

AWS Load Balancer

Overview

AWS Load Balancer is a fully managed service that distributes incoming application traffic across multiple targets, such as EC2 instances, in different Availability Zones. It ensures high availability, fault tolerance, and scalability.

Why Use a Load Balancer?

- Distributes traffic efficiently to prevent overloading a single resource.
- Enhances application availability by routing traffic to healthy instances.
- Provides automatic failover in case of server failure.
- Supports security features like TLS termination and integrated authentication.
- Enables flexible routing and advanced request handling with listener rules.

Connection Between AWS Load Balancer and OSI Model

- **Layer 4 (Transport Layer):** Network Load Balancer (NLB) operates at this layer, handling TCP/UDP traffic.
- **Layer 7 (Application Layer):** Application Load Balancer (ALB) operates here, managing HTTP/HTTPS requests.
- **Layer 3 (Network Layer):** Gateway Load Balancer (GLB) processes network packets between appliances.
- **Layer 2 (Data Link Layer):** Classic Load Balancer (CLB) provides basic load balancing capabilities.
- Load balancers interact with lower OSI layers via security groups and networking configurations.

Types of Load Balancers

1) Application Load Balancer (ALB)

- Clients send requests to your application.
- Listeners match requests based on protocol and port configuration.
- Incoming requests are evaluated against rules and routed to the appropriate target group.
- HTTPS listeners can offload TLS encryption and decryption.
- Healthy targets receive traffic based on routing rules and load balancing algorithms.

2) Network Load Balancer (NLB)

- Clients send requests to your application.
- The load balancer receives requests directly or via AWS PrivateLink.
- Listeners match protocol and port, routing requests based on the default action.
- TLS listeners offload encryption and decryption.
- Traffic is distributed to healthy targets using the flow hash algorithm.

3) Gateway Load Balancer (GLB)

- Routes requests based on VPC route table, Internet Gateway, or Transit Gateway.
- Directs traffic to a target group of virtual appliances (e.g., firewalls, deep packet inspection systems).
- Virtual appliances process and forward or drop traffic based on configurations.
- Functions as a bump-in-the-wire for inline traffic inspection.

4) Classic Load Balancer (CLB)

- Routes requests based on VPC route table, Internet Gateway, or Transit Gateway.
- Uses a fleet of appliances for traffic filtering and deep packet inspection.
- Processes traffic flows before forwarding or dropping traffic.
- Functions as a basic load balancer for legacy applications.

Load Balancer Schemes

- **Internet Facing:** Exposes a public IP to route traffic from the internet.
- **Internal:** Routes traffic only within a private VPC, restricting external access.

Target Group in Load Balancers

A Target Group defines a group of registered targets (such as EC2 instances) that receive traffic from the load balancer. It supports different protocols and health checks to determine target availability.

Project :- AWS Load Balancer Path-Based Routing Project For EC2 Hosted Web Applications.

Step 1: Create Two EC2 Instances in Different Availability Zones

We are launching two EC2 instances in the same AWS region but in separate Availability Zones (AZs). This setup improves high availability and fault tolerance.

Instance 1: "Test EC2 1a"

- **AMI Selection:** Choose an appropriate Amazon Machine Image (AMI) based on the required operating system.
- **Key Pair (KP):** Select a key pair for secure SSH access.
- **Networking:**
 - Place the instance in a specific subnet belonging to **Availability Zone ap-south-1a** (e.g., `subnet-0603aa0dd1018f970`).
 - Use the **security group** (`ec2sg`) to allow necessary inbound/outbound traffic.
- **User Data:** The following script executes automatically upon instance launch:

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

- **Launch the instance** named "Test EC2 1a".

Instance 2: "Test EC2 1b"

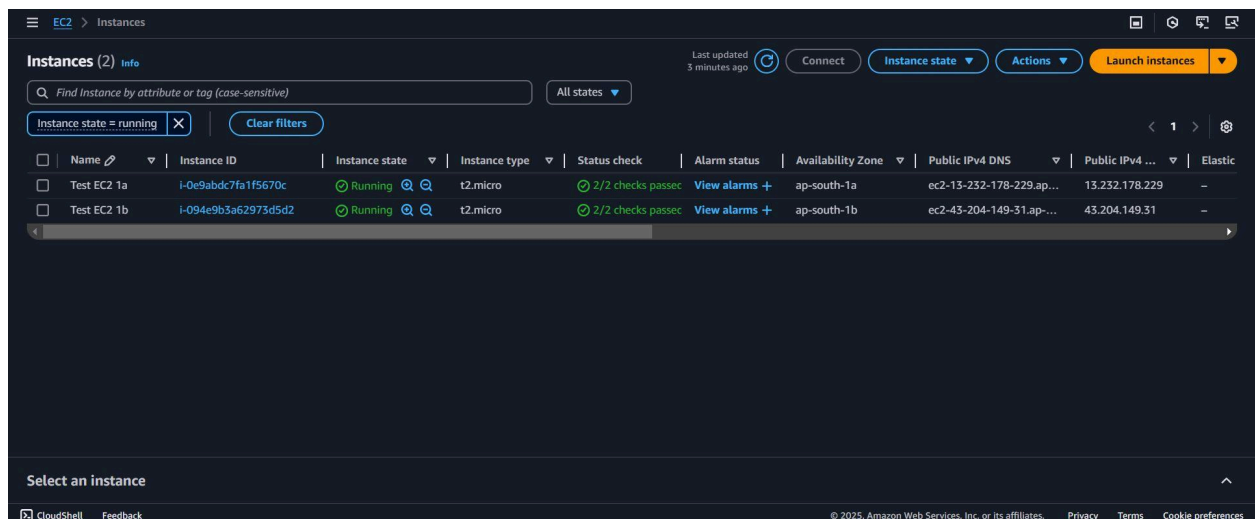
- **Follow the same steps as "Test EC2 1a"** but change:
 - **Subnet:** `subnet-006d8c52d39e28865` (Availability Zone: `ap-south-1b`).
- **Launch the instance** named "Test EC2 1b".

