

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

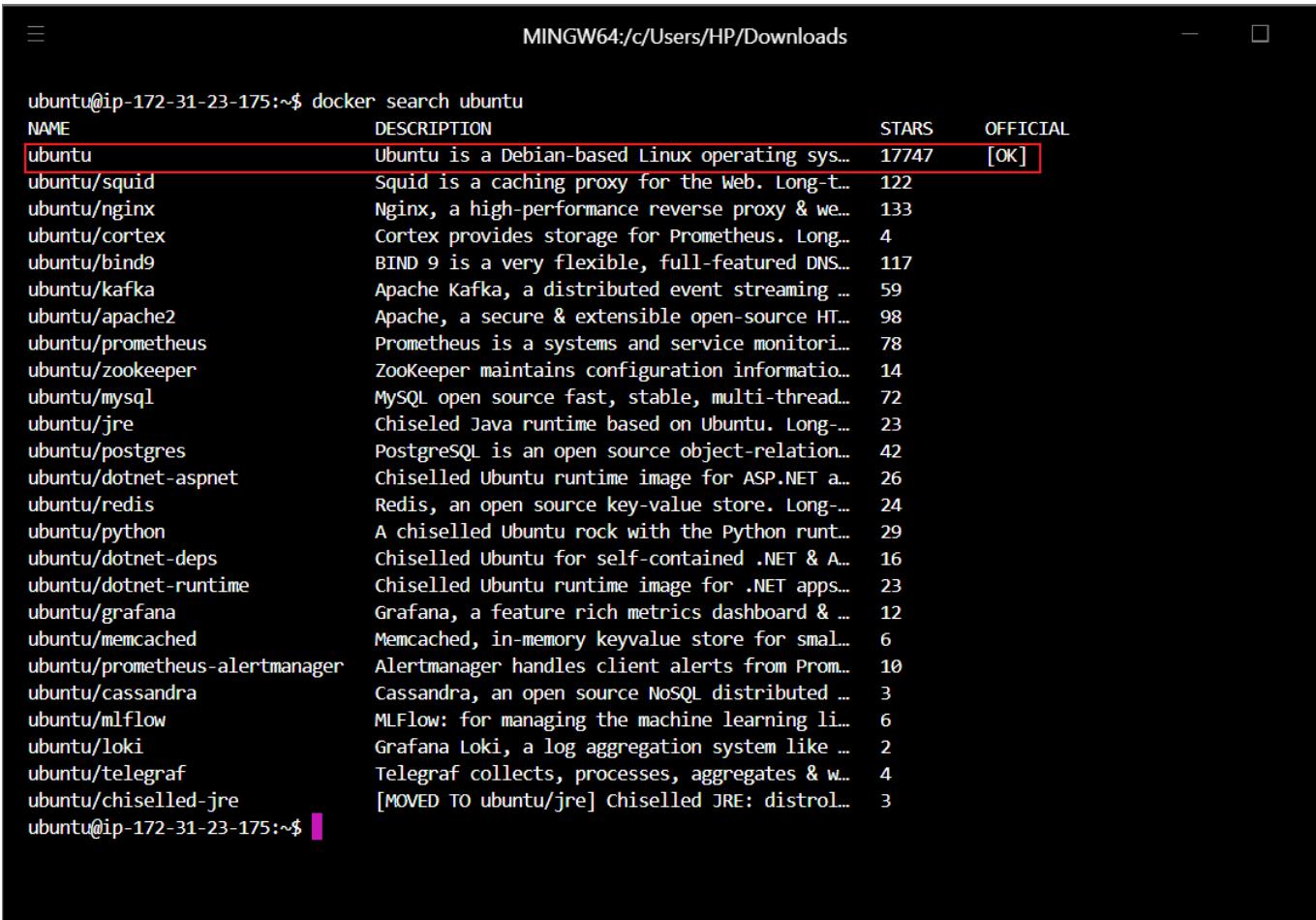
- Docker
 - Docker Hub
 - Nginx
 - AWS EC2
 - Linux (Ubuntu)
 - HTML
 - Vim & Nano editors
-

Project Execution Steps

Step 1: Discover Docker Images from Docker Hub

Search for available Ubuntu images on Docker Hub.

Screenshot:

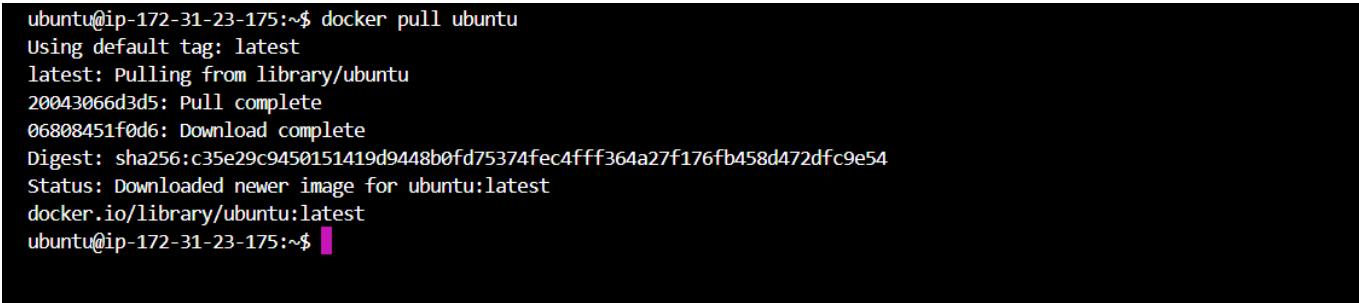


```
MINGW64:/c/Users/HP/Downloads
ubuntu@ip-172-31-23-175:~$ docker search ubuntu
NAME                           DESCRIPTION                               STARS   OFFICIAL
ubuntu                          Ubuntu is a Debian-based Linux operating sys... 17747  [OK]
ubuntu/squid                    Squid is a caching proxy for the Web. Long-t...
ubuntu/nginx                   Nginx, a high-performance reverse proxy & we...
ubuntu/cortex                   Cortex provides storage for Prometheus. Long...
ubuntu/bind9                    BIND 9 is a very flexible, full-featured DNS...
ubuntu/kafka                     Apache Kafka, a distributed event streaming ...
ubuntu/apache2                  Apache, a secure & extensible open-source HT...
ubuntu/prometheus                Prometheus is a systems and service monitori...
ubuntu/zookeeper                 ZooKeeper maintains configuration informatio...
ubuntu/mysql                     MySQL open source fast, stable, multi-thread...
ubuntu/jre                       Chiseled Java runtime based on Ubuntu. Long...
ubuntu/postgres                  PostgreSQL is an open source object-relation...
ubuntu/dotnet-aspnet              Chiselled Ubuntu runtime image for ASP.NET a...
ubuntu/redis                     Redis, an open source key-value store. Long ...
ubuntu/python                    A chiseled Ubuntu rock with the Python runt...
ubuntu/dotnet-deps               Chiselled Ubuntu for self-contained .NET & A...
ubuntu/dotnet-runtime             Chiselled Ubuntu runtime image for .NET apps...
ubuntu/grafana                   Grafana, a feature rich metrics dashboard & ...
ubuntu/memcached                 Memcached, in-memory keyvalue store for smal...
ubuntu/prometheus-alertmanager  Alertmanager handles client alerts from Prom...
ubuntu/cassandra                 Cassandra, an open source NoSQL distributed ...
ubuntu/mlflow                     MLflow: for managing the machine learning li...
ubuntu/loki                       Grafana Loki, a log aggregation system like ...
ubuntu/telegraf                  Telegraf collects, processes, aggregates & w...
ubuntu/chiselled-jre             [MOVED TO ubuntu/jre] Chiselled JRE: distrol...
ubuntu@ip-172-31-23-175:~$
```

Step 2: Pull the Ubuntu Image

Pull the official Ubuntu image successfully from Docker Hub.

