Report- Challenge 3

Full-Stack Application Setup Using Docker

Introduction

Objective: The purpose of this project is to demonstrate how to set up a full-stack application using Docker. Docker simplifies deployment by creating isolated environments (containers), making it easy to manage, deploy, and scale applications.

Scope: The project includes three main components: a Node.js application, a MariaDB database, and a Nginx web server.

Prerequisites

- **Software Requirements**: Docker and Docker Compose are necessary tools for this project. You can install Docker from here and Docker Compose from here.
- **Basic Knowledge**: Some basic understanding of Docker concepts like images, containers, and Dockerfiles will be helpful.

Project Structure Overview

- **Directory Layout**: The project directory consists of directories for each component (api, db, nginx) and essential files such as **docker-compose.yml**, **.env**, and configuration files (**nginx.conf**, **server.js**, **init.sql**).
- **Key Files Explained**:
    - **docker-compose.yml**: Manages the multi-container setup.
    - **Dockerfile**: Contains instructions for building Docker images for each component.
    - **.env**: Contains environment variables required for configuration.
    - **nginx.conf**: Configuration file for the Nginx server.
    - **server.js**: Script for the Node.js application.
    - **init.sql**: SQL script to initialize the database.

Detailed Setup Steps

1. **Creating the .env File**:
    - Create a **.env** file and define necessary environment variables like **DB_USERNAME**, **DB_PASSWORD**, and others as mentioned in the task sheet.
2. **Writing Dockerfiles**:
    - **Node.js Application**: Set up the Dockerfile in the Node.js application directory, explaining each command (**FROM**, **COPY**, **RUN**, **CMD**).

- **Database Initialization**: Configure the Dockerfile in the database directory and explain the initialization process with **init.sql**.
- **Nginx Configuration**: Explain the Dockerfile setup for Nginx and how to configure the **nginx.conf** file.

***Note**: Make sure the path you give in dockerfile to COPY are given considering the root directory where respective Dockerfile is situated so here as example I had to change

COPY docker/api/package*.json ./ *to*

COPY package*.json ./

3. **Setting up docker-compose.yml**:
    - Create the **docker-compose.yml** file, defining each service and their interactions.
    - Explain build context, ports, volumes, and **depends_on**.
4. **Building and Running the Containers**:
    - Build and run the containers using **docker-compose up --build**.
    - Check the status of the containers using **docker-compose ps**.
5. **Testing the Application**:
    - Access the application through a web browser to ensure it is working as expected.
6. **Debugging Common Issues**:
    - Address potential common errors (like the COPY error) and provide guidance on how to solve them.

Conclusion

- **Summary**: This report covered the setup of a full-stack application using Docker, including the configuration of each component and their interaction.
- **Further Reading**: For deeper understanding, refer to the Docker documentation and additional resources on Docker best practices and component-specific configurations.

Appendices

- **Appendix A**: Full **docker-compose.yml** listing.
- **Appendix B**: Full **Dockerfile** listings for each service.
- **Appendix C**: Example **.env** file.

