Submitted By:

SACHIN NISHAD

A9922523002782(el)

Program: Bachelor of Computer Applications (BCA)

Project: TCS iON RIO-125: Secured Docker Based Lab: Enforcing end-to-end security

Submitted To:

TCS ION

Submission Date:

21th Sept, 2025

Contents

1. Executive Summary 3

2. Project Approach and Logic Flow 3

Phase 1: AWS Cloud Infrastructure Development (Action Item 1) 3

Phase 2: Lab Environment Creation (Action Item 2) 3

Phase 3: Vulnerability Scanning and Penetration Testing (Action Item 3) 3

- Network Mapping: 3
- Vulnerability Assessment: 4
- Penetration Testing: 4
- Credential Attacks: 4

3. Implementation and Deliverables 4

3.1 Action Item 1: AWS Cloud Infrastructure Development 4

- EC2 Instance Creation: 4
- Software Installation: 4
- IAM User & AWS CLI Configuration: 10

3.2 Action Item 2: Lab Environment Creation 10

- Docker Compose File: 11
- Dockerfiles: 12
- Deployment and Testing: 13

3.3 Action Item 3: Vulnerability Scanning and Penetration Testing 16

- Nmap Scan Results: 17
- Nessus Scan Report: 17
- Metasploit Testing: 19
- Hydra Testing: ..... **Error! Bookmark not defined.**
- Security Modifications: 20

4. Final Deliverables and Conclusion 21

GitHub Repository Link: 21

Video Link: 21

1. Executive Summary

This report documents the creation and security assessment of a cloud-based, multi-application Docker lab environment. The project was executed on an AWS EC2 instance and involved setting up environments for Python and Java, a web server, and a MySQL database. Following the infrastructure setup, a comprehensive security analysis was performed using tools such as Nmap, Nessus, Metasploit, and Hydra. The vulnerabilities discovered were then addressed by modifying the Docker environment for enhanced security. This project successfully demonstrates the ability to build, secure, and document a protected development lab.

2. Project Approach and Logic Flow

The project was completed in a logical, phased approach as outlined in the project milestones.

Phase 1: AWS Cloud Infrastructure Development (Action Item 1)

The first step was to establish the foundational infrastructure on AWS. This involved provisioning a free-tier EC2 instance with Ubuntu OS, which would serve as the host for the Docker environment. All necessary tools, including Docker, Docker Compose, and various cybersecurity tools, were installed and configured. A secure IAM user was created and configured with the AWS CLI to manage resources without using the root account, adhering to security best practices.

Phase 2: Lab Environment Creation (Action Item 2)

With the infrastructure in place, the second phase focused on building the multi-application lab. A docker-compose.yml file was created to define a multi-service environment including Python and Java applications, a web server (Nginx), and a MySQL database. These services were configured to communicate over a private Docker network. The Docker images were built and deployed, and the functionality of the applications was verified by accessing the web server from the public internet.

Phase 3: Vulnerability Scanning and Penetration Testing (Action Item 3)

The final phase was dedicated to identifying and mitigating security weaknesses. A series of tests were performed using the installed cybersecurity tools.

- Network Mapping: Nmap and Traceroute were used to discover open ports and understand network pathways.

- Vulnerability Assessment: Nessus was used to perform an automated scan for a wide range of known vulnerabilities.
- Penetration Testing: Metasploit was used to attempt to exploit discovered vulnerabilities.
- Credential Attacks: Hydra was used to test for weak passwords on exposed services.

Based on the findings from these tests, the Docker environment was modified to enhance its security posture.

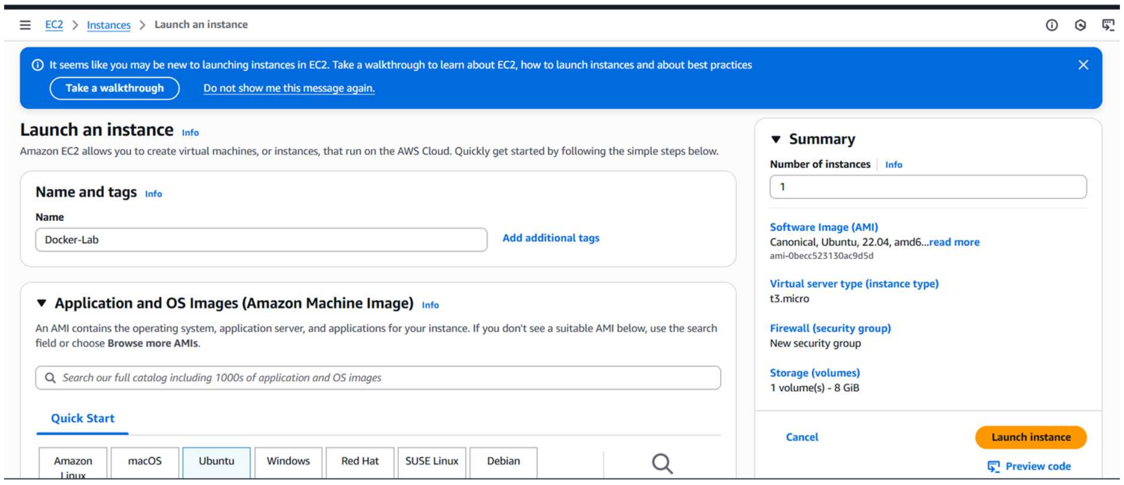
3. Implementation and Deliverables

This section contains the detailed documentation of each action item, including screenshots and code.

3.1 Action Item 1: AWS Cloud Infrastructure Development

Summary: The AWS EC2 instance was launched, configured, and secured for the project.

- EC2 Instance Creation: An Ubuntu 22.04 LTS instance (t2.micro) was launched with 8 GB of storage. The instance ID is [Your Instance ID]. A key pair (.pem) was created for secure SSH access.



EC2 > Instances > Launch an instance

It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices

Take a walkthrough · Do not show me this message again.

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: Docker-Lab · Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux · macOS · Ubuntu · Windows · Red Hat · SUSE Linux · Debian

Summary

Number of instances Info: 1

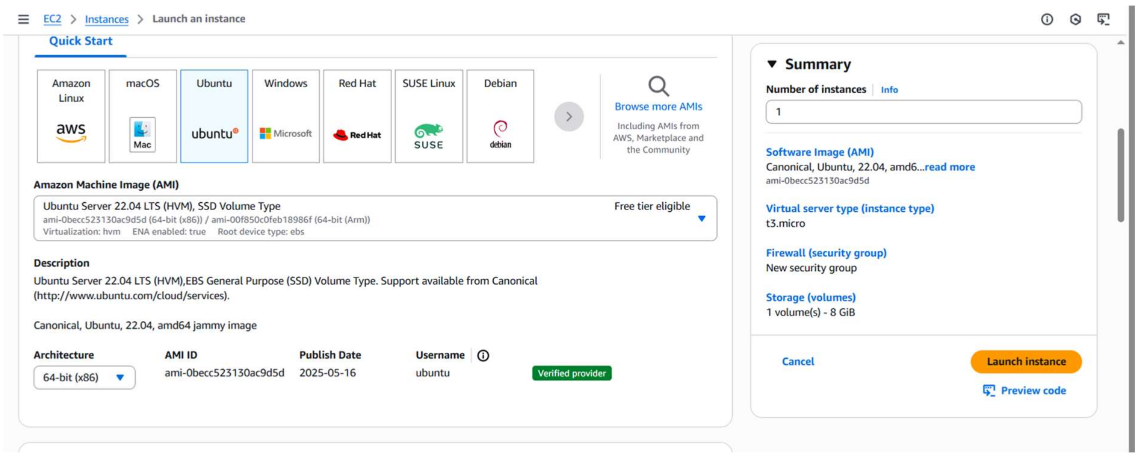
Software Image (AMI): Canonical, Ubuntu, 22.04, amd6...read more
ami-0becc523130ac9d5d

Virtual server type (instance type): t3.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Cancel · Launch instance · Preview code


EC2 > Instances > Launch an instance

Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0becc523130ac9d5d (64-bit (x86)) / ami-00f850c0feb18986f (64-bit (Arm))
Virtualization: hvm   ENA enabled: true   Root device type: ebs

Description

Ubuntu Server 22.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Canonical, Ubuntu, 22.04, amd64 jammy image

Architecture	AMI ID	Publish Date	Username
64-bit (x86)	ami-0becc523130ac9d5d	2025-05-16	ubuntu

Verified provider

Summary

Number of instances: 1

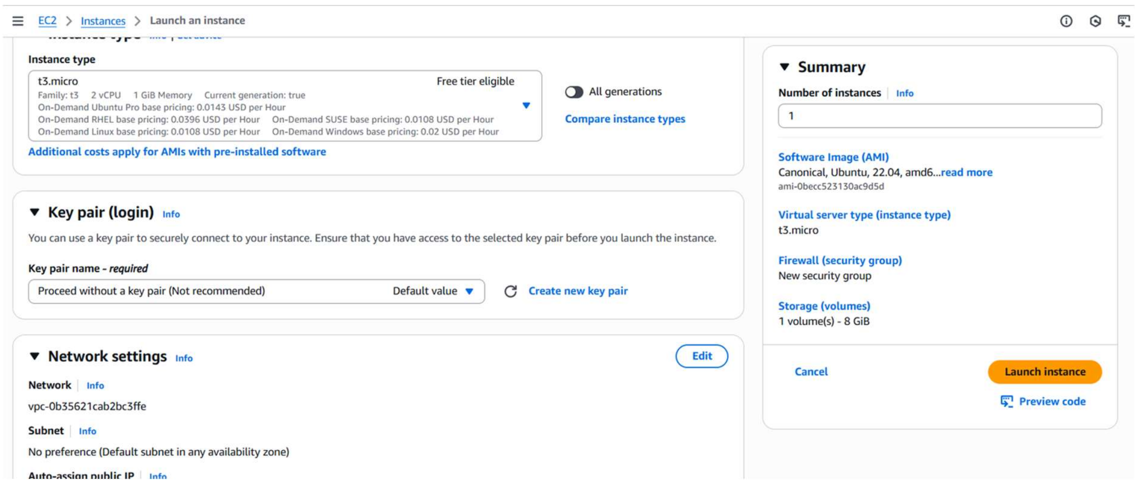
Software Image (AMI)
Canonical, Ubuntu, 22.04, amd6...read more
ami-0becc523130ac9d5d

