

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

The screenshot shows the AWS EC2 Instances page. The top navigation bar includes the AWS logo, search bar, and account information (Account ID: 3336-7327-1195, Region: Asia Pacific (Mumbai), User: Abineshwari B). The left sidebar has a 'EC2' section with links like Dashboard, AWS Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Capacity Manager. The main content area is titled 'Instances Info' with a search bar and filters for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. It displays a message: 'No instances' and 'You do not have any instances in this region'. A prominent blue 'Launch instances' button is at the bottom of the list. Below it, a section titled 'Select an instance' is visible.

1.2 Choose AMI (Operating System)

Select:

Ubuntu

Why:

- AWS optimized
- Stable
- Docker works perfectly

Name and tags

Name: deployment-project

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64... [read more](#)
ami-02b8269d5e85954ef

Virtual server type (Instance type)

t3.micro

Summary

Number of instances: 1

Launch instance

1.3 Choose Instance Type

Select:

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

Click **Next / Continue**

Instance type

t3.micro
Family: t3 - 2 vCPU, 1 GB Memory, Current generation: true, On-Demand Linux base pricing: 0.0112 USD per Hour
On-Demand SUSE base pricing: 0.0112 USD per Hour, On-Demand Windows base pricing: 0.0204 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0147 USD per Hour, On-Demand RHEL base pricing: 0.04 USD per Hour

Key pair (login)

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

Key pair name - required: Select [Create new key pair](#)

Summary

Number of instances: 1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64... [read more](#)
ami-02b8269d5e85954ef

Virtual server type (Instance type)

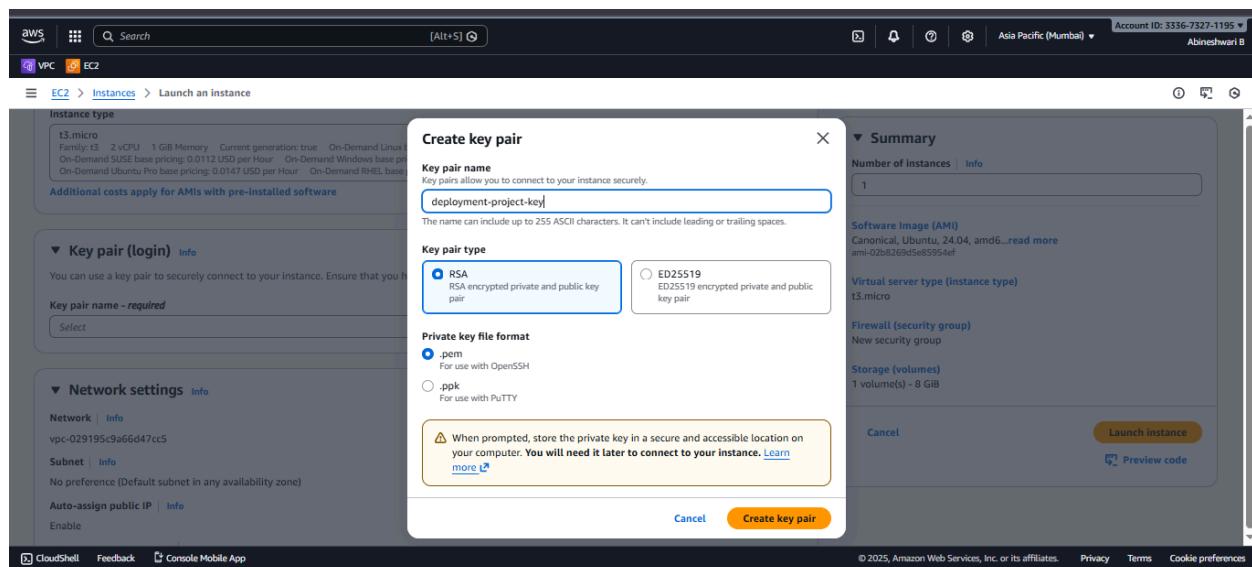
t3.micro

Launch instance

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 AWS EC2 'Launch an instance' wizard. In the 'Network settings' section, the 'Auto-assign public IP' option is selected. Other visible options include 'Network' (set to vpc-029195c9a66d47cc5), 'Subnet' (set to 'No preference'), and 'Auto-assign public IP' (set to 'Enable'). On the right side, the 'Summary' panel shows 1 instance being launched with the 'deployment-project-key.pem' key pair and the 't3.micro' instance type.

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 AWS VPC 'Create VPC' wizard. In the 'Preview' section, it displays the network structure. Under 'Subnets (4)', there are four subnets: 'deployment-project-VPC-subnet-' under 'ap-south-1a' and 'deployment-project-VPC-subnet-' under 'ap-south-1b'. Under 'Route tables (3)', there are three route tables: 'deployment-project-VPC-rtb-pul', 'deployment-project-VPC-rtb-pr1', and 'deployment-project-VPC-rtb-pr1'. The 'VPC settings' section shows the VPC is named 'deployment-project-VPN' and the CIDR block is '10.0.0.0/16'.

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.

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

Tenancy Info
 Default

Encryption settings - optional

Number of Availability Zones (AZs) Info
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.
 1 2 3

Customize AZs

Number of public subnets Info
The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.
 0 2

Preview

VPC Show details
Your AWS virtual network
deployment-project-VPC-vpc

Subnets (4)
Subnets within this VPC
ap-south-1a
deployment-project-VPC-subnet-
deployment-project-VPC-subnet-
ap-south-1b
deployment-project-VPC-subnet-
deployment-project-VPC-subnet-

Route tables (3)
Route network traffic to resources
deployment-project-VPC-rtb-pub
deployment-project-VPC-rtb-pr1
deployment-project-VPC-rtb-pr2

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 the endpoint policies.

Preview

Subnets (4)
Subnets within this VPC
ap-south-1a
deployment-project-VPC-subnet-
deployment-project-VPC-subnet-
ap-south-1b
deployment-project-VPC-subnet-
deployment-project-VPC-subnet-

Route tables (3)
Route network traffic to resources
deployment-project-VPC-rtb-public
deployment-project-VPC-rtb-private1
deployment-project-VPC-rtb-private2

Network connections (2)
Connections to other networks
deployment-project-VPC-igw
deployment-project-VPC-vpce-s2

NAT gateways (\$) - updated Info

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

None **Regional - new** **Zonal**

Introducing regional NAT gateway X

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

VPC endpoints Info

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

None **S3 Gateway**

DNS options Info

- Enable DNS hostnames
- Enable DNS resolution

Additional tags

Create VPC

The preview diagram illustrates the VPC structure. It shows four subnets under 'ap-south-1a' and four subnets under 'ap-south-1b'. Three route tables are present: 'deployment-project-VPC-rtb-public', 'deployment-project-VPC-rtb-private1-', and 'deployment-project-VPC-rtb-private2-'. Two network connections are shown: 'deployment-project-VPC-igw' and 'deployment-project-VPC-vpc-e-s'.

Details

- Create VPC: `vpc-0bba916360bb4e70d`
- Enable DNS hostnames
- Enable DNS resolution
- Verifying VPC creation: `vpc-0bba916360bb4e70d`
- Create S3 endpoint: `vpc-0c8177644f23a803b`
- Create subnet: `subnet-0624ce757b655ffff`
- Create subnet: `subnet-03f4256660184fbfb`
- Create subnet: `subnet-026b0b0360a5788bc`
- Create subnet: `subnet-01e19a410fb6a040b`
- Create internet gateway: `igw-03610fb7a9c41a3b4`
- Attach internet gateway to the VPC
- Create route table: `rtb-04d53d7a1b619b005`
- Create route
- Associate route table
- Associate route table
- Create route table: `rtb-0922888181857f389`
- Associate route table
- Create route table: `rtb-05f1a940a9ff91fb7`
- Associate route table
- Verifying route table creation
- Associate S3 endpoint with private subnet route tables: `vpc-0c8177644f23a803b`

View VPC

Your VPCs

Name	VPC ID	State	Encryption c...	Encryption control ...	Block Public...	IPv4 CIDR
vpc-029195c9a66d47cc5	vpc-029195c9a66d47cc5	Available	-	-	Off	172.31.0.0/16
deployment-project-VPC	vpc-0bba916360bb4e70d	Available	-	-	Off	10.0.0.0/16

Key pair (login)

Key pair name - required: deployment-project-key

Network settings

VPC - required: vpc-0bba916360bb4e70d (deployment-project-VPC)

Subnet: subnet-03f425e660184ff8b

Auto-assign public IP: Enable

Summary

Number of instances: 1

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

Launch instance

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

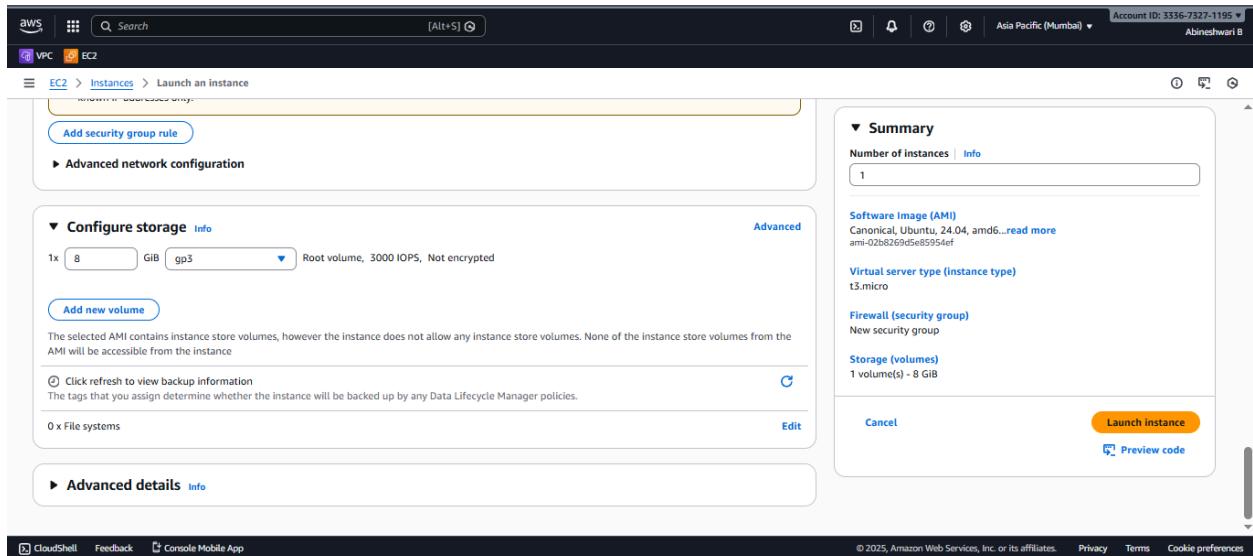
The screenshot shows the AWS EC2 'Launch an instance' wizard. On the left, under 'Inbound Security Group Rules', there is one rule: 'Security group rule 1 (TCP, 22, 0.0.0.0/0)'. The rule details are: Type: ssh, Protocol: TCP, Port range: 22, Source type: Anywhere. A note below states: '⚠️ Rules with 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.' On the right, the 'Summary' section shows 1 instance, the software image (ami-02b8269d5e8595def), the virtual server type (t3.micro), and the storage (1 volume(s) - 8 GiB). At the bottom right are 'Launch instance' and 'Preview code' buttons.

The screenshot shows the AWS EC2 'Launch an instance' wizard with two security group rules. Rule 1: 'Security group rule 2 (TCP, 0-65535, 0.0.0.0/0)' with Type: All TCP, Protocol: TCP, Port range: 0-65535, Source type: Custom. Rule 2: 'Security group rule 3 (TCP, 80, 0.0.0.0/0)' with Type: HTTP, Protocol: TCP, Port range: 80, Source type: Custom. A note below states: '⚠️ 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.' At the bottom right are 'Add security group rule', 'Launch instance', and 'Preview code' buttons.

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 'Launch an Instance' success page. At the top, there's a green success banner stating 'Successfully initiated launch of instance (i-0e2ae...). Below it, a 'Launch log' section is shown. A 'Next Steps' box contains links to various AWS services: 'Create billing usage alerts', 'Connect to your instance', 'Connect an RDS database', 'Create EBS snapshot policy', 'Manage detailed monitoring', 'Create Load Balancer', 'Create AWS budget', and 'Manage CloudWatch alarms'. The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and Console Mobile App.

The screenshot shows the AWS EC2 Instances page. On the left, a sidebar lists categories like Dashboard, AWS Global View, Events, Instances (selected), Images, and Elastic Block Store. The main area displays a table titled 'Instances (1/1) Info' with one row for the newly launched instance: 'i-0e2ae... deployment-project'. The instance is listed as 'Running' with an 't3.micro' type. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IPv4 . The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and Console Mobile App.

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.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with links for Dashboard, AWS Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, AMI Catalog, and Elastic Block Store Volumes. The main content area displays a table titled "Instances (1/1) Info". The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IPv4 . The single instance listed is named "deployment-project", with Instance ID i-0e2ae... and State Running. It is a t3.micro instance in ap-south-1b with Public IPv4 DNS ec2-3-110-207-94.ap-south-1.compute.amazonaws.com and Public IPv4 3.110.207.94. Below the table, a detailed view for instance i-0e2ae... is shown, including tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The Details tab shows the Instance ID i-0e2ae... and Instance state Running. The Public IPv4 address is 3.110.207.94. The Public DNS is ec2-3-110-207-94.ap-south-1.compute.amazonaws.com.

