

# WEB APPLICATION DEPLOYMENT

Deployed an existing static web application sourced from a GitHub repository using a containerized, cloud-native AWS architecture. The focus of the project was on containerization, cloud deployment, networking, and scalability rather than frontend development.

## Tech Stacks Used

- **GitHub** – Source code repository (cloned existing application)
- **Docker** – Containerization (installed and used on an EC2 virtual server)
- **Amazon EC2** – Virtual server for building Docker images
- **Amazon ECR** – Container image repository
- **Amazon ECS (Fargate)** – Serverless container orchestration
- **Application Load Balancer (ALB)** – Traffic routing, health checks, and high availability
- **AWS CLI** – AWS service interaction and automation

## STEP 1: 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)

### 1.1 Go to EC2 Console

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

## 1.2 Choose AMI (Operating System)

Select:

**Ubuntu**

Why:

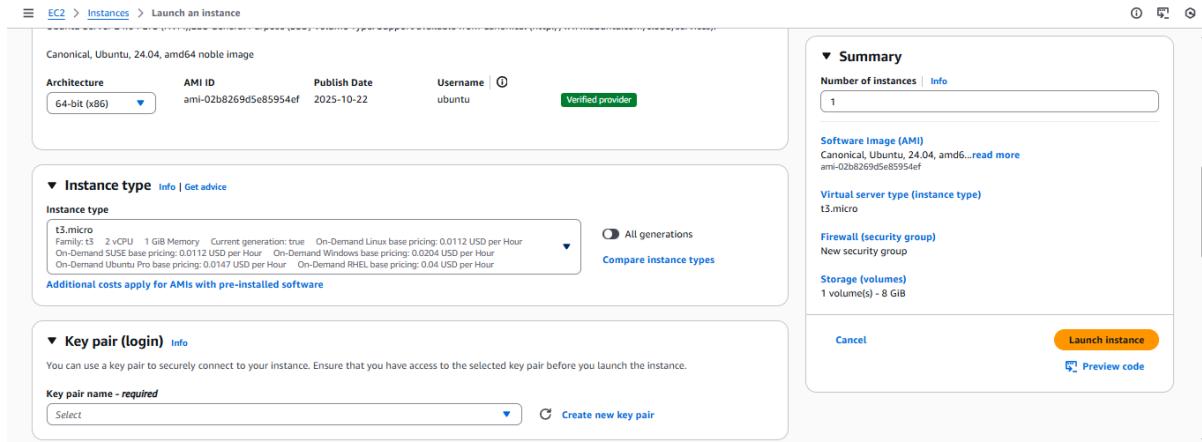
- AWS optimized
- Stable
- Docker works perfectly

## 1.3 Choose Instance Type

Select:

**t2.micro or t3.micro**  
(Free-tier eligible)

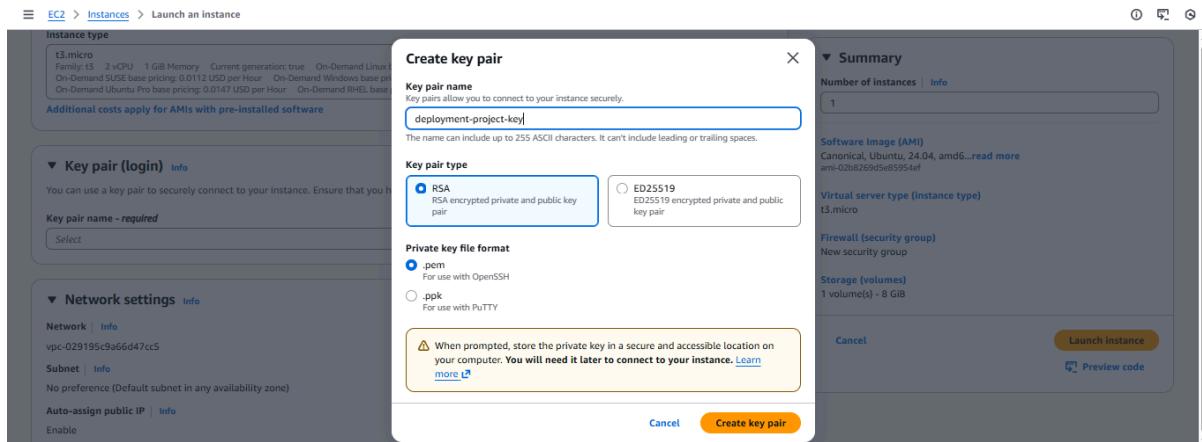
Click **Next / Continue**



## 1.4 Key Pair (VERY IMPORTANT)

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

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



The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. It includes sections for:

- Instance type:** T3.micro selected, with details about Family, vCPU, Memory, Current generation, and pricing.
- Key pair (login):** A dropdown menu for selecting a key pair, with 'deployment-project-key' selected and a 'Create new key pair' button.
- Network settings:** Includes Network (vpc-029195c9a66d47cc), Subnet (No preference), and Auto-assign public IP (Enabled).
- Summary:** Shows 1 instance, Software Image (AMI) Canonical, Ubuntu, 24.04, Virtual server type (instance type) T3.micro, Firewall (security group) New security group, Storage (volumes) 1 volume(s) - 8 GiB, and a 'Launch instance' button.

## 1.5 Network Settings

Configure:

- VPC: Use your created VPC - **deployment-project-VPC**
- Subnet: Public subnet
- Auto-assign public IP: **Enabled**

The screenshot shows the 'Create VPC' wizard in the AWS VPC console. It includes:

- VPC settings:** Set to 'VPC and more'. Options include Name tag auto-generation (Auto-generate checked, deployment-project-VPC), IPv4 CIDR block (10.0.0.0/16, 65,536 IPs), IPv6 CIDR block (No IPv6 CIDR block), and Tenancy (Hosted).
- Preview:** Shows the VPC structure with:
  - VPC:** Your AWS virtual network deployment-project-VPC-vpc
  - Subnets (4):** ap-south-1a (deployment-project-VPC-subnet-0), ap-south-1a (deployment-project-VPC-subnet-1), ap-south-1b (deployment-project-VPC-subnet-2), ap-south-1b (deployment-project-VPC-subnet-3)
  - Route tables (3):** deployment-project-VPC-rtb-pub, deployment-project-VPC-rtb-prv, 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.

 65,536 IPs  
 CIDR block size must be between /16 and /28.

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

**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**

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 (\$ - 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.

**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-s2

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.

**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-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-s2

**Create VPC**

The screenshot shows the 'Create VPC resources' step in the AWS VPC creation wizard. It lists 20 successful actions, each with a green checkmark and a blue link icon:

- Create VPC: vpc-0bba916360bb4e70d
- Enable DNS hostnames
- Enable DNS resolution
- Verifying VPC creation: vpc-0bba916360bb4e70d
- Create S3 endpoint: vpce-0c8177644f23a803b
- Create subnet: subnet-0624e7570655f19
- Create subnet: subnet-03f425e660184ff8b
- Create subnet: subnet-026b0b0360a578bbcc
- Create subnet: subnet-01e19a410fb6a040b
- 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: vpce-0c8177644f23a803b

**View VPC** button at the bottom right.

The screenshot shows the 'Your VPCs' page in the AWS VPC dashboard. It displays a table of VPCs with the following columns: Name, VPC ID, State, Encryption controls, Block Public, and IPv4 CIDR. Two VPCs are listed:

Name	VPC ID	State	Encryption controls	Block Public	IPv4 CIDR
vpc-029195c9a66d47cc5	vpc-029195c9a66d47cc5	Available	-	Off	172.31.0.0/16
<b>deployment-project-VPC</b>	vpc-0bba916360bb4e70d	Available	-	Off	10.0.0.0/16

The 'deployment-project-VPC' row is selected. Below the table, there's a 'Details' tab showing the VPC configuration.

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 service. It includes sections for Key pair (login), Network settings, and Summary.

**Key pair (login)**: A dropdown menu shows 'deployment-project-key'.

**Network settings**: Includes VPC selection (vpc-0bba916360bb4e70d), Subnet selection (subnet-03f425e660184ff8b), Auto-assign public IP (Enable), and Firewall (security group) selection (Create security group).

**Summary**: Shows 1 instance, Software Image (AMI) (Canonical, Ubuntu, 24.04, amd64), Virtual server type (instance type) (t3.micro), Firewall (security group) (New security group), Storage (volumes) (1 volume(s) - 8 GiB), and buttons for 'Cancel', 'Launch instance', and 'Preview code'.

# 1.6 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

**EC2 > Instances > Launch an Instance**

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

Type   Info	Protocol   Info	Port range   Info
All TCP	TCP	0-65535
Remove		

Source type | Info

Custom

Source | Info

Add CIDR, prefix list or security group

Description - optional | Info

e.g. SSH for admin desktop

0.0.0.0/0 X

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

Type   Info	Protocol   Info	Port range   Info
HTTP	TCP	80
Remove		

Source type | Info

Custom

Source | Info

Add CIDR, prefix list or security group

Description - optional | Info

e.g. SSH for admin desktop

0.0.0.0/0 X

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

Add security group rule

**Summary**

Number of instances | Info

1

**Software Image (AMI)**  
Canonical, Ubuntu, 24.04, amd64... [read more](#)  
ami-02ba8269de683954ef

**Virtual server type (instance type)**  
t3.micro

**Firewall (security group)**  
New security group

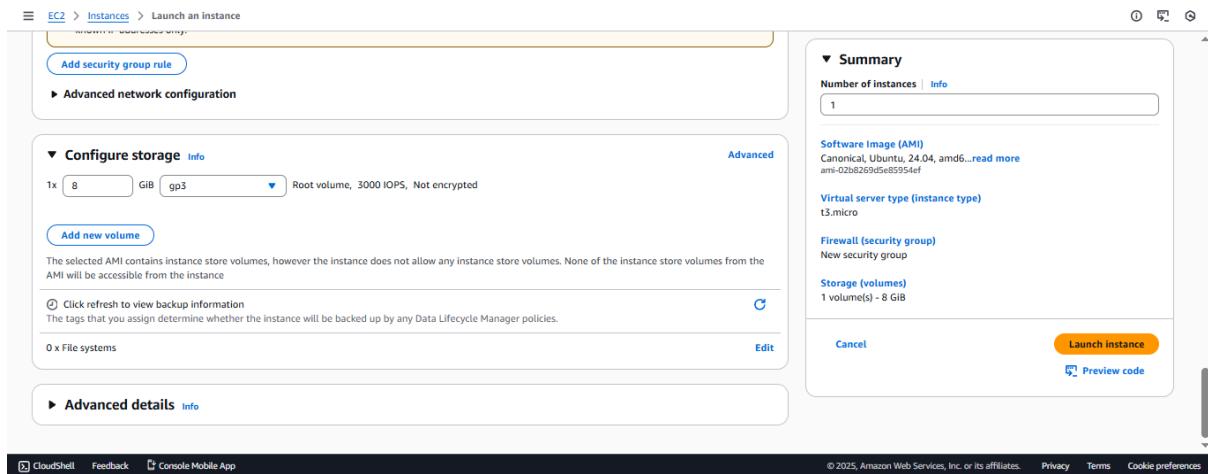
**Storage (volumes)**  
1 volume(s) - 8 GiB

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

## 1.7 Storage

**Leave default:**

8 GB gp3

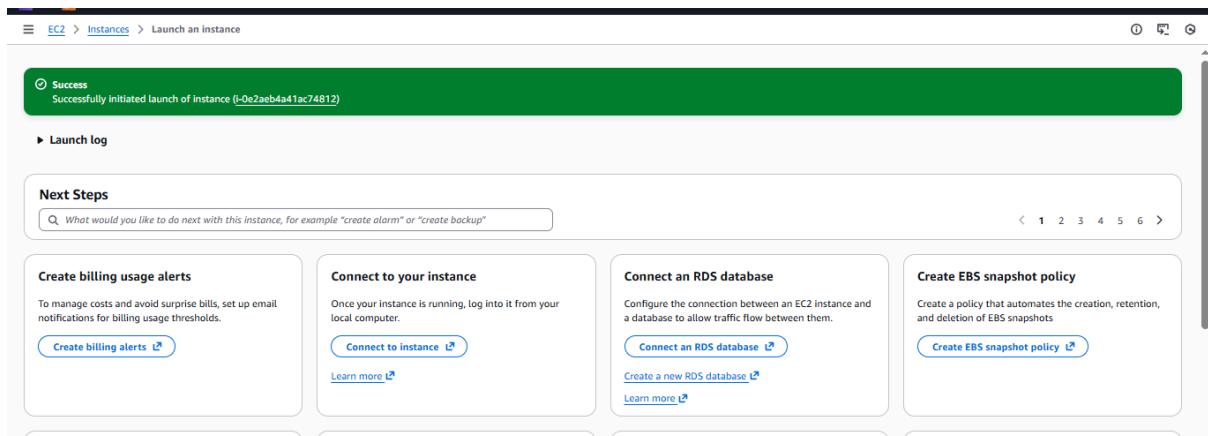


## 1.8 Launch Instance

Click **Launch Instance**

Wait until:

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



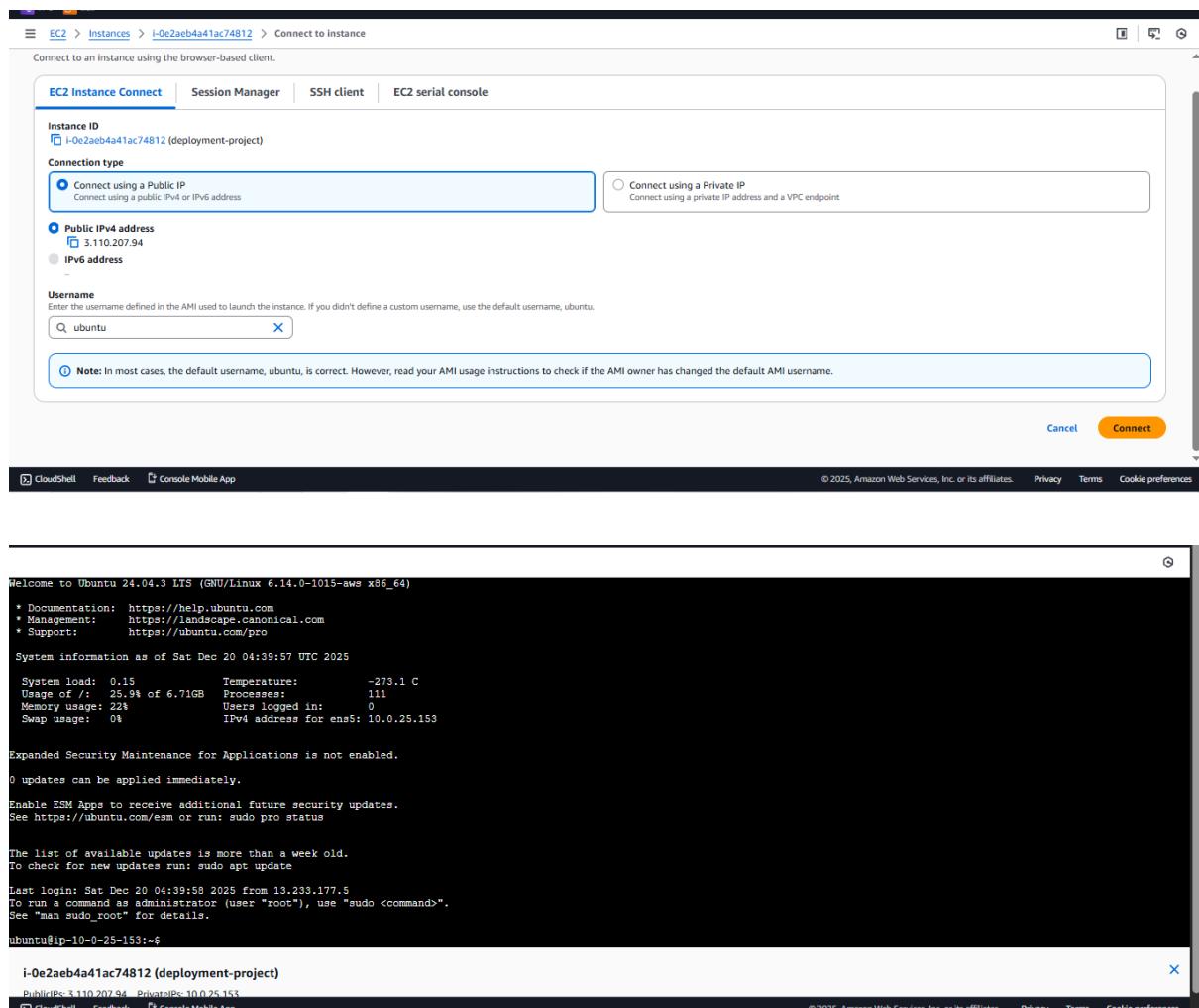
The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with 'EC2' selected. The main area displays a table of instances. One instance is highlighted: 'i-0e2aeab4a41ac74812 (deployment-project)'. Below the table, the instance details are shown, including its Public IPv4 address (3.110.207.94), Instance state (Running), and Public DNS (ec2-3-110-207-94.ap-s...). The bottom of the screen includes standard AWS footer links.

