Swinburne University of Technology

COS20019 Cloud Computing Architecture

Assignment 2

Saturday 14th November 2023

Phan Vu
Student ID: 104222099
104222099@student.swin.edu.au

I. INTRODUCTION

The Photo Album website project employs EC2 web servers as the hosting infrastructure for the website, using many AWS services including S3, RDS, and Lambda. Users have the ability to upload visual images onto the digital platform, where they may then engage in the exploration of various collections including these photographs. Furthermore, the website provides the functionality of automatically producing and resizing thumbnail photos. The comprehensive source code and accompanying instructions have been supplied to assist the seamless integration of the website with the S3 bucket, RDS database, and Lambda function. This project utilizes the capabilities of Amazon Web Services (AWS) to create a dynamic and user-centric platform for image albums. It incorporates many functionalities, including photo retrieval, uploading, and thumbnail generation.

II. WEBSITE ARCHITECTURE

A. Infrastructure Requirements

1) **Virtual Private Cloud (VPC):** VPC configured with 2AZs both with public and private subnets. Public and private route tables route to IGW and NAT, respectively.

Figure 1. Configuring VPC

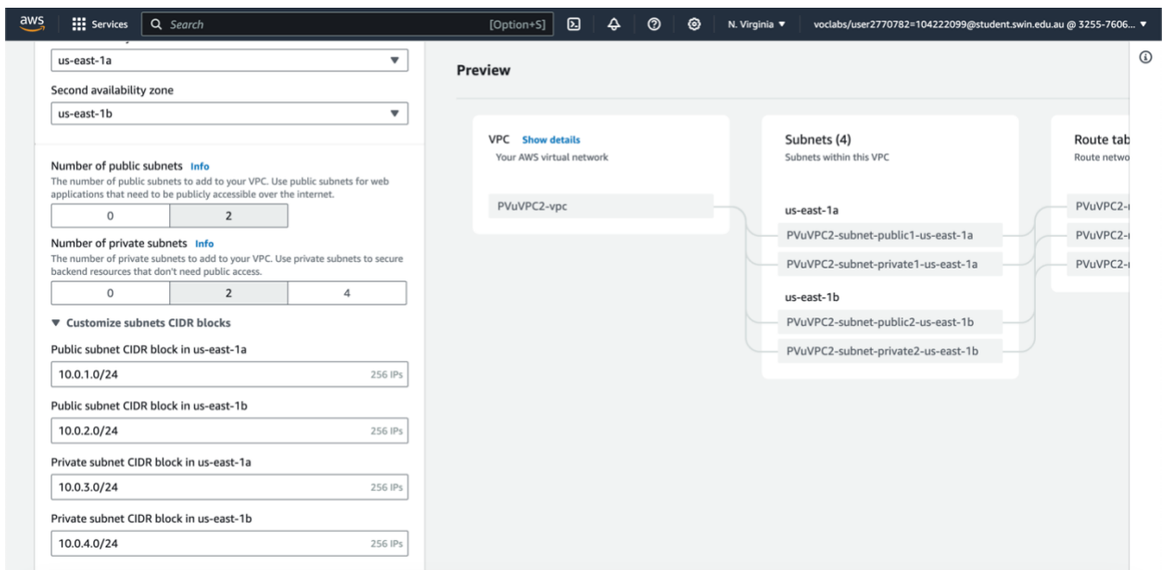

Figure 2. VPC Subnet CIDR Block

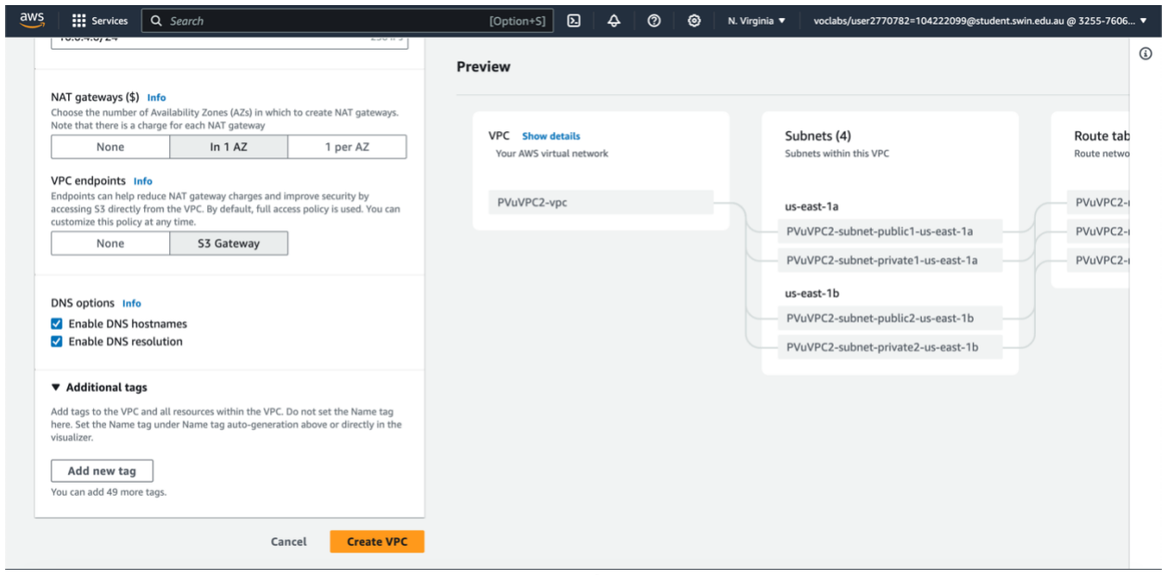

Figure 3. Configuring VPC

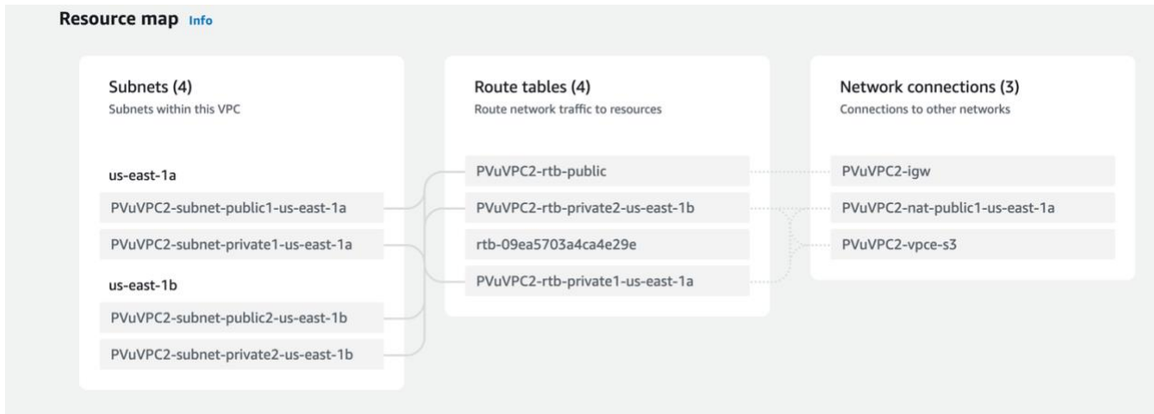

Figure 4. Resource Map

The VPC is ready for the photo album web server to be launched.

2) **Security Group:** The architecture has five security groups: ELBSG, WebServerSG, DBServerSG, DevServerSG, and NATServerSG. Since NAT gateway is utilized instead of NAT instance, NATServerSG is unnecessary. All security group outbound rule defaults to IPv4 traffic everywhere.

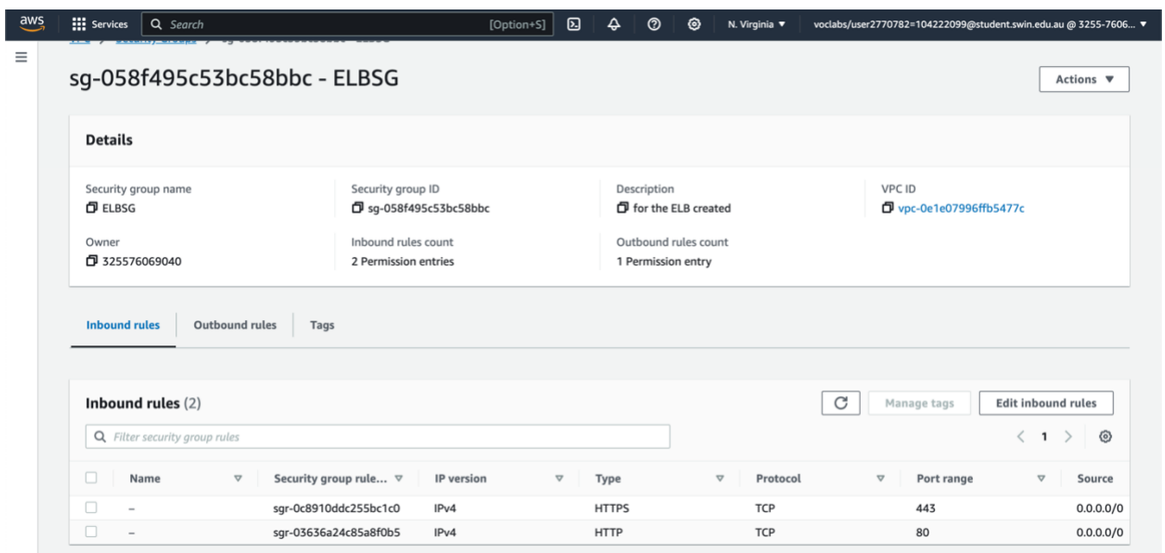

Figure 5. ELBSG for the ELB created above.

