

CI/CD PIPELINE USING GITHUB ACTIONS + DOCKER + AWS EC2

Project Overview

This project focuses on designing and implementing a **complete CI/CD (Continuous Integration and Continuous Deployment) pipeline** using **AWS Cloud and DevOps tools**. The main objective of the project is to **automate the process of building, deploying, and hosting a web application** using Docker and GitHub Actions on an AWS EC2 instance.

In this project, a **custom Virtual Private Cloud (VPC)** is created to provide a secure and isolated network environment. The VPC includes **public subnets**, routing configurations, and internet access to host the application securely on AWS.

The web application source code (HTML, CSS, JavaScript, and images) is stored in a **GitHub repository** along with a **Dockerfile**. Whenever a developer pushes code to the GitHub repository, **GitHub Actions automatically triggers the CI/CD pipeline**. This pipeline connects securely to the AWS EC2 instance using SSH, builds a Docker image from the updated source code, and deploys the application by running it inside a Docker container.

Docker is used to **containerize the application**, ensuring consistency across development and deployment environments. AWS EC2 acts as the **hosting server**, where the Docker container runs and serves the application through the Nginx web server.

The entire deployment process is fully automated, reducing manual effort, minimizing errors, and enabling faster delivery of application updates. This project demonstrates real-world DevOps practices such as **automation, infrastructure management, containerization, and cloud deployment**.

Key Objectives of the Project

- To create a secure AWS VPC for application hosting
- To automate application deployment using GitHub Actions
- To containerize a web application using Docker
- To deploy and host the application on AWS EC2
- To implement a real-time CI/CD workflow

STEP 1: Prepare the Project Locally

1.1 Project Structure

Your GitHub repository should look like this:

```
devops/
└── .github
    └── workflows/
        └── ci.yml
└── CI-CD Pipeline Project/
    └── Source Code/
        ├── index.html
        ├── style.css
        ├── script.js
        └── images/
            └── Dockerfile
```

STEP 2: Create Dockerfile (Inside Source Code Folder)

```
FROM nginx:alpine
COPY . /usr/share/nginx/html
EXPOSE 80
```

STEP 3: Create a VPC

3.1 Login to AWS Console

- Go to <https://aws.amazon.com>
- Sign in to the **AWS Management Console**

3.2 Open VPC Service

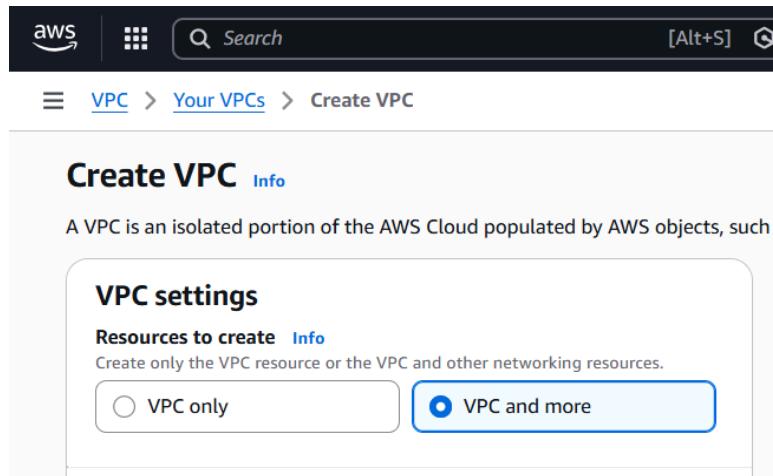
- In the search bar, type **VPC**
- Click **VPC → Create VPC**

3.3 Choose VPC Creation Option

Under **Resources to create**, select:

- **VPC and more**

(This automatically creates subnets, route tables, and gateways)



3.4 Name the VPC

- Enable **Name tag auto-generation**
- Enter name: **cicd-vpc**

3.5 Configure IPv4 CIDR Block

- IPv4 CIDR block:

10.0.0.0/16

Name tag auto-generation [Info](#)

Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

Auto-generate
cicd-vpc

IPv4 CIDR block [Info](#)

Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16 65,536 IPs

CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)

- No IPv6 CIDR block
 Amazon-provided IPv6 CIDR block

Provides **65,536 IP addresses**

(Leave IPv6 as **No IPv6 CIDR block**)

3.6 Select Tenancy

- Choose **Default**
(Default is cost-effective)

3.7 Choose Availability Zones

- Number of AZs: **2** (recommended for high availability)
- Example:
 - ap-south-1a
 - ap-south-1b

Number of Availability Zones (AZs) [Info](#)
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1 | **2** | 3

▼ Customize AZs

First availability zone
aps1-az1 (ap-south-1a) ▾

Second availability zone
aps1-az3 (ap-south-1b) ▾

3.8 Configure Subnets

Public Subnets: 2

Private Subnets: 2

Example CIDR blocks:

- Public subnet 1: **10.0.0.0/20**
- Public subnet 2: **10.0.16.0/20**
- Private subnet 1: **10.0.128.0/20**
- Private subnet 2: **10.0.144.0/20**

Number of public subnets [Info](#)
The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 | 2

Number of private subnets [Info](#)
The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 | 2 | 4

▼ Customize subnets CIDR blocks

Public subnet CIDR block in ap-south-1a

10.0.0.0/20	4,096 IPs
-------------	-----------

Public subnet CIDR block in ap-south-1b

10.0.16.0/20	4,096 IPs
--------------	-----------

Private subnet CIDR block in ap-south-1a

10.0.128.0/20	4,096 IPs
---------------	-----------

Private subnet CIDR block in ap-south-1b

10.0.144.0/20	4,096 IPs
---------------	-----------

3.9 Configure NAT Gateway

- Select **None** (for learning / free tier)
- NAT is used when private subnets need internet access

3.10 Configure VPC Endpoints

- Select **None** (optional)
- Used to access services like S3 without internet

3.11 Enable DNS Options

- Enable **DNS hostnames**
- Enable **DNS resolution**

NAT gateways (\$) - *updated* [Info](#)

NAT gateway allows private resources to access the internet from any availability zone within a VPC, providing a single managed internet exit point for the entire region. Additional charges apply.

