

Hands-On with AWS Load Balancer: EC2 Setup, Target Groups & Health Checks

Introduction

A **Load Balancer (LB)** in AWS helps distribute incoming traffic across multiple EC2 instances. This ensures:

- No single instance is overloaded
- Higher **availability**
- Improved **fault tolerance**

How Load Balancer Works

Load Balancer automatically routes client requests to healthy targets (EC2s) based on various **strategies** like:

- **Round Robin**
- **Least Outstanding Requests**

LB performs **health checks** on targets and only routes traffic to **healthy** ones. It listens on specific ports (e.g., 80 or 443) and routes the requests to a **Target Group** of EC2 instances.

Types of Load Balancers in AWS

1. Application Load Balancer (ALB)
2. Network Load Balancer (NLB)
3. Classic Load Balancer (CLB)
4. Gateway Load Balancer (GWLB)

Type	Layer	Protocols	Use Case
ALB	Layer 7	HTTP/HTTPS	Application routing, content-based
NLB	Layer 4	TCP/UDP	High performance, low latency
CLB	Both 4 & 7	HTTP, TCP	Legacy systems
GWLB	Layer 3/4	Custom flows	Used with firewalls, inspection tools

TCP vs UDP

- **TCP:** Reliable, ordered, but slower (e.g., Email)
- **UDP:** Faster, connectionless, some data loss possible (e.g., video/voice calls)

OSI Model :

The Open Systems Interconnection (OSI) model defines a framework for network communication. It consists of 7 layers:

Layer 7 to Layer 1:

Application → Presentation → Session → Transport → Network → Data Link → Physical

LB primarily works at **Layer 4 (NLB)** and **Layer 7 (ALB)**.

Step-by-Step Practical: Load Balancer Setup in AWS

1. Create a Security Group

Go to EC2 > Security Groups

- Create a new SG allowing:
 - Inbound: TCP 22, TCP 80
 - Outbound: All traffic

The screenshot shows the AWS Security Groups console. A green success message at the top states: "Security group (sg-0ac40be40db7ad191 | SG-targets) was created successfully". Below it, the security group details are shown: Name: sg-0ac40be40db7ad191, Description: SG-targets, VPC ID: vpc-0d08096a0a77e4f38. Under the "Inbound rules" tab, there are two entries: one for port 22 (SSH) and one for port 80 (HTTP). Both rules are set to TCP and are associated with the security group rule ID sgr-0a75581422f10d69a.

Screenshot 1: Security Group allowing ports 22 (SSH) and 80 (HTTP)

2. Launch Two Web Servers

- Launch 2 EC2 instances using Amazon Linux or Ubuntu
- SSH into each instance
- Install Apache or Nginx:
- sudo yum install httpd -y
- sudo systemctl start httpd
- sudo systemctl enable httpd

- Modify index.html with unique content:
- echo "This is Web Server 1" > /var/www/html/index.html

And similarly for Server 2.

The screenshot shows the AWS EC2 Instances page. It displays two instances: Target1 and Target2. Both instances are in the 'Running' state, t2.micro type, and located in the 'ap-south-1b' availability zone. The 'Instances' section is selected in the sidebar.

Name	Instance ID	Instance State	Type	Status Check	Alarm Status	Availability Zone
Target1	i-02d9549239f4b7b79	Running	t2.micro	Initializing	View alarms +	ap-south-1b
Target2	i-0d05b266719d39cdf	Running	t2.micro	-	View alarms +	ap-south-1b

Screenshot 2: Running EC2 Instances

```
[root@ip-172-31-15-32 ~]# cat /var/www/html/server1/index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <title>Server 1 - DevOps Projects</title>
    <style>
        body {
            margin: 0;
            font-family: 'Segoe UI', sans-serif;
            color: #ffffff;
            text-align: center;
            height: 100vh;
            /* Purple to sky blue gradient background */
            background: linear-gradient(135deg, #6a11cb, #2575fc);
            display: flex;
            flex-direction: column;
            justify-content: center;
            align-items: center;
        }
        h1 {
            font-size: 3em;
            text-shadow: 2px 2px 10px #000000aa;
            margin-bottom: 0.3em;
        }
        p {
            font-size: 1.2em;
            text-shadow: 1px 1px 5px #00000099;
            margin-top: 0;
        }
    </style>
</head>
```

