

Implementation of Application Load Balancer project

Indroduction

In today's online world, websites and apps need to be available, reliable, and able to handle a lot of users at the same time. A Load Balancer helps make this possible by spreading traffic across different servers. In this project, we will learn how an Application Load Balancer (ALB) works by setting up a real example where it shares incoming traffic between multiple servers like EC2 instances or containers.

This project showcases the deployment of an Application Load Balancer (ALB) on AWS by launching three EC2 instances, each representing a different service: Home, Laptop, and Mobile. Three separate target groups are created and associated with these instances. The ALB is then set up to route incoming traffic to the appropriate target based on predefined rules. Finally, the DNS name of the ALB is obtained and tested to confirm that traffic is being properly distributed among the instances.

Project Highlights

1] Three EC2 Instances: In this project, we launch three EC2 instances, each running an Apache web server. To make each server unique, we use a User Data script that customizes the index.html page with different content for each one — labeled Home, Laptop, and Mobile.

2] Target Groups: We create three separate target groups, assigning each EC2 instance to its own group. This setup ensures that traffic can be routed individually to the Home, Laptop, and Mobile servers.

3] Application Load Balancer (ALB): An Application Load Balancer is then set up and linked to all three target groups. The ALB is configured to distribute incoming traffic based on routing rules. To verify everything is working correctly, we access the DNS name of the ALB in a browser and check that traffic is being directed to the correct server.

Steps of Implementation

Step1: Launch 3 instance Home, Laptop, Mobile.

- Click on "Launch Instances" to start creating servers.
- Name your first Instance "Mobile".
- Choose a key pair so you can connect to the server later.
- Select a security group to allow internet access (like HTTP or SSH).
- Scroll down and click on "Advanced Details".
- Find the "User Data" box and paste your script there (this sets up the server automatically).
- Set the "Number of Instances" to 2 so that two servers will be created at the same time.

The screenshot shows the AWS EC2 'Launch an instance' wizard. On the left, under 'User data - optional', a script is pasted:

```
#!/bin/bash
sudo yum update -y
sudo yum install httpd -y
sudo systemctl start httpd
sudo systemctl enable httpd
echo "<h1>This is a home page ${hostname} </h1>" > /var/www/html/index.html
```

On the right, the 'Summary' section shows the configuration for launching 2 instances. It includes:

- Software Image (AMI)**: Amazon Linux 2023 AMI 2023.8.2... (ami-08982f1c5bf93d976)
- Virtual server type (instance type)**: t3.micro
- Firewall (security group)**: launch-wizard-9
- Storage (volumes)**: 1 volume(s) - 8 GiB

At the bottom right are 'Cancel', 'Launch instance', and 'Preview code' buttons.

- Create a Second instance "Laptop" same as first instance
- Find the "User Data" box and paste your script there as shown below

The screenshot shows the AWS EC2 'Launch an instance' wizard. On the left, under 'User data - optional', a script is pasted:

```
#!/bin/bash
sudo yum update -y
sudo yum install httpd -y
sudo systemctl start httpd
sudo systemctl enable httpd
sudo mkdir /var/www/html/Laptop
echo "<h1>This is a Laptop page ${hostname} </h1>" > /var/www/html/Laptop/index.html
```

On the right, the 'Summary' section shows the configuration for launching 1 instance. It includes:

- Software Image (AMI)**: Amazon Linux 2023 AMI 2023.8.2... (ami-08982f1c5bf93d976)
- Virtual server type (instance type)**: t3.micro
- Firewall (security group)**: launch-wizard-9
- Storage (volumes)**: 1 volume(s) - 8 GiB

At the bottom right are 'Cancel', 'Launch instance', and 'Preview code' buttons.

- Create a Third instance "Mobile" same as first instance
- Find the "User Data" box and paste your Mobile script Same as Laptop
- Now, we Launch the Instances like Home, Laptop & Mobile

The screenshot shows the AWS EC2 Instances page. A green notification bar at the top indicates "Successfully initiated termination (deletion) of i-0e4106ddfbcdcdca3". The main table lists 18 instances. One instance, "Mobile" (i-0604a33172063f1f9), is selected and highlighted in blue. It is currently "Running" with a status check of "3/3 checks passed". Other instances include "home", "Laptop", and various "server" and "laptop" entries. The interface includes filters for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. Buttons for Notifications, Connect, Instance state, Actions, and Launch instances are visible.