[None](#) | [Regional - new](#) | [Zonal](#)

[Introducing regional NAT gateway](#) X

AWS now offers a multi-AZ NAT Gateway, eliminating the need for separate NAT Gateways across availability zones.

VPC endpoints [Info](#)

Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

[None](#) | [S3 Gateway](#)

DNS options [Info](#)

- Enable DNS hostnames
- Enable DNS resolution

3.12 Create VPC

- Review all settings
- Click **Create VPC**

Your VPC is now created successfully!

Your VPCs (3) Info							Last updated	Actions	Create VPC
<input type="checkbox"/>	Name	VPC ID	State	Encryption c...	Encryption control ...	Block Pu...			
<input type="checkbox"/>	-	vpc-0f9e4ee013c04e521	 Available	-	-	 Off			
<input type="checkbox"/>	cicdpro-vpc	vpc-0636be3e63e7b79c9	 Available	-	-	 Off			
<input type="checkbox"/>	cicd-vpc-vpc	vpc-0cb28b2142676a28d	 Available	-	-	 Off			

STEP 4: Launch an EC2 Virtual Server (Docker Build Server)

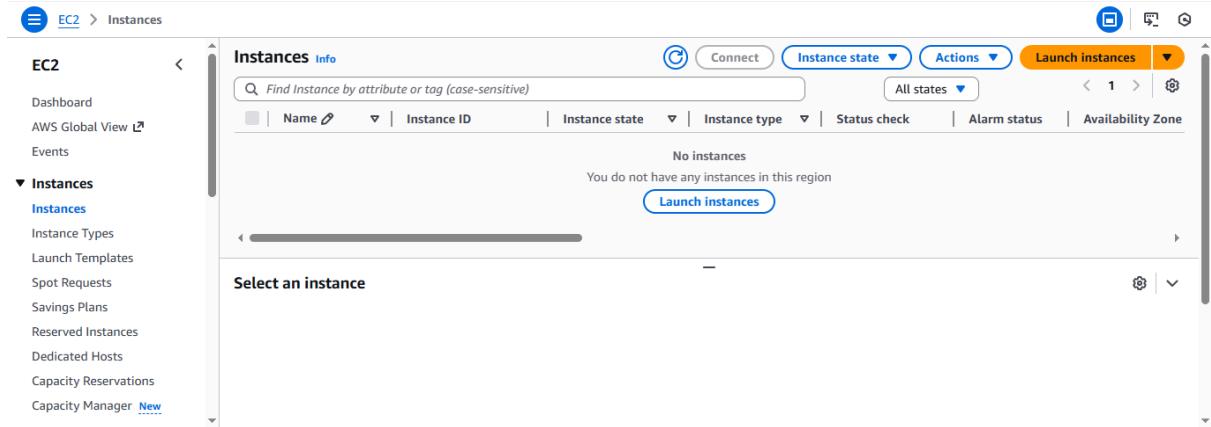
This EC2 instance will:

- Run Docker
- Build your website image

- Push image to ECR
(No Docker Desktop needed)

4.1 Go to EC2 Console

AWS Console → **EC2** → **Instances** → **Launch instance**



4.2 Choose AMI (Operating System)

Select:

Ubuntu

Why:

- AWS optimized
- Stable
- Docker works perfectly

4.3 Choose Instance Type

Select:

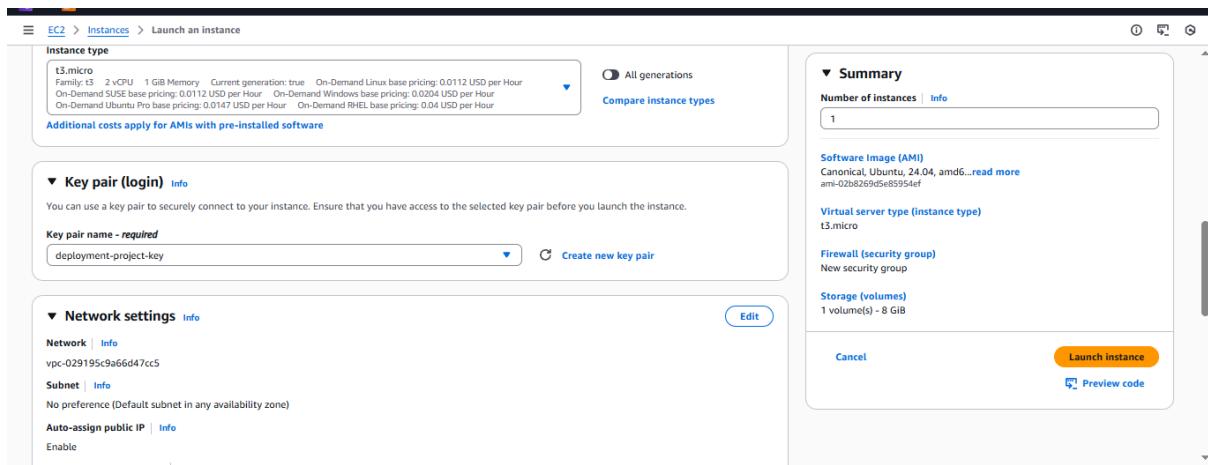
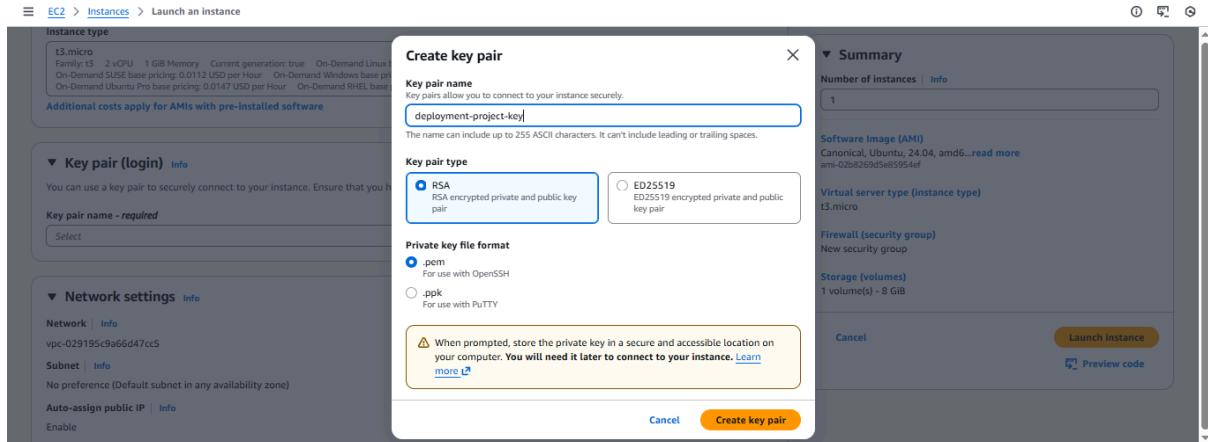
t3.micro
(Free-tier eligible)

Click **Next / Continue**

4.4 Key Pair (VERY IMPORTANT)

- Select **Create new key pair**
- Key pair type: RSA
- File format: **.pem**
- Name: **cicd-key**

Download and save this file safely (You cannot download it again)



4.5 Network Settings

Configure:

- VPC: Use your created VPC - `cicd-vpc`
- Subnet: Public subnet
- Auto-assign public IP: **Enabled**

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag auto-generation Info
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

Auto-generate
deployment-project-VPC

IPv4 CIDR block Info
Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16 65,536 IPs

CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
 No IPv6 CIDR block Amazon-provided IPv6 CIDR block

Tenancy Info
Default

Preview

VPC Show details
Your AWS virtual network

deployment-project-VPC-vpc

Subnets (4)
Subnets within this VPC

ap-south-1a
deployment-project-VPC-subnet-1
deployment-project-VPC-subnet-2

ap-south-1b
deployment-project-VPC-subnet-3
deployment-project-VPC-subnet-4

Route tables (3)
Route network traffic to resources

deployment-project-VPC-rtb-pub
deployment-project-VPC-rtb-prf
deployment-project-VPC-rtb-prv

VPC > Your VPCs > Create VPC

IPv4 CIDR block Info
Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16 65,536 IPs

CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
 No IPv6 CIDR block Amazon-provided IPv6 CIDR block

Tenancy Info
Default

Encryption settings - optional

Number of Availability Zones (AZs) Info
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1 2 3

Customize AZs

Number of public subnets Info
The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the Internet.

0 2

Preview

VPC Show details
Your AWS virtual network

deployment-project-VPC-vpc

Subnets (4)
Subnets within this VPC

ap-south-1a
deployment-project-VPC-subnet-1
deployment-project-VPC-subnet-2

ap-south-1b
deployment-project-VPC-subnet-3
deployment-project-VPC-subnet-4

Route tables (3)
Route network traffic to resources

deployment-project-VPC-rtb-pub
deployment-project-VPC-rtb-prf
deployment-project-VPC-rtb-prv

VPC > Your VPCs > Create VPC

Number of public subnets Info
The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the Internet.

0 2

Number of private subnets Info
The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 2 4

Customize subnets CIDR blocks

NAT gateways (5) - updated Info
NAT gateway allows private resources to access the internet from any availability zone within a VPC, providing a single managed internet exit point for the entire region. Additional charges apply.

None Regional - new Zonal

Introducing regional NAT gateway
AWS now offers a multi-AZ NAT Gateway, eliminating the need for separate NAT Gateways across availability zones.

VPN endpoints Info
Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy as needed.

Preview

Subnets (4)
Subnets within this VPC

ap-south-1a
deployment-project-VPC-subnet-1
deployment-project-VPC-subnet-2

ap-south-1b
deployment-project-VPC-subnet-3
deployment-project-VPC-subnet-4

Route tables (3)
Route network traffic to resources

deployment-project-VPC-rtb-public
deployment-project-VPC-rtb-private1
deployment-project-VPC-rtb-private2

Network connections (2)
Connections to other networks

deployment-project-VPC-igw
deployment-project-VPC-vpce-s

VPC > Your VPCs > Create VPC

NAT gateways (\$ - updated) Info
NAT gateway allows private resources to access the internet from any availability zone within a VPC, providing a single managed internet exit point for the entire region. Additional charges apply.

