COS20019 - Cloud Computing Architecture

Assignment 2

Developing a highly available Photo Album website

Name: Pham Do Tien Phong
Student Id: 104189767

Link to the ELB album.php: http://elb-assignment2-434376215.us-east-1.elb.amazonaws.com/photoalbum/album.php
Link to the ELB photouploader.php: http://elb-assignment2-434376215.us-east-1.elb.amazonaws.com/photoalbum/photouploader.php

I. Introduction

Amazon Web Services (AWS) offers a wide range of services that can be used to create a reliable and highly accessible photo album website. By utilizing various AWS services for storage, hosting, databases, caching, load balancing, and monitoring, we can develop a photo album website with enhanced capabilities compared to previous assignments.

II. Implementation

1.Create VPC

First of all, I need to create a new VPC for this assignment
_ Name: PPhamVPC2
_Region: us-east-1
_AZ : I assigned subnets in 2 different AZ (Public subnet 1.2 and Private subnet 1.2 in us-east-1a & Public subnet 2.2 and Private subnet 2.2 in us-east-1b)
_Rout table :
. I create 2 route table beside the default route table
+ *RouteTable2-Public* is created to associate with 2 public subnets as well as routes to Internet Gateway (*InternetGateway2*)
+ *RouteTable2-Private* is created to associate with 2 private subnets as well as routes to NAT Gateway (*NAT gateway2*)

Figure 1: VPC resource map .

This is my resource map

2.Create Security Groups & Network ACLs

Secondly, I will create Security groups and Network ACLs to ensure security and accessibility to and from Web server

