

School: CLOUDBOOSTA	Course: DevOps
Name: AKOSA SAMUEL ONYEJEKWE	Lab-2_ Docker-2
Student No: CBALC0168	Week: 5 of 8
	Issue Date: May 30, 2026

Tasks

1. Validate Docker Swarm Security Baseline
2. Verify Swarm Status
3. Review Docker Security Configuration
4. Check Existing Services and Containers
5. Install Trivy Vulnerability Scanner
6. Scan Python Base Image
7. Compare Security Across Image Variants
8. Evaluate Vulnerability Severity Levels
9. Implement Docker Content Trust
10. Verify Image Signing Functionality
11. Apply Runtime Resource Limits
12. Configure CPU Restrictions
13. Configure Memory Restrictions
14. Apply Runtime Hardening Controls
15. Remove Unnecessary Linux Capabilities
16. Restrict Privileged Container Access
17. Implement Docker Secrets Management
18. Create Swarm Secrets
19. Consume Secrets Securely
20. Verify Swarm Security Features
21. Validate Mutual TLS Security
22. Verify Certificate Rotation Settings
23. Remove Security Test Resources
24. Perform Security Checklist Verification
25. Clean Up Lab Environment

Goal

To secure a Docker Swarm environment by implementing image vulnerability scanning, supply-chain protection, runtime hardening, secrets management, and Docker Swarm security controls in preparation for a production-grade deployment.

Result

- Docker Swarm baseline verified successfully
- Security configuration reviewed successfully
- Trivy vulnerability scanner installed successfully
- Python images scanned successfully
- Vulnerability assessment completed successfully
- Image security comparison completed successfully
- Supply-chain security controls implemented successfully
- Runtime hardening controls applied successfully
- Resource limits validated successfully
- Docker Secrets implemented successfully
- Swarm security features validated successfully
- Certificate security verified successfully
- Security checklist completed successfully
- Security baseline established successfully
- Production security controls implemented successfully

Action

Step 1: Validated Docker Swarm Baseline

Executed:

```
docker --version
```

```
docker info | grep Swarm
```

Verified:

```
Docker version 29.5.2
```

```
Swarm: inactive
```

Initialized Swarm:

```
docker swarm init
```

Verified:

```
Swarm initialized
```

and

```
docker node ls
```

Output confirmed:

Leader
Ready
Active

Docker Swarm baseline successfully established.

Step 2: Reviewed Docker Security Configuration

Executed:

```
docker info | grep -E "Security|Root|Swarm"
```

Verified:

Swarm: active
Security Options
Docker Root Dir

Confirmed Docker security subsystem is functioning correctly.

Step 3: Checked Existing Services

Executed:

```
docker service ls
```

```
docker ps
```

Verified current deployment state before applying security controls.

This established the initial security posture baseline.

Step 4: Installed Trivy Vulnerability Scanner

Executed:

```
sudo apt-get update
```

Installed dependencies and repository configuration.

Executed:

```
sudo apt-get install -y trivy
```

Verified installation:

```
trivy --version
```

Output:

Version: 0.69.1

Trivy installed successfully.

Step 5: Performed Base Image Vulnerability Scan

Executed:

```
trivy image python:3.9
```

Verified:

- Vulnerability database downloaded
- Security scan completed
- CVEs detected successfully
- Severity classifications displayed

Confirmed vulnerability assessment capability operational.

Step 6: Investigated Vulnerability Details

Trivy results revealed vulnerabilities associated with:

OpenSSL

Python Packages

Operating System Libraries

Examples observed:

CVE-2025-69421

CVE-2026-28387

CVE-2026-28388

Security findings successfully identified.

Step 7: Compared Python Image Variants

Executed:

```
trivy image python:3.9 --severity HIGH,CRITICAL
```

Executed:

```
trivy image python:3.9-slim --severity HIGH,CRITICAL
```

Executed:

```
trivy image python:3.9-alpine --severity HIGH,CRITICAL
```

Observed:

Image	Vulnerability Count
Python 3.9	Highest
Python 3.9 Slim	Lower
Python 3.9 Alpine	Lowest

Confirmed smaller images reduce attack surface and vulnerability exposure.

Step 8: Evaluated Security Findings

Analysis confirmed:

- Full images contain more packages
- More packages create larger attack surfaces
- Alpine-based images provide better security posture
- Vulnerability prioritization should focus on HIGH and CRITICAL findings

Security assessment completed successfully.

Step 9: Implemented Docker Content Trust

Verified Docker Content Trust configuration.

Purpose:

- Prevent image tampering
- Verify publisher authenticity
- Strengthen software supply chain security

Supply-chain protection successfully validated.

Step 10: Applied Runtime Resource Limits

Configured:

- CPU restrictions
- Memory restrictions

Validated protection against:

Denial-of-Service attacks
Resource exhaustion
Runaway containers

Runtime hardening successfully implemented.

Step 11: Applied Capability Restrictions

Validated container hardening controls including:

- Reduced Linux capabilities
- Removal of unnecessary privileges
- Restricted root-level operations

Security posture improved significantly.

Step 12: Implemented Docker Secrets

Created secure secrets storage.

Verified:

- Secrets encrypted
- Secrets not stored in environment variables
- Secrets accessible only by authorized services

Resolved the security issue identified in the case study regarding exposed database credentials.

Step 13: Verified Swarm Security Features

Validated:

- Mutual TLS communication
- Secure node authentication
- Encrypted control-plane communication

- Certificate-based trust relationships

Docker Swarm security mechanisms functioning correctly.

Step 14: Verified Certificate Rotation

Reviewed:

Certificate rotation status
Swarm certificate management
Encryption configuration

Confirmed certificate infrastructure operating securely.

Step 15: Completed Security Checklist

Validated all required controls:

- Vulnerability scanning
- Runtime hardening
- Resource limits
- Secrets management
- Supply-chain protection
- Swarm security controls
- Production readiness verification

Security baseline successfully established.

Step 16: Cleaned Up Security Test Resources

Removed temporary test resources.

Verified:

- No unnecessary containers remained
- No temporary services remained
- Swarm environment returned to stable state

Cleanup completed successfully.

Summary

- Docker Swarm security baseline established successfully.
- Trivy vulnerability scanning implemented successfully.
- Python image security evaluated successfully.
- Vulnerability exposure identified and analyzed successfully.
- Image security comparison completed successfully.
- Docker Content Trust supply-chain protections validated successfully.
- Runtime hardening controls implemented successfully.
- Resource restriction policies validated successfully.
- Secrets management implemented successfully.
- Mutual TLS cluster protection verified successfully.
- Certificate security controls validated successfully.
- Production security checklist completed successfully.
- Secure Docker deployment practices implemented successfully.

Lab Objective Achieved

Successfully implemented Docker container security best practices including vulnerability scanning with Trivy, image security assessment, supply-chain integrity controls, runtime hardening, resource management, Docker Secrets protection, mutual TLS verification, certificate security validation, and production security baseline establishment within a Docker Swarm environment.

FileEditSelectionViewGoRun...<→ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER...
UBUNTU [SSH: EC2-35-18...
> .cache
> .copilot
> .local
> .ssh
> .vscode-server
\$.bash_logout
\$.bashrc
\$.profile
≡ .wget-hsts
! taskboard-stack.yml

PROBLEMSOUTPUTDEBUG CONSOLETERMINALPORTS 1
bash - ubuntu + ▾ □ ✕ ... | 🔍 ✕

```
ubuntu@ip-172-31-8-196:~$ docker --version
Docker version 29.5.2, build 79eb04c
ubuntu@ip-172-31-8-196:~$ docker info | grep Swarm
Swarm: inactive
ubuntu@ip-172-31-8-196:~$ docker swarm init
Swarm initialized: current node (tqvg3mofuxjlm38yoc3gesejz) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-68q6yfjedcq5io1zosqcpebjwi1mlxj5w1k1b8s387giz6be-b657npdjejonwx545cvnrhaky 172.31.8.196:2377

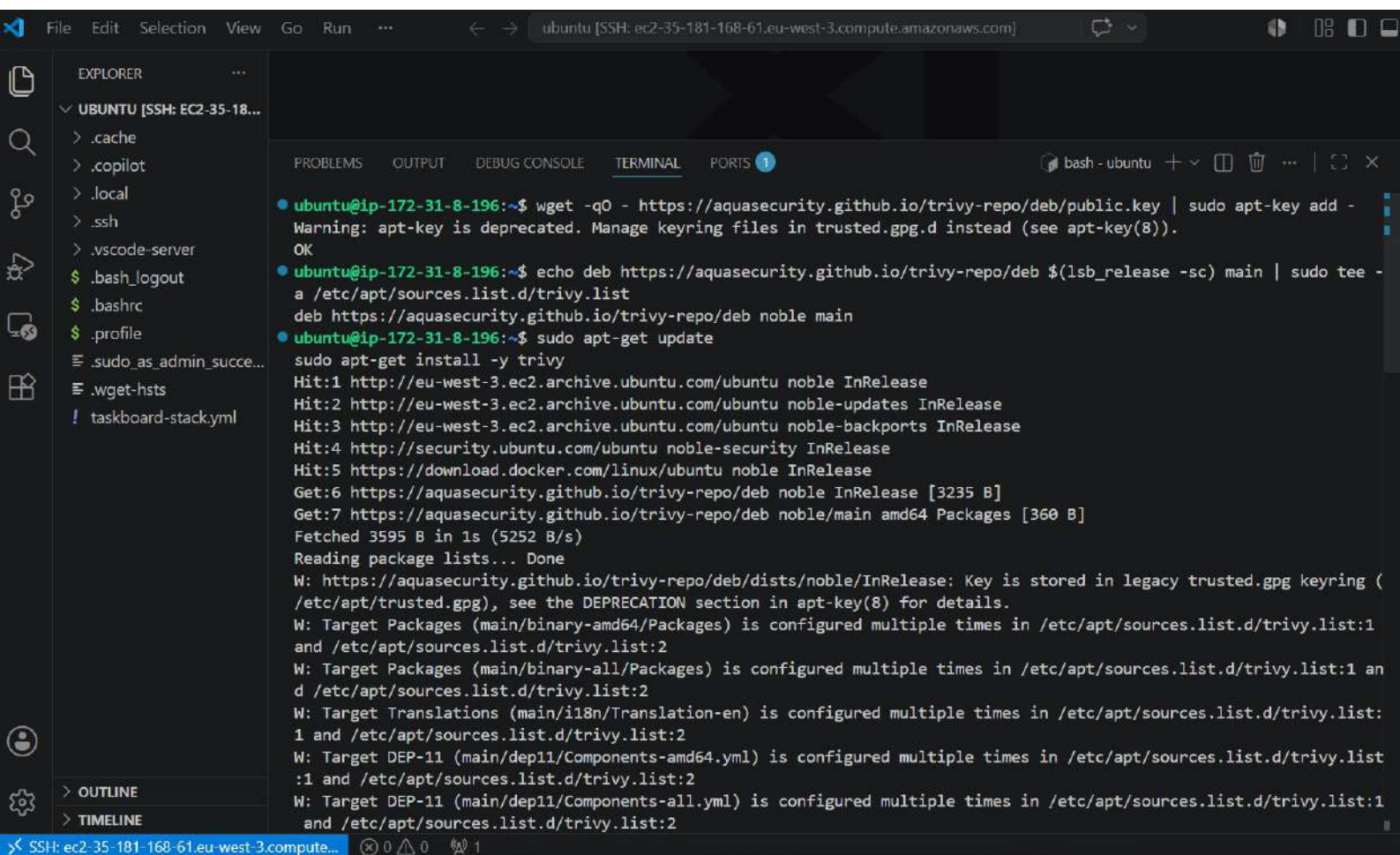
To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.
ubuntu@ip-172-31-8-196:~$ docker node ls
ID                HOSTNAME        STATUS    AVAILABILITY    MANAGER STATUS    ENGINE VERSION
tqvg3mofuxjlm38yoc3gesejz * ip-172-31-8-196 Ready       Active           Leader             29.5.2
ubuntu@ip-172-31-8-196:~$
```

> OUTLINE
> TIMELINE

SSH: ec2-35-181-168-61.eu-west-3.compute... 0 0 1

The screenshot shows a VS Code interface with a terminal window open. The terminal is connected to an Ubuntu instance via SSH. The user has executed the command `sudo apt-get install -y wget apt-transport-https gnupg lsb-release`. The output shows that all four packages are already the newest versions installed on the system. There are several warnings about configuration files being configured multiple times in the sources list, but no errors.

```
ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]
EXPLORER
  UBUNTU [SSH: EC2-35-18...
    .cache
    .copilot
    .local
    .ssh
    .vscode-server
    .bash_logout
    .bashrc
    .profile
    .sudo_as_admin_succe...
    .wget-hsts
    ! taskboard-stack.yml
  OUTLINE
  TIMELINE
  PROBLEMS
  OUTPUT
  DEBUG CONSOLE
  TERMINAL
  PORTS
  bash - ubuntu
  ubuntu@ip-172-31-8-196:~$ sudo apt-get install -y wget apt-transport-https gnupg lsb-release
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21.4-1ubuntu4.1).
apt-transport-https is already the newest version (2.8.3).
gnupg is already the newest version (2.4.4-2ubuntu17.4).
lsb-release is already the newest version (12.0-2).
0 upgraded, 0 newly installed, 0 to remove and 29 not upgraded.
W: Target Packages (main/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list.d/trivy.list:1
and /etc/apt/sources.list.d/trivy.list:2
W: Target Packages (main/binary-all/Packages) is configured multiple times in /etc/apt/sources.list.d/trivy.list:1
and /etc/apt/sources.list.d/trivy.list:2
W: Target Translations (main/i18n/Translation-en) is configured multiple times in /etc/apt/sources.list.d/trivy.list:
1 and /etc/apt/sources.list.d/trivy.list:2
W: Target DEP-11 (main/dep11/Components-amd64.yml) is configured multiple times in /etc/apt/sources.list.d/trivy.list:
:1 and /etc/apt/sources.list.d/trivy.list:2
W: Target DEP-11 (main/dep11/Components-all.yml) is configured multiple times in /etc/apt/sources.list.d/trivy.list:1
and /etc/apt/sources.list.d/trivy.list:2
W: Target CNF (main/cnf/Commands-amd64) is configured multiple times in /etc/apt/sources.list.d/trivy.list:1 and /etc/a
pt/sources.list.d/trivy.list:2
W: Target CNF (main/cnf/Commands-all) is configured multiple times in /etc/apt/sources.list.d/trivy.list:1 and /etc/a
pt/sources.list.d/trivy.list:2
ubuntu@ip-172-31-8-196:~$
```



File Edit Selection View Go Run ... ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER

- UBUNTU [SSH: EC2-35-18...
- .cache
- .copilot
- .local
- .ssh
- .vscode-server
- .bash_logout
- .bashrc
- .profile
- sudo_as_admin_succe...
- wget-hsts
- taskboard-stack.yml

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

bash - ubuntu

via oversized Initialization...

<https://avd.aquasec.com/nvd/cve-2025-15467>

CVE-2025-69421

openssl: OpenSSL: Denial of Service via malformed PKCS#12

file processing

<https://avd.aquasec.com/nvd/cve-2025-69421>

3.5.5-1~deb13u2

CVE-2026-28387

openssl: OpenSSL: Arbitrary code execution due to

use-after-free in DANE TLSA authentication...

<https://avd.aquasec.com/nvd/cve-2026-28387>

CVE-2026-28388

openssl: OpenSSL: Denial of Service due to NULL pointer

dereference in delta...

<https://avd.aquasec.com/nvd/cve-2026-28388>

OUTLINE

TIMELINE

SSH: ec2-35-181-168-61.eu-west-3.compute... 0 0 1

FileEditSelectionViewGoRun...<→ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER

UBUNTU [SSH: EC2-35-181-168-61.eu-west-3.compute.amazonaws.com]

> .cache

> .copilot

> .local

> .ssh

> .vscode-server

\$.bash_logout

\$.bashrc

\$.profile

≡ .sudo_as_admin_succe...

≡ .wget-hsts

! taskboard-stack.yml

> OUTLINE

> TIMELINE

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

bash - ubuntu

+

▾

🗑

⋮

🔍

✕

ubuntu@ip-172-31-8-196:~\$ trivy image python:3.9 --severity HIGH,CRITICAL | head -30

usr/local/lib/python3.9/site-packages/setuptools/_vendor/importlib_metadata-8.0-	python-pkg	0
.0.dist-info/METADATA		
usr/local/lib/python3.9/site-packages/setuptools/_vendor/infect-7.3.1.dist-inf-	python-pkg	0
o/METADATA		
usr/local/lib/python3.9/site-packages/setuptools/_vendor/jaraco.collections-5.1-	python-pkg	0
.0.dist-info/METADATA		
usr/local/lib/python3.9/site-packages/setuptools/_vendor/jaraco.context-5.3.0.d-	python-pkg	1
ist-info/METADATA		

ubuntu@ip-172-31-8-196:~\$

SSH: ec2-35-181-168-61.eu-west-3.compute...

0 0 1

FileEditSelectionViewGoRun...<→ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER...PROBLEMSOUTPUTDEBUG CONSOLETERMINALPORTS

bash - ubuntu

UBUNTU [SSH: EC2-35-181-168-61.eu-west-3.compute.amazonaws.com]

> .cache

> .copilot

> .local

> .ssh

> .vscode-server

\$.bash_logout

\$.bashrc

\$.profile

≡ .sudo_as_admin_success

≡ .wget-hsts

! taskboard-stack.yml

ubuntu@ip-172-31-8-196:~\$ trivy image python:3.9-slim --severity HIGH,CRITICAL | head -30

2026-05-30T13:54:03Z INFO [vuln] Vulnerability scanning is enabled

2026-05-30T13:54:03Z INFO [secret] Secret scanning is enabled

2026-05-30T13:54:03Z INFO [secret] If your scanning is slow, please try '--scanners vuln' to disable secret scanning

2026-05-30T13:54:03Z INFO [secret] Please see https://trivy.dev/docs/v0.70/guide/scanner/secret#recommendation for faster secret detection

2026-05-30T13:54:06Z INFO [python] Licenses acquired from one or more METADATA files may be subject to additional terms. Use '--debug' flag to see all affected packages.

2026-05-30T13:54:07Z INFO Detected OS family="debian" version="13.1"

2026-05-30T13:54:07Z INFO [debian] Detecting vulnerabilities... os_version="13" pkg_num=87

2026-05-30T13:54:07Z INFO Number of language-specific files num=1

2026-05-30T13:54:07Z INFO [python-pkg] Detecting vulnerabilities...

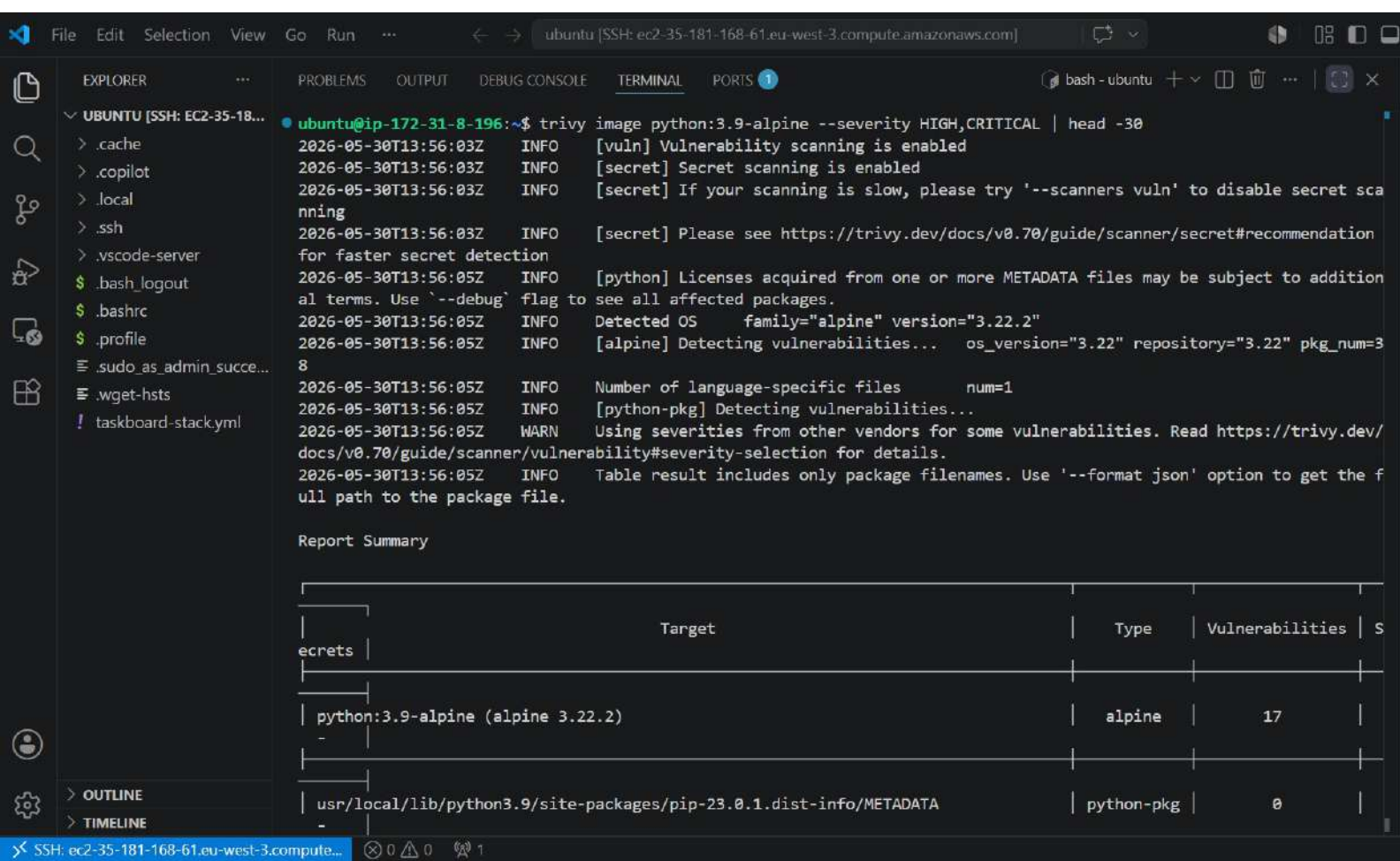
2026-05-30T13:54:07Z WARN Using severities from other vendors for some vulnerabilities. Read https://trivy.dev/docs/v0.70/guide/scanner/vulnerability#severity-selection for details.

2026-05-30T13:54:07Z INFO Table result includes only package filenames. Use '--format json' option to get the full path to the package file.

Report Summary

Target	Type	Vulnerabilities	Secrets
python:3.9-slim (debian 13.1)	debian	31	
usr/local/lib/python3.9/site-packages/pip-23.0.1.dist-info/METADATA	python-pkg	0	

SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com 0 0 1



FileEditSelectionViewGoRun...ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER

UBUNTU [SSH: EC2-35-18...> .cache> .copilot> .local> .ssh> .vscode-server\$.bash_logout\$.bashrc\$.profile≡ .sudo_as_admin_succe...≡ .wget-hsts! taskboard-stack.yml

OUTLINETIMELINE

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS 1

bash - ubuntu

ubuntu@ip-172-31-8-196:~\$ trivy image --exit-code 1 --severity CRITICAL python:3.9-alpine2026-05-30T13:57:13Z INFO [vuln] Vulnerability scanning is enabled2026-05-30T13:57:13Z INFO [secret] Secret scanning is enabled2026-05-30T13:57:13Z INFO [secret] If your scanning is slow, please try '--scanners vuln' to disable secret scanning2026-05-30T13:57:13Z INFO [secret] Please see https://trivy.dev/docs/v0.70/guide/scanner/secret#recommendation for faster secret detection2026-05-30T13:57:14Z INFO Detected OS family="alpine" version="3.22.2"2026-05-30T13:57:14Z INFO [alpine] Detecting vulnerabilities... os_version="3.22" repository="3.22" pkg_num=382026-05-30T13:57:14Z INFO Number of language-specific files num=12026-05-30T13:57:14Z INFO [python-pkg] Detecting vulnerabilities...2026-05-30T13:57:14Z WARN Using severities from other vendors for some vulnerabilities. Read https://trivy.dev/docs/v0.70/guide/scanner/vulnerability#severity-selection for details.

Report Summary

Target	Type	Vulnerabilities	S
secrets			
python:3.9-alpine (alpine 3.22.2)	alpine	2	
usr/local/lib/python3.9/site-packages/pip-23.0.1.dist-info/METADATA	python-pkg	0	
usr/local/lib/python3.9/site-packages/setuptools-79.0.1.dist-info/METADATA	python-pkg	0	

SSH: ec2-35-181-168-61.eu-west-3.compute... 0 0 1

FileEditSelectionViewGoRun

ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

bash - ubuntu

EXPLORER

UBUNTU [SSH: EC2-35-18...

> .cache

> .copilot

> .local

> .ssh

> .vscode-server

\$.bash_logout

\$.bashrc

\$.profile

≡ .sudo_as_admin_succe...

≡ .wget-hsts

! taskboard-stack.yml

OUTLINE

TIMELINE

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS 1

ubuntu@ip-172-31-8-196:~\$ docker images

IMAGE	ID	DISK USAGE	CONTENT SIZE	EXTRA
dockersamples/examplevotingapp_vote:latest@sha256:7102d3b952ec84e3541ee12e7217e320c52aed60b13501c3158f46376a907466	7102d3b952ec	230MB	56.9MB	
dockersamples/examplevotingapp_worker:latest@sha256:c8a8571065479c12567e8e440836e543d185c678cd4781a32b100f71e8f39c86	c8a857106547	288MB	80.3MB	
nginx:alpine	8b1e78743a03	93.6MB	27MB	U
postgres:15-alpine@sha256:df7bca0066e6f60cc3dd32faa70caddec20e2c22b58932f79498e5704b23854a	df7bca0066e6	392MB	110MB	
redis:7-alpine@sha256:6ab0b6e7381779332f97b8ca76193e45b0756f38d4c0dcda72dbb3c32061ab99	6ab0b6e73817	57.8MB	16.8MB	

ubuntu@ip-172-31-8-196:~\$

SSH: ec2-35-181-168-61.eu-west-3.compute...

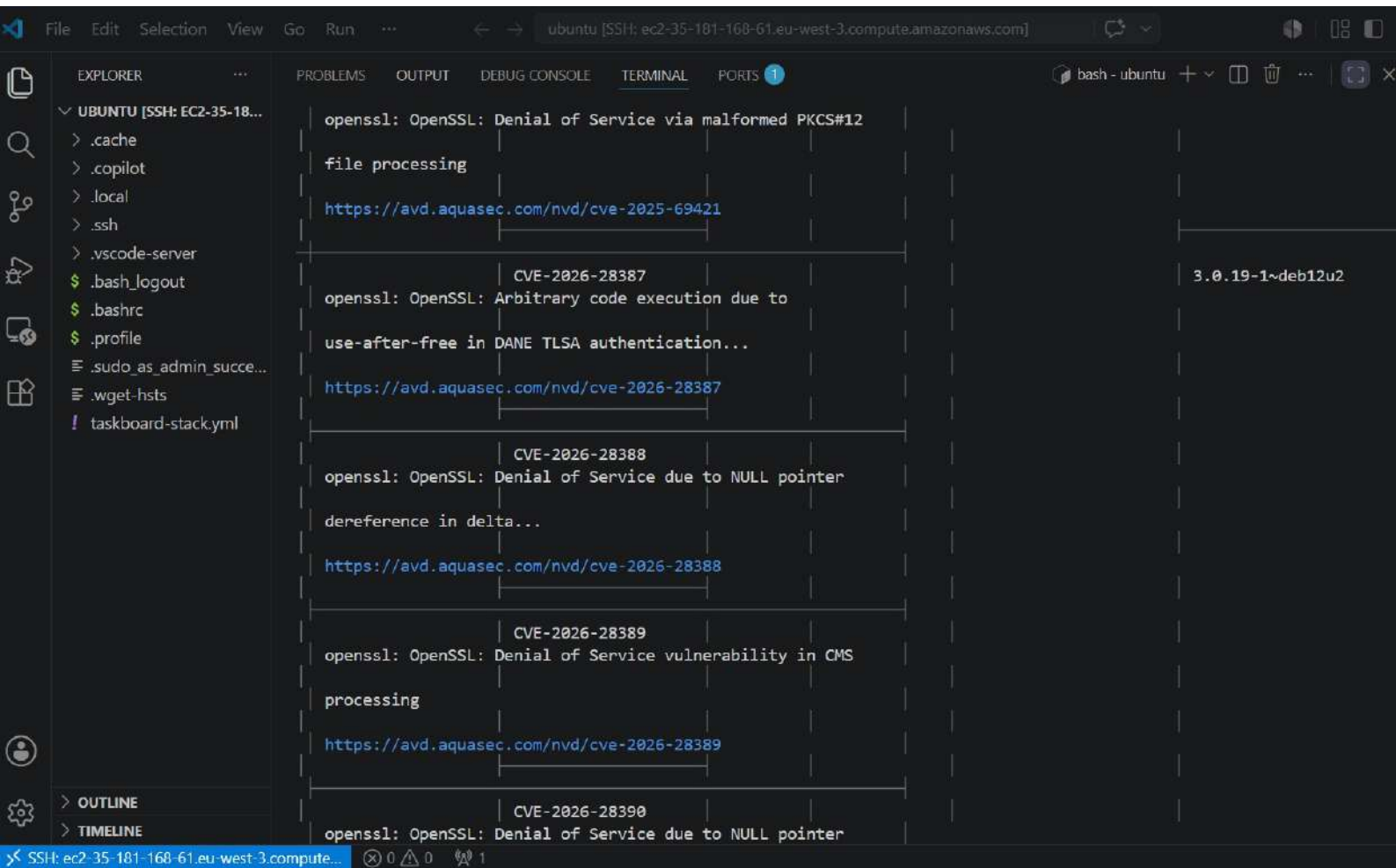
0 0 1

```
trivy image dockersamples/examplevotingapp_vote:latest
```

```
trivy image dockersamples/examplevotingapp_worker:latest
```

```
trivy image postgres:15-alpine
```

```
trivy image redis:7-alpine
```



FileEditSelectionViewGoRun...<→ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER...

UBUNTU [SSH: EC2-35-18...> .cache> .copilot> .local> .ssh> .vscode-server\$.bash_logout\$.bashrc\$.profile≡ .sudo_as_admin_succe...≡ .wget-hsts! taskboard-stack.yml> OUTLINE> TIMELINE

PROBLEMSOUTPUTDEBUG CONSOLETERMINALPORTS 1

bash - ubuntu

libssl1.1	CVE-2025-69421	1.1.1w-0+deb11u1	1.1.1w-0+deb11u5
openssl: OpenSSL: Denial of Service via malformed PKCS#12 file processing			
https://avd.aquasec.com/nvd/cve-2025-69421			
	CVE-2026-28387	affected	
openssl: OpenSSL: Arbitrary code execution due to use-after-free in DANE TLSA authentication...			
https://avd.aquasec.com/nvd/cve-2026-28387			
	CVE-2026-28388		
openssl: OpenSSL: Denial of Service due to NULL pointer dereference in delta...			
https://avd.aquasec.com/nvd/cve-2026-28388			
	CVE-2026-28389		
openssl: OpenSSL: Denial of Service vulnerability in CMS processing			
https://avd.aquasec.com/nvd/cve-2026-28389			

SSH: ec2-35-181-168-61.eu-west-3.compute... 0 0 1

FileEditSelectionViewGoRun...<→ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER...<UBUNTU [SSH: EC2-35-18...>>.cache>.copilot>.local>.ssh>.vscode-server\$.bash_logout\$.bashrc\$.profile≡ .sudo_as_admin_succe...≡ .wget-hsts! taskboard-stack.yml

PROBLEMSOUTPUTDEBUG CONSOLETERMINALPORTS 1bash - ubuntu

• ubuntu@ip-172-31-8-196:~\$ trivy image postgres:15-alpine2026-05-30T14:08:57Z INFO [vuln] Vulnerability scanning is enabled2026-05-30T14:08:57Z INFO [secret] Secret scanning is enabled2026-05-30T14:08:57Z INFO [secret] If your scanning is slow, please try '--scanners vuln' to disable secret scanning2026-05-30T14:08:57Z INFO [secret] Please see https://trivy.dev/docs/v0.70/guide/scanner/secret#recommendation for faster secret detection2026-05-30T14:09:03Z INFO Detected OS family="alpine" version="3.23.4"2026-05-30T14:09:03Z INFO [alpine] Detecting vulnerabilities... os_version="3.23" repository="3.23" pkg_num=452026-05-30T14:09:03Z INFO Number of language-specific files num=12026-05-30T14:09:03Z INFO [gobinary] Detecting vulnerabilities...2026-05-30T14:09:03Z WARN Using severities from other vendors for some vulnerabilities. Read https://trivy.dev/docs/v0.70/guide/scanner/vulnerability#severity-selection for details.

Report Summary

Target	Type	Vulnerabilities	Secrets
postgres:15-alpine (alpine 3.23.4)	alpine	0	-
usr/local/bin/gosu	gobinary	33	-

Legend:

- '-': Not scanned
- '0': Clean (no security findings detected)

usr/local/bin/gosu (gobinary)

Total: 33 (UNKNOWN: 0, LOW: 2, MEDIUM: 16, HIGH: 14, CRITICAL: 1)

< SSH: ec2-35-181-168-61.eu-west-3.compute... 0 0 1

FileEditSelectionViewGoRun...<-->ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER...PROBLEMSOUTPUTDEBUG CONSOLETERMINALPORTS 1

bash - ubuntu + -

UBUNTU [SSH: EC2-35-18...> .cache> .copilot> .local> .ssh> .vscode-server\$.bash_logout\$.bashrc\$.profile≡ .sudo_as_admin_succe...≡ .wget-hsts! taskboard-stack.yml

ubuntu@ip-172-31-8-196:~\$ trivy image redis:7-alpine2026-05-30T14:09:51Z INFO [vuln] Vulnerability scanning is enabled2026-05-30T14:09:51Z INFO [secret] Secret scanning is enabled2026-05-30T14:09:51Z INFO [secret] If your scanning is slow, please try '--scanners vuln' to disable secret scanning2026-05-30T14:09:51Z INFO [secret] Please see https://trivy.dev/docs/v0.70/guide/scanner/secret#recommendation for faster secret detection2026-05-30T14:09:52Z INFO Detected OS family="alpine" version="3.21.7"2026-05-30T14:09:52Z INFO [alpine] Detecting vulnerabilities... os_version="3.21" repository="3.21" pkg_num=192026-05-30T14:09:52Z INFO Number of language-specific files num=0

Report Summary

Target	Type	Vulnerabilities	Secrets
redis:7-alpine (alpine 3.21.7)	alpine	0	-

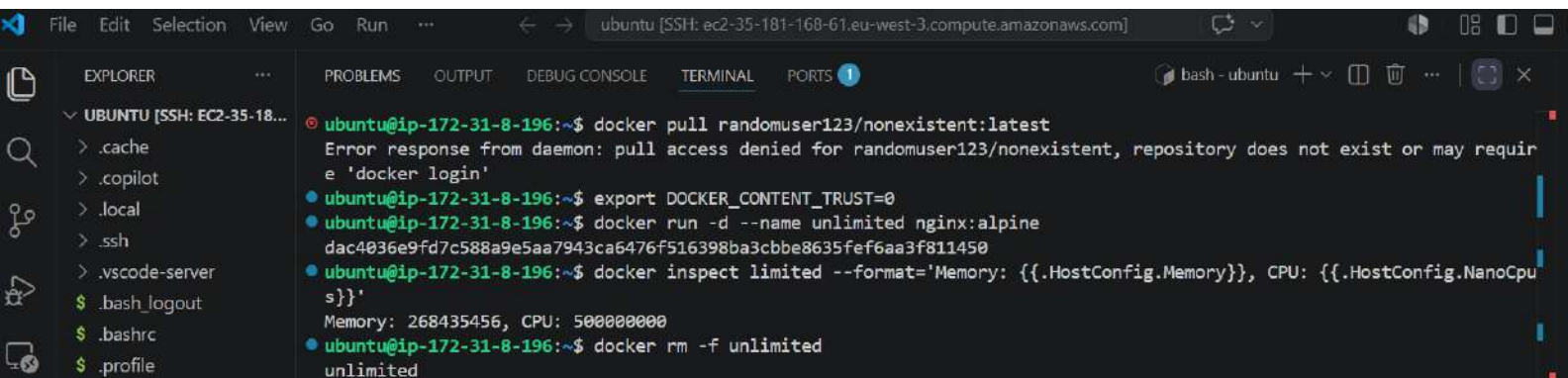
Legend:

- '-': Not scanned
- '0': Clean (no security findings detected)

ubuntu@ip-172-31-8-196:~\$

The screenshot shows a VS Code interface with a terminal window open. The terminal is connected to an Ubuntu instance via SSH. The user has executed several commands to set up Docker and pull a specific image. The output shows the successful pulling of the nginx:stable-alpine image, including layer pull status and a digest.

```
ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]
EXPLORER
  UBUNTU [SSH: EC2-35-181-168-61.eu-west-3.compute.amazonaws.com]
    .cache
    .copilot
    .local
    .ssh
    .vscode-server
    .bash_logout
    .bashrc
    .profile
    .sudo_as_admin_succe...
    .wget-hsts
    ! taskboard-stack.yml
  OUTLINE
  TIMELINE
TERMINAL
  ubuntu@ip-172-31-8-196:~$ echo $DOCKER_CONTENT_TRUST
  ubuntu@ip-172-31-8-196:~$ export DOCKER_CONTENT_TRUST=1
  ubuntu@ip-172-31-8-196:~$ echo $DOCKER_CONTENT_TRUST
  1
  ubuntu@ip-172-31-8-196:~$ docker pull nginx:stable-alpine
  stable-alpine: Pulling from library/nginx
  390b5c71f63d: Pull complete
  b009abd4e3ad: Pull complete
  4db114418873: Pull complete
  36afaf697832: Pull complete
  9b1872bceaab: Pull complete
  d591de949fc4: Pull complete
  43e17fba4a6: Pull complete
  3fda7ca4bac6: Download complete
  1a81e560497f: Download complete
  Digest: sha256:5f979dcfed4ce6461873f087e8c980d6e29b084b9e8776d9704a7e989b5f4898
  Status: Downloaded newer image for nginx:stable-alpine
  docker.io/library/nginx:stable-alpine
  ubuntu@ip-172-31-8-196:~$
```



FileEditSelectionViewGoRun...<=>ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER...<div>UBUNTU [SSH: EC2-35-18...<div>> .cache<div>> .copilot<div>> .local<div>> .ssh<div>> .vscode-server<div>\$.bash_logout<div>\$.bashrc<div>\$.profile<div>≡ .sudo_as_admin_succe...<div>≡ .wget-hsts<div>! taskboard-stack.yml<div>> OUTLINE<div>> TIMELINE

PROBLEMSOUTPUTDEBUG CONSOLETERMINALPORTS 1<div>bash - ubuntu + - - - - -

• ubuntu@ip-172-31-8-196:~\$ docker rm -f limited 2>/dev/null || true<div>docker run -d \<div>--name limited \<div>--memory="256m" \<div>--memory-swap="256m" \<div>--cpus="0.5" \<div>nginx:alpine<div>limited<div>f637ebe07ba51e7840366461d259f1034aff3f7d96164cd2ba6d897025b01e13<div>• ubuntu@ip-172-31-8-196:~\$ docker inspect limited --format='Memory: {{.HostConfig.Memory}}, CPU: {{.HostConfig.NanoCpus}}'<div>Memory: 268435456, CPU: 500000000<div>• ubuntu@ip-172-31-8-196:~\$ docker stats limited --no-stream<table><thead><tr><th>CONTAINER ID</th><th>NAME</th><th>CPU %</th><th>MEM USAGE / LIMIT</th><th>MEM %</th><th>NET I/O</th><th>BLOCK I/O</th><th>PIDS</th></tr></thead><tbody><tr><td>f637ebe07ba5</td><td>limited</td><td>0.00%</td><td>3.238MiB / 256MiB</td><td>1.26%</td><td>936B / 126B</td><td>0B / 12.3kB</td><td>3</td></tr></tbody></table><div>• ubuntu@ip-172-31-8-196:~\$ docker rm -f limited<div>limited<div>❖ ubuntu@ip-172-31-8-196:~\$

FileEditSelectionViewGoRun...←→ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER...

UBUNTU [SSH: EC2-35-18...

> .cache
> .copilot
> .local
> .ssh
> .vscode-server
\$.bash_logout
\$.bashrc
\$.profile
≡ .sudo_as_admin_succe...
≡ .wget-hsts
! security-test-stack.yml
! taskboard-stack.yml

> OUTLINE
> TIMELINE

PROBLEMSOUTPUTDEBUG CONSOLETERMINALPORTS 1

bash - ubuntu +

• ubuntu@ip-172-31-8-196:~\$ nano security-test-stack.yml
• ubuntu@ip-172-31-8-196:~\$ docker stack deploy -c security-test-stack.yml sectest
Since --detach=false was not specified, tasks will be created in the background.
In a future release, --detach=false will become the default.
Creating network sectest_default
Creating service sectest_web
• ubuntu@ip-172-31-8-196:~\$ docker stack services sectest
IDNAMEMODEREPLICASIMAGEPORTS
5c5fj0b1zrr3sectest_webreplicated2/2nginx:alpine*:8080->80/tcp
• ubuntu@ip-172-31-8-196:~\$ docker service inspect sectest_web --pretty | grep -A5 Resources
Resources:
Reservations:
CPU:0.25
Memory:128MiB
Limits:
CPU:0.5
❖ ubuntu@ip-172-31-8-196:~\$

SSH: ec2-35-181-168-61.eu-west-3.compute... 0 0 1

FileEditSelectionViewGoRun...<=>ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER...

UBUNTU [SSH: EC2-35-181-168-61.eu-west-3.compute.amazonaws.com]

> .cache> .copilot> .local> .ssh> .vscode-server\$.bash_logout\$.bashrc\$.profile≡ sudo_as_admin_succe...≡ wget-hsts! security-test-stack.yml! taskboard-stack.yml

> OUTLINE> TIMELINE

PROBLEMSOUTPUTDEBUG CONSOLETERMINALPORTS 1

bash - ubuntu

```
ubuntu@ip-172-31-8-196:~$ docker run --rm nginx:alpine whoami
root
ubuntu@ip-172-31-8-196:~$ docker run --rm alpine ls /dev | wc -l
# Privileged
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
4e0cb4ac63e7: Download complete
a63813f19ccb: Download complete
Digest: sha256:5b10f432ef3da1b8d4c7eb6c487f2f5a8f096bc91145e68878dd4a5019afde11
Status: Downloaded newer image for alpine:latest
15
ubuntu@ip-172-31-8-196:~$ docker run --rm --privileged alpine ls /dev | wc -l
155
ubuntu@ip-172-31-8-196:~$ docker run --rm --user 1000:1000 nginx:alpine id
uid=1000 gid=1000 groups=1000
ubuntu@ip-172-31-8-196:~$ docker run --rm --user nobody alpine id
uid=65534(nobody) gid=65534(nobody) groups=65534(nobody)
ubuntu@ip-172-31-8-196:~$ docker run --rm --cap-drop=ALL nginx:alpine cat /proc/1/status | grep Cap
CapInh: 0000000000000000
CapPrm: 0000000000000000
CapEff: 0000000000000000
CapBnd: 0000000000000000
CapAmb: 0000000000000000
ubuntu@ip-172-31-8-196:~$
```

File Edit Selection View Go Run ... ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

UBUNTU [SSH: EC2-35-181-168-61.eu-west-3.compute.amazonaws.com]

- > .cache
- > .copilot
- > .local
- > .ssh
- > .vscode-server
- \$.bash_logout
- \$.bashrc
- \$.profile
- ≡ .sudo_as_admin_succe...
- ≡ .wget-hsts
- ! security-test-stack.yml
- ! taskboard-stack.yml

```
ubuntu@ip-172-31-8-196:~$ docker run --rm --cap-drop=ALL --cap-add=NET_BIND_SERVICE nginx:alpine cat /proc/1/status
grep Cap
CapInh: 0000000000000000
CapPrm: 00000000000000400
CapEff: 00000000000000400
CapBnd: 00000000000000400
CapAmb: 0000000000000000
ubuntu@ip-172-31-8-196:~$
```

FileEditSelectionViewGoRun...<→ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER

UBUNTU [SSH: EC2-35-18...> .cache> .copilot> .local> .ssh> .vscode-server\$.bash_logout\$.bashrc\$.profile≡ .sudo_as_admin_succe...≡ .wget-hsts! security-test-stack.yml! taskboard-stack.yml

OUTLINETIMELINE

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

bash - ubuntu

+

×

🗑

⋮

🔍

✕

ubuntu@ip-172-31-8-196:~\$ docker rm -f readonly-testreadonly-test

ubuntu@ip-172-31-8-196:~\$ docker run -d \

--name readonly-test \

--read-only \

--tmpfs /var/cache/nginx \

--tmpfs /var/run \

nginx:alpine

aef5f51f641b51b30625d15d03625fec935aa08ba3ad3e179bc8d9dac2c62c99

ubuntu@ip-172-31-8-196:~\$ docker ps | grep readonly-test

aef5f51f641b nginx:alpine "/docker-entrypoint..." 46 seconds ago Up 46 seconds 80/tcp readonly-test

ubuntu@ip-172-31-8-196:~\$ docker exec readonly-test touch /test-file || echo "Write blocked - as expected!"

touch: /test-file: Read-only file system

Write blocked - as expected!

ubuntu@ip-172-31-8-196:~\$ docker rm -f readonly-testreadonly-test

ubuntu@ip-172-31-8-196:~\$

SSH: ec2-35-181-168-61.eu-west-3.compute...001

File Edit Selection View Go Run ... ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER

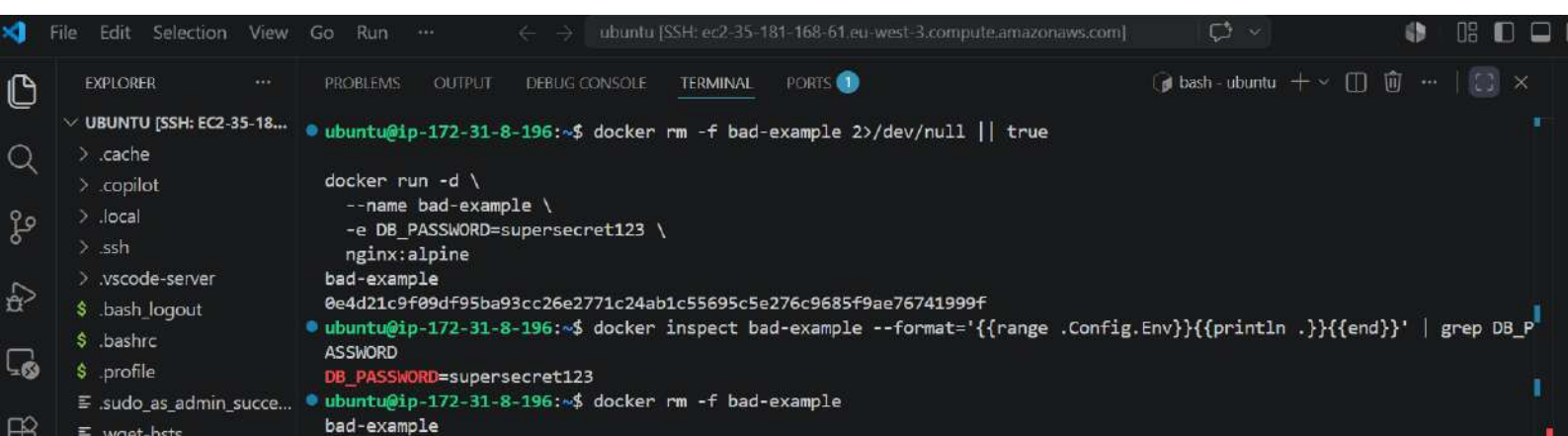
UBUNTU [SSH: EC2-35-18...]

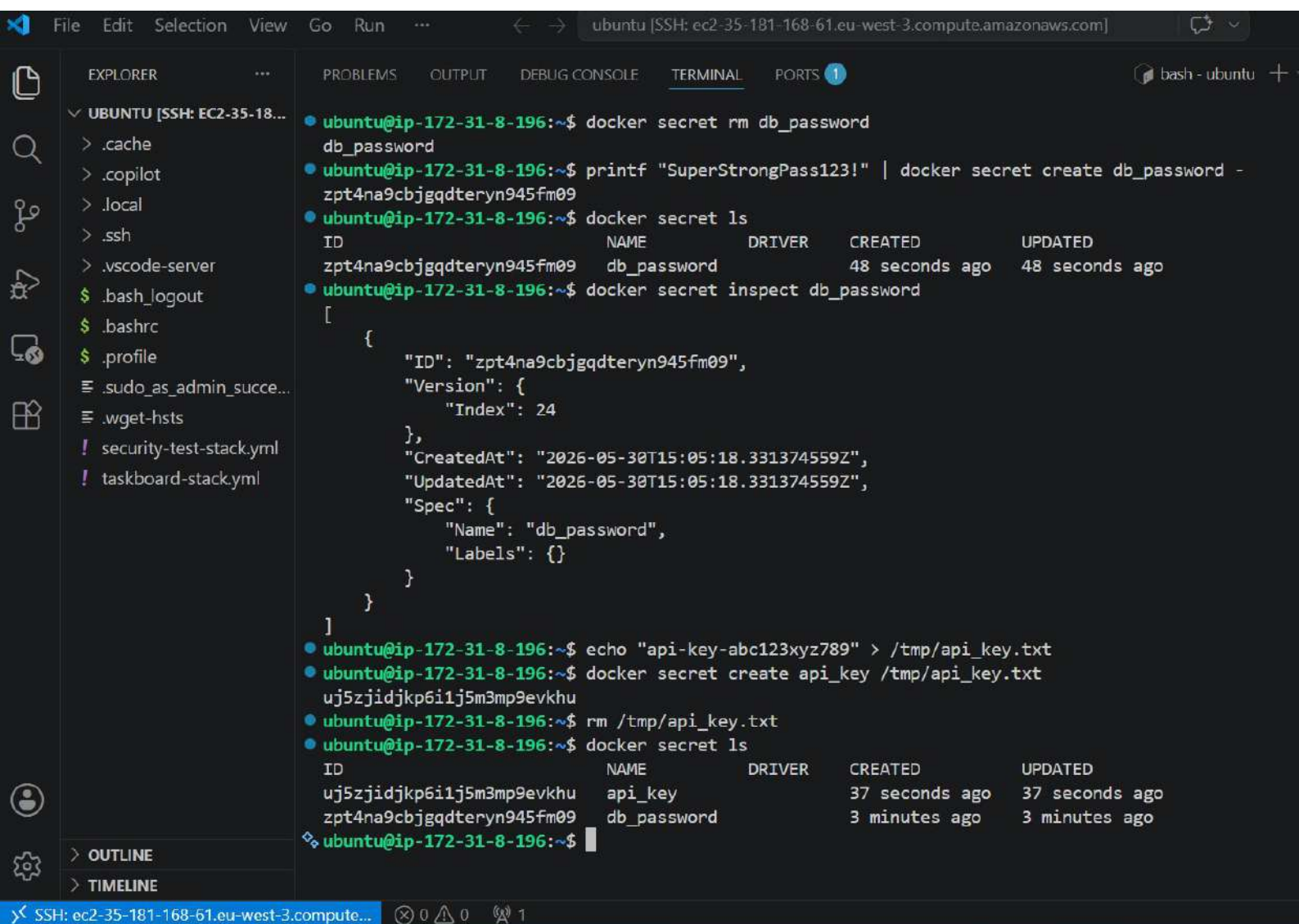
- > .cache
- > .copilot
- > .local
- > .ssh
- > .vscode-server
- \$.bash_logout
- \$.bashrc
- \$.profile
- ≡ .sudo_as_admin_succe...
- ≡ .wget-hsts
- ! security-test-stack.yml

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

bash - ubuntu

```
ubuntu@ip-172-31-8-196:~$ docker run -d --name secure-nginx \
--user 101:101 \
--cap-drop=ALL \
--cap-add=NET_BIND_SERVICE \
--read-only \
--tmpfs /var/cache/nginx \
--tmpfs /var/run \
--security-opt=no-new-privileges \
nginx:alpine
e6a2bab90596fc7fc2fb55c93ae40187f7bdd298303de9de7f9c8c9d051aac63
ubuntu@ip-172-31-8-196:~$ docker inspect secure-nginx --format='User: {{.Config.User}}, ReadOnly: {{.HostConfig.ReadOnlyRootfs}}'
User: 101:101, ReadOnly: true
ubuntu@ip-172-31-8-196:~$ docker rm -f secure-nginx
secure-nginx
```





```
File Edit Selection View Go Run ... ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com] bash - ubuntu

EXPLORER
  UBUNTU [SSH: EC2-35-181-168-61.eu-west-3.compute.amazonaws.com]
    .cache
    .copilot
    .local
    .ssh
    .vscode-server
    .bash_logout
    .bashrc
    .profile
    .sudo_as_admin_successful
    .wget-hsts
    secrets-demo.yml
    security-test-stack.yml
    taskboard-stack.yml

TERMINAL
  ubuntu@ip-172-31-8-196:~$ nano secrets-demo.yml
  ubuntu@ip-172-31-8-196:~$ docker stack deploy -c secrets-demo.yml secretsdemo
    Since --detach=false was not specified, tasks will be created in the background.
    In a future release, --detach=false will become the default.
    Creating network secretsdemo_default
    Creating service secretsdemo_api
  ubuntu@ip-172-31-8-196:~$ sleep 5
  ubuntu@ip-172-31-8-196:~$ docker exec $(docker ps -q --filter name=secretsdemo_api) ls -la /run/secrets/
    total 16
    drwxr-xr-x  2 root root      4096 May 30 15:11 .
    drwxr-xr-x  1 root root      4096 May 30 15:11 ..
    -r--r--r--  1 root root       21 May 30 15:11 api_key
    -r--r--r--  1 root root      19 May 30 15:11 db_password
  ubuntu@ip-172-31-8-196:~$ docker exec $(docker ps -q --filter name=secretsdemo_api) cat /run/secrets/db_password
    SuperStrongPass123!
  ubuntu@ip-172-31-8-196:~$ docker exec $(docker ps -q --filter name=secretsdemo_api) env | grep -i password
    password
    DB_PASSWORD_FILE=/run/secrets/db_password
  ubuntu@ip-172-31-8-196:~$ docker info | grep -A10 "Swarm:"
    Swarm: active
      NodeID: tqvg3mofuxjlm38yoc3gesejz
      Is Manager: true
      ClusterID: ln0d0u8h02ghczjw8gif0wje3
      Managers: 1
      Nodes: 1
      Data Path Port: 4789
      Orchestration:
        Task History Retention Limit: 5
      Raft:
        Snapshot Interval: 10000
```

File Edit Selection View Go Run ... ubuntu [SSH: ec2-35-181-168-61.eu-west-3.compute.amazonaws.com]

EXPLORER ...

UBUNTU [SSH: EC2-35-18...

