Docker Journal

Zainul Malik

Introduction

This is a tutorial and outline of the steps I followed to complete this bonus assignment. It will take you through all the steps that you need to complete challenge 1 and challenge 2 outlined in the assignment document. Along with this, I will provide helpful references, screenshots, and general tips that you can use to learn how to use Docker to deploy both static and dynamic web applications.

Prerequisites

Before you start, ensure you have the following installed on your computer:

- **Git**: For version control and cloning repositories.
- **Docker**: The main tool for creating, deploying, and running containers.
- **GitHub Account**: For hosting your project code and Dockerfiles.
- **Visual Studio Code:** A versatile code editor by Microsoft for Windows, macOS, and Linux.

Download Links

These are the download links for the above programs:

- **Git**: https://git-scm.com/downloads
- **Docker**: https://www.docker.com/products/docker-desktop/
- **GitHub Account**: https://github.com/
- **Visual Studio Code:** https://code.visualstudio.com/

Helpful Links

These are some helpful links I used to complete the challenges, and may be helpful for you as well.

- **Docker Docs**: https://docs.docker.com/get-started/
- **Creating a Dockerfile & Docker Image for HTML Application:** https://youtu.be/UXAoZg1W3Q4?si=oriYwvfmHE6OVC5l
- **NGINX + Docker:** https://youtu.be/prn68HXjlwQ?si=ql9ceDd3x-Ca8CP3
- **Understanding Docker Networking:** https://www.docker.com/blog/understanding-docker-networking-drivers-use-cases/

Getting Started

1. **Fork the Project Repository**: Visit https://github.com/eduluz1976/docker-challenge-template and fork the repository to your GitHub account. Keep the forked repository public.
2. **Clone the Repository**: Open your terminal and clone the forked repository to your local machine using:
    a. git clone <repository-url>
    b. Replace <repository-url> with the URL of your forked repository.

Challenge 1: Deploy a Static Web Page

Setup Your Project Directory

1. Inside docker-challenge-template, create a directory for the first challenge using the command below if not already created in the template: mkdir challenge1.

```
C:\Users\zainu\Desktop\OS_BONUS_ASSIGNMENT\docker-challenge-template>mkdir challenge1
```

2. Navigate into this directory: cd challenge1.

```
C:\Users\zainu\Desktop\OS_BONUS_ASSIGNMENT\docker-challenge-template>cd challenge1
```

3. Inside challenge1, create a public directory: mkdir public.

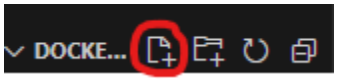
```
C:\Users\zainu\Desktop\OS_BONUS_ASSIGNMENT\docker-challenge-template\challenge1>mkdir public
```

Create Your Static Web Page

1. Navigate into the public directory: cd public.

```
C:\Users\zainu\Desktop\OS_BONUS_ASSIGNMENT\docker-challenge-template\challenge1\public>
```

2. Open the challenge1 folder in Visual Studio Code and create an index.html file by clicking the new file button:



3. Add the following code to your index.html file and replace [Your Name] and [Your Student ID] with your actual name and ID.

Dockerize Your Application

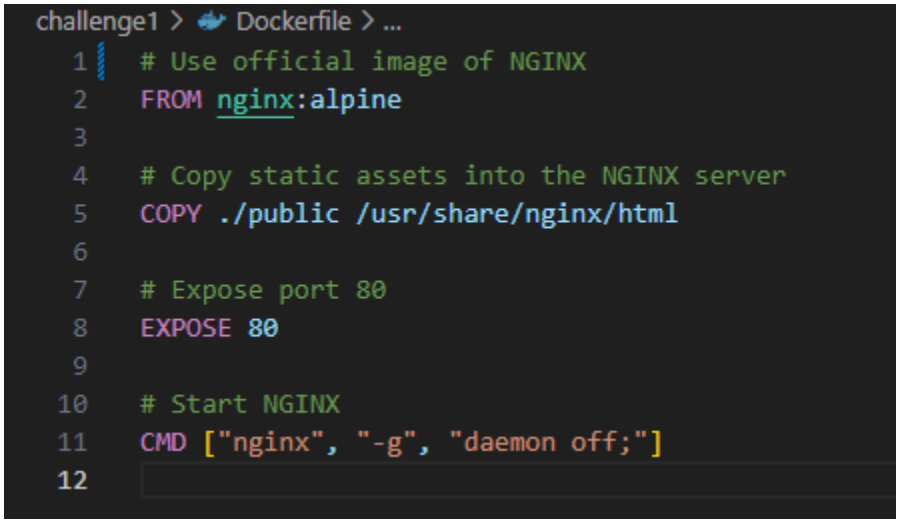
1. Return to the challenge1 directory: cd ..

```
C:\Users\zainu\Desktop\OS_BONUS_ASSIGNMENT\docker-challenge-template\challenge1\public>cd ..
```