Step 2: Launch the Instance Called "ap-south-1b"

Follow the same procedure while ensuring the instance is created in the correct **subnet** and **AZ**.

Step 3: Verify Instances Are Running

1. Navigate to the **EC2 Dashboard** in the AWS Management Console.
2. Check the status of the instances to ensure they are in a "running" state.



Step 4: Test Web Server Accessibility

1. Copy the **public IP address** of each instance.
2. Open a web browser and paste the IP address into the address bar.
3. You should see a webpage displaying:

Hello World from <hostname>



Step 2: Create an Application Load Balancer

1. Navigate to **Load Balancers** in the **EC2 Dashboard**.
2. Click **Create Load Balancer** and select **Application Load Balancer**.
3. Configure the Load Balancer:
 - **Name:** DemoALB
 - **Scheme:** Internet Facing
 - **IP Address Type:** IPv4
 - **Network Mapping:** Keep **VPC Default** and select **all Availability Zones and subnets**.
 - **Security Group:** Select `launch-wizard-1` (we will configure it later).

Listeners and Routing

Listeners: A listener checks for connection requests using the specified protocol and port. Here, we set:

- **Protocol:** HTTP
- **Port:** 80 (A port is a communication endpoint that allows data transfer between devices.)
- **Default Action:** Forward requests to a **Target Group**.

What is a Target Group in Load Balancers?

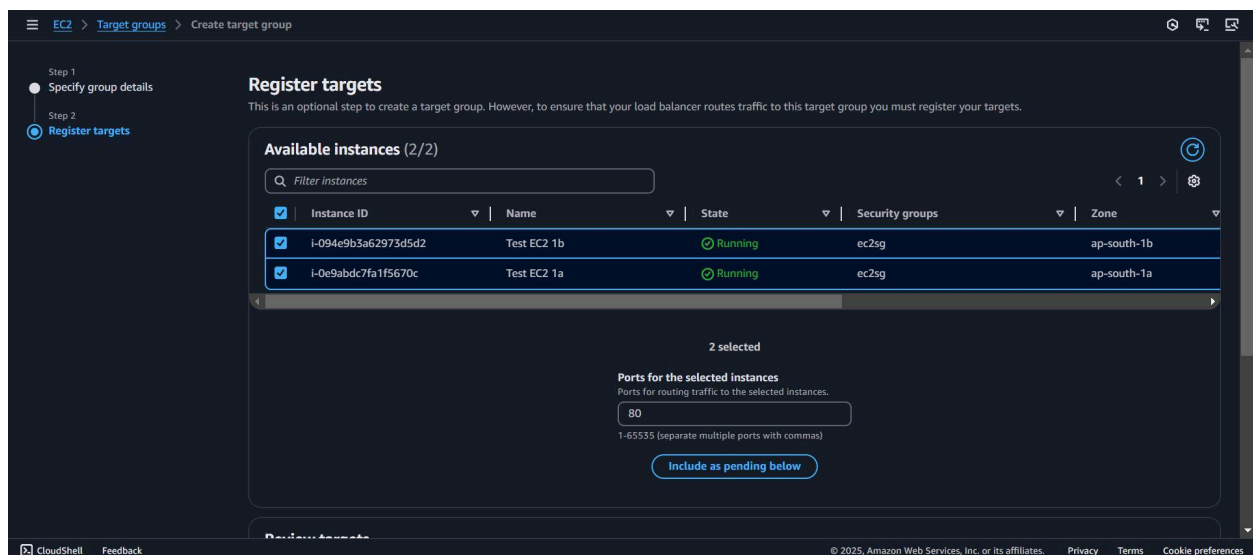
A **Target Group** is a logical group of registered targets (EC2 instances) that receive traffic from the Load Balancer. It defines how traffic is directed based on routing rules.

1. Click **Create Target Group**.
2. **Target Group Configuration:**
 - **Target Type:** **Instances**
 - **Name:** **TG1**
 - **Protocol:** HTTP, **Port:** 80
 - **VPC:** Select the appropriate VPC
 - **Protocol Version:** HTTP1

What are Health Checks?

