**Little Bit Advance Labs**

**Part 1: EC2 with ELB and ASG**

**Objective:** Learn how to create a scalable and highly available web application environment using Amazon EC2 instances, ELB, and ASG.

**Approach:**

**1. Launch EC2 Instances**: Start by launching two or more EC2 instances. These instances will run a simple web application (e.g., a "Hello World" page or any basic web service).

**2. Configure Load Balancer:** Set up an Elastic Load Balancer (ELB) to distribute incoming web traffic across your EC2 instances. This step ensures high availability and fault tolerance.

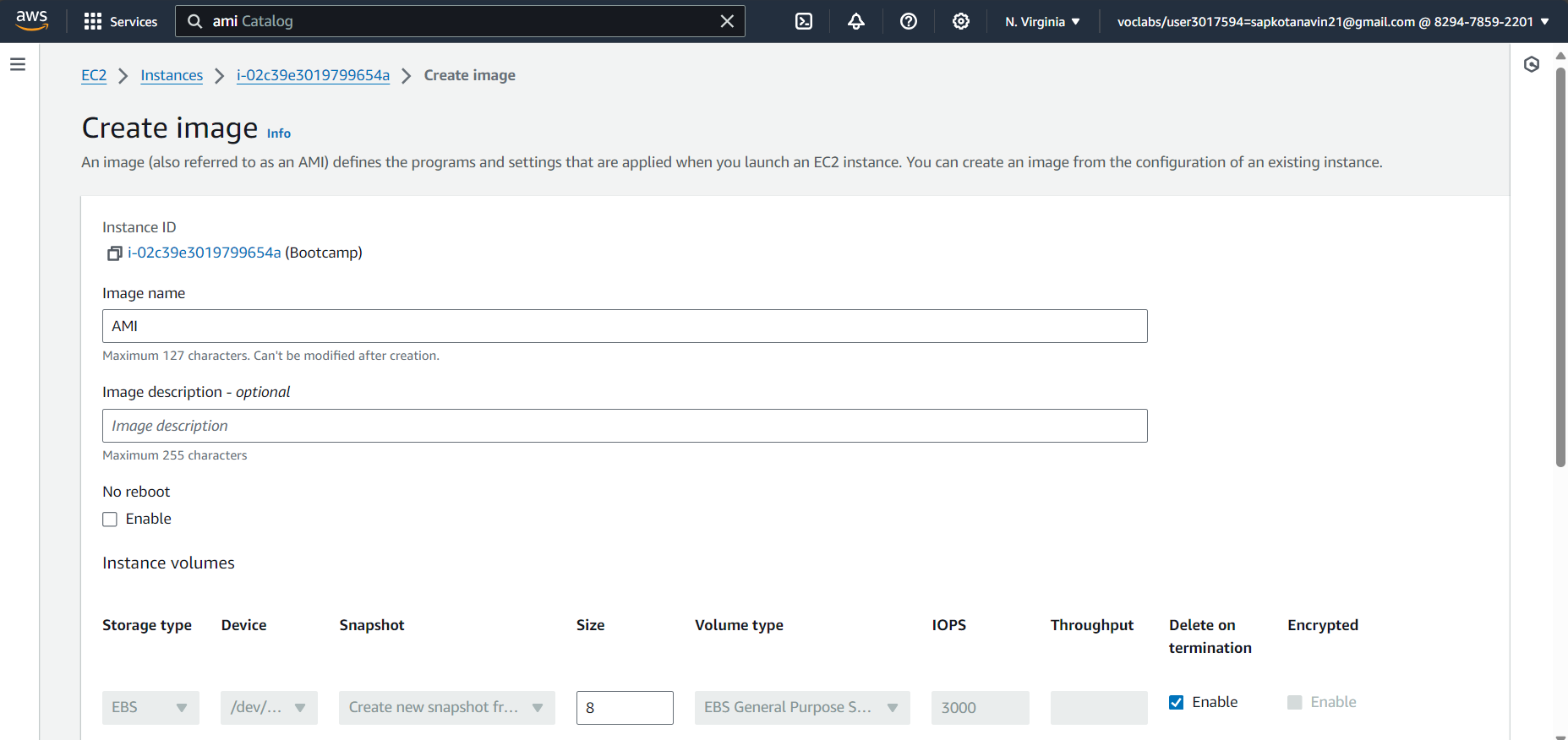
**3. Set Up Auto Scaling Group (ASG):** Create an ASG that uses the launched EC2 instances. Configure ASG policies to automatically scale the number of instances up or down based on criteria like CPU usage or network traffic.

