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.

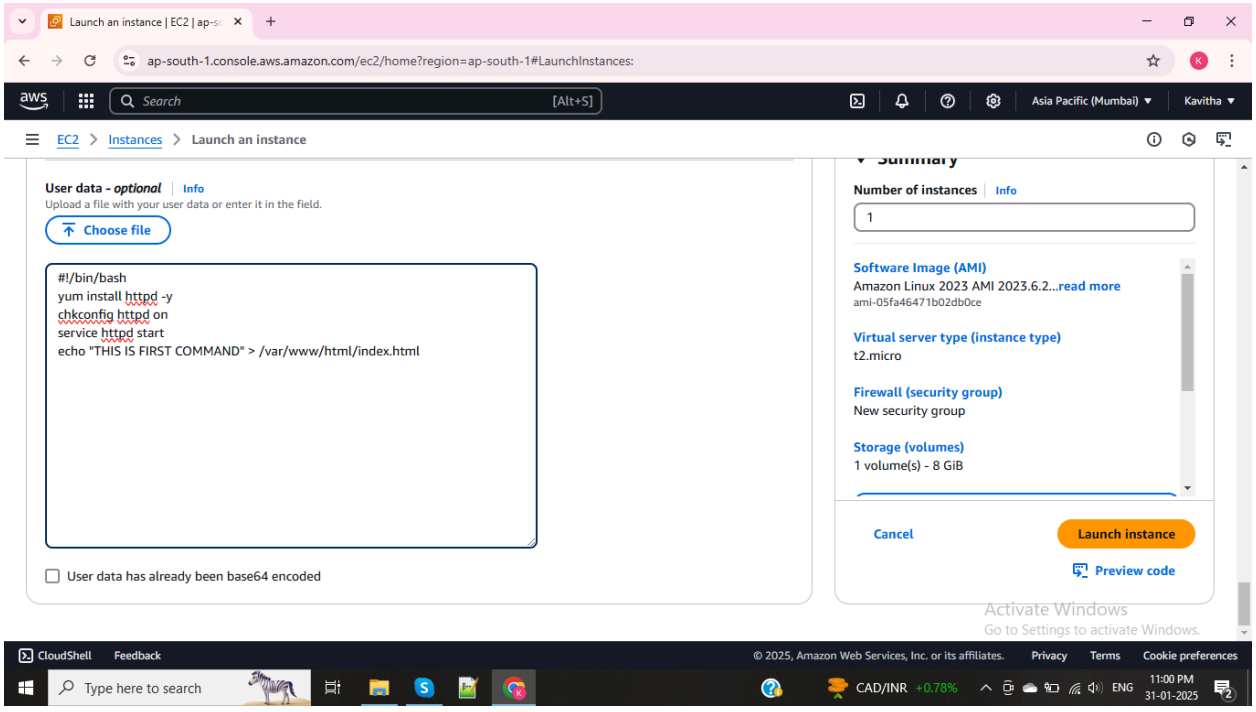
3. Gateway Load Balancer (GWLB) - Layer 3

- 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**.

