

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

Set Up a Load Balancer in the Cloud Configure a load balancer to distribute traffic across multiple VMs hosting your web application.

Name: Harivasan

Department: ADS



# Introduction :

This Proof of Concept (POC) explores the implementation of a cloud-based Load Balancer on AWS to efficiently distribute traffic across multiple EC2 instances. Load Balancers are essential in modern cloud infrastructures, enhancing high availability, fault tolerance, and scalability for web applications. This POC showcases the fundamental setup of an AWS Load Balancer, enabling traffic distribution between two EC2 instances hosting basic web servers

# Overview:

The POC covers the following:

1. **Launching EC2 Instances**: Deploying two virtual machines (WebServer1 and WebServer2) within the AWS Free Tier.

2. **Configuring Web Servers**: Installing and setting up Apache HTTP Server on both instances to serve simple HTML web pages.

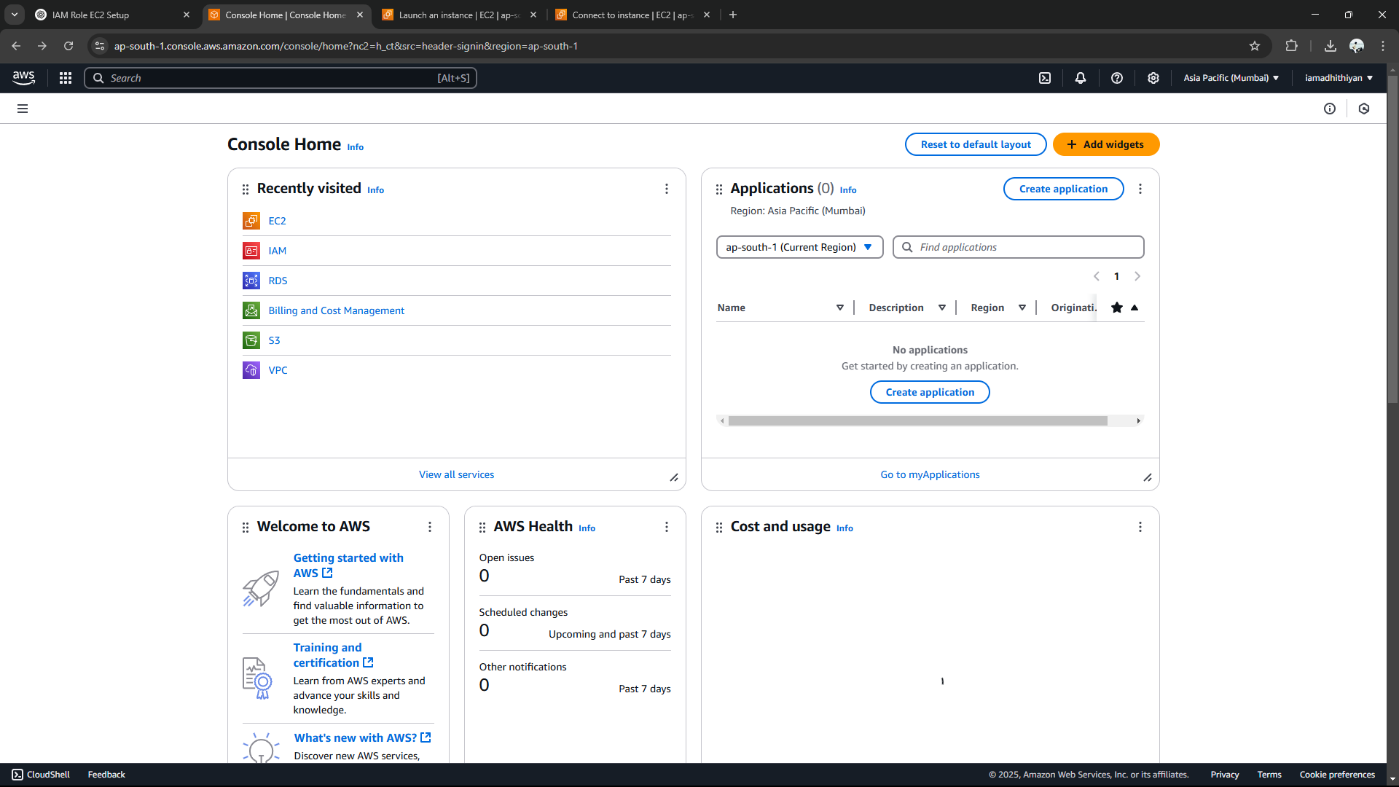
3. **Deploying a Load Balancer**: Setting up an Application Load Balancer (ALB) to evenly distribute incoming traffic across the two EC2 instances.

