Working with Docker Images

Author: Oluwaseun Osunsola
Environment & Tools: AWS, EC2 Ubuntu, Windows, and, VSCode
Project Link: https://github.com/Oluwaseunoa/DevOps-Projects/tree/main/Docker-Projects

Project Overview

This project demonstrates **hands-on work with Docker images**, starting from discovering images on Docker Hub, pulling official images, creating a **custom Docker image using a Dockerfile**, running containers, exposing applications via AWS EC2 security groups, and finally **pushing the custom image to Docker Hub** for reuse.

The project uses a simple **CV website (HTML)** served through **Nginx** as the custom Docker image use case.

Objectives

- Explore and understand Docker images
- Pull official images from Docker Hub
- Create a custom Docker image using a Dockerfile
- Run and manage containers
- Configure AWS EC2 Security Groups
- Push Docker images to Docker Hub
- Reuse images via pull commands

Tools & Technologies

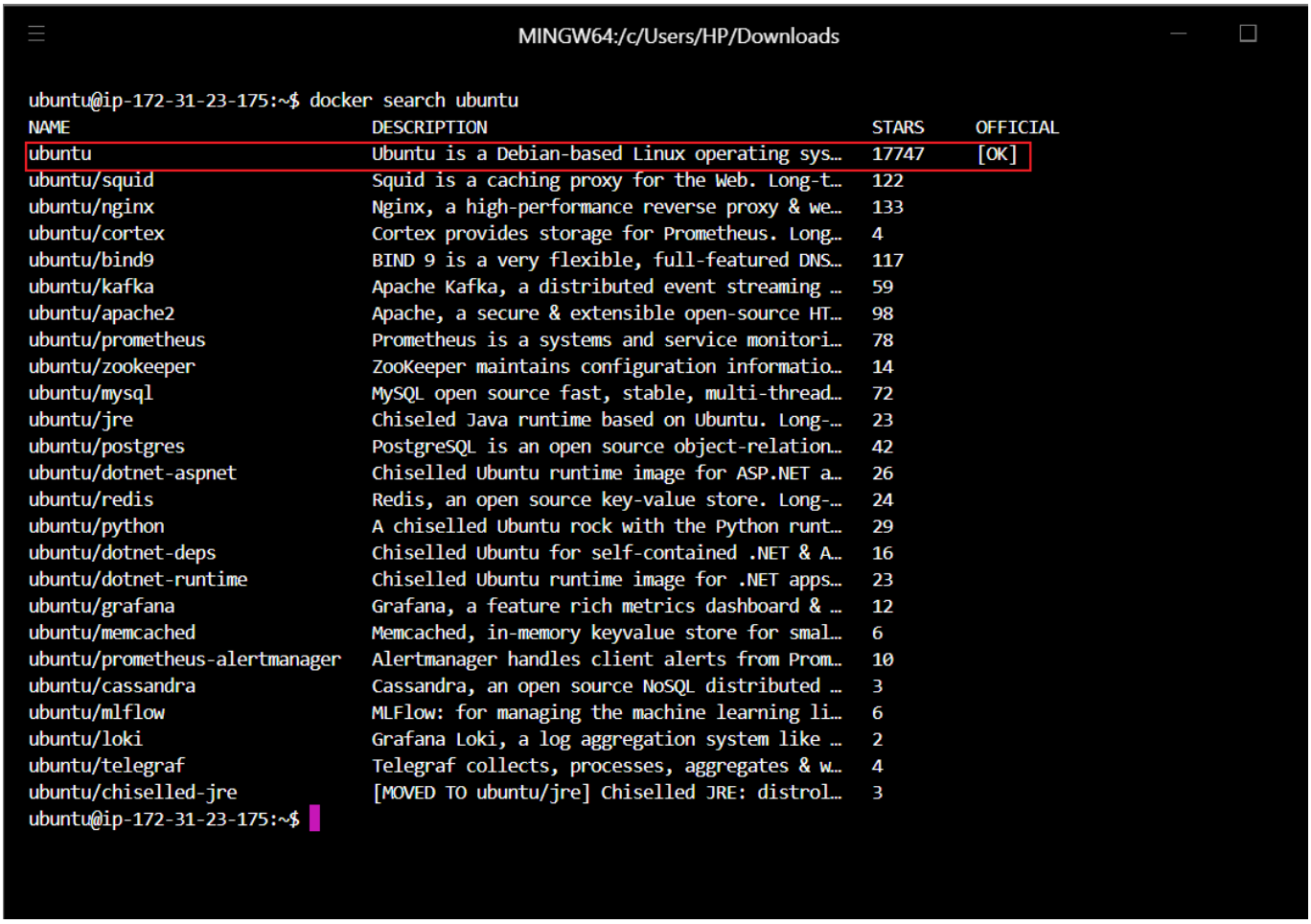
- Docker
- Docker Hub
- Nginx
- AWS EC2
- Linux (Ubuntu)
- HTML
- Vim & Nano editors

Project Execution Steps

Step 1: Discover Docker Images from Docker Hub

Search for available Ubuntu images on Docker Hub.

📸 Screenshot:



Step 2: Pull the Ubuntu Image

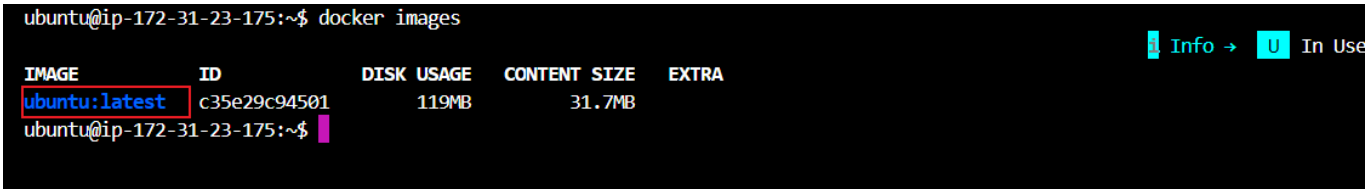
Pull the official Ubuntu image successfully from Docker Hub.

📸 Screenshot:

Step 3: Confirm Pulled Images

List Docker images to confirm that the Ubuntu image was pulled.

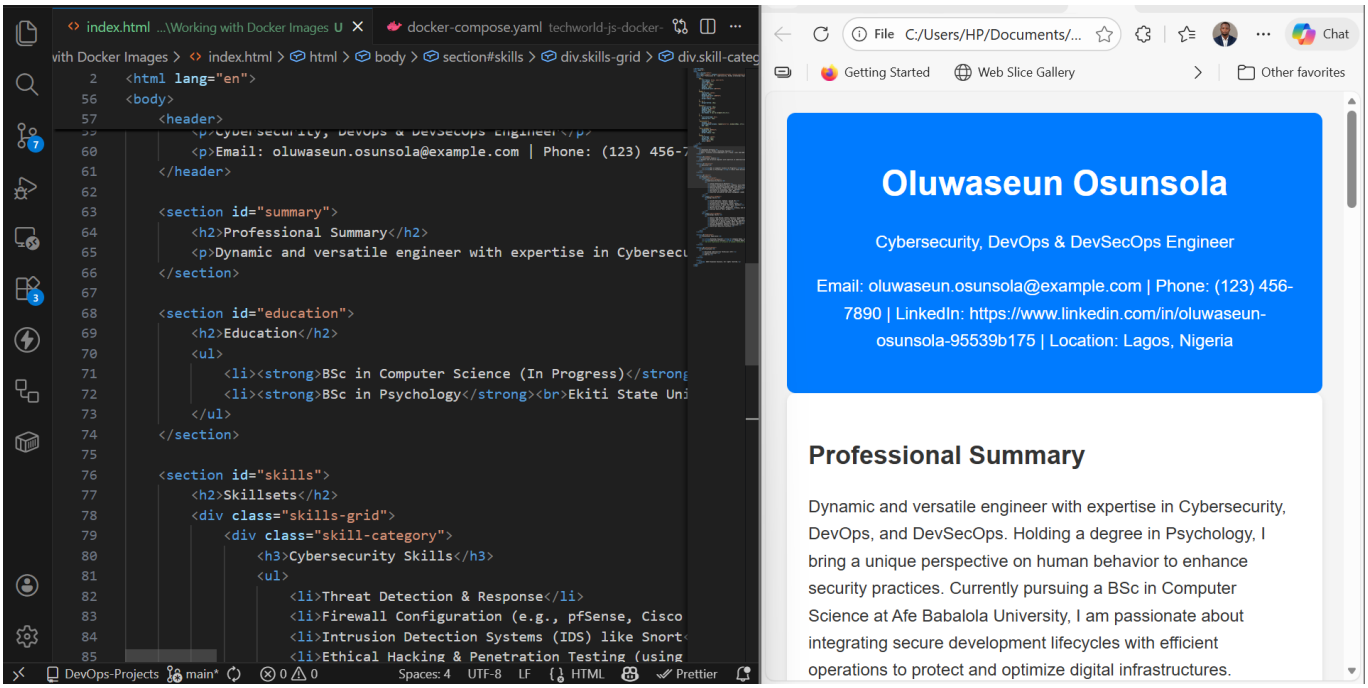
📸 Screenshot:



Step 4: Plan Custom Image Creation

Prepare to create a custom Docker image for a CV website using a Dockerfile.

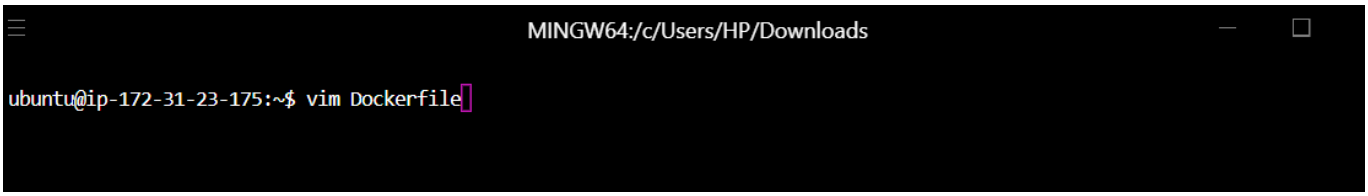
Screenshot:



Step 5: Create Dockerfile

Open and edit the Dockerfile using Vim.