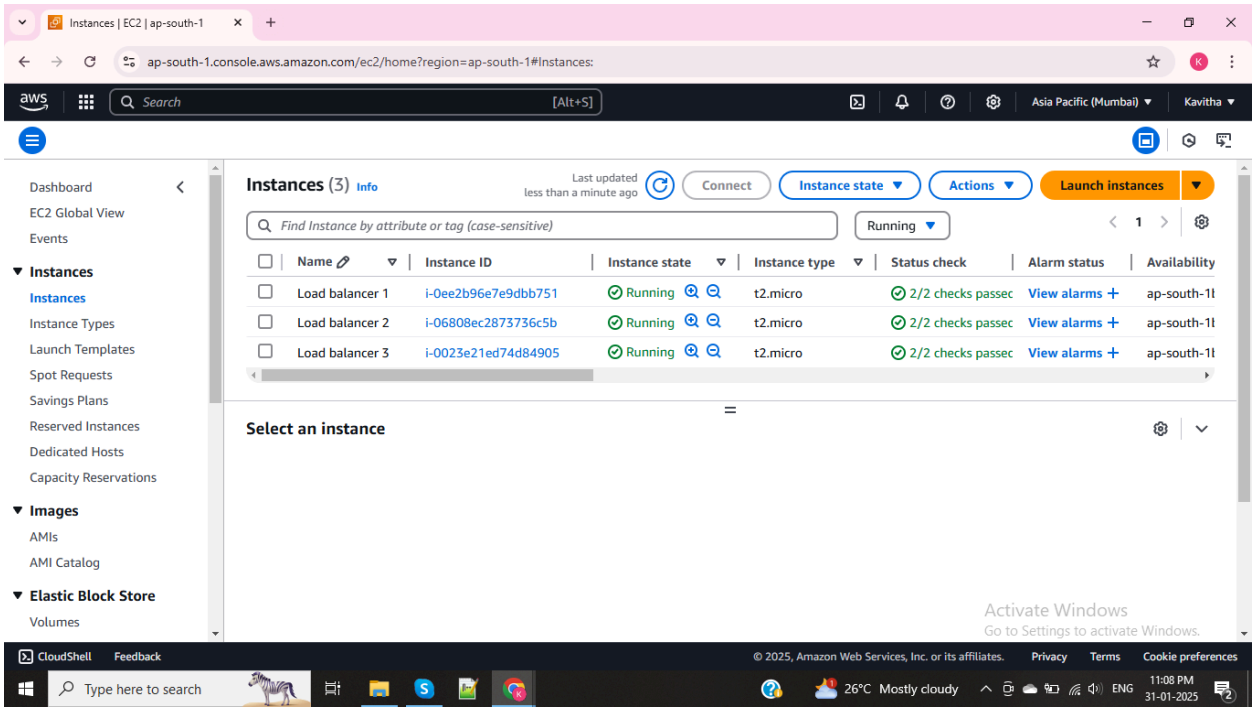
TASK : CREATING 3 EC2 INSTANCE AND REGISTER THOSE INSTANCE TO LOAD BALANCER

Go to the AWS Management Console.

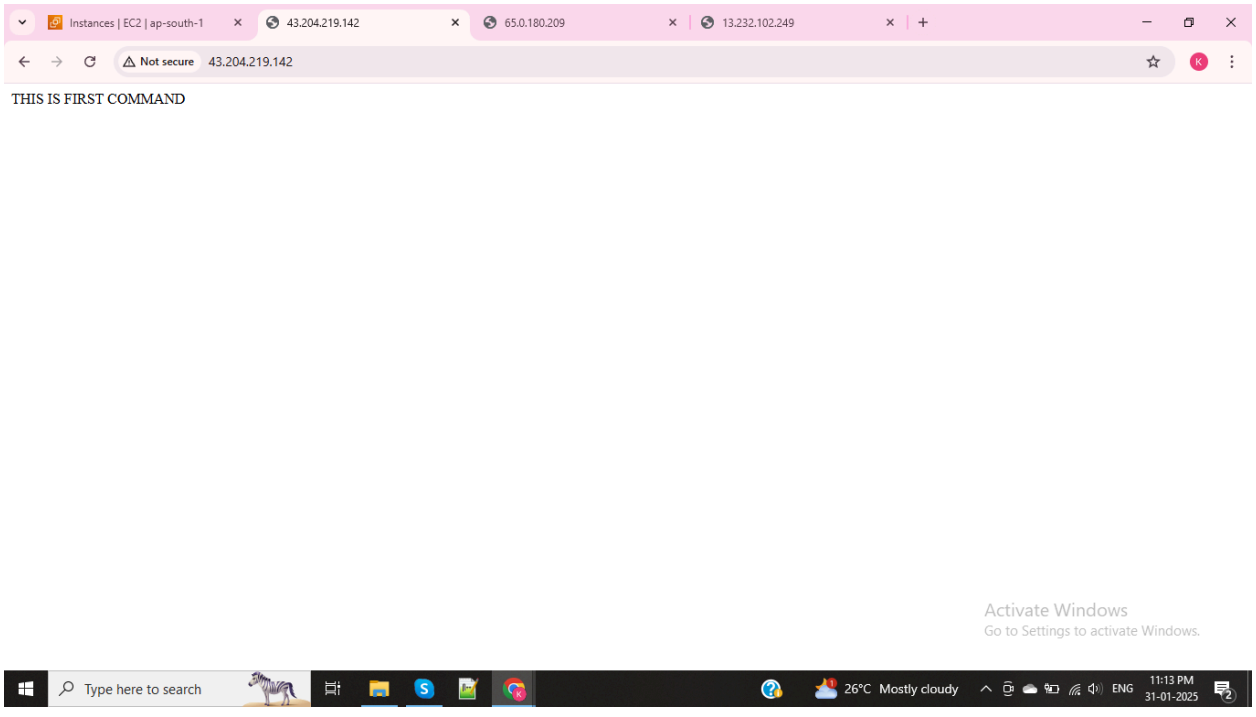
Navigate to **EC2 Dashboard** > Create EC2 Instance and add advanced script in userdata > Launch instance > After Launch instance select that Using actions > launch more like this option launched another two EC2 instance with different echo commands.