## STEP 2: 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.

This screenshot is identical to the one above, showing the AWS EC2 Instances page with the same instance details. The URL at the bottom of the browser window is https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Conn..., indicating the user is currently connected to the instance via the AWS console.



# STEP 3: Install Docker on EC2 (Virtual Docker)

## 3.1 Update the EC2 System

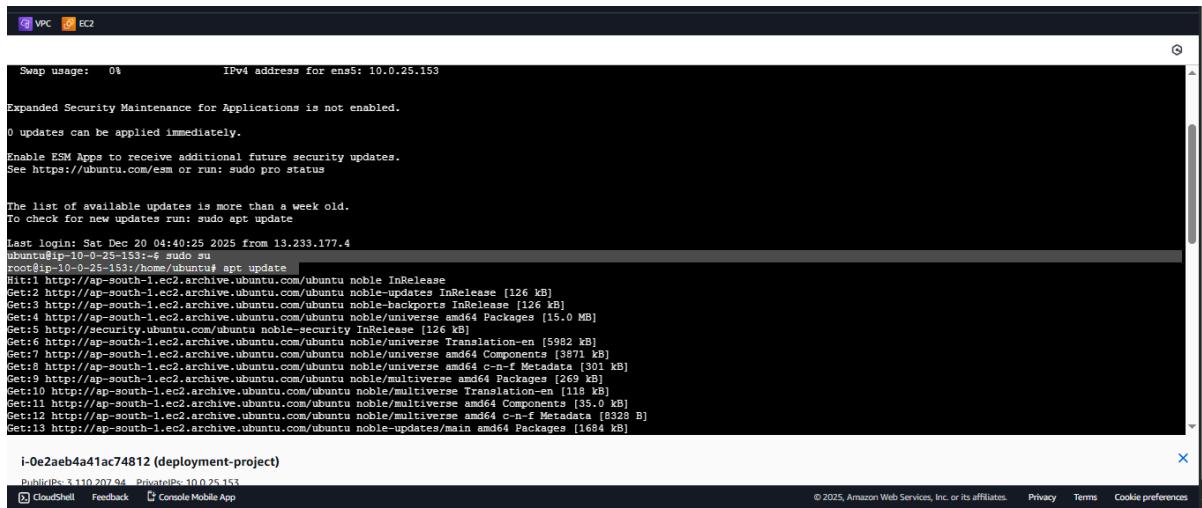
Run this **exact command** in the EC2 terminal:

```
sudo su
apt update
```

### What this does

- Updates system packages
- Prevents Docker installation issues

Wait until it completes.



```
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

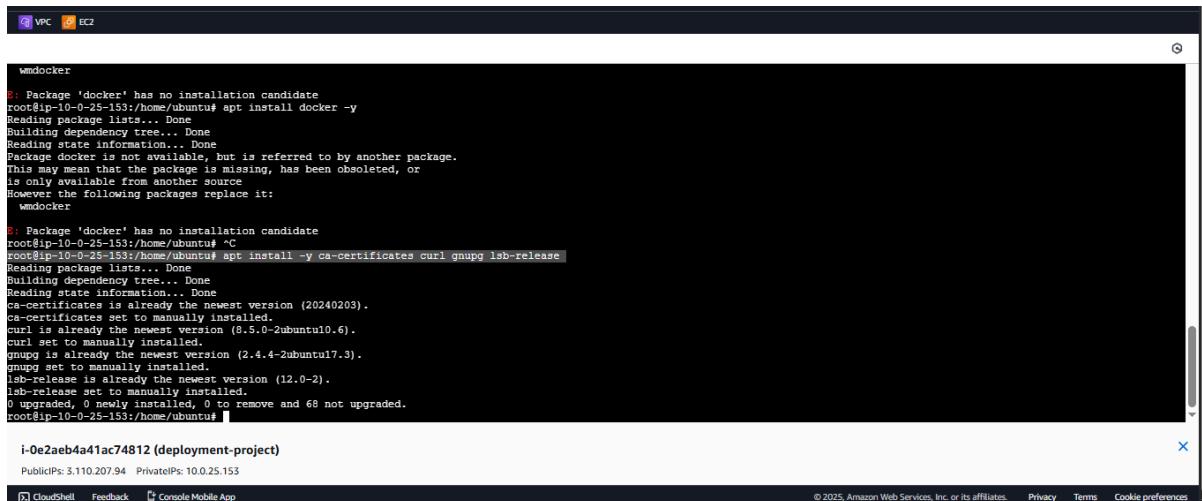
Last login: Sat Dec 20 04:40:25 2025 from 13.233.177.4
ubuntu@ip-10-0-25-153:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease [126 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu universe InRelease [15.0 MB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 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/multiverse amd64 Packages [269 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [111 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [1664 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1664 kB]

i-0e2aeb4a41ac74812 (deployment-project)

PublicIP: 3.110.207.94 PrivateIP: 10.0.25.153
```

## 3.2 Install Required Packages

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



```
wmdocker
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# 

i-0e2aeb4a41ac74812 (deployment-project)

PublicIP: 3.110.207.94 PrivateIP: 10.0.25.153
```

## 3.3 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:
  docker

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 -fsS https://download.docker.com/linux/ubuntu/gpg | gpg --dearmor -o /etc/apt/keyrings/docker.gpg
root@ip-10-0-25-153:/home/ubuntu#
```

i-0e2aeb4a41ac74812 (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

## 3.4 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:
  docker

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 -fsS 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=$(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
root@ip-10-0-25-153:/home/ubuntu#
```

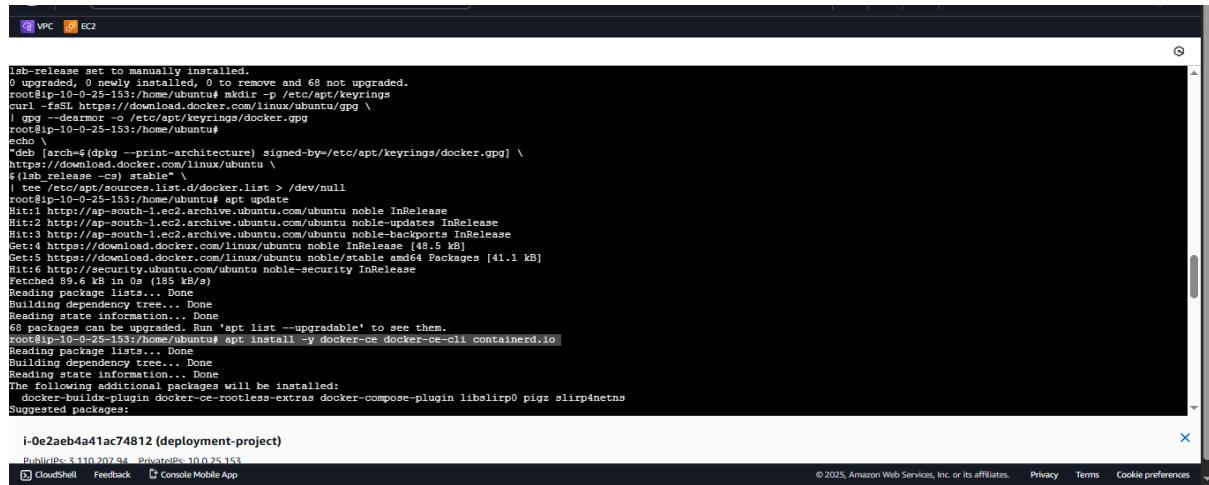
i-0e2aeb4a41ac74812 (deployment-project)  
Public IPs: 3.110.207.94 Private IPs: 10.0.25.153  
CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 3.5 Update Package List Again

```
apt update
```

## 3.6 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 stable" > /dev/null  
| 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/stable amd64 Packages [48.5 kB]  
Get:5 https://download.docker.com/linux/ubuntu/noble/stable amd64 Packages [41.1 kB]  
Hit:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-security InRelease  
Fetched 89.6 kB in 0s (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)
```

Public IPs: 3.110.207.94 Private IPs: 10.0.25.153

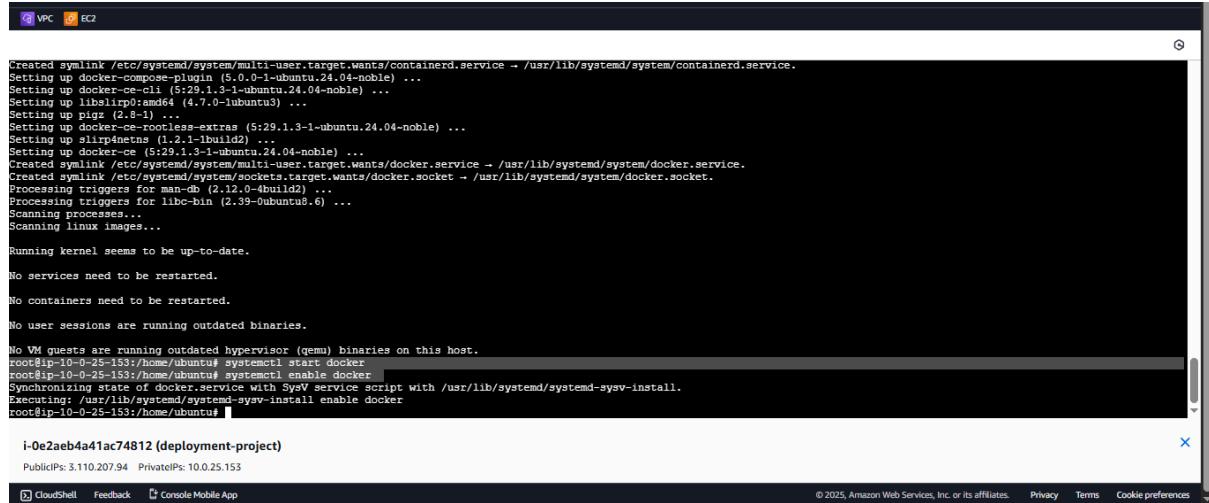
CloudShell Feedback Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 3.7 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 (5.0.0-1~ubuntu.24.04-noble) ...  
Setting up docker-ce-cli (5:29.1.3-1~ubuntu.24.04-noble) ...  
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...  
Setting up pigz (2.8-1) ...  
Setting up docker-ce-rootless-extras (5:29.1.3-1~ubuntu.24.04-noble) ...  
Setting up slirp4netns (1.2.1-1build2) ...  
Setting up docker-ce (5:29.1.3-1~ubuntu.24.04-noble) ...  
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.  
Processing triggers for man-db (2.12.0-4build2) ...  
Processing triggers for libc-bin (2.39-1ubuntu8.6) ...  
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 (qemu) 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: /var/lib/systemd/systemd-sysv-install enable docker  
root@ip-10-0-25-153:/home/ubuntu#  
  
i-0e2aeb4a41ac74812 (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

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 (qemu) 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/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
TriggeredBy: • docker.socket
    Docs: https://docs.docker.com
      Main PID: 3402 (dockerd)
        Tasks: 2
          Memory: 41.4M (peak: 42.3M)
            CPU: 528ms
          CGroup: /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="2025-12-20T05:00:05.246299885Z" level=info msg="Loading containers: done."
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.261074012Z" level=info msg="Docker daemon" commit=fbd3ed2 containerd-snapshotter=true storage-driver=overlayfs v
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.261074012Z" level=info msg="Initializing buildkit"
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.324328109Z" level=info msg="Buildkit has completed buildkit initialization"
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.343116001Z" level=info msg="Daemon has completed initialization"
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.345229359Z" 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-0e2ae84a41ac74812 (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

## 3.8 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
TriggeredBy: • docker.socket
    Docs: https://docs.docker.com
      Main PID: 3402 (dockerd)
        Tasks: 2
          Memory: 41.4M (peak: 42.3M)
            CPU: 528ms
          CGroup: /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="2025-12-20T05:00:05.246299885Z" level=info msg="Loading containers: done."
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.261074012Z" level=info msg="Docker daemon" commit=fbd3ed2 containerd-snapshotter=true storage-driver=overlayfs v
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.261074012Z" level=info msg="Initializing buildkit"
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.324328109Z" level=info msg="Buildkit has completed buildkit initialization"
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.343116001Z" level=info msg="Daemon has completed initialization"
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.345229359Z" 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-0e2ae84a41ac74812 (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-0e2ae84a41ac74812 > 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-0e2ae84a41ac74812 (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

**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.

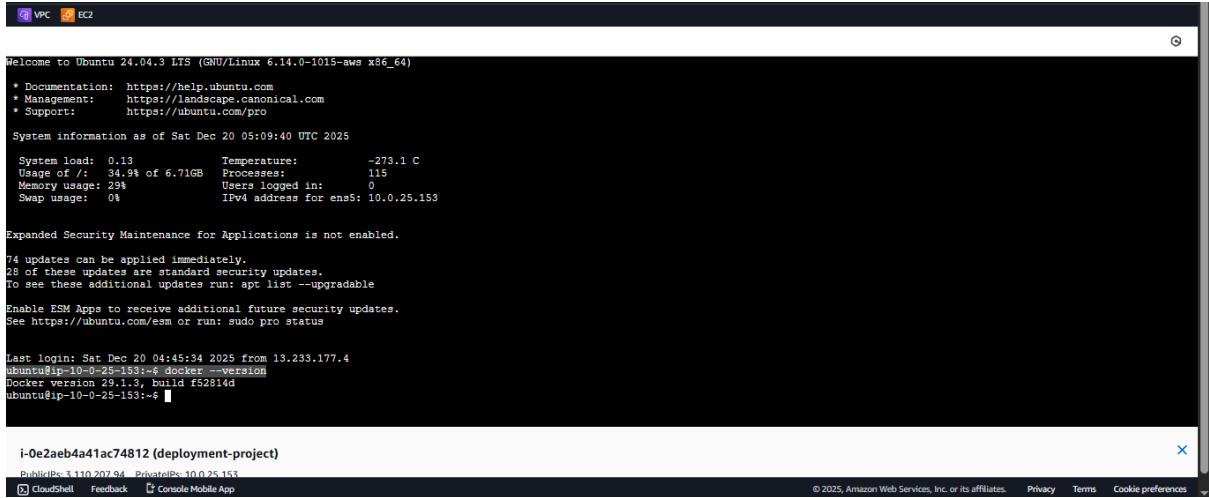
Cancel Connect

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 3.9 Verify Docker Installation

After reconnecting, run:

```
docker --version
```



The screenshot shows a terminal window titled "CloudShell" with the AWS logo. The window has tabs for "VPC" and "EC2". The terminal content is as follows:

```
Welcome to Ubuntu 24.04.3 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 04:45:34 2025 from 13.233.177.4
ubuntu@ip-10-0-25-153:~$ docker --version
Docker version 29.1.3, build f52814d
ubuntu@ip-10-0-25-153:~$
```

The terminal also shows the user's session ID: i-0e2aeb4a41ac74812 (deployment-project) and the IP address: 10.0.25.153.

## STEP 4: Get Your Website Code from GitHub (Inside EC2)

You are:

- Inside EC2 (browser terminal)
- Inside an **empty my-website/ directory**
- Website source code is already in **GitHub**
- Git is **not installed yet**

### STEP 4.1: Create a Project Directory on EC2

In the EC2 terminal, run:

```
mkdir my-website
cd my-website
```

Verify:

```
pwd
```

Expected:

/home/ubuntu/my-website



```
root@ip-10-0-25-153:/home/ubuntu# mkdir my-website
root@ip-10-0-25-153:/home/ubuntu# cd my-website/
root@ip-10-0-25-153:/home/ubuntu/my-website# pwd
/home/ubuntu/my-website
root@ip-10-0-25-153:/home/ubuntu/my-website#
```

i-0e2aeb4a41ac74812 (deployment-project)  
PrivateIPs: 3.110.207.94 PrivateIPv4: 10.0.25.153  
CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

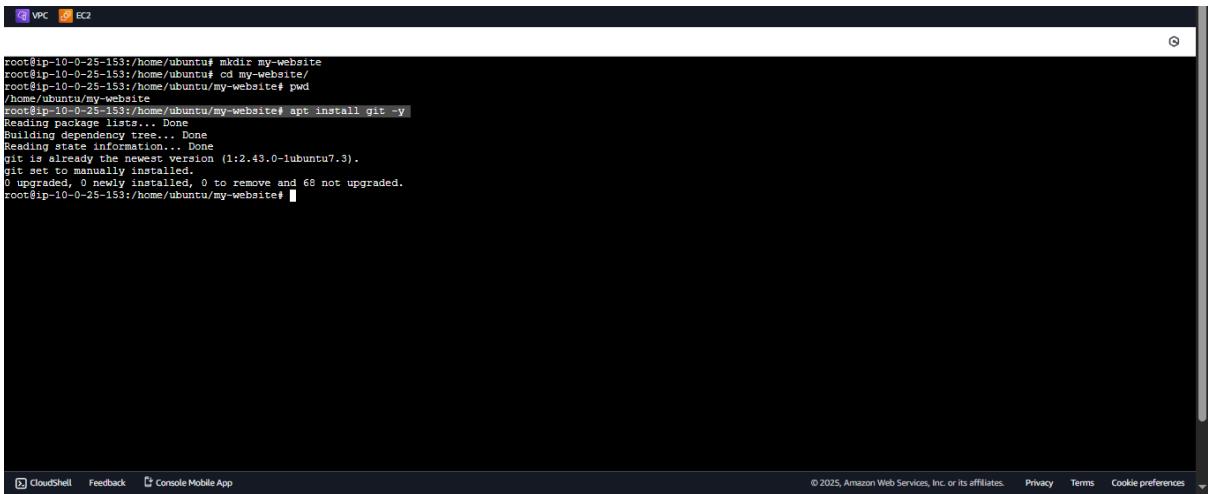
## 4.2 Install Git on EC2 (Ubuntu)

Run:

```
sudo apt install git -y
```

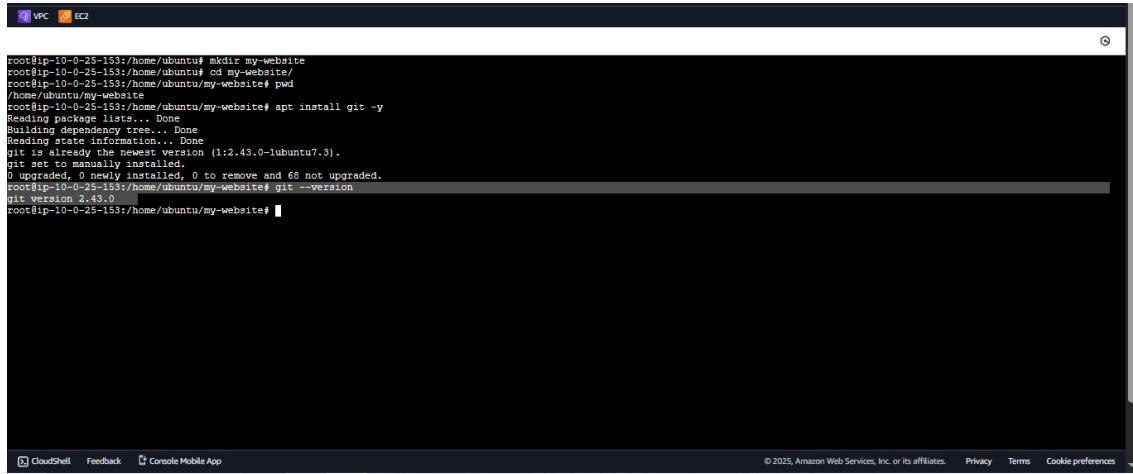
Verify:

```
git --version
```



```
root@ip-10-0-25-153:/home/ubuntu# mkdir my-website
root@ip-10-0-25-153:/home/ubuntu# cd my-website/
root@ip-10-0-25-153:/home/ubuntu/my-website# pwd
/home/ubuntu/my-website
root@ip-10-0-25-153:/home/ubuntu/my-website# apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-ubuntu7.3).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
root@ip-10-0-25-153:/home/ubuntu/my-website#
```

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



```
root@ip-10-0-25-153:/home/ubuntu# mkdir my-website
root@ip-10-0-25-153:/home/ubuntu# cd my-website/
root@ip-10-0-25-153:/home/ubuntu/my-website# pwd
/home/ubuntu/my-website
root@ip-10-0-25-153:/home/ubuntu/my-website# apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.3).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
root@ip-10-0-25-153:/home/ubuntu/my-website# git --version
git version 2.43.0
root@ip-10-0-25-153:/home/ubuntu/my-website#
```

## STEP 4.3: Clone Your Repo INTO Current Directory

```
git clone https://github.com/your-username/your-repo-name.git .
```

Example:

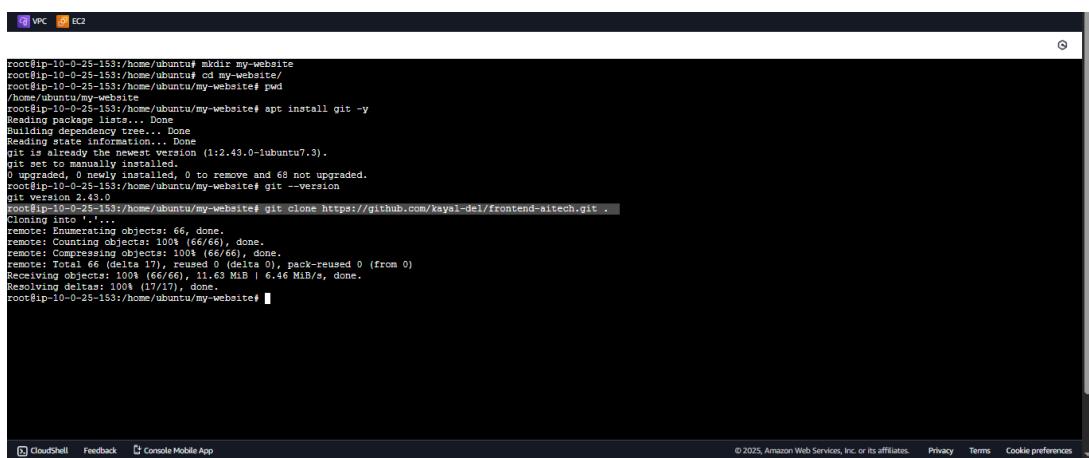
```
https://github.com/kayal-del/frontend-aitech
```

Run **this exact command** (note the dot . at the end):

```
git clone https://github.com/kayal-del/frontend-aitech.git .
```

### Why the . matters

- It clones **files directly into my-website/**
- Avoids extra folder nesting
- Docker build will work cleanly



```
root@ip-10-0-25-153:/home/ubuntu# mkdir my-website
root@ip-10-0-25-153:/home/ubuntu# cd my-website/
root@ip-10-0-25-153:/home/ubuntu/my-website# pwd
/home/ubuntu/my-website
root@ip-10-0-25-153:/home/ubuntu/my-website# apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.3).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
root@ip-10-0-25-153:/home/ubuntu/my-website# git --version
git version 2.43.0
root@ip-10-0-25-153:/home/ubuntu/my-website# git clone https://github.com/kayal-del/frontend-aitech.git .
Cloning into '.'...
remote: Enumerating objects: 66, done.
remote: Counting objects: 100% (66/66), done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 66 (delta 17), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (66/66), 11.63 MiB / 6.46 MiB/s, done.
Resolving deltas: 100% (17/17), done.
root@ip-10-0-25-153:/home/ubuntu/my-website#
```

```

root@ip-10-0-25-153:/home/ubuntu# mkdir my-website
root@ip-10-0-25-153:/home/ubuntu# cd my-website/
root@ip-10-0-25-153:/home/ubuntu/my-website# pwd
/home/ubuntu/my-website
root@ip-10-0-25-153:/home/ubuntu/my-website# apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.3).
git is set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
root@ip-10-0-25-153:/home/ubuntu/my-website# git --version
git version 2.43.0
root@ip-10-0-25-153:/home/ubuntu/my-website# git clone https://github.com/kayal-del/frontend-aitech.git .
Cloning into '.'.
remote: Enumerating objects: 66, done.
remote: Counting objects: 100% (66/66), done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 66 (delta 17), reused 0 (delta 0), pack-reused 0 (from 0).
Receiving objects: 100% (66/66) 16.63 MiB | 6.46 MiB/s, done.
Resolving deltas: 100% (17/17), done.
root@ip-10-0-25-153:/home/ubuntu/my-website# ls
'Cheese Nachos.html'          'Grilled Paneer.jpg'      contact.html     heart.jpg      'palaya sooru.html'   'salty food2.jpg'
'Cheese Nachos.jpg'           'Ice Cream Sundae.jpg'  dish1.jpg       hotel.jpg     panipuri.html    signup.html
'Chicken Swarma.html'         'Mini Burgers.html'    background1.jpg dish2.jpg     ice.html       style.css
'Chocolate Muffin.jpg'        'Nacho Fries.html'    background2.jpg dish3.jpg     index.html    style1.css
'Cold Noodles.html'           'Pasta Curry Kuli.html' cart.html      dish4.jpg     pickle.html   style1.css
'Fries.html'                  'Veggie Sandwich.jpg'  cart2.html    dish4.jpg     kuli.jpg      tandoori.jpg
'Fried Chicken.html'          'Vintage airmail envelope with stamps.png'  choco.mp4     dish4.jpg     lastmp4.html  tandoori.jpg
'Falafel Wrap.jpg.jpeg'       'about.html'          chesky.mp4     dish4.jpg     logo.png     popcorn.html  videotoplayback.mp4
'Fruit Salad.jpg'             'chicken swarma.jpg'  frenchfries.html dish4.jpg     muru.html    salil11.mp4
'Grilled Paneer.html'         'account.html'       'fruit salad.html' ordernow1.html 'salty food11.jpg'
root@ip-10-0-25-153:/home/ubuntu/my-website#

```

CloudShell Feedback + Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## STEP 5: Dockerize Your Website (Now We Package It)

Now we will:

1. Create a Dockerfile
2. Build the Docker image
3. Test it inside EC2

### STEP 5.1: Create Dockerfile

You are already inside:

/home/ubuntu/my-website

Create Dockerfile:

`vim Dockerfile`

Paste **exactly this**:

`FROM nginx:alpine`

`COPY . /usr/share/nginx/html`

`EXPOSE 80`

```
root@ip-10-0-25-153:/home/ubuntu# mkdir my-website
root@ip-10-0-25-153:/home/ubuntu# cd my-website/
root@ip-10-0-25-153:/home/ubuntu/my-website# pwd
/home/ubuntu/my-website
root@ip-10-0-25-153:/home/ubuntu/my-website# apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu3).
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
root@ip-10-0-25-153:/home/ubuntu/my-website# git --version
git version 2.43.0
root@ip-10-0-25-153:/home/ubuntu/my-website# git clone https://github.com/kayal-del/frontend-aitech.git .
Cloning into 'frontend-aitech'...
remote: Counting objects: 66, done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 66 (delta 17), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (66/66), 11.63 MiB | 6.46 MiB/s, done.
Resolving deltas: 100% (17/17), done.
root@ip-10-0-25-153:/home/ubuntu/my-website# ls
'Cheese Nachos.jpg'          'Grilled Panier.jpg'           appstore.png.jpeg   contact.html    heart.jpg      'palaya socrout.html' 'salty food2.jpg'
'Cheese Nachos.jpg'          'Ice Cream Sundae.jpg'        background1.jpg     dish1.jpg       hotel.jpg     'panipuri.html'    'signup.html'
'Chicken Swarma.html'        'Mini Burgers.html'         background2.jpg     dish2.jpg       ice.html      'payment.html'   'style.css'
'Chocolate Muffin.jpg'       'Mirage Burgers.jpg'        cart.html          dish3.html     index.html    'pickle.html'    'style1.css'
'Cold Foodie.jpg'            'Strawberry Kebab.html'    cart1.png         dish3.jpg      kufi1.jpg     'pickle.jpg'    'tan.mp4'
'Fajita Wrap.jpg.jpeg'       'Vintage airmail envelope with stamps.png'  'chicken pizza.html' dish4.html     kufi2.jpg     'popcorn.mp4'   'videoplayback.mp4'
'Fruit Salad.html'           'about.html'                 'chesy.mp4'        dish4.jpg      logo.png     'muru.html'     'sal111.mp4'
'Grilled Panier.html'        'account.html'             'choco.mp4'       'chicken swarma.jpg' frenchfries.html 'ordernow1.html' 'salty food11.jpg'
'Grilled Panier.html'        'account.html'             'choco.mp4'       'fruit salad.html'  muru.html     'sal111.mp4'
root@ip-10-0-25-153:/home/ubuntu/my-website# vim Dockerfile
root@ip-10-0-25-153:/home/ubuntu/my-website#
```

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

6,0-1 All

## STEP 5.2: Build Docker Image

Run:

```
docker build -t frontend-aitech .
```

Wait until it finishes.

Verify image:

## docker images

You should see:

frontend-aitech latest

## **STEP 5.3: Run Container (Local EC2 Test)**

Run:

```
docker run -d -p 80:80 frontend-aitech
```

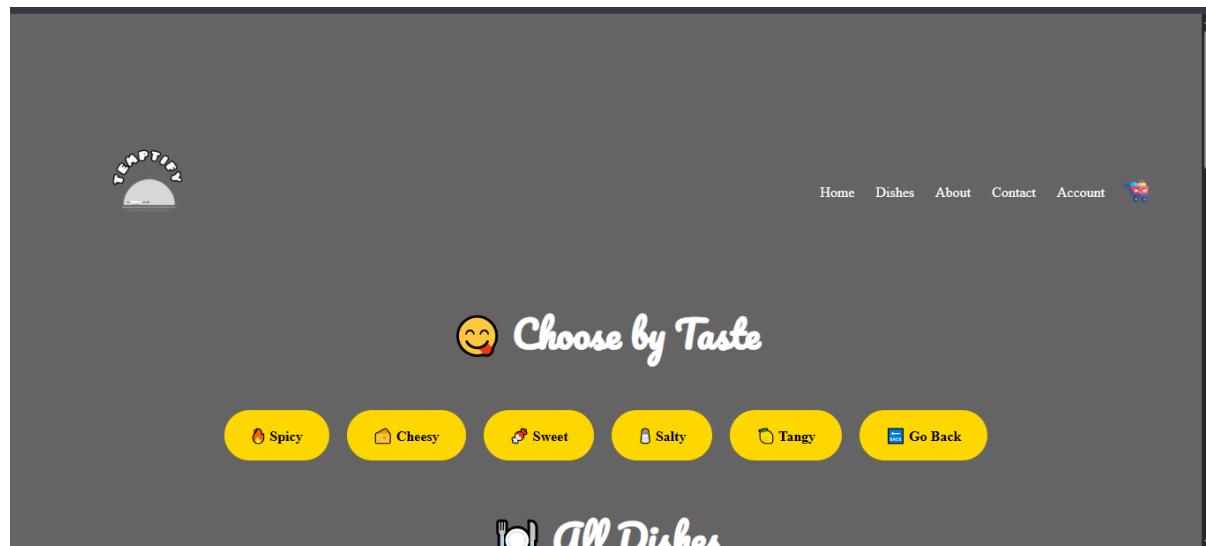
Check container:

docker ps

## **STEP 5.4: Test Website in Browser**

Open browser and visit:

[http://<EC2\\_PUBLIC\\_IP>](http://<EC2_PUBLIC_IP>)



Your website **homepage** should load.

Try clicking:

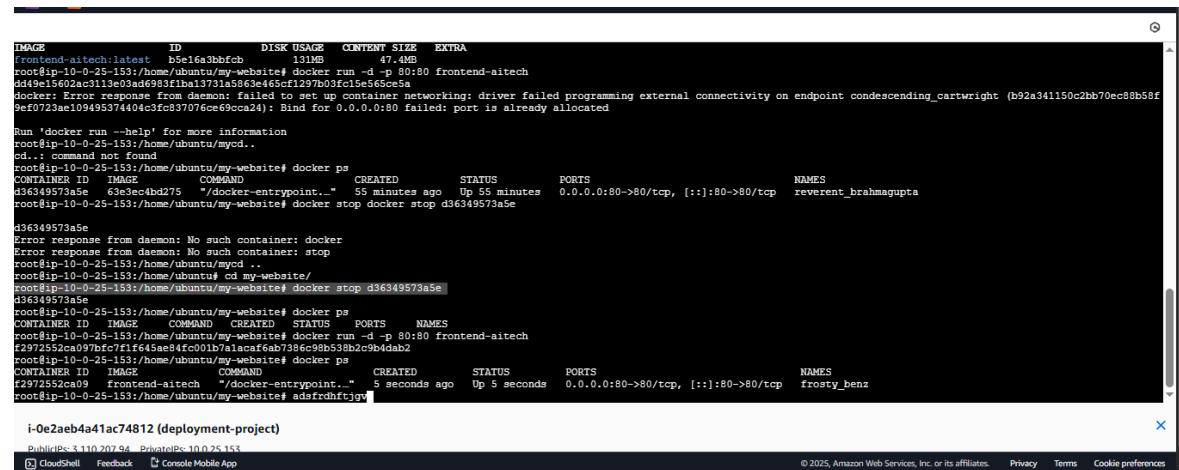
- Menu links
  - Other HTML pages
  - Images

Everything should work exactly like local.

# STEP 5.5: Stop Test Container

After confirming it works:

```
docker stop <container-id>
```

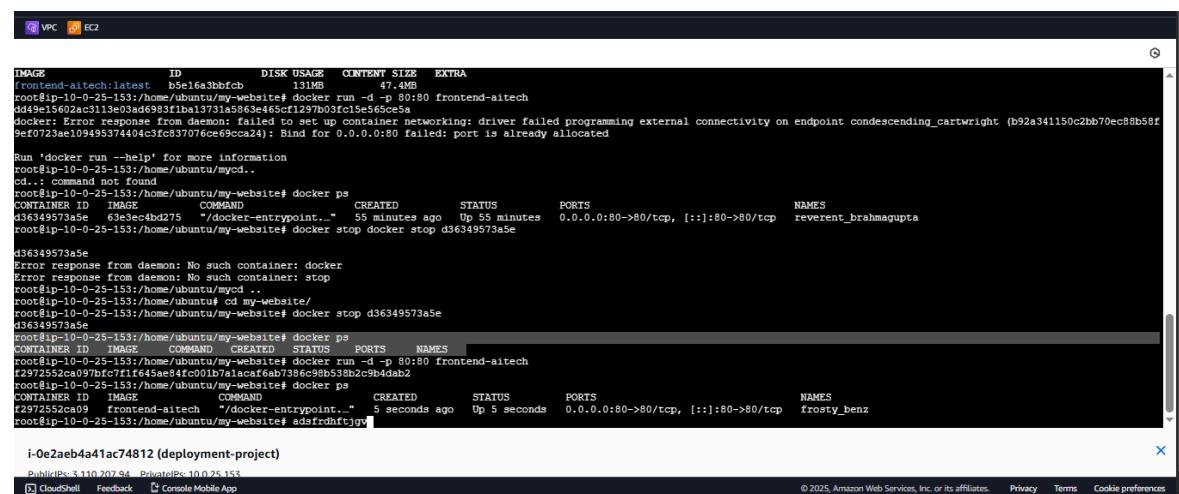


```
IMAGE ID      DISK USAGE   CONTENT SIZE  EXTRA
frontend-aitech:latest b5e16a3bbfcbb 131MB    47.4MB
root@ip-10-0-25-153:/home/ubuntu/my-website# docker run -d -p 80:80 frontend-aitech
dd49c15602a3c113e03ad6983f1ba13731a5863e465cf1297b03fc15e565ce5a
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint condescending_cartwright (b92a341150c2bb70ec88b58f
9ef0723ae109495374404c3fc837076ce69cc24): Bind for 0.0.0.80 failed: port is already allocated

Run 'docker run --help' for more information
root@ip-10-0-25-153:/home/ubuntu/mycd...
cd: command not found
root@ip-10-0-25-153:/home/ubuntu/my-website# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d36349573a5e 63e3ec4bd275 "/docker-entrypoint..." 55 minutes ago Up 55 minutes 0.0.0.80->80/tcp, [:]:80->80/tcp reverent_brahmagupta
root@ip-10-0-25-153:/home/ubuntu/my-website# docker stop d36349573a5e

d36349573a5e
Error response from daemon: No such container: docker
Error response from daemon: No such container: stop
root@ip-10-0-25-153:/home/ubuntu/mycd...
root@ip-10-0-25-153:/home/ubuntu/my-website# cd my-website/
root@ip-10-0-25-153:/home/ubuntu/my-website# docker stop d36349573a5e
root@ip-10-0-25-153:/home/ubuntu/my-website# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
root@ip-10-0-25-153:/home/ubuntu/my-website# docker run -d -p 80:80 frontend-aitech
f2972552ca097bf771f645ae84fc001b7acafab7386c9b538b2c9b4dab2
root@ip-10-0-25-153:/home/ubuntu/my-website# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
f2972552ca09 frontend-aitech "/docker-entrypoint..." 5 seconds ago Up 5 seconds 0.0.0.80->80/tcp, [:]:80->80/tcp frosty_benz
root@ip-10-0-25-153:/home/ubuntu/my-website# adsfrdhftjgv

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

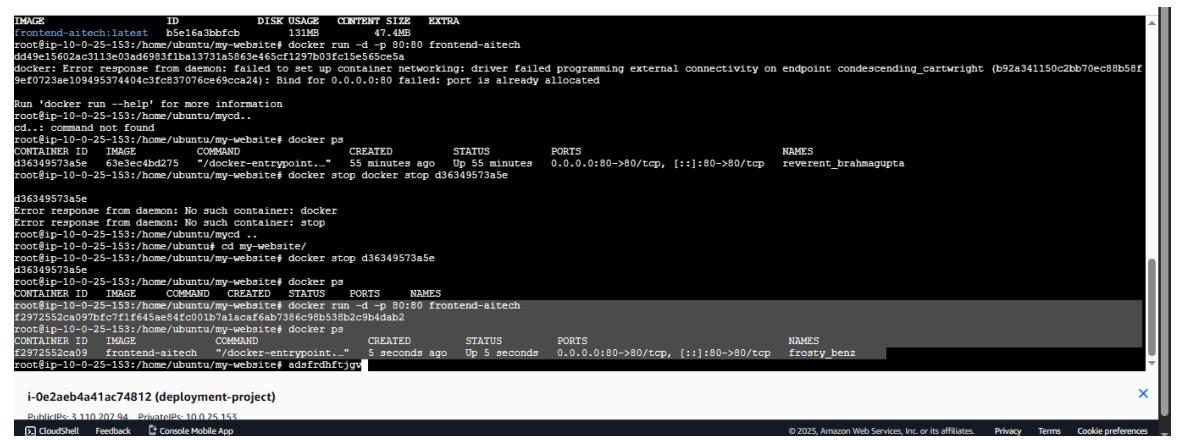


```
IMAGE ID      DISK USAGE   CONTENT SIZE  EXTRA
frontend-aitech:latest b5e16a3bbfcbb 131MB    47.4MB
root@ip-10-0-25-153:/home/ubuntu/my-website# docker run -d -p 80:80 frontend-aitech
dd49c15602a3c113e03ad6983f1ba13731a5863e465cf1297b03fc15e565ce5a
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint condescending_cartwright (b92a341150c2bb70ec88b58f
9ef0723ae109495374404c3fc837076ce69cc24): Bind for 0.0.0.80 failed: port is already allocated

Run 'docker run --help' for more information
root@ip-10-0-25-153:/home/ubuntu/mycd...
cd: command not found
root@ip-10-0-25-153:/home/ubuntu/my-website# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d36349573a5e 63e3ec4bd275 "/docker-entrypoint..." 55 minutes ago Up 55 minutes 0.0.0.80->80/tcp, [:]:80->80/tcp reverent_brahmagupta
root@ip-10-0-25-153:/home/ubuntu/my-website# docker stop d36349573a5e

d36349573a5e
Error response from daemon: No such container: docker
Error response from daemon: No such container: stop
root@ip-10-0-25-153:/home/ubuntu/mycd...
root@ip-10-0-25-153:/home/ubuntu/my-website# cd my-website/
root@ip-10-0-25-153:/home/ubuntu/my-website# docker stop d36349573a5e
root@ip-10-0-25-153:/home/ubuntu/my-website# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
root@ip-10-0-25-153:/home/ubuntu/my-website# docker run -d -p 80:80 frontend-aitech
f2972552ca097bf771f645ae84fc001b7acafab7386c9b538b2c9b4dab2
root@ip-10-0-25-153:/home/ubuntu/my-website# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
f2972552ca09 frontend-aitech "/docker-entrypoint..." 5 seconds ago Up 5 seconds 0.0.0.80->80/tcp, [:]:80->80/tcp frosty_benz
root@ip-10-0-25-153:/home/ubuntu/my-website# adsfrdhftjgv

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



```
IMAGE ID      DISK USAGE   CONTENT SIZE  EXTRA
frontend-aitech:latest b5e16a3bbfcbb 131MB    47.4MB
root@ip-10-0-25-153:/home/ubuntu/my-website# docker run -d -p 80:80 frontend-aitech
dd49c15602a3c113e03ad6983f1ba13731a5863e465cf1297b03fc15e565ce5a
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint condescending_cartwright (b92a341150c2bb70ec88b58f
9ef0723ae109495374404c3fc837076ce69cc24): Bind for 0.0.0.80 failed: port is already allocated

Run 'docker run --help' for more information
root@ip-10-0-25-153:/home/ubuntu/mycd...
cd: command not found
root@ip-10-0-25-153:/home/ubuntu/my-website# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d36349573a5e 63e3ec4bd275 "/docker-entrypoint..." 55 minutes ago Up 55 minutes 0.0.0.80->80/tcp, [:]:80->80/tcp reverent_brahmagupta
root@ip-10-0-25-153:/home/ubuntu/my-website# docker stop d36349573a5e

d36349573a5e
Error response from daemon: No such container: docker
Error response from daemon: No such container: stop
root@ip-10-0-25-153:/home/ubuntu/mycd...
root@ip-10-0-25-153:/home/ubuntu/my-website# cd my-website/
root@ip-10-0-25-153:/home/ubuntu/my-website# docker stop d36349573a5e
root@ip-10-0-25-153:/home/ubuntu/my-website# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
root@ip-10-0-25-153:/home/ubuntu/my-website# docker run -d -p 80:80 frontend-aitech
f2972552ca097bf771f645ae84fc001b7acafab7386c9b538b2c9b4dab2
root@ip-10-0-25-153:/home/ubuntu/my-website# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
f2972552ca09 frontend-aitech "/docker-entrypoint..." 5 seconds ago Up 5 seconds 0.0.0.80->80/tcp, [:]:80->80/tcp frosty_benz
root@ip-10-0-25-153:/home/ubuntu/my-website# adsfrdhftjgv

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

# STEP 6: Push Docker Image to Amazon ECR

This step has **3 sub-steps**:

1. Create ECR repository
2. Authenticate Docker to ECR
3. Tag & push image

We'll do them **one by one**.

## STEP 6.1: Create an ECR Repository

Go to AWS Console

- Open **ECR (Elastic Container Registry)**
- Click **Create repository**

### Repository settings

Repository name:

`frontend-aitech`

- Visibility: **Private**
- Leave everything else as default
- Click **Create repository**

The screenshot shows the 'Private repositories' section of the Amazon ECR console. On the left, there's a sidebar with navigation links for 'Amazon Elastic Container Service', 'Private registry' (selected), 'Public registry', 'ECR public gallery', 'Getting started', and 'Documentation'. The main area has a search bar and a table header with columns: 'Repository name', 'URI', 'Created at', 'Tag immutability', and 'Encryption type'. Below the header, it says 'No repositories' and 'No repositories were found'. At the top right, there are buttons for 'View push commands', 'Delete', 'Actions', and 'Create repository'.

This screenshot shows the first step of the 'Create a private repository' wizard. It's titled 'General settings'. It includes a 'Repository name' field containing '333675271195.dkr.ecr.ap-south-1.amazonaws.com/ frontend-aitech'. Below it are sections for 'Image tag settings' (with tabs for 'Info', 'Mutability', and 'Exclusions'), 'Mutable tag exclusions' (with a note about wildcards and a 'Add filter' button), and 'Encryption settings' (with a note that configurations can't be changed once created). The bottom right contains standard AWS footer links.

This screenshot shows the second step of the 'Create a private repository' wizard, specifically the 'Mutable tag exclusions' section. It lists a single exclusion: '333675271195.dkr.ecr.ap-south-1.amazonaws.com/ frontend-aitech'. Below this is a note about wildcards and a 'Add filter' button. The bottom right contains standard AWS footer links.

After creation, **COPY the Repository URI**.  
It will look like this:

3336xxxxxxxx.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech

Keep this copied — we'll use it in commands.