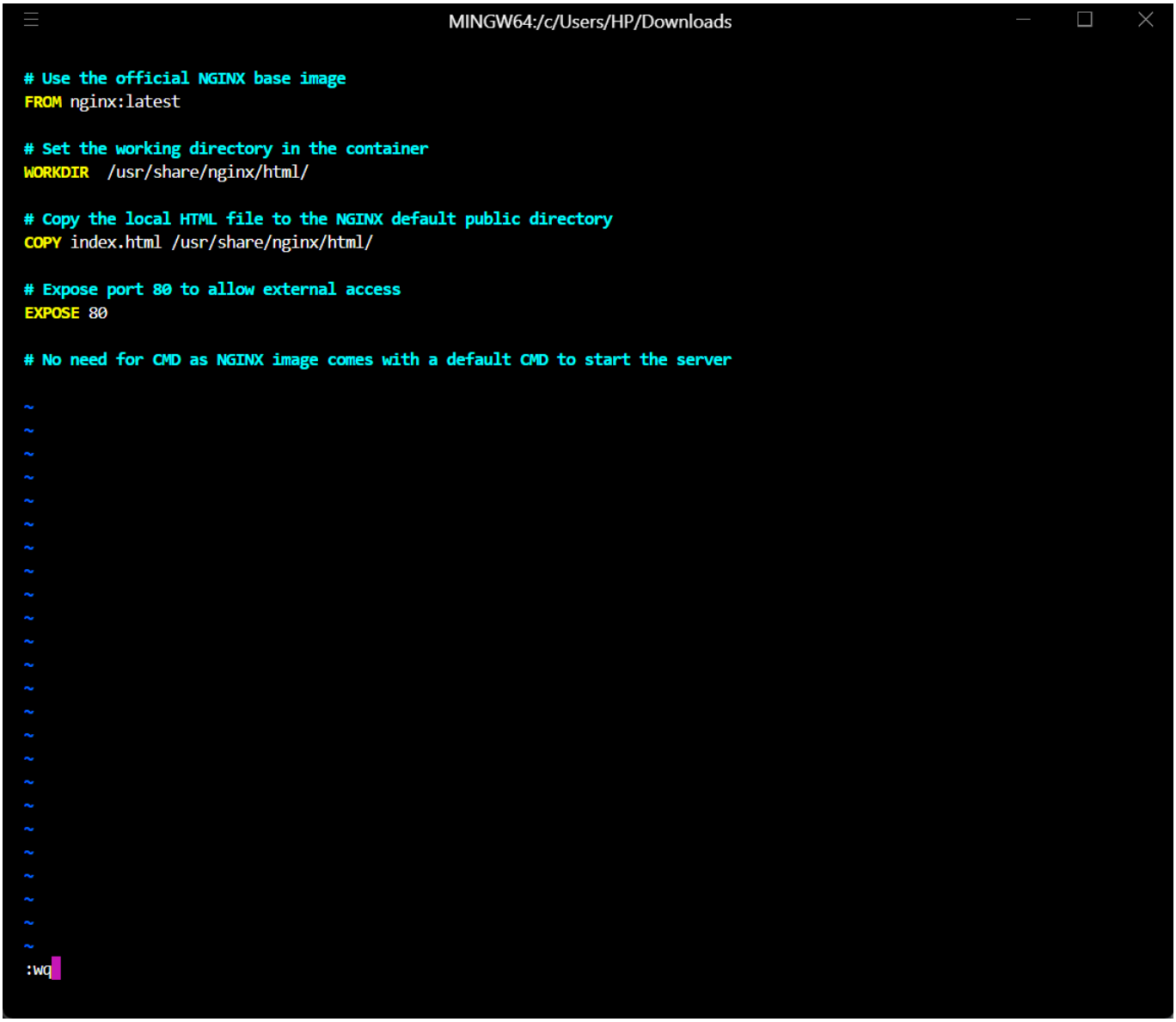
Screenshot:



Step 6: Dockerfile Code

Add Dockerfile instructions and save the file.

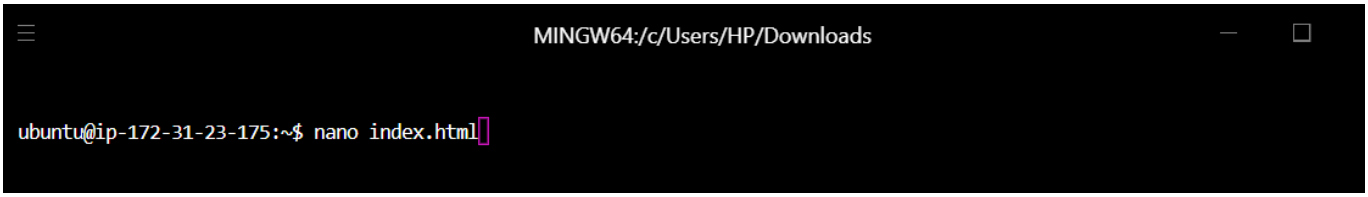
📸 Screenshot:



Step 7: Create HTML File

Create the `index.html` file using Nano editor.

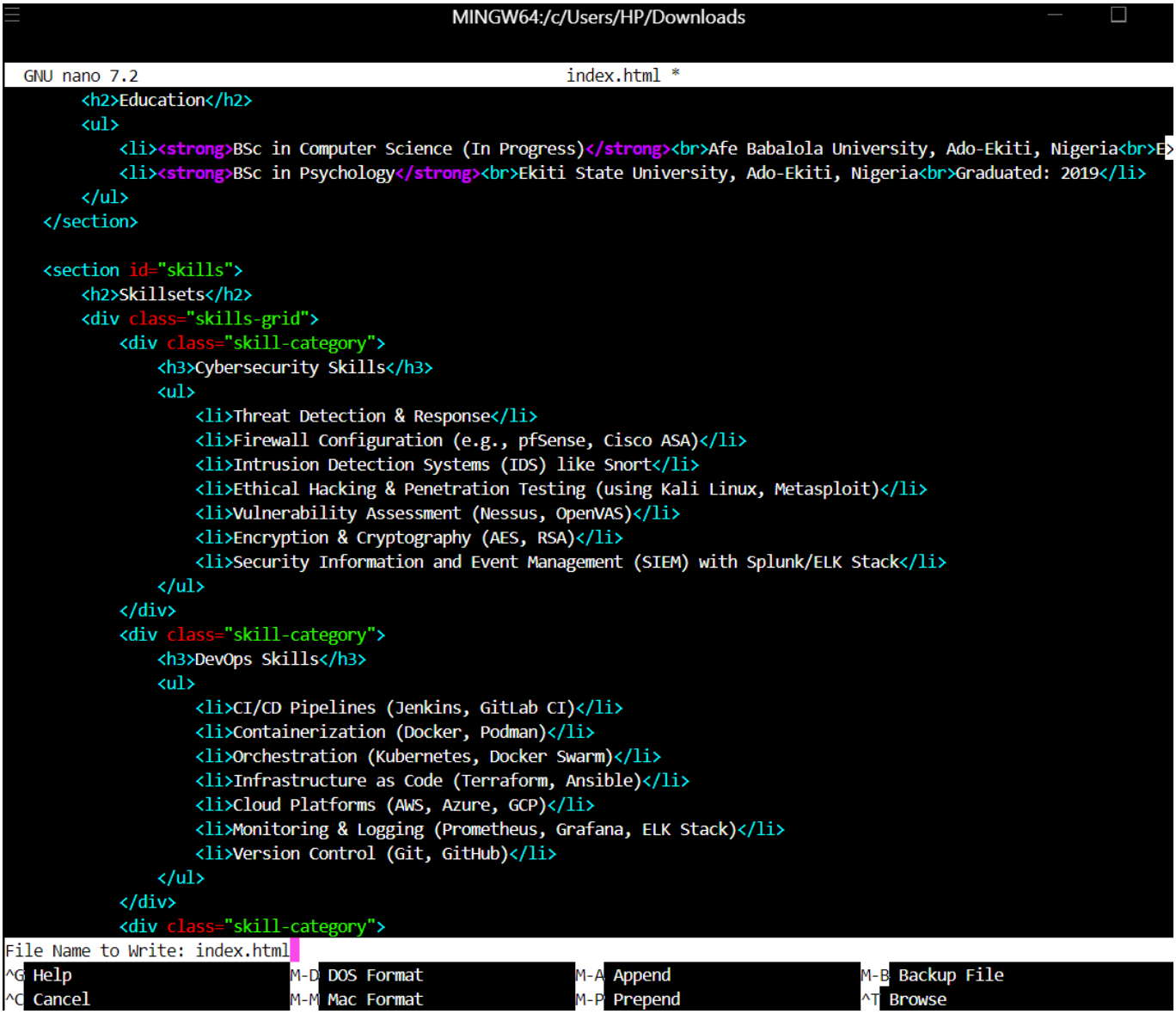
📸 Screenshot:



Step 8: HTML Code

Add website content to the HTML file and save.

