

Assignment 6: Deploying with Docker

Project Description

For this assignment, I containerized my React application using Docker. First, I created a Dockerfile that builds and serves the app. I then created a Docker image and pushed it to my DockerHub repository. Finally, I pulled this image from an AWS EC2 instance and ran it there, exposing it on a specific port to make it accessible via the browser.

Project Description

Step 1: Install Docker on Ubuntu

Step 2: Create a Dockerfile for the React Application

Step 3: Build the Docker Image

Step 4: Push the Docker Image to Docker Hub

Step 5: Deploy the Docker Image on an AWS EC2 Instance

Step 6: Verify the Application

Optional: Stop the Container

Step 1: Install Docker on Ubuntu

1. Update packages and install dependencies:

```
sudo apt-get update
sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg \
  lsb-release
```

2. Add Docker's official GPG key:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

3. Set up the stable Docker repository:

```
$ echo \
    "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
    https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

4. Install Docker Engine:

```
sudo apt-cache madison docker-ce
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

5. Verify Docker installation:

```
sudo docker run hello-world
```

For further details, visit the [Docker installation guide](https://docs.docker.com/engine/install/ubuntu/).(https://docs.docker.com/engine/install/ubuntu/).

Step 2: Create a Dockerfile for the React Application

Inside the project repository, create a Dockerfile to containerize the React application. Here is the content of the Dockerfile:

```
FROM node:18

# Set the working directory
WORKDIR /usr/src/app
```

```
# Copy package.json and package-lock.json first to leverage Docker caching
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy all the files to the container
COPY . .

# Build the app
RUN npm run build

# Install serve to serve the app
RUN npm install -g serve

# Expose port 3000
EXPOSE 3000

# Run the app using serve
CMD ["serve", "-s", "build"]
```

Step 3: Build the Docker Image

To build the Docker image from the Dockerfile, use the following command:

```
docker push eduardobautistamacieli/react-gameapp
```

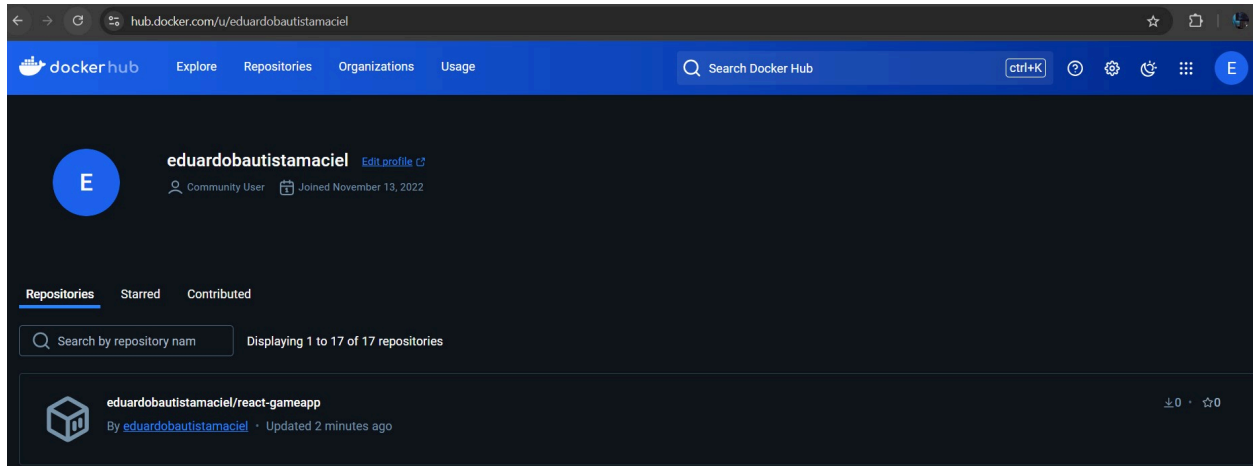
This creates a Docker image of the React application that I can push to Docker Hub.

Step 4: Push the Docker Image to Docker Hub

After building the image, I will push it to my Docker Hub account:

```
sudo docker login -u eduardobautistamaciell  
sudo docker push eduardobautistamaciell/react-gameapp
```

Below is a screenshot of the image on my Docker Hub account.