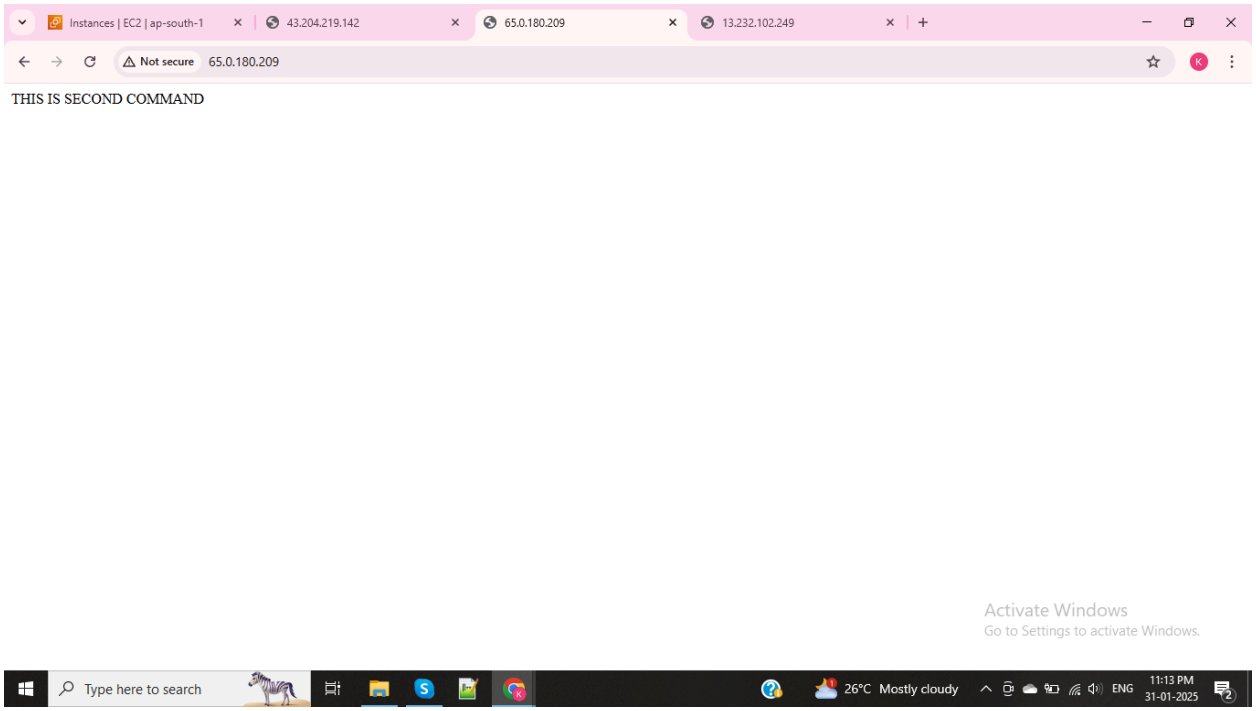
Created 3 EC2 Instance



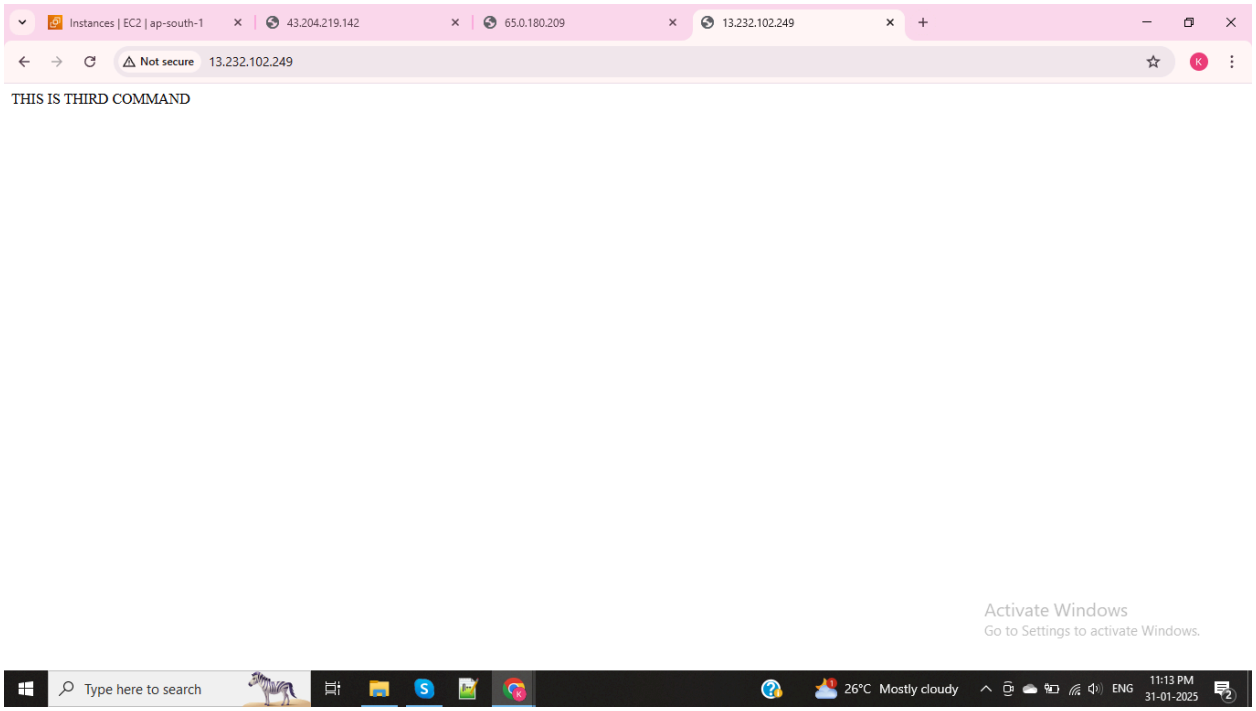
Using public ip address of each instance checking the output for Instance whether userdata is working or not.



