# 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:Shahana.M.S Department: ADS



**Introduction**

A load balancer is a crucial component of cloud architecture that distributes incoming traffic across multiple virtual machines (VMs) to ensure high availability, reliability, and performance of web applications. By setting up a cloud-based load balancer, organizations can prevent server overload, improve response times, and provide fault tolerance. Cloud providers like AWS, Azure, and Google Cloud offer managed load-balancing services that simplify deployment and management.

**Overview**

Load balancing enhances application scalability by efficiently distributing requests among multiple instances. It ensures that no single server bears excessive traffic, reducing the risk of downtime and improving user experience. Cloud platforms provide various types of load balancers, including:

* **Application Load Balancer (ALB)** – Routes traffic based on HTTP/HTTPS requests and application-layer parameters.
* **Network Load Balancer (NLB)** – Operates at the transport layer (TCP/UDP) for handling high-speed traffic.
* **Classic Load Balancer (CLB)** – Provides basic load balancing at both the application and network layers.

Configuring a load balancer involves setting up target instances, defining routing rules, and ensuring firewall settings allow traffic distribution.

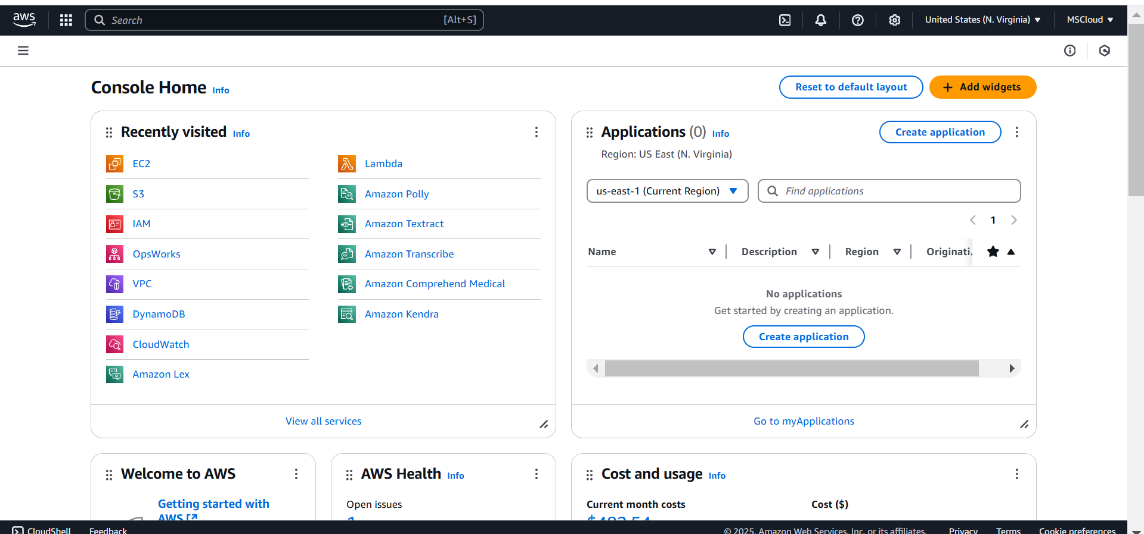
**Objective**

* To configure a cloud-based load balancer (AWS Elastic Load Balancer, Azure Load Balancer, or GCP Load Balancing).
* To distribute traffic efficiently across multiple VMs hosting a web application.
* To ensure high availability and fault tolerance for applications.
* To test and verify load balancing by monitoring traffic distribution and failover mechanisms.