**4. Test Your Setup:** Simulate traffic to test the scaling policies and the load balancer. Observe how ASG adds or removes instances and how ELB distributes traffic.

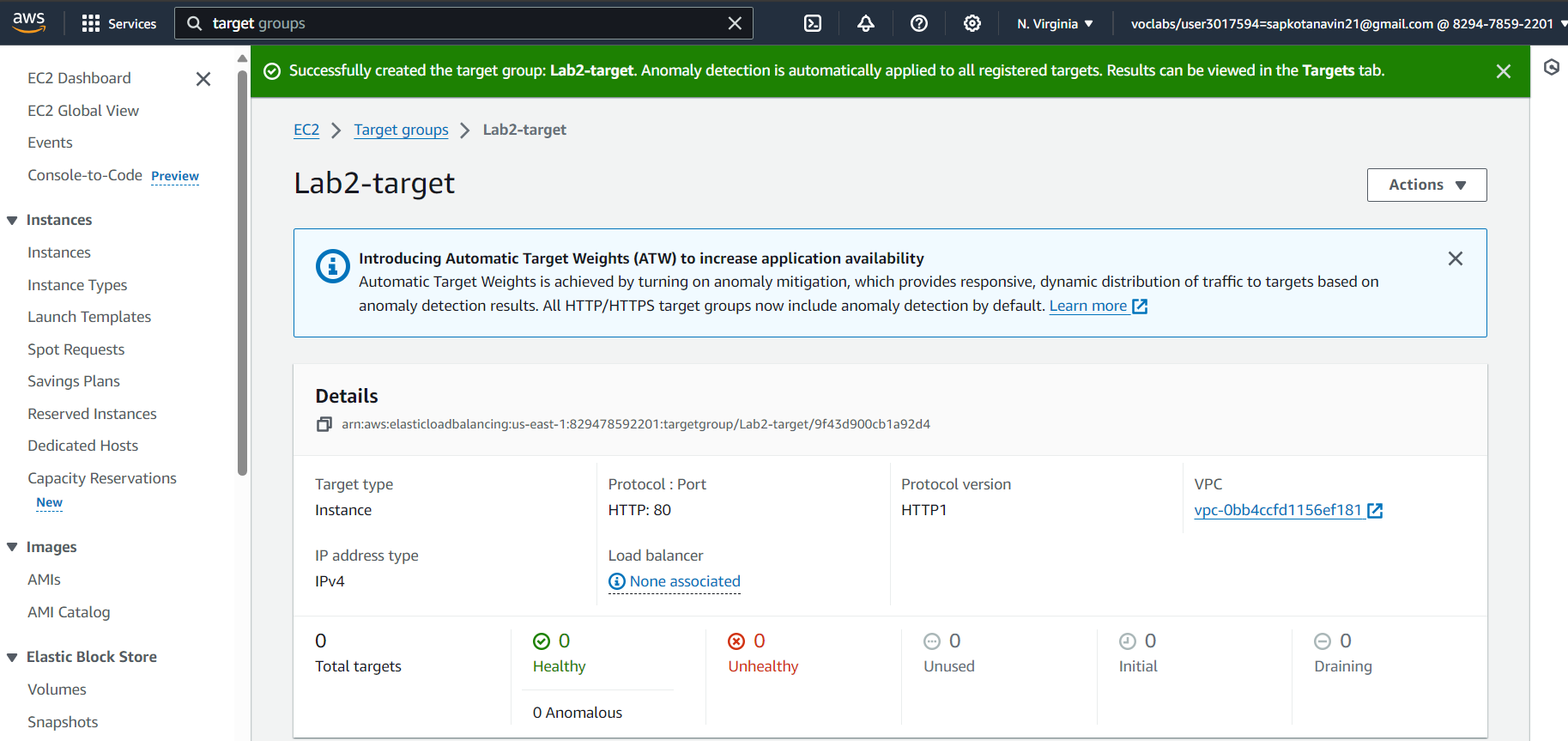
**5. Verify Website Functionality:** Ensure that the website hosted on EC2 instances remains accessible and functional during scaling operations. Goal: By the end of this lab, students will have a hands-on understanding of setting up a load-balanced and auto-scaled web application using AWS services.