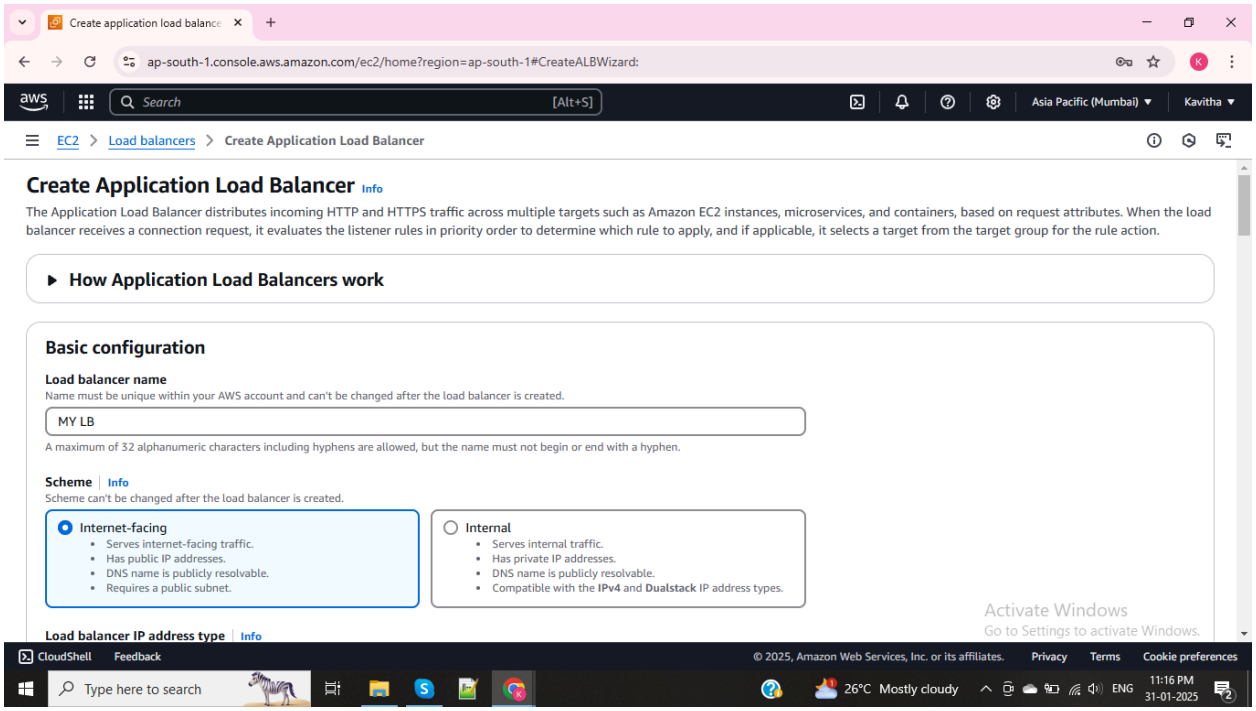
Second Instance output



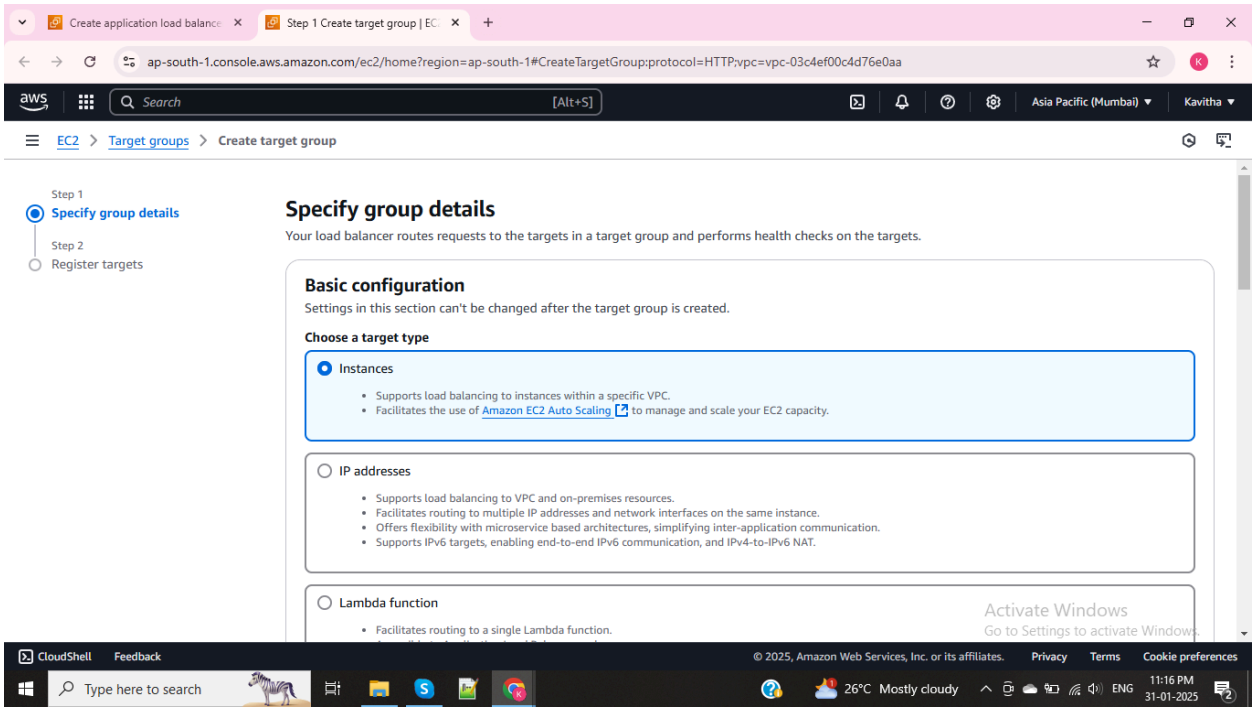
Third Instance output