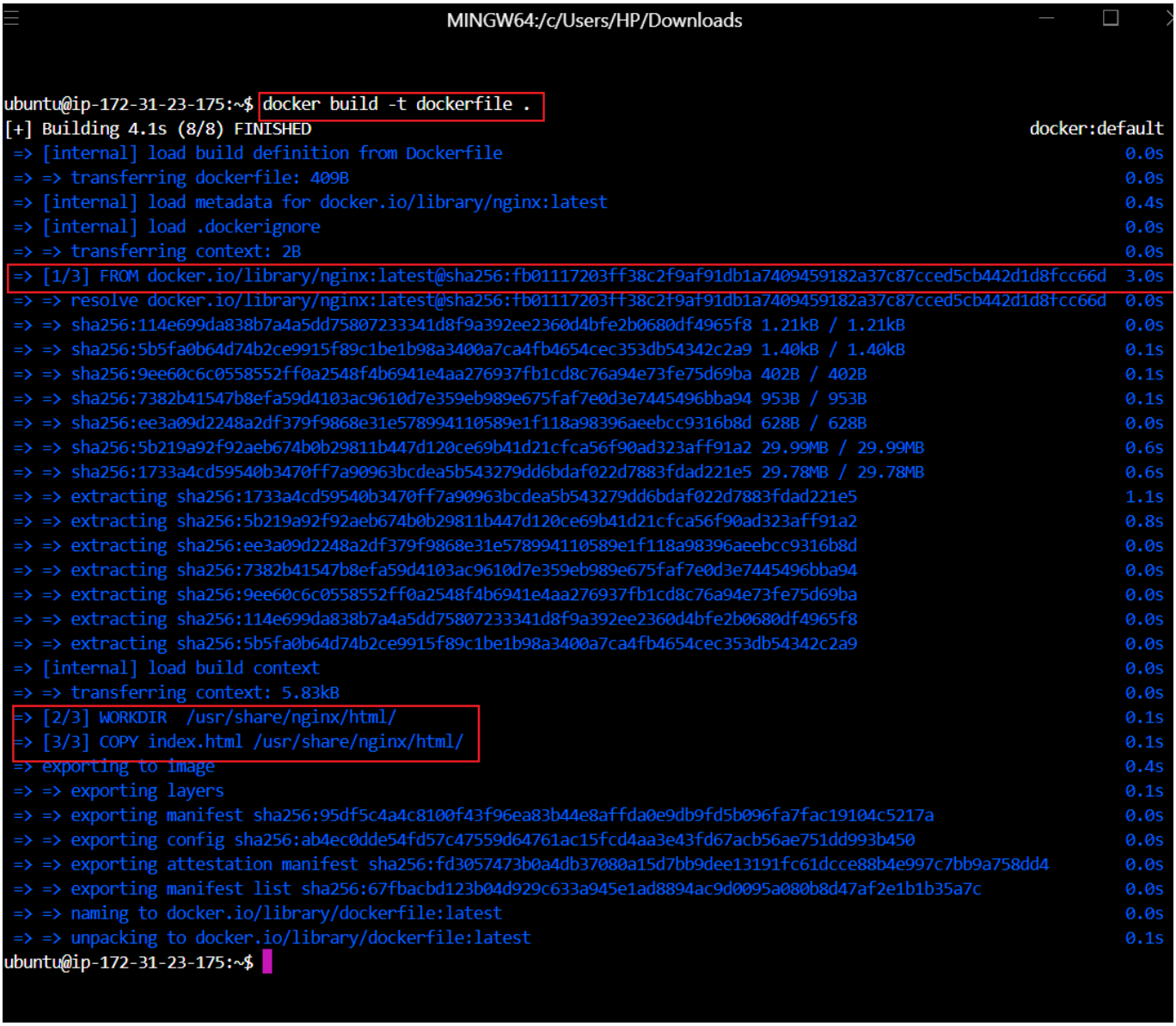
Screenshot:



Step 9: Build Docker Image

Build the Docker image and tag it as `dockerfile`.

📸 Screenshot:



Step 10: Confirm Image Creation

List Docker images to verify successful creation.

📸 Screenshot:

Step 11: Run Container

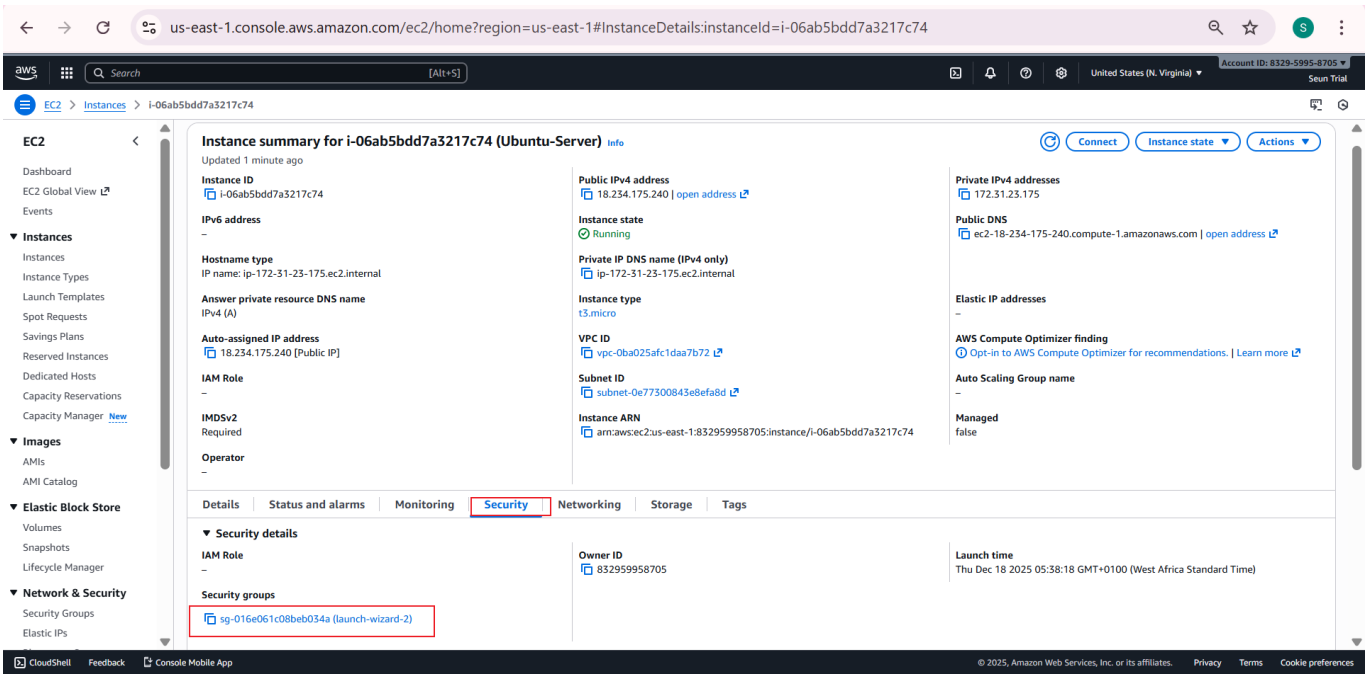
Run a container from the custom Nginx image on port **8080**.

Screenshot:

Step 12: Access EC2 Security Settings

Navigate to the **Security** tab on the EC2 instance dashboard.

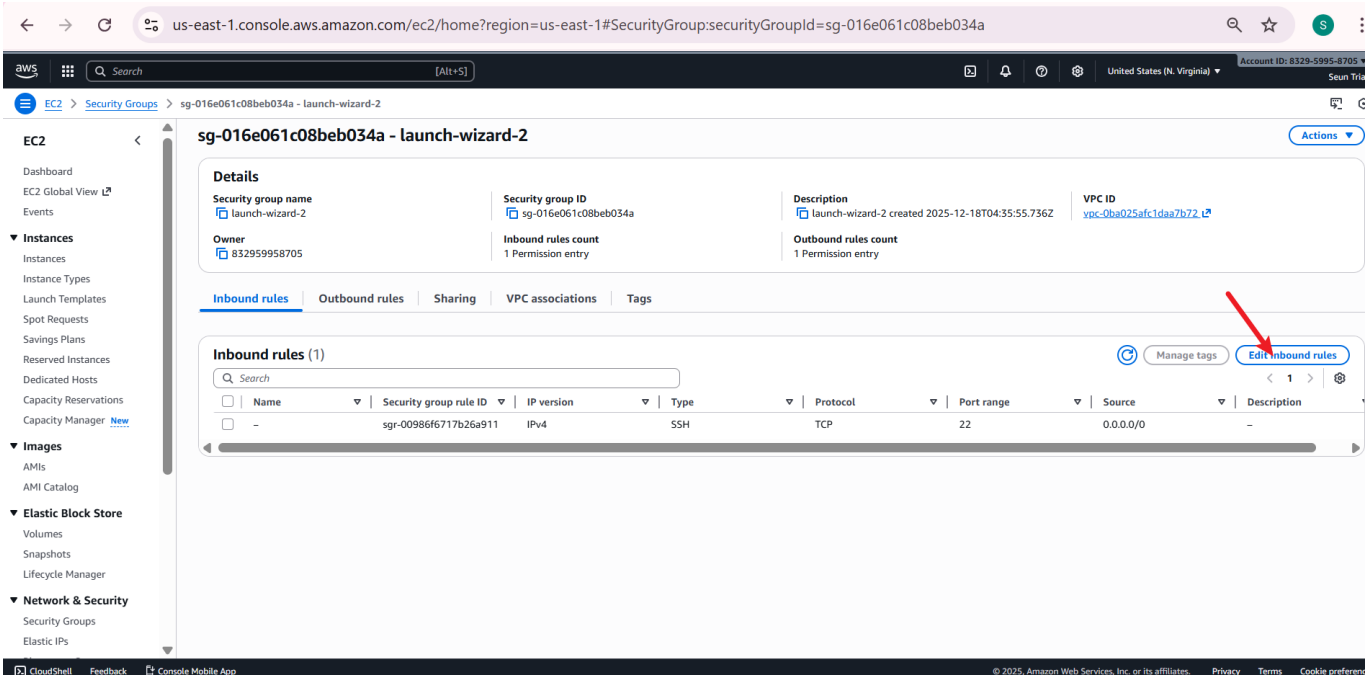
Screenshot:



Step 13: Edit Inbound Rules

Edit inbound rules of the EC2 security group.

