COS20019

Cloud Computing Architechture

Week 7 – ACF Lab 6:
Scaling and Load Balance you Architecture

Truong Ngoc Gia Hieu
105565520

Lab 6: Scale and Load Balance Your Architecture

A. Lab Overview and objectives

This lab walks you through using the Elastic Load Balancing (ELB) and Auto Scaling services to load balance and automatically scale your infrastructure.

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve fault tolerance in your applications by seamlessly providing the required amount of load balancing capacity needed to route application traffic.

Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity out or in automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances. Auto Scaling can also automatically increase the number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs. Auto Scaling is well suited to applications that have stable demand patterns or that experience hourly, daily, or weekly variability in usage.

By the end of this lab, you will be able to:

- Create an Amazon Machine Image (AMI) from a running instance.
- Create a load balancer.
- Create a launch template and an Auto Scaling group.
- Automatically scale new instances
- Create Amazon CloudWatch alarms and monitor performance of your infrastructure.

Duration

This lab takes approximately **30 minutes**.

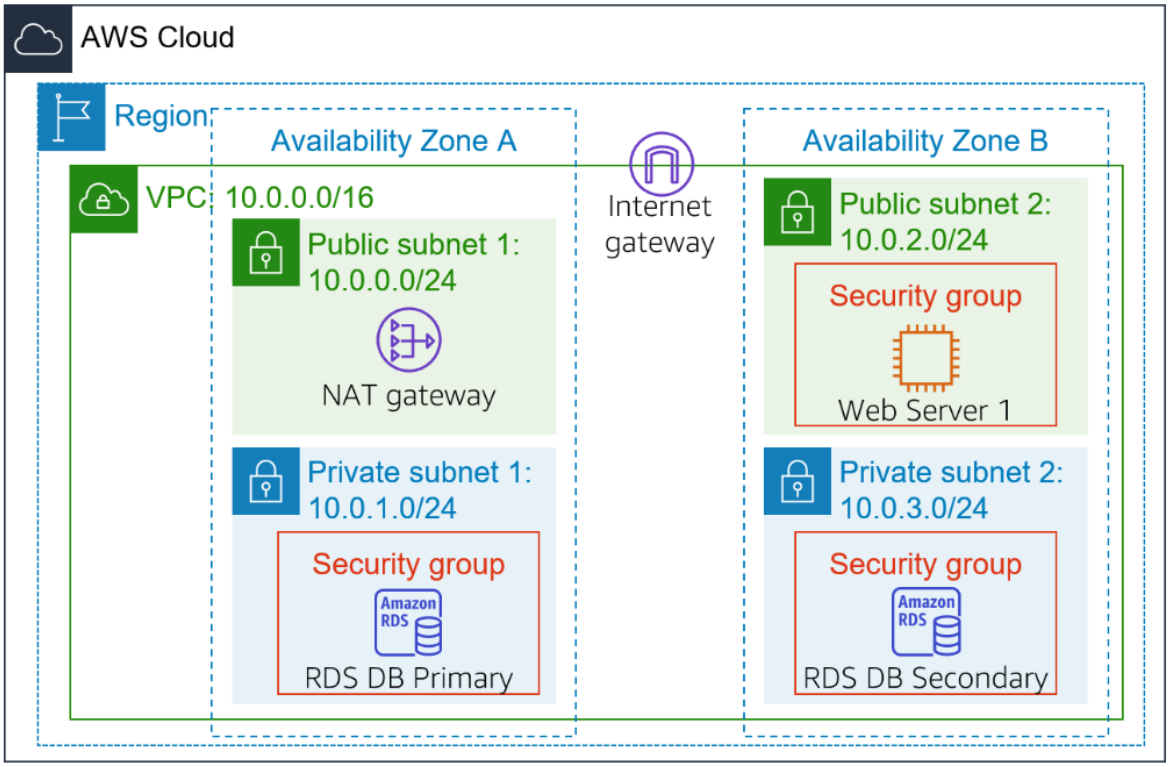
AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

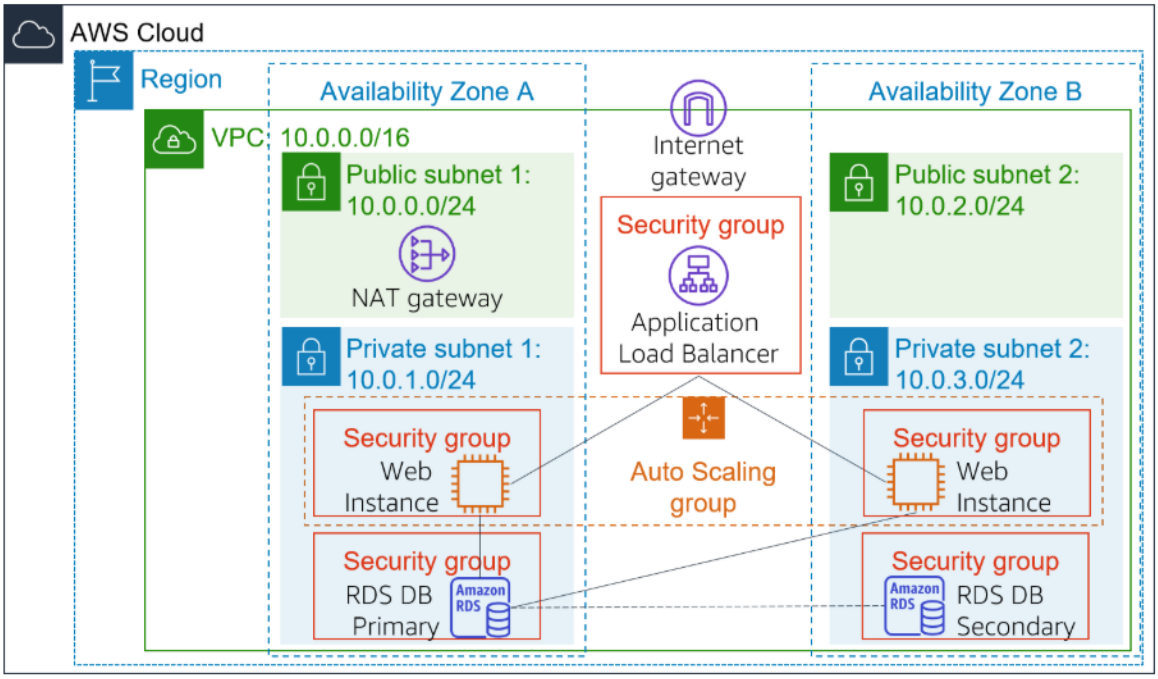
Caution: Any attempt to have 20 or more concurrently running instances (regardless of size), will result in immediate deactivation of the AWS account and all resources in the account will be immediately deleted.

Scenario

You start with the following infrastructure:



The final state of the infrastructure is:



B. Accessing the AWS Management Console

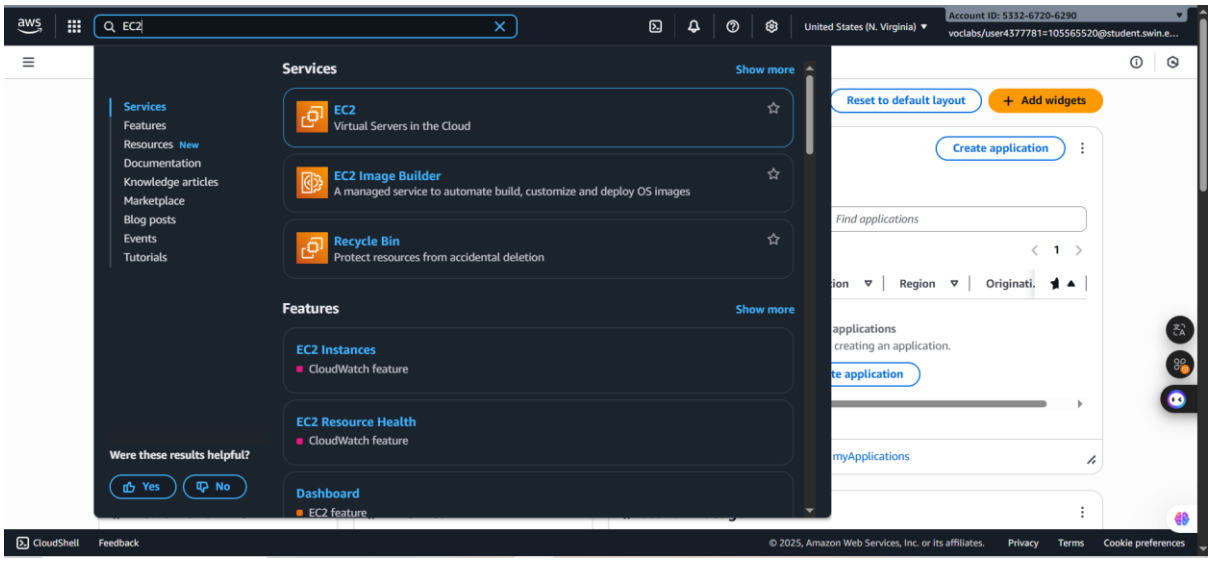
To access the **AWS Management Console**, clicking the button ▸ Start Lab. and wait until the circle next to the **AWS** turns from yellow to green

AWS

Figure 1: AWS Management Console activate

C. Task 1: Create an AMI for Auto Scaling

Step 1.1: In the serach box next to the **Services and its status**, search and select



the **EC2**

Figure 2: Search and select the EC2

Step 1.2: After accessing the ***EC2*** successfully, choose the Instances in the left navigation pane and makes sure that the ***Status check*** of the ***Web Server 1*** displays *2/2 checks passed.* Then, tick the box next to the *2/2 checks passed*

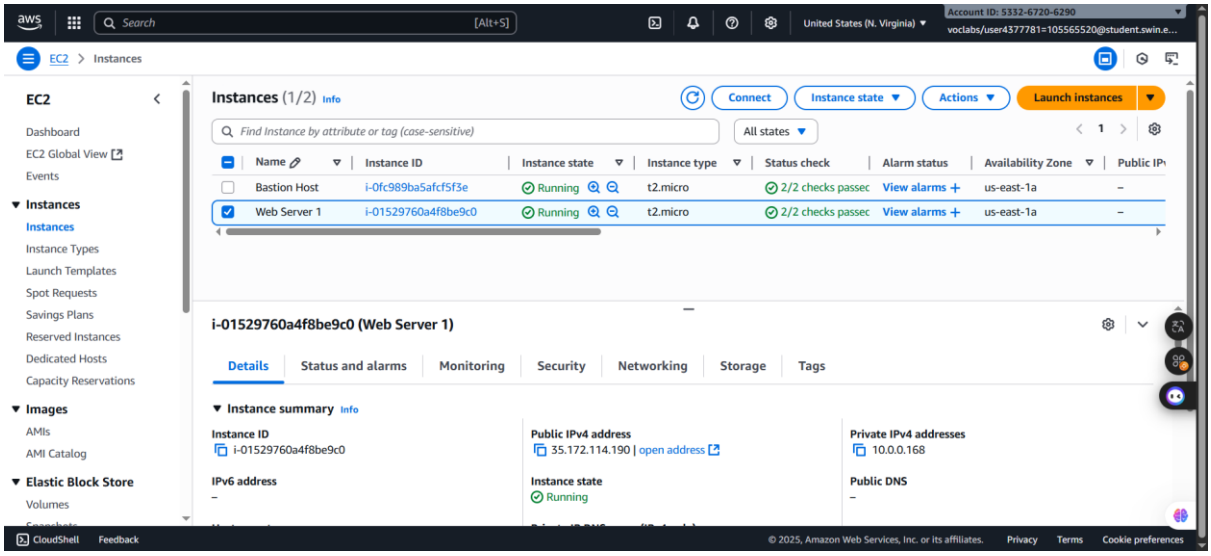


Figure 3: Weeb Server 1 selected

Step 1.3: In the next step, choose **Actions** => **Images and templates** => **Create image**. Then, configure based following requiremenrs:

- **Image name:** WebServerAMI
- **Image description:** Lab AMI for Web Server

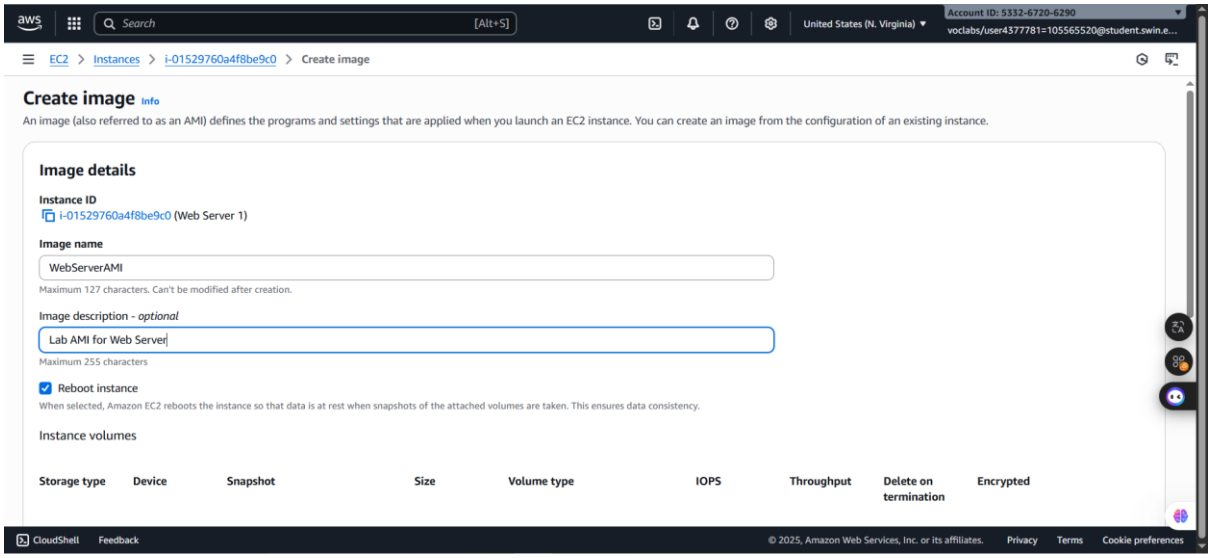


Figure 4: Create Image configuration

Step 1.4: After confirming that these settings are correct, click the button Create Image on the bottom of the configuration window and wait until the announcement that created successfully.