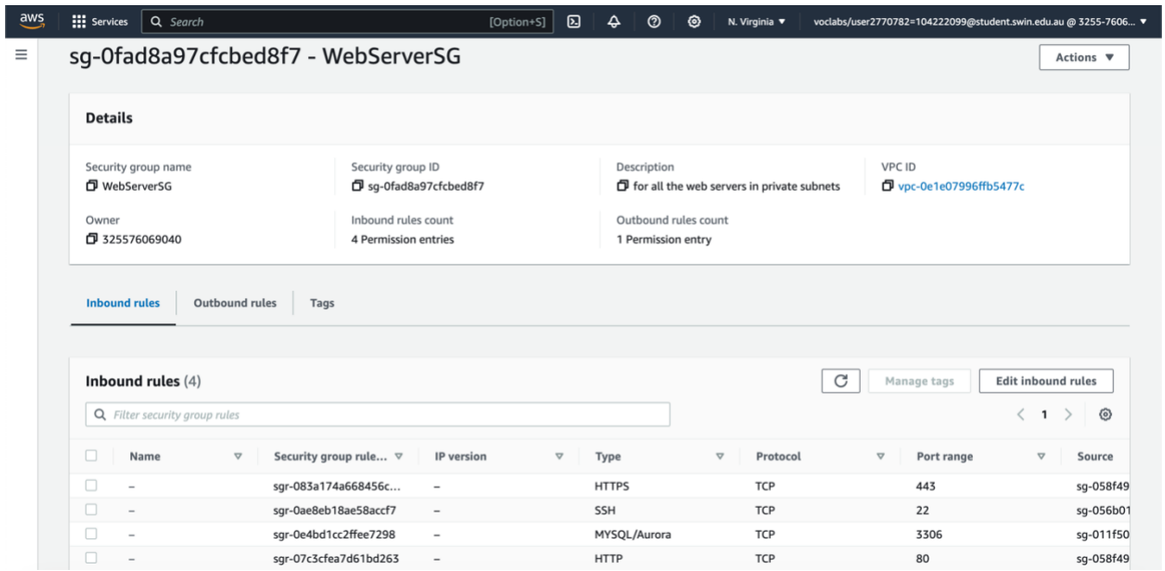

Figure 6. WebServerSG for all the web servers in private subnets.

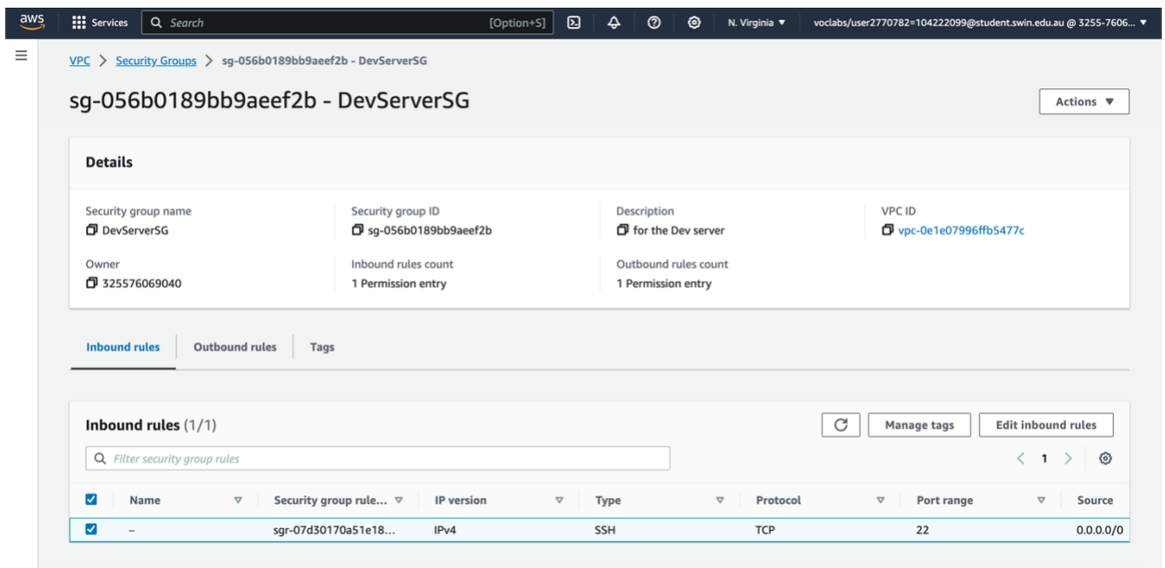

Figure 7. DevServerSG for the Dev server.

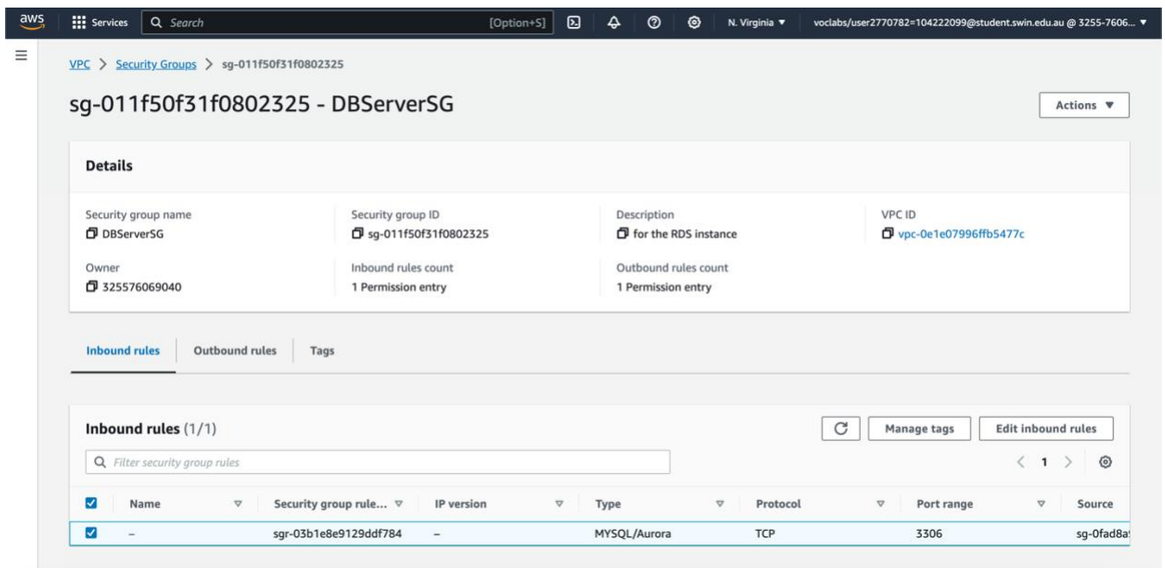

Figure 8. DBServerSG for the RDS instance.

Security group name	Protocols	Source
WebServerSG	HTTP (80), HTTPS (443)	ELBSG
	SSH (22)	DevServerSG
	MYSQL/Aurora (3306)	DBServerSG
DBServerSG	MYSQL/Aurora (3306)	WebServerSG
DevServerSG	SSH (22)	Anywhere-IPv4
ELBSG	HTTP (80), HTTPS (443)	Anywhere-IPv4

Figure 9. Inbound security rules summary

3) **Network ACLs (NACL):** The "PrivateSubnetsNACL" NACL was established to enhance web server security on private subnets. The NACL limits ICMP communication to and from the DevServer in both directions.

