

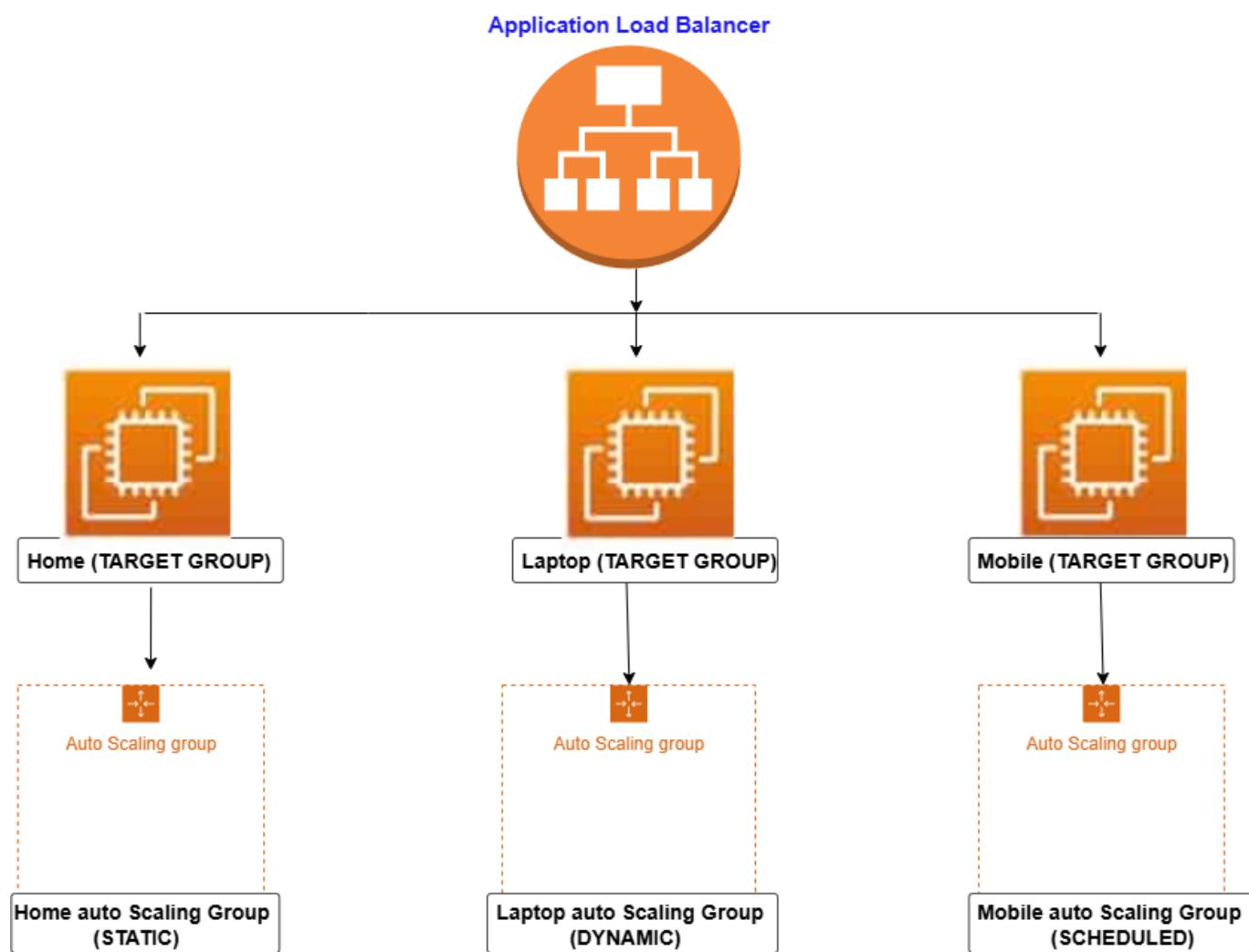
Auto Scaling Group with Application Load Balancer (ALB) in AWS

This guide explains how to set up an Auto Scaling Group (ASG) behind an Application Load Balancer (ALB) in AWS. The ALB will distribute traffic across EC2 instances, and the ASG will automatically adjust the number of instances based on demand.