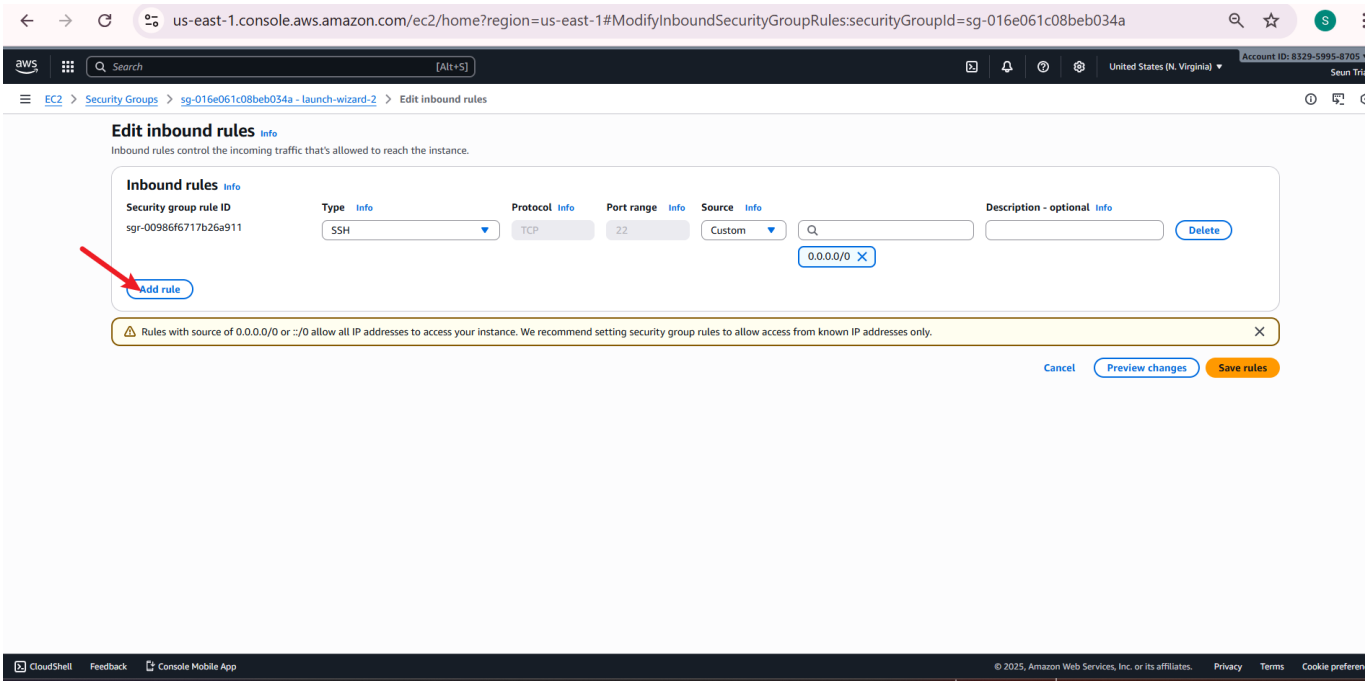
Screenshot:



Step 14: Add New Rule

Add a new inbound rule.

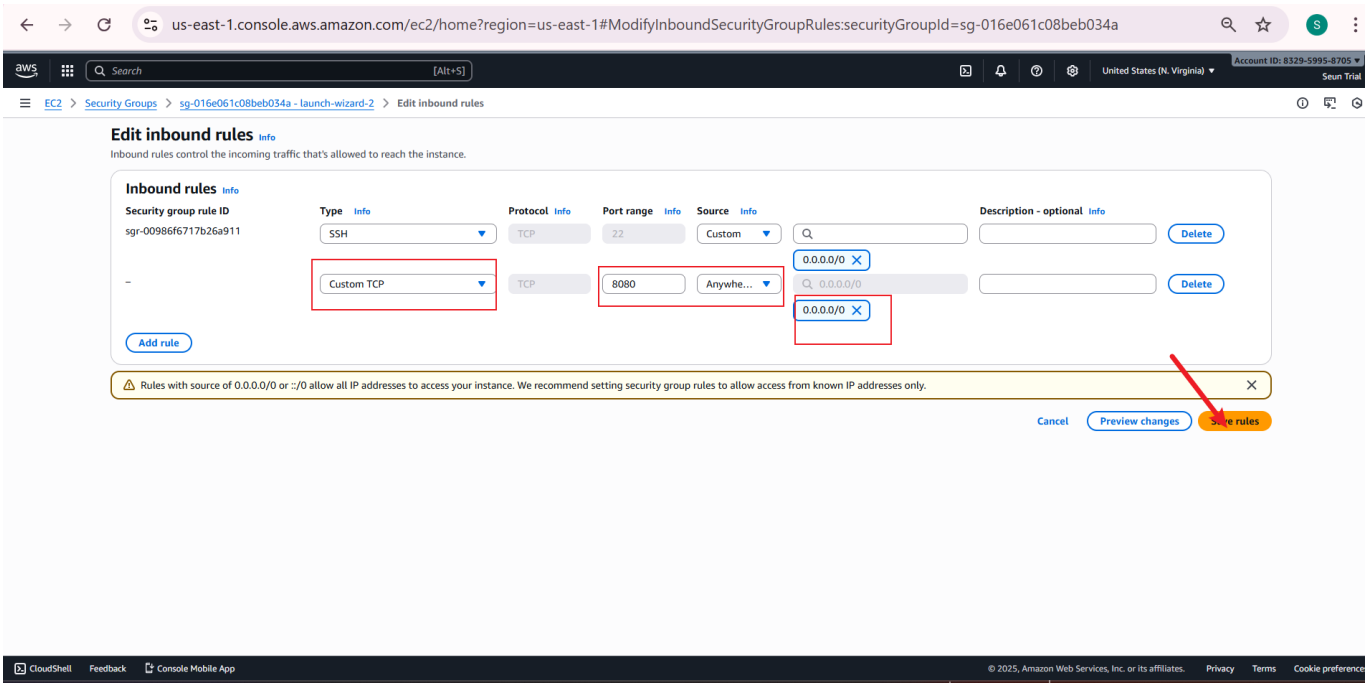
Screenshot:



Step 15: Allow Port 8080

Allow **Custom TCP** traffic on port **8080** from anywhere.

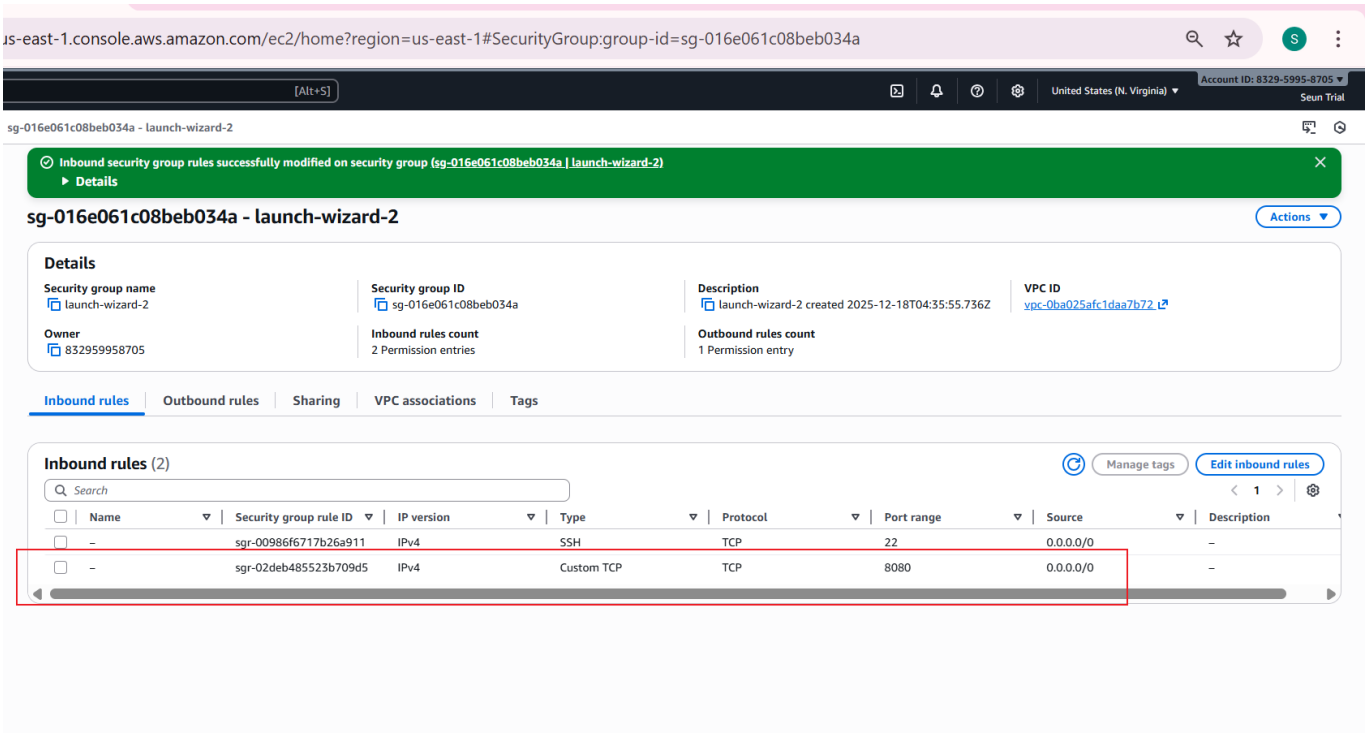
Screenshot:



Step 16: Confirm Rule Update

Inbound rule successfully modified.

Screenshot:



Step 17: Stop & List Containers

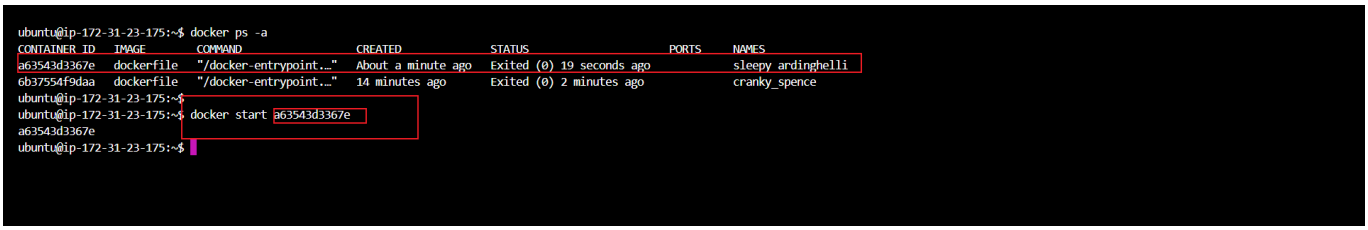
Stop the container and list all containers.

📸 Screenshot:

Step 18: Restart Container

Restart container using its container ID.