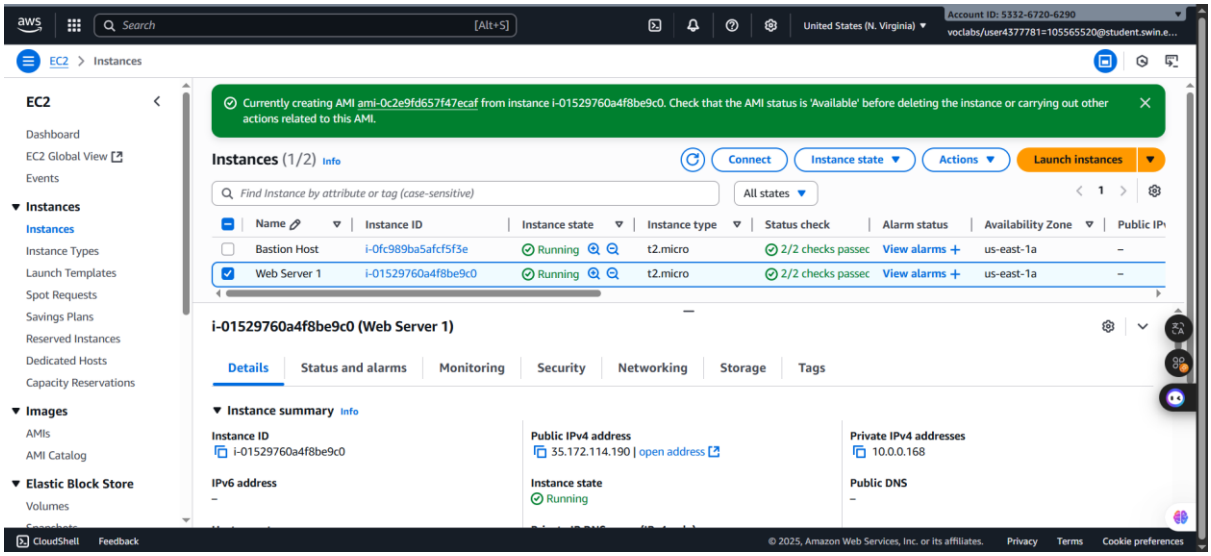


Figure 5: Create successfully

D. Task 2: Create a Load Balancer

In this task, you will first create a target group and then you will create a load balancer that can balance traffic across multiple EC2 instances and Availability Zones.

Step 2.1: In the left naviagtion pane, choose **Target Groups**, select the **Create Target group** and configure based on following requirements in the ***Step 1: Specify group details***

- Choose a target type: **Instances**
- **Target group name**, enter: LabGroup
- Select **Lab VPC** from the **VPC** drop-down menu.

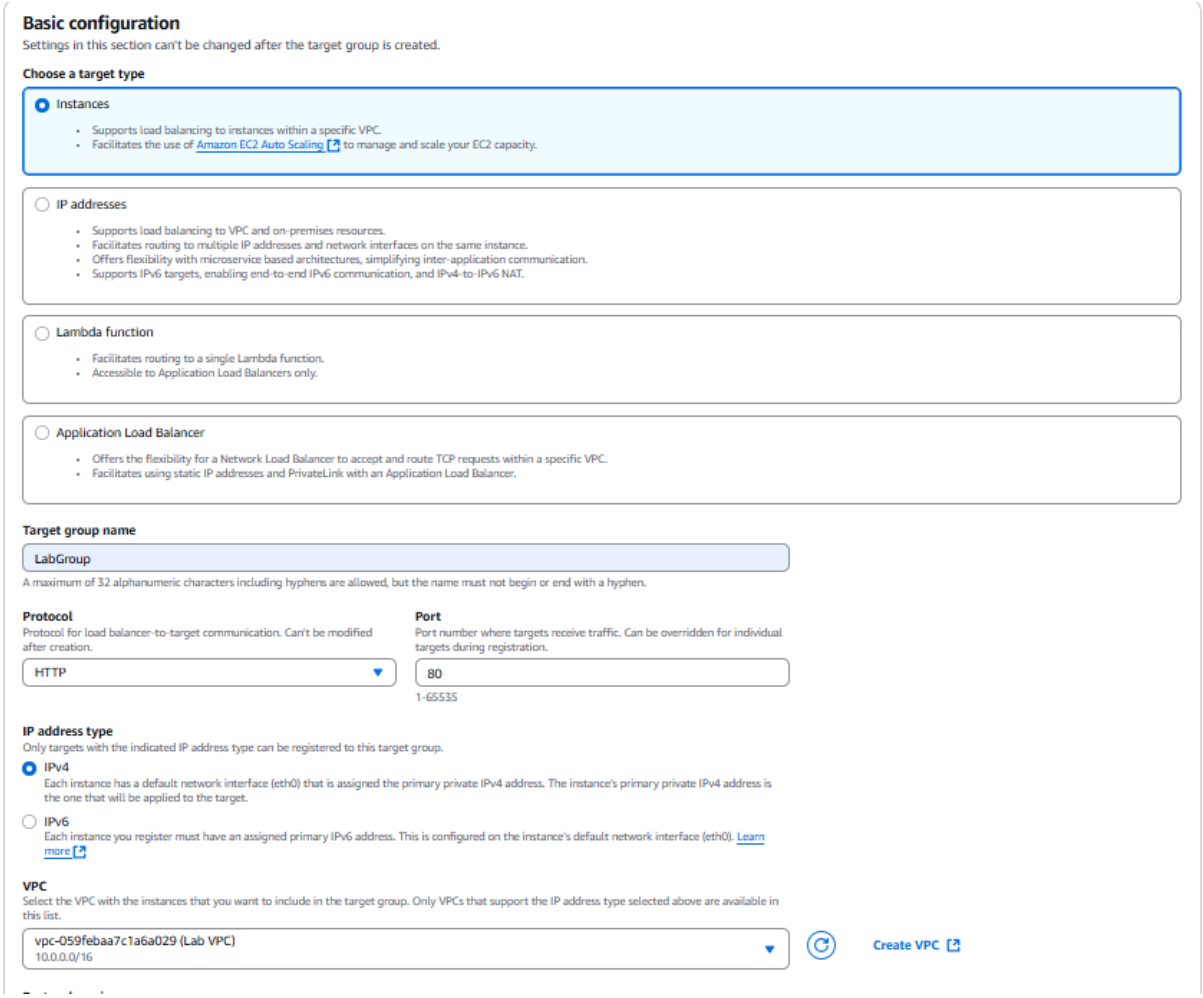


Figure 6: Basic Configuration

Step 2.2: After completing the previous step, move to ***Step 2: Register targets***. However, no configuration in this step because there is not any web application instances yet.

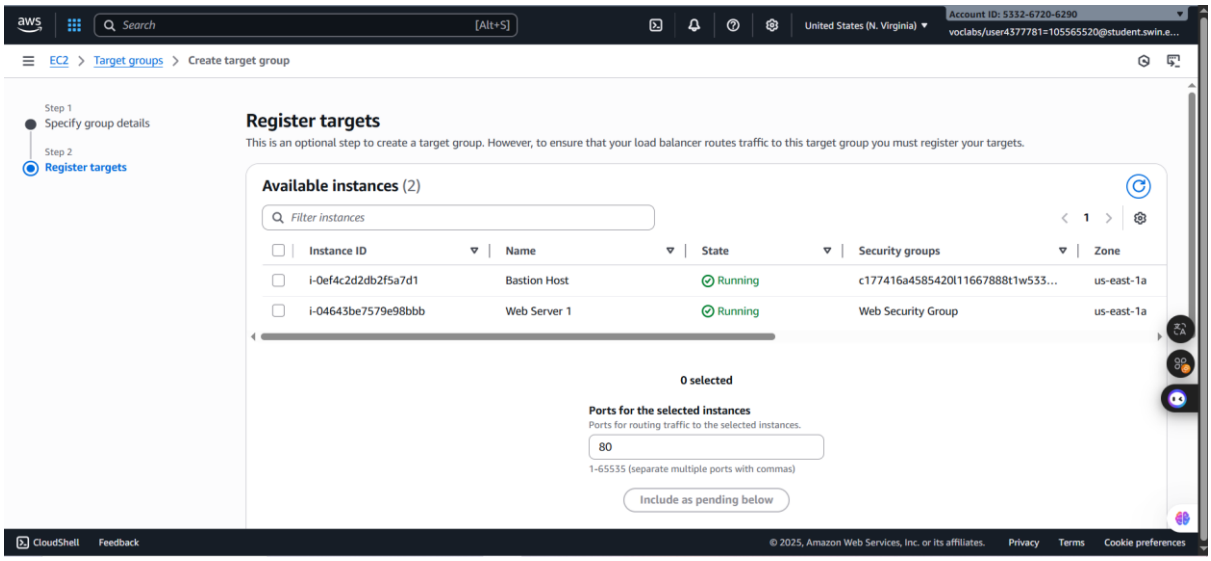


Figure 7: Register Targets window

Step 2.3: By checking all configurations are correct, choose the Create target groups and wait until the notification dusplay created successfully.

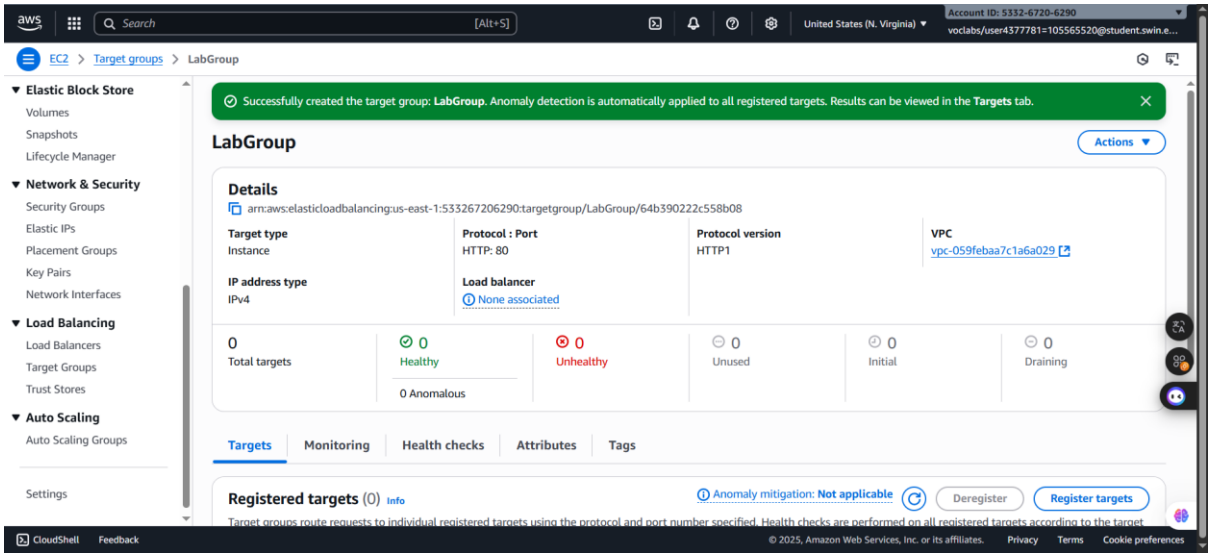


Figure 8: Created successfully

Step 2.4: In the left navigation pane, choose **Load Balancers** and select the button **Create Load Balancer.** After that, configure step-by-step with following settings:

- Under **Application Load Balancer**, choose **Create**

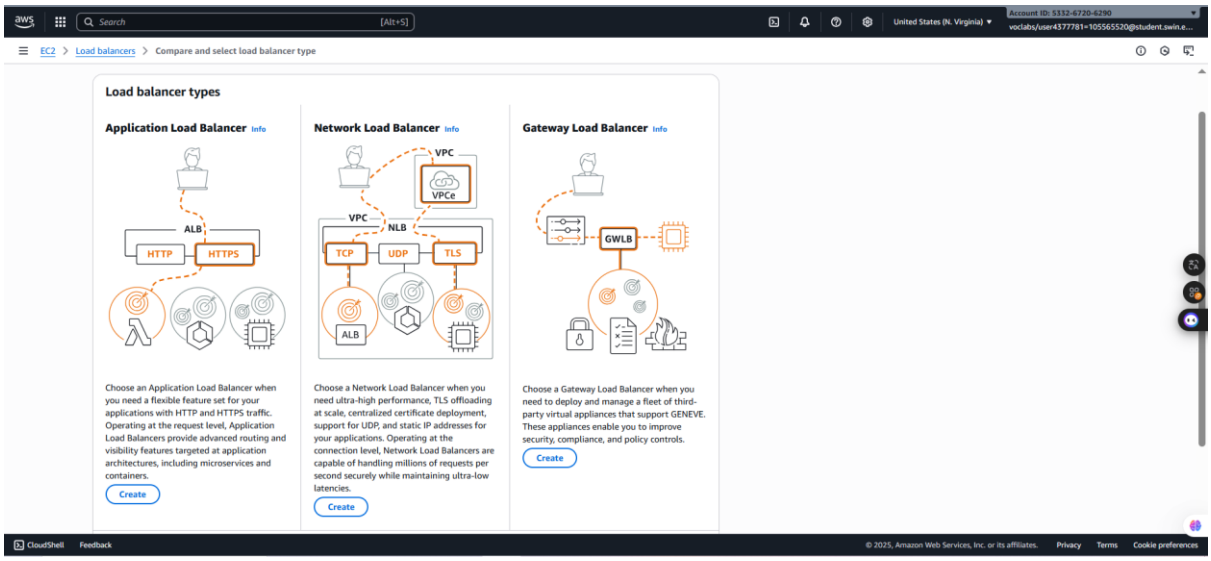


Figure 9: Application Load Balancer selected

- Under **Load balancer name**, enter: **LabELB**

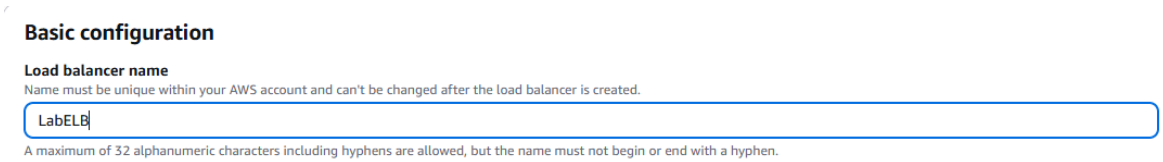


Figure 10: Set Load Balancer name

- Scroll down to the **Network mapping** section:
  - For **VPC**, choose **Lab VPC**
  - Choose the **first** displayed Availability Zone, then select **Public Subnet 1** from the Subnet drop down menu that displays beneath it.
  - Choose the **second** displayed Availability Zone, then select **Public Subnet 2** from the Subnet drop down menu that displays beneath it.

