

Assignment: Containerization with Docker Report

Personal

- Name: Zackaria Osman
- ID: 000885686
- Email:
 - Primary: cobalt.zr86@gmail.com
 - Secondary: zackaria.osman@edu.sait.ca

Links

GitHub repository: <https://github.com/ZackariaOsman/docker-challenge-template>

Basic introduction of Docker

Why is it important for me, in the context of software development?

- Consistency
 - Ensures applications behave consistently across different environments.
- Isolation
 - Encapsulates applications and dependencies, enhancing security and reliability.
- Efficiency
 - Lightweight containers enable faster deployment and scaling.
- Portability

- Runs on any system supporting Docker, simplifying deployment and collaboration.
- Version Control
 - Images are versioned and shareable, ensuring reproducibility and consistency.

Challenge 1 - Simple static page server

Docker Commands Used

*** '.' = current working directory ***

- `docker build`
 - Builds an image from a Dockerfile in the current directory.
- `docker images`
 - Lists all available Docker images.
- `docker build -t <repository_name>`
 - Builds an image and tags it with a repository name.
- `docker run -d -p <host_port>:<container_port> <image_id>`
 - Runs a container in detached mode, mapping ports between host and container.
- `docker ps`
 - Lists running containers.
- `docker stop <container_id> or docker stop <container_name>:`
 - Stops a running container.
- `docker ps -a`
 - Lists all containers (both running and stopped).
- `docker rm <container_id> or docker rm <container_name>`
 - Removes a container.
- `docker rmi <image_id> or docker rmi <image_name>`
 - Removes an image.

Steps To Make It Work

Installation

Install Docker

- Downloaded the Docker Engine from the Docker website followed the instructions for the installation process.

Configuration

Creating a DockerFile:

```
# To use the Nginx image from Docker Hub
FROM nginx:alpine

# Copy contents of public folder to Nginx html directory
COPY public/ /usr/share/nginx/html/

# Expose port 80 but it is not mandatory
#EXPOSE 80

# Start Nginx when container starts
#Also not mandatory
#CMD ["nginx", "-g", "daemon off;"]

#For Nginx in debug mode
#CMD ["nginx-debug", "-g", "daemon off;"]
```

Building an Image:

- Use `docker build .` to build an image from a Dockerfile located in the current directory (.).
- Optionally, you can tag the image with a repository name using
 - `docker build -t <repository_name> ..`

Listing Images

- After building, use `docker images` to list all locally available Docker images.

Creation of Files

Creating and Running Containers:

- Create a container from an image with `docker run`:
 - Use `docker run -d -p <host_port>:<container_port> <image_id>` to run a container in detached mode (`-d`) and map ports (`-p`). Replace `<host_port>` and `<container_port>` with appropriate values.
 - Example: `docker run -d -p 80:80 8b0a2c43a149` runs a container from image `8b0a2c43a149`, mapping host port 80 to container port 80.

Managing Containers:

- View running containers with
 - `docker ps`.
- Stop a container using
 - `docker stop <container_id>` OR `docker stop <container_name>`.
- List all containers (including stopped ones) with
 - `docker ps -a`.
- Remove a container with
 - `docker rm <container_id>` OR `docker rm <container_name>`.

Managing Images:

- Remove an image using
 - `docker rmi <image_id>` OR `docker rmi <image_name>`.