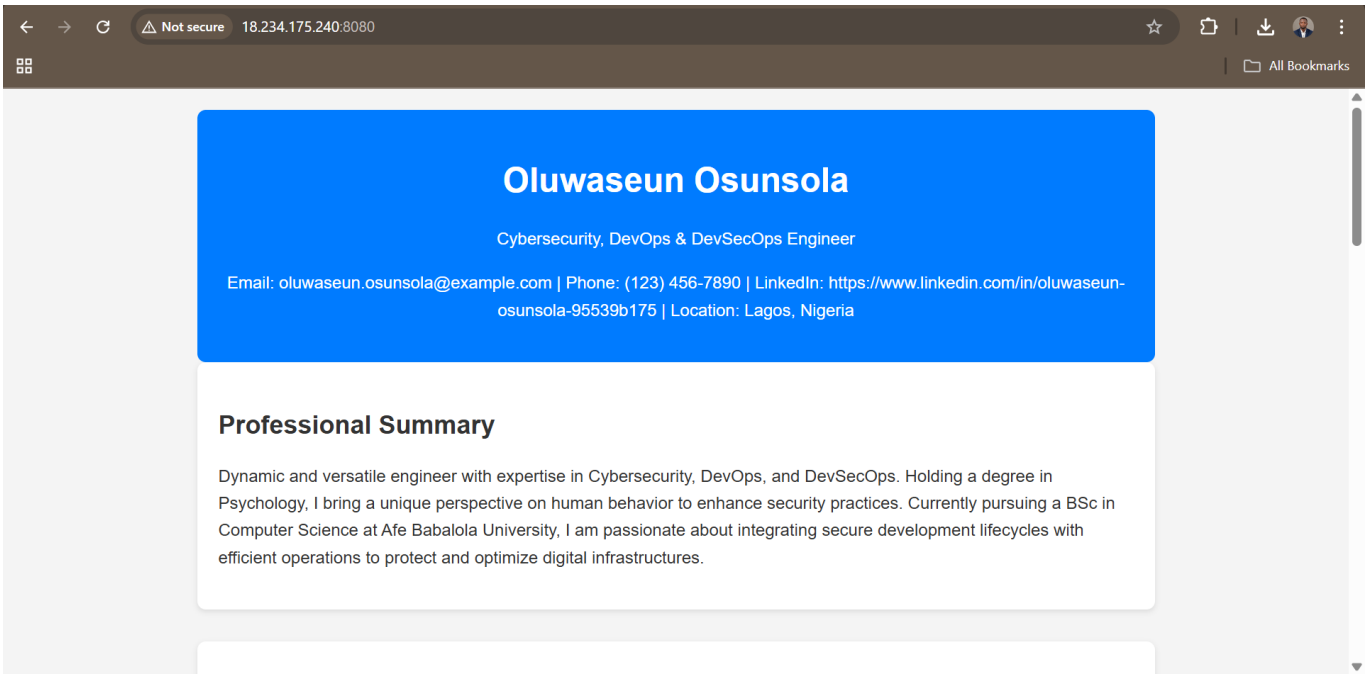
📸 Screenshot:



Step 19: Access Live Website

Access the CV website using EC2 public IP and port **8080**.

📸 Screenshot:

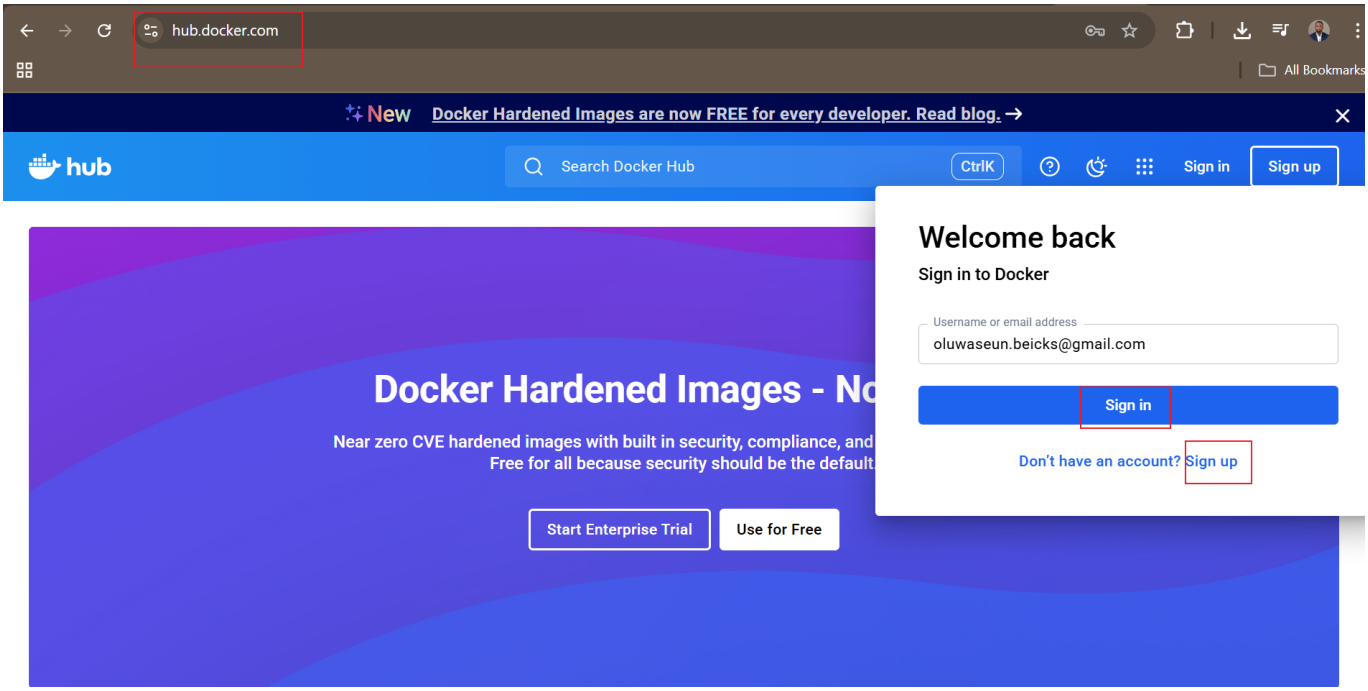


🚀 Docker Hub Integration

Step 20: Sign in to Docker Hub

Proceed to Docker Hub and sign in.

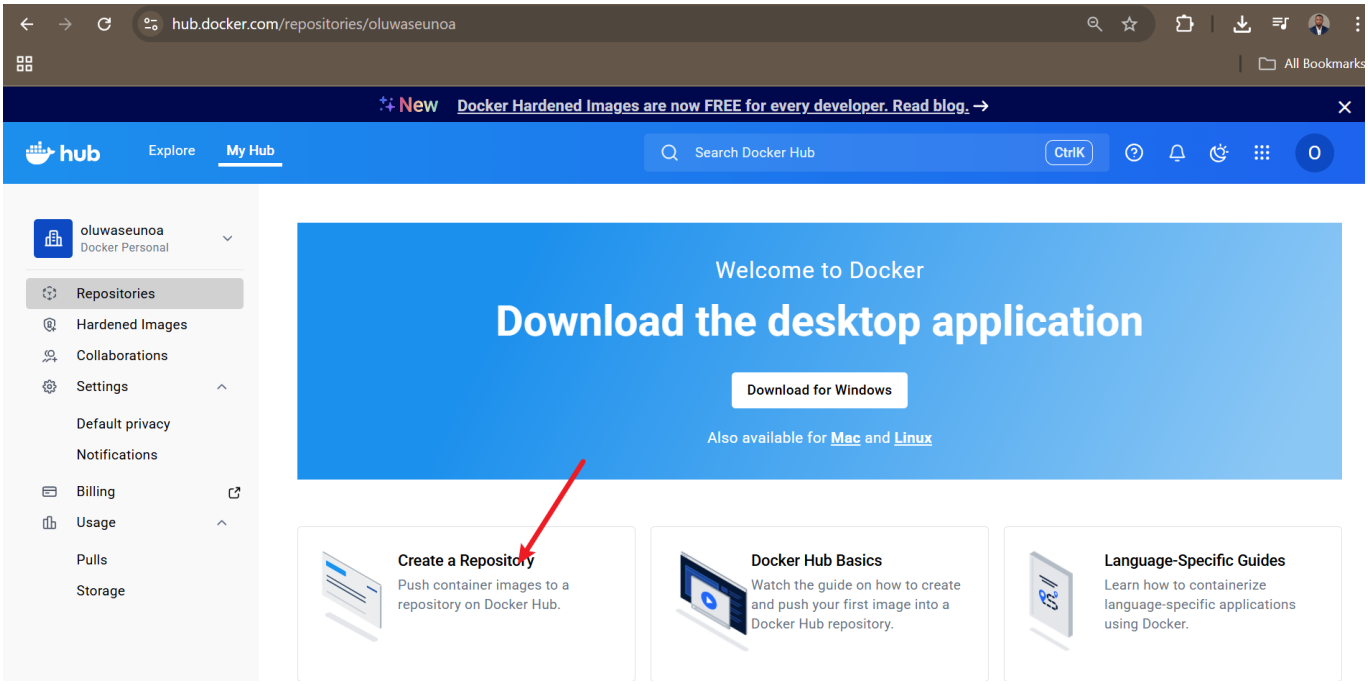
📸 Screenshot:



Step 21: Create Repository

Click **Create Repository** after logging in.

