Lesson 03 Demo 07

Configuring an Application Load Balancer

Objective: To configure an Application Load Balancer in AWS to distribute traffic across multiple EC2 instances for load balancing and redundancy

Tools required: AWS Management Console, AWS EC2, and web browser

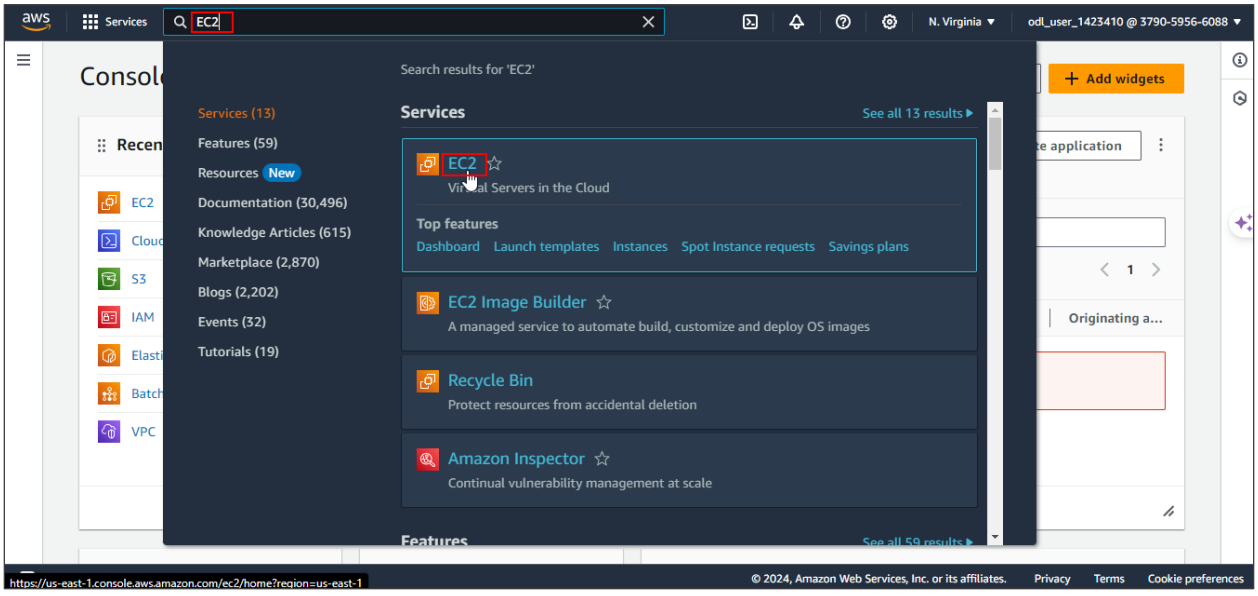
Prerequisites: None

Steps to be followed:

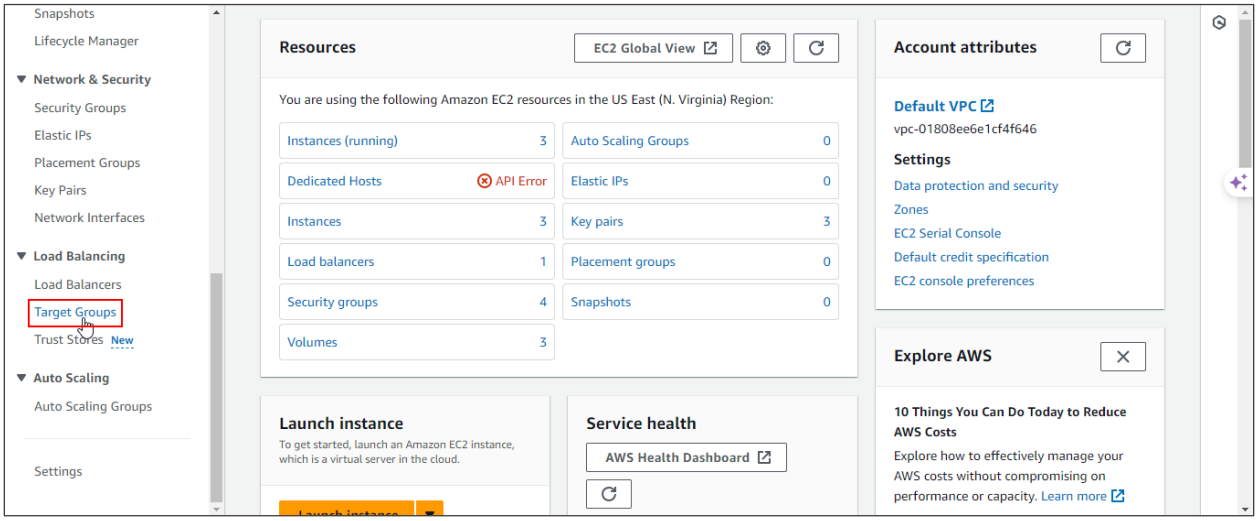
1. Create a target group
2. Launch EC2 instances
3. Configure the target group
4. Create a Load Balancer
5. Test the Load Balancer

Step 1: Create a target group

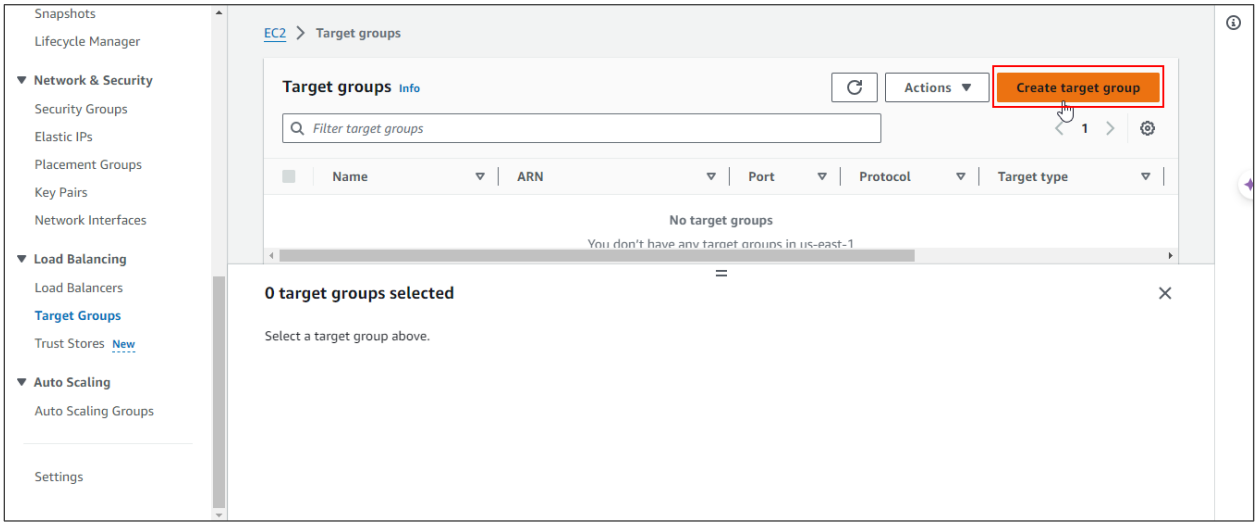
1.1 Navigate to the AWS console home dashboard, search for and click on **EC2**



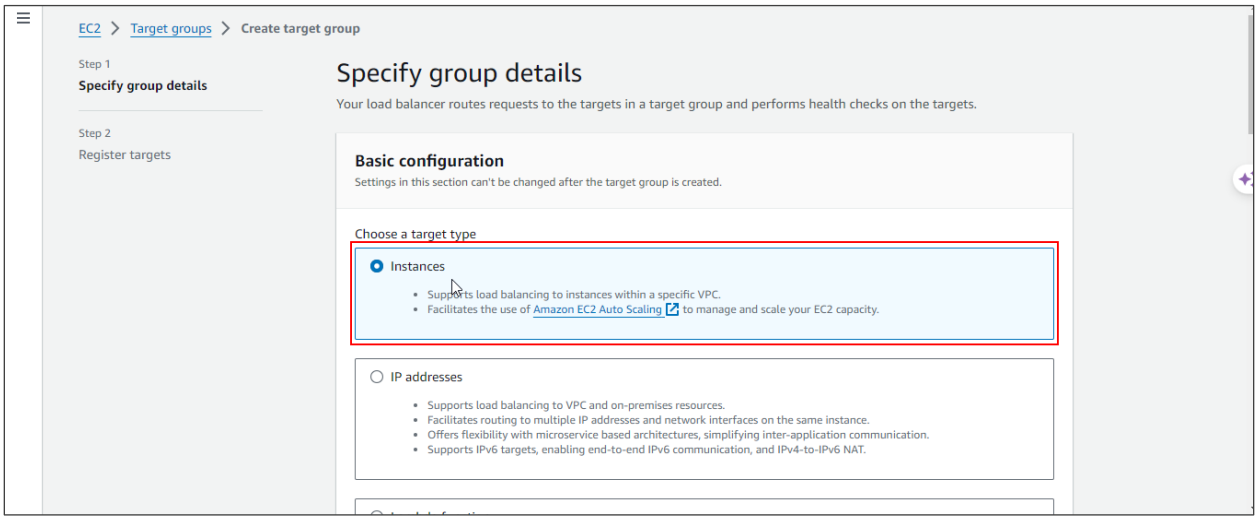
1.2 Navigate to the **Load Balancing** section and click on **Target Groups**

