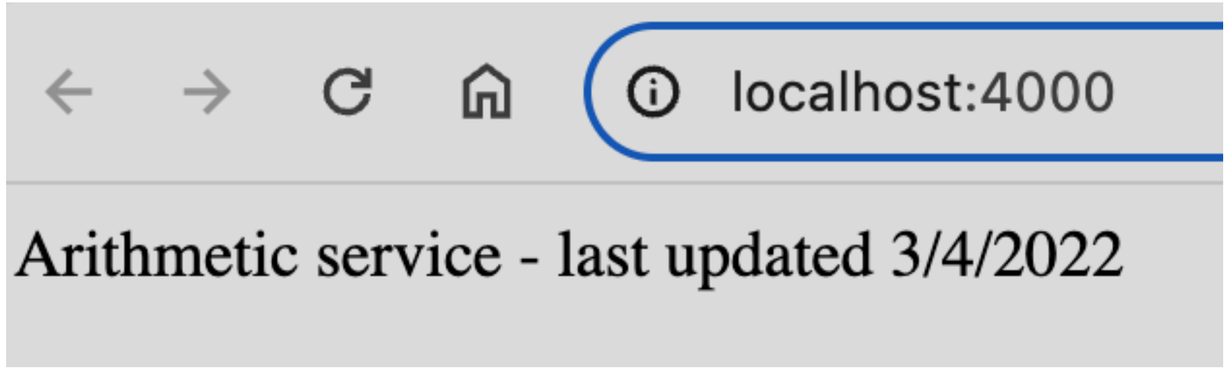
Steps

- Add the following to the yaml file to describe the image and the container that we want to be created for our arithmetic-service.



Steps

- Run the docker compose: **docker compose up --build**
- Verify that the container is running: http://localhost:4000/



Steps

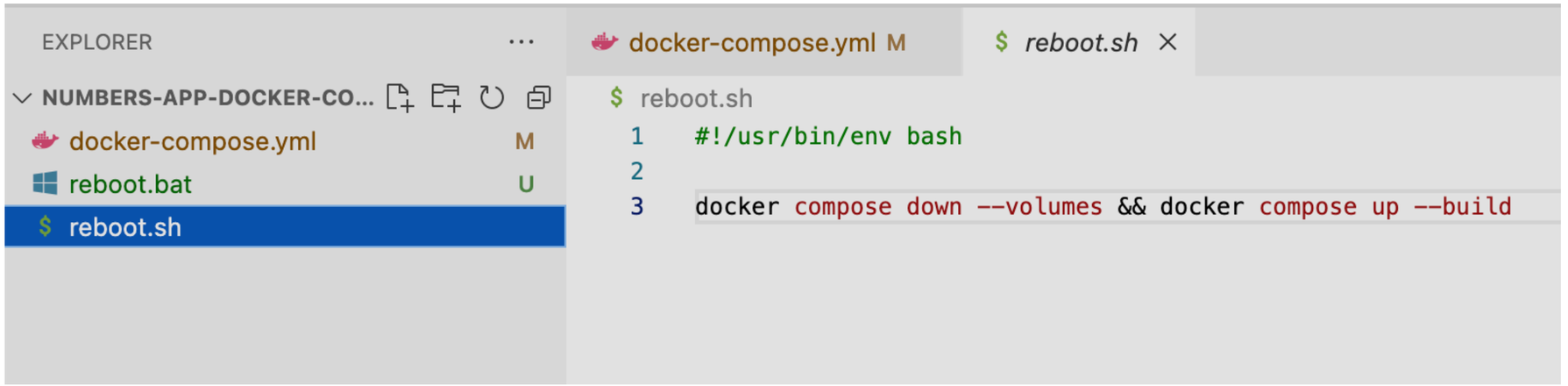
- To delete the container(s), run the following in the same folder where your docker-compose is located:
  docker compose down --volumes

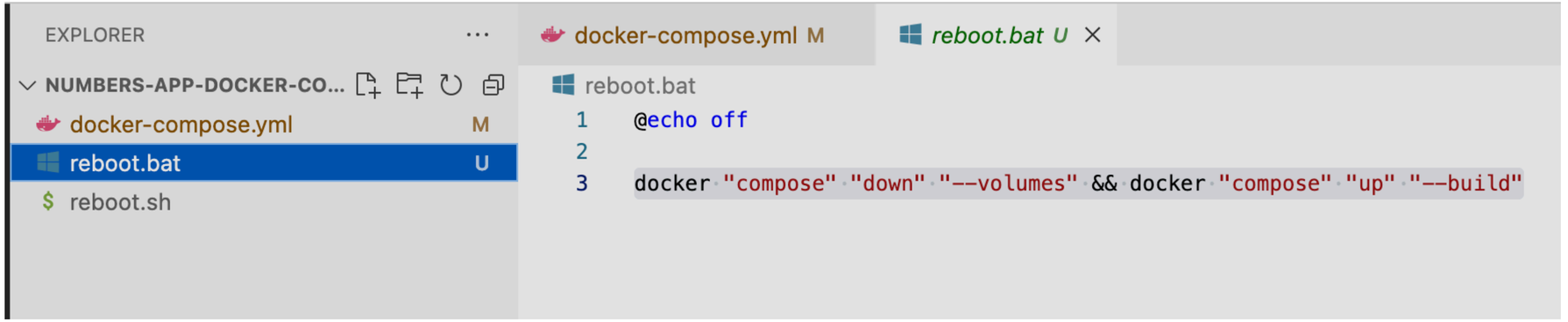
Steps

- Or we can combine both to reboot the application when we update the code:
  docker compose down --volumes && docker compose up —build

- You can create a bash file (or bat file for windows machines) containing the above command so that every time you need to stop-delete-create-start the docker images and containers, you can simply run the bash or bat file.

Steps


EXPLORER
NUMBERS-APP-DOCKER-CO...
docker-compose.yml M
reboot.bat U
reboot.sh
docker-compose.yml M
reboot.sh
reboot.sh
1 #!/usr/bin/env bash
2
3 docker compose down --volumes && docker compose up --build


EXPLORER
NUMBERS-APP-DOCKER-CO...
docker-compose.yml M
reboot.bat U
reboot.sh
docker-compose.yml M
reboot.bat U
reboot.bat
1 @echo off
2
3 docker "compose" "down" "--volumes" && docker "compose" "up" "--build"

Data Management

Tow types of data

- File storage
- Database
  - SQL databases
  - NOSQL databases

File storage using Azure

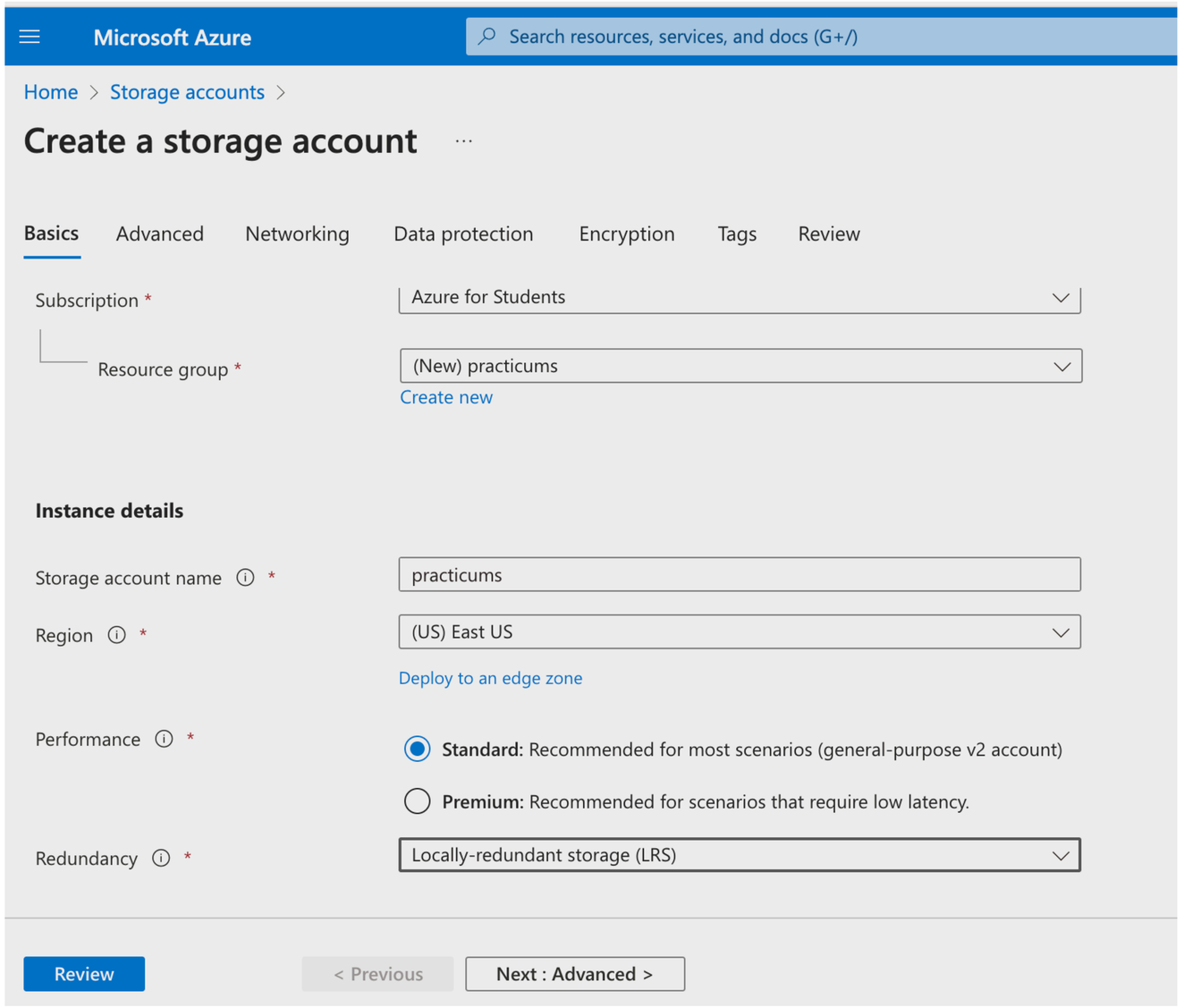
- *Azure Storage* is a Microsoft Azure service for hosting private or public files in the cloud. You upload your files to Azure Storage and can then access these through the Azure Storage API.