The screenshot shows the AWS EC2 Instance Connect page. At the top, it says "Connect to instance | EC2 | ap-south-1 | vpcs | VPC Console | DevOps - Docker ECS website | Website Deployment - ECS Pro". The main content area is titled "EC2 Instance Connect" and shows the instance details: Instance ID i-0e2ae... (deployment-project). It provides two connection options: "Connect using a Public IP" (selected) and "Connect using a Private IP". Under "Public IPv4 address", it shows 3.110.207.94. There's a note about the default username "ubuntu". At the bottom, there are "Cancel" and "Connect" buttons. The browser's address bar shows the URL ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ConnectToInstance:instanceId=i-0e2ae... The footer includes standard AWS links like CloudShell, Feedback, and Console Mobile App, along with system status icons and the date/time 20-12-2025 10:09.

```
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 04:39:57 UTC 2025

System load: 0.15 Temperature: -273.1 C
Usage of /: 25.9% of 6.71GB Processes: 111
Memory usage: 228 Users logged in: 0
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 04:39:58 2025 from 13.233.177.5
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-25-153:~$ i-0e2ae84a41ac74812 (deployment-project)
PublicIPs 3.110.207.94 PrivateIPs 10.0.25.153
```

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

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.

```
aws VPC EC2 Search [Alt+Shift] Account ID: 3336-7327-1195 Asia Pacific (Mumbai) Abinashwari B

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 logon: Sat Dec 20 04:40:25 2025 from 13.233.177.4
ubuntu@ip-10-0-25-153:~$ sudo su
root@ip-10-0-25-153:~/home/ubuntu$ apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
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 noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.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 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [111 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [155 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1684 kB]

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

3.2 Install Required Packages

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

```
wmdocker
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

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|
```

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
```

The screenshot shows a terminal session in AWS CloudShell. The user has run the command to download the Docker GPG key and save it to the /etc/apt/keyrings directory. The output shows standard apt-get dependency resolution and package installation logs.

```
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:
  wodocker

E: Package 'docker' has no installation candidate
root@ip-10-0-25-153:/home/ubuntu# curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
| gpg --dearmor -o /etc/apt/keyrings/docker.gpg
root@ip-10-0-25-153:/home/ubuntu#
```

3.4 Add Docker Repository

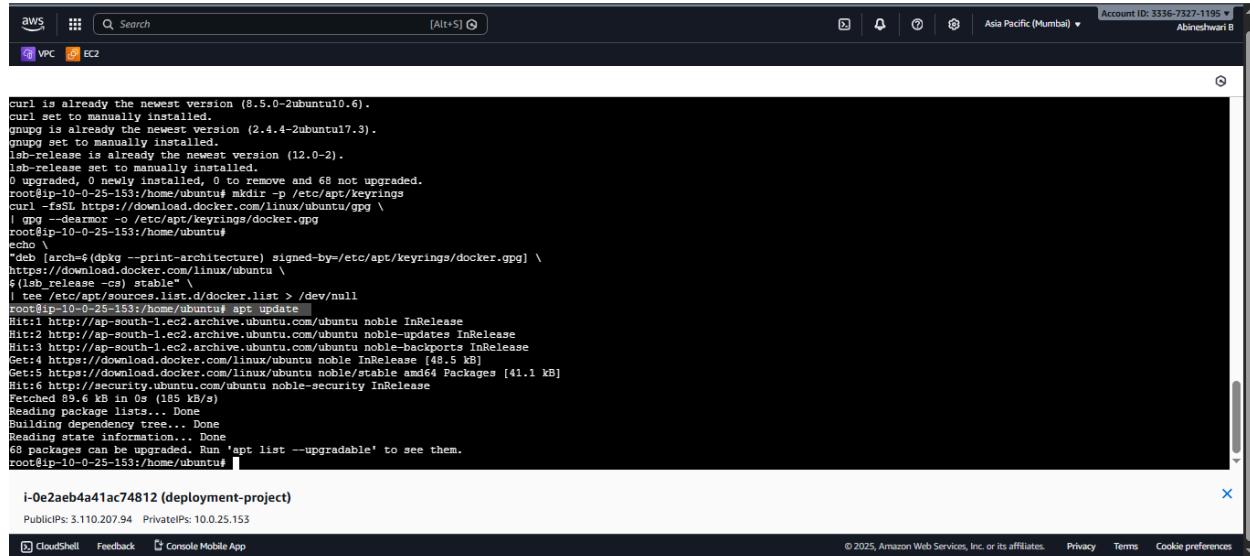
```
echo \
"deb [arch=$(dpkg --print-architecture)
signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" \
| tee /etc/apt/sources.list.d/docker.list > /dev/null
```

The screenshot shows a terminal session in AWS CloudShell. The user has run the command to add the Docker repository to /etc/apt/sources.list.d/docker.list. The output shows standard apt-get dependency resolution and package installation logs.

```
root@ip-10-0-25-153:/home/ubuntu# curl -fsSL https://download.docker.com/linux/ubuntu \
| 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#
```

3.5 Update Package List Again

```
apt update
```



```
curl is already the newest version (8.5.0-2ubuntu10.6).
curl set to manually installed.
gnupg is already the newest version (2.4.4-2ubuntu17.3).
gnupg set to manually installed.
lsb-release is already the newest version (12.0-2).
lsb-release set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
root@ip-10-0-25-153:/home/ubuntu# mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
| gpg --dearmor -o /etc/apt/keyrings/docker.gpg
root@ip-10-0-25-153:/home/ubuntu#
echo \
"deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" \
> /etc/apt/sources.list.d/docker.list > /dev/null
root@ip-10-0-25-153:/home/ubuntu# apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 https://download.docker.com/linux/ubuntu/noble InRelease [48.5 kB]
Get:5 https://download.docker.com/linux/ubuntu/noble/stable amd64 Packages [41.1 kB]
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Fetched 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#
```

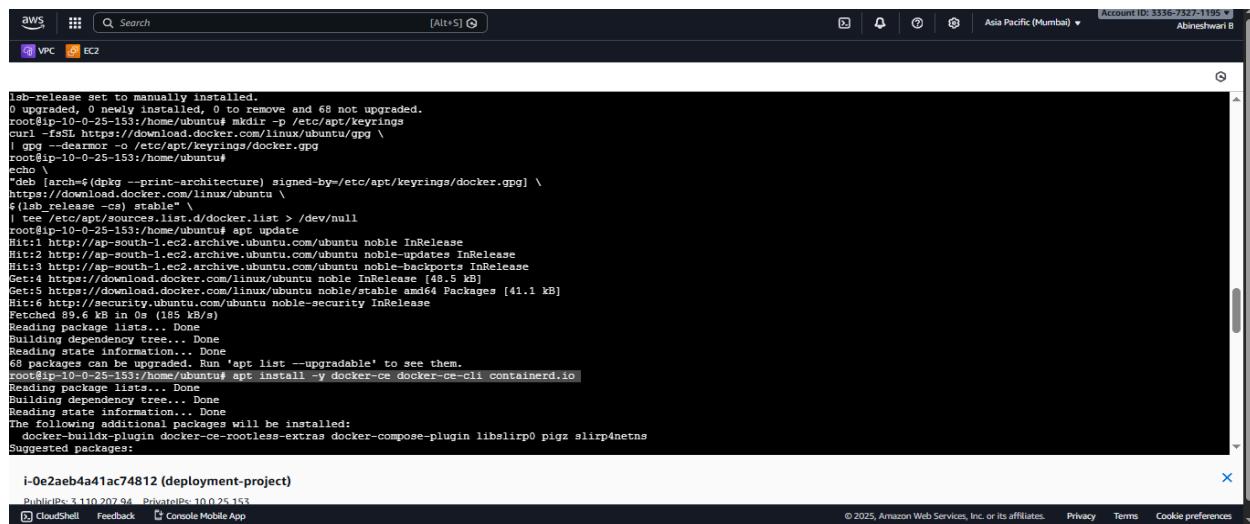
i-0e2aeb4a41ac74812 (deployment-project)
PublicIPs: 3.110.207.94 PrivateIPs: 10.0.25.153

CloudShell Feedback Console Mobile App

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

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 \
$(lsb_release -cs) stable" \
> /etc/apt/sources.list.d/docker.list > /dev/null
root@ip-10-0-25-153:/home/ubuntu# apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 https://download.docker.com/linux/ubuntu/noble InRelease [48.5 kB]
Get:5 https://download.docker.com/linux/ubuntu/noble/stable amd64 Packages [41.1 kB]
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Fetched 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)

PublicIPs: 3.110.207.94 PrivateIPs: 10.0.25.153

CloudShell Feedback Console Mobile App



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


```

3.7 Start and Enable Docker

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

```
Created symlink /etc/systemd/system/multi-user.target.wants/containererd.service → /usr/lib/systemd/system/containererd.service.
Setting up docker-compose-plugin (5.0.0-1~ubuntu.24.04-noble) ...
Setting up docker-ce-cl1 (5:29.1.3-1~ubuntu.24.04-noble) ...
Setting up libslirp0:amd64 (4.7.0~ubuntus3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:29.1.3-1~ubuntu.24.04-noble) ...
Setting up liblpmem0 (1.2.1-1~ubuntus2) ...
Setting up docker-ce (5:29.1.3-1~ubuntu.24.04-noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Processing triggers for man-db (2.12.0-4ubuntus2) ...
Processing triggers for libc-bin (2.39-0ubuntus8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

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

i-0e2aeb4a41ac74812 (deployment-project)
PublicIPs: 3.110.207.94 PrivateIPs: 10.0.25.153
```

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

Check status:

`systemctl status docker`

You should see:

`active (running)`

```
No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (gemu) binaries on this host.
root@ip-10-0-25-153:/home/ubuntu# systemctl start docker
root@ip-10-0-25-153:/home/ubuntu# systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
root@ip-10-0-25-153:/home/ubuntu# 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 Dec 20 05:00:05 UTC 4min 52s ago
     Docs: https://docs.docker.com
     Main PID: 3402 (dockerd)
        Tasks: 9
       Memory: 41.4M (peak: 42.3M)
         CPU: 528ms
      CGroup: /system.slice/docker.service
              └─3402 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containererd.sock

Dec 20 05:00:04 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:04.761182285Z" level=info msg="Restoring containers: start."
Dec 20 05:00:04 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:04.809726955Z" level=info msg="Deleting nftables IPv4 rules" error="exit status 1"
Dec 20 05:00:04 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:04.818586659Z" level=info msg="Deleting nftables IPv6 rules" error="exit status 1"
Dec 20 05:00:05 ip-10-0-25-153 dockerd[3402]: time="2025-12-20T05:00:05.250529885Z" level=info msg="Loading containers: done."
```

i-0e2aeb4a41ac74812 (deployment-project)
PublicIPs: 3.110.207.94 PrivateIPs: 10.0.25.153

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

3.8 Allow Docker Without sudo (VERY IMPORTANT)

```
usermod -aG docker ubuntu
```

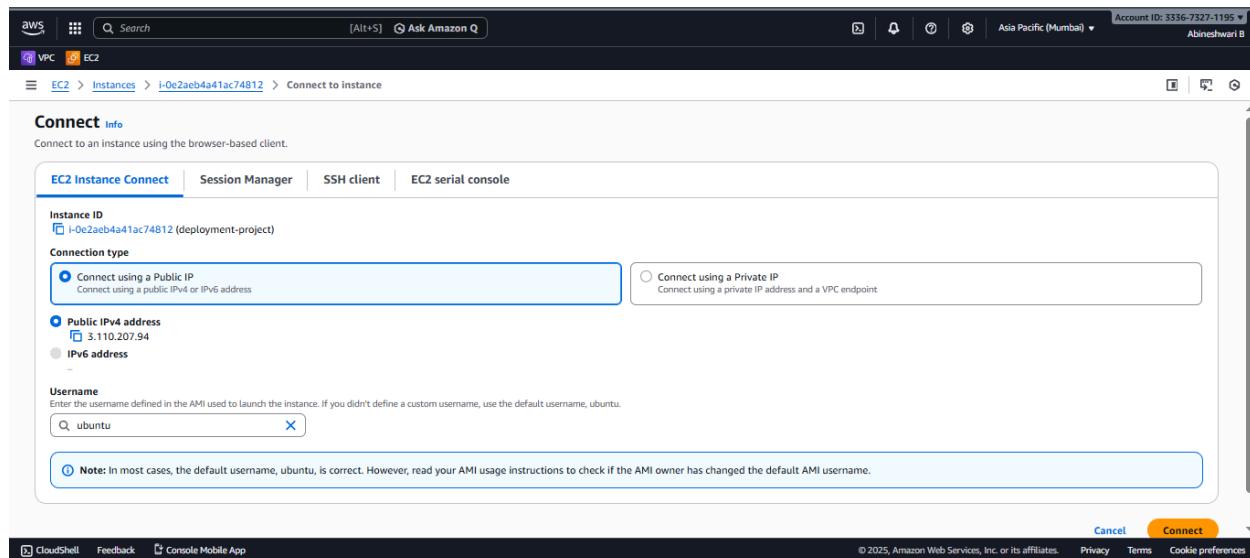
The screenshot shows a terminal session in AWS CloudShell. The user has run the command `usermod -aG docker ubuntu`. Below this, the terminal displays the Docker daemon's log output, which shows the daemon starting up and initializing buildkit. The log ends with the message "Docker has completed initialization".

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

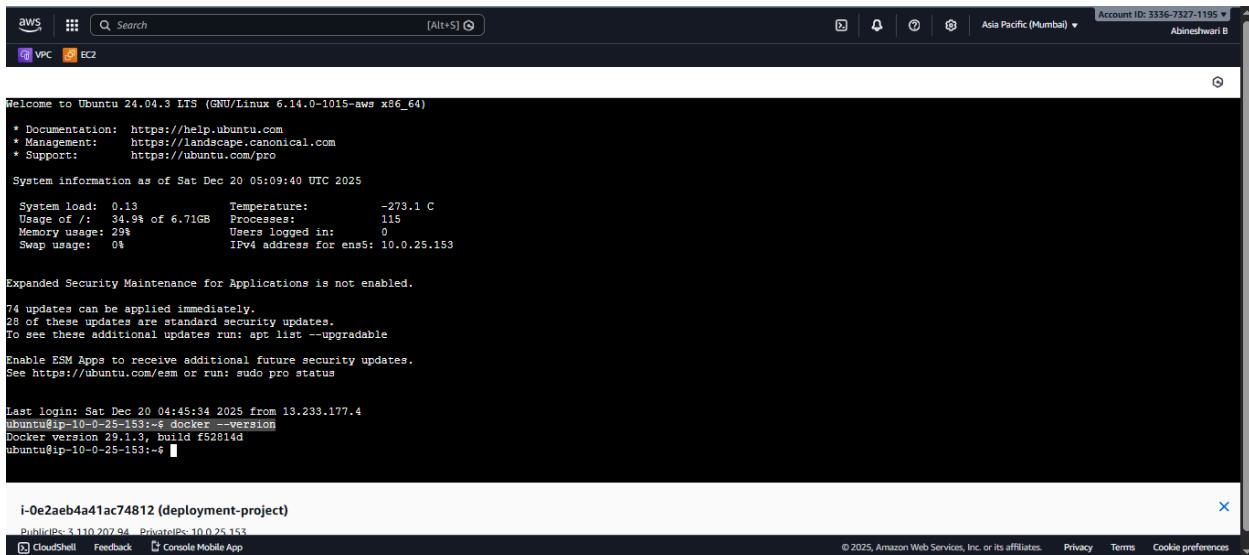
Now disconnect and reconnect the EC2 console session.



3.9 Verify Docker Installation

After reconnecting, run:

```
docker --version
```



The screenshot shows a CloudShell terminal window within the AWS Management Console. The terminal output 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:~$
```

Below the terminal, the CloudShell interface shows the session ID (i-0e2aeb4a41ac74812), public IP (3.110.207.94), private IP (10.0.25.153), and links for CloudShell Feedback and Console Mobile App.

