

Load Balancer

A **Load Balancer** is a system that **distributes incoming network traffic** across multiple servers (or instances) to ensure that no single server is overwhelmed. It **improves the availability, performance, and reliability** of applications by managing workload distribution.

AWS provides **Elastic Load Balancing (ELB)**, which automatically distributes incoming application traffic across multiple targets, such as Amazon **EC2 instances, containers, and IP addresses**.

How Does a Load Balancer Work?

1. Traffic Distribution Process

1. **Client Request** → A user accesses an application (e.g., a website or API).
2. **DNS Resolution** → The request is resolved to the **Load Balancer's public IP address**.
3. **Health Checks** → The Load Balancer checks the health of backend servers and routes traffic only to healthy instances.
4. **Traffic Routing** → Based on configured **routing algorithms** (Round Robin, Least Connections, etc.), traffic is sent to an available backend server.
5. **Response to Client** → The backend server processes the request and sends the response back through the Load Balancer.

2. Key Features of Load Balancers

- ✓ **Automatic Scaling** → Handles increased traffic by adding/removing backend servers dynamically.
- ✓ **Health Monitoring** → Regularly checks if backend servers are healthy and removes unhealthy ones.
- ✓ **SSL/TLS Termination** → Manages HTTPS encryption to offload processing from backend servers.
- ✓ **Sticky Sessions** → Ensures a client connects to the same backend server for session consistency.
- ✓ **Cross-Zone Load Balancing** → Distributes traffic across multiple Availability Zones for reliability.

OSI Layers in Load Balancer

A **Load Balancer** operates at different layers of the **OSI (Open Systems Interconnection) model**, depending on the type of Load Balancer used. AWS **Elastic Load Balancer (ELB)** supports **four types of load balancers**, each working at different OSI layers.

Layer 7 - Application layer or HTTP layer

Layer 6 - Presentation layer

Layer 5 - Sessions layer

Layer 4 - Transport layer

Layer 3 - Network layer

Layer 2 - Data link layer

Layer 1 - Physical layer

OSI Model Overview

Layer	Name	Function	Example in Load Balancing
Layer 7	Application	Handles HTTP/HTTPS requests, URL-based routing, host-based routing	Application Load Balancer (ALB)
Layer 4	Transport	Manages TCP/UDP connections, port-based traffic distribution	Network Load Balancer (NLB)
Layer 3	Network	Routes IP-based traffic between networks	Gateway Load Balancer (GWLB)
Layer 2	Data Link	Handles MAC address-based switching (not used in AWS Load Balancers)	✗ Not used in AWS
Layer 1	Physical	Concerned with physical network connections	✗ Not applicable

Types of Load Balancers in AWS

Load Balancer Type	OSI Layer	Use Case
Application Load Balancer (ALB)	Layer 7 (HTTP/HTTPS)	Web applications, microservices, API gateways
Network Load Balancer (NLB)	Layer 4 (TCP/UDP)	Low-latency applications, gaming, real-time communication
Gateway Load Balancer (GLB)	Layer 3 (Network)	Distributes traffic across firewall appliances
Classic Load Balancer (CLB)	Layer 4 & 7	Legacy applications (deprecated in modern AWS setups)

1. Application Load Balancer (ALB) - Layer 7

- Works at **Layer 7 (Application Layer)**.
- Routes requests based on **URLs, HTTP headers, cookies, and request parameters**.
- Supports **host-based, path-based, and query-based routing**.
- Used for **web applications, microservices, and API gateways**.

2. Network Load Balancer (NLB) - Layer 4

- Works at **Layer 4 (Transport Layer)**.
- Routes traffic based on **TCP/UDP connections**.
- Handles **millions of requests per second** with ultra-low latency.
- Suitable for **real-time applications, gaming servers, VoIP, and databases**.
- Example:
 - Client connects to an **NLB** using **TCP port 443 (HTTPS)**.
 - The **NLB** forwards traffic to EC2 instances without inspecting the application data.