# Step-by-Step Overview

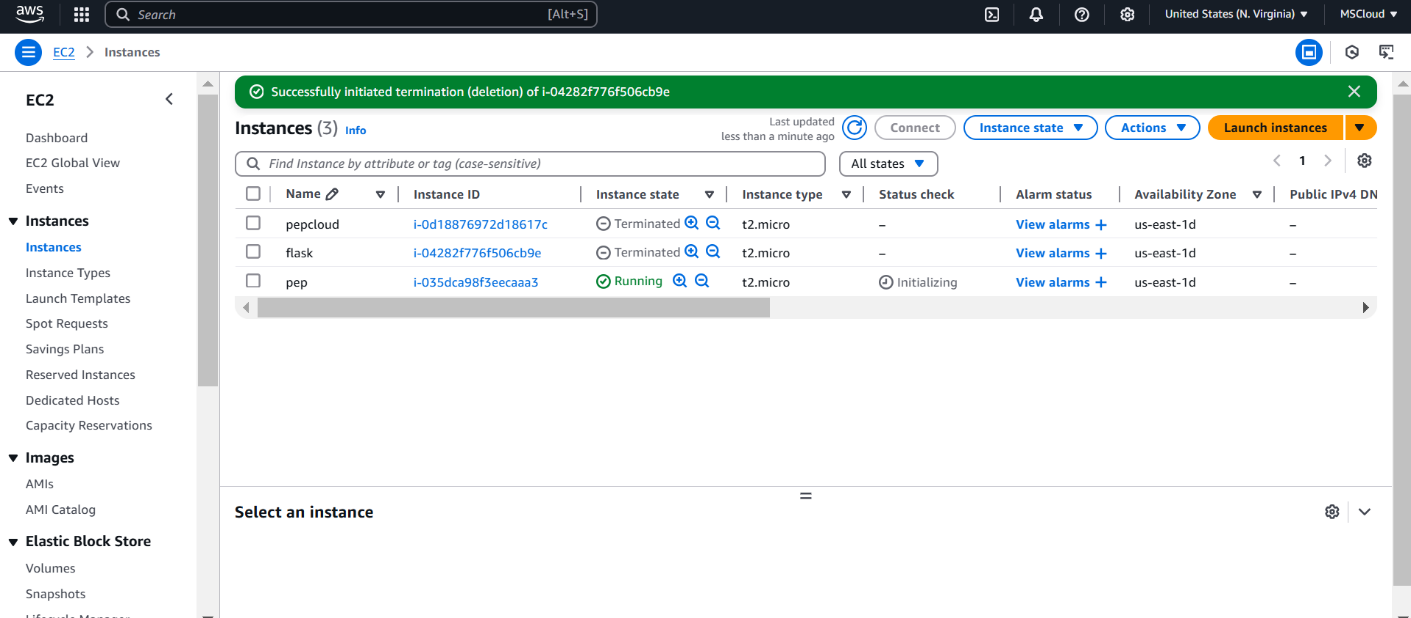
## Step 1:

1. Go to [AWS Management Console](https://aws.amazon.com/console/).
2. Enter your username and password to log in.



## Step 2:

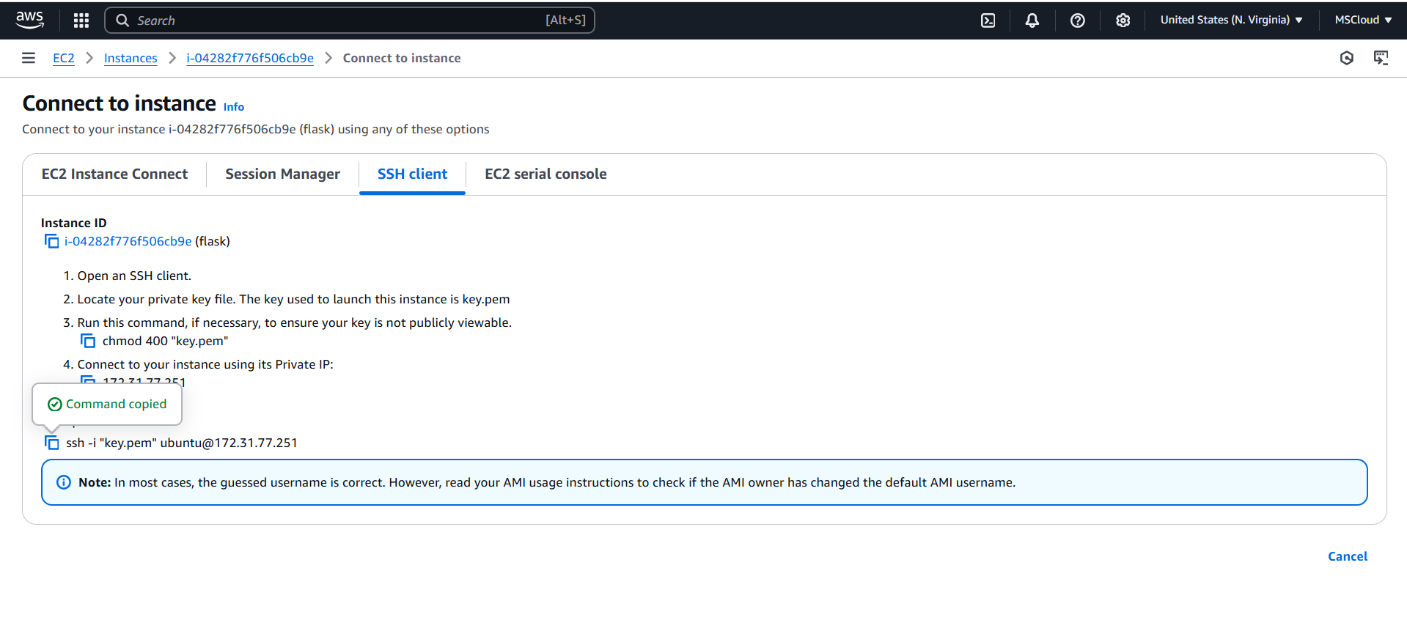
## Create instance , and and get the Amazon Linux , free tier and allow HTTP.