4. **Validating the Load Balancer**: Testing functionality by accessing the Load Balancer's DNS name and confirming traffic alternates between both servers.

**Step-by-Step Overview**

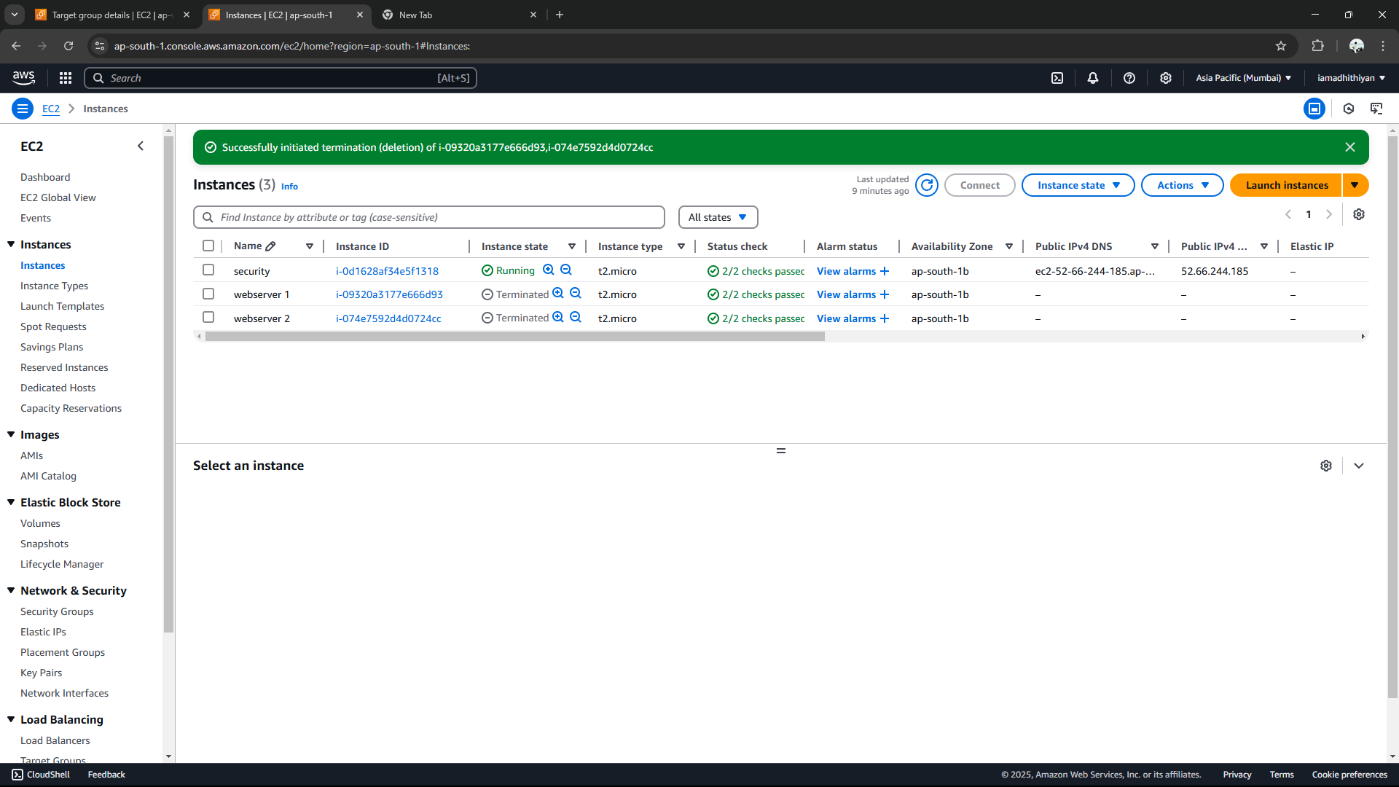
Step 1:

1. Enter your username and password to log in.



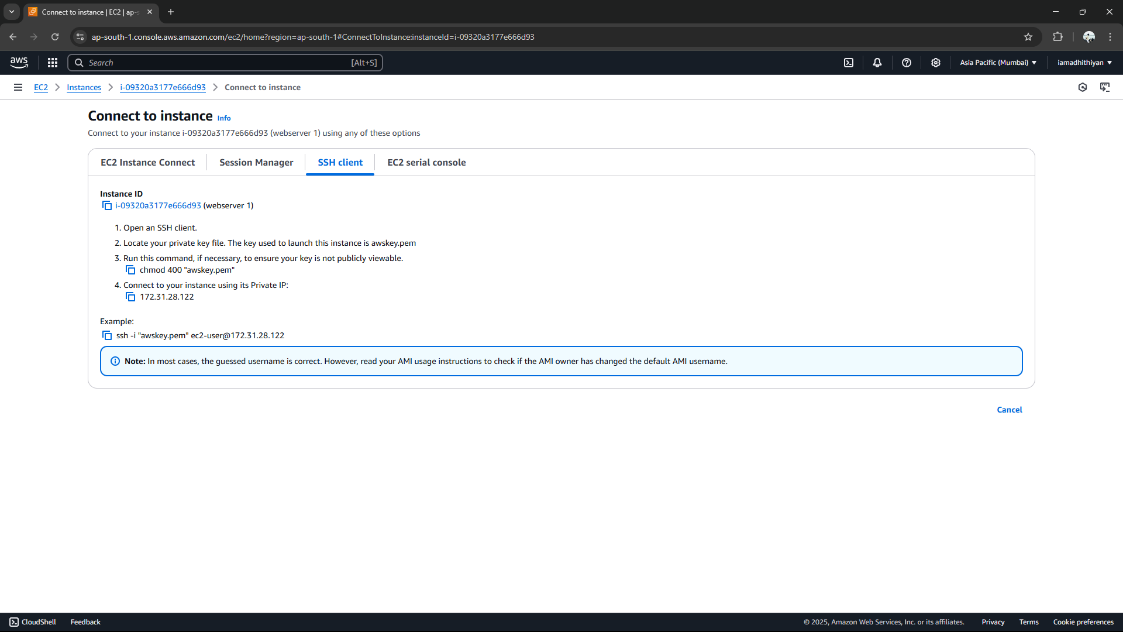
Step 2:

To set up your instances, click **Launch Instance** and enter the necessary details: name the first instance **"WebServer1"**, select **Amazon Linux 2 AMI (Free Tier eligible)** as the operating system, and choose the **t2.micro** instance type. For the **Key Pair**, either use an existing one or generate a new key pair for SSH access. In the **Network Settings**, click **Edit** and ensure **"Allow HTTP traffic from the internet"** is enabled to permit web traffic. Keep the storage size at the default **8 GB**, then click **Launch Instance**. Repeat these steps for the second instance, naming it **"WebServer2**



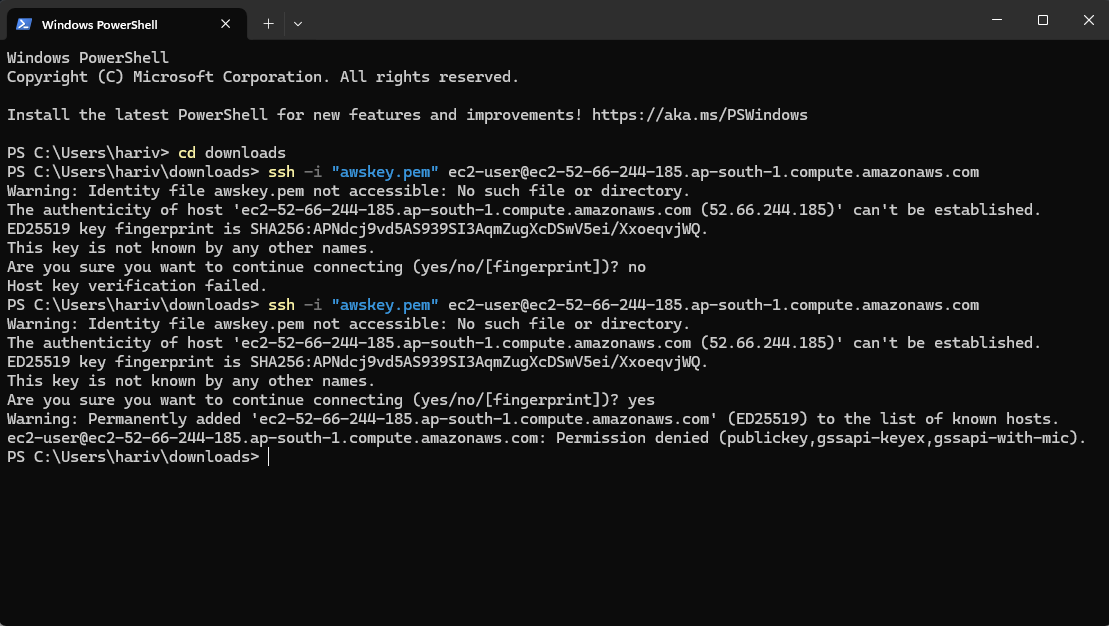
Step 3:

Click on **WebServer1**, then click **Connect**.