2. Create a Dockerfile: touch Dockerfile.

```
C:\Users\zainu\Desktop\OS_BONUS_ASSIGNMENT\docker-challenge-template\challenge1>touch Dockerfile
```

3. Open the Dockerfile in Visual Studio Code and insert the following:



4. Save and close the Dockerfile.

Build and Run Your Docker Container

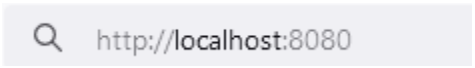
1. In the challenge1 directory, build your Docker image: docker build -t webserver-static .

```
C:\Users\zainu\Desktop\OS_BONUS_ASSIGNMENT\docker-challenge-template\challenge1>docker build -t webserver-static .
```

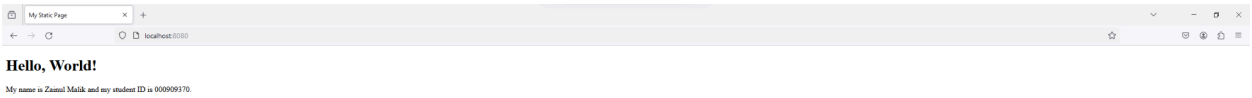
2. Run your container: docker run -d -p 8080:80 webserver-static.

```
C:\Users\zainu\Desktop\OS_BONUS_ASSIGNMENT\docker-challenge-template\challenge1>docker run -d -p 8080:80 webserver-static
```

3. Open a web browser and navigate to http://localhost:8080. You should see your static page.



4. Your page should look similar to the page below with your name and your id.



Documentation

1. Take a screenshot of your browser displaying the static page.
2. Take a screenshot of your terminal showing the last build and run commands.
3. Using a text editor such as Word or Google Docs, write down the steps you went through to achieve your results.

Challenge 2: Deploy a Dynamic NodeJS Application

Prepare Your Application Files

1. Navigate back to the root directory: cd ..

```
C:\Users\zainu\Desktop\OS_BONUS_ASSIGNMENT\docker-challenge-template\challenge1>cd ..
```

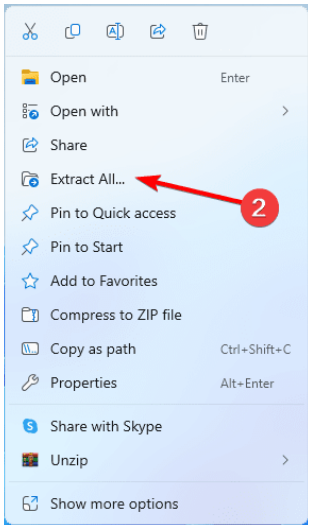
2. Create a directory for the second challenge: mkdir challenge2 (if it is not already created from the template).

```
C:\Users\zainu\Desktop\OS_BONUS_ASSIGNMENT\docker-challenge-template>mkdir challenge2
```