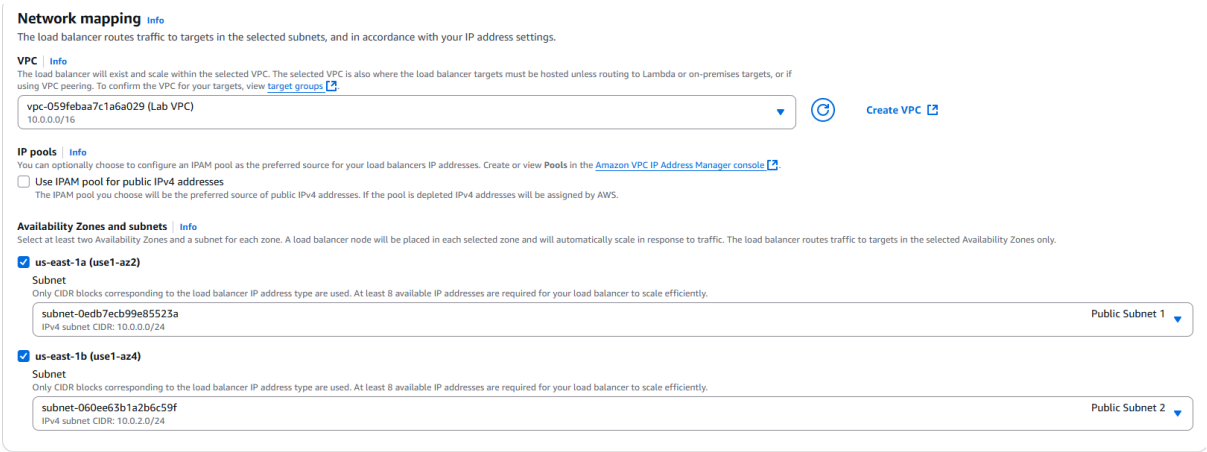


Figure 11: Network mapping configuration

- In the Security groups section, select the **Web Security Group** and deselect the **default**

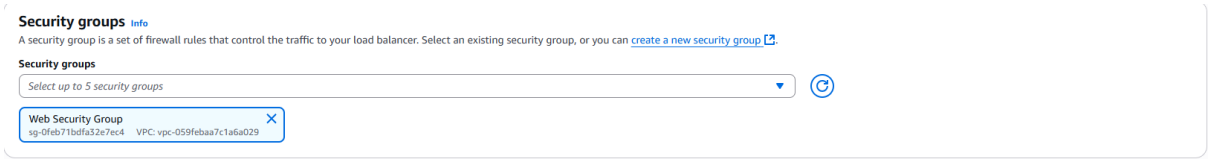


Figure 12: Security Groups configuration

- For the Listener HTTP:80 row, set the Default action to forward to **LabGroup.**

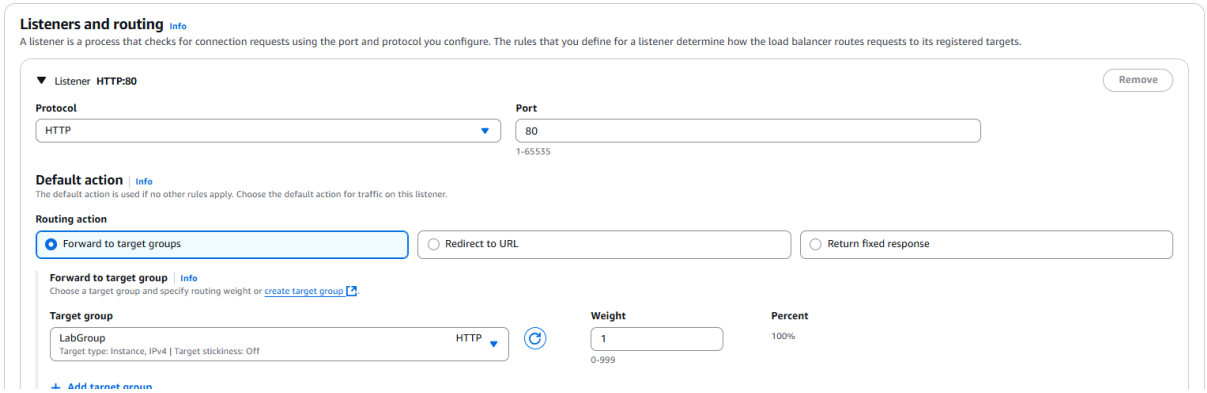


Figure 13: Target and routign configuration

Step 2.5: After checking all the configuration in each part is correct, click the **Create load balancer** button to view and wait until the successful notification.

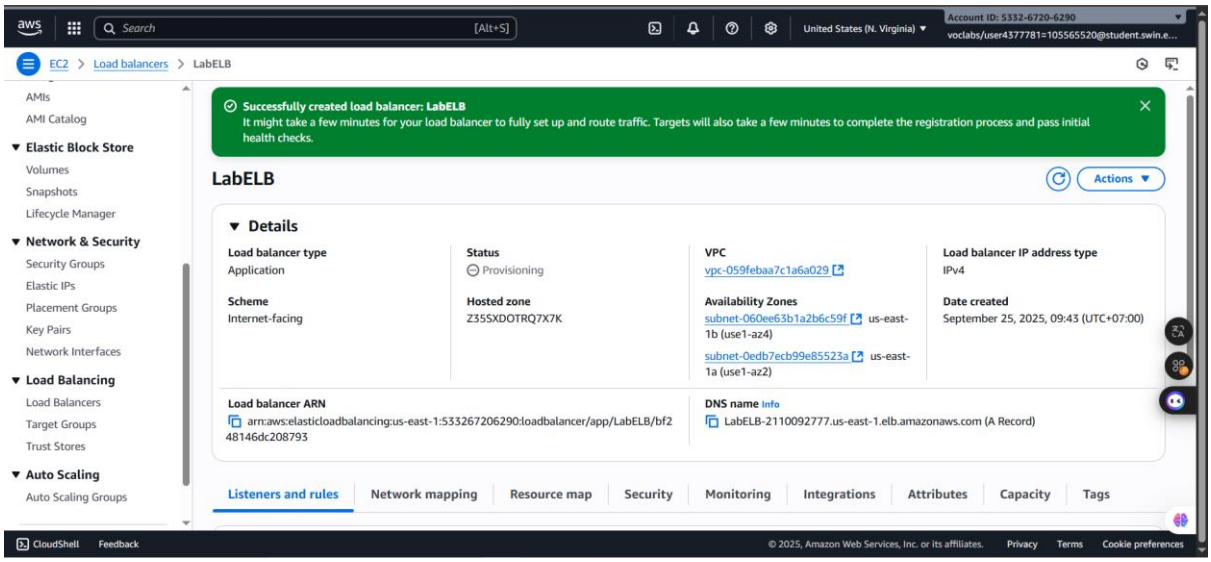


Figure 14: Successfully created

E. Task 3: Create a Launch Template and an Auto Scaling Group

In this task, you will create a *launch template* for your Auto Scaling group. A launch template is a template that an Auto Scaling group uses to launch EC2 instances. When you create a launch template, you specify information for the instances such as the AMI, the instance type, a key pair, and security group.

Step 3.1: In the left navigation pane, choose the ***Lauch Templates*** and select the ***Create launch teplate*** button

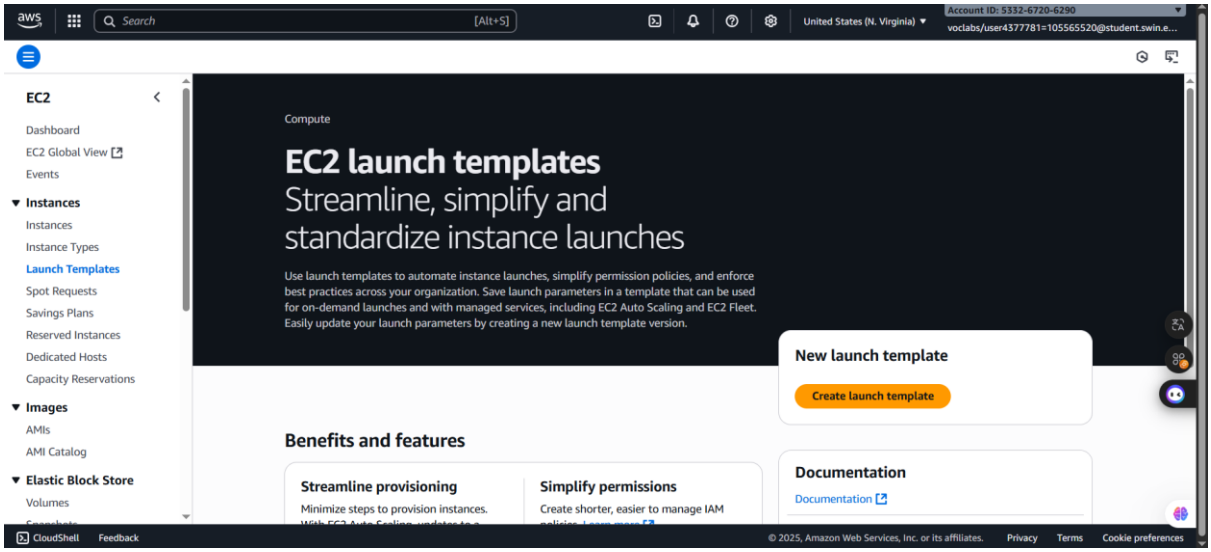


Figure 15: Launch template window

Step 3.2: Configure these requirement settings:

- **Launch template name**: LabConfig
- Under **Auto Scaling guidance**, select *Provide guidance to help me set up a template that I can use with EC2 Auto Scaling*

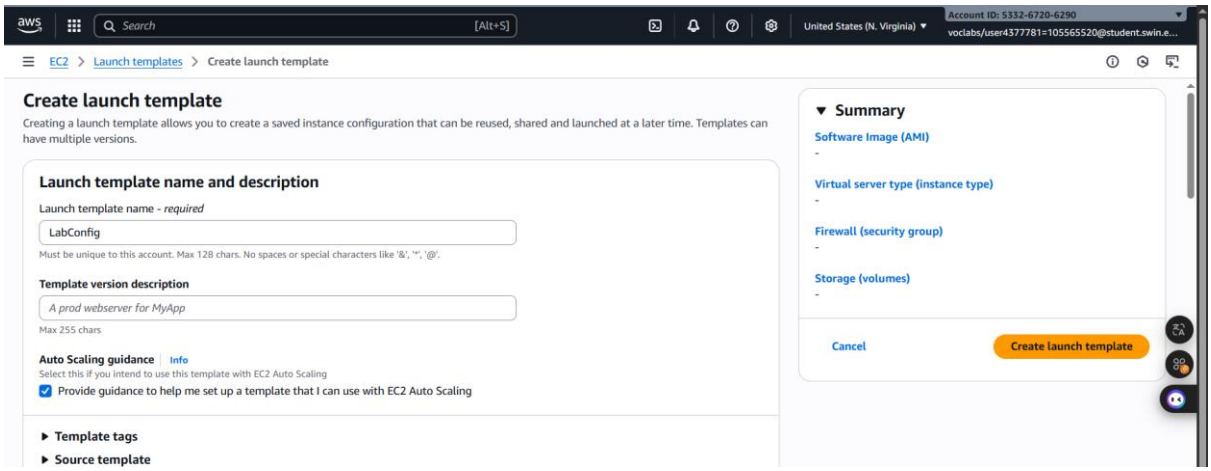


Figure 16: Launch template name and description configuration

- In the **Application and OS Images (Amazon Machine Image)** area, choose My AMIs.