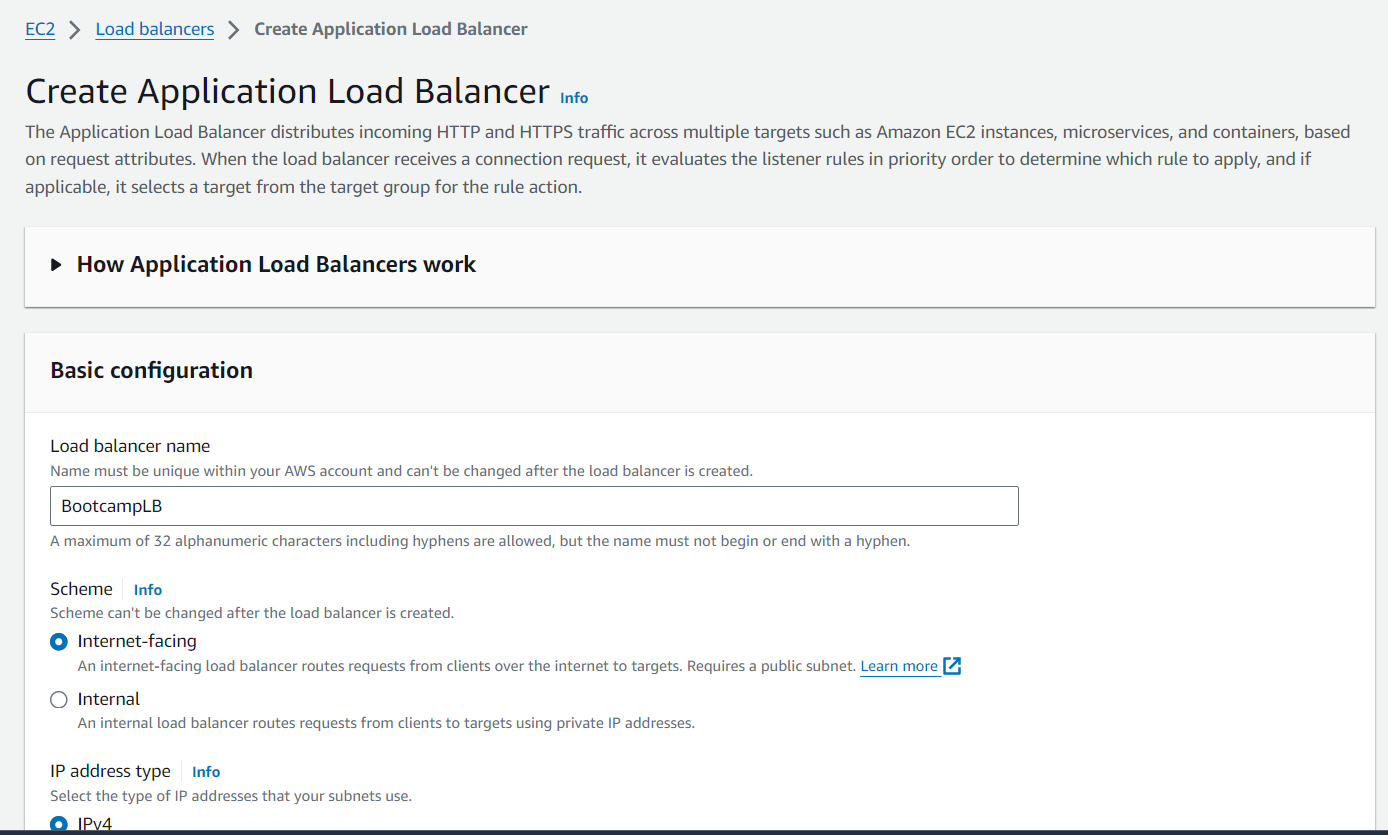
**Steps:**

1. Open AWS Management Console and create an Auto Scaling Image as