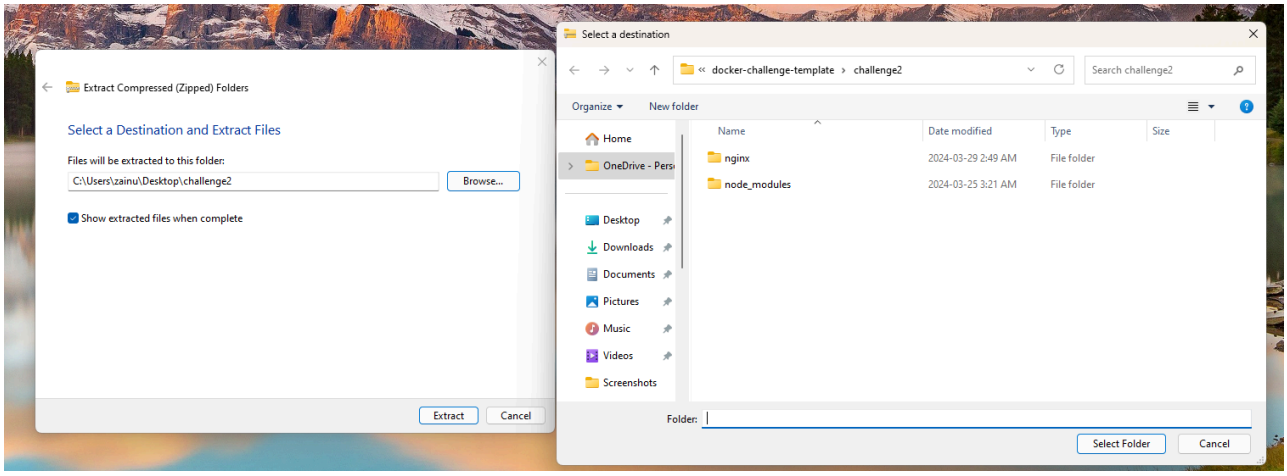
3. Navigate into it: cd challenge2

```
C:\Users\zainu\Desktop\OS_BONUS_ASSIGNMENT\docker-challenge-template>cd challenge2
```

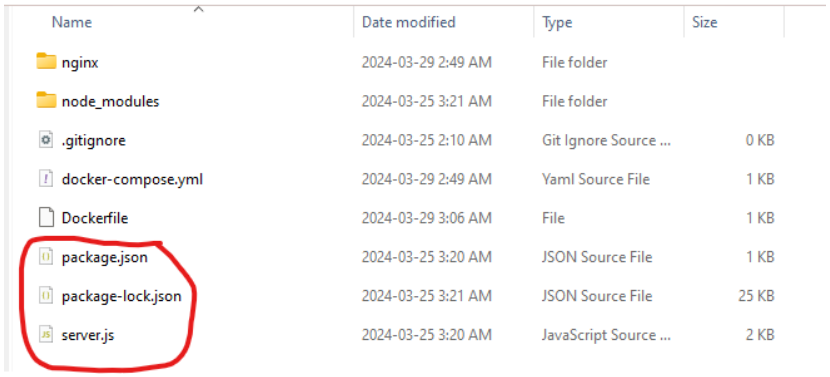
4. Unzip the provided challenge2.zip file into this directory by clicking extract all.



5. After clicking extract all, click the browse button and select the challenge2 folder. Then click the extract button to extract all the files from the challenge2.zip folder to your docker-challenge-template/challenge2 folder.



6. Now the following files should be in your docker-challenge-template/challenge2 folder directory.

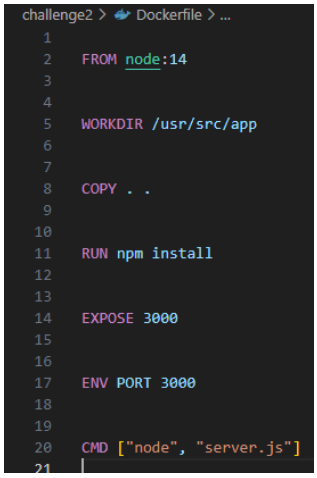


Dockerize the NodeJS Application

1. Inside challenge2, create a Dockerfile: touch Dockerfile.

```
C:\Users\zainu\Desktop\OS_BONUS_ASSIGNMENT\docker-challenge-template\challenge2>touch Dockerfile
```

