

# Deploying A Highly Available Photo Album Website

Minh Phuong (Jade) Hoang

October 8, 2023

[jadehoang167@gmail.com](mailto: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.

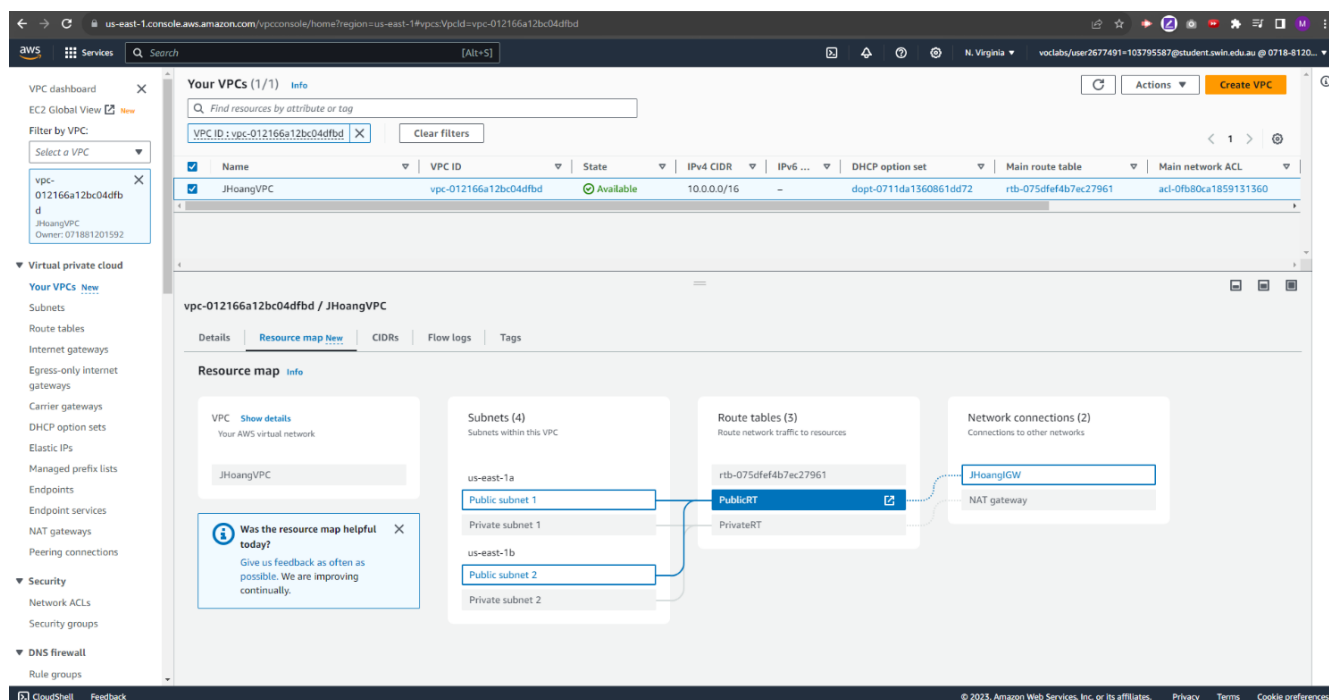


Figure 1 - JHoangVPC resource map.

### 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.

The figure consists of two side-by-side screenshots of the AWS Management Console, specifically the 'Route tables' section. Both screenshots show a list of route tables and a detailed view of a specific route table's routes.

**Left Screenshot: PrivateRT Details**

The top section shows a list of route tables:

Name	Route table ID	Explicit subnet associations
-	rtb-075dfe4b7ec27961	-
PublicRT	rtb-02a96804e0363531b	2 subnets
<b>PrivateRT</b>	<b>rtb-0e8fd079f209afc63</b>	2 subnets
-	rtb-0a03662b438729c06	-

The bottom section shows the 'Routes' for 'rtb-0e8fd079f209afc63 / PrivateRT':

Destination	Target	Status	Propagated
0.0.0.0/0	nat-039c83f146030e42e	Active	No
10.0.0.0/16	local	Active	No

**Right Screenshot: PublicRT Details**

The top section shows a list of route tables:

Name	Route table ID	Explicit subnet associations
-	rtb-075dfe4b7ec27961	-
<b>PublicRT</b>	<b>rtb-02a96804e0363531b</b>	2 subnets
PrivateRT	rtb-0e8fd079f209afc63	2 subnets
-	rtb-0a03662b438729c06	-

The bottom section shows the 'Routes' for 'rtb-02a96804e0363531b / PublicRT':

Destination	Target	Status	Propagated
0.0.0.0/0	igw-042e97d90ac3ec860	Active	No
10.0.0.0/16	local	Active	No

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.

The screenshot displays the AWS Management Console interface. On the left, the navigation menu shows various services, with 'Network & Security' expanded to show 'Security Groups'. The main content area is titled 'Security Groups (1/6)' and contains a table listing several security groups. The 'sg-0ef88fe505233474b' group, named 'ELBSG', is selected. Below the table, the 'Inbound rules' tab is active, showing a single rule with the following details:

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sg-0c746b02d097dd...	IPv4	HTTP	TCP	80	0.0.0.0/0	-

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.