Screenshots

Png1:

```
C:\SENSENGIN\W204-AUG2024\OPERATINGSYSTEMS\UI\termDocker\docker-challenge-template\challenge1>type Dockerfile
# To use the Nginx image from Docker hub
FROM nginx:alpine

# Copy contents of public folder to Nginx html directory
COPY public/ /usr/share/nginx/html/

# Expose port 80 but it is not mandatory
EXPOSE 80

# Start Nginx when container starts
# Also not mandatory
#CMD ["nginx", "-g", "daemon off;"]

#or Nginx in debug mode
#CMD ["nginx-debug", "-g", "daemon off;"]

C:\SENSENGIN\W204-AUG2024\OPERATINGSYSTEMS\UI\termDocker\docker-challenge-template\challenge1>docker build .
[+] Building 4.9s (7/7) FINISHED          docker:desktop-linux
-> [internal] load build definition from Dockerfile
-> transferring Dockerfile: 423B
-> [internal] load metadata for docker.io/library/nginx:alpine
-> [internal] load .dockerignore
-> transferring context: 2B
-> [internal] load build context
-> transferring context: 389B
-> [1/2] FROM docker.io/library/nginx:alpine@sha256:a5ee5d042aa9e81e013f97ae40c3dda26f6e08f22b6251acd728e57956
-> resolve docker.io/library/nginx:alpine@sha256:a5ee5d042aa9e81e013f97ae40c3dda26f6e08f22b6251acd728e57956
-> sha256:a5ee5d042aa9e81e013f97ae40c3dda26f6e08f22b6251acd728e57956005 9.07kB / 9.07kB
-> sha256:d6d4d323ef1c45ee5c555a6e231ef50b35e0f575a81fa14779e5c 2.00kB / 2.00kB
-> sha256:099a2d78d1f36dc012419b0e4b7da299f48b4d2854fab051e70491e233 11.03kB / 11.03kB
-> sha256:b04a74c0e5f5c51996f2c2ca0c051a29a7716d1732e923f1a1c4c870a0580 3.42kB / 3.42kB
-> sha256:ee7392345c47d688a3d0a7b0301b1a0d33a1f2c44803105ac7386e09923a00a 1.92kB / 1.92kB
-> sha256:4838760d22df5943da3aad764e3df0477ebc2051724efdb110723fbcfb3 620B / 620B
-> sha256:005493210c66a270e07c6b0a321d4a9f0eb0c4223813540913b065d09a 950B / 950B
-> extracting sha256:b04a74c0e5f5c51996f2c2ca0c051a29a7716d1732e923f1a1c4c870a0580 0.15s
-> sha256:811a4232b3b6423b363d2b2b3dc028c690809c941e0b731c335399305461c 397B / 397B
-> sha256:5080260c4e02270a83affcbe50a0f52aacc3bd3a8f05706c0f204149c 1.21kB / 1.21kB
-> sha256:86c35a3a2ef3fc036c4d25dc91090bcb7a182ccde0b8dc3d8f2441cc1d114 1.40kB / 1.40kB
-> extracting sha256:ee7392345c47d688a3aad764e3df0477ebc2051724efdb110723fbcfb3 0.25s
-> sha256:011a23012072c72ae51d8a8b50dec8330732f4a1023c0b0c7e0b053c 12.04kB / 12.04kB
-> extracting sha256:4838760d22df5943da3aad764e3df0477ebc2051724efdb110723fbcfb3 0.15s
-> extracting sha256:085492633b0b6ac7f0e0720c012bdfa9f0b6d8c42381354b0f15a965da9a 0.05s
-> extracting sha256:811a4232b3b6423b363d2b2b3dc028c690809c941e0b731c335399305461c 0.05s
-> extracting sha256:5080260c4e02270a83affcbe50a0f52aacc3bd3a8f05706c0f204149c 0.05s
-> extracting sha256:86c35a3a2ef3fc036c4d25dc91090bcb7a182ccde0b8dc3d8f2441cc1d114 0.05s
-> extracting sha256:011a23012072c72ae51d8a8b50dec8330732f4a1023c0b0c7e0b053c7c0b053c 0.25s
-> [2/2] COPY public/ /usr/share/nginx/html/
-> exporting to image
-> exporting layers
-> writing image sha256:8bba2c43a140ce032e5124903d7af2f8e082717cfe5f8009f2765bba5dca 0.05s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/cihyjbhhsxf65ibhyf62dtr

[+] Building 0.8s (7/7) FINISHED          docker:desktop-linux
-> [internal] load build definition from Dockerfile
-> transferring Dockerfile: 423B
-> [internal] load metadata for docker.io/library/nginx:alpine
-> [internal] load .dockerignore
-> [internal] load build context
0s
-> [internal] load build context
0s
-> transferring context: 2B
0.85s -> transferring context: 65B
```