The screenshot shows the Amazon ECR console interface. On the left, there's a sidebar with navigation links for VPC, EC2, Amazon ECR (selected), Private registry, Public registry, ECR public gallery, Amazon ECS, Amazon EKS, Getting started, and Documentation. The main area is titled 'Private repositories (1)' and shows a single repository named 'frontend-aitech'. A green banner at the top indicates 'Successfully created private repository, frontend-aitech'. Below the banner, there's a search bar with 'Repository name' and 'Repository URI copied'. The repository details table includes columns for Repository name, URI, Created at, Tag immutability, and Encryption type. The repository URI is listed as 533673271195.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech. The 'Created at' column shows 20 December 2025, 12:27:07 (UTC+05:5). The 'Tag immutability' column shows 'Mutable', and the 'Encryption type' column shows 'AES-256'. At the bottom of the page, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

## STEP 6.2: Install AWS CLI v2

### Download AWS CLI v2 Installer

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o awscliv2.zip
```

The screenshot shows a terminal window with the following command and output:

```
root@ip-10-0-25-153:/home/ubuntu/my-website# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o awscliv2.zip
  % Total    % Received % Xferd  Average Speed   Time     Time  Current
          Dload  Upload Total Spent   Left Speed
100 60.2M  100 60.2M    0     0  328M  0:00:00 --:--:-- 329M
root@ip-10-0-25-153:/home/ubuntu/my-website# apt install unzip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
0 upgraded, 1 newly installed, 0 to remove and 46 not upgraded.
Need to get 174 kB of archives.
After this operation, 532 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 unzip amd64 6.0-28ubuntu4.1 [174 kB]
Fetched 174 kB in 0s (7439 kB/s)
Selecting previously unselected package unzip.
(Reading database ... 103632 files and directories currently installed.)
Preparing to unpack .../unzip 6.0-28ubuntu4.1 amd64.deb ...
Unpacking unzip (6.0-28ubuntu4.1) ...
Setting up unzip (6.0-28ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning triggers...
Scanning candidates...
Scanning linux images...
Pending kernel upgrade!
Running kernel version:
  6.14.0-1015-aws
```

### Install unzip (if not installed)

```
apt install unzip -y
```

```

root@ip-10-0-25-153:/home/ubuntu/my-website# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o awscliv2.zip
  % Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total Spent   Left Speed
100 60.2M  100 60.2M    0     0  328M  0:--:--:--:--:--:--: 329M
root@ip-10-0-25-153:/home/ubuntu/my-website# apt install unzip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
0 upgraded, 1 newly installed, 0 to remove and 46 not upgraded.
Need to get 174 kB of archives.
After this operation, 384 kB of additional disk space will be used.
Get:1 https://archive.ubuntu.com/ubuntu focal-updates/main amd64 unzip amd64 6.0-28ubuntu4.1 [174 kB]
Fetched 174 kB in 0s (7439 kB/s)
Selecting previously unselected package unzip.
(Reading database ... 103632 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-28ubuntu4.1_amd64.deb ...
Unpacking unzip (6.0-28ubuntu4.1) ...
Setting up unzip (6.0-28ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning crontriggers...
Scanning linux images...
Pending kernel upgrade!
Running kernel version:
 6.14.0-1015-aws

```

i-0e2aeb4a41ac74812 (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

## Unzip Installer

`unzip awscliv2.zip`

## Install AWS CLI

`sudo ./aws/install`

## Verify Installation

`aws --version`

Expected output:

`aws-cli/2.x.x Python/3.x Linux/x86_64`

```

inflating: aws/dist/awscli/customizations/wizard/wizards/iam/new-role.yml
inflating: aws/dist/awscli/customizations/sso/index.html
inflating: aws/dist/awscli/topics/dbb-expressions.rst
inflating: aws/dist/awscli/topics/return-codes.rst
inflating: aws/dist/awscli/topics/config-options.rst
inflating: aws/dist/awscli/topics/config-override.rst
inflating: aws/dist/awscli/topics/topicc-tags.json
inflating: aws/dist/awscli/topics/s3-faq.rst
  creating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/RECORD
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/INSTALLER
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/WHEEL
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/METADATA
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/top_level.txt
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/LICENSE
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/AUTHORS.rst
inflating: aws/dist/wheel-0.45.1.dist-info/WHEEL
inflating: aws/dist/wheel-0.45.1.dist-info/RECORD
inflating: aws/dist/wheel-0.45.1.dist-info/LICENSE.txt
inflating: aws/dist/wheel-0.45.1.dist-info/METADATA
inflating: aws/dist/wheel-0.45.1.dist-info/direct_url.json
inflating: aws/dist/wheel-0.45.1.dist-info/requires_points.txt
inflating: aws/dist/wheel-0.45.1.dist-info/INSTALLER
inflating: aws/dist/wheel-0.45.1.dist-info/REQUESTED
root@ip-10-0-25-153:/home/ubuntu/my-website# ./aws/install
You can now run: /usr/local/bin/aws --version
root@ip-10-0-25-153:/home/ubuntu/my-website# aws --version
aws-cli/2.32.21 Python/3.11.11 Linux/6.14.0-1015-aws exe/x86_64.ubuntu.24
root@ip-10-0-25-153:/home/ubuntu/my-website# 
```

i-0e2aeb4a41ac74812 (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 6.3: Configure AWS CLI

Now configure credentials:

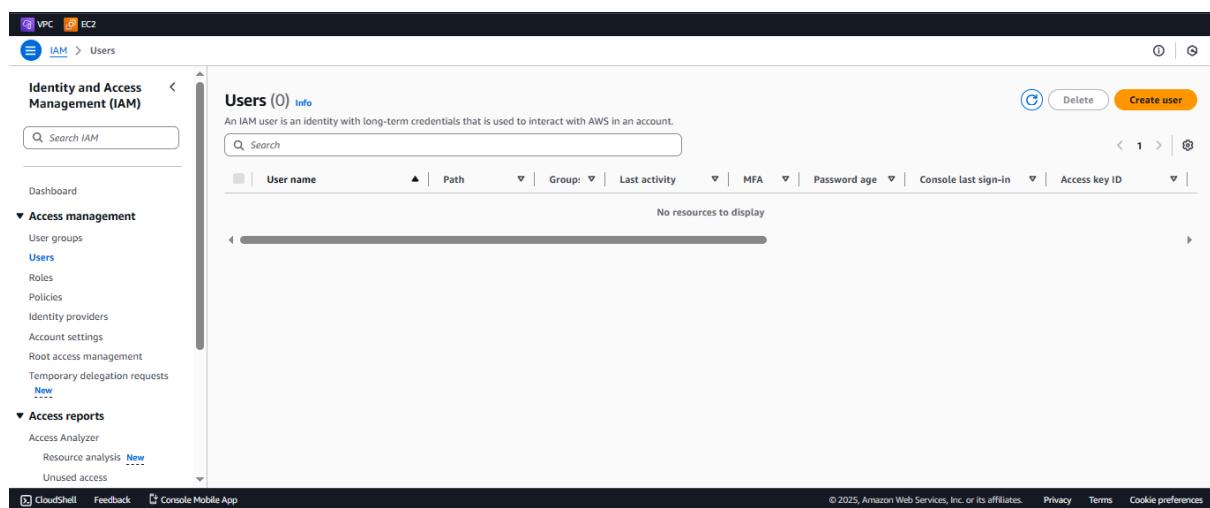
```
aws configure
```

Enter:

- **AWS Access Key ID** → your IAM user key
- **AWS Secret Access Key** → your secret key
- **Default region name** → ap-south-1
- **Default output format** → json

IAM user must have:

- **AmazonEC2ContainerRegistryFullAccess**
- **AmazonECSFullAccess**



VPC EC2

IAM > Users > Create user

Step 1 Specify user details

User name abi

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +, -, @, \_ (hyphen)

Provide user access to the AWS Management Console - optional

In addition to console access, users with SigninLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user. [Learn more](#)

Cancel Next

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

VPC EC2

IAM > Users > Create user

Step 1 Specify user details

Step 2 Set permissions

Step 3 Review and create

**Set permissions**

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Add user to group Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions Copy all group memberships, attached managed policies, and inline policies from an existing user.

Attach policies directly Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

**Permissions policies (1440)**

Choose one or more policies to attach to your new user.

Policy name	Type	Attached entities
AccessAnalyzerServiceRolePolicy	AWS managed	0
AccountManagementFromVercel	AWS managed	0
AdministratorAccess	AWS managed - job function	0

Create policy

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

VPC EC2

IAM > Users > Create user

Step 3 Review and create

**User details**

User name abi

Console password type None

Require password reset No

**Permissions summary**

Name	Type	Used as
AmazonEC2ContainerRegistryFullAccess	AWS managed	Permissions policy
AmazonEC2FullAccess	AWS managed	Permissions policy
AmazonECS_FullAccess	AWS managed	Permissions policy

**Tags - optional**

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Previous Create user

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**User created successfully**

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

[View user](#)

**Users (1) Info**

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Access key ID
abi	/	0	-	-	-	-	-

[Create user](#)

**Summary**

ARN  
arn:aws:iam::333673271195:user/abi

Created  
December 20, 2025, 12:49 (UTC+05:30)

Console access  
Disabled

Last console sign-in  
-

Access key 1  
[Create access key](#)

**Permissions** | Groups | Tags | Security credentials | Last Accessed

**Permissions policies (3)**

Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via
AmazonEC2ContainerRegistryFullAccess	AWS managed	Directly
AmazonEC2FullAccess	AWS managed	Directly

**Step 1**

**Access key best practices & alternatives**

Step 2 - optional

Set description tag

Step 3

Retrieve access keys

**Access key best practices & alternatives Info**

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

**Use case**

- Command Line Interface (CLI)**  
You plan to use this access key to enable the AWS CLI to access your AWS account.
- Local code**  
You plan to use this access key to enable application code in a local development environment to access your AWS account.
- Application running on an AWS compute service**  
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- Third-party service**  
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- Application running outside AWS**  
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
- Other**

VPC EC2

IAM > Users > abi > Create access key

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1

- Access key best practices & alternatives
- Step 2 - optional
- Set description tag
- Step 3
- Retrieve access keys**

## Retrieve access keys Info

**Access key**  
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAU3MD2B6NZRPN5XG	<span>XXXXXXXXXXXXXX</span> <a href="#">Show</a>

### Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

[Download .csv file](#) [Done](#)

## STEP 6.4: Authenticate Docker to Amazon ECR

You already created the ECR repository **frontend-aitech** (if not, do it now in the console).

## 6.4.1 Get your ECR Registry URI

From ECR → **Repositories** → **frontend-aitech**, copy the URI:

<ACCOUNT\_ID>.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech

I'll refer to it as:

**ECR\_URI**

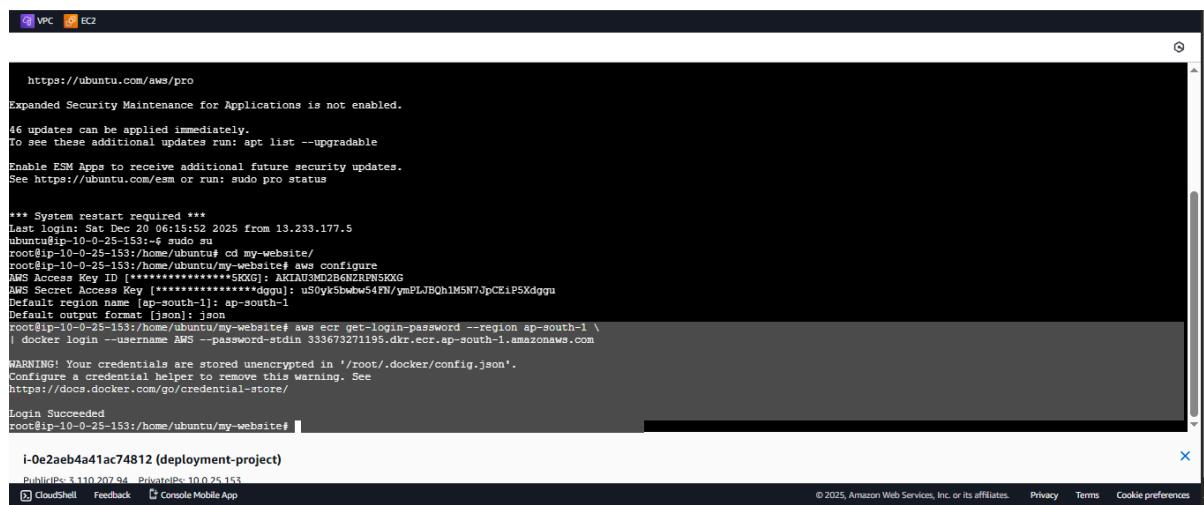
## 6.4.2 Login Docker to ECR (MOST IMPORTANT COMMAND)

Run this **exactly** (replace `ECR_URI` with your real one):

```
aws ecr get-login-password --region ap-south-1 \
| docker login --username AWS --password-stdin
<ACCOUNT_ID>.dkr.ecr.ap-south-1.amazonaws.com
```

### Expected output:

Login Succeeded



The screenshot shows a terminal window within the AWS CloudShell interface. The terminal output is as follows:

```
https://ubuntu.com/aws/pro
Expanded Security Maintenance for Applications is not enabled.
46 updates can be applied immediately.
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

*** System restart required ***
Last login: Sat Dec 20 06:15:52 2025 from 13.233.177.5
ubuntu@ip-10-0-25-153:~$ sudo su
root@ip-10-0-25-153:/home/ubuntu$ cd my-website/
root@ip-10-0-25-153:/home/ubuntu/my-website$ aws configure
AWS Access Key ID [*****5KXG]: ARIAU3MD2BNZRN5KXG
AWS Secret Access Key [*****dgu]: u5Oy5bwW54FN/ympLJBQh1M5N7JpCEiP5Xdggu
Default region [ap-south-1]: ap-south-1
Default profile [default]: default
root@ip-10-0-25-153:/home/ubuntu/my-website$ aws ecr get-login-password --region ap-south-1 \
| docker login --username AWS --password-stdin 333673271195.dkr.ecr.ap-south-1.amazonaws.com

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

Login Succeeded
root@ip-10-0-25-153:/home/ubuntu/my-website#
```

The terminal window has a dark background with light-colored text. The AWS CloudShell interface at the bottom includes tabs for CloudShell, Feedback, and Console Mobile App, along with links for Privacy, Terms, and Cookie preferences.

If this works → Docker can now push images to AWS.

## STEP 6.5: Tag Docker Image for ECR

Check your local image name:

```
docker images
```

You already have:

```
frontend-aitech:latest
```

```

46 updates can be applied immediately.
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

*** System restart required ***
Last login: Sat Dec 20 06:15:52 2025 from 13.233.177.5
ubuntu@ip-10-0-25-153:~$ sudo su
root@ip-10-0-25-153:/home/ubuntu# cd my-website/
root@ip-10-0-25-153:/home/ubuntu/my-website# aws configure
AWS Access Key ID [*****]: AKIAU3MD2BNZRN5KXG
AWS Secret Access Key [*****]: u5OykhMw54FN/ymPLBQh1MSN7JpCEiP5Xdggu
Default region name [ap-south-1]: ap-south-1
Default output format [json]: json
root@ip-10-0-25-153:/home/ubuntu/my-website# aws ecr get-login-password --region ap-south-1 \
| docker login --username AWS --password $stdin 333673271195.dkr.ecr.ap-south-1.amazonaws.com
WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Succeeded
root@ip-10-0-25-153:/home/ubuntu/my-website# docker images
Use
IMAGE           ID          DISK USAGE   CONTENT SIZE   EXTRA
frontend-aitech:latest b5e16a3bbfcb    131MB        47.4MB      [ ]
root@ip-10-0-25-153:/home/ubuntu/my-website# [REDACTED]

```

i-0e2aeb4a41ac74812 (deployment-project)

Public IP: 3.110.207.94 Private IP: 10.0.25.153

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

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

Now tag it for ECR:

`docker tag frontend-aitech:latest ECR_URI:latest`

Example:

`docker tag frontend-aitech:latest  
3336xxxxxx.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech:latest`

```

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

*** System restart required ***
Last login: Sat Dec 20 06:15:52 2025 from 13.233.177.5
ubuntu@ip-10-0-25-153:~$ sudo su
root@ip-10-0-25-153:/home/ubuntu# cd my-website/
root@ip-10-0-25-153:/home/ubuntu/my-website# aws configure
AWS Access Key ID [*****]: AKIAU3MD2BNZRN5KXG
AWS Secret Access Key [*****]: u5OykhMw54FN/ymPLBQh1MSN7JpCEiP5Xdggu
Default region name [ap-south-1]: ap-south-1
Default output format [json]: json
root@ip-10-0-25-153:/home/ubuntu/my-website# aws ecr get-login-password --region ap-south-1 \
| docker login --username AWS --password $stdin 333673271195.dkr.ecr.ap-south-1.amazonaws.com
WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Succeeded
root@ip-10-0-25-153:/home/ubuntu/my-website# docker images
Use
IMAGE           ID          DISK USAGE   CONTENT SIZE   EXTRA
frontend-aitech:latest b5e16a3bbfcb    131MB        47.4MB      [ ]
root@ip-10-0-25-153:/home/ubuntu/my-website# ^[[200-docker tag frontend-aitech:latest ECR_URI:latest
docker: command not found
root@ip-10-0-25-153:/home/ubuntu/my-website# docker tag frontend-aitech:latest 333673271195.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech:latest
root@ip-10-0-25-153:/home/ubuntu/my-website# [REDACTED]

```

i-0e2aeb4a41ac74812 (deployment-project)

Public IP: 3.110.207.94 Private IP: 10.0.25.153

## STEP 6.6: Push Image to ECR

`docker push ECR_URI:latest`

You'll see layer upload progress.  
Wait until it completes.