Steps

- Create a free Azure for students account: https://azure.microsoft.com/en-us/free/students
- Login to Azure portal: https://portal.azure.com

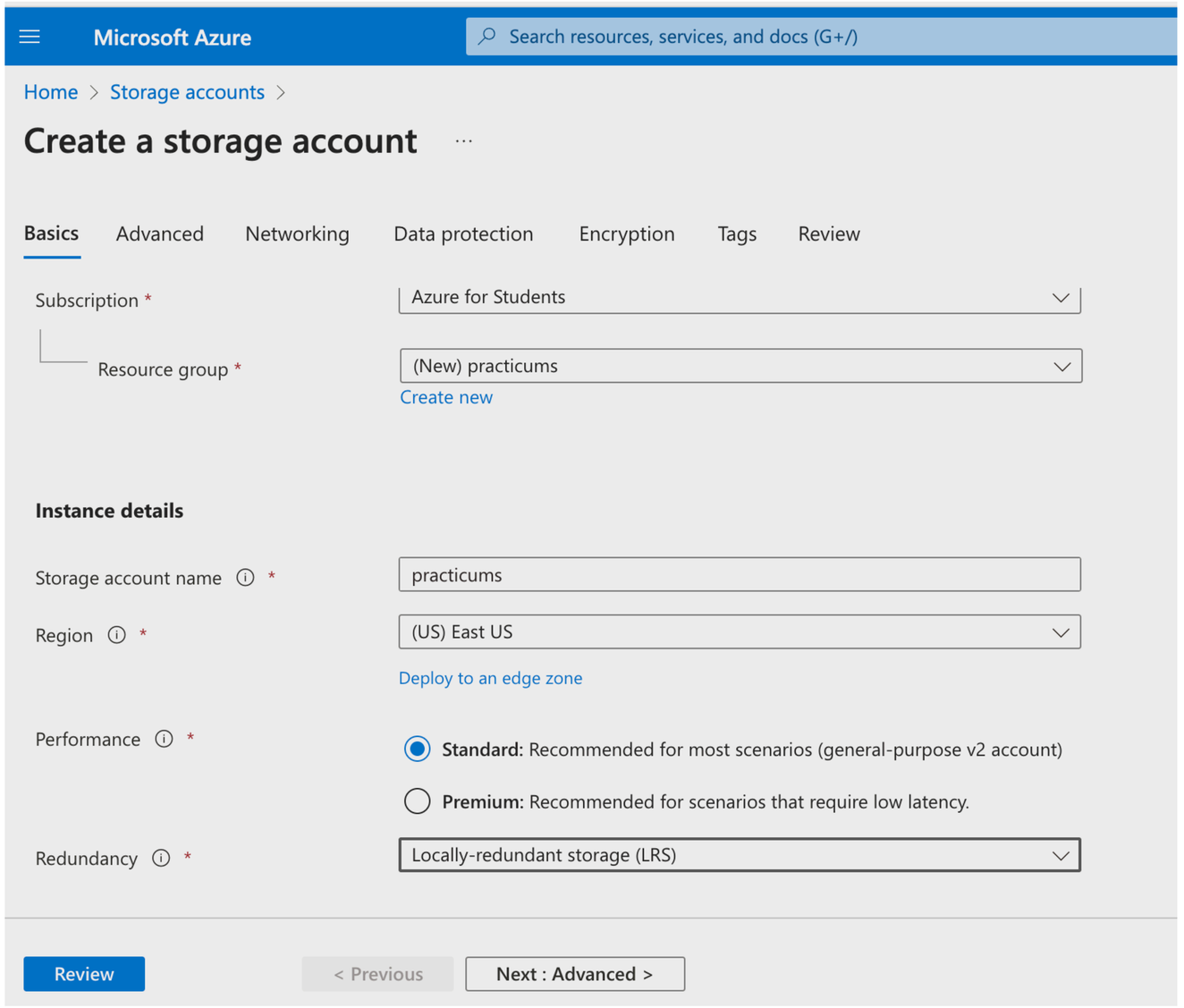
Steps

- Create a storage account called “practicums”:



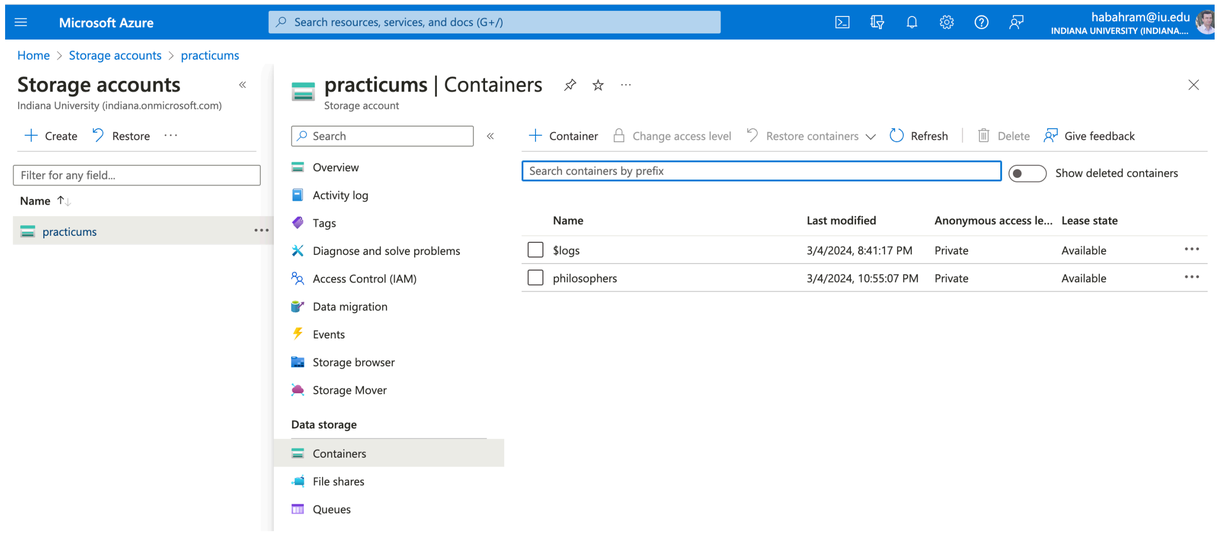
Steps

- Create a storage account called “practicums”:



Steps

- Then create a folder called “philosophers” and add a picture file to it. In Azure storage system a folder is called a container! But that “container” has nothing to do with “docker container”. It is simply a folder.



The azure-storage-service

Steps

- Create a folder called “azure-storage-service” in the same folder where your other services are located.
- Change into the new directory:
  - **cd azure-storage-service**
- Create an empty nodejs project:
  - **npm init -y**
- Install Express:
  - **npm install express**

Steps

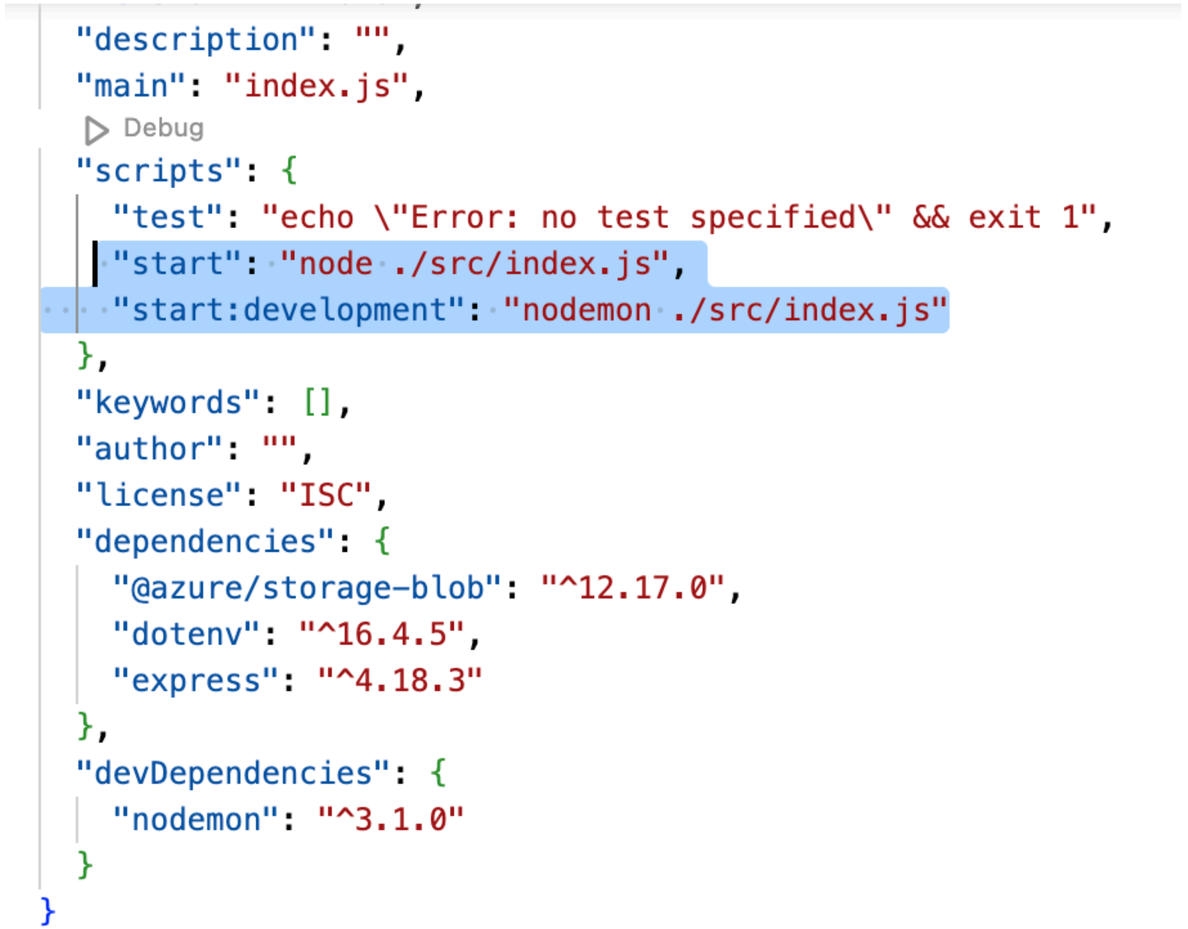
- Add the azure storage package to your service:
  npm install --save @azure/storage-blob

