Report Challenge 2

Deploying a NodeJS Application with NGINX Reverse Proxy

Introduction:

In this challenge, we'll deploy a dynamic NodeJS application using Docker and orchestrate it with NGINX as a reverse proxy using Docker Compose. By the end, you'll be able to access your NodeJS application through NGINX on port 8080.

Prerequisites:

Basic knowledge of NodeJS and Express framework.

Docker and Docker Compose installed on your machine.

Basic understanding of NGINX configuration.

Setup and Configuration:

Step 1: Project Structure

Create a folder named challenge2 and navigate into it.

Unzip the provided challenge2.zip file in this directory, ensuring your NodeJS application files are directly within the folder.

Step 2: Writing the NodeJS Application

Within your project, ensure you have a NodeJS application ready. File "server.js" includes routes for fetching books and system stats as provided in your question.

Step 3: Creating the Dockerfile for NodeJS Application

In the root of your challenge2 directory, create a Dockerfile with the following content:

----------Dockerfile----------

```
FROM node:14

WORKDIR /app

COPY package*.json ./
```

```
RUN npm install

COPY . .

EXPOSE 3000

CMD ["node", "server.js"]
```

----------Dockerfile----------

This Dockerfile sets up a Node environment, installs dependencies, and starts your application.

Step 4: Setting Up NGINX as a Reverse Proxy

Create a nginx.conf file in your project directory with the following configuration to route requests to your NodeJS application:

----------nginx----------

```
events {}


http {
    server {
        listen 80;


        location / {
            proxy_pass http://app:3000/;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;
        }
    }
}
```

----------nginx----------

Ensure NGINX listens on port 8080 in your Docker Compose configuration, as described in the next step.

Step 5: Docker Compose Configuration

Create a docker-compose.yml file at the root of your project with the following services defined:

----------yaml----------

```
version: '3'


services:
  app:
    build: .
    ports:
      - "3000:3000"


  nginx:
    image: nginx:alpine
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf
    ports:
      - "8080:80"
    depends_on:
      - app
```

----------yaml----------

This configuration tells Docker Compose how to build your app and how to configure NGINX to proxy requests to it.

Step 6: Building and Running Your Containers

Open a terminal in your project directory.

Run docker-compose up --build to start your services.

Once the services are running, navigate to http://localhost:8080/api/books and http://localhost:8080/api/books/1 in a browser to view the output.

----------**Step 7:** Debugging and Troubleshooting on Problems encountered by me----------Extra----------

If the application doesn't work as expected, check the logs using docker-compose logs.

Ensure no other services are running on port 8080. You can stop them using docker stop <container_id> or by terminating the process occupying the port.

Conclusion:

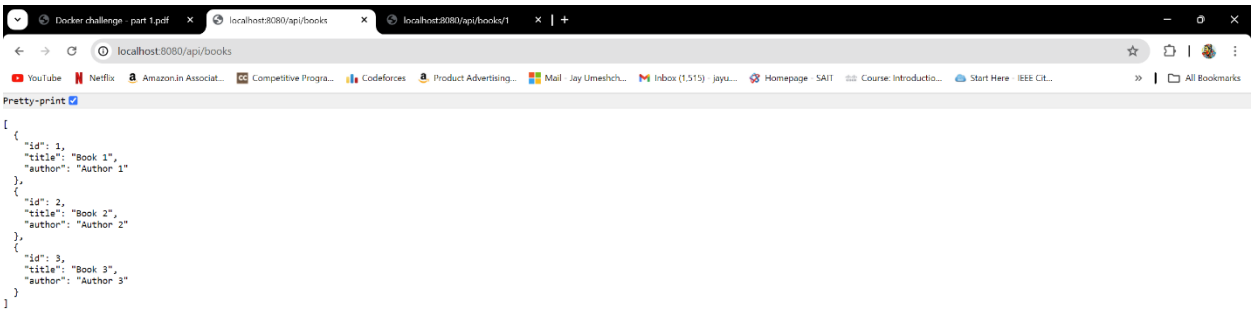
You've now set up a NodeJS application with a NGINX reverse proxy using Docker and Docker Compose. This setup is commonly used in real-world applications, providing you with a solid foundation in Docker container management and application deployment.