Architecture Overview

The system is designed to handle different types of user traffic (Home, Laptop, Mobile) via an **Application Load Balancer (ALB)**, which routes requests to appropriate **Target Groups**. Each target group is associated with an **Auto Scaling Group** that manages compute resources based on the nature of traffic.

Architecture



Architecture Overview

- **ALB** routes traffic to 3 target groups: **Home, Laptop, and Mobile**.

- Each target group is connected to an Auto Scaling Group (ASG).
 - Each ASG uses a different type of scaling policy:
 - Home: Static
 - Laptop: Dynamic (based on CPU, etc.)
 - Mobile: Scheduled
-

Prerequisites

- Before you start, ensure you have:
 1. An AWS account with admin access.
 2. AWS CLI installed and configured.
 3. Basic knowledge of EC2, VPC, and Security Groups.
-

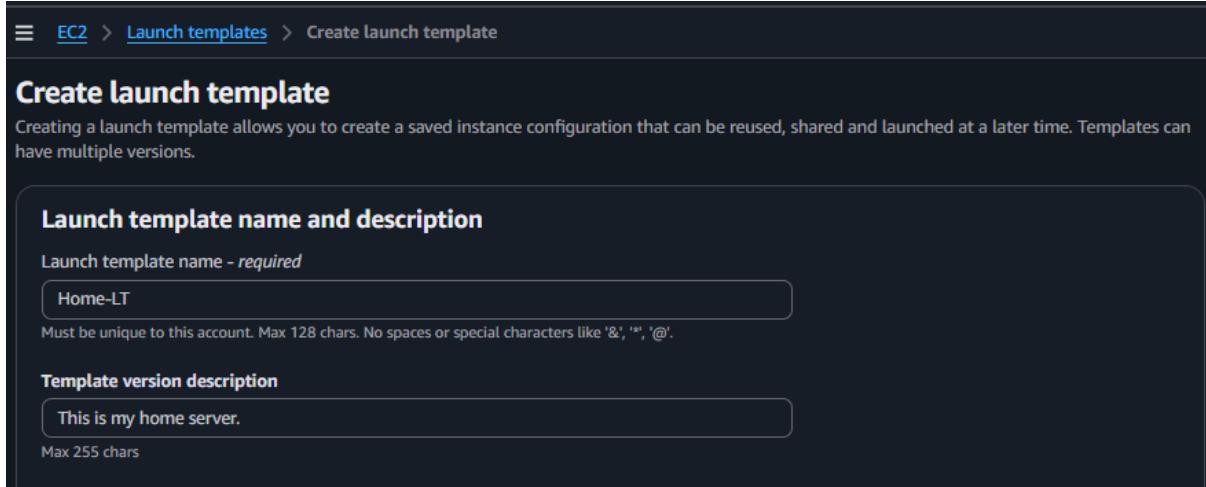
Step-by-Step Implementation Guide

Step 1: Prepare EC2 Launch Template (for all ASGs)

1. Home Launch Template

1. Go to the EC2 Console → Launch Templates → Create launch template
2. Fill in:

- **Template Name:** Home-LT
- **Description :** This is my Home instance



- **AMI ID:** Use Amazon Linux 2 AMI
- **Instance Type:** t3.micro
- **Key Pair:** Select your existing key or create a new one

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

Key pair (login) [Info](#)

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

Key pair name

north-v-key2 [Create new key pair](#)

Security Group: Allow ports 22 (SSH) and 80 (HTTP)

3. In Advanced Details → paste this User Data script:

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "<h1>Hello World from Home instance $(hostname -f)</h1>" >
/var/www/html/index.html
```

EC2 > Launch templates > Create launch template

Metadata response hop limit [Info](#) 2

Allow tags in metadata [Info](#) Don't include in launch template

User data - optional [Info](#) Upload a file with your user data or enter it in the field.

[Choose file](#)

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "<h1>Hello from HOME instance - $(hostname -f)</h1>" > /var/www/html/index.html
```