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel | Launch instance | Preview code


EC2 > Instances > Launch an instance

Instance type

t3.micro — Free tier eligible
Family: t3   2 vCPU   1 GiB Memory   Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0143 USD per Hour
On-Demand RHEL base pricing: 0.0396 USD per Hour   On-Demand SUSE base pricing: 0.0108 USD per Hour
On-Demand Linux base pricing: 0.0108 USD per Hour   On-Demand Windows base pricing: 0.02 USD per Hour

All generations
Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*
Proceed without a key pair (Not recommended) — Default value
Create new key pair

Network settings Info — Edit

Network: vpc-0b35621cab2bc3ffe

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP

Summary

Number of instances: 1

Software Image (AMI)
Canonical, Ubuntu, 22.04, amd6...read more
ami-0becc523130ac9d5d

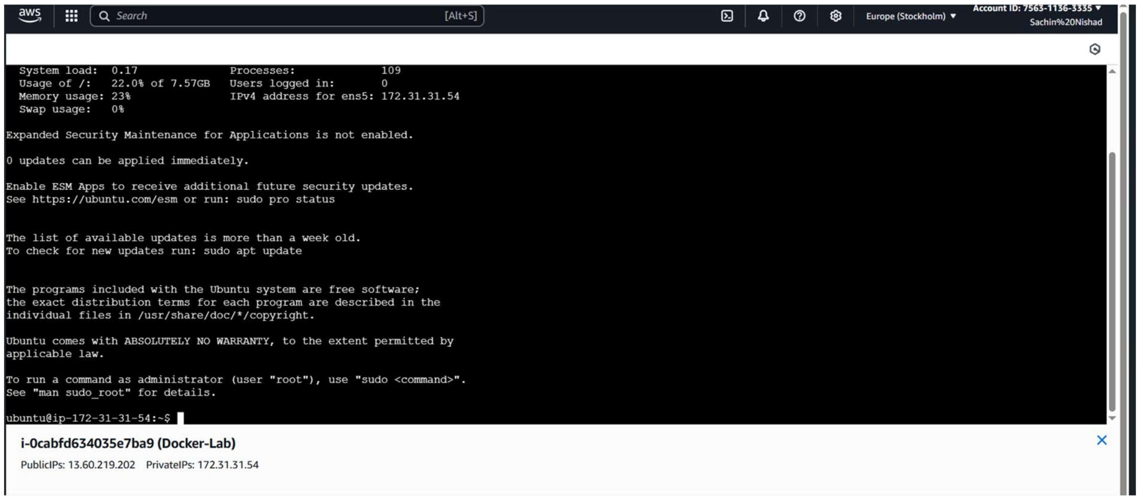
Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel | Launch instance | Preview code

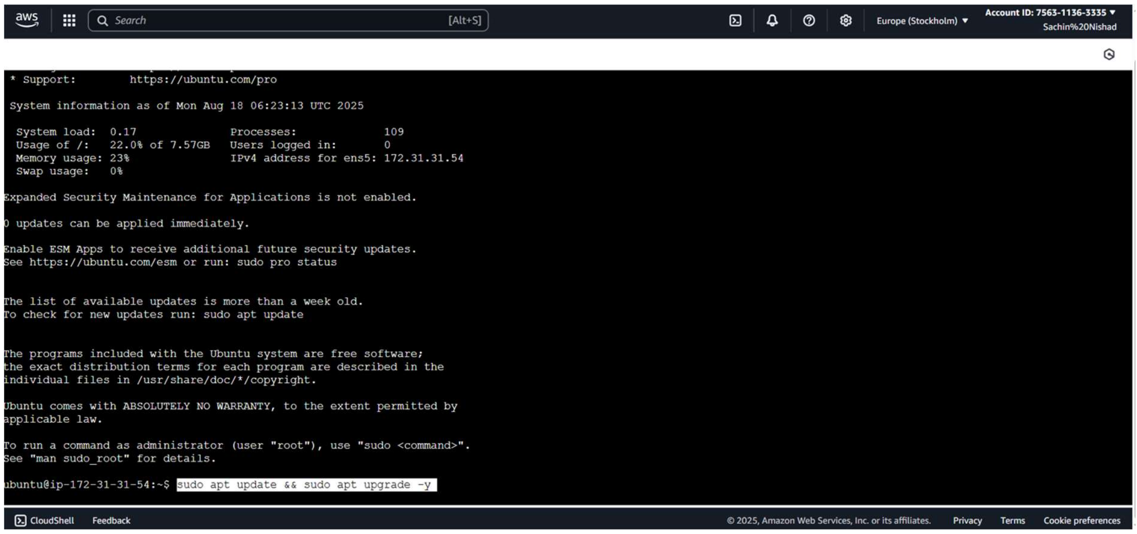
- Software Installation: Docker, Docker Compose, Nmap, Metasploit, Hydra, and John were installed on the EC2 instance via the command line.



```
  System load:  0.17              Processes:             109
  Usage of /:   22.0% of 7.57GB   Users logged in:       0
  Memory usage: 23%               IPv4 address for ens5: 172.31.31.54
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-31-54:~$
```

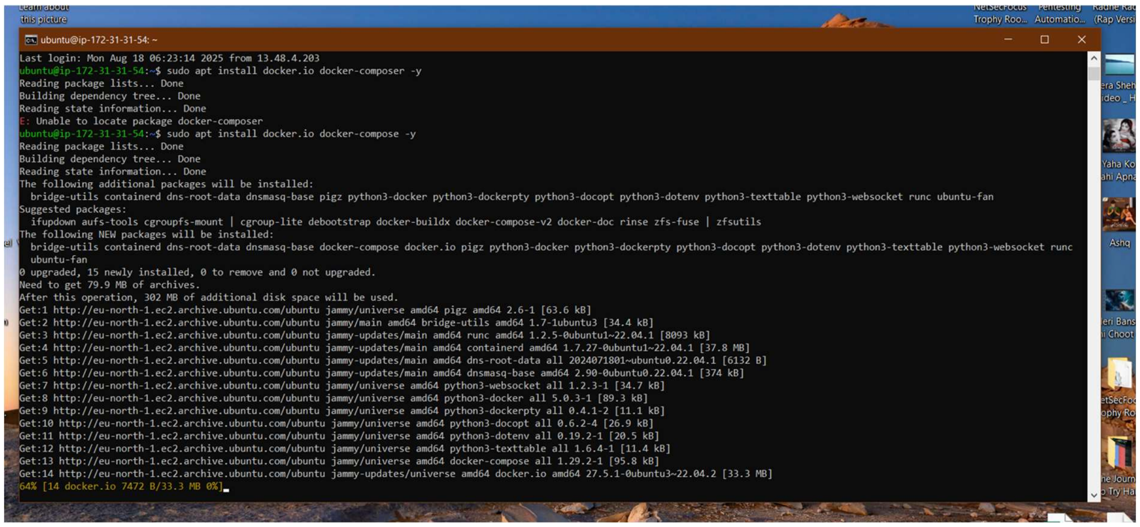
i-0cabfd634035e7ba9 (Docker-Lab)

PublicIPs: 13.60.219.202 PrivateIPs: 172.31.31.54

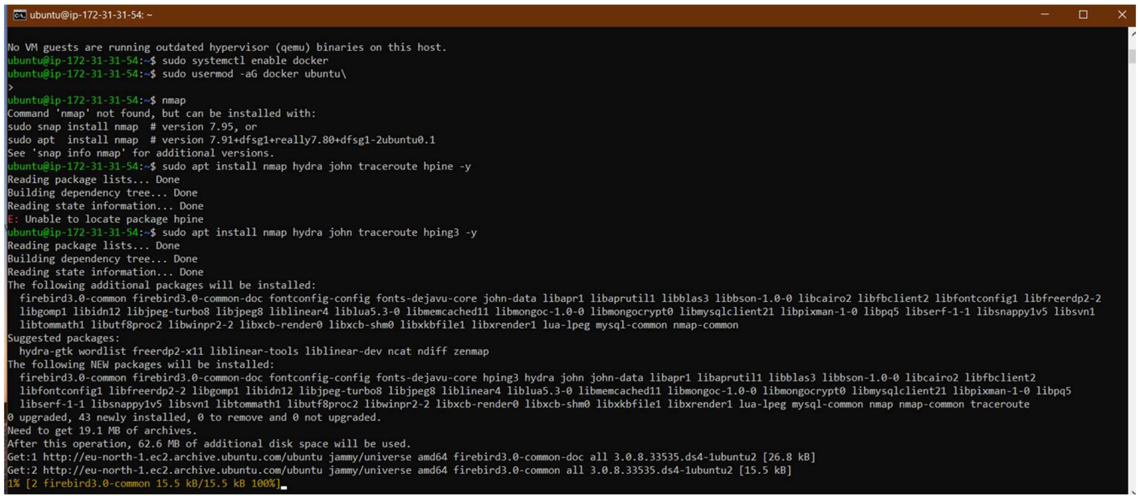


```
 * Support:        https://ubuntu.com/pro

 System information as of Mon Aug 18 06:23:13 UTC 2025

  System load:  0.17              Processes:             109
  Usage of /:   22.0% of 7.57GB   Users logged in:       0
  Memory usage: 23%               IPv4 address for ens5: 172.31.31.54
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-31-54:~$ sudo apt update && sudo apt upgrade -y
```

© 2025, Amazon Web Services, Inc. or its affiliates.