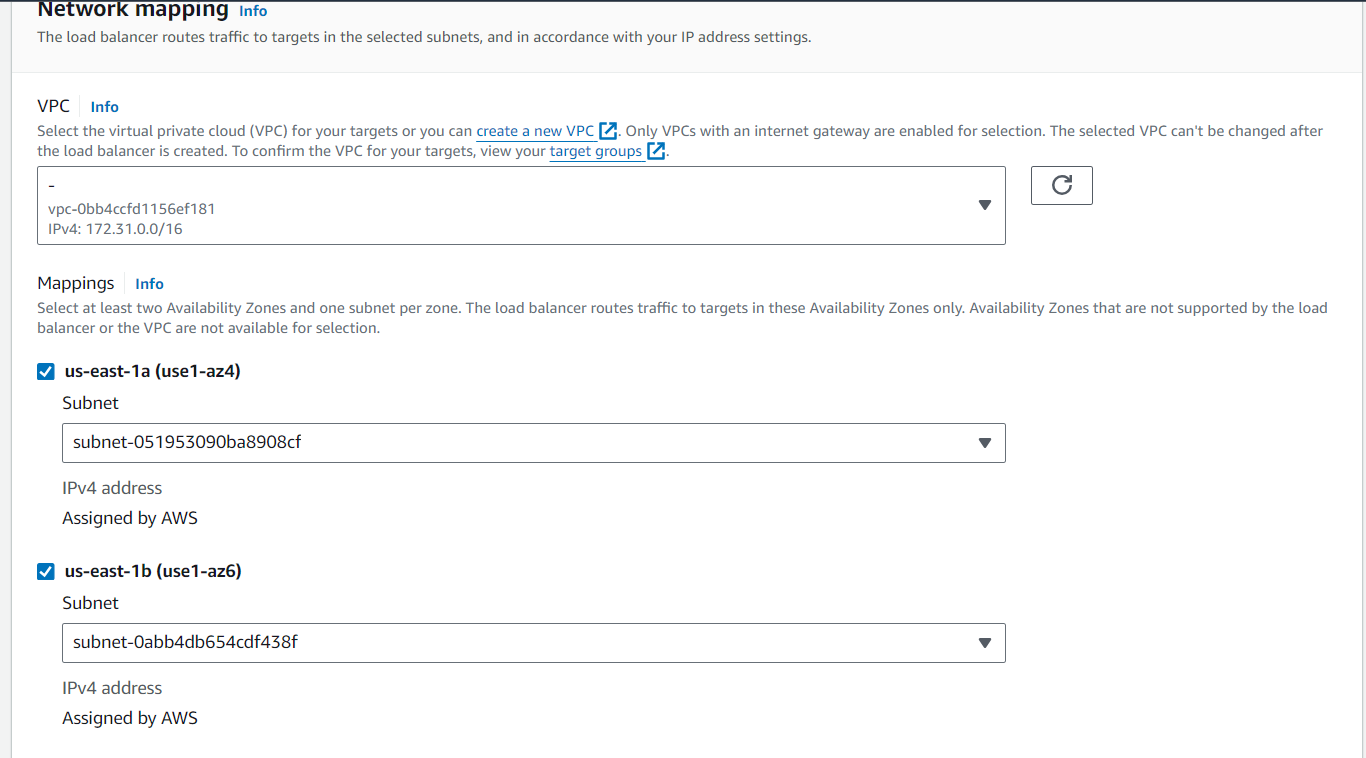
****

1. Create a target group and attached with load balancer as

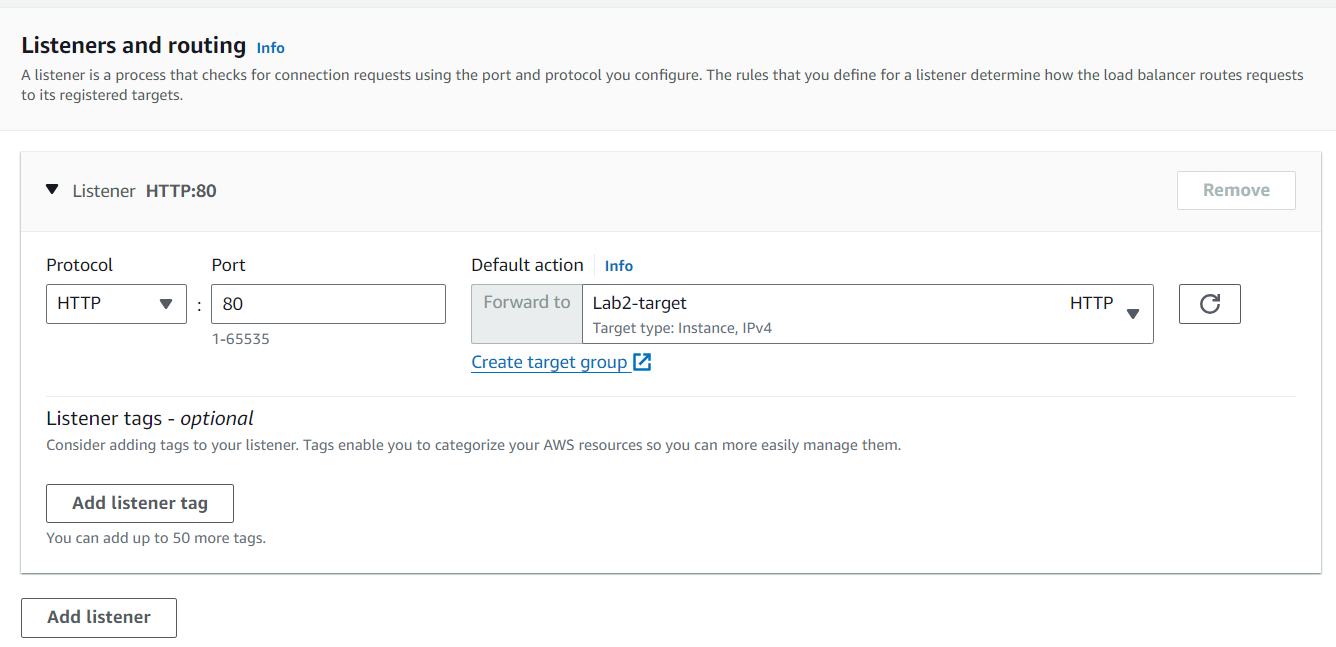