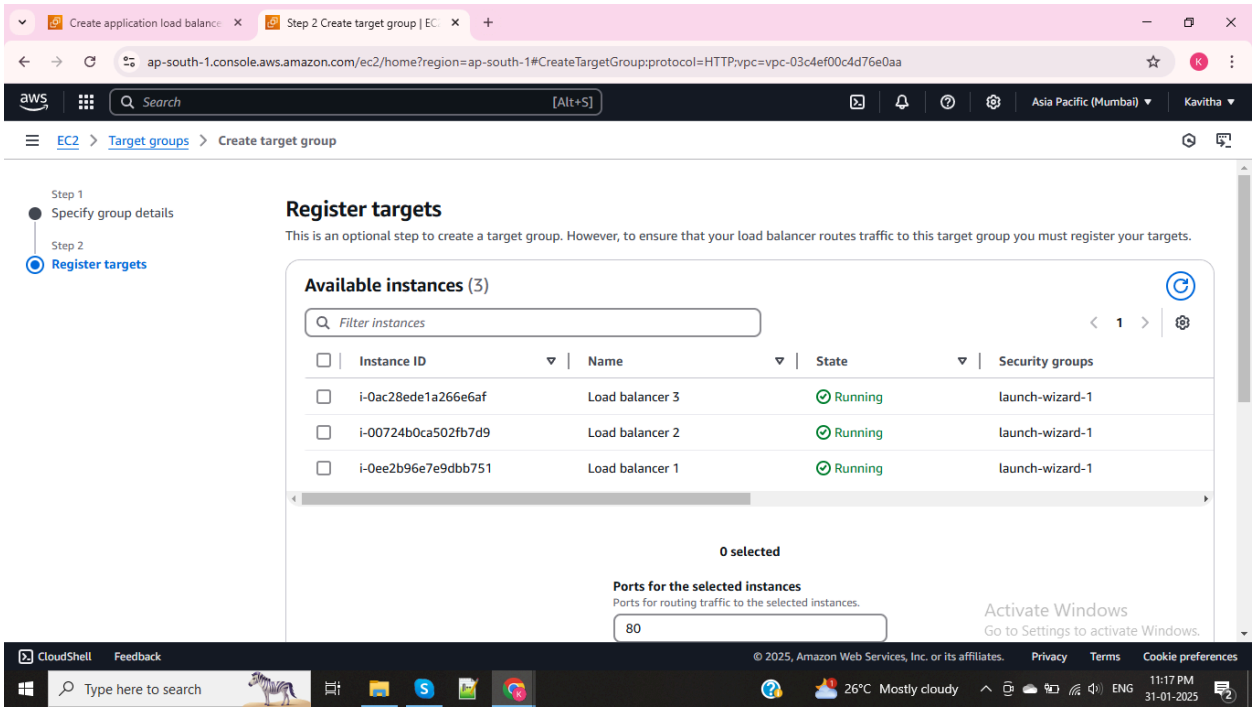
Go to EC2 Dashboard > Load balancer > click load balancer > Creating Application load balancer with basic configurations