Screenshot 3: index.html content for Web Server 1

```
[root@ip-172-31-15-32 ~]# cat > /var/www/html/server2/index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <title>Server 2 - DevOps Projects</title>
    <style>
        body {
            margin: 0;
            font-family: 'Segoe UI', sans-serif;
            color: #ffffff;
            text-align: center;
            height: 100vh;
            /* Purple to sky blue gradient background */
            background: linear-gradient(135deg, #6a11cb, #2575fc);
            display: flex;
            flex-direction: column;
            justify-content: center;
            align-items: center;
        }
        h1 {
            font-size: 3em;
            text-shadow: 2px 2px 10px #000000aa;
            margin-bottom: 0.3em;
        }
        p {
            font-size: 1.2em;
            text-shadow: 1px 1px 5px #00000099;
            margin-top: 0;
        }
    </style>
</head>
```

Screenshot 4: index.html content for Web Server 2

3. Create a Target Group

- Navigate to EC2 > Target Groups
- Create a new target group (choose instance-based, protocol: HTTP)
- Add both EC2 instances as **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.

The screenshot shows the 'Available instances (2/2)' section of the AWS Lambda console. Two instances are listed:

Instance ID	Name	State	Security groups	Zone	Private IPv4 address	Subnet
i-0d05b266719d39cdf	Target2	Running	SG-targets	ap-south-1b	172.31.14.154	sub
i-02d9549239f4b7b79	Target1	Running	SG-targets	ap-south-1b	172.31.15.32	sub

Below the table, it says '2 selected'. Under 'Ports for the selected instances', the port '80' is selected. A button 'Include as pending below' is visible.

Screenshot 5: Target Group creation

Review targets

The screenshot shows the 'Targets (2)' section of the AWS Lambda console. Two targets are listed:

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
i-0d05b266719d39cdf	Target2	80	Running	SG-targets	ap-south-1b	172.31.14.154	subnet-087c15be1693bf0e5	May 29, 2025, 11:18 (UTC+05:30)
i-02d9549239f4b7b79	Target1	80	Running	SG-targets	ap-south-1b	172.31.15.32	subnet-087c15be1693bf0e5	May 29, 2025, 11:17 (UTC+05:30)

Screenshot 6: Both EC2s registered as targets

4. Create a Load Balancer

- Navigate to EC2 > Load Balancers
- Create **Application Load Balancer**
- Select **internet-facing**, port 80
- Choose the Security Group created in step 1
- Assign the **Target Group** from step 3

Create Application Load Balancer Info

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.

LB1

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.

Screenshot 7: Creating an Application Load Balancer (ALB)