****

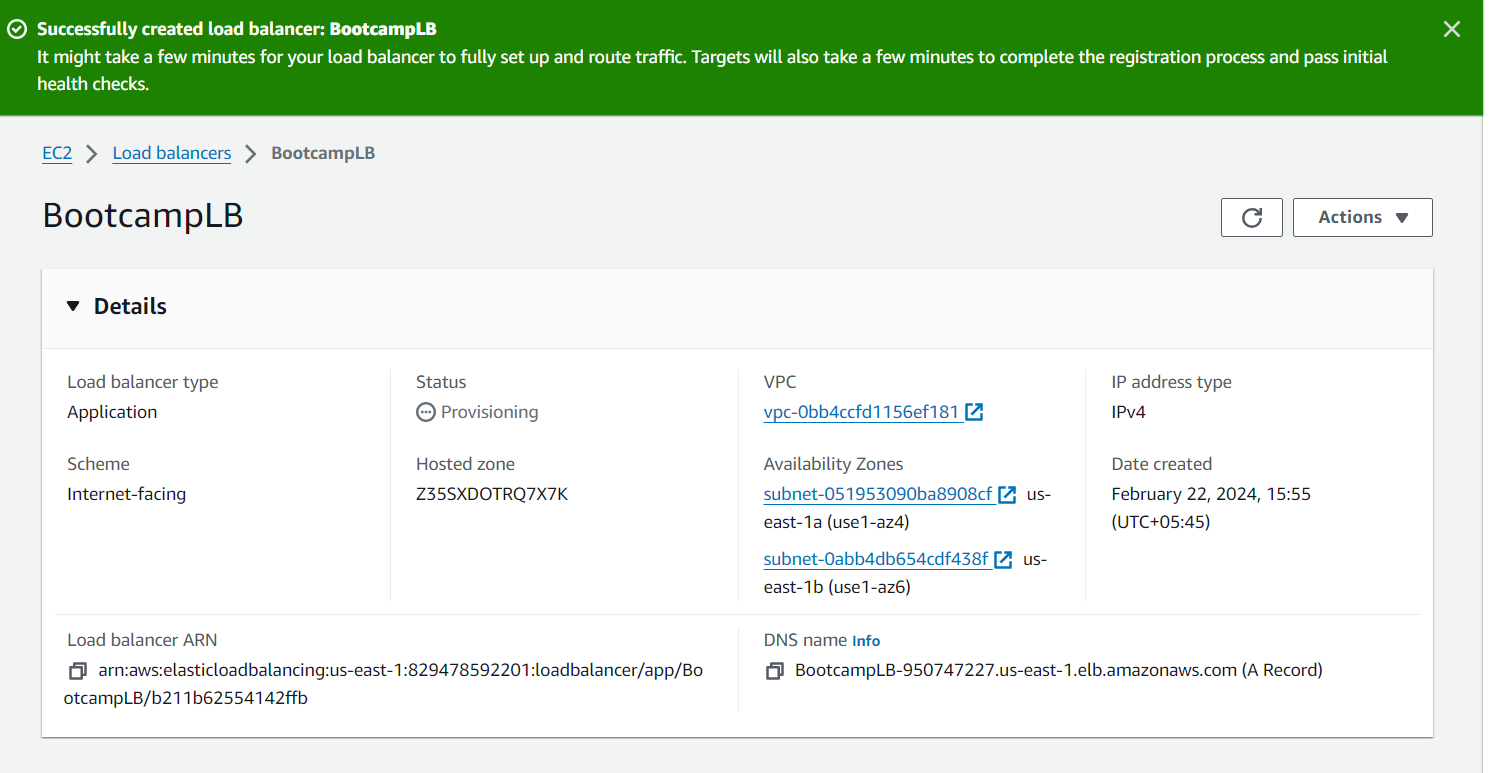
****

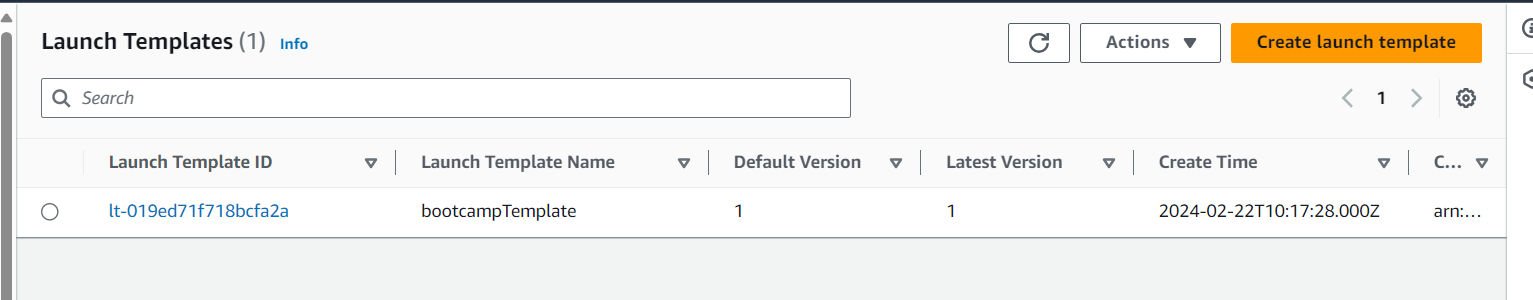
**Attaching a VPC**