This screenshot shows the same AWS EC2 Instances page as the previous one, but with two instances selected: "Mobile" (i-0604a33172063f1f9) and "Laptop" (i-0eb4d2cd30347056). Both are currently "Running". The interface includes monitoring tabs and a CloudWatch agent configuration button. The left sidebar shows the navigation menu for EC2, including Instances, Images, and Elastic Block Store.

Step 2: Create Target Groups

- Create 3 target groups in AWS console. Register each EC2 instance to its respective target group
- Go to the Target Groups

The screenshot shows the AWS EC2 Target groups page. On the left, there is a navigation sidebar with sections for 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 Info" and contains a search bar and a table header with columns: Name, ARN, Port, Protocol, Target type, Load balancer, and VPC ID. A message at the top says "No target groups" and "You don't have any target groups in us-east-1". A blue "Create target group" button is located at the bottom right of the table area.

- Create three target groups

Target groups (3) Info						
<input type="button" value="Actions"/> <input type="button" value="Create target group"/>						
<input type="button" value="Filter target groups"/>						
Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
home-tar	arn:aws:elasticloadbalancin...	80	HTTP	Instance	ALB	vpc-0f20
laptop-tar	arn:aws:elasticloadbalancin...	80	HTTP	Instance	ALB	vpc-0f20
mobile-tar	arn:aws:elasticloadbalancin...	80	HTTP	Instance	ALB	vpc-0f20

Step 3: Create Application Load Balancer

- Click on load balancer

The screenshot shows the AWS Load balancers page. At the top, it says "Compare and select load balancer type". Below that, there is a section titled "Load balancer types" with three cards: "Application Load Balancer", "Network Load Balancer", and "Gateway Load Balancer". Each card has a diagram and a brief description. The "Application Load Balancer" card shows a flow from a client to an ALB, which then routes traffic to targets via HTTP and HTTPS. The "Network Load Balancer" card shows a flow from a client to an ALB, which then routes traffic to NLBs (TCP, UDP, TLS) connected to VPCs. The "Gateway Load Balancer" card shows a flow from a client to a GWLB, which then routes traffic to targets via various protocols like TCP, UDP, and TLS. At the bottom of each card, there is a note: "Choose an Application Load Balancer when you", "Choose a Network Load Balancer when you", and "Choose a Gateway Load Balancer when you".

- Select Application Load Balancer and click on create

Application Load Balancer (ALB)

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.

Network Load Balancer (NLB)

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.

Gateway Load Balancer (GLB)

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.

Classic Load Balancer - previous generation

Create

Create

Create

Close

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

- Put Load Balancer name like ALB

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.

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

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

- Select all AZ's

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-az2)**
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-022c5b89c1b339215
IPv4 subnet CIDR: 172.31.80.0/20
- us-east-1b (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-042354815a750c8d0
IPv4 subnet CIDR: 172.31.16.0/20
- us-east-1c (use1-az6)**
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-05cd0e8e33324b316
IPv4 subnet CIDR: 172.31.32.0/20
- us-east-1d (use1-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-0a7312ab67e516365
IPv4 subnet CIDR: 172.31.0.0/20
- us-east-1e (use1-az3)**
Subnet

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

- Select Existing Security Group which allows port 80 & port 22

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

launch-wizard-9
sg-078882a10d21d386e VPC: vpc-070d03d6e19cf3b8

Listeners and routing | [Info](#)

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

▼ Listener HTTP:80

Protocol	Port	Remove
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 URI Return fixed response