Health checks monitor the status of registered targets to ensure traffic is only sent to healthy instances. It includes:

- **Health Check Protocol:** HTTP
 - **Health Check Path:** /
3. Click **Next**.
 4. **Register Targets:**
 - Select the two previously created EC2 instances.
 - **Ports for the selected instances:** These define the communication channel for routing traffic to the instances.
 - Click **Include as pending below**.



5. Click **Create Target Group**.

Step 3: Assign Target Group to Load Balancer

1. Go to **Listeners and Routing** in the Load Balancer settings.
2. Refresh the page and **add TG1 as the target group for HTTP**.

EC2 > Load balancers > Create Application Load Balancer

sg-027825e43d3b55fa3 VPC: vpc-0360d11019ee41ea0

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

Port

Default action

Info

HTTP

80

Forward to

TG1

Target type: Instance, IPv4

HTTP

Create target group

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

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.

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

3. Review the **Summary** to check all Load Balancer configurations.

EC2 > Load balancers > Create Application Load Balancer

Review

Review the load balancer configurations and make changes if needed. After you finish reviewing the configurations, choose Create load balancer.

Summary

Review and confirm your configurations. [Estimate cost](#)

Basic configuration

Edit

Name: DemoALB

Scheme: Internet-facing

IP address type: IPv4

Network mapping

Edit

VPC: [vpc-0360d11019ee41ea0](#)

Public IPv4 IPAM pool: -

Availability Zones and subnets:

- ap-south-1a
 - [subnet-0603aa0dd1018f970](#)
- ap-south-1b
 - [subnet-006d8c52d39e28865](#)
- ap-south-1c
 - [subnet-0be0ce07400f57397](#)

Security groups

Edit

launch-wizard-1

[sg-027825e43d3b55fa3](#)

Listeners and routing

Edit

HTTP:80 | Target group: [TG1](#)

Service integrations

Edit

Amazon CloudFront + AWS Web Application Firewall (WAF): -

AWS WAF: -

AWS Global Accelerator: -

Tags

Edit

-

Attributes

1

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

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

4. Click **Create Load Balancer**.

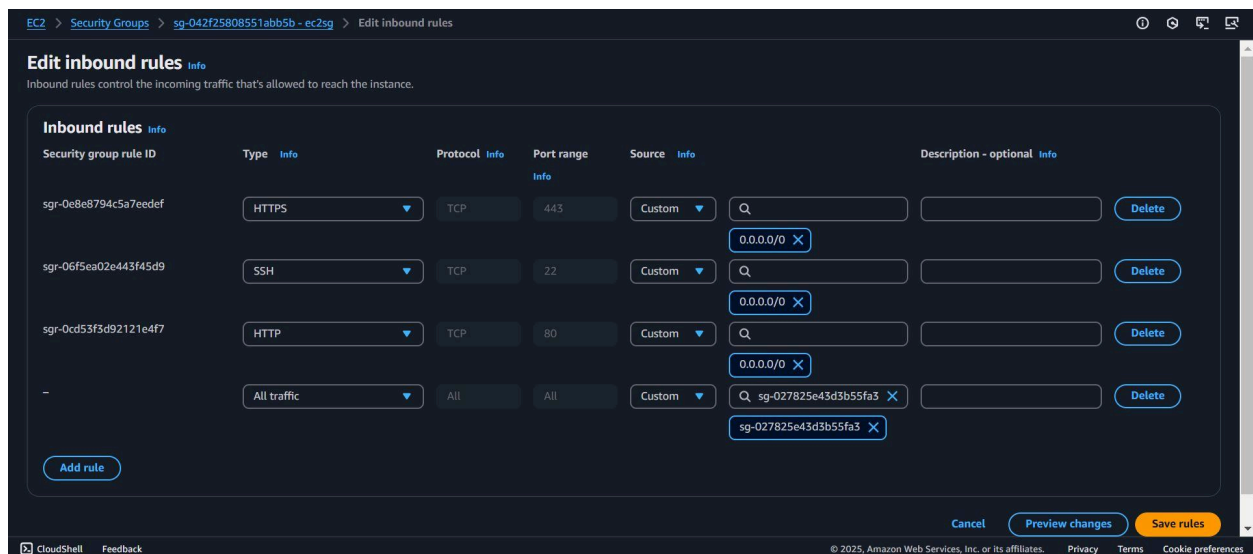
Step 4: Security Group Configuration

Now we have two **Security Groups (SGs)**:

1. **Load Balancer Security Group**: Controls incoming traffic to the Load Balancer.
2. **Instance Security Group**: Controls traffic to EC2 instances.

