# Deploying A Highly Available Photo Album Website

Minh Phuong (Jade) Hoang
October 8, 2023
jadehoang167@gmail.com

## I. Introduction

This report outlines the configuration and documentation for Assignment 2 (Cloud Computing Architecture unit), which aims to create a highly available Photo Album website. The step involves the deployment of an Elastic Load Balancer (ELB), an Auto Scaling group (ASG) that operates across multiple Availability Zones (AZs), a Lambda function, S3 storage, and an RDS instance. Security Groups and an NACL were also implemented to enhance security.

The Photo Album website can be accessed via the following ELB's URL:

http://assign2elb-1292790470.us-east-1.elb.amazonaws.com/photoalbum/album.php

## II. VPC Configuration

The figure shows the "JHoangVPC" setup with 4 subnets across 2 AZs and their corresponding route table associations. The "PublicRT" route table is linked to the IGW for internet access, while the "PrivateRT" is connected to the NAT gateway.

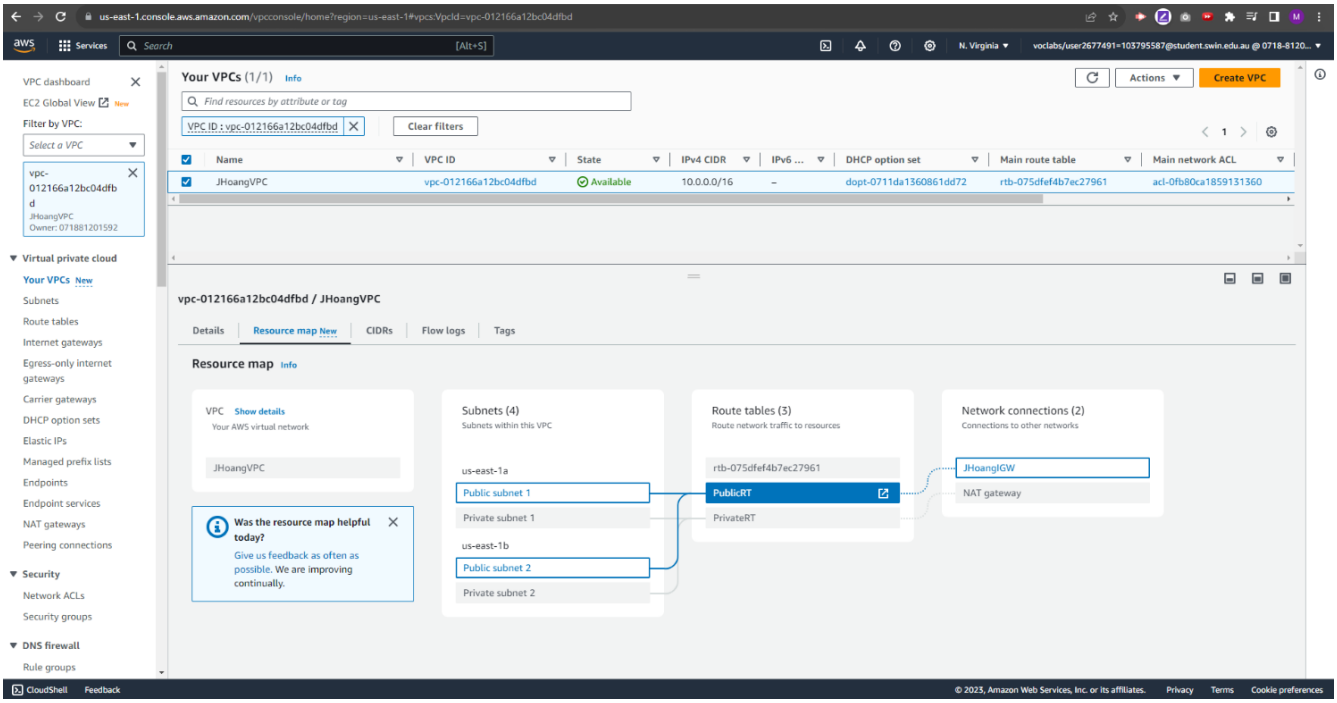


*Figure 1 - JHoangVPC resource map.*

## III. Route table configurations

The "PublicRT" includes a route for 0.0.0.0/0 directing traffic to the IGW and is associated with Public Subnet 1 and Public Subnet 2. Additionally, “PrivateRT" has a route to the NAT gateway, enabling Web Servers in Private Subnet 1 and Private Subnet 2 to initiate Internet access.

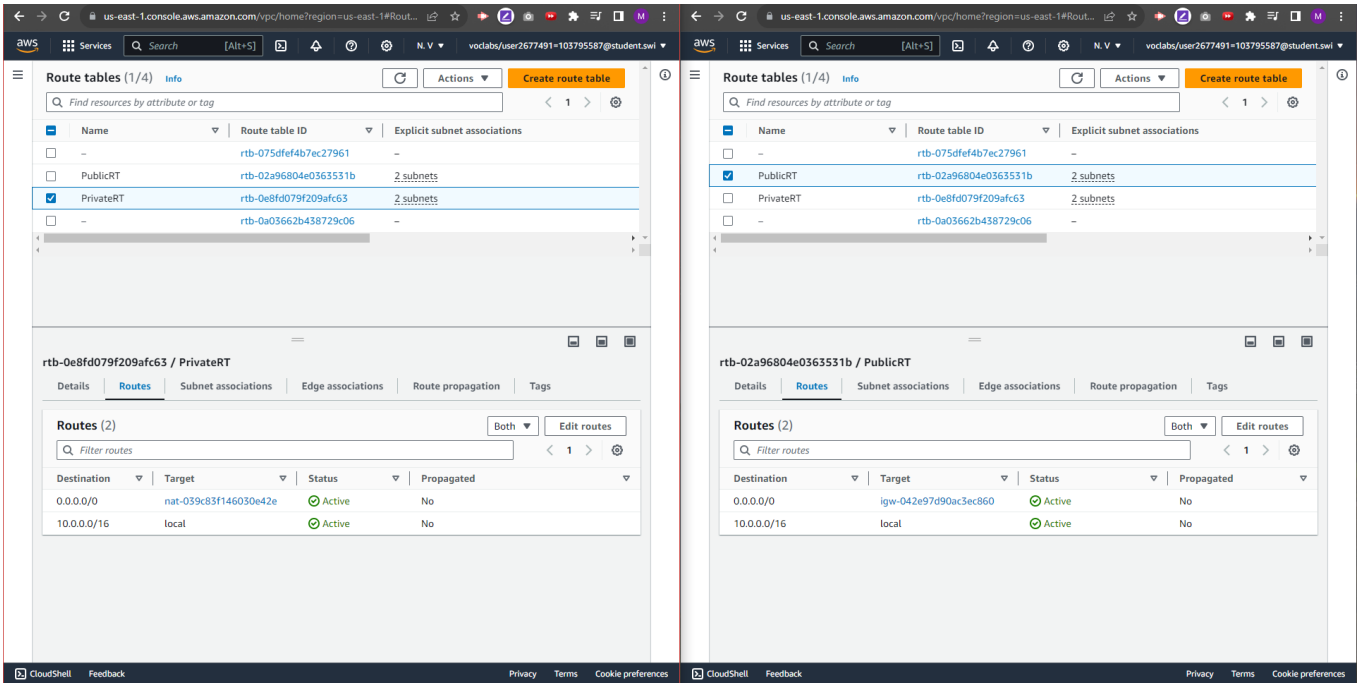


*Figure 2 – PublicRT and PrivateRT route details.*

## IV. Security Group Configurations

All security groups created share the same outbound rule: allow all traffic to anywhere.

### A. *Create ELBSG*

The ELBSG has one inbound rule to allow HTTP traffic from anywhere (0.0.0.0/0) and is associated with the “assign2elb” application load balancer.