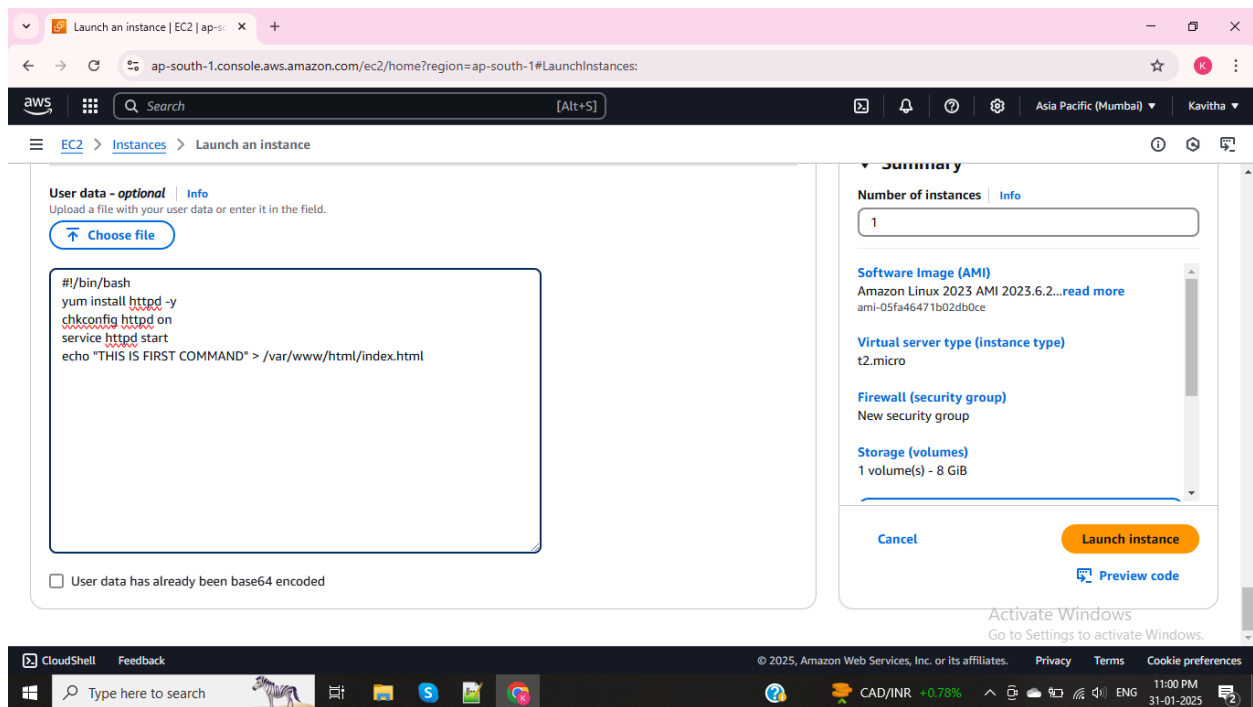
3. Gateway Load Balancer (GWLB) - Layer 3

- Works at **Layer 3 (Network Layer)**.
- Routes **IP-based** traffic.
- Used for **security appliances** like **firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS)**.
- Helps with **centralized security inspection and traffic filtering**.