To ensure secure communication, configure the **instance's security group** to allow traffic only from the **Load Balancer's security group**:

1. Navigate to **EC2 Dashboard**.
2. Select **Security Groups** and choose **ec2sg**.
3. Click **Edit Inbound Rules**.
4. **Modify the inbound rules**:
 - **Add Rule**:
 - **Type**: **All Traffic**
 - **Source**: **Custom**
 - **Select Security Group of Load Balancer (**launch-wizard-1**)**.
 -



5. **Delete previous three inbound rules** (Reason: Removing public access ensures that instances only receive traffic from the Load Balancer, enhancing security).

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID

Type Info

Protocol Info

Port range Info

Source Info

Description - optional Info

All traffic

All

All

Custom

Q

sg-027825e43d3b55fa3 X

Delete

Add rule

Cancel

Preview changes

Save rules

CloudShell

Feedback

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

6. Click **Save Rules**.

Step 5: Verify Target Group and Load Balancer Status

1. Check Target Group Details:

- Navigate to **Target Groups**.
- Verify that both instances appear as **Healthy**.

EC2 > Target groups > TG1

EC2

Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups

IP address type

Load balancer

IPv4

DemoALB

2

0 Anomalous

2 Healthy

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.

Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (2) Info

Anomaly mitigation: Not applicable

Deregister

Register targets

Filter targets

1

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details	Admini...	Overri...	Launch...
<input type="checkbox"/>	i-094e9b3a62973d5d2	Test EC2 1b	80	ap-south-1b (a...	Healthy	-	No override.	No overri...	March 26,...
<input type="checkbox"/>	i-0e9abdc7fa1f5670c	Test EC2 1a	80	ap-south-1a (a...	Healthy	-	No override.	No overri...	March 26,...

CloudShell

Feedback

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

2. Check Load Balancer Status:

- Navigate to **Load Balancers**.
- Verify that the **DemoALB** status is **Active** or **Provisioning**.

The screenshot displays the AWS Management Console for the **DemoALB** Load Balancer. The **Details** tab is active, showing the following information:

- Load balancer type:** Application
- Scheme:** Internet-facing
- Status:** Active
- Hosted zone:** ZP97RAFLXTNZK
- VPC:** vpc-0360d11019ee41ea0
- Availability Zones:** subnet-006d8c52d39e28865 (ap-south-1b (aps1-az3)), subnet-0be0ce07400f57397 (ap-south-1c (aps1-az2)), subnet-0603aa0dd1018f970 (ap-south-1a (aps1-az1))
- Load balancer IP address type:** IPv4
- Date created:** March 26, 2025, 12:59 (UTC+05:30)
- Load balancer ARN:** arn:aws:elasticloadbalancing:ap-south-1:463470956366:loadbalancer/app/DemoALB/406a64ecec725ccb
- DNS name:** DemoALB-1767573967.ap-south-1.elb.amazonaws.com (A Record)

The **Resource map** tab is also visible, showing the architecture of the load balancer. It includes a diagram with the following components:

- Listeners (1):** HTTP:80
- Rules (1):** Priority default, Forward to target group
- Target groups (1):** Instance TG1, 2 targets
- Targets (2):** I-094e9b3a62973d5d2 (Port 80, Healthy), I-0e9abdc7fa1f5670c (Port 80, Healthy)

Step 6: Verify Load Balancer Traffic Distribution

1. Copy the "DNS Name" of the Load Balancer.
2. Open a browser and paste the **DNS Name** to check the webpage.

The screenshot shows a web browser window with the address bar displaying the DNS name of the load balancer: **demoalb-1767573967.ap-south-1.elb.amazonaws.com**. The page content displays the text: **Hello World from Mumbai 1a ip-172-31-43-97.ap-south-1.compute.internal**.

3. **Refresh the page multiple times** to see the traffic distribution between the two instances.



Now, you can see that the traffic is being **distributed equally** between the two EC2 instances. This demonstrates **high availability**, a key feature of AWS Load Balancer.

To allow users to access the application, simply share the **DNS Name of DemoALB**.

Currently, the Load Balancer has only one **Target Group (TG1)** with the following **Listener Rule**:

- **Forward to Target Group**
 - **TG1** (100% traffic distribution)
 - **Target group stickiness**: Off

Understanding Listener Rule Configuration

Currently, the Load Balancer has only one **Target Group (TG1)** with the following **Listener Rule**:

- **Forward to Target Group**
 - **TG1** (100% traffic distribution)
 - **Target group stickiness**: Off

What does this mean?

- **"Forward to Target Group TG1: 1 (100%)"** → All incoming traffic is directed to the **TG1** target group.
- **"Target group stickiness: Off"** → The Load Balancer does not persistently direct requests from the same client to the same instance; instead, it distributes traffic evenly across all healthy targets.

This setup ensures efficient load balancing and redundancy within the AWS region.