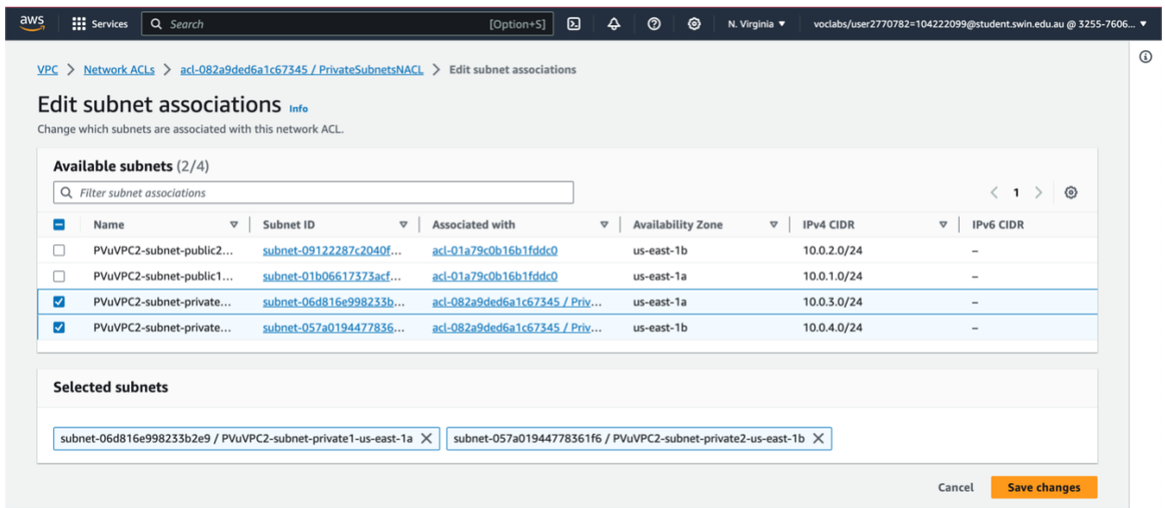

Figure 10. NACL associated with private subnets in the VPC

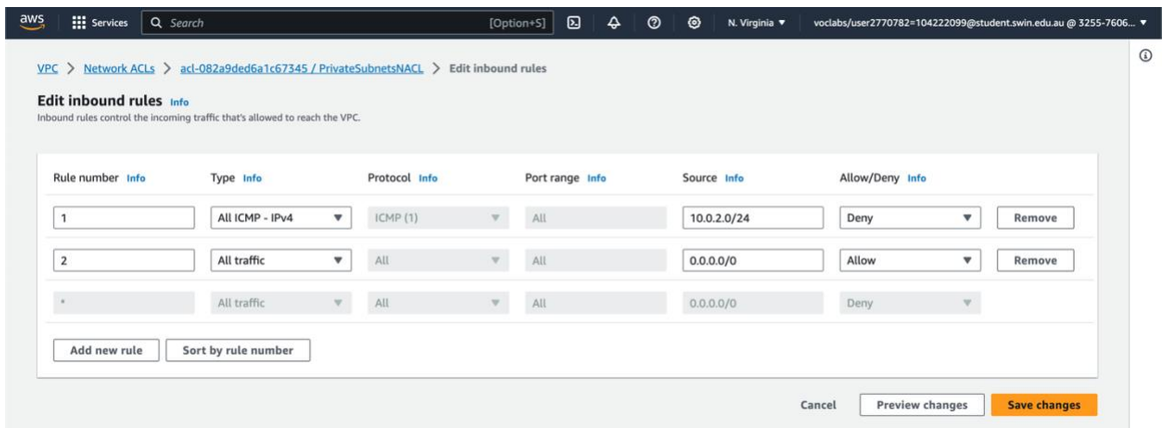

Figure 11. Inbound rules of PrivateSubnetsNACL

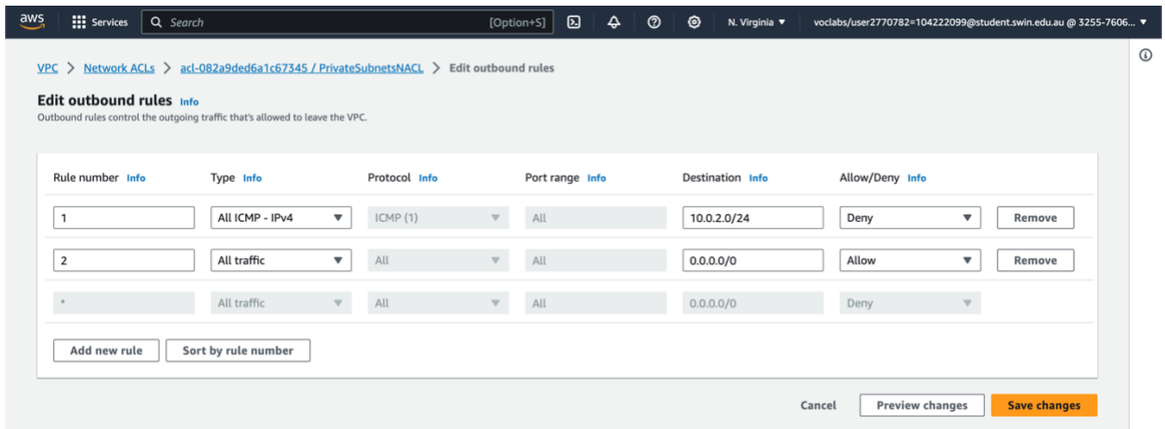

Figure 12. Outbound rules of PrivateSubnetsNACL

```
[ec2-user@ip-10-0-2-99 ~]$ ping 10.0.3.78
PING 10.0.3.78 (10.0.3.78) 56(84) bytes of data.
^C
--- 10.0.3.78 ping statistics ---
162 packets transmitted, 0 received, 100% packet loss, time 164868ms

[ec2-user@ip-10-0-2-99 ~]$ ping 10.0.4.26
PING 10.0.4.26 (10.0.4.26) 56(84) bytes of data.
^C
--- 10.0.4.26 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5126ms
```

Figure 13. Testing the NACL (ping from DevServer to WebInstance ASG)

```
[ec2-user@ip-10-0-4-26 ~]$ ping 10.0.2.99
PING 10.0.2.99 (10.0.2.99) 56(84) bytes of data.
^C
--- 10.0.2.99 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2051ms
```

Figure 14. Testing NACL (ping from WebInstance ASG to DevServer)

```
[ec2-user@ip-10-0-3-78 ~]$ ping 10.0.2.99
PING 10.0.2.99 (10.0.2.99) 56(84) bytes of data.
^C
--- 10.0.2.99 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4075ms
```

Figure 15. Testing NACL (ping from WebInstance ASG to DevServer)

4) **IAM Role:** The management console contains IAM roles such as “LabRole” and “Labinstancerole” with essential permissions for this assignment.

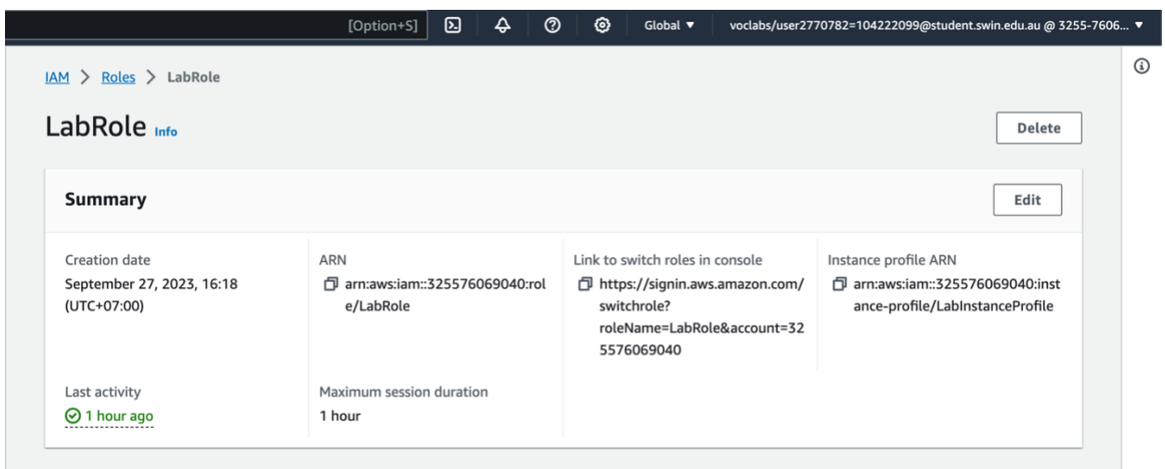

Figure 16. IAM role

5) **CreateThumbnail Lambda function:** The IAM execution role was assigned to the Lambda function to ensure appropriate security measures and permissions. The IAM role named LabRole, which adheres to the concept of least privilege, grants the Lambda function the necessary access and control over the items within the designated S3 bucket.

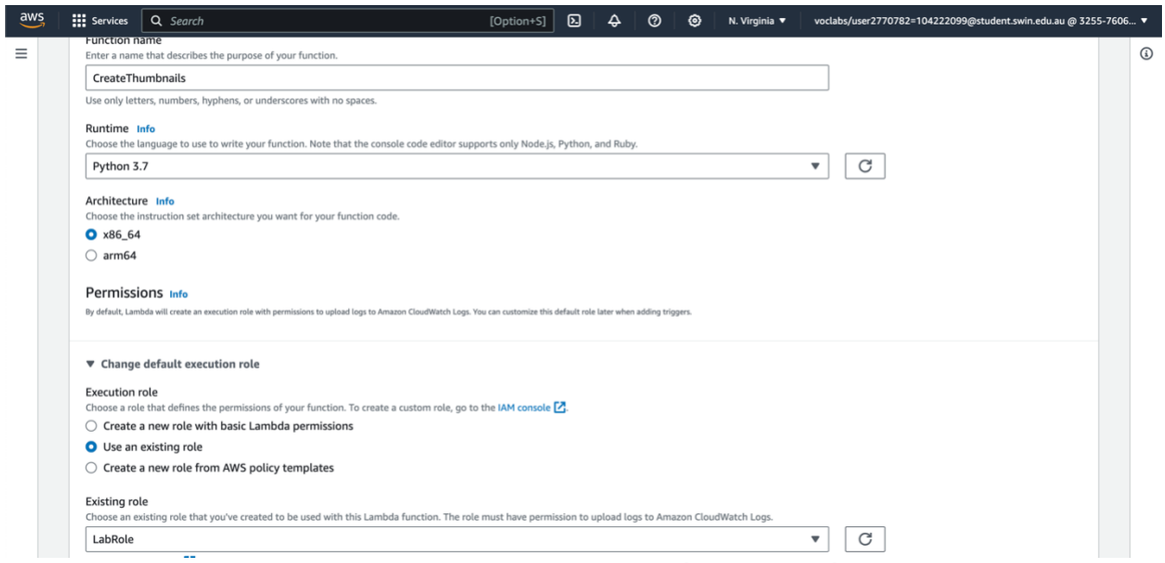

Figure 17. CreateThumbnail Lambda function configuration

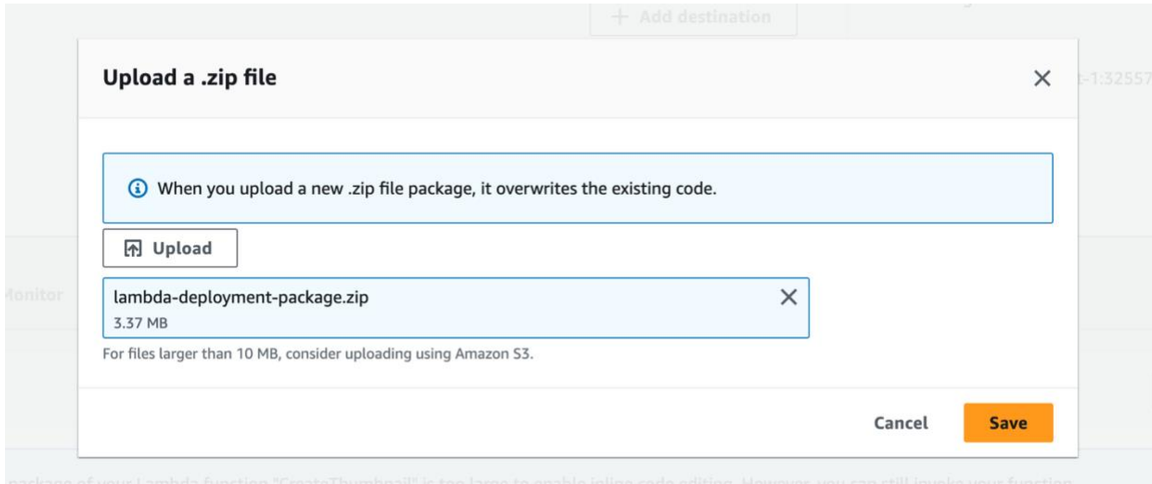

Figure 18. CreateThumbnail Lambda function configuration

To grant the web server the necessary permissions, an IAM role entitled LabInstanceProfile was created using the principle of least privilege. The role was created to grant the selected S3 bucket Web Server rights for adding objects and calling the CreateThumbnail Lambda function.