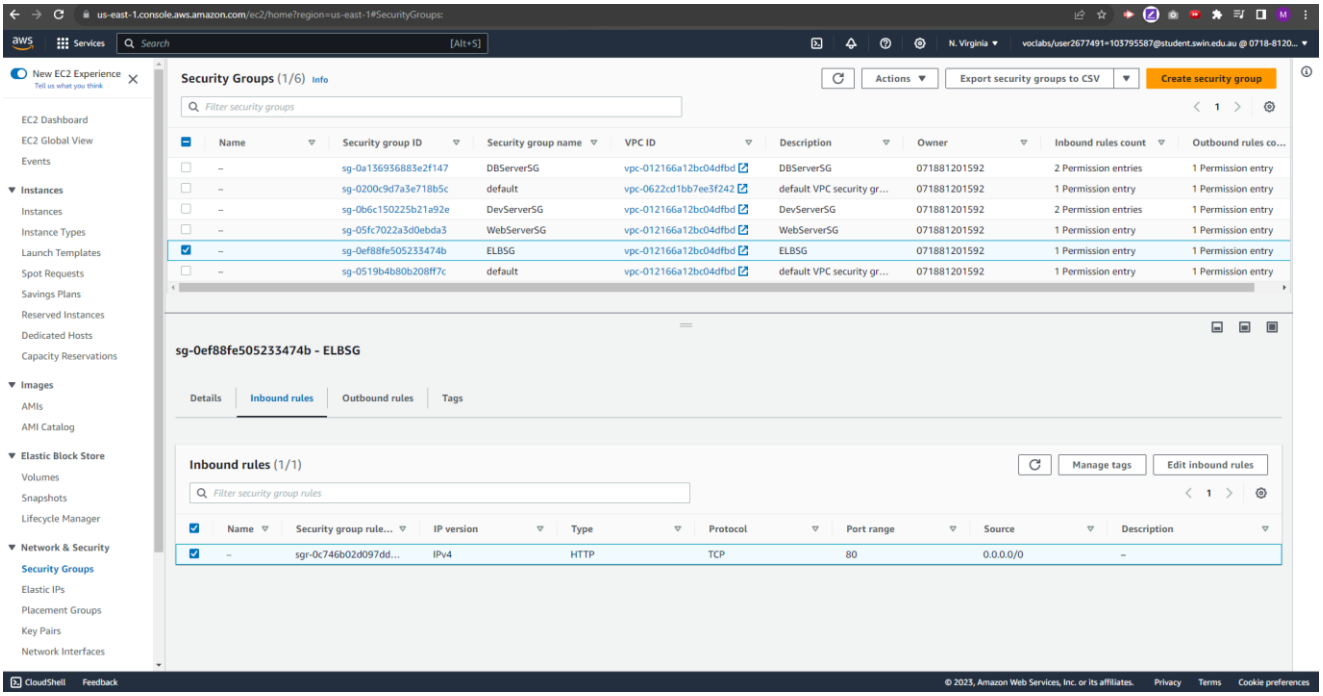


*Figure 3 - ELBSG inbound rules details.*

### B. *Create WebServerSG*

The WebServerSG for the web servers in private subnets has an inbound rule that allows HTTP (80) from only the ELBSG.

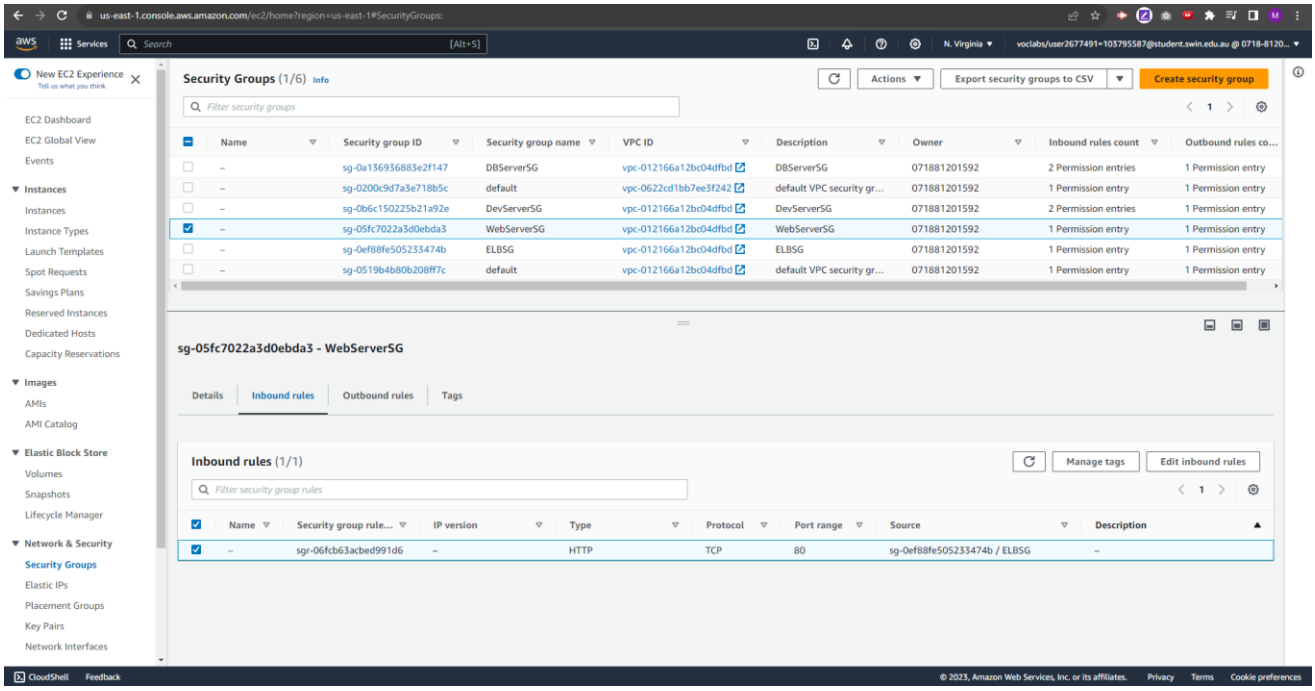


*Figure 4 - WebServerSG inbound rules details.*

### C. *Create DBSERVERSG*

The DBServerSG has 2 inbound rules that allow MySQL (3306) from WebServerSG to retrieve metadata, and from DevServerSG to manage the RDS database through phpMyAdmin.

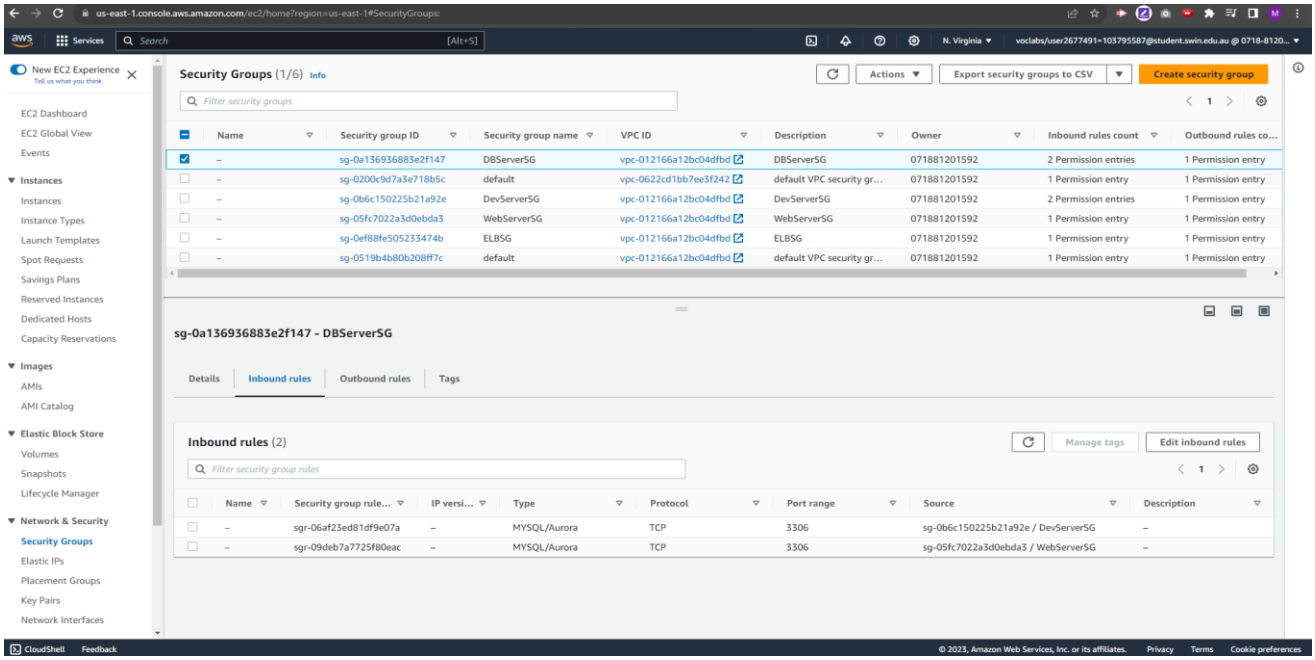


*Figure 5 - DBServerSG inbound rules details.*

*D. Create DevServerSG*

The DevServerSG has 2 inbound rules that allow SSH and HTTP traffic from anywhere.