- Install dotenv module: **npm install dotenv**

- Install nodemon to run the server in the development mode:
  npm install --save-dev nodemon

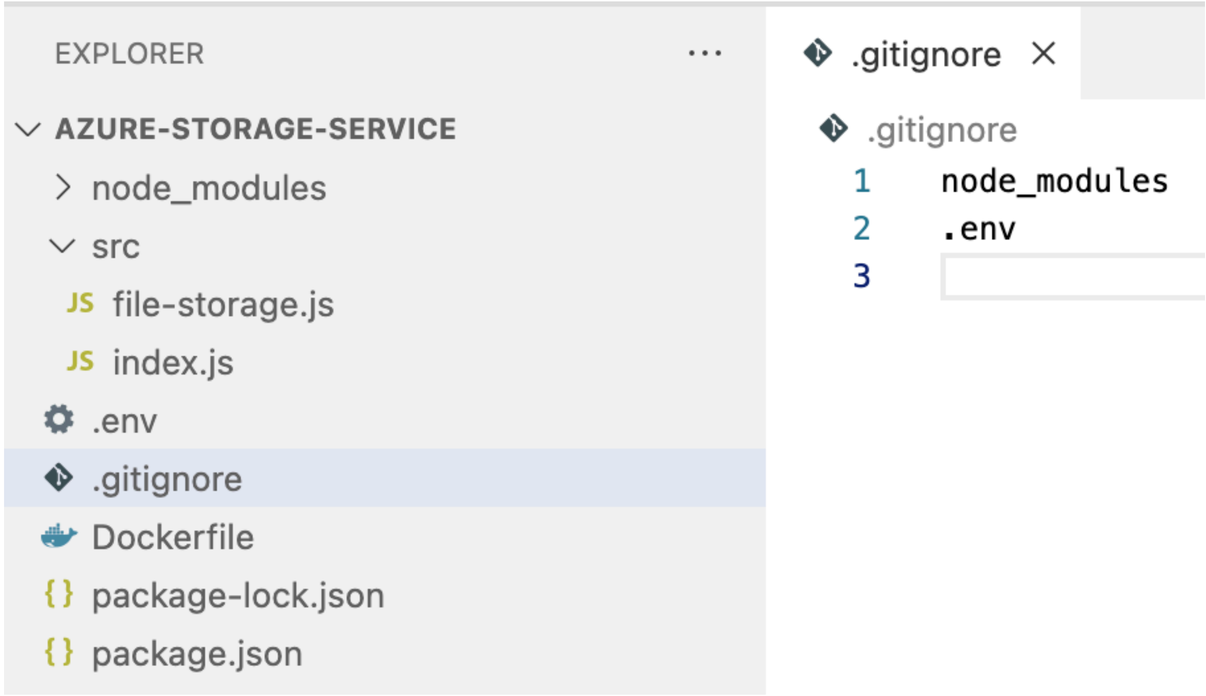
Steps

- Add the start commands to the package.json file:



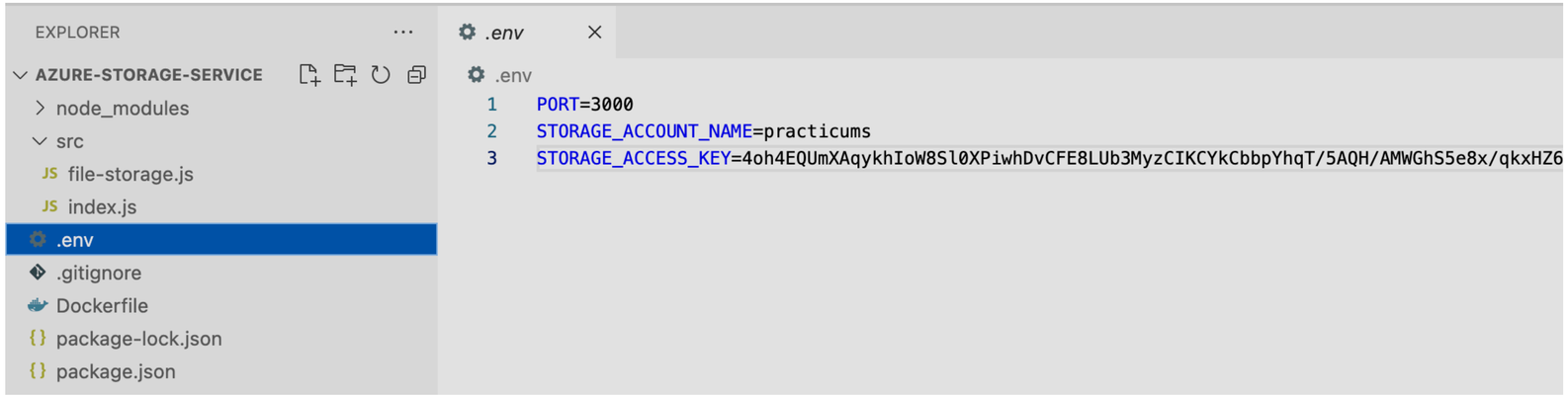
Steps

- Add a .gitignore file to the folder:



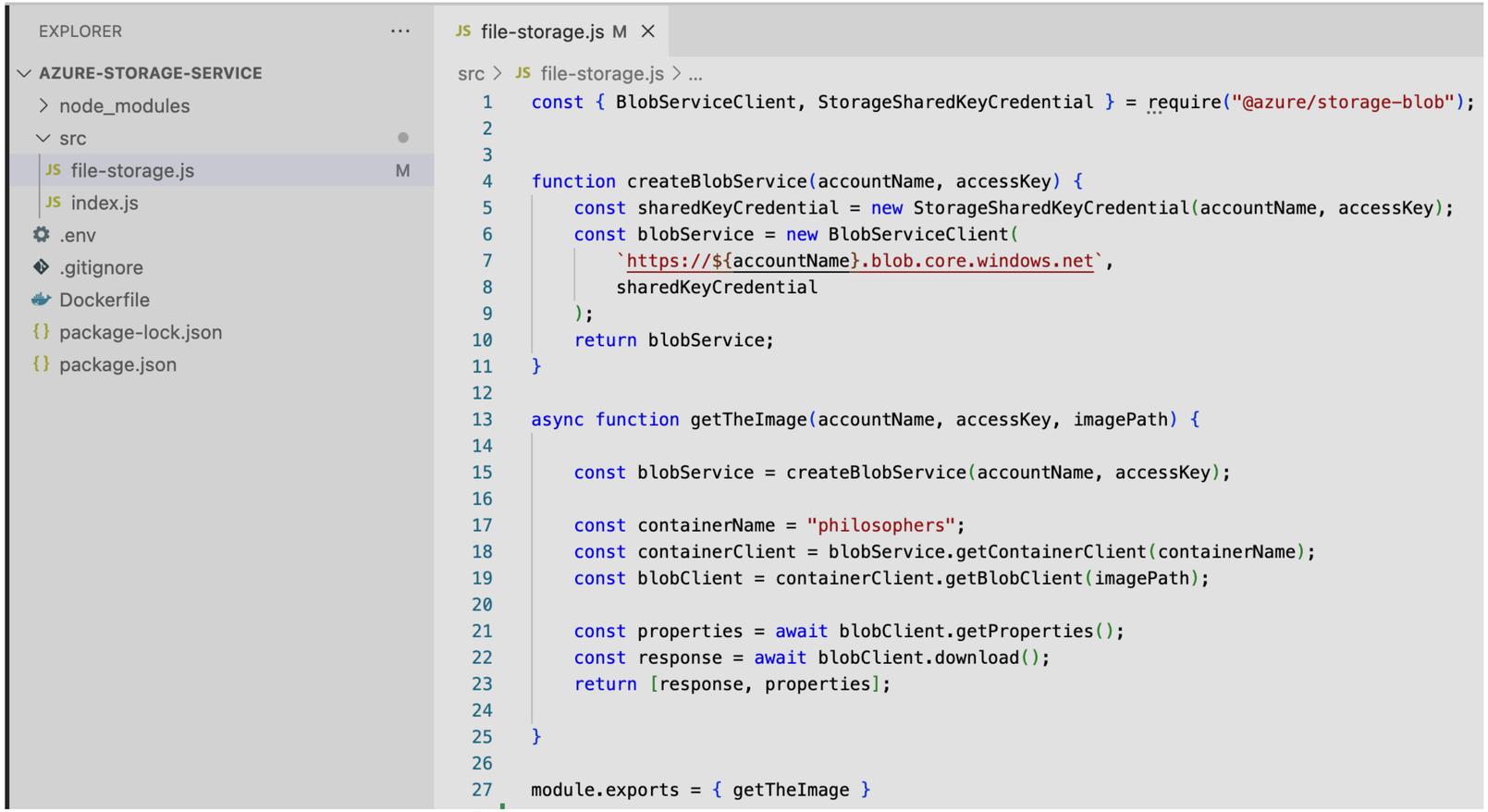
Steps

- Add a .env file to the folder.
- To find your access key go to your azure portal and navigate to your storage account. Under the Access Keys page you can find two keys. Use any of those.