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

- Click on create target groups

The screenshot shows the AWS Management Console interface for creating an Application Load Balancer. The top navigation bar includes the AWS logo, a search bar with the placeholder 'Search', and a keyboard shortcut '[Alt+S]'. Below the navigation, the breadcrumb trail shows 'EC2 > Load balancers > Create Application Load Balancer'. The main content area is titled 'Default action' with an 'Info' link. A note states: 'The default action is used if no other rules apply. Choose the default action for traffic on this listener.' Two radio button options are shown: 'Forward to target groups' (selected) and 'Redirect to URL'. The 'Forward to target group' section includes an 'Info' link and a note: 'Choose a target group and specify routing weight or [create target group](#)'. A 'Target group' section has a dropdown menu labeled 'Select a target group' with a 'C' icon and a '0' count. A blue '+ Add target group' button is available. A note says: 'You can add up to 4 more target groups.' The 'Target group stickiness' section includes an 'Info' link and a note: 'Enables the load balancer to bind a user's session to a specific target group. To use stickiness the client must support co...' A checkbox labeled 'Turn on target group stickiness' is present. The 'Listener tags - optional' section includes an 'Info' link and a note: 'Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them...'. A blue 'Add listener tag' button is available. A note says: 'You can add up to 50 more tags.' At the bottom of the page, there is a horizontal progress bar.

- Specify all group details

Specify group details

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

- Put Health check path for home is /

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

Attributes

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

Tags - optional

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

Cancel Next

- Then select the both home targets for register targets

Available instances (2/6)

Instance ID	Name	State	Security groups	Zone
i-0604a33172063f1f9	Mobile	Running	launch-wizard-9	us-east-1b
i-047f62cad6e9b935c	Mobile	Running	launch-wizard-9	us-east-1b
i-0eeb4d2cd30347056	Laptop	Running	launch-wizard-9	us-east-1b
i-0035a419e117268a9	Laptop	Running	launch-wizard-9	us-east-1b
i-0bc2857add064837d	Home	Running	launch-wizard-9	us-east-1b
i-01df7b436b6e77b73	Home	Running	launch-wizard-9	us-east-1b

Ports for the selected instances
Ports for routing traffic to the selected instances.

- Same target group create for Laptop
- Put health check path for laptop is /Laptop/
- Then select the both Laptop targets for register targets

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: /Laptop/

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

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

- Same target group create for Mobile
- Put health check path for laptop is /Mobile/

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.
/Mobile/

Attributes

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

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

- Then select the both Mobile targets for register targets

Step 1
Specify group details

Step 2
Register targets

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

Available instances (2/6)					
<input type="text"/> Filter instances					
	Instance ID	Name	State	Security groups	Zone
<input checked="" type="checkbox"/>	i-0604a33172063f1f9	Mobile	Running	launch-wizard-9	us-east-1b
<input checked="" type="checkbox"/>	i-047f62cad6e9b935c	Mobile	Running	launch-wizard-9	us-east-1b
<input type="checkbox"/>	i-0eeb4d2cd30347056	Laptop	Running	launch-wizard-9	us-east-1b
<input type="checkbox"/>	i-0035a419e117268a9	Laptop	Running	launch-wizard-9	us-east-1b
<input type="checkbox"/>	i-0bc2857add064837d	Home	Running	launch-wizard-9	us-east-1b
<input type="checkbox"/>	i-01df7b436b6e77b73	Home	Running	launch-wizard-9	us-east-1b

2 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.

- Then Add a Home target group on the Load balancer

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 Target type: Instance, IPv4 Target stickiness: Off	1	100%
Laptop Target type: Instance, IPv4 Target stickiness: Off		
Mobile Target type: Instance, IPv4 Target stickiness: Off		

It must support cookies. If you want to bind a user's session to a specific target, turn on the Target Group attribute 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.

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

- Now, Click on load balancer

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.

[Cancel](#) [Create load balancer](#)

- Sucessfully created a load balancer

The screenshot shows the AWS Cloud Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancer:loadBalancerArn=arn:aws:elasticloadbalancing:us-east-1:824901035886:loadbalancer/app/ALB/140cfdb686eb713d>. The page displays a green success message: "Successfully created load balancer: ALB. It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks." Below this, the "ALB" details are shown in a card:

Details	
Load balancer type	Application
Status	Provisioning
Scheme	Internet-facing
VPC	vpc-070d03d6e19cfe3bb [2]
Availability Zones	subnet-0a7312ab67e516365 [2] us-east-1d (use1-az1) subnet-05cd0e8e33324b316 [2] us-east-1c (use1-az2) subnet-042354815a750c8d0 [2] us-east-1b (use1-az4) subnet-022c5b89c1b339215 [2] us-east-1a (use1-az2) subnet-0aa92490e2e4cb085 [2] us-east-1e (use1-az3) subnet-0ef33c861cf14bcae [2] us-east-1f (use1-az5)
Load balancer IP address type	IPv4
Date created	September 18, 2025, 15:24 (UTC+05:30)

The sidebar on the left includes sections for Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, and Settings.

- Go to the listeners and rules

The screenshot shows the AWS Cloud Console with the same URL as the previous screenshot. The main content area displays the ALB configuration, including the ARN and DNS name (ALB-299581374.us-east-1.elb.amazonaws.com (A Record)). Below this, the "Listeners and rules" tab is selected, showing one listener rule:

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS
HTTP:80	Forward to target group Home [2]: 1 (100%) Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable

The sidebar on the left remains the same as the previous screenshot.

- Click on manage rule and add the rules like Laptop and Mobile rule

The screenshot shows the AWS CloudWatch Metrics interface. A log stream named 'lambda-1' is selected. One log entry is expanded, revealing a JSON payload. The payload contains the following structure:

```

{
    "stage": "prod",
    "path": "/",
    "body": "Hello, world! This is a test message from a Lambda function running in production stage. The timestamp is 2025-09-19T10:00:00Z."
}
  
```

The screenshot shows the AWS Lambda function configuration page. Under the 'Handler' section, the value is set to 'lambda-1.lambda_handler'. In the 'Code' section, the ARN of the Lambda function is displayed.

- Add a laptop-rule first

The screenshot shows the AWS Lambda function configuration page. The top navigation bar includes the AWS logo, search bar, and account information (Account ID: 8249-0103-5886, United States (N. Virginia), Sayali Patil). The main content area is titled 'Code' and contains sections for 'Handler' (lambda.handler), 'Runtime' (Node.js 18.x), 'Memory limit' (128 MB), and 'Timeout' (3 minutes). Below these are tabs for 'Environment', 'Triggers', and 'Logs'. A large 'Edit' button is at the bottom right.

- Select a condition as a path

The screenshot shows the AWS Load Balancer configuration page. The top navigation bar includes the AWS logo, search bar, and account information (Account ID: 8249-0103-5886, United States (N. Virginia), Sayali Patil). The main content area is titled 'Actions' and contains sections for 'Name and tags' (Laptop-rule), 'Conditions (0 values)', and 'Actions'. The 'Actions' section includes options for 'Forward to target groups', 'Redirect to URL', and 'Return fixed response'. A large 'Edit' button is at the bottom right.

- Give Path /Laptop/*

The screenshot shows the AWS Lambda console interface for creating a new rule. The top navigation bar includes the AWS logo, search bar, and account information (Account ID: 8249-0103-5886, United States (N. Virginia), Sayali Patil). The main area is titled "Add rule" under "HTTP:80 listener".

Name: Laptop-rule

Conditions (1 value) Info

Define 1-5 condition values. Additional conditions can't be added once the limit is reached.

Path = /Laptop/* Remove

Path condition value Case sensitive.

= /Laptop/*

Valid characters are a-z, A-Z, 0-9 and special characters. Path must be 1-128 characters.

+ Add OR condition value

Add condition ▼

You can add up to 4 more condition values for this rule.

Actions Info

Requests matching all rule conditions route according to the rule actions.

- Then add Laptop target group

The screenshot shows the continuation of the rule configuration. The "Actions" section is active, with the "Forward to target groups" option selected.

Forward to target group Info

Choose a target group and specify routing weight or [create target group](#).

Target group

Target group	Weight	Percent
Laptop Target type: Instance, IPv4 Target stickiness: Off	HTTP ▲ 1 0-999	100%
Home Target type: Instance, IPv4 Target stickiness: Off	HTTP	
Mobile Target type: Instance, IPv4 Target stickiness: Off	HTTP	

Turn on target group stickiness

Cancel Next

- Set rule priority 1st to the laptop rule

Set rule priority Info

Each rule has a priority. The default rule is evaluated last. You can change the priority of a non-default rule at any time. You can't change the priority of the default rule.

Listener rules (2) Info

Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from the lowest value to the highest value.

Priority	Name tag	Conditions (If)	Actions (Then)	ARN
1	Laptop-rule	Path = /Laptop/*	<ul style="list-style-type: none"> Forward to target group Laptop: 1 (100%) Target group stickiness: Off 	Pending
Last (default)	Default	If no other rule applies	<ul style="list-style-type: none"> Forward to target group Home: 1 (100%) Target group stickiness: Off 	ARN

Cancel **Previous** **Next**

- Now, add the rule Mobile

Add rule Info

Requests that reach this rule are evaluated against its conditions. If a request matches all of the rule conditions, then the request is routed according to the rule's actions.

Listener details: HTTP:80

Name and tags Info

Tags can help you manage, identify, organize, search for and filter resources.

Name **Add additional tags**

Conditions (0 values) Info

Define 1-5 condition values. Additional conditions can't be added once the limit is reached.

Add condition ▼

You can add up to 5 condition values for this rule.

Actions Info

Requests matching all rule conditions route according to the rule actions.

- Select Path like /Mobile/*

General

Origins

Add origin

Mobile

Port: 80 Path prefix:

Default Cache Behavior

Custom Error Responses

Custom Headers

Custom Origins

CloudFront Metrics

CloudFront Metrics

- Give 2nd Priority to Mobile

General

Origins

Add origin

Mobile

Port: 80 Path prefix:

Default Cache Behavior

Custom Error Responses

Custom Headers

Custom Origins

CloudFront Metrics

CloudFront Metrics

- Now, successfully we create a Load balancer named as ALB

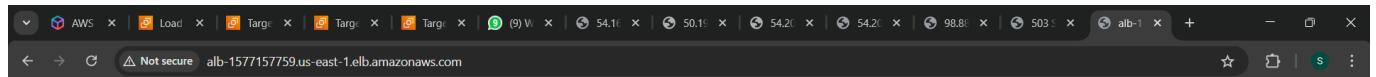
The screenshot shows the AWS EC2 Load Balancers console. On the left, a navigation sidebar includes sections for Elastic Block Store, Network & Security, Load Balancing (selected), Auto Scaling, and CloudWatch Metrics. Under Load Balancing, 'Load Balancers' is selected, showing 'Target Groups' and 'Trust Stores'. The main area displays 'Load balancers (1/2)' with one entry: 'classic-LB' (classic type, Internet-facing, IPv4, VPC ID: vpc-070d03d6e19cfe3b8, 6 Availability Zones). A table below lists 'Load balancer: ALB' with tabs for Details, Listeners and rules (selected), Network mapping, Resource map, Security, Monitoring, Integrations, Attributes, and Capacity. The 'Listeners and rules' tab shows 'Listeners and rules (1) Info' with a table for managing listeners.

- Copy the DNS name of load balancer and hit on browser

The screenshot shows the details of the 'ALB' load balancer. The left sidebar is identical to the previous screenshot. The main area displays the 'ALB' details page. Under the 'Details' section, it shows the Load balancer type as 'Application', Status as 'Active', Scheme as 'Internet-facing', Hosted zone as 'Z35SXDOTRQ7X7K', and VPC as 'vpc-070d03d6e19cfe3b8'. The 'Availability Zones' section lists multiple subnets across three AZs: us-east-1a, us-east-1b, and us-east-1c. The 'Load balancer ARN' is listed as 'arn:aws:elasticloadbalancing:us-east-1:824901035886:loadbalancer/app/ALB/140cfdb686eb713d' and the 'DNS name' is listed as 'ALB-299581374.us-east-1.elb.amazonaws.com (A Record)'.

- Final output

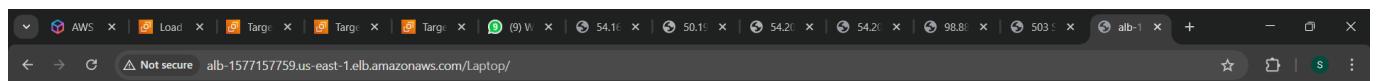
This is Output of server- Home Instance



This is a home page ip-172-31-16-203.ec2.internal



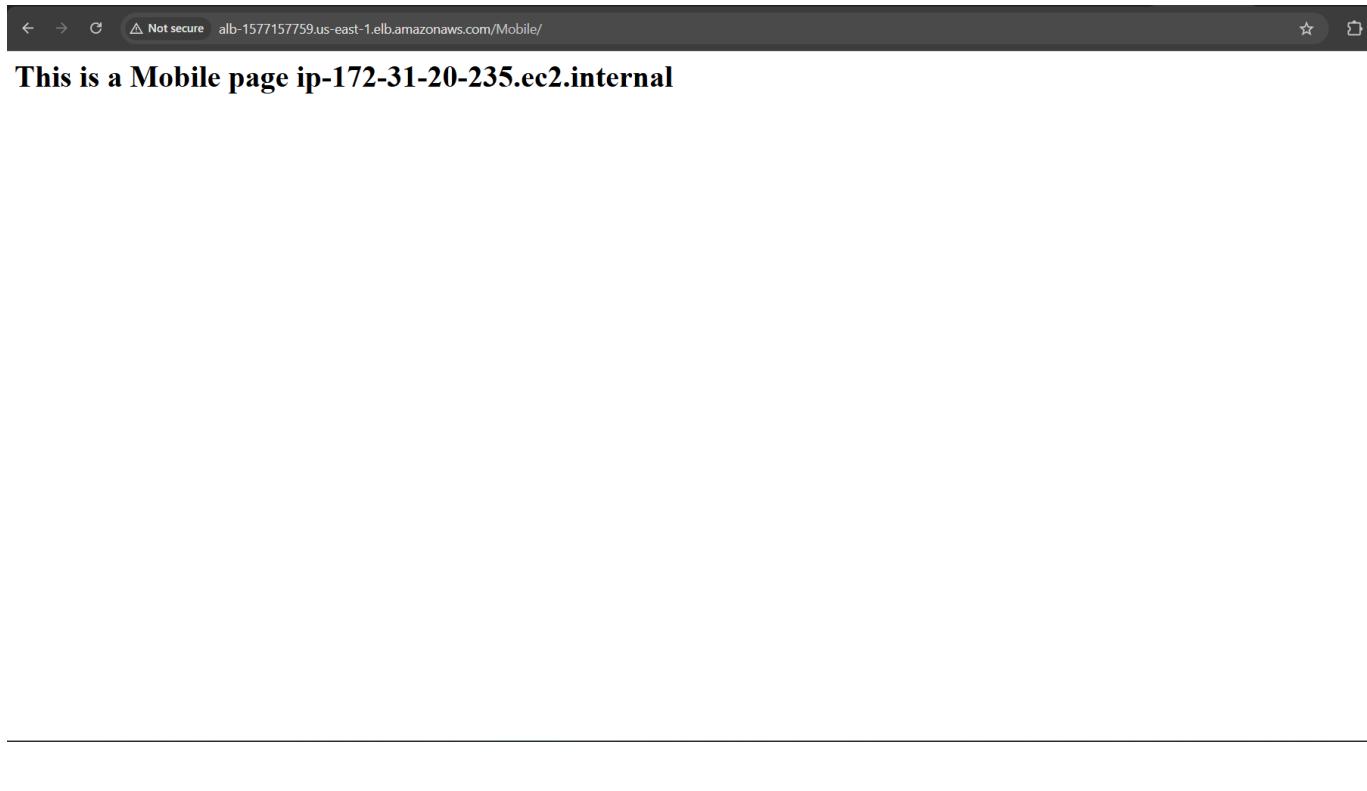
This is Output of server- Laptop Instance



This is a Laptop page ip-172-31-26-141.ec2.internal



This is Output of server- Mobile Instance



Summary

In this project, we set up an Application Load Balancer (ALB) on AWS to share traffic between three EC2 instances: Home, Laptop, and Mobile. Each instance ran a script when it started to install Apache and show a custom webpage. We created three target groups, and each instance was added to its own group. Then, we created an Application Load Balancer with a listener on port 80 (HTTP), which sends traffic to these target groups. Finally, we tested the ALB's DNS link in a web browser. The page showed different responses from Home, Laptop, and Mobile, proving that the load balancer was working correctly.