## Step 3:

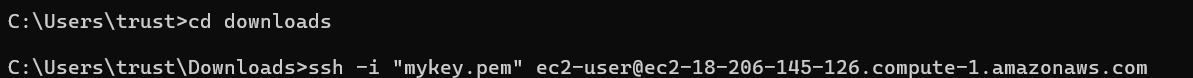
Click on **WebServer1**, then click **Connect**.

Use the instructions under **SSH client** to connect to your instance via terminal.



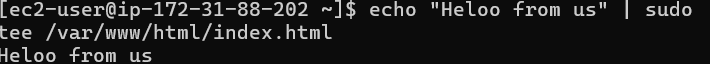
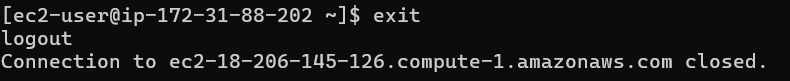
## Step 3:

Run the following commands to install and start a web server



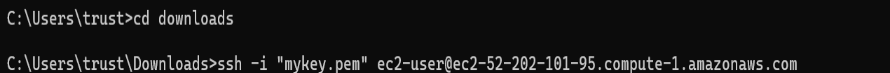




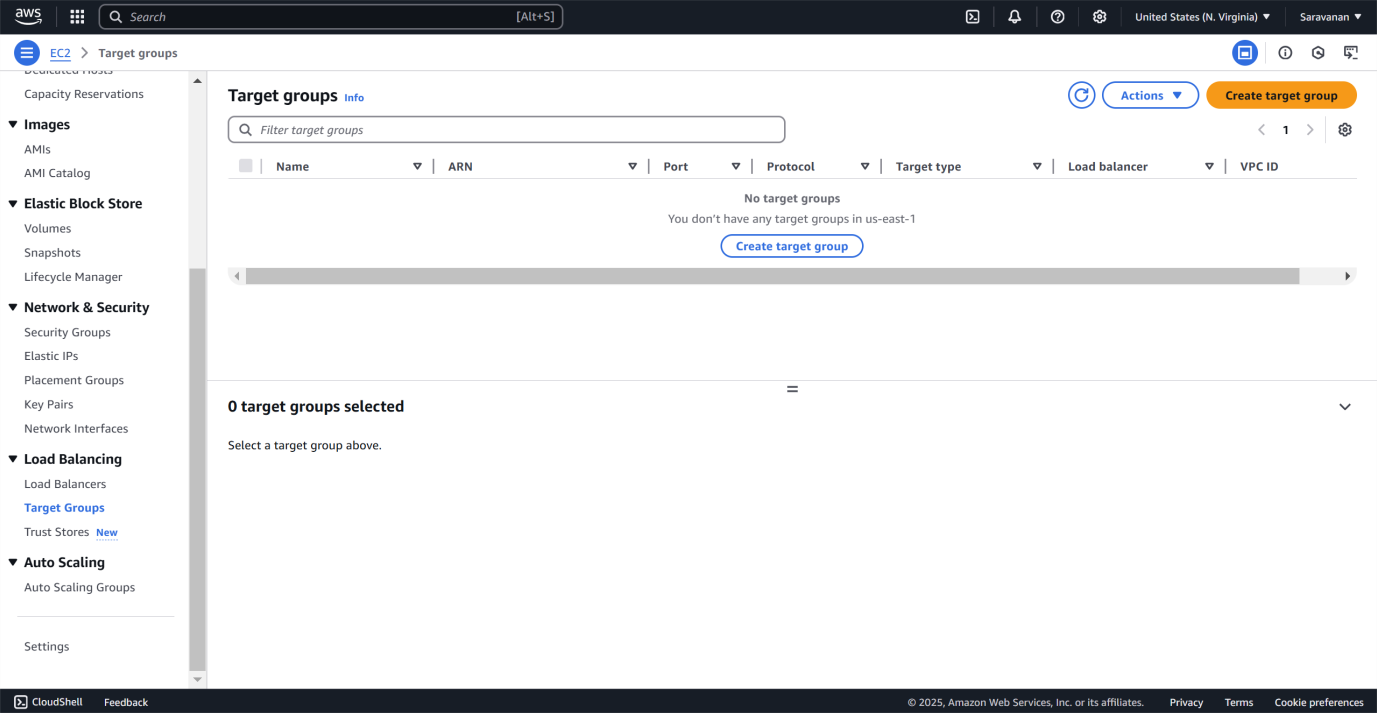


## Step 4:

Repeat these steps for **WebServer2** but change the message in the last command to:

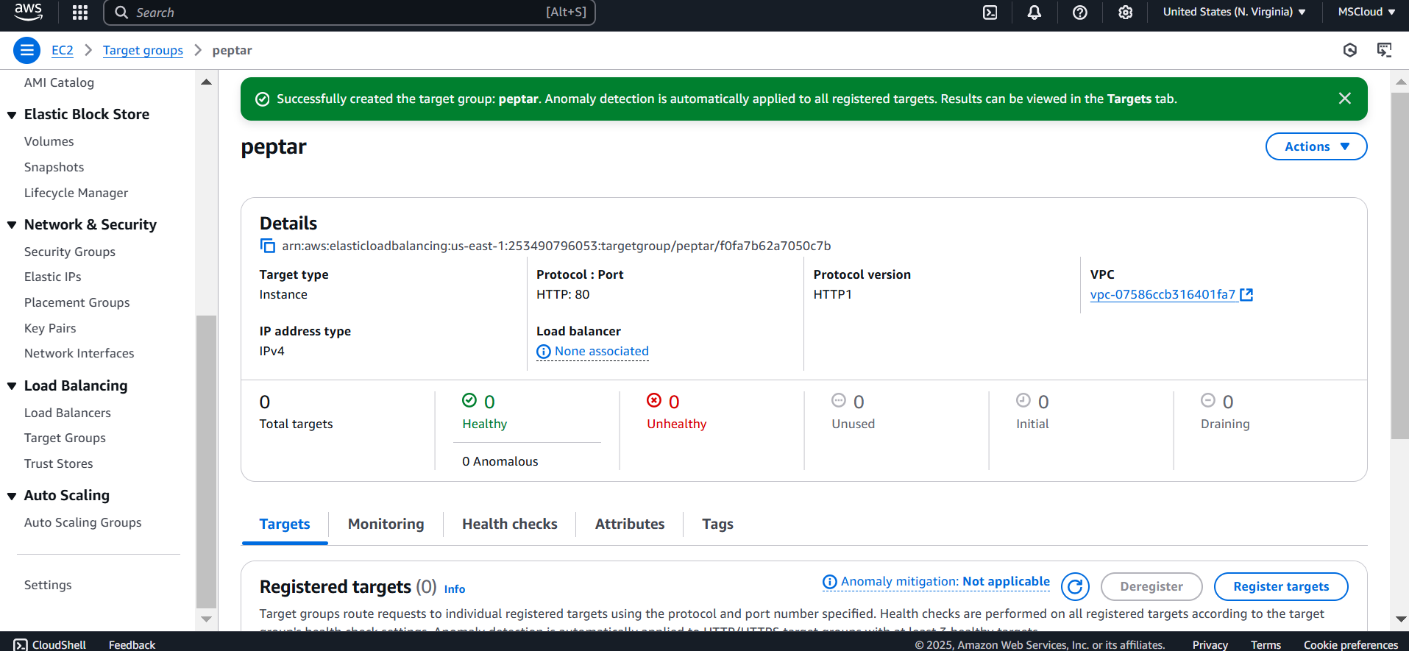
Step 5:

1. In the **AWS Management Console**, go to the **EC2 Dashboard**.
2. Scroll down and click on **Target Groups** under "Load Balancing."
3. Click **Create Target Group**.



To create a target group:

* select **Instances** as the target type, name it.
* set the **Protocol** to HTTP and **Port** to 80, and choose the same VPC as your EC2 instances (usually the default VPC). Keep the **Health Check Path** as / to verify the web server's status.
* Click **Next**, select both WebServer1 and WebServer2 under "Register Targets," click **Include as pending below**, and then create the target group.