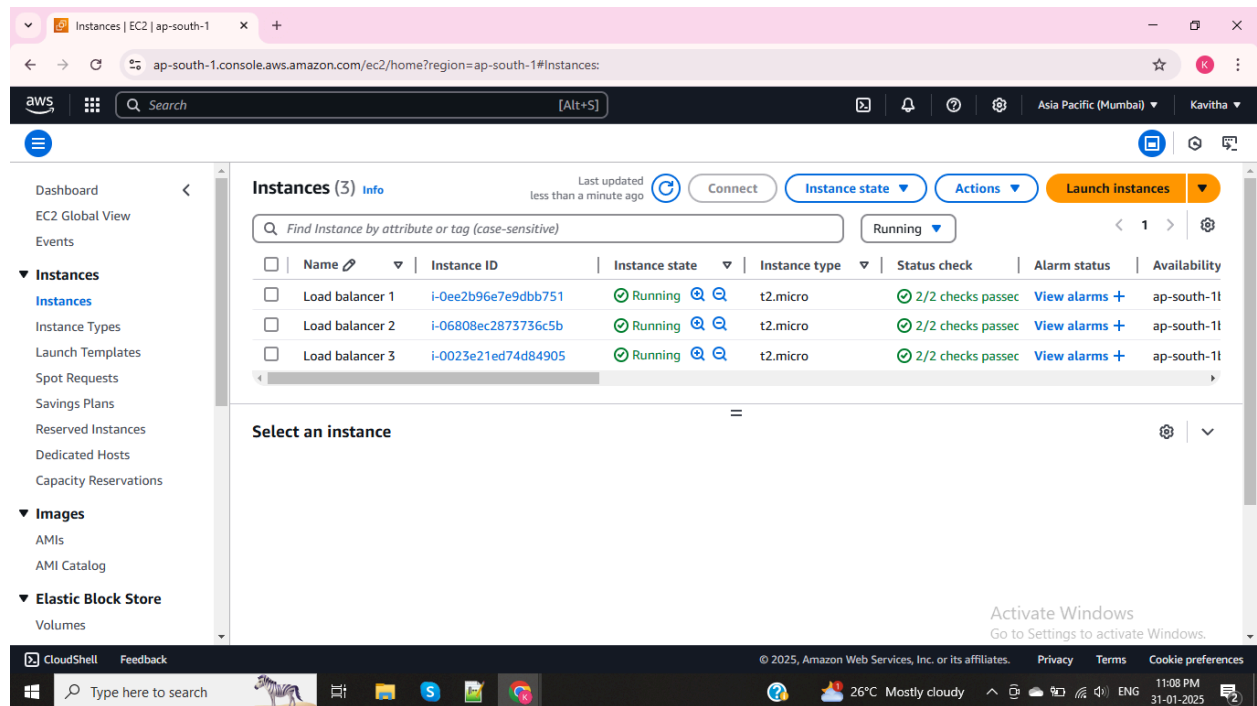
TASK : CREATING 3 EC2 INSTANCE AND REGISTER THOSE INSTANCE TO LOAD BALANCER

Go to the AWS Management Console.

Navigate to **EC2 Dashboard** > Create EC2 Instance and add advanced script in userdata > Launch instance > After Launch instance select that Using actions > launch more like this option launched another two EC2 instance with different echo commands.