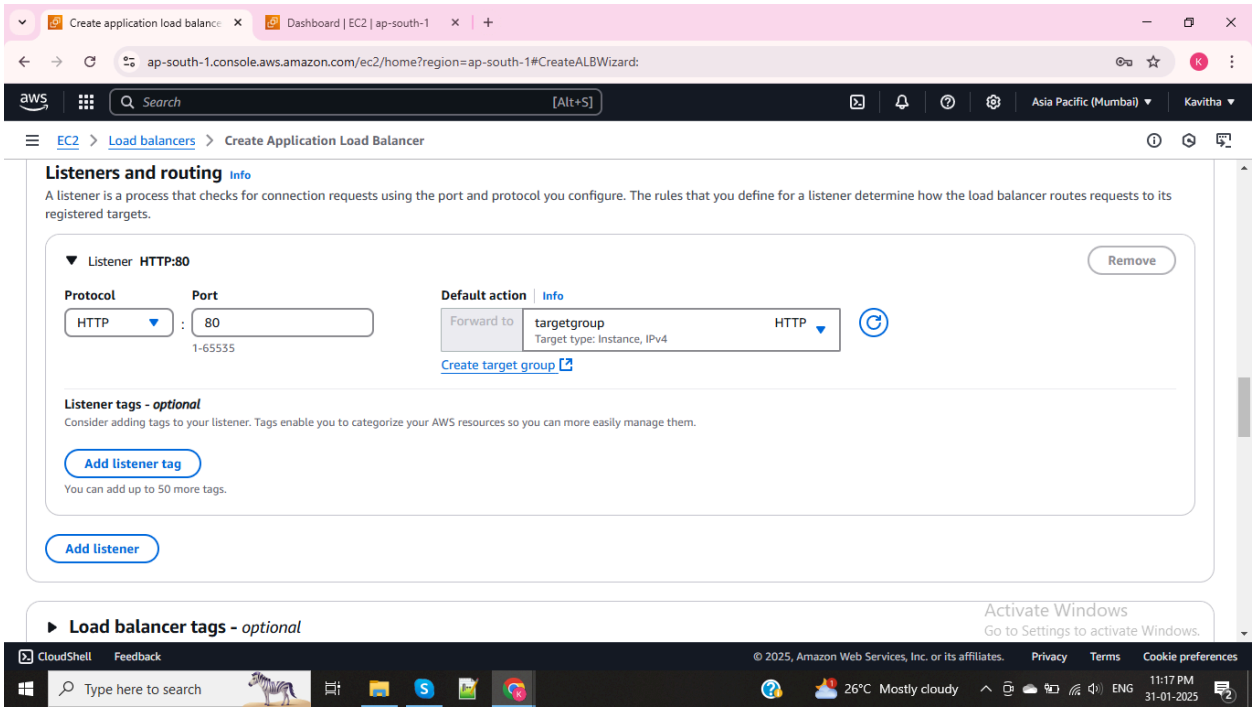
For attached load balancer created Target Group > Choose Instance target type