- **Amazon Machine Image (AMI):** choose Web Server AMI

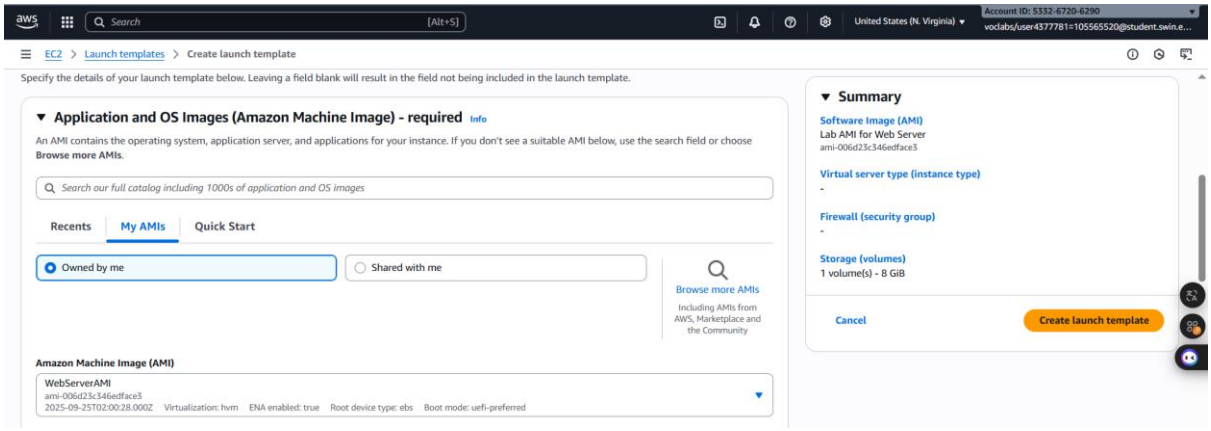


Figure 17: Application and OS images (Amazon Machine Image) – required configuration

- **Instance type**: choose t2.micro

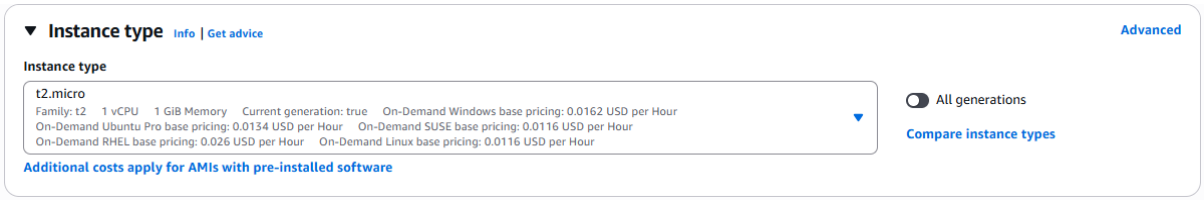


Figure 18: Instance type configuration

- **Key pair name**: choose vockey

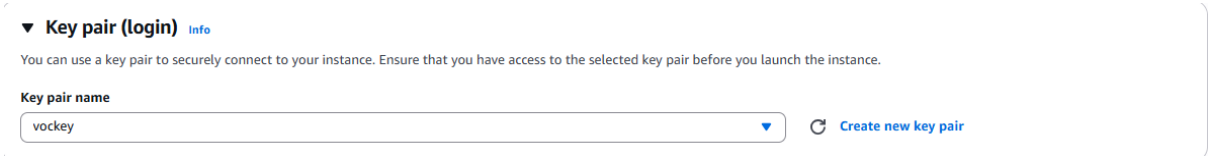


Figure 19: Key pair configuration

- **Firewall (security groups):** choose Select existing security group
- **Security groups**: choose Web Security Group

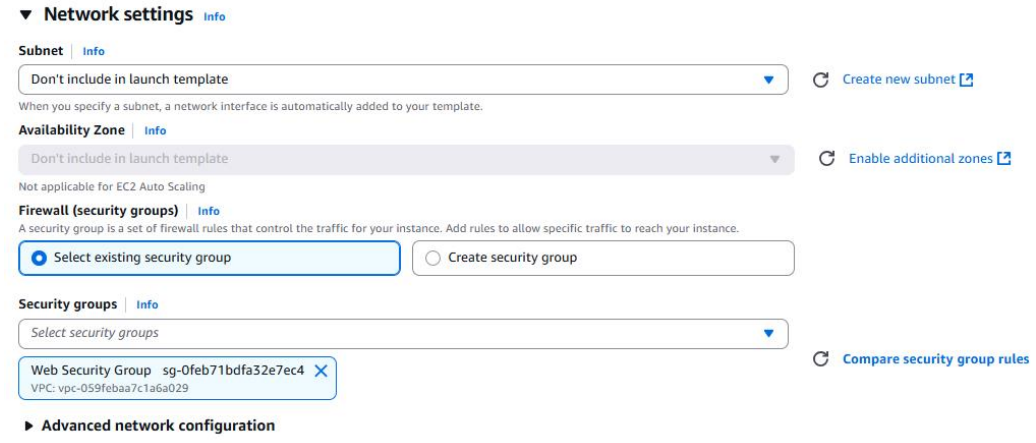


Figure 20: Netwoork Settings configuration

- Scroll down to the **Advanced details** area and expand it.
- Scroll down to the **Detailed CloudWatch monitoring** setting. Select Enable

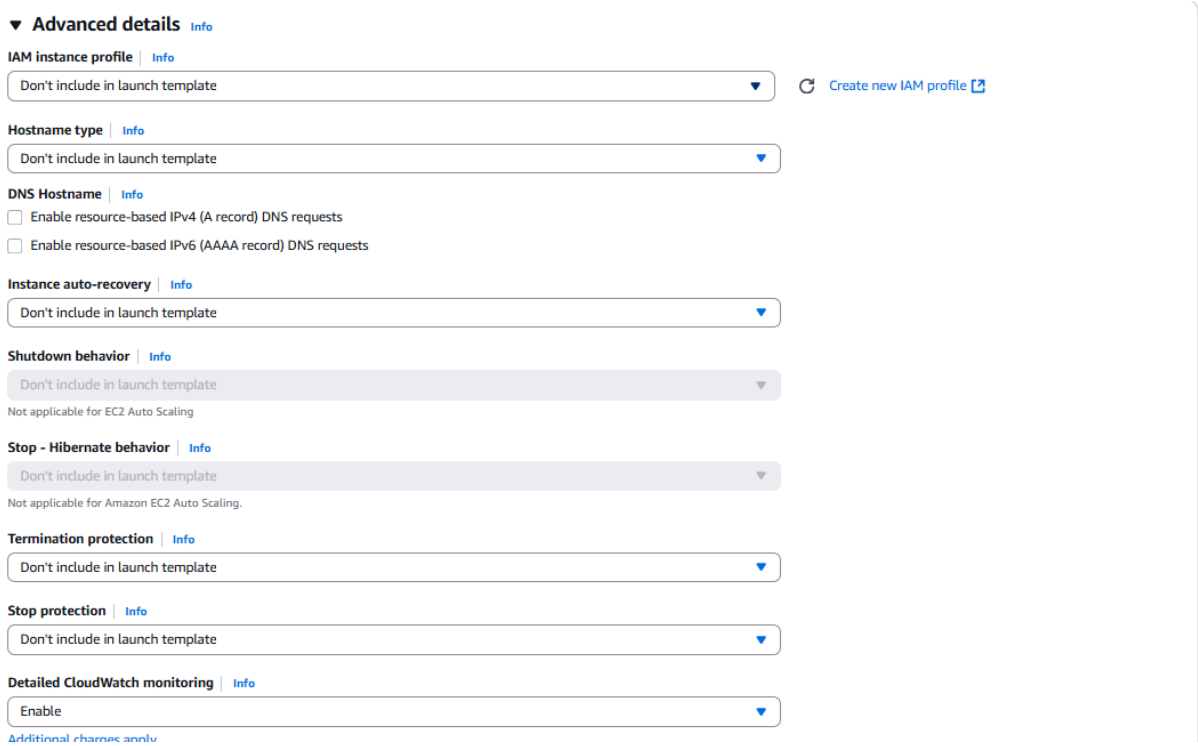


Figure 21: Advanced details configuration

Step 3.3: By making sure that all configuration settings are correct, click the button Create launch template and then window display the successful notification

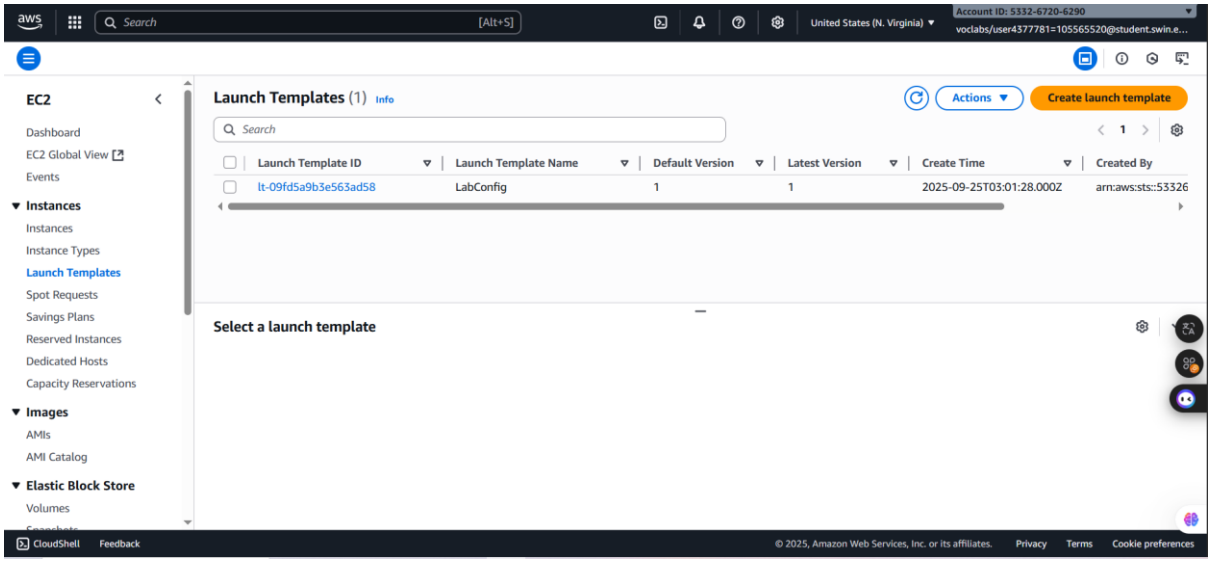


Figure 22: Launch Template successfully

Step 3.4: Then, select the previous template which has created, choose **Actions** and select the **Create Auto Scaling group**

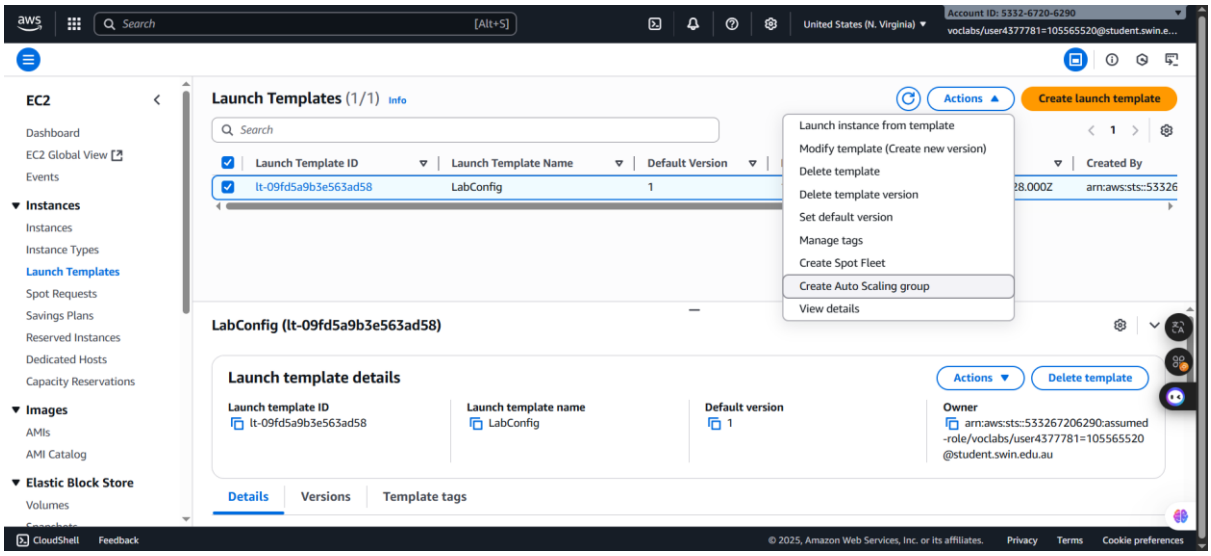


Figure 23: Create Auto Scaling group selection

Step 3.5: Configure the details in **Step 1** (Choose launch template or configuration):

- **Auto Scaling group name**: Lab Auto Scaling Group
- **Launch template**: confirm that the LabConfig template you just created is selected.

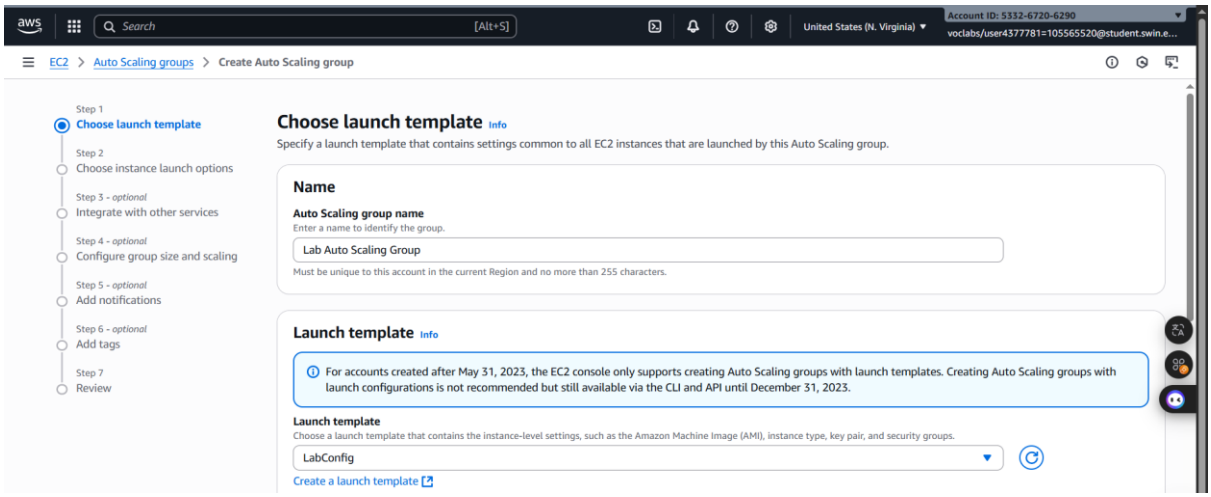


Figure 24: Step 1 configuration

Step 3.6: Configure the details in Step 2 (Choose instance launch options):

- VPC: choose Lab VPC
- Availability Zones and subnets: Choose Private Subnet 1 and then choose Private Subnet 2.