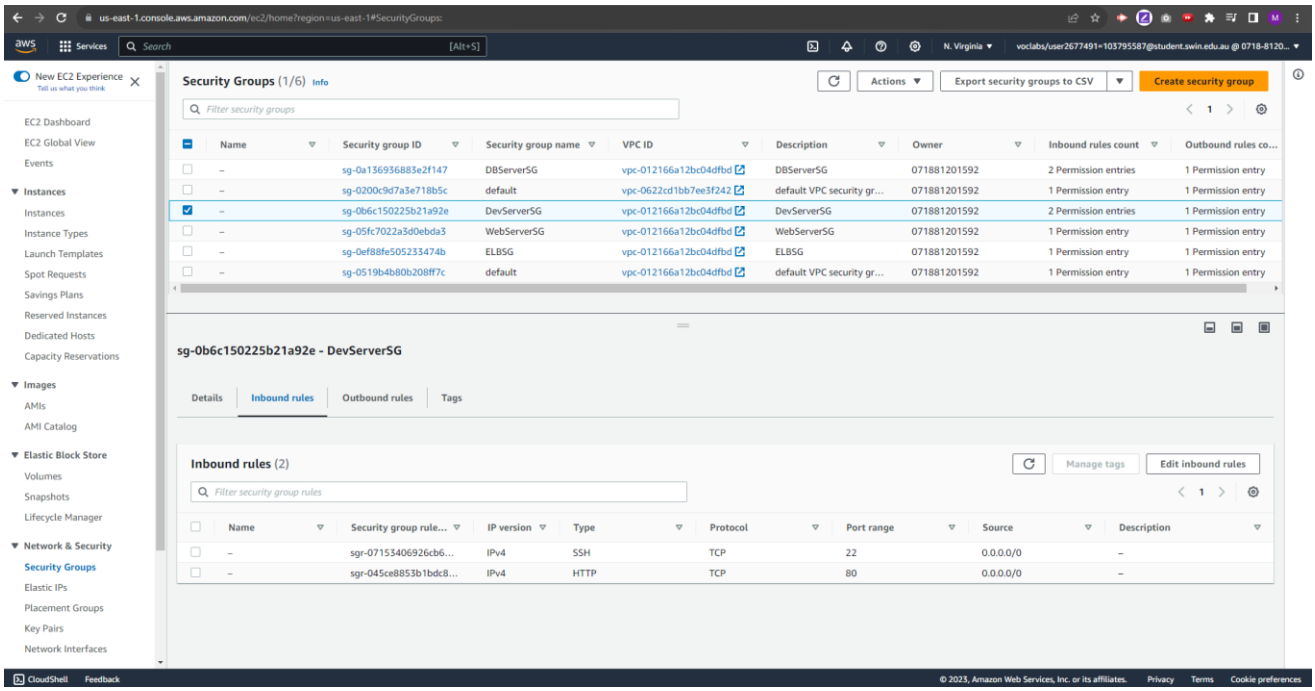


*Figure 6 – DevServerSG inbound rules details.*

## V. NACL CONFIGURATION

"PrivateSubnetsNACL" for the web servers has an inbound and outbound rule that denies bidirectional ICMP traffic to/from Dev Server (10.0.2.0/24). And because NACL is stateless, a subsequent rule allowing all traffic from/to anywhere was added. The more specific rule (i.e., blocking ICMP traffic) is placed before the more general rule to prevent unintended traffic that should be blocked.

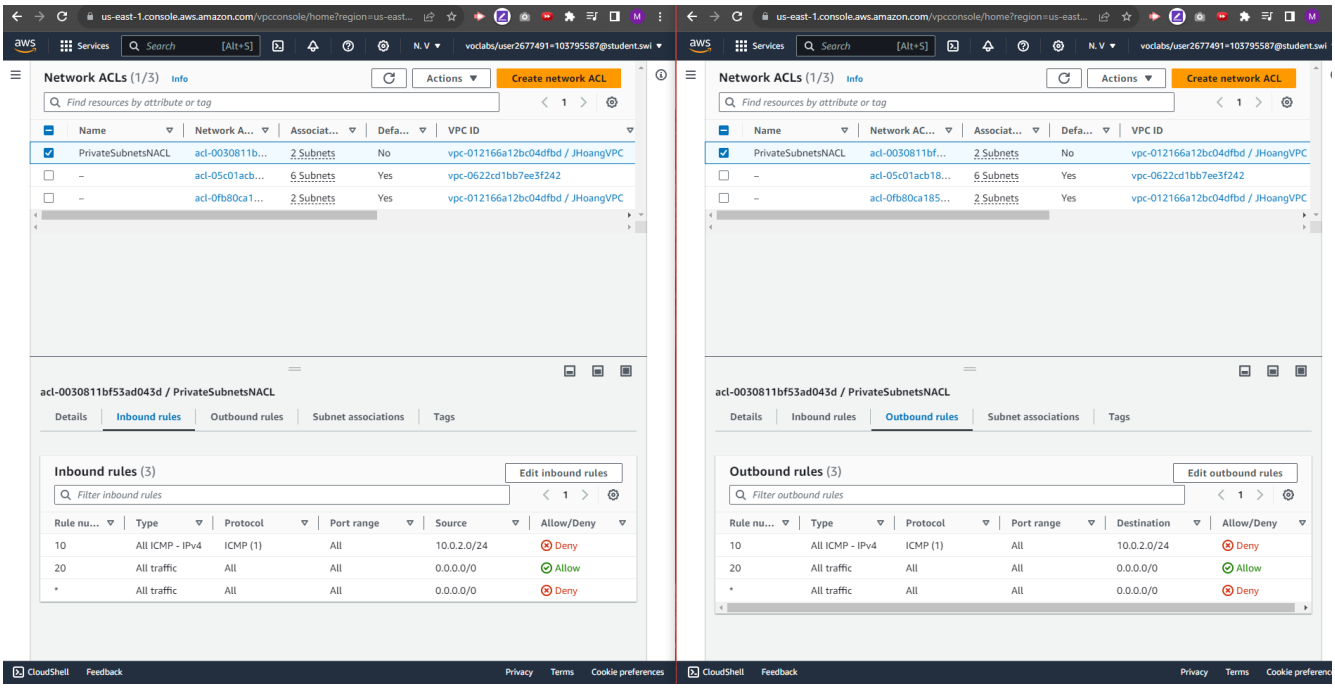


*Figure 7 - PrivateSubnetsNACL inbound and outbound rules details.*

## VI. IAM ROLES

The IAM role “LabRole” has been assigned to the “CreateThumbnail” Lambda function, which allows the function to put objects into the S3 bucket.