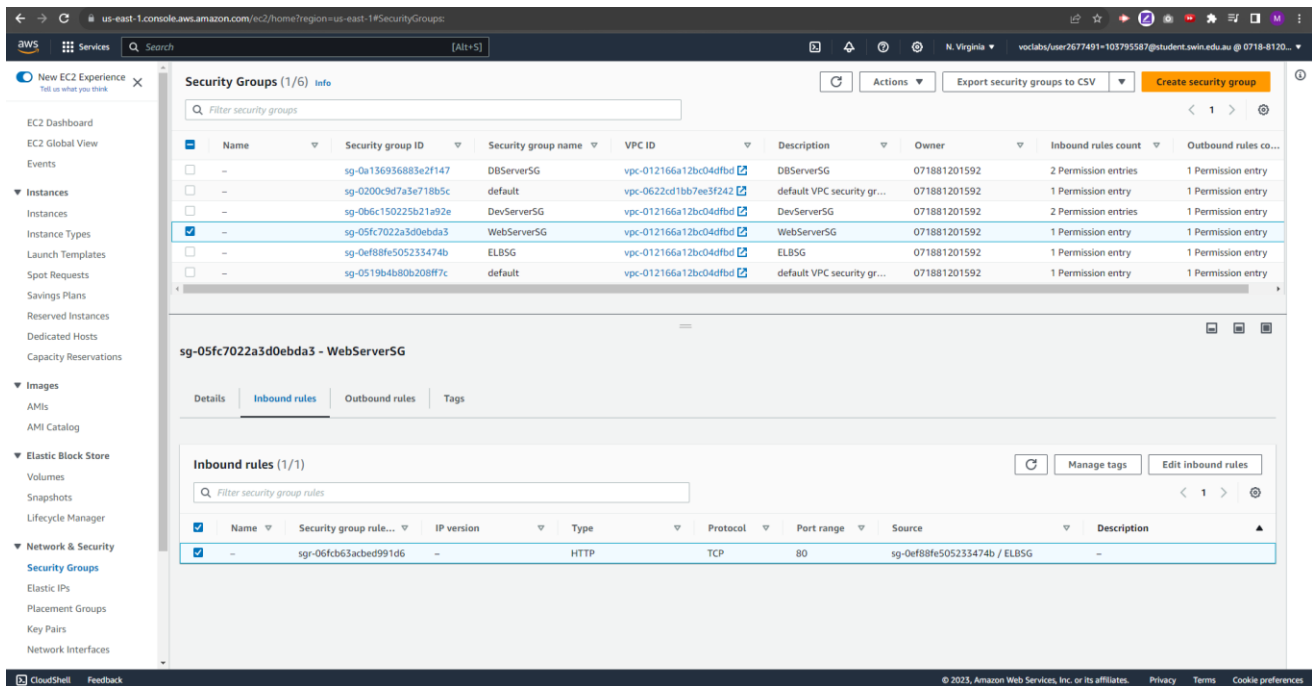


Figure 4 - WebServerSG inbound rules details.

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.

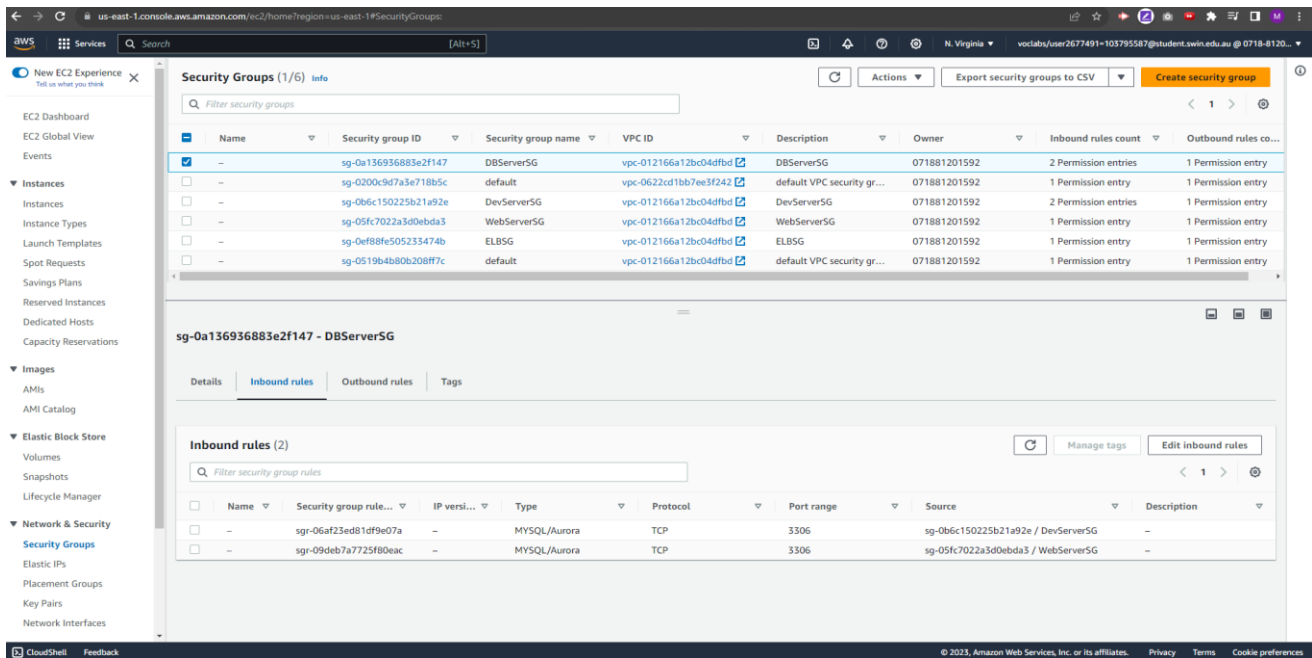


Figure 5 - DBServerSG inbound rules details.

#### D. Create DevServerSG

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

The screenshot displays the AWS Management Console interface for the 'us-east-1' region. The left-hand navigation pane shows various AWS services, with 'Network & Security' and 'Security Groups' highlighted. The main content area is titled 'Security Groups (1/6) Info'. A table lists several security groups, with 'sg-0b6c150225b21a92e' (DevServerSG) selected. Below this, the 'Inbound rules' tab is active, showing two rules: one for SSH (port 22) and one for HTTP (port 80), both allowing traffic from any source (0.0.0.0/0).

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules co...
-	sg-0a136936883e2f147	DBServerSG	vpc-012166a12bc04dfbd	DBServerSG	071881201592	2 Permission entries	1 Permission entry
-	sg-0200c9d7a3e718b5c	default	vpc-0622cd1bb7ee3f242	default VPC security gr...	071881201592	1 Permission entry	1 Permission entry
✓	sg-0b6c150225b21a92e	DevServerSG	vpc-012166a12bc04dfbd	DevServerSG	071881201592	2 Permission entries	1 Permission entry
-	sg-05fc7022a3d0ebda3	WebServerSG	vpc-012166a12bc04dfbd	WebServerSG	071881201592	1 Permission entry	1 Permission entry
-	sg-0ef88fe505233474b	ELBSG	vpc-012166a12bc04dfbd	ELBSG	071881201592	1 Permission entry	1 Permission entry
-	sg-0519b4b80b208f7c	default	vpc-012166a12bc04dfbd	default VPC security gr...	071881201592	1 Permission entry	1 Permission entry

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-07153406926cb6...	IPv4	SSH	TCP	22	0.0.0.0/0	-
-	sgr-045ce8853b1bdc8...	IPv4	HTTP	TCP	80	0.0.0.0/0	-

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.

The figure displays two screenshots of the AWS Management Console, showing the configuration of the PrivateSubnetsNACL (Network ACL) for the web servers. The left screenshot shows the Inbound rules tab, and the right screenshot shows the Outbound rules tab. Both screenshots show a list of rules for the Network ACL, with the first rule being a deny rule for ICMP traffic from/to 10.0.2.0/24, and the second rule being an allow rule for all traffic. The third rule is a deny rule for all traffic.

**Left Screenshot: Inbound rules (3)**

Rule no...	Type	Protocol	Port range	Source	Allow/Deny
10	All ICMP - IPv4	ICMP (1)	All	10.0.2.0/24	Deny
20	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

**Right Screenshot: Outbound rules (3)**

Rule no...	Type	Protocol	Port range	Destination	Allow/Deny
10	All ICMP - IPv4	ICMP (1)	All	10.0.2.0/24	Deny
20	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

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.