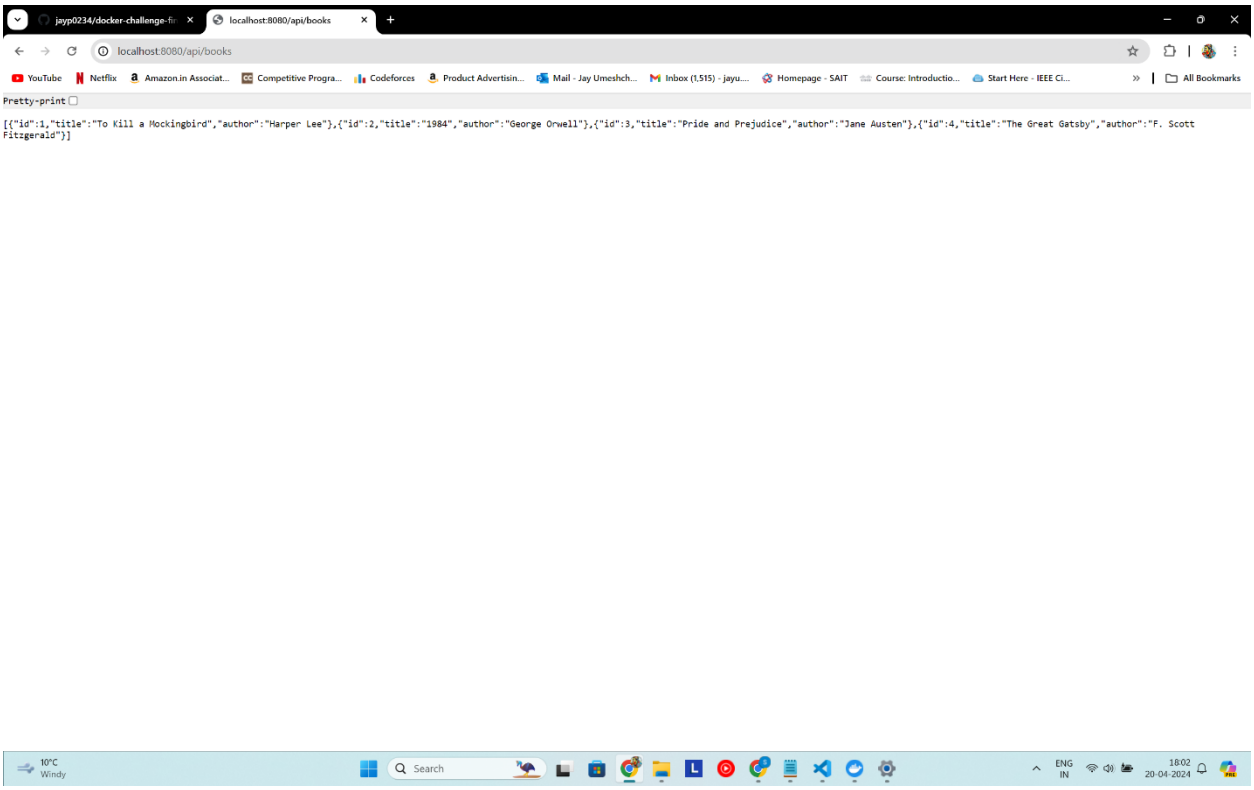
References

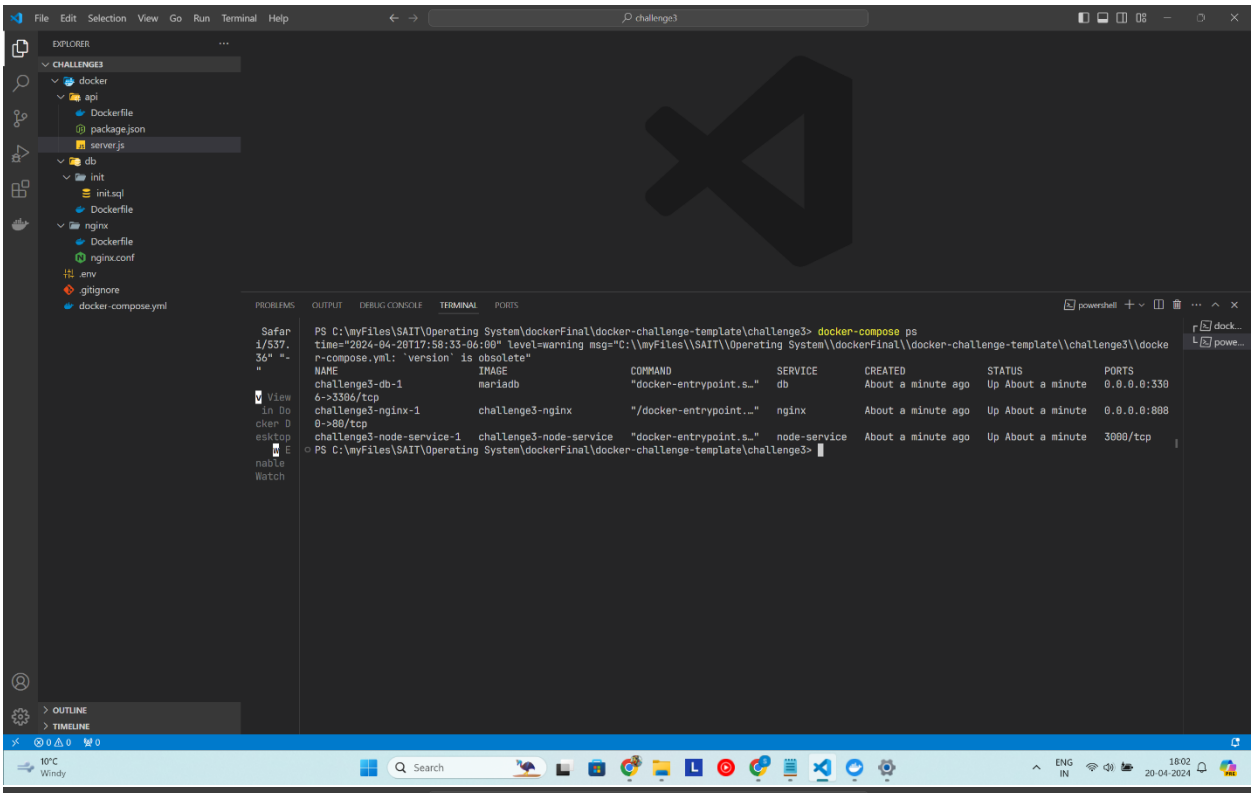
Docker, "Docker Documentation," Docker. [Online]. Available: https://docs.docker.com. [Accessed: Apr. 20, 2024].