User data has already been base64 encoded

Summary

Software Image (AMI) Amazon Linux 2023 AMI 2023.8.2... [read more](#) ami-08982fc5bf93d976

Virtual server type (instance type) t3.micro

Firewall (security group) launch-wizard-1

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

[Cancel](#) [Create launch template](#)

4. Leave all other settings as default unless you have specific requirements.

5. Click Create launch template.

2. Laptop Launch Template

- Create new launch template → Name: Laptop-LT
- Use the same instance type, AMI, security group and key pair.
- In Advanced Details, paste this User Data script:

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
mkdir -p /var/www/html/laptop/
echo "<h1>Hello World from Laptop instance $(hostname -f)</h1>" >
/var/www/html/laptop/index.html
```

4. Click Create launch template

3. Mobile Launch Template

1. Create new launch template → Name: Mobile-LT
2. Same setup for AMI, instance type, security group and key pair.
3. In Advanced Details, paste this User Data script:

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
mkdir -p /var/www/html/mobile/
echo "<h1>Hello World from Mobile instance $(hostname -f)</h1>" >
/var/www/html/mobile/index.html
```

4. click Create launch template

- Launch templates are created successfully :

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
lt-06cc9beb75d314721	Mobile-LT	1	1	2025-09-25T17:13:04.000Z	arn:aws:iam::5763
lt-03c5f272af0a0bfe6	Laptop-LT	1	1	2025-09-25T17:18:54.000Z	arn:aws:iam::5763
lt-04f9188f7cafcd668	Home-LT	1	1	2025-09-25T17:03:21.000Z	arn:aws:iam::5763

Step 2: Create Target Groups

1. Go to **EC2 Console** → **Target Groups** → **Create Target Group**.
2. Choose:
 - **Target type:** Instances

- **Protocol:** HTTP
- **Port:** 80

Target group name
Home-TG
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-096cbaa5f93452502 (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.

3. Name each target group:

- **Home-TG**
- **Laptop-TG**
- **Mobile-TG**

4. Register EC2 instances (optional for now — ASGs will do this automatically).

5. Click **Create**.

- **Do this two times — one for each (Laptop, Mobile).**

Target groups (3) Info

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
Mobile-TG	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/MyApp/1234567890123456	80	HTTP	Instance	None associated	vpc-096cbaa5f93452502
Laptop-TG	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/MyApp/1234567890123456	80	HTTP	Instance	None associated	vpc-096cbaa5f93452502
Home-TG	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/MyApp/1234567890123456	80	HTTP	Instance	None associated	vpc-096cbaa5f93452502

0 target groups selected
Select a target group above.

Step 3: Create Auto Scaling Groups

Do this three times with different scaling types:

Home Auto Scaling Group (STATIC)

1. Go to **EC2 Console** → **Auto Scaling Groups** → **Create**.
2. Group Name: **Home-ASG**

3. Attach to Launch Template: Home-LT

Choose launch template Info
Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name
Auto Scaling group name
Enter a name to identify the group.
Home-ASG
Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info
For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
Home-LT
Create a launch template [?]
Version
Default (1) [?] (C)
Create a launch template version [?]
Description
This is my home server.

Launch template
Home-LT [?]
lt-04f9188f7cafcd668

Instance type
t3.micro

Activate Windows

4. Choose VPC and Subnets.

Network Info
For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-096cbaaf93452502
172.31.0.0/16 Default
Create a VPC [?]

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.
Select Availability Zones and subnets
use1-az4 (us-east-1d) | subnet-037a86bd5802d01b0 X
172.31.16.0/20 Default
use1-az5 (us-east-1f) | subnet-0e2178f266b4b3d59 X
172.31.64.0/20 Default
Create a subnet [?]