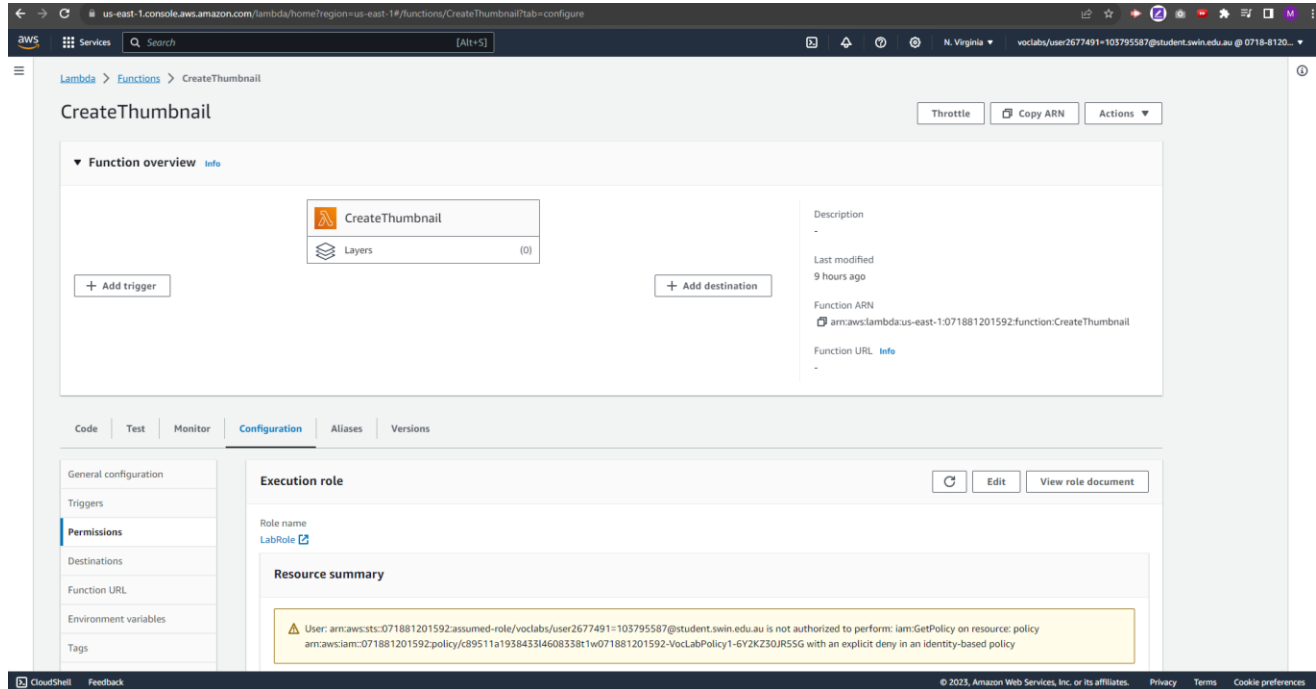


Figure 8 – Lambda function execution role details.

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

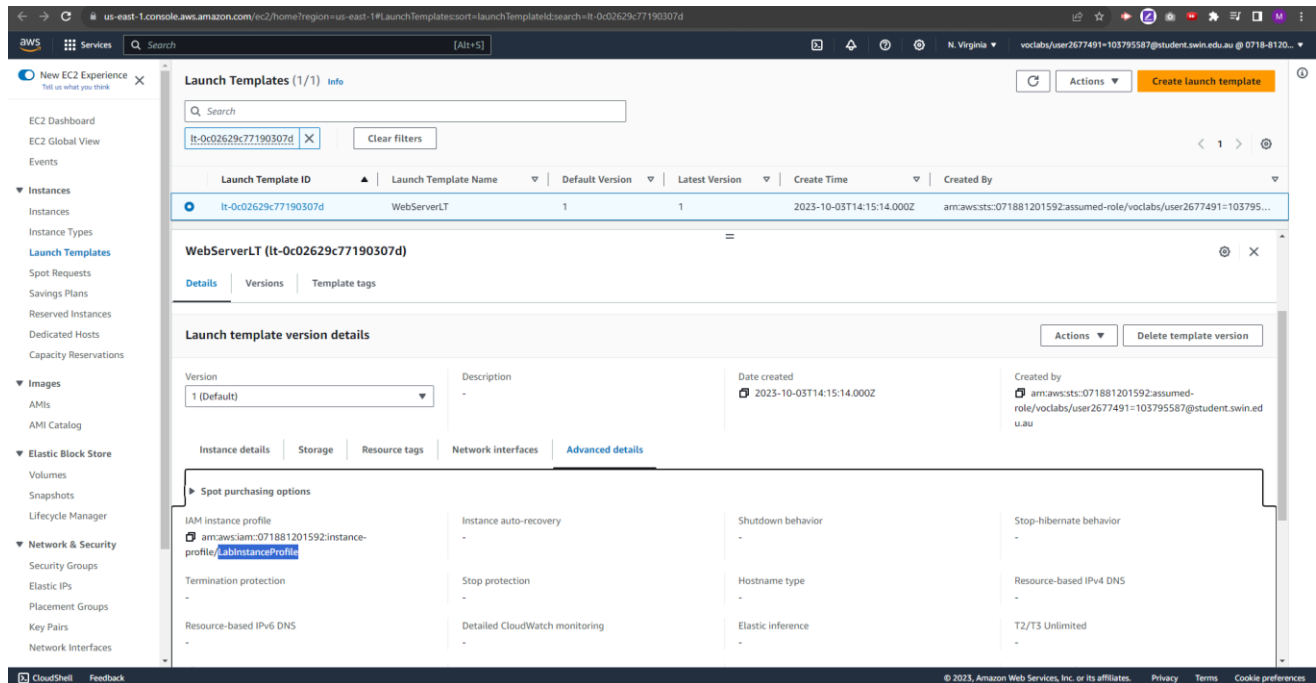


Figure 9 – Launch template advanced details.