None | Regional - new | Zonal

Introducing regional NAT gateway X
AWS now offers a multi-AZ NAT Gateway, eliminating the need for separate NAT Gateways across availability zones.

VPC endpoints Info
Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None | S3 Gateway

DNS options Info
 Enable DNS hostnames
 Enable DNS resolution

Additional tags

Preview

Subnets (4)
Subnets within this VPC

- ap-south-1a
 - deployment-project-VPC-subnet-
 - deployment-project-VPC-subnet-
- ap-south-1b
 - deployment-project-VPC-subnet-
 - deployment-project-VPC-subnet-

Route tables (3)
Route network traffic to resources

- deployment-project-VPC-rtb-public
- deployment-project-VPC-rtb-private1
- deployment-project-VPC-rtb-private2

Network connections (2)
Connections to other networks

- deployment-project-VPC-igw
- deployment-project-VPC-vpc-e-s

Create VPC

VPC > Your VPCs > Create VPC > Create VPC resources

Details

- Create VPC: vpc-0bba916360bb4e70d
- Enable DNS hostnames
- Enable DNS resolution
- Verifying VPC creation: vpc-0bba916360bb4e70d
- Create S3 endpoint: vpc-0c177644f23a803b
- Create subnet: subnet-0624ce757655ff79
- Create subnet: subnet-03f425e660184ff8b
- Create subnet: subnet-026b0b0360a5788b
- Create subnet: subnet-01e19a10fb60a040b
- Create internet gateway: igw-03610fb7a9c41a3b4
- Attach internet gateway to the VPC
- Create route table: rtb-0ad53d7a1b619b005
- Create route
- Associate route table
- Associate route table
- Create route table: rtb-0922888181857f389
- Associate route table
- Create route table: rtb-05f1a940a9ff91fb7
- Associate route table
- Verifying route table creation
- Associate S3 endpoint with private subnet route tables: vpc-0c8177644f23a803b

Create VPC

VPC > Your VPCs

Your VPCs

VPCs **VPC encryption controls**

Your VPCs (1/2) Info

Name	VPC ID	State	Encryption c...	Encryption control...	Block Public...	IPv4 CIDR
vpc-029195c9a66d47c5	vpc-0bba916360bb4e70d	Available	-	-	Off	172.31.0.0/16
deployment-project-VPC	vpc-0bba916360bb4e70d	Available	-	-	Off	10.0.0.0/16

Last updated less than a minute ago

Actions **Create VPC**

vpc-0bba916360bb4e70d / deployment-project-VPC

Details **Resource map** **CIDRs** **Flow logs** **Tags** **Integrations**

4.6 Security Group Configuration

Create a new security group:

Inbound Rules

Add these rules:

```
SSH      | TCP | 22 | 0.0.0.0/0
HTTP     | TCP | 80 | 0.0.0.0/0
All TCP  | TCP | 0.0.0.0/0
```

Why:

- SSH → connect to EC2
- HTTP → test website temporarily

Security group rule 2 (TCP, 0-65535, 0.0.0.0/0)

- Type: All TCP
- Protocol: TCP
- Port range: 0-65535

Source type: Custom

Source: 0.0.0.0/0

Description - optional: e.g. SSH for admin desktop

Security group rule 3 (TCP, 80, 0.0.0.0/0)

- Type: HTTP
- Protocol: TCP
- Port range: 80

Source type: Custom

Source: 0.0.0.0/0

Description - optional: e.g. SSH for admin desktop

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6... [read more](#)

Virtual server type (instance type): t3.micro

Firewall (security group): New security group

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

[Launch instance](#) [Preview code](#)

4.7 Storage

Leave default:

8 GB gp3

Add security group rule

Advanced network configuration

Configure storage

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.

[Edit](#)

Advanced details

[CloudShell](#) [Feedback](#) [Console Mobile App](#)

[Launch instance](#) [Preview code](#)

4.8 Launch Instance

Click **Launch Instance**

Wait until:

- Instance state = **Running**
- Status checks = **3/3 passed**

Success
Successfully initiated launch of instance [i-0e2ae4a41ac74812](#)

▶ [Launch log](#)

Next Steps

Q What would you like to do next with this instance, for example "create alarm" or "create backup"

1 2 3 4 5 6 >

Create billing usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing usage thresholds.
[Create billing alerts](#)

Connect to your instance
Once your instance is running, log into it from your local computer.
[Connect to instance](#)
[Learn more](#)

Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
[Connect an RDS database](#)
[Create a new RDS database](#)
[Learn more](#)

Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots.
[Create EBS snapshot policy](#)

EC2 > Instances

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
deployment-p...	i-0e2ae4a41ac74812	Running	t3.micro	3/3 checks passed	View alarms	ap-south-1b	ec2-3-110-207-94.ap-s...	3.110.207.94

i-0e2ae4a41ac74812 (deployment-project)

[Details](#) | [Status and alarms](#) | [Monitoring](#) | [Security](#) | [Networking](#) | [Storage](#) | [Tags](#)

Instance summary

Instance ID i-0e2ae4a41ac74812	Public IPv4 address 3.110.207.94 open address	Private IPv4 addresses 10.0.25.153
IPv6 address -	Instance state Running	Public DNS ec2-3-110-207-94.ap-south-1.compute.amazonaws.com open

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

STEP 5: Connect to EC2 (Using AWS Console)

You are already inside the EC2 terminal.
Now we prepare this virtual server to act as a Docker machine.