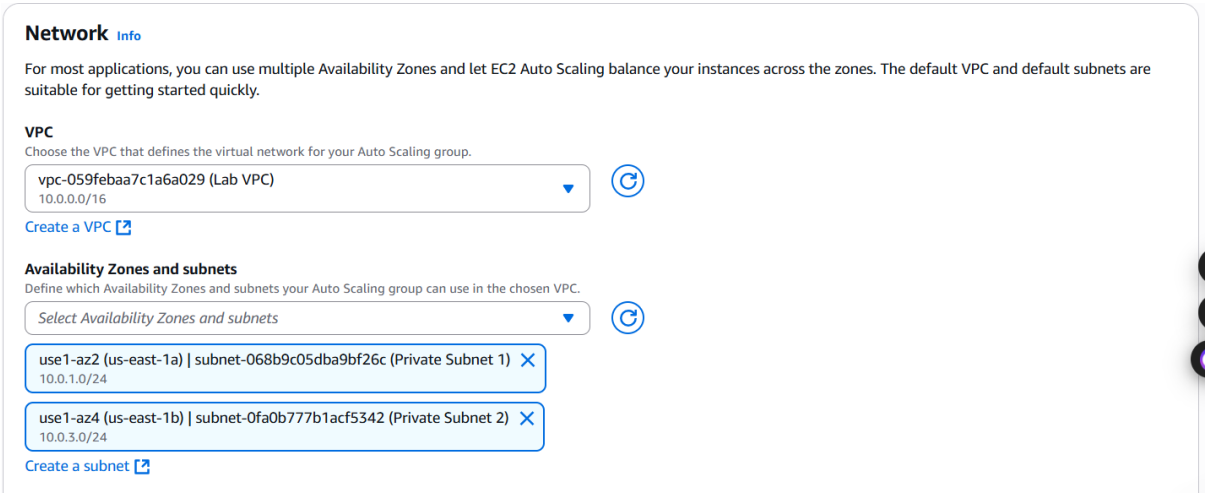


Figure 25: Step 2 configuration

Step 3.7: Configure the details in Step 3 (Configure advanced options):

- Choose **Attach to an existing load balancer**
  - **Existing load balancer target groups**: select LabGroup.

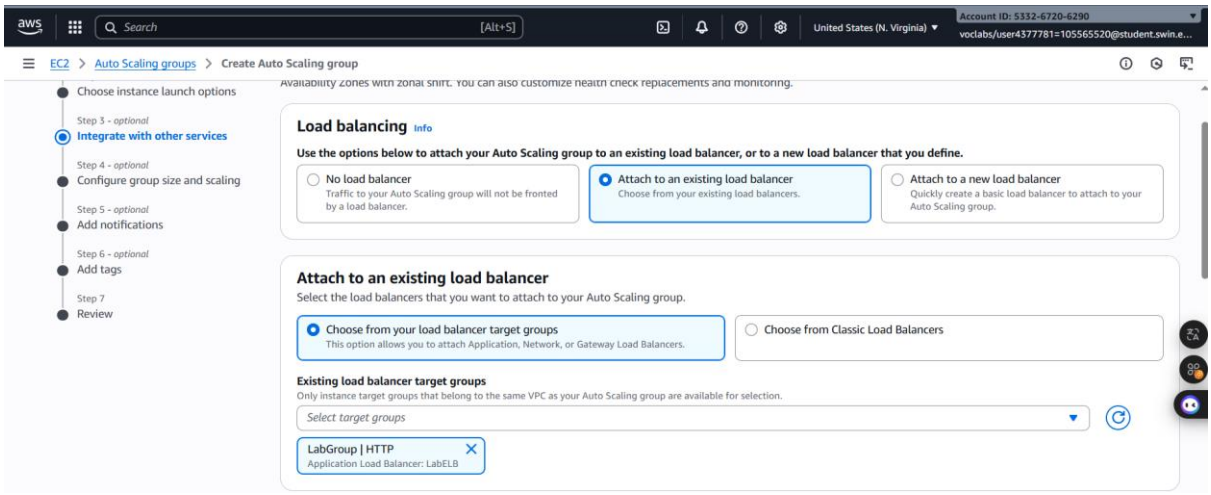


Figure 26: Step 3 configuration

Step 3.8: Configure the details in Step 4 (Configure group size and scaling policies - optional):

- Under **Group size**, configure:
  - **Desired capacity**: 2
  - **Minimum capacity**: 2
  - **Maximum capacity**: 6
- Under **Scaling policies**, choose Target tracking scaling policy and configure:
  - **Scaling policy name**: LabScalingPolicy
  - **Metric type**: Average CPU Utilization
  - **Target value**: 60
- In the **Additional settings** pane:
  - Select **Enable group metrics collection within CloudWatch**

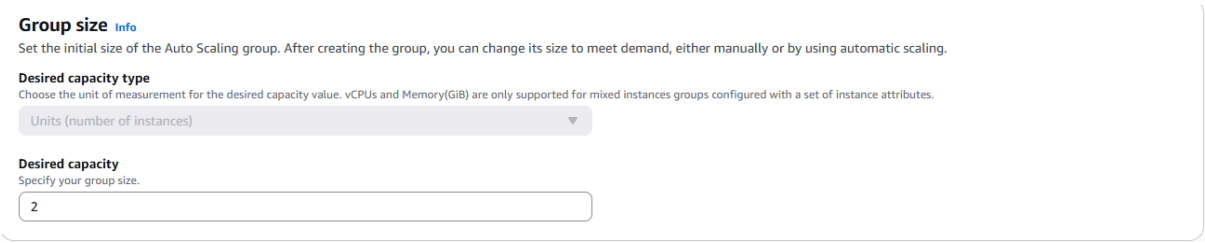


Figure 27: Group Size configuration

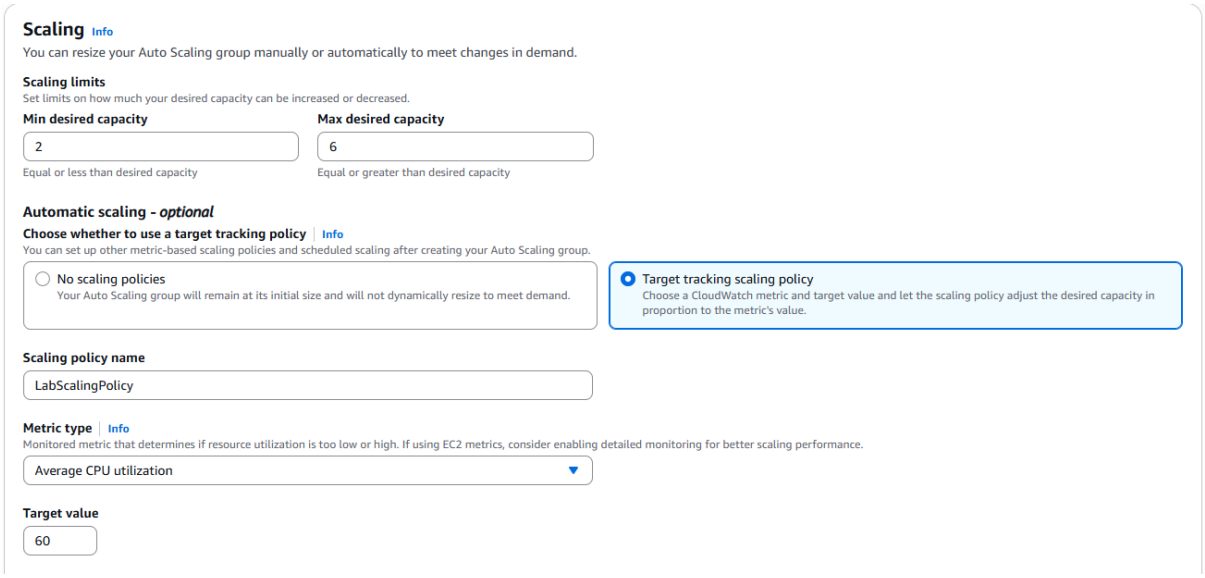


Figure 28: Scaling configuration

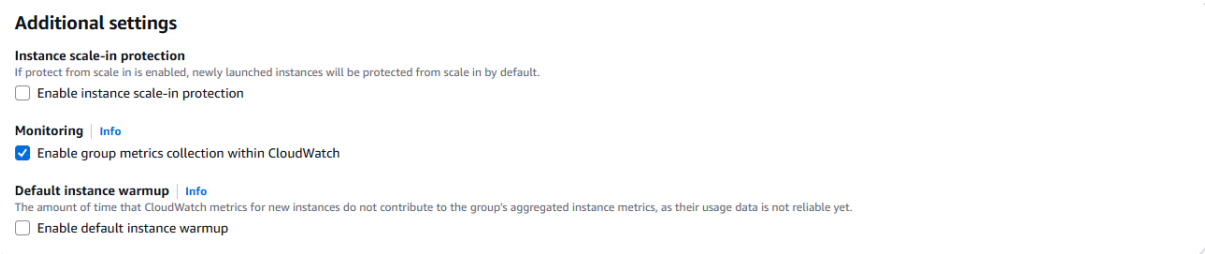


Figure 29: Additional Settings configuration

Step 3.9: Configure the details in Step 5 (Add notifications - optional):

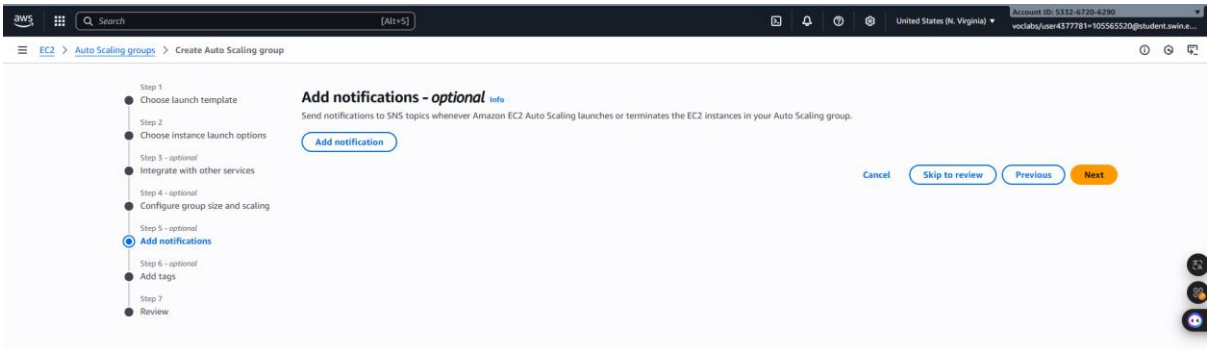


Figure 30: Step 5 configuration

Step 3.10: Configure the details in Step 6 (Add tags – optional):

- Choose **Add tag** and Configure the following:
  - **Key:** Name
  - **Value:** Lab Instance
- Choose **Next**

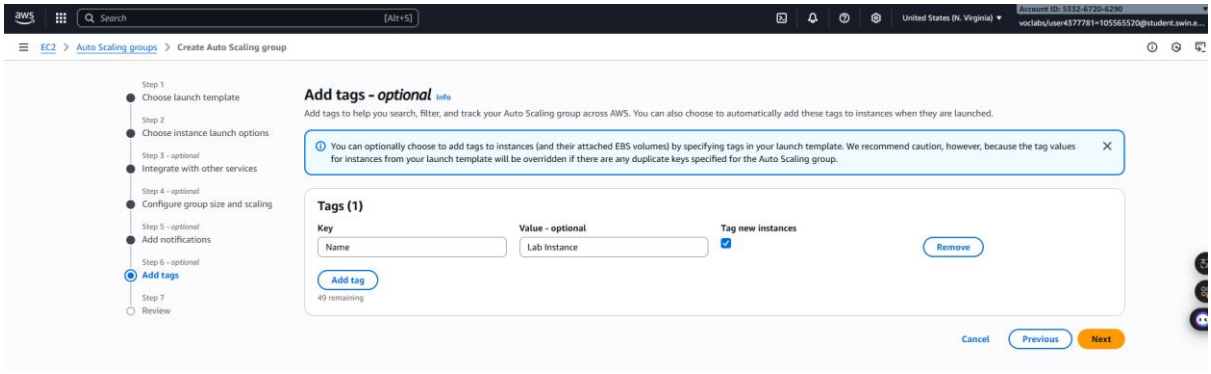


Figure 31: Tag configuration

Step 3.11*:* After making sure that all configurations are correctly, click the button **Create Auto Scaling group** to create:

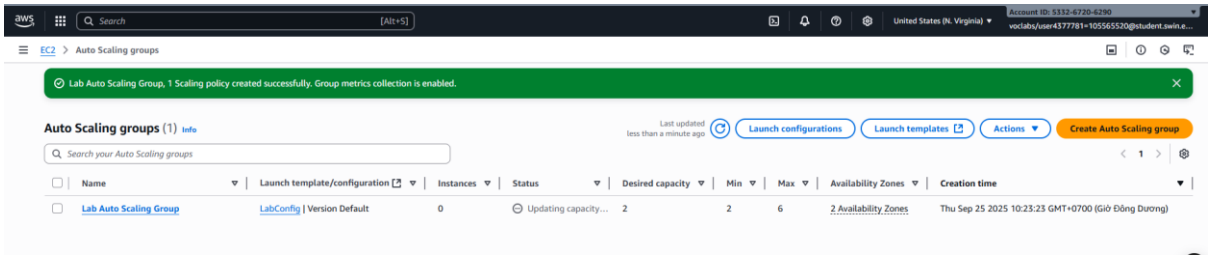


Figure 32: Created successfully

F. Task 4: Verify that Load Balancing is Working

In this task, you will verify that Load Balancing is working correctly.

Step 4.1*:* In the left navigation pane, choose Instances and view that there are two new instances named ***Lab Instance.***

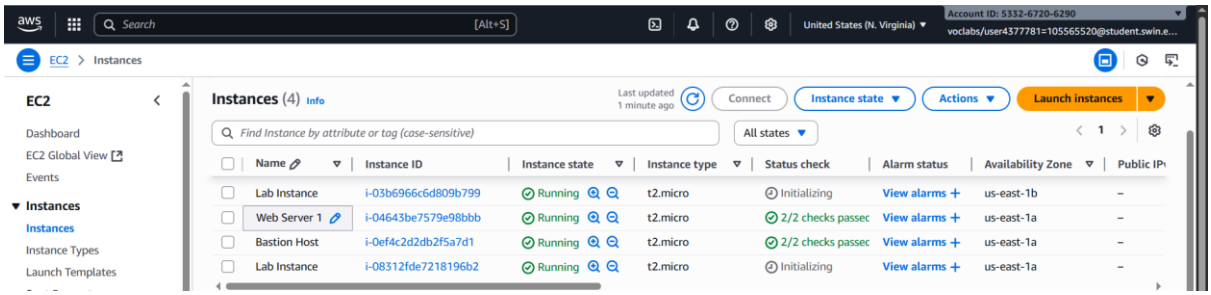


Figure 33: EC2 mainpage

Step 4.2: In the left navigation pane, choose **Target Groups** and select **LabGroup.** In the Details pane, check the status is **Healthy** or not.