Reference Links:

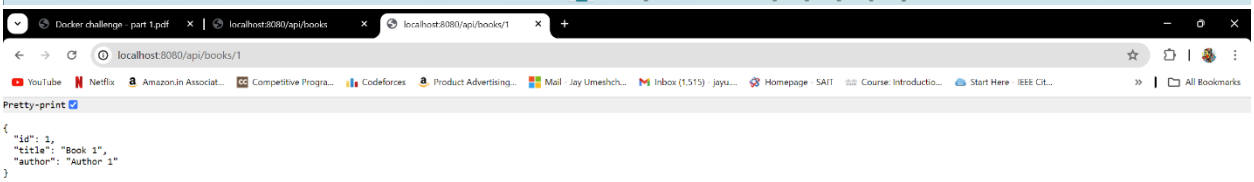
https://docs.docker.com/get-started/overview/

https://docker-curriculum.com/

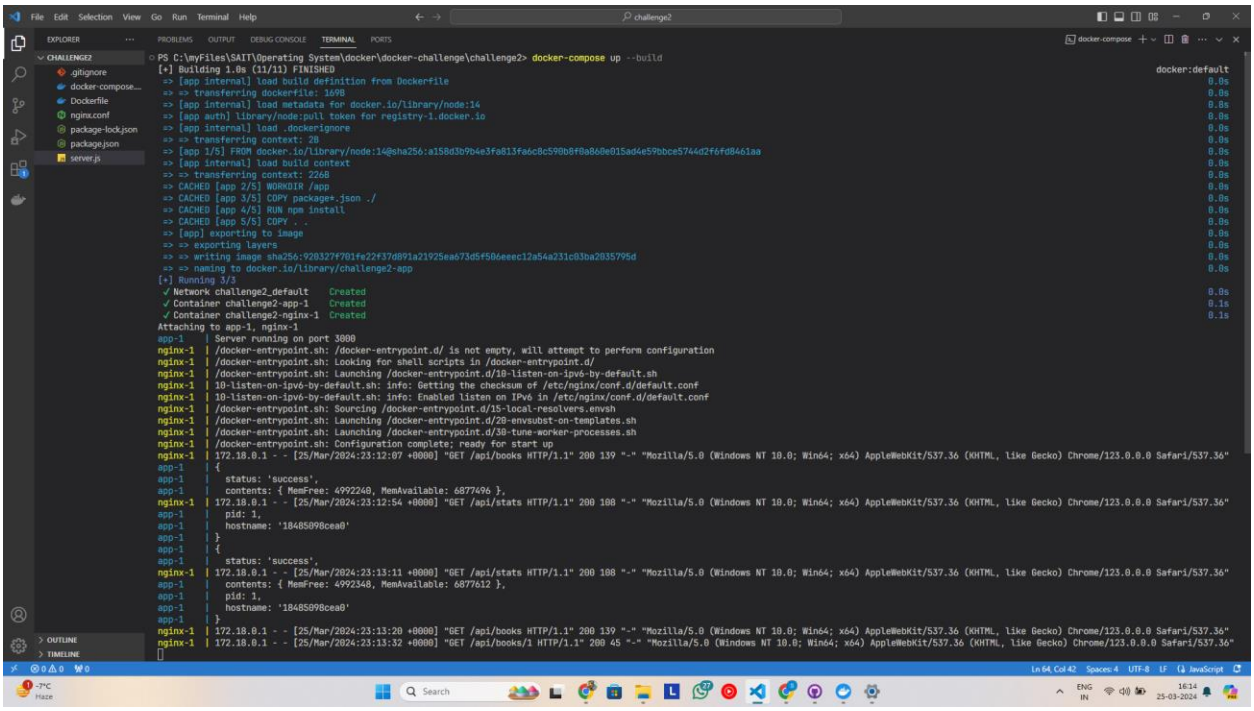
https://expressjs.com


Docker challenge - part 1.pdf
localhost:8080/api/books
localhost:8080/api/books/1
localhost:8080/api/books
YouTube
Netflix
Amazon.in Associat...
Competitive Progra...
Codeforces
Product Advertising...
Mail - Jay Umeshch...
Inbox (1,515) - jayu...
Homepage - SAIT
Course: Introductio...
Start Here - IEEE Cit...
All Bookmarks
Pretty-print
[
{
"id": 1,
"title": "Book 1",
"author": "Author 1"
},
{
"id": 2,
"title": "Book 2",
"author": "Author 2"
},
{
"id": 3,
"title": "Book 3",
"author": "Author 3"
}
]