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
HTTP	: 80 1-65535

Default action Info

Forward to	TG1	HTTP
Target type: Instance, IPv4		



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

Screenshot 8: Listener configuration and Target Group selection

5. Access the Application via DNS

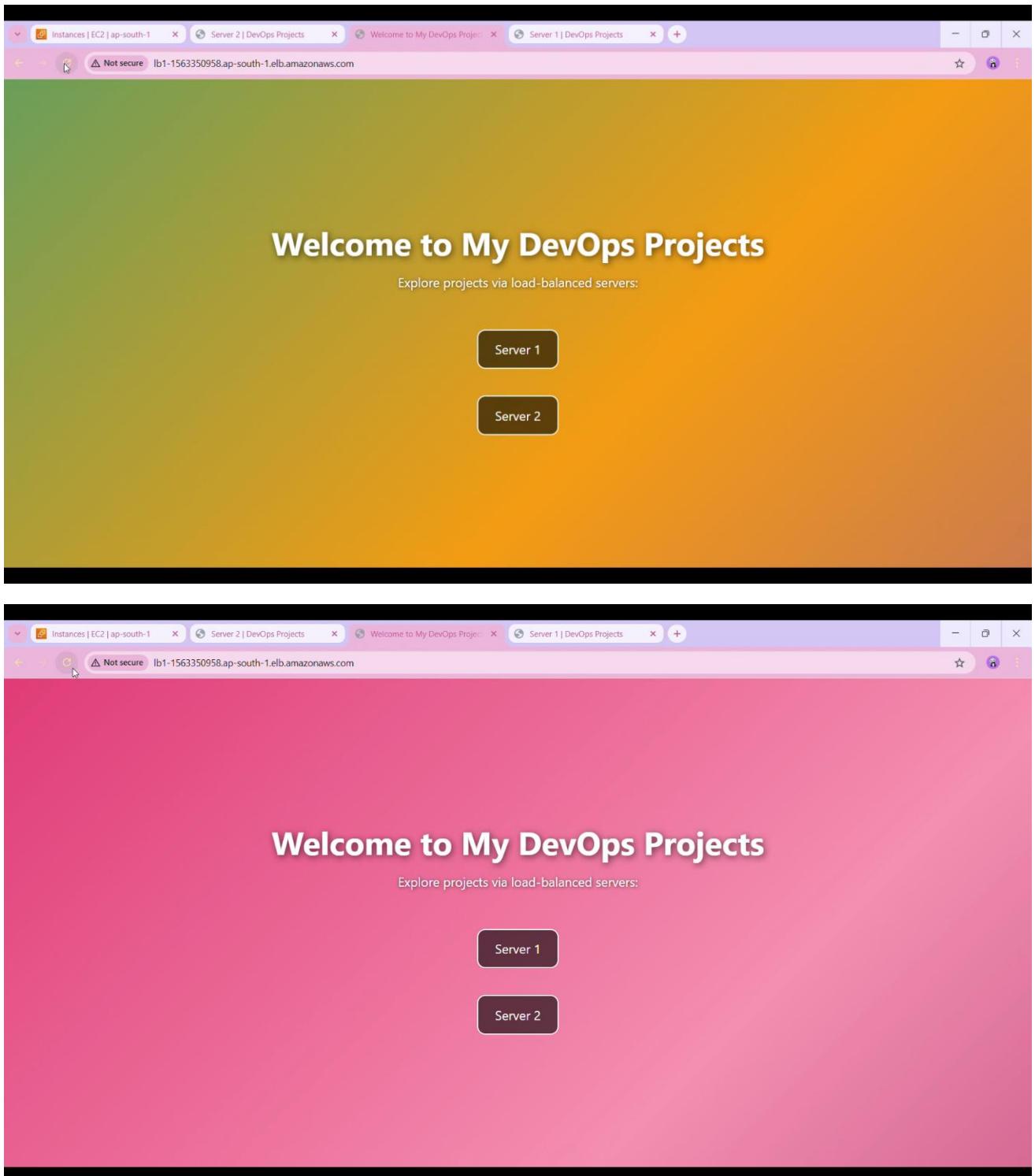
- Go to Load Balancer > Description
- Copy the **DNS name**
- Paste in browser → You should see content from Web Server 1 or 2 randomly (due to round robin)

Load balancers (1)

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

Filter load balancers						
Name	DNS name	State	VPC ID	Availability Zones	Type	
LB1	LB1-1313006664.ap-south-1.elb.amazonaws.com	Active	vpc-0d08096a0a77e4f38	3 Availability Zones	application	

Screenshot 9: DNS Name of Load Balancer in EC2 Console



Screenshot 10: Browser output showing Web Server 1 and 2 responses after refresh

Load Balancer Health Check Parameters

- **Timeout:** Wait time for instance to respond (e.g., 5 sec)
- **Interval:** Time between two health checks (e.g., 30 sec)
- **Healthy Threshold:** Min. success attempts to be considered healthy
- **Unhealthy Threshold:** Min. failures before marking unhealthy

The LB continuously **pings** instances. If it gets a 200 OK response within timeout, the instance is healthy.

▼ Advanced health check settings Restore defaults

Health check port
The port the load balancer uses when performing health checks on targets. By default, the health check port is the same as the target group's traffic port. However, you can specify a different port as an override.
 Traffic port
 Override

Healthy threshold
The number of consecutive health checks successes required before considering an unhealthy target healthy.

2-10

Unhealthy threshold
The number of consecutive health check failures required before considering a target unhealthy.

2-10

Timeout
The amount of time, in seconds, during which no response means a failed health check.
 seconds
2-120

Interval
The approximate amount of time between health checks of an individual target
 seconds
5-300

Success codes
The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,202") or a range of values (for example, "200-299").

Screenshot 11: Health check configuration page

Load Balancing Strategies

1.Round Robin:

Request 1 → Server 1

Request 2 → Server 2

Request 3 → Server 1 (repeat)

2.Least Outstanding Requests:

Sends traffic to server with **least active requests**

Conclusion

With just a few clicks and commands, you can set up a **robust, fault-tolerant Load Balancer** in AWS. This hands-on:

- Reinforces OSI Layer understanding
- Covers real-world deployment
- Teaches scaling strategies with ALB/NLB