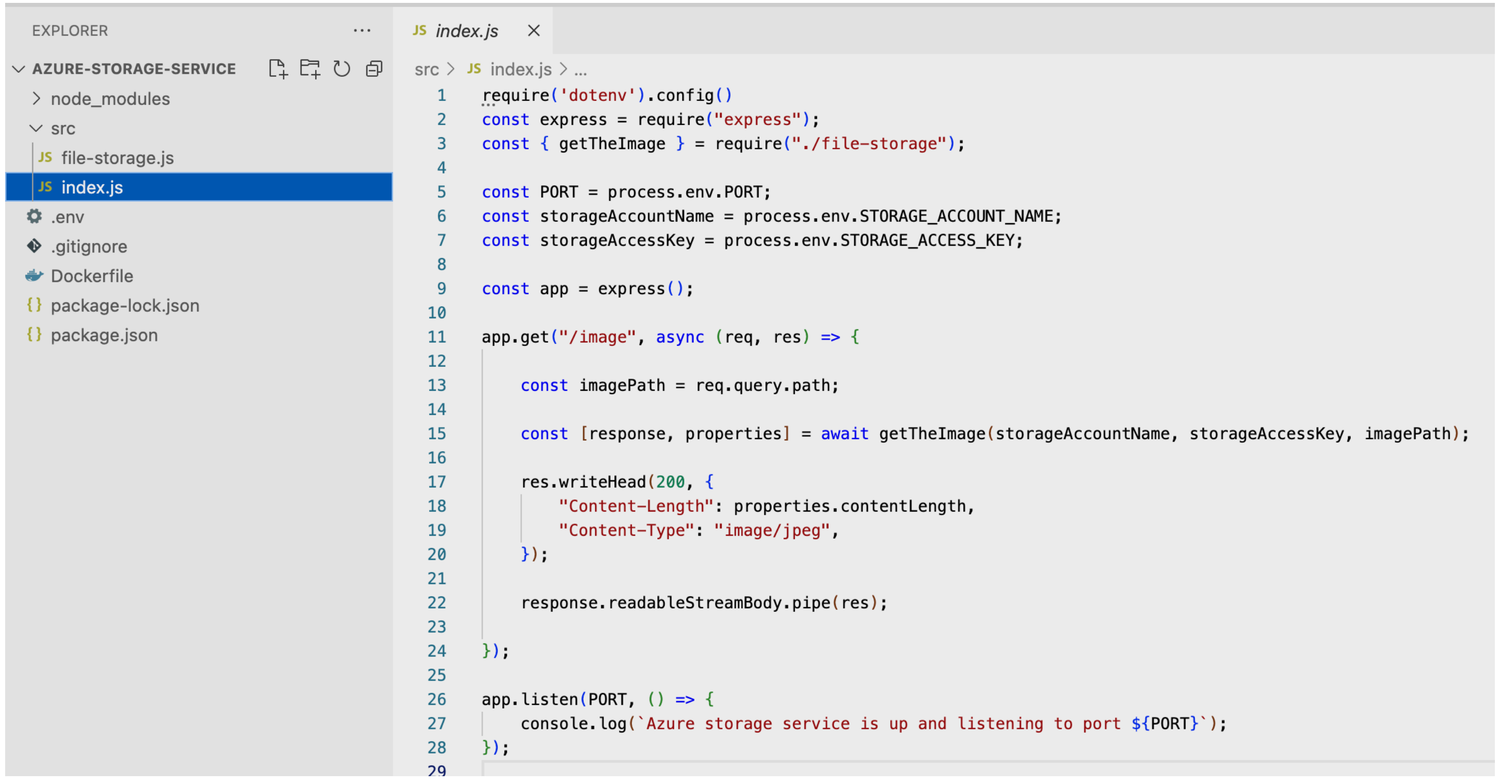
Steps

- Create a src folder and add a file called “file-storage.js”.



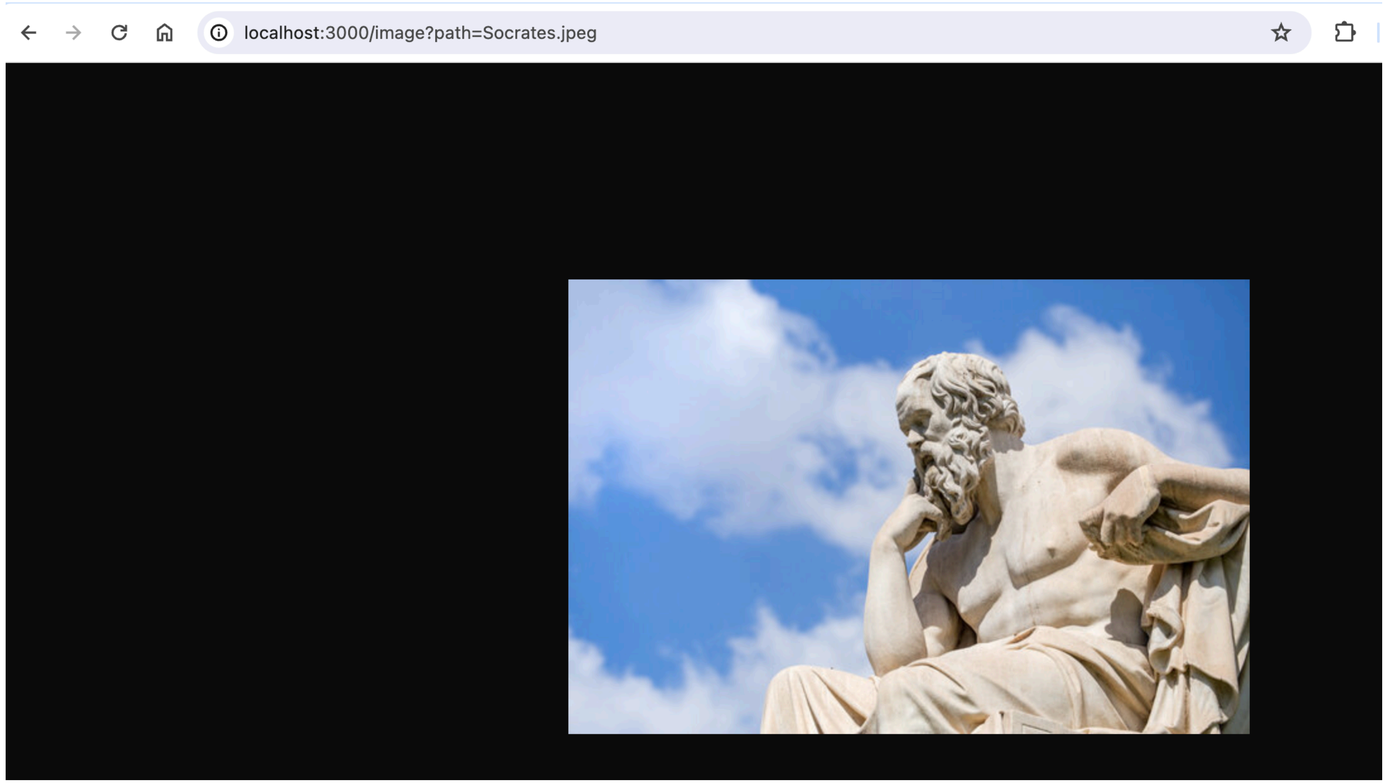
Steps

- In the src folder add the index.js file:



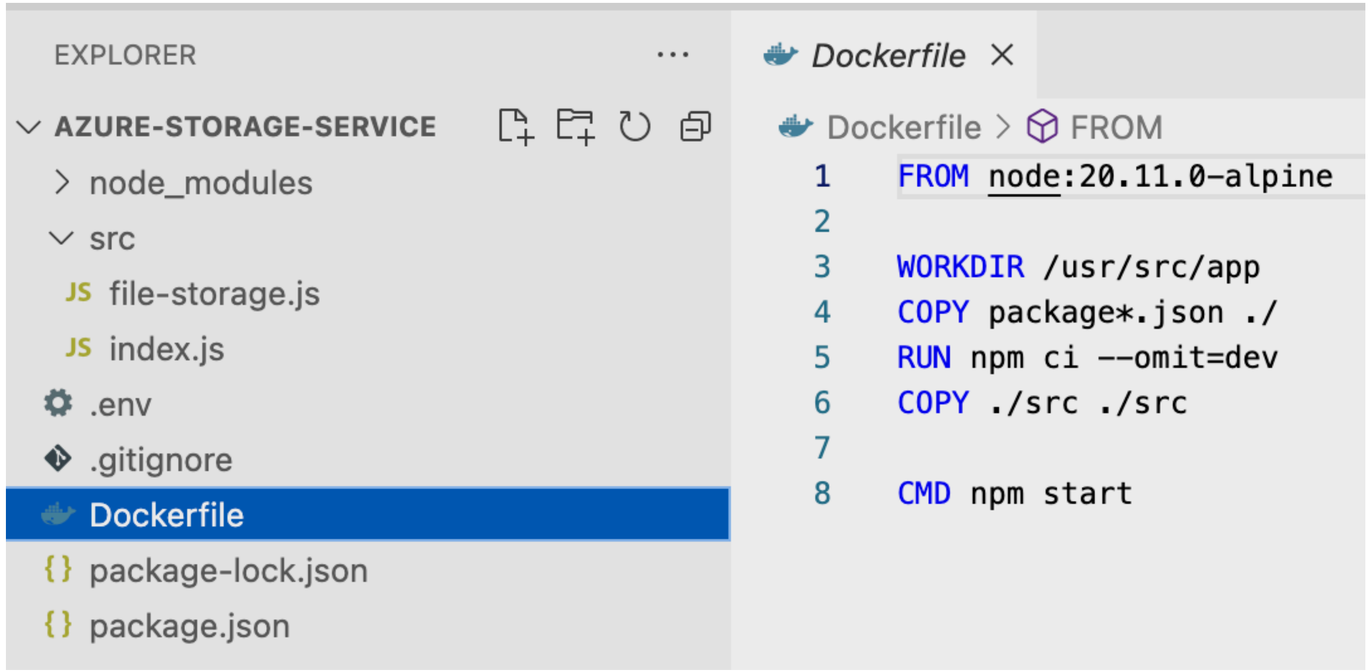
Steps

- Run the application in the development mode:
  npm run start:development
- Verify that it is working.