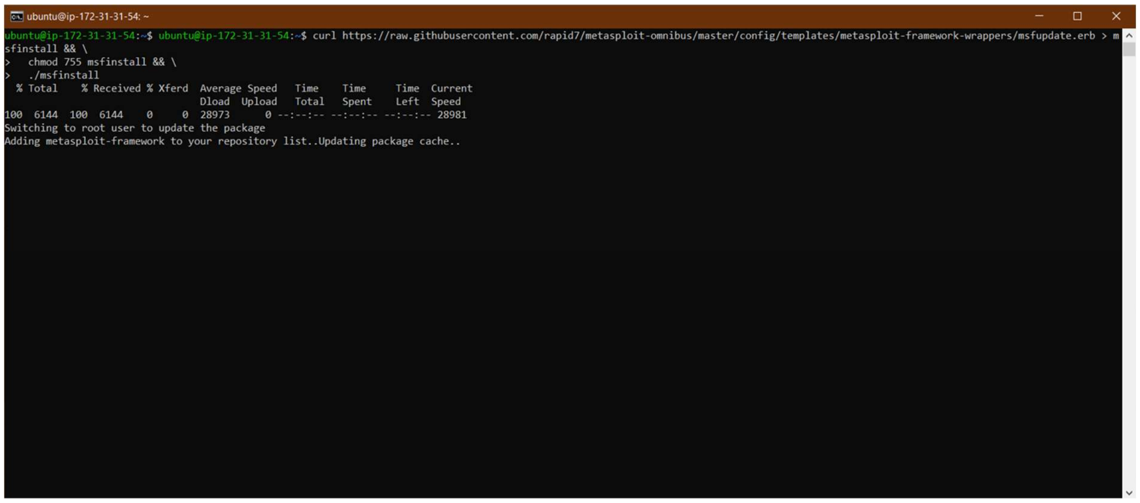


```
Last login: Mon Aug 18 06:23:14 2025 from 13.48.4.203
ubuntu@ip-172-31-31-54:~$ sudo apt install docker.io docker-composer -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package docker-composer
ubuntu@ip-172-31-31-54:~$ sudo apt install docker.io docker-compose -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz python3-docker python3-dockerpty python3-docopt python3-dotenv python3-texttable python3-websocket runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker-compose docker.io pigz python3-docker python3-dockerpty python3-docopt python3-dotenv python3-texttable python3-websocket runc
  ubuntu-fan
0 upgraded, 15 newly installed, 0 to remove and 0 not upgraded.
Need to get 79.9 MB of archives.
After this operation, 302 MB of additional disk space will be used.
Get:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]
Get:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.2.5-0ubuntu1~22.04.1 [8093 kB]
Get:4 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.7.27-0ubuntu1~22.04.1 [37.8 MB]
Get:5 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dns-root-data all 2024071801~ubuntu0.22.04.1 [6132 B]
Get:6 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dnsmasq-base amd64 2.90-0ubuntu0.22.04.1 [374 kB]
Get:7 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-websocket all 1.2.3-1 [34.7 kB]
Get:8 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-docker all 5.0.3-1 [89.3 kB]
Get:9 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-dockerpty all 0.4.1-2 [11.1 kB]
Get:10 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-docopt all 0.6.2-4 [26.9 kB]
Get:11 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-dotenv all 0.19.2-1 [20.5 kB]
Get:12 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-texttable all 1.6.4-1 [11.4 kB]
Get:13 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 docker-compose all 1.29.2-1 [95.8 kB]
Get:14 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 docker.io amd64 27.5.1-0ubuntu3~22.04.2 [33.3 MB]
64% [14 docker.io 7472 B/33.3 MB 0%]
```



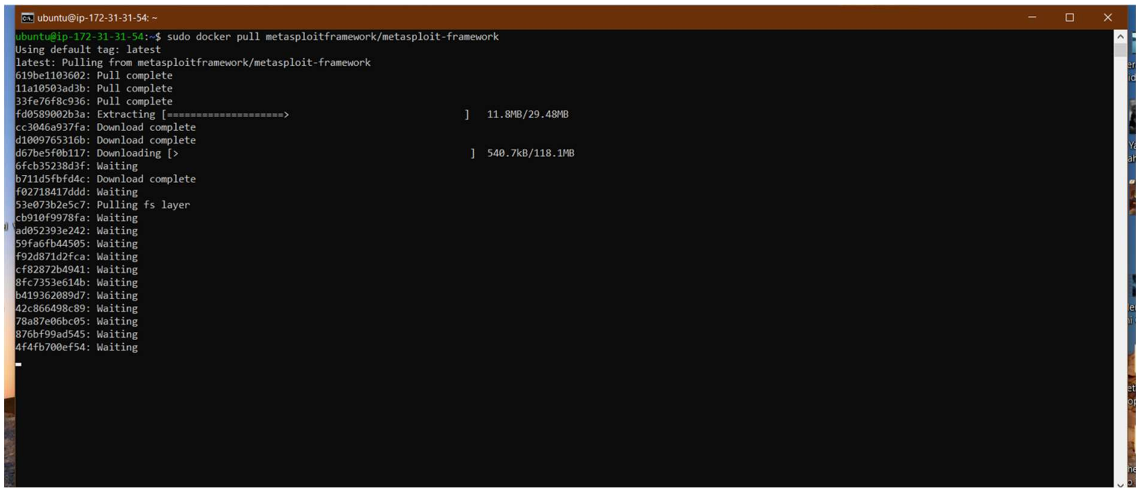
ubuntu@ip-172-31-31-54: ~

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-31-54:~$ sudo systemctl enable docker
ubuntu@ip-172-31-31-54:~$ sudo usermod -aG docker ubuntu\
>
ubuntu@ip-172-31-31-54:~$ nmap
Command 'nmap' not found, but can be installed with:
sudo snap install nmap  # version 7.95, or
sudo apt  install nmap  # version 7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1
See 'snap info nmap' for additional versions.
ubuntu@ip-172-31-31-54:~$ sudo apt install nmap hydra john traceroute hpine -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package hpine
ubuntu@ip-172-31-31-54:~$ sudo apt install nmap hydra john traceroute hping3 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  firebird3.0-common firebird3.0-common-doc fontconfig-config fonts-dejavu-core john-data libapr1 libaprutil1 libblas3 libbson-1.0-0 libcairo2 libfbclient2 libfontconfig1 libfreerdp2-2
  libgomp1 libidn12 libjpeg-turbo8 libjpeg8 liblinear4 liblua5.3-0 libmemcached11 libmongoc-1.0-0 libmongocrypt0 libmysqlclient21 libpixman-1-0 libpq5 libserf-1-1 libsnappy1v5 libsvn1
  libtommath1 libutf8proc2 libwinpr2-2 libxcb-render0 libxcb-shm0 libxkbfile1 libxrender1 lua-lpeg mysql-common nmap-common
Suggested packages:
  hydra-gtk wordlist freerdp2-x11 liblinear-tools liblinear-dev ncat ndiff zenmap
The following NEW packages will be installed:
  firebird3.0-common firebird3.0-common-doc fontconfig-config fonts-dejavu-core hping3 hydra john john-data libapr1 libaprutil1 libblas3 libbson-1.0-0 libcairo2 libfbclient2
  libfontconfig1 libfreerdp2-2 libgomp1 libidn12 libjpeg-turbo8 libjpeg8 liblinear4 liblua5.3-0 libmemcached11 libmongoc-1.0-0 libmongocrypt0 libmysqlclient21 libpixman-1-0 libpq5
  libserf-1-1 libsnappy1v5 libsvn1 libtommath1 libutf8proc2 libwinpr2-2 libxcb-render0 libxcb-shm0 libxkbfile1 libxrender1 lua-lpeg mysql-common nmap nmap-common traceroute
0 upgraded, 43 newly installed, 0 to remove and 0 not upgraded.
Need to get 19.1 MB of archives.
After this operation, 62.6 MB of additional disk space will be used.
Get:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 firebird3.0-common-doc all 3.0.8.33535.ds4-1ubuntu2 [26.8 kB]
Get:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 firebird3.0-common all 3.0.8.33535.ds4-1ubuntu2 [15.5 kB]
1% [2 firebird3.0-common 15.5 kB/15.5 kB 100%]
```



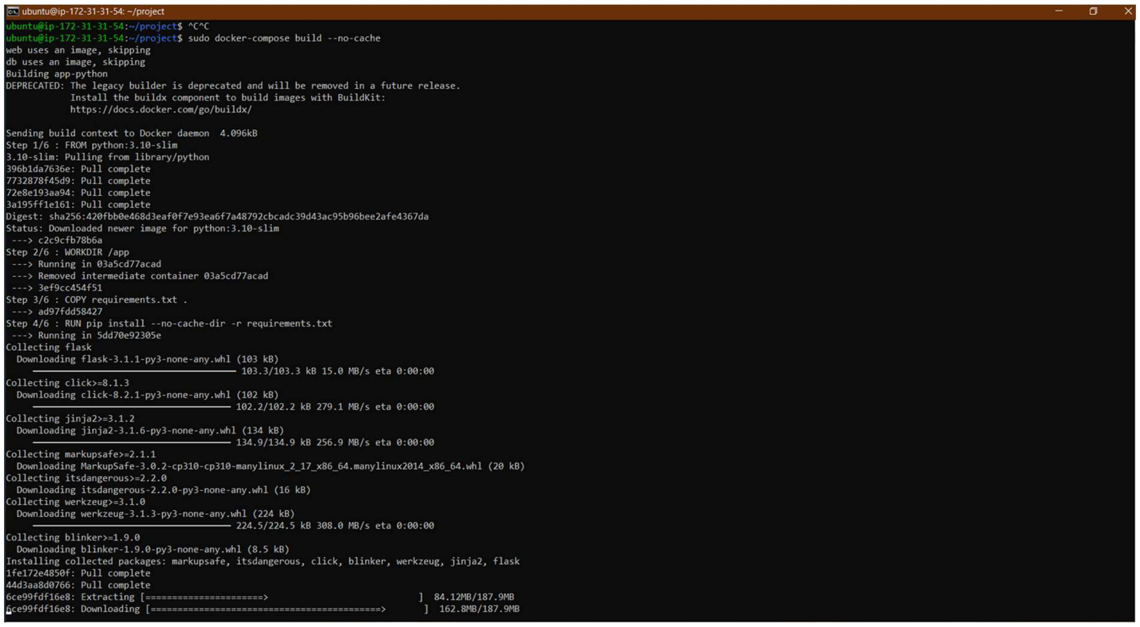
ubuntu@ip-172-31-31-54: ~