6) **Auto Scaling Group (ASG):** To create ASG, we need to create a DevServer instance first, which is used to develop the custom AMI for the web server and make a launch template later

DevServer Instance: Since it only develops the specific AMI needed to run PhotoAlbum, DevServer does not get traffic from ELB. The custom AMI includes the AWS PHP SDK, Apache web server, and website source code. The DevServer also manages MySQL RDS using phpMyAdmin.

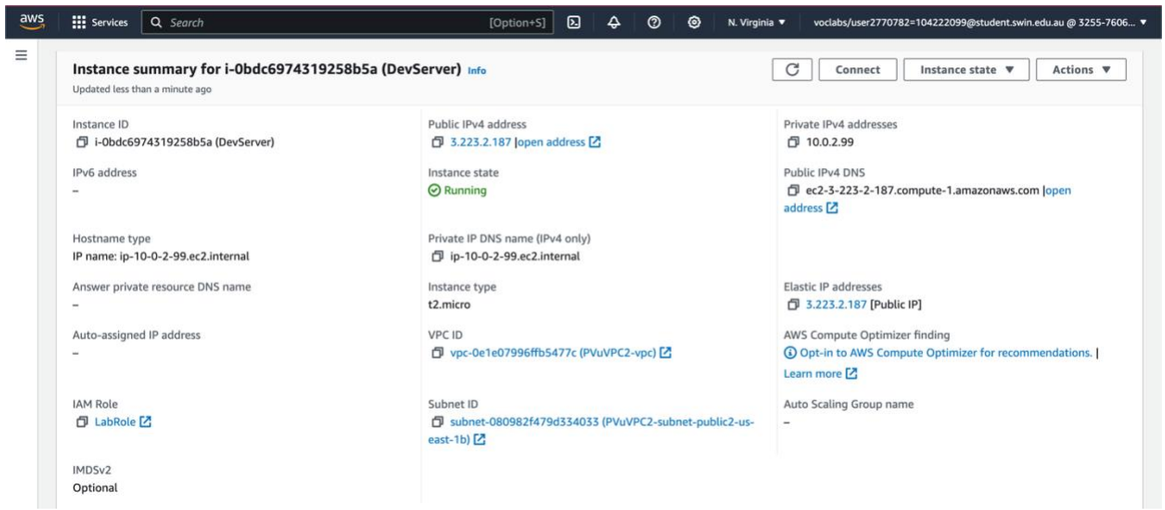

Figure 19. DevServer instance resides in Public Subnet 2 (CIDR: 10.0.2.0/24) with an Elastic IP associated

DevServer has t2-micro instance type, Amazon Linux 2 AMI (HVM), SSD Volume Type, and bash script-installed Apache Web Server in assignment 1a. AWS Learner Lab already has LabRole IAM Role.

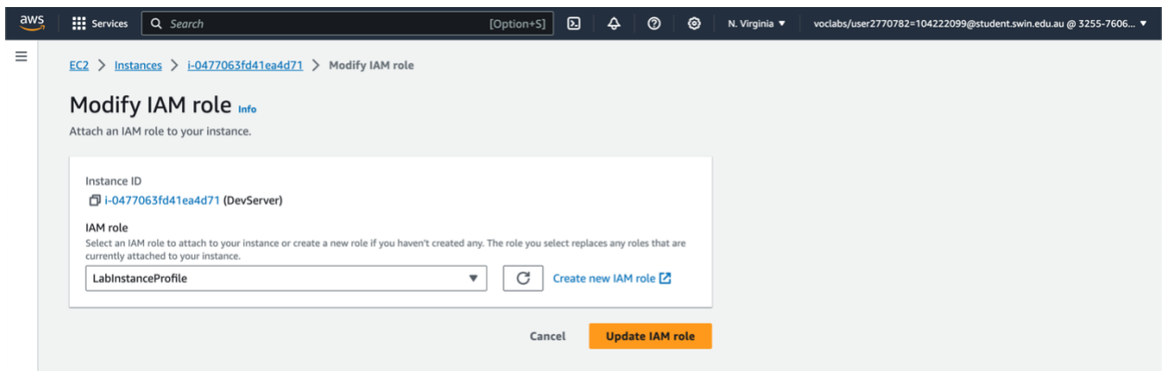

Figure 20. DevServer IAM Role

The DevServer is associated with an Elastic IP to allow SSH connection to manage the DevServer directory.

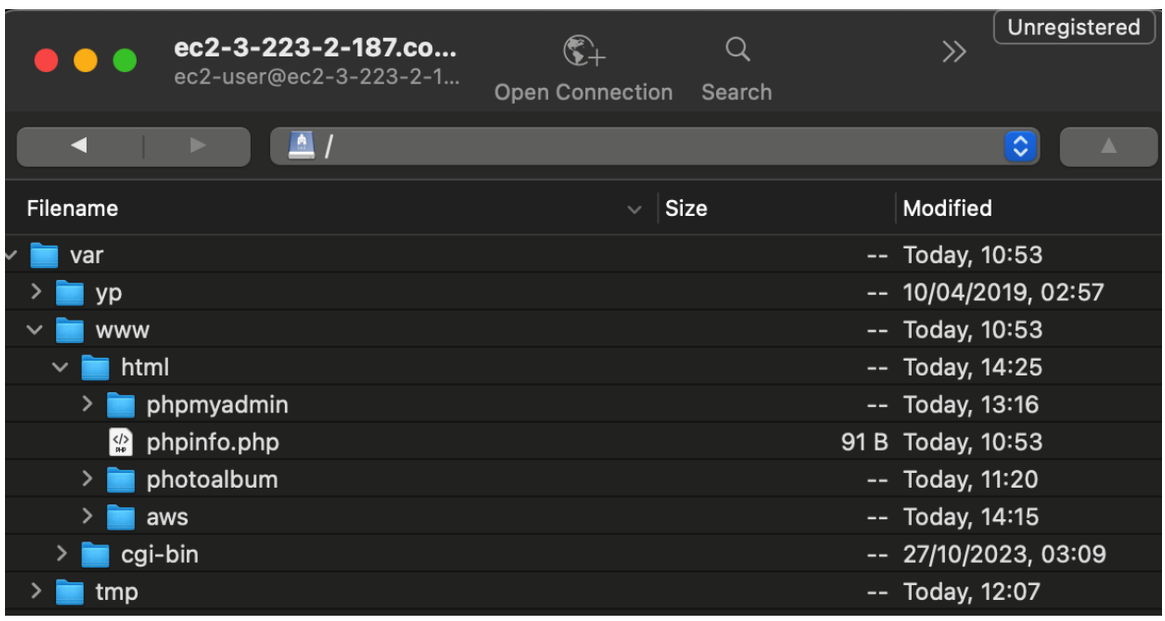

Figure 21. DevServer Directory Structure with necessary SDKs and components included

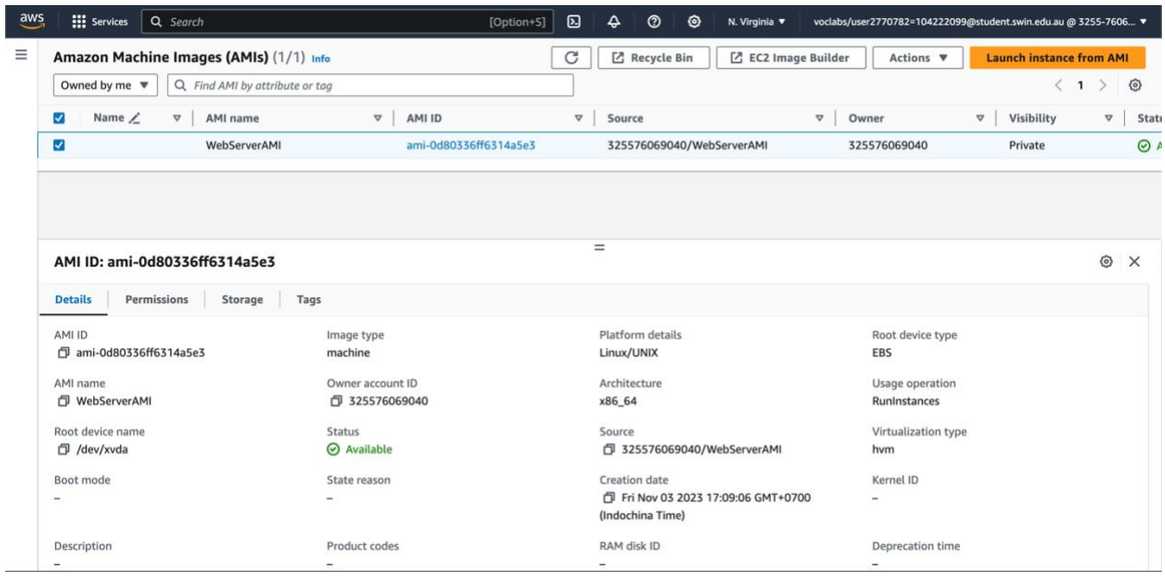

Figure 22. DevServer image created, ready to use

Launch template: Create a launch template with instance type t2.micro and IAM role LabInstanceProfile using the AMI created.