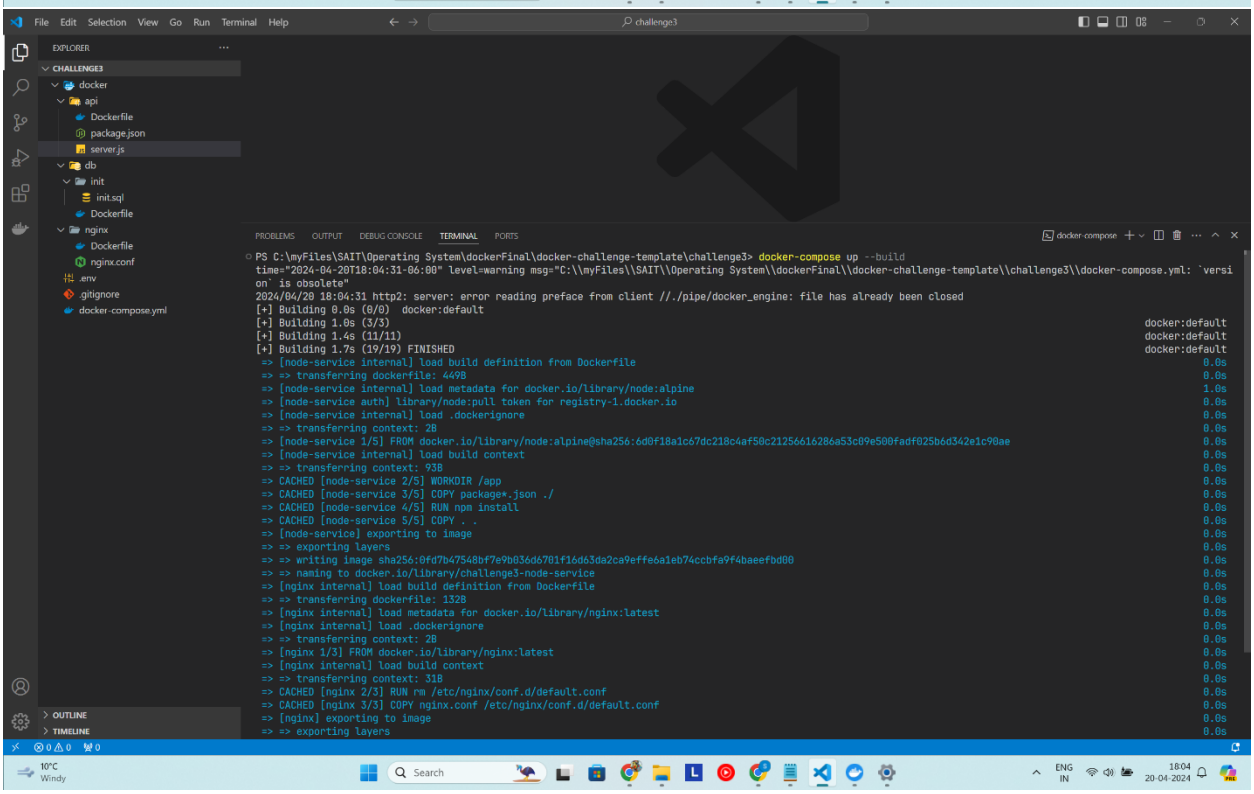
Node.js, "Node.js Docker Best Practices," Node.js. [Online]. Available: https://nodejs.org/en/docs/guides/nodejs-docker-webapp/. [Accessed: Apr. 20, 2024].

