Shaikh Ateeb Ahmed
03-10-2025

CloudFront with ALB + EC2 (Free Tier)

Step 1: Launch an EC2 instance

Open the EC2 Console

Sign in to the AWS Management Console → **EC2**.

Create the ALB security group (ALB-SG)

In the EC2 left menu → **Network & Security** → **Security Groups** → **Create security group**.

Set:

Name: ALB-SG

Description: Allow HTTP from Internet (for ALB)

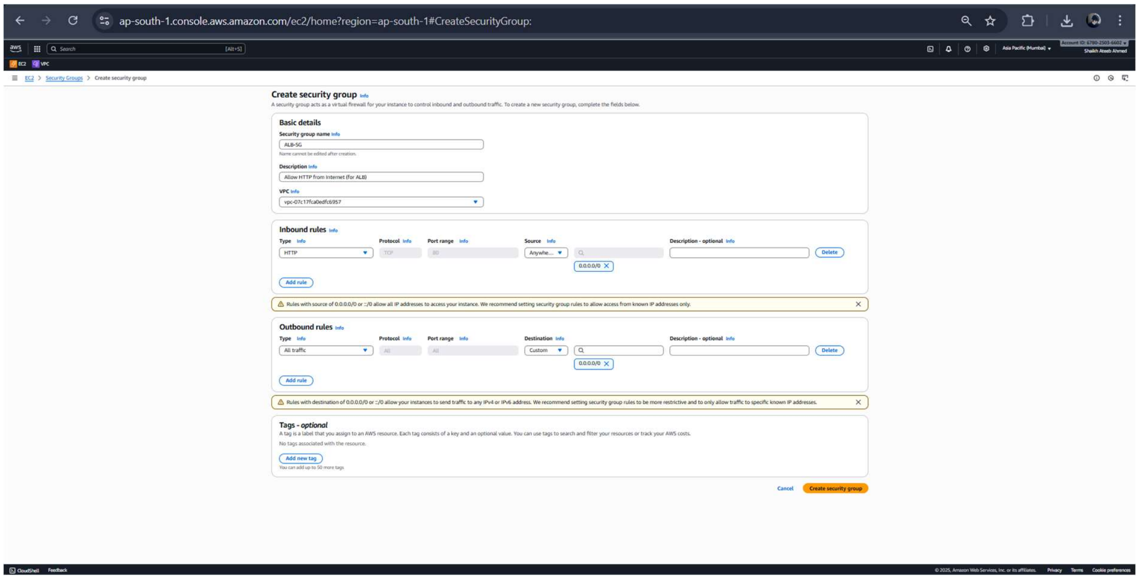
VPC: choose the same VPC you will use for your instance.

Inbound rules → **Add rule**:

Type: **HTTP**, Port: **80**, Source: **Anywhere (0.0.0.0/0)**

Leave outbound as default (allow all).

Click **Create security group**.



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

EC2 > Security Groups > Create security group

Create security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name: ALB-SG

Name cannot be edited after creation.

Description: Allow HTTP from Internet (for ALB)

VPC: vpc-07c17fca0edfc6957

Inbound rules

Type	Protocol	Port range	Source		Description - optional
HTTP	TCP	80	Anywhe...	0.0.0.0/0	

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

Outbound rules

Type	Protocol	Port range	Destination		Description - optional
All traffic	All	All	Custom	0.0.0.0/0	

Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses.

Tags - *optional*

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

You can add up to 50 more tags

Create the EC2 security group (EC2-SG)

Still in **Security Groups** → **Create security group**.

Set:

Shaikh Ateeb Ahmed
03-10-2025

Name: EC2-SG

Description: Allow SSH from my IP and HTTP from ALB

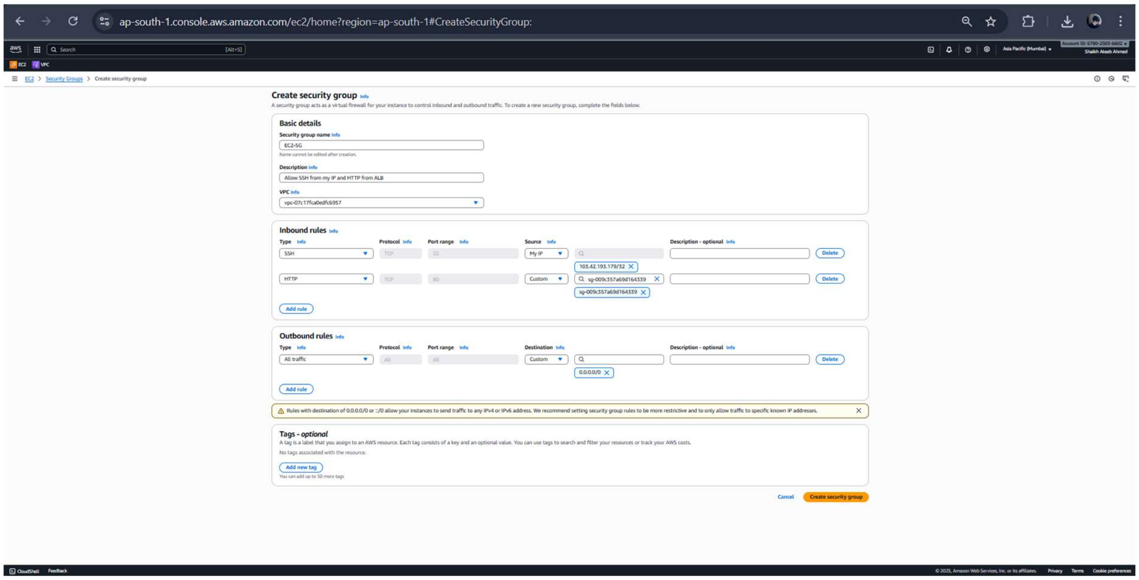
VPC: same VPC as ALB-SG and instance.

Inbound rules → add:

SSH — Port **22** — **Source**: **My IP** (click "My IP" to auto-fill your public IP).

HTTP — Port **80** — **Source**: **Custom** → *select the ALB security group you created*.

Create the security group.



ap-south-1.console.amazon.com/ec2/home?region=ap-south-1#CreateSecurityGroup:
EC2 > Security Groups > Create security group
Asia Pacific (Mumbai)
Shaikh Ateeb Ahmed
Add rule
Delete
Cancel
Create security group
Add new tag

Create security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name: EC2-SG

Name cannot be edited after creation.

Description: Allow SSH from my IP and HTTP from ALB

VPC: vpc-07c17fca0edfc6957

Inbound rules

Type	Protocol	Port range	Source		Description - optional
SSH	TCP	22	My IP	103.42.193.179/32	
HTTP	TCP	80	Custom	sg-009c357a69d164339	

Outbound rules

Type	Protocol	Port range	Destination		Description - optional
All traffic	All	All	Custom	0.0.0.0/0	

Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses.

Tags - *optional*

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

You can add up to 50 more tags

Launch EC2 instance (Amazon Linux 2, Free Tier)

EC2 → **Instances** → **Launch instances**.

Configure:

Name: e.g., web-server-1.

AMI: **Amazon Linux 2** (Free Tier eligible).

Instance type: **t3.micro** (or t2.micro if t3 not available).

Key pair: choose existing.

Network: choose the same **VPC** as your security groups.

Subnet: choose a **public subnet** (one that assigns a public IP) if you want to test via the public IP.