EC2 > Instances

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
deployment-p...	i-0e2ae4a41ac74812	Running	t3.micro	3/3 checks passed	View alarms	ap-south-1b	ec2-3-110-207-94.ap-s...	3.110.207.94

i-0e2ae4a41ac74812 (deployment-project)

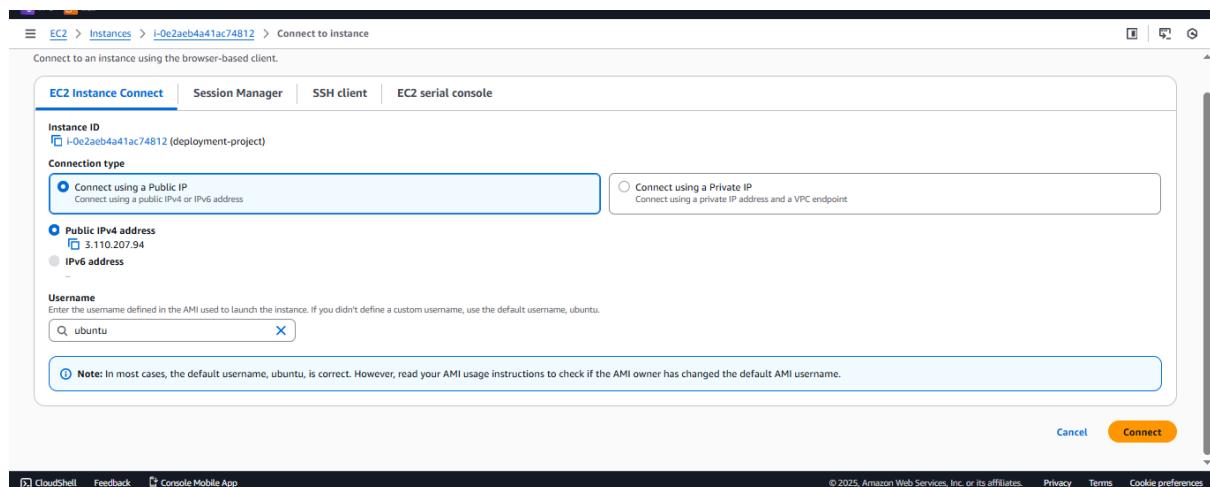
[Details](#) | [Status and alarms](#) | [Monitoring](#) | [Security](#) | [Networking](#) | [Storage](#) | [Tags](#)

Instance summary

Instance ID i-0e2ae4a41ac74812	Public IPv4 address 3.110.207.94 open address	Private IPv4 addresses 10.0.25.153
IPv6 address -	Instance state Running	Public DNS ec2-3-110-207-94.ap-south-1.compute.amazonaws.com open

https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Conn...

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)



STEP 6: Install Docker on EC2 (Virtual Docker)

6.1 Update the EC2 System

Run this **exact command** in the EC2 terminal:

```
sudo su
apt update
```

What this does

- Updates system packages
- Prevents Docker installation issues

Wait until it completes.

```
VPC EC2

Swap usage: 0%          IPv4 address for ens5: 10.0.25.153

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

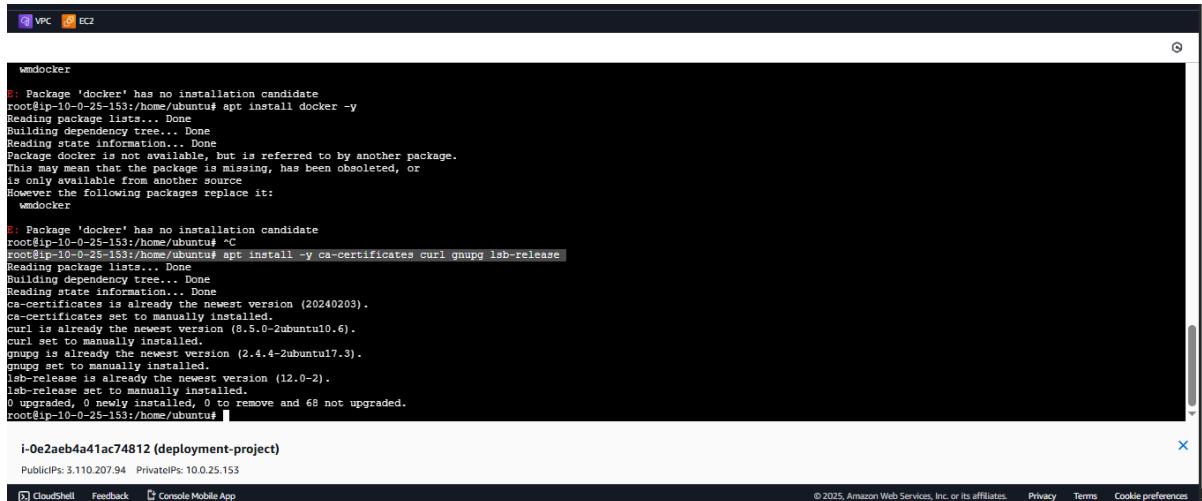
Last login: Sat Dec 20 04:40:25 2025 from 13.233.177.4
ubuntu@ip-10-0-25-153:~$ sudo su
root@ip-10-0-25-153:/home/ubuntu# apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease [126 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease [5982 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [3871 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [269 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [118 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1684 kB]

i-0e2ae84a41ac74812 (deployment-project)
PrivateIP: 3.110.207.94  PrivateIP: 10.0.25.153

CloudShell Feedback Console Mobile App
```