```
root@ip-10-0-25-153:/home/ubuntu/my-website# aws ecr get-login-password --region ap-south-1 \
| docker login --username AWS --password-stdin 333673271195.dkr.ecr.ap-south-1.amazonaws.com
WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Succeeded
root@ip-10-0-25-153:/home/ubuntu/my-website# docker images
Info - In
Use
IMAGE           ID             DISK USAGE   CONTENT SIZE   EXTRA
frontend-aitech:latest  b5e16a3bbfcb    131MB      47.4MB  [ ] 
root@ip-10-0-25-153:/home/ubuntu/my-website# |[200-docker tag frontend-aitech:latest ECR_URI:latest
root@ip-10-0-25-153:/home/ubuntu/my-website# docker push 333673271195.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech:latest
The push refers to repository [333673271195.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech]
d7c973d892: Pushed
3ffef8a0: Pushed
b5e16a3bbfcb: Pushed
de54ca821236: Pushed
1074353sec0d: Pushed
25f4530e4fd3: Pushed
0abf9e567266: Pushed
5aee5f8d6f3d: Pushed
50203434: Pushed
567f84da6f8fd: Pushed
latest: digest: sha256:b5e16a3bbfcbae82fce5c2d4f3256471ba93b8b43c59e9206a58aee5f8d04775 size: 856
root@ip-10-0-25-153:/home/ubuntu/my-website#
```

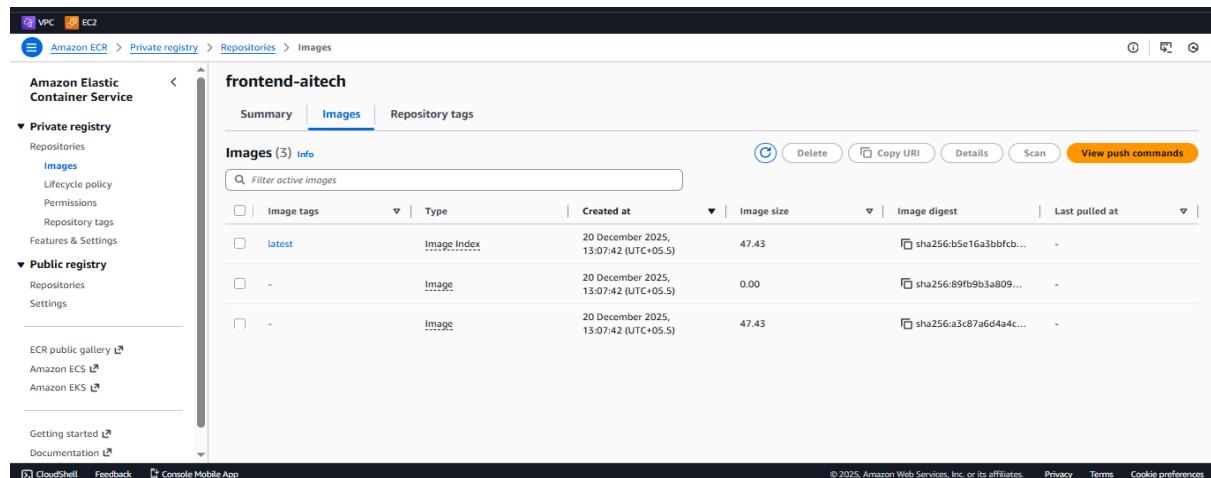
## **STEP 6.7: Verify in AWS Console**

Go to:

ECR → frontend-aitech → Images

You should see:

latest

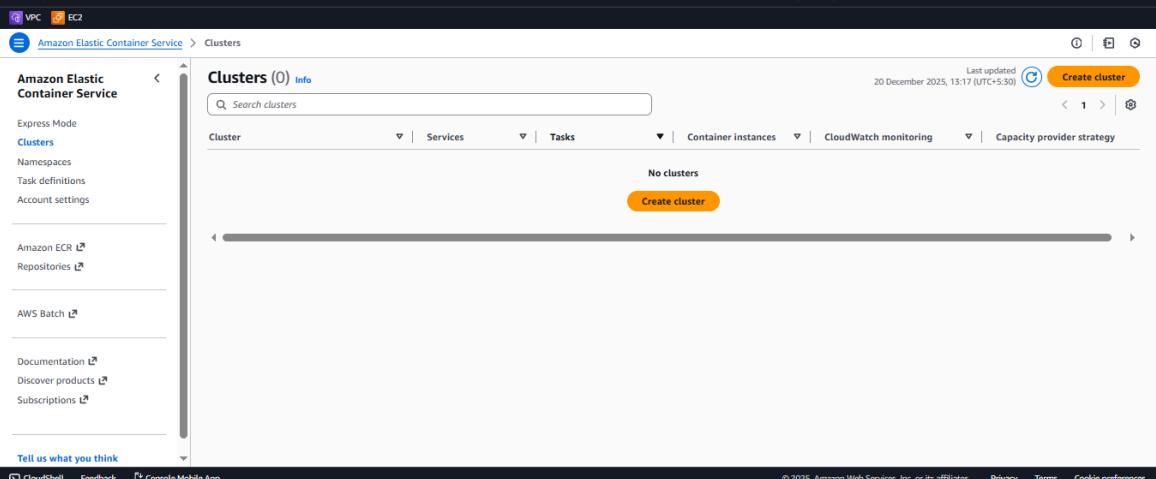


## **STEP 7: Create ECS Cluster (Fargate)**

This cluster will run your container image from ECR.

# STEP 7.1: Open ECS Console

AWS Console → Elastic Container Service (ECS)  
Click Clusters → Create cluster



The screenshot shows the AWS Elastic Container Service (ECS) Clusters page. On the left, there's a sidebar with links like 'Clusters', 'Namespaces', 'Task definitions', and 'Account settings'. The main area has a heading 'Clusters (0) Info' and a search bar. Below it, there are tabs for 'Cluster', 'Services', 'Tasks', 'Container instances', 'CloudWatch monitoring', and 'Capacity provider strategy'. A message 'No clusters' is displayed, followed by a large orange 'Create cluster' button. At the bottom, there's a footer with links for 'CloudShell', 'Feedback', and 'Console Mobile App', along with copyright information.

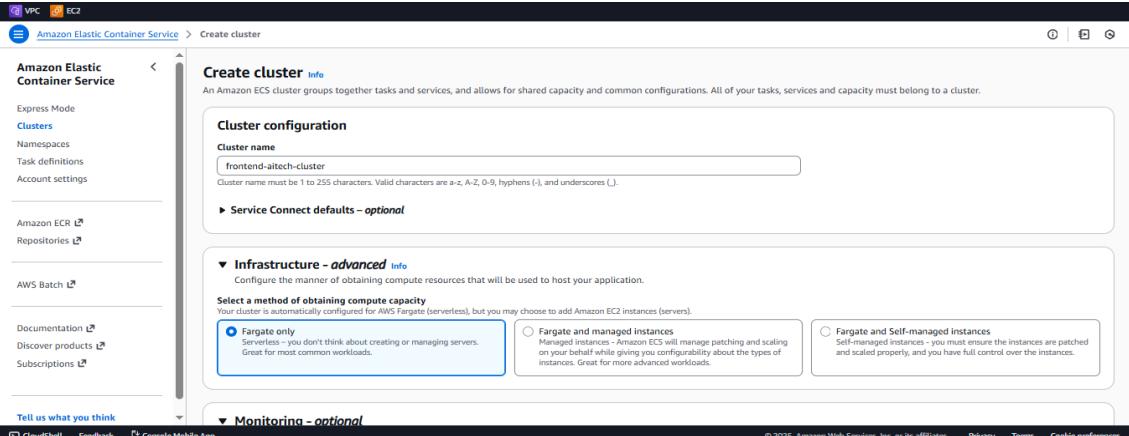
# STEP 7.2: Cluster Configuration

Fill the details as follows:

Setting	Value
Cluster name	frontend-aitech-cluster
Infrastructure	
Monitoring	Default

Click **Create**

Wait until cluster status = **Active**



The screenshot shows the 'Create cluster' configuration page. On the left, there's a sidebar with links like 'Clusters', 'Namespaces', 'Task definitions', and 'Account settings'. The main area has a heading 'Create cluster Info' with a note about cluster requirements. It then splits into sections: 'Cluster configuration' (with 'Cluster name' set to 'frontend-aitech-cluster'), 'Service Connect defaults - optional', 'Infrastructure - advanced' (with a note about compute capacity and three options: 'Fargate only' (selected), 'Fargate and managed instances', and 'Fargate and Self-managed instances'), and 'Monitoring - optional'. At the bottom, there's a footer with links for 'CloudShell', 'Feedback', and 'Console Mobile App', along with copyright information.

The screenshot shows the AWS ECS Clusters page. A green success message at the top states "Cluster frontend-aitech-cluster has been created successfully." Below this, the "Clusters (1) Info" section displays a table with one row for the "frontend-aitech-cluster". The table columns include Cluster, Services, Tasks, Container instances, CloudWatch monitoring, and Capacity provider strategy. The cluster details are: 0 services, 0 tasks running, 0 EC2 container instances, Default CloudWatch monitoring, and No default found for capacity provider strategy. The page also includes a sidebar with navigation links for VPC, EC2, Express Mode, Clusters, Namespaces, Task definitions, Account settings, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions. At the bottom, there are links for CloudShell, Feedback, and Console Mobile App.

The screenshot shows the AWS ECS Services page for the "frontend-aitech-cluster". A green success message at the top states "Cluster frontend-aitech-cluster has been created successfully." Below this, the "frontend-aitech-cluster" section displays a "Cluster overview" table with columns for ARN, Status (Active), CloudWatch monitoring (Default), and Registered container instances (-). The "Services" section shows a table with one row for "Draining" (Status: Active, Pending, Running). Below these sections is a navigation bar with tabs for Services, Tasks, Infrastructure, Metrics, Scheduled tasks, Configuration, Event history, and Tags. At the bottom, there is a "Services (0) Info" table with columns for Service name, ARN, Status, Schedu..., Launch..., Task de..., Deployments and tasks, and Last de... . The page also includes a sidebar with navigation links for VPC, EC2, Express Mode, Clusters, Namespaces, Task definitions, Account settings, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions. At the bottom, there are links for CloudShell, Feedback, and Console Mobile App.

# STEP 8: Create ECS Task Definition

Task definition = blueprint of your container.

## STEP 8.1: Create Task Definition

ECS → Task definitions → Create new task definition

Choose:

- Fargate

The screenshot shows the 'Task definitions' page of the Amazon Elastic Container Service. The left sidebar includes links for Express Mode, Clusters, Namespaces, Task definitions (which is selected), Account settings, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions. The main content area displays a table with one row for 'Task definition'. The table has columns for 'Task definition' and 'Status of last revision'. A message at the bottom says 'No task definitions' and 'No task definitions to display.' A 'Create new task definition' button is located at the bottom right of the table.

## STEP 8.2: Task Definition Settings

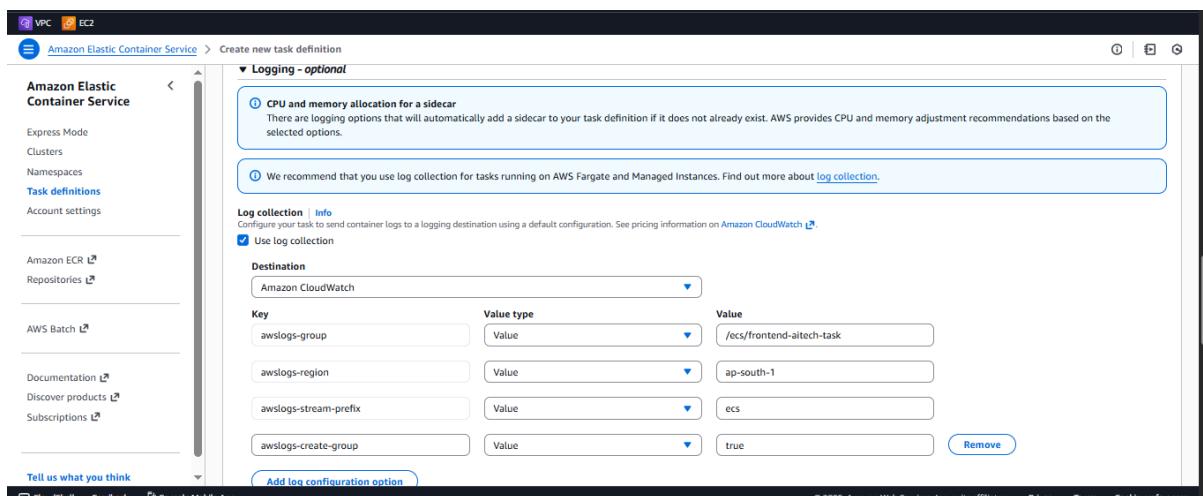
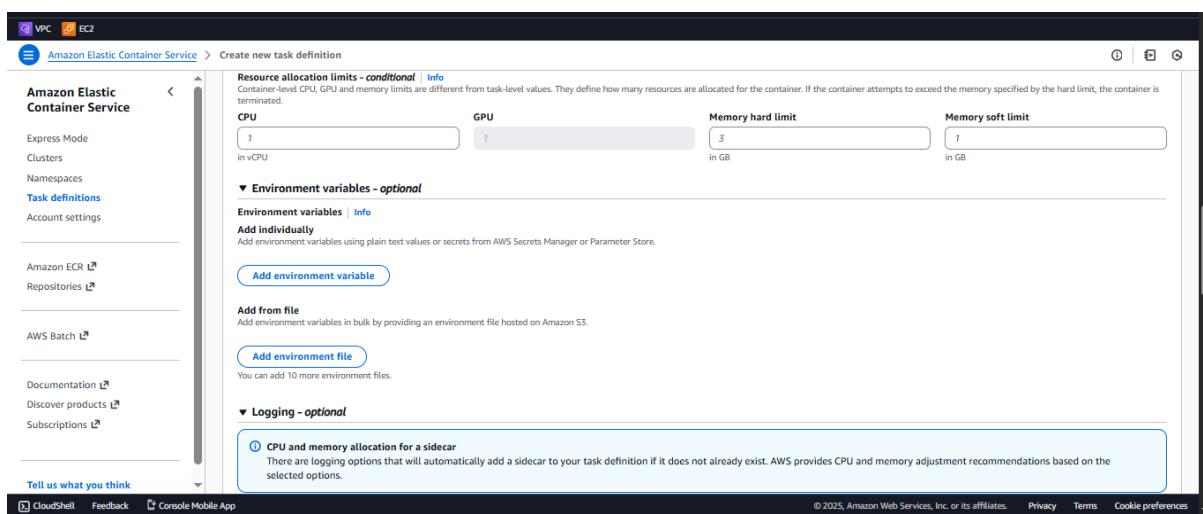
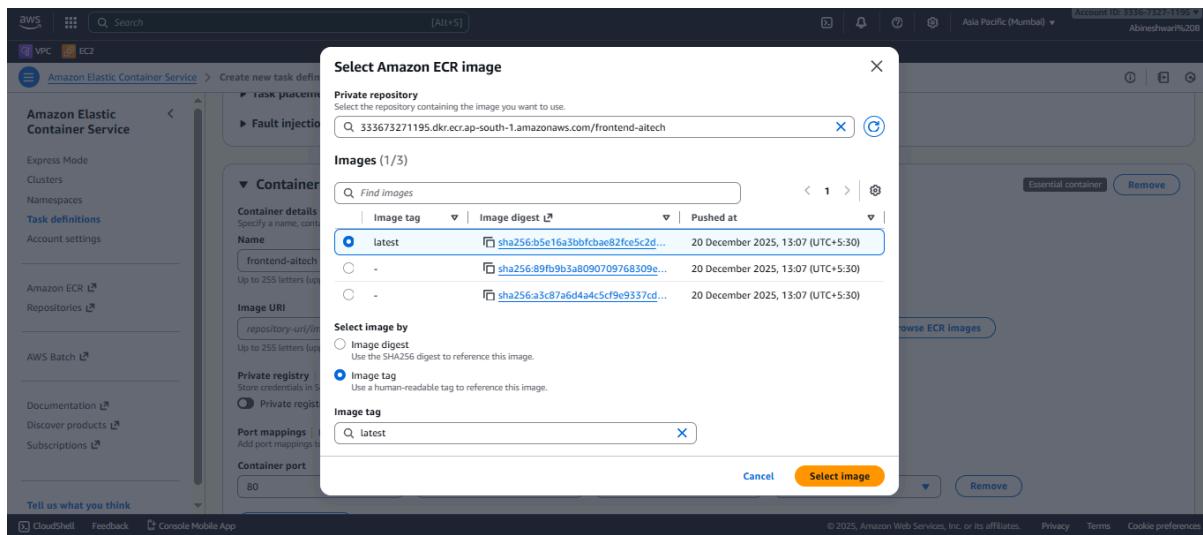
Setting	Value
Task definition name	frontend-aitech-task
Launch type	Fargate
Operating system	Linux
CPU	0.25 vCPU
Memory	0.5 GB

The screenshot shows the 'Create new task definition' page. The left sidebar is identical to the previous screenshot. The main form is titled 'Create new task definition'. It has several sections: 'Task definition configuration' (with a 'Task definition family' field containing 'frontend-aitech-task'), 'Infrastructure requirements' (with a 'Launch type' section showing 'AWS Fargate' selected), 'OS, Architecture, Network mode' (with tabs for 'Operating system/Architecture' and 'Network mode'), and a footer with standard AWS links.

## STEP 8.3: Container Configuration

Click **Add container**

Field	Value
Container name	frontend-aitech
Image	333673271195.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech:latest
Port mappings	80 / TCP
Essential	Yes



The screenshot shows the 'Create new task definition' interface in the AWS ECS console. On the left, a sidebar lists various services: Express Mode, Clusters, Namespaces, Task definitions (which is selected), Account settings, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions. The main panel is titled 'Create new task definition' and contains several optional configuration sections:

- Storage - optional**:
  - Ephemeral storage**: Info. The amount of ephemeral storage, in GiB, to allocate for the task. By default, your tasks hosted on AWS Fargate receive a minimum of 20 GiB of ephemeral storage.
  - Amount**: A text input field containing '21'. A note below says: 'To specify a custom amount of ephemeral storage, specify a value between 21 GiB up to a maximum of 200 GiB.'
  - Volumes**: Info. Add one or more data volumes for your task to provide additional storage for the containers in the task. For each data volume, you must add a mount point to specify where to mount the data volume in the container.
  - Add volume**: A button.
  - Volumes from**: Info. Mount data volumes from another container.
  - Add volume from**: A button.
- Monitoring - optional**: Configure your application trace and metric collection settings using the AWS Distro for OpenTelemetry integration.
- Tags - optional**: Info. Tags help you to identify and organise your task definitions.

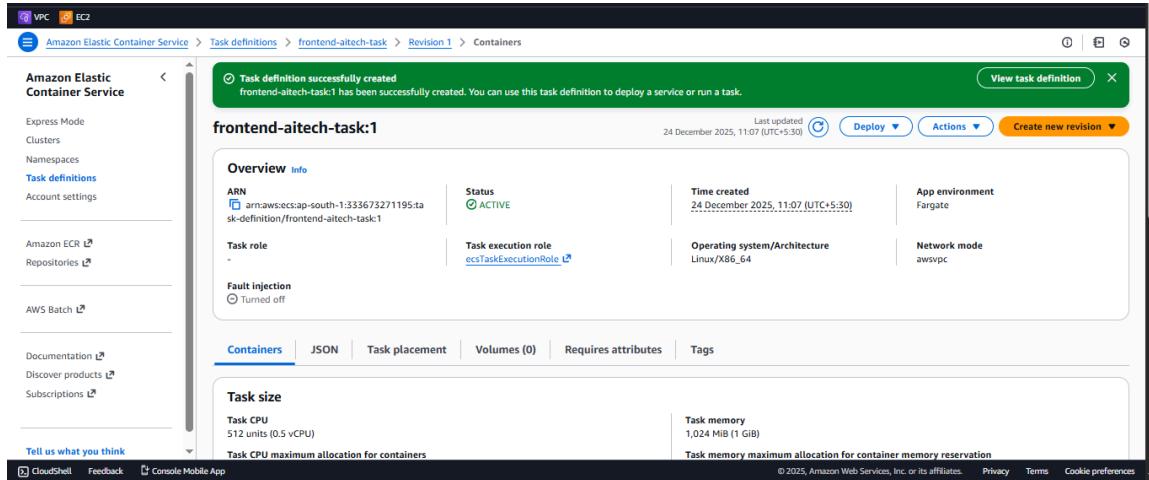
At the bottom right are 'Cancel' and 'Create' buttons. The footer includes links for CloudShell, Feedback, Console Mobile App, and navigation icons.

## Logging

- Enable CloudWatch logs
- Log group: auto-create

Click **Add**, then **Create task definition**

This screenshot shows the same 'Create new task definition' interface as the previous one, but with the 'Tags - optional' section expanded. The 'Create' button at the bottom right is highlighted in orange, indicating it is the next step. The rest of the interface is identical to the first screenshot, including the sidebar and other optional configuration sections.



## STEP 9: Create Target Group (for ALB → ECS)

This target group is where the **Application Load Balancer** will forward traffic.

Go to **EC2 → Target Groups → Create target group**

### Target Group Settings

Setting	Value
Target type	IP
Protocol	HTTP
Port	80
VPC	Default VPC
Health check path	/

- Click **Next**
- **DO NOT register targets manually**
- Click **Create target group**

VPC EC2

EC2 > Target groups

Target groups Info | What's new?

Filter target groups

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
No target groups You don't have any target groups in ap-south-1						

Create target group

0 target groups selected

Select a target group above.

Actions Create target group

CloudShell Feedback Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

VPC EC2

EC2 > Target groups > Create target group

Step 1 Create target group

Step 2 - recommended

Register targets

Step 3

Review and create

Create target group

A target group can be made up of one or more targets. Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Settings - immutable

Choose a target type and the load balancer and listener will route traffic to your target. These settings can't be modified after target group creation.

Instances

Supports load balancing to instances in a VPC. Integrate with Auto Scaling Groups or ECS services for automatic management.

Suitable for: ALB NLB GWLB

IP address

Supports load balancing to VPC and on-premises resources. Facilitates routing to IP addresses and network interfaces on the same instance. Supports IPv6 targets.

Suitable for: ALB NLB GWLB

Lambda function

Supports load balancing to a single Lambda function. ALB required as traffic source.

Suitable for: ALB

Application Load Balancer

Allows use of static IP addresses and PrivateLink with an Application Load Balancer. NLB required as traffic source.

Suitable for: NLB

Target group name

Name must be unique per Region per AWS account.

frontend-aitech-tg

Accepts: a-z, A-Z, 0-9, and hyphen (-). Can't begin or end with hyphen. 1-32 total characters; Count: 18/32

CloudShell Feedback Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

VPC EC2

EC2 > Target groups > Create target group

Name must be unique per Region per AWS account.

frontend-aitech-tg

Protocol

Protocol for communication between the load balancer and targets.

HTTP

Port

Port number where targets receive traffic. Can be overridden for individual targets during registration.

80

1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.

IPv4

IPv6

VPC

Select the VPC that hosts the load balancer. Only VPCs that support the IP address type selected above are available in this list. On the Register targets page, you can register IP addresses from this VPC, or from private IP addresses located outside of this load balancer's VPC (such as a peered VPC, EC2-Classic, or on-premises targets that are reachable over Direct Connect or VPN).