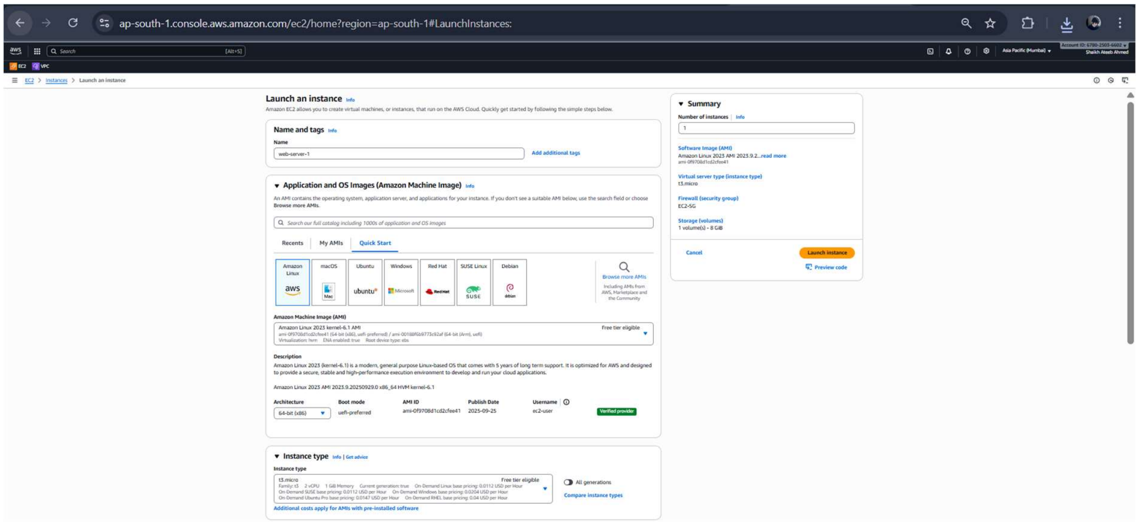
Auto-assign Public IP: **Enable** (for quick testing).

Shaikh Ateeb Ahmed
03-10-2025

Security group: select **Select an existing security group** → choose EC2-SG.

Storage/Tags: set as desired (default 8 GiB is fine).

Click **Launch instance.**



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

EC2 > Instances > Launch an instance

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: web-server-1

Application and OS Images (Amazon Machine Image)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose Browse more AMIs.

Quick Start: Amazon Linux

Amazon Machine Image (AMI): Amazon Linux 2023 kernel-6.1 AMI — Free tier eligible
ami-0f9708d1cd2cfee41 (64-bit (x86), uefi-preferred) / ami-00188f6b9773c92af (64-bit (Arm), uefi)
Virtualization: hvm   ENA enabled: true   Root device type: ebs

Description
Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.9.20250929.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Publish Date	Username
64-bit (x86)	uefi-preferred	ami-0f9708d1cd2cfee41	2025-09-25	ec2-user

Instance type

Instance type: t3.micro — Free tier eligible
Family: t3   2 vCPU   1 GiB 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

Summary

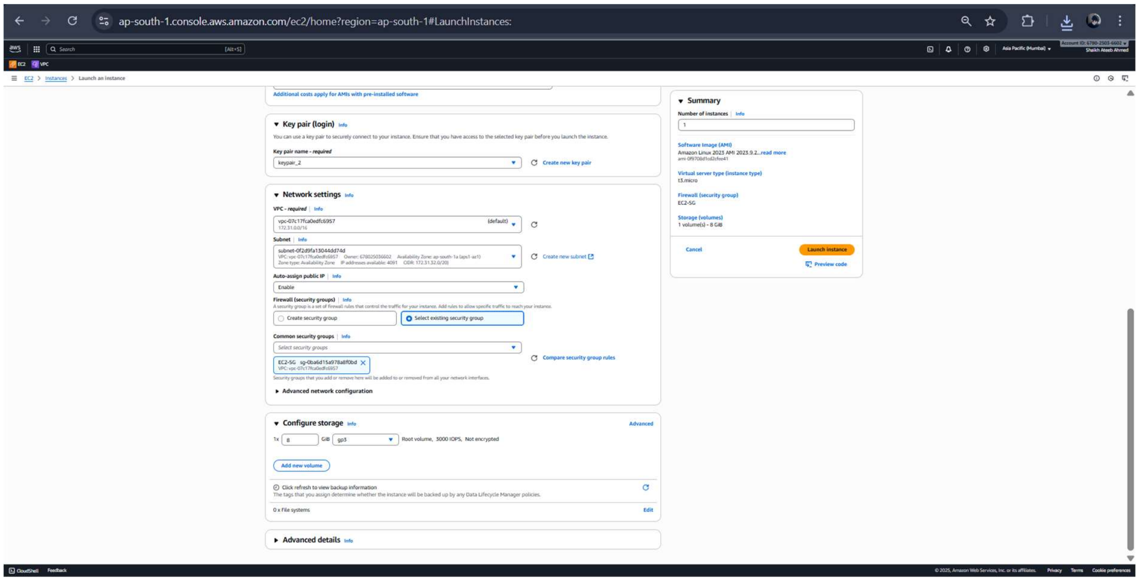
Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.9.2...read more
ami-0f9708d1cd2cfee41

Virtual server type (instance type): t3.micro

Firewall (security group): EC2-SG

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



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

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*: keypair_2

Network settings

VPC - *required*: vpc-07c17fca0edfc6957 (default)
172.31.0.0/16

Subnet: subnet-0f2d9fa13044dd74d
VPC: vpc-07c17fca0edfc6957   Owner: 678025036602   Availability Zone: ap-south-1a (aps1-az1)
Zone type: Availability Zone   IP addresses available: 4091   CIDR: 172.31.32.0/20)

Auto-assign public IP: Enable

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

○ Create security group   ● Select existing security group

Common security groups: EC2-SG sg-0ba6d15a978a8f0bd
VPC: vpc-07c17fca0edfc6957

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

Configure storage

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

Click refresh to view backup information
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.9.2...read more
ami-0f9708d1cd2cfee41

Virtual server type (instance type): t3.micro

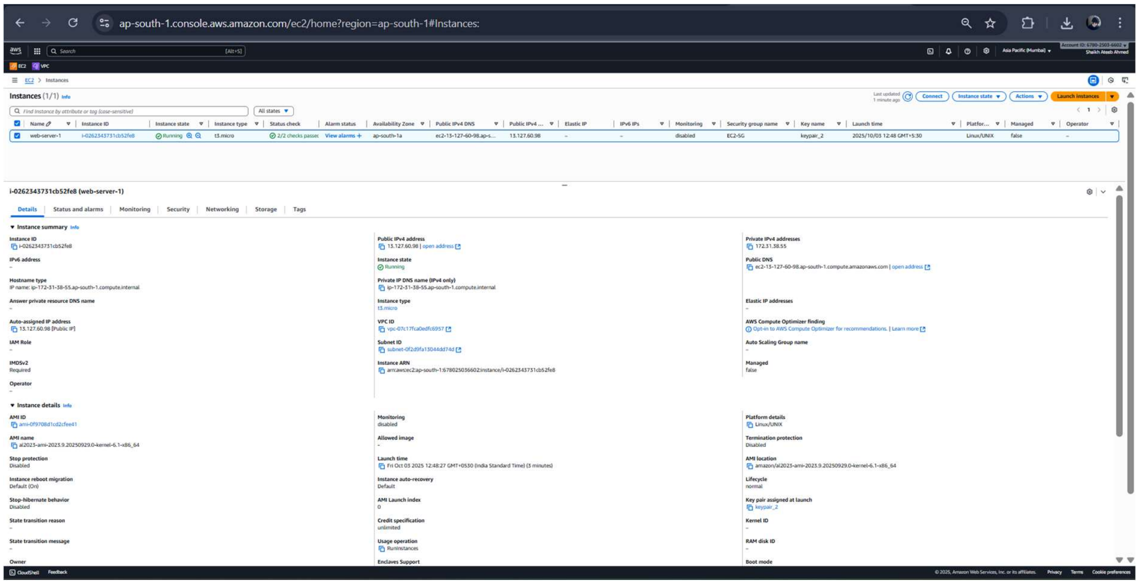
Firewall (security group): EC2-SG

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

Wait & confirm instance state

EC2 → **Instances** → wait for **Instance state** = running and **Status checks** = 2/2 checks passed.