****

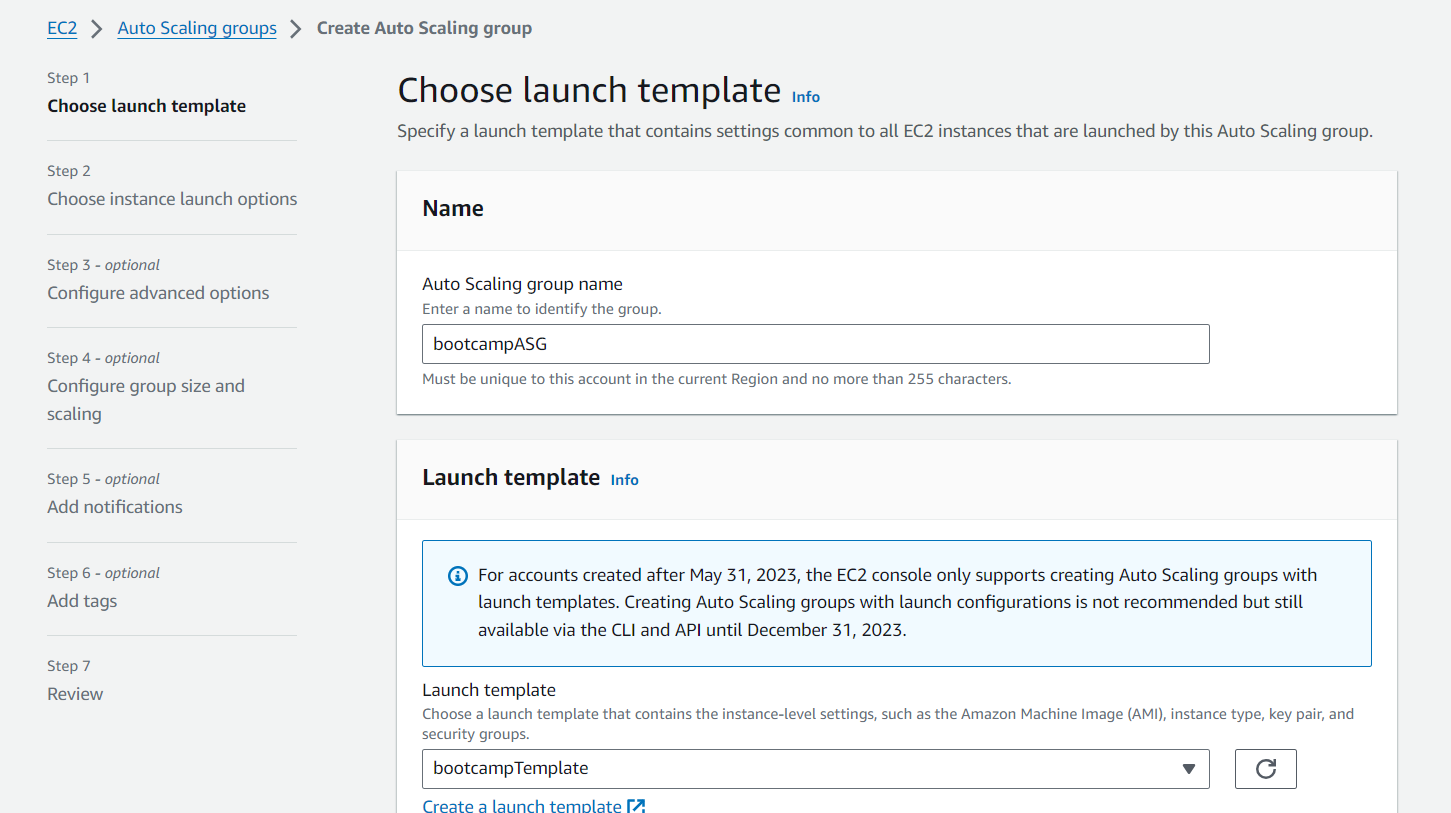
**Selecting listener and routing**

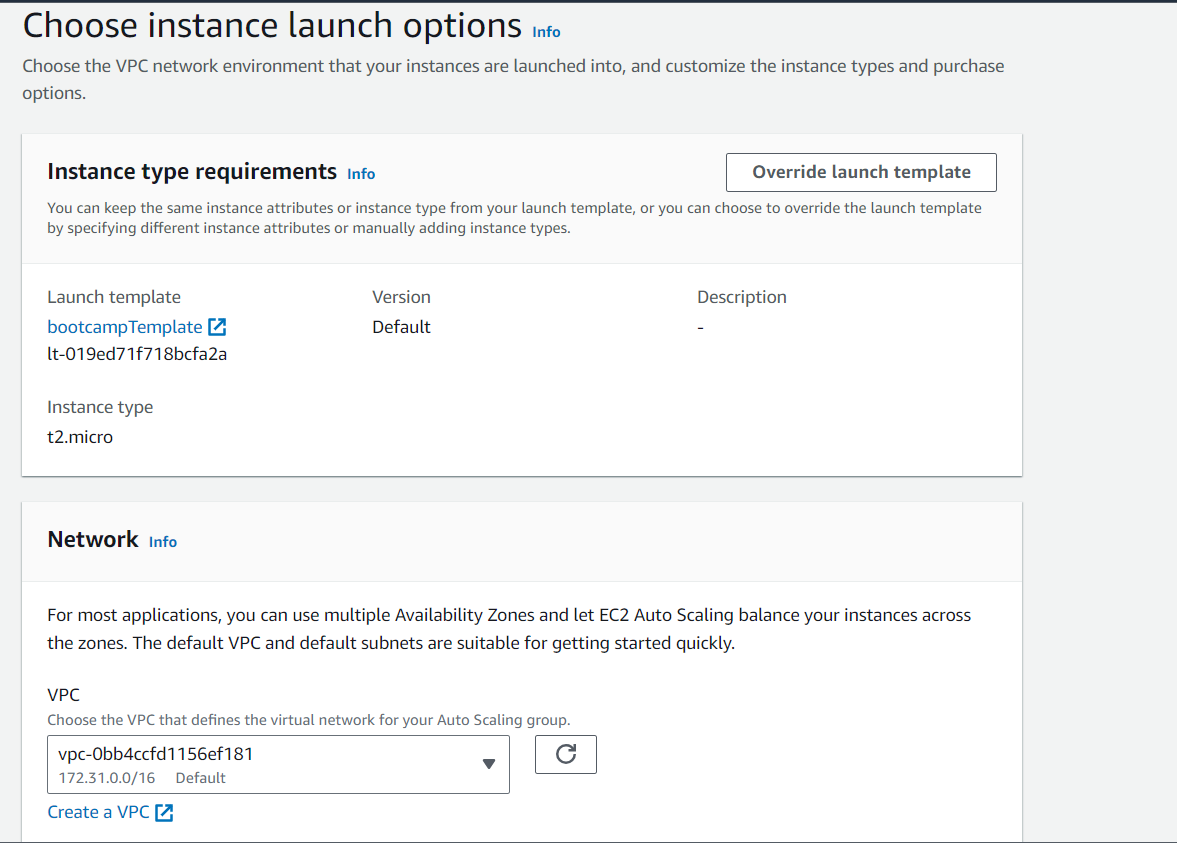