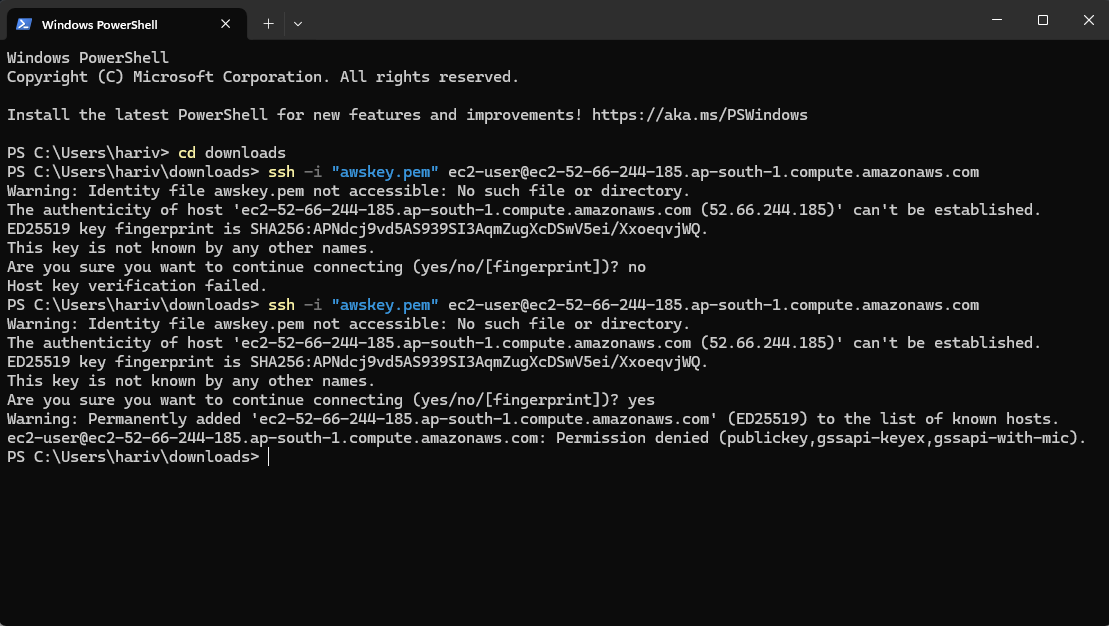
Step 3:

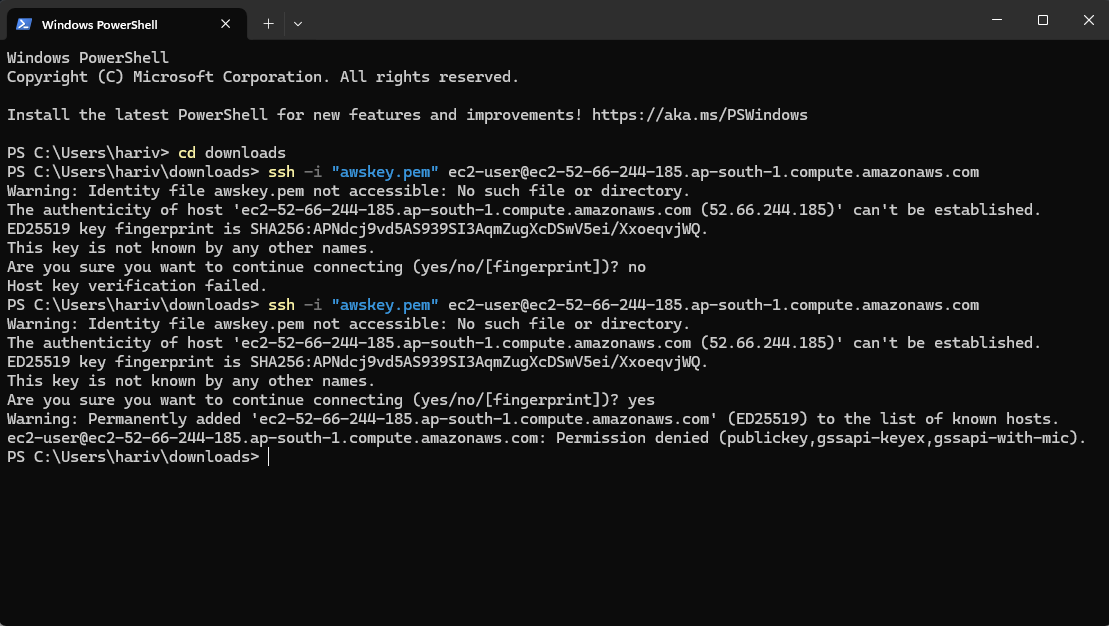
Run the following commands to install and start a web server