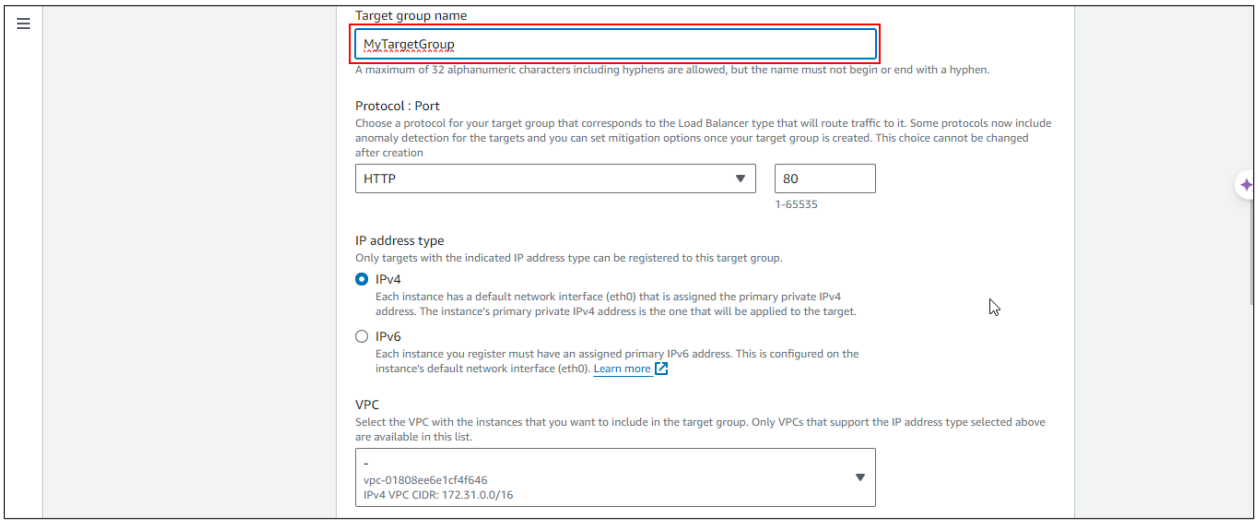


1.3 Click on **Create target group**

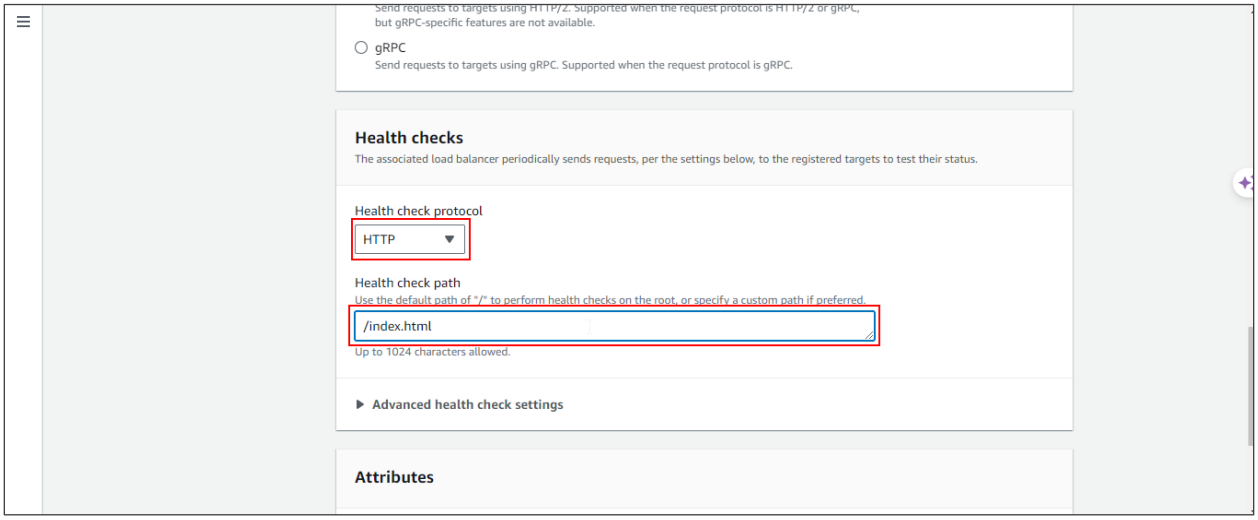


1.4 In the **Basic configuration** section, choose **Instances** as the target type and enter a name for the target group, such as **MyTargetGroup**

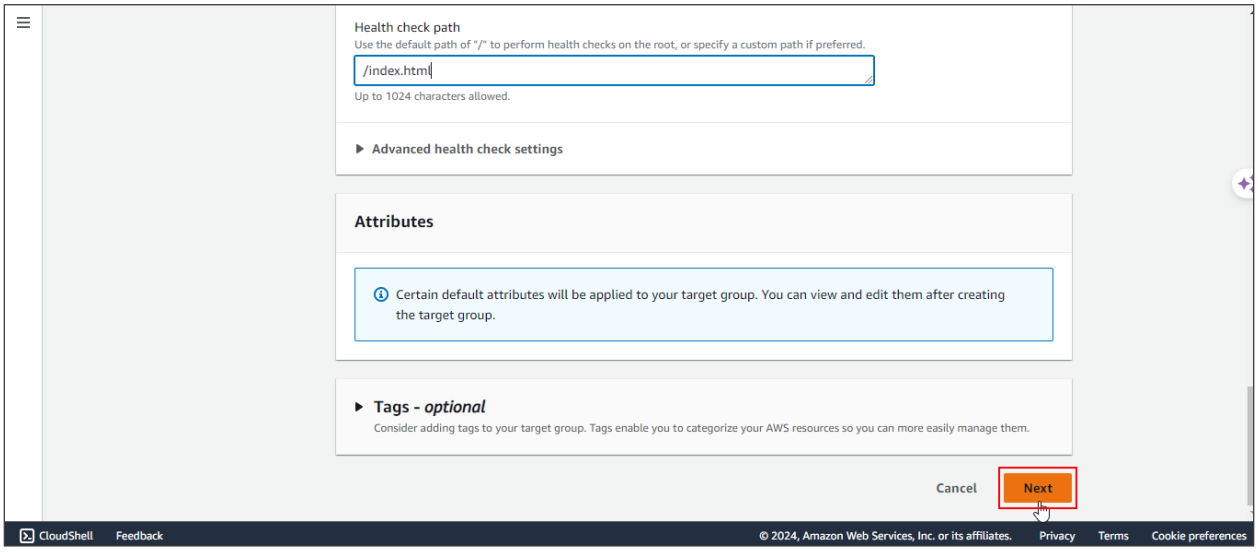




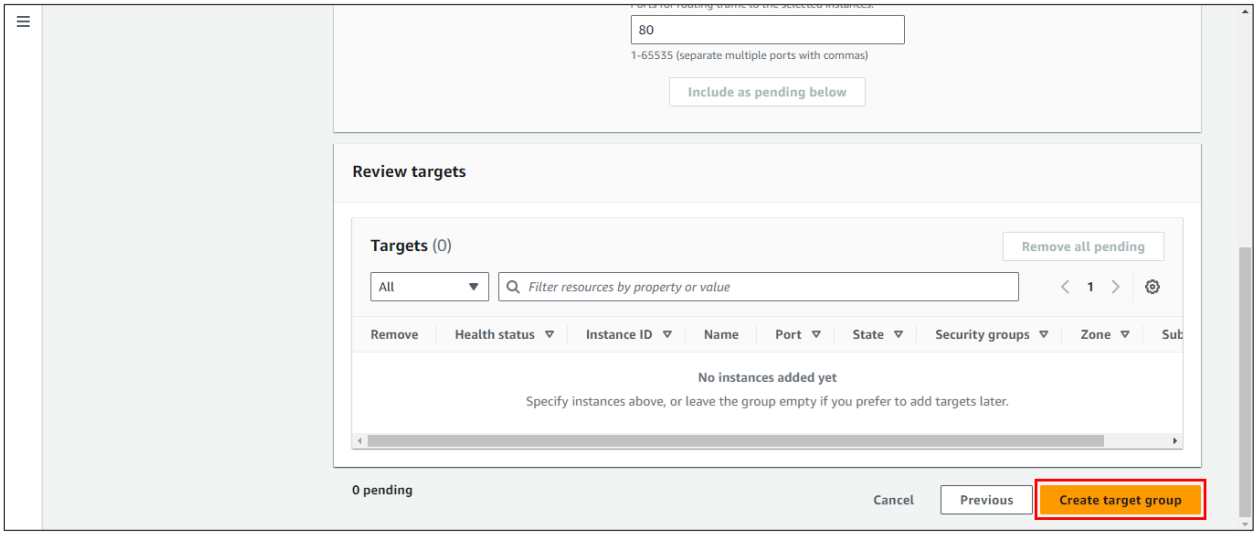
1.5 Set the protocol to **HTTP** and the path to **/index.html** in the **Health checks** section