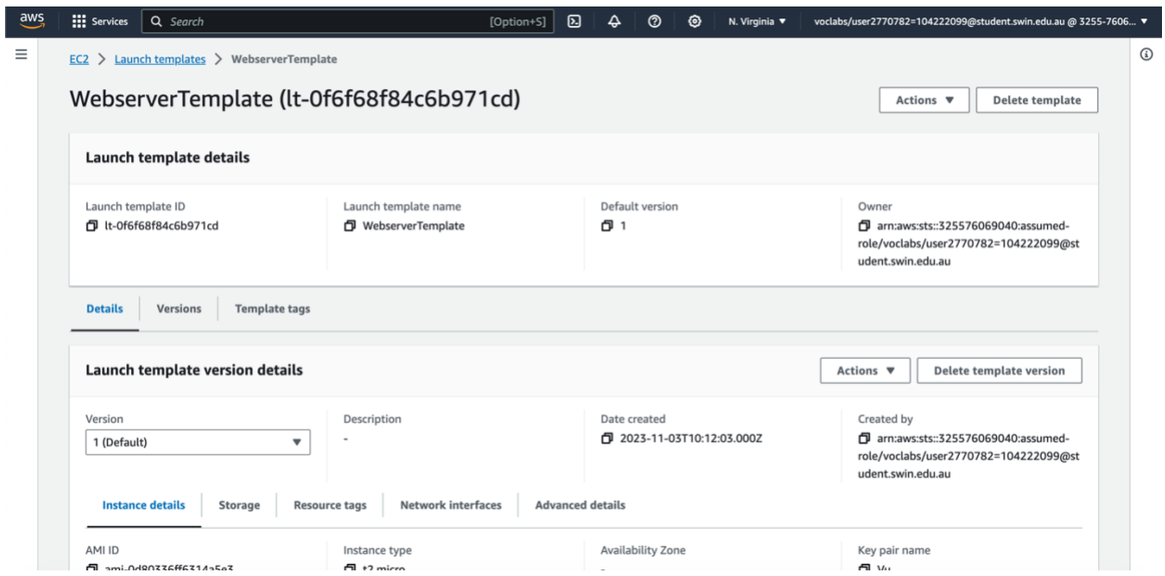

Figure 23. Launch template for ASG

Auto Scaling Group: The ASG is configured to launch instances exclusively into the private subnets, with a minimum of two and a maximum of three instances maintained, with two instances being the optimal number. This guarantees that the application maintains a minimum number of available instances while restricting the infrastructure's scalability to the specified maximum.

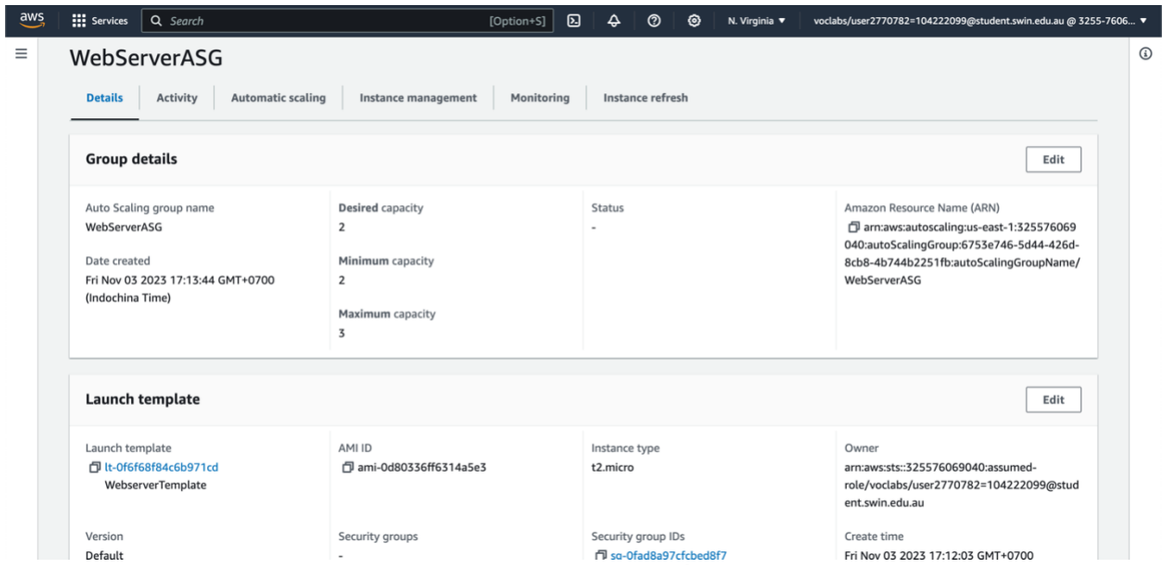

Figure 24. ASG basic configuration

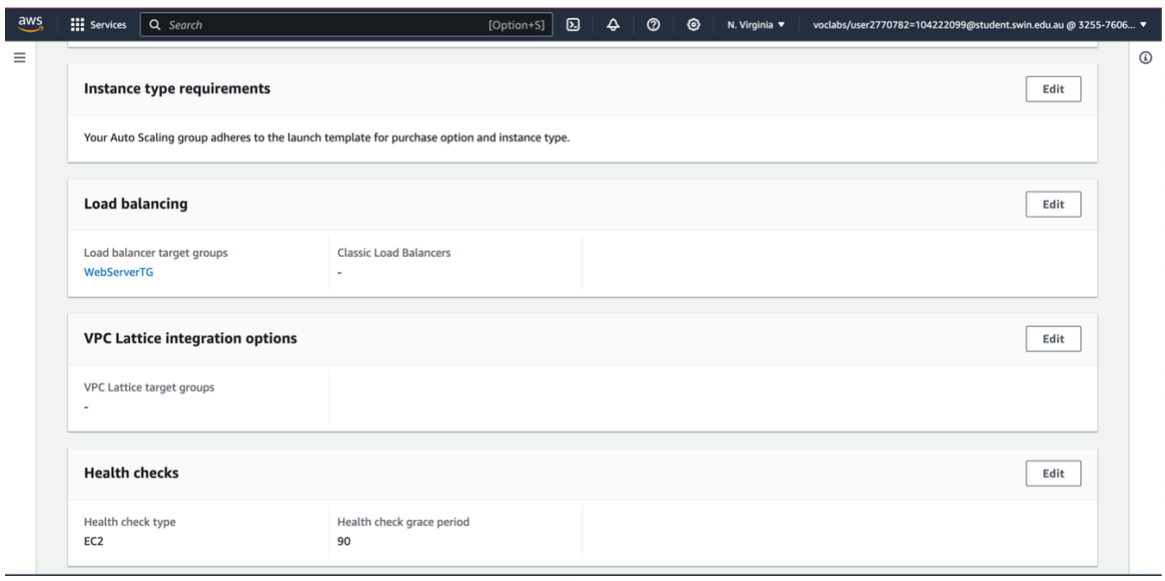

Figure 25. ASG health check configuration with target group attached

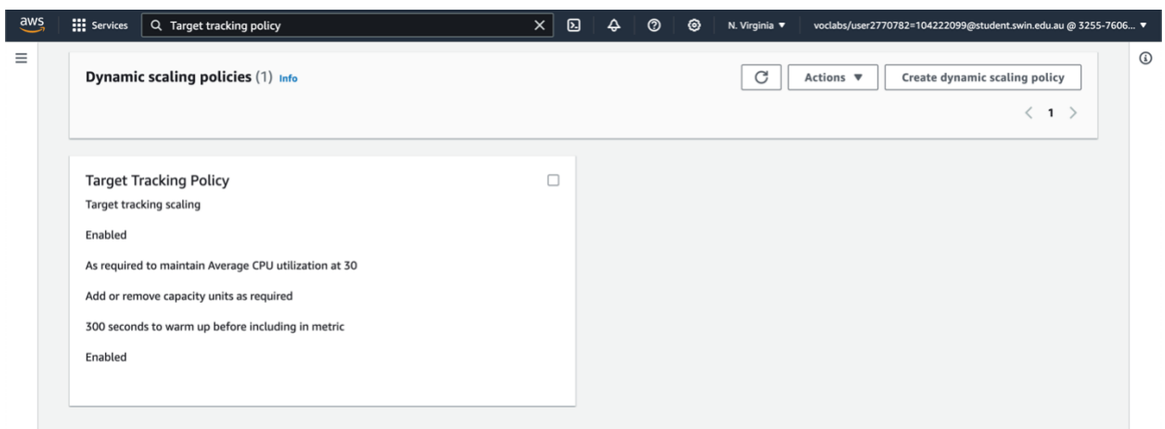

Figure 26. Target tracking policy based on application load balancer request

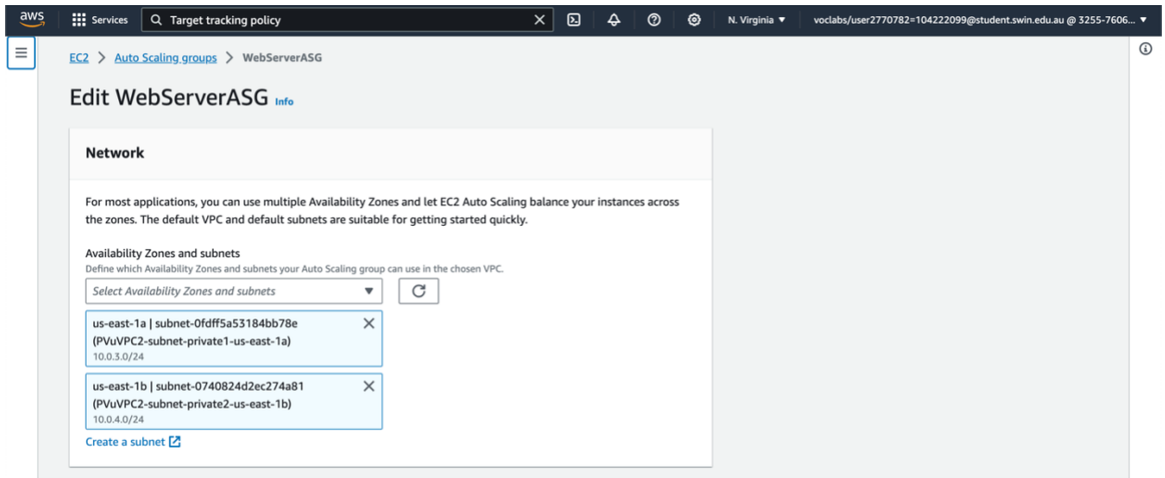

Figure 27. ASG network mapped to private subnets

Target tracking scaling was used to limit instances based on ELB target group requests. The policy sets a target request count of 30. The auto-scaling group automatically scales instances up or down to maintain the required request count per target, optimizing performance and resource use.