Shaikh Ateeb Ahmed
03-10-2025



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

Instances (1/1)

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name	Launch time	Platfor...	Managed	Operator
☑	web-server-1	i-0262343731cb52fe8	Running	t3.micro	2/2 checks passec	View alarms +	ap-south-1a	ec2-13-127-60-98.ap-s...	13.127.60.98	–	–	disabled	EC2-SG	keypair_2	2025/10/03 12:48 GMT+5:30	Linux/UNIX	false	–

i-0262343731cb52fe8 (web-server-1)

▼ Instance summary

Instance ID
i-0262343731cb52fe8

IPv6 address
–

Hostname type
IP name: ip-172-31-38-55.ap-south-1.compute.internal

Answer private resource DNS name
–

Auto-assigned IP address
13.127.60.98 [Public IP]

IAM Role
–

IMDSv2
Required

Operator
–

Public IPv4 address
13.127.60.98 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-38-55.ap-south-1.compute.internal

Instance type
t3.micro

VPC ID
vpc-07c17fca0edfc6957

Subnet ID
subnet-0f2d9fa13044dd74d

Instance ARN
arn:aws:ec2:ap-south-1:678025036602:instance/i-0262343731cb52fe8

Private IPv4 addresses
172.31.38.55

Public DNS
ec2-13-127-60-98.ap-south-1.compute.amazonaws.com | open address

Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name
–

Managed
false

▼ Instance details

AMI ID
ami-0f9708d1cd2cfee41

AMI name
al2023-ami-2023.9.20250929.0-kernel-6.1-x86_64

Stop protection
Disabled

Instance reboot migration
Default (On)

Stop-hibernate behavior
Disabled

State transition reason
–

State transition message
–

Owner

Monitoring
disabled

Allowed image
-

Launch time
Fri Oct 03 2025 12:48:27 GMT+0530 (India Standard Time) (3 minutes)

Instance auto-recovery
Default

AMI Launch index
0

Credit specification
unlimited

Usage operation
RunInstances

Enclaves Support

Platform details
Linux/UNIX

Termination protection
Disabled

AMI location
amazon/al2023-ami-2023.9.20250929.0-kernel-6.1-x86_64

Lifecycle
normal

Key pair assigned at launch
keypair_2

Kernel ID
–

RAM disk ID
–

Boot mode

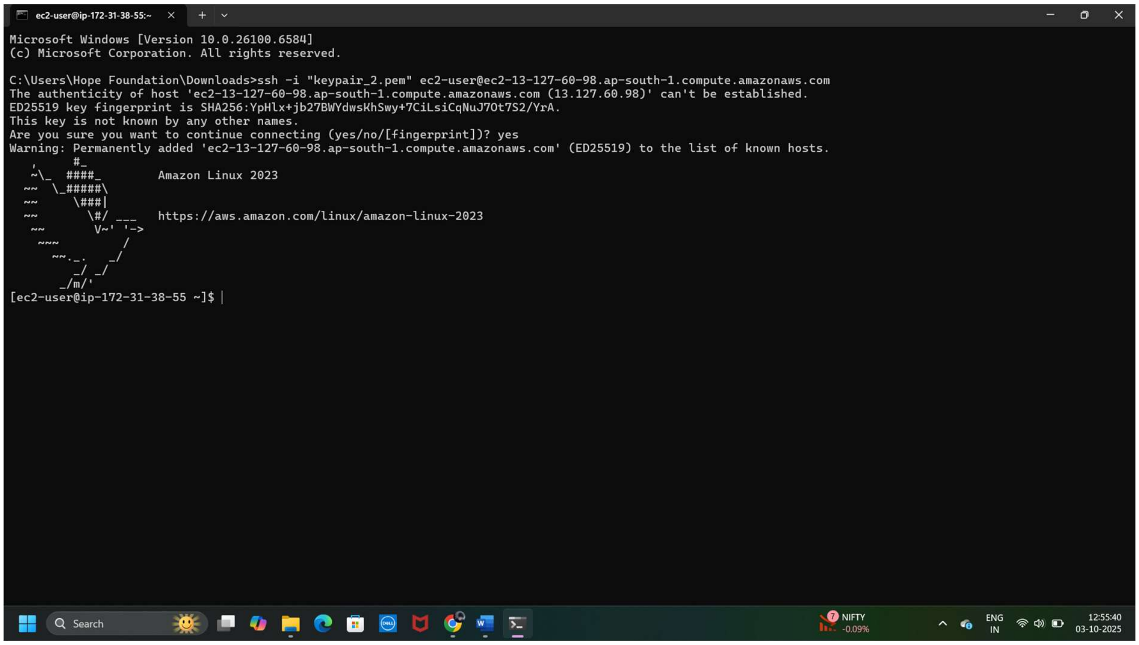
SSH in and install Apache (quick verification)

On your workstation:

# replace path and address

chmod 400 /path/to/your-key.pem

ssh -i /path/to/your-key.pem ec2-user@<EC2-Public-IP>



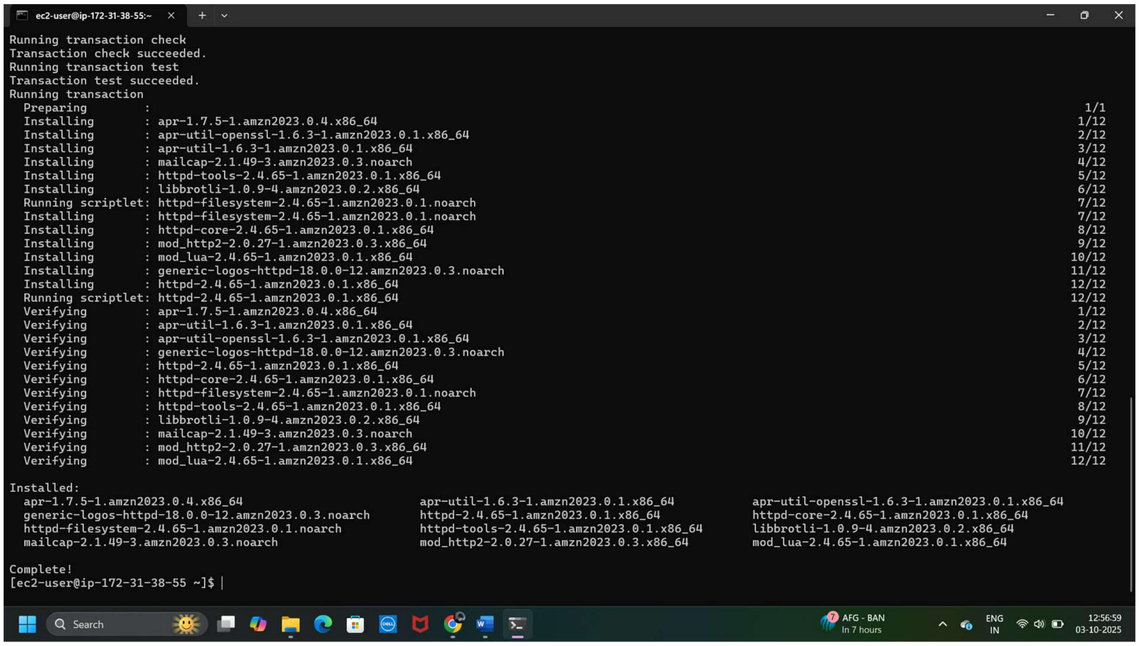
```
Microsoft Windows [Version 10.0.26100.6584]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Hope Foundation\Downloads>ssh -i "keypair_2.pem" ec2-user@ec2-13-127-60-98.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-127-60-98.ap-south-1.compute.amazonaws.com (13.127.60.98)' can't be established.
ED25519 key fingerprint is SHA256:YpHlx+jb27BWYdwsKhSwy+7CiLsiCqNuJ7Ot7S2/YrA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-127-60-98.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-38-55 ~]$ |
```