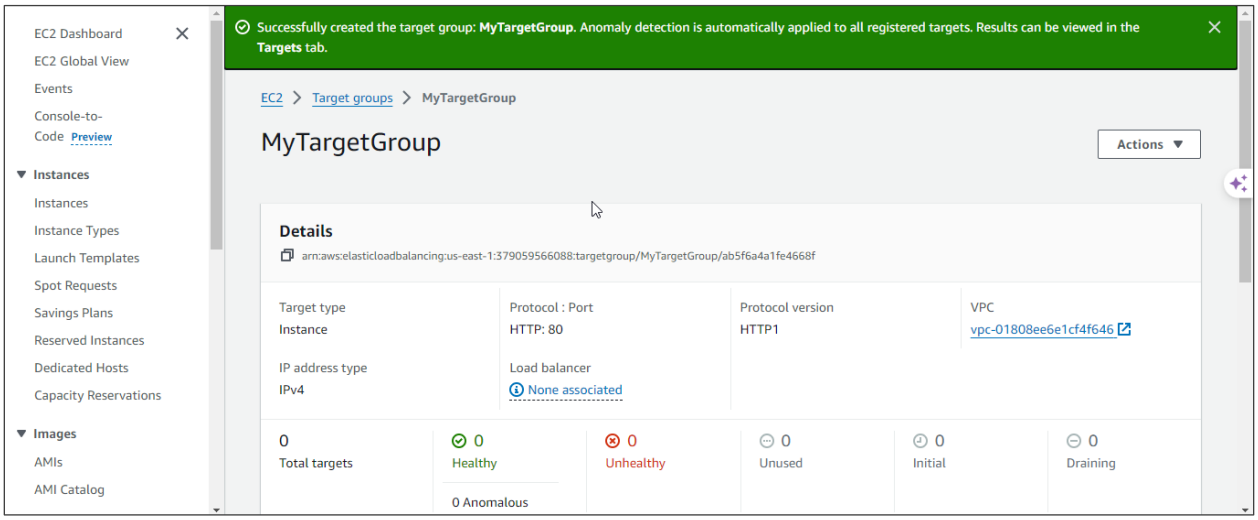


1.6 Click on **Next**



1.7 Review the configurations and click on **Create target group**

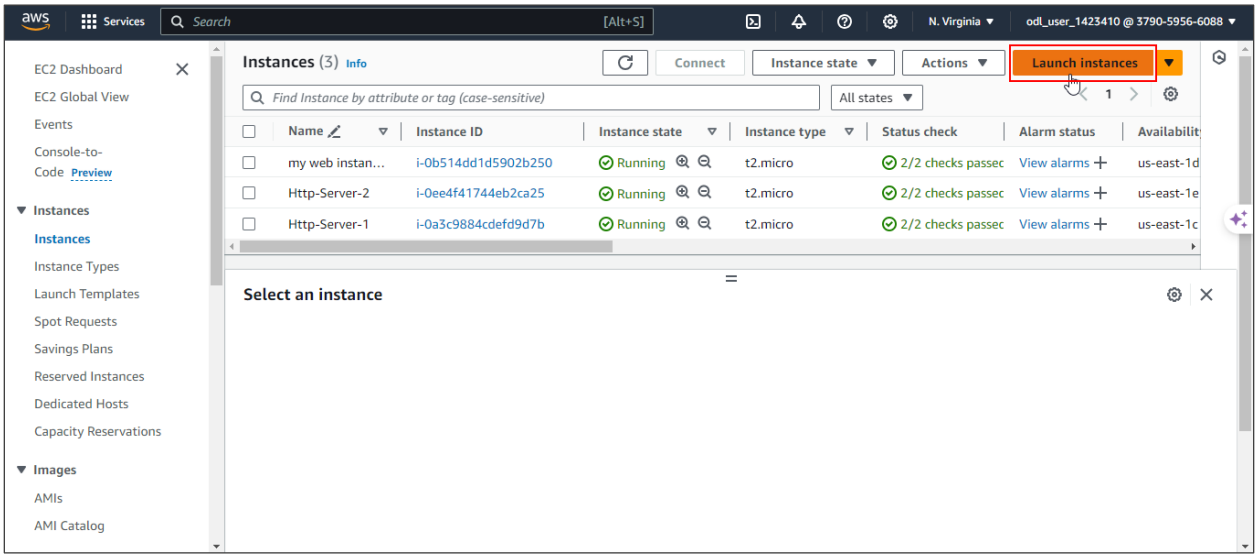




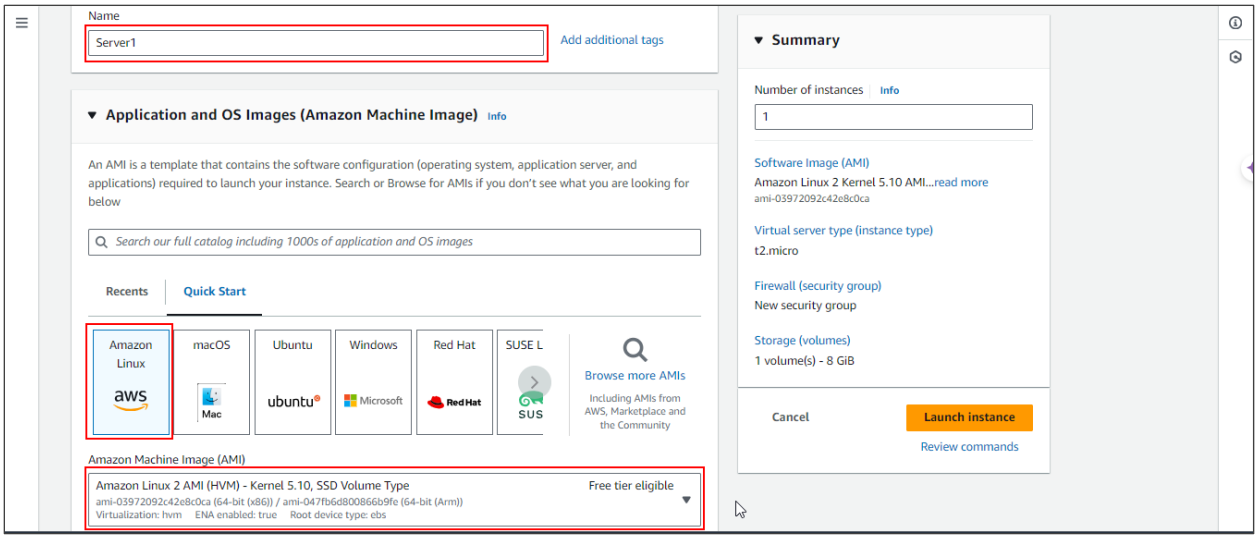
The target group has been successfully created.

Step 2: Launch EC2 instances

2.1 Navigate to the **Instances** section and click on **Launch instances**

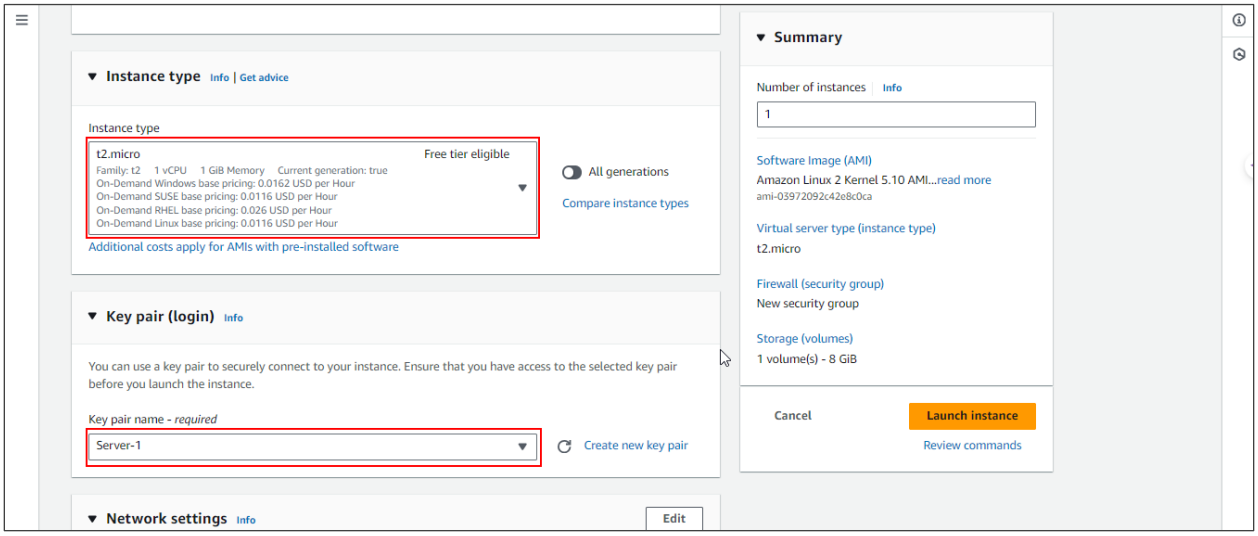


2.2 Provide a name for the instance and choose an appropriate AMI (**Amazon Linux 2**)

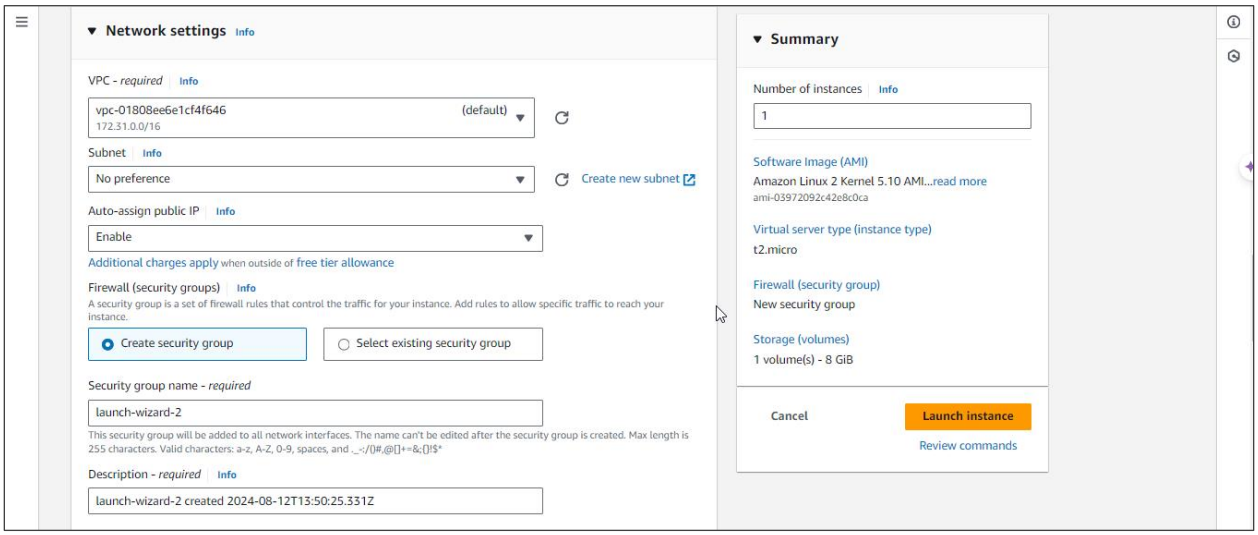


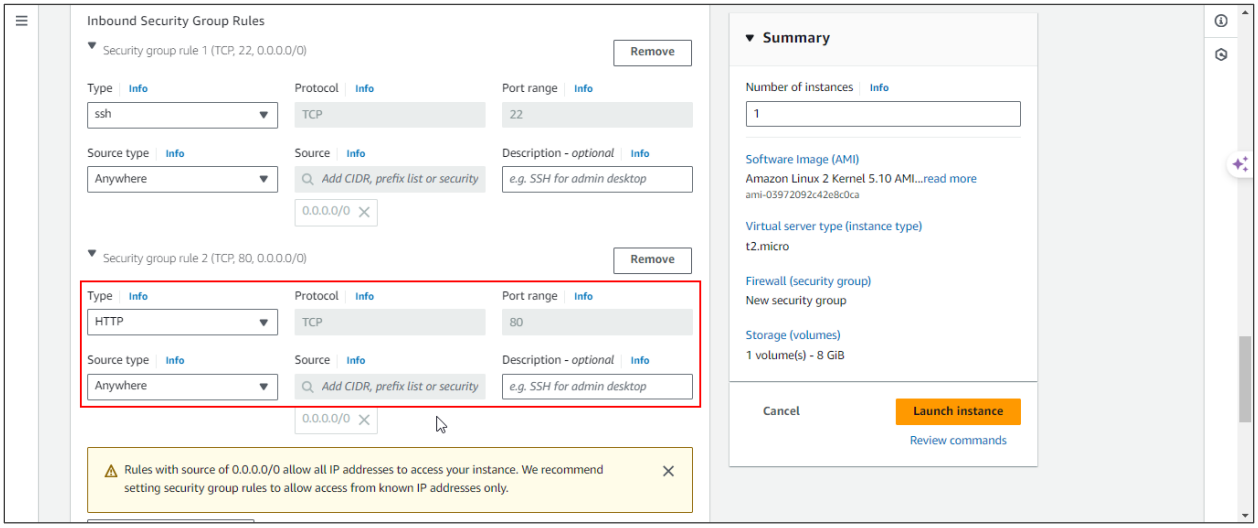
.