6.2 Install Required Packages

```
sudo apt install -y ca-certificates curl gnupg lsb-release
```

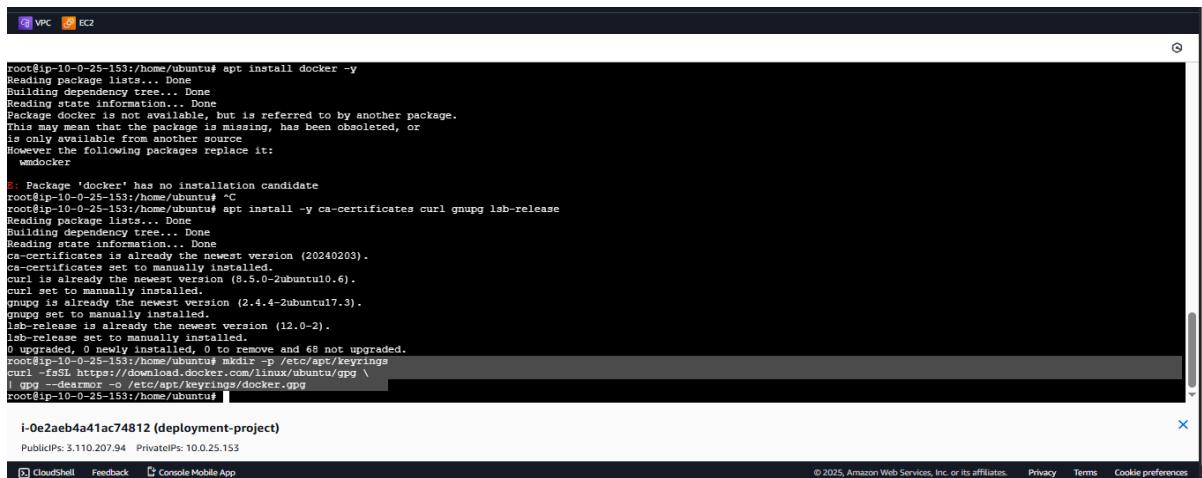


```
wm docker
E: Package 'docker' has no installation candidate
root@ip-10-0-25-153:/home/ubuntu# apt install docker -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package docker is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
However the following packages replace it:
  wmdocker

E: Package 'docker' has no installation candidate
root@ip-10-0-25-153:/home/ubuntu# ^C
root@ip-10-0-25-153:/home/ubuntu# apt install -y ca-certificates curl gnupg lsb-release
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
curl is already the newest version (8.5.0-2ubuntu10.6).
curl set to manually installed.
gnupg is already the newest version (2.4.4-2ubuntu17.3).
gnupg set to manually installed.
lsb-release is already the newest version (12.0-2).
lsb-release set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
root@ip-10-0-25-153:/home/ubuntu#
```

6.3 Add Docker's Official GPG Key

```
mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
| gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```



```
root@ip-10-0-25-153:/home/ubuntu# apt install docker -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package docker is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
However the following packages replace it:
  wmdocker

E: Package 'docker' has no installation candidate
root@ip-10-0-25-153:/home/ubuntu# ^C
root@ip-10-0-25-153:/home/ubuntu# apt install -y ca-certificates curl gnupg lsb-release
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
curl is already the newest version (8.5.0-2ubuntu10.6).
curl set to manually installed.
gnupg is already the newest version (2.4.4-2ubuntu17.3).
gnupg set to manually installed.
lsb-release is already the newest version (12.0-2).
lsb-release set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
root@ip-10-0-25-153:/home/ubuntu# mkdir -p /etc/apt/keyrings
root@ip-10-0-25-153:/home/ubuntu# curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
| gpg --dearmor -o /etc/apt/keyrings/docker.gpg
root@ip-10-0-25-153:/home/ubuntu#
```

6.4 Add Docker Repository

```
echo \
"deb [arch=$(dpkg --print-architecture)
signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" \
| tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
is only available from another source  
However the following packages replace it:  
  wodocker  
#  
# Package 'docker' has no installation candidate  
root@ip-10-0-25-153:/home/ubuntu# ^C  
root@ip-10-0-25-153:/home/ubuntu# apt install -y ca-certificates curl gnupg lsb-release  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
ca-certificates is already the newest version (20240203).  
ca-certificates set to manually installed.  
curl is already the newest version (8.5.0-2ubuntu10.6).  
curl set to manually installed.  
gnupg is already the newest version (2.4.4-2ubuntu17.3).  
gnupg set to manually installed.  
lsb-release is already the newest version (12.0-2).  
lsb-release set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.  
root@ip-10-0-25-153:/home/ubuntu# mkdir -p /etc/apt/keyrings  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg  
| gpg --dearmor -o /etc/apt/keyrings/docker.gpg  
root@ip-10-0-25-153:/home/ubuntu#  
echo '  
deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu  
$lsb_release -cs" stable"  
| tee /etc/apt/sources.list.d/docker.list > /dev/null  
root@ip-10-0-25-153:/home/ubuntu#
```

i-0e2aeb4a41ac74812 (deployment-project)
PublicIPs: 3.110.207.94 PrivateIPs: 10.0.25.153
CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

6.5 Update Package List Again

```
apt update
```

6.6 Install Docker Engine (Correct Package)