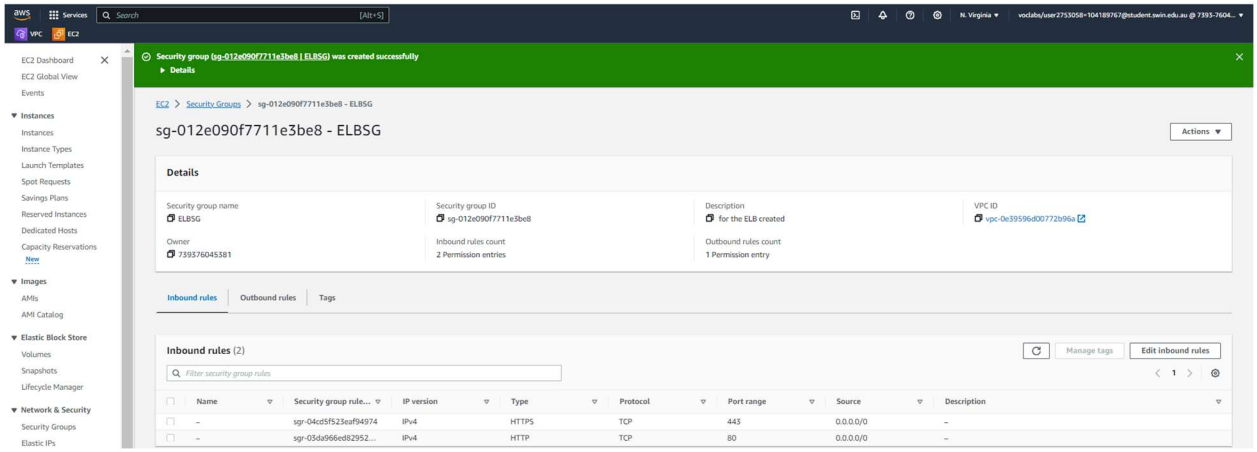


Figure 2: ELBSG security group

This is ELBSG allowing traffic for the web application from port 80 to port 443

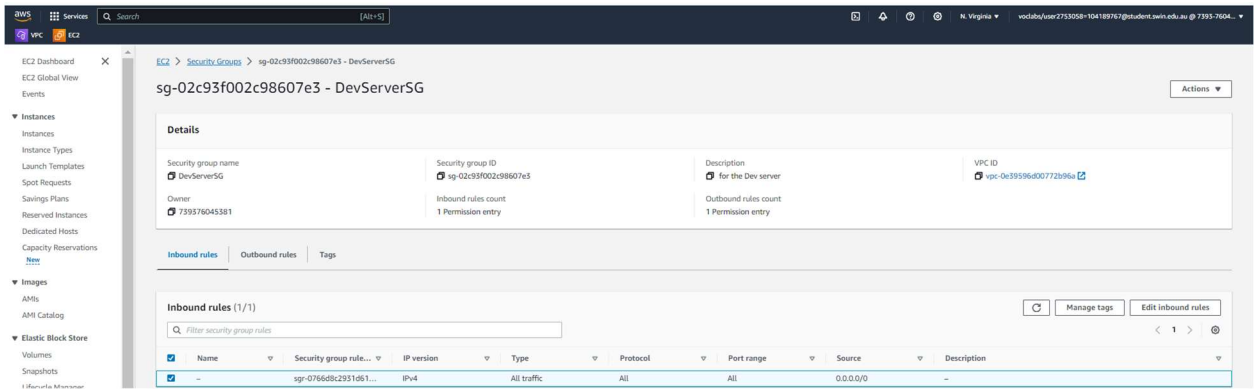


Figure 3: DevServerSG security group

This is DevServerSG can accept all traffic and SSH from port 22

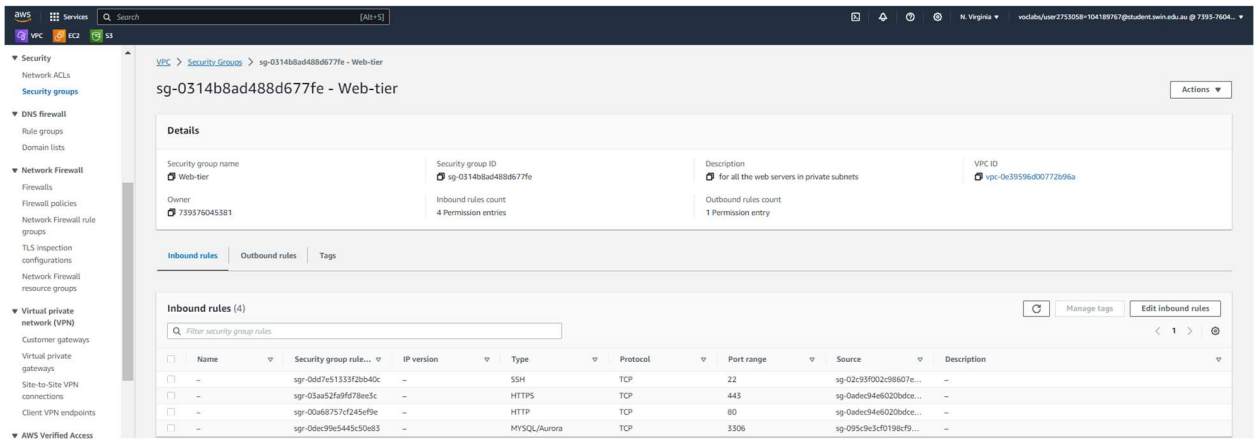


Figure 4: Web-tier security group

This is Web-tier (Security Group for Web server) allowing the traffic from the web application,SSH from port 22 and the DB security group.

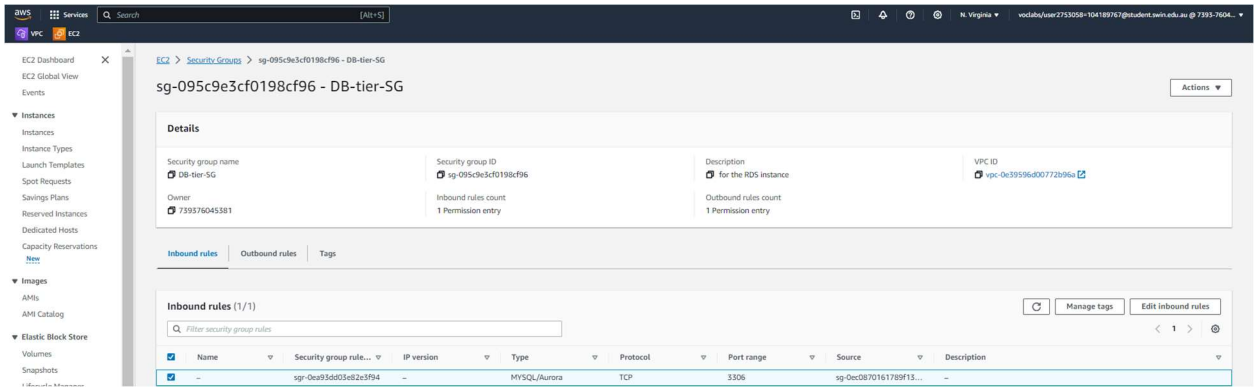


Figure 5: DB-tier-SG security group

This is DB-tier-SG, which just allows traffic from the port 3306 from the Web-tier security group above

