

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: K Swathika

DEPARTMENT: AML

INTRODUCTION:

An Elastic Load Balancer (ELB) is an AWS-managed service that automatically distributes incoming network traffic across multiple targets, such as EC2 instances, containers, and IP addresses, in different Availability Zones. It ensures high availability, fault tolerance, and scalability of applications. In this PoC, we'll demonstrate the basic setup of an AWS Load Balancer, allowing traffic to be distributed between two EC2 instances running simple web servers.

IMPORTANCE:

1. High Availability & Fault Tolerance - Spreads traffic across multiple instances to prevent a single point of failure and supports multiple Availability Zones for redundancy.
2. Scalability - Automatically scales with the incoming traffic load.
3. Security - Supports SSL/TLS encryption to secure traffic also works with AWS WAF (Web Application Firewall) to block malicious traffic.
4. Health Monitoring - Continuously checks the health of backend instances and automatically stops routing traffic to unhealthy instances.
5. Improved Performance - Supports content-based routing (Layer 7) for optimized request handling.
6. Integration with AWS Services - Works with Auto Scaling to dynamically add/remove instances. Also, supports AWS CloudWatch for monitoring and logging.

STEP BY STEP OVERVIEW:

Step 1: CREATE INSTANCE

- Login into your AWS console and navigate to the EC2 dashboard.

- Click on Launch Instance and specify name the first instance, select Amazon Linux 2 AMI as the OS, and choose the t2.micro instance type. For the Key Pair, either select an existing one or create a new key pair to use for SSH access. Under Network Settings, ensure "Allow HTTP traffic from the internet" is checked to enable web traffic. Keep the storage size at the default 8 GB, then click Launch Instance.
- Repeat the same steps for the second instance.

The screenshot shows the AWS Management Console's EC2 Instances page. On the left is a navigation menu with options like Dashboard, EC2 Global View, Events, and a list of instance types. The main area displays a table of two instances: 'my-server-2' (ID: i-0864340f33de5e14a) and 'my-server-1' (ID: i-090842e4ae59e2301). Both are t2.micro instances in the 'Running' state. Above the table are filters and a search bar. Below the table is a 'Select an instance' section.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
my-server-2	i-0864340f33de5e14a	Running	t2.micro	Initializing	View alarms +	us-east-1d	ec2-54-87-9-50.compute-1.amazonaws.com
my-server-1	i-090842e4ae59e2301	Running	t2.micro	Initializing	View alarms +	us-east-1d	ec2-54-87-9-50.compute-1.amazonaws.com

- Click on my-server-1, then click Connect. Use the instructions under SSH client to connect to your instance via terminal.

The screenshot shows the 'Connect to instance' page in the AWS Management Console for instance 'my-server-1' (ID: i-090842e4ae59e2301). The 'SSH client' tab is selected, showing a list of instructions for connecting via SSH. A green box with a checkmark indicates that the command has been copied. Below the instructions is a terminal window showing the command: `ssh -i "newpair.pem" ec2-user@ec2-54-87-9-50.compute-1.amazonaws.com`. A note at the bottom states: "Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username."

Connect to instance Info

Connect to your instance i-090842e4ae59e2301 (my-server-1) using any of these options

EC2 Instance Connect | Session Manager | **SSH client** | EC2 serial console

Instance ID
i-090842e4ae59e2301 (my-server-1)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is newpair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
`chmod 400 "newpair.pem"`
4. Connect to your instance using its Public DNS:
`ec2-54-87-9-50.compute-1.amazonaws.com`

✓ Command copied

```
ssh -i "newpair.pem" ec2-user@ec2-54-87-9-50.compute-1.amazonaws.com
```

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Step 2: INSTALL & RUN THE WEB-SERVER

- Run the following command to install and run the web-server.

```
PS C:\Users\Arulldhas> cd downloads
PS C:\Users\Arulldhas\downloads> ssh -i "sample.pem" ec2-user@ec2-13-201-20-140.ap-south-1.compute.amazonaws.com
```

```
~/m/'
[ec2-user@ip-172-31-9-33 ~]$ sudo yum update -y
```

```
[ec2-user@ip-172-31-9-33 ~]$ sudo yum install httpd -y
```

```
[ec2-user@ip-172-31-9-33 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-9-33 ~]$ sudo systemctl enable httpd
```

- Repeat the same steps for the second instance and add the following command.

```
[ec2-user@ip-172-31-9-33 ~]$ echo "hello from server-2" | sudo tee /var/www/html/index.html
hello from server-2
[ec2-user@ip-172-31-9-33 ~]$ |
```

Step 3: CREATE TARGET GROUP

- In the AWS Management Console, go to the EC2 Dashboard and scroll down and click on Target Groups under "Load Balancing."
- Click Create Target Group.
- To create a target group, select Instances as the target type, name it and set the Protocol to HTTP and Port to 80, and choose the VPC (usually the default VPC). Keep the Health Check Path Click Next, select both instances under "Register Targets," and then create the target group.