## 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.

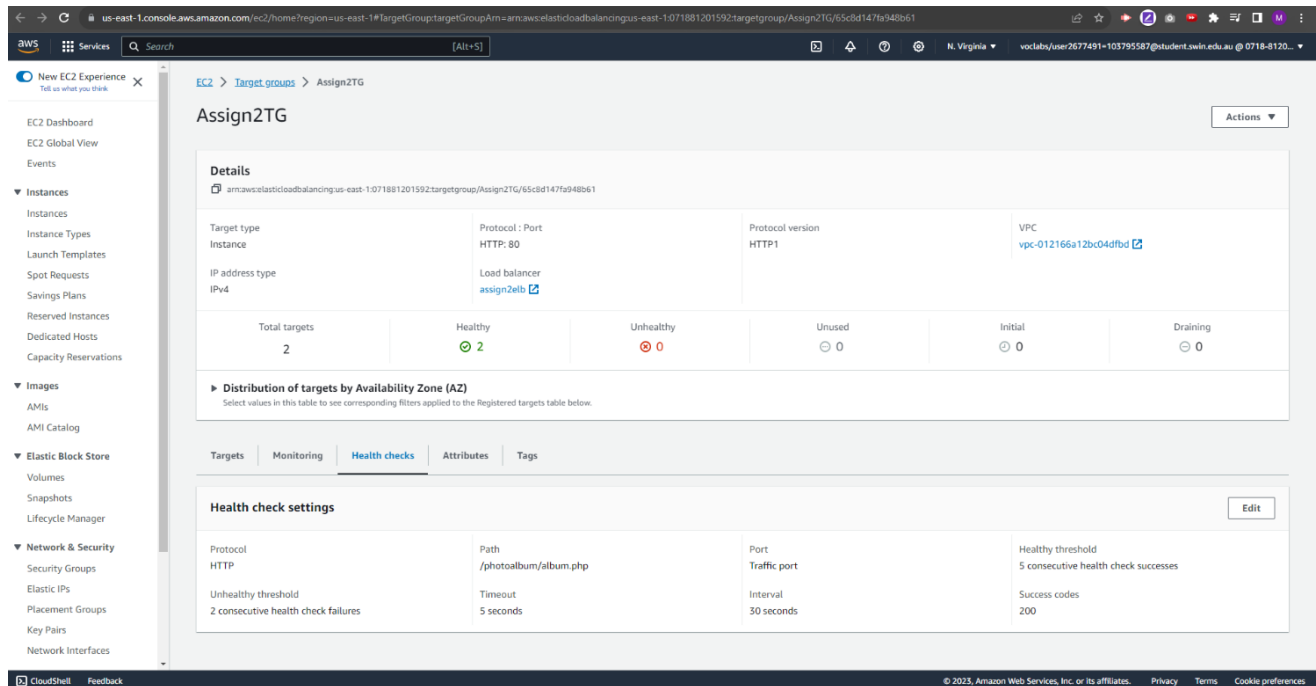


Figure 10 – “Assign2TG” target group details.

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.

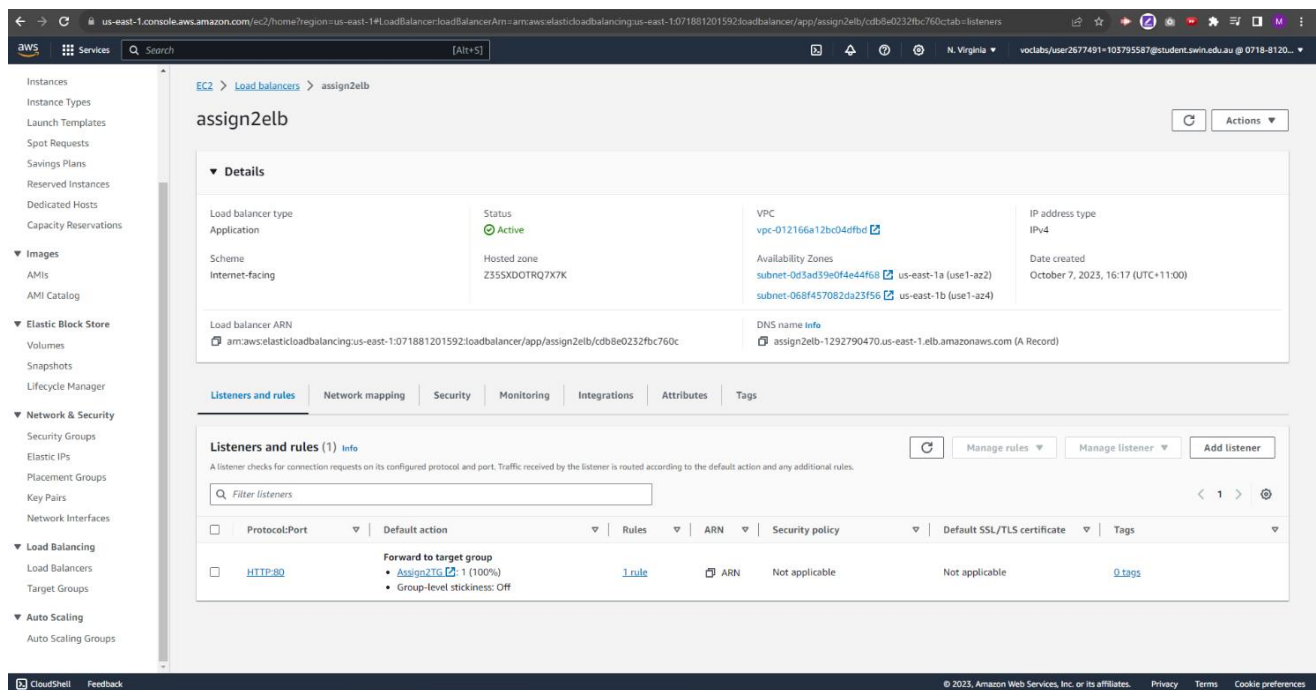


Figure 11 – “assign2elb” ELB details.



## 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.

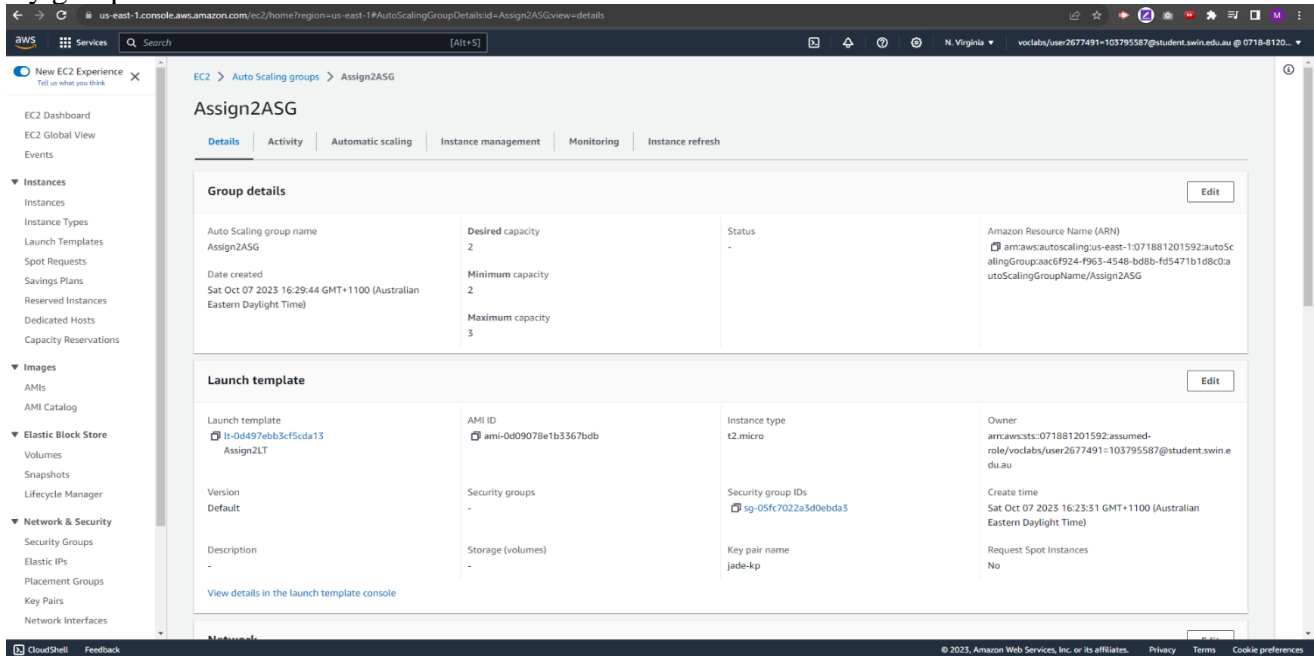


Figure 12 – “Assign2ASG” details.