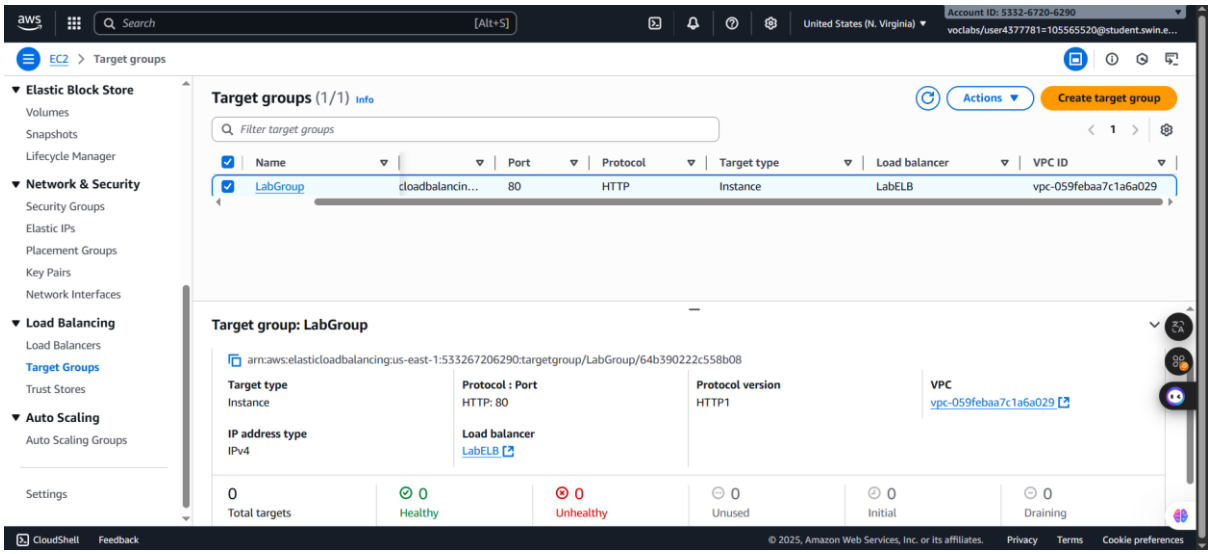


Figure 34: Healthy checked

Step 4.3: Then, In the left navigation pane, choose the **Load Balancers.,** and select the **LabELB,** copy **the DNS name of the load balancer** and paste It into a new browser tab

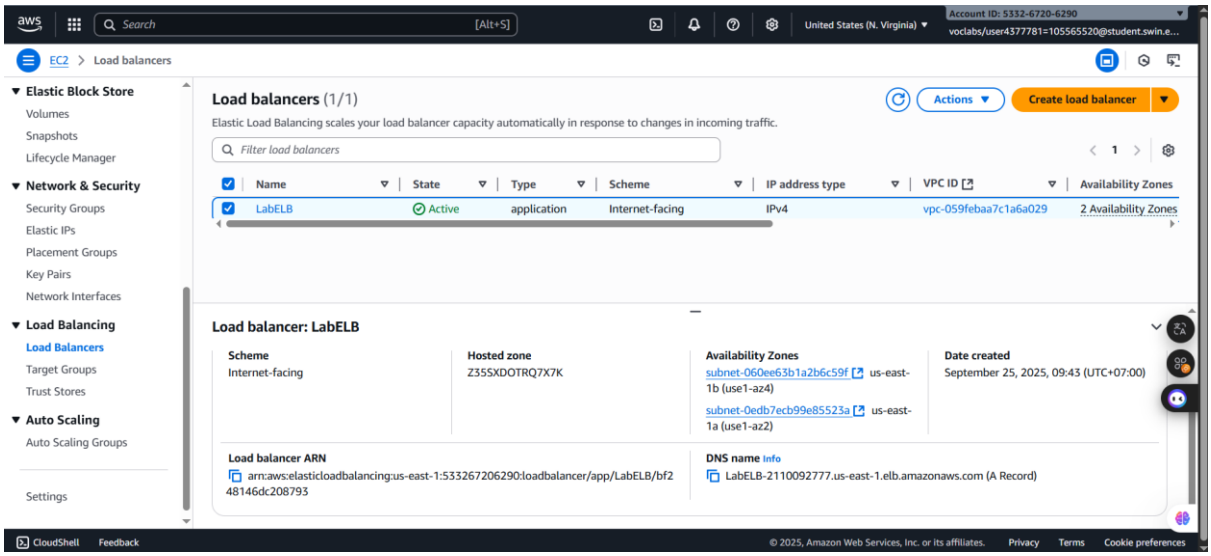


Figure 34: LabELB selected

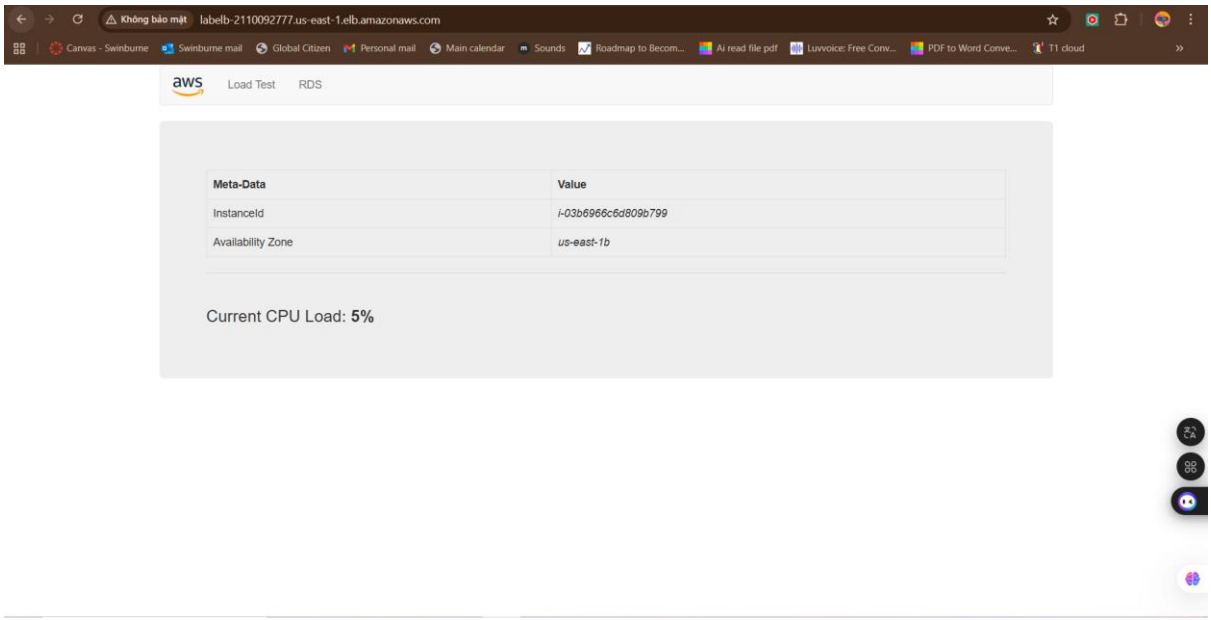


Figure 35: DNS link result

G. Task 5: Test Auto Scaling

You created an Auto Scaling group with a minimum of two instances and a maximum of six instances. Currently two instances are running because the minimum size is two and the group is currently not under any load. You will now increase the load to cause Auto Scaling to add additional instances.

Step 5.1: Return to the AWS Management Console and do not close the application tab. Then, search and choose the **CloudWatch**.

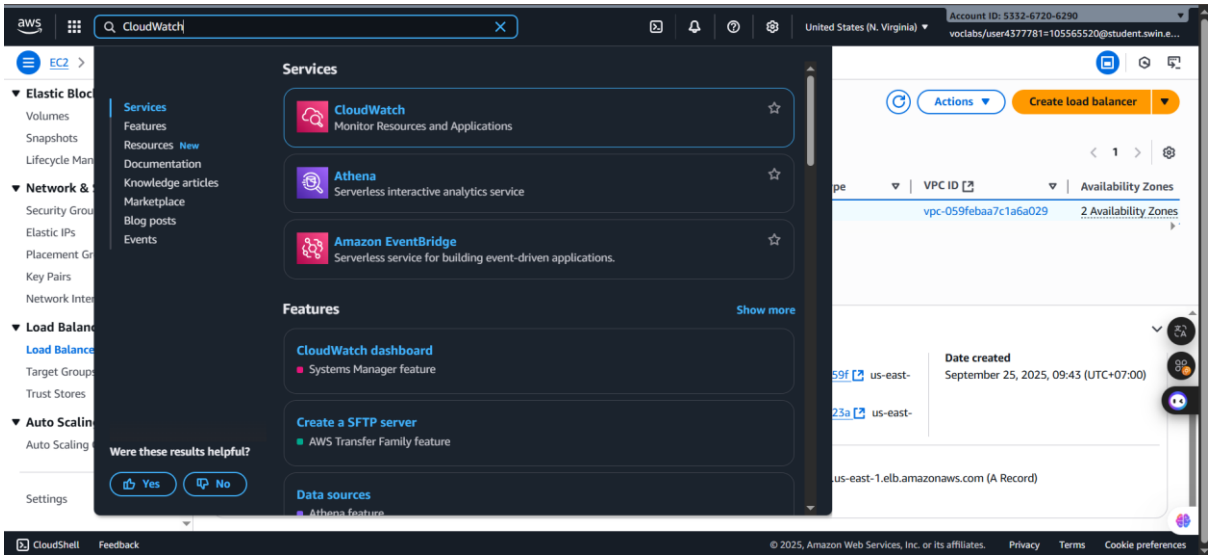


Figure 36: Search and select the CloudWatch

***Step 5.2*:** In the left navigation pane, choose **All alarms.**

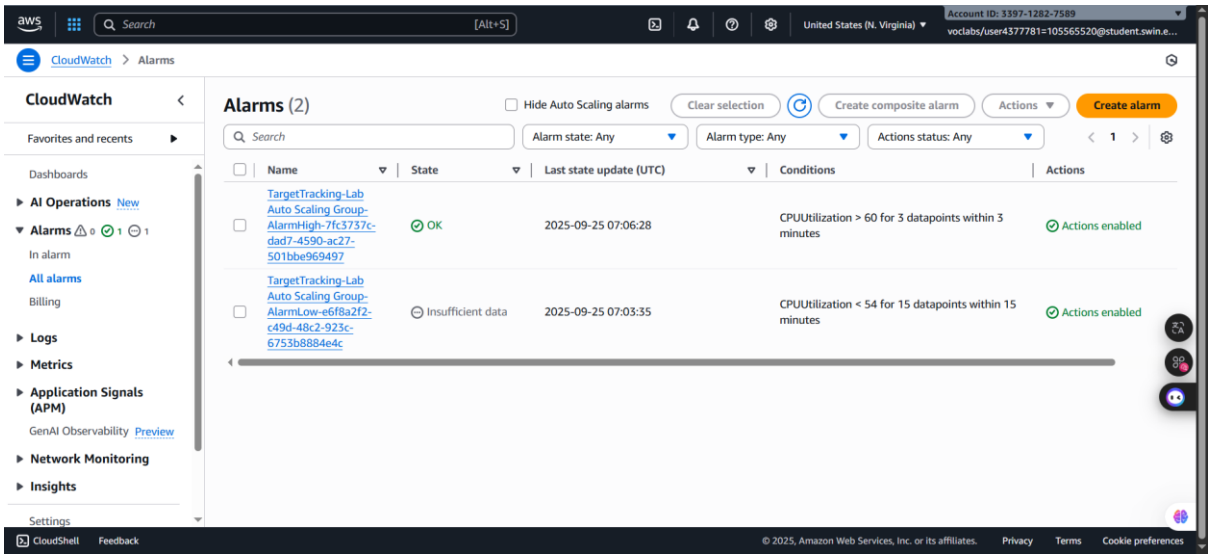


Figure 37: All alarms window

Step 5.3: On the **Services** menu, choose **EC2**

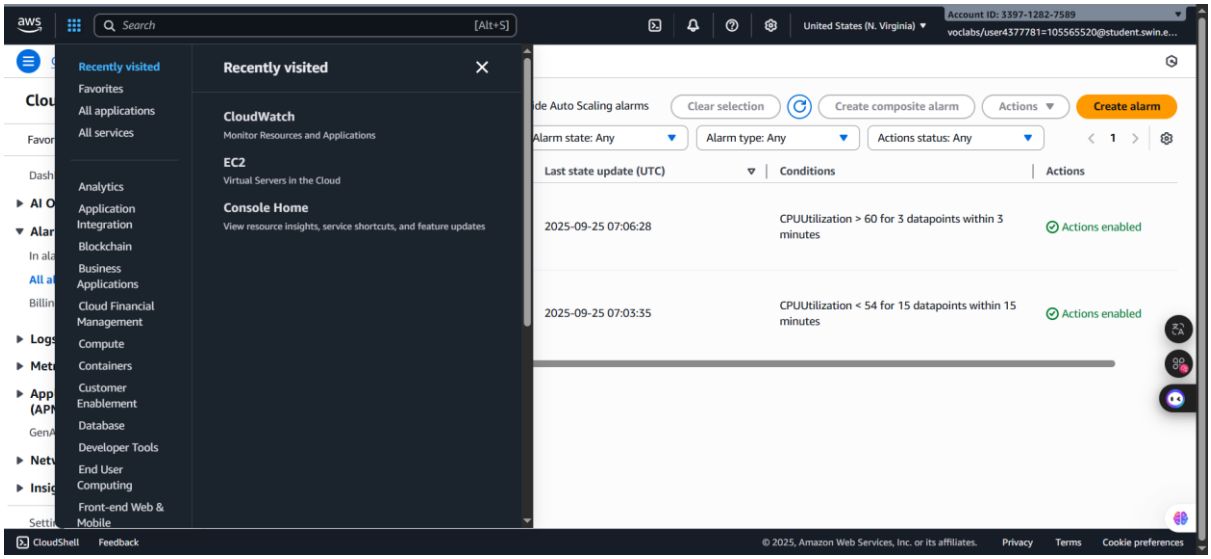


Figure 38: EC2 selection

Step 5.4: After accessing successfully to the ***EC2*** homepage, choose the ***Auto Scaling Groups*** in the left navigation pane.

