Challenge 2 Report: Configuring a Node.js Application with Docker

Overview

The main goal of this project is to configure and run a dynamic Node.js application using Docker, highlighting the advantages of containerization for easy deployment and scalability. The application's architecture utilizes Docker containers for both the Node.js server and NGINX, which serves as a reverse proxy.

Preparation Requirements

Tools Required:

- **Docker:** Essential for creating and managing containers.
- **Docker Compose:** Utilized for defining and running multi-container Docker applications.

Installation instructions for both tools are available on their official website.

Detailed Implementation Steps

1. **Environment Setup:**
    - Begin by extracting the contents of **challenge2.zip** into the root folder of the project to set up your working environment.
2. **Dockerfile Construction:**
    - **Node.js Application:**
        - Create a **Dockerfile** within the Node.js application directory. This file should use the Node.js official image, set a working directory, copy application files, install dependencies, expose port 3000, and define the command to start the server.
3. **NGINX Configuration:**
    - Prepare an **nginx.conf** file to configure NGINX to listen on port 80 and proxy requests to the Node.js application running on port 3000.
4. **Compose File Configuration:**
    - Create a **docker-compose.yml** file that defines the Node.js and NGINX services, specifying build contexts, ports, and dependencies.
5. **Container Operations:**
    - Execute **docker-compose up --build** to build and start the containers.
    - Check the containers' status with **docker-compose ps**.
6. **Application Testing:**
    - Open a browser and navigate to **http://localhost:8080/api/books** to see if the application returns the expected JSON output.
7. **Troubleshooting:**

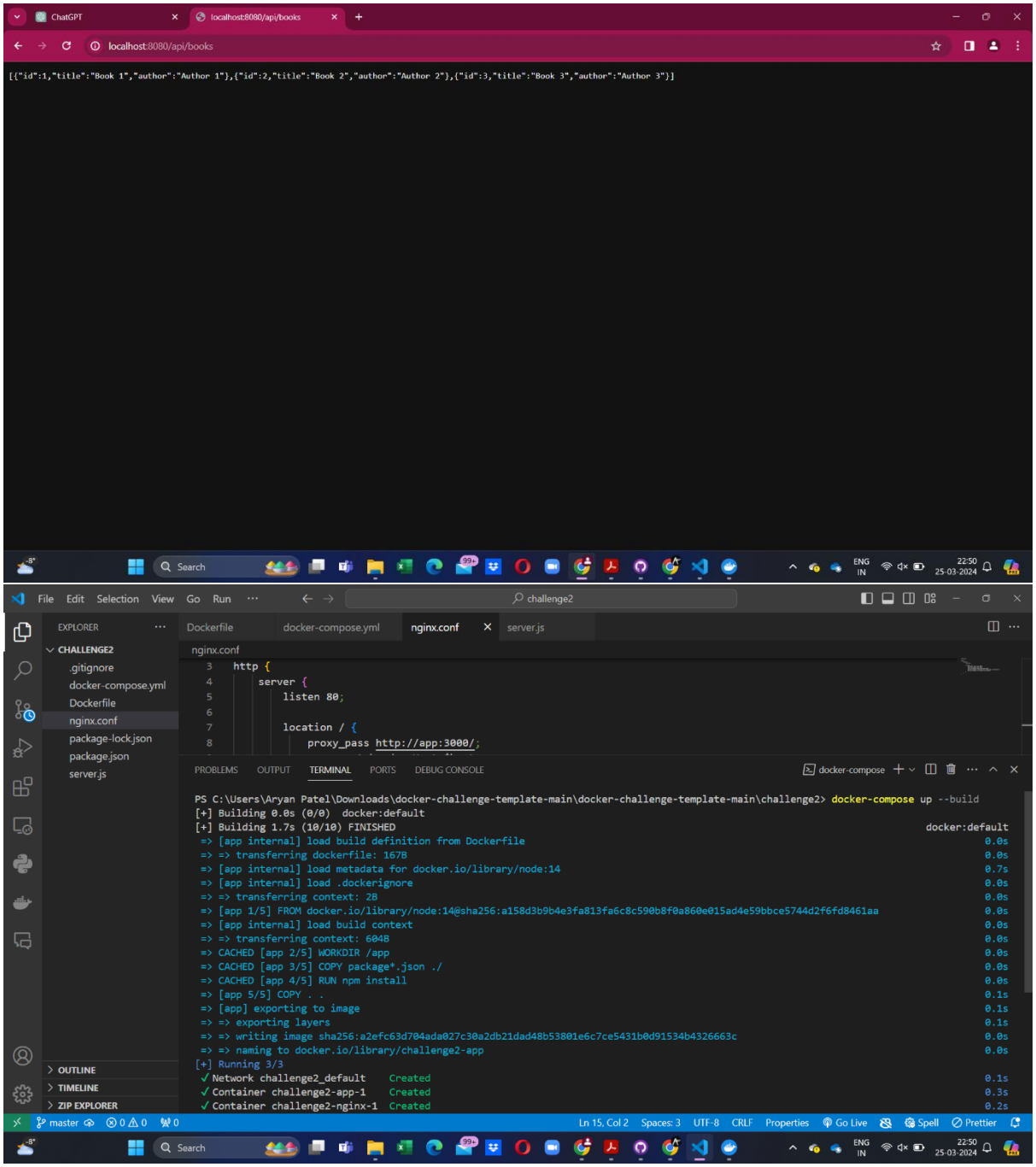
- If the expected results are not met, check the NGINX and Node.js logs for any errors and adjust the configurations as necessary.