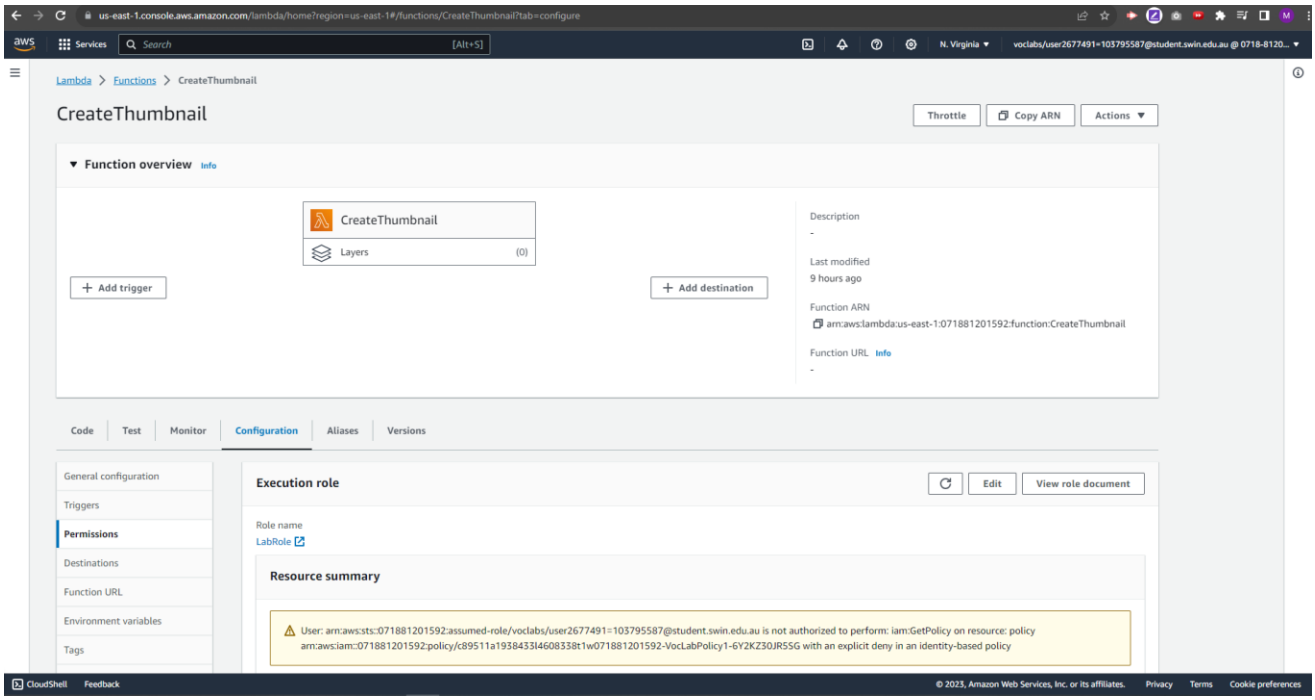


*Figure 8 – Lambda function execution role details.*

Similarly, the role “LabInstanceProfile” is also assigned to the WebServers (inside the ASG launch template) as their IAM instance profile.

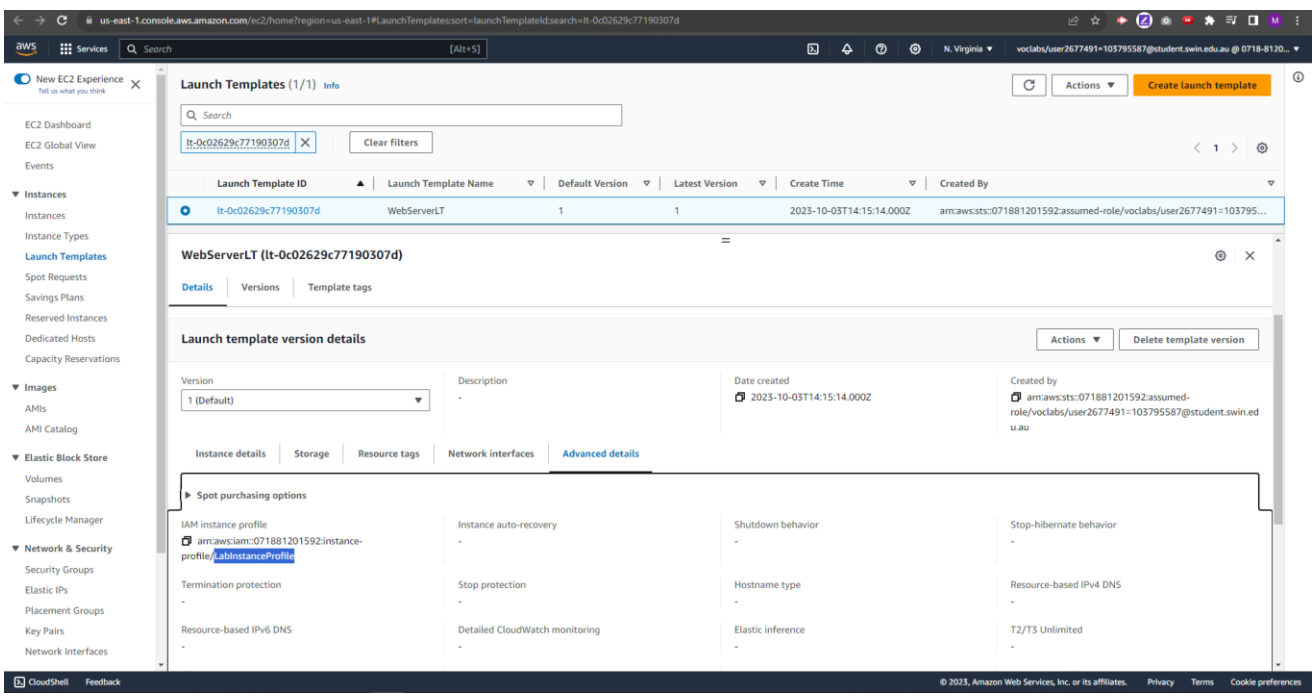


*Figure 9 – Launch template advanced details.*

## VII. ELB CONFIGURATION

A target group, used by the ELB, is configured with a health check path set to '/photoalbum/album.php.' Currently, there are 2 healthy targets, and no targets are marked as unhealthy.

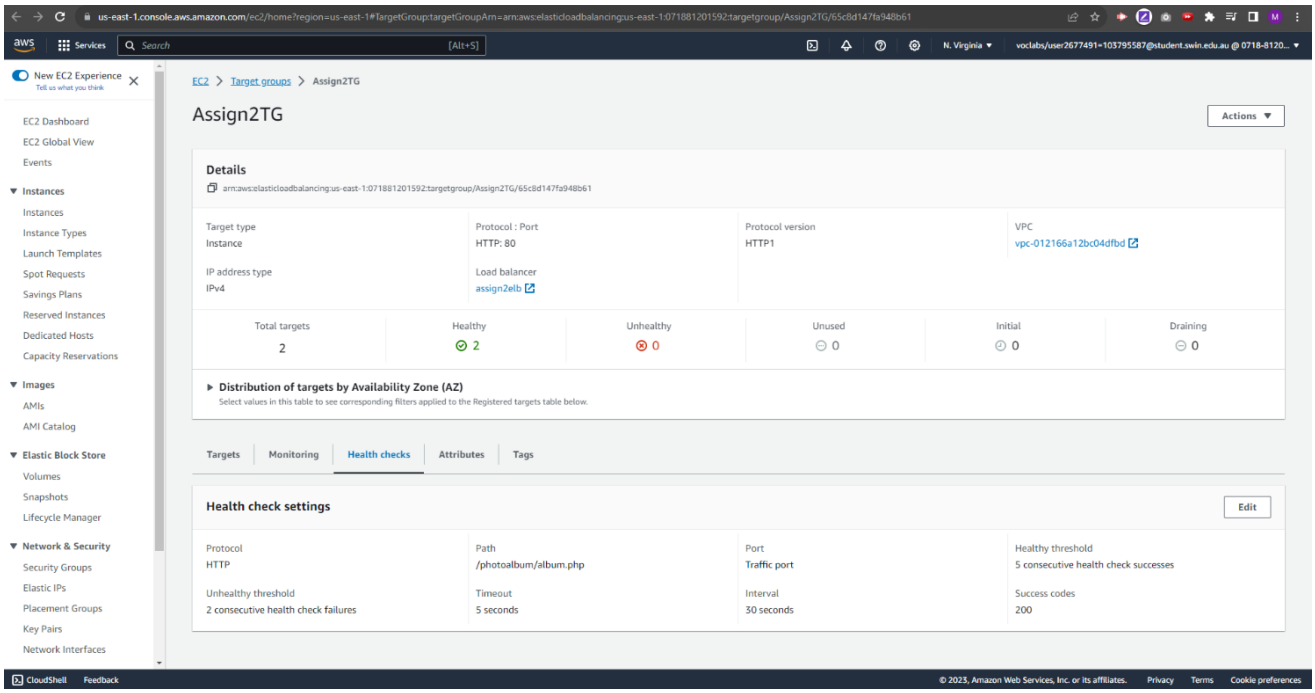


*Figure 10 – "Assign2TG" target group details.*

The "Assign2LB" Application Load Balancer uses the TG created earlier, is associated with ELBSG, and is deployed across 2 public subnets. It uses ELBSG and is placed into 2 public subnets.