```
apt install -y docker-ce docker-ce-cli containerd.io
```

```
lsb-release set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.  
root@ip-10-0-25-153:/home/ubuntu# mkdir -p /etc/apt/keyrings  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg  
| gpg --dearmor -o /etc/apt/keyrings/docker.gpg  
root@ip-10-0-25-153:/home/ubuntu#  
echo '  
deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu  
$lsb_release -cs" stable"  
| tee /etc/apt/sources.list.d/docker.list > /dev/null  
root@ip-10-0-25-153:/home/ubuntu# apt update  
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Get:4 https://download.docker.com/linux/ubuntu noble InRelease [48.5 kB]  
Get:5 https://download.docker.com/linux/ubuntu/noble/stable amd64 Packages [41.1 kB]  
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease  
Fetched 98 kB in 1s (185 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
68 packages can be upgraded. Run 'apt list --upgradable' to see them.  
root@ip-10-0-25-153:/home/ubuntu# apt install -y docker-ce docker-ce-cli containerd.io  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  docker-buildx-plugin docker-ce-rootless-extras docker-compose-plugin libslirp0 pigz slirp4netns  
Suggested packages:
```

i-0e2aeb4a41ac74812 (deployment-project)
PublicIPs: 3.110.207.94 PrivateIPs: 10.0.25.153
CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

6.7 Start and Enable Docker

```
systemctl start docker  
systemctl enable docker
```

```

Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (5.0.0-1ubuntu.24.04-noble) ...
Setting up docker-ce-cli (5:29.1.3-1ubuntu.24.04-noble) ...
Setting up docker (5:29.1.3-1ubuntu.24.04-noble) ...
Setting up liblz4-1 (4.7.0-1ubuntu3) ...
Setting up libpigz1 (2.8-1)
Setting up docker-ce-rootless-extras (5:29.1.3-1ubuntu.24.04-noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:29.1.3-1ubuntu.24.04-noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4ubuntu2) ...
Processing triggers for libc-bin (2.39-Dubuntu8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (gemu) binaries on this host.
root@ip-10-0-25-153:~/.home/ubuntu$ systemctl start docker
root@ip-10-0-25-153:~/.home/ubuntu$ systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
root@ip-10-0-25-153:~/.home/ubuntu# 
```

i-0e2ae4a41ac74812 (deployment-project)
PublicIPs: 3.110.207.94 PrivateIPs: 10.0.25.153

[CloudShell](#) [Feedback](#) [Console Mobile App](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Check status:

`systemctl status docker`

You should see:

`active (running)`

```

No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (gemu) binaries on this host.
root@ip-10-0-25-153:~/.home/ubuntu$ systemctl start docker
root@ip-10-0-25-153:~/.home/ubuntu$ systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
root@ip-10-0-25-153:~/.home/ubuntu# 
```

- docker.service - Docker Application Container Engine
 - Loaded: loaded (/usr/lib/systemd/system/docker.service; **enabled; preset: enabled**)
 - Active: **active (running)** since Sat 2025-12-20 01:00:05 UTC; 4min 52s ago
 - TriggeredBy: • docker.socket
 - Docs: <https://docs.docker.com>
 - Main PID: 3402 (dockerd)
 - Tasks: 9
 - Memory: 41.4M (peak: 42.3M)
 - CPU: 0.82ms
 - CGROUPS: /system.slice/docker.service
 - └─3402 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

```

Dec 20 05:00:04 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:04.761182295Z" level=info msg="Restoring containers: start."
Dec 20 05:00:04 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:04.809726955Z" level=info msg="Deleting nftables IPv4 rules" error="exit status 1"
Dec 20 05:00:04 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:04.818586659Z" level=info msg="Deleting nftables IPv6 rules" error="exit status 1"
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.250529885Z" level=info msg="Loading containers: done." 
```

i-0e2ae4a41ac74812 (deployment-project)
PublicIPs: 3.110.207.94 PrivateIPs: 10.0.25.153

[CloudShell](#) [Feedback](#) [Console Mobile App](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

6.8 Allow Docker Without sudo (VERY IMPORTANT)

`usermod -aG docker ubuntu`

```

root@ip-10-0-25-153:/home/ubuntu# systemctl enable docker
Synchronizing state of docker service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemctl enable docker
root@ip-10-0-25-153:/home/ubuntu# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-12-20 05:00:05 UTC; 4min 52s ago
       Docs: man:docker(1)
    TriggeredBy: ● docker.socket
      Main PID: 3402 (dockerd)
        Tasks: 9
       Memory: 41.4M (peak: 42.3M)
          CPU: 528ms
         CGroub: /system.slice/docker.service
             ↳ 3402 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Dec 20 05:00:04 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:04.761z" level=info msg="Restoring containers: start."
Dec 20 05:00:04 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:04.809z" level=info msg="Deleting nftables IPv4 rules" error="exit status 1"
Dec 20 05:00:04 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:04.819z" level=info msg="Deleting nftables IPv6 rules" error="exit status 1"
Dec 20 05:00:04 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:04.829z" level=info msg="Restoring containers: done."
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.261z" level=info msg="Docker daemon initialized"
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.261z" level=info msg="Using buildkit"
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.328z" level=info msg="Completed buildkit initialization"
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.343z" level=info msg="Daemon has completed initialization"
Dec 20 05:00:05 ip-10-0-25-153 systemd[1]: Started docker.service - Docker Application Container Engine.
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.345z" level=info msg="API listen on /run/docker.sock"

root@ip-10-0-25-153:/home/ubuntu# usermod -aG docker ubuntu
root@ip-10-0-25-153:/home/ubuntu#

```

i-0e2ae4a41ac74812 (deployment-project)
Public IPs: 3.110.207.94 Private IPs: 10.0.25.153

[CloudShell](#) [Feedback](#) [Console Mobile App](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Now disconnect and reconnect the EC2 console session.

EC2 > Instances > i-0e2ae4a41ac74812 > Connect to instance

Connect Info

Connect to an instance using the browser-based client.

[EC2 Instance Connect](#) [Session Manager](#) [SSH client](#) [EC2 serial console](#)

Instance ID
i-0e2ae4a41ac74812 (deployment-project)

Connection type

Connect using a Public IP
Connect using a public IPv4 or IPv6 address

Connect using a Private IP
Connect using a private IP address and a VPC endpoint

Public IPv4 address
3.110.207.94

IPv6 address

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

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

[Cancel](#) [Connect](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

6.9 Verify Docker Installation

After reconnecting, run:

`docker --version`