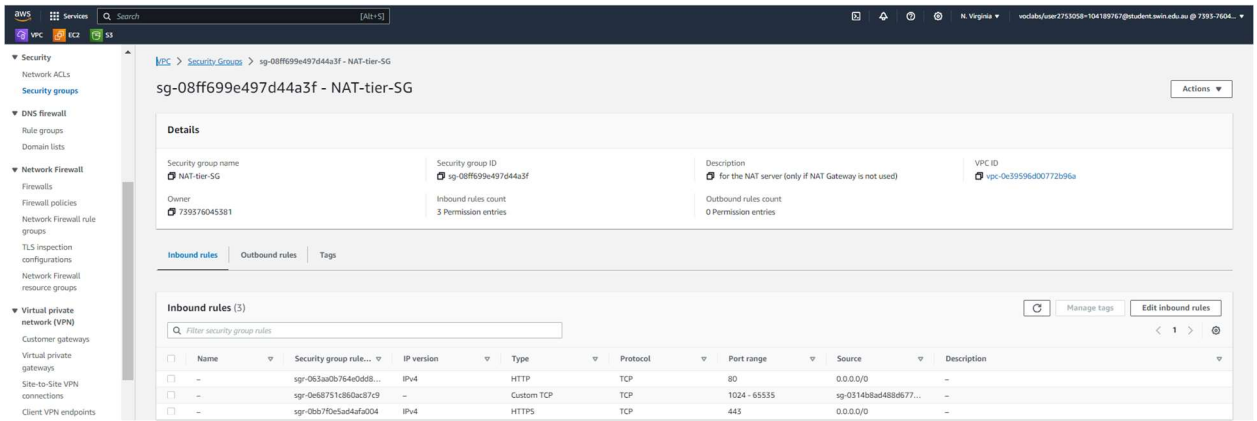
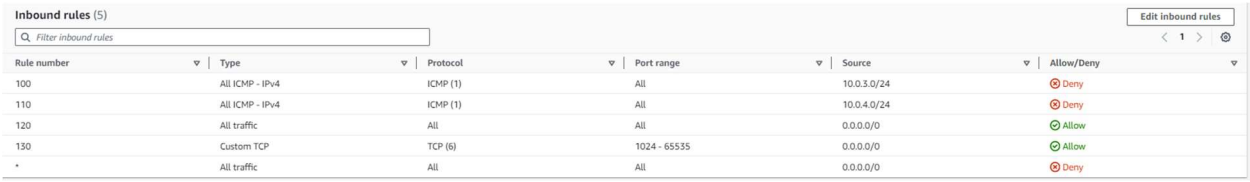


Figure 6: NAT-tier-SG security group

This is NAT-tier-SG that allow access the traffic from HTTP/HTTPs for the web application and traffic from the web server.

The next step is creating Network ACLs that is an additional security layer for the Web server. It will restrict DevServer from sending ICMP packet to the WebServer.

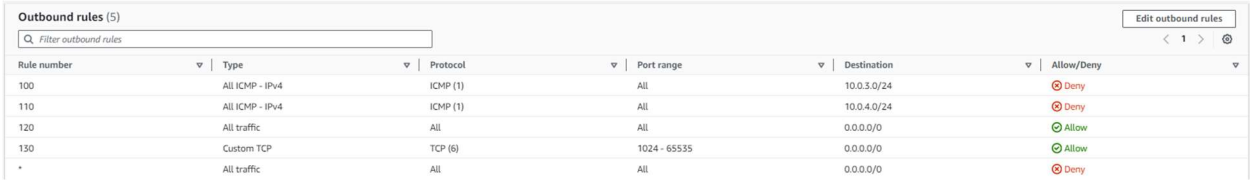


Inbound rules (5)

Edit inbound rules

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	All ICMP - IPv4	ICMP (1)	All	10.0.3.0/24	Deny
110	All ICMP - IPv4	ICMP (1)	All	10.0.4.0/24	Deny
120	All traffic	All	All	0.0.0.0/0	Allow
130	Custom TCP	TCP (6)	1024 - 65535	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Figure 7: Inbound rules



Outbound rules (5)

Edit outbound rules

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
100	All ICMP - IPv4	ICMP (1)	All	10.0.3.0/24	Deny
110	All ICMP - IPv4	ICMP (1)	All	10.0.4.0/24	Deny
120	All traffic	All	All	0.0.0.0/0	Allow
130	Custom TCP	TCP (6)	1024 - 65535	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Figure 8: Outbound rules