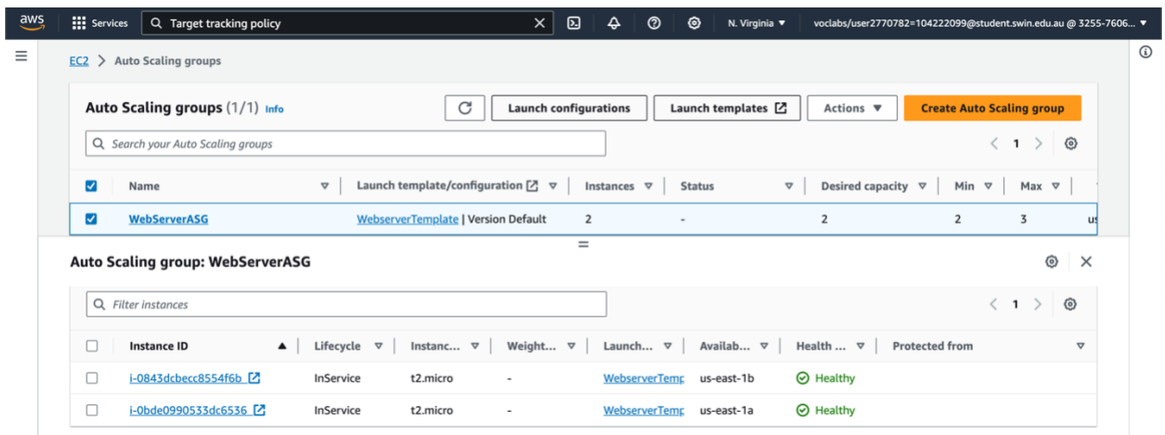

Figure 28. EC2 instances are properly distributed across two private subnets with healthy states

The web server may now adjust its capacity based on request load, maintaining a consistent number of requests per target and optimizing resource use.

7) **Elastic Load Balancing (ELB):** First, create a new target group as the load balancer needs to route requests to the targets in a target group and perform health checks on the targets.

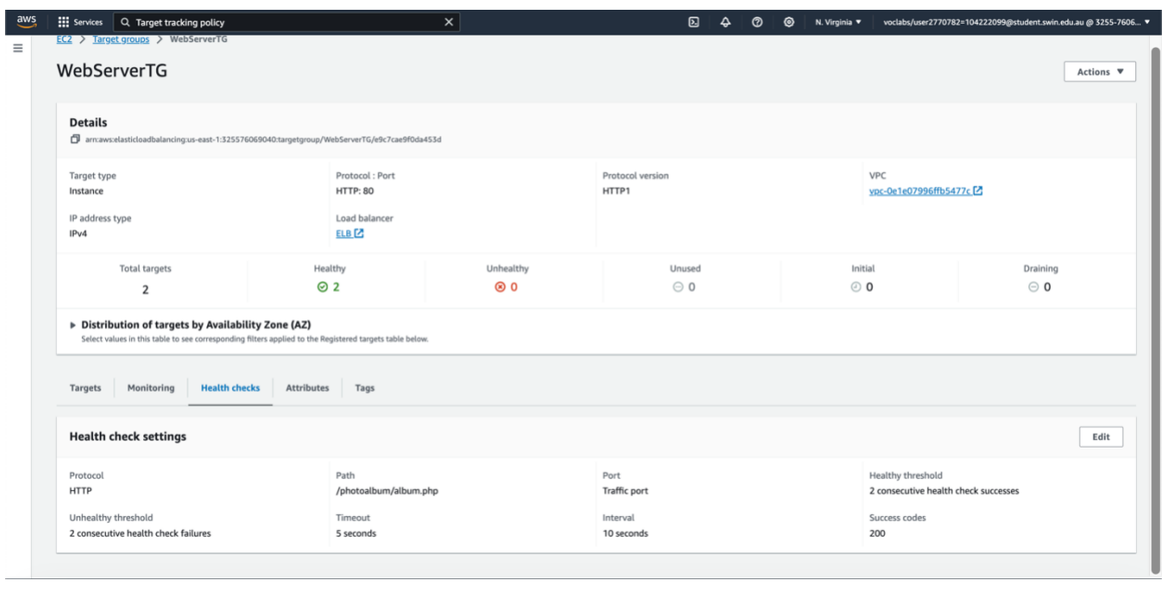

Figure 29. Target group configuration, with health check path set to /photoalbum/album.php

Create a new load balancer and attach it to the target group.

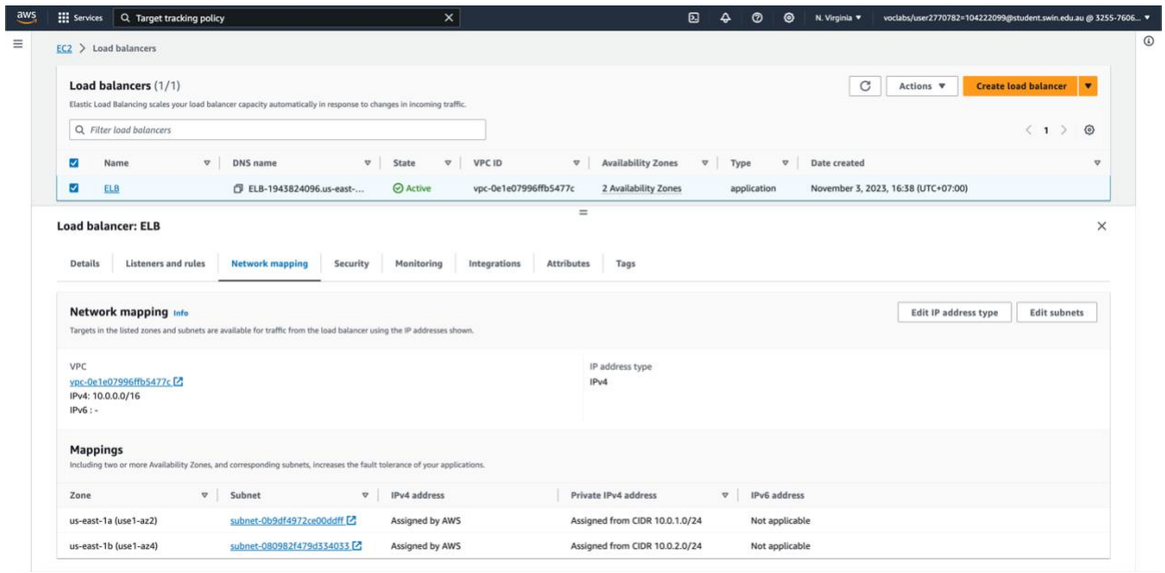

Figure 30. Application Load Balancer mapped to Public Subnet 1 and Public Subnet 2

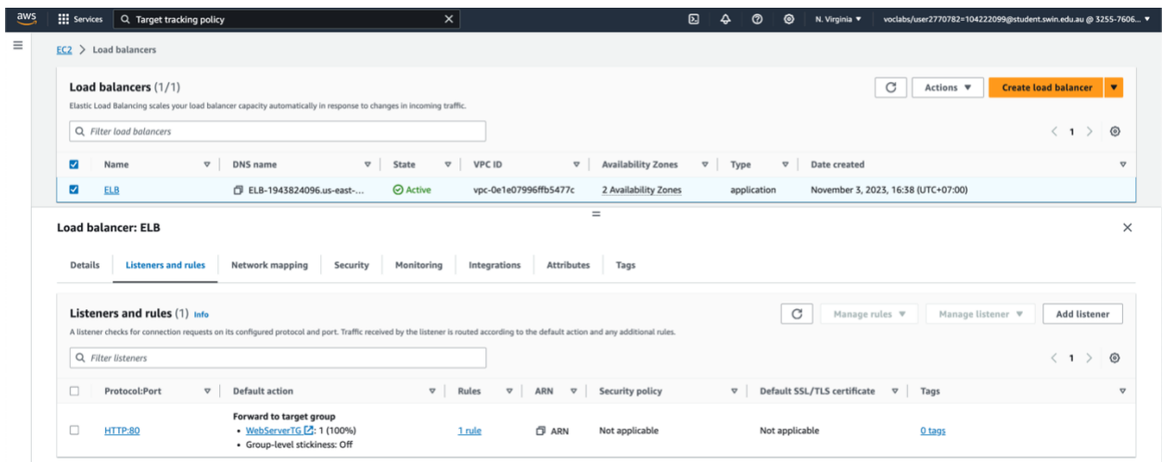

Figure 31. ELB Listener check rule which forwarded to the target group created in Figure 26

Now, The ELB can distribute incoming HTTP and HTTPS traffic across multiple EC2 targets.

8) **Simple Storage Service (S3):** Assignment 1b's S3 bucket for photos is similar. This S3 bucket has permissions and policies to make objects accessible. These steps set photo access permissions correctly.