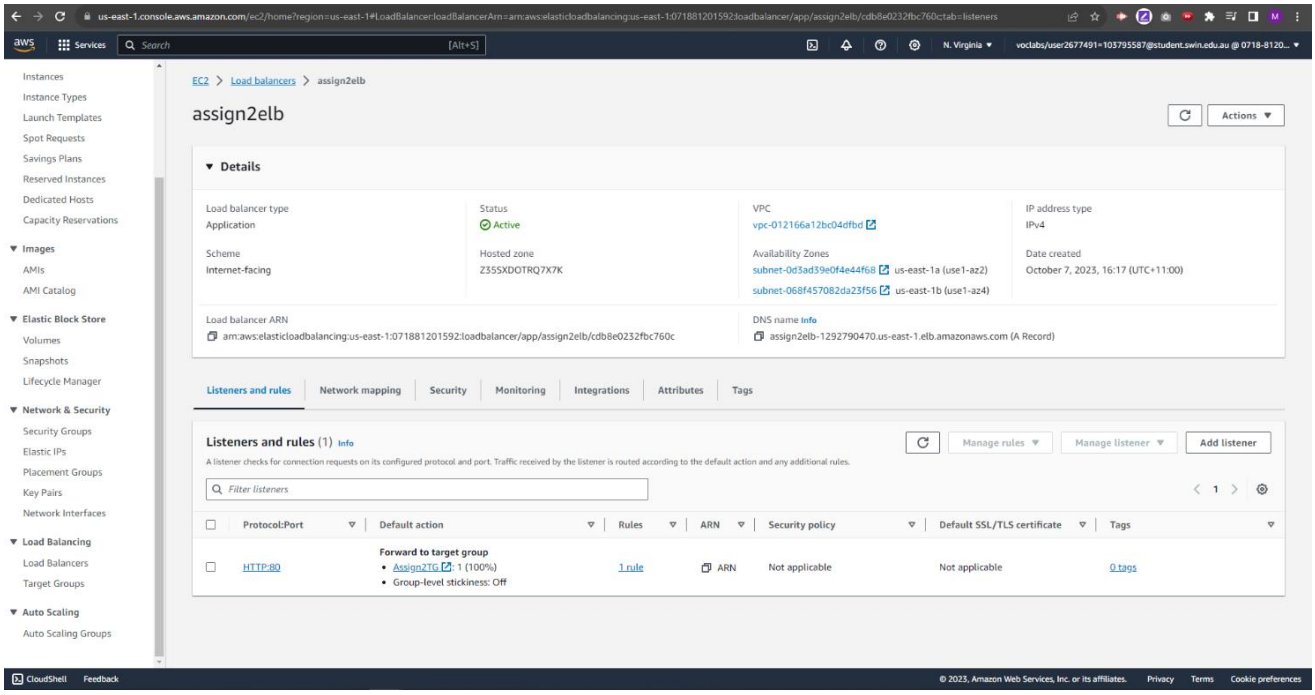


*Figure 11 – "assign2elb" ELB details.*

## VIII. ASG CONFIGURATION

The ASG to launch Web Server instances has a desired capacity, minimum capacity, and maximum capacity set to 2, 2, and 3, respectively. The figure also shows the "Assign2LT" launch template used by the ASG, which uses the DevServerAMI, has an instance type as t2.micro, and associates the instances with the WebServerSG security group.

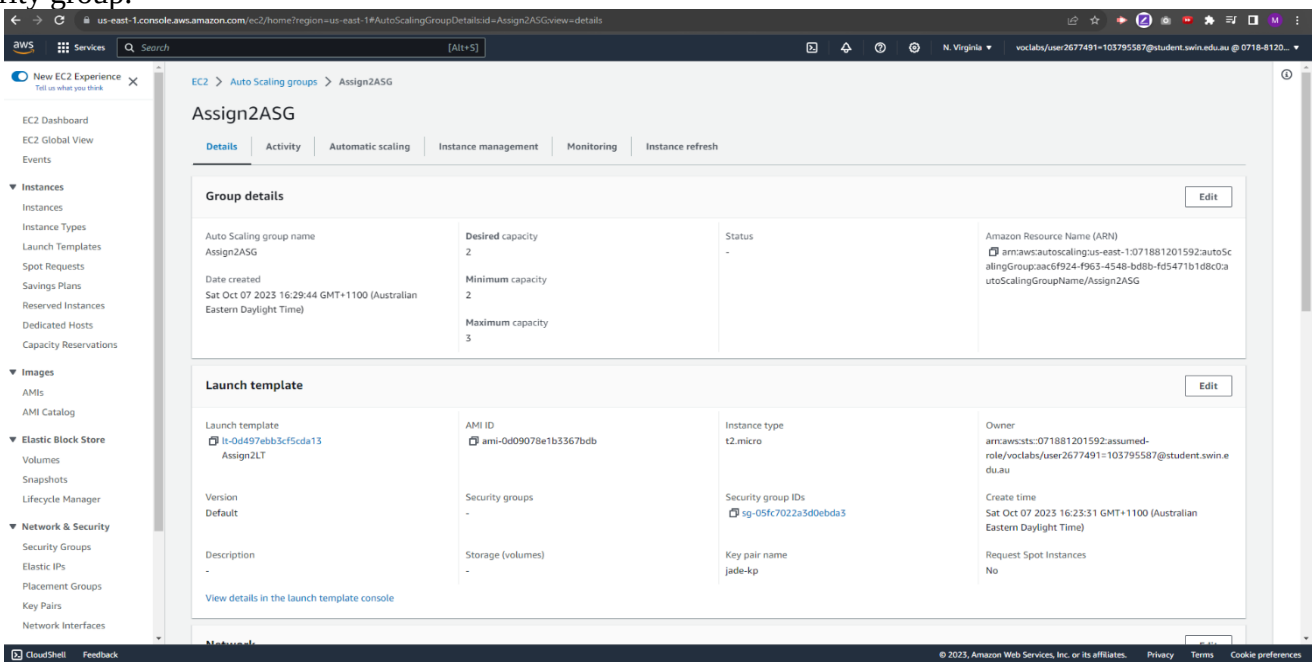


*Figure 12 – "Assign2ASG" details.*

The ASG is also configured with a target tracking scaling policy set to the ELB request count per target of 30 as per the assignment specification.

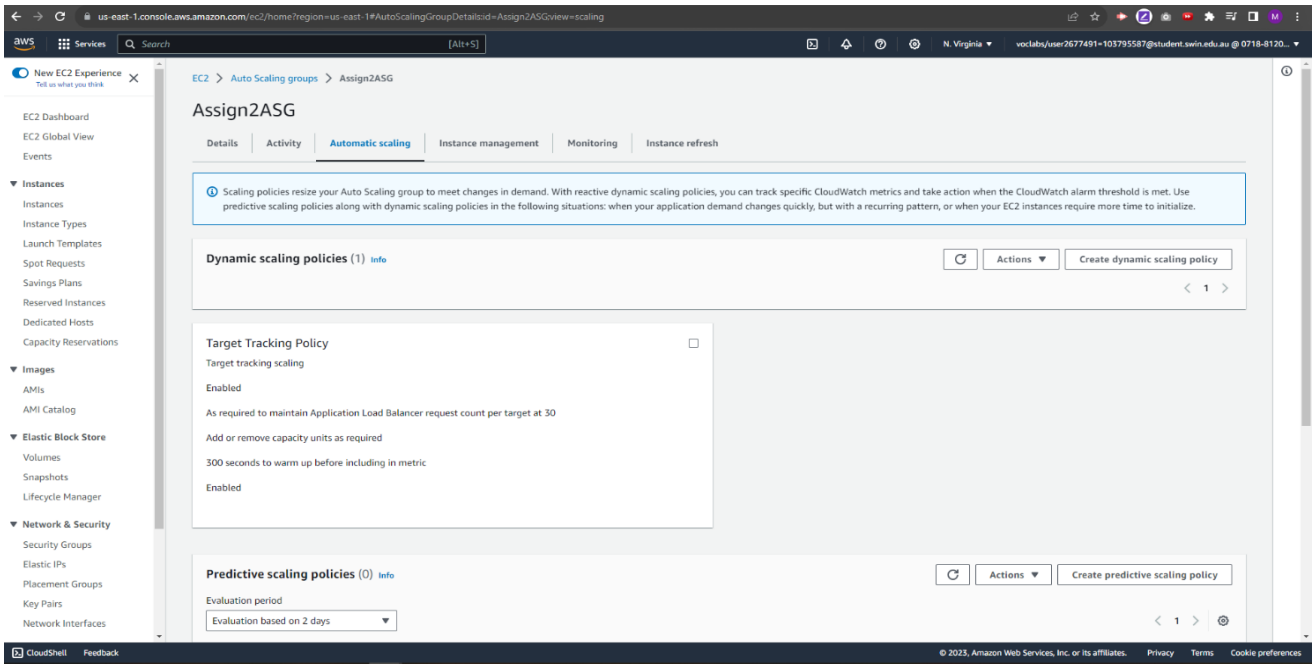


*Figure 13 – "Assign2ASG" target tracking policy details.*

## IX. S3 BUCKET CONFIGURATIONS

The 'jade-photos' S3 bucket is configured with block public access settings, as shown in the figure below, adhering to the principle of least privilege. This configuration ensures that direct access to the S3 photos is not accessible.