2.3 Select the instance type as **t2.micro**, create a key pair, and name it **Server-1**



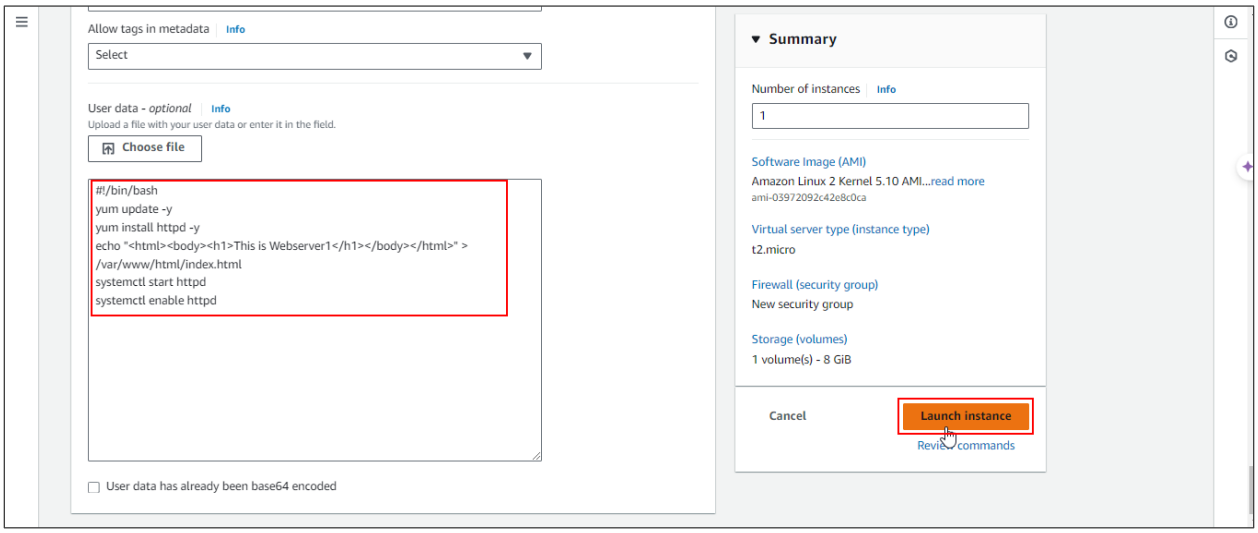
2.4 Configure the network settings as shown:



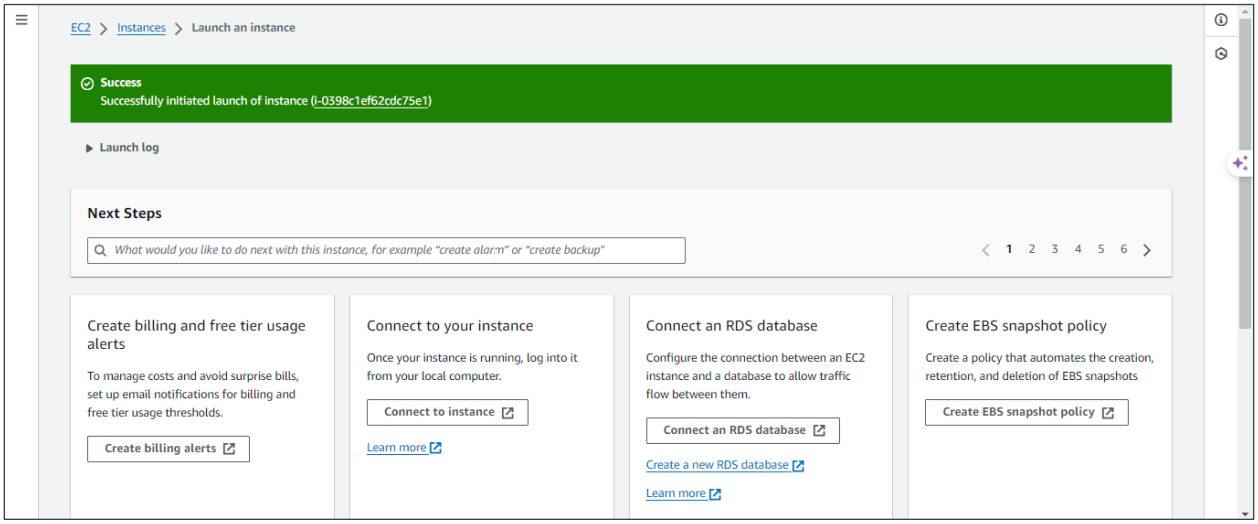


2.5 Add the following user data script in the **Advance details** section, and click on **Launch instance:**

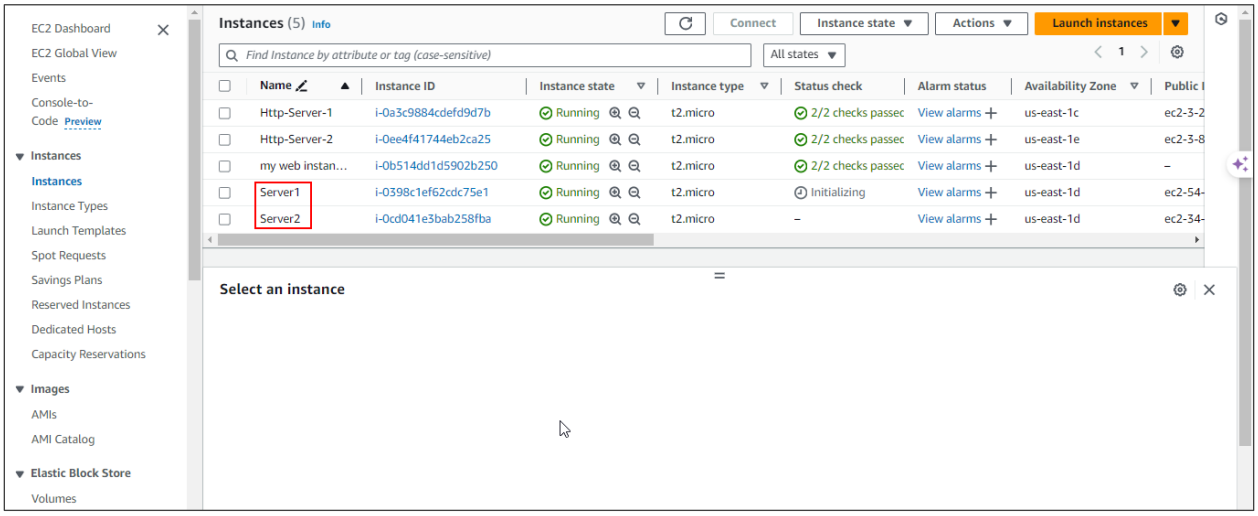
```
#!/bin/bash
yum update -y
yum install httpd -y
echo "<html><body><h1>This is Webserver1</h1></body></html>" >
/var/www/html/index.html
systemctl start httpd
systemctl enable httpd
```



You will see the following interface:



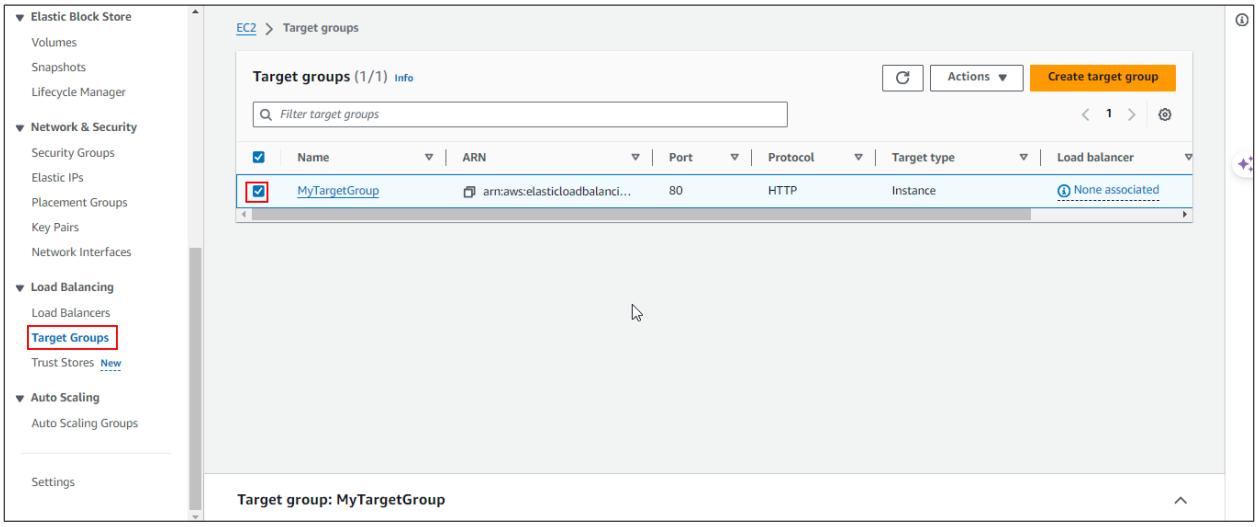
2.6 Launch another EC2 instance using the same steps, but modify the user data script to display the message **This is Webserver2**



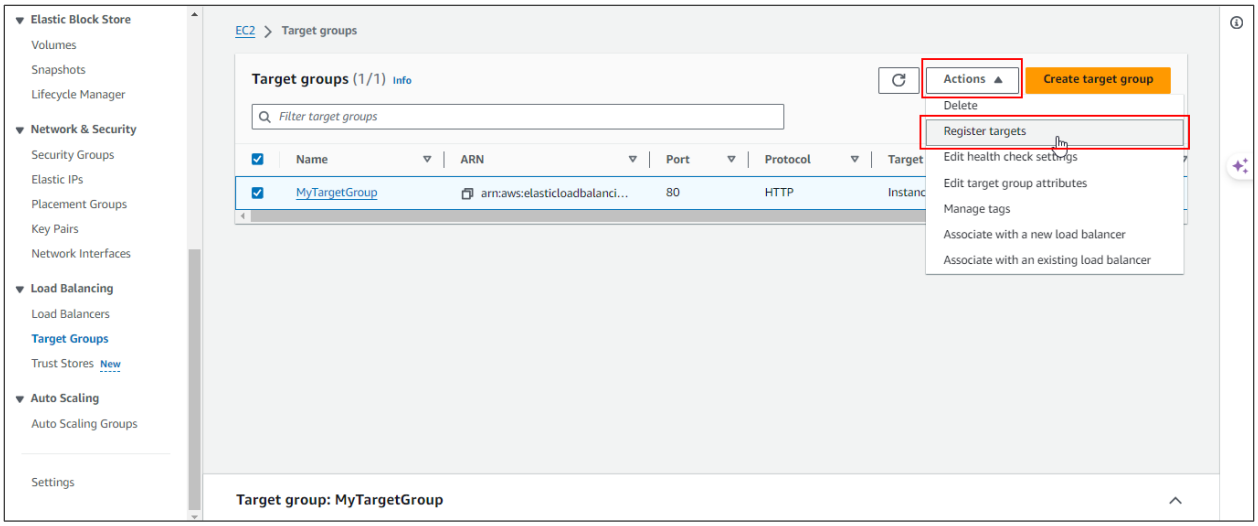
The EC2 instances have been successfully launched.