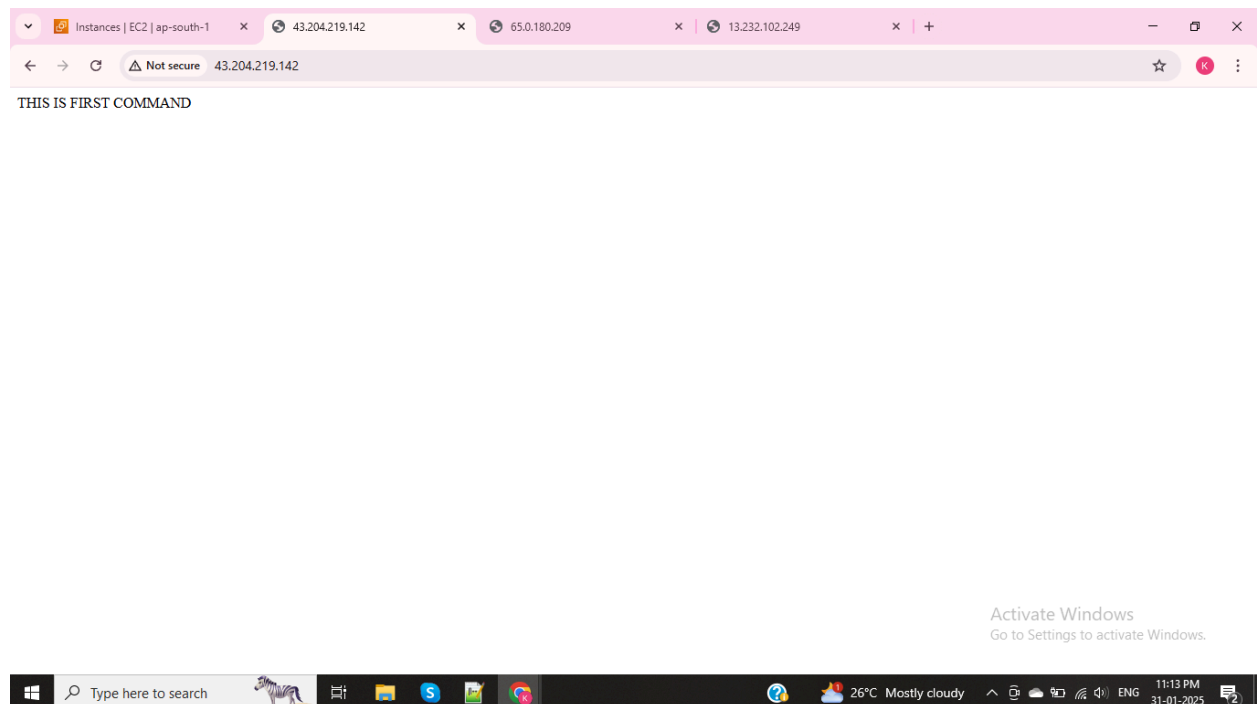
Created 3 EC2 Instance



The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'Instances' page displays three EC2 instances, all in a 'Running' state. The instances are named 'Load balancer 1', 'Load balancer 2', and 'Load balancer 3'. Each instance has a unique Instance ID and is using the 't2.micro' instance type. The status check for all instances shows '2/2 checks passed'. The console also includes a sidebar with navigation options like Dashboard, EC2 Global View, Events, and a list of services under 'Instances', 'Images', and 'Elastic Block Store'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
Load balancer 1	i-0ee2b96e7e9dbb751	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1l
Load balancer 2	i-06808ec2873736c5b	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1l
Load balancer 3	i-0023e21ed74d84905	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1l

Using public ip address of each instance checking the output for Instance whether userdata is working or not.



The screenshot shows a web browser with three tabs. The first tab is titled 'Instances | EC2 | ap-south-1' and shows the public IP address '43.204.219.142'. The second tab is titled '65.0.180.209' and shows the public IP address '65.0.180.209'. The third tab is titled '13.232.102.249' and shows the public IP address '13.232.102.249'. The browser address bar shows 'Not secure 43.204.219.142'. The page content displays the text 'THIS IS FIRST COMMAND'. The browser also shows a Windows taskbar at the bottom with the date '31-01-2025' and time '11:13 PM'.

Second Instance output

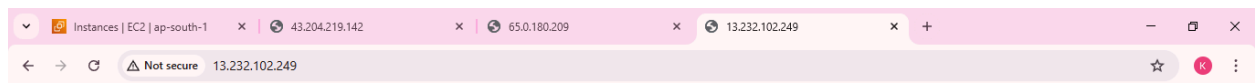


THIS IS SECOND COMMAND

Activate Windows
Go to Settings to activate Windows.