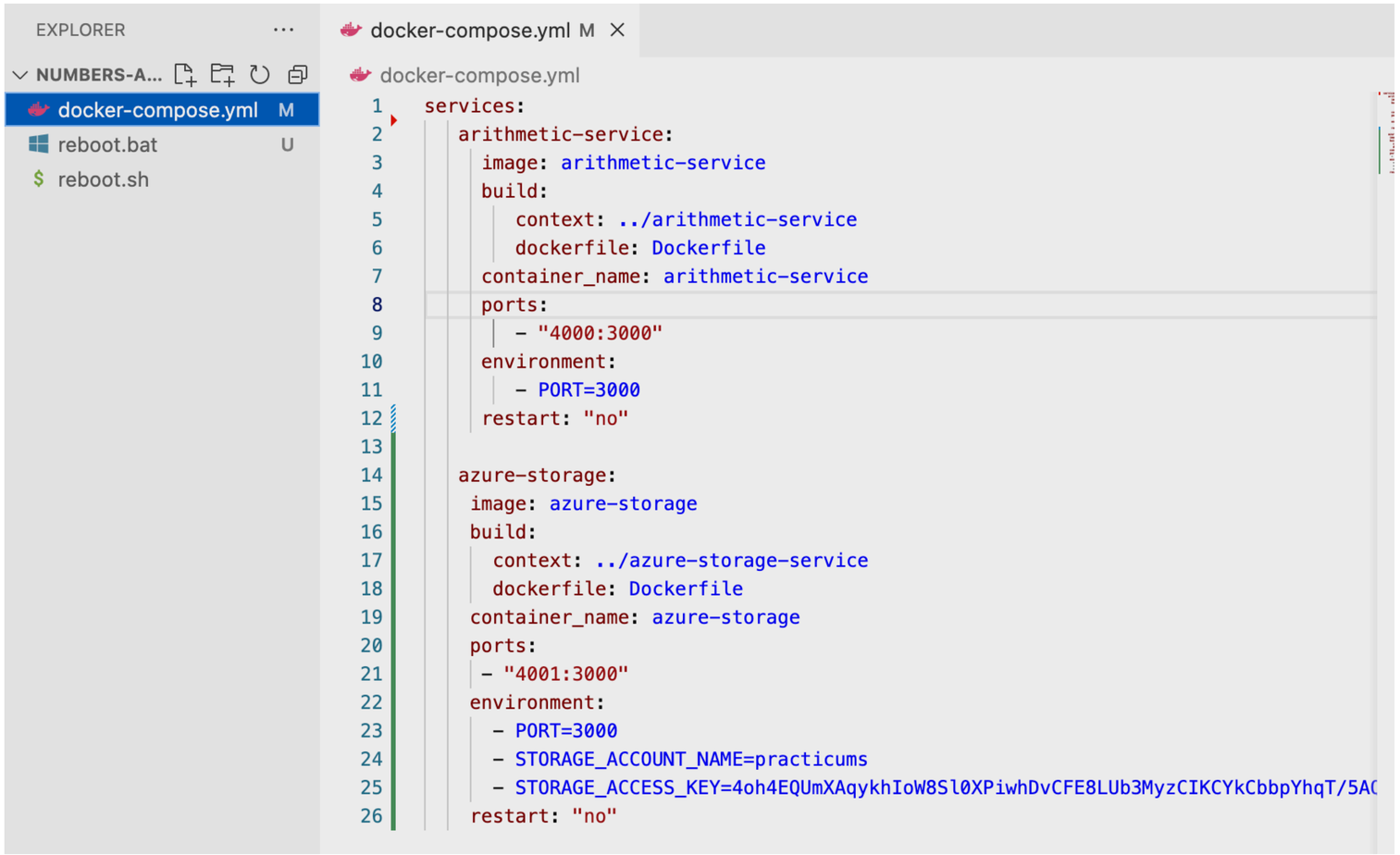
Steps

- Add the docker file:



Steps

- Add the service to your docker-compose file:

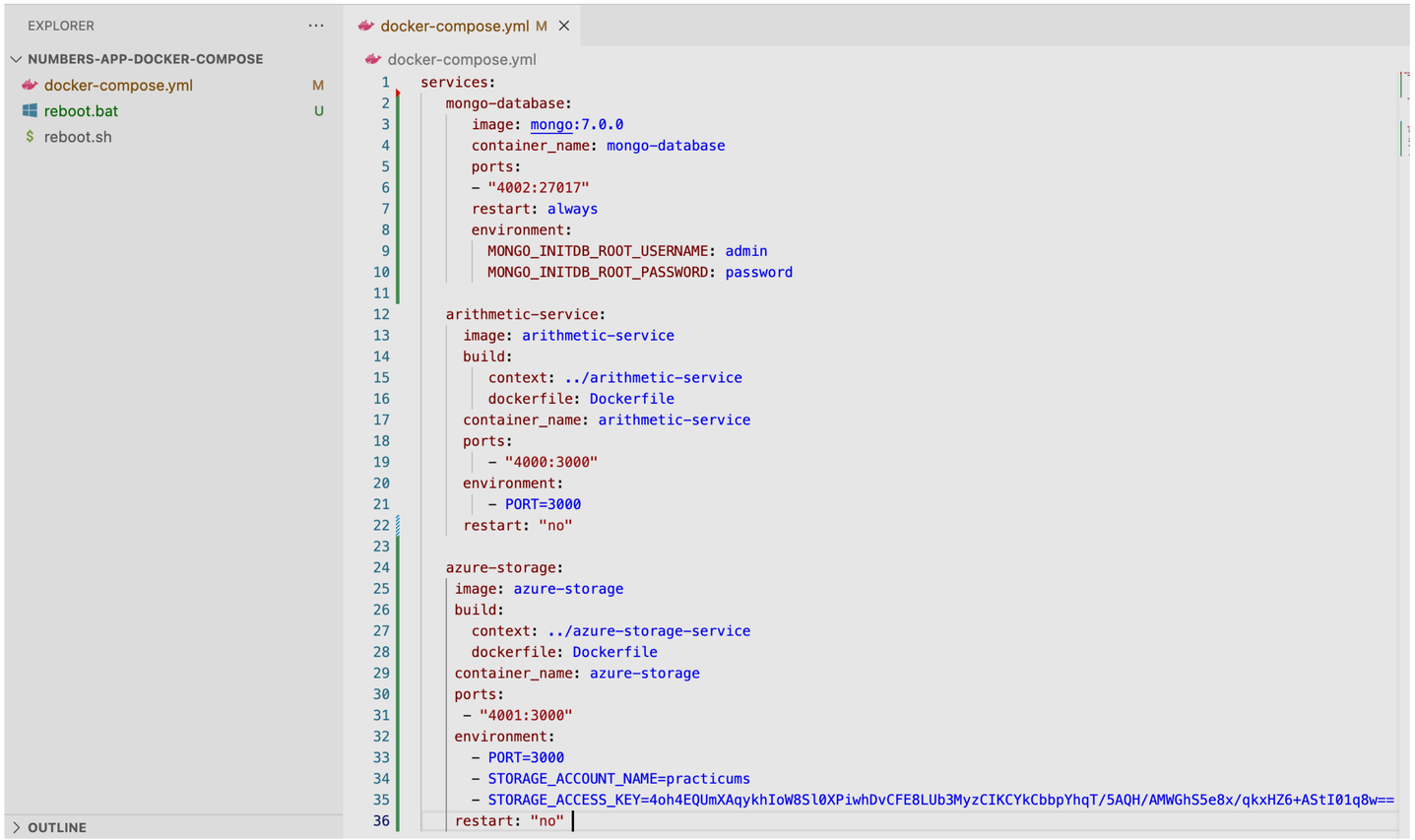


- Run the reboot.sh file (reboot.bat file) and verify that both containers are up and running.

Adding a database

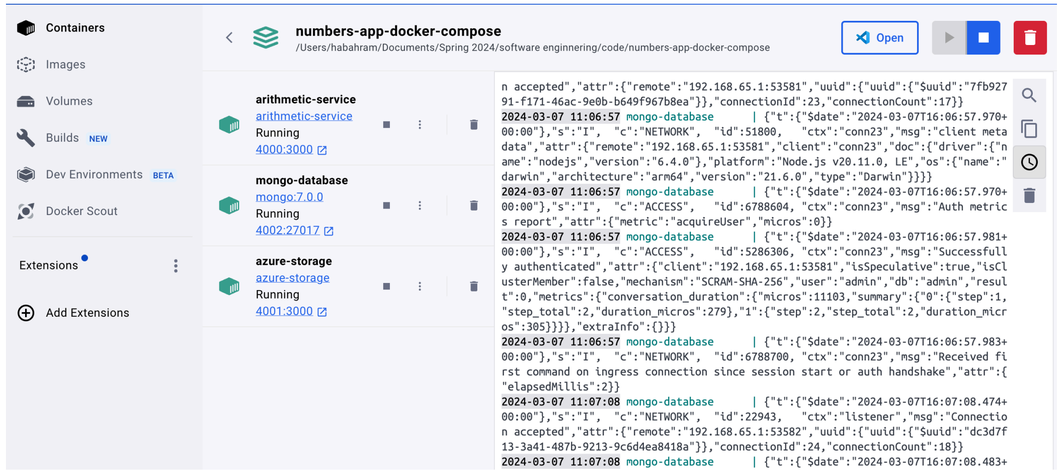
Steps

- Add a mongodb service to your docker-compose.yml file:



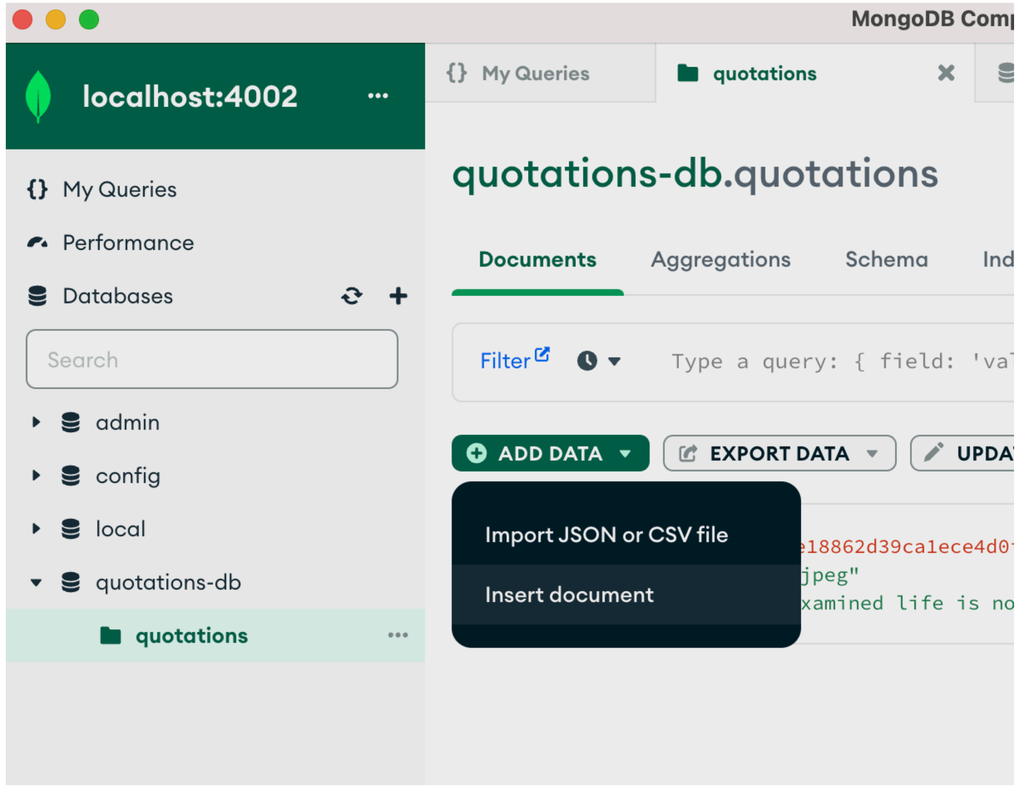
Steps

- Restart the containers by running your reboot.sh (or reboot.bat on windows).



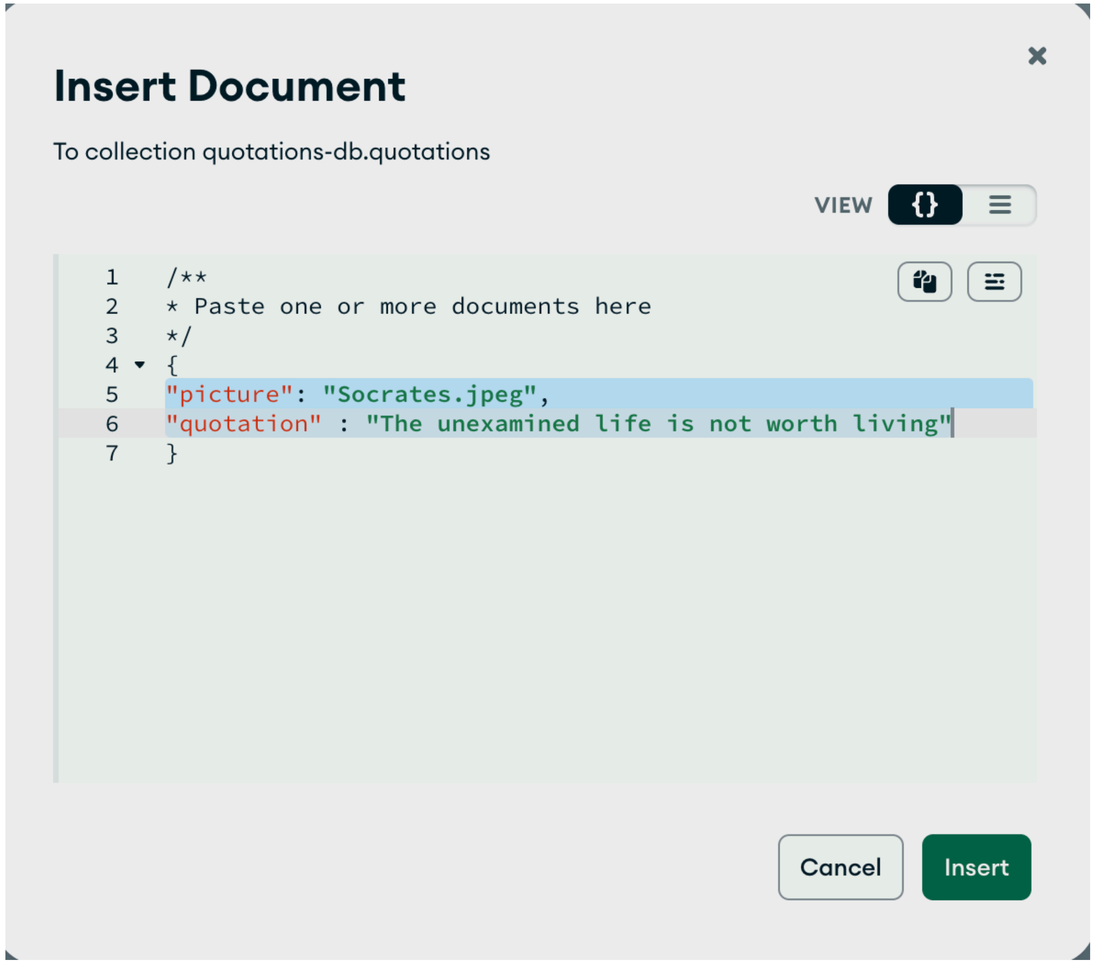
Steps

- Install MongoDB Compass tool: https://www.mongodb.com/products/tools/compass
- Open the tool and connect to your mongo-database container.
- Create a database called “quotations-db” and in that database create a collection called “quotations”
- In the collection, create a document:



Steps

- Add the following to keys to the document:



Quotation service

Steps

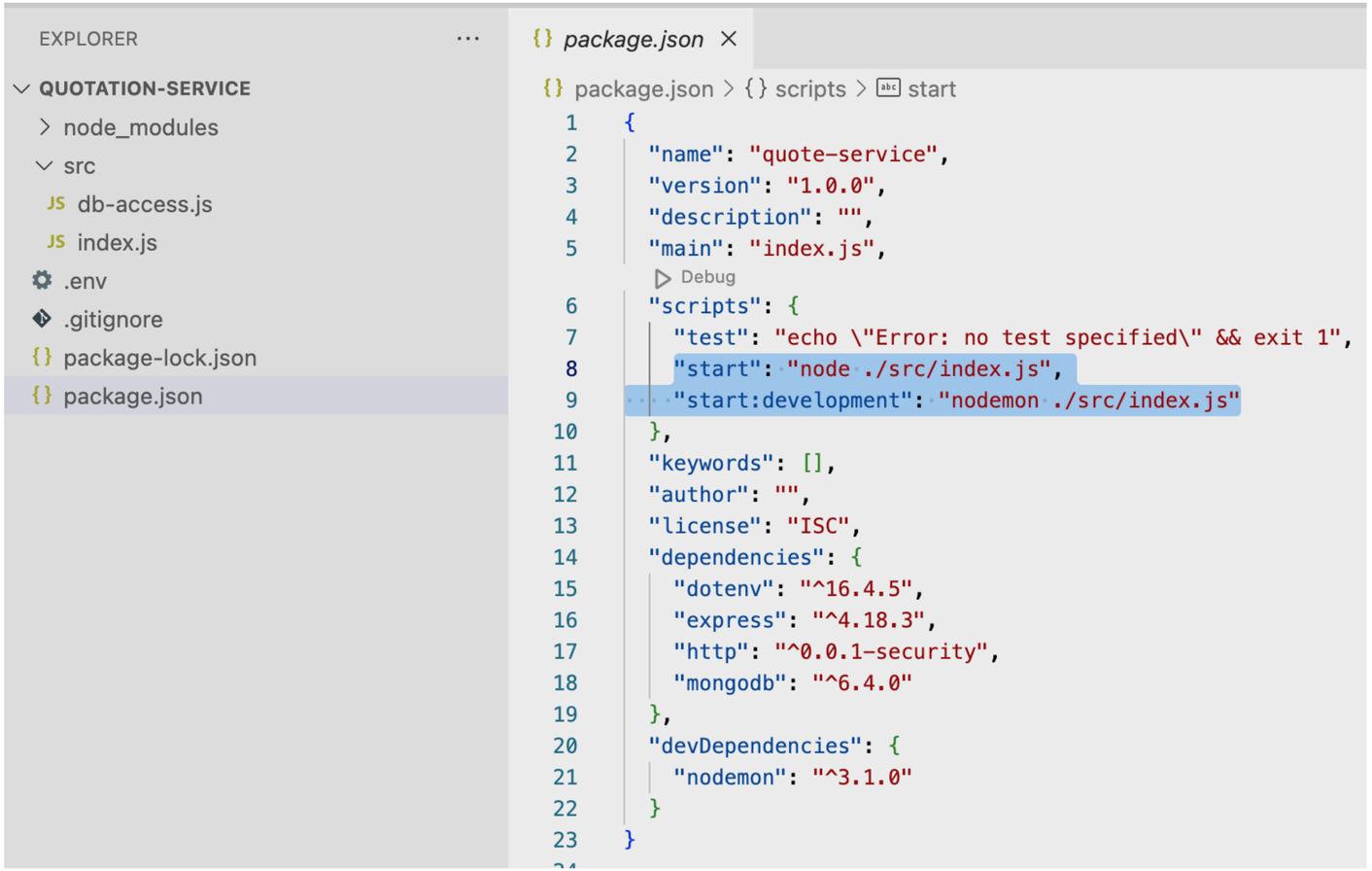
- Create a folder called “quotation-service” in the same folder where your other services are located.
- Change into the new directory:
  - **cd quotation-service**
- Create an empty nodejs project:
  - **npm init -y**
- Install Express:
  - **npm install express**

Steps

- Install dotenv module: **npm install dotenv**
- Install nodemon to run the server in the development mode: **npm install --save-dev nodemon**
- Install the mongodb package: **npm install mongodb**