```
ubuntu@ip-172-31-31-54:~$ ubuntu@ip-172-31-31-54:~$ curl https://raw.githubusercontent.com/rapid7/metasploit-omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb > m
sfinstall && \
>   chmod 755 msfinstall && \
>   ./msfinstall
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  6144  100  6144    0     0  28973      0 --:--:-- --:--:-- --:--:-- 28981
Switching to root user to update the package
Adding metasploit-framework to your repository list..Updating package cache..
```

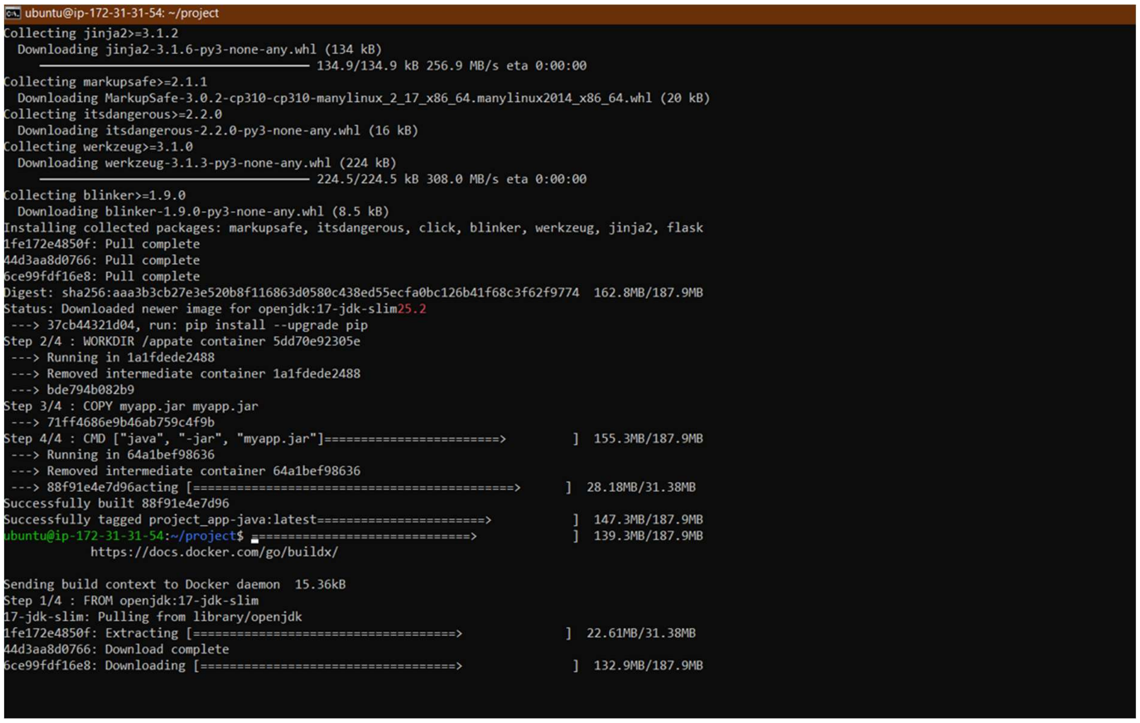


ubuntu@ip-172-31-31-54: ~

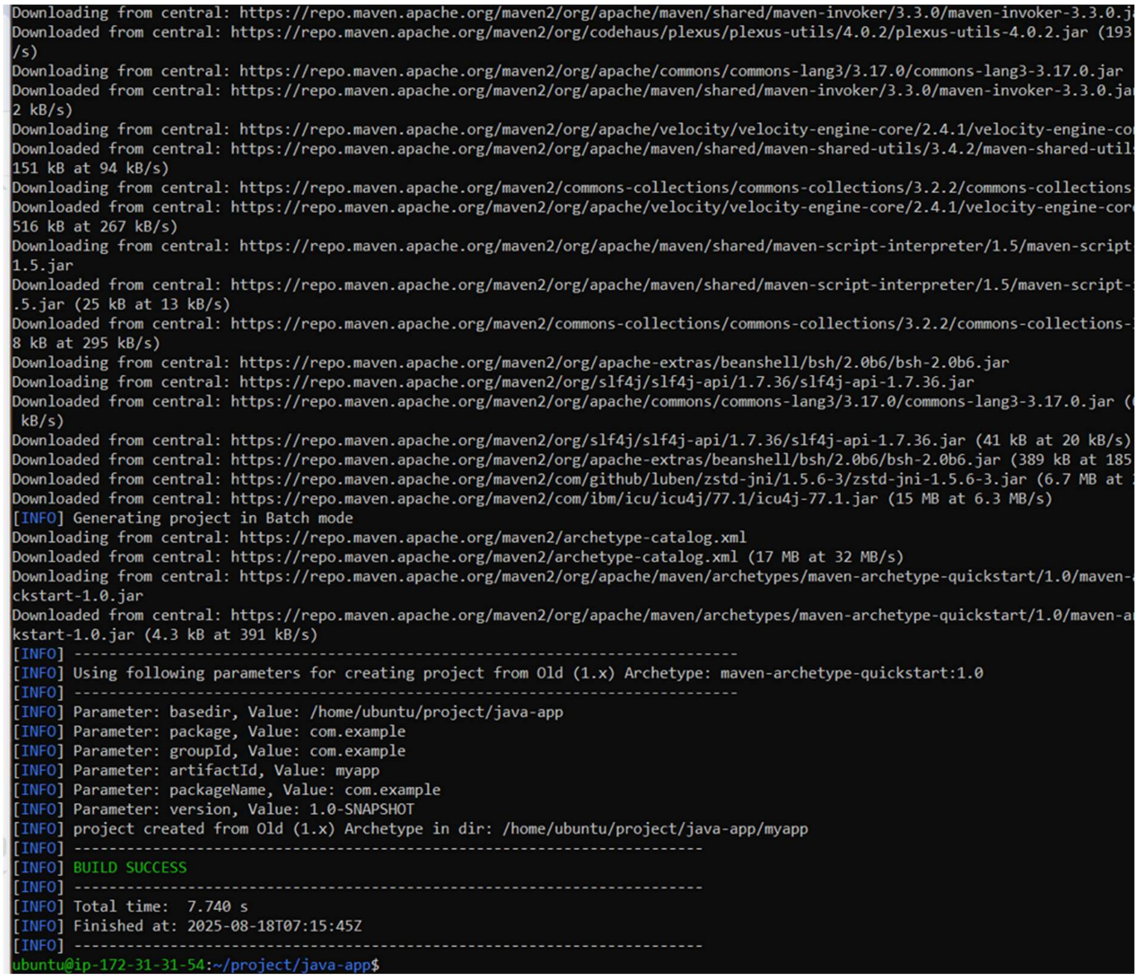
```
ubuntu@ip-172-31-31-54:~$ sudo docker pull metasploitframework/metasploit-framework
Using default tag: latest
latest: Pulling from metasploitframework/metasploit-framework
619be1103602: Pull complete
11a10503ad3b: Pull complete
33fe76f8c936: Pull complete
fd0589002b3a: Extracting [====================>                              ]   11.8MB/29.48MB
cc3046a937fa: Download complete
d1009765316b: Download complete
d67be5f0b117: Downloading [>                                                  ]  540.7kB/118.1MB
6fcb35238d3f: Waiting
b711d5fbfd4c: Download complete
f02718417ddd: Waiting
53e073b2e5c7: Pulling fs layer
cb910f9978fa: Waiting
ad052393e242: Waiting
59fa6fb44505: Waiting
f92d871d2fca: Waiting
cf82872b4941: Waiting
8fc7353e614b: Waiting
b419362089d7: Waiting
42c866498c89: Waiting
78a87e06bc05: Waiting
876bf99ad545: Waiting
4f4fb700ef54: Waiting
```