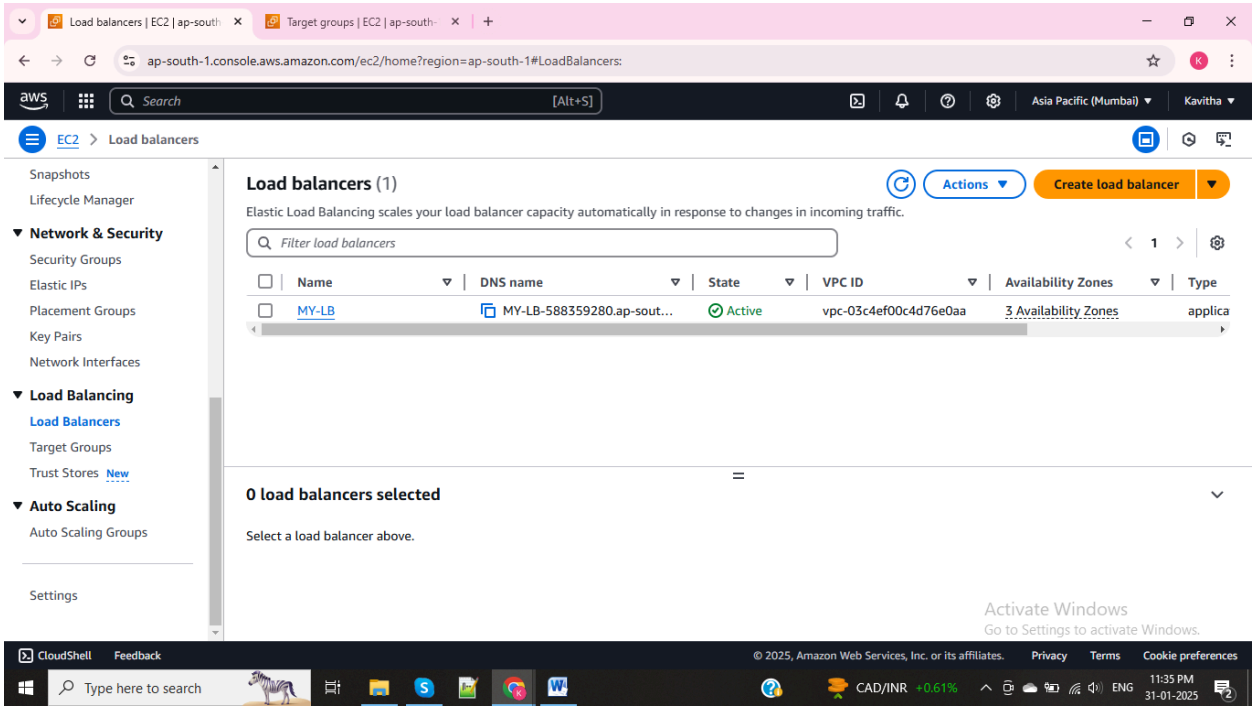
Next > In Available instance > Register the Created Instance to Application load balancer