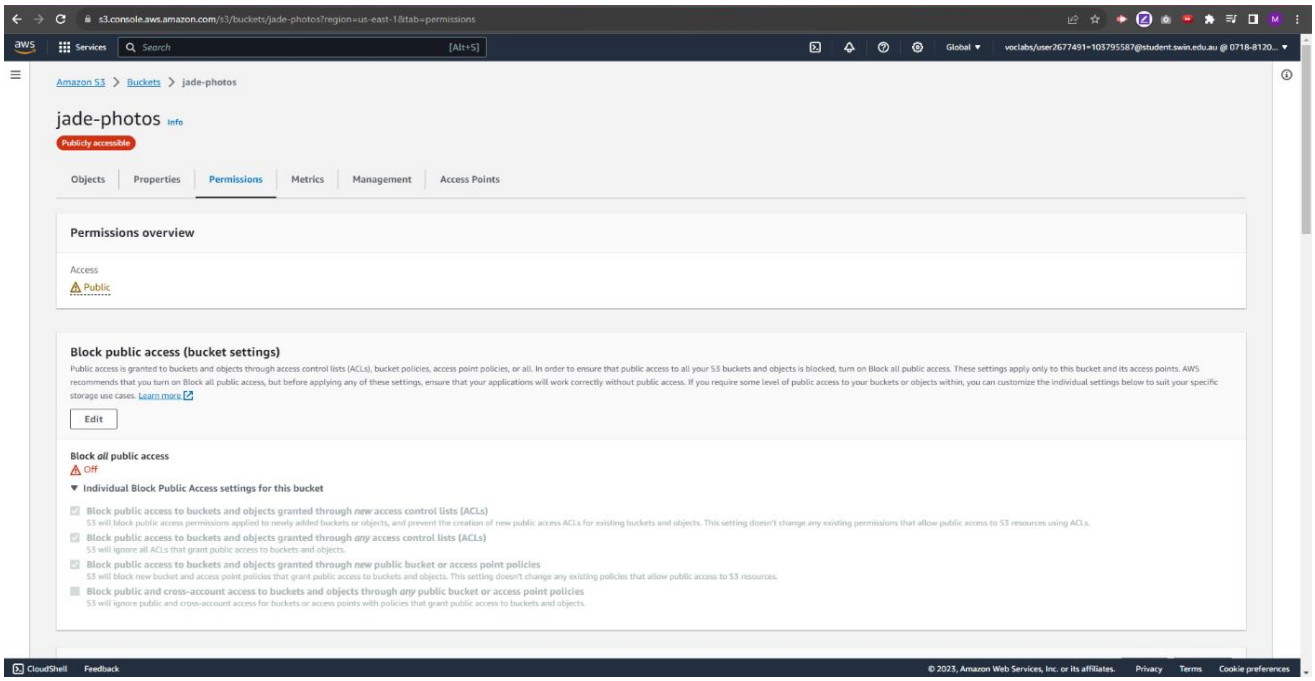


*Figure 14 - S3 block public access bucket settings.*

Following the least-privilege principle, an HTTP referrer part has been added to the S3 bucket policy. This is to only allow GET requests originating from the ELB's DNS address so that the album.php can properly display the photos from S3 buckets.

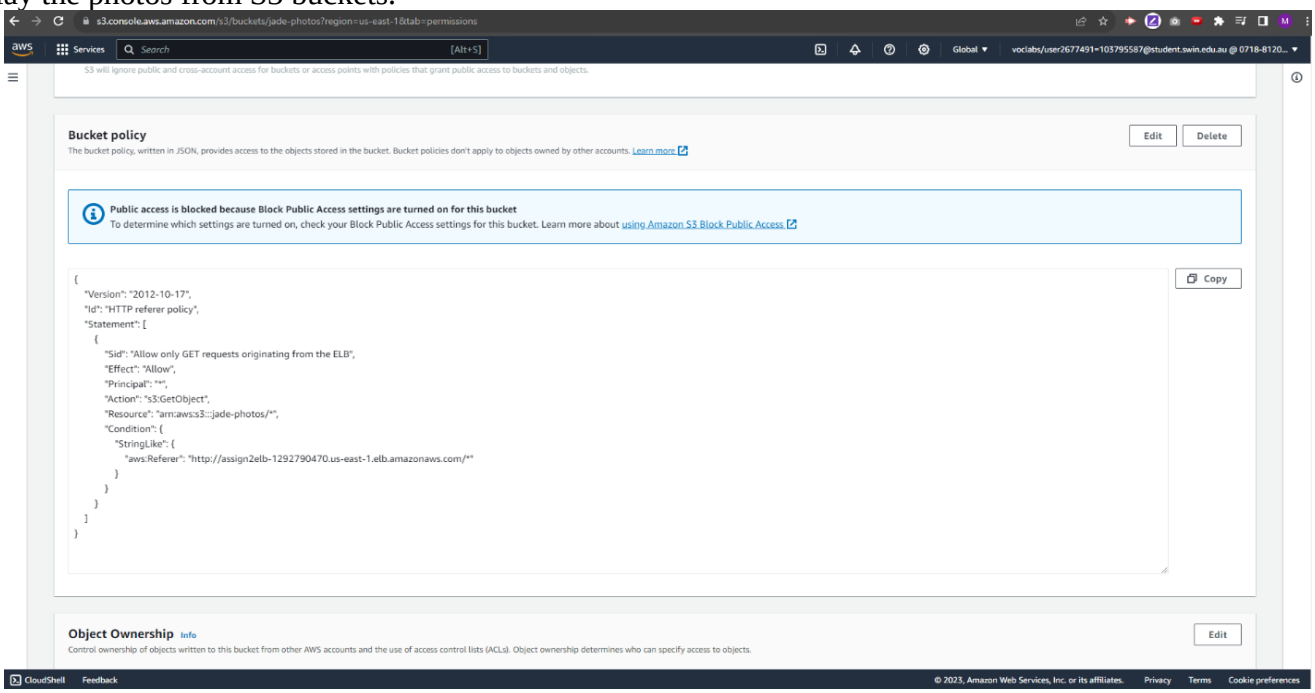


*Figure 15 - S3 bucket policy details.*

## X. LAMBDA CONFIGURATION

The ‘CreateThumbnail’ Lambda function resizes the images and uploads the resized versions to the same S3 bucket. It is configured with Python 3.11 runtime and arm64 architecture, and it uses “LabRole” as mentioned before. I also increased the Lambda function’s timeout settings so that it can handle larger photos, not just small ones.