Availability Zone distribution - new
Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.
 Balanced best effort
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.
 Balanced only
If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

5. Desired, Min, Max capacity: Set all to 2

Configure group size and scaling - optional Info
Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size Info
Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.
Units (number of instances)

Desired capacity
Specify your group size.
2

Scaling Info
You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits
Set limits on how much your desired capacity can be increased or decreased.
Min desired capacity
2 Equal or less than desired capacity
Max desired capacity
2 Equal or greater than desired capacity

6. Skip scaling policies (static)

7. Create ASG.

Laptop Auto Scaling Group (DYNAMIC)

1. Create ASG as above → Name: **Laptop-ASG**

2. Attach to Launch Template: **Laptop-LT**

3. Capacity:

- Min: 2
- Max: 7
- Desired: 3

4. On **Scaling Policies** step:

- Choose **Target tracking scaling policy**
- Metric: Average CPU Utilization
- Target Value: 50%

Automatic scaling - optional
Choose whether to use a target tracking policy | Info
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Scaling policy name
Target Tracking Policy

Metric type | Info
Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.
Average CPU utilization

Target value
50

Instance warmup | Info
300 seconds

Disable scale in to create only a scale-out policy

-

5. Create ASG.

Mobile Auto Scaling Group (SCHEDULED)

1. Create ASG → Name: **Mobile-ASG**

2. Attach to Launch Template: **Mobile-LT**

3. Capacity:

- Min: 2
- Max: 7
- Desired: 3

4. On **Scaling Policies** step:

- Choose **Target tracking scaling policy**
- Metric: Average CPU Utilization
- Target Value: 50%

5. Create ASG.

- Review the ASG

The screenshot shows the AWS Auto Scaling Groups page. At the top, there's a search bar and navigation links for 'Launch configurations', 'Launch templates', 'Actions', and 'Create Auto Scaling group'. Below the header, a table lists three Auto Scaling groups:

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones	Creation time
Mobile-ASG	Mobile-LT Version Default	3	-	3	2	7	2 Availability Zones	Thu Sep 25 2025 23:19:26 GMT+0530 (India Standard Time)
Laptop-ASG	Laptop-LT Version Default	3	-	3	2	7	2 Availability Zones	Thu Sep 25 2025 23:18:25 GMT+0530 (India Standard Time)
Home-ASG	Home-LT Version Default	2	-	2	2	2	2 Availability Zones	Thu Sep 25 2025 23:15:49 GMT+0530 (India Standard Time)

Step 4: Make Mobile-ASG a Scheduled Action

1. Go to Mobile-ASG → scroll down to Scheduled actions.

2. Click Create scheduled action.

The screenshot shows two stacked pages from the AWS console. The top page is titled 'Predictive scaling policies (0) Info' and shows a table with columns: Name, Metric pair, Forecast and scale, Recommendation, Chart, Availability impact, and Cost impact. A message states 'No predictive scaling policies have been created' and provides a 'Create predictive scaling policy' button. The bottom page is titled 'Scheduled actions (0) Info' and shows a table with columns: Name, Start time, End time, Recurrence, Time zone, Desired capa..., Min, and Max. A message states 'No scheduled actions are currently specified' and provides a 'Create scheduled action' button.

3. Enter Name = **BigBillionSale**.

4. Set the Capacity values:

- Min = 5
- Max = 15
- Desired = 8

5. In Recurrence, select **Cron** → enter:

- 0 10 21 10 ***

- (This means: Start at 10:00 AM, 21st October every year).

6. Under End By, set the end date and time:

- Date: 2025/10/31**
- Time: 10:30**

Create scheduled action

Name
Big Billion Sale

Desired capacity
Min: 5, Max: 15
8

Recurrence
Cron: 0 10 21 10 *

Time zone
Etc/UTC

Current time in selected time zone is 2025-09-25/18:05 UTC

Specific start time
Schedule a specific date and time for the first scheduled action to run. Interpreted in recurrence time zone: Etc/UTC
YYYY/MM/DD: 2025/09/25, 00:00, Etc/UTC

End by
2025/10/30, 10:30, Etc/UTC

Actions
Cancel, Create, Go

7. Click **Create**.

Step 5: Attach Target Group to Auto Scaling Group

1. Go to the **Auto Scaling Groups** section in the AWS Console.
2. Select the **Mobile-ASG**.
3. Click on **Actions → Edit**.