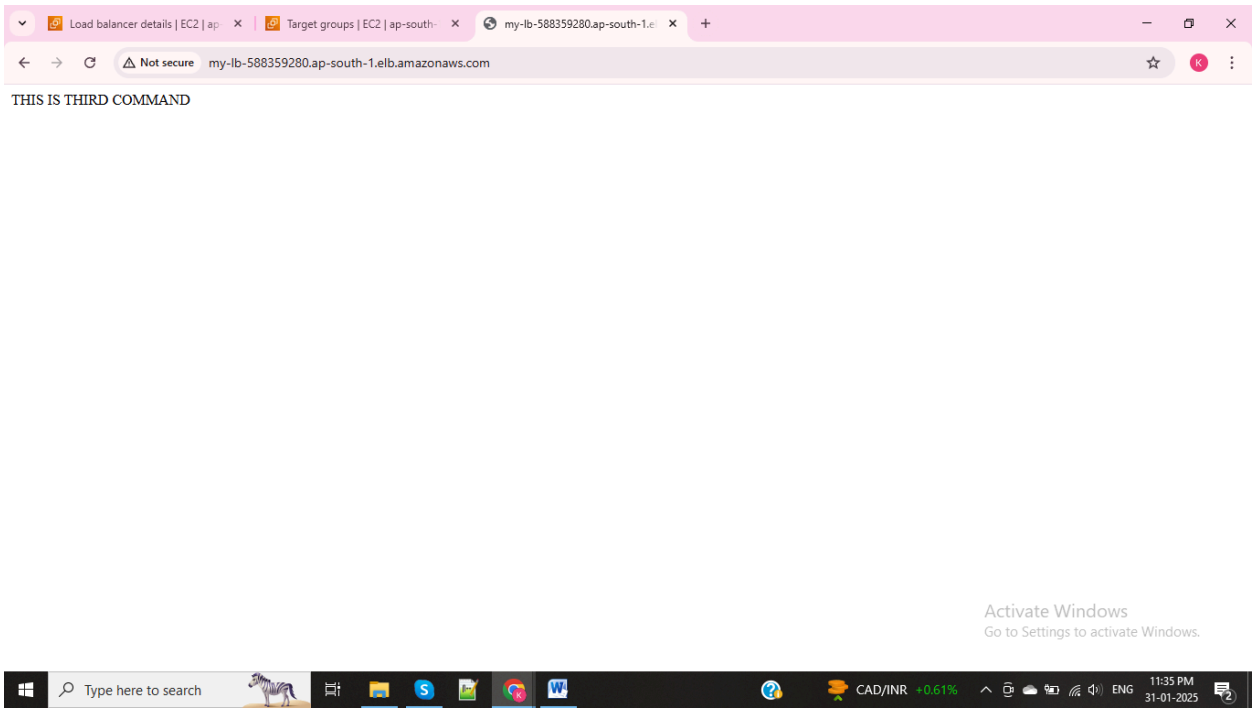
Now successfully Target Group is attached

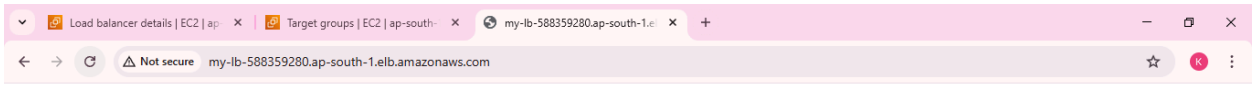


Load Balancer created

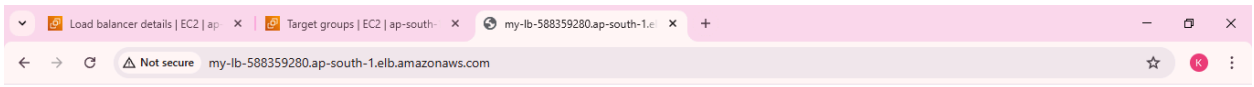


After creating load balancer it provides DNS URL (Domain Naming System). Using URL copy and paste in the browser to display output of Load balancer, it redirect to Instance 1,2 and 3 (Refresh the page to see all outputs)





THIS IS FIRST COMMAND



THIS IS SECOND COMMAND