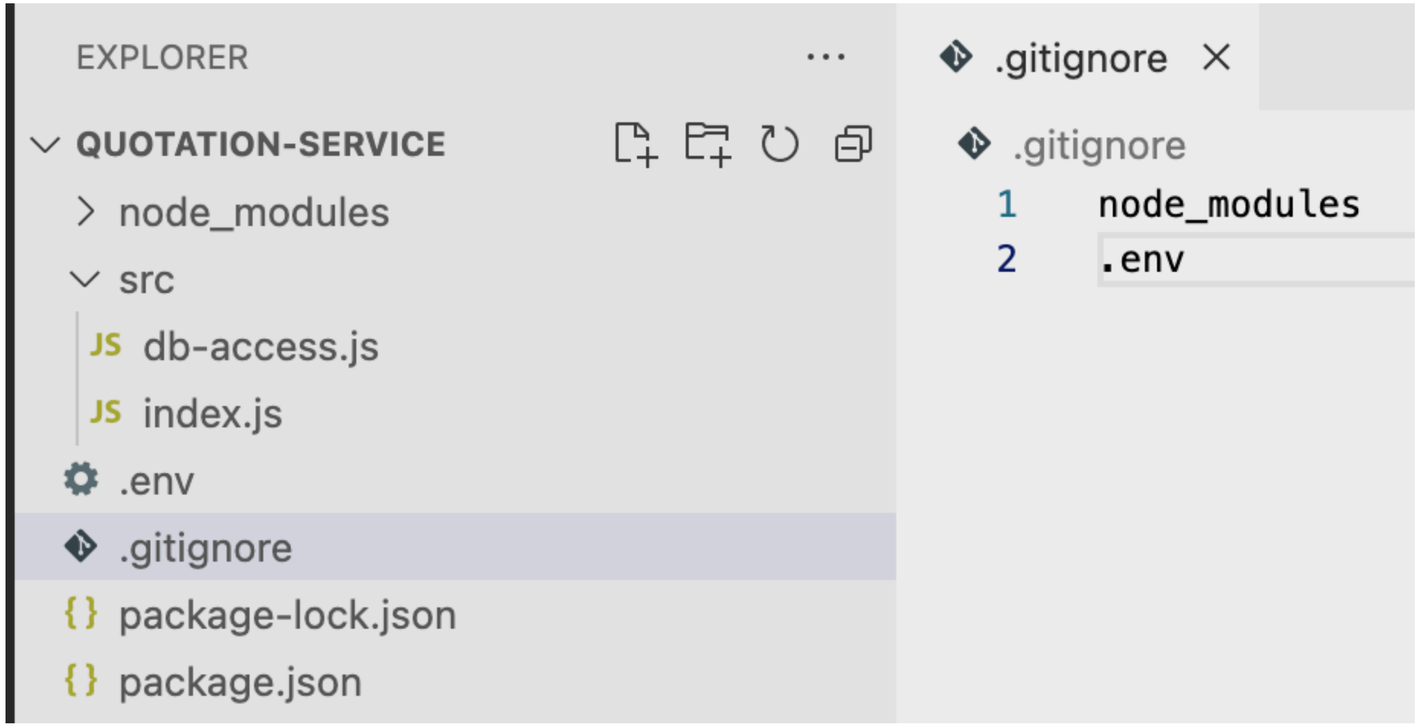
Steps

- Add the start commands to the package.json file:



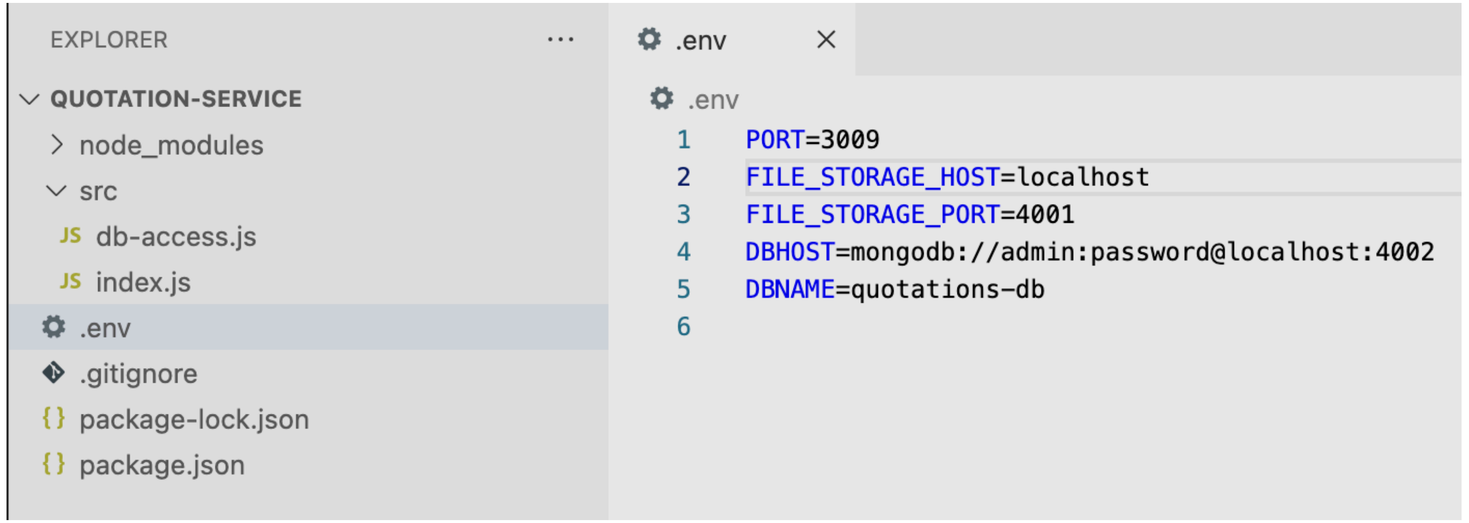
Steps

- Add a .gitignore file to the folder:



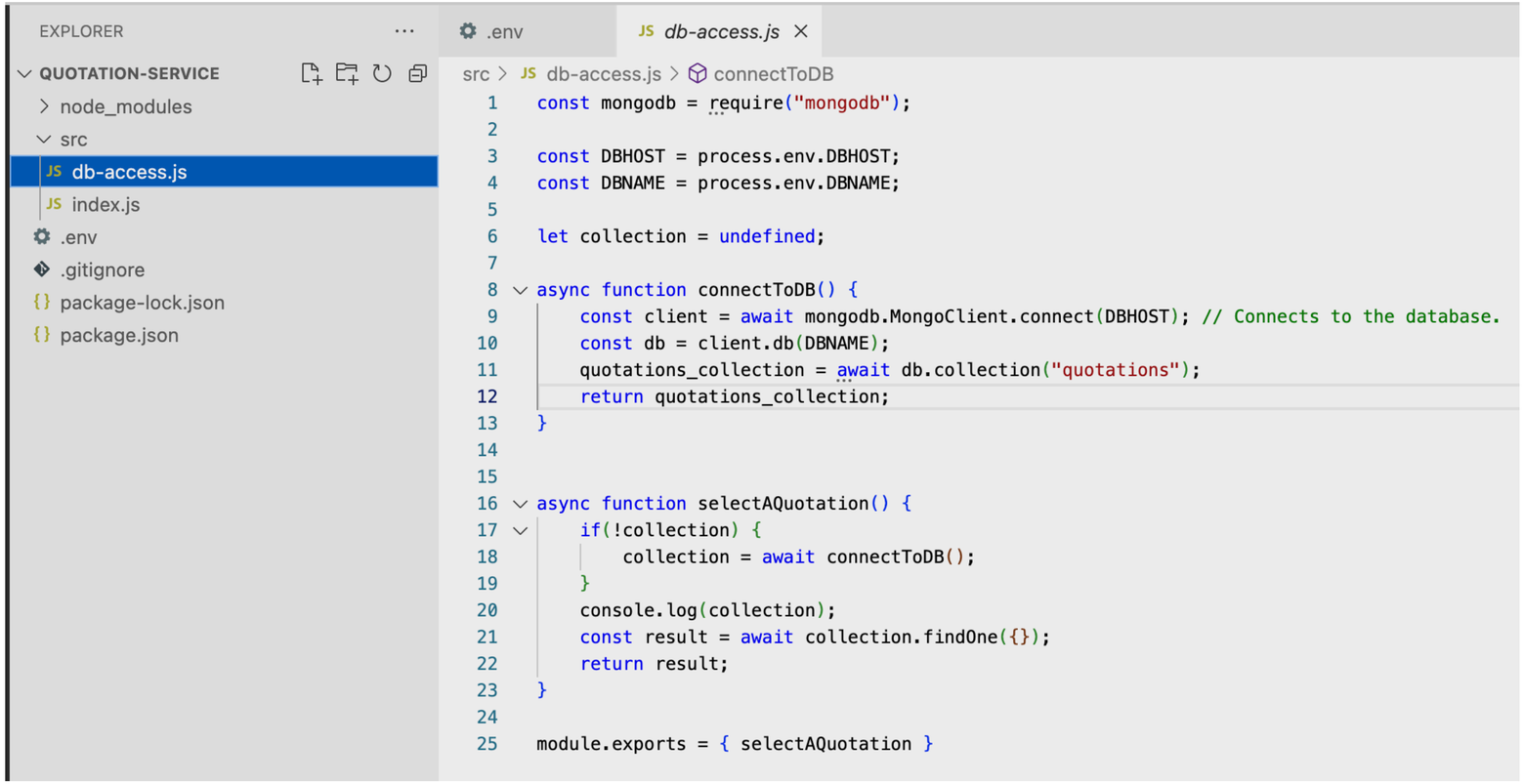
Steps

- Add a .env file to the folder.