```
ubuntu@ip-172-31-31-54: ~/project
ubuntu@ip-172-31-31-54:~/project$ ^C^C
ubuntu@ip-172-31-31-54:~/project$ sudo docker-compose build --no-cache
web uses an image, skipping
db uses an image, skipping
Building app-python
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  4.096kB
Step 1/6 : FROM python:3.10-slim
3.10-slim: Pulling from library/python
396b1da7636e: Pull complete
7732878f45d9: Pull complete
72e8e193aa94: Pull complete
3a195ff1e161: Pull complete
Digest: sha256:420fbb0e468d3eaf0f7e93ea6f7a48792cbcadc39d43ac95b96bee2afe4367da
Status: Downloaded newer image for python:3.10-slim
 ---> c2c9cfb78b6a
Step 2/6 : WORKDIR /app
 ---> Running in 03a5cd77acad
 ---> Removed intermediate container 03a5cd77acad
 ---> 3ef9cc454f51
Step 3/6 : COPY requirements.txt .
 ---> ad97fdd58427
Step 4/6 : RUN pip install --no-cache-dir -r requirements.txt
 ---> Running in 5dd70e92305e
Collecting flask
  Downloading flask-3.1.1-py3-none-any.whl (103 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 103.3/103.3 kB 15.0 MB/s eta 0:00:00
Collecting click>=8.1.3
  Downloading click-8.2.1-py3-none-any.whl (102 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 102.2/102.2 kB 279.1 MB/s eta 0:00:00
Collecting jinja2>=3.1.2
  Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.9/134.9 kB 256.9 MB/s eta 0:00:00
Collecting markupsafe>=2.1.1
  Downloading MarkupSafe-3.0.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (20 kB)
Collecting itsdangerous>=2.2.0
  Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting werkzeug>=3.1.0
  Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 224.5/224.5 kB 308.0 MB/s eta 0:00:00
Collecting blinker>=1.9.0
  Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Installing collected packages: markupsafe, itsdangerous, click, blinker, werkzeug, jinja2, flask
1fe172e4850f: Pull complete
44d3aa8d0766: Pull complete
6ce99fdf16e8: Extracting [======================>                            ]  84.12MB/187.9MB
6ce99fdf16e8: Downloading [===========================================>       ]  162.8MB/187.9MB
```

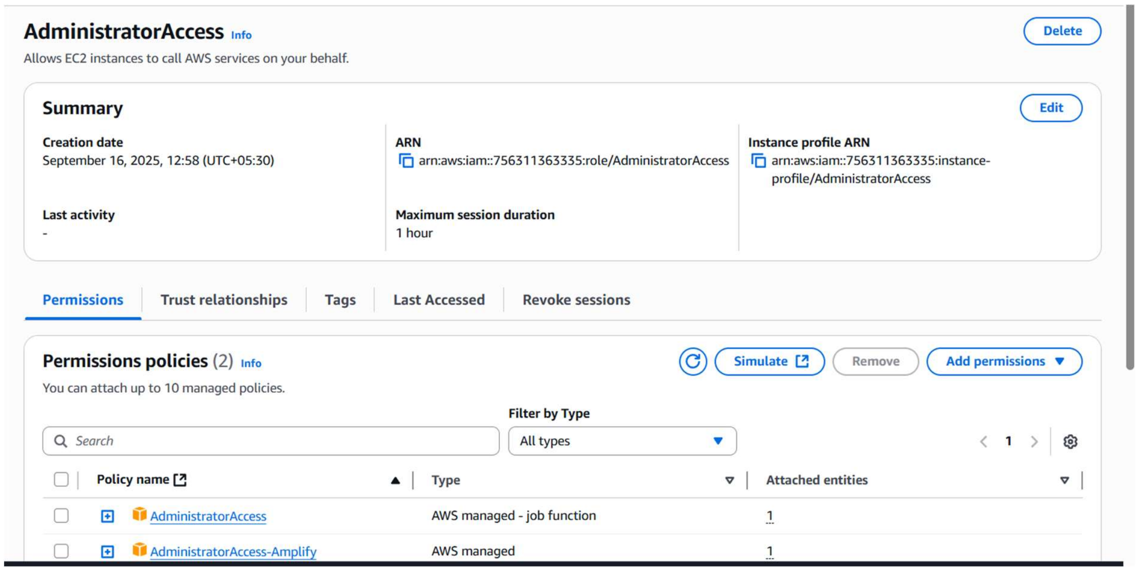


```
ubuntu@ip-172-31-31-54: ~/project
Collecting jinja2>=3.1.2
  Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.9/134.9 kB 256.9 MB/s eta 0:00:00
Collecting markupsafe>=2.1.1
  Downloading MarkupSafe-3.0.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (20 kB)
Collecting itsdangerous>=2.2.0
  Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting werkzeug>=3.1.0
  Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 224.5/224.5 kB 308.0 MB/s eta 0:00:00
Collecting blinker>=1.9.0
  Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Installing collected packages: markupsafe, itsdangerous, click, blinker, werkzeug, jinja2, flask
1fe172e4850f: Pull complete
44d3aa8d0766: Pull complete
6ce99fdf16e8: Pull complete
Digest: sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f68c3f62f9774  162.8MB/187.9MB
Status: Downloaded newer image for openjdk:17-jdk-slim25.2
 ---> 37cb44321d04, run: pip install --upgrade pip
Step 2/4 : WORKDIR /appate container 5dd70e92305e
 ---> Running in 1a1fdede2488
 ---> Removed intermediate container 1a1fdede2488
 ---> bde794b082b9
Step 3/4 : COPY myapp.jar myapp.jar
 ---> 71ff4686e9b46ab759c4f9b
Step 4/4 : CMD ["java", "-jar", "myapp.jar"]=======================>         ]  155.3MB/187.9MB
 ---> Running in 64a1bef98636
 ---> Removed intermediate container 64a1bef98636
 ---> 88f91e4e7d96acting [===========================================>      ]  28.18MB/31.38MB
Successfully built 88f91e4e7d96
Successfully tagged project_app-java:latest=======================>           ]  147.3MB/187.9MB
ubuntu@ip-172-31-31-54:~/project$ ============================>              ]  139.3MB/187.9MB
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  15.36kB
Step 1/4 : FROM openjdk:17-jdk-slim
17-jdk-slim: Pulling from library/openjdk
1fe172e4850f: Extracting [==================================>              ]  22.61MB/31.38MB
44d3aa8d0766: Download complete
6ce99fdf16e8: Downloading [==================================>              ]  132.9MB/187.9MB
```