****

****

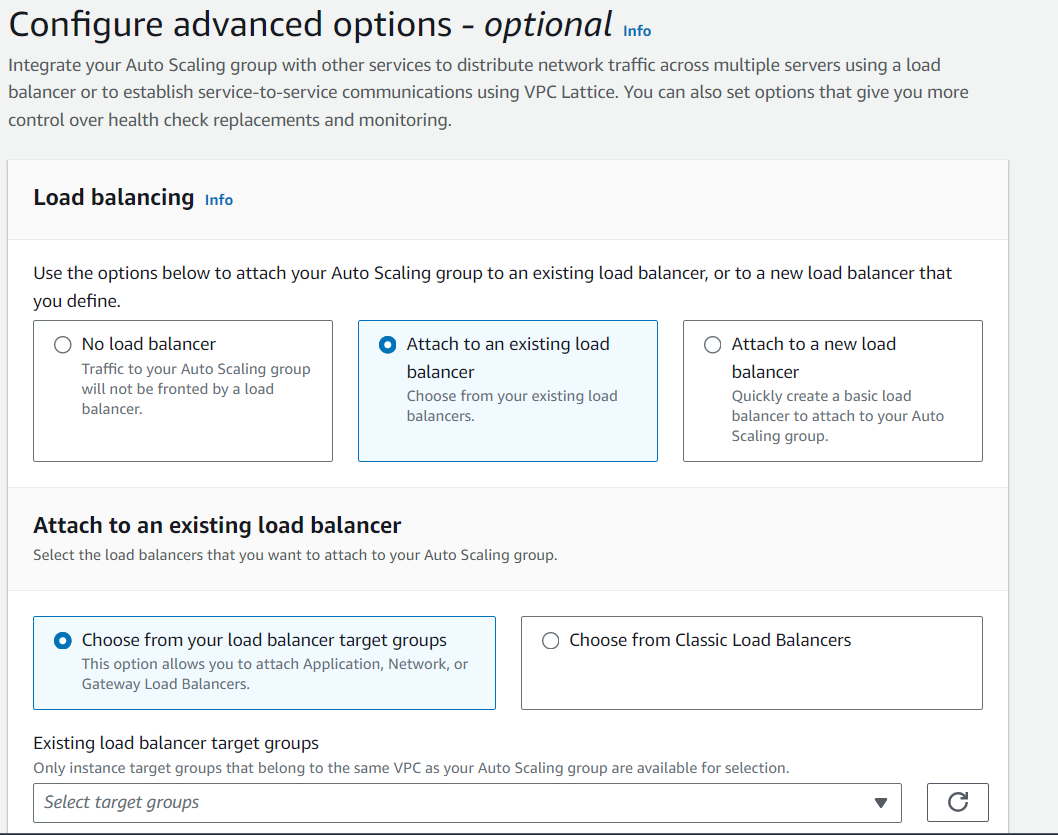
****

1. Create an auto scaling group. Before it, create a launch template