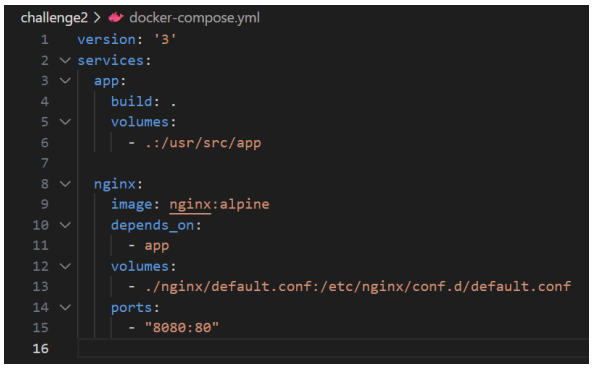
2. Open the Dockerfile in Visual Studio Code and write the following code:



3. Save and close the Dockerfile.
4. In the challenge2 directory, create a docker-compose.yml file: touch docker-compose.yml.

```
C:\Users\zainu\Desktop\OS_BONUS_ASSIGNMENT\docker-challenge-template\challenge2>touch docker-compose.yml
```

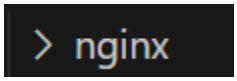
5. Open docker-compose.yml in Visual Studio Code and insert the following: (Make sure NGinx is listening on port 8080)



6. Save and close the docker-compose.yml.

7. Open visual studio code, and in the challenge2 folder click the new folder button:

8. Name it "nginx"



9. Create a default.conf file inside the nginx folder, using the new file button shown earlier in challenge1 and insert the following code.

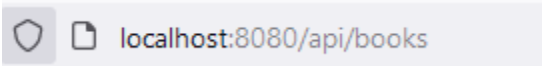
```
challenge2 > nginx > default.conf
server {
  listen 80;

  location /api/ {
    proxy_pass http://app:3000/;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
  }
}
```

Build and Run Your Application

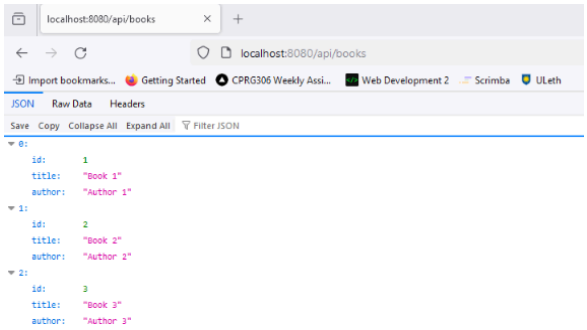
1. Build and start your application with Docker Compose: docker-compose up --build in the challenge2 directory.

```
C:\Users\zainu\Desktop\OS_BONUS_ASSIGNMENT\docker-challenge-template\challenge2>docker-compose up --build
```

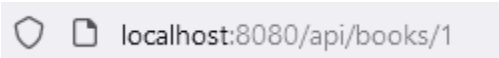
2. Open a web browser and navigate to http://localhost:8080/api/books first



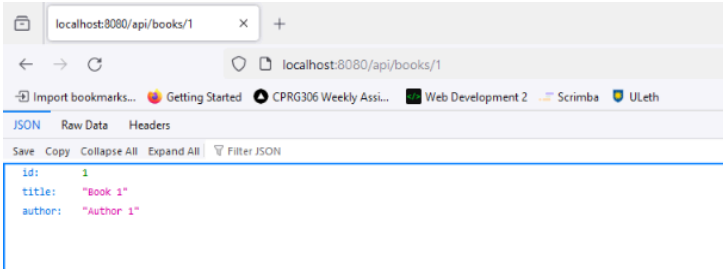
3. This is what your page should look like:



4. Now open a web browser and navigate to http://localhost:8080/api/books/1.



5. You should see this output on your page:



Documentation

1. Take a screenshot of your browser displaying the dynamic application.
2. Take screenshots of your terminal showing the Docker Compose commands and output.
3. Using a text editor such as Word or Google Docs, write down the steps you went through to achieve your results.