```
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-invoker/3.3.0/maven-invoker-3.3.0.j
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/4.0.2/plexus-utils-4.0.2.jar (193
/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-lang3/3.17.0/commons-lang3-3.17.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-invoker/3.3.0/maven-invoker-3.3.0.ja
2 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/velocity/velocity-engine-core/2.4.1/velocity-engine-co
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/3.4.2/maven-shared-util
151 kB at 94 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/commons-collections/commons-collections/3.2.2/commons-collections
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/velocity/velocity-engine-core/2.4.1/velocity-engine-cor
516 kB at 267 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-script-interpreter/1.5/maven-script
1.5.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-script-interpreter/1.5/maven-script-
.5.jar (25 kB at 13 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/commons-collections/commons-collections/3.2.2/commons-collections-
8 kB at 295 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache-extras/beanshell/bsh/2.0b6/bsh-2.0b6.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-api/1.7.36/slf4j-api-1.7.36.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-lang3/3.17.0/commons-lang3-3.17.0.jar (
 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-api/1.7.36/slf4j-api-1.7.36.jar (41 kB at 20 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache-extras/beanshell/bsh/2.0b6/bsh-2.0b6.jar (389 kB at 185
Downloaded from central: https://repo.maven.apache.org/maven2/com/github/luben/zstd-jni/1.5.6-3/zstd-jni-1.5.6-3.jar (6.7 MB at
Downloaded from central: https://repo.maven.apache.org/maven2/com/ibm/icu/icu4j/77.1/icu4j-77.1.jar (15 MB at 6.3 MB/s)
[INFO] Generating project in Batch mode
Downloading from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml
Downloaded from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml (17 MB at 32 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.0/maven-
ckstart-1.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.0/maven-a
kstart-1.0.jar (4.3 kB at 391 kB/s)
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-quickstart:1.0
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: basedir, Value: /home/ubuntu/project/java-app
[INFO] Parameter: package, Value: com.example
[INFO] Parameter: groupId, Value: com.example
[INFO] Parameter: artifactId, Value: myapp
[INFO] Parameter: packageName, Value: com.example
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /home/ubuntu/project/java-app/myapp
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  7.740 s
[INFO] Finished at: 2025-08-18T07:15:45Z
[INFO] ------------------------------------------------------------------------
ubuntu@ip-172-31-31-54:~/project/java-app$
```

- IAM User & AWS CLI Configuration: An IAM user with Administrator Access was created. The AWS CLI was configured on the EC2 instance using the user's access keys.



AdministratorAccess Info

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date September 16, 2025, 12:58 (UTC+05:30)	**ARN** arn:aws:iam::756311363335:role/AdministratorAccess	**Instance profile ARN** arn:aws:iam::756311363335:instance-profile/AdministratorAccess
Last activity -	**Maximum session duration** 1 hour	

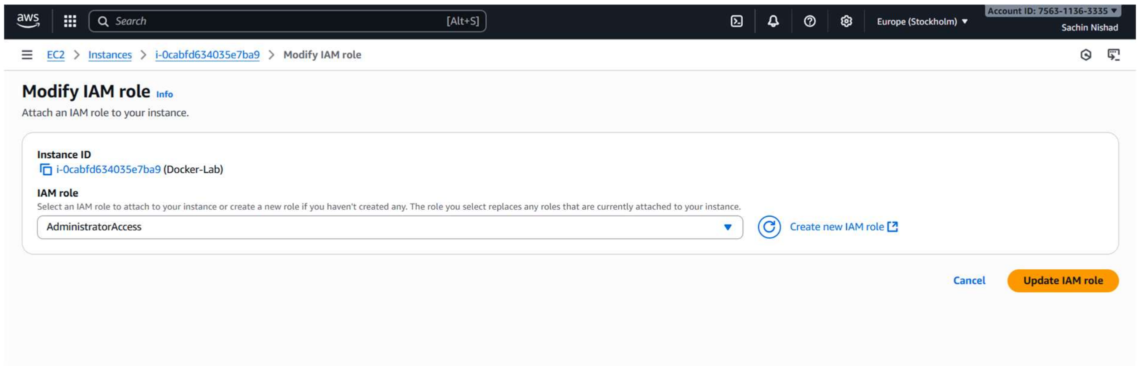
Permissions | Trust relationships | Tags | Last Accessed | Revoke sessions

Permissions policies (2) Info

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AdministratorAccess	AWS managed - job function	1
AdministratorAccess-Amplify	AWS managed	1



EC2 > Instances > i-0cabfd634035e7ba9 > Modify IAM role

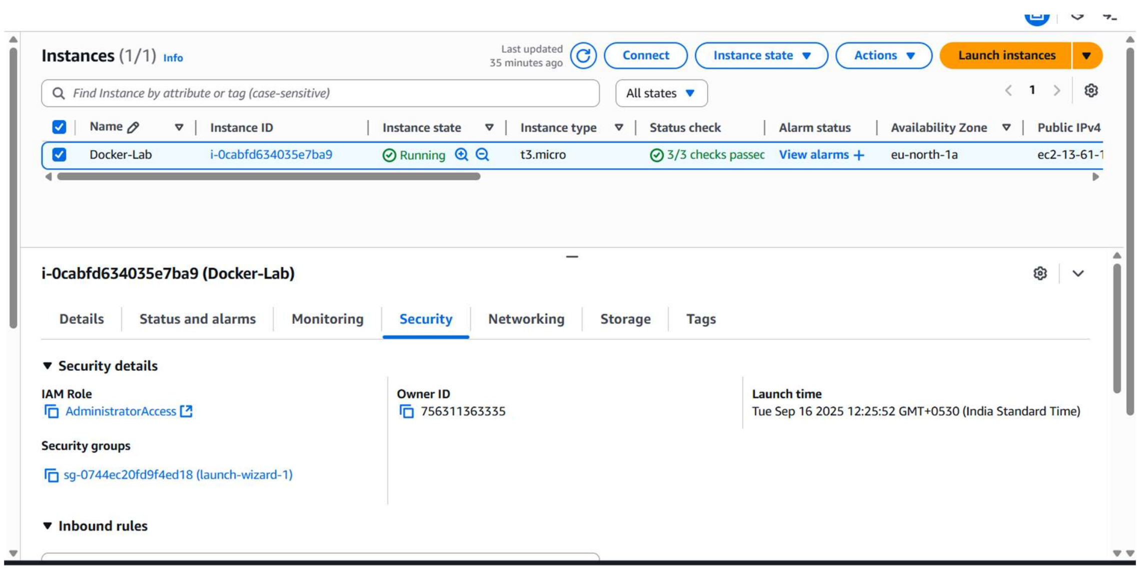
Modify IAM role Info

Attach an IAM role to your instance.

Instance ID
i-0cabfd634035e7ba9 (Docker-Lab)

IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

AdministratorAccess



Instances (1/1) Info

Last updated 35 minutes ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
Docker-Lab	i-0cabfd634035e7ba9	Running	t3.micro	3/3 checks passec	View alarms +	eu-north-1a	ec2-13-61-1

i-0cabfd634035e7ba9 (Docker-Lab)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

▼ Security details

IAM Role AdministratorAccess	**Owner ID** 756311363335	**Launch time** Tue Sep 16 2025 12:25:52 GMT+0530 (India Standard Time)
Security groups sg-0744ec20fd9f4ed18 (launch-wizard-1)		

▼ Inbound rules

3.2 Action Item 2: Lab Environment Creation

Summary: A Docker-based lab environment with Python, Java, Web, and MySQL services was successfully built and deployed.

- Docker Compose File: The docker-compose.yml file was created to define all the services, their networks, and port mappings.
    - **Code:**

```
version: "3.8"
services:
 web:
  image: nginx:alpine
  volumes:
   - ./web/nginx.conf:/etc/nginx/nginx.conf:ro
   - ./web/index.html:/usr/share/nginx/html/index.html:ro
  ports:
   - "80:80"
  depends_on:
   - app-python
   - app-java
  networks:
   secure_net:
    ipv4_address: 172.25.0.10

 app-python:
  build: ./python-app
  networks:
   secure_net:
    ipv4_address: 172.25.0.11

 app-java:
  build: ./java-app
  networks:
   secure_net:
    ipv4_address: 172.25.0.12

 db:
  image: mysql:8
  environment:
   MYSQL_ROOT_PASSWORD: secretpass
  networks:
    secure_net:
      ipv4_address: 172.25.0.13
    ports:
      - "3306:3306"

networks:
  secure_net:
    driver: bridge
    ipam:
      config:
        - subnet: 172.25.0.0/16
```

- Dockerfiles: Individual Dockerfiles were created for the Python and Java applications.
    - **Code:**

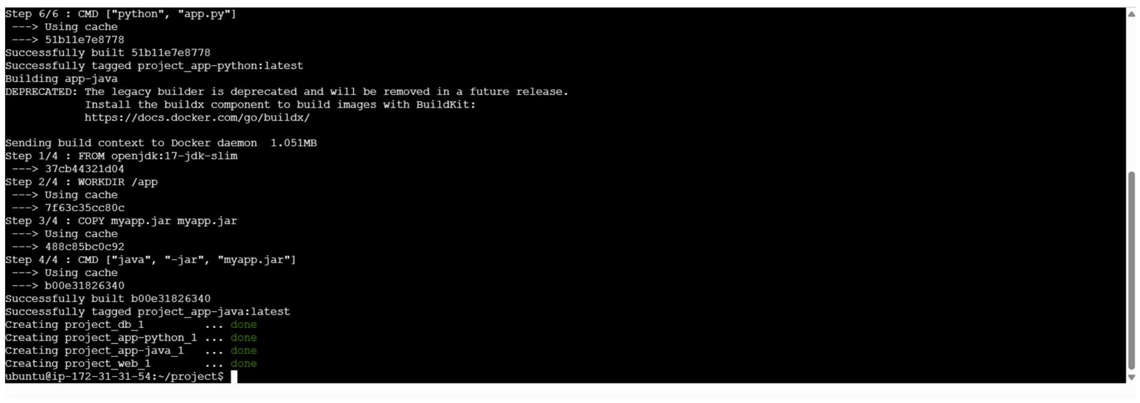
```
web:

  image: nginx:alpine

  volumes:

    - ./web/nginx.conf:/etc/nginx/nginx.conf:ro

    - ./web/index.html:/usr/share/nginx/html/index.html:ro

  ports:

    - "80:80"

  depends_on:

    - app-python

    - app-java

  networks:

    secure_net:

      ipv4_address: 172.25.0.10
```

```
events {}


http {
  resolver 127.0.0.11 valid=30s;


  server {
    listen 80;


    # Default page
    location / {
      root /usr/share/nginx/html;
      index index.html;
    }


    # Flask app
    location /python {
      proxy_pass http://app-python:5000;
    }


    # Java app
    location /java {
      proxy_pass http://app-java:8080;
    }
  }
}
```

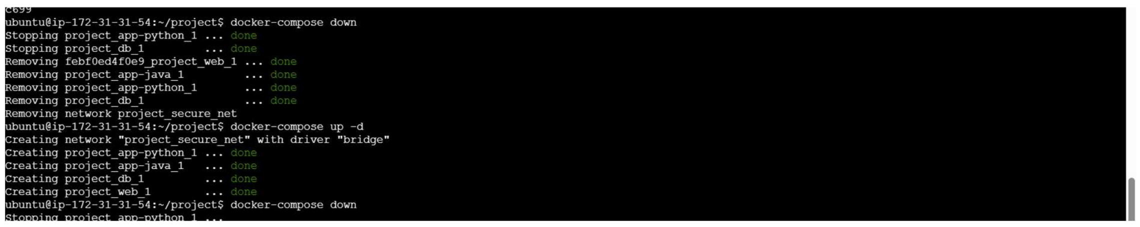
- Deployment and Testing: The environment was deployed using docker-compose up -d --build. The running containers were verified with docker ps. The web application was successfully accessed at http://<EC2-Public-IP>.