Step 3: Configure the target group

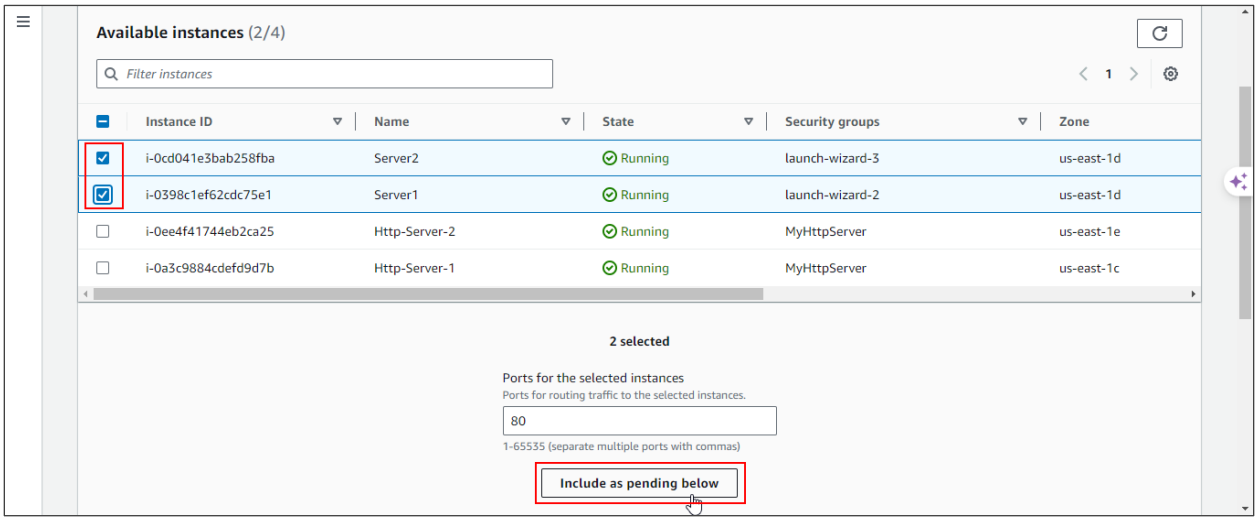
3.1 Navigate to the **Target Groups** section and select the target group created in Step 1



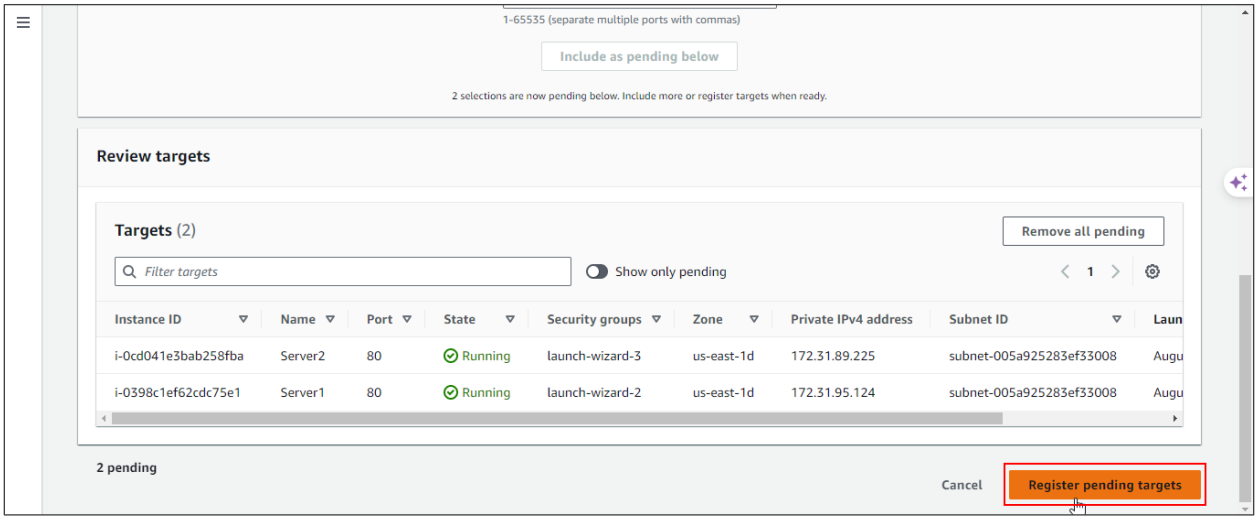
3.2 Click on **Register targets** from the **Actions** menu

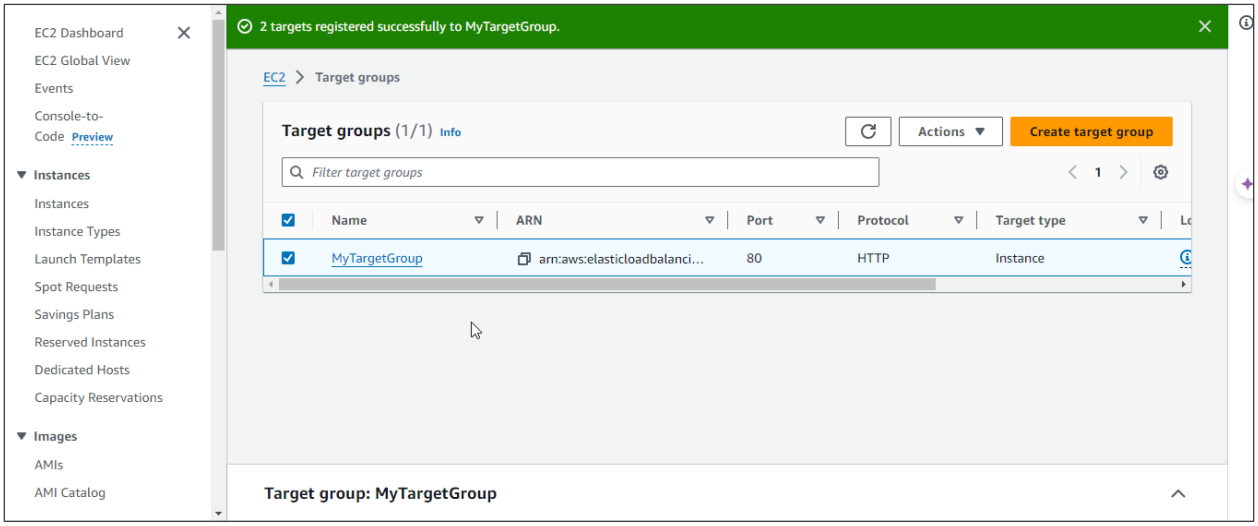


3.3 Select the instances (**Server1** and **Server2**) that were launched in Step 2 and click on **Include as pending below**



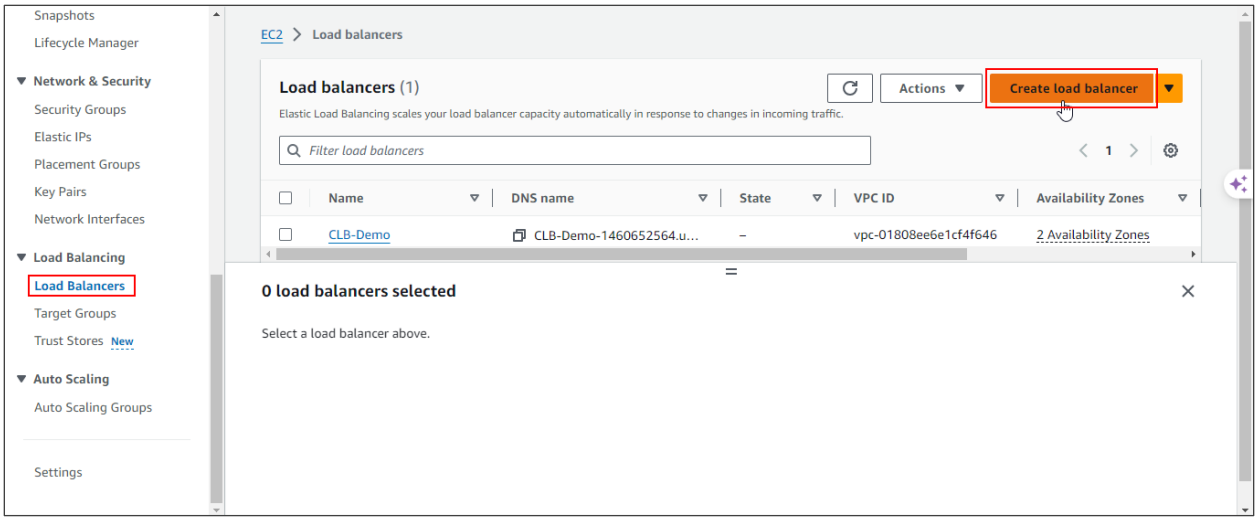
3.4 Click on **Register pending targets** to register the instances with the target group



