

School: CLOUDBOOSTA	Course: DevOps
Name: AKOSA SAMUEL ONYEJEKWE	Lab-1_ Docker-2
	Week: 5 of 8
Student No: CBALC0168	Issue Date: May 30, 2026

Tasks

1. Create Swarm Manager EC2 Instance
2. Configure Swarm Security Group
3. Create Swarm Worker EC2 Instance
4. Connect to EC2 Instances using VS Code
5. Initialize Docker Swarm Cluster
6. Join Worker Node to Swarm Cluster
7. Verify Swarm Node Membership
8. Deploy TaskBoard Application Stack
9. Verify TaskBoard Service Deployment
10. Access TaskBoard Web Application
11. Scale TaskBoard Services
12. Verify Service Replication
13. Inspect Swarm Services
14. Review Service Logs
15. Force Service Update
16. Verify Load Balancing
17. Drain Worker Node
18. Verify Container Rescheduling
19. Remove TaskBoard Stack
20. Leave Docker Swarm Cluster
21. Terminate AWS Resources
22. Validate Docker Swarm Orchestration Workflow

Goal

To deploy a Docker Swarm cluster on AWS EC2 infrastructure, configure secure Swarm networking, deploy a multi-container TaskBoard application, perform service scaling and orchestration operations, validate Swarm high availability mechanisms, and implement cloud-native container orchestration using Docker Swarm.

Result

- Swarm Manager EC2 instance created successfully
- Swarm Worker EC2 instance created successfully

- Swarm security group configured successfully
- Docker installed successfully on all nodes
- Docker Swarm initialized successfully
- Worker node joined Swarm cluster successfully
- Swarm node membership verified successfully
- TaskBoard application stack deployed successfully
- TaskBoard services created successfully
- TaskBoard web application accessible successfully
- Service scaling operations completed successfully
- Swarm service replication validated successfully
- Service inspection workflow validated successfully
- Service logging workflow validated successfully
- Rolling update workflow validated successfully
- Swarm load balancing validated successfully
- Node draining operation validated successfully
- Automatic service rescheduling validated successfully
- Stack removal workflow validated successfully
- Swarm cluster cleanup validated successfully
- Docker orchestration workflow implemented successfully

Action

Step 1: Created Swarm Manager EC2 Instance

Opened:

AWS Console → EC2 → Launch Instance

Configured:

- Instance Name: SwarmManager
- Operating System: Ubuntu Server 24.04 LTS
- Instance Type: t3.small
- Key Pair: cba_keypair
- Security Group: swarm_sg

Verified:

- EC2 instance launched successfully
- Instance entered Running state successfully

Evidence available in AWS EC2 launch screenshots.

Step 2: Configured Swarm Security Group

Configured inbound rules:

Port	Protocol	Purpose
22	TCP	SSH
2377	TCP	Swarm Management
7946	TCP	Node Communication
7946	UDP	Node Communication
4789	UDP	Overlay Network
8080	TCP	TaskBoard Web Application

Verified security group rules created successfully.

Step 3: Created Swarm Worker EC2 Instance

Opened:

AWS Console → EC2 → Launch Instance

Configured:

- Instance Name: SwarmWorker
- Operating System: Ubuntu Server 24.04 LTS
- Instance Type: t3.small
- Key Pair: cba_keypair
- Security Group: swarm_sg

Verified:

- Worker instance launched successfully
- Worker instance entered Running state successfully

Step 4: Connected to Swarm Manager using VS Code

Configured:

- Remote SSH Extension

- SSH Client Connection
- Private Key Authentication

Verified:

- Successful remote connection to Swarm Manager
- Terminal access established successfully

Step 5: Verified Docker Installation

Executed:

```
docker --version
```

Verified:

```
Docker version 29.5.2
```

Confirmed Docker installed successfully on Swarm Manager.

Step 6: Initialized Docker Swarm Cluster

Executed:

```
MANAGER_IP=172.31.8.196
```

```
docker swarm init --advertise-addr $MANAGER_IP
```

Verified:

- Swarm initialized successfully
- Join token generated successfully

Step 7: Verified Swarm Manager Node

Executed:

```
docker node ls
```

Verified:

- Manager node present
- Leader status assigned
- Node availability active

Step 8: Joined Worker Node to Swarm

Executed on Worker Node:

```
docker swarm join --token <join-token> \
<manager-ip>:2377
```

Verified:

This node joined a swarm as a worker.

Confirmed worker successfully joined cluster.

Step 9: Verified Cluster Membership

Executed:

```
docker node ls
```

Verified:

- Swarm Manager Ready
- Swarm Worker Ready
- Leader status active

Confirmed cluster membership functioning successfully.

Step 10: Deployed TaskBoard Application Stack

Created:

```
taskboard-stack.yml
```

Executed:

```
docker stack deploy -c taskboard-stack.yml taskboard
```

Verified:

- Stack deployed successfully
- Swarm services created successfully

Step 11: Verified TaskBoard Web Application

Opened browser:

`http://<public-ip>:8080`

Verified:

Cats vs Dogs!

Application accessible successfully through Swarm ingress routing.

Step 12: Verified Stack Services

Executed:

```
docker stack services taskboard
```

Verified services:

- taskboard_web
- taskboard_api
- taskboard_postgres
- taskboard_redis

Confirmed all services deployed successfully.

Step 13: Scaled TaskBoard API Service

Executed:

```
docker service scale taskboard_api=4
```

Verified:

4/4 running

Confirmed service scaling completed successfully.

Step 14: Inspected Swarm Services

Executed:

```
docker service inspect taskboard_web
```

Verified:

- Service configuration
- Update strategy
- Rollback policy
- Replica settings

Confirmed service inspection workflow functioning successfully.

Step 15: Reviewed Service Logs

Executed:

```
docker service logs taskboard_web
```

and

```
docker service logs taskboard_api
```

Verified:

- Application startup logs
- API service logs
- Database connectivity messages

Confirmed centralized logging functioning successfully.

Step 16: Executed Rolling Update

Executed:

```
docker service update --force taskboard_web
```

Verified:

```
4 out of 4 tasks running
```

Confirmed rolling update completed successfully.

Step 17: Verified Swarm Load Balancing

Executed multiple requests:

```
curl http://<public-ip>:8080
```

Verified:

- Application remained available
- Requests served successfully
- Swarm routing mesh functioning successfully

Step 18: Drained Worker Node

Executed:

```
docker node update --availability drain ip-172-31-8-81
```

Verified:

- Worker node drained successfully
- Tasks migrated automatically

Step 19: Verified Automatic Rescheduling

Executed:

```
docker service ps taskboard_web
```

and

```
docker service ps taskboard_api
```

Verified:

- Running containers reassigned
- Service availability maintained
- Swarm self-healing functioning successfully

Step 20: Removed TaskBoard Stack

Executed:

```
docker stack rm taskboard
```

Verified:

- Services removed successfully
- Network removed successfully
- Containers removed successfully

Step 21: Left Docker Swarm Cluster

Executed on Manager:

```
docker swarm leave --force
```

Executed on Worker:

```
docker swarm leave
```

Verified:

Node left the swarm.

Cluster cleanup completed successfully.

Summary

- Docker Swarm cluster configured successfully
- Swarm Manager and Worker nodes deployed successfully
- Secure Swarm networking configured successfully
- TaskBoard application stack deployed successfully
- Multi-container orchestration validated successfully
- Service scaling workflow validated successfully
- Rolling update workflow validated successfully
- Load balancing workflow validated successfully
- Automatic recovery workflow validated successfully
- Node draining and task migration validated successfully
- Stack cleanup workflow validated successfully
- Cloud-native Docker orchestration implemented successfully

Lab Objective Achieved

Implemented Docker Swarm cluster deployment, EC2-based container orchestration, secure Swarm networking, multi-node service deployment, TaskBoard application hosting, service scaling, rolling updates, centralized service management, load balancing, self-healing container recovery, node draining operations, and cloud-native Docker orchestration successfully within AWS EC2 and Docker Swarm infrastructure.

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

SwarmManager

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

Recents

Quick Start



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

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0adb8ca49015e0901 (64-bit (x86)) / ami-0d4488a9dfc5a29c6 (64-bit (Arm))
Virtualization: hvm FFA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EB5 General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Publish Date	Username	
64-bit (x...)	ami-0adb8ca49015e0901	2026-04-24	ubuntu	Verified provider

Instance type [Info](#) [Get advice](#)

Instance type

t3.small

Family: t3 2 vCPU 2 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.042 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0271 USD per Hour
On-Demand RHEL base pricing: 0.0524 USD per Hour
On-Demand SUSE base pricing: 0.0546 USD per Hour
On-Demand Linux base pricing: 0.0236 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

cba_keypair

Create new key pair

Network settings [Info](#)

Edit

Network [Info](#)

vpc-07e61444329818154

Subnet [Info](#)

Name

SwarmManager

Add additional tags

Summary

Number of instances [Info](#)

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-0adb8ca49015e0901

Virtual server type (instance type)

t3.small

Firewall (security group)

swarm_sg

Cancel

Launch instance

Preview code

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-0adb8ca49015e0901

▼ 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
Select security groups

[Compare security group rules](#)

swarm_sg sg-074a0a69742031c5c

VPC: vpc-07e61444329b18154

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▼ Configure storage [Info](#)

[Advanced](#)

1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted

[Add new volume](#)

The selected AMI contains instance store volumes, however the instance does not allow any instance store volumes. None of the instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

File systems

☐ S3 Files - new

☐ EFS

☐ FSx

☒ None

▼ Advanced details [Info](#)

Domain join directory [Info](#)

Select

[Create new directory](#)

IAM instance profile [Info](#)

Select

[Create new IAM profile](#)

Hostname type [Info](#)

IP name

DNS Hostname [Info](#)

☒ Enable IP name (IPv4 (A record) DNS requests

☒ Enable resource-based IPv4 (A record) DNS requests

☐ Enable resource-based IPv6 (AAAA record) DNS requests

Instance auto-recovery [Info](#)

Select

Shutdown behavior [Info](#)

Stop

Stop - Hibernate behavior [Info](#)

Select

Termination protection [Info](#)

Select

Stop protection [Info](#)

Select

Detailed CloudWatch monitoring [Info](#)

Select

Credit specification [Info](#)

Unlimited

[Additional charges apply](#)

Placement group [Info](#)

Select

[Create new placement group](#)

EBS-optimized instance [Info](#)

Virtual server type (instance type)

t3.small

Firewall (security group)

swarm_sg

[Cancel](#)

[Launch Instance](#)

enable

Instance bandwidth configuration [Info](#)

Select

Purchasing option [Info](#)

☒ None

☐ Interruptible Capacity Reservations

Launch instances for your interruptible Capacity Reservations

☐ Spot instances

Request Spot instances at the Spot price, capped at the On-Demand price

Capacity reservation [Info](#)

Select

Tenancy [Info](#)

Select

RAM disk ID [Info](#)

Select

Kernel ID [Info](#)

Select

Nitro Enclave [Info](#)

Select

CPU options - update [Info](#)

Configure CPUs for your instance to optimize performance and save on licensing costs.

☒ Use default CPU options

☐ Specify CPU options

Default active vCPUs

2

Total vCPUs

2

Nested virtualization [Info](#)

Select

Nested virtualization is not supported with the selected instance type.

Metadata accessible [Info](#)

Enabled

Metadata IPv6 endpoint [Info](#)

Select

Metadata version [Info](#)

V2 only (token required)

⚠ For V2 requests, you must include a session token in all instance metadata requests. Applications or agents that use V1 for instance metadata access will break.

Metadata response hop limit [Info](#)

2

Allow tags in metadata [Info](#)

Select

User data - optional [Info](#)

Upload a file with your user data or enter it in the field.

Choose file

```
apt-get install -y ca-certificates curl
install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | gpg --
dearmor -o /etc/apt/keyrings/docker.gpg
chmod a+r /etc/apt/keyrings/docker.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu $(. /etc/os-release && echo
"$VERSION_CODENAME") stable" | tee /etc/apt/sources.list.d/docker.list >
/dev/null
apt-get update
apt-get install -y docker-ce docker-ce-cli containerd.io
usermod -aG docker ubuntu
systemctl enable docker
systemctl start docker
```

☐ User data has already been base64 encoded

EC2 > Instances

EC2

Dashboard

AWS Global View

Events

Instances

Instances (3) Info

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	Running	t3.small	3/3 checks passed	eu-west-3a

EC2

<

Dashboard

AWS Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Capacity Manager

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

▼ Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

▼ Load Balancing

Load Balancers

Target Groups

Trust Stores

▼ Auto Scaling

Auto Scaling Groups

Settings

Inbound rules

Outbound rules

Sharing

VPC associations

Related resources - new

Tags

Inbound rules (6)

Manage tags

Edit inbound rules

< 1 >

☐	Name	Security group rule ID	IP version	Type	Protocol
☐	-	sgr-03e33245180859538	-	Custom TCP	TCP
☐	-	sgr-0e5dcfd1fd1ccff13	-	Custom UDP	UDP
☐	-	sgr-0233386b83843cf48	-	Custom TCP	TCP
☐	-	sgr-095ae870ad8eff1	IPv4	SSH	TCP
☐	-	sgr-0e463d63fa2ec801e	IPv4	Custom TCP	TCP
☐	-	sgr-0a11be23ca5f127f1	-	Custom UDP	UDP

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

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

Name

SwarmWorker

Add additional tags

Application and OS Images (Amazon Machine Image)

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

Recents

Quick Start



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

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0adb8ca49015e0901 (64-bit (x86)) / ami-0d4488a9dfc5a29c6 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EB5 General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture

64-bit (x...

AMI ID

ami-0adb8ca49015e0901

Publish Date

2026-04-24

Username

ubuntu

Verified provider

Instance type

Instance type

t3.small

Family: t3 2 vCPU 2 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.042 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0271 USD per Hour
On-Demand RHEL base pricing: 0.0524 USD per Hour
On-Demand SUSE base pricing: 0.0546 USD per Hour
On-Demand Linux base pricing: 0.0236 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login)

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

cba_keypair

Create new key pair

Network settings

Network

vpc-07e61444329818154

Subnet

No preference (Default subnet in any availability zone)

Auto-assign public IP

Summary

Number of instances

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64...read more
ami-0adb8ca49015e0901

Virtual server type (instance type)

t3.small

Firewall (security group)

swarm_sg

Cancel

Launch instance

Preview code

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group

☒ Select existing security group

Common security groups [Info](#)

Select security groups

Compare security group rules

swarm_sg sg-074a0a69742031c5c

VPC: vpc-07e61444329b18154

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▼ Configure storage [Info](#)

Advanced

1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted

[Add new volume](#)

The selected AMI contains instance store volumes, however the instance does not allow any instance store volumes. None of the instance store volumes from the AMI will be accessible from the instance.

Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

File systems

☐ S3 Files - new

☐ EFS

☐ FSx

☒ None

▼ Advanced details [Info](#)

Domain join directory [Info](#)

Select

Create new directory [↗](#)

IAM instance profile [Info](#)

Select

Create new IAM profile

Hostname type [Info](#)

IP name

DNS Hostname [Info](#)

☒ Enable IP name (IPv4 (A record) DNS requests

☒ Enable resource-based IPv4 (A record) DNS requests

☐ Enable resource-based IPv6 (AAAA record) DNS requests

Instance auto-recovery [Info](#)

Select

Shutdown behavior [Info](#)

Stop

Stop - Hibernate behavior [Info](#)

Select

Termination protection [Info](#)

Select

Stop protection [Info](#)

Select

Detailed CloudWatch monitoring [Info](#)

Select

Credit specification [Info](#)

Unlimited

[Additional charges apply](#)

Placement group [Info](#)

Select

Create new placement group [↗](#)

EBS-optimized instance [Info](#)

enable

Instance bandwidth configuration [Info](#)

Select

Purchasing option [Info](#)

☒ None

☐ Interruptible Capacity Reservations

Launch instances for your interruptible Capacity Reservations

☐ Spot instances

Request Spot instances at the Spot price, capped at the On-Demand price

Capacity reservation [Info](#)

Select

Tenancy [Info](#)

Select

RAM disk ID [Info](#)

Select

Kernel ID [Info](#)

Select

Nitro Enclave [Info](#)

Select

CPU options - update [Info](#)

Configure CPUs for your instance to optimize performance and save on licensing costs.

☒ Use default CPU options

☐ Specify CPU options

Default active vCPUs

2

Total vCPUs

2

Nested virtualization [Info](#)

Select

Nested virtualization is not supported with the selected instance type.

Metadata accessible [Info](#)

Enabled

Metadata IPv6 endpoint [Info](#)

Select

Metadata version [Info](#)

V2 only (token required)

 For V2 requests, you must include a session token in all instance metadata requests. Applications or agents that use V1 for instance metadata access will break.

Metadata response hop limit [Info](#)

2

Allow tags in metadata [Info](#)

Select

User data - optional [Info](#)

Upload a file with your user data or enter it in the field.

 Choose file

```
apt-get install -y ca-certificates curl
install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | gpg --dearmor
-o /etc/apt/keyrings/docker.gpg
chmod a+r /etc/apt/keyrings/docker.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu $(. /etc/os-release && echo
"$VERSION_CODENAME") stable" | tee /etc/apt/sources.list.d/docker.list >
/dev/null
apt-get update
apt-get install -y docker-ce docker-ce-cli containerd.io
usermod -aG docker ubuntu
systemctl enable docker
systemctl start docker
```

☐ User data has already been base64 encoded

Connect

Info

Connect to an instance using the browser-based client.

Instance ID

i-041d4d7ba69dbace6 (SwarmManager)

VPC ID

vpc-07e61444329818154

Security groups

sg-074a0a69742031c5c (swarm_sg)

IAM role

-

EC2 Instance Connect

SSM Session Manager

SSH client

EC2 serial console

Instance ID

i-041d4d7ba69dbace6 (SwarmManager)

1. Open an SSH client.

2. Locate your private key file. The key used to launch this instance is cba_keypair.pem

3. Run this command, if necessary, to ensure your key is not publicly viewable.

chmod 400 "cba_keypair.pem"

4. Connect to your instance using its Public DNS:

ec2-35-181-168-61.eu-west-3.compute.amazonaws.com

Example:

ssh -i "cba_keypair.pem" ubuntu@ec2-35-181-168-61.eu-west-3.compute.amazonaws.com

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

Connect

Info

Connect to an instance using the browser-based client.

Instance ID

i-0e688238088b9eb1d (SwarmWorker)

VPC ID

vpc-07e61444329818154

Security groups

sg-074a0a69742031c5c (swarm_sg)

IAM role

-

EC2 Instance Connect

SSM Session Manager

SSH client

EC2 serial console

Instance ID

i-0e688238088b9eb1d (SwarmWorker)

1. Open an SSH client.

2. Locate your private key file. The key used to launch this instance is cba_keypair.pem

3. Run this command, if necessary, to ensure your key is not publicly viewable.

chmod 400 "cba_keypair.pem"

4. Connect to your instance using its Public DNS:

ec2-13-38-102-221.eu-west-3.compute.amazonaws.com

Example:

ssh -i "cba_keypair.pem" ubuntu@ec2-13-38-102-221.eu-west-3.compute.amazonaws.com

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

CloudShell

Feedback

Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

EC2

<

Dashboard

AWS Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

EC2 > Instances > i-041d4d7ba69dbace6

Instance summary for i-041d4d7ba69dbace6 (SwarmManager) Info

Refresh

Connect

Instance state

Actions

Updated less than a minute ago

Instance ID

i-041d4d7ba69dbace6

IPv6 address

—

Hostname type

IP name: ip-172-31-8-196.eu-west-3.compute.internal

Public IPv4 address

35.181.168.61 | open address

Instance state

Running

Private IP DNS name (IPv4 only)

ip-172-31-8-196.eu-west-3.compute.internal

Private IPv4 addresses

172.31.8.196

Public DNS

ec2-35-181-168-61.eu-west-3.compute.amazonaws.com | open address

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.CO...]

.cache

.copilot

.ssh

.vscode-server

.bash_logout

.bashrc

.profile

.wget-hsts

OUTLINE

TIMELINE

PROBLEMSOUTPUTTERMINAL

bash - ubuntu

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

- ubuntu@ip-172-31-8-196:~\$ docker --version
Docker version 29.5.2, build 79eb04c
- ubuntu@ip-172-31-8-196:~\$ MANAGER_IP=172.31.8.196
- ubuntu@ip-172-31-8-196:~\$ docker swarm init --advertise-addr \$MANAGER_IP
Swarm initialized: current node (zwusnosh1lg0uajsrami58fnz) is now a manager.

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

```
docker swarm join --token SWMTKN-1-0b8y8yqcesjc4xb025fqfzan87g5syrnafbyvf9j4uy04ayqms-1ebsvdmj99x63sn5f0jtajepb 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:~\$

meet.google.com is sharing your screen.

Stop sharing

Hide

EXPLORER

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

> .cache

> .copilot

> .ssh

> .vscode-server

\$.bash_logout

\$.bashrc

\$.profile

≡ .wget-hsts

> OUTLINE

> TIMELINE

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

bash - ubuntu

+

▾

📄

🗑

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

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
qu2f5eqxzq2xwscb06umdm3j5	ip-172-31-8-81	Ready	Active		29.5.2
zwusnosh1lg0uajsrmi58fnz *	ip-172-31-8-196	Ready	Active	Leader	29.5.2

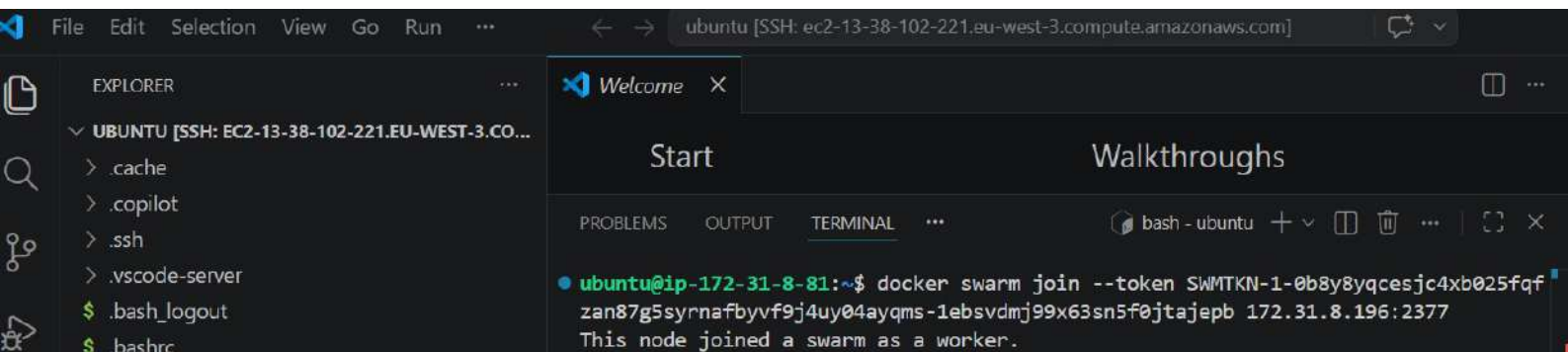
ubuntu@ip-172-31-8-196:~\$

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

✖ 0

⚠ 0

🔍 0



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

≡ .wget-hsts

! taskboard-stack.yml

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

bash - ubuntu + ▾

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

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
qu2f5eqxzq2xwscb06umdm3j5	ip-172-31-8-81	Ready	Active		29.5.2
zwusnosh1lg0uajsrmi58fnz *	ip-172-31-8-196	Ready	Active	Leader	29.5.2

ubuntu@ip-172-31-8-196:~\$ nano taskboard-stack.yml

ubuntu@ip-172-31-8-196:~\$ docker stack deploy -c taskboard-stack.yml taskboard

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 taskboard_frontend

Creating network taskboard_backend

Creating service taskboard_web

Creating service taskboard_api

Creating service taskboard_db

Creating service taskboard_redis

Cats vs Dogs!

CATS

DOGS

(Tip: you can change your vote)

Processed by container ID
258f22021a0a

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

.wget-hsts

taskboard-stack.yml

OUTLINE

TIMELINE

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

bash - ubuntu

ID	NAME	MODE	REPLICAS	IMAGE	PORTS
io4tohk7bfr7	taskboard_api	replicated	2/2	dockersamples/examplevotingapp_worker:latest	
rwmz8qw8ivgo	taskboard_db	replicated	1/1	postgres:15-alpine	
zj27ntxueez0	taskboard_redis	replicated	1/1	redis:7-alpine	
jao38tb168a3	taskboard_web	replicated	2/2	dockersamples/examplevotingapp_vote:latest	*:8080->80/tcp

ubuntu@ip-172-31-8-196:~\$ docker stack ps taskboard

ID	NAME	IMAGE	NODE	DESIRED STATE	C
w4ec0pyr1mrc	taskboard_api.1	dockersamples/examplevotingapp_worker:latest	ip-172-31-8-81	Running	R
r0tgpp00kvz6	taskboard_api.2	dockersamples/examplevotingapp_worker:latest	ip-172-31-8-196	Running	R
ldfxmv102azh	taskboard_db.1	postgres:15-alpine	ip-172-31-8-81	Running	R
p9201utv4c14	taskboard_redis.1	redis:7-alpine	ip-172-31-8-196	Running	R
ctlt7jjanzipz	taskboard_web.1	dockersamples/examplevotingapp_vote:latest	ip-172-31-8-81	Running	R
am5nuanbjz3u	taskboard_web.2	dockersamples/examplevotingapp_vote:latest	ip-172-31-8-196	Running	R

ubuntu@ip-172-31-8-196:~\$ docker service scale taskboard_web=4

taskboard_web scaled to 4

overall progress: 4 out of 4 tasks

1/4: running [=====>]

2/4: running [=====>]

3/4: running [=====>]

4/4: running [=====>]

verify: Service taskboard_web converged

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

EXPLORER

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

> .cache

> .copilot

> .local

> .ssh

> .vscode-server

\$.bash_logout

\$.bashrc

\$.profile

≡ .wget-hsts

! taskboard-stack.yml

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

bash - ubuntu

+

⌵

🗑

⋮

🔄

✕

ubuntu@ip-172-31-8-196:~\$ watch docker service ps taskboard_web

ubuntu@ip-172-31-8-196:~\$ docker service scale taskboard_api=4

taskboard_api scaled to 4

overall progress: 4 out of 4 tasks

1/4: running [=====>]

2/4: running [=====>]

3/4: running [=====>]

4/4: running [=====>]

verify: Service taskboard_api converged

ubuntu@ip-172-31-8-196:~\$ docker stack services taskboard

ID	NAME	MODE	REPLICAS	IMAGE	PORTS
io4tohk7bfr7	taskboard_api	replicated	4/4	dockersamples/examplevotingapp_worker:latest	
rwmz8qw8ivgo	taskboard_db	replicated	1/1	postgres:15-alpine	
zj27ntxuueez0	taskboard_redis	replicated	1/1	redis:7-alpine	
jao38tb168a3	taskboard_web	replicated	4/4	dockersamples/examplevotingapp_vote:latest	*:8080->80/tc

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

≡ .wget-hsts

! taskboard-stack.yml

> OUTLINE

> TIMELINE

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

bash - ubuntu

+

▾

🗑

⋮

🔍

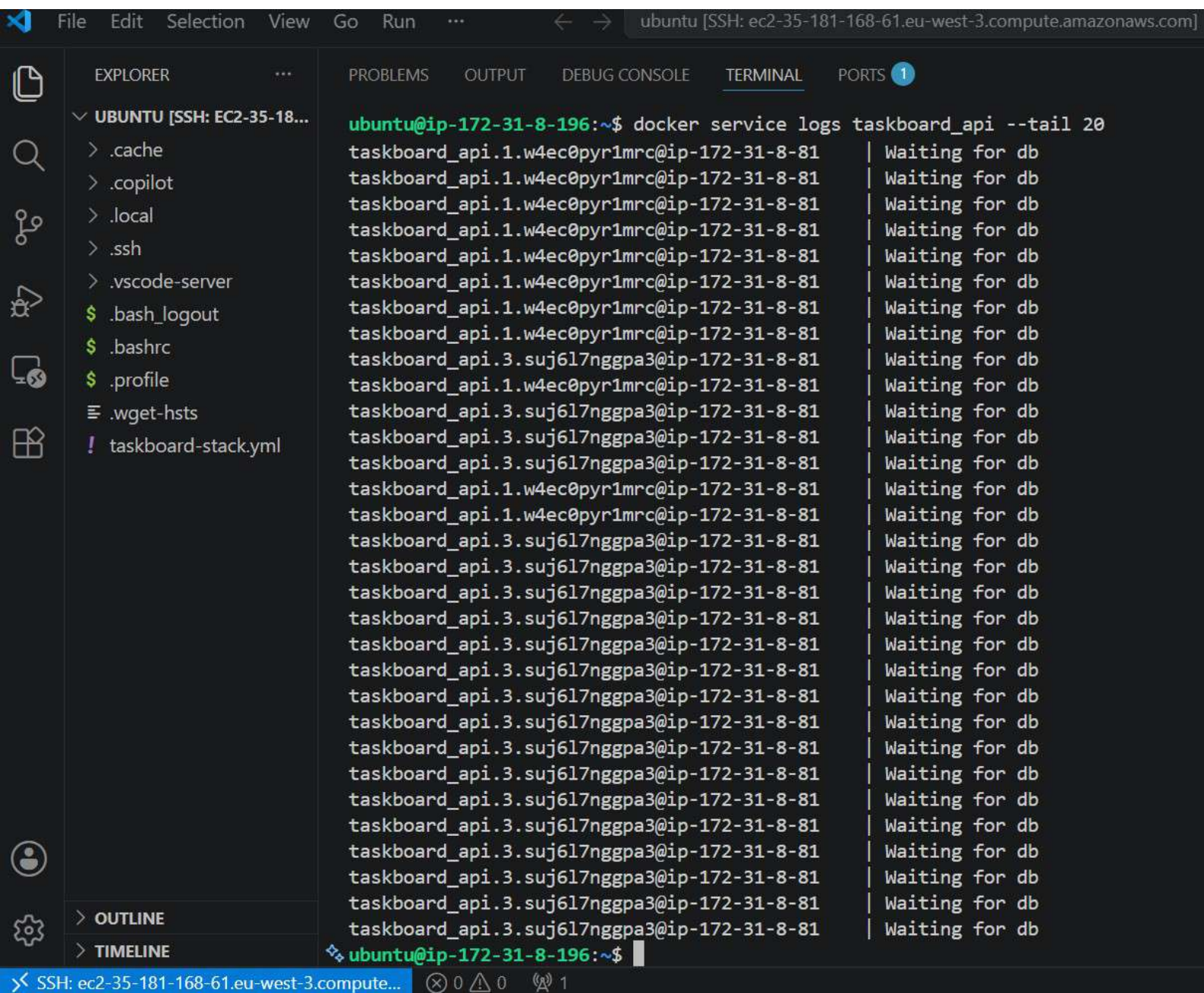
✕

```
ubuntu@ip-172-31-8-196:~$ docker service inspect taskboard_web --pretty
Levels:
  com.docker.stack.image=dockersamples/examplevotingapp_vote
  com.docker.stack.namespace=taskboard
Service Mode: Replicated
Replicas: 4
Placement:
UpdateConfig:
  Parallelism: 1
  Delay: 10s
  On failure: pause
  Monitoring Period: 5s
  Max failure ratio: 0
  Update order: stop-first
RollbackConfig:
  Parallelism: 1
  On failure: pause
  Monitoring Period: 5s
  Max failure ratio: 0
  Rollback order: stop-first
ContainerSpec:
  Image: dockersamples/examplevotingapp_vote:latest@sha256:7102d3b952ec84e3541ee12e7217e320c52aed60b13501c3158f46376a907466
  Env: API_URL=http://api:3000
Resources:
  Networks: owz4yyhxnidhvatfx09w4d5fo
  Endpoint Mode: vip
  Ports:
    PublishedPort = 8080
    Protocol = tcp
    TargetPort = 80
    PublishMode = ingress
```

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

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

```
File Edit Selection View Go Run ... 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
    .wget-hsts
    ! taskboard-stack.yml
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - ubuntu
ubuntu@ip-172-31-8-196:~$ docker service logs taskboard_web --tail 20
80 (1)
taskboard_web.1.cltt7janzipz@ip-172-31-8-81 | [2026-05-30 12:46:39 +0000] [1] [INFO] Using worker: sync
taskboard_web.1.cltt7janzipz@ip-172-31-8-81 | [2026-05-30 12:46:39 +0000] [8] [INFO] Booting worker with pid: 8
taskboard_web.1.cltt7janzipz@ip-172-31-8-81 | [2026-05-30 12:46:39 +0000] [9] [INFO] Booting worker with pid: 9
taskboard_web.1.cltt7janzipz@ip-172-31-8-81 | [2026-05-30 12:46:39 +0000] [10] [INFO] Booting worker with pid: 10
taskboard_web.1.cltt7janzipz@ip-172-31-8-81 | [2026-05-30 12:46:39 +0000] [11] [INFO] Booting worker with pid: 11
taskboard_web.1.cltt7janzipz@ip-172-31-8-81 | 10.0.0.3 - - [30/May/2026:12:54:22 +0000] "GET /static/stylesheets/s
tyle.css HTTP/1.1" 200 0 "http://13.38.102.221:8080/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/148.0.0.0 Safari/537.36"
taskboard_web.2.am5nuanbjz3u@ip-172-31-8-196 | [2026-05-30 12:46:36 +0000] [1] [INFO] Starting gunicorn 23.0.0
taskboard_web.2.am5nuanbjz3u@ip-172-31-8-196 | [2026-05-30 12:46:36 +0000] [1] [INFO] Listening at: http://0.0.0.0
:80 (1)
taskboard_web.2.am5nuanbjz3u@ip-172-31-8-196 | [2026-05-30 12:46:36 +0000] [1] [INFO] Using worker: sync
taskboard_web.2.am5nuanbjz3u@ip-172-31-8-196 | [2026-05-30 12:46:36 +0000] [7] [INFO] Booting worker with pid: 7
taskboard_web.2.am5nuanbjz3u@ip-172-31-8-196 | [2026-05-30 12:46:36 +0000] [8] [INFO] Booting worker with pid: 8
taskboard_web.2.am5nuanbjz3u@ip-172-31-8-196 | [2026-05-30 12:46:36 +0000] [9] [INFO] Booting worker with pid: 9
taskboard_web.2.am5nuanbjz3u@ip-172-31-8-196 | [2026-05-30 12:46:36 +0000] [10] [INFO] Booting worker with pid: 10
taskboard_web.2.am5nuanbjz3u@ip-172-31-8-196 | 10.0.0.3 - - [30/May/2026:12:54:22 +0000] "GET / HTTP/1.1" 200 1285
 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/148.0.0.0 Safari/537.36
"
taskboard_web.2.am5nuanbjz3u@ip-172-31-8-196 | 10.0.0.3 - - [30/May/2026:12:54:23 +0000] "GET /favicon.ico HTTP/1.
1" 404 207 "http://13.38.102.221:8080/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Ge
cko) Chrome/148.0.0.0 Safari/537.36"
taskboard_web.4.7mzgjwr8a3h9@ip-172-31-8-196 | [2026-05-30 12:55:30 +0000] [1] [INFO] Starting gunicorn 23.0.0
taskboard_web.4.7mzgjwr8a3h9@ip-172-31-8-196 | [2026-05-30 12:55:30 +0000] [1] [INFO] Listening at: http://0.0.0.0
:80 (1)
taskboard_web.4.7mzgjwr8a3h9@ip-172-31-8-196 | [2026-05-30 12:55:30 +0000] [1] [INFO] Using worker: sync
taskboard_web.4.7mzgjwr8a3h9@ip-172-31-8-196 | [2026-05-30 12:55:30 +0000] [7] [INFO] Booting worker with pid: 7
taskboard_web.4.7mzgjwr8a3h9@ip-172-31-8-196 | [2026-05-30 12:55:31 +0000] [8] [INFO] Booting worker with pid: 8
taskboard_web.4.7mzgjwr8a3h9@ip-172-31-8-196 | [2026-05-30 12:55:31 +0000] [9] [INFO] Booting worker with pid: 9
taskboard_web.4.7mzgjwr8a3h9@ip-172-31-8-196 | [2026-05-30 12:55:31 +0000] [10] [INFO] Booting worker with pid: 10
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

≡ .wget-hsts

! taskboard-stack.yml

OUTLINE

TIMELINE

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

ubuntu@ip-172-31-8-196:~\$ docker service update --force taskboard_web taskboard_web

overall progress: 4 out of 4 tasks

1/4: running [=====>]

2/4: running [=====>]

3/4: running [=====>]

4/4: running [=====>]

verify: Service taskboard_web converged

ubuntu@ip-172-31-8-196:~\$ watch docker service ps taskboard_web

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

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

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

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1
bash - ubuntu

ubuntu@ip-172-31-8-196:~\$ # Make multiple requests
PUBLIC_IP=\$(curl -s ifconfig.me)
for i in {1..10}; do
 echo "=== Request \$i ==="
 curl -s --max-time 5 http://\$PUBLIC_IP:8080 | head -10 ...

<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8">
 <title>Cats vs Dogs!</title>
 <base href="/index.html">
 <meta name = "viewport" content = "width=device-width, initial-scale = 1.0">
 <meta name="keywords" content="docker-compose, docker, stack">
 <meta name="author" content="Tutum dev team">
 <link rel='stylesheet' href="/static/stylesheets/style.css" />

=== Request 10 ===
<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8">
 <title>Cats vs Dogs!</title>
 <base href="/index.html">
 <meta name = "viewport" content = "width=device-width, initial-scale = 1.0">
 <meta name="keywords" content="docker-compose, docker, stack">
 <meta name="author" content="Tutum dev team">
 <link rel='stylesheet' href="/static/stylesheets/style.css" />

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: EC2-35-18...

.cache

.copilot

.local

.ssh

.vscode-server

.bash_logout

.bashrc

.profile

.wget-hsts

taskboard-stack.yml

OUTLINE

TIMELINE

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS 1

bash - ubuntu

+ v

...

ubuntu@ip-172-31-8-196:~\$ docker node update --availability drain ip-172-31-8-81

ip-172-31-8-81

ubuntu@ip-172-31-8-196:~\$ docker service ps taskboard_web

ID	NAME	IMAGE	NODE	DESIRED STATE	C
fm7h5w34mzmi	taskboard_web.1	dockersamples/examplevotingapp_vote:latest	ip-172-31-8-196	Running	R
unrunning about a minute ago					
vqn2g1fbocgz	_ taskboard_web.1	dockersamples/examplevotingapp_vote:latest	ip-172-31-8-81	Shutdown	S
hutdown about a minute ago					
ctlt7janzipz	_ taskboard_web.1	dockersamples/examplevotingapp_vote:latest	ip-172-31-8-81	Shutdown	S
hutdown 19 minutes ago					
box8xk3oieji	taskboard_web.2	dockersamples/examplevotingapp_vote:latest	ip-172-31-8-196	Running	R
unrunning 19 minutes ago					
am5nuanbjz3u	_ taskboard_web.2	dockersamples/examplevotingapp_vote:latest	ip-172-31-8-196	Shutdown	S
hutdown 19 minutes ago					
wq7hzvtlgak8	taskboard_web.3	dockersamples/examplevotingapp_vote:latest	ip-172-31-8-196	Running	R
unrunning about a minute ago					
pm9buhtpzg8c	_ taskboard_web.3	dockersamples/examplevotingapp_vote:latest	ip-172-31-8-81	Shutdown	S
hutdown about a minute ago					
mpecy3kytp4	_ taskboard_web.3	dockersamples/examplevotingapp_vote:latest	ip-172-31-8-81	Shutdown	S
hutdown 18 minutes ago					
hkecjytfypqr	taskboard_web.4	dockersamples/examplevotingapp_vote:latest	ip-172-31-8-196	Running	R
unrunning 19 minutes ago					
7mzgjwr8a3h9	_ taskboard_web.4	dockersamples/examplevotingapp_vote:latest	ip-172-31-8-196	Shutdown	S
hutdown 19 minutes ago					

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

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

FileEditSelectionViewGoRun...

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

EXPLOREUR

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

.cache

.copilot

.local

.ssh

.vscode-server

.bash_logout

.bashrc

.profile

.wget-hsts

taskboard-stack.yml

OUTLINE

TIMELINE

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

bash - ubuntu

ubuntu@ip-172-31-8-196:~\$ docker service ps taskboard_api

ID	NAME	IMAGE	NODE	DESIRED STATE
kwkh16bx2bpx	taskboard_api.1	dockersamples/examplevotingapp_worker:latest	ip-172-31-8-196	Running
w4ec0pyr1mrc	_ taskboard_api.1	dockersamples/examplevotingapp_worker:latest	ip-172-31-8-81	Shutdown
r0tgppo0kvz6	taskboard_api.2	dockersamples/examplevotingapp_worker:latest	ip-172-31-8-196	Running
vcj9ey6m3pnw	taskboard_api.3	dockersamples/examplevotingapp_worker:latest	ip-172-31-8-196	Running
suj6l7nggpa3	_ taskboard_api.3	dockersamples/examplevotingapp_worker:latest	ip-172-31-8-81	Shutdown
ut32rkvjle8v	taskboard_api.4	dockersamples/examplevotingapp_worker:latest	ip-172-31-8-196	Running

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]