On the instance, run:

sudo yum update -y

Shaikh Ateeb Ahmed
03-10-2025

sudo yum install -y httpd

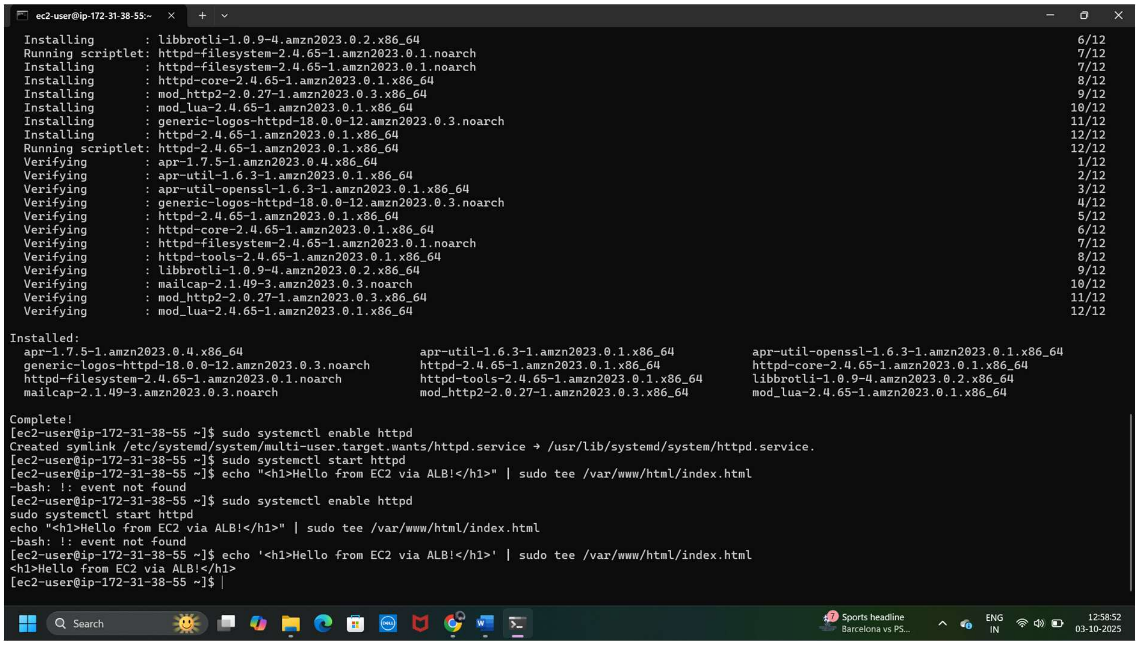


```
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                            1/1
  Installing       : apr-1.7.5-1.amzn2023.0.4.x86_64                           1/12
  Installing       : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64              2/12
  Installing       : apr-util-1.6.3-1.amzn2023.0.1.x86_64                      3/12
  Installing       : mailcap-2.1.49-3.amzn2023.0.3.noarch                      4/12
  Installing       : httpd-tools-2.4.65-1.amzn2023.0.1.x86_64                  5/12
  Installing       : libbrotli-1.0.9-4.amzn2023.0.2.x86_64                     6/12
  Running scriptlet: httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch             7/12
  Installing       : httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch             7/12
  Installing       : httpd-core-2.4.65-1.amzn2023.0.1.x86_64                   8/12
  Installing       : mod_http2-2.0.27-1.amzn2023.0.3.x86_64                    9/12
  Installing       : mod_lua-2.4.65-1.amzn2023.0.1.x86_64                     10/12
  Installing       : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch        11/12
  Installing       : httpd-2.4.65-1.amzn2023.0.1.x86_64                       12/12
  Running scriptlet: httpd-2.4.65-1.amzn2023.0.1.x86_64                       12/12
  Verifying        : apr-1.7.5-1.amzn2023.0.4.x86_64                           1/12
  Verifying        : apr-util-1.6.3-1.amzn2023.0.1.x86_64                      2/12
  Verifying        : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64              3/12
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch         4/12
  Verifying        : httpd-2.4.65-1.amzn2023.0.1.x86_64                        5/12
  Verifying        : httpd-core-2.4.65-1.amzn2023.0.1.x86_64                   6/12
  Verifying        : httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch             7/12
  Verifying        : httpd-tools-2.4.65-1.amzn2023.0.1.x86_64                  8/12
  Verifying        : libbrotli-1.0.9-4.amzn2023.0.2.x86_64                     9/12
  Verifying        : mailcap-2.1.49-3.amzn2023.0.3.noarch                     10/12
  Verifying        : mod_http2-2.0.27-1.amzn2023.0.3.x86_64                   11/12
  Verifying        : mod_lua-2.4.65-1.amzn2023.0.1.x86_64                     12/12

Installed:
  apr-1.7.5-1.amzn2023.0.4.x86_64                        apr-util-1.6.3-1.amzn2023.0.1.x86_64              apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch      httpd-2.4.65-1.amzn2023.0.1.x86_64                httpd-core-2.4.65-1.amzn2023.0.1.x86_64
  httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch          httpd-tools-2.4.65-1.amzn2023.0.1.x86_64          libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  mailcap-2.1.49-3.amzn2023.0.3.noarch                   mod_http2-2.0.27-1.amzn2023.0.3.x86_64            mod_lua-2.4.65-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-38-55 ~]$ |
```

sudo systemctl enable httpd

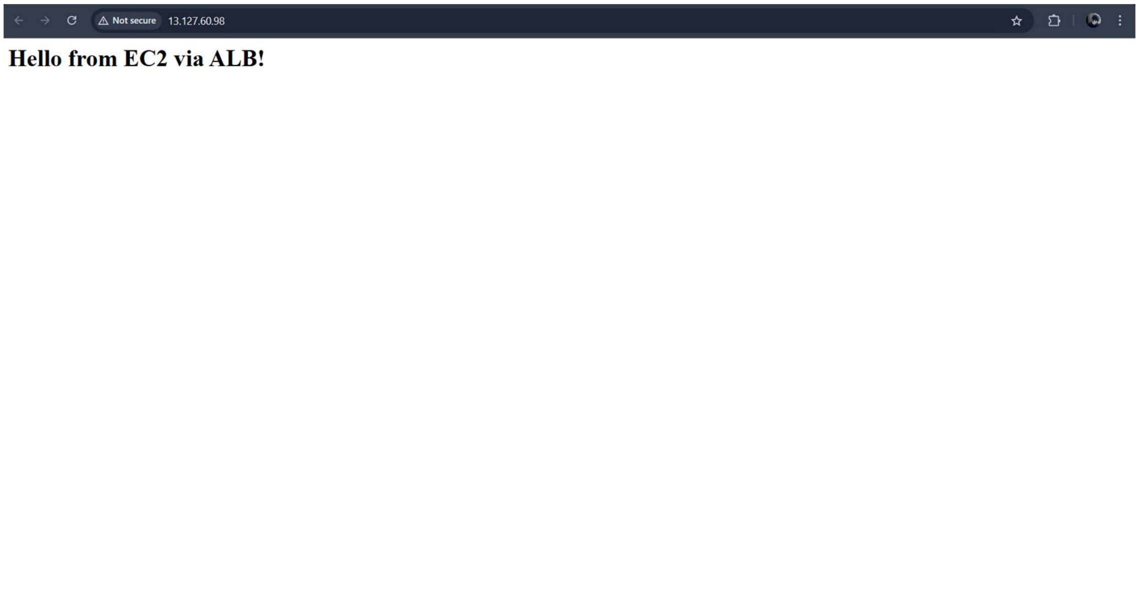
sudo systemctl start httpd

echo '<h1>Hello from EC2 via ALB!</h1>' | sudo tee /var/www/html/index.html



```
  Installing       : libbrotli-1.0.9-4.amzn2023.0.2.x86_64                     6/12
  Running scriptlet: httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch             7/12
  Installing       : httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch             7/12
  Installing       : httpd-core-2.4.65-1.amzn2023.0.1.x86_64                   8/12
  Installing       : mod_http2-2.0.27-1.amzn2023.0.3.x86_64                    9/12
  Installing       : mod_lua-2.4.65-1.amzn2023.0.1.x86_64                     10/12
  Installing       : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch        11/12
  Installing       : httpd-2.4.65-1.amzn2023.0.1.x86_64                       12/12
  Running scriptlet: httpd-2.4.65-1.amzn2023.0.1.x86_64                       12/12
  Verifying        : apr-1.7.5-1.amzn2023.0.4.x86_64                           1/12
  Verifying        : apr-util-1.6.3-1.amzn2023.0.1.x86_64                      2/12
  Verifying        : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64              3/12
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch         4/12
  Verifying        : httpd-2.4.65-1.amzn2023.0.1.x86_64                        5/12
  Verifying        : httpd-core-2.4.65-1.amzn2023.0.1.x86_64                   6/12
  Verifying        : httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch             7/12
  Verifying        : httpd-tools-2.4.65-1.amzn2023.0.1.x86_64                  8/12
  Verifying        : libbrotli-1.0.9-4.amzn2023.0.2.x86_64                     9/12
  Verifying        : mailcap-2.1.49-3.amzn2023.0.3.noarch                     10/12
  Verifying        : mod_http2-2.0.27-1.amzn2023.0.3.x86_64                   11/12
  Verifying        : mod_lua-2.4.65-1.amzn2023.0.1.x86_64                     12/12

Installed:
  apr-1.7.5-1.amzn2023.0.4.x86_64                        apr-util-1.6.3-1.amzn2023.0.1.x86_64              apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch      httpd-2.4.65-1.amzn2023.0.1.x86_64                httpd-core-2.4.65-1.amzn2023.0.1.x86_64
  httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch          httpd-tools-2.4.65-1.amzn2023.0.1.x86_64          libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  mailcap-2.1.49-3.amzn2023.0.3.noarch                   mod_http2-2.0.27-1.amzn2023.0.3.x86_64            mod_lua-2.4.65-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-38-55 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-38-55 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-38-55 ~]$ echo "<h1>Hello from EC2 via ALB!</h1>" | sudo tee /var/www/html/index.html
-bash: !: event not found
[ec2-user@ip-172-31-38-55 ~]$ sudo systemctl enable httpd
sudo systemctl start httpd
echo "<h1>Hello from EC2 via ALB!</h1>" | sudo tee /var/www/html/index.html
-bash: !: event not found
[ec2-user@ip-172-31-38-55 ~]$ echo '<h1>Hello from EC2 via ALB!</h1>' | sudo tee /var/www/html/index.html
<h1>Hello from EC2 via ALB!</h1>
[ec2-user@ip-172-31-38-55 ~]$ |
```