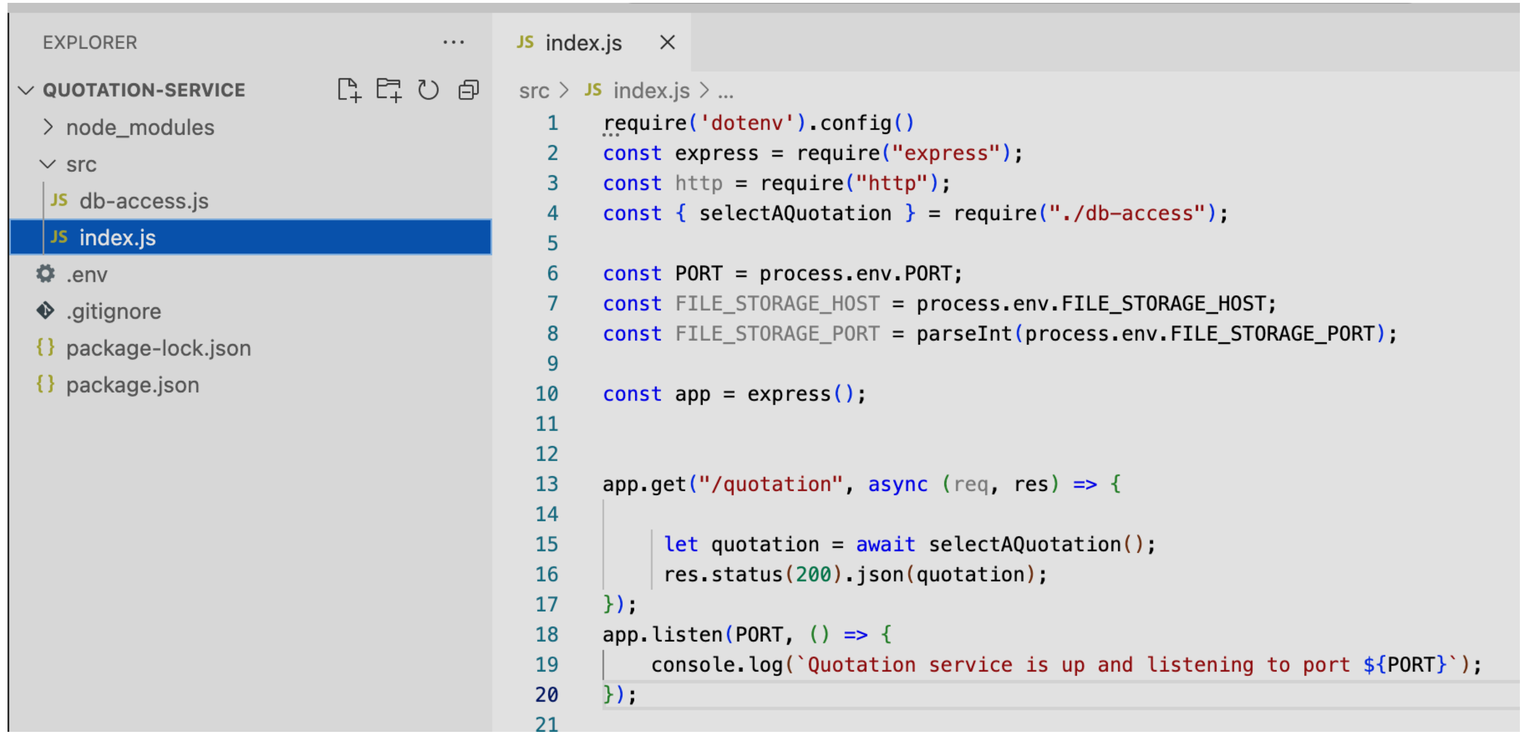
Steps

- Create a src folder and add two js files to it: index.js and db-access.js.
- Add the following to the db-access.js file:

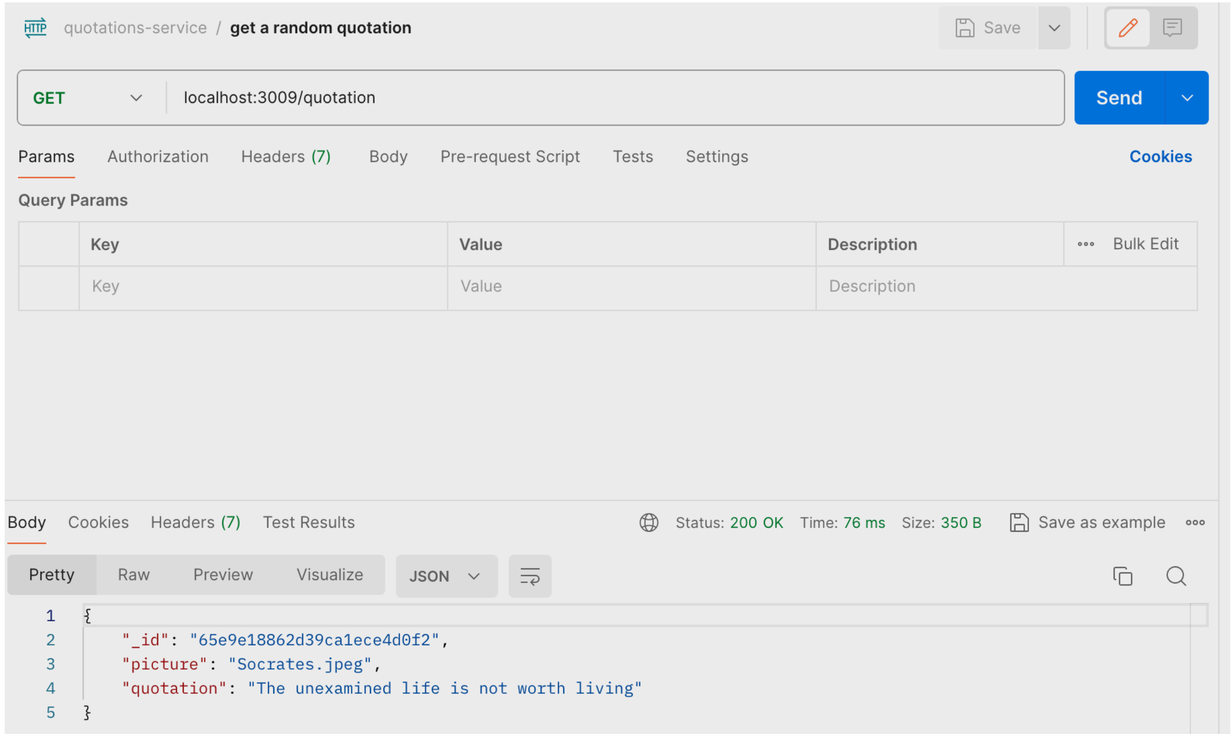


Steps

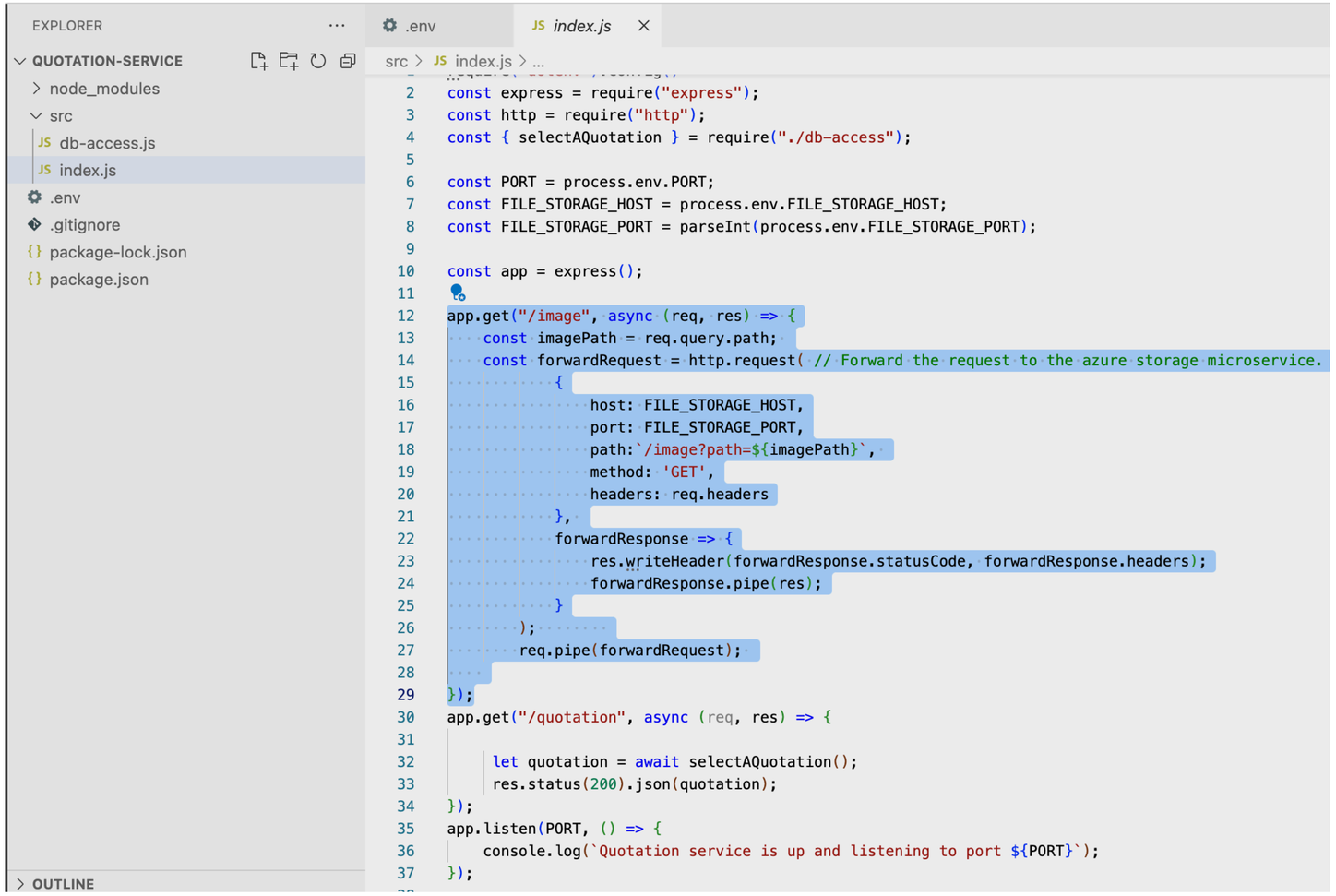
- Add the following to the index.js file:



- Using postman, verify that the endpoint “/quotations” is working.



- Add the “/image” endpoint to your index.js file:



- Rerun the application.
- Using postman verify that the endpoint is working: