

AWS Application Load Balancer (Mini Project)

Overview

This project walks through deploying an **Application Load Balancer (ALB)** in AWS to evenly distribute traffic across three EC2 instances.

Each instance (named **Home**, **Laptop**, and **Mobile**) hosts a custom Apache web page. The ALB is then configured to route incoming requests and verify distribution by refreshing the DNS link.

Architecture Summary

- 1. **EC2 Instances**
 - Launch **3 Amazon Linux t2.micro instances** (Free Tier).
 - Use User Data scripts to install Apache (httpd) and serve unique pages.
- 2. **Target Groups**
 - Create three target groups (one per instance).
 - Register each instance to its respective target group.
- 3. **Application Load Balancer**
 - Create an ALB and attach the target groups.
 - Configure **HTTP:80 listener** to forward traffic.

Implementation Steps

Step 1: Launch EC2 Instances

Instances (6/8) [Info](#)

🔄

Connect

Instance state ▾

Actions ▾

Launch instances ▾

🔍 Find Instance by attribute or tag (case-sensitive)

All states ▾

<input type="checkbox"/>	Name 🔗 ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾	Public IP
<input checked="" type="checkbox"/>	MOBILE 1	i-0c0abad2396f6e011	🟢 Running 🔍 🔍	t3.micro	🟢 3/3 checks passed	View alarms +	eu-north-1b	ec2-13-61
<input checked="" type="checkbox"/>	MOBILE 2	i-0bc02da91f84c721f	🟢 Running 🔍 🔍	t3.micro	🟢 3/3 checks passed	View alarms +	eu-north-1b	ec2-13-61
<input checked="" type="checkbox"/>	HOME 1	i-091dc036f99f7502b	🟢 Running 🔍 🔍	t3.micro	🟢 3/3 checks passed	View alarms +	eu-north-1b	ec2-13-61
<input checked="" type="checkbox"/>	HOME 2	i-0d8199a799198c008	🟢 Running 🔍 🔍	t3.micro	🟢 3/3 checks passed	View alarms +	eu-north-1b	ec2-13-61
<input checked="" type="checkbox"/>	LAPTOP 1	i-023f1438fadc61f3a	🟢 Running 🔍 🔍	t3.micro	🟢 3/3 checks passed	View alarms +	eu-north-1b	ec2-13-61
<input checked="" type="checkbox"/>	LAPTOP 2	i-001a49a6074fff6a8	🟢 Running 🔍 🔍	t3.micro	🟢 3/3 checks passed	View alarms +	eu-north-1b	ec2-13-61

6 instances selected

For each instance, use a **User Data script** to install Apache and serve a custom page.

Home Instance

```
#!/bin/bash
yum update -y
yum install -y httpd
```

```
systemctl start httpd
systemctl enable httpd
echo "<h1> Welcome to the Home Page - $(hostname -f)</h1>" >
/var/www/html/index.html
```

Laptop Instance

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

Mobile Instance

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

Step 2: Configure Target Groups

- In the **AWS Console** → **Target Groups**, create 3 target groups.
- Register each instance under its group (Home → TG-Home, Laptop → TG-Laptop, Mobile → TG-Mobile).

Target groups (3) [Info](#) [Refresh](#) [Actions](#) [Create target group](#)

Filter target groups

<input type="checkbox"/>	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
<input type="checkbox"/>	laptop-tg	arn:aws:elasticloadbalancin...	80	HTTP	Instance	None associated	vpc-0d2:
<input type="checkbox"/>	mobile-tg	arn:aws:elasticloadbalancin...	80	HTTP	Instance	None associated	vpc-0d2:
<input type="checkbox"/>	home-tg	arn:aws:elasticloadbalancin...	80	HTTP	Instance	None associated	vpc-0d2:

Step 3: Create Application Load Balancer


- Go to **EC2** → **Load Balancers** → **Create ALB**.
- Add the **three target groups**.


- Configure **Listener on Port 80 (HTTP)** to forward requests to the groups.

Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.


Filter load balancers

< 1 > 

<input checked="" type="checkbox"/>	Name	State	Type	Scheme	IP address type	VPC ID	Availability Zones
<input checked="" type="checkbox"/>	ALB	 Provisioning..	application	Internet-facing	IPv4	vpc-0d2320dc882d8c89b	3 Availability Zones

Step 4: Test the Load Balancer

- Copy the **ALB DNS name** from the console.
- Paste it into a browser and refresh multiple times.
- You should alternately see:
 - Home instance response
 - Laptop instance response
 - Mobile instance response

DNS name [Info](#) ALB-1521863408.eu-north-1.elb.amazonaws.com (A Record)

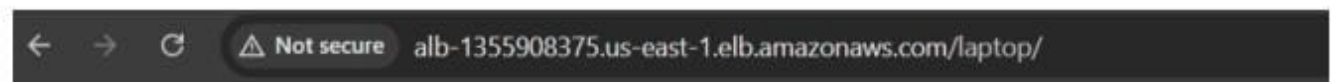
Expected Output

- **Home Page:** This is the Home Instance



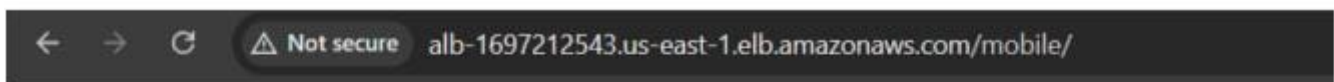
this is home page ip-172-31-21-216.ec2.internal

- **Laptop Page:** This is the Laptop Instance



this is laptop page ip-172-31-27-238.ec2.internal

- **Mobile Page:** This is the Mobile Instance



this is mobile page ip-172-31-24-206.ec2.internal

Conclusion

We successfully set up an **Application Load Balancer** in AWS that distributes requests across three EC2 servers. Each instance serves a custom page, and the ALB ensures traffic is shared evenly, verifying the load balancing feature.