Step 5: Deploy the Docker Image on an AWS EC2 Instance

Now that the Docker image is on Docker Hub, I will deploy it to an AWS EC2 instance. The following steps will help in setting up the EC2 instance:

1. Prepare your AWS key

```
mkdir ~/KEY_FOR_AWS  
cp /mnt/hgfs/key_pairs/edu.pem ~/KEY_FOR_AWS/  
chmod 400 ~/KEY_FOR_AWS/edu.pem
```

2. SSH into your EC2 instance

```
ssh -i ~/KEY_FOR_AWS/edu.pem ubuntu@<EC2_PUBLIC_IP>
```

3. Run the Docker container

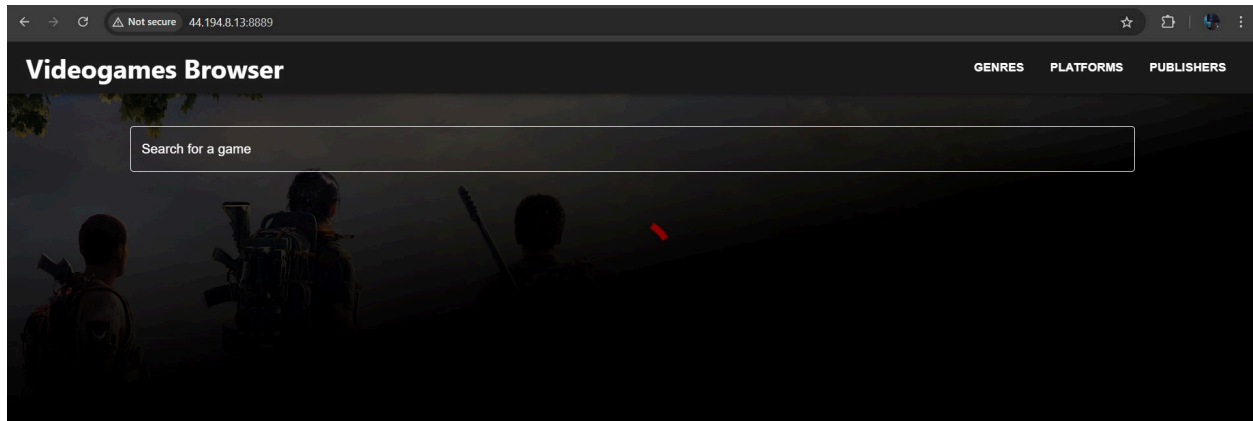
On the EC2 instance, I will run the Docker image with port mapping to make the application accessible via the public IP address.

```
sudo docker run -p 8889:3000 eduardobautistamaciell/react-gameapp
```

```
ubuntu@ip-172-31-83-202:~$ sudo docker run -p 8889:3000 eduardobautistamaciell/react-gameapp
Unable to find image 'eduardobautistamaciell/react-gameapp:latest' locally
latest: Pulling from eduardobautistamaciell/react-gameapp
7d98d813d54f: Pull complete
da802df85c96: Pull complete
7aad5092c3b: Pull complete
ad1c7cfc347f: Pull complete
59ee42d02ee5: Pull complete
eb173c1dbe92: Pull complete
f110c757afc5: Pull complete
e3d8693bad2f: Pull complete
95e5469fb9b0: Pull complete
0fc5cbb5a270: Pull complete
86be7ad8bd24: Pull complete
8a667c0100cc: Pull complete
fd075f9632d6: Pull complete
a1df3cd91b4c: Pull complete
Digest: sha256:192575fd119807bff3794698a1a0025a21d757ad7393542071807676c1cfc1e8
Status: Downloaded newer image for eduardobautistamaciell/react-gameapp:latest
INFO Accepting connections at http://localhost:3000
HTTP 10/24/2024 6:39:17 PM 179.24.47.152 GET /
HTTP 10/24/2024 6:39:17 PM 179.24.47.152 Returned 200 in 101 ms
HTTP 10/24/2024 6:39:18 PM 179.24.47.152 GET /static/js/main.cab8bf94.js
HTTP 10/24/2024 6:39:18 PM 179.24.47.152 Returned 200 in 5 ms
HTTP 10/24/2024 6:39:18 PM 179.24.47.152 GET /static/css/main.a620d49d.css
HTTP 10/24/2024 6:39:18 PM 179.24.47.152 Returned 200 in 3 ms
HTTP 10/24/2024 6:39:18 PM 179.24.47.152 GET /static/media/background-image.0d99efb296d85a4832e1.jpg
HTTP 10/24/2024 6:39:18 PM 179.24.47.152 Returned 200 in 4 ms
HTTP 10/24/2024 6:39:19 PM 179.24.47.152 GET /gamepad.ico
HTTP 10/24/2024 6:39:19 PM 179.24.47.152 Returned 200 in 2 ms
```

Step 6: Verify the Application

React app will be accessible via http://<EC2_PUBLIC_IP>:8889 in a browser.



Optional: Stop the Container

```
sudo docker ps # to get the container ID  
sudo docker stop eduardobautistamaciell/react-gameapp`
```