Open your browser to http://<EC2-Public-IP>/ — you should see the test page.



Hello from EC2 via ALB!

Step 2: Open EC2 console

In the left menu → **Load Balancers** → **Create Load Balancer** → **Application Load Balancer**.

Configure Load Balancer

Name: my-alb-lab

Scheme: **Internet-facing** (so it's accessible from the Internet)

IP address type: IPv4

Listeners: default **HTTP 80**

Availability Zones / Subnets: select your **VPC** and at least **2 public subnets** (recommended for redundancy)

Click **Next: Configure Security Settings** → skip HTTPS for now (HTTP only, lab purpose)

Configure Security Group for ALB

Select existing security group:

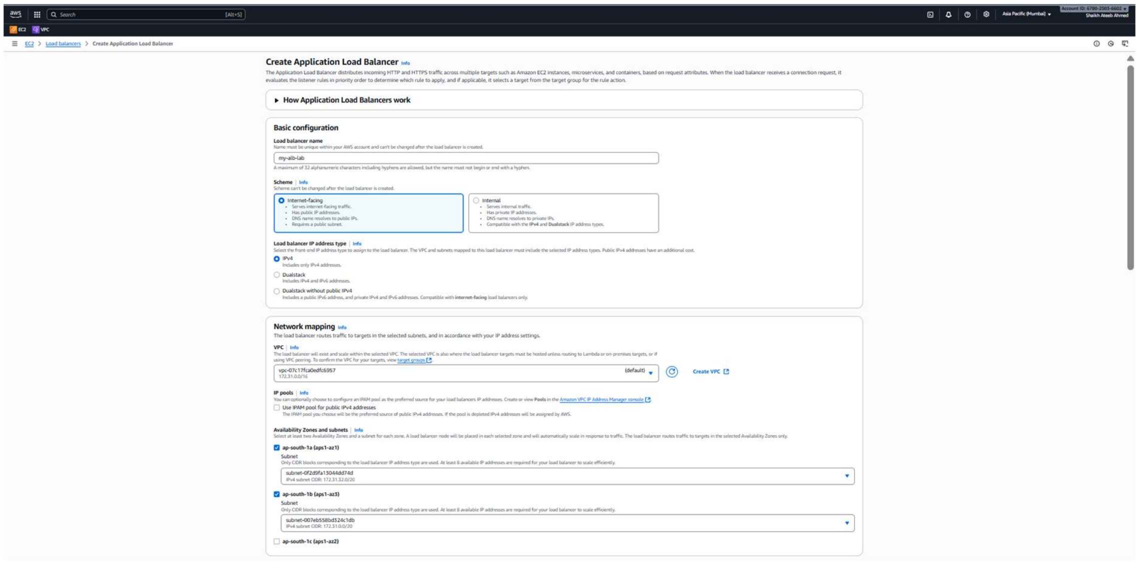
Name: ALB-SG

Inbound rules: HTTP 80 from **0.0.0.0/0**

Outbound: leave default (allow all)

Click **Next: Configure Routing**

Shaikh Ateeb Ahmed
03-10-2025



Create Application Load Balancer Info

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.
my-alb-lab
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme | Info
Scheme can't be changed after the load balancer is created.

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

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

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

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

Network mapping Info
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC | Info
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view target groups.
vpc-07c17fca0edfc6957 (default)
172.31.0.0/16
Create VPC

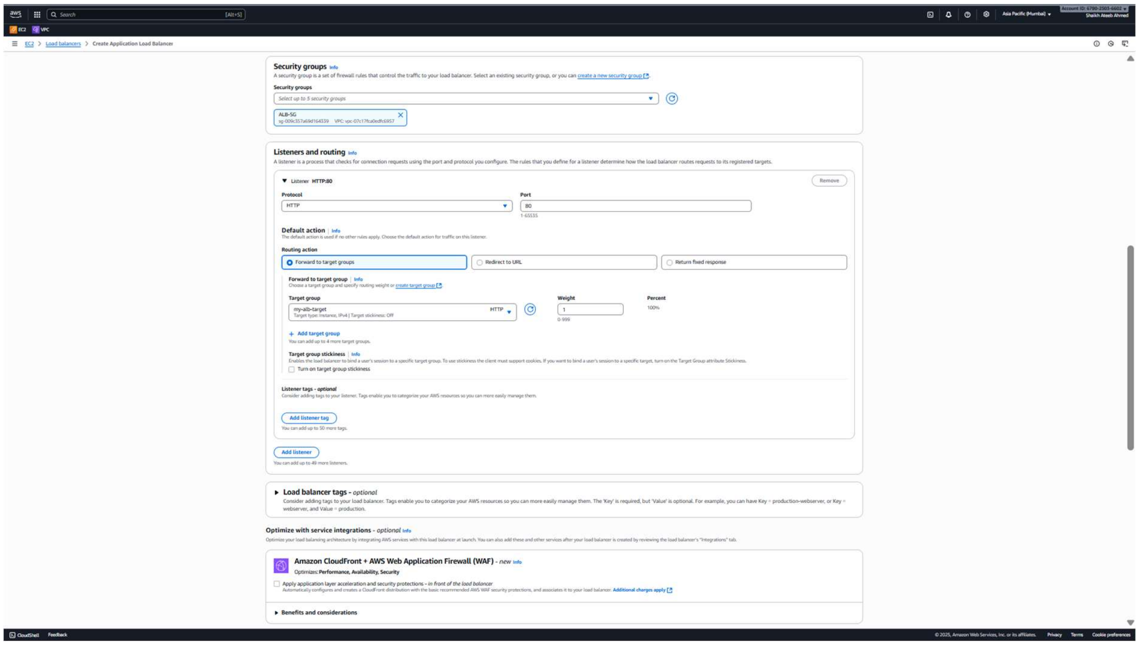
IP pools | Info
You can optionally choose to configure an IPAM pool as the preferred source for your load balancers IP addresses. Create or view **Pools** in the Amazon VPC IP Address Manager console.
□ Use IPAM pool for public IPv4 addresses
The IPAM pool you choose will be the preferred source of public IPv4 addresses. If the pool is depleted IPv4 addresses will be assigned by AWS.

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

☑ ap-south-1a (aps1-az1)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
subnet-0f2d9fa13044dd74d
IPv4 subnet CIDR: 172.31.32.0/20

☑ ap-south-1b (aps1-az3)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
subnet-007eb558bd324c1db
IPv4 subnet CIDR: 172.31.0.0/20

□ ap-south-1c (aps1-az2)



EC2 > Load balancers > Create Application Load Balancer
Asia Pacific (Mumbai)
Shaikh Ateeb Ahmed
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Security groups Info
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can create a new security group.

Security groups
Select up to 5 security groups
ALB-SG
sg-009c357a69d164339 VPC: vpc-07c17fca0edfc6957

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

▼ Listener **HTTP:80** Remove

Protocol: HTTP
Port: 80
1-65535

Default action | Info
The default action is used if no other rules apply. Choose the default action for traffic on this listener.