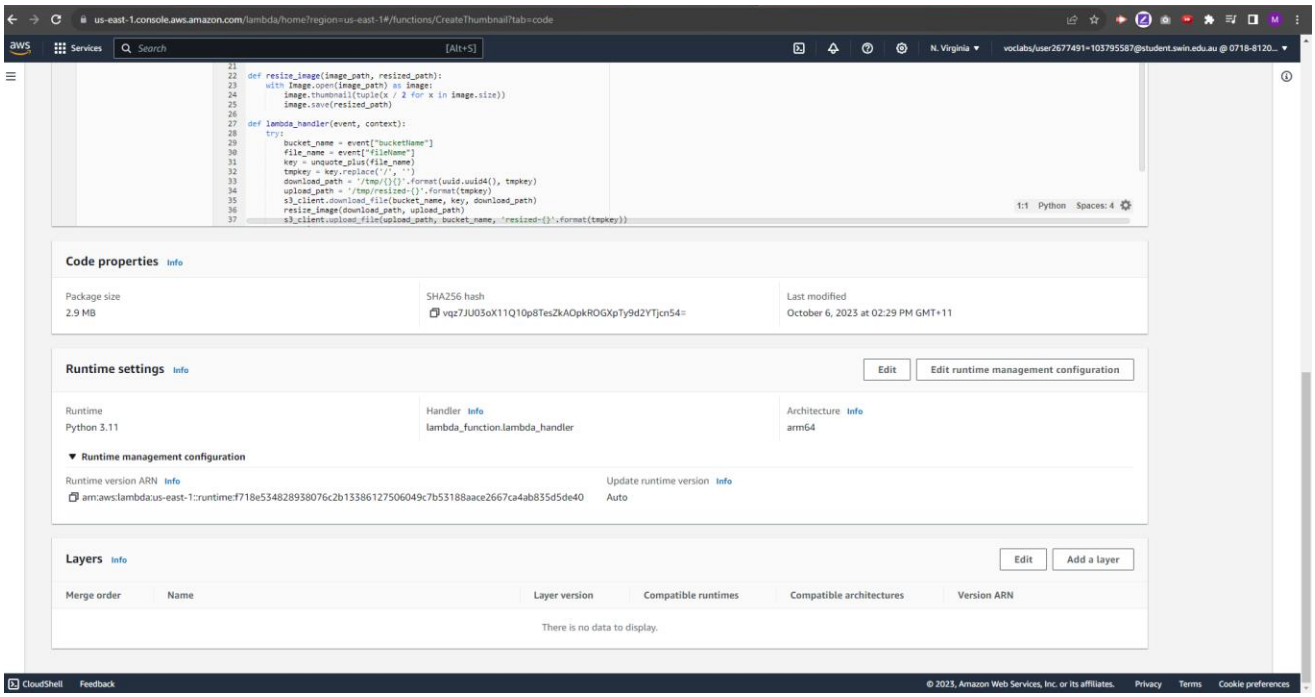


*Figure 16 – Lambda function runtime settings.*

## XI. RDS INSTANCE CONFIGURATIONS

### A. *Create RDS instance*

The RDS instance is configured to use the "jhoangsubnetgroup" subnet group, ensuring that it remains in the private subnets. Specifically, it is located in Private Subnet 1 and uses the DBServerSG security group.

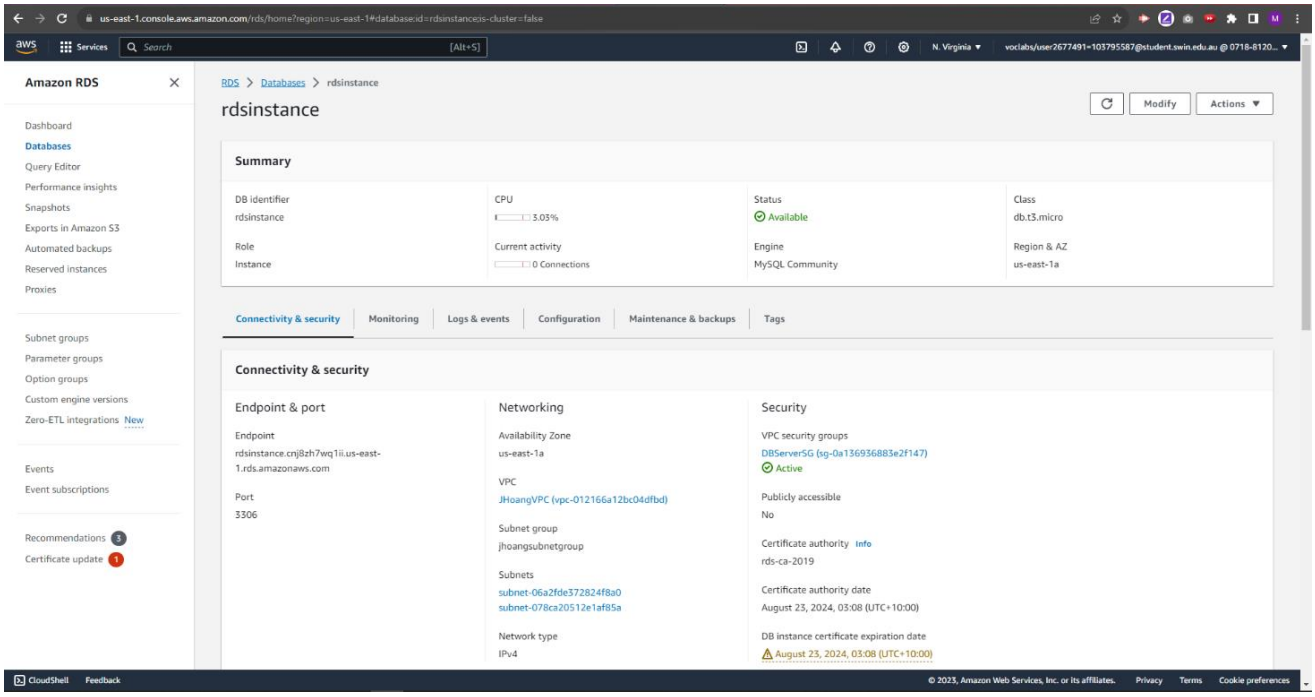


*Figure 17 - RDS instance details.*

### B. *Database schema*

The RDS database's "photos" table has a schema with 5 columns and their respective data types, as shown in the figure below. The RDS database is managed via phpMyAdmin on the DevServer.

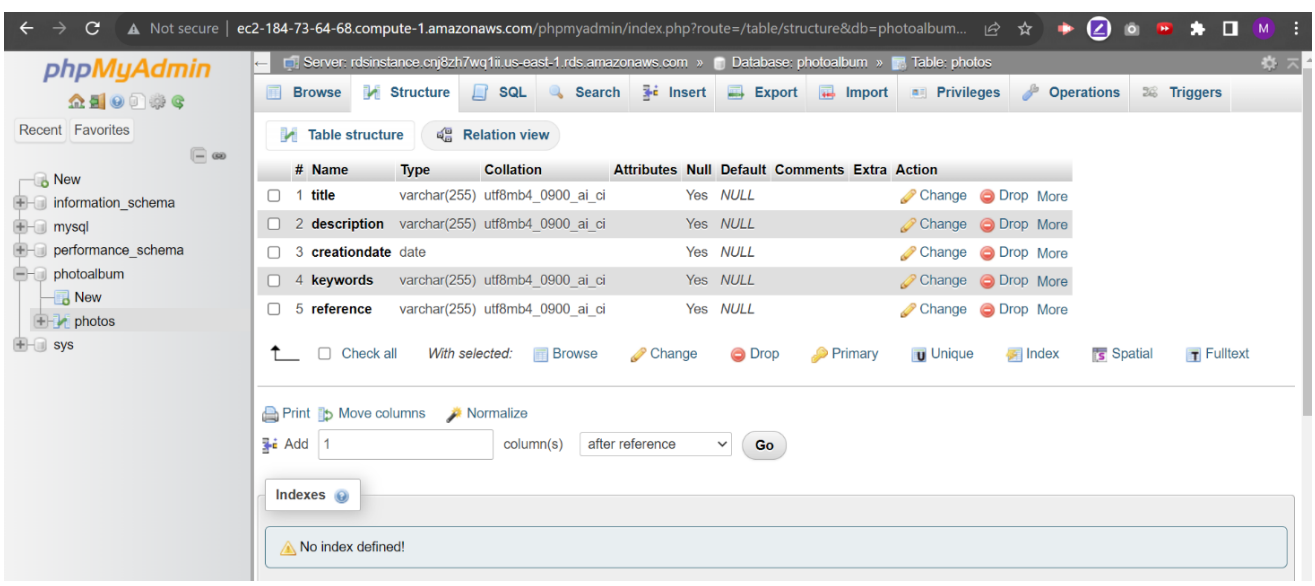


*Figure 18 - RDS "photos" table structure.*

## XII. Website verification

### A. *Accessing the website*

The website displays 4 photos and their corresponding metadata on the album.php page. Here is the URL via ELB: http://assign2elb-1292790470.us-east-1.elb.amazonaws.com/photoalbum/album.php