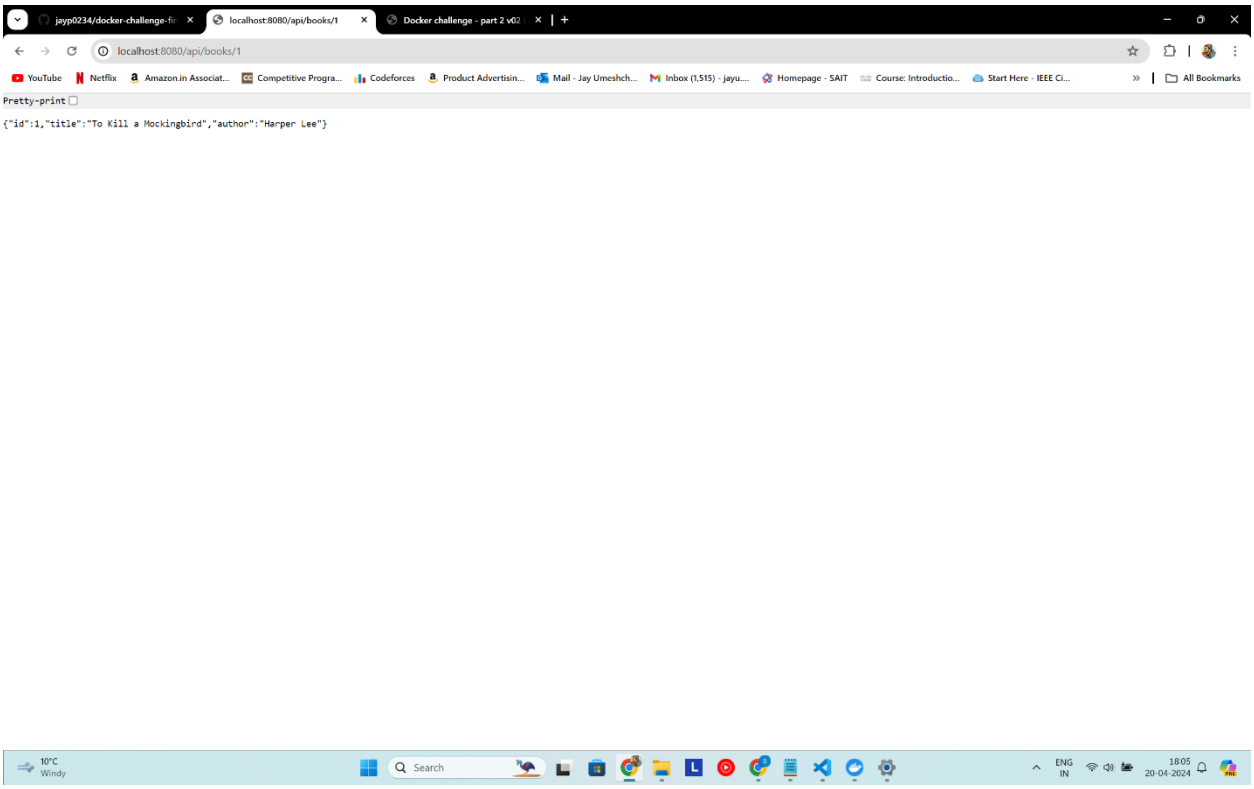
MariaDB, "MariaDB Docker Image," Docker Hub. [Online]. Available: https://hub.docker.com/_/mariadb. [Accessed: Apr. 20, 2024].