Scheduled action created or edited successfully

Auto Scaling groups (1/3) Info Last updated less than a minute ago

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
<input checked="" type="checkbox"/> Mobile-ASG	Mobile-LT Version Default	2	-	2	2	7	2 Availability Zones
<input type="checkbox"/> Laptop-ASG	Laptop-LT Version Default	2	-	2	2	7	2 Availability Zones
<input type="checkbox"/> Home-ASG	Home-LT Version Default	2	-	2	2	2	2 Availability Zones

Actions ▾ Create Auto Scaling group

Edit, Delete

4. In the **Load balancing** section, choose:
 - **Application, Network, or Gateway** Load Balancer target groups (depending on your setup).

5. Add the **Mobile-TG** to the Auto Scaling Group.

Load balancing - optional

Load balancers

Application, Network or Gateway Load Balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups ▾ (C)

Mobile-TG | HTTP X
Load balancer: Not associated with any load balancer

One of your target groups is not yet associated with any load balancer. In order for routing and scaling to occur, you will need to attach the target group to a load balancer. This can be done later in the [Load Balancing console](#). [?]

Classic Load Balancers

Create and attach new load balancers

Add a new load balancer

6. Click **Update** to save changes.

- Repeat the same process for the Home-ASG and Laptop-ASG so that each Auto Scaling Group is attached to its respective Target Group.

Step 6: Create Application Load Balancer (ALB)

1. Go to **EC2 Console** → **Load Balancers** → **Create Load Balancer**.
2. Choose **Application Load Balancer**.

Application Load Balancer	Network Load Balancer	Gateway Load Balancer
<p>Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.</p> <p>Create</p>	<p>Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.</p> <p>Create</p>	<p>Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.</p> <p>Create</p>

3. Provide:

- **Name:** ALB
- **Scheme:** Internet-facing
- **IP address type:** IPv4

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.
 ALB

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.

4. Configure listeners:

- Protocol: HTTP
- Port: 80

5. In Default action, select Forward to Target Groups → choose Home-TG.

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.

▼ Listener HTTP:80

Protocol	Port
HTTP	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	Weight	Percent
Home-TG Target type: Instance, IPv4 Target stickiness: Off	1 0-999	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 affinity setting.

Turn on target group stickiness

6. Select at least two public subnets.

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-096cbaa5f93452502 172.31.0.0/16	(default)	<input type="button"/> Create VPC
--	-----------	-----------------------------------

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

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

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

us-east-1a (use1-az6)

us-east-1b (use1-az1)

us-east-1c (use1-az2)

us-east-1d (use1-az4)

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-037a86bd5802d01b0 IPv4 subnet CIDR: 172.31.16.0/20
--

7. Configure Security Group:

- Allow inbound **HTTP (80)** traffic.

8. Leave **Target Groups** rules for later (do not configure at this step).

9. Click **Create**.

Step 7: Configure ALB Listener Rules

1. Go to the **Listeners** tab of the ALB.

2. Click on the listener (**HTTP: 80**) → **View/Edit rules**.

3. Add a new rule to route traffic:

- **Name/Tag:** mobile-rule
- **Condition:** Select **Path** → set path as /mobile/*

- **Action:** Forward to **Mobile-TG**

The screenshot shows the AWS Lambda function configuration page. Under the 'Actions' section, the 'Forward to target groups' option is selected. Below it, there's a 'Target group' section for 'Mobile-TG' with a weight of 1 and 100% percent. There are options to add more target groups and turn on target group stickiness.

- o Click **Next**
- o Set **Priority** = 1
- o Click **Next → Add rule**

4. Repeat the same process for the **Laptop-TG**:

- o **Name/Tag:** laptop-rule
- o **Condition:** Path = /laptop/*
- o **Action:** Forward to **Laptop-TG**
- o Assign the next available **Priority** (e.g., 2).