Screenshot:

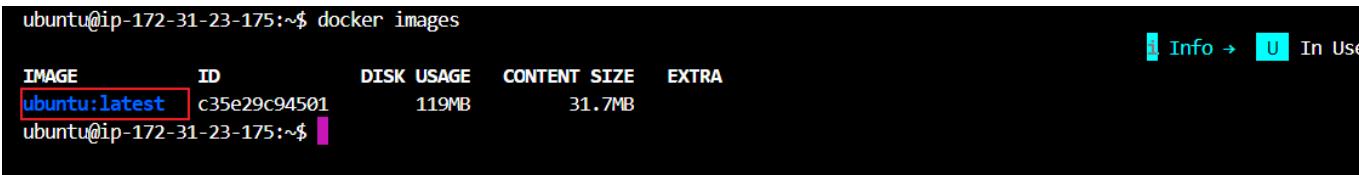


```
ubuntu@ip-172-31-23-175:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
20043066d3d5: Pull complete
06808451f0d6: Download complete
Digest: sha256:c35e29c9450151419d9448b0fd75374fec4fff364a27f176fb458d472dfc9e54
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
ubuntu@ip-172-31-23-175:~$
```

Step 3: Confirm Pulled Images

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

Screenshot:

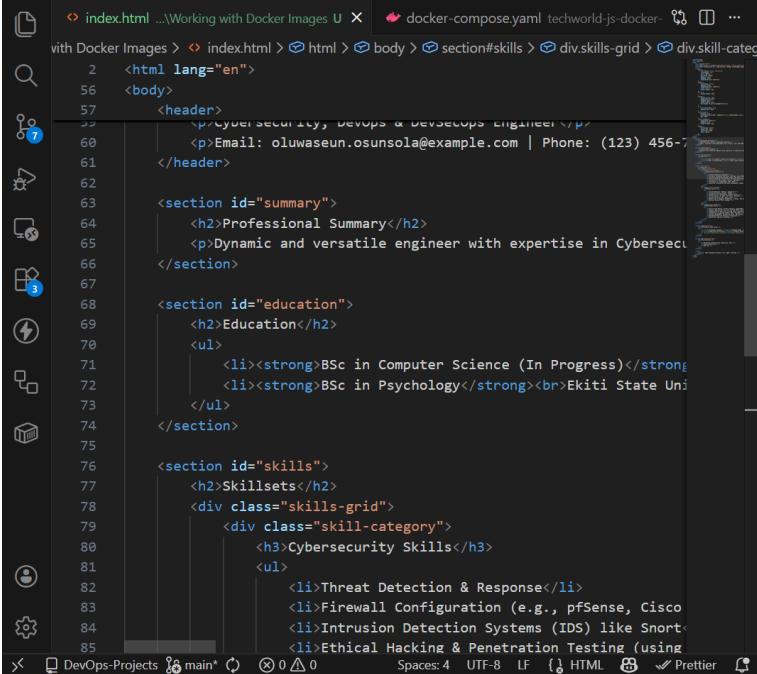
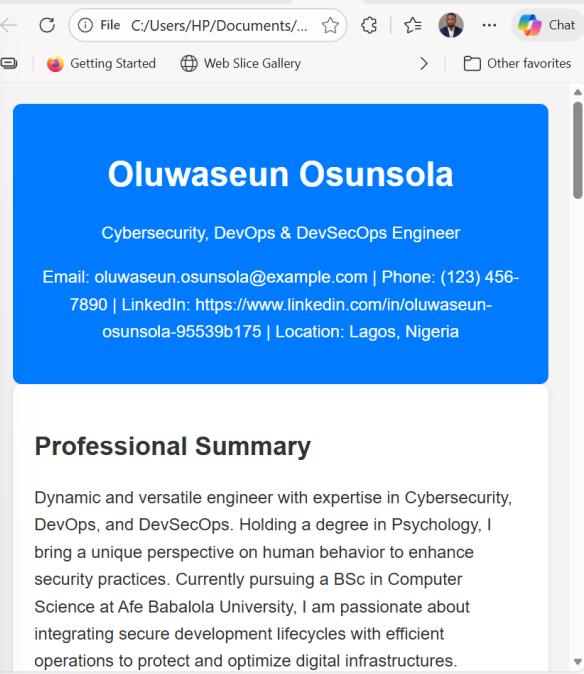


| IMAGE | ID | DISK USAGE | CONTENT SIZE | EXTRA |
|-----------------------------|--------------|------------|--------------|---|
| ubuntu:latest | c35e29c94501 | 119MB | 31.7MB |  Info →  In Use |
| ubuntu@ip-172-31-23-175:~\$ | | | | |

Step 4: Plan Custom Image Creation

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

 Screenshot:

```

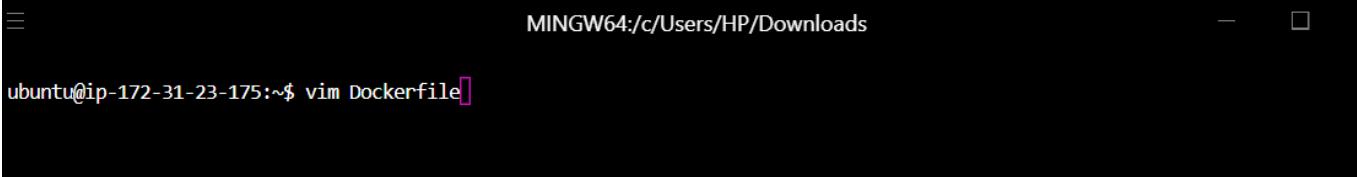
index.html ...\\Working with Docker Images U docker-compose.yaml techworld-js-docker- with Docker Images > index.html > html > body > section#skills > div.skills-grid > div.skill-category
2   <html lang="en">
56  <body>
57    <header>
58      <p>Cybersecurity, DevOps & DevSecOps Engineer</p>
59      <p>Email: oluwaseun.osunsola@example.com | Phone: (123) 456-7890</p>
60    </header>
61
62
63    <section id="summary">
64      <h2>Professional Summary</h2>
65      <p>Dynamic and versatile engineer with expertise in Cybersecurity, DevOps, and DevSecOps. Currently pursuing a BSc in Computer Science at Afe Babalola University, Nigeria. Passionate about integrating secure development lifecycles with efficient operations to protect and optimize digital infrastructures.</p>
66    </section>
67
68    <section id="education">
69      <h2>Education</h2>
70      <ul>
71        <li><strong>BSc in Computer Science (In Progress)</strong></li>
72        <li><strong>BSc in Psychology</strong><br>Ekiti State University, Nigeria</li>
73      </ul>
74    </section>
75
76    <section id="skills">
77      <h2>Skillsets</h2>
78      <div class="skills-grid">
79        <div class="skill-category">
80          <h3>Cybersecurity Skills</h3>
81          <ul>
82            <li>Threat Detection & Response</li>
83            <li>Firewall Configuration (e.g., pfSense, Cisco ASA, MikroTik) </li>
84            <li>Intrusion Detection Systems (IDS) like Snort, Suricata</li>
85            <li>Ethical Hacking & Penetration Testing (using Kali Linux, Metasploit, Nmap, Wireshark)</li>
86          </ul>
87        </div>
88      </div>
89    </section>
90
91  </body>
92</html>

```

Step 5: Create Dockerfile

Open and edit the Dockerfile using Vim.

 Screenshot:



```
ubuntu@ip-172-31-23-175:~$ vim Dockerfile
```

Step 6: Dockerfile Code

Add Dockerfile instructions and save the file.

Screenshot:

```
# 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 file 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

~
~
~
```

Step 7: Create HTML File

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

Screenshot:

ubuntu@ip-172-31-23-175:~\$ nano index.html

Step 8: HTML Code

Add website content to the HTML file and save.