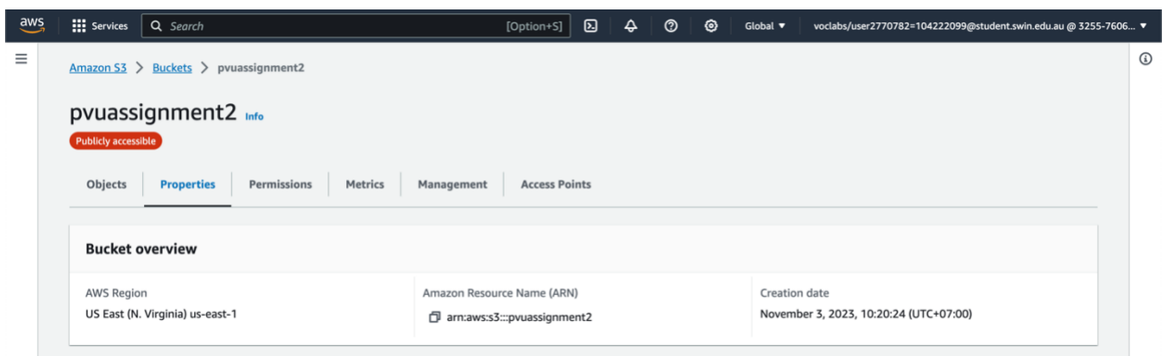

Figure 32. Properties of S3 bucket

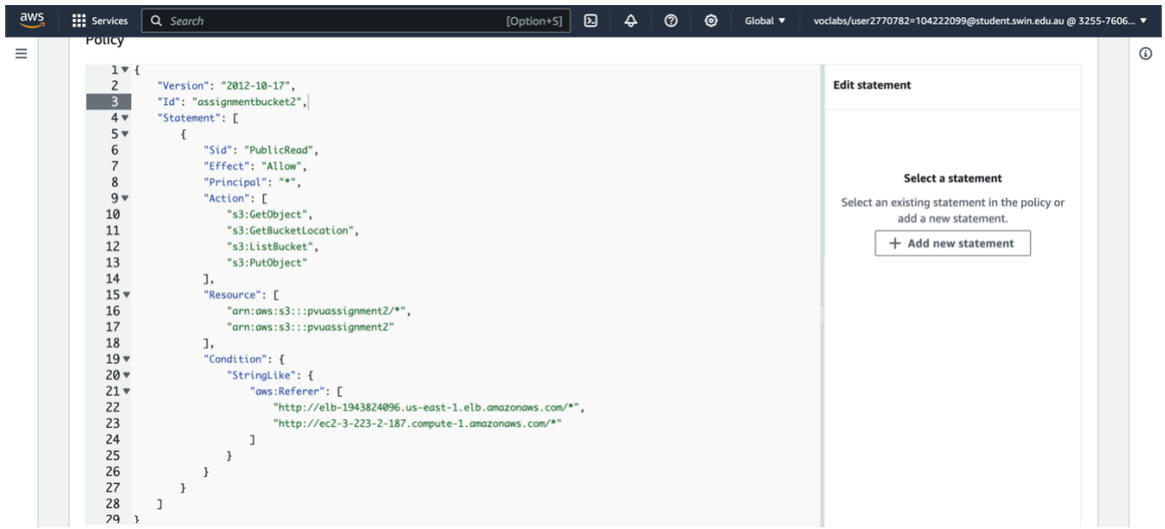

Figure 33. S3 policy to restrict access to a specific HTTP referer from Dev Server and Elastic Load Balancer

This policy limits access to the S3 bucket to GET requests from specific domains, ensuring restricted and secure access to the bucket's objects.

9) **Lambda Function:** A Lambda function named “CreateThumbnail” was created using Python 3.7 as the runtime environment.

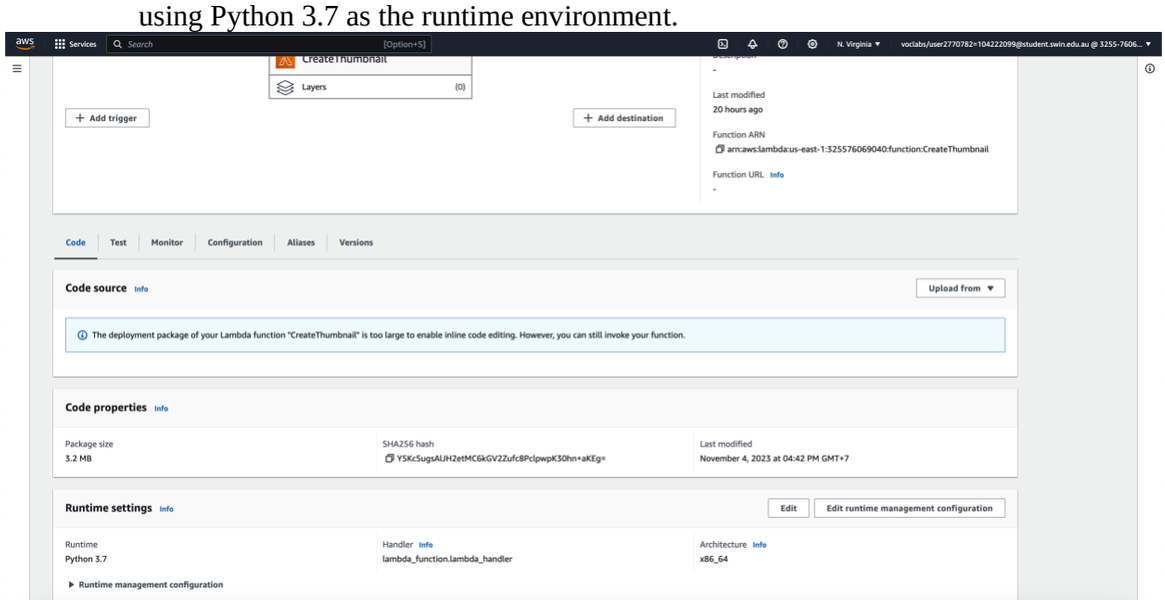

Figure 34. CreateThumbnail Lambda function configuration

Uploading “lambda-deployment-package.zip”. This package includes picture resizing and S3 bucket download/upload libraries and source code.

10) **Relational Database Service (RDS):** The RDS instance used in this assignment is configured like the previous one.

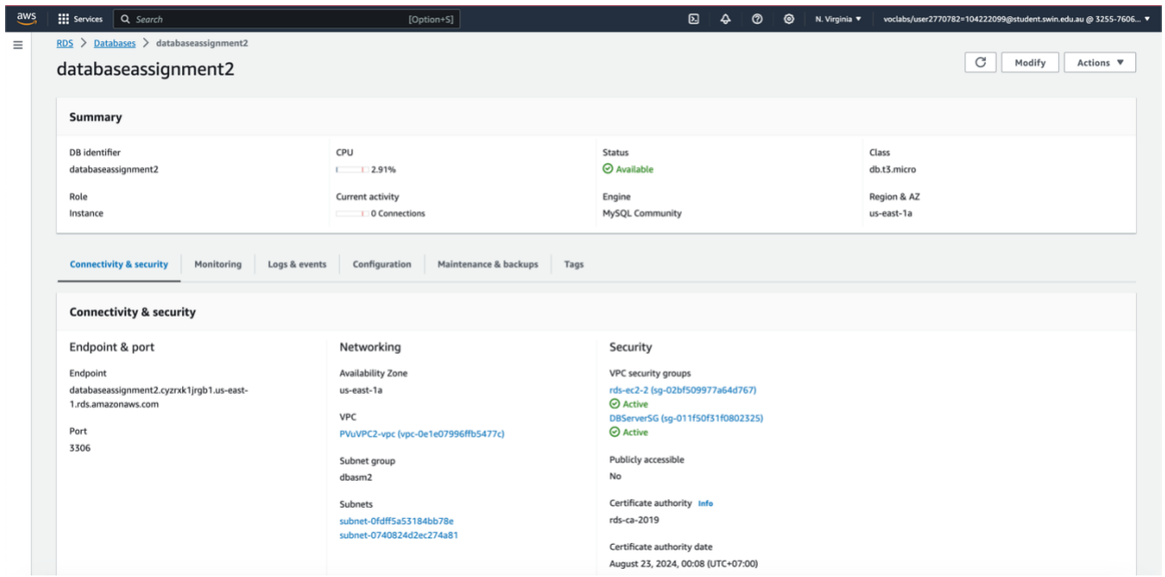

Figure 35. RDS instance

Configuration:

- Template: Free-tier
- Database engine: MySQL Community 8.0.28
- Public access set to No.
- Use DBServerSG for VPC security group
- AZ set to us-east-1a (According to the provided diagram).
- The RDS instance is associated with a subnet group “dbasm2” comprising private subnets in both AZs.