Step 4:

Repeat these steps for **WebServer2**

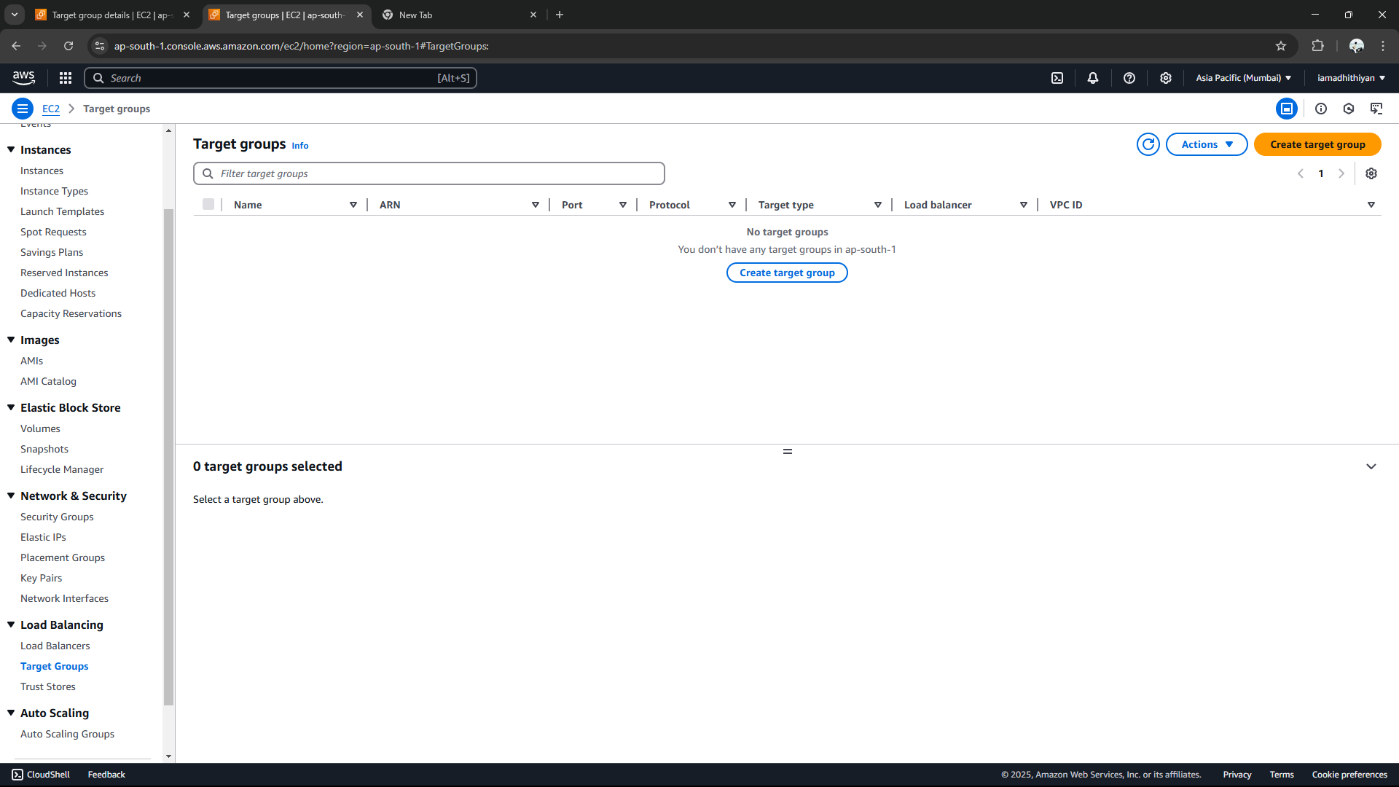




Step 5:

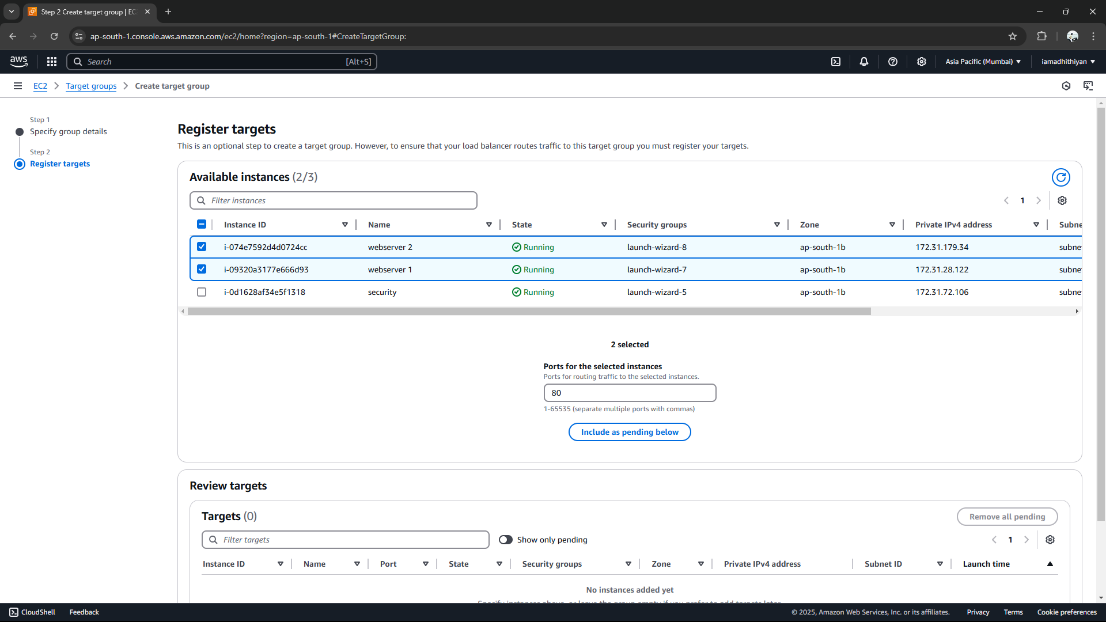
1. In the **AWS Management Console**, go to the **EC2 Dashboard**.

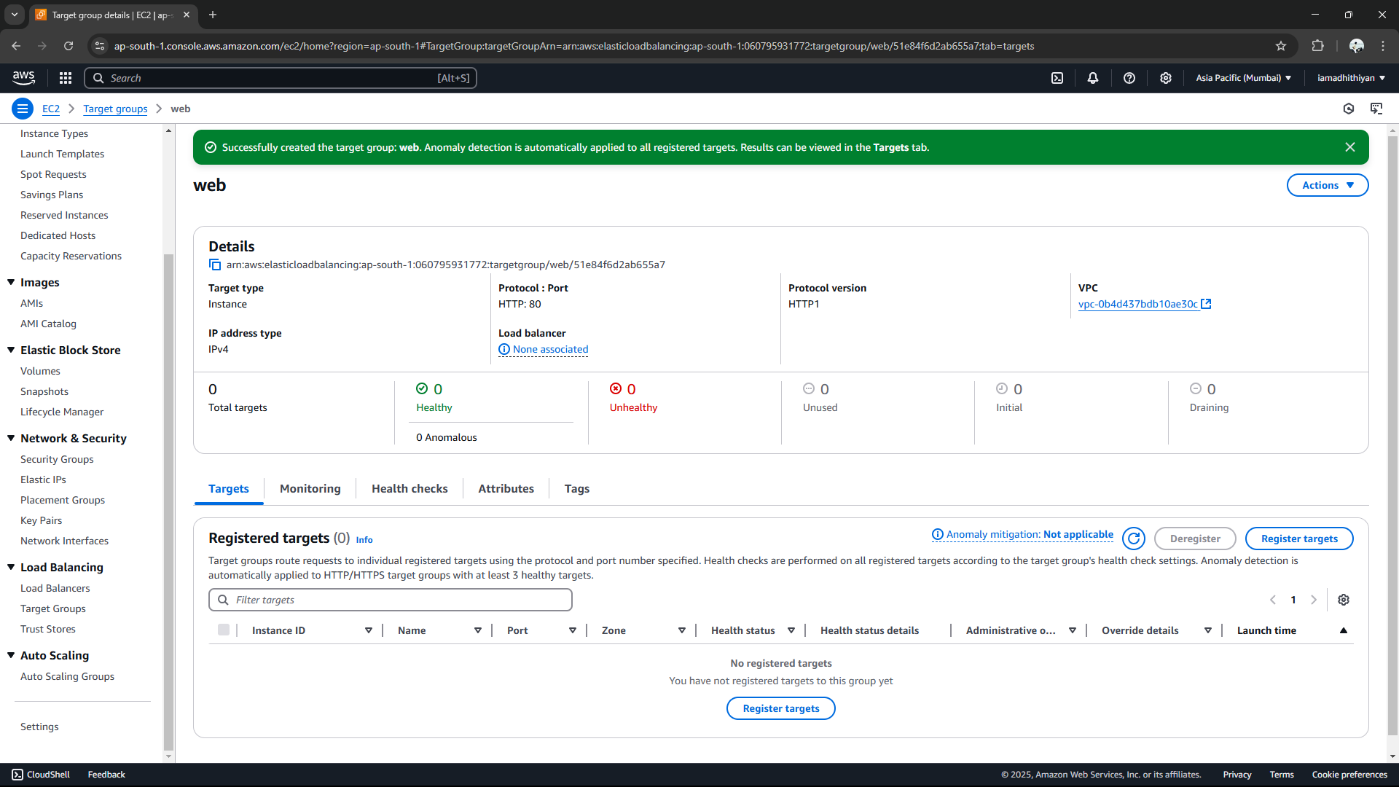
1. Click on **Target Groups** under Load Balancing.
2. Click **Create Target Group**.