5. Save the rules to apply changes.

Step 8: Test the ALB Setup

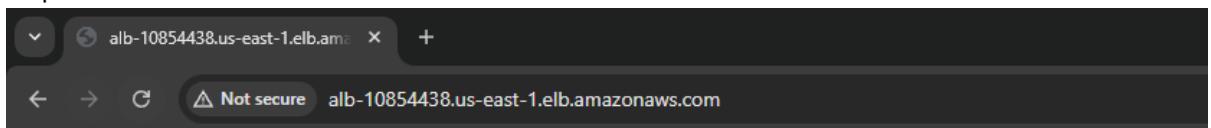
1. Go to the **EC2 Console → Load Balancers**.
2. Select your **Application Load Balancer (ALB)**.
3. Copy the **DNS name** of the ALB (e.g., ALB-123456789.ap-south-1.elb.amazonaws.com).

The screenshot shows the AWS Load Balancers console. It lists one load balancer named 'ALB' which is active and Internet-facing. The details page for 'Load balancer: ALB' shows the following information:

Load balancer type Application	Status Active	VPC vpc-096cbaa5f93452502	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z35SXDOTRQ7X7K	Availability Zones subnet-037a86bd5802d01b0 us-east-1d (use1-az4) subnet-0112b3ff32a16fe68 us-east-1e (use1-az3)	Date created September 25, 2025, 23:50 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:576394208077:loadbalancer/app/ALB/eb25eae71838e4ed		DNS name ALB-619582672.us-east-1.elb.amazonaws.com	Activate Windows Go to Settings to activate Windows

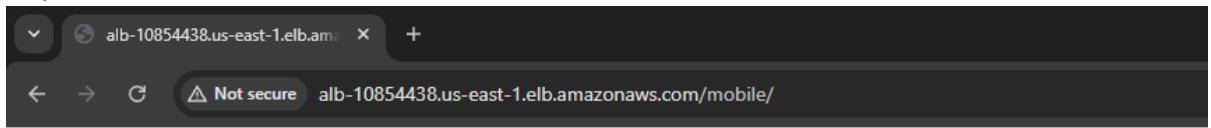
4. Open a browser and test the following paths:

- `http://` → should route to **Home-TG**.



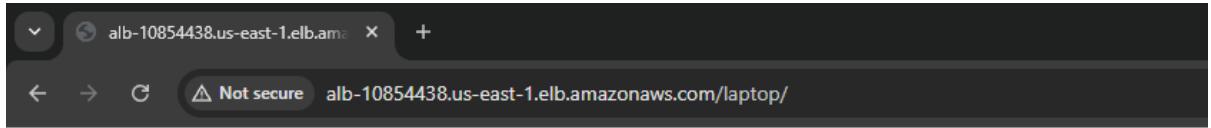
Hello from HOME instance - ip-172-31-92-175.ec2.internal

- `http://mobile/` → should route to **Mobile-TG**.



Hello from MOBILE instance - ip-172-31-84-199.ec2.internal

- `http://laptop/` → should route to **Laptop-TG**.



Hello from LAPTOP instance - ip-172-31-28-49.ec2.internal

Conclusion

In this project, we successfully set up an **Application Load Balancer (ALB)** integrated with **Auto Scaling Groups (ASGs)** and **Target Groups** in AWS.

- We created **separate Target Groups** for Home, Mobile, and Laptop applications.
- Configured **Auto Scaling Groups** to ensure high availability and automatic scaling during peak traffic (e.g., Big Billion Sale).
- Deployed an **Application Load Balancer** with **listener rules** to route traffic based on URL paths (`/mobile/*`, `/laptop/*`, etc.).
- Finally, we tested the configuration using the **ALB DNS name**, confirming that traffic is routed correctly and scaling works as expected.

With this setup:

- The application is **highly available**,
- Can **scale automatically** during demand spikes,
- And provides **efficient traffic distribution** across healthy EC2 instances.

This project demonstrates a real-world implementation of **scalable, load-balanced architectures** in AWS that can be applied to e-commerce platforms, web applications, and enterprise solutions.