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
```

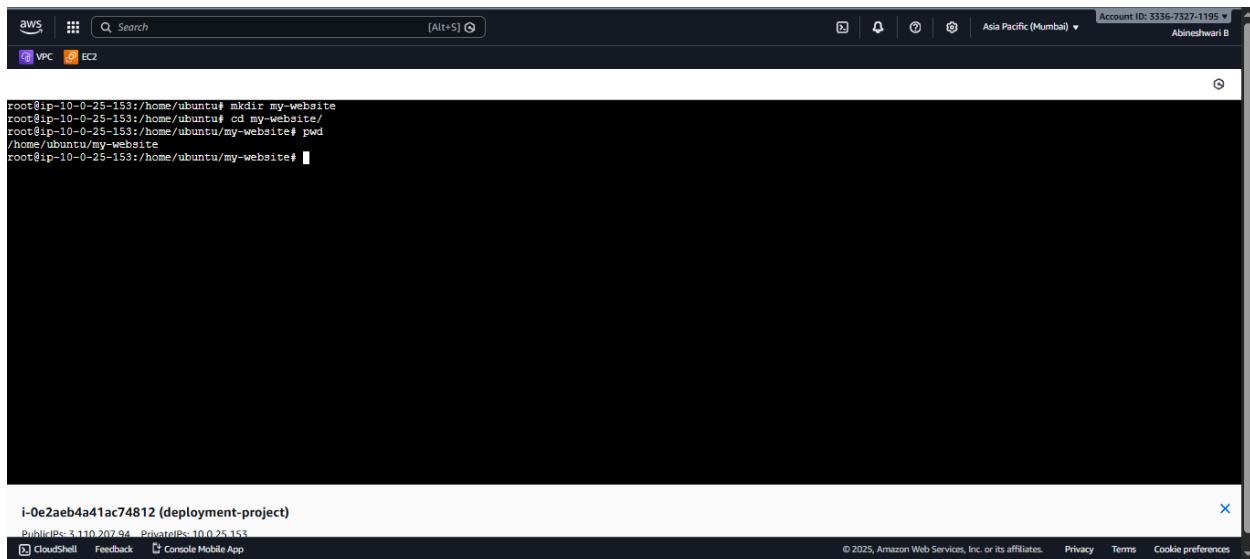
```
cd my-website
```

Verify:

```
pwd
```

Expected:

```
/home/ubuntu/my-website
```



The screenshot shows a terminal session in an AWS CloudShell window. The terminal output is as follows:

```
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#
```

The terminal window has a header bar with the AWS logo, search bar, and account information (Account ID: 3336-7327-1195, Asia Pacific (Mumbai), Abinashwar B). Below the terminal area, there is a footer bar with project details (i-0e2aeb4a41ac74812 (deployment-project)), network info (Public IPs: 3.110.207.94, Private IPs: 10.0.25.153), and links for CloudShell, Feedback, Console Mobile App, Privacy, Terms, and 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 (1:2.43.0-1ubuntu7.3).
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#
```

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#
```

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

STEP 4.3: Clone Your Repo INTO Current Directory

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

Example:

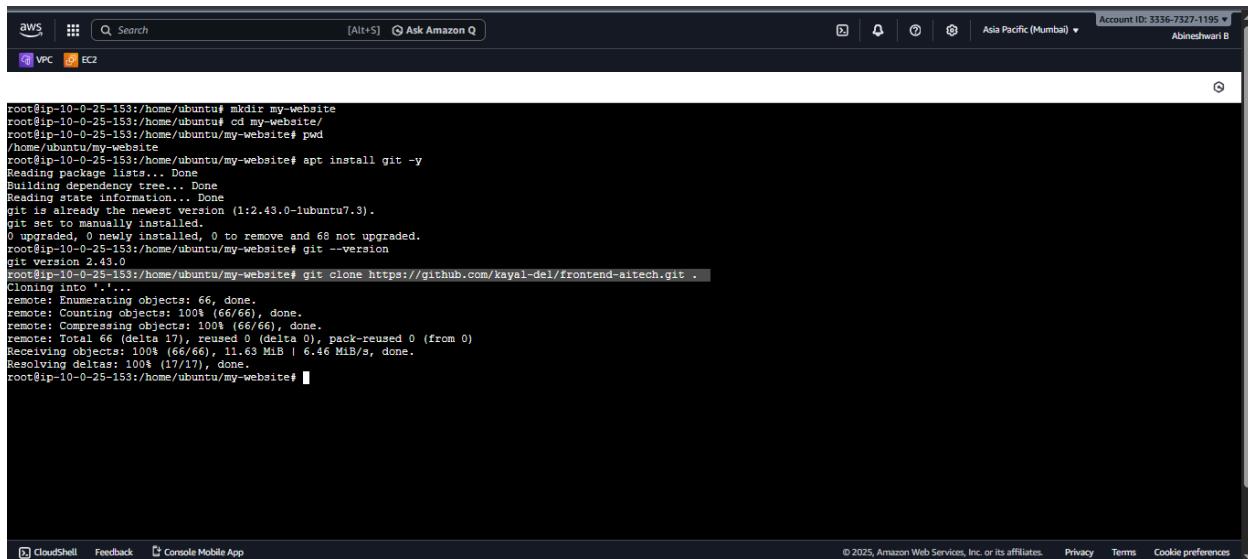
<https://github.com/kayal-del/frontend-aitech>

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

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

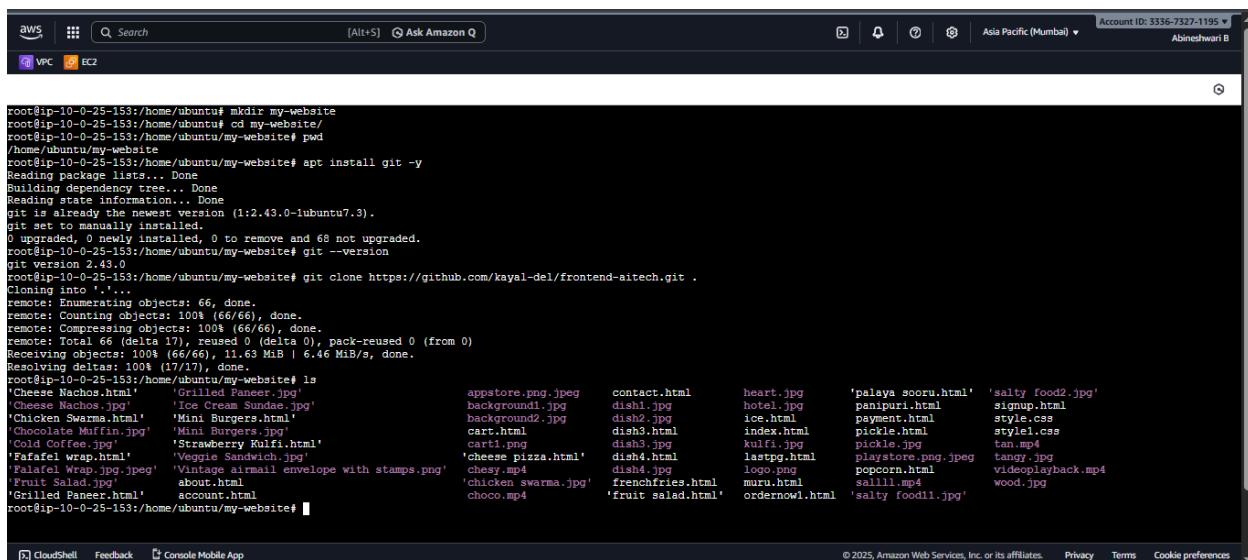
Why the . matters

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



```
aws CloudShell Search [Alt+S] Ask Amazon Q Account ID: 3336-7327-1195 ▾ Asia Pacific (Mumbai) ▾ Abineshwar B
@ VPC EC2

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#
```



```
aws CloudShell Search [Alt+S] Ask Amazon Q Account ID: 3336-7327-1195 ▾ Asia Pacific (Mumbai) ▾ Abineshwar B
@ VPC EC2

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# ls
'Cheese Nachos.html'          'Grilled Paneer.jpeg'      contact.html      heart.jpg      'palaya scoru.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'     dish2.jpg       ice.html      payment.html    style.css
'Chocolate Muffin.jpg'        'Strawberry Kulfi.html' dish3.html     index.html    pickle.html     style1.css
'Cold Coffee.jpg'             'Veggie Sandwich.jpg'   dish4.html     dish1.jpg      logo.jpg       tan.mp4
'Fafafel wrap.html'           'Vintage airmail envelope with stamps.png' dish5.jpg       dish2.jpg      popcorn.html   videoplayback.mp4
'Fried Wrap.jpg.jpeg'         about.html           dish6.jpg       dish3.jpg      dish3.jpg      wood.jpg
'Fruit Salad.jpg'             account.html          dish7.jpg       dish4.jpg      dish5.jpg      ''
'Grilled Paneer.html'         chocochoco.html       dish8.jpg       dish6.jpg      dish7.jpg      ''
root@ip-10-0-25-153:/home/ubuntu/my-website#
```

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:

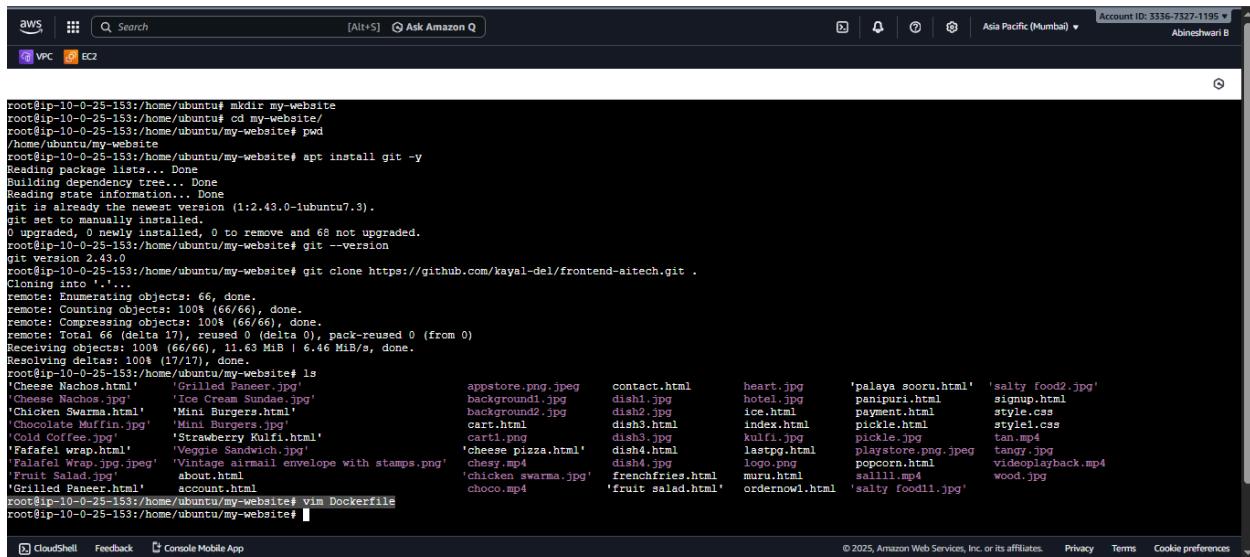
```
vim Dockerfile
```

Paste **exactly this**:

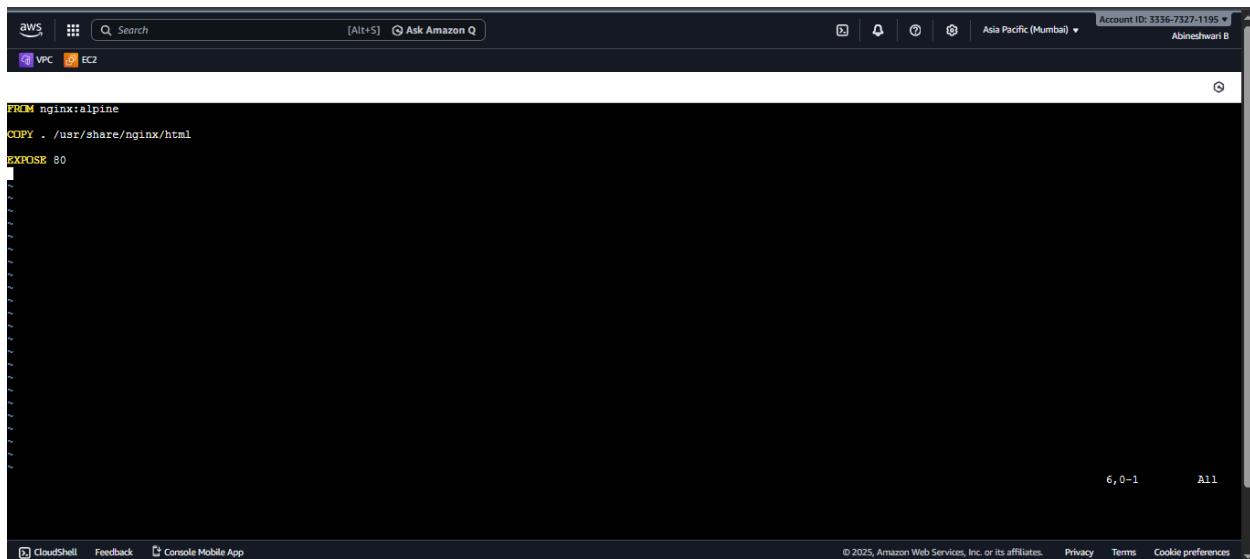
```
FROM nginx:alpine
```

```
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 (1:2.43.0-1ubuntu7.3).
git-set-locale is already installed.
git is fully 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# ls
'Cheese Nachos.html'          'Grilled Paneer.jpg'
'Cheese Nachos.jpg'           'Ice Cream Sundae.jpg'
'Chicken Swarma.html'         'Mini Burgers.html'
'Chocolate Muffin.jpg'        'Mini Burgers.jpg'
'Cold Coffee.jpg'             'Strawberry Kulfi.html'
'Fafafel wrap.html'           'Veggie Sandwich.jpg'
'Fafafel Wrap.jpg.jpeg'       'Vintage airmail envelope with stamps.png'
'Fruit Salad.jpg'             'about.html'
'Grilled Paneer.html'         'account.html'
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

```

STEP 5.2: Build Docker Image

Run:

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

Wait until it finishes.