Png2:

```
C:\SENSENGIN\W204-AUG2024\OPERATINGSYSTEMS\UI\termDocker\docker-challenge-template\challenge1>type Dockerfile
# To use the Nginx image from Docker hub
FROM nginx:alpine

# Copy contents of public folder to Nginx html directory
COPY public/ /usr/share/nginx/html/

# Expose port 80 but it is not mandatory
EXPOSE 80

# Start Nginx when container starts
# Also not mandatory
#CMD ["nginx", "-g", "daemon off;"]

#or Nginx in debug mode
#CMD ["nginx-debug", "-g", "daemon off;"]

C:\SENSENGIN\W204-AUG2024\OPERATINGSYSTEMS\UI\termDocker\docker-challenge-template\challenge1>docker build .
[+] Building 4.9s (7/7) FINISHED          docker:desktop-linux
-> [internal] load build definition from Dockerfile
-> transferring Dockerfile: 423B
-> [internal] load metadata for docker.io/library/nginx:alpine
-> [internal] load .dockerignore
-> transferring context: 2B
-> [internal] load build context
-> transferring context: 389B
-> [1/2] FROM docker.io/library/nginx:alpine@sha256:a5ee5d042aa9e81e013f97ae40c3dda26f6e08f22b6251acd728e57956
-> resolve docker.io/library/nginx:alpine@sha256:a5ee5d042aa9e81e013f97ae40c3dda26f6e08f22b6251acd728e57956
-> sha256:a5ee5d042aa9e81e013f97ae40c3dda26f6e08f22b6251acd728e57956005 9.07kB / 9.07kB
-> sha256:d6d4d323ef1c45ee5c555a6e231ef50b35e0f575a81fa14779e5c 2.00kB / 2.00kB
-> sha256:099a2d78d1f36dc012419b0e4b7da299f48b4d2854fab051e70491e233 11.03kB / 11.03kB
-> sha256:b04a74c0e5f5c51996f2c2ca0c051a29a7716d1732e923f1a1c4c870a0580 3.42kB / 3.42kB
-> sha256:ee7392345c47d688a3d0a7b0301b1a0d33a1f2c44803105ac7386e09923a00a 1.92kB / 1.92kB
-> sha256:4838760d22df5943da3aad764e3df0477ebc2051724efdb110723fbcfb3 620B / 620B
-> sha256:005493210c66a270e07c6b0a321d4a9f0eb0c4223813540913b065d09a 950B / 950B
-> extracting sha256:b04a74c0e5f5c51996f2c2ca0c051a29a7716d1732e923f1a1c4c870a0580 0.15s
-> sha256:811a4232b3b6423b363d2b2b3dc028c690809c941e0b731c335399305461c 397B / 397B
-> sha256:5080260c4e02270a83affcbe50a0f52aacc3bd3a8f05706c0f204149c 1.21kB / 1.21kB
-> sha256:86c35a3a2ef3fc036c4d25dc91090bcb7a182ccde0b8dc3d8f2441cc1d114 1.40kB / 1.40kB
-> extracting sha256:ee7392345c47d688a3aad764e3df0477ebc2051724efdb110723fbcfb3 0.25s
-> sha256:011a23012072c72ae51d8a8b50dec8330732f4a1023c0b0c7e0b053c 12.04kB / 12.04kB
-> extracting sha256:4838760d22df5943da3aad764e3df0477ebc2051724efdb110723fbcfb3 0.15s
-> extracting sha256:085492633b0b6ac7f0e0720c012bdfa9f0b6d8c42381354b0f15a965da9a 0.05s
-> extracting sha256:811a4232b3b6423b363d2b2b3dc028c690809c941e0b731c335399305461c 0.05s
-> extracting sha256:5080260c4e02270a83affcbe50a0f52aacc3bd3a8f05706c0f204149c 0.05s
-> extracting sha256:86c35a3a2ef3fc036c4d25dc91090bcb7a182ccde0b8dc3d8f2441cc1d114 0.05s
-> extracting sha256:011a23012072c72ae51d8a8b50dec8330732f4a1023c0b0c7e0b053c7c0b053c 0.25s
-> [2/2] COPY public/ /usr/share/nginx/html/
-> exporting to image
-> exporting layers
-> writing image sha256:8bba2c43a140ce032e5124903d7af2f8e082717cfe5f8009f2765bba5dca 0.05s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/cihyjbhhsxf65ibhyf62dtr

[+] Building 0.8s (7/7) FINISHED          docker:desktop-linux
-> [internal] load build definition from Dockerfile
-> transferring Dockerfile: 423B
-> [internal] load metadata for docker.io/library/nginx:alpine
-> [internal] load .dockerignore
-> [internal] load build context
0s
-> [internal] load build context
0s
-> transferring context: 2B
0.85s -> transferring context: 65B
```