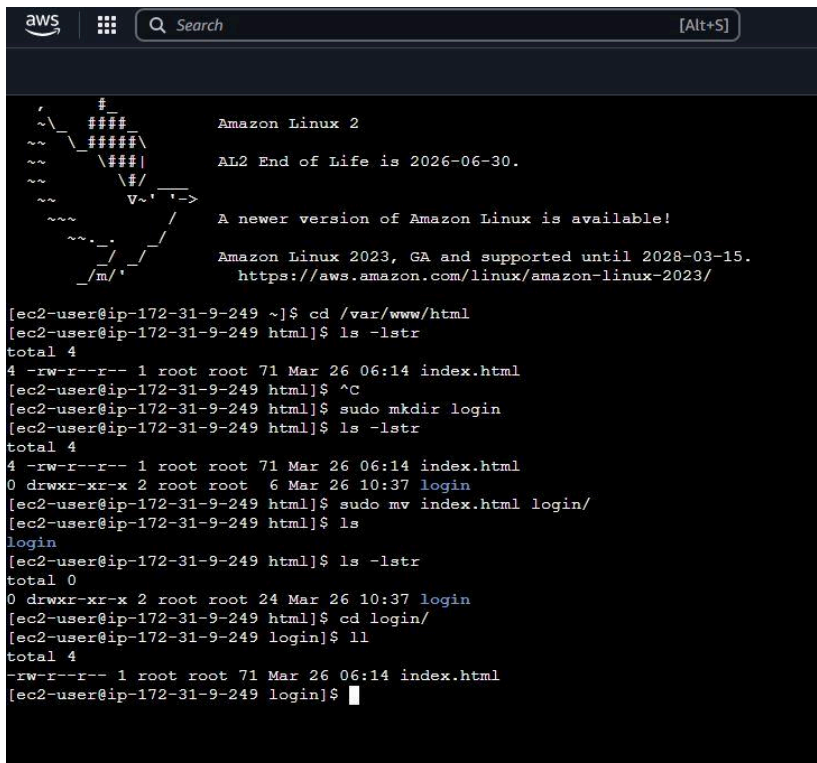
Step 7 :Advanced Use Cases for Load Balancer

Modify the Path of `index.html`

1. Go to **EC2 Dashboard**.
2. Select **EC2 Instance "Test EC2 1b"**.
3. Click **Connect** to access the instance (Ensure SSH access is enabled in the Security Group, as it was previously removed).
4. Run the following commands:

```
cd /var/www/html
ls -lstr
sudo mkdir login
sudo mv index.html login/
```

This changes the path of `index.html` on "Test EC2 1b".



```
aws [Alt+S]
Amazon Linux 2
AL2 End of Life is 2026-06-30.
A newer version of Amazon Linux is available!
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-9-249 ~]$ cd /var/www/html
[ec2-user@ip-172-31-9-249 html]$ ls -lstr
total 4
4 -rw-r--r-- 1 root root 71 Mar 26 06:14 index.html
[ec2-user@ip-172-31-9-249 html]$ ^C
[ec2-user@ip-172-31-9-249 html]$ sudo mkdir login
[ec2-user@ip-172-31-9-249 html]$ ls -lstr
total 4
4 -rw-r--r-- 1 root root 71 Mar 26 06:14 index.html
0 drwxr-xr-x 2 root root 6 Mar 26 10:37 login
[ec2-user@ip-172-31-9-249 html]$ sudo mv index.html login/
[ec2-user@ip-172-31-9-249 html]$ ls
login
[ec2-user@ip-172-31-9-249 html]$ ls -lstr
total 0
0 drwxr-xr-x 2 root root 24 Mar 26 10:37 login
[ec2-user@ip-172-31-9-249 html]$ cd login/
[ec2-user@ip-172-31-9-249 login]$ ll
total 4
-rw-r--r-- 1 root root 71 Mar 26 06:14 index.html
[ec2-user@ip-172-31-9-249 login]$
```

1. Navigate to the **Target Group Dashboard**.
2. Locate **Target Group "TG1"**.
3. Notice that the health check will fail for **EC2 Instance "Test EC2 1b"** because the default `/index.html` path no longer exists at the expected location.

Target groups > TG1

TG1

Details

arn:aws:elasticloadbalancing:ap-south-1:463470956366:targetgroup/TG1/23b3065278db1d04

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-0360d11019ee41ea0
IP address type IPv4	Load balancer DemoALB		

2 Total targets	1 Healthy	1 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

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.

Filter targets

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details	Admini...	Overrid...	Launch...	Anomaly detection...
<input type="checkbox"/>	i-Oe9abdc7fa1f5670c	Test EC2 1a	80	ap-south-1a (a...	Healthy	-	No override.	No overrid...	March 26, ...	Normal
<input type="checkbox"/>	i-094e9b3a62973d5d2	Test EC2 1b	80	ap-south-1b (a...	Unhealthy	Health checks failed wi...	No override.	No overrid...	March 26, ...	Normal

Load Balancer Path Awareness Issue

1. You can see in the below image that the **Load Balancer is unaware** that we changed the path of our web application (`index.html`) to `/var/www/html/login/index.html` for EC2 "Test EC2 1b".
2. However, it still redirects requests to the old path `/var/www/html`.