```

Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Sat Dec 20 05:09:40 UTC 2025

System load: 0.13 Temperature: -273.1 C
Usage of /: 34.9% of 6.71GB Processes: 115
Memory usage: 29% Users logged in: 0
Swap usage: 0% IPv4 address for ens5: 10.0.25.153

Expanded Security Maintenance for Applications is not enabled.

74 updates can be applied immediately.
28 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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

Last login: Sat Dec 20 04:45:14 2025 from 13.233.177.4
ubuntu@ip-10-0-25-153:~$ docker --version
Docker version 29.1.3, build f52b14d
ubuntu@ip-10-0-25-153:~$ 

```

i-0e2ae4a41ac74812 (deployment-project)
Public IPs: 3.110.207.94 Private IPs: 10.0.25.153

[CloudShell](#) [Feedback](#) [Console Mobile App](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

STEP 7: Add GitHub Actions Secrets (VERY IMPORTANT)

Go to:

GitHub Repo → Settings → Secrets and variables → Actions

Add 3 Repository Secrets:

Secret Name	Value
EC2_HOST	EC2 Public IPv4
EC2_USER	ubuntu
EC2_KEY	Full .pem file content (open the .pem using notepad & copy the entire thing and paste it in the secret)

These are used for **SSH from GitHub Actions to EC2**.

General

Actions secrets and variables

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Models

Webhooks

Copilot

Environments

Codespaces

Pages

Security

Advanced Security

Deploy keys

Secrets and variables

Actions

Actions secrets and variables

Secrets Variables

Environment secrets

This environment has no secrets.

Manage environment secrets

New repository secret

Name	Last updated
EC2_HOST	50 minutes ago
EC2_KEY	49 minutes ago
EC2_USER	50 minutes ago

STEP 8: Create GitHub Actions Workflow

Create file:

```
.github/workflows/ci.yml
```

CI/CD YAML (ci.yml)

```
name: CI-CD Build and Deploy Docker Image

on:
  push:
    branches:
      - main

jobs:
  deploy:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout source code
        uses: actions/checkout@v4

      - name: Deploy to EC2
        uses: appleboy/ssh-action@v1.0.3
        with:
          host: ${{ secrets.EC2_HOST }}
          username: ${{ secrets.EC2_USER }}
          key: ${{ secrets.EC2_KEY }}
          script: |
            docker stop frontend-app || true
            docker rm frontend-app || true

            rm -rf devops || true
            git clone https://github.com/<username>/devops.git
            cd devops/CI-CD\ Pipeline\ Project/Source\ Code

            docker build -t frontend-app .
            docker run -d -p 80:80 --name frontend-app frontend-app
```

STEP 9: Verify Deployment

9.1 GitHub Actions

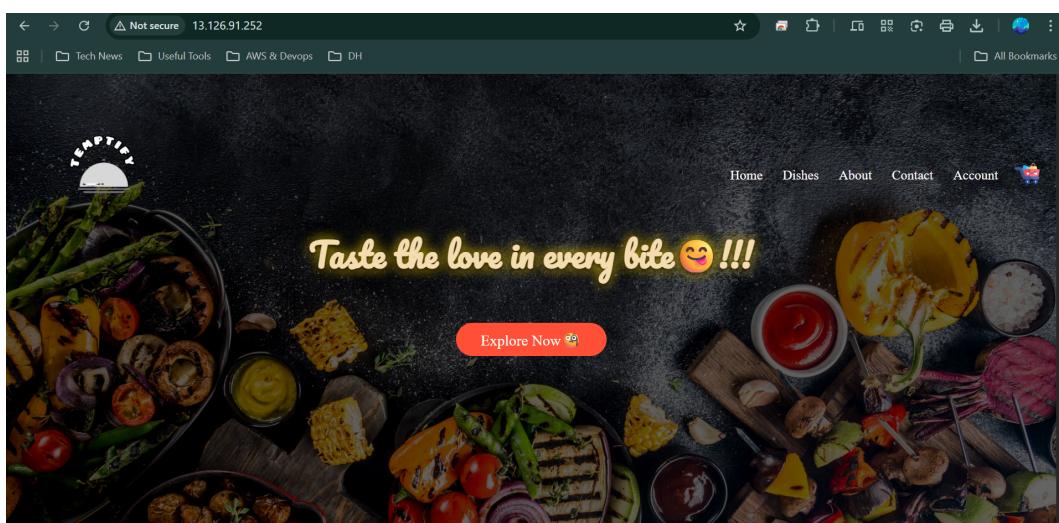
- Go to **Actions**
- Check **green tick (Success)**

9.2 Verify on EC2

`docker ps`

9.3 Open Website

`http://<EC2_PUBLIC_IP>`



STEP 10: CI/CD FLOW (EXPLAIN THIS)

Developer pushes code to GitHub



GitHub Actions triggers workflow



SSH connection to EC2



Clone repository on EC2



Build Docker image



Run Docker container



Website deployed automatically