Png3:

```
C:\SENSENGIN\W204-AUG2024\OPERATINGSYSTEMS\UI\termDocker\docker-challenge-template\challenge1>type Dockerfile
# To use the Nginx image from Docker hub
FROM nginx:alpine

# Copy contents of public folder to Nginx html directory
COPY public/ /usr/share/nginx/html/

# Expose port 80 but it is not mandatory
EXPOSE 80

# Start Nginx when container starts
# Also not mandatory
#CMD ["nginx", "-g", "daemon off;"]

#or Nginx in debug mode
#CMD ["nginx-debug", "-g", "daemon off;"]

C:\SENSENGIN\W204-AUG2024\OPERATINGSYSTEMS\UI\termDocker\docker-challenge-template\challenge1>docker build .
[+] Building 4.9s (7/7) FINISHED          docker:desktop-linux
-> [internal] load build definition from Dockerfile
-> transferring Dockerfile: 423B
-> [internal] load metadata for docker.io/library/nginx:alpine
-> [internal] load .dockerignore
-> transferring context: 2B
-> [internal] load build context
-> transferring context: 389B
-> [1/2] FROM docker.io/library/nginx:alpine@sha256:a5ee5d042aa9e81e013f97ae40c3dda26f6e08f22b6251acd728e57956
-> resolve docker.io/library/nginx:alpine@sha256:a5ee5d042aa9e81e013f97ae40c3dda26f6e08f22b6251acd728e57956
-> sha256:a5ee5d042aa9e81e013f97ae40c3dda26f6e08f22b6251acd728e57956005 9.07kB / 9.07kB
-> sha256:d6d4d323ef1c45ee5c555a6e231ef50b35e0f575a81fa14779e5c 2.00kB / 2.00kB
-> sha256:099a2d78d1f36dc012419b0e4b7da299f48b4d2854fab051e70491e233 11.03kB / 11.03kB
-> sha256:b04a74c0e5f5c51996f2c2ca0c051a29a7716d1732e923f1a1c4c870a0580 3.42kB / 3.42kB
-> sha256:ee7392345c47d688a3d0a7b0301b1a0d33a1f2c44803105ac7386e09923a00a 1.92kB / 1.92kB
-> sha256:4838760d22df5943da3aad764e3df0477ebc2051724efdb110723fbcfb3 620B / 620B
-> sha256:005493210c66a270e07c6b0a321d4a9f0eb0c4223813540913b065d09a 950B / 950B
-> extracting sha256:b04a74c0e5f5c51996f2c2ca0c051a29a7716d1732e923f1a1c4c870a0580 0.15s
-> sha256:811a4232b3b6423b363d2b2b3dc028c690809c941e0b731c335399305461c 397B / 397B
-> sha256:5080260c4e02270a83affcbe50a0f52aacc3bd3a8f05706c0f204149c 1.21kB / 1.21kB
-> sha256:86c35a3a2ef3fc036c4d25dc91090bcb7a182ccde0b8dc3d8f2441cc1d114 1.40kB / 1.40kB
-> extracting sha256:ee7392345c47d688a3aad764e3df0477ebc2051724efdb110723fbcfb3 0.25s
-> sha256:011a23012072c72ae51d8a8b50dec8330732f4a1023c0b0c7e0b053c 12.04kB / 12.04kB
-> extracting sha256:4838760d22df5943da3aad764e3df0477ebc2051724efdb110723fbcfb3 0.15s
-> extracting sha256:085492633b0b6ac7f0e0720c012bdfa9f0b6d8c42381354b0f15a965da9a 0.05s
-> extracting sha256:811a4232b3b6423b363d2b2b3dc028c690809c941e0b731c335399305461c 0.05s
-> extracting sha256:5080260c4e02270a83affcbe50a0f52aacc3bd3a8f05706c0f204149c 0.05s
-> extracting sha256:86c35a3a2ef3fc036c4d25dc91090bcb7a182ccde0b8dc3d8f2441cc1d114 0.05s
-> extracting sha256:011a23012072c72ae51d8a8b50dec8330732f4a1023c0b0c7e0b053c7c0b053c 0.25s
-> [2/2] COPY public/ /usr/share/nginx/html/
-> exporting to image
-> exporting layers
-> writing image sha256:8bba2c43a140ce032e5124903d7af2f8e082717cfe5f8009f2765bba5dca 0.05s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/cihyjbhhsxf65ibhyf62dtr

[+] Building 0.8s (7/7) FINISHED          docker:desktop-linux
-> [internal] load build definition from Dockerfile
-> transferring Dockerfile: 423B
-> [internal] load metadata for docker.io/library/nginx:alpine
-> [internal] load .dockerignore
-> [internal] load build context
0s
-> [internal] load build context
0s
-> transferring context: 2B
0.85s -> transferring context: 65B
```