This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting [www.example.com](#), you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:



Fix Load Balancer Path Routing

1. We need to add **intelligence** to the Load Balancer so that when the web application is accessed from the new path, it correctly routes to the appropriate EC2 instance.
2. Navigate to **TG1 Target Group**.
3. **Deregister** the unhealthy EC2 instance **"Test EC2 1b"** from **TG1**.

Details
arn:aws:elasticloadbalancing:ap-south-1:463470956366:targetgroup/TG1/23b3065278db1d04

Target type
Instance

Protocol version
HTTP1

VPC
vpc-0360d11019ee41ea0

IP address type
IPv4

Load balancer
DemoALB

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

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

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.

Filter targets

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details	Admini...	Overrid...	Launch...	Anomaly detection...
<input type="checkbox"/>	i-0e9abdc7fa1f5670c	Test EC2 1a	80	ap-south-1a (a...	Healthy	-	No override.	No overrid...	March 26, ...	Normal
<input checked="" type="checkbox"/>	i-094e9b3a62973d5d2	Test EC2 1b	80	ap-south-1b (a...	Draining	Target deregistration is...	No override.	No overrid...	March 26, ...	Normal

Draining in AWS Load Balancer (Short Explanation)

Connection draining (or deregistration delay) ensures that active connections to an instance are **allowed to complete** before the instance is **removed** from the load balancer. This prevents ongoing requests from being abruptly terminated.

Step 8 : Create a New Target Group

1. Go to **Target Groups** in the AWS Console.
2. Click **Create Target Group**.
3. Set the **Name** as **"loginTG"**.
4. Keep **IP address type** as **IPv4**.
5. Select **VPC = HTTP1**.
6. Configure **Health Checks**:
 - The associated load balancer periodically sends requests to the registered targets to test their status.
 - **Health check protocol: HTTP**.
 - Use the default path **/** to perform health checks on the root, or specify a custom path.
 - Set the new health check path as **/login/**.
7. Click **Next**.

Create target group

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

Health check path

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/login/

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

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

Register EC2 Instance to New Target Group

1. Include EC2 "Test EC2 1b" under **Include as pending** below.
2. Click **Create Target Group**.

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)

<input type="checkbox"/>	Instance ID	Name	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
<input type="checkbox"/>	i-0e7ab067fa1f5670c	Test EC2 1a	Running	ec2sg	ap-south-1a	172.31.43.97	subnet-0603aa0001018970	March 26, 2025, 15:44 (UTC+05:30)
<input type="checkbox"/>	i-094e9b3a62973d5d2	Test EC2 1b	Running	ec2sg	ap-south-1b	172.31.9.249	subnet-006d8c52d39c28865	March 26, 2025, 15:39 (UTC+05:30)

0 selected

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

80

1-65535 (separate multiple ports with comma)

Include as pending below

1 selection is now pending below. Include more or register targets when ready.

Review targets

Targets (1)

Show only pending
Remove all pending

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
i-094e9b3a62973d5d2	Test EC2 1b	80	Running	ec2sg	ap-south-1b	172.31.9.249	subnet-006d8c52d39c28865	March 26, 2025, 15:39 (UTC+05:30)

1 pending

Cancel Previous Create target group

Target Group Overview

Now, you have the following Target Groups:

- 1. **TG1** – Includes **EC2 "Test EC2 1a"**.
- 2. **loginTG** – Includes **EC2 "Test EC2 1b"**.

EC2 > Target groups

Target groups (1/2)

Actions

Create target group

Filter target groups

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
loginTG	arn:aws:elasticloadbalancing...	80	HTTP	Instance	None associated	vpc-0360d11019ee41ea0
TG1	arn:aws:elasticloadbalancing...	80	HTTP	Instance	DemoALB	vpc-0360d11019ee41ea0

Target group: TG1

Details

Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (1)

Anomaly mitigation: Not applicable

Deregister

Register targets

Filter targets

Instance ID	Name	Port	Zone	Health status	Health status details	Admini...	Overri...	Launch...	Anomaly detection.
i-0e9abdc7fa1f5670c	Test EC2 1a	80	ap-south-1a (a...	Healthy	-	No override.	No overri...	March 26,...	Normal

EC2 > Target groups

Target groups (1/2)

Actions

Create target group

Filter target groups

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
loginTG	arn:aws:elasticloadbalancing...	80	HTTP	Instance	None associated	vpc-0360d11019ee41ea0
TG1	arn:aws:elasticloadbalancing...	80	HTTP	Instance	DemoALB	vpc-0360d11019ee41ea0

Target group: loginTG

Details

Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (1)

Anomaly mitigation: Not applicable

Deregister

Register targets

Filter targets

Instance ID	Name	Port	Zone	Health status	Health status details	Admini...	Overri...	Launch...	Anomaly detection.
i-094e9b3a62973d5d2	Test EC2 1b	80	ap-south-1b (a...	Unused	Target group is not co...	-	-	March 26,...	Normal

Step 9: Creating a New Listener Rule for Load Balancer

In this step, we will create a new listener rule for our Load Balancer to handle traffic for the path `"/login/"`.