Routing action
● Forward to target groups ○ Redirect to URL ○ Return fixed response

Forward to target group | Info
Choose a target group and specify routing weight or create target group.

Target group: my-alb-target HTTP
Target type: Instance, IPv4 | Target stickiness: Off
Weight: 1
0-999
Percent: 100%

+ Add target group
You can add up to 4 more target groups.

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

Listener tags - *optional*
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.
Add listener tag
You can add up to 50 more tags.

Add listener
You can add up to 49 more listeners.

▶ Load balancer tags - *optional*
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value = production.

Optimize with service integrations - *optional* Info
Optimize your load balancing architecture by integrating AWS services with this load balancer at launch. You can also add these and other services after your load balancer is created by reviewing the load balancer's "Integrations" tab.

Amazon CloudFront + AWS Web Application Firewall (WAF) - *new* Info
Optimizes: **Performance, Availability, Security**
□ Apply application layer acceleration and security protections - *in front of the load balancer*
Automatically configures and creates a CloudFront distribution with the basic recommended AWS WAF security protections, and associates it to your load balancer. **Additional charges apply**

▶ Benefits and considerations

Create Target Group

Target group name: my-alb-target

Target type: **Instance**

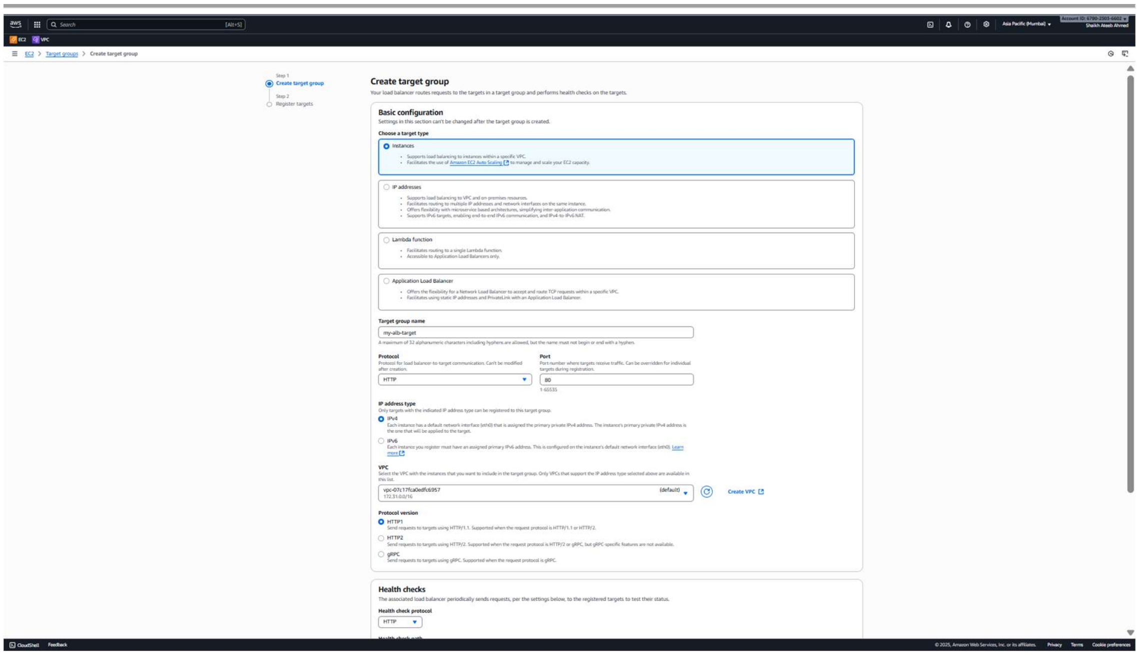
Protocol / Port: HTTP / 80

VPC: choose the same VPC as your EC2 instance

Health checks: default path /

Click **Next: Register Targets**

Shaikh Ateeb Ahmed
03-10-2025



Create target group

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

● Instances
- Supports load balancing to instances within a specific VPC.
- Facilitates the use of Amazon EC2 Auto Scaling to manage and scale your EC2 capacity.

○ IP addresses
- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

○ Lambda function
- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

○ Application Load Balancer
- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name
my-alb-target
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol
Protocol for load balancer-to-target communication. Can't be modified after creation.
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
Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.
○ IPv6
Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). Learn more

VPC
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.
vpc-07c17fca0edfc6957 (default)
172.31.0.0/16
Create VPC

Protocol version
● HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
○ HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.
○ gRPC
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.

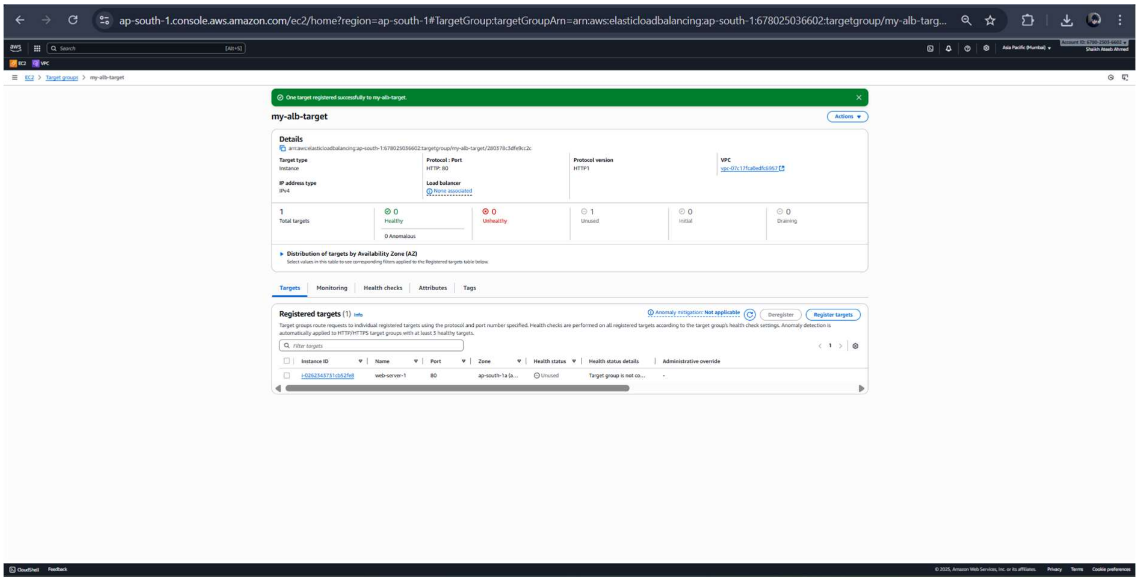
Health check protocol
HTTP

Register EC2 instance

From the list of instances, select your **EC2 instance**.

Click **Add to registered** → it should appear in **Registered targets**

Click **Next: Review** → **Create Load Balancer**



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#TargetGroup:targetGroupArn=arn:aws:elasticloadbalancing:ap-south-1:678025036602:targetgroup/my-alb-targ...

One target registered successfully to my-alb-target.

my-alb-target

Details
arn:aws:elasticloadbalancing:ap-south-1:678025036602:targetgroup/my-alb-target/280378c3dfe9cc2c

Target type: Instance
Protocol : Port: HTTP: 80
Protocol version: HTTP1
VPC: vpc-07c17fca0edfc6957
IP address type: IPv4
Load balancer: None associated

1 Total targets | 0 Healthy | 0 Anomalous | 0 Unhealthy | 1 Unused | 0 Initial | 0 Draining

Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Registered targets (1)