```
aws CloudShell Feedback Console Mobile App [Alt+S] Ask Amazon Q Account ID: 3336-7327-1195 Asia Pacific (Mumbai) Abinashwani B

[+] Building 6.6s (6/7)
[+] 1/6 : internal load build definition from Dockerfile
[+] 2/6 : transferring dockerfile: 97B
[+] 3/6 : internal load metadata for docker.io/library/nginx:alpine
[+] 4/6 : internal load .dockerignore
[+] 5/6 : transferring context: 25.6MB
[+] 6/6 : internal load build context
[+] 7/7 : transferring context: 25.6MB
[+] 8/8 : FROM docker.io/library/nginx:alpine@sha256:8491795299c8e739bfcc6285d531d9812ce2666fe07bd3dd8dh00020ad132295
[+] 9/9 : resolve docker.io/library/nginx:alpine@sha256:8491795299c8e739bfcc6285d531d9812ce2666fe07bd3dd8dh00020ad132295
[+] 10/10 : sha256_0abf9e572665202679f26f22ef50bd12559e2ea51ec76fb65d43558dfe 17.26MB / 17.26MB
[+] 11/11 : sha256_0abf9e572665202679f26f22ef50bd12559e2ea51ec76fb65d43558dfe 1.40KB / 1.40KB
[+] 12/12 : sha256_0abf9e572665202679f26f22ef50bd12559e2ea51ec76fb65d43558dfe 1.21KB / 1.21KB
[+] 13/13 : sha256_33f95e505e8a4b594770526801b89452b1fc5c043939410ceb5eb066fd458c71b9 403B / 403B
[+] 14/14 : sha256_6d47973db92a1555060973ed849a392b938e260501cfaeeee20984c4rbe80d3 953B / 953B
[+] 15/15 : sha256_d47973db92a1555060973ed849a392b938e260501cfaeeee20984c4rbe80d3 626B / 626B
[+] 16/16 : sha256_085c5e8a4b594770526801b89452b1fc5c043939410ceb5eb066fd458c71b9 5.86MB / 5.86MB
[+] 17/17 : sha256_085c5e8a4b594770526801b89452b1fc5c043939410ceb5eb066fd458c71b9 3.86MB / 3.86MB
[+] 18/18 : extracting sha256_01074535aee00b201d15af21971e50000cf5733495f762a0933da06ff8361c 0.29
[+] 19/19 : extracting sha256_01074535aee00b201d15af21971e50000cf5733495f762a0933da06ff8361c 0.18
[+] 20/20 : extracting sha256_567f824da46fb42874945837485469435c0a81f9a893995b6395d3600643a 0.09
[+] 21/21 : extracting sha256_d7c793db92a1555060973ed849a392b938e260501b89452b1fc5c043939410ceb5eb066fd458c71b9 0.09
[+] 22/22 : extracting sha256_33f95e505e8a4b594770526801b89452b1fc5c043939410ceb5eb066fd458c71b9 0.09
[+] 23/23 : extracting sha256_085c5e8a4b594770526801b89452b1fc5c043939410ceb5eb066fd458c71b9 0.09
[+] 24/24 : extracting sha256_0abf9e572665202679f26f22ef50bd12559e2ea51ac32807721b76fd244ab0 0.09
```

Verify image:

docker images

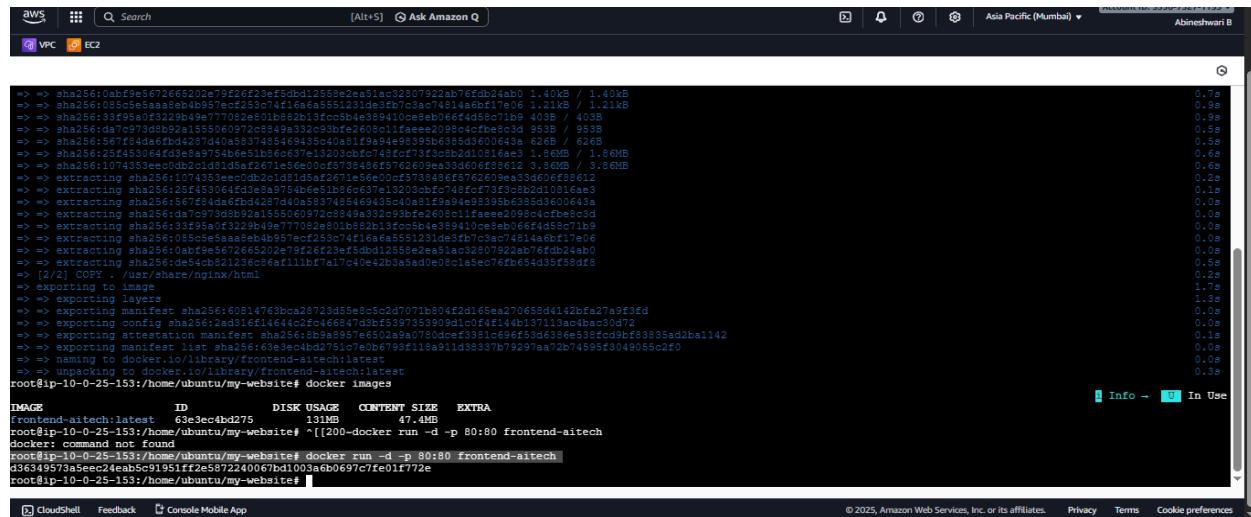
You should see:

frontend-aitech latest

STEP 5.3: Run Container (Local EC2 Test)

Run:

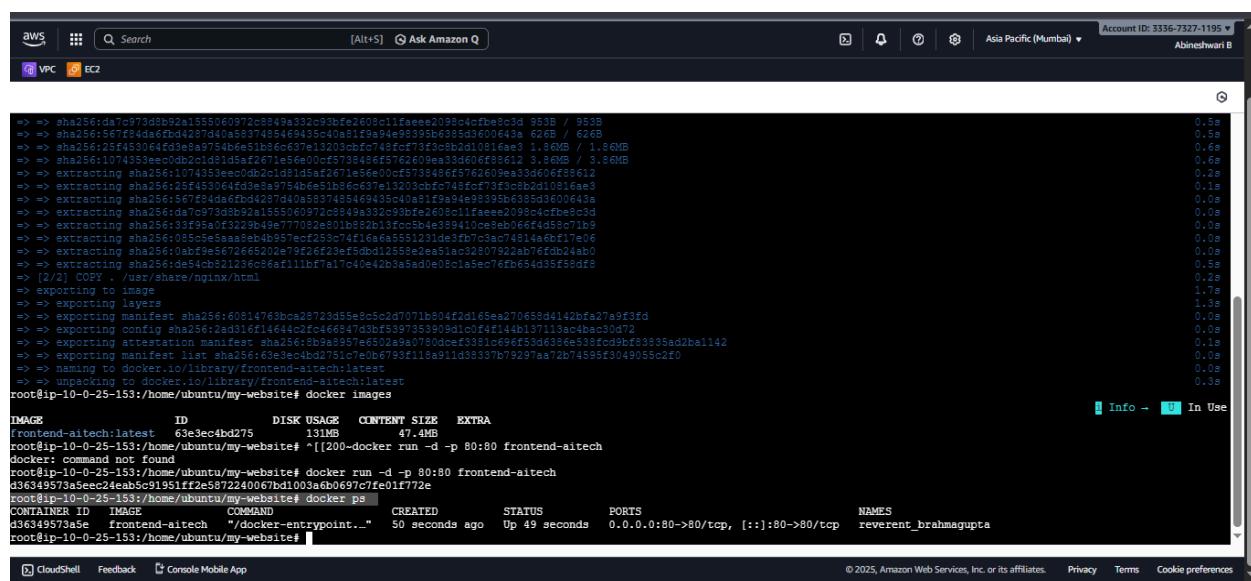
```
docker run -d -p 80:80 frontend-aitech
```



```
>> => sha256:0abf3e5176650567561672d4e5dbd1255e8c5a5380782d0b76fd24b0 1.40KB / 1.40KB
=> => sha256:1955c55a5aa8bb92a57ae4253a74f16a8551533d2527cfae74114aeb1756 1.21KB / 1.21KB
=> => sha256:d33f795a0f33229b49e77702e01b892b13fc5b04e399410ce39e0b06624d5cc71b9 403B / 403B
=> => sha256:d4a7c973d8b92a1555060972c849a332c530fe2605c11faeee2098c4cfbe8c3d 953B
=> => sha256:677f54d4a6fd4287d4a058374a5469435c40a81f9a94e983395b6385d3600643a 626B / 626B
=> => sha256:25f453064d3de8a9754bde51bae537e13203cbfc748fc73f3c0bdc10816aae3 1.86MB
=> => sha256:10743533ee00bd2c1d91da52a7f71e56e000f579348ef76d209ea3d3a606f88612 3.86MB / 3.86MB
=> => extracting sha256:0abf3e5176650567561672d4e5dbd1255e8c5a5380782d0b76fd24b0 1.40KB
=> => extracting sha256:1955c55a5aa8bb92a57ae4253a74f16a8551533d2527cfae74114aeb1756 1.21KB
=> => extracting sha256:d33f795a0f33229b49e77702e01b892b13fc5b04e399410ce39e0b06624d5cc71b9 403B
=> => extracting sha256:d4a7c973d8b92a1555060972c849a332c530fe2605c11faeee2098c4cfbe8c3d 953B
=> => sha256:13f95a0f3229b49e77702e01b892b13fc5b04e399410ce39e0b06624d5cc71b9 403B
=> => extracting sha256:058c5eaaa8eb4b57ecf253a74f16a6551231de3fb7cfae74114aeb17e6
=> => extracting sha256:0abf3e517665020e75f26f23ef50bd1255e8e2a51a32807922ab76fd24ab0
=> => extracting sha256:e04c0b521236e86af11bf7a17c40e42b3a5ad0e08cia5ec76fb654d35f58df8
(2/2) [2/2] COPY . /user/share/nginx/html
=> => exporting to Image
=> => exporting layers
=> => exporting manifest sha256:60814763bca28723d55e8c5c2d7071b804f2d165ea270658d4142bfa27a9f3fd
=> => exporting config sha256:2ad316f1444c2fc4668a7d3bf53973539090d1cf0f4f1446137113ac4bac30d72
=> => exporting attestation manifest sha256:9b9a9597e6502a9a780dcef3381c696f53d6596e5358fcd9bf83835ad2ba1142
=> => exporting manifest list sha56:643e39e44bd2751c760b6793f1l8a911d38337b79297aa72b74595f3049055c2f0
=> => naming to docker.io/library/frontend-aitech:latest
=> => unpacking to docker.io/library/frontend-aitech:latest
root@ip-10-0-25-153:/home/ubuntu/my-website# docker images
[Info - In Use]
IMAGE           ID             SIZE      EXTRA
frontend-aitech:latest 63e3ec4bd275 131MB    47.4MB
root@ip-10-0-25-153:/home/ubuntu/my-website# ^[[200-docker run -d -p 80:80 frontend-aitech
root@ip-10-0-25-153:/home/ubuntu/my-website# docker run -d -p 80:80 frontend-aitech
d36349573a5ec24eb5c91951ff2e5872240067bd1003a6b0697c7fe01f772e
root@ip-10-0-25-153:/home/ubuntu/my-website#
[CloudShell Feedback Console Mobile App]
```

Check container:

```
docker ps
```

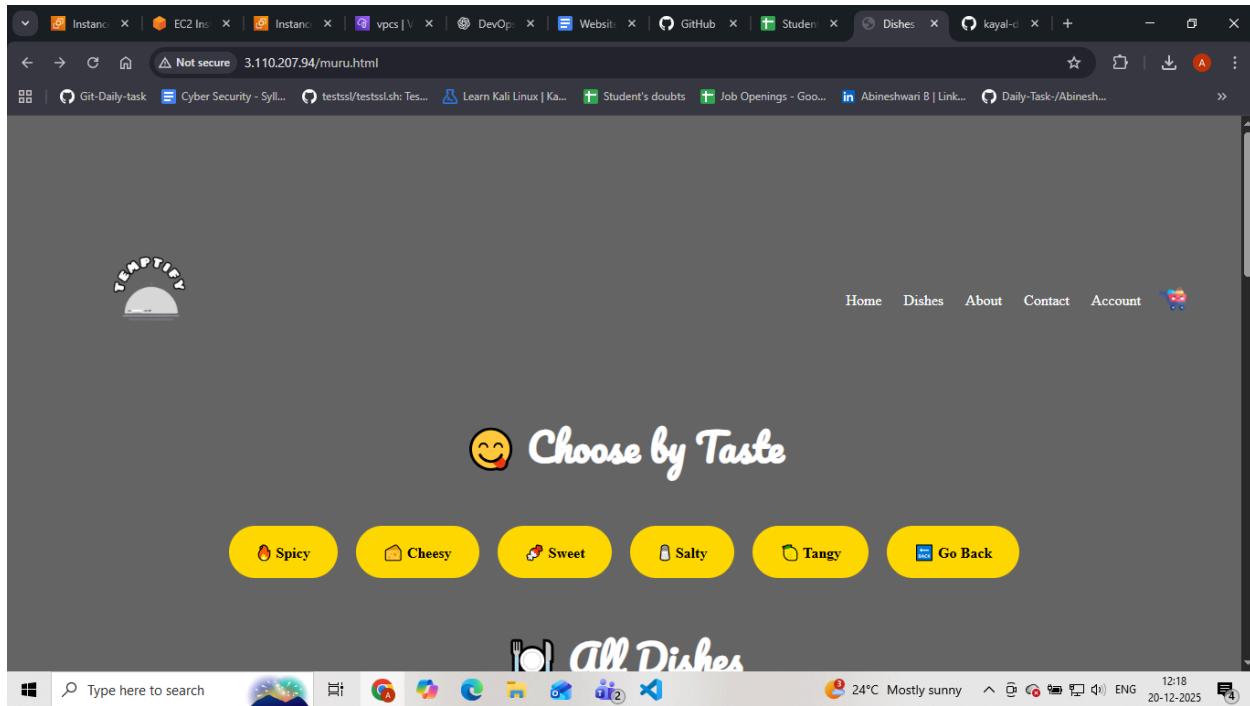


```
>> => sha256:da7c973d8b92a1555060972c849a332c53bfe2605c11faeee2098c4cfbe8c3d 953B / 953B
=> => sha256:677f54d4a6fd4287d4a058374a5469435c40a81f9a94e983395b6385d3600643a 626B / 626B
=> => sha256:10743533ee00bd2c1d91da52a7f71e56e000f579348ef76d209ea3d3a606f88612 3.86MB
=> => extracting sha256:0abf3e5176650567561672d4e5dbd1255e8c5a5380782d0b76fd24b0 1.40KB
=> => extracting sha256:1955c55a5aa8bb92a57ae4253a74f16a8551533d2527cfae74114aeb1756 1.21KB
=> => extracting sha256:d33f795a0f33229b49e77702e01b892b13fc5b04e399410ce39e0b06624d5cc71b9 403B
=> => extracting sha256:d4a7c973d8b92a1555060972c849a332c530fe2605c11faeee2098c4cfbe8c3d 953B
=> => sha256:13f95a0f3229b49e77702e01b892b13fc5b04e399410ce39e0b06624d5cc71b9 403B
=> => extracting sha256:058c5eaaa8eb4b57ecf253a74f16a6551231de3fb7cfae74114aeb17e6
=> => extracting sha256:0abf3e517665020e75f26f23ef50bd1255e8e2a51a32807922ab76fd24ab0
=> => extracting sha256:e04c0b521236e86af11bf7a17c40e42b3a5ad0e08cia5ec76fb654d35f58df8
(2/2) COPY . /user/share/nginx/html
=> => exporting to image
=> => exporting manifest sha256:60814763bca28723d55e8c5c2d7071b804f2d165ea270658d4142bfa27a9f3fd
=> => exporting config sha256:2ad316f1444c2fc4668a7d3bf53973539090d1cf0f4f1446137113ac4bac30d72
=> => exporting attestation manifest sha256:9b9a9597e6502a9a780dcef3381c696f53d6596e5358fcd9bf83835ad2ba1142
=> => exporting manifest list sha56:643e39e44bd2751c760b6793f1l8a911d38337b79297aa72b74595f3049055c2f0
=> => naming to docker.io/library/frontend-aitech:latest
=> => unpacking to docker.io/library/frontend-aitech:latest
root@ip-10-0-25-153:/home/ubuntu/my-website# docker images
[Info - In Use]
IMAGE           ID             SIZE      EXTRA
frontend-aitech:latest 63e3ec4bd275 131MB    47.4MB
root@ip-10-0-25-153:/home/ubuntu/my-website# ^[[200-docker run -d -p 80:80 frontend-aitech
docker: command not found
root@ip-10-0-25-153:/home/ubuntu/my-website# docker run -d -p 80:80 frontend-aitech
d36349573a5ec24eb5c91951ff2e5872240067bd1003a6b0697c7fe01f772e
root@ip-10-0-25-153:/home/ubuntu/my-website# docker ps
CONTAINER ID   IMAGE          COMMAND   CREATED          STATUS          PORTS     NAMES
d36349573a5ec24eb5c91951ff2e5872240067bd1003a6b0697c7fe01f772e   "/docker-entrypoint..-"  50 seconds ago  Up 49 seconds  0.0.0.0:80->80/tcp, ::1:80->80/tcp   reverent_brahmagupta
root@ip-10-0-25-153:/home/ubuntu/my-website#
[CloudShell Feedback Console Mobile App]
```

STEP 5.4: Test Website in Browser

Open browser and visit:

`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>
```

```
aws Search [Alt+S] Ask Amazon Q Account ID: 3336-7327-1195 ▾ Asia Pacific (Mumbai) ▾ Abineshwari B
VPC EC2

IMAGE           ID   DISK USAGE   CONTENT SIZE   EXTRA
Frontend-aitech:latest  b5e16a3bbfcb  131MB    47.4MB
root@ip-10-0-25-153:/home/ubuntu/my-website# docker run -d -p 80:80 frontend-aitech
d4f9e15602ac3113e03ad693f1ba13731a5963e465cf1297bd03fc15e65ce5a
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint condescending_cartwright (b92a341150c2bb70ec88b58f
9ef0723ae109495374404c3fc837076ce69cca24): Bind for 0.0.0.0:80 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  0.0.0.0:80->80/tcp, [:]:80->80/tcp  reverent_brahmagupta
root@ip-10-0-25-153:/home/ubuntu/my-website# docker stop 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
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
f2972552ca097bf7f1f615ae4fc011b7a1aca6fah7386c99b53b2c9b4dab2
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  0.0.0.0:80->80/tcp, [:]:80->80/tcp  frosty_benz
root@ip-10-0-25-153:/home/ubuntu/my-website# adsfrdhtjgv

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

```
aws Search [Alt+S] Ask Amazon Q Account ID: 3336-7327-1195 ▾ Asia Pacific (Mumbai) ▾ Abineshwari B
VPC EC2

IMAGE           ID   DISK USAGE   CONTENT SIZE   EXTRA
Frontend-aitech:latest  b5e16a3bbfcb  131MB    47.4MB
root@ip-10-0-25-153:/home/ubuntu/my-website# docker run -d -p 80:80 frontend-aitech
d4f9e15602ac3113e03ad693f1ba13731a5963e465cf1297bd03fc15e65ce5a
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint condescending_cartwright (b92a341150c2bb70ec88b58f
9ef0723ae109495374404c3fc837076ce69cca24): Bind for 0.0.0.0:80 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  0.0.0.0:80->80/tcp, [:]:80->80/tcp  reverent_brahmagupta
root@ip-10-0-25-153:/home/ubuntu/my-website# docker stop 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
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
f2972552ca097bf7f1f615ae4fc011b7a1aca6fah7386c99b53b2c9b4dab2
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  0.0.0.0:80->80/tcp, [:]:80->80/tcp  frosty_benz
root@ip-10-0-25-153:/home/ubuntu/my-website# adsfrdhtjgv

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

```

IMAGE          ID             DISK USAGE   CONTENT SIZE  EXTRA
frontend-aitech:latest  b5a16a3bbfcb    131MB      47.4MB
root@ip-10-0-25-153:/home/ubuntu/my-website# docker run -d -p 80:80 frontend-aitech
d419e15602a3c113e03df6983f1ba13731a59f3e465cf1297b03fc15e565ce5a
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint condescending_cartwright (b92a341150c2bb70ec88b58f
9ef0723ae109495374404c3fc837076c6e69ca24): Bind for 0.0.0.0:80 failed: port is already allocated
Run 'docker run --help' for more information
root@ip-10-0-25-153:/home/ubuntu/my-website# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
d36349573a5e        frontend-aitech   "/docker-entrypoint..."  35 minutes ago     Up 55 minutes    0.0.0.0:80->80/tcp, [::]:80->80/tcp   reverent_brahmagupta
root@ip-10-0-25-153:/home/ubuntu/my-website# docker stop 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/my-website# cd my-website/
root@ip-10-0-25-153:/home/ubuntu/my-website# docker stop d36349573a5e
d36349573a5e
root@ip-10-0-25-153:/home/ubuntu/my-website# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
root@ip-10-0-25-153:/home/ubuntu/my-website# docker run -d -p 80:80 frontend-aitech
f2972552ca097bfc7f1f1645e54fc001b7a1acafab7386c98b535b2c3b4dab2
root@ip-10-0-25-153:/home/ubuntu/my-website# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
f2972552ca09        frontend-aitech   "/docker-entrypoint..."  5 seconds ago     Up 5 seconds    0.0.0.0:80->80/tcp, [::]:80->80/tcp   frosty_benz
root@ip-10-0-25-153:/home/ubuntu/my-website# adsfdrdfjgvgv

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: 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 AWS Management Console interface for the Amazon Elastic Container Registry (Amazon ECR). The top navigation bar includes the AWS logo, search bar, and account information (Account ID: 5556-7827-1195, Region: Asia Pacific (Mumbai), User: Abinashwari B). The left sidebar is titled "Amazon Elastic Container Service" and contains sections for "Private registry" (Repositories, Features & Settings) and "Public registry" (Repositories, Settings). The main content area is titled "Private repositories" and displays a table header with columns: Repository name, URI, Created at, Tag immutability, and Encryption type. A search bar at the top of the table says "Search by repository substring". Below the table, a message states "No repositories" and "No repositories were found". 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) and links for Privacy, Terms, and Cookie preferences.

The screenshot shows the 'Create a private repository' page in the AWS ECR console. In the 'General settings' section, the repository name is set to '333673271195.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech'. Below this, under 'Image tag settings', there are sections for 'Image tag mutability' (with 'Mutable' selected), 'Mutable tag exclusions' (containing a single filter '.*'), and 'Encryption settings' (warning that settings can't be changed once created). The 'Encryption configuration' section shows 'AES-256' selected. At the bottom right are 'Cancel' and 'Create' buttons.

This screenshot shows the same 'Create a private repository' page, but the 'Image tag settings' section is collapsed. The 'Mutable tag exclusions' section is expanded, showing a single filter '.*'. The 'Encryption settings' section is also expanded, displaying the warning about不可更改的设置。The 'Encryption configuration' section shows 'AES-256' selected. The bottom right features 'Cancel' and 'Create' buttons.

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 AWS ECR console interface. On the left, there's a sidebar with navigation links for VPC, EC2, Amazon ECR, Private registry (Repositories), Public registry (Repositories, Settings), ECR public gallery, Amazon ECS, Amazon EKS, Getting started, and Documentation. The main area is titled 'Private repositories (1)' and shows a table with one row. The row details are: Repository name: frontend-aitech, Repository URI: 333673271195.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech, Created at: 20 December 2025, 12:27:07 (UTC+05:5), Tag immutability: Mutable, Encryption type: AES-256. A green success message at the top says 'Successfully created private repository, frontend-aitech'. A 'Repository URI copied' notification is also present. Action buttons include 'View push commands', 'Delete', 'Actions', and 'Create repository'.

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 terminal window shows the command being run: curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o awscliv2.zip. The output shows the progress of the download and the extraction of the zip file. It includes package management logs for apt install unzip -y, listing packages like unzip, amd64, and libcurl4. The terminal ends with a message about pending kernel upgrade and running kernel version 6.14.0-1015-aws.

```
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 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.
(Running in parallel download with 10360 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 candidates...
Scanning linux images...
Pending kernel upgrade!
Running kernel version:
  6.14.0-1015-aws
```

Install unzip (if not installed)

```
apt install unzip -y
```

The screenshot shows a terminal window within the AWS CloudShell interface. The user has run several commands to download and install the AWS CLI:

```
root@ip-10-0-25-153:/home/ubuntu/my-website# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o awscliv2.zip
root@ip-10-0-25-153:/home/ubuntu/my-website# unzip awscliv2.zip
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 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 processes...
Scanning candidates...
Scanning linux images...
Pending kernel upgrade!
Running kernel version:
  6.14.0-1015-aws
```

At the bottom of the terminal, the command `i-0e2aeb4a41ac74812 (deployment-project)` is shown, along with links for CloudShell, Feedback, and Console Mobile App.

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
```

The screenshot shows the AWS CloudShell interface. At the top, there's a navigation bar with 'aws' and 'VPC' buttons, a search bar, and account information ('Account ID: 3336-7327-1195' and 'Asia Pacific (Mumbai)'). Below the navigation bar is a terminal window displaying the output of the 'aws --version' command. The output shows the AWS CLI version 3.13.11 and Python version 3.11.0. At the bottom of the terminal, there are links for 'CloudShell', 'Feedback', and 'Console Mobile App', along with copyright and legal information.

```

inflating: aws/dist/awscli/customizations/wizard/wizards/iam/new-role.yml
inflating: aws/dist/awscli/customizations/seo/index.html
inflating: aws/dist/awscli/topics/ddb-expressions.rst
inflating: aws/dist/awscli/topics/route53-geoip.rst
inflating: aws/dist/awscli/topics/config-vars.rst
inflating: aws/dist/awscli/topics/s3-config.rst
inflating: aws/dist/awscli/topics/topic-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/entry_points.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.rst
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/entry_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/3.13.11 Linux/6.14.0-1015-aws 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`

Screenshot of the AWS IAM Users page. The left sidebar shows navigation options like Dashboard, Access management, and Access reports. The main area displays a table titled "Users (0) Info" with columns for User name, Path, Group, Last activity, MFA, Password age, Console last sign-in, and Access key ID. A message states "No resources to display".

Screenshot of the "Create user" wizard, Step 1: Specify user details. The left sidebar shows the current step: "Specify user details". The main area has a "User details" section with a "User name" input field containing "abi". Below it is a note about character restrictions and a checkbox for "Provide user access to the AWS Management Console - optional". A callout box provides instructions for generating programmatic access keys. Navigation buttons "Cancel" and "Next" are at the bottom.

Screenshot of the AWS IAM 'Create user' wizard Step 2: Set permissions.

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](#)

Permissions options

- 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

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

Screenshot of the AWS IAM 'Create user' wizard 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

Screenshot of the AWS IAM 'Users' page.

Identity and Access Management (IAM)

User created successfully

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

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	-	-	-	-	-

View user Delete Create user

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

Screenshot of the AWS IAM User Details page for user 'abi'.

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

Permissions policies (3):

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

Screenshot of the 'Create access key' wizard for user 'abi'.

Step 1: Access key best practices & alternatives

The 'Access key best practices & alternatives' section is highlighted.

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**