Screenshot:

The screenshot shows a terminal window titled "MINGW64:/c/Users/HP/Downloads". The file being edited is "index.html". The content of the file is an HTML resume. It includes sections for education (BSc in Computer Science and BSc in Psychology) and skills (Cybersecurity Skills and DevOps Skills). The resume also lists various tools and technologies used.

```
GNU nano 7.2                               index.html *
```

```
<h2>Education</h2>
<ul>
    <li><strong>BSc in Computer Science (In Progress)</strong><br>Afe Babalola University, Ado-Ekiti, Nigeria<br>E>
        <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">
    
```

File Name to Write: index.html

File Operations:

- ^G Help
- M-D DOS Format
- M-A Append
- M-B Backup File
- ^C Cancel
- M-M Mac Format
- M-P Prepend
- ^T Browse

Step 9: Build Docker Image

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

Screenshot:

```
ubuntu@ip-172-31-23-175:~$ docker build -t dockerfile .
[+] Building 4.1s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 409B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load .dockerrcignore
=> => transferring context: 2B
=> [1/3] FROM docker.io/library/nginx:latest@sha256:fb01117203ff38c2f9af91db1a7409459182a37c87cced5cb442d1d8fcc66d 3.0s
=> => resolve docker.io/library/nginx:latest@sha256:fb01117203ff38c2f9af91db1a7409459182a37c87cced5cb442d1d8fcc66d 0.0s
=> => sha256:114e699da838b7a4a5dd75807233341d8f9a392ee2360d4bfe2b0680df4965f8 1.21kB / 1.21kB 0.0s
=> => sha256:b5bfa0b64d74b2ce9915f89c1be1b98a3400a7ca4fb4654cec353db54342c2a9 1.40kB / 1.40kB 0.1s
=> => sha256:9ee60c6c0558552ff0a2548f4b6941e4aa276937fb1cd8c76a94e73fe75d69ba 402B / 402B 0.1s
=> => sha256:7382b41547b8efa59d4103ac9610d7e359eb989e675faf7e0d3e7445496bba94 953B / 953B 0.1s
=> => sha256:ee3a09d2248a2df379f9868e31e578994110589e1f118a98396aeecbcc9316b8d 628B / 628B 0.0s
=> => sha256:5b219a92f92aeb674b0b29811b447d120ce69b41d21cfca56f90ad323aff91a2 29.99MB / 29.99MB 0.6s
=> => sha256:1733a4cd59540b3470ff7a90963bcdea5b543279dd6bdaf022d7883fdad221e5 29.78MB / 29.78MB 0.6s
=> => extracting sha256:1733a4cd59540b3470ff7a90963bcdea5b543279dd6bdaf022d7883fdad221e5 1.1s
=> => extracting sha256:5b219a92f92aeb674b0b29811b447d120ce69b41d21cfca56f90ad323aff91a2 0.8s
=> => extracting sha256:ee3a09d2248a2df379f9868e31e578994110589e1f118a98396aeecbcc9316b8d 0.0s
=> => extracting sha256:7382b41547b8efa59d4103ac9610d7e359eb989e675faf7e0d3e7445496bba94 0.0s
=> => extracting sha256:9ee60c6c0558552ff0a2548f4b6941e4aa276937fb1cd8c76a94e73fe75d69ba 0.0s
=> => extracting sha256:114e699da838b7a4a5dd75807233341d8f9a392ee2360d4bfe2b0680df4965f8 0.0s
=> => extracting sha256:b5bfa0b64d74b2ce9915f89c1be1b98a3400a7ca4fb4654cec353db54342c2a9 0.0s
=> [internal] load build context
=> => transferring context: 5.83kB
=> [2/3] WORKDIR /usr/share/nginx/html/
=> [3/3] COPY index.html /usr/share/nginx/html/
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:95df5c4a4c8100f43f96ea83b44e8affda0e9db9fd5b096fa7fac19104c5217a 0.0s
=> => exporting config sha256:ab4ec0dde54fd57c4759d64761ac15fd4aa3e43fd67acb56ae751dd993b450 0.0s
=> => exporting attestation manifest sha256:fd3057473b0a4db37080a15d7bb9dee13191fc61dcce88b4e997c7bb9a758dd4 0.0s
=> => exporting manifest list sha256:67fbacbd123b04d929c633a945e1ad8894ac9d0095a080b8d47af2e1b1b35a7c 0.0s
=> => naming to docker.io/library/dockerfile:latest
=> => unpacking to docker.io/library/dockerfile:latest 0.0s
ubuntu@ip-172-31-23-175:~$
```

Step 10: Confirm Image Creation

List Docker images to verify successful creation.

Screenshot:

| IMAGE | ID | DT | USAGE | CONTENT SIZE | EXTRA |
|--------------------------|--------------|----|-------|--------------|-------|
| dockerfile:latest | 67fbacbd123b | | 225MB | 59.8MB | |
| ubuntu:latest | c35e29c94501 | | 119MB | 31.7MB | |

```
ubuntu@ip-172-31-23-175:~$ docker images
          Info →   In Use
IMAGE           ID      DT    USAGE   CONTENT SIZE  EXTRA
dockerfile:latest  67fbacbd123b      225MB   59.8MB
ubuntu:latest   c35e29c94501      119MB   31.7MB
ubuntu@ip-172-31-23-175:~$
```

Step 11: Run Container

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

Screenshot:

```
ubuntu@ip-172-31-23-175:~$ docker run -p 8080:80 dockerfile
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/12/18 05:17:47 [notice] 1#1: using the "epoll" event method
2025/12/18 05:17:47 [notice] 1#1: nginx/1.29.4
2025/12/18 05:17:47 [notice] 1#1: built by gcc 4.2.0 (Debian 4.2.0-19)
2025/12/18 05:17:47 [notice] 1#1: OS: Linux 6.14.0-1015-aws
2025/12/18 05:17:47 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1024:524288
2025/12/18 05:17:47 [notice] 1#1: start worker processes
2025/12/18 05:17:47 [notice] 1#1: start worker process 29
2025/12/18 05:17:47 [notice] 1#1: start worker process 30
```

Step 12: Access EC2 Security Settings

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

Screenshot:

| Instance summary for i-06ab5bdd7a3217c74 (Ubuntu-Server) | |
|--|---|
| Public IPv4 address | 18.234.175.240 open address |
| Instance state | Running |
| Private IP DNS name (IPv4 only) | ip-172-31-23-175.ec2.internal |
| Instance type | t3.micro |
| VPC ID | vpc-0ba025afc1daa7b72 |
| Subnet ID | subnet-0e77300843e8fa8d |
| Instance ARN | arn:aws:ec2:us-east-1:832959958705:instance/i-06ab5bdd7a3217c74 |
| Owner ID | 832959958705 |
| Launch time | Thu Dec 18 2025 05:38:18 GMT+0100 (West Africa Standard Time) |

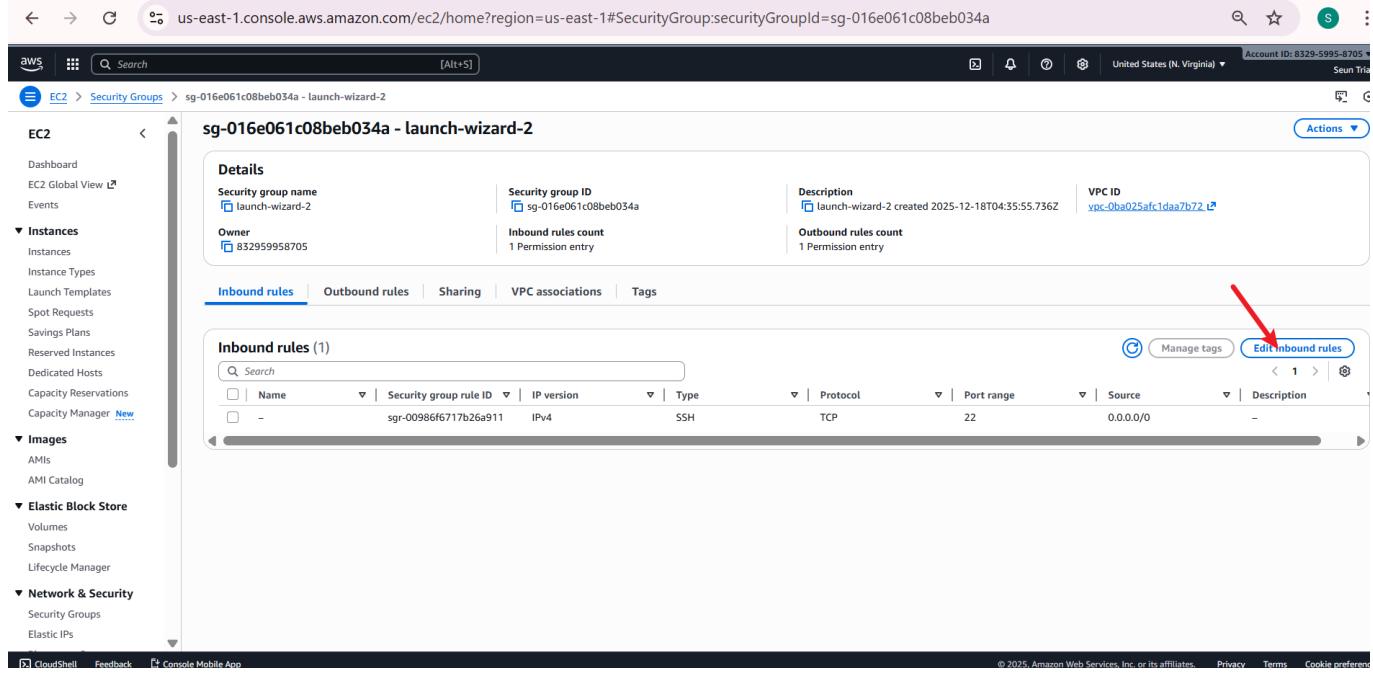
Security details

- IAM Role
- Security groups
 - sg-016e061c08beb034a (launch-wizard-2)

Step 13: Edit Inbound Rules

Edit inbound rules of the EC2 security group.