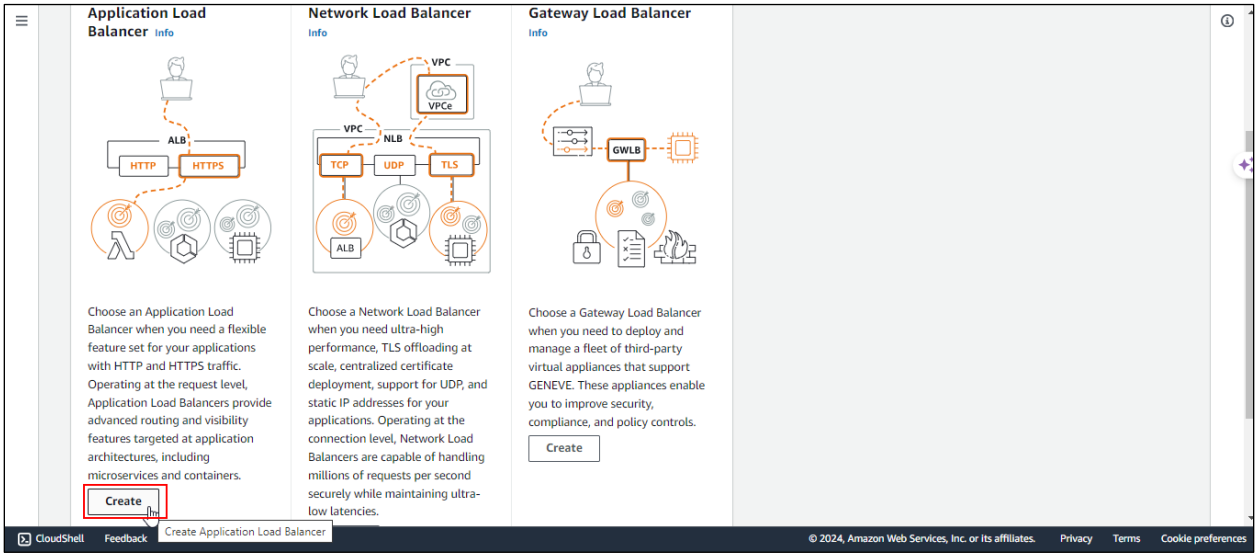


Step 4: Create a Load Balancer

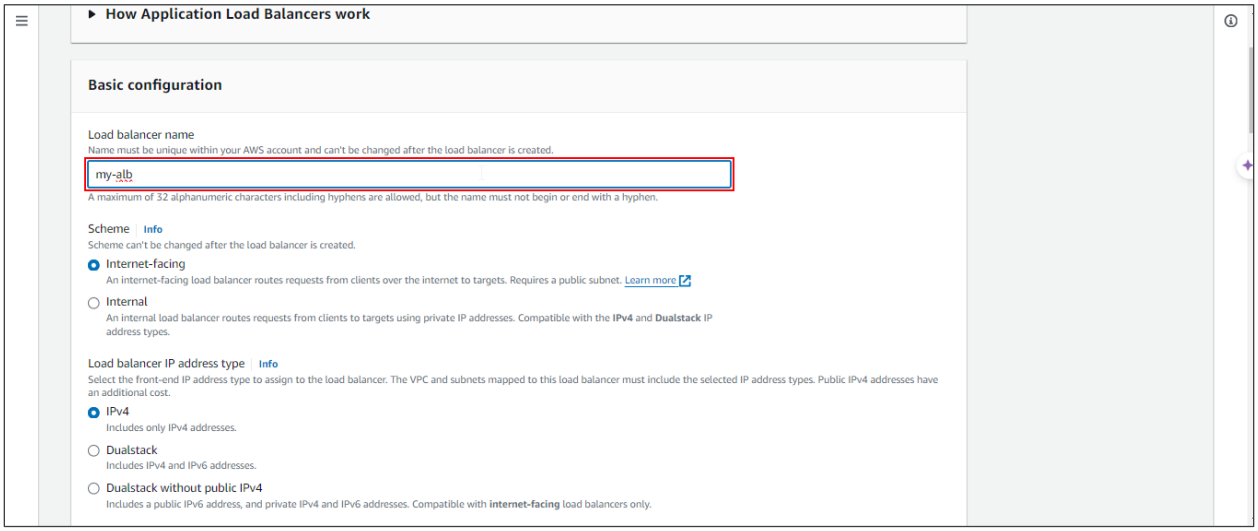
4.1 Navigate to the **Load Balancers** section under **Load Balancing** and click **Create load balancer**

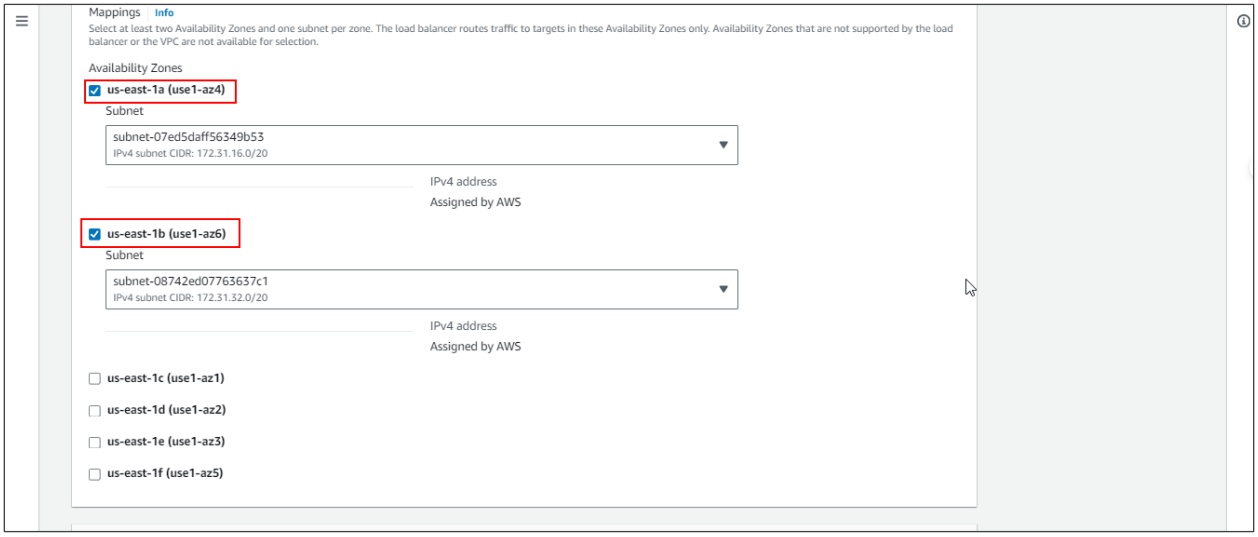


4.2 Choose **Application Load Balancer** and click **Create**

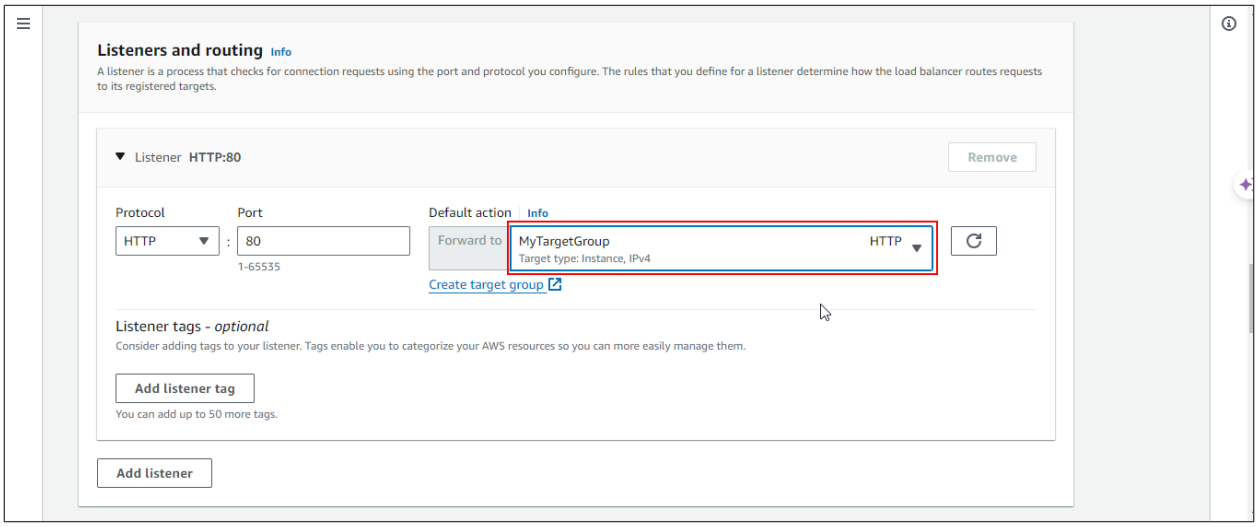


4.3 Configure the load balancer settings, enter a name for the load balancer, such as **my-alb**, and select availability zones, such as **us-east-1a** and **us-east-1b**

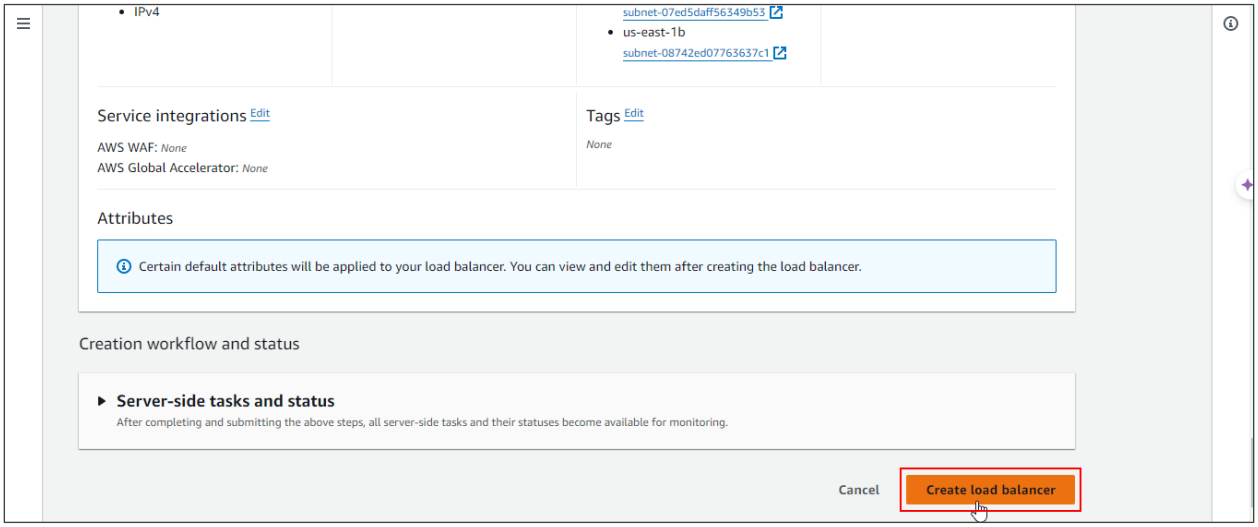




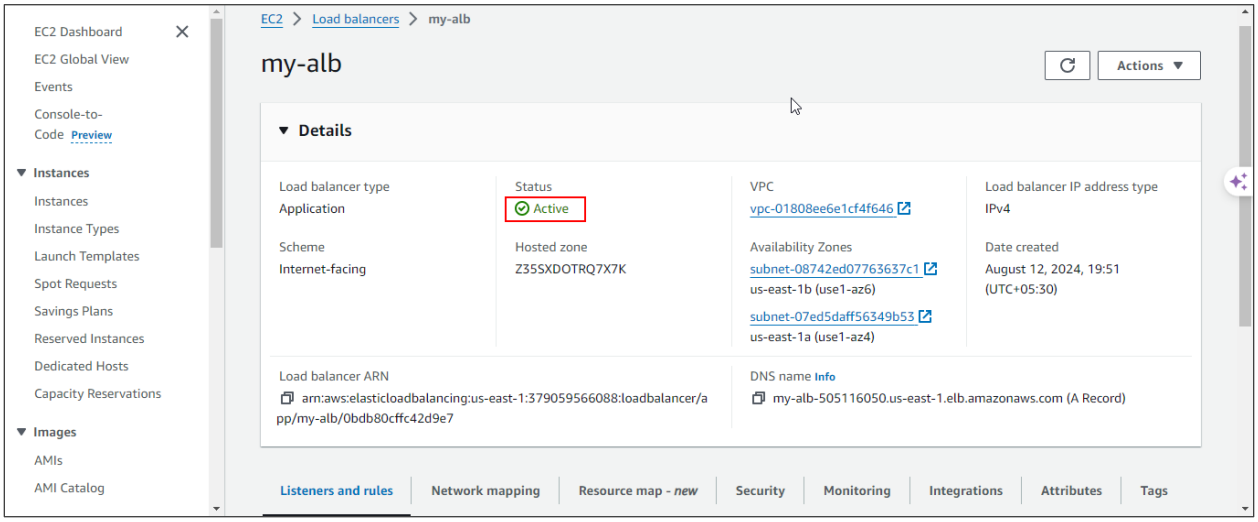
4.4 Choose the default action for the listener configuration to accept HTTP traffic on port **80**, and select the target group created in Step 1