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.

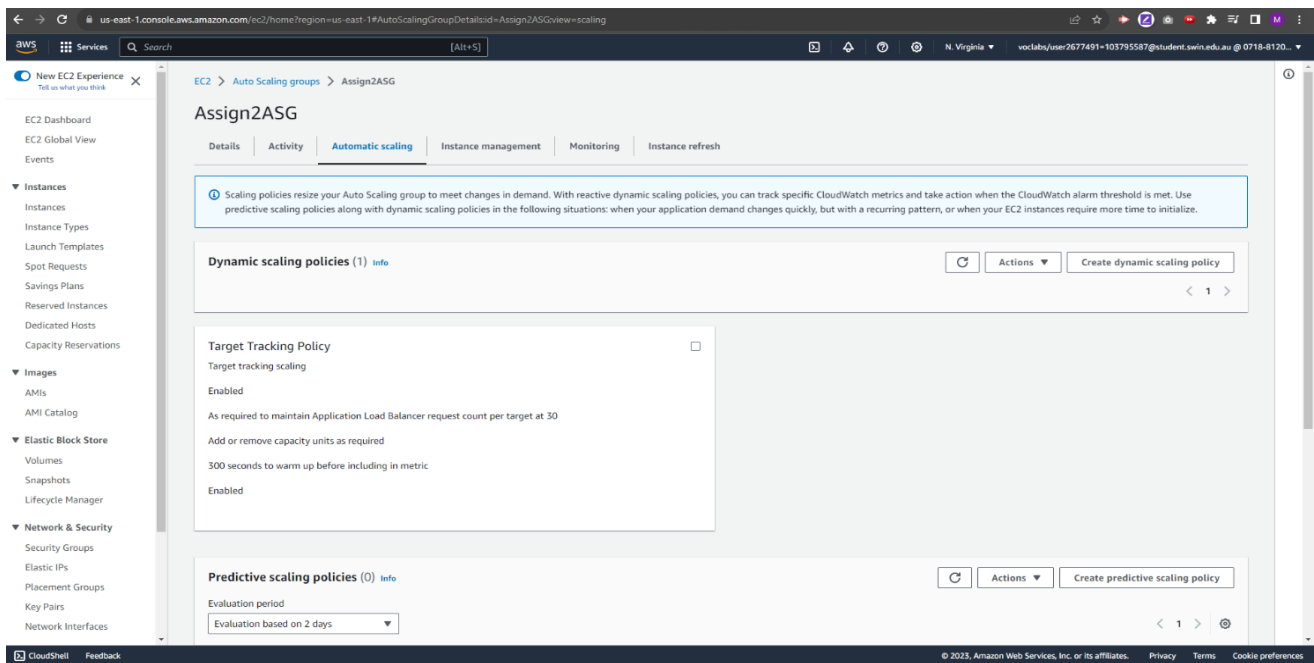


Figure 13 – “Assign2ASG” target tracking policy details.

## 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.

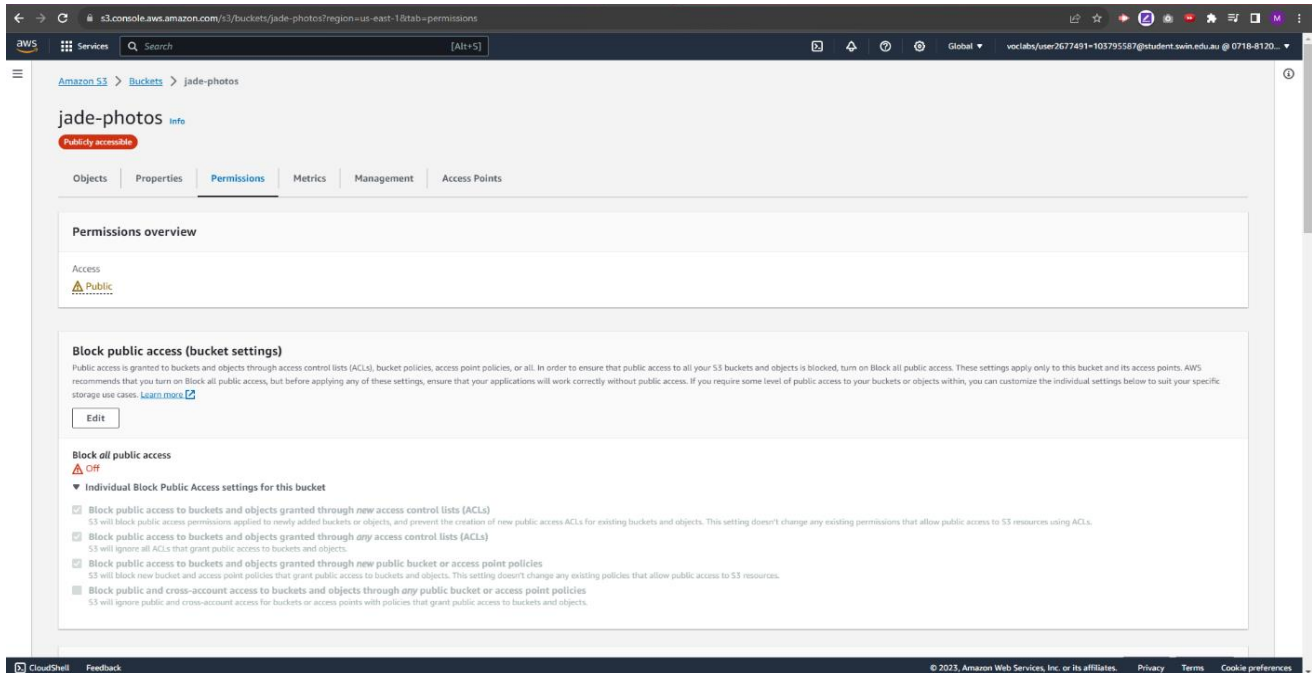


Figure 14 - S3 block public access bucket settings.

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.