 Screenshot:

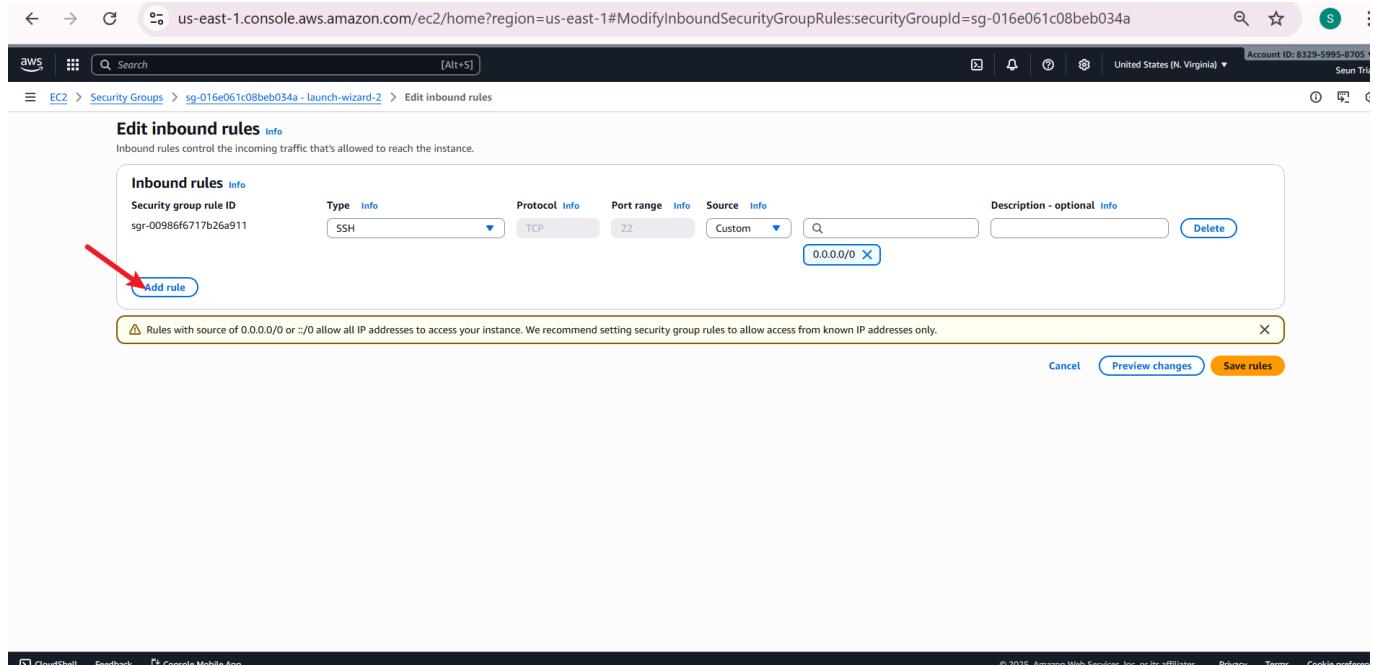


The screenshot shows the AWS EC2 Security Groups console. The left sidebar is collapsed. The main area displays the details of the security group 'sg-016e061c08beb034a - launch-wizard-2'. The 'Inbound rules' tab is selected, showing one rule: sgr-00986f6717b26a911. The rule is for IPv4, SSH protocol, TCP port 22, and source 0.0.0.0/0. A red arrow points to the 'Edit inbound rules' button at the top right of the table.

Step 14: Add New Rule

Add a new inbound rule.

 Screenshot:



The screenshot shows the 'Edit inbound rules' page for the 'sg-016e061c08beb034a - launch-wizard-2' security group. The 'Add rule' button is highlighted with a red arrow. A warning message at the bottom states: '⚠️ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' A red arrow also points to this message. The page includes tabs for Type, Protocol, Port range, Source, and Description.

Step 15: Allow Port 8080

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

Screenshot:

Inbound rules info

Inbound rules control the incoming traffic that's allowed to reach the instance.

| Type | Protocol | Port range | Source | Description - optional |
|------------|----------|------------|-------------|------------------------|
| SSH | TCP | 22 | Custom | 0.0.0.0/0 |
| Custom TCP | TCP | 8080 | Anywhere... | 0.0.0.0/0 |

Add rule

⚠️ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Preview changes Save rules

Step 16: Confirm Rule Update

Inbound rule successfully modified.

Screenshot:

Inbound security group rules successfully modified on security group (sg-016e061c08beb034a | launch-wizard-2)

Details

sg-016e061c08beb034a - launch-wizard-2

Inbound rules Outbound rules Sharing VPC associations Tags

Inbound rules (2)

| Name | Security group rule ID | IP version | Type | Protocol | Port range | Source | Description |
|------|------------------------|------------|------------|----------|------------|-----------|-------------|
| - | sgr-00986f6717b26a911 | IPv4 | SSH | TCP | 22 | 0.0.0.0/0 | - |
| - | sgr-02deb485523b709d5 | IPv4 | Custom TCP | TCP | 8080 | 0.0.0.0/0 | - |

Step 17: Stop & List Containers

Stop the container and list all containers.

Screenshot:

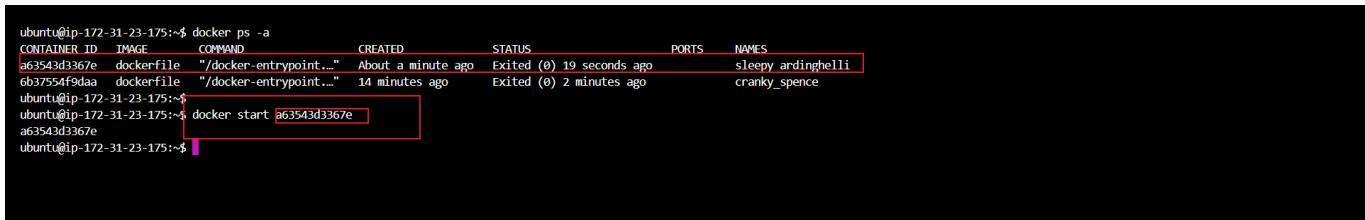


```
ubuntu@ip-172-31-23-175:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a63543d367e dockerfile "/docker-entrypoint..." About a minute ago Exited (0) 19 seconds ago
6b37554f9daa dockerfile "/docker-entrypoint..." 14 minutes ago Exited (0) 2 minutes ago
ubuntu@ip-172-31-23-175:~$
```

Step 18: Restart Container

Restart container using its container ID.

Screenshot:

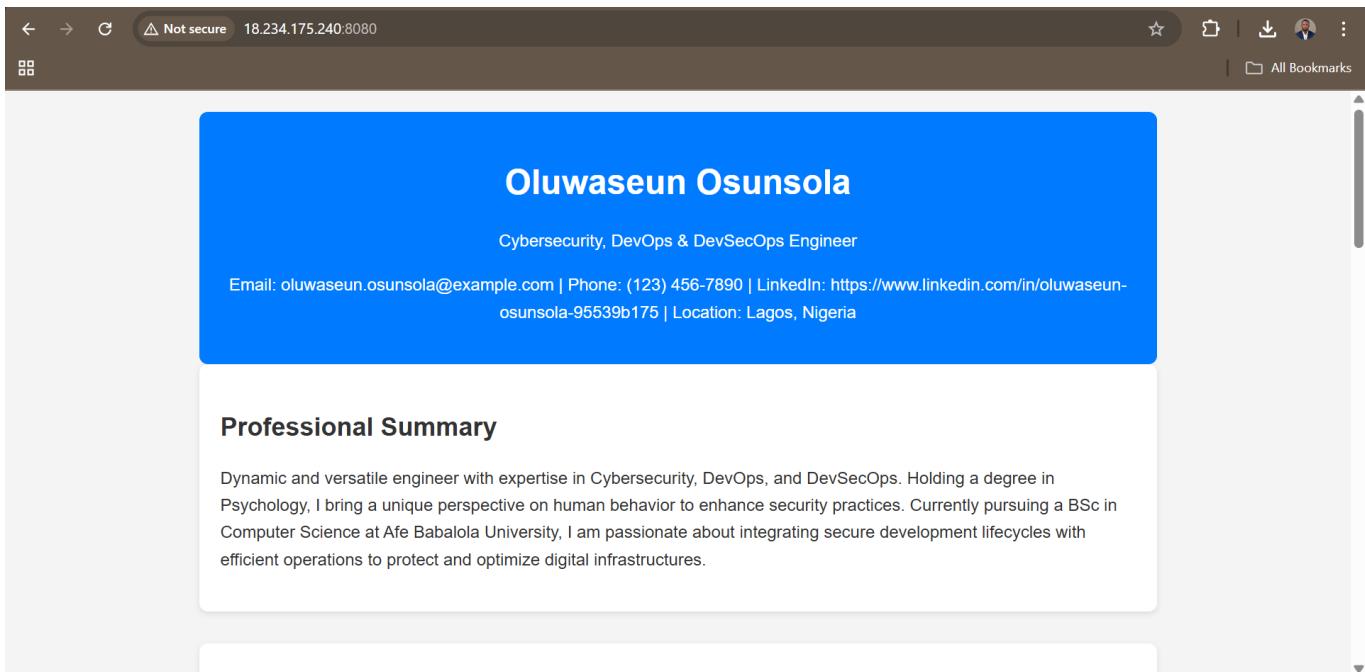


```
ubuntu@ip-172-31-23-175:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a63543d367e dockerfile "/docker-entrypoint..." About a minute ago Exited (0) 19 seconds ago
6b37554f9daa dockerfile "/docker-entrypoint..." 14 minutes ago Exited (0) 2 minutes ago
ubuntu@ip-172-31-23-175:~$ docker start a63543d367e
a63543d367e
ubuntu@ip-172-31-23-175:~$
```

Step 19: Access Live Website

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

Screenshot:

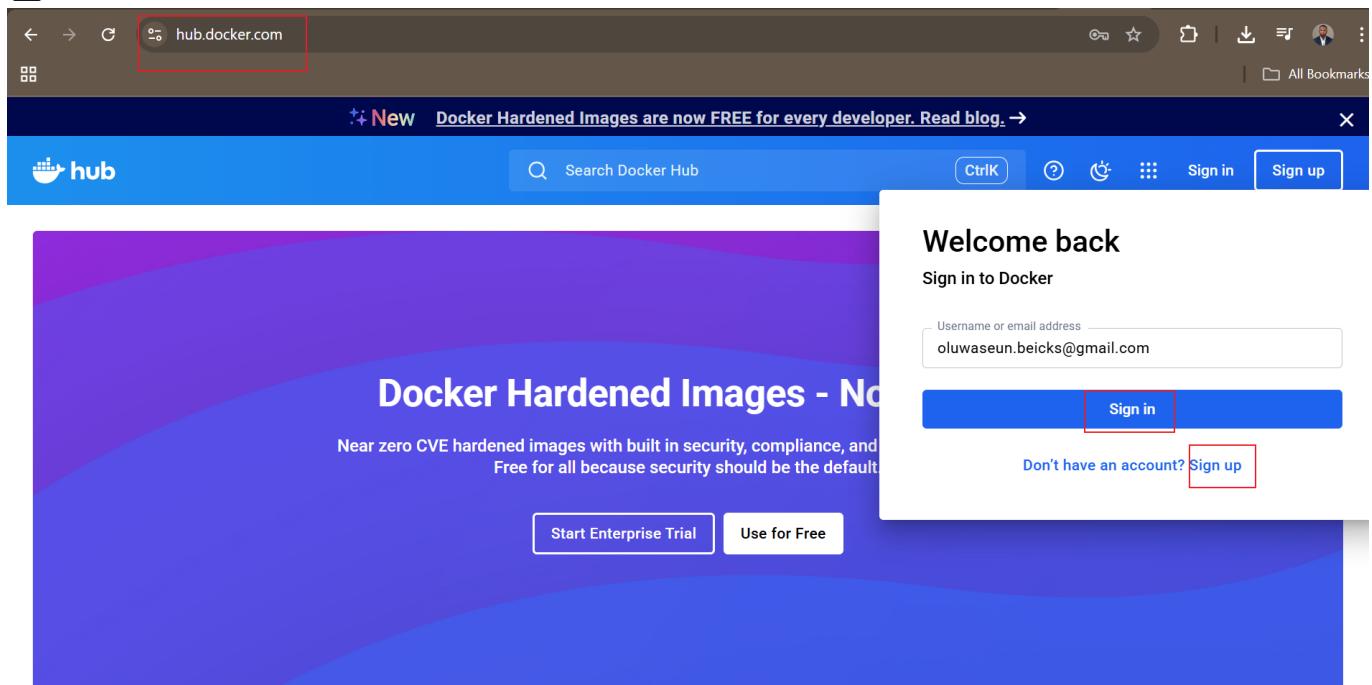


Docker Hub Integration

Step 20: Sign in to Docker Hub

Proceed to Docker Hub and sign in.

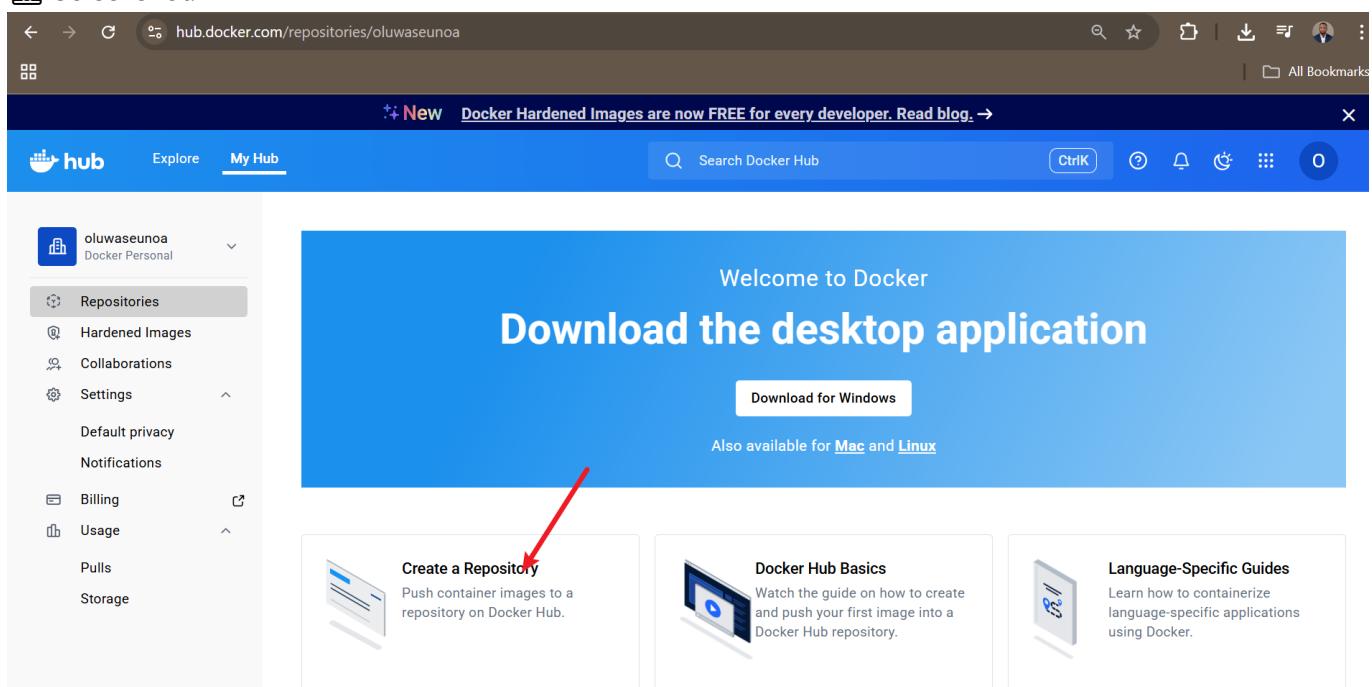
Screenshot:



Step 21: Create Repository

Click **Create Repository** after logging in.