Third Instance output

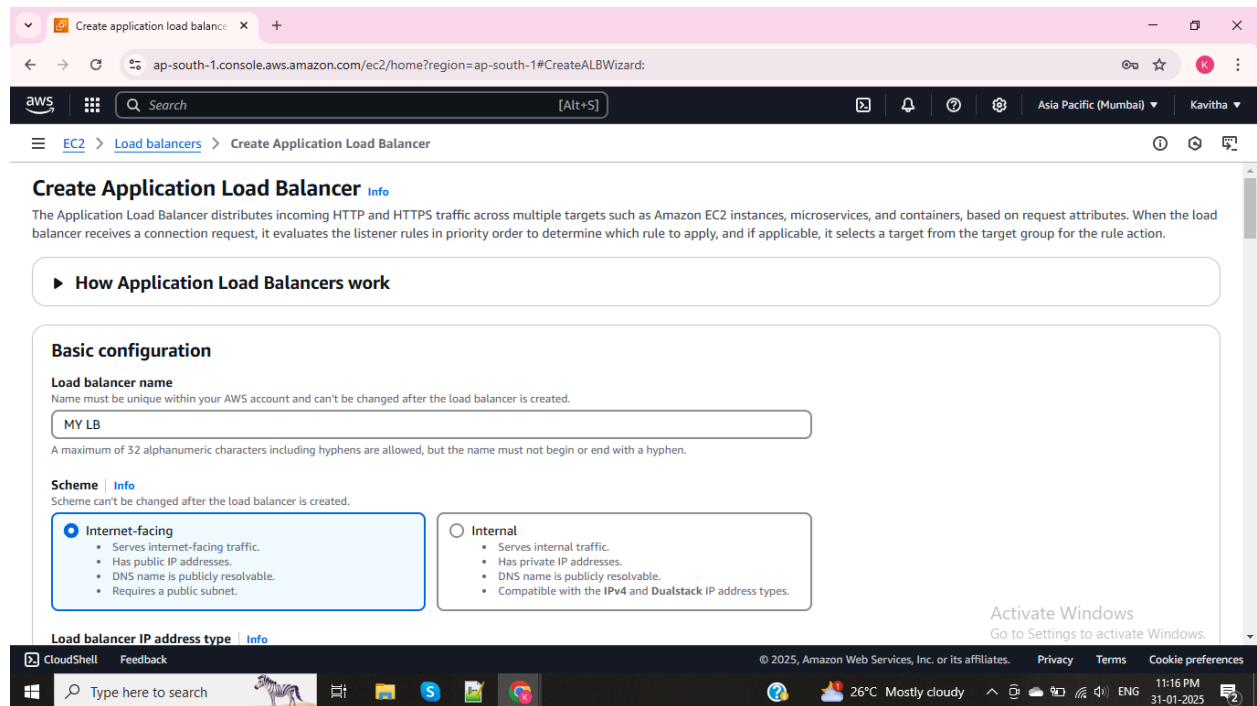


THIS IS THIRD COMMAND

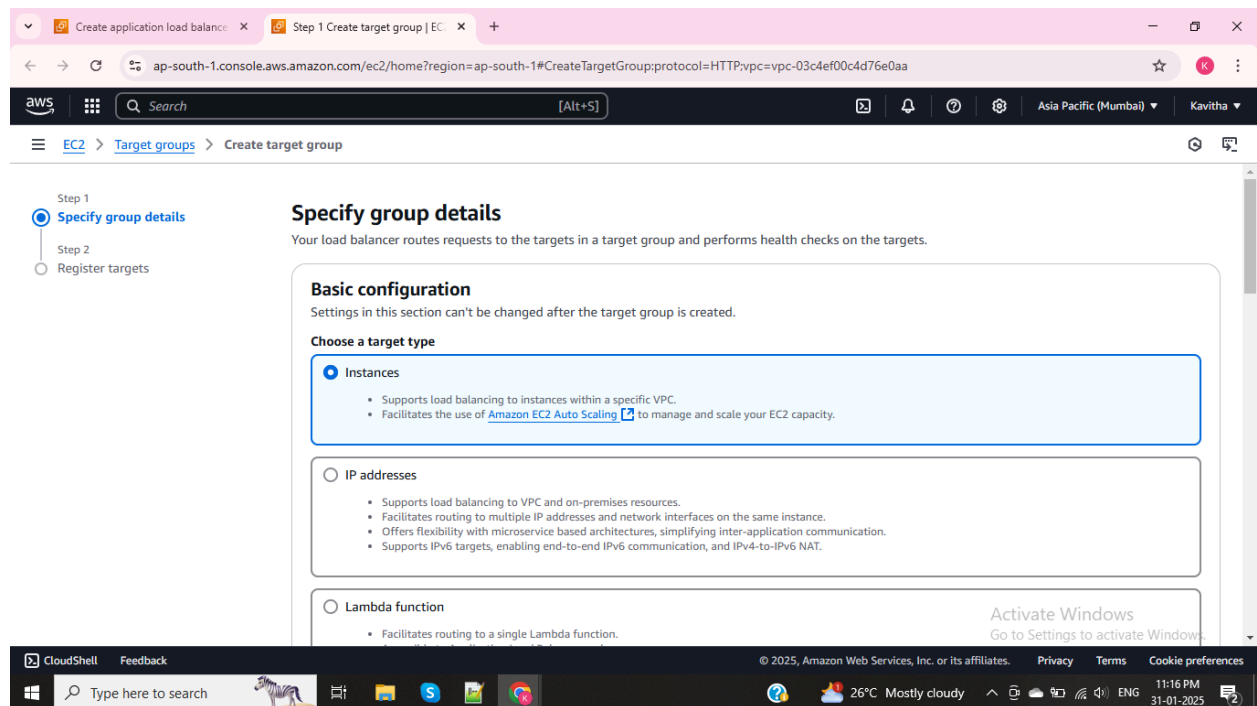
Activate Windows
Go to Settings to activate Windows.