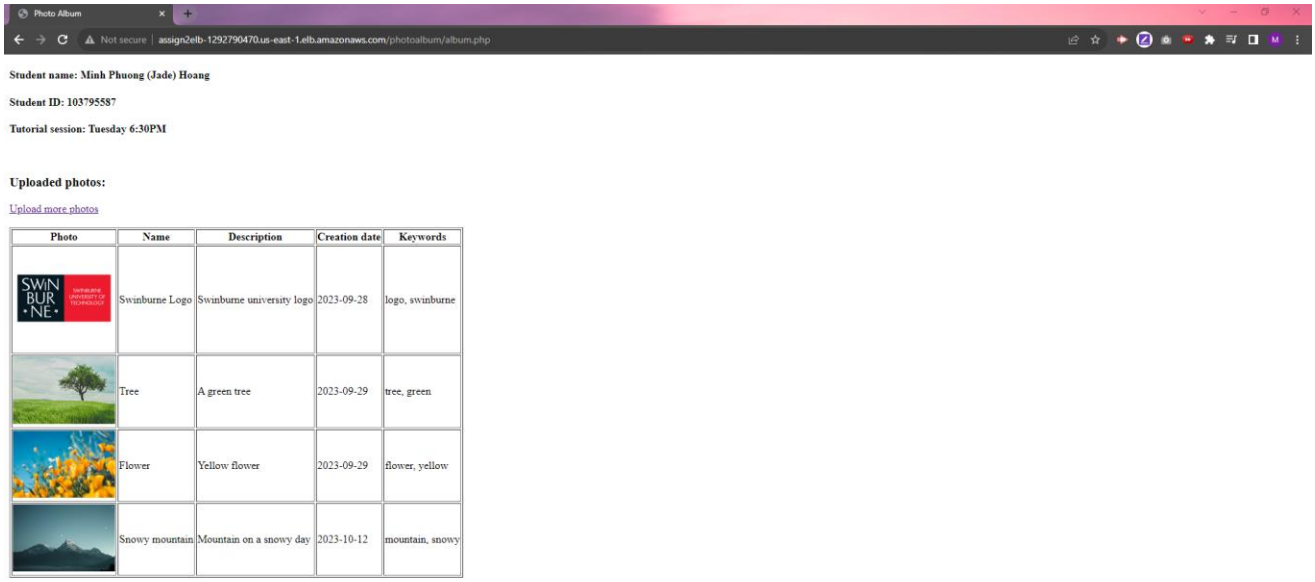


*Figure 19 - Album.php webpage.*

### B. *RDS data records inserted and photos uploaded to the S3 bucket*

As a result, there are 4 data records inserted by the EC2 instances into the RDS "photos" table.

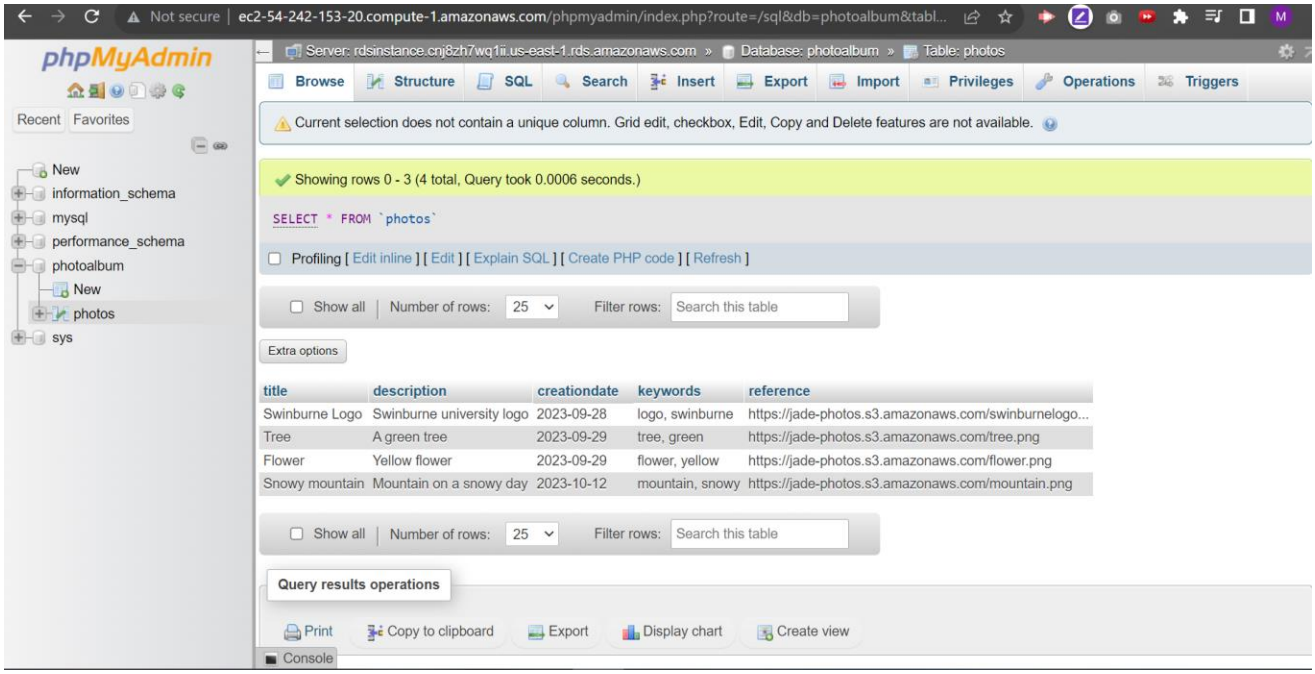


*Figure 20 – Rows inserted in the RDS 'photos' table.*

These records correspond to the 4 photos that were uploaded to the S3 bucket via the photouploader.php page. Additionally, each uploaded photo has an associated "resized" version created by the Lambda function. This results in a total of 8 photos in the S3 bucket, as indicated in Figure 21.

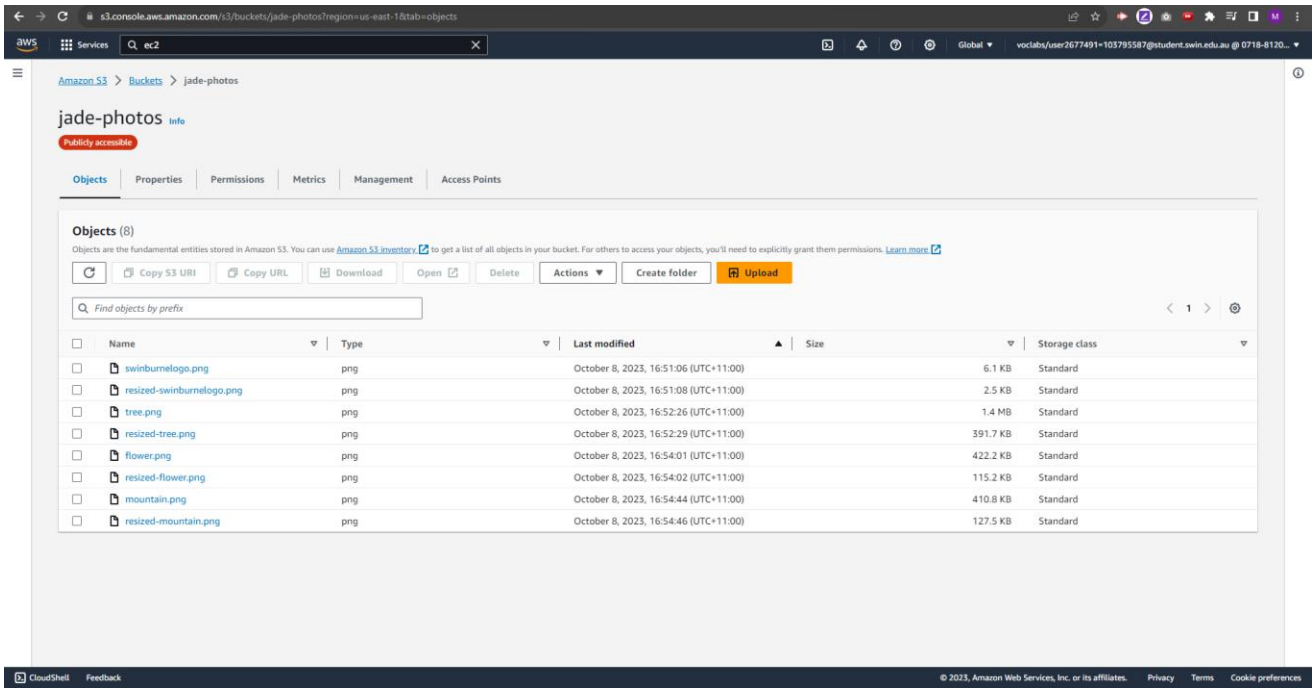


*Figure 21 - S3 photos with their resized versions.*