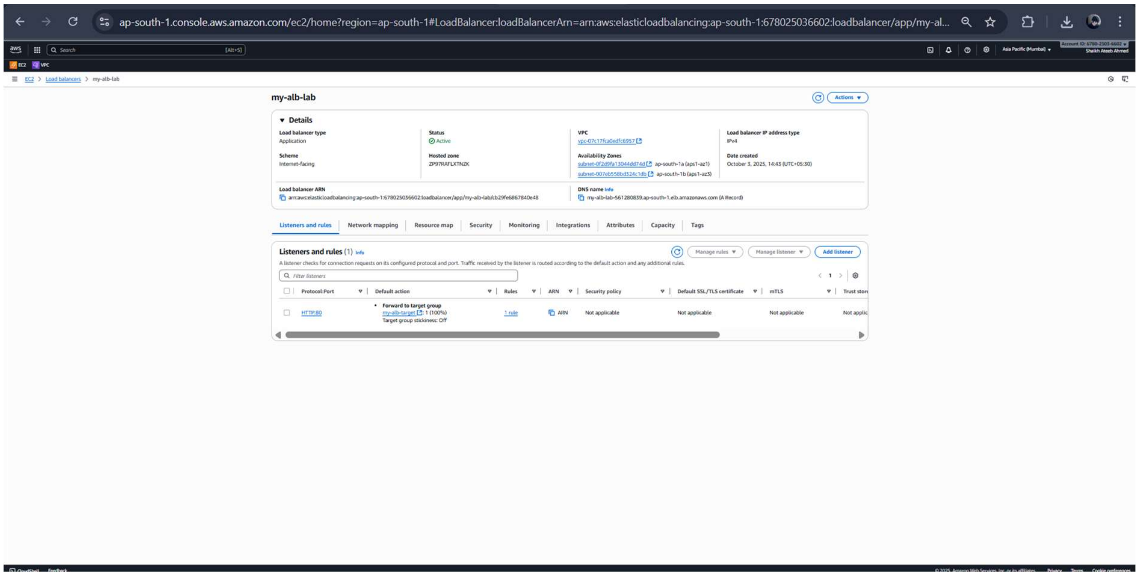
Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Instance ID	Name	Port	Zone	Health status	Health status details	Administrative override
i-0262343731cb52fe8	web-server-1	80	ap-south-1a (a...	Unused	Target group is not co...	-

Wait for ALB to be active

After creation, in **Load Balancers** → select my-alb-lab → check **DNS name**:
Example: my-alb-lab-123456.ap-south-1.elb.amazonaws.com

Shaikh Ateeb Ahmed
03-10-2025



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LoadBalancer:loadBalancerArn=arn:aws:elasticloadbalancing:ap-south-1:678025036602:loadbalancer/app/my-al...

EC2 > Load balancers > my-alb-lab

my-alb-lab

▼ **Details**

Load balancer type Application	**Status** Active	**VPC** vpc-07c17fca0edfc6957	**Load balancer IP address type** IPv4
Scheme Internet-facing	**Hosted zone** ZP97RAFLXTNZK	**Availability Zones** subnet-0f2d9fa13044dd74d ap-south-1a (aps1-az1) subnet-007eb558bd324c1db ap-south-1b (aps1-az3)	**Date created** October 3, 2025, 14:43 (UTC+05:30)

Load balancer ARN
arn:aws:elasticloadbalancing:ap-south-1:678025036602:loadbalancer/app/my-alb-lab/cb29fe6867840e48

DNS name Info
my-alb-lab-561280839.ap-south-1.elb.amazonaws.com (A Record)

Listeners and rules (1) Info

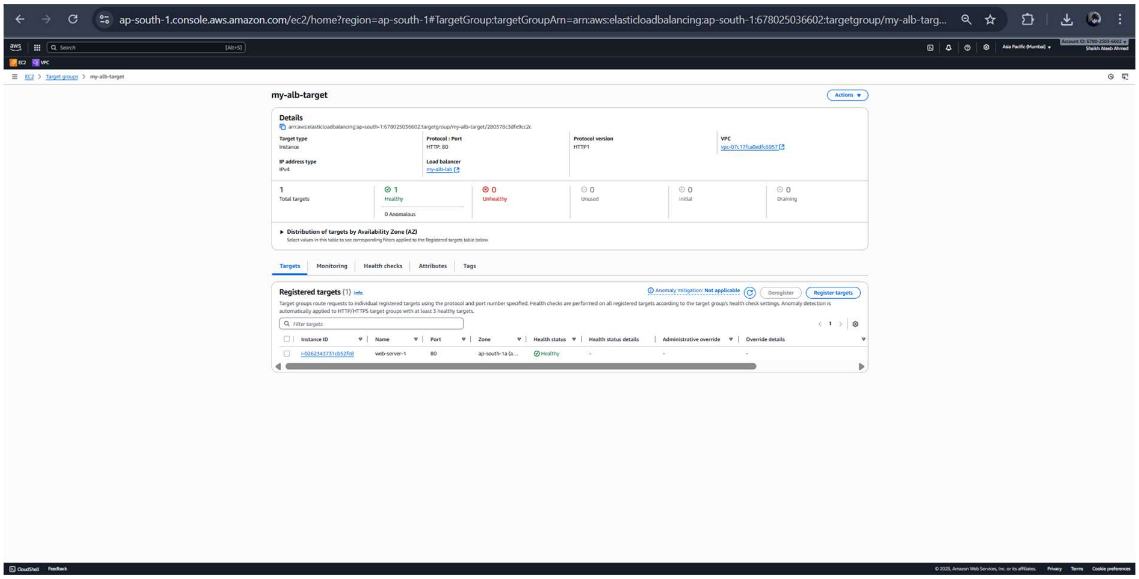
A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS	Trust stor
HTTP:80	• Forward to target group my-alb-target: 1 (100%) Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable	Not applicable	Not applic

Verify Target Health

Load Balancers → select your ALB → **Target Groups → Targets tab**

Your EC2 instance should show **healthy** (green).



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#TargetGroup:targetGroupArn=arn:aws:elasticloadbalancing:ap-south-1:678025036602:targetgroup/my-alb-targ...

EC2 > Target groups > my-alb-target

my-alb-target

Details
arn:aws:elasticloadbalancing:ap-south-1:678025036602:targetgroup/my-alb-target/280378c3dfe9cc2c

Target type Instance	**Protocol : Port** HTTP: 80	**Protocol version** HTTP1	**VPC** vpc-07c17fca0edfc6957
IP address type IPv4	**Load balancer** my-alb-lab		

1 Total targets	1 Healthy 0 Anomalous	0 Unhealthy	0 Unused	0 Initial	0 Draining

▶ **Distribution of targets by Availability Zone (AZ)**
Select values in this table to see corresponding filters applied to the Registered targets table below.

Registered targets (1) Info

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Instance ID	Name	Port	Zone	Health status	Health status details	Administrative override	Override details
i-0262343731cb52fe8	web-server-1	80	ap-south-1a (a...	Healthy	-	-	-

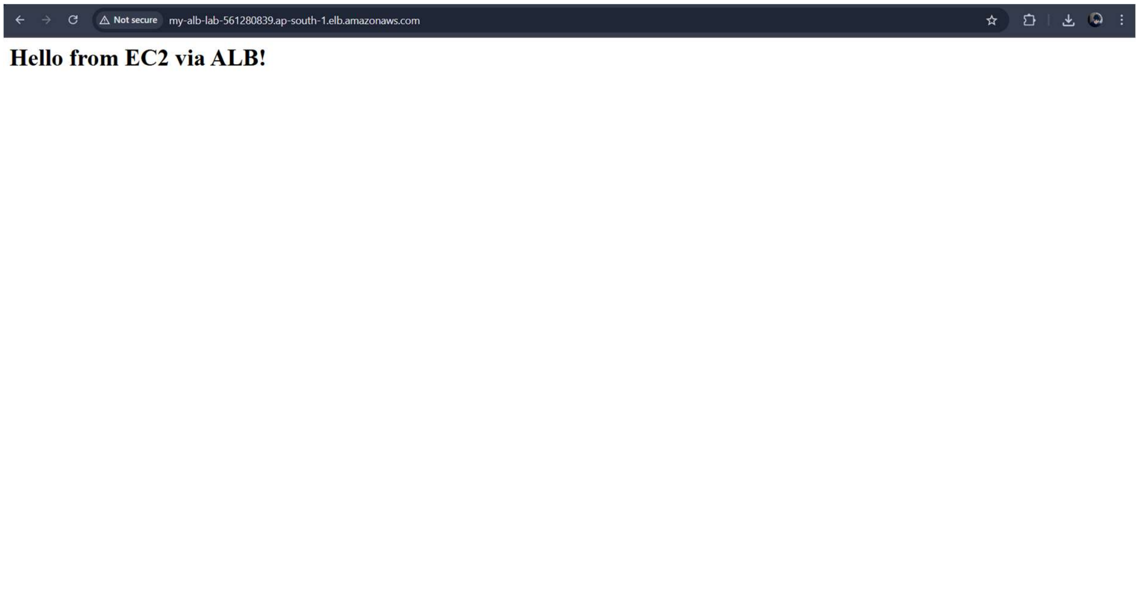
Test the ALB in Browser

Open browser → enter the **ALB DNS name**:

http://my-alb-lab-123456.ap-south-1.elb.amazonaws.com

You should see the Apache test page:

<h1>Hello from EC2 via ALB!</h1>



Hello from EC2 via ALB!