Go to EC2 Dashboard > Load balancer > click load balancer > Creating Application load balancer with basic configurations



For attached load balancer created Target Group > Choose Instance target type



Next > In Available instance > Register the Created Instance to Application load balancer

The screenshot shows the 'Register targets' step in the AWS Management Console. The breadcrumb navigation is 'EC2 > Target groups > Create target group'. The page title is 'Register targets'. A sub-header states: '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.'

On the left, there are two steps: 'Step 1 Specify group details' and 'Step 2 Register targets' (which is the active step). Below the steps, there is a section titled 'Available instances (3)'. It contains a table with the following data:

<input type="checkbox"/>	Instance ID	Name	State	Security groups
<input type="checkbox"/>	i-0ac28ede1a266e6af	Load balancer 3	Running	launch-wizard-1
<input type="checkbox"/>	i-00724b0ca502fb7d9	Load balancer 2	Running	launch-wizard-1
<input type="checkbox"/>	i-0ee2b96e7e9dbb751	Load balancer 1	Running	launch-wizard-1

Below the table, it says '0 selected'. There is a section for 'Ports for the selected instances' with a text input field containing '80'. At the bottom right, there is a note: 'Activate Windows. Go to Settings to activate Windows.'

Now successfully Target Group is attached

The screenshot shows the 'Listeners and routing' step in the AWS Management Console. The breadcrumb navigation is 'EC2 > Load balancers > Create Application Load Balancer'. The page title is 'Listeners and routing'. A sub-header states: '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.'

Below the header, there is a section for 'Listener HTTP:80'. It contains a 'Protocol' dropdown set to 'HTTP' and a 'Port' input field set to '80'. To the right, there is a 'Default action' section with a 'Forward to' dropdown set to 'targetgroup' and a 'Target type' dropdown set to 'Instance, IPv4'. There is a 'Remove' button to the right of the listener name.

Below the listener configuration, there is a section for 'Listener tags - optional'. It contains a text input field for 'Add listener tag' and a button 'Add listener tag'. Below this, there is a button 'Add listener'.

At the bottom, there is a section for 'Load balancer tags - optional'.

Load Balancer created

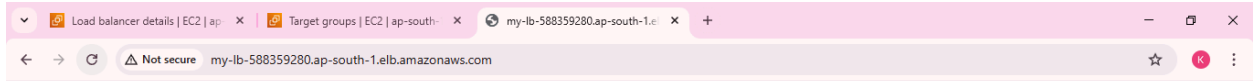
The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'Load balancers' page is active, displaying a table with one load balancer:

Name	DNS name	State	VPC ID	Availability Zones	Type
MY-LB	MY-LB-588359280.ap-south-1.elb.amazonaws.com	Active	vpc-03c4ef00c4d76e0aa	3 Availability Zones	application

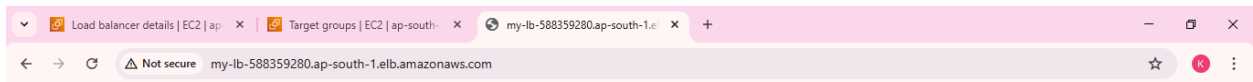
The left sidebar shows the 'Load Balancing' section selected. The top navigation bar includes the AWS logo and search bar. The bottom taskbar shows the Windows operating system with various application icons.

After creating load balancer it provides DNS URL (Domain Naming System). Using URL copy and paste in the browser to display output of Load balancer, it redirect to Instance 1,2 and 3 (Refresh the page to see all outputs)

The screenshot shows a web browser window with the address bar containing the DNS URL: `my-lb-588359280.ap-south-1.elb.amazonaws.com`. The page content is blank, indicating a redirect to the EC2 instances. The browser's taskbar shows the Windows operating system with various application icons.



THIS IS FIRST COMMAND



THIS IS SECOND COMMAND

