**Docker Challenges Report**

**Khrisell Marie Porte**

**000899460**

Introduction

This report documents the steps taken to complete two Docker challenges for the Operating Systems and Cloud Computing course. These challenges introduce containerization with Docker, which is vital for modern software development, especially in DevOps, architecture, and cloud computing.

Challenge 1: Simple Static Page Server

Step-by-Step Tutorial

1. Cloning the Repository

- Cloned the repository using:

**```**

**git clone https://github.com/eduluz1976/docker-challenge-template**

**```**

2. Creating and Configuring the Dockerfile

- Created a `public` folder and added `index.html` with name and SAIT ID:

**```html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Docker Challenge 1</title>**

**</head>**

**<body>**

**<h1>Khrisell Marie Porte </h1>**

**<h3> SAIT ID : 000899460 </h3>**

**</body>**

**</html>**

**```**

- Created a `Dockerfile`:

**```Dockerfile**

**FROM nginx:alpine**

**COPY public /usr/share/nginx/html**

**EXPOSE 80**

**CMD ["nginx", "-g", "daemon off;"]**

**```**

3. Building and Running the Docker Container

- Built the Docker image:

**```**

**docker build -t static-page-server .**

**```**

- Ran the Docker container:

**```**

**docker run -d -p 8080:80 static-page-server**

**```**

4. Testing the Application

- Opened `http://localhost:8080` in a browser to see the static page.

Screenshots

A screenshot of a computer

Description automatically generated

Challenge 2: NodeJS Application

1. Cloning the Repository

- Navigated to the `challenge2` directory and extracted `challenge2.zip`.

2. Creating and Configuring the Dockerfile

- Created a `Dockerfile` for the NodeJS application:

**```Dockerfile**

**FROM node:14**

**WORKDIR /usr/src/app**

**COPY package\*.json ./**

**RUN npm install**

**COPY . .**

**EXPOSE 3000**

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

**```**

3. Creating the Docker Compose File

- Created a `docker-compose.yml`:

**```yaml**

**version: '3'**

**services:**

**web:**

**image: nginx:alpine**

**ports:**

**- "8080:80"**

**volumes:**

**- ./nginx.conf:/etc/nginx/nginx.conf**

**depends\_on:**

**- app**

**app:**

**build: .**

**ports:**

**- "3000:3000"**

**environment:**

**NODE\_ENV: production**

**```**

**- Created `nginx.conf`:**

**```nginx**

**events {}**

**http {**

**server {**

**listen 80;**

**location /api/ {**

**proxy\_pass http://app:3000/;**

**}**

**}**

**}**

**```**

4. Building and Running the Docker Containers

- Built and ran the containers:

**``**

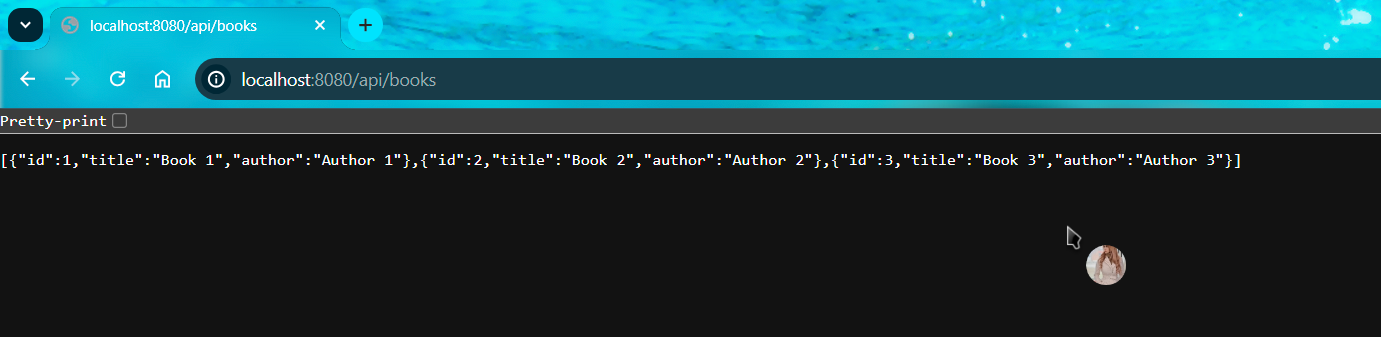
**docker-compose up --build**

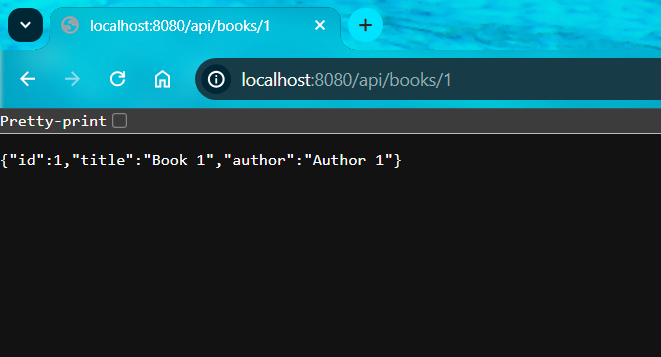
**```**

5. Testing the Application

- Opened `http://localhost:8080/api/books` and `http://localhost:8080/api/books/1` in a browser to see the JSON responses.

Screenshots





Conclusion

This report outlines the completion of two Docker challenges, providing practical experience with containerization and reinforcing concepts from the Operating Systems and Cloud Computing course.