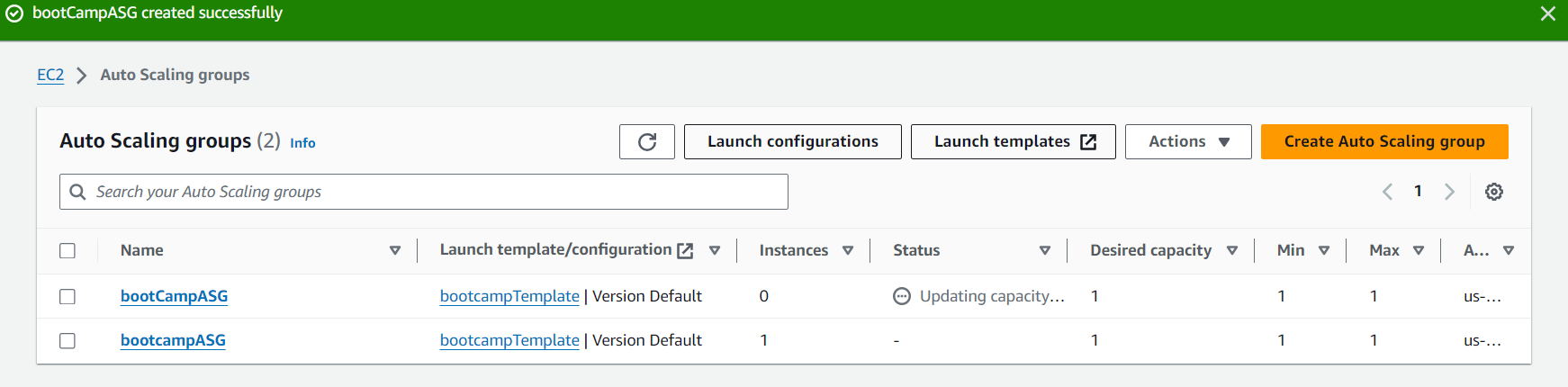
****

****

**Attaching a load balancer**

****

**Auto Scaling Group is created as**

****