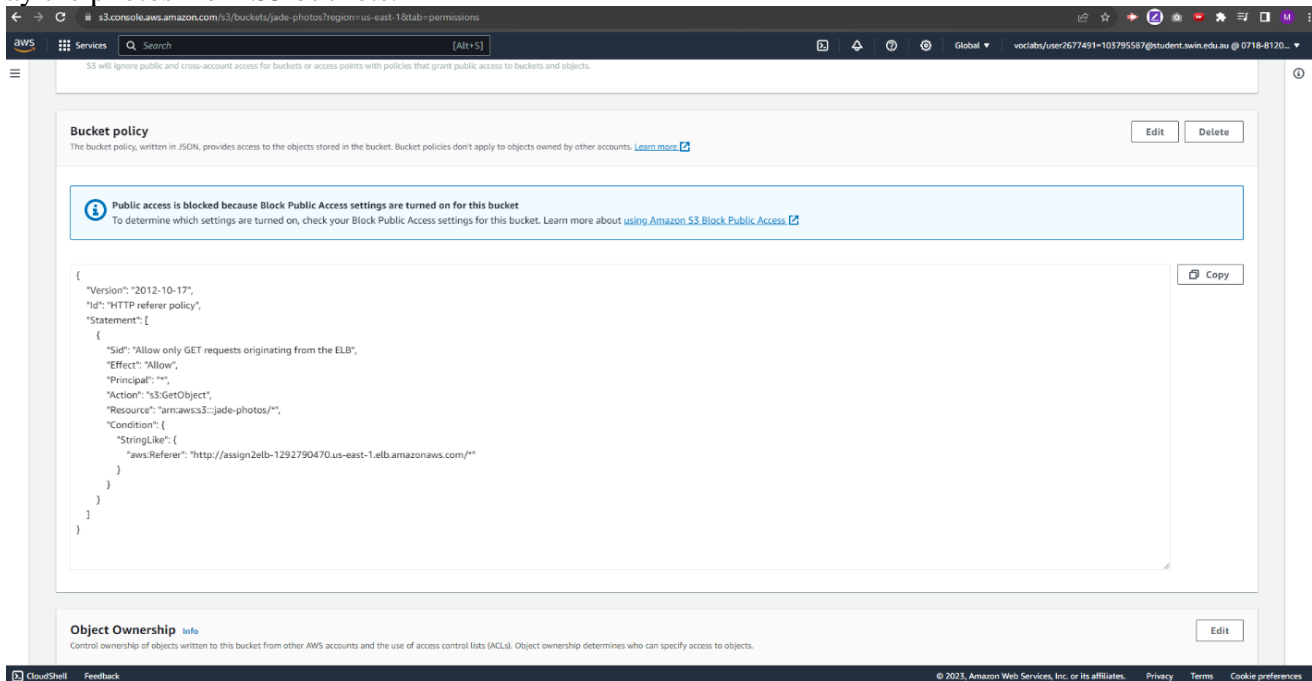


Figure 15 - S3 bucket policy details.

## 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.

The screenshot displays the AWS Lambda console for the 'CreateThumbnail' function in the us-east-1 region. The top section shows the function's code in Python 3.11, which uses the boto3 library to resize images and upload them to an S3 bucket. Below the code, the 'Code properties' section shows a package size of 2.9 MB, a SHA256 hash, and a last modified date of October 6, 2023. The 'Runtime settings' section shows the runtime as Python 3.11, the handler as 'lambda\_function.lambda\_handler', and the architecture as arm64. The 'Runtime management configuration' section shows the runtime version as 'auto' and the runtime version ARN as 'arn:aws:lambda:us-east-1::runtime:f718e534828938076c2b13386127506049c7b53188ace2667ca4ab835d5de40'. The 'Layers' section shows a table with columns for Merge order, Name, Layer version, Compatible runtimes, Compatible architectures, and Version ARN, but it currently displays 'There is no data to display.'

```
21
22 def resize_image(image_path, resized_path):
23     with Image.open(image_path) as image:
24         image.thumbnail((tuple(x / 2 for x in image.size)))
25         image.save(resized_path)
26
27 def lambda_handler(event, context):
28     """
29     bucket_name = event['bucketname']
30     file_name = event['filename']
31     key = unquote_plus(file_name)
32     tmpkey = key.replace('/', '-')
33     download_path = f'/tmp/{tmpkey}'.format(uuid.uuid4(), tmpkey)
34     upload_path = f'/tmp/resized-{tmpkey}'.format(tmpkey)
35     s3_client.download_file(bucket_name, key, download_path)
36     resize_image(download_path, upload_path)
37     s3_client.upload_file(upload_path, bucket_name, f'resized-{tmpkey}')
```

**Code properties** [Info](#)

Package size	SHA256 hash	Last modified
2.9 MB	<a href="#">vqz7JU03oX11Q10p8Tes2kAOpkROGXpTy9d2YTJcn54=</a>	October 6, 2023 at 02:29 PM GMT+11

**Runtime settings** [Info](#) [Edit](#) [Edit runtime management configuration](#)

Runtime	Handler	Architecture
Python 3.11	<a href="#">Info</a> lambda_function.lambda_handler	<a href="#">Info</a> arm64

**Runtime management configuration**

Runtime version ARN	Update runtime version
<a href="#">Info</a> <a href="#">arn:aws:lambda:us-east-1::runtime:f718e534828938076c2b13386127506049c7b53188ace2667ca4ab835d5de40</a>	<a href="#">Info</a> Auto

**Layers** [Info](#) [Edit](#) [Add a layer](#)

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
There is no data to display.					

© 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

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.