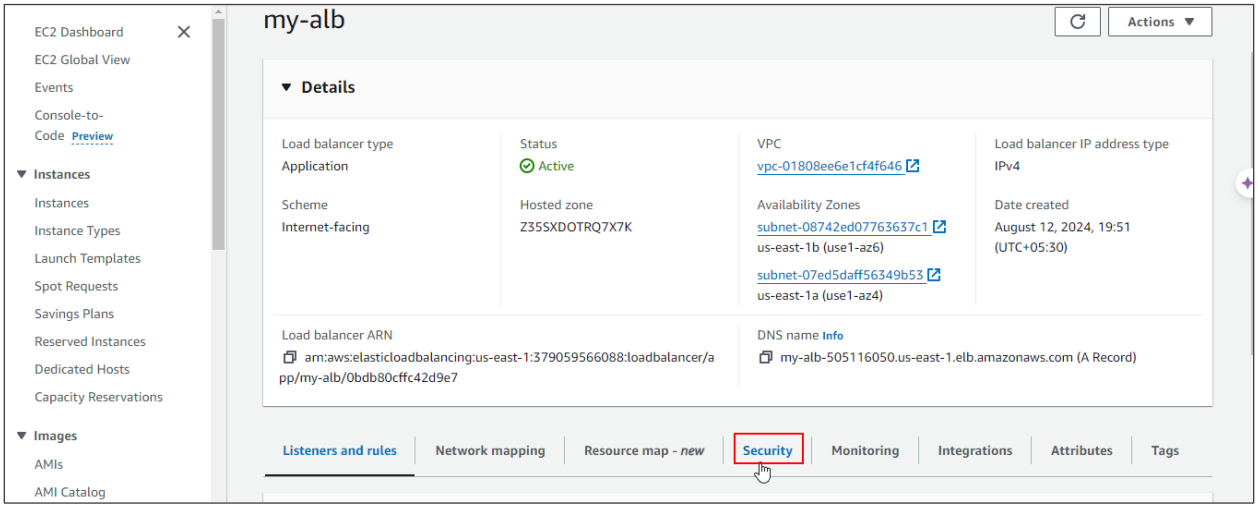
4.5 Review the configuration and click **Create load balancer**



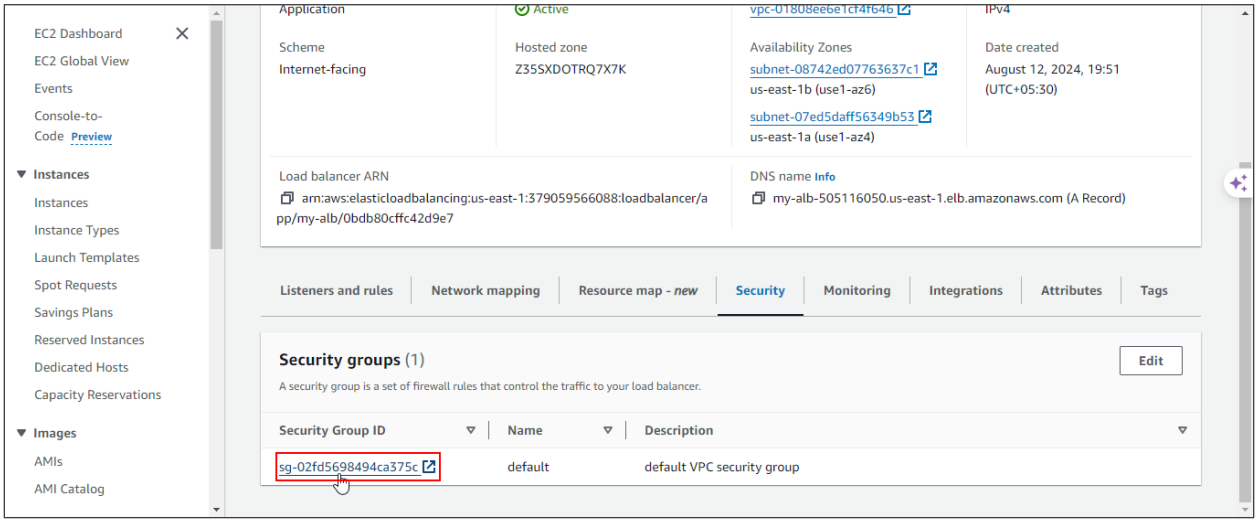
Wait until the **Status** changes from **Provisioning** to **Active**



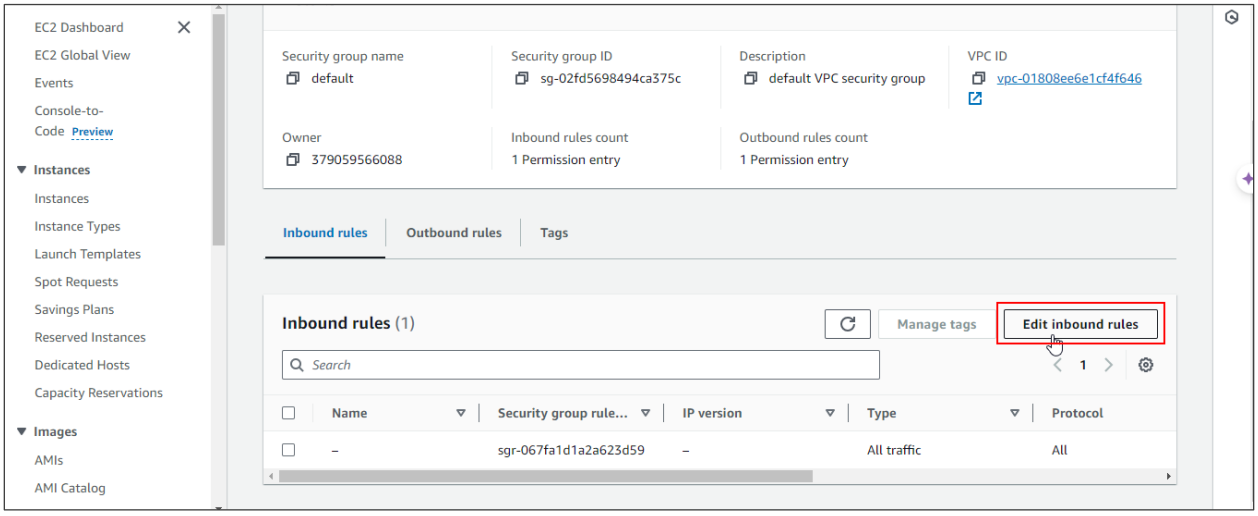
4.6 Click on the **Security** tab



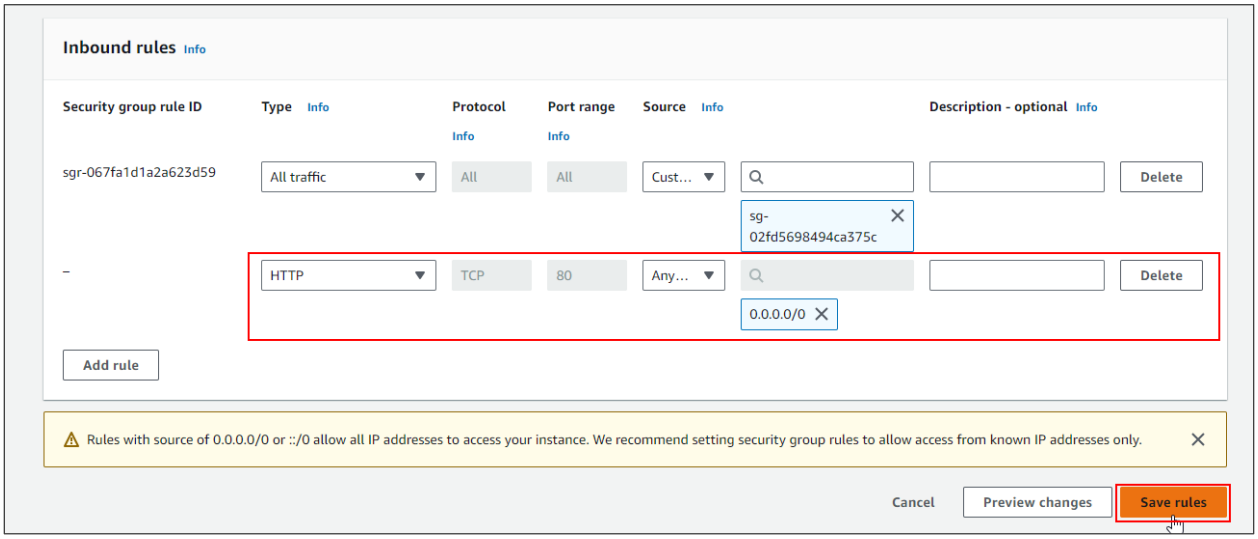
4.7 Click on the **Security Group ID** name