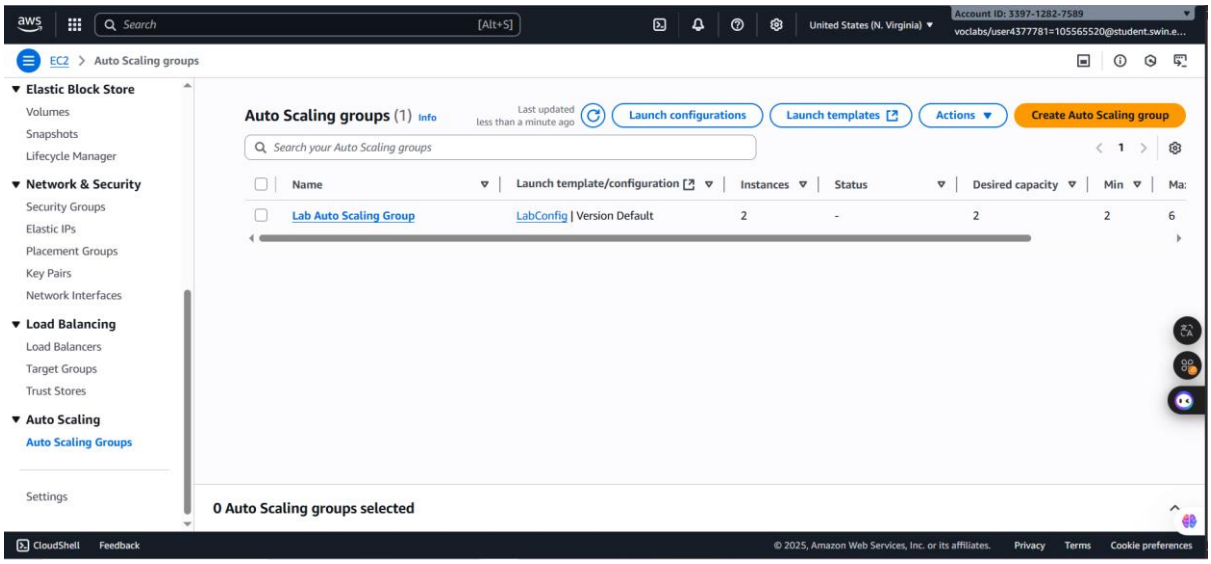


Figure 39: Auto Scaling Group window

Step 5.5: Tick the box next to the ***Lab Auto Scaling Group*** window and in the bottom half of the page, choose the ***Automatic Scaling*** tab. Next, Select **LabScalingPolicy**. Then, choose **Actions => Edit**

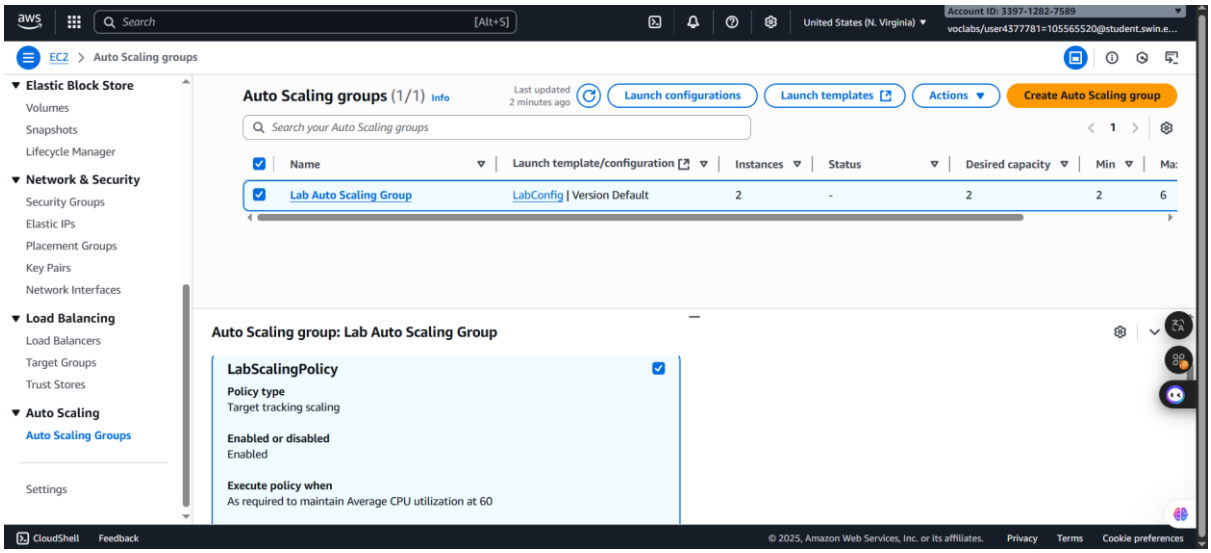


Figure 40: Confiure preparation

Step 5.6: Change the **Target Value** to **50** and click **Update**

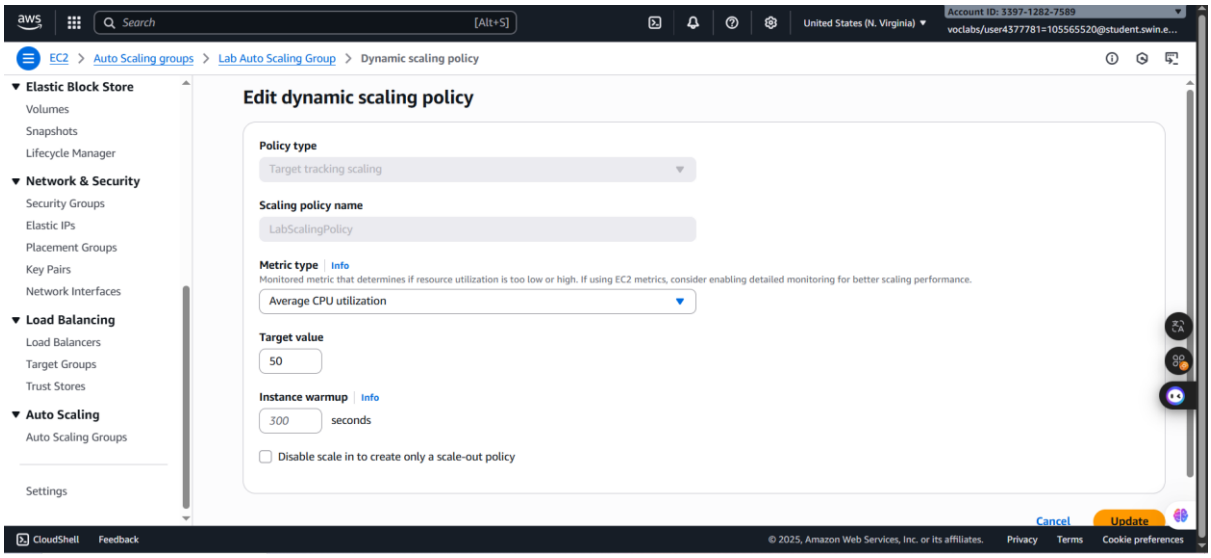


Figure 41: Target value configuration

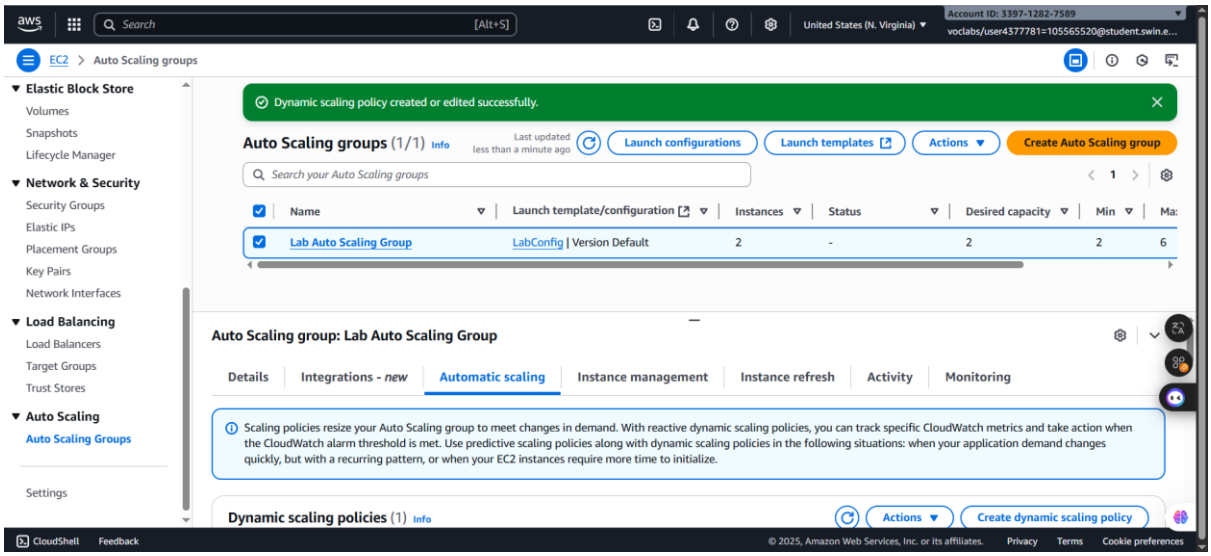


Figure 42: Target change successfully

Step 5.7: In the following step, choose the **Cloudwatch** in AWS Service, select **All alarms**, and tick the **AlarmHigh** with State is **OK**

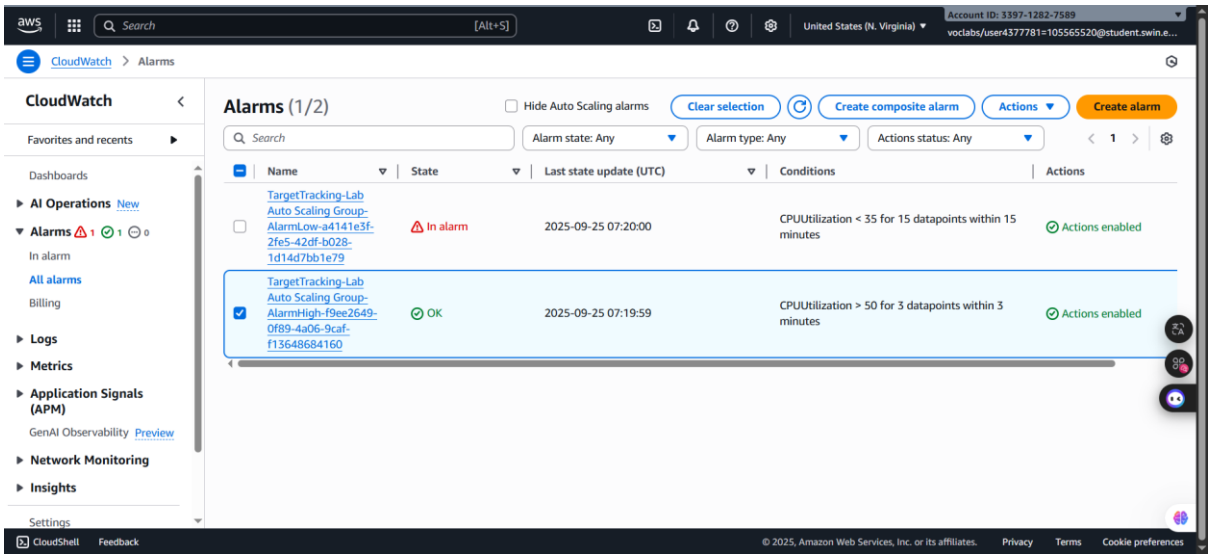


Figure 43: AlarmHigh selected

Step 5.8: Return to the browser tab with the web application and choose **Load Test** beside the AWS logo to makes sure the application to generate high loads

Figure 44: High loads

Step 5.9: Return to browser tab with the **CloudWatch** console. In less than 5 minutes, the **AlarmLow** alarm should change to *OK* and the **AlarmHigh** alarm status should change to *In alarm.*

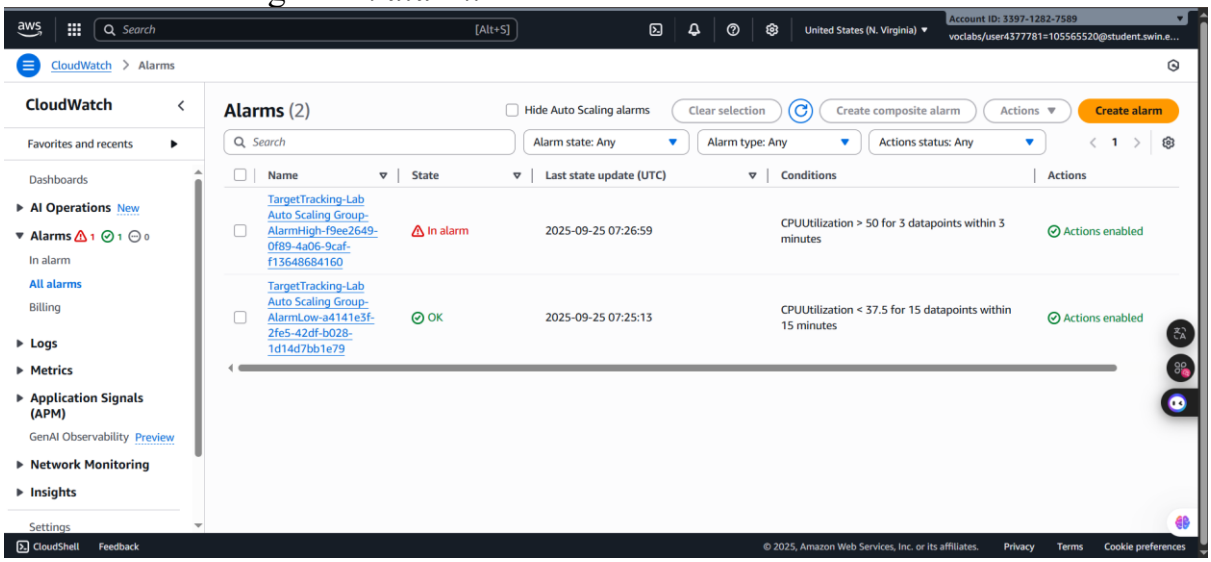


Figure 45: Alarm changed successfully

Step 5.10: Turn back to the EC2 and select the Insances to makes sure that there more then two instances labeled Lab Instance should now be running.