-6°C
Haze
Search
ENG
IN
16:15
25-03-2024


Docker challenge - part 1.pdf
localhost:8080/api/books
localhost:8080/api/books/1
localhost:8080/api/books/1
YouTube
Netflix
Amazon.in Associat...
Competitive Progra...
Codeforces
Product Advertising...
Mail - Jay Umeshch...
Inbox (1,515) - jayu...
Homepage - SAIT
Course: Introductio...
Start Here - IEEE Cit...
All Bookmarks
Pretty-print
{
"id": 1,
"title": "Book 1",
"author": "Author 1"
}


-6°C
Haze
Search
ENG
IN
16:15
25-03-2024


File Edit Selection View Go Run Terminal Help
challenge2
EXPLORER
CHALLENGE2
.gitignore
docker-compose....
Dockerfile
nginx.conf
package-lock.json
package.json
server.js
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
docker-compose
PS C:\myFiles\SAIT\Operating System\docker\docker-challenge\challenge2> docker-compose up --build
[+] Building 1.0s (11/11) FINISHED docker:default
=> [app internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 169B 0.0s
=> [app internal] load metadata for docker.io/library/node:14 0.0s
=> [app auth] library/node:pull token for registry-1.docker.io 0.0s
=> [app internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [app 1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f8a860e015ad4e59bbce5744d2f6fd8461aa 0.0s
=> [app internal] load build context 0.0s
=> => transferring context: 226B 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
=> CACHED [app 5/5] COPY . . 0.0s
=> [app] exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:920327f701fe22f37d891a21925ea673d5f506eeec12a54a231c03ba2035795d 0.0s
=> => naming to docker.io/library/challenge2-app 0.0s
[+] Running 3/3
✓ Network challenge2_default Created 0.0s
✓ Container challenge2-app-1 Created 0.1s
✓ Container challenge2-nginx-1 Created 0.1s
Attaching to app-1, nginx-1
app-1 | Server running on port 3000
nginx-1 | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
nginx-1 | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
nginx-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
nginx-1 | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
nginx-1 | 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
nginx-1 | /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
nginx-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
nginx-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
nginx-1 | /docker-entrypoint.sh: Configuration complete; ready for start up
nginx-1 | 172.18.0.1 - - [25/Mar/2024:23:12:07 +0000] "GET /api/books HTTP/1.1" 200 139 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36"
app-1 | {
app-1 | status: 'success',
app-1 | contents: { MemFree: 4992240, MemAvailable: 6877496 },
nginx-1 | 172.18.0.1 - - [25/Mar/2024:23:12:54 +0000] "GET /api/stats HTTP/1.1" 200 108 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36"
app-1 | pid: 1,
app-1 | hostname: '18485098cea0'
app-1 | }
app-1 | {
app-1 | status: 'success',
nginx-1 | 172.18.0.1 - - [25/Mar/2024:23:13:11 +0000] "GET /api/stats HTTP/1.1" 200 108 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36"
app-1 | contents: { MemFree: 4992348, MemAvailable: 6877612 },
app-1 | pid: 1,
app-1 | hostname: '18485098cea0'
app-1 | }
nginx-1 | 172.18.0.1 - - [25/Mar/2024:23:13:20 +0000] "GET /api/books HTTP/1.1" 200 139 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36"
nginx-1 | 172.18.0.1 - - [25/Mar/2024:23:13:32 +0000] "GET /api/books/1 HTTP/1.1" 200 45 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36"
Ln 64, Col 42 Spaces: 4 UTF-8 LF JavaScript
-7°C Haze
Search
ENG IN
16:14 25-03-2024