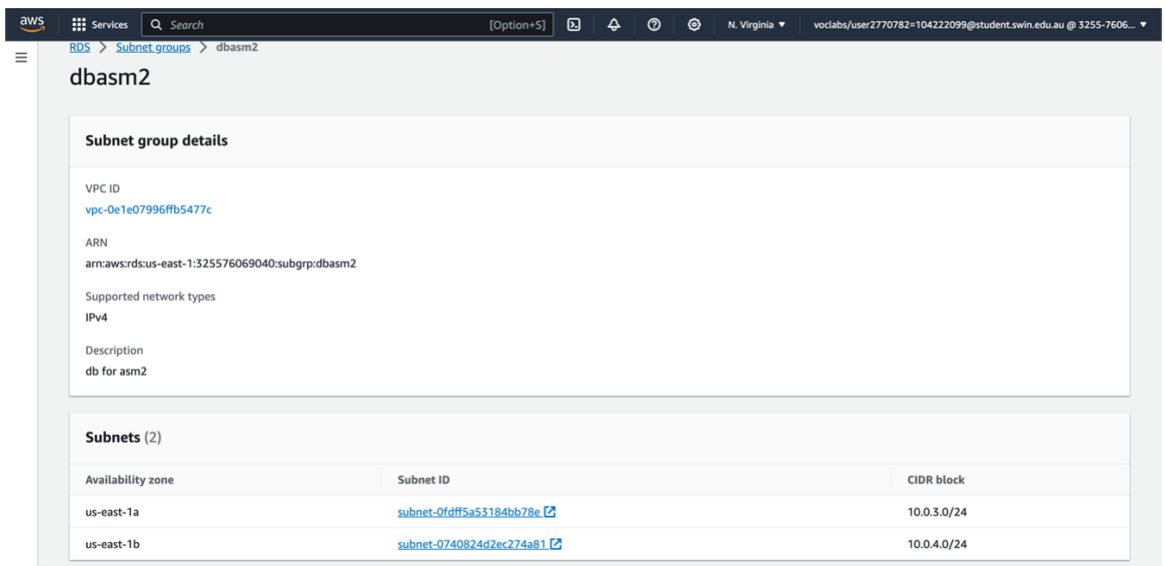

Figure 36. Subnet dbasm2 with Private subnet 3 and Private subnet 4

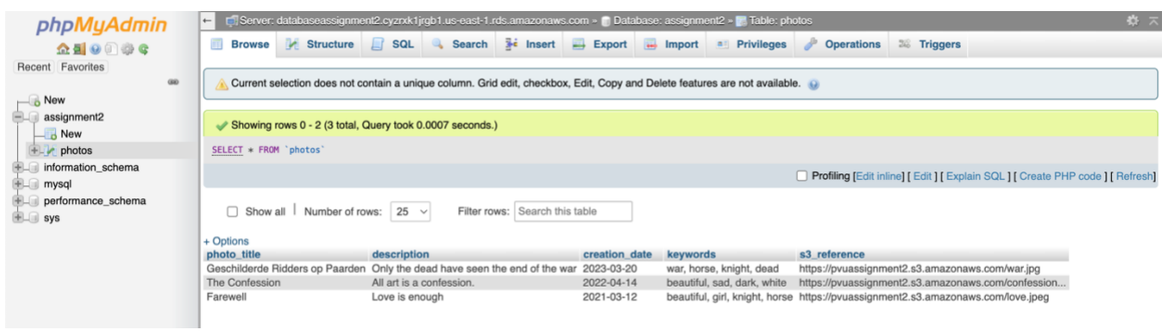

Figure 37. Data records of the database

B. Functional requirements

1) **Website accessibility:** To access the PhotoAlbum website, use the URL: http://elb-1943824096.us-east-1.elb.amazonaws.com/photoalbum/album.php. It allows you to view and interact with the PhotoAlbum web application. Additionally, to upload photos and their associated metadata, utilize the PhotoUploader web page at http://elb-1943824096.us-east-1.elb.amazonaws.com/photoalbum/photouploader.php. Using this page, multiple photos and their corresponding metadata can be easily uploaded to enhance the functionality of the PhotoAlbum website.

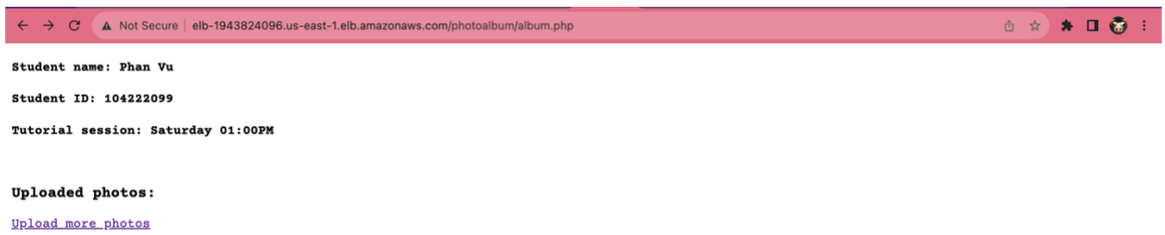

Figure 38. Website accessible through ELB DNS

2) **Photo display function:**

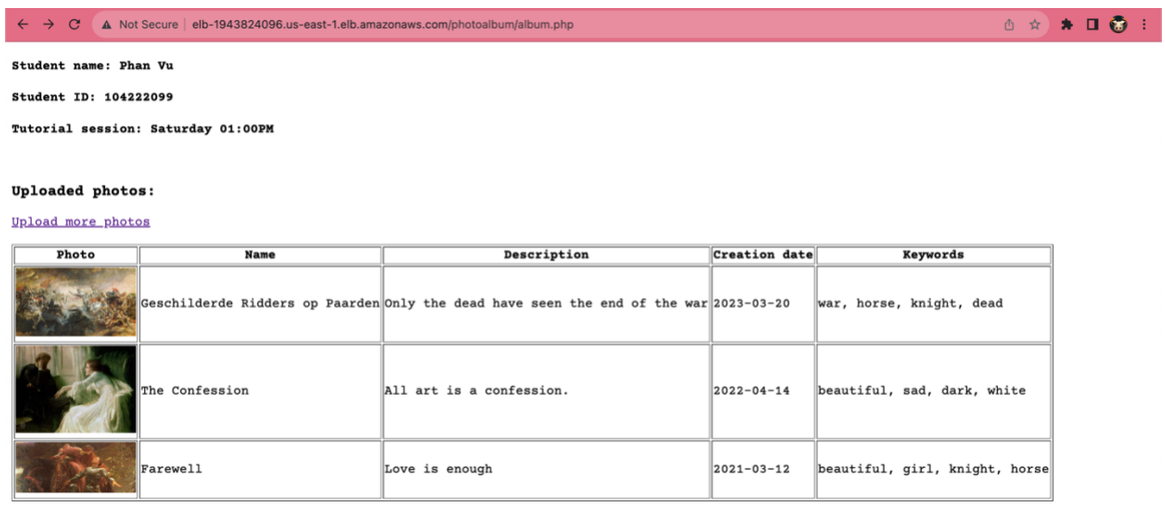

Figure 39. Photo display function

The photo display function is working correctly.

3) **Photo uploader function:**

Figure 40. Photo uploader function

Student name: Phan Vu

Student ID: 104222099

Tutorial session: Saturday 01:00PM

Uploaded photos:

Upload more photos

Photo	Name	Description	Creation date	Keywords
	Geschilderde Ridders op Paarden	Only the dead have seen the end of the war	2023-03-20	war, horse, knight, dead
	The Confession	All art is a confession.	2022-04-14	beautiful, sad, dark, white
	Farewell	Love is enough	2021-03-12	beautiful, girl, knight, horse
	Mary - Mother of Jesus	Mary was a first-century Judean woman of Nazareth,[6] the wife of Joseph and the mother of Jesus.	0001-12-24	Catholic, Church, Maria

Figure 41. Photo uploaded

The photo uploader function is working correctly.

4) **Resizing Lambda:**

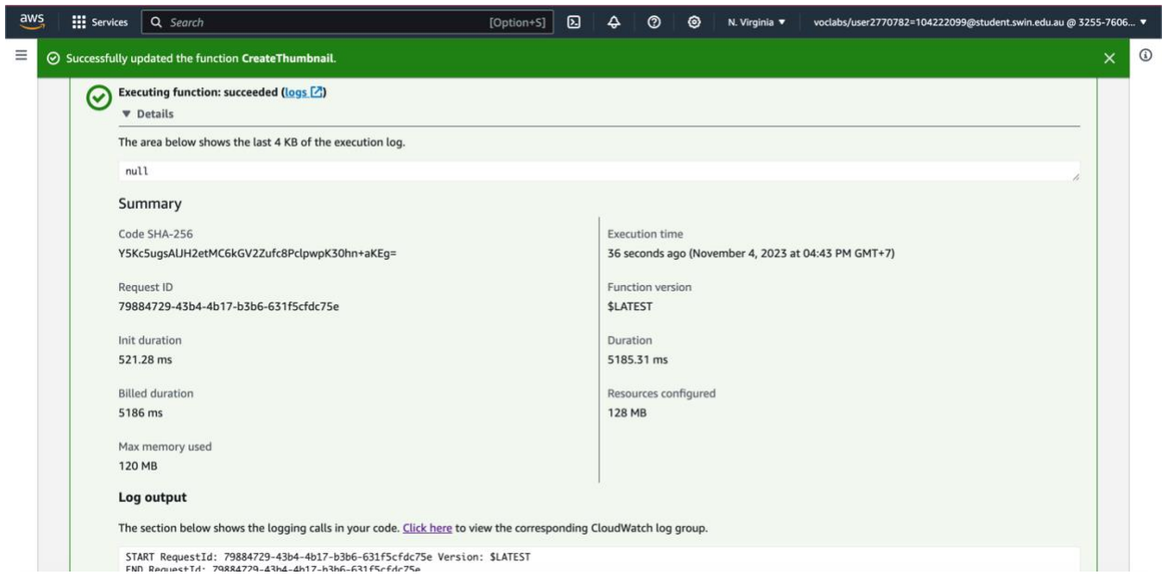

Figure 42. Successfully resize image

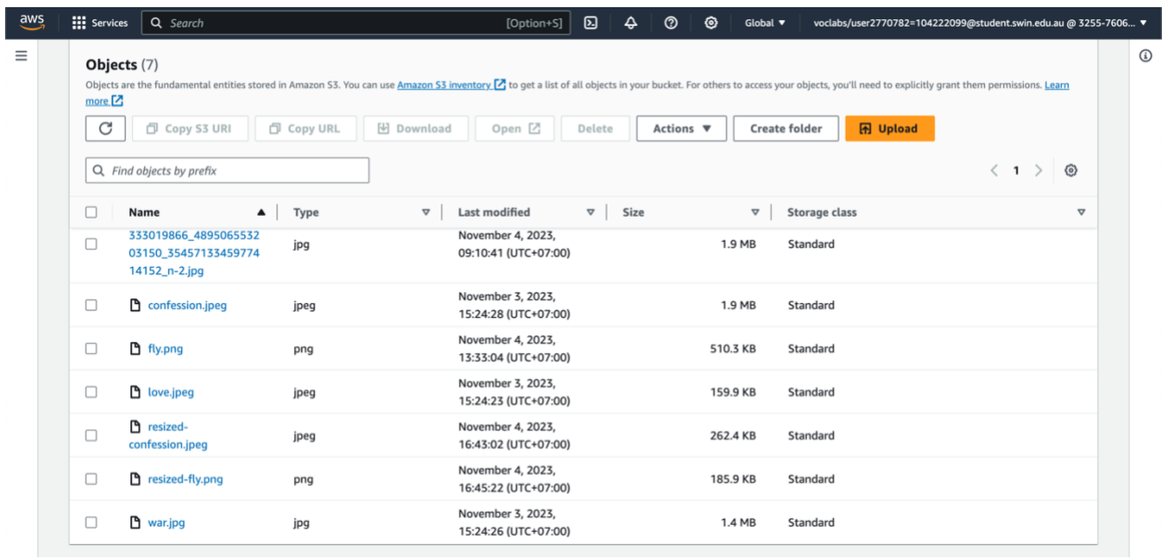

Figure 43. Resized images uploaded