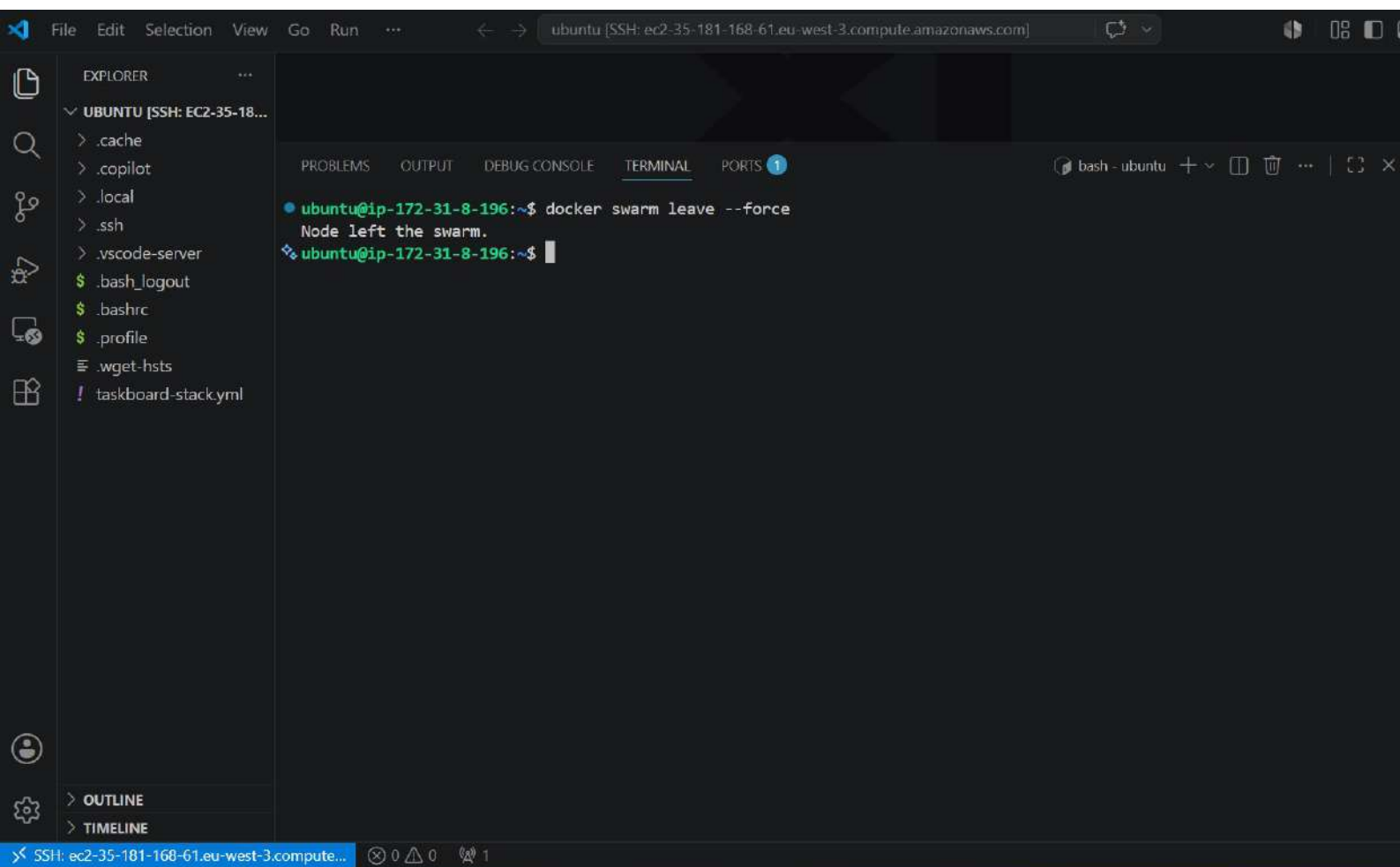
EXPLORER

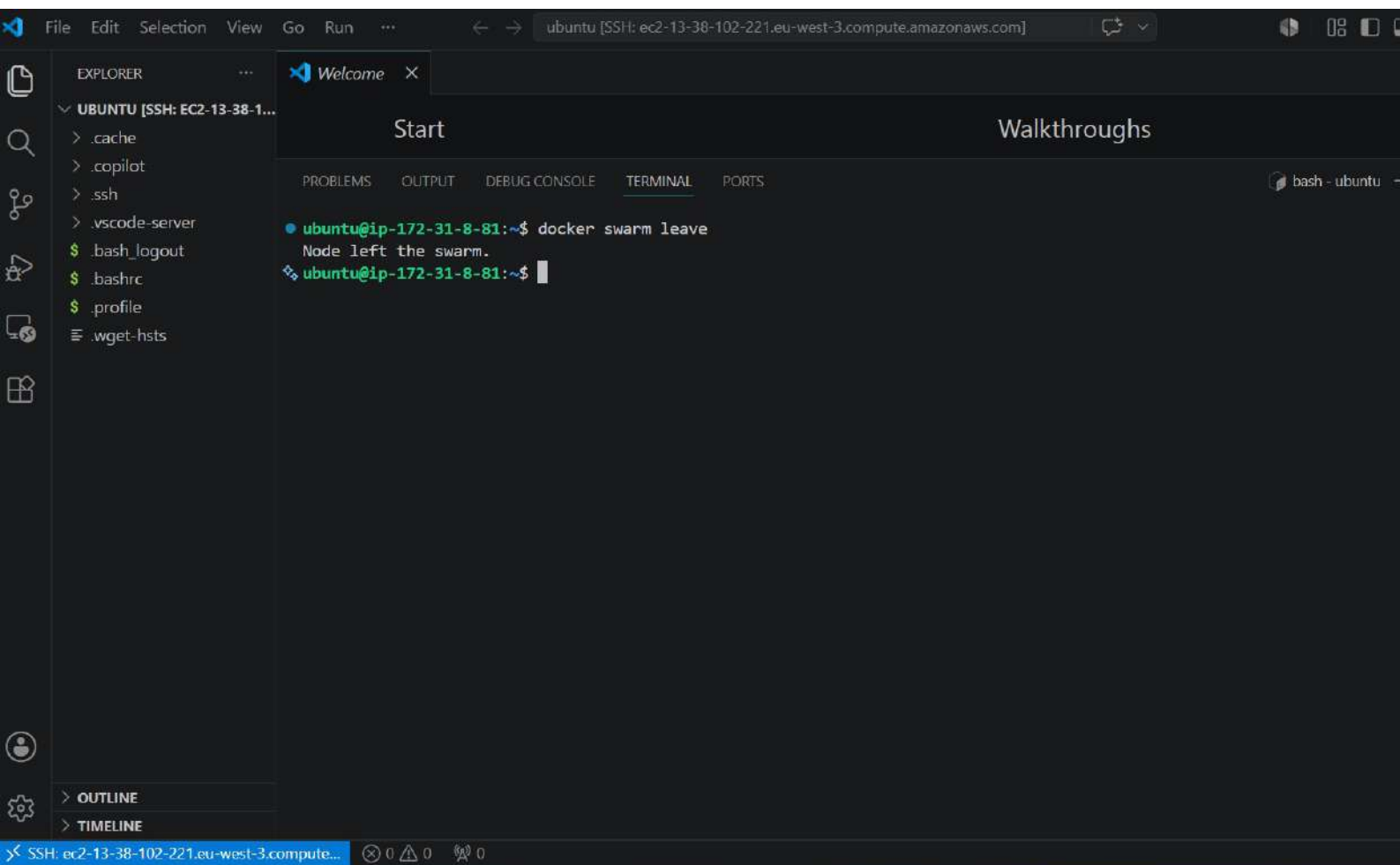
UBUNTU [SSH: EC2-35-18...> .cache> .copilot> .local> .ssh> .vscode-server\$.bash_logout\$.bashrc\$.profile≡ .wget-hsts! taskboard-stack.yml

PROBLEMSOUTPUTDEBUG CONSOLETERMINALPORTS 1

bash - ubuntu

ubuntu@ip-172-31-8-196:~\$ docker stack rm taskboard
Removing service taskboard_api
Removing service taskboard_db
Removing service taskboard_redis
Removing service taskboard_web
Removing network taskboard_backend
Removing network taskboard_frontend
ubuntu@ip-172-31-8-196:~\$ sleep 10
ubuntu@ip-172-31-8-196:~\$ docker stack ls
NAME SERVICES
ubuntu@ip-172-31-8-196:~\$ docker service ls
ID NAME MODE REPLICAS IMAGE PORTS





EC2 > Instances

EC2

Dashboard

AWS Global View

Events

Instances

Instances

Instance Types

Successfully initiated termination (deletion) of i-0e688238088b9eb1d

Instances (1/2) Info

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
☒	SwarmWorker	i-0e688238088b9eb1d	Shutting-d...	t3.small	-	eu-west-3a