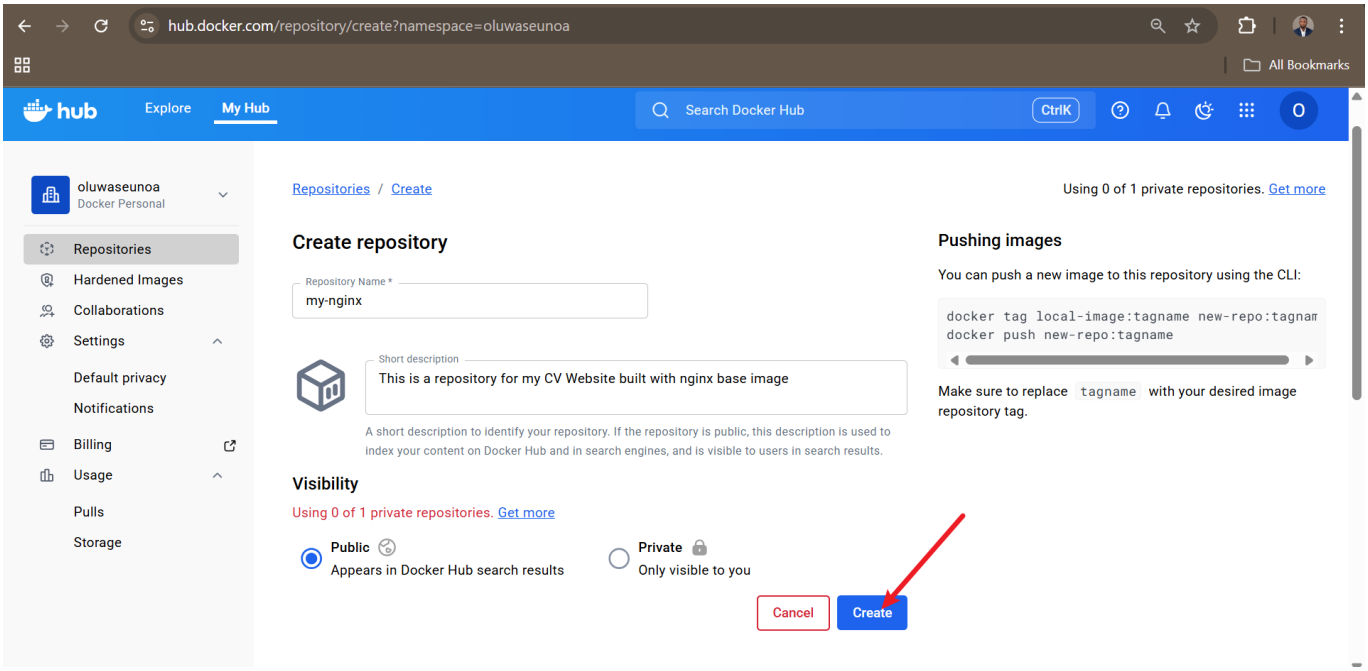
📸 Screenshot:



Step 22: Configure Repository

Name the repository, describe it, set visibility, and create.

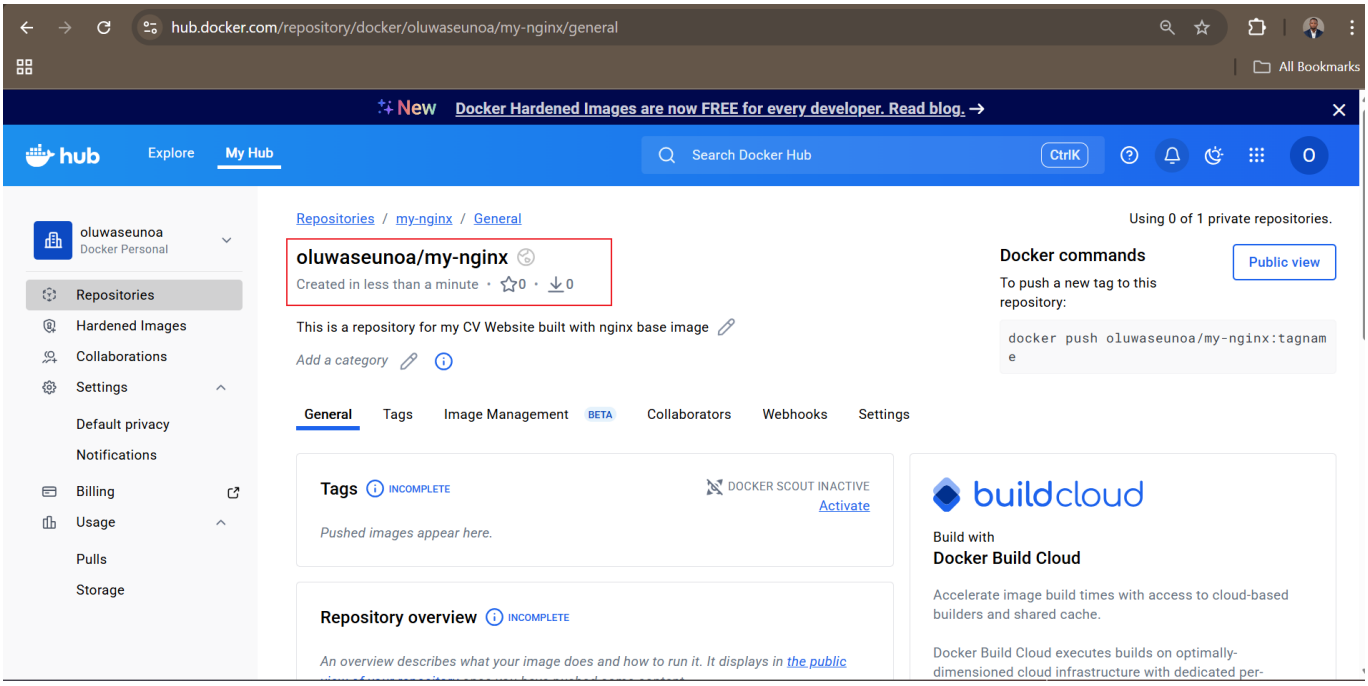
📸 Screenshot:



Step 23: Repository Created

Docker Hub repository created successfully.

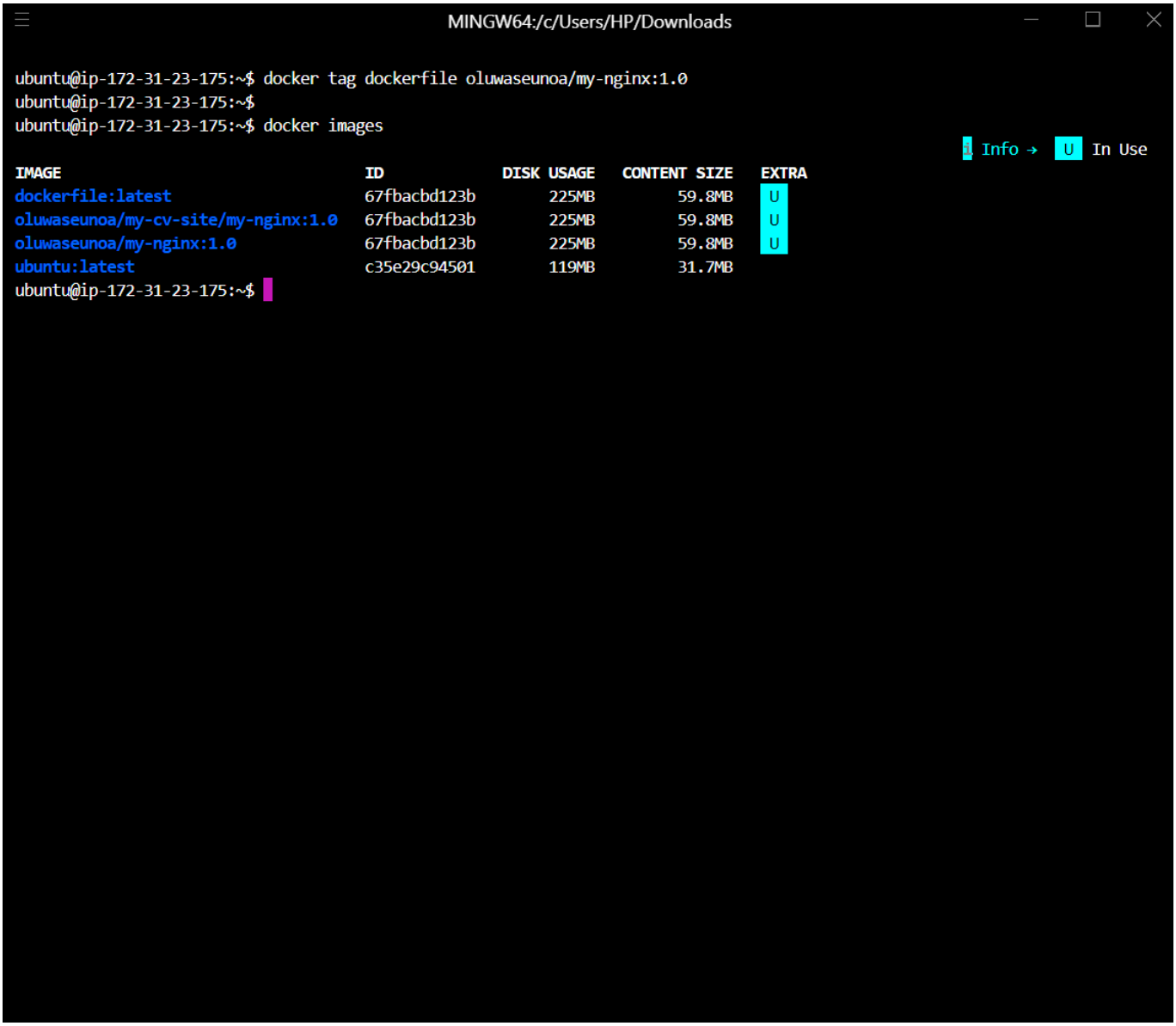
📸 Screenshot:



Step 24: Tag Docker Image

Tag the Docker image and confirm using `docker images`.