3.Create & Configure NAT instance & EC2 instance

The next step is create NAT and EC2 instance

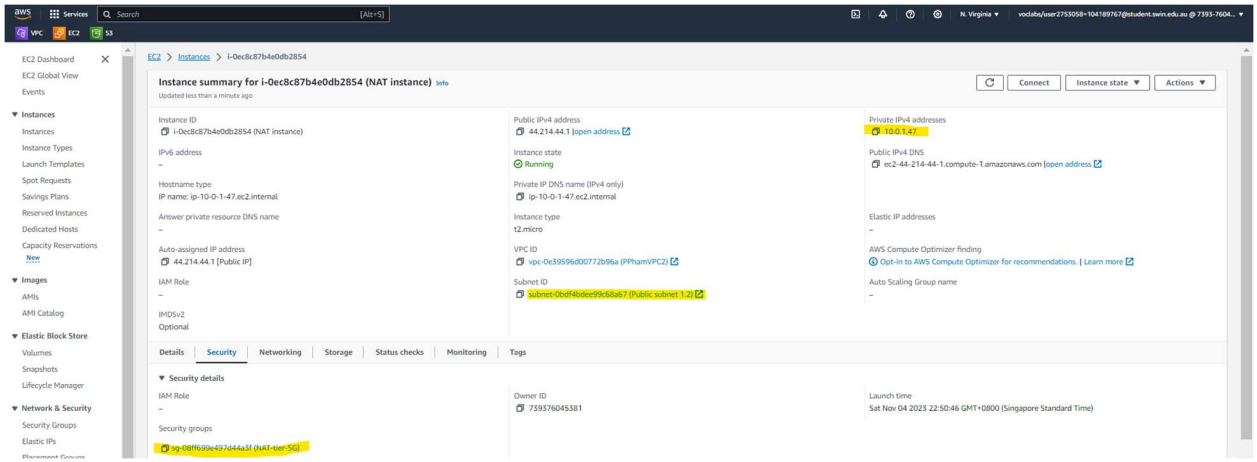


Figure 9: Information of NAT instance

I create NAT instance with Public Subnet 1.2 (*10.0.1.47*) and add security group into this instance (*NAT-tier-SG*)

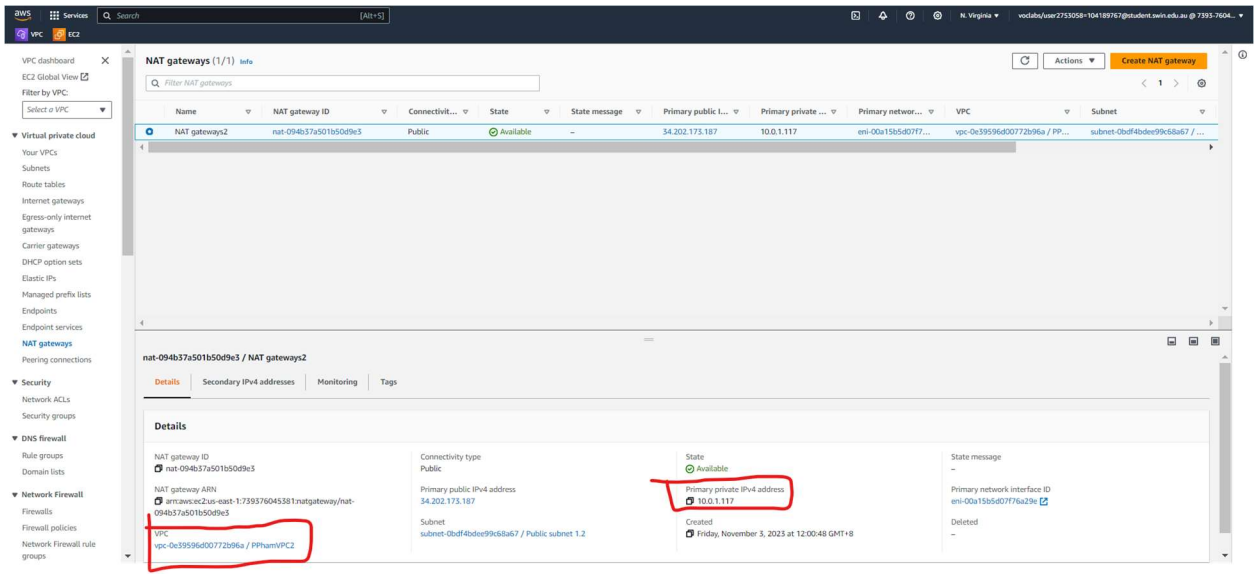


Figure 10: Information of NAT gateways

Configure NAT instance to NAT gateway , which will help private instances can communicate with public internet through NAT device