```

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/_meta/_marker
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.rst
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/entry_points.txt
inflating: aws/dist/wheel-0.45.1.dist-info/MAINTAINER
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/3.13.11 Linux/6.14.0-1015-aws exe/x86_64.ubuntu.24
root@ip-10-0-25-153:/home/ubuntu/my-website# aws configure
root@ip-10-0-25-153:/home/ubuntu/my-website# aws configure
AWS Access Key ID [None]: AKIAU3MD2B6N2RPN5KXG
AWS Secret Access Key [None]: u50y45bmw54Fn/ymPLJBQh1MSN7JpCEiP5Xdggu
Default Region Name [None]: ap-south-1
Default Output Format [None]: json
root@ip-10-0-25-153:/home/ubuntu/my-website#

```

i-0e2aeb4a41ac74812 (deployment-project)
PublicIPs: 3.110.207.94 PrivateIPs: 10.0.25.153

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

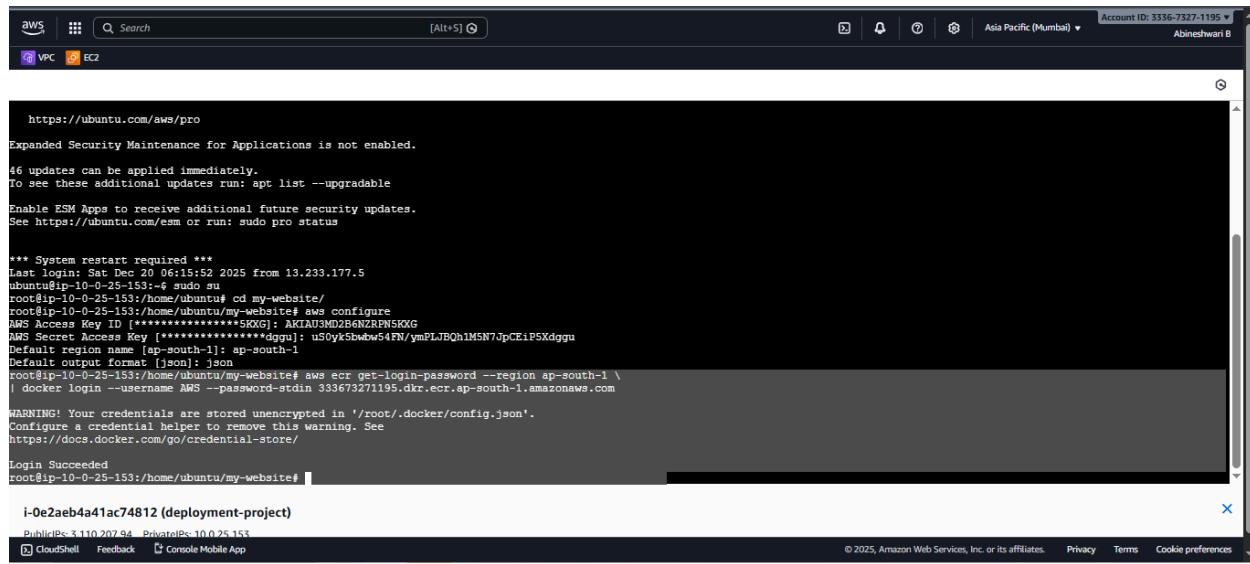
6.4.2 Login Docker to ECR (MOST IMPORTANT COMMAND)

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 with the AWS CLI command being run. The command is:

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

The terminal output shows the password being retrieved and passed to the docker login command, resulting in a "Login Succeeded" message.

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

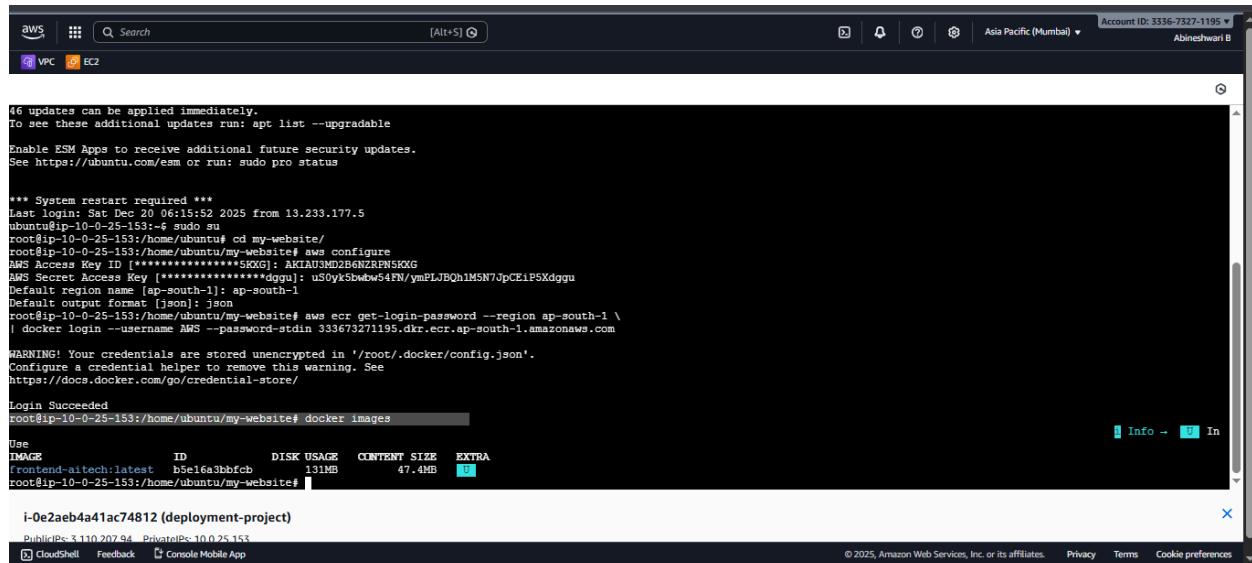
STEP 6.5: Tag Docker Image for ECR

Check your local image name:

```
docker images
```

You already have:

```
frontend-aitech:latest
```



The screenshot shows a terminal window within the AWS CloudShell interface. The terminal displays the output of the command `docker images`. The output shows one image named `frontend-aitech:latest` with details such as ID, disk usage, content size, and extra information.

```
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 [*****5XKG]: AKIAU3M02B6NZN2BN5XKG
AWS Secret Access Key [*****5dggu]: uS0ykBwdw54EN/ymlJJBQh1mSN7JpCEiP5Xdggu
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
[REDACTED]
Use
IMAGE          ID      DISK USAGE   CONTENT SIZE   EXTRA
frontend-aitech:latest  b5e16a3bbfc...  131MB        47.4MB  [REDACTED]
root@ip-10-0-25-153:/home/ubuntu/my-website# i-0e2ae... (deployment-project)
Public IPs: 3.110.207.94 Private IPs: 10.0.25.153
CloudShell Feedback Console Mobile App
Info - In
© 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:~/my-website# aws configure
AWS Access Key ID [*****5KKG]: AKIAU3M026N2ZRN5KKG
AWS Secret Access Key [*****5dgg]: usOyksbw6w64FN/ymlJ5Qh1M5N7JpCEiP5Xdgg
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
[Info] In
Use
IMAGE           ID          DISK USAGE   CONTENT SIZE   EXTRA
frontend-aitech:latest b5e16a3bbfcfcb 131MB      47.4MB  U
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# ^[[200-docker push 333673271195.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech:latest
root@ip-10-0-25-153:/home/ubuntu/my-website# ^[[200-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]
da7c973db9b2: Pushed
33f95af0f3229: Pushed
085cbe10404d: Pushed
daec5cb21236: Pushed
1074353eecc0d: Pushed
25f453064fd3: Pushed
0abfe567266: Pushed
5afbeec10ed: Pushed
79902eba3043: Pushed
567f184daefbd: Pushed
latest: digest: sha256:b5e16a3bbfcbae82fce5c2d4f3256471ba93b8b43c59e9206a58aee5f8d04775 size: 856
root@ip-10-0-25-153:/home/ubuntu/my-website# ^[[200-docker push i-De2aeb4a41ac74812 (deployment-project)
PublicIPs: 3.110.207.94 PrivateIPs: 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.

```

aws | Search [Alt+5] ⓘ Account ID: 333673271195
VPC EC2
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 b5e16a3bbfcfcb 131MB      47.4MB  U
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# ^[[200-docker push 333673271195.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech:latest
root@ip-10-0-25-153:/home/ubuntu/my-website# ^[[200-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]
da7c973db9b2: Pushed
33f95af0f3229: Pushed
085cbe10404d: Pushed
daec5cb21236: Pushed
1074353eecc0d: Pushed
25f453064fd3: Pushed
0abfe567266: Pushed
5afbeec10ed: Pushed
79902eba3043: Pushed
567f184daefbd: Pushed
latest: digest: sha256:b5e16a3bbfcbae82fce5c2d4f3256471ba93b8b43c59e9206a58aee5f8d04775 size: 856
root@ip-10-0-25-153:/home/ubuntu/my-website# ^[[200-docker push i-De2aeb4a41ac74812 (deployment-project)
PublicIPs: 3.110.207.94 PrivateIPs: 10.0.25.153
CloudShell Feedback Console Mobile App

```

STEP 6.7: Verify in AWS Console

Go to:

`ECR → frontend-aitech → Images`

You should see:

latest

The screenshot shows the AWS ECR console interface. On the left, there's a navigation sidebar for 'Amazon Elastic Container Service' with sections for 'Private registry' (Repositories, Images, Lifecycle policy, Permissions, Repository tags, Features & Settings) and 'Public registry' (Repositories, Settings). Below that are links for 'ECR public gallery', 'Amazon ECS', and 'Amazon EKS'. At the bottom of the sidebar are 'Getting started' and 'Documentation' links, along with 'CloudShell', 'Feedback', and 'Console Mobile App' buttons.

The main area displays a repository named 'frontend-aitech'. It has three images listed:

Image tags	Type	Created at	Image size	Image digest	Last pulled at
latest	Image Index	20 December 2025, 13:07:42 (UTC+05:5)	47.43	sha256:b5e16a3abbfc...	-
-	Image	20 December 2025, 13:07:42 (UTC+05:5)	0.00	sha256:89fb9b3a809...	-
-	Image	20 December 2025, 13:07:42 (UTC+05:5)	47.43	sha256:a3c87a6d4a4c...	-

At the top of the main area, there are tabs for 'Summary', 'Images' (which is selected), and 'Repository tags'. Below the tabs are buttons for 'Delete', 'Copy URI', 'Details', 'Scan', and 'View push commands'. A search bar labeled 'Filter active images' is also present.

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**

Clusters (0) Info

No clusters

Create cluster

Last updated 20 December 2025, 13:17 (UTC+5:30)

Cluster Services Tasks Container instances CloudWatch monitoring Capacity provider strategy

Express Mode Clusters Namespaces Task definitions Account settings

Amazon ECR Repositories AWS Batch Documentation Discover products Subscriptions

Tell us what you think

CloudShell Feedback Console Mobile App

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

STEP 7.2: Cluster Configuration

Fill the details as follows:

Setting	Value
Cluster name	<code>frontend-aitech-cluster</code>
Infrastructure	
Monitoring	Default

Click **Create**

Wait until cluster status = **Active**

Create cluster Info

An Amazon ECS cluster groups together tasks and services, and allows for shared capacity and common configurations. All of your tasks, services and capacity must belong to a cluster.

Cluster configuration

Cluster name

Cluster name must be 1 to 255 characters. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Service Connect defaults – optional

Infrastructure – advanced Info

Configure the manner of obtaining compute resources that will be used to host your application.

Select a method of obtaining compute capacity

Your cluster is automatically configured for AWS Fargate (serverless), but you may choose to add Amazon EC2 instances (servers).

Fargate only
Serverless – you don't think about creating or managing servers. Great for most common workloads.

Fargate and managed instances
Managed instances - Amazon ECS will manage patching and scaling on your behalf while giving you configurability about the types of instances. Great for more advanced workloads.

Fargate and Self-managed instances
Self-managed instances - you must ensure the instances are patched and scaled properly, and you have full control over the instances.

Monitoring – optional

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

Clusters

Cluster frontend-aitech-cluster has been created successfully.

Clusters (1) Info

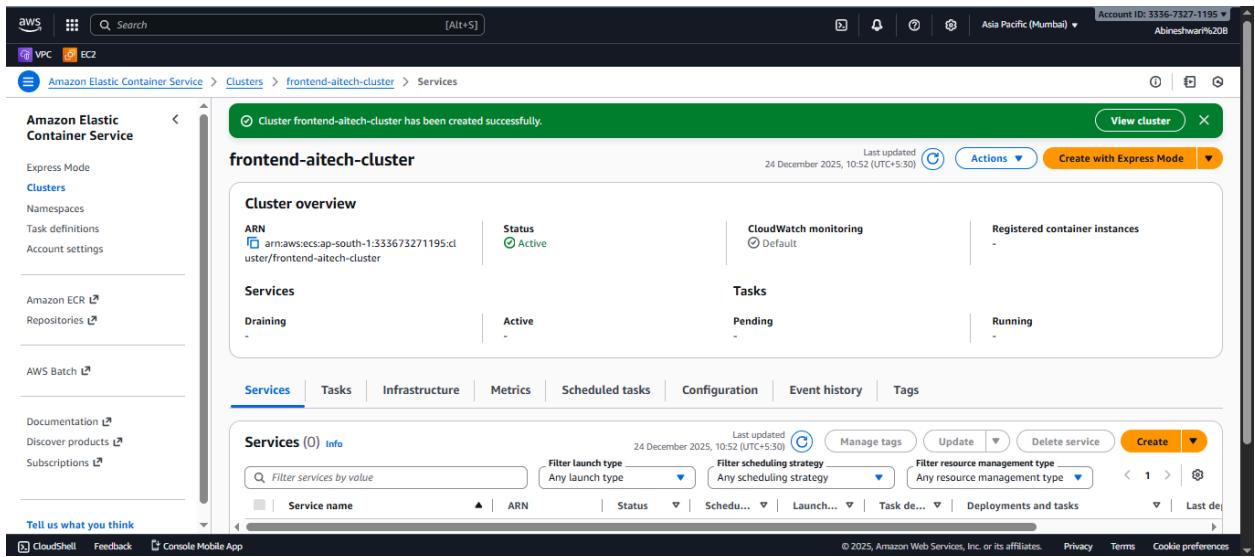
Last updated: 24 December 2025, 10:51 (UTC+5:30)

Create cluster

Cluster	Services	Tasks	Container instances	CloudWatch monitoring	Capacity provider strategy
frontend-aitech-cluster	0	No tasks running	0 EC2	Default	No default found

View cluster

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



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:

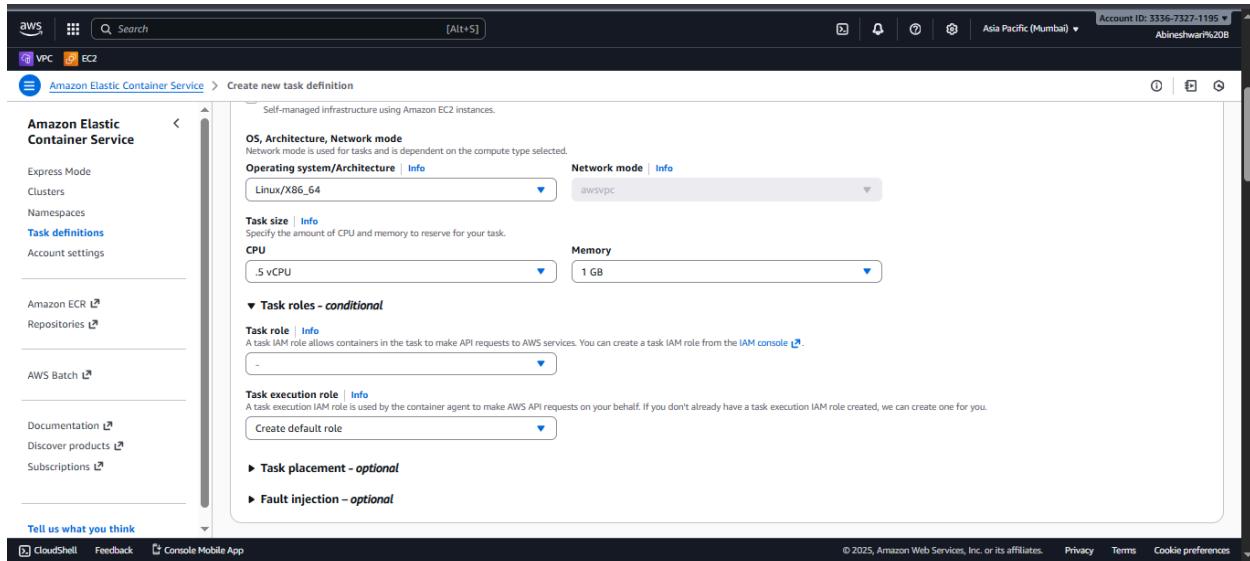
- Fargate

The screenshot shows the AWS Elastic Container Service (ECS) Task definitions page. The left sidebar includes links for Express Mode, Clusters, Namespaces, Task definitions (which is selected and highlighted in blue), and Account settings. Other sections like Amazon ECR, AWS Batch, Documentation, and Subscriptions are also listed. The main content area is titled "Task definitions (0)" and displays a message stating "No task definitions" and "No task definitions to display." A prominent orange button at the bottom right says "Create new task definition". The top navigation bar shows the account ID (3336-7327-1195), region (Asia Pacific (Mumbai)), and user (Abineshwar%20B). The bottom of the page includes standard AWS footer links for CloudShell, Feedback, and Console Mobile App, along with copyright information for 2025.