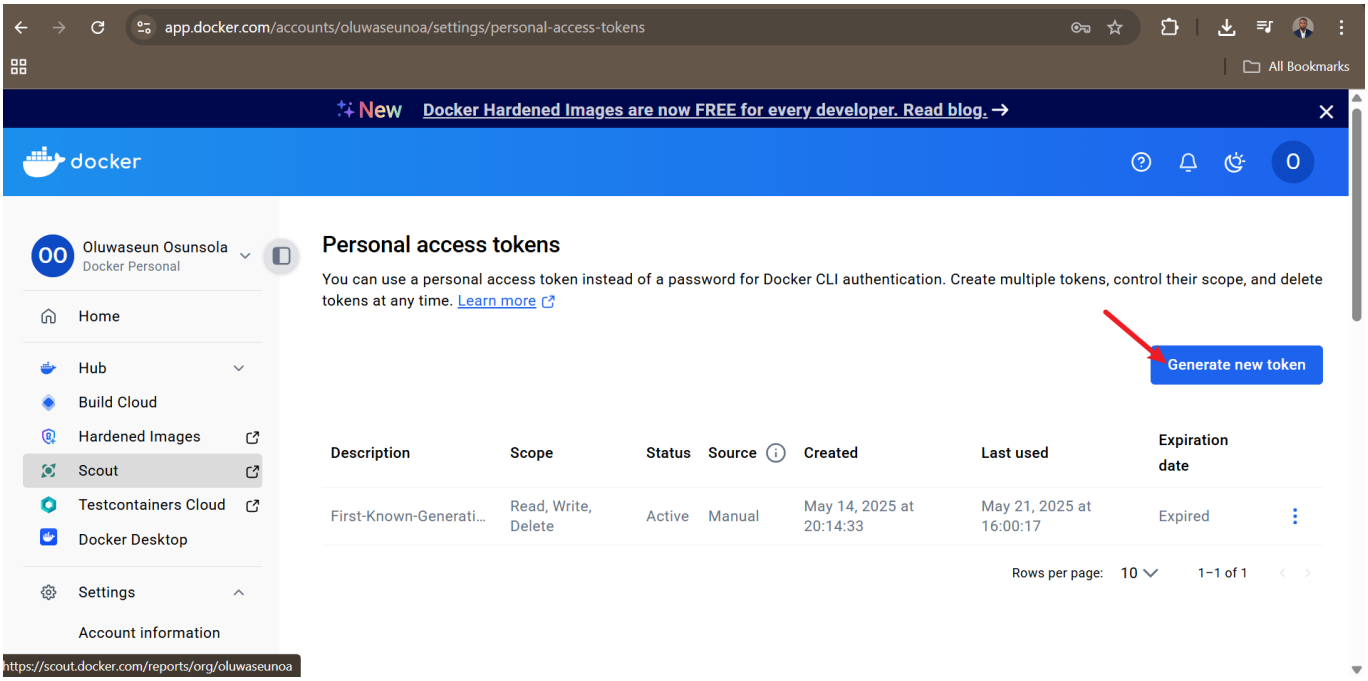
📸 Screenshot:



Step 25: Generate Access Token

Navigate to account settings and generate a personal access token.

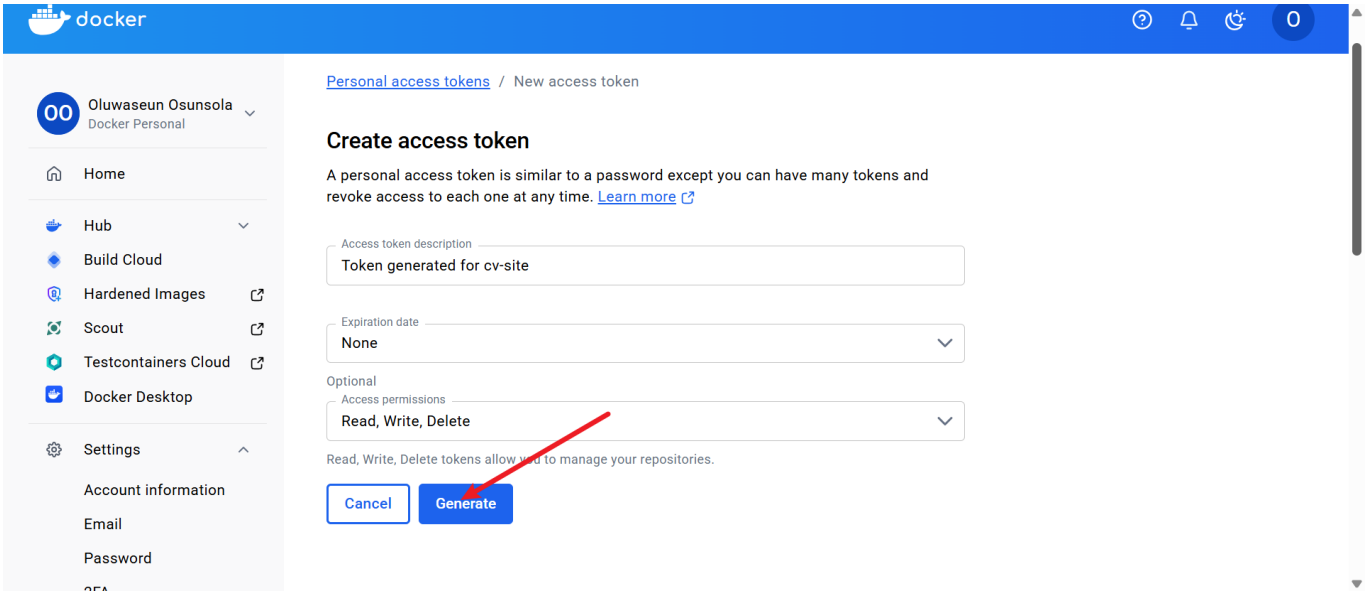
📸 Screenshot:



Step 26: Configure Token

Describe token, set expiration and permissions, then generate.

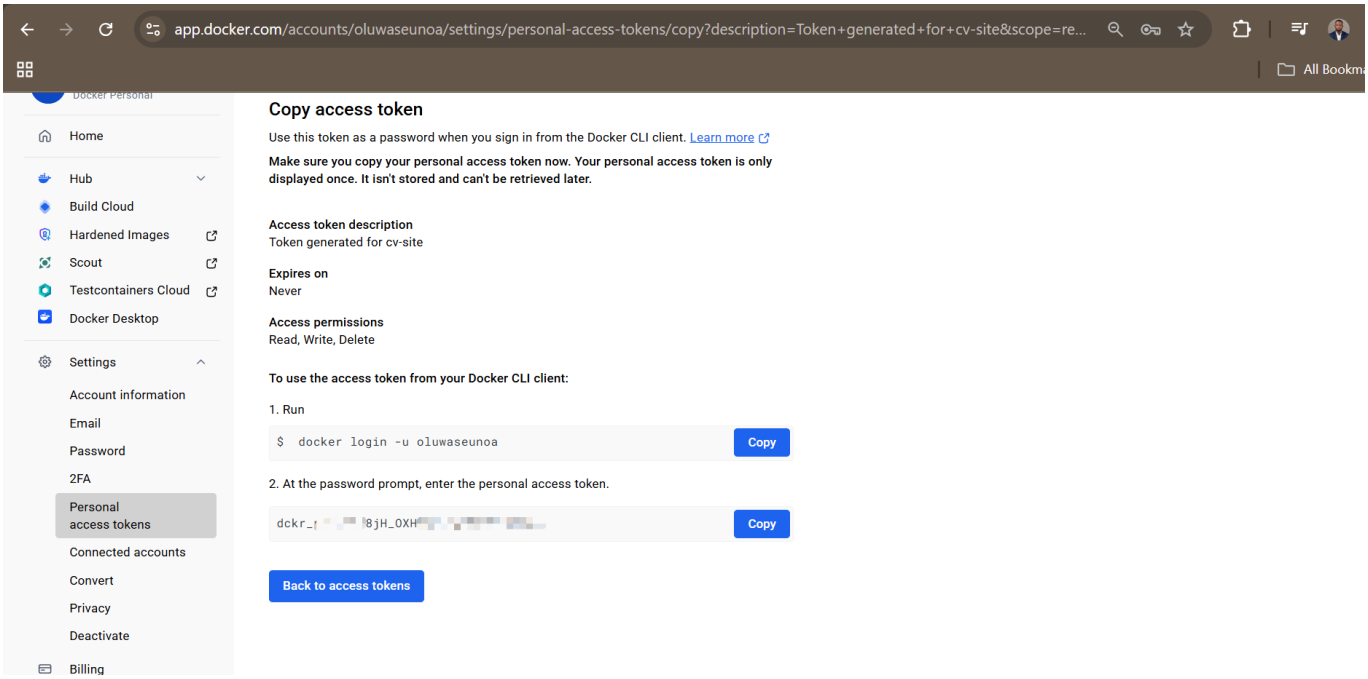
📸 Screenshot:



Step 27: Save Token

Save the generated token securely.

📸 Screenshot:



Step 28: Docker Login via Terminal

Log in to Docker Hub using username and access token.

📸 Screenshot:

Step 29: Push Image to Docker Hub

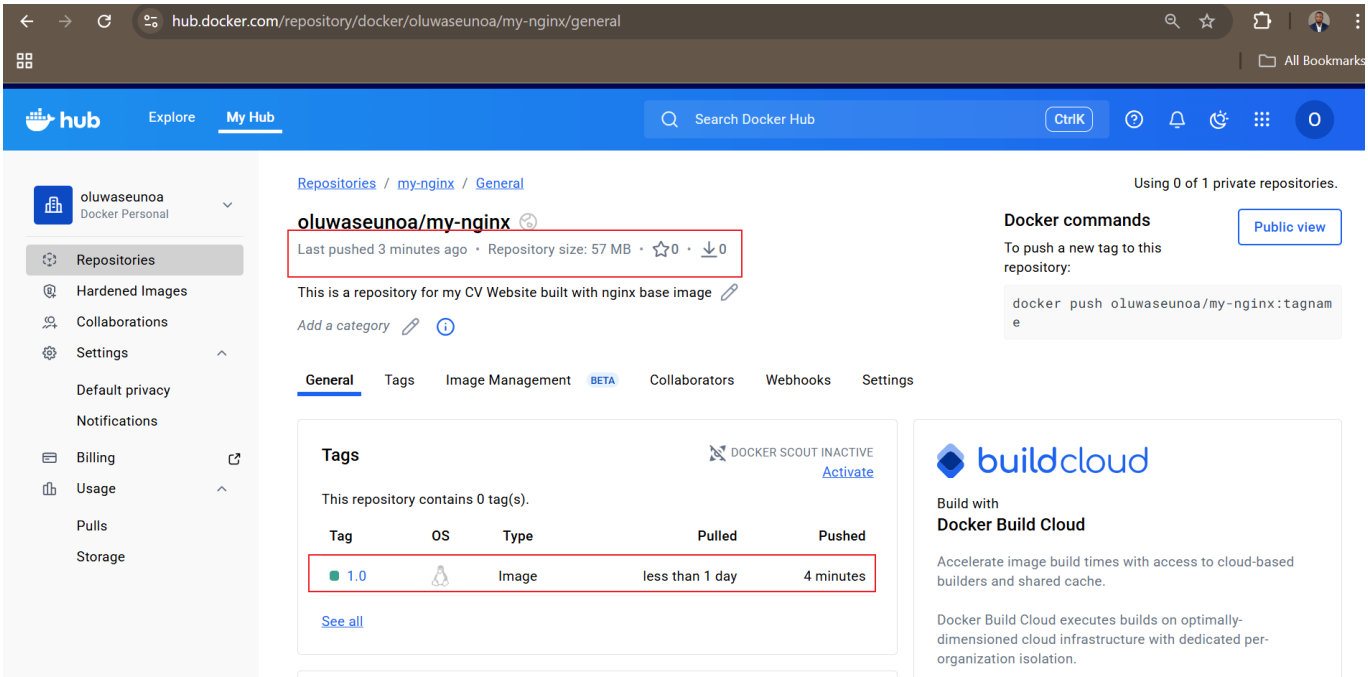
Push the tagged Docker image.

