

# AWS Application Load Balancer (ALB) LAB

## Complete Steps: Setup ALB with EC2 in AWS

### Pre-requisites:

- Two running EC2 instances in **different Availability Zones**
- Apache/Nginx or any web server installed
- EC2 instances should be in the **same VPC**

## Step-by-Step Guide

### STEP 1: Install Web Server on EC2 Instances

- Two running EC2 instances in **different Availability Zones**

Instances (2) <a href="#">Info</a>										
<a href="#">Find Instance by attribute or tag (case-sensitive)</a> <span>Last updated 3 minutes ago</span> <span> Connect</span> <span>Instance state ▾</span> <span>Actions ▾</span> <span>Launch instances</span> <span>▼</span>										
	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾	Public IPv4 DNS ▾	Private IP ▾	Port ▾
<input type="checkbox"/>	linux-server-1b	i-06bc2f5257b23ba8a	 Running  	t2.micro	 2/2 checks passed	<a href="#">View alarms +</a>	ap-south-1b	ec2-3-108-235-57.ap-s...	3.	
<input type="checkbox"/>	linux-server-1a	i-0adf145b8d5c7fcf7	 Running  	t2.micro	 2/2 checks passed	<a href="#">View alarms +</a>	ap-south-1a	ec2-13-126-112-36.ap...	1:	

Connect to each EC2 instance via SSH and run:

```
#!/bin/bash
dnf install httpd -y
systemctl enable httpd --now
echo "<h1>This is my web Web Server - 1</h1>" >> /var/www/html/index.html
echo "<h2>From ap-south-1a </h2>" >> /var/www/html/index.html
```

Do the same on the second instance, but change the message:

```
#!/bin/bash
dnf install httpd -y
systemctl enable httpd --now
echo "<h1>This is my web Web Server - 2</h1>" >> /var/www/html/index.html
echo "<h2>From ap-south-1b</h2>" >> /var/www/html/index.html
```

Confirm both instances return their message via browser or curl:

<http://<public-ip-of-ec2-1a>> eg :<http://39.98.44.1>

**This is my web Web Server - 1**

**From ap-south-1a**

Linkedin : <https://www.linkedin.com/in/uppalavenkatasai/>

Github : <https://github.com/Uppalavenkatasai>

<http://<public-ip-of-ec2-1b>>

## This is my web Web Server - 2

From ap-south-1b

### STEP 2: Create a Security Group for the ALB

1. Go to **VPC > Security Groups > Create Security Group**
2. **Name:** ALB-SG
3. **Inbound Rules:**
  - o Type: **HTTP**
  - o Protocol: **TCP**
  - o Port: **80**
  - o Source: **0.0.0.0/0** (allow from all)
4. **Outbound Rules:** Leave as default (all traffic allowed)

The screenshot shows the AWS VPC Security Groups console. At the top, it displays the security group name "sg-Off8022a87768d05c - ALB-SG" and an "Actions" dropdown. Below this is a "Details" section with fields for Security group name (ALB-SG), Security group ID (sg-Off8022a87768d05c), Description (ALB SG for load balancer), VPC ID (vpc-0f14e8837e8e713bc), Owner (398211280527), Inbound rules count (2 Permission entries), and Outbound rules count (1 Permission entry). Below the details is a navigation bar with tabs: Inbound rules (selected), Outbound rules, Sharing - new, VPC associations - new, and Tags. The "Inbound rules (2)" table lists two entries:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0054201bec8ba4741	IPv4	HTTP	TCP	80	0.0.0.0/0
-	sgr-0127a60d30c4f02aa	IPv6	HTTP	TCP	80	::/0

### STEP 3: Create Target Group

1. Go to **EC2 > Target Groups**
2. Click **Create Target Group**
  - o **Target type:** Instances
  - o **Protocol:** HTTP
  - o **Port:** 80
  - o **VPC:** Choose the correct one
  - o **Name:** my-target-group
3. Register your **2 EC2 instances** to this target group
4. Click **Create**

## Aviz Academy

### Linux-ELB-TG

Actions ▾

#### Details

[arn:aws:elasticloadbalancing:ap-south-1:398211280527:targetgroup/Linux-ELB-TG/1ad41a9888488bd3](#)

Target type  
Instance

Protocol : Port  
HTTP: 80

Protocol version  
HTTP1

VPC  
[vpc-0f14e8837e8e713bc](#)

IP address type  
IPv4

Load balancer  
[L-APP-LB](#)

2  
Total targets

2  
Healthy

0  
Unhealthy

0  
Unused

0  
Initial

0  
Draining

0 Anomalous

#### Distribution of targets by Availability Zone (AZ)

Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets | Monitoring | Health checks | Attributes | Tags

#### Registered targets (2) [Info](#)

Anomaly mitigation: Not applicable



Deregister

Register targets

Registered targets (2) <a href="#">Info</a>									
<span style="float: right;">Anomaly mitigation: Not applicable</span> <span style="float: right;"></span> <span style="float: right;">Deregister</span> <span style="float: right;">Register targets</span>									
<span style="float: left;">Filter targets</span> <span style="float: right;">1</span>									
Instance ID	Name	Port	Zone	Health status	Health status details	Administ...	Overrid...	Launch...	Anomaly detectio...
<a href="#">i-06bc2f5257b23ba8a</a>	linux-server-1b	80	ap-south-1b (a...)	Healthy	-	<input type="radio"/> No override.	<input type="radio"/> No overrid...	September...	Normal
<a href="#">i-0adf145b8d5c7fcf7</a>	linux-server-1a	80	ap-south-1a (a...)	Healthy	-	<input type="radio"/> No override.	<input type="radio"/> No overrid...	September...	Normal