1. Navigate to Listener Settings:

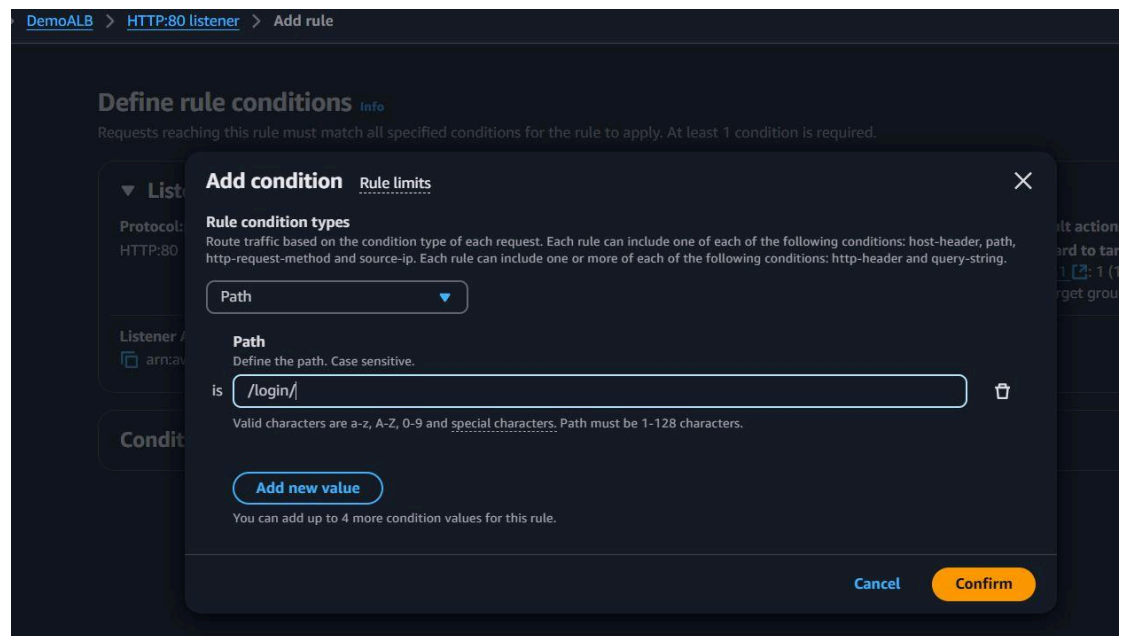
- Go to the Listener section of your Load Balancer.
- Click on Add Rule.

2. Add a Name to the Rule:

- Provide a meaningful name for the rule.

3. Define Rule Conditions:

- Click on Add Condition.
- Select "Path" as the condition type.
- Click Add Path and enter the desired path `"/login/"`.



- Click Next.

4. Define Rule Actions:

- Under Actions, choose Forward to Target Group.
- Select the Target Group as "loginTG".
- Click Next.

DemoALB > HTTP:80 listener > Add rule

► Listener details: HTTP:80

Actions

Action types

Routing actions

☒ 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: loginTG Target type: Instance, IPv4 HTTP 1 100%
0-999

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

5. Set Rule Priority:

- Specify the rule priority. In this case, set it to 1.
- Click Next.

EC2 > Load balancers > DemoALB > HTTP:80 listener > Add rule

Step 1: Add rule
Step 2: Define rule conditions
Step 3: Define rule actions
Step 4: **Set rule priority**
Step 5: Review and create

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 details: HTTP:80

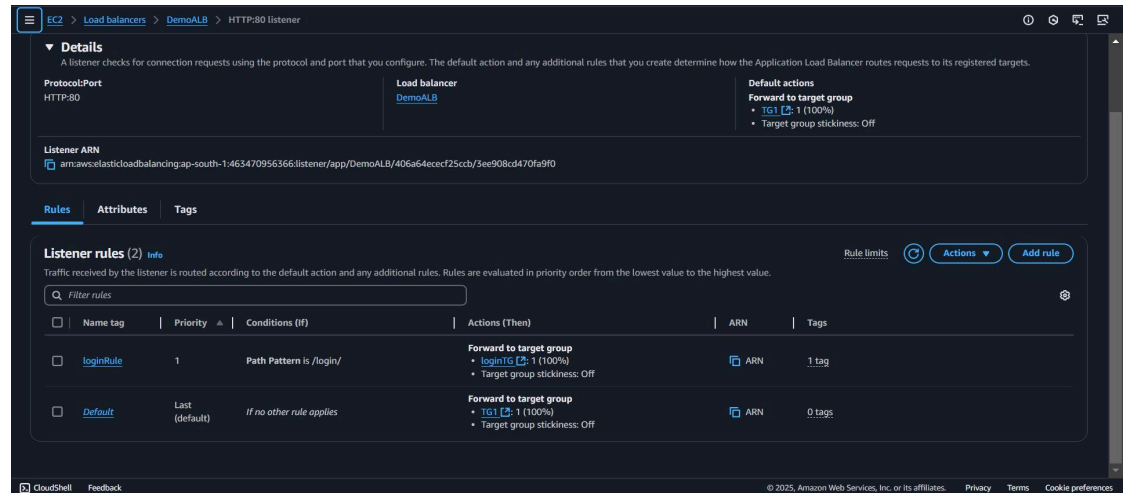
Listener rules (2) Info

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.