vpc-0ba916360b04e70d (deployment-project-VPC)

Create VPC

Protocol version

HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

HTTP2

Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

gRPC

CloudShell Feedback Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

VPC EC2

EC2 > Target groups > Create target group

**Protocol version**

**HTTP1**  
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

**HTTP2**  
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

**gRPC**  
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

**Health checks**

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

**Health check protocol**

HTTP

**Health check path**

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.  
/

Up to 1024 characters allowed.

**Advanced health check settings**

**Target optimizer - optional** Info

Use a target control port when the target has a strict concurrency limit.

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

VPC EC2

EC2 > Target groups > Create target group

**Advanced health check settings**

**Target optimizer - optional** Info

Use a target control port when the target has a strict concurrency limit.

**Target control port**

The port on which the target communicates its capacity. This value can't be modified after target group creation.

Enter target control port  
Valid range: 1-65535

**Attributes**

Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

**Tags - optional**

Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel Next

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

VPC EC2

EC2 > Target groups > Create target group

Step 1  
 Create target group  
 Step 2 - recommended  
 Register targets  
 Step 3  
 Review and create

**Register targets - recommended**

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

**IP addresses**

**Step 1: Choose a network**

You can add IP addresses from the VPC selected for your target group or from outside the VPC. Note that you can assemble a mix of targets from multiple network sources by returning to this step and choosing another network.

**Network**

deployment-project-VPC  
vpc-0ba916360bb-de70d  
IPv4 VPC CIDR: 10.0.0.0/16

**Step 2: Specify IPs and define ports**

You can manually enter IP addresses from the selected network.

Enter an IPv4 address from a VPC subnet.

Add IPv4 address

You can add up to 5 more IP addresses.

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

VPC EC2

EC2 > Target groups > Create target group

80  
1-65535 (separate multiple ports with commas)

Include as pending below

**Review targets**

**Step 3: Review IP targets to include in your group**

Confirm the IP targets to include in your target group. Add more IP targets by repeating steps 1 and 2 on this page. You can also register additional targets after your target group is created.

**Targets (0)**

Filter targets Show only pending Remove all pending

Remove IPv4 address Health status IP address Port Zone

No IP addresses included yet  
Specify IP addresses above and add to list.

0 pending

Cancel Previous Next

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

VPC EC2

EC2 > Target groups > Create target group

Step 1 Create target group  
Step 2 - recommended Register targets  
Step 3 Review and create

**Review and create**

Review your target group configuration before creating

**Step 1: Target group details**

**Target group details**

Name: frontend-aitech-tg	Target type: IP	Protocol: Port: HTTP: 80	Protocol version: HTTP1
VPC: vpc-0bba916560bb4e70d	IP address type: IPv4		

**Health check details**

Health check protocol: HTTP	Health check path: /	Health check port: traffic-port	Interval: 30 seconds
Timeout: 5 seconds	Healthy threshold: 5	Unhealthy threshold: 2	Success codes: 200

**Step 2: Register targets**

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

VPC EC2

EC2 > Target groups > Create target group

VPC: vpc-0bba916560bb4e70d

**Health check details**

Health check protocol: HTTP	Health check path: /	Health check port: traffic-port	Interval: 30 seconds
Timeout: 5 seconds	Healthy threshold: 5	Unhealthy threshold: 2	Success codes: 200

**Step 2: Register targets**

**Targets (0)**

IP address Port Zone

No targets added

Cancel Previous Create target group

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# STEP 10: Create Application Load Balancer (ALB)

Go to **EC2 → Load Balancers → Create load balancer**

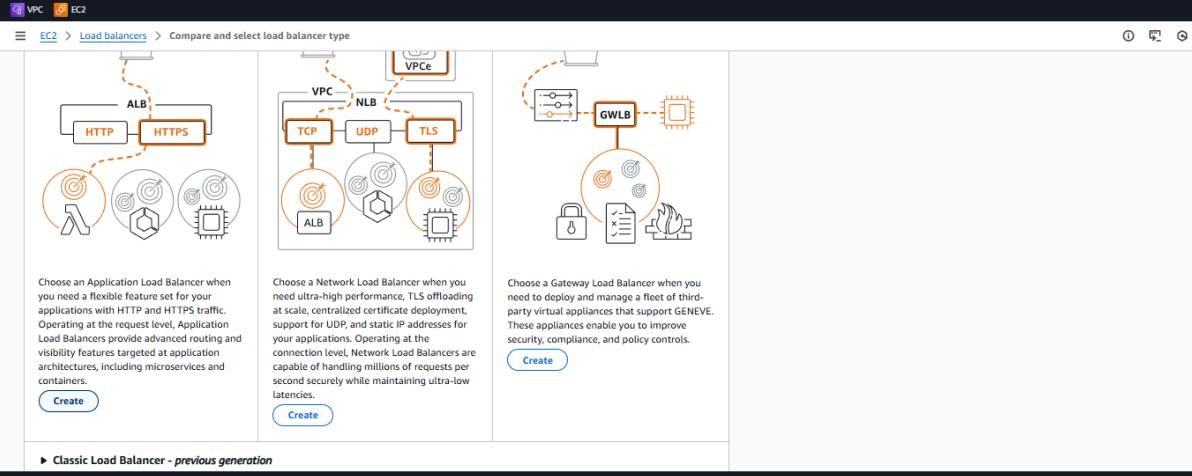
Choose:

- Application Load Balancer

## STEP 10.1: ALB Basic Configuration

Setting	Value
Name	frontend-aitech-alb

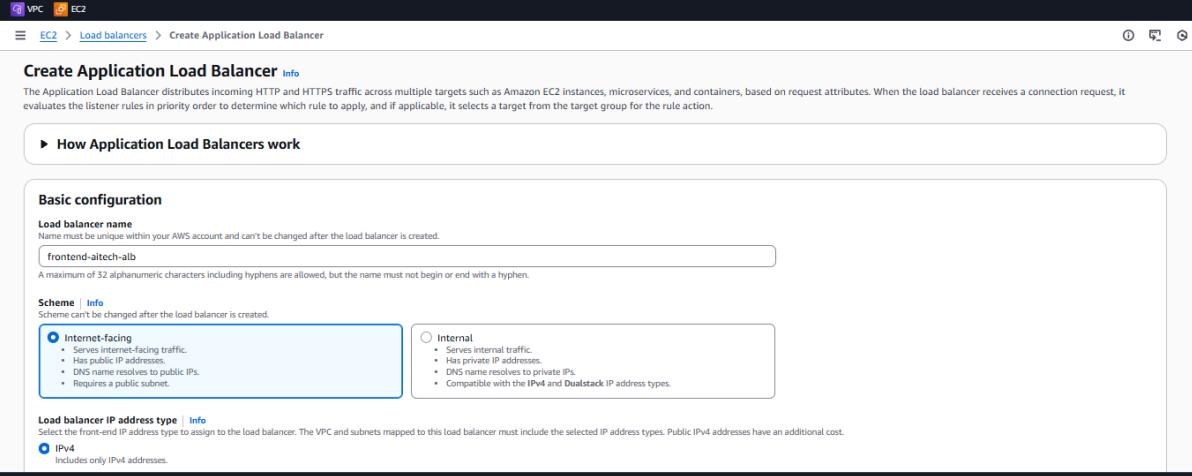
Scheme	<b>Internet-facing</b>
IP address type	IPv4
VPC	Default
Subnets	<b>Select at least 2 public subnets</b>



The screenshot shows the 'Compare and select load balancer type' section. It includes three cards:

- Application Load Balancer (ALB):** Handles HTTP and HTTPS traffic. Diagram shows ALB connecting to multiple targets (Amazon EC2 instances, microservices, containers) via TCP, UDP, and TLS.
- Network Load Balancer (NLB):** Handles ultra-high performance, TLS offloading, and centralized certificate deployment. Diagram shows NLB connecting to targets via TCP, UDP, and TLS.
- Gateway Load Balancer (GWLB):** Handles third-party virtual appliances supporting GENVE. Diagram shows GWLB connecting to targets via TCP, UDP, and TLS.

Each card has a 'Create' button.



The screenshot shows the 'Create Application Load Balancer' page. The 'Basic configuration' section includes:

- Load balancer name:** 'frontend-attach-alb'. A note says: 'Name must be unique within your AWS account and can't be changed after the load balancer is created.'
- Scheme:** 'Internet-facing' (selected) or 'Internal'.
- Load balancer IP address type:** 'IPv4' (selected).

Notes for 'Internet-facing' scheme:

- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name resolves to public IPs.
- Requires a public subnet.

Notes for 'Internal' scheme:

- Serves internal traffic.
- Has private IP addresses.
- DNS name resolves to private IPs.
- Compatible with the IPv4 and Dualstack IP address types.

**Load balancer IP address type** | [Info](#)  
 Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

**IPv4**  
 Includes only IPv4 addresses.

**Dualstack**  
 Includes IPv4 and IPv6 addresses.

**Dualstack without public IPv4**  
 Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with internet-facing load balancers only.

**Network mapping** | [Info](#)  
 The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

**VPC** | [Info](#)  
 The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view [target groups](#).

vpc-0bba916360bb4e70d (deployment-project-VPC)  
 10.0.0.0/16

[Create VPC](#)

**IP pools** | [Info](#)  
 You can optionally choose to configure an IPAM pool as the preferred source for your load balancer's IP addresses. Create or view Pools in the [Amazon VPC IP Address Manager console](#).

**Use IPAM pool for public IPv4 addresses**  
 The IPAM pool you choose will be the preferred source of public IPv4 addresses. If the pool is depleted, IPv4 addresses will be assigned by AWS.

**Availability Zones and subnets** | [Info](#)  
 Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

ap-south-1a (aps1-a21)

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

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

## STEP 10.2: Security Group for ALB(Optional)

Create or select a security group with:

Inbound :

HTTP | TCP | 80 | 0.0.0.0/0

Outbound: Allow all

## STEP 10.3: Listener & Routing

Setting	Value

Protocol	HTTP
Port	80
Forward to	<b>Target group created in STEP 9</b>

Click **Create load balancer**

Wait until **State = Active**

Listeners and routing [Info](#)  
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener **HTTP:80**

Protocol	HTTP	Port	80	<a href="#">Remove</a>
----------	------	------	----	------------------------

**Default action** | [Info](#)  
The default action is used if no other rules apply. Choose the default action for traffic on this listener.

**Routing action**

- Forward to target groups
- Redirect to URL
- Return fixed response

**Forward to target group** | [Info](#)  
Choose a target group and specify routing weight or [create target group](#).

**Target group**

frontend-altech-tg	HTTP	Weight	1	Percent	100%
Target type: IP, IPv4   Target stickiness: Off					

[+ Add target group](#)  
You can add up to 4 more target groups.

**Target group stickiness** | [Info](#)  
Enables the load balancer to bind a user's session to a specific target group. To use stickiness the client must support cookies. If you want to bind a user's session to a specific target, turn on the Target Group attribute Stickiness.

[Turn on target group stickiness](#)

Review the load balancer configurations and make changes if needed. After you finish reviewing the configurations, choose **Create load balancer**.

**Summary**  
Review and confirm your configurations. [Estimate cost](#)

<b>Basic configuration</b> <a href="#">Edit</a> Name: frontend-altech-alb Scheme: Internet-facing IP address type: IPv4	<b>Network mapping</b> <a href="#">Edit</a> VPC: vpc-0ba916360bb4e70d Public IPv4 IPAM pool: - Availability Zones and subnets: • ap-south-1a subnet-0624ce757b655ff9 deployment-project-VPCC-subnet-public1-ap-south-1a • ap-south-1b subnet-03f425e660184ff8b deployment-project-VPCC-subnet-public2-ap-south-1b	<b>Security groups</b> <a href="#">Edit</a> deployment-project-sg sg-0bd64cf7bd3b0a40d default sg-0dcc14845b66358c6	<b>Listeners and routing</b> <a href="#">Edit</a> HTTP:80   Forward to 1 target group
--	--	---	--

**Service integrations** [Edit](#)  
Amazon CloudFront + AWS Web Application Firewall (WAF): -  
AWS WAF: -  
AWS Global Accelerator: -

**Attributes**

[Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.](#)

The screenshot shows the 'Service integrations' section of the 'Create Application Load Balancer' wizard. It includes a table with one row containing the text: 'ap-south-1b', 'subnet-03f425e660184ff8b', 'deployment-project-VPC-subnet-public2-ap-south-1b'. Below the table, there's a note: 'Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.'

**Creation workflow and status**

**Server-side tasks and status**  
After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

**Create load balancer**

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the 'frontend-aitech-alb' load balancer details page. Key information includes:

- Load balancer type:** Application (Active)
- Status:** Active
- Hosted zone:** ZP97RAFLXNZK
- VPC:** vpc-0bba916360bb4c70d
- Availability Zones:** subnet-03f425e660184ff8b (ap-south-1b, ap-s1-az3), subnet-0624ce757b655ff9 (ap-south-1a, ap-s1-az1)
- Load balancer ARN:** arn:aws:elasticloadbalancing:ap-south-1:333673271195:loadbalancer/app/frontend-aitech-alb/b1bbf29dc9a9f6a9
- DNS name:** frontend-aitech-alb-10762547.ap-south-1.elb.amazonaws.com (A Record)

Listeners and rules (1) Manage rules Manage listener Add listener

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# STEP 11: Create ECS Service (Attach ALB)

Go to:

**Amazon ECS → Clusters → frontend-aitech-cluster → Create service**

## STEP 11.1: Service Configuration

Setting	Value
---------	-------

Launch type	Fargate
Task definition	frontend-aitech-task : 1
Service name	frontend-aitech-service
Desired tasks	1