The screenshot displays the AWS Management Console interface for the 'my-target' Target Group. The left sidebar shows the navigation menu with 'Load Balancing' selected. The main content area shows the 'Details' tab for the target group 'my-target'. A green success message at the top states: 'Successfully created the target group: my-target. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab.'

The 'Details' section includes the following information:

- Target type:** Instance
- IP address type:** IPv4
- Protocol : Port:** HTTP: 80
- Protocol version:** HTTP1
- VPC:** vpc-02a6d6d355b6ee357
- Load balancer:** None associated

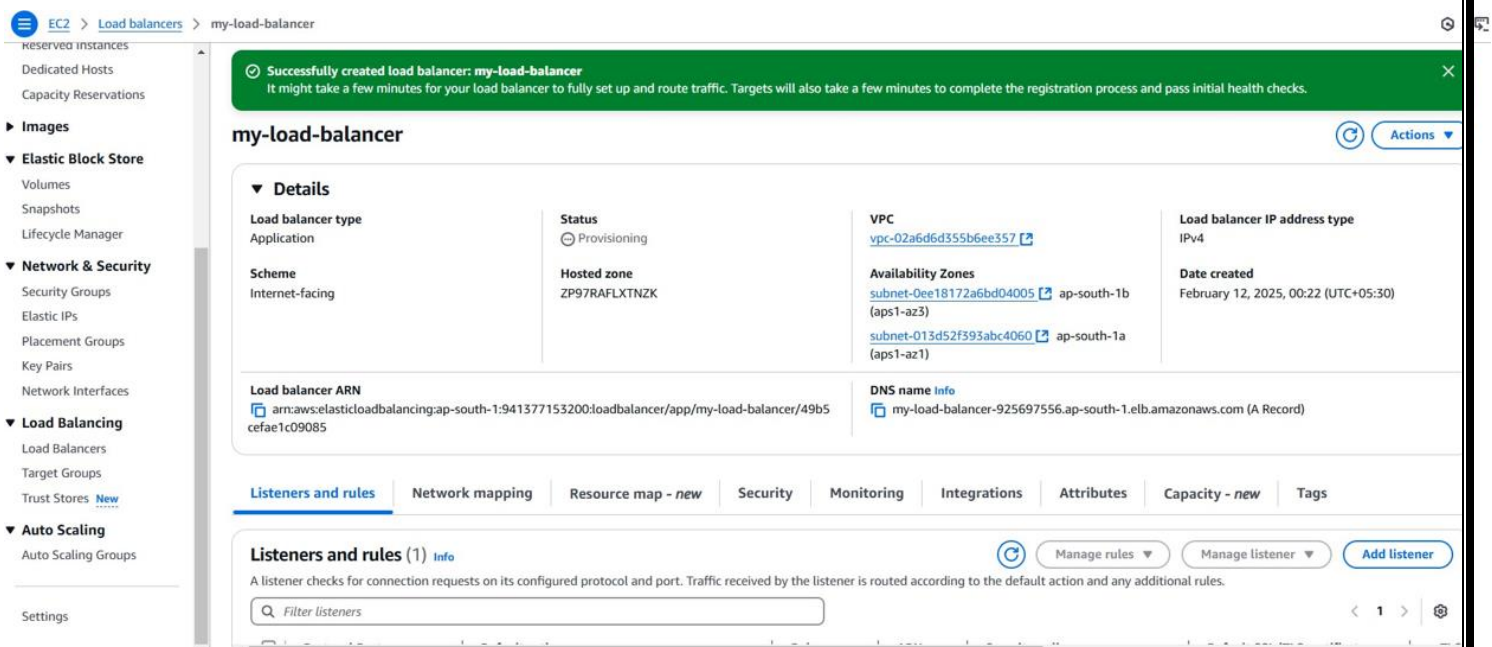
Below the details, there are status indicators for the target group:

- Total targets:** 0
- Healthy:** 0
- Unhealthy:** 0
- Unused:** 0
- Initial:** 0
- Draining:** 0

The 'Targets' tab is selected, showing a list of registered targets. The 'Registered targets (0)' section includes a search bar and a filter button. The 'Anomaly mitigation: Not applicable' status is also visible.

Step 4: CREATE LOAD BALANCER

- 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 & 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 and click Create Load Balancer.



Step 5: TESTING THE FUNCTIONALITY

To verify the functionality of your Load Balancer:

- Go to the Load Balancers section in the AWS Management Console. Select your Load Balancer and find its DNS name under the Description tab.

- Copy the DNS name and open it in your browser. Refresh the page to confirm that traffic is being alternated between the two EC2 instances.
- You should see the messages "Hello from my-server1" and "Hello from my-server-2" displayed alternately. This confirms that the Load Balancer is correctly distributing traffic and ensuring high availability.

CONCLUSION:

By completing this PoC, you will:

- Launch and configure two EC2 instances with Amazon Linux 2, each hosting a simple web server with unique content.
- Create and configure an Application Load Balancer to distribute incoming traffic between the two EC2 instances.
- Verify the functionality of the Load Balancer by accessing the DNS name and observing traffic alternation between the two web servers.
- Understand the importance of Load Balancers in ensuring high availability and fault tolerance for web applications.