Result:

Zackaria Osman
ID:000885686

Lessons learned

No, nothing went wrong, but this is what I learned:

- Docker Installation and Set up
- Creating Dockerfiles
- Building Images
- Managing Containers and Images

References

- YouTube:
 - <https://www.youtube.com/watch?v=SnSH8Ht3MIc&list=LL&index=7&t=929s>
- LinkedIn
 - <https://www.linkedin.com/pulse/why-every-software-engineer-should-learn-docker-s-gouse-basha-2amxe/>

Challenge 2 - NodeJS application

Docker Commands Used

*** '.' = current working directory ***

```
#to build the image
#docker build .

#to list the images
#docker images

#to add a tag to the image
#docker-compose build

#to create a container
#docker-compose up

#See the running containers
#docker-compose ps
```

Steps To Make It Work

Configuration

Creating a DockerFile:

```
# To use the Nginx image from Docker Hub
FROM nginx:alpine

# Use the Node.js image from Docker Hub
FROM node:14.17.0-alpine3.13

WORKDIR /app

# Copy application dependency manifests to the container image.
COPY package*.json ./

# Install application dependencies.
RUN npm install

# Bundle app source
```



```
COPY . .

# Expose the port the app runs on
EXPOSE 3000

# Start the Node.js app
CMD ["npm", "start"]
```

Creating a Docker-Compose:

```
version: '3'

services:
  web:
    build:
      context: .
      dockerfile: Dockerfile
    ports:
      - "8080:3000"
```

Steps to Build and Run

1. **Navigate to the challenge2 Directory and open CMD**
2. **Build Docker Images:**
 - docker-compose build
3. **Run Docker Containers**
 - docker-compose up
4. **Check Running Containers:**
 - docker-compose ps
5. **Open your web browser and navigate to:**
 - <http://localhost:8080/api/books> to get a JSON message with all books.
 - <http://localhost:8080/api/books/1> to get a JSON message with the book of ID 1.