ScreenShots




File Edit Selection View Go Run Terminal Help
challenge3
EXPLORER
CHALLENGE3
docker
api
Dockerfile
package.json
server.js
db
init
init.sql
Dockerfile
nginx
Dockerfile
nginx.conf
.env
.gitignore
docker-compose.yml
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
PS C:\myFiles\SAIT\Operating System\dockerFinal\docker-challenge-template\challenge3> docker-compose ps
time="2024-04-20T17:58:33-06:00" level=warning msg="C:\\myFiles\\SAIT\\Operating System\\dockerFinal\\docker-challenge-template\\challenge3\\docker-compose.yml: `version` is obsolete"
NAME IMAGE COMMAND SERVICE CREATED STATUS PORTS
challenge3-db-1 mariadb "docker-entrypoint.s…" db About a minute ago Up About a minute 0.0.0.0:3306->3306/tcp
challenge3-nginx-1 challenge3-nginx "/docker-entrypoint.…" nginx About a minute ago Up About a minute 0.0.0.0:8080->80/tcp
challenge3-node-service-1 challenge3-node-service "docker-entrypoint.s…" node-service About a minute ago Up About a minute 3000/tcp
PS C:\myFiles\SAIT\Operating System\dockerFinal\docker-challenge-template\challenge3>
OUTLINE
TIMELINE
10°C Windy
Search
ENG IN
18:02 20-04-2024


File Edit Selection View Go Run Terminal Help
challenge3
EXPLORER
CHALLENGE3
docker
api
Dockerfile
package.json
server.js
db
init
init.sql
Dockerfile
nginx
Dockerfile
nginx.conf
.env
.gitignore
docker-compose.yml
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
docker-compose
PS C:\myFiles\SAIT\Operating System\dockerFinal\docker-challenge-template\challenge3> docker-compose up --build
time="2024-04-20T18:04:31-06:00" level=warning msg="C:\\myFiles\\SAIT\\Operating System\\dockerFinal\\docker-challenge-template\\challenge3\\docker-compose.yml: `version` is obsolete"
2024/04/20 18:04:31 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed
[+] Building 0.0s (0/0) docker:default
[+] Building 1.0s (3/3) docker:default
[+] Building 1.4s (11/11) docker:default
[+] Building 1.7s (19/19) FINISHED docker:default
=> [node-service internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 449B 0.0s
=> [node-service internal] load metadata for docker.io/library/node:alpine 1.0s
=> [node-service auth] library/node:pull token for registry-1.docker.io 0.0s
=> [node-service internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [node-service 1/5] FROM docker.io/library/node:alpine@sha256:6d0f18a1c67dc218c4af50c2125661628 6a53c09e500fadf025b6d342e1c90ae 0.0s
=> [node-service internal] load build context 0.0s
=> => transferring context: 93B 0.0s
=> CACHED [node-service 2/5] WORKDIR /app 0.0s
=> CACHED [node-service 3/5] COPY package*.json ./ 0.0s
=> CACHED [node-service 4/5] RUN npm install 0.0s
=> CACHED [node-service 5/5] COPY . . 0.0s
=> [node-service] exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:0fd7b47548bf7e9b036d6701f16d63da2ca9effe6a1eb74ccbfa9f4baeefbd00 0.0s
=> => naming to docker.io/library/challenge3-node-service 0.0s
=> [nginx internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 132B 0.0s
=> [nginx internal] load metadata for docker.io/library/nginx:latest 0.0s
=> [nginx internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [nginx 1/3] FROM docker.io/library/nginx:latest 0.0s
=> [nginx internal] load build context 0.0s
=> => transferring context: 318B 0.0s
=> CACHED [nginx 2/3] RUN rm /etc/nginx/conf.d/default.conf 0.0s
=> CACHED [nginx 3/3] COPY nginx.conf /etc/nginx/conf.d/default.conf 0.0s
=> [nginx] exporting to image 0.0s
=> => exporting layers 0.0s
OUTLINE
TIMELINE
10°C Windy
Search
ENG IN
18:04 20-04-2024


jayp0234/docker-challenge-fi
localhost:8080/api/books/1
Docker challenge - part 2 v02
localhost:8080/api/books/1
YouTube
Netflix
Amazon.in Associat...
Competitive Progra...
Codeforces
Product Advertisin...
Mail - Jay Umeshch...
Inbox (1,515) - jayu....
Homepage - SAIT
Course: Introductio...
Start Here - IEEE Ci...
All Bookmarks
Pretty-print
{"id":1,"title":"To Kill a Mockingbird","author":"Harper Lee"}
10°C
Windy
Search
ENG
IN
18:05
20-04-2024