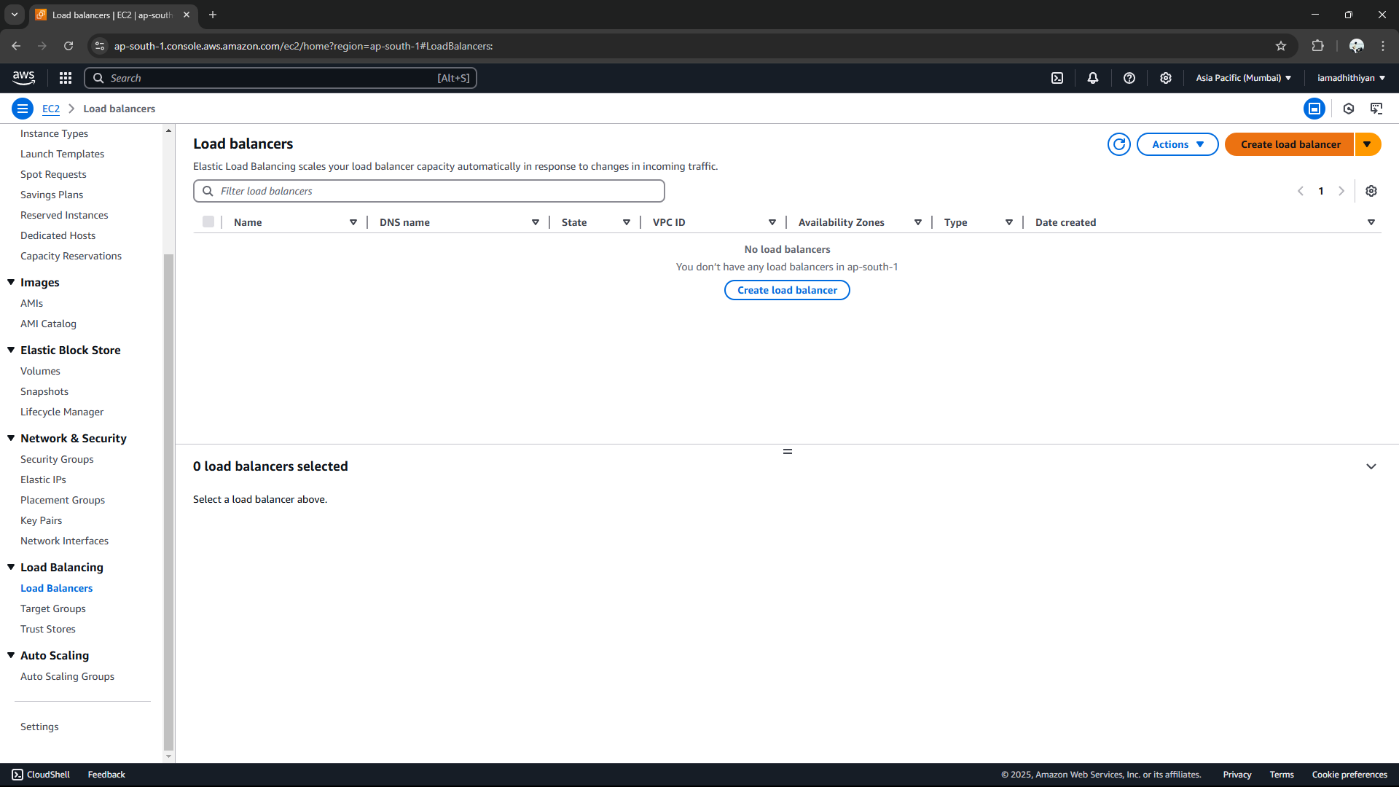
Step 6:

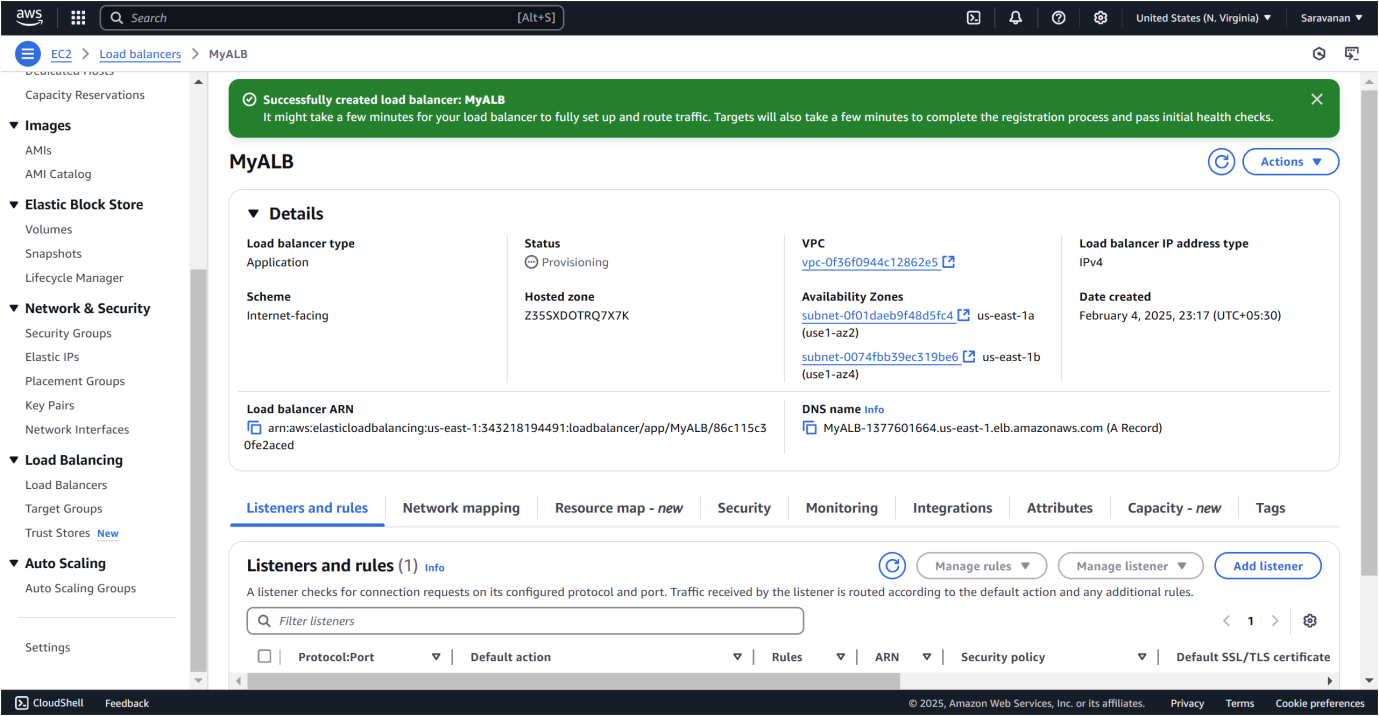
To create a target group





Step 7:

In the **EC2 Dashboard**, go to **Load Balancers** and click **Create LoadBalancer**. Select **Application Load Balancer**, name it (e.g., **"MyALB"**), set it as **Internet-facing** with **IPv4**, and enable **HTTP (port 80)**. Choose a **VPC** with two **subnets**, allow **HTTP traffic** in **Security Groups**, link a **target group** (e.g., **"MyTargetGroup"**), and click **Create**.



Step 8:

1. In the **AWS Console**, go to **Load Balancers** and select yours.
2. Copy the **DNS name** from the **Description** tab.
3. Open it in a browser and refresh the page.
4. Ensure traffic alternates between **WebServer1** and **WebServer2**, displaying their respective messages.

This confirms proper traffic distribution and high availability.

# Expected Outcome:

1. Deploy and configure two **EC2 instances** running **Amazon Linux 2**, each hosting a simple web server with distinct content.
2. Set up an **Application Load Balancer** to evenly distribute traffic between the two instances.
3. Validate the Load Balancer by accessing its **DNS name** and confirming traffic alternates between both servers.
4. Gain insight into the role of **Load Balancers** in enhancing **high availability** and **fault tolerance** for web applications.