Screenshots

Png1:

```
C:\Windows\System32\cmd.exe - docker-compose up
Microsoft Windows [Version 10.0.19045.4529]
(c) Microsoft Corporation. All rights reserved.

C:\SESPRINGMAY2024-AUG2024\OPERATINGSYSTEMS\DockerChallenges\docker-challenge-template\challenge2>docker-compose build
'docker-compose' is not recognized as an internal or external command,
operable program or batch file.

C:\SESPRINGMAY2024-AUG2024\OPERATINGSYSTEMS\DockerChallenges\docker-challenge-template\challenge2>docker-compose build
'docker-compose' is not recognized as an internal or external command,
operable program or batch file.

C:\SESPRINGMAY2024-AUG2024\OPERATINGSYSTEMS\DockerChallenges\docker-challenge-template\challenge2>docker-compose build .
'docker-compose' is not recognized as an internal or external command,
operable program or batch file.

C:\SESPRINGMAY2024-AUG2024\OPERATINGSYSTEMS\DockerChallenges\docker-challenge-template\challenge2>docker-compose build .
time="2024-07-02T00:11:05-07:00" level=warning msg="C:\\SESPRINGMAY2024-AUG2024\\OPERATINGSYSTEMS\\DockerChallenges\\docker-challenge-template\\challenge2\\docker-compose.yml: 'version' is obsolete"
no such service".

C:\SESPRINGMAY2024-AUG2024\OPERATINGSYSTEMS\DockerChallenges\docker-challenge-template\challenge2>docker-compose build
time="2024-07-02T00:11:09-07:00" level=warning msg="C:\\SESPRINGMAY2024-AUG2024\\OPERATINGSYSTEMS\\DockerChallenges\\docker-challenge-template\\challenge2\\docker-compose.yml: 'version' is obsolete"

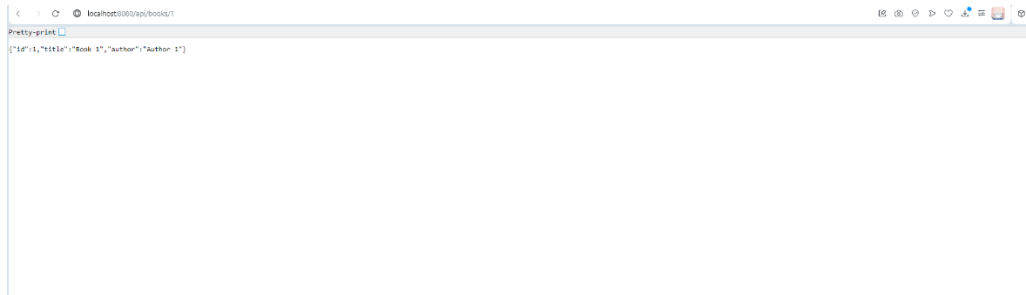
[*] Building 1.3s (18/18) FINISHED
-> [web internal] load build definition from Dockerfile                                docker:desktop-linux
-> -- transferring dockerfile: 480B                                                  0.0s
-> [web internal] load metadata for docker.io/library/node:14.17.0-alpine3.13       1.2s
-> [web internal] load dockerignore                                                 0.0s
-> -- transferring context: 2B                                                       0.0s
-> [web 1/5] FROM docker.io/library/node:14.17.0-alpine3.13@sha256:782e89198bf16cc6b1fe928d8d163d4dbecf5c8545 0.0s
-> [web internal] load build context                                                0.0s
-> -- transferring context: 1.00KB                                                  0.0s
-> CACHED [web 2/5] WORKDIR /app                                                    0.0s
-> CACHED [web 3/5] COPY package*.json ./                                           0.0s
-> CACHED [web 4/5] RUN npm install                                                  0.0s
-> [web 5/5] COPY . .                                                                0.0s
-> [web] exporting to image                                                         0.0s
-> -- exporting layers                                                              0.0s
-> -- writing image sha256:b421d70a28a5189ab1bf0d2a5c8cabdb0ee71491228d376ba3facb3aab36f3 0.0s
-> -- naming to docker.io/library/challenge2-web                                    0.0s

[*] Running 2/02024-AUG2024\OPERATINGSYSTEMS\DockerChallenges\docker-challenge-template\challenge2>docker-compose up
[*] Network challenge2_default Created                                              0.0s
[*] Container challenge2-web-1 Created                                              0.0s

Attaching to web-1
web-1 | > srv@1.0.0 start /app
web-1 | > node server.js
web-1 |
web-1 | Server running on port 3000

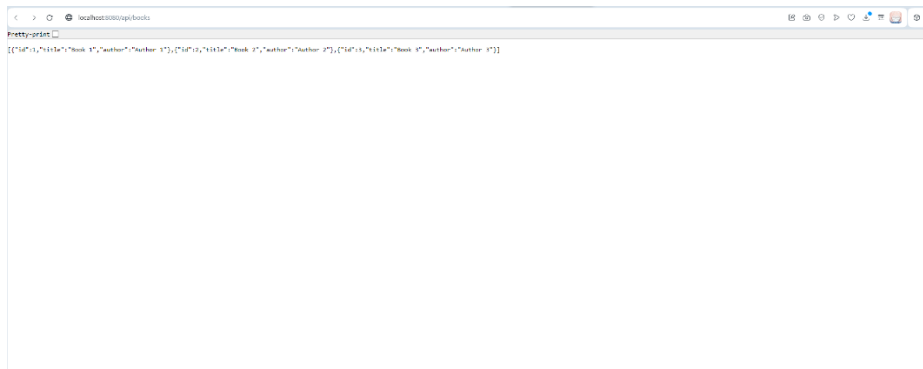
View in Docker Desktop  Enable Watch
```

Result1:



A screenshot of a web browser window with the address bar showing 'localhost:3000/api/books/1'. The page content displays a JSON object in a 'Pretty-print' view. The JSON object is: `{ "id": 1, "title": "Book 1", "author": "Author 1" }`.

Result2:



A screenshot of a web browser window with the address bar showing 'localhost:3000/api/books'. The page content displays an array of three JSON objects in a 'Pretty-print' view. The array is: `[{"id": 1, "title": "Book 1", "author": "Author 1"}, {"id": 2, "title": "Book 2", "author": "Author 2"}, {"id": 3, "title": "Book 3", "author": "Author 3"}]`.

Lessons learned

Again No, nothing went wrong, but this is what I learned:

- Docker Compose
- YouTube:
 - <https://www.youtube.com/watch?v=0B2raYYH2fE>
 - <https://www.youtube.com/watch?v=hXhI2ZLDgQM>