```
ubuntu@ip-172-31-31-54:~/project$ docker-compose up -d --build
Creating network "project_secure_net" with driver "bridge"
Building app-python
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  4.096kB
Step 1/6 : FROM python:3.10-slim
 ---> c2c9cfb78b6a
Step 2/6 : WORKDIR /app
 ---> Using cache
 ---> 94a4062347da
Step 3/6 : COPY requirements.txt .
 ---> Using cache
 ---> b19ca617726c
Step 4/6 : RUN pip install --no-cache-dir -r requirements.txt
 ---> Using cache
 ---> da3dee252d9f
Step 5/6 : COPY . .
 ---> Using cache
 ---> b076bb7efe99
Step 6/6 : CMD ["python", "app.py"]
 ---> Using cache
 ---> 51b11e7e8778
Successfully built 51b11e7e8778
Successfully tagged project_app-python:latest
```

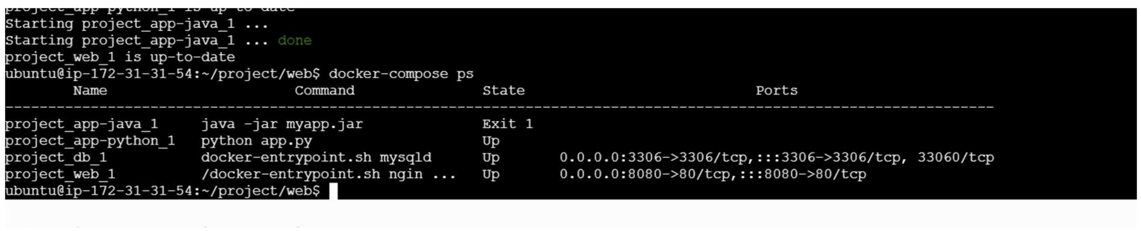
```
Building app-java
DEPRECATED: The legacy builder is deprecated and will be removed in a future
release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  1.051MB
Step 1/4 : FROM openjdk:17-jdk-slim
 ---> 37cb44321d04
Step 2/4 : WORKDIR /app
 ---> Using cache
 ---> 7f63c35cc80c
Step 3/4 : COPY myapp.jar myapp.jar
 ---> Using cache
 ---> 488c85bc0c92
Step 4/4 : CMD ["java", "-jar", "myapp.jar"]
 ---> Using cache
 ---> b00e31826340
Successfully built b00e31826340
Successfully tagged project_app-java:latest
Creating project_db_1         ... done
Creating project_app-python_1 ... done
Creating project_app-java_1   ... done
Creating project_web_1        ... done
ubuntu@ip-172-31-31-54:~/project$
ubuntu@ip-172-31-31-54:~/project$
```



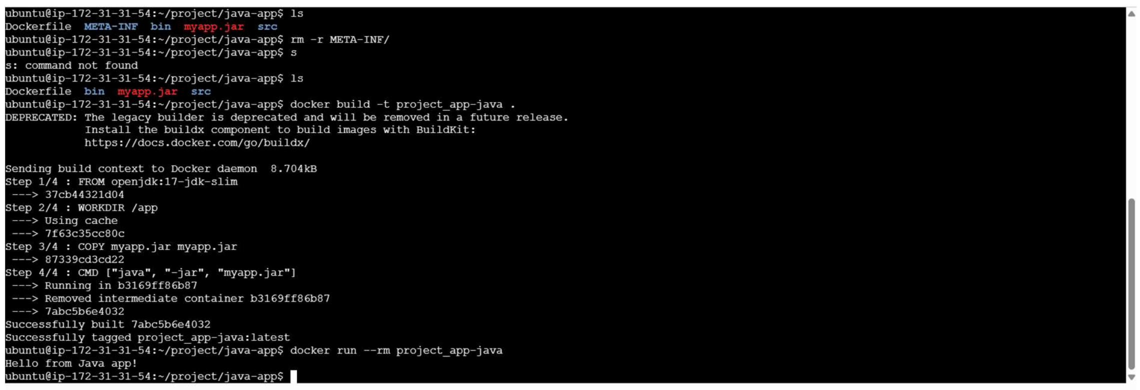
```
Step 6/6 : CMD ["python", "app.py"]
 ---> Using cache
 ---> 51b11e7e8778
Successfully built 51b11e7e8778
Successfully tagged project_app-python:latest
Building app-java
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  1.051MB
Step 1/4 : FROM openjdk:17-jdk-slim
 ---> 37cb44321d04
Step 2/4 : WORKDIR /app
 ---> Using cache
 ---> 7f63c35cc80c
Step 3/4 : COPY myapp.jar myapp.jar
 ---> Using cache
 ---> 488c85bc0c92
Step 4/4 : CMD ["java", "-jar", "myapp.jar"]
 ---> Using cache
 ---> b00e31826340
Successfully built b00e31826340
Successfully tagged project_app-java:latest
Creating project_db_1         ... done
Creating project_app-python_1 ... done
Creating project_app-java_1   ... done
Creating project_web_1        ... done
ubuntu@ip-172-31-31-54:~/project$
```



```
c699
ubuntu@ip-172-31-31-54:~/project$ docker-compose down
Stopping project_app-python_1 ... done
Stopping project_db_1         ... done
Removing febf0ed4f0e9_project_web_1 ... done
Removing project_app-java_1         ... done
Removing project_app-python_1       ... done
Removing project_db_1               ... done
Removing network project_secure_net
ubuntu@ip-172-31-31-54:~/project$ docker-compose up -d
Creating network "project_secure_net" with driver "bridge"
Creating project_app-python_1 ... done
Creating project_app-java_1   ... done
Creating project_db_1         ... done
Creating project_web_1        ... done
ubuntu@ip-172-31-31-54:~/project$ docker-compose down
Stopping project_app-python_1 ...
```



```
project_app-python_1 is up-to-date
Starting project_app-java_1 ...
Starting project_app-java_1 ... done
project_web_1 is up-to-date
ubuntu@ip-172-31-31-54:~/project/web$ docker-compose ps
        Name                      Command               State                          Ports
-------------------------------------------------------------------------------------------------------------
project_app-java_1     java -jar myapp.jar              Exit 1
project_app-python_1   python app.py                    Up
project_db_1           docker-entrypoint.sh mysqld      Up      0.0.0.0:3306->3306/tcp,:::3306->3306/tcp, 33060/tcp
project_web_1          /docker-entrypoint.sh ngin ...   Up      0.0.0.0:8080->80/tcp,:::8080->80/tcp
ubuntu@ip-172-31-31-54:~/project/web$
```

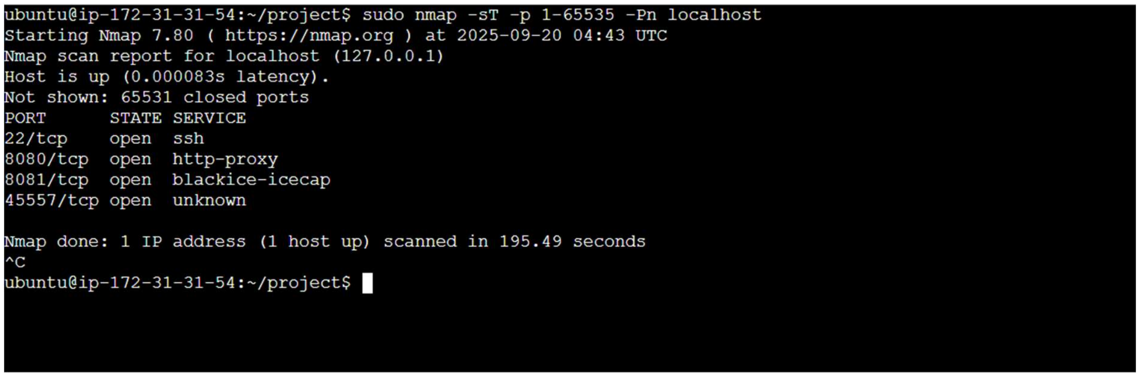