Step 3: Open CloudFront console

Sign in to AWS Management Console → **CloudFront**.

Click **Create distribution** → choose **Web** (or “Web” for new UI).

Configure Origin Settings

Origin Domain Name: paste your **ALB DNS name** (e.g., my-alb-lab-123456.ap-south-1.elb.amazonaws.com).

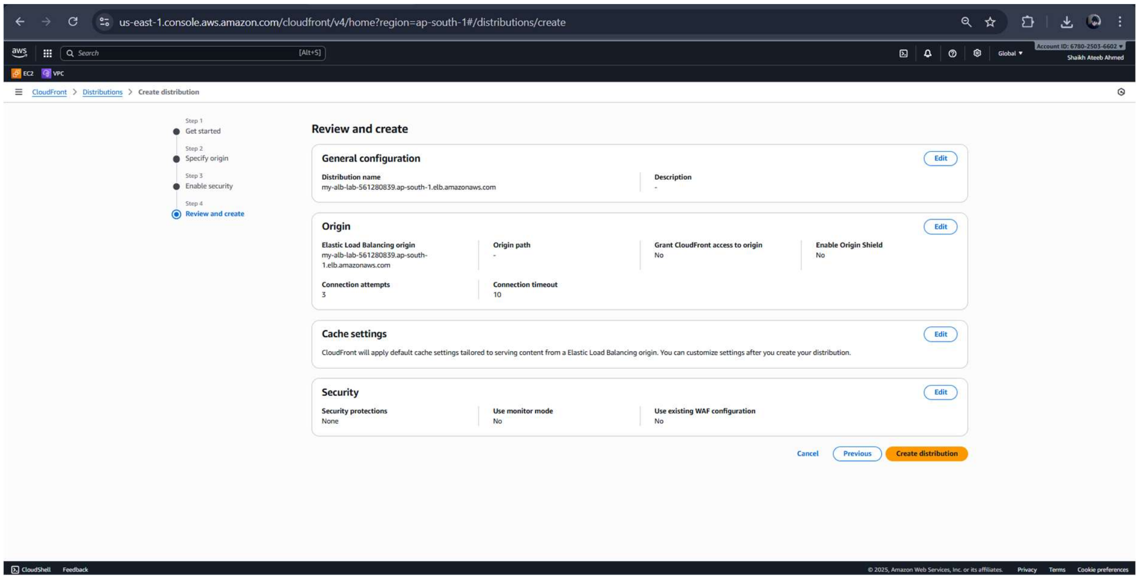
Origin type: choose **Elastic Load Balancer**.

Origin settings: Use recommended origin settings.

Cache settings: Use recommended cache settings tailored to serving Elastic Load Balancing content.

Compress Objects Automatically: Enable (optional, improves speed).

Shaikh Ateeb Ahmed
03-10-2025



us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=ap-south-1#/distributions/create

CloudFront > Distributions > Create distribution

Step 1 Get started
Step 2 Specify origin
Step 3 Enable security
Step 4 Review and create

Review and create

General configuration

Distribution name: my-alb-lab-561280839.ap-south-1.elb.amazonaws.com
Description: -

Origin

Elastic Load Balancing origin: my-alb-lab-561280839.ap-south-1.elb.amazonaws.com
Origin path: -
Grant CloudFront access to origin: No
Enable Origin Shield: No
Connection attempts: 3
Connection timeout: 10

Cache settings

CloudFront will apply default cache settings tailored to serving content from a Elastic Load Balancing origin. You can customize settings after you create your distribution.

Security

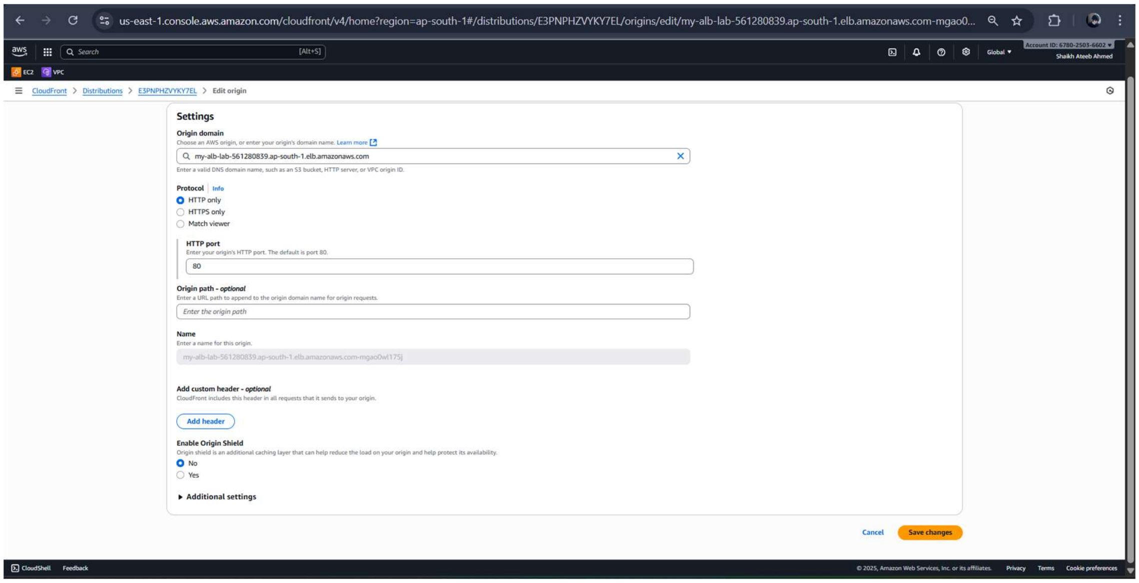
Security protections: None
Use monitor mode: No
Use existing WAF configuration: No

WAF: optional, leave default.

Create Distribution

Origin Protocol Policy mismatch

If your ALB only listens on HTTP (80) but in CloudFront you selected HTTPS only, CloudFront can't connect.
Fix: In CloudFront → Origins → select your origin → set Origin Protocol Policy to HTTP Only (or "Match Viewer" if you added HTTPS on ALB).



us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=ap-south-1#/distributions/E3PNPHZVYKY7EL/origins/edit/my-alb-lab-561280839.ap-south-1.elb.amazonaws.com-mgao0...

CloudFront > Distributions > E3PNPHZVYKY7EL > Edit origin

Settings

Origin domain
Choose an AWS origin, or enter your origin's domain name. Learn more
my-alb-lab-561280839.ap-south-1.elb.amazonaws.com
Enter a valid DNS domain name, such as an S3 bucket, HTTP server, or VPC origin ID.

Protocol
● HTTP only
○ HTTPS only
○ Match viewer

HTTP port
Enter your origin's HTTP port. The default is port 80.
80

Origin path - *optional*
Enter a URL path to append to the origin domain name for origin requests.
Enter the origin path

Name
Enter a name for this origin.
my-alb-lab-561280839.ap-south-1.elb.amazonaws.com-mgao0wl175j

Add custom header - *optional*
CloudFront includes this header in all requests that it sends to your origin.

Enable Origin Shield
Origin shield is an additional caching layer that can help reduce the load on your origin and help protect its availability.
● No
○ Yes

Click **Create Distribution**.

CloudFront status will show **In Progress** (it usually takes ~10–15 minutes to deploy).

Shaikh Ateeb Ahmed
03-10-2025

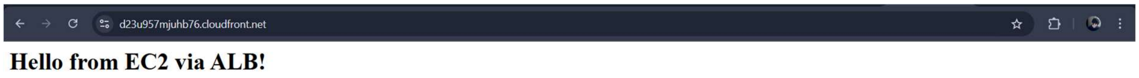
Test CloudFront

Wait until **Status = Deployed**.

Copy the **Domain Name** of the distribution (e.g., d12345abcdef.cloudfront.net).

Open in browser:

https://d12345abcdef.cloudfront.net/



Hello from EC2 via ALB!

You should see your test page:

Hello from EC2 via ALB!