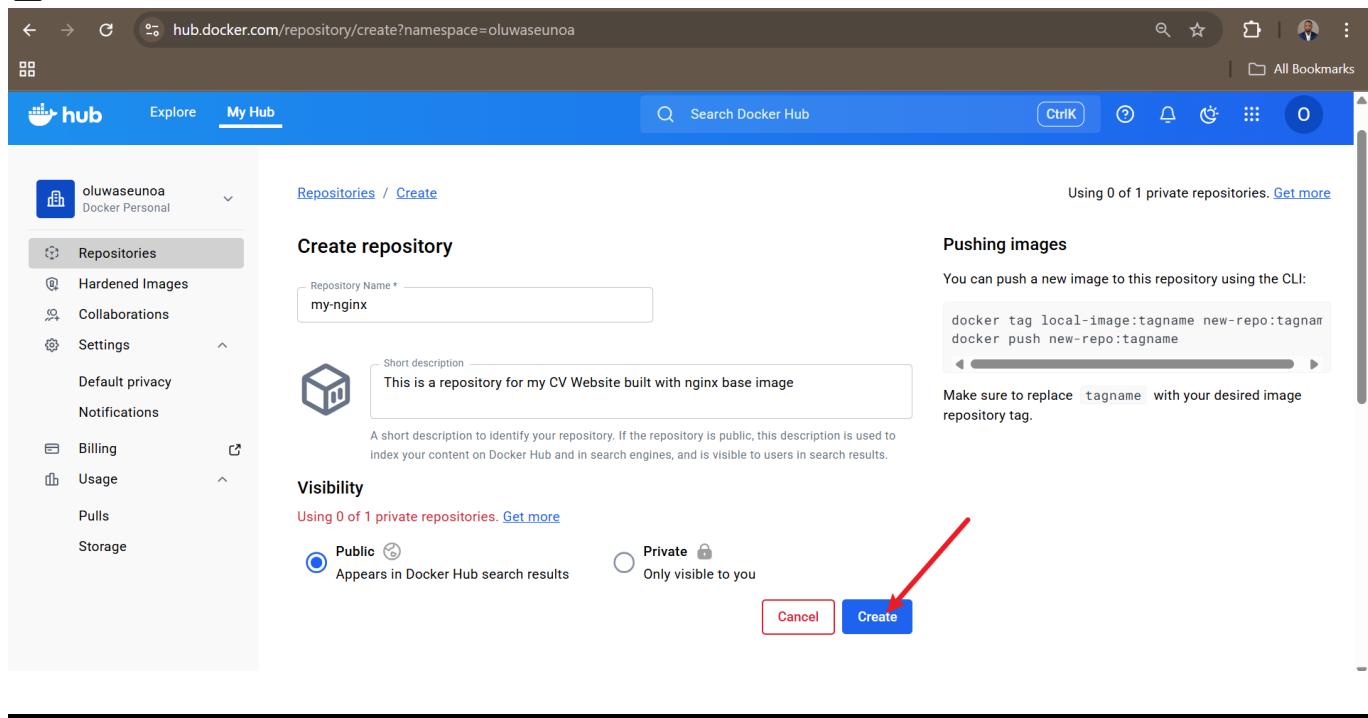
Screenshot:



Step 22: Configure Repository

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

Screenshot:

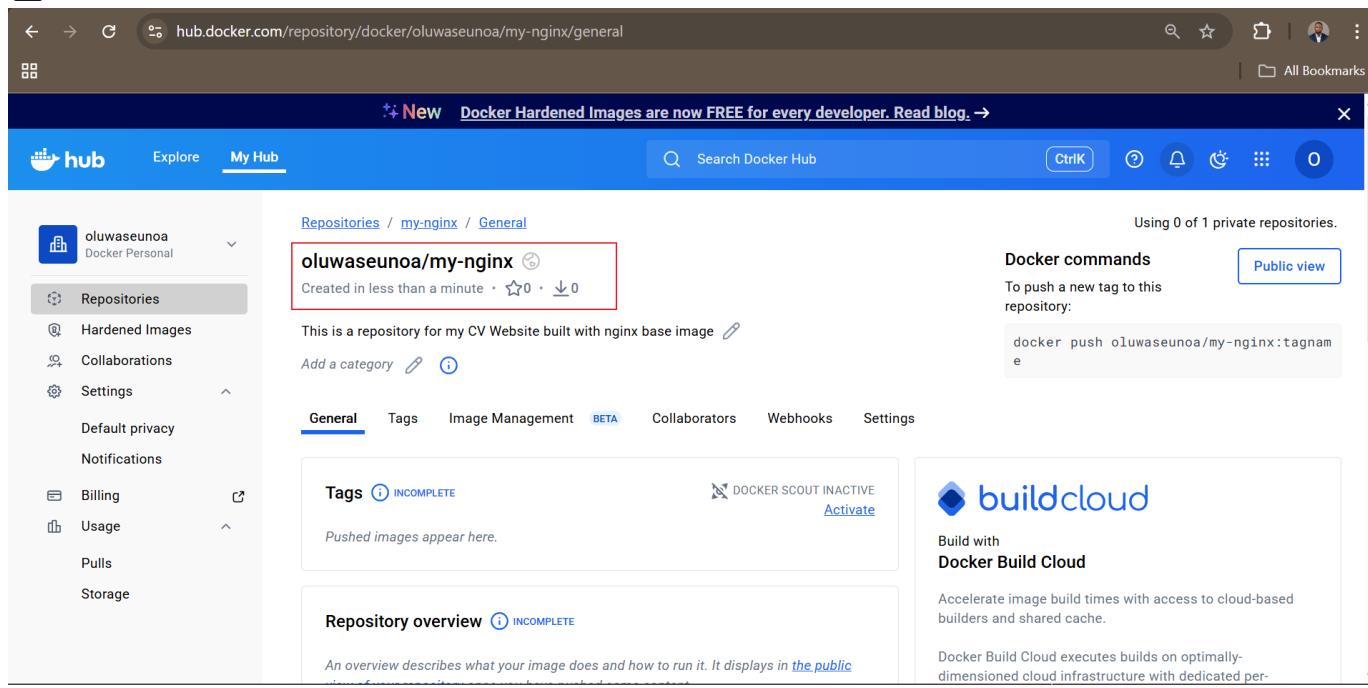


The screenshot shows the Docker Hub 'Create repository' interface. On the left, there's a sidebar with options like 'Repositories', 'Hardened Images', 'Collaborations', 'Settings', 'Default privacy', 'Notifications', 'Billing', 'Usage', 'Pulls', and 'Storage'. The main area has a 'Create repository' form. It includes fields for 'Repository Name' (set to 'my-nginx'), 'Short description' (set to 'This is a repository for my CV Website built with nginx base image'), and 'Visibility' (set to 'Public'). A red arrow points to the blue 'Create' button at the bottom right of the form.

Step 23: Repository Created

Docker Hub repository created successfully.

Screenshot:



The screenshot shows the Docker Hub repository details page for 'oluwaseunoa/my-nginx'. The top navigation bar includes links for 'Explore' and 'My Hub'. The main content area shows the repository name 'oluwaseunoa/my-nginx' with a note that it was 'Created in less than a minute'. Below this is a description: 'This is a repository for my CV Website built with nginx base image'. There are tabs for 'General', 'Tags', 'Image Management (BETA)', 'Collaborators', 'Webhooks', and 'Settings'. The 'Tags' tab is active, showing 'INCOMPLETE'. To the right, there's a 'Docker commands' section with a 'Public view' link and a command line snippet: 'docker push oluwaseunoa/my-nginx:tagname'. A 'buildcloud' advertisement is also present.

Step 24: Tag Docker Image

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

Screenshot:

The screenshot shows a terminal window titled "MINGW64:/c/Users/HP/Downloads". The terminal output is as follows:

```
ubuntu@ip-172-31-23-175:~$ docker tag dockerfile oluwaseunoa/my-nginx:1.0
ubuntu@ip-172-31-23-175:~$ docker images


| IMAGE                               | ID           | DISK USAGE | CONTENT SIZE | EXTRA |
|-------------------------------------|--------------|------------|--------------|-------|
| dockerfile:latest                   | 67fbacbd123b | 225MB      | 59.8MB       | U     |
| oluwaseunoa/my-cv-site/my-nginx:1.0 | 67fbacbd123b | 225MB      | 59.8MB       | U     |
| oluwaseunoa/my-nginx:1.0            | 67fbacbd123b | 225MB      | 59.8MB       | U     |
| ubuntu:latest                       | c35e29c94501 | 119MB      | 31.7MB       |       |



ubuntu@ip-172-31-23-175:~$


```

Step 25: Generate Access Token

Navigate to account settings and generate a personal access token.