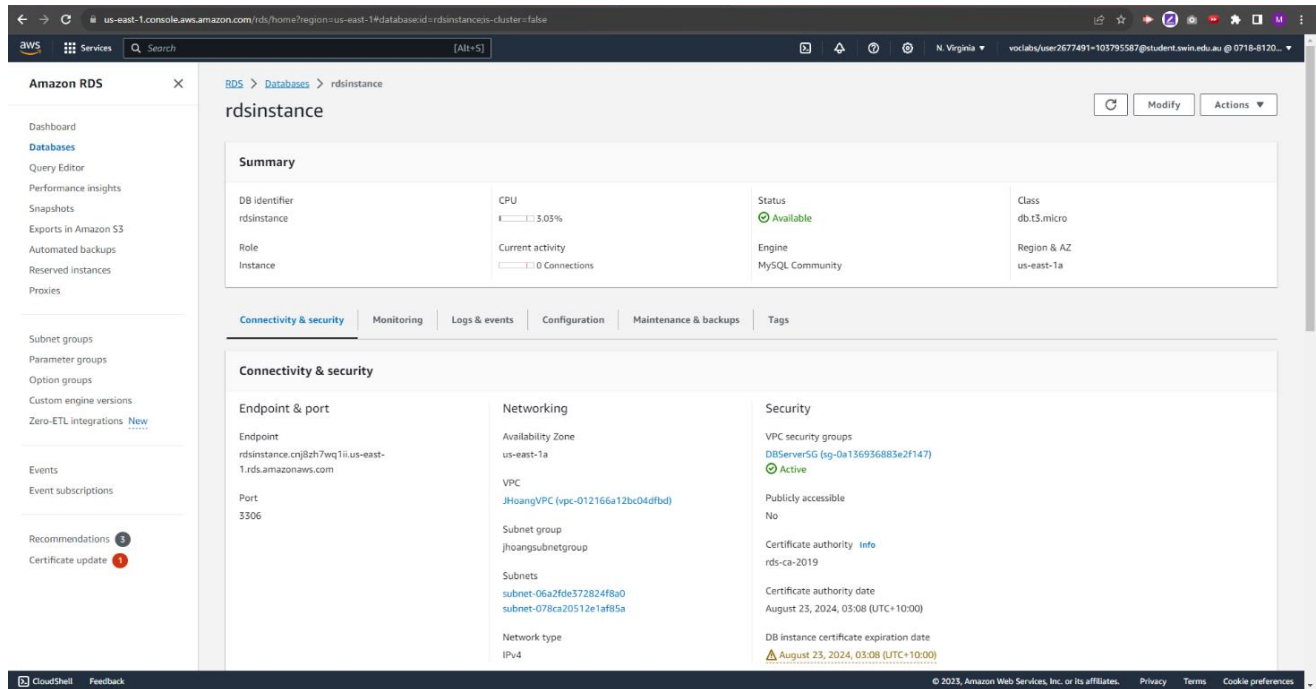


Figure 17 - RDS instance details.

### 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.

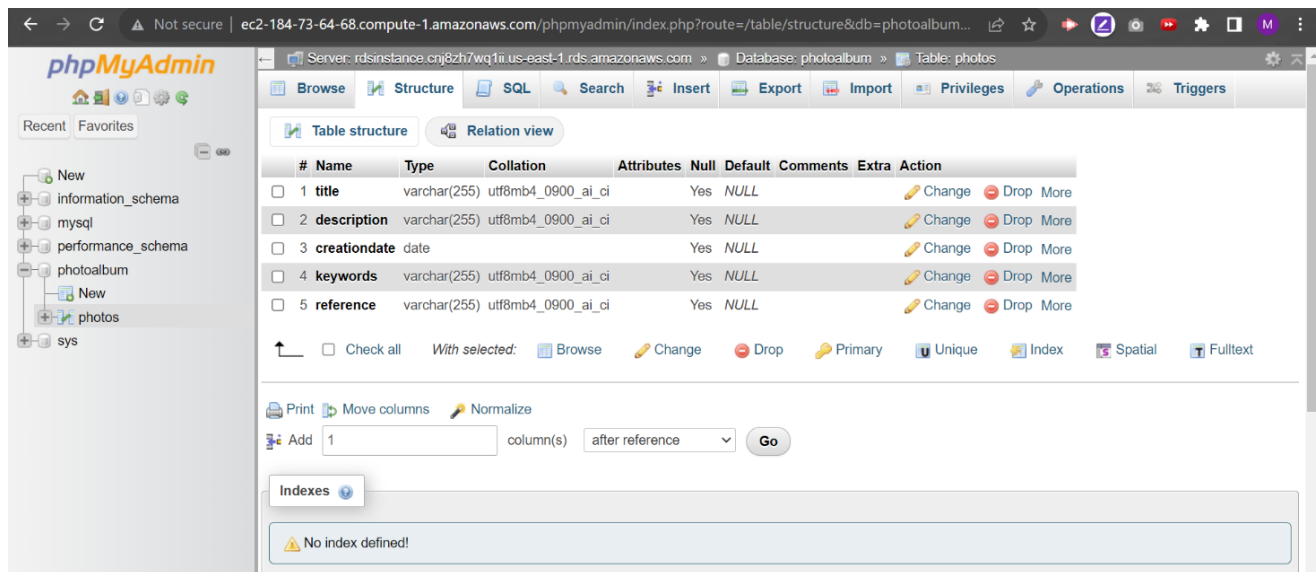


Figure 18 - RDS "photos" table structure.

## 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>

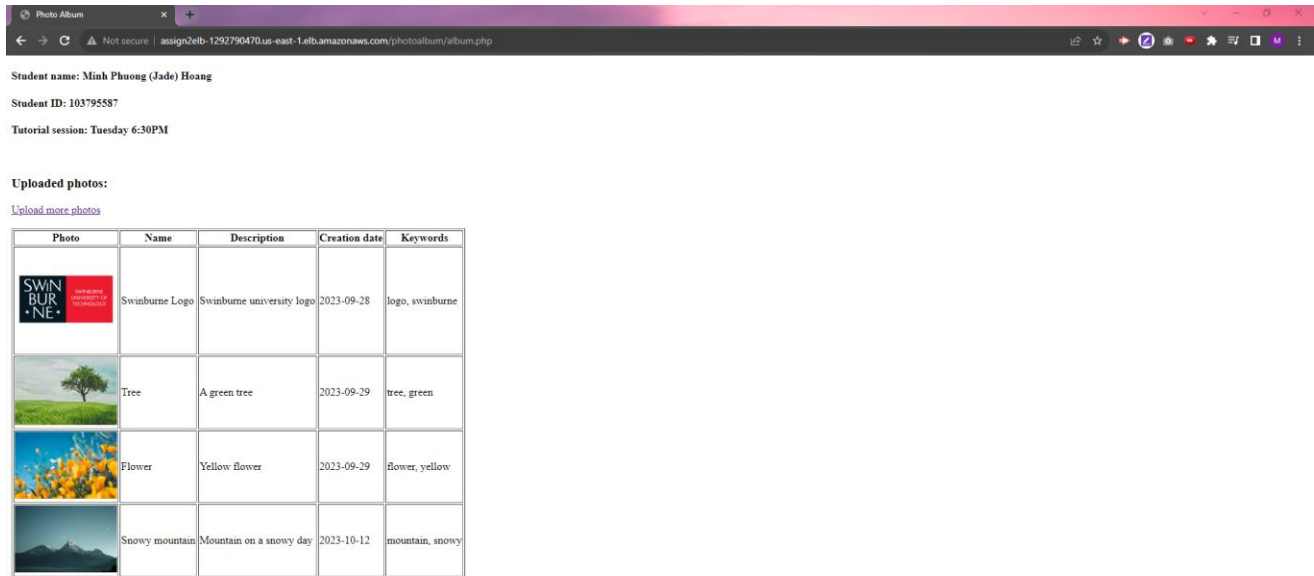


Photo Album

Student name: Minh Phuong (Jade) Hoang

Student ID: 103795587

Tutorial session: Tuesday 6:30PM

Uploaded photos:

[Upload more photos](#)