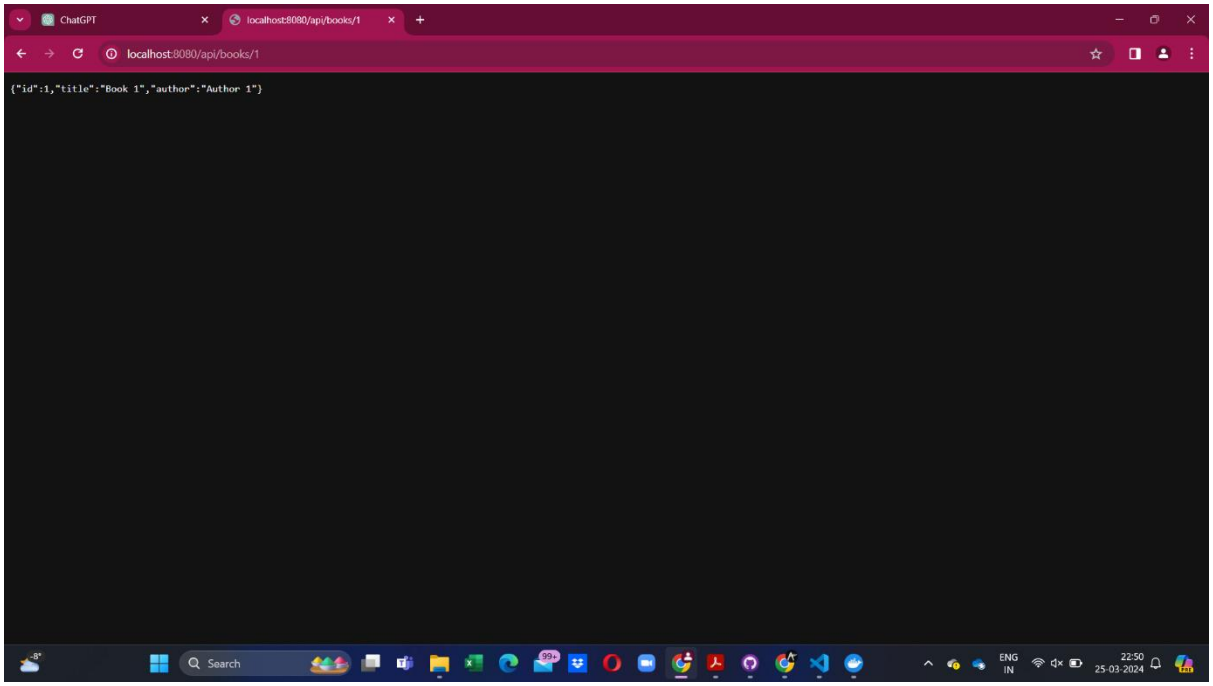
Conclusion

This report details the complete setup of a Node.js application with NGINX using Docker, illustrating each step from the initial setup to the final testing. This method ensures the application is efficiently containerized, making it easier to manage and deploy.

Cited References

- Docker Documentation, Docker Docs. [Online] Available: https://docs.docker.com [Accessed: Apr 22, 2024]


ChatGPT
localhost:8080/api/books
localhost:8080/api/books
[{"id":1,"title":"Book 1","author":"Author 1"},{"id":2,"title":"Book 2","author":"Author 2"},{"id":3,"title":"Book 3","author":"Author 3"}]
File Edit Selection View Go Run
challenge2
EXPLORER
CHALLENGE2
.gitignore
docker-compose.yml
Dockerfile
nginx.conf
package-lock.json
package.json
server.js
Dockerfile
docker-compose.yml
nginx.conf
server.js
nginx.conf
3 http {
4 server {
5 listen 80;
6
7 location / {
8 proxy_pass http://app:3000/;
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
docker-compose
PS C:\Users\Aryan Patel\Downloads\docker-challenge-template-main\docker-challenge-template-main\challenge2> docker-compose up --build
[+] Building 0.0s (0/0) docker:default
[+] Building 1.7s (10/10) FINISHED docker:default
=> [app internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 167B 0.0s
=> [app internal] load metadata for docker.io/library/node:14 0.7s
=> [app internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [app 1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa 0.0s
=> [app internal] load build context 0.0s
=> => transferring context: 604B 0.0s
=> CACHED [app 2/5] WORKDIR /app 0.0s
=> CACHED [app 3/5] COPY package*.json ./ 0.0s
=> CACHED [app 4/5] RUN npm install 0.0s
=> [app 5/5] COPY . . 0.1s
=> [app] exporting to image 0.1s
=> => exporting layers 0.1s
=> => writing image sha256:a2efc63d704ada027c30a2db21dad48b53801e6c7ce5431b0d91534b4326663c 0.0s
=> => naming to docker.io/library/challenge2-app 0.0s
[+] Running 3/3
✓ Network challenge2_default Created 0.1s
✓ Container challenge2-app-1 Created 0.3s
✓ Container challenge2-nginx-1 Created 0.2s
OUTLINE
TIMELINE
ZIP EXPLORER
master
Ln 15, Col 2 Spaces: 3 UTF-8 CRLF Properties Go Live Spell Prettier
Search
ENG IN
22:50 25-03-2024


ChatGPT
localhost:8080/api/books/1
localhost:8080/api/books/1
{"id":1,"title":"Book 1","author":"Author 1"}
Search
ENG
IN
22:50
25-03-2024