Screenshot:

The screenshot shows the Docker Personal Access Tokens page. On the left is a sidebar with navigation links: Home, Hub, Build Cloud, Hardened Images, Scout (selected), Testcontainers Cloud, Docker Desktop, Settings, and Account information. The main content area has a heading 'Personal access tokens' and a sub-instruction: 'You can use a personal access token instead of a password for Docker CLI authentication. Create multiple tokens, control their scope, and delete tokens at any time. [Learn more](#)'.

| Description | Scope | Status | Source | Created | Last used | Expiration date |
|-------------------------|---------------------|--------|--------|--------------------------|--------------------------|-----------------|
| First-Known-Generati... | Read, Write, Delete | Active | Manual | May 14, 2025 at 20:14:33 | May 21, 2025 at 16:00:17 | Expired |

At the bottom right of the table, there is a blue button labeled 'Generate new token' with a red arrow pointing to it. Below the table, there are pagination controls: 'Rows per page: 10' and '1-1 of 1'.

Step 26: Configure Token

Describe token, set expiration and permissions, then generate.

Screenshot:

The screenshot shows the 'Create access token' form. The sidebar on the left is identical to the previous screenshot. The main form has a title 'Create access token' and a sub-instruction: 'A personal access token is similar to a password except you can have many tokens and revoke access to each one at any time. [Learn more](#)'.

The form fields are:

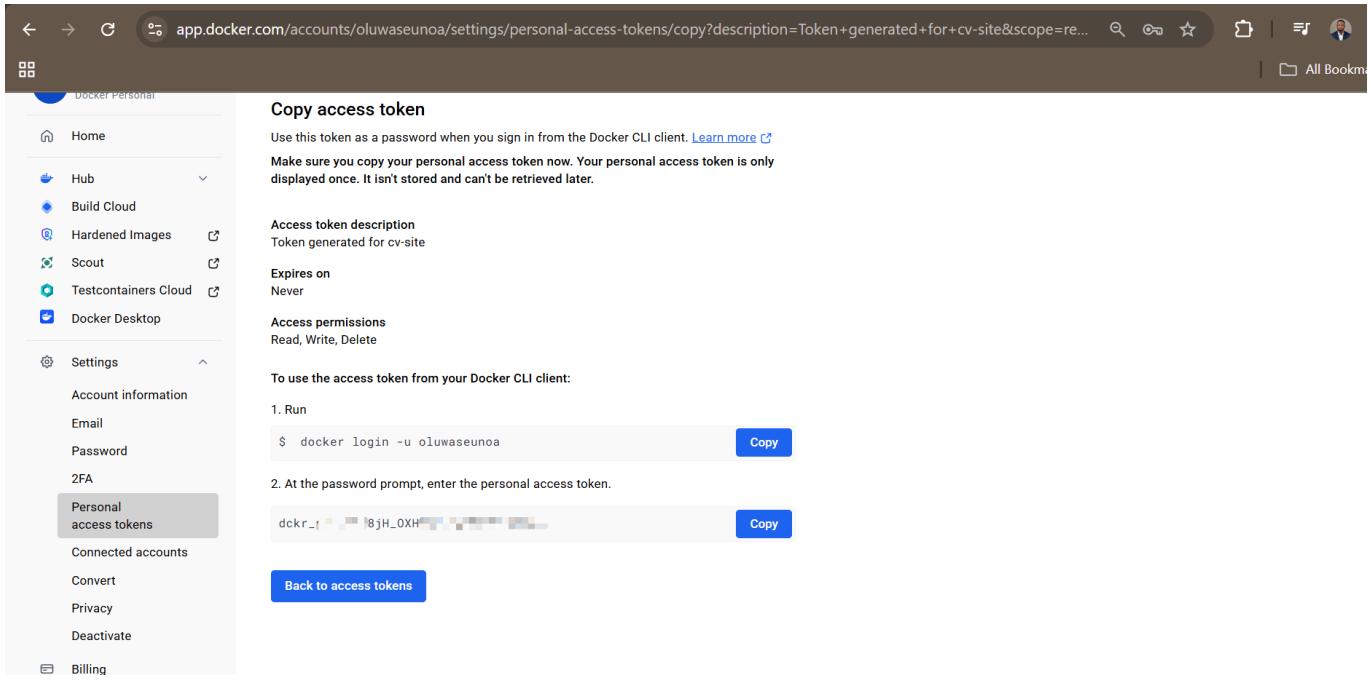
- Access token description: 'Token generated for cv-site'
- Expiration date: 'None'
- Access permissions: 'Read, Write, Delete'

Below the permissions, there is a note: 'Read, Write, Delete tokens allow you to manage your repositories.' At the bottom of the form are two buttons: 'Cancel' and 'Generate' (which is highlighted with a red arrow).

Step 27: Save Token

Save the generated token securely.

Screenshot:



The screenshot shows the Docker Personal access token generation page. The URL is app.docker.com/accounts/oluwaseunoa/settings/personal-access-tokens/copy?description=Token+generated+for+cv-site&scope=re.... The page title is "Copy access token". It displays a generated token: "Token generated for cv-site". The token has an expiration date of "Never" and access permissions of "Read, Write, Delete". Instructions for using the token with the Docker CLI are provided, including a command: "\$ docker login -u oluwaseunoa" with a "Copy" button.

Step 28: Docker Login via Terminal

Log in to Docker Hub using username and access token.

Screenshot:



```
ubuntu@ip-172-31-23-175:~$ docker login -u oluwaseunoa
Info: A Personal Access Token (PAT) can be used instead.
      To create a PAT, visit https://app.docker.com/settings

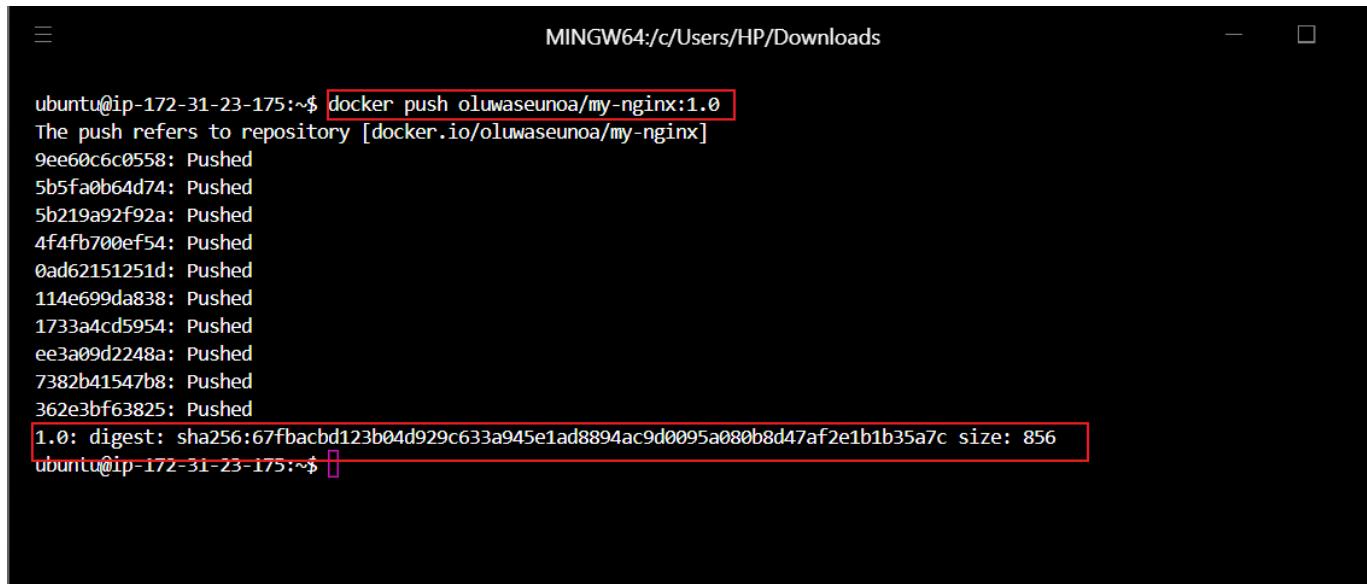
Password:
WARNING! Your credentials are stored unencrypted in '/home/ubuntu/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
ubuntu@ip-172-31-23-175:~$
```

Step 29: Push Image to Docker Hub

Push the tagged Docker image.