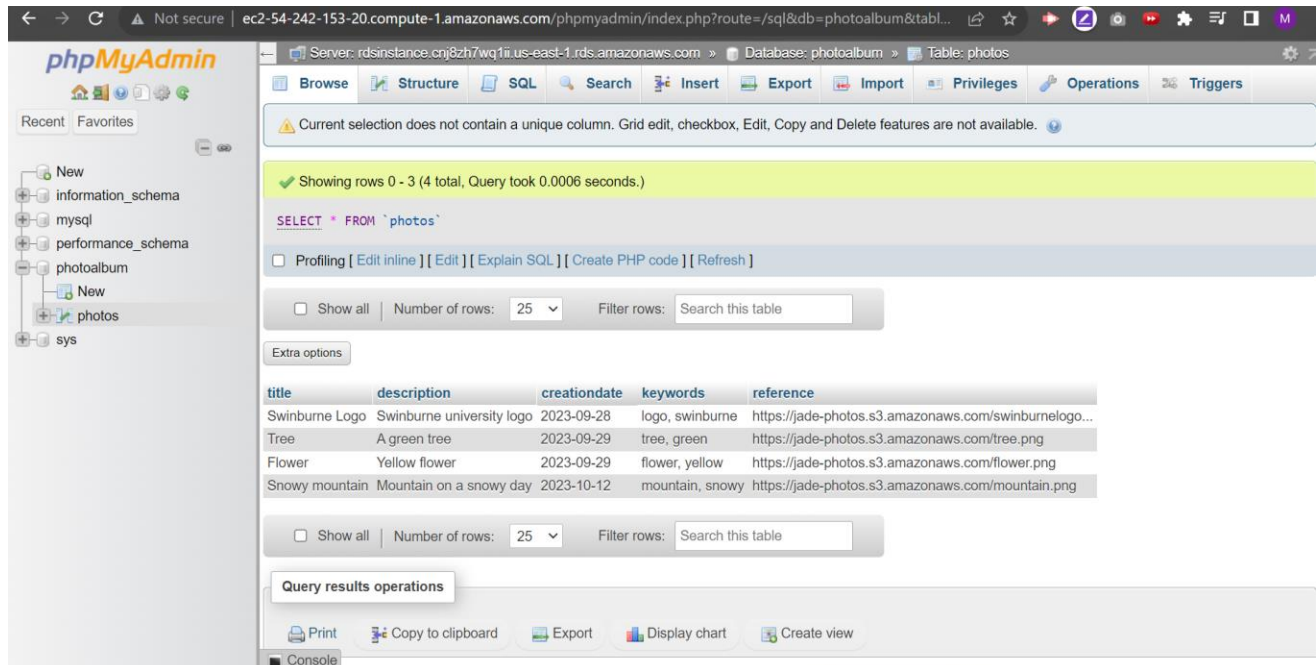
Photo	Name	Description	Creation date	Keywords
	Swinburne Logo	Swinburne university logo	2023-09-28	logo, swinburne
	Tree	A green tree	2023-09-29	tree, green
	Flower	Yellow flower	2023-09-29	flower, yellow
	Snowy mountain	Mountain on a snowy day	2023-10-12	mountain, snowy

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.



phpMyAdmin

Server: rdsinstance.cnj8zh7wq1.us-east-1.rds.amazonaws.com » Database: photoalbum » Table: photos

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 3 (4 total, Query took 0.0006 seconds.)

SELECT \* FROM 'photos'

Number of rows: 25 Filter rows: Search this table

title	description	creationdate	keywords	reference
Swinburne Logo	Swinburne university logo	2023-09-28	logo, swinburne	https://jade-photos.s3.amazonaws.com/swinburnelogo...
Tree	A green tree	2023-09-29	tree, green	https://jade-photos.s3.amazonaws.com/tree.png
Flower	Yellow flower	2023-09-29	flower, yellow	https://jade-photos.s3.amazonaws.com/flower.png
Snowy mountain	Mountain on a snowy day	2023-10-12	mountain, snowy	https://jade-photos.s3.amazonaws.com/mountain.png

Query results operations

Print Copy to clipboard Export Display chart Create view

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.

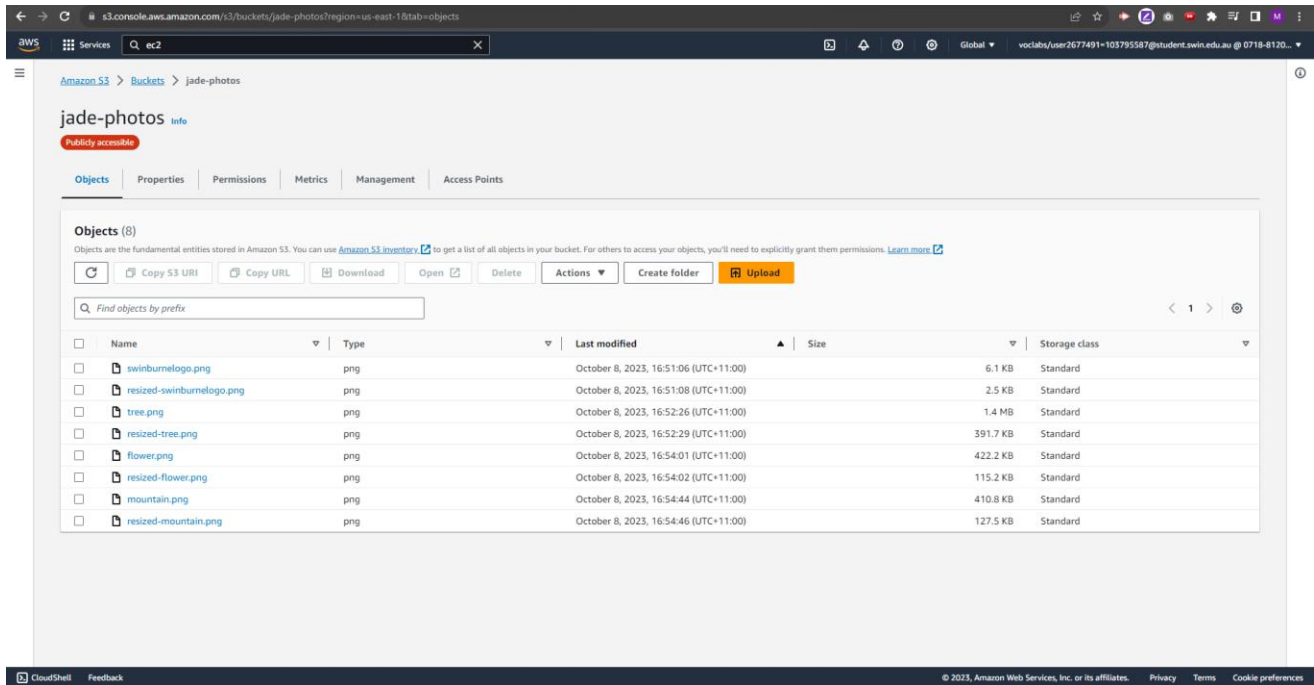


Figure 21 - S3 photos with their resized versions.