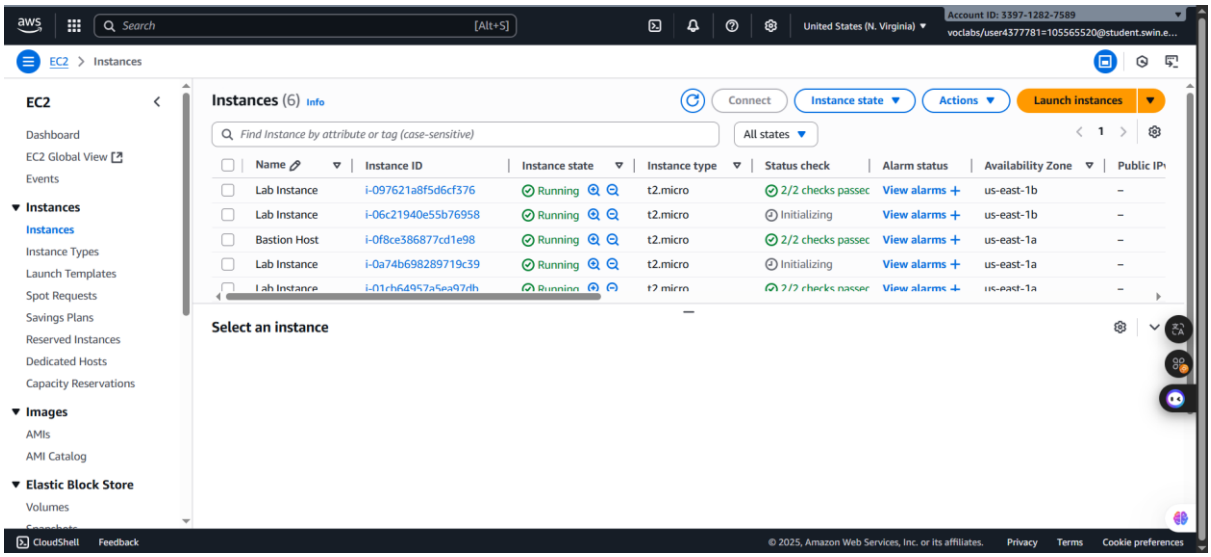


Figure 46: Instances confirmation

H. Task 6: Terminate Web Server 1

In this task, you will terminate *Web Server 1*. This instance was used to create the AMI used by your Auto Scaling group, but it is no longer needed.

Step 6.1: Select the ***Web Server 1,*** *then in the* ***Instance menu,*** choose ***Instance State => Terminate Instance.***

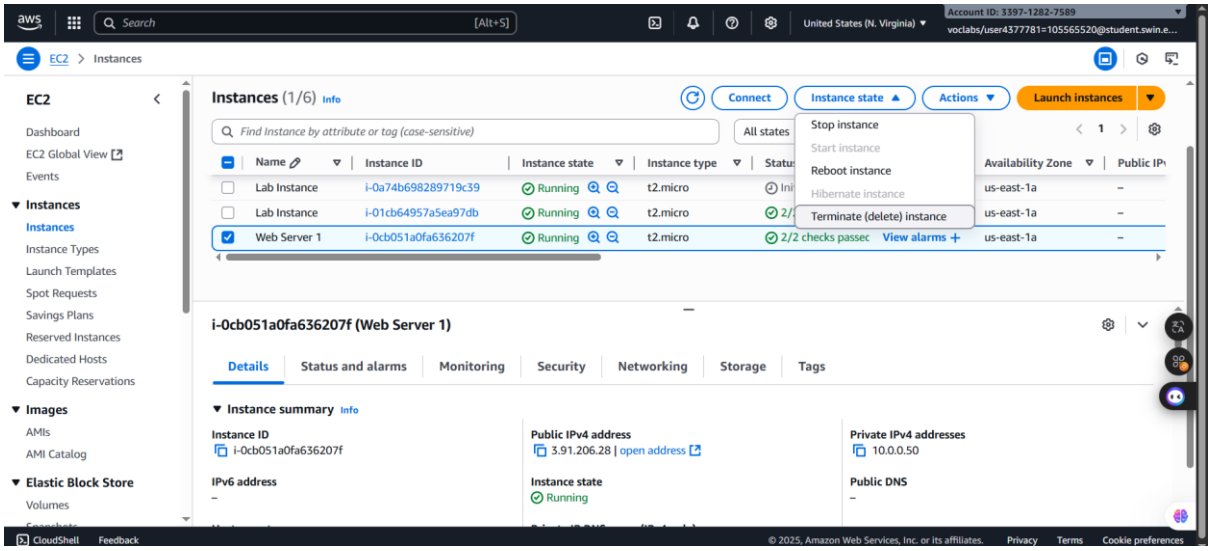


Figure 47: Instance state selection

Step 6.2: After clicking the Terminate Instance, the annuncement will display before terminating the ***Web Server 1*** instance

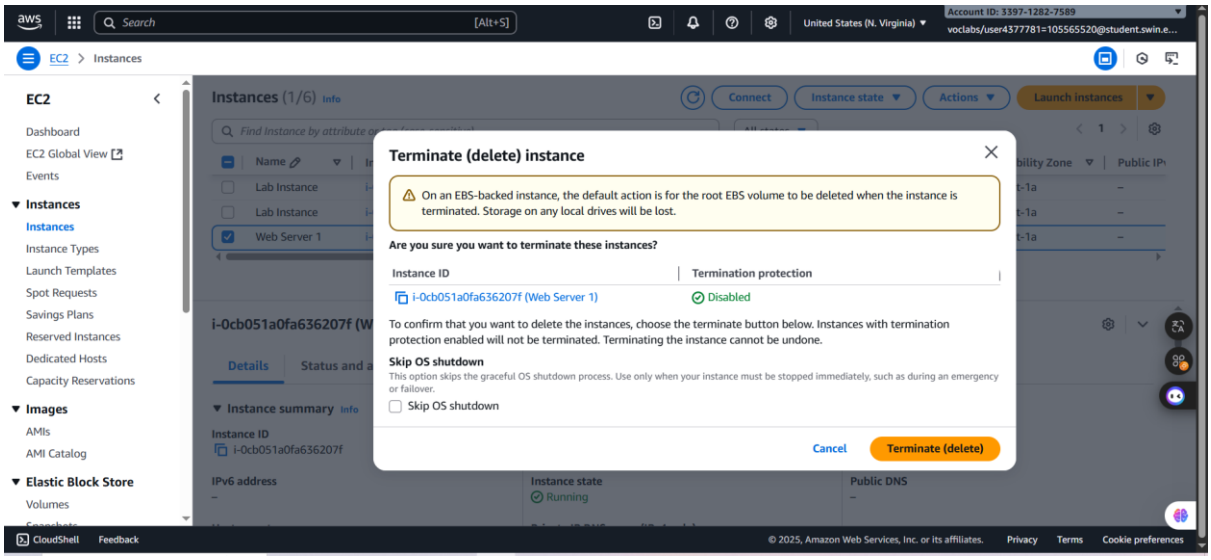


Figure 48: Announcement before terminating

Step 6.3: After choosing button ***Terminate,*** the instance will be terminated

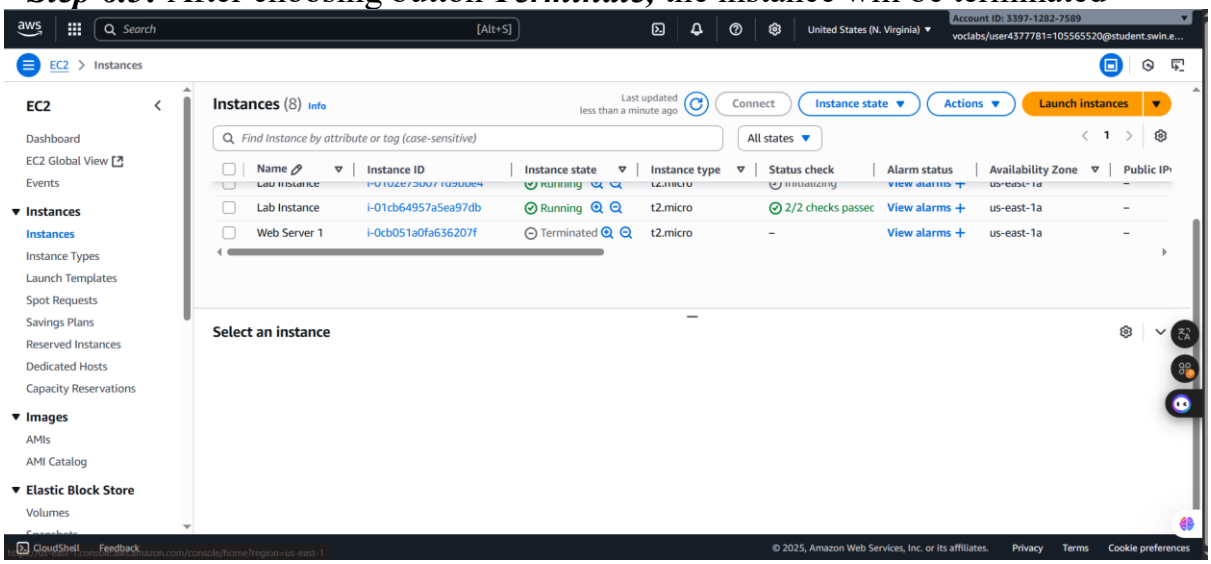
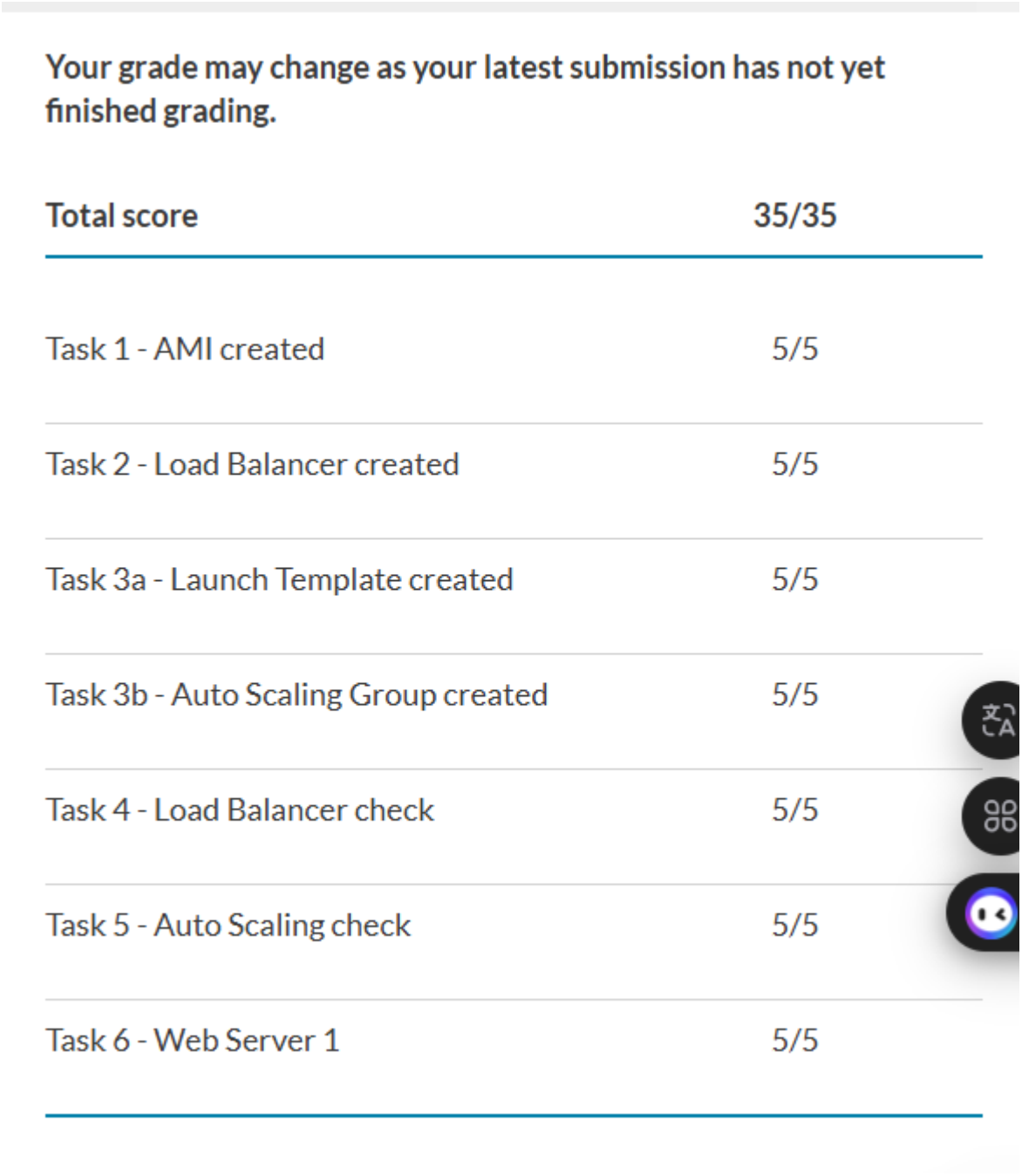


Figure 49: Stop instance successfully

I. Submitting your work



Your grade may change as your latest submission has not yet finished grading.

Total score	35/35
Task 1 - AMI created	5/5
Task 2 - Load Balancer created	5/5
Task 3a - Launch Template created	5/5
Task 3b - Auto Scaling Group created	5/5
Task 4 - Load Balancer check	5/5
Task 5 - Auto Scaling check	5/5
Task 6 - Web Server 1	5/5

Figure 50: Lab completed

J. Conclusion

- In this lab, I learned how to implement scaling and load balancing in AWS to improve both performance and reliability of applications. I created an Amazon Machine Image (AMI), set up an Elastic Load Balancer, and configured an Auto Scaling group to automatically adjust the number of instances based on demand. I also monitored the infrastructure using Amazon CloudWatch and tested how the system responded under increased load.

- Through this process, I gained hands-on experience with two important AWS services: Elastic Load Balancing and Auto Scaling. I understood how they work together to distribute traffic evenly, maintain application availability, and reduce costs by scaling resources dynamically. Overall, this lab gave me practical skills in designing a scalable and fault-tolerant cloud architecture.