Screenshot:

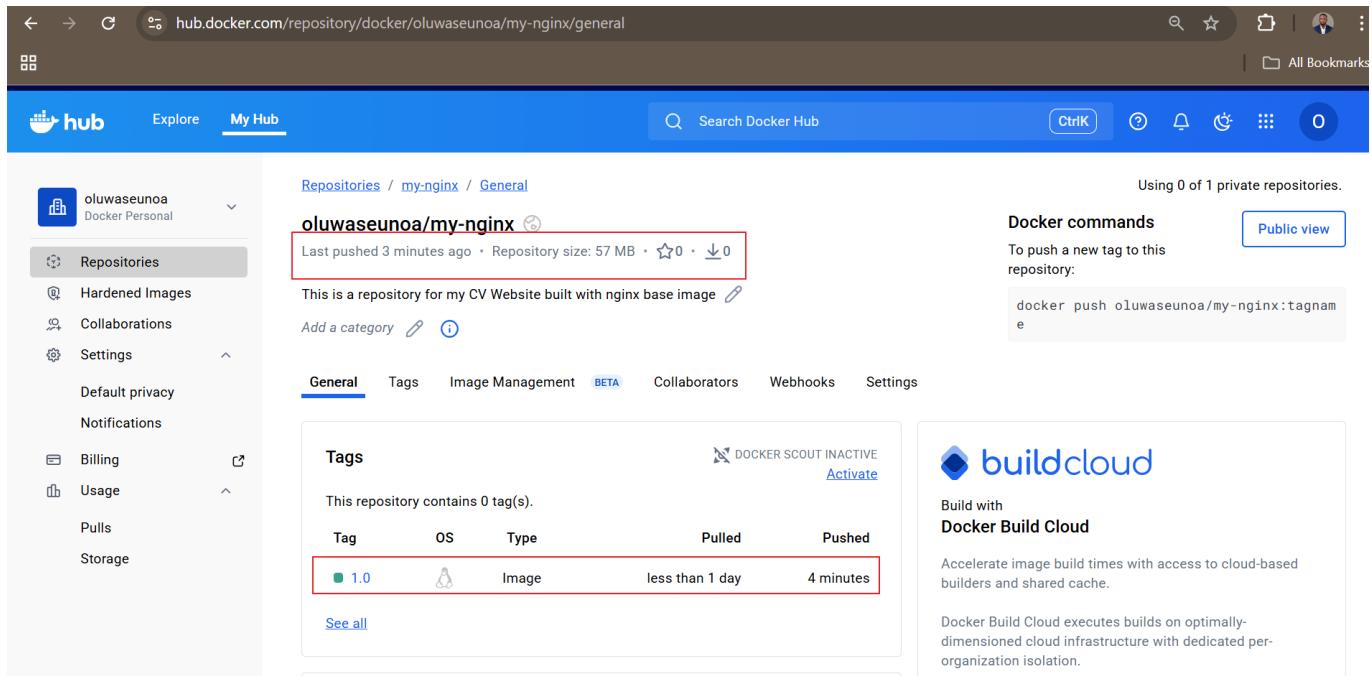


```
MINGW64:/c/Users/HP/Downloads
ubuntu@ip-172-31-23-175:~$ docker push oluwaseunoa/my-nginx:1.0
The push refers to repository [docker.io/oluwaseunoa/my-nginx]
9ee60c6c0558: Pushed
5b5fa0b64d74: Pushed
5b219a92f92a: Pushed
4f4fb700ef54: Pushed
0ad62151251d: Pushed
114e699da838: Pushed
1733a4cd5954: Pushed
ee3a09d2248a: Pushed
7382b41547b8: Pushed
362e3bf63825: Pushed
1.0: digest: sha256:67fbacbd123b04d929c633a945e1ad8894ac9d0095a080b8d47af2e1b1b35a7c size: 856
ubuntu@ip-172-31-23-175:~$
```

Step 30: Verify Image Push

Confirm successful image push on Docker Hub.

Screenshot:



hub.docker.com/repository/docker/oluwaseunoa/my-nginx/general

Repositories / my-nginx / General

oluwaseunoa/my-nginx 

Last pushed 3 minutes ago • Repository size: 57 MB • ⭐0 • ↓0

This is a repository for my CV Website built with nginx base image 

Add a category  

General Tags Image Management  Collaborators Webhooks Settings

Tags

This repository contains 0 tag(s).

| Tag | OS | Type | Pulled | Pushed |
|-----|---|-------|-----------------|-----------|
| 1.0 |  | Image | less than 1 day | 4 minutes |

[See all](#)

Docker commands

To push a new tag to this repository:

```
docker push oluwaseunoa/my-nginx:tagname
```

Using 0 of 1 private repositories.

Public view

buildcloud

Build with Docker Build Cloud

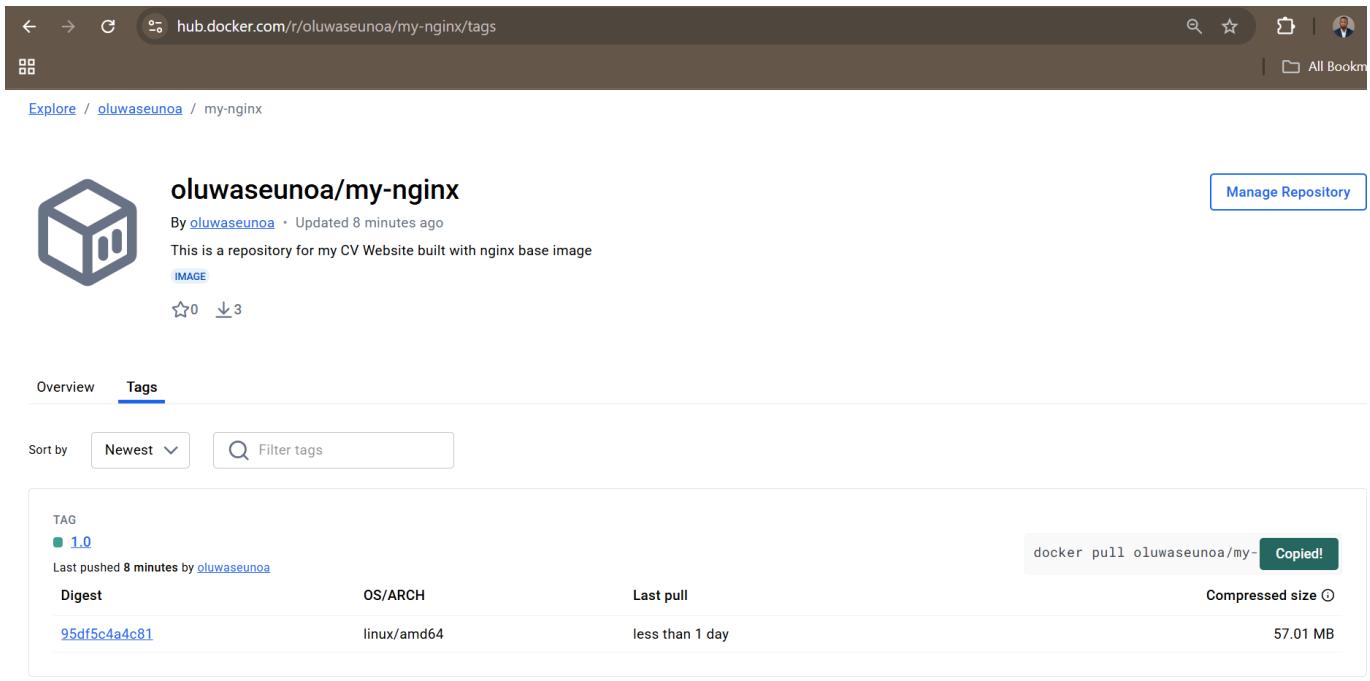
Accelerate image build times with access to cloud-based builders and shared cache.

Docker Build Cloud executes builds on optimally-dimensioned cloud infrastructure with dedicated per-organization isolation.

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 file 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>
                </ul>
            </div>
        </div>
    </section>
</body>
```

```
        <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>  
        </ul>  
    </div>  
    <div>  
        <ul>  
            <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>
```