4.8 Click on **Edit inbound rules**

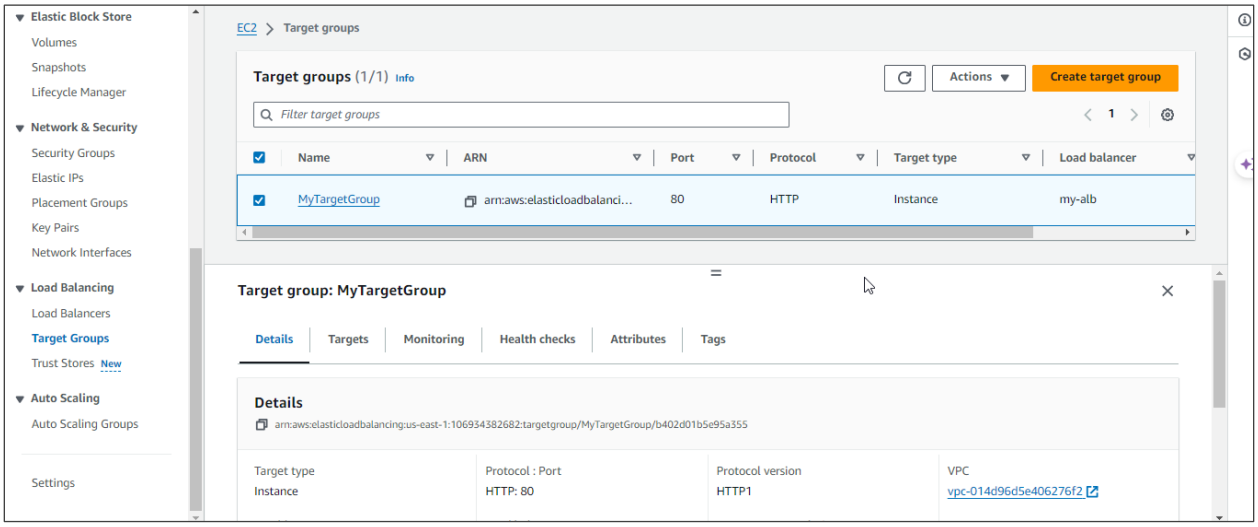


4.9 Create an inbound rule to permit port **80** access for all, and click on **Save rules** as shown:

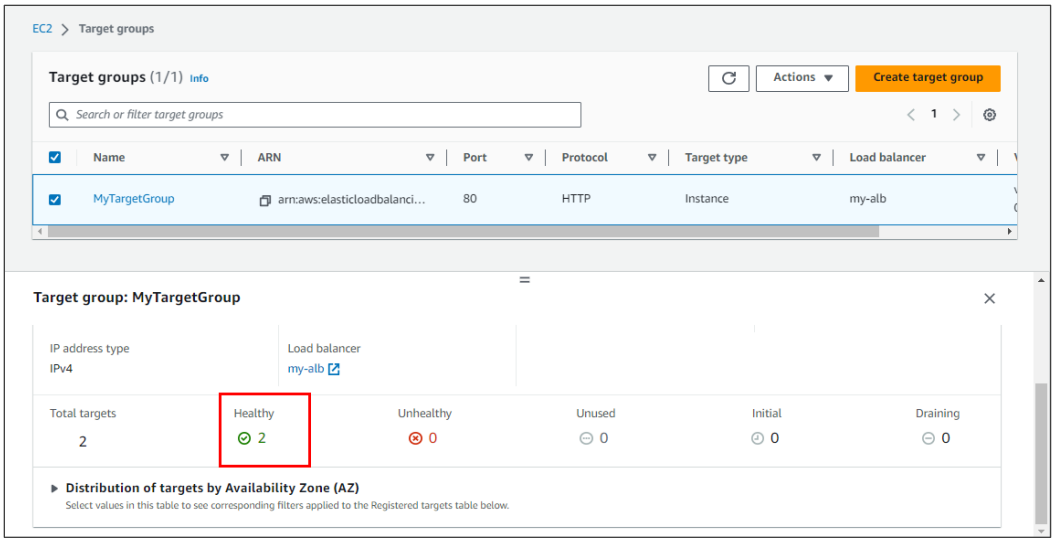


Step 5: Test the Load Balancer

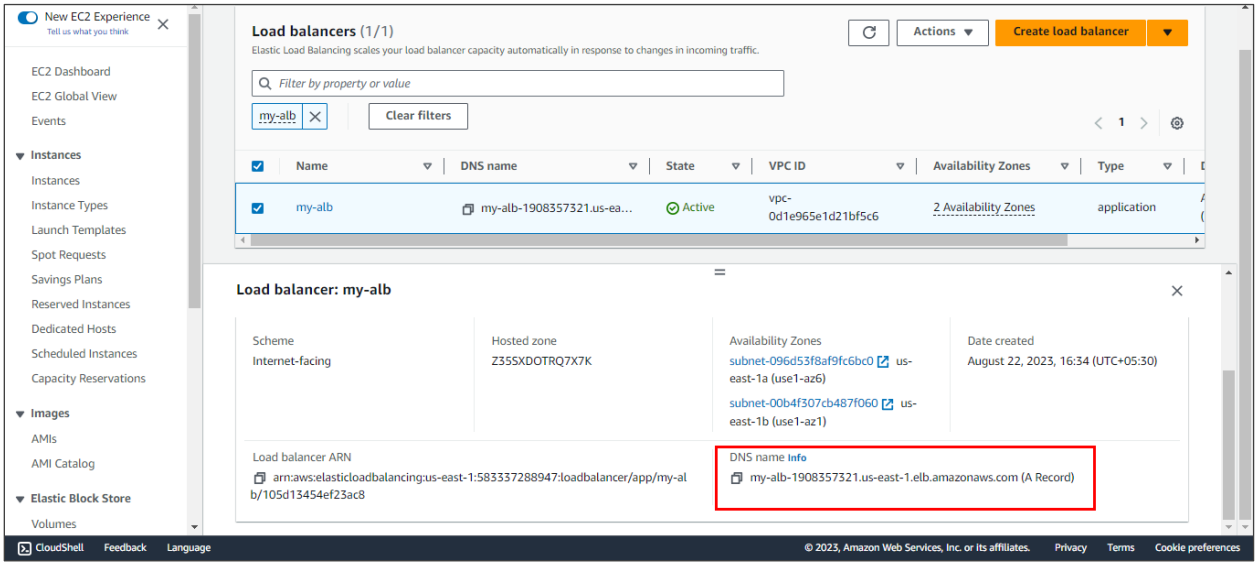
5.1 Navigate to the **Target Groups** section and select the target group you created



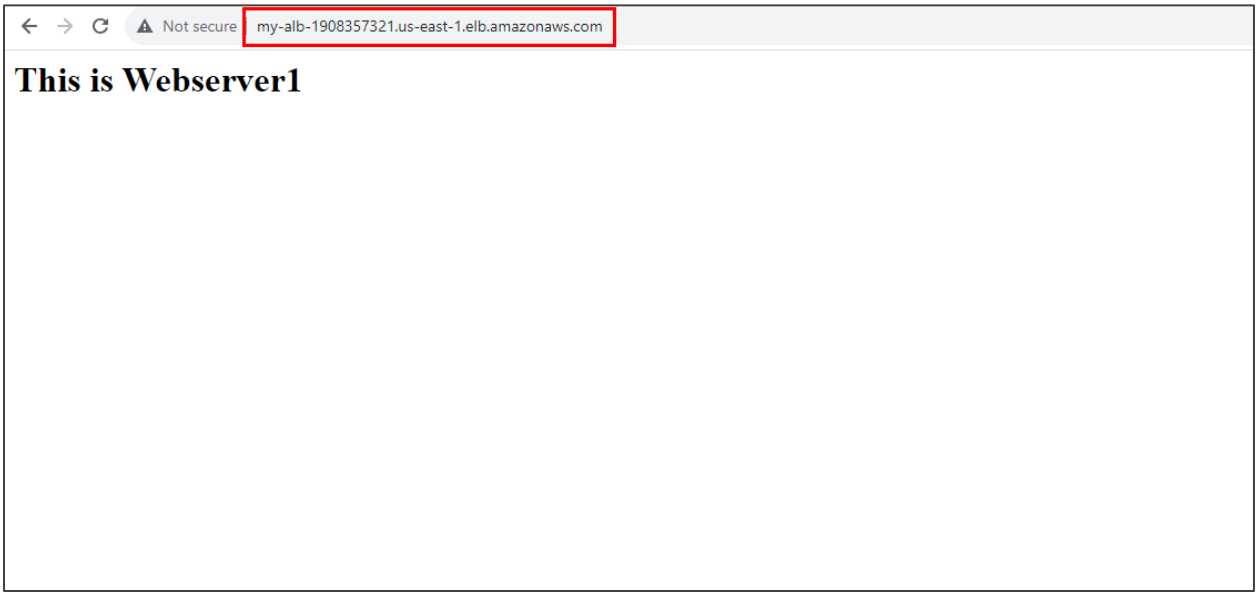
5.2 Click on **Details** to verify that your instances are registered and healthy



5.3 Navigate to the **Load Balancers** section and copy the DNS name of the Load Balancer

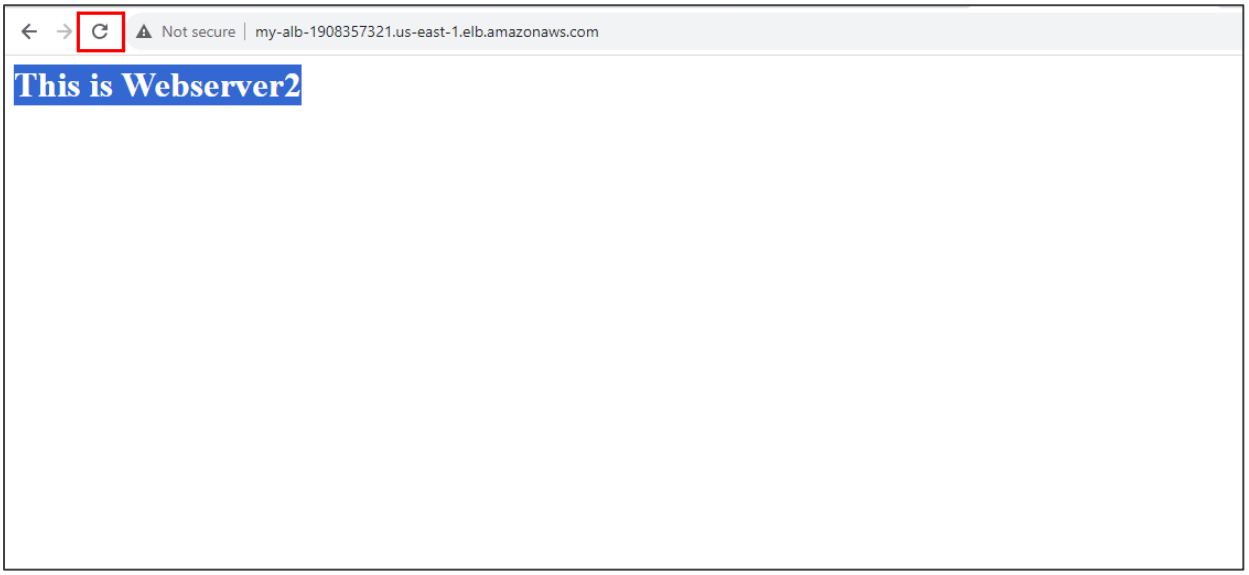


5.4 Open a browser window and paste the DNS URL into the address bar



You will observe the header message originating from the **Server1** instance.

5.5 Refresh the web page multiple times to see the header message originating from the **Server2** instance



By following these steps, you have successfully configured an Application Load Balancer in AWS to distribute traffic across multiple EC2 instances, ensuring load balancing and redundancy.