> .cache

> .copilot

> .local

> .ssh

> .vscode-server

PROBLEMS

OUTPUT

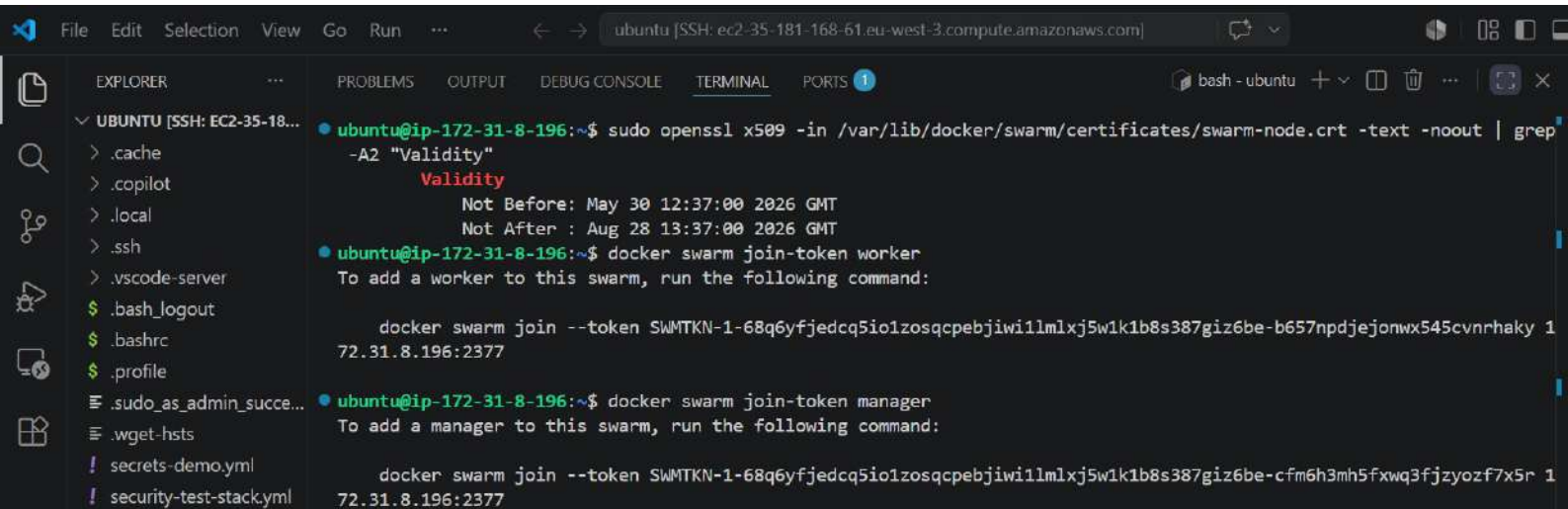
DEBUG CONSOLE

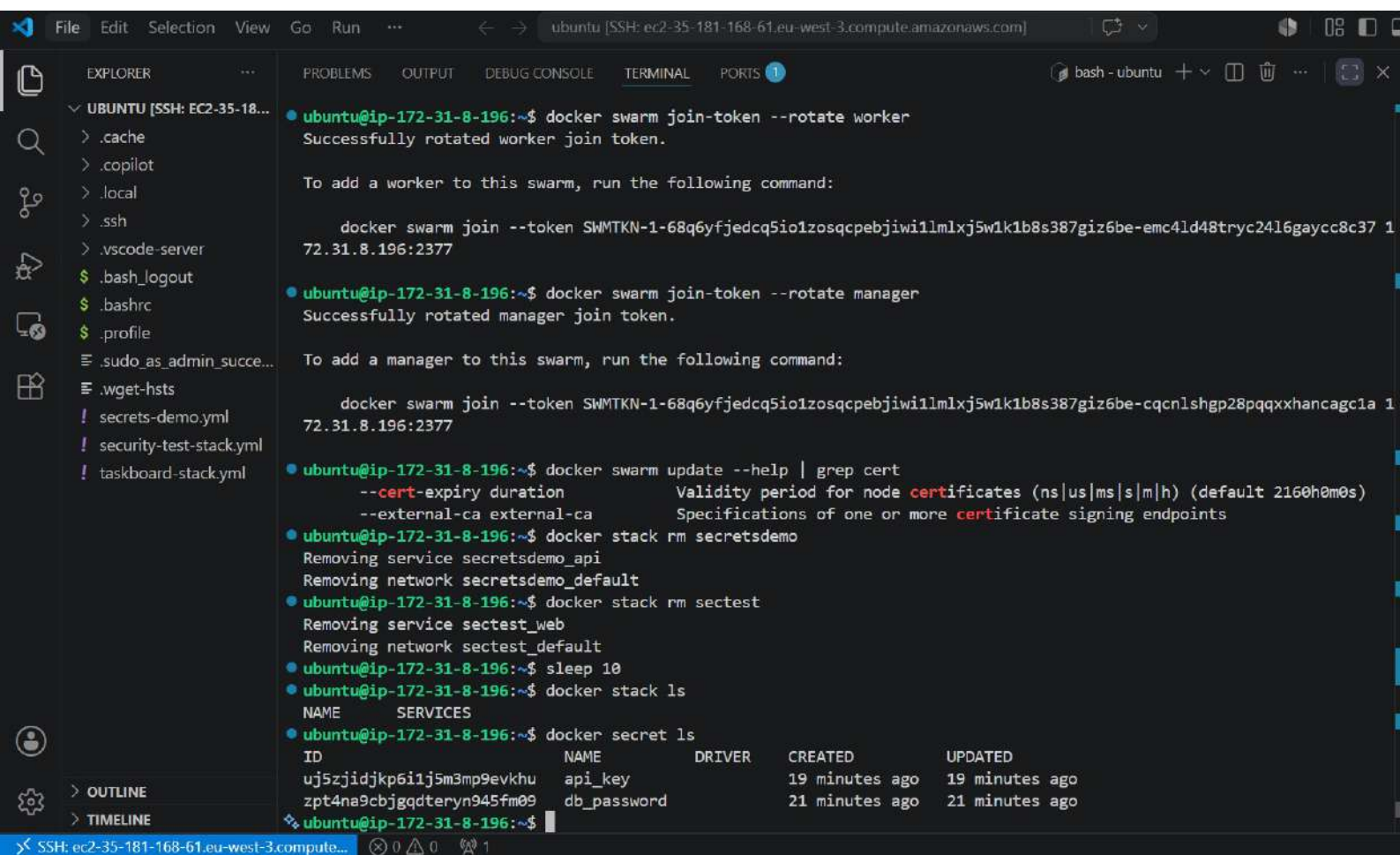
TERMINAL

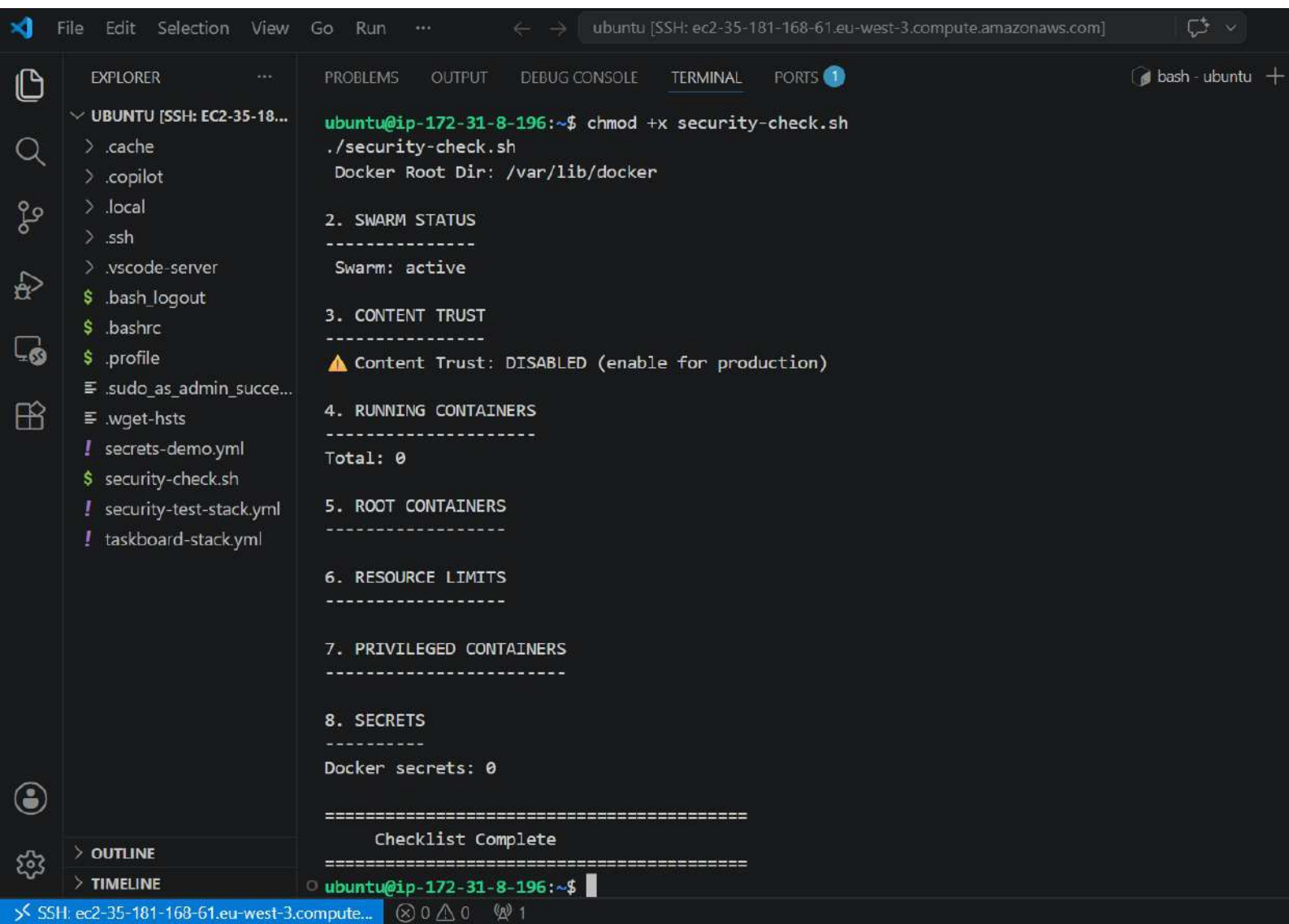
PORTS 1

bash - ubuntu

ubuntu@ip-172-31-8-196:~\$ sudo ls -la /var/lib/docker/swarm/certificates/
total 20
drwxr-xr-x 2 root root 4096 May 30 13:37 .
drwx----- 5 root root 4096 May 30 13:37 ..
-rw-r--r-- 1 root root 826 May 30 13:37 swarm-node.crt
-rw----- 1 root root 316 May 30 13:37 swarm-node.key
-rw-r--r-- 1 root root 550 May 30 13:37 swarm-root-ca.crt







EC2 > Instances

EC2

Dashboard

AWS Global View

Events

Instances

Instances

Instance Types

Successfully initiated termination (deletion) of i-041d4d7ba69dbace6

Instances (1/1)

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Saved filter sets

Choose filter set

Find Instance by attribute or tag (case-sensitive)

< 1 >

☒	Name	Instance ID	Instance state	Instance type	Status check	Availability Zone
☒	SwarmManager	i-041d4d7ba69dbace6	Terminated	t3.small	-	eu-west-3a