The screenshot shows the 'Create service' step in the AWS Elastic Container Service wizard. On the left, a sidebar lists 'Amazon Elastic Container Service' options like Express Mode, Clusters, Namespaces, Task definitions, Account settings, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions. The main area is titled 'Create service' with a 'Info' link. It has two tabs: 'Service details' and 'Compute configuration - advanced'. Under 'Service details', the 'Task definition family' dropdown is set to 'frontend-aitech-task'. The 'Task definition revision' dropdown is set to 'Latest'. The 'Service name' input field contains 'frontend-aitech-task-service-of9juxqq'. Under 'Environment', it shows 'Existing cluster' set to 'frontend-aitech-cluster'. A note indicates that the service name must be unique within a cluster. The bottom right corner shows 'AWS Fargate'.

This screenshot shows the 'Compute configuration - advanced' section of the 'Create service' wizard. It includes fields for 'Compute options' (Capacity provider strategy), 'Launch type' (set to 'FARGATE'), 'Platform version' (set to 'LATEST'), and 'Deployment configuration'. The 'Launch type' field is highlighted with a blue border. The bottom right corner shows 'AWS Fargate'.

## STEP 11.2: Networking

Setting	Value
VPC	Default
Subnets	Public subnets
Public IP	<b>Disabled</b>
Security group (Task)	New or existing

### ECS Task Security Group – Inbound rule:

HTTP | TCP | 80 | ALB-Security-Group

## STEP 11.3: Load Balancing (IMPORTANT)

Field	Value
Load balancer type	Application Load Balancer
Listener	HTTP : 80
Target group	frontend-aitech-tg
Container name	frontend-aitech
Container port	80

Click **Create service**

Wait for service status = **RUNNING**

**Amazon Elastic Container Service**

**Clusters**

**Create service**

**Host port:Container port**

**Application Load Balancer**

Specify whether to create a new load balancer or choose an existing one.

Create a new load balancer

Use an existing load balancer

**Load balancer**

Choose an existing load balancer to distribute traffic. View existing load balancers and create new one in [EC2 Console](#).

frontend-aitech-alb  
frontend-aitech-alb-10762547.ap-south-1.elb.amazonaws.com

**Listener**

Specify the port and protocol that the load balancer will listen for connection requests on.

Create new listener

Use an existing listener

Listener: **HTTP-80**

**Listener rules for 80:HTTP (1)**

Traffic received by the listener is routed according to its rules. Rules are evaluated in priority order, from the lowest value to the highest value. The default rule is evaluated last.

Priority	Rule path	Target group
default	/	<a href="#">frontend-aitech-tg</a>

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

**Amazon Elastic Container Service**

**Clusters**

**Create service**

**Listener rules for 80:HTTP (1)**

Traffic received by the listener is routed according to its rules. Rules are evaluated in priority order, from the lowest value to the highest value. The default rule is evaluated last.

Priority	Rule path	Target group
default	/	<a href="#">frontend-aitech-tg</a>

**Target group**

Specify whether to create a new target group or choose an existing one that the load balancer will use to route requests to the tasks in your service.

Create new target group

Use an existing target group

Target group name: **frontend-aitech-tg**

Health check path: /

Health check protocol: **HTTP**

**VPC Lattice - optional**

Fully managed application networking service to connect, secure, and monitor your services across multiple accounts and virtual private clouds (VPCs). When you use VPC Lattice, there is a cost associated with it.

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

**Amazon Elastic Container Service**

**Clusters**

**Create service**

**HTTP**

**VPC Lattice - optional**

Fully managed application networking service to connect, secure, and monitor your services across multiple accounts and virtual private clouds (VPCs). When you use VPC Lattice, there is a cost associated with it.

**Service auto scaling - optional**

Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your service auto scaling configuration at any time to meet the needs of your application.

**Volume - optional**

Configure a data volume to provide additional storage for the containers in the task.

**Tags - optional**

Tags help you to identify and organise your resources.

**Create**

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

Amazon Elastic Container Service > Clusters > frontend-aitech-cluster > Services

**Cluster overview**

ARN: arn:aws:ecs:ap-south-1:333673271195:cluster/ frontend-aitech-cluster Status: Active CloudWatch monitoring: Default Registered container instances: -

**Services**

Draining	Active	Pending	Running
-	-	-	-

**Scheduled tasks**

Last updated: 24 December 2025, 11:44 (UTC+5:30)

Service name	ARN	Status	Schedul...	Launch...	Task de...	Deployments and tasks	Last de...
frontend-aitech-task-service-of9juxqq	arn:aws:ecs:ap-s...	Active	REPLICA	FARGATE	frontend-...	0/1 tasks running	...

Tell us what you think

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

You'll get an error after completing this.

Click your service → Update → Verify the below images.

Amazon Elastic Container Service > Clusters > frontend-aitech-cluster > Services > frontend-aitech-task-service-of9juxqq > Update

**Update front-end-aitech-task-service-of9juxqq**

**Deployment configuration**

Force new deployment

**Task definition family**  
Select an existing task definition. To create a new task definition, go to [Task definitions](#).  
frontend-aitech-task

**Task definition revision** [Latest](#)  
Select the task definition revision from the 100 most recent entries, or enter a revision. Leave the field blank to use the latest revision.  
Q 1 C

**Scheduling strategy**  
REPLICA

**Desired tasks**  
Specify the number of tasks to launch.  
1

**Availability Zone re-balancing** [Info](#)  
 Turn on Availability Zone re-balancing  
Amazon ECS automatically detects Availability Zone imbalances in task distributions across an ECS service, and evenly redistributes ECS service tasks across Availability Zones.

**Health check grace period** [Info](#)  
0 seconds

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon Elastic Container Service > Clusters > frontend-aitech-cluster > Services > frontend-aitech-task-service-of9juxqq > Update

**Deployment options**

**Deployment failure detection** [Info](#)

**Compute configuration - advanced**

**Troubleshooting configuration - recommended**

**Networking** [Info](#)  
Configure the subnets and security group for your service's tasks, and choose whether to automatically assign public IP addresses.

**VPC** [Info](#)  
Choose the virtual private cloud or create a new Virtual Private Cloud.  
vpc-0bb916360bb4e70d deployment-project-VPC C

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Amazon Elastic Container Service**

- Express Mode
- Clusters**
- Namespaces
- Task definitions
- Account settings

- Amazon ECR
- Repositories

- AWS Batch

- Documentation
- Discover products
- Subscriptions

Tell us what you think

VPC | EC2

Amazon Elastic Container Service > Clusters > frontend-aitech-cluster > Services > frontend-aitech-task-service-of9juxqq > Update

**Networking** Info

Configure the subnets and security group for your service's tasks, and choose whether to automatically assign public IP addresses.

**VPC** | Info

Choose the virtual private cloud or create a new Virtual Private Cloud.

vpc-0bb91916360b4e70d deployment-project-VPC

**Subnets** | Info

Choose existing subnets.

Choose subnets

subnet-03f1425e6e50184ff0 deployment-project-VPC-autonet-public2-ap-south-1b ap-south-1b 10.0.16.0/20

subnet-0624ce757b655ff19 deployment-project-VPC-subnet-public1-ap-south-1a ap-south-1a 10.0.0.0/20

**Security groups** | Info

Choose an existing security group or create a new security group.

Choose security groups

sg-0bd664cf7bd3b0a40d deployment-project-sg sg-0dcc14845b66358c6 default

**Public IP** | Info

Auto-assign a public IP to the task's elastic network interface (ENI)

Service Connect - optional

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Amazon Elastic Container Service**

- Express Mode
- Clusters**
- Namespaces
- Task definitions
- Account settings

- Amazon ECR
- Repositories

- AWS Batch

- Documentation
- Discover products
- Subscriptions

Tell us what you think

VPC | EC2

Amazon Elastic Container Service > Clusters > frontend-aitech-cluster > Services > frontend-aitech-task-service-of9juxqq > Update

**VPC Lattice - optional** Info

Fully managed application networking service to connect, secure, and monitor your services across multiple accounts and virtual private clouds (VPCs). When you use VPC Lattice, there is a cost associated with it.

**Load balancing - optional**

Configure load balancing using Amazon Elastic Load Balancing to distribute traffic evenly across the healthy tasks in your service.

**Service auto scaling - optional**

Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your service auto scaling configuration at any time to meet the needs of your application.

**Volume - optional** Info

Configure a data volume to provide additional storage for the containers in the task.

**Tags - optional**

Tags help you to identify and organise your clusters.

Cancel Update

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Amazon Elastic Container Service**

- Express Mode
- Clusters**
- Namespaces
- Task definitions
- Account settings

- Amazon ECR
- Repositories

- AWS Batch

- Documentation
- Discover products
- Subscriptions

Tell us what you think

VPC | EC2

Amazon Elastic Container Service > Clusters > frontend-aitech-cluster > Services > frontend-aitech-task-service-of9juxqq > Deployments

Service updated: frontend-aitech-cluster/frontend-aitech-task-service-of9juxqq

frontend-aitech-task-service-of9juxqq Info

Last updated 24 December 2025, 12:11 (UTC+5:30) Delete service Update service

**Service overview** Info

Status: Active	Tasks (1 Desired): 0 pending   1 running	Task definition: revision frontend-aitech-task:1	Deployment status: Success
----------------	--	--	----------------------------

Health and metrics Tasks Logs Deployments Events Configuration and networking Service auto scaling Event history Tags

**Last deployment** Info

Deployment ID: UAGPikqz08K7eBZjDTGoT	Deployment status: Success	Deployment controller type: ECS	Deployment strategy: Rolling update
Min and max running tasks: 100% min and 200% max	Deployment duration: 1 minute, 2 seconds	Created at: 24 December 2025, 12:10 (UTC+5:30)	Started at: 24 December 2025, 12:10 (UTC+5:30)
Stopped at: -	Finished at: 24 December 2025, 12:11 (UTC+5:30)		

**Service revisions** (2) Info

A service revision includes the number of tasks involved in the service deployment. You can choose to view details for all service revisions created on or after 24 October 2024.

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# STEP 12: Verify & Access Website

Go to EC2 → Load Balancers

1. Copy the DNS name of **frontend-aitech-alb**

Example:

`http://frontend-aitech-alb-xxxx.ap-south-1.elb.amazonaws.com`

Open it in your browser.

Your **HTML + CSS + Bootstrap + JS website is now LIVE via ECS + ALB**

The screenshot shows the AWS CloudFront console with the following details for the 'frontend-aitech-alb' load balancer:

- Details:**
  - Load balancer type: Application
  - Status: Active
  - Scheme: Internet-facing
  - Hosted zone: ZP97RAFLXTNZK
  - VPC: vpc-0ba916360b04c70d
  - Availability Zones:
    - subnet-03f425e660184ff8b (ap-south-1b)
    - subnet-0624ce757b655ff9 (ap-south-1a)
  - Load balancer ARN: arn:aws:elasticloadbalancing:ap-south-1:1333673271195:loadbalancer/app/frontend-aitech-alb/bbf29dc9a9f6a9
  - DNS name info: frontend-aitech-alb-10762547.ap-south-1.elb.amazonaws.com (A Record)- Listeners and rules:** 1 rule defined.