Name tag	Priority	Conditions (if)	Actions (Then)	ARN	Tags
loginRule	1 <small>Priority value must be 1-50,000.</small>	Path Pattern is /login/	Forward to target group <ul style="list-style-type: none"> loginTG (100%) Target group stickiness: Off 	Pending	1 tag
Default	Last (default)	If no other rule applies	Forward to target group <ul style="list-style-type: none"> TG1 (100%) Target group stickiness: Off 	ARN	0 tags

6. Review and Create:

- Verify all configurations.
- Click Create Rule.



Now, the listener will have two rules, and they will be executed in a top-to-bottom order based on priority.

Step 10 :- Verifying the Load Balancer Routing

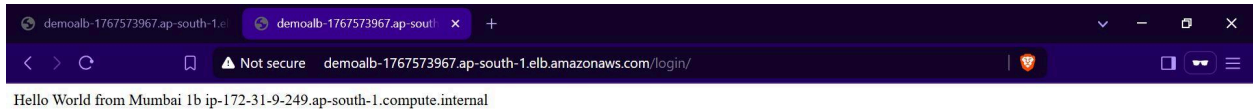
1. Test Default Load Balancer Behavior:

- Hit the URL of the Load Balancer "DemoALB".
- You should see the load balancer redirecting traffic to EC2 "Test EC2 1a".



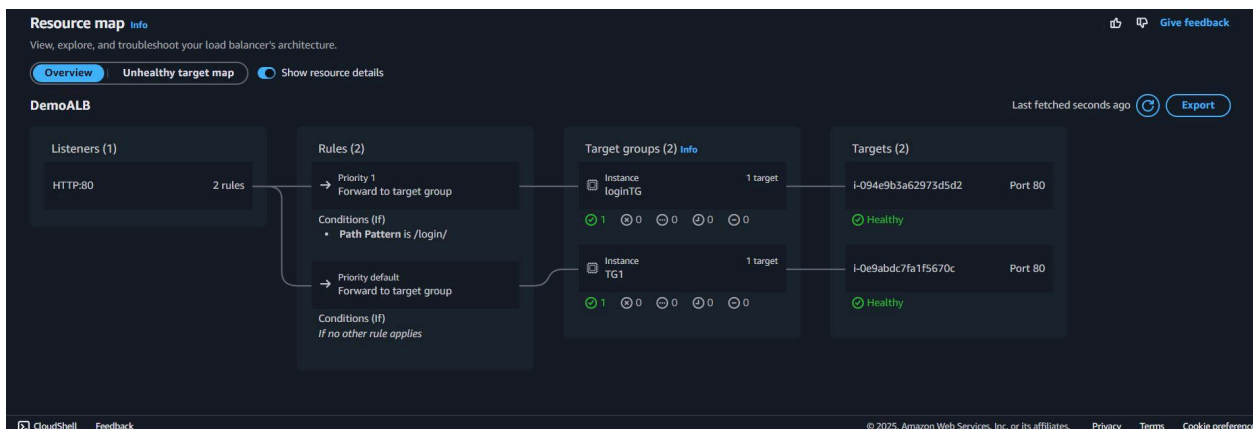
2. Test Path-Based Routing:

- Hit the URL of the Load Balancer "DemoALB" and append '/login/' at the end.
- You should see the load balancer redirecting traffic to EC2 "Test EC2 1b".



3. Resource Mapping for Clear Understanding:

- To get a better understanding, you can check the Resource Map for the "DemoALB" Load Balancer.





Understanding the Round Robin Algorithm in AWS ALB:

The Application Load Balancer (ALB) in AWS uses the Round Robin Algorithm to distribute incoming traffic across multiple registered targets within a target group.

- **How It Works:** Each new request is sent to the next available target in a circular order, ensuring even distribution of traffic.
- **Example in ALB:** If there are two EC2 instances (Test EC2 1a and Test EC2 1b) in the target group, the first request will be routed to Test EC2 1a, the second to Test EC2 1b, the third back to Test EC2 1a, and so on.
- **Load Balancing Benefits:** This approach ensures that no single instance is overwhelmed while others remain idle, providing efficient resource utilization and improved application availability.

Target groups (1/2) [Info](#)

<input type="checkbox"/>	Name	ARN	Port
<input type="checkbox"/>	loginTG	 arn:aws:elasticloadbalancin...	80
<input checked="" type="checkbox"/>	TG1	 arn:aws:elasticloadbalancin...	80

Target group: TG1

Details

Targets

Monitoring

Health checks

Attributes

Attributes

Target deregistration management

Deregistration delay (draining interval)
300 seconds

Traffic configuration

Load balancing algorithm
Round robin