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" configuration page. The left sidebar is identical to the previous screenshot, featuring links for Express Mode, Clusters, Namespaces, Task definitions (selected), and Account settings. The main form is titled "Create new task definition" and contains several sections: "Task definition configuration" (with a "Task definition family" field containing "frontend-aitech-task"), "Infrastructure requirements" (with "Launch type" set to "AWS Fargate" and other options like "Managed instances" and "Amazon EC2 instances" available), and "OS, Architecture, Network mode" (with "Network mode" currently selected). The bottom of the page includes the standard AWS footer links and copyright information for 2025.



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

Screenshot of the Amazon Elastic Container Service (Amazon ECS) console showing the "Create new task definition" page.

The left sidebar shows navigation links for Express Mode, Clusters, Namespaces, Task definitions (selected), Account settings, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions.

The main content area displays the "Container - 1" configuration:

- Container details**: Name is set to "frontend-aitech". The "Essential container" dropdown is set to "Yes".
- Image URI**: The URL is "333673271195.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech:latest". A "Browse ECR images" button is available.
- Private registry**: A note says "Store credentials in Secrets Manager, and then use the credentials to reference images in private registries." A radio button for "Private registry authentication" is selected.
- Port mappings**: Container port 80 is mapped to host port TCP 80, with the app protocol set to HTTP. A "Remove" button is present.
- Read-only root file system**: A note states "When this parameter is turned on, the container is given read-only access to its root file system." A checkbox for "Read only" is checked.

At the bottom, there are CloudShell, Feedback, and Console Mobile App links, along with copyright information: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

Screenshot of the "Select Amazon ECR image" modal window.

The modal has the following fields:

- Private repository**: The search bar contains the URL "333673271195.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech".
- Images (1/3)**: A table lists three images:

Image tag	Image digest	Pushed at
latest	sha256:b5e16a3bbfcbae82fce5c2d...	20 December 2025, 13:07 (UTC+5:30)
-	sha256:89fb9b3a8090709768309e...	20 December 2025, 13:07 (UTC+5:30)
-	sha256:a3c87a6d4a4c5cf9e9337cd...	20 December 2025, 13:07 (UTC+5:30)
- Select image by**: A radio button is selected for "Image tag" (Use a human-readable tag to reference this image). The input field shows "latest".
- Image tag**: An input field containing "latest".

At the bottom right of the modal are "Cancel" and "Select image" buttons.

Searched for "Create new task definition".

Resource allocation limits - conditional

Container-level CPU, GPU and memory limits are different from task-level values. They define how many resources are allocated for the container. If the container attempts to exceed the memory specified by the hard limit, the container is terminated.

CPU	GPU	Memory hard limit	Memory soft limit
1 in vCPU	1	3 in GB	1

Environment variables - optional

Environment variables

Add individually: Add environment variables using plain text values or secrets from AWS Secrets Manager or Parameter Store.

Add environment variable

Add from file

Add environment variables in bulk by providing an environment file hosted on Amazon S3.

Add environment file

You can add 10 more environment files.

Logging - optional

CPU and memory allocation for a sidecar

There are logging options that will automatically add a sidecar to your task definition if it does not already exist. AWS provides CPU and memory adjustment recommendations based on the selected options.

We recommend that you use log collection for tasks running on AWS Fargate and Managed Instances. Find out more about [log collection](#).

Log collection

Configure your task to send container logs to a logging destination using a default configuration. See pricing information on [Amazon CloudWatch](#).

Use log collection

Destination

Amazon CloudWatch

Key	Value type	Value
awslogs-group	Value	/ecs/frontend-aitech-task
awslogs-region	Value	ap-south-1
awslogs-stream-prefix	Value	ecs
awslogs-create-group	Value	true

Add log configuration option

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

Searched for "Create new task definition".

Logging - optional

CPU and memory allocation for a sidecar

There are logging options that will automatically add a sidecar to your task definition if it does not already exist. AWS provides CPU and memory adjustment recommendations based on the selected options.

We recommend that you use log collection for tasks running on AWS Fargate and Managed Instances. Find out more about [log collection](#).

Log collection

Configure your task to send container logs to a logging destination using a default configuration. See pricing information on [Amazon CloudWatch](#).

Use log collection

Destination

Amazon CloudWatch

Key	Value type	Value
awslogs-group	Value	/ecs/frontend-aitech-task
awslogs-region	Value	ap-south-1
awslogs-stream-prefix	Value	ecs
awslogs-create-group	Value	true

Add log configuration option

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

Amazon Elastic Container Service > Create new task definition

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
21
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

Volumes from Info
Mount data volumes from another container.

Add volume from

Monitoring - optional

Configure your application trace and metric collection settings using the AWS Distro for OpenTelemetry integration.

Tags - optional Info

Tell us what you think

CloudShell Feedback Console Mobile App

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

Logging

- Enable CloudWatch logs
- Log group: auto-create

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

Amazon Elastic Container Service > Create new task definition

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
21
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

Volumes from Info
Mount data volumes from another container.

Add volume from

Monitoring - optional

Configure your application trace and metric collection settings using the AWS Distro for OpenTelemetry integration.

Tags - optional Info

Tell us what you think

CloudShell Feedback Console Mobile App

Cancel Create

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

The screenshot shows the AWS ECS Task Definitions page. A green success message at the top states: "Task definition successfully created" and "frontend-aitech-task:1 has been successfully created. You can use this task definition to deploy a service or run a task." Below this, the task definition details are shown:

- ARN:** arn:aws:ecs:ap-south-1:333673271195:task-definition/frontend-aitech-task:1
- Status:** ACTIVE
- Time created:** 24 December 2025, 11:07 (UTC+5:30)
- Task role:** - (ecsTaskExecutionRole)
- Fault injection:** Turned off
- Operating system/Architecture:** Linux/X86_64
- App environment:** Fargate
- Network mode:** awsvpc

The "Containers" tab is selected. Under "Task size", it shows Task CPU (512 units (0.5 vCPU)) and Task memory (1,024 MiB (1 GiB)).

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**

The screenshot shows the AWS EC2 Target groups page. The left sidebar includes sections for AMI Catalog, Elastic Block Store, Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), Auto Scaling (Auto Scaling Groups), and Settings. The main content area is titled "Target groups" and displays a table with columns: Name, ARN, Port, Protocol, Target type, Load balancer, and VPC ID. A message states "No target groups" and "You don't have any target groups in ap-south-1". A "Create target group" button is located at the bottom of the table. Below the table, a section says "0 target groups selected" and "Select a target group above."

The screenshot shows the "Create target group" wizard, Step 1: Create target group. It has three tabs: Step 1 (selected), Step 2 - recommended, and Step 3: Review and create. The main content area is titled "Create target group" and contains a sub-section "Settings - immutable". It explains that a target group can consist of one or more targets and that the load balancer routes requests to the targets in the target group and performs health checks on the targets. It also notes that these settings can't be modified after target group creation. The "Target type" section allows selecting between Instances, IP addresses, Lambda function, and Application Load Balance. The "Instances" option is selected, showing it supports load balancing to instances in a VPC, integrates with Auto Scaling Groups or ECS services for automatic management, and is suitable for ALB, NLB, and GWLB. The "IP addresses" option is also shown. The "Target group name" field is set to "frontend-altech-tg".

AWS Search [Alt+S] Asia Pacific (Mumbai) Abineshwari%208

VPC EC2

EC2 Target groups Create target group

Name must be unique per Region per AWS account.
frontend-altech-tg
Accepts: a-z, A-Z, 0-9, and hyphen (-). Can't begin or end with hyphen. 1-32 total characters; Count: 18/32

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-0bb99163560bb4e70d (deployment-project-VPC)
10.0.0/16

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.

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

AWS Search [Alt+S] Account ID: 3336-7327-1195 Asia Pacific (Mumbai) Abineshwari%208

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

The screenshot shows the AWS CloudShell interface with a terminal window open. The command entered is:

```
aws elbv2 create-target-group --name my-target-group --port 80 --protocol HTTP --health-check-type EC2 --idle-timeout 60 --timeout 60 --healthy-threshold 2 --unhealthy-threshold 3
```

The output of the command is displayed in the terminal window.

The screenshot shows the AWS CloudShell interface with a terminal window open. The command entered is:

```
aws elbv2 register-targets --target-group-name my-target-group --targets "Ip=192.168.0.11,Port=80,Protocol=HTTP"
```

The output of the command is displayed in the terminal window.

Screenshot of the AWS EC2 Target Groups 'Create target group' wizard Step 3: Review targets.

The page shows a summary of the target group configuration:

- Targets (0)**: A table with columns: Remove IPv4 address, Health status, IP address, Port, Zone. A note says "No IP addresses included yet. Specify IP addresses above and add to list."
- 0 pending**
- Cancel**, **Previous**, **Next** buttons at the bottom.

At the bottom, there are links to CloudShell, Feedback, and Console Mobile App, along with copyright and legal information.

Screenshot of the AWS EC2 Target Groups 'Create target group' wizard Step 1: Target group details.

The page shows the target group details:

- Step 1**: Create target group (selected)
- Step 2 - recommended**
- Step 3**
- Review and create** (highlighted)

Review and create: Review your target group configuration before creating.

Step 1: Target group details

Name	Target type	Protocol : Port	Protocol version
frontend-aitech-tg	IP	HTTP: 80	HTTP1

Step 2: Register targets

At the bottom, there are links to CloudShell, Feedback, and Console Mobile App, along with copyright and legal information.

Step 2: Register targets

IP address	Port	Zone
No targets added		

[Cancel](#) [Previous](#) [Create target group](#)

frontend-aitech-tg

Details

arn:aws:elasticloadbalancing:ap-south-1:333673271195:targetgroup/frontend-aitech-tg/fd8437c0914cba41	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-0bba916360bb4e70d
Target type IP	IP address type IPv4	Load balancer None associated	
0 Total targets	0 Healthy	0 Unhealthy	0 Unused
	0 Anomalous		0 Initial
			0 Draining

[Actions](#)

Targets [Monitoring](#) [Health checks](#) [Attributes](#) [Tags](#)

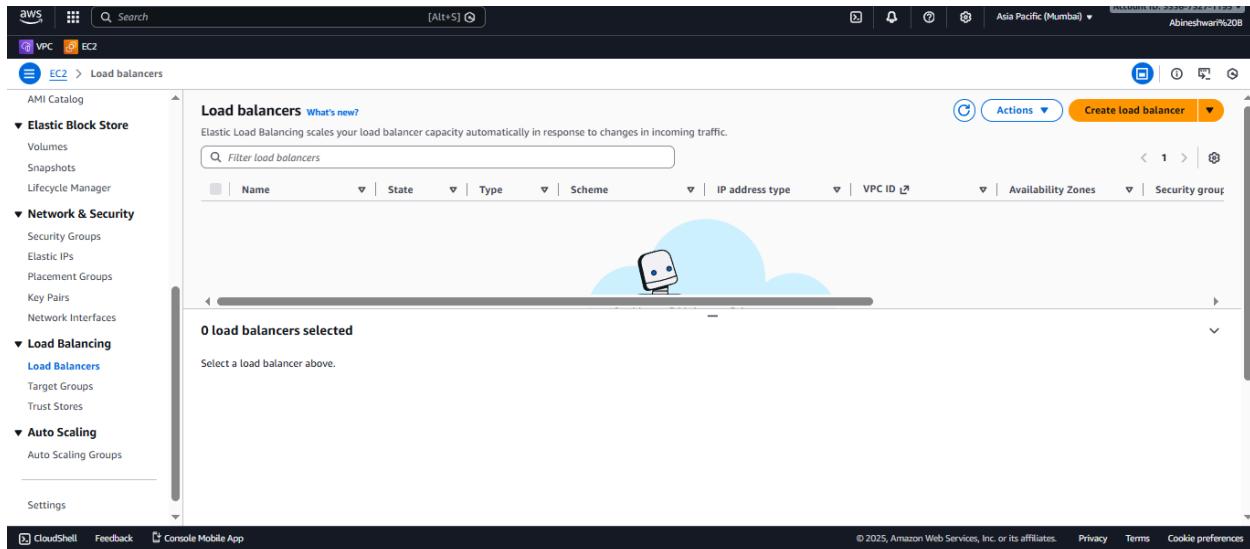
Registered targets (0) [Info](#) [Anomaly mitigation: Not applicable](#) [Deregister](#) [Register targets](#)

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

EC2 > Load balancers > Compare and select load balancer type

ALB: Handles HTTP and HTTPS traffic. It routes requests to targets (Amazon EC2 instances, Lambda functions, or containers) based on listener rules.

NLB: Handles TCP, UDP, and TLS traffic. It routes requests to targets based on connection-level rules.

GWLB: Handles traffic from third-party virtual appliances supporting GENEVE. It routes requests to targets based on security, compliance, and policy controls.

Classic Load Balancer - previous generation: A separate section for the previous generation of load balancers.

[Create ALB](#) | [Create NLB](#) | [Create GWLB](#)

EC2 > Load balancers > Create Application Load Balancer

Create Application Load Balancer

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

How Application Load Balancers work

Basic configuration

Load balancer name: Name must be unique within your AWS account and can't be changed after the load balancer is created.
frontend-aitech-alb
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme: Internet-facing (selected) or Internal.
Scheme can't be changed after the load balancer is created.

Load balancer IP address type: IPv4 (selected).
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.

[Create](#)

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	Target group created in STEP 9

👉 Click **Create load balancer**

Wait until **State = Active**

The screenshot shows the AWS CloudFormation Create Stack Wizard. The 'Resources' tab is selected. A single resource, 'CloudWatch Metrics', is listed. Its properties are: Metric Name: CloudWatchMetrics, Metric Type: CloudWatchMetrics, and Resource Type: AWS::Logs::LogGroup. At the bottom, there is a 'Next Step' button.