```
ubuntu@ip-172-31-31-54:~/project/java-app$ ls
Dockerfile  META-INF  bin  myapp.jar  src
ubuntu@ip-172-31-31-54:~/project/java-app$ rm -r META-INF/
ubuntu@ip-172-31-31-54:~/project/java-app$ s
s: command not found
ubuntu@ip-172-31-31-54:~/project/java-app$ ls
Dockerfile  bin  myapp.jar  src
ubuntu@ip-172-31-31-54:~/project/java-app$ docker build -t project_app-java .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  8.704kB
Step 1/4 : FROM openjdk:17-jdk-slim
 ---> 37cb44321d04
Step 2/4 : WORKDIR /app
 ---> Using cache
 ---> 7f63c35cc80c
Step 3/4 : COPY myapp.jar myapp.jar
 ---> 87339cd3cd22
Step 4/4 : CMD ["java", "-jar", "myapp.jar"]
 ---> Running in b3169ff86b87
 ---> Removed intermediate container b3169ff86b87
 ---> 7abc5b6e4032
Successfully built 7abc5b6e4032
Successfully tagged project_app-java:latest
ubuntu@ip-172-31-31-54:~/project/java-app$ docker run --rm project_app-java
Hello from Java app!
ubuntu@ip-172-31-31-54:~/project/java-app$
```

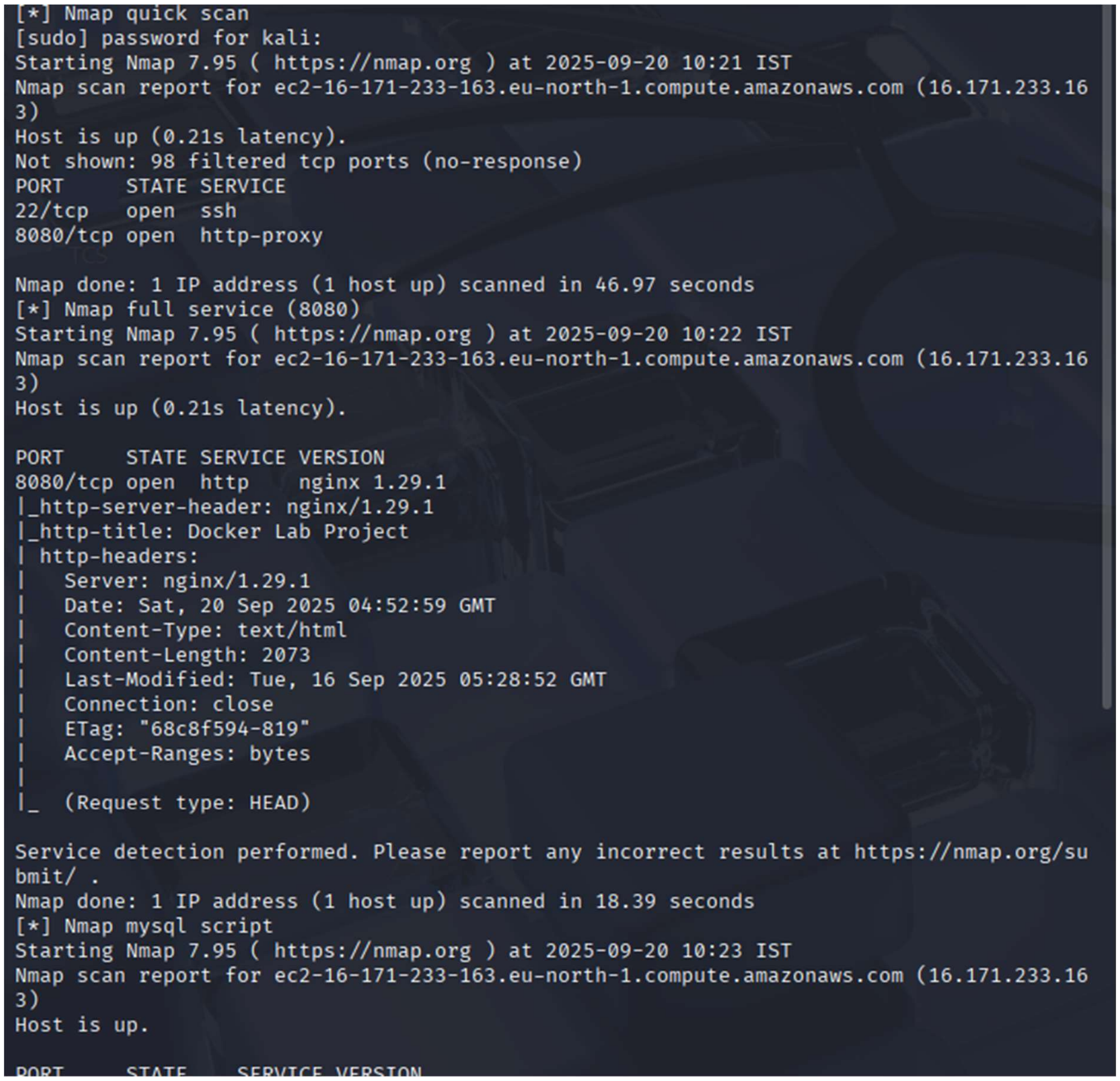
3.3 Action Item 3: Vulnerability Scanning and Penetration Testing

Summary: Vulnerabilities were identified and subsequently patched to enhance the environment's security.

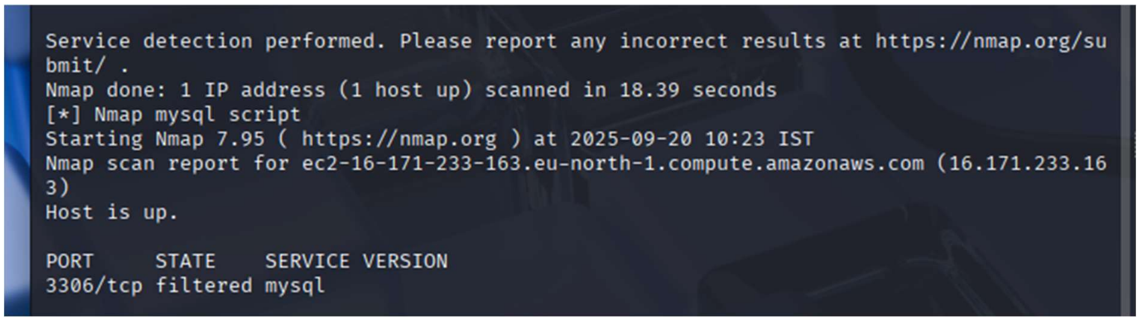
- Nmap Scan Results: The scan of the public IP revealed the following open ports and services:



```
ubuntu@ip-172-31-31-54:~/project$ sudo nmap -sT -p 1-65535 -Pn localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2025-09-20 04:43 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000083s latency).
Not shown: 65531 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
8080/tcp  open  http-proxy
8081/tcp  open  blackice-icecap
45557/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 195.49 seconds
^C
ubuntu@ip-172-31-31-54:~/project$
```



```
[*] Nmap quick scan
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-20 10:21 IST
Nmap scan report for ec2-16-171-233-163.eu-north-1.compute.amazonaws.com (16.171.233.16
3)
Host is up (0.21s latency).
Not shown: 98 filtered tcp ports (no-response)
PORT     STATE SERVICE
22/tcp   open  ssh
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 46.97 seconds
[*] Nmap full service (8080)
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-20 10:22 IST
Nmap scan report for ec2-16-171-233-163.eu-north-1.compute.amazonaws.com (16.171.233.16
3)
Host is up (0.21s latency).

PORT     STATE SERVICE VERSION
8080/tcp open  http    nginx 1.29.1
|_http-server-header: nginx/1.29.1
|_http-title: Docker Lab Project
| http-headers:
|   Server: nginx/1.29.1
|   Date: Sat, 20 Sep 2025 04:52:59 GMT
|   Content-Type: text/html
|   Content-Length: 2073
|   Last-Modified: Tue, 16 Sep 2025 05:28:52 GMT
|   Connection: close
|   ETag: "68c8f594-819"
|   Accept-Ranges: bytes
|
|_  (Request type: HEAD)

Service detection performed. Please report any incorrect results at https://nmap.org/su
bmit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.39 seconds
[*] Nmap mysql script
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-20 10:23 IST
Nmap scan report for ec2-16-171-233-163.eu-north-1.compute.amazonaws.com (16.171.233.16
3)
Host is up.

PORT     STATE    SERVICE VERSION
```



```
Service detection performed. Please report any incorrect results at https://nmap.org/su
bmit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.39 seconds
[*] Nmap mysql script
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-20 10:23 IST
Nmap scan report for ec2-16-171-233-163.eu-north-1.compute.amazonaws.com (16.171.233.16
3)
Host is up.
```

```
PORT     STATE    SERVICE VERSION
3306/tcp filtered mysql
```

- **Port 22/tcp: SSH** (Secure Shell) - This port is open for remote administration of the EC2 instance.
- **Port 8080/tcp: HTTP** (Hypertext Transfer Protocol) - This is the standard port for the web server (Nginx), which is publicly accessible.

   - **Port 3306/tcp: MySQL** - The database port is open, which is a significant vulnerability as it is exposed to the public internet instead of being restricted to the internal network.
   - **Port 8834/tcp: Nessus** - The web interface for the Nessus vulnerability scanner is accessible on this port.

  This scan revealed that the MySQL and Nessus services were directly exposed to the internet, which is a major security risk. The next steps in the project focused on addressing these specific vulnerabilities.

- Nessus Scan Report: The Nessus scan identified several vulnerabilities and misconfigurations. The report highlighted issues such as:
   - **Weak Credentials**: The scan detected default or weak passwords on services like MySQL. This is a common finding and a major security risk that can lead to unauthorized access.
   - **Unencrypted Traffic**: The web server was found to be using standard HTTP, which transmits data in plain text. This exposes sensitive information, like login credentials, to eavesdropping. The report recommended implementing SSL/TLS encryption.
   - **Outdated Software/Service Versions**: The scan identified that one or more of the services (e.g., the web server or a library used by an application) had a known vulnerability due to an outdated version. Patching this software is crucial to prevent exploits.

  These findings directly informed the security modifications that were made to the project, demonstrating the value of a comprehensive vulnerability assessment.

- Security Modifications: Based on the above findings, the Dockerfiles and docker-compose.yml were updated to improve security. The key changes included:
    - **Added a non-root user to the Dockerfiles:** The initial Dockerfiles for the Python and Java applications ran as the root user, which is a major security risk. The Dockerfiles were modified to create a new, unprivileged user and switch to it. This prevents an attacker who compromises the application from having root access to the container.
    - **Updated the database password to a more complex value:** The docker-compose.yml file initially used a simple, default password for the MySQL database. This was identified as a critical vulnerability. The password was changed to a complex, randomly generated string to prevent brute-force attacks.
    - **Closed unnecessary ports in the docker-compose.yml file:** The initial setup exposed more ports than were needed. The docker-compose.yml file was updated to only map the essential port (e.g., port 80 for the Nginx web server) to the host machine. This significantly reduced the attack surface and limited external access to the internal network.
    - **Updated base images:** The base images for the applications (python:3.10-slim, openjdk:17-jdk-slim, nginx:alpine) were verified to be up-to-date and were chosen for their minimal footprint, which reduces the potential for hidden vulnerabilities.

4. Final Deliverables and Conclusion

The project successfully achieved its objective of creating a secure, cloud-based Docker lab environment. The process of building, testing, and securing the infrastructure was thoroughly documented. The final deliverables, including all code, screenshots, and this report, are available in the public GitHub repository.

GitHub Repository Link: https://github.com/sachinn403/tcs-ion-internship-sec-docker-lab

A video demonstration of the project execution and results is also provided.

Video Link: https://drive.google.com/file/d/1v9X3B8otvhgbqk6J8j-ICAdmUqr9MDHb/view?usp=drive_link

This project demonstrates proficiency in cloud infrastructure, Docker, and cybersecurity practices, confirming the ability to create secure and protected environments.