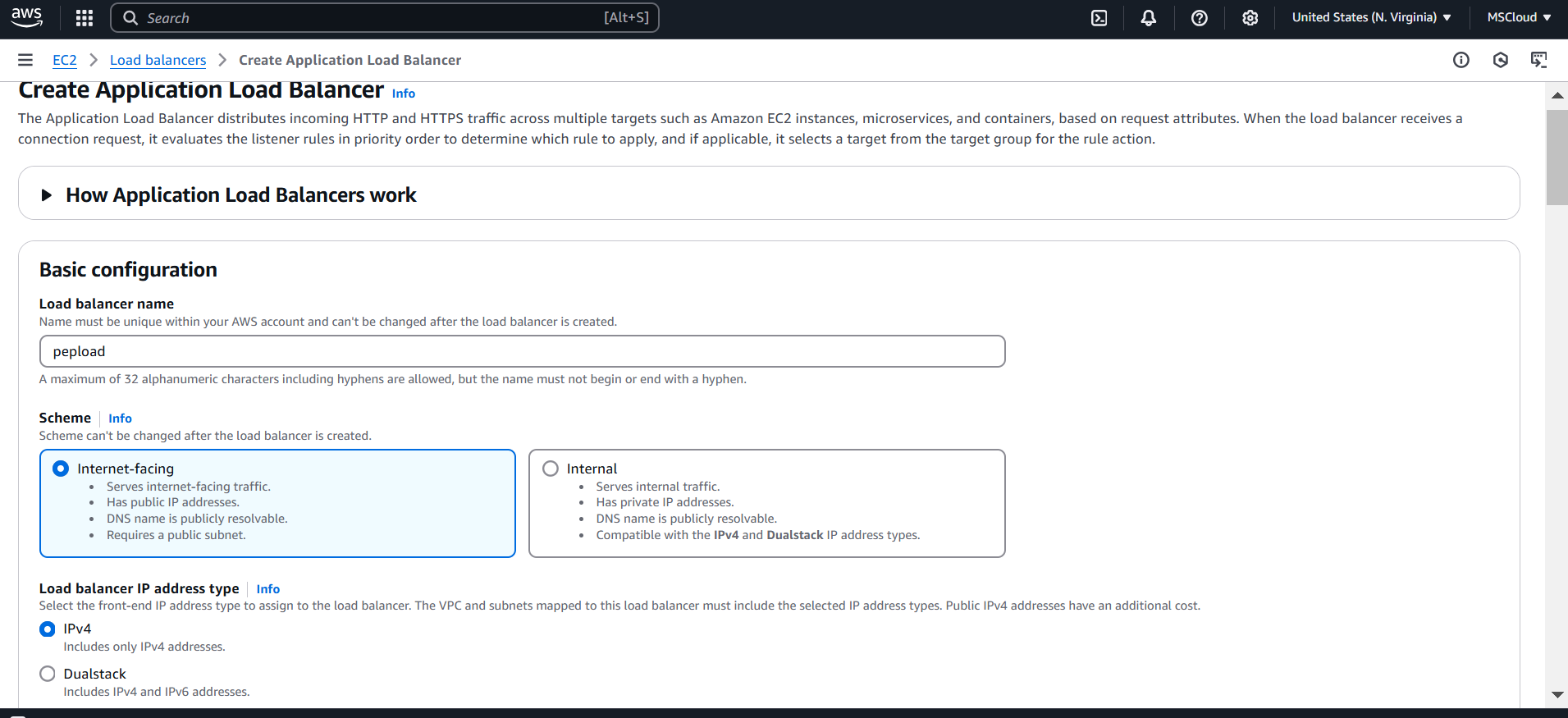


## Step 7:

In the EC2 Dashboard, go to **Load Balancers** under "Load Balancing" and click **Create Load Balancer**. Select **Application Load Balancer (free tier eligible)** and configure it: name it (e.g., "MyALB"), set the **Scheme** to Internet-facing, **IP Address Type** to IPv4, and ensure the listener is HTTP on port 80. Select the VPC and at least two subnets for high availability. Skip the security settings since this is HTTP. On the **Security Groups** page, choose or create a security group that allows HTTP traffic. On the **Routing** page, select

the previously created target group (e.g., "MyTargetGroup") and click

**Create Load Balancer**.



## Step 8:

To verify the functionality of your Load Balancer:

1. Go to the **Load Balancers** section in the AWS Management Console.
2. Select your Load Balancer and find its **DNS name** under the

**Description** tab.

1. Copy the DNS name and open it in your browser.
2. Refresh the page to confirm that traffic is being alternated between the two EC2 instances. You should see the messages **"Hello from WebServer1!"** and **"Hello from WebServer2!"** displayed alternately.

This confirms that the Load Balancer is correctly distributing traffic and ensuring high availability.

**Outcome**

* Successful deployment of a load balancer to distribute traffic across multiple VMs.
* Improved scalability and high availability of the web application.
* Enhanced fault tolerance, ensuring uninterrupted service even if one VM fails.
* Better performance and load distribution across instances.