## STEP 4: Create Application Load Balancer (ALB)

1. Go to **EC2 > Load Balancers > Create Load Balancer**
2. Select **Application Load Balancer**
3. **Name:** my-app-lb
4. **Scheme:** Internet-facing
5. **Listeners:** HTTP, Port 80
6. **Availability Zones:**
  - o Select 2 subnets in different AZs (same as EC2s)
7. **Security Groups:**
  - o Select **ALB-SG** created earlier
8. **Target Group:**
  - o Choose **existing target group** → my-target-group
9. Click **Create Load Balancer**

### L-APP-LB

Actions ▾

#### ▼ Details

Load balancer type  
Application

Status  
 Active

VPC  
[vpc-0f14e8837e8e713bc](#)

Load balancer IP address type  
IPv4

Scheme  
Internet-facing

Hosted zone  
ZP97RAFLXTNZK

Availability Zones  
[subnet-0ec8a554198204dd2](#) ap-south-1b (aps1-az3)  
[subnet-0c6c345f38458c250](#) ap-south-1a (aps1-az1)

Date created  
September 11, 2025, 12:49 (UTC+05:30)

#### Load balancer ARN

[arn:aws:elasticloadbalancing:ap-south-1:398211280527:loadbalancer/app/L-APP-LB/b4b138fef0c8103b](#)

#### DNS name [Info](#)

[L-APP-LB-812647431.ap-south-1.elb.amazonaws.com](#) (A Record)

Listeners and rules

Network mapping

Resource map

Security

Monitoring

Integrations

Attributes

Capacity

Tags

#### Listeners and rules (1) [Info](#)



Manage rules ▾

Manage listener ▾

Add listener

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.

[Filter listeners](#)

< 1 >

Protocol:Port

Default action

Rules

ARN

Security policy

mTLS

Trust store

[HTTP:80](#)

Forward to target group  
[Linux-ELB-TG](#) 1 (100%)  
Target group stickiness: Off

1 rule

[ARN](#)

Not applicable

Not applicable

Not applicable

Not applicable

LinkedIn : <https://www.linkedin.com/in/uppalavenkatasai/>

Github : <https://github.com/UppalavenkataSai>

## ■ STEP 5: Adjust EC2 Security Group for Better Security (Optional Best Practice)

Currently, your EC2 instances probably allow HTTP from all (0.0.0.0/0).

Let's **lock it down** so only the ALB can access EC2s:

1. Go to **EC2 > Security Groups**
2. Select your EC2's group (e.g., launch-wizard-2)
3. **Edit Inbound Rules**
  - Remove:
    - HTTP from 0.0.0.0/0
  - Add:
    - **Type:** HTTP
    - **Source:** Choose ALB-SG (from security groups list)

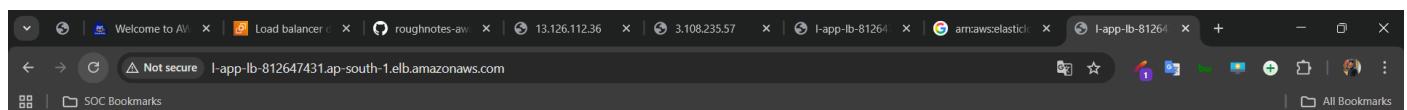
Now only your ALB can send traffic to EC2 instances.

---

## ■ STEP 6: Test Your ALB

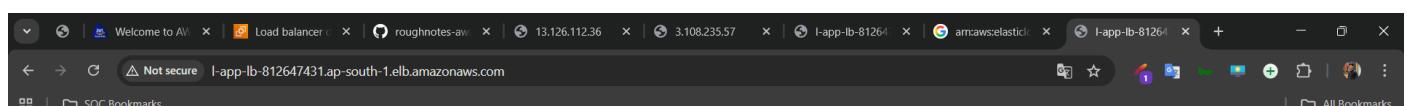
1. Go to **EC2 > Load Balancers**
2. Copy the **DNS name** of your ALB (e.g.):  
3. `http://L-APP-LB-812648431.ap-south-1.elb.amazonaws.com`
4. Open in a browser or run:  
5. `curl http://<your-alb-dns>`

 Refresh multiple times — you should see:



**This is my web Web Server - 1**

**From ap-south-1a**



**This is my web Web Server - 2**

**From ap-south-1b**

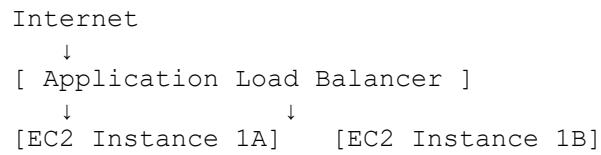
Which confirms **load balancing is working**.

---

Linkedin : <https://www.linkedin.com/in/uppalavenkatasai/>

Github : <https://github.com/UppalavenkataSai>

## 🎯 Final Architecture Diagram (Simplified)



- ALB-SG allows HTTP from the Internet
- EC2-SG allows HTTP only from ALB-SG