📸 Screenshot:

Step 30: Verify Image Push

Confirm successful image push on Docker Hub.

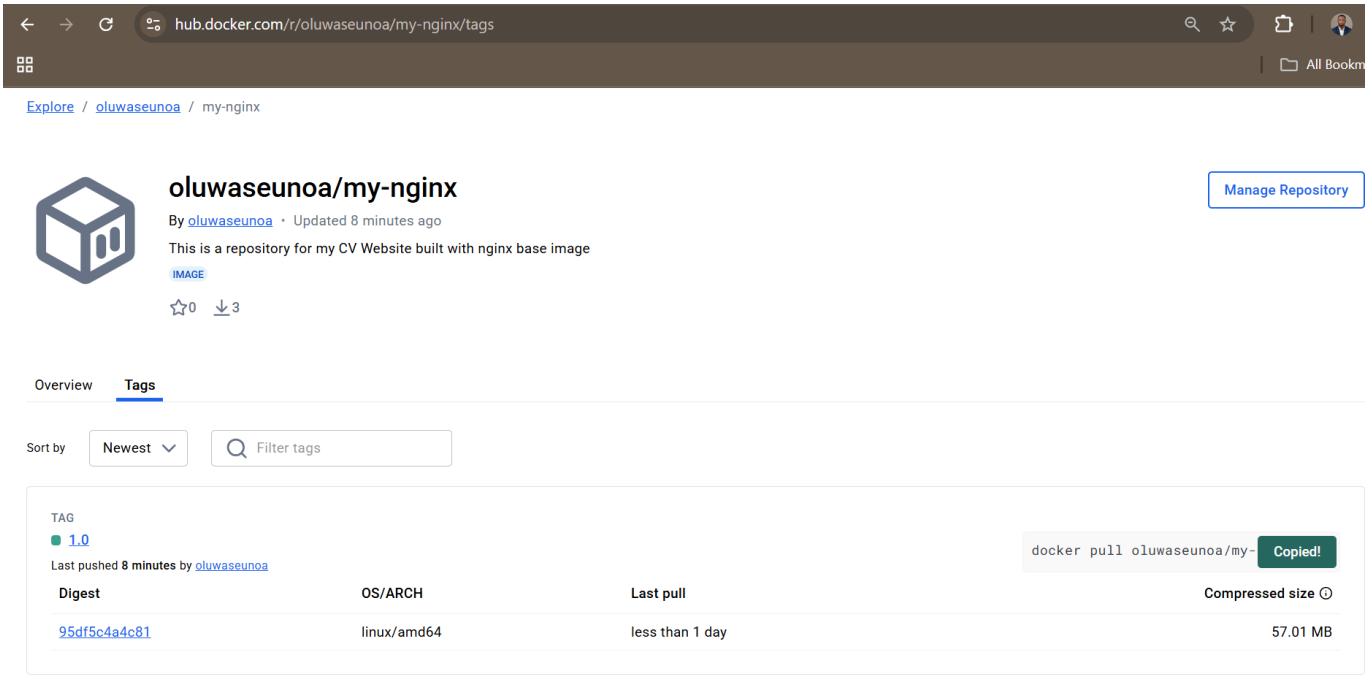
📸 Screenshot:



Step 31: Copy Pull Command

Copy the Docker pull command for reuse.

Screenshot:



☑ Conclusion

This project demonstrates the **complete lifecycle of Docker image management**, from exploration and creation to deployment and distribution. It reinforces best practices for Docker image handling, container management, cloud networking, and image sharing via Docker Hub.

Repository Structure

```
.
├── Dockerfile
├── index.html
├── img/
│   ├── *.png
└── README.md
```

Dockerfile Snippet

```
# Use the official NGINX base image
FROM nginx:latest

# Set the working directory in the container
WORKDIR /usr/share/nginx/html/

# Copy the local HTML fil to the NGINX default public directory
COPY index.html /usr/share/nginx/html/

# Expose port 80 to allow external access
```

```
EXPOSE 80

# No need for CMD as NGINX image comes with a default CMD to start the server
```

index.html Snippet

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Your Professional CV - Cybersecurity, DevOps & DevSecOps
Engineer</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            line-height: 1.6;
            color: #333;
            max-width: 900px;
            margin: 0 auto;
            padding: 20px;
            background-color: #f4f4f4;
        }
        header {
            text-align: center;
            padding: 20px;
            background-color: #007bff;
            color: white;
            border-radius: 8px;
        }
        h1, h2 {
            margin-bottom: 10px;
        }
        section {
            margin-bottom: 30px;
            background: white;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 2px 5px rgba(0,0,0,0.1);
        }
        ul {
            list-style-type: disc;
            padding-left: 20px;
        }
        .skills-grid {
            display: grid;
            grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
            gap: 20px;
        }
        .skill-category {
```

```
            background: #e9ecef;
            padding: 15px;
            border-radius: 8px;
        }
        footer {
            text-align: center;
            margin-top: 40px;
            font-size: 0.9em;
            color: #666;
        }
    </style>
</head>
<body>
    <header>
        <h1>Oluwaseun Osunsola</h1>
        <p>Cybersecurity, DevOps & DevSecOps Engineer</p>
        <p>Email: oluwaseun.osunsola@example.com | Phone: (123) 456-7890 |
LinkedIn: https://www.linkedin.com/in/oluwaseun-osunsola-95539b175 | Location:
Lagos, Nigeria</p>
    </header>

    <section id="summary">
        <h2>Professional Summary</h2>
        <p>Dynamic and versatile engineer with expertise in Cybersecurity, DevOps,
and DevSecOps. Holding a degree in Psychology, I bring a unique perspective on
human behavior to enhance security practices. Currently pursuing a BSc in Computer
Science at Afe Babalola University, I am passionate about integrating secure
development lifecycles with efficient operations to protect and optimize digital
infrastructures.</p>
    </section>

    <section id="education">
        <h2>Education</h2>
        <ul>
            <li><strong>BSc in Computer Science (In Progress)</strong><br>Afe
Babalola University, Ado-Ekiti, Nigeria<br>Expected Graduation: 2028</li>
            <li><strong>BSc in Psychology</strong><br>Ekiti State University, Ado-
Ekiti, Nigeria<br>Graduated: 2019</li>
        </ul>
    </section>

    <section id="skills">
        <h2>Skillsets</h2>
        <div class="skills-grid">
            <div class="skill-category">
                <h3>Cybersecurity Skills</h3>
                <ul>
                    <li>Threat Detection & Response</li>
                    <li>Firewall Configuration (e.g., pfSense, Cisco ASA)</li>
                    <li>Intrusion Detection Systems (IDS) like Snort</li>
                    <li>Ethical Hacking & Penetration Testing (using Kali Linux,
Metasploit)</li>
                    <li>Vulnerability Assessment (Nessus, OpenVAS)</li>
                    <li>Encryption & Cryptography (AES, RSA)</li>
```

```
                    <li>Security Information and Event Management (SIEM) with
Splunk/ELK Stack</li>
                </ul>
            </div>
            <div class="skill-category">
                <h3>DevOps Skills</h3>
                <ul>
                    <li>CI/CD Pipelines (Jenkins, GitLab CI)</li>
                    <li>Containerization (Docker, Podman)</li>
                    <li>Orchestration (Kubernetes, Docker Swarm)</li>
                    <li>Infrastructure as Code (Terraform, Ansible)</li>
                    <li>Cloud Platforms (AWS, Azure, GCP)</li>
                    <li>Monitoring & Logging (Prometheus, Grafana, ELK Stack)</li>
                    <li>Version Control (Git, GitHub)</li>
                </ul>
            </div>
            <div class="skill-category">
                <h3>DevSecOps Skills</h3>
                <ul>
                    <li>Secure Code Review & Static Analysis (SonarQube)</li>
                    <li>Dynamic Application Security Testing (DAST) with OWASP
ZAP</li>
                    <li>Integration of Security in CI/CD (Snyk, Checkmarx)</li>
                    <li>Compliance & Auditing (HIPAA, GDPR, PCI-DSS)</li>
                    <li>Secrets Management (HashiCorp Vault, AWS Secrets Manager)
</li>
                    <li>Automated Security Testing in Pipelines</li>
                    <li>Shift-Left Security Practices</li>
                </ul>
            </div>
        </div>
    </section>

    <section id="experience">
        <h2>Professional Experience</h2>
        <ul>
            <li><strong>DevSecOps Engineer</strong><br>[Company Name], [Location]
<br>[Dates]<br>- Implemented secure DevOps pipelines reducing vulnerabilities by
40%.<br>- Managed Kubernetes clusters with integrated security controls.</li>
            <li><strong>Cybersecurity Analyst</strong><br>[Company Name],
[Location]<br>[Dates]<br>- Conducted penetration tests and remediated findings.
<br>- Monitored network traffic for anomalies using SIEM tools.</li>
            <!-- Add more experiences or projects as needed; customize with your
real details -->
        </ul>
    </section>

    <section id="certifications">
        <h2>Certifications</h2>
        <ul>
            <li>Certified Cybersecurity Technician (CCT)</li>
            <li>CompTIA Security+</li>
            <li>ISC2 CC</li>
        </ul>
```

```
    </section>

    <footer>
        <p>&copy; 2025 Oluwaseun Osunsola. All rights reserved.</p>
    </footer>
</body>
</html>
```