AWS Search [Alt+S] Account ID: 3336-7327-1195 Asia Pacific (Mumbai) Abineshwari%208

VPC EC2

EC2 > Load balancers > Create Application Load Balancer

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](#)

Basic configuration Edit

Name: frontend-ai-tech-alb
Scheme: Internet-facing
IP address type: IPv4

Network mapping Edit

VPC: [vpc-0bb916360bb4e70d](#)
Public IPv4 IPAM pool: -
Availability Zones and subnets:

- ap-south-1a
[subnet-0624ce757655ff9](#)
deployment-project-VPC-subnet-public1-ap-south-1a
- ap-south-1b
[subnet-03f425e660184ff8b](#)
deployment-project-VPC-subnet-public2-ap-south-1b

Security groups Edit

deployment-project-sg
[sg-0bd64cf7bd350a40d](#)
default
[sg-0dc1c4845b66358c6](#)

Listeners and routing Edit

HTTP-80 | Forward to 1 target group

Service integrations Edit

Amazon CloudFront + AWS Web Application Firewall (WAF): -
AWS WAF: -
AWS Global Accelerator: -

Tags Edit

-

Attributes

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

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

AWS Search [Alt+S] Account ID: 3336-7327-1195 Asia Pacific (Mumbai) Abineshwari%208

VPC EC2

EC2 > Load balancers > Create Application Load Balancer

1a
ap-south-1b
[subnet-03f425e660184ff8b](#)
deployment-project-VPC-subnet-public2-ap-south-1b

Service integrations Edit

Amazon CloudFront + AWS Web Application Firewall (WAF): -
AWS WAF: -
AWS Global Accelerator: -

Tags Edit

-

Attributes

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 Cancel

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

The screenshot shows the AWS CloudWatch Metrics interface. A log stream named 'lambda-1' is selected. The log entry contains the following message:

```
2023-07-10T10:00:00Z {"@version": "1", "@timestamp": "2023-07-10T10:00:00Z", "log_id": "12345678901234567890123456789012", "source": "aws.lambda", "function_name": "lambda-1", "log_group": "/aws/lambda/lambda-1", "log_stream": "lambda-1", "message": "Hello, World!"}
```

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	<code>frontend-aitech-task : 1</code>
Service name	<code>frontend-aitech-service</code>
Desired tasks	<code>1</code>

aws Search [Alt+S] Account ID: 3336-7327-1195 Asia Pacific (Mumbai) Abineshwar%208

VPC EC2

Amazon Elastic Container Service > Clusters > frontend-aitech-cluster > Create service

Create service Info

Service details

Task definition family
Select an existing task definition. To create a new task definition, go to [Task definitions](#).
 [\(C\)](#)

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.
 [\(X\)](#) [\(C\)](#)

Service name
Assign a service name that is unique for this cluster.
 Up to 255 letters (uppercase and lowercase), numbers, underscores and hyphens are allowed. Service names must be unique within a cluster.

Environment AWS Fargate

Existing cluster

Compute configuration - advanced

Compute options Info
To ensure task distribution across your compute types, use appropriate compute options.
 Capacity provider strategy
Specify a launch strategy to distribute your tasks across one or more capacity providers.
 Launch type
Launch tasks directly without the use of a capacity provider strategy.

Launch type Info
Select either managed capacity (Fargate), or custom capacity (EC2 or user-managed, External instances). External instances are registered to your cluster using the ECS Anywhere capability.
 [\(▼\)](#)

Platform version Info
Specify the platform version on which to run your service.
 [\(▼\)](#)

Troubleshooting configuration - recommended

Deployment configuration

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

CloudShell Feedback Console Mobile App

aws Search [Alt+S] Account ID: 3336-7327-1195 Asia Pacific (Mumbai) Abineshwar%208

VPC EC2

Amazon Elastic Container Service > Clusters > frontend-aitech-cluster > Create service

Create service Info

Environment AWS Fargate

Existing cluster

Compute configuration - advanced

Compute options Info
To ensure task distribution across your compute types, use appropriate compute options.
 Capacity provider strategy
Specify a launch strategy to distribute your tasks across one or more capacity providers.
 Launch type
Launch tasks directly without the use of a capacity provider strategy.

Launch type Info
Select either managed capacity (Fargate), or custom capacity (EC2 or user-managed, External instances). External instances are registered to your cluster using the ECS Anywhere capability.
 [\(▼\)](#)

Platform version Info
Specify the platform version on which to run your service.
 [\(▼\)](#)

Troubleshooting configuration - recommended

Deployment configuration

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

CloudShell Feedback Console Mobile App

The screenshot shows the AWS ECS console with the following details:

- Scheduling strategy:** Replica (selected)
- Desired tasks:** 1
- Availability Zone re-balancing:** Turned on (checkbox checked)
- Health check grace period:** 0 seconds
- Deployment options:** Deployment failure detection

STEP 11.2: Networking

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

ECS Task Security Group – Inbound rule:

HTTP | TCP | 80 | ALB-Security-Group

VPC | Info
Select a VPC to use for your Amazon ECS resources.

vpc-0bb916360bb4e70d deployment-project-VPC

Subnets
Choose the subnets within the VPC that the task scheduler should consider for placement.

Choose subnets Clear current selection

subnet-03f425e660184ff8b deployment-project-VPC-subnet-public2-ap-south-1b ap-south-1b 10.0.16.0/20

subnet-0624ce757b655ff9 deployment-project-VPC-subnet-public1-ap-south-1a ap-south-1a 10.0.0.0/20

Security group | Info
Choose an existing security group or create a new security group.

Use an existing security group Create a new security group

Security group name
Choose an existing security group.

Choose security groups

sg-0bd64cf7fd3b0940d deployment-project-sg sg-0dcc14845b66358c6 default

Public IP | Info
Choose whether to auto-assign a public IP to the task's elastic network interface (ENI).

Public IP | Info
Choose whether to auto-assign a public IP to the task's elastic network interface (ENI).

Turned off

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**

The screenshot shows the 'Create service' wizard for an Amazon Elastic Container Service cluster named 'frontend-aitech-cluster'. The first step, 'Load balancing - optional', is displayed. It includes a checkbox for 'Use load balancing' which is checked, and a dropdown menu showing the VPC 'vpc-0bba916360bb4e70d'. Below this, there are two options for 'Load balancer type': 'Application Load Balancer' (selected) and 'Network Load Balancer'. Under 'Container', it shows 'frontend-aitech 80:80'. A note says 'Specify whether to create a new load balancer or choose an existing one.' with radio buttons for 'Create a new load balancer' (unchecked) and 'Use an existing load balancer' (checked). The bottom of the screen shows standard AWS navigation and footer links.

The screenshot shows the second step of the 'Create service' wizard, focusing on the 'Application Load Balancer' configuration. It shows the 'Host port:Container port' field set to 'frontend-aitech-alb' and 'internet-facing'. The 'Listener' dropdown is set to 'HTTP:80'. The 'Listener rules for 80:HTTP (1)' section shows a single rule with priority 'default' and target group 'frontend-aitech-tg'. The bottom of the screen shows standard AWS navigation and footer links.

aws | ⚙️ | Search [Alt+S] | Account ID: 3336-7327-1195 | Asia Pacific (Mumbai) | Abineshwar%208

VPC EC2

Amazon Elastic Container Service > Clusters > frontend-aitech-cluster > Create service

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

CloudShell Feedback Console Mobile App

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	/	frontend-aitech-tg

Target group | Info

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 | Info

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.

Cancel Create

aws | ⚙️ | Search [Alt+S] | Account ID: 3336-7327-1195 | Asia Pacific (Mumbai) | Abineshwar%208

VPC EC2

Amazon Elastic Container Service > Clusters > frontend-aitech-cluster > Create service

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

CloudShell Feedback Console Mobile App

▶ 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.

Cancel Create

The screenshot shows the AWS Elastic Container Service (ECS) Cluster overview for the 'frontend-aitech-cluster'. The cluster ARN is arn:aws:ecs:ap-south-1:1333673271195:cluster/frontend-aitech-cluster, and it is currently active. CloudWatch monitoring is set to 'Default'. There are no registered container instances. Under the 'Services' section, there is one service named 'Training' which is active. Under the 'Tasks' section, there is one task named 'Pending' which is running. The navigation bar at the top includes links for VPC, EC2, and the Amazon Elastic Container Service. The left sidebar provides access to various AWS services like Express Mode, Clusters, Namespaces, Task definitions, Account settings, Amazon ECR, and AWS Batch.

You'll get an error after completing this.

Click your service → Update → Verify the below images.

The screenshot shows the 'Update' page for the 'frontend-aitech-task-service-of9juxqq' service. Under the 'Deployment configuration' section, the 'Force new deployment' checkbox is checked. In the 'Task definition family' dropdown, 'frontend-aitech-task' is selected. The 'Task definition revision' dropdown is set to 'Latest'. Under 'Scheduling strategy', 'REPLICAS' is chosen. In the 'Desired tasks' section, the value '1' is entered. The 'Availability Zone re-balancing' section has the 'Turn on Availability Zone re-balancing' checkbox checked. The 'Health check grace period' section is also visible. The left sidebar and top navigation bar are identical to the previous screenshot.

Screenshot of the AWS Elastic Container Service (ECS) console showing the configuration of a service named "frontend-aitech-task-service-of9juxqq" within a cluster "frontend-aitech-cluster".

The left sidebar shows the navigation path: Amazon Elastic Container Service > Clusters > frontend-aitech-cluster > Services > frontend-aitech-task-service-of9juxqq > Update.

The main configuration pane includes the following sections:

- Availability Zone re-balancing**: A checkbox labeled "Turn on Availability Zone re-balancing" is checked, with a note that ECS automatically detects imbalances and redistributes tasks.
- Health check grace period**: Set to 0 seconds.
- Deployment options**: Includes links for Deployment failure detection, Compute configuration - advanced, and Troubleshooting configuration - recommended.
- Networking**:
 - VPC**: Set to "vpc-0ba916360bb4e70d deployment-project-VPC".
 - Subnets**: Set to "subnet-03f425e60184ff8b deployment-project-VPC-subnet-public2-ap-south-1b" and "subnet-0624ce757b655ff19 deployment-project-VPC-subnet-public1-ap-south-1a".
 - Security groups**: Set to "sg-0bd64c77bd3b0a40d deployment-project-sg" and "sg-0dcc14845b66558c6 default".
 - Public IP**: A checkbox for "Auto-assign a public IP to the task's elastic network interface (ENI)" is checked.
- Service Connect - optional**: A link for configuring service connect.

At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information and privacy terms.

Screenshot of the AWS Elastic Container Service (ECS) console showing the continuation of the configuration for the service "frontend-aitech-task-service-of9juxqq".

The left sidebar shows the same navigation path: Amazon Elastic Container Service > Clusters > frontend-aitech-cluster > Services > frontend-aitech-task-service-of9juxqq > Update.

The main configuration pane continues the Networking section from the previous screenshot:

- VPC**: Set to "vpc-0ba916360bb4e70d deployment-project-VPC".
- Subnets**: Set to "subnet-03f425e60184ff8b deployment-project-VPC-subnet-public2-ap-south-1b" and "subnet-0624ce757b655ff19 deployment-project-VPC-subnet-public1-ap-south-1a".
- Security groups**: Set to "sg-0bd64c77bd3b0a40d deployment-project-sg" and "sg-0dcc14845b66558c6 default".
- Public IP**: A checkbox for "Auto-assign a public IP to the task's elastic network interface (ENI)" is checked.

Below the networking section, there is a link for "Service Connect - optional".

At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information and privacy terms.

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**

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)

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: UAGPikg08k7e8ZjDTG0t	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.

CloudShell Feedback Console Mobile App

STEP 12: Verify & Access Website

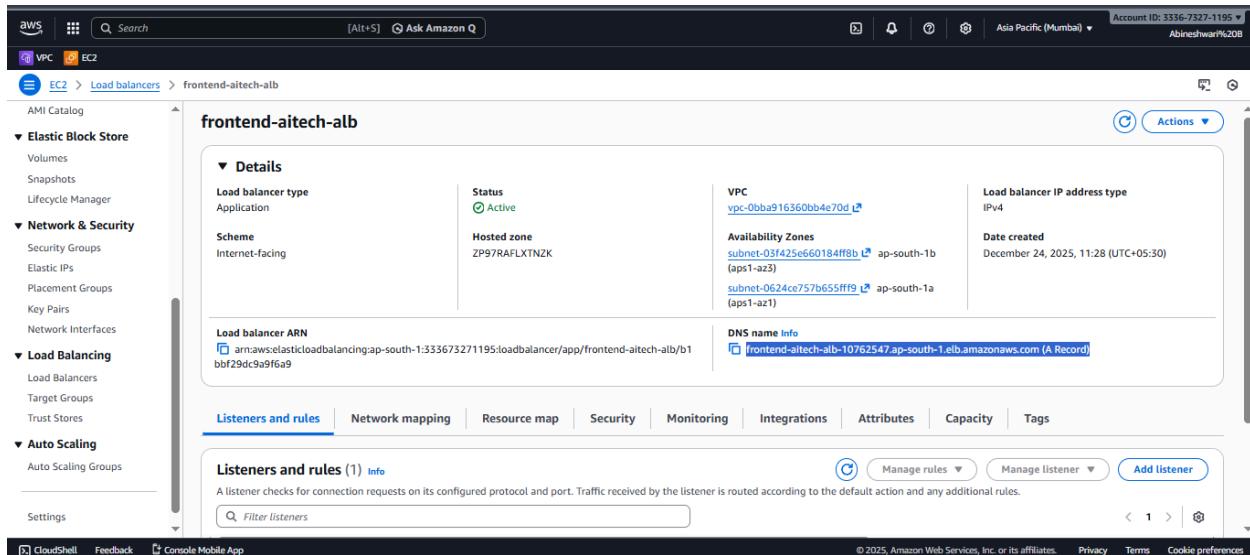
1. Go to EC2 → Load Balancers
2. Copy the DNS name of **frontend-aitech-alb**

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 CloudWatch Metrics interface. On the left, a navigation sidebar includes links for VPC, EC2, AMI Catalog, Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, and Settings. The main content area displays a metric named 'frontend-aitech-alb' with a 'Details' section. It shows the load balancer type as Application, status as Active, and hosted zone as ZP97RAFLXTNZK. The VPC is listed as vpc-0bba916360bb4e70d. Availability zones include subnet-03f425e660184ff8b (ap-south-1b, aps1-az3) and subnet-0624ce757b655ff9 (ap-south-1a, aps1-az1). The DNS name is frontend-aitech-alb-10762547.ap-south-1.elb.amazonaws.com (A Record). Below this, there are tabs for Listeners and rules, Network mapping, Resource map, Security, Monitoring, Integrations, Attributes, Capacity, and Tags. The 'Listeners and rules' tab is selected, showing one rule with the option to manage rules or add a listener.