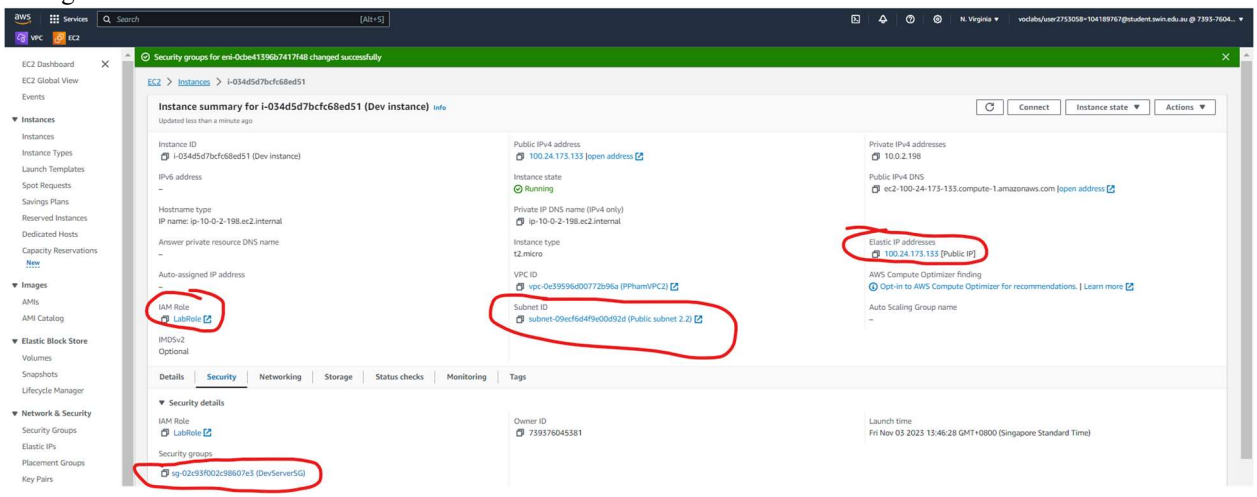


Figure 11: Information of Dev instance

I create Dev instance with public subnet 2.2 (*10.0.2.198*) , attach EIP (*100.24.173.133*) for having unchanged public IP addrress and assign IAM role (*LabRole)*

4.RDS Database

The following stage is that create a RDS database with the same steps I implemented in the assignment 1b

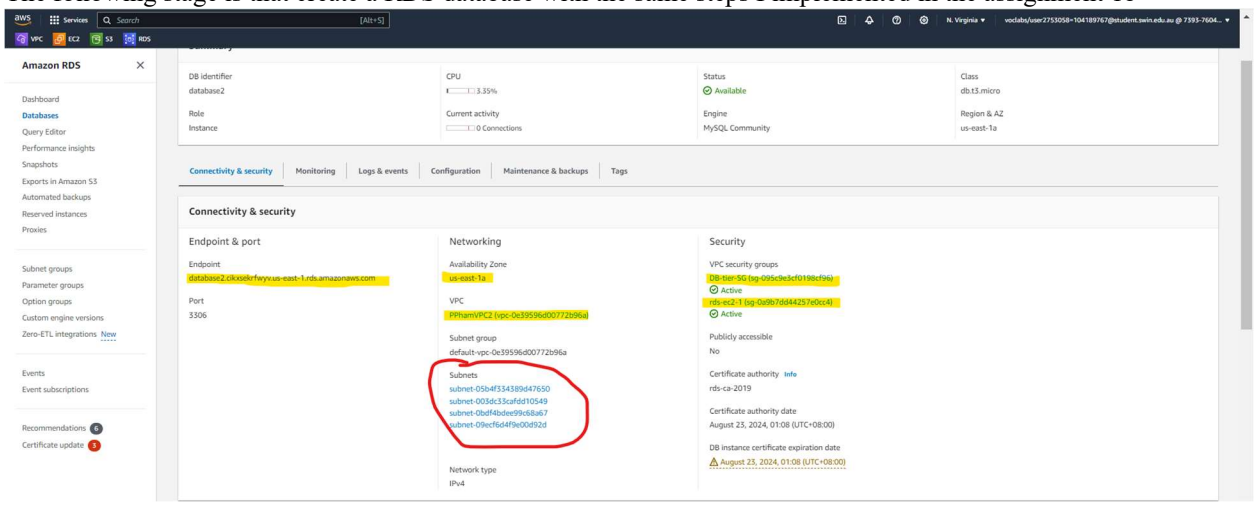


Figure 12: Information of database

I create the database (*databased2*) and add security group for them (*DB-tier-SG*). There are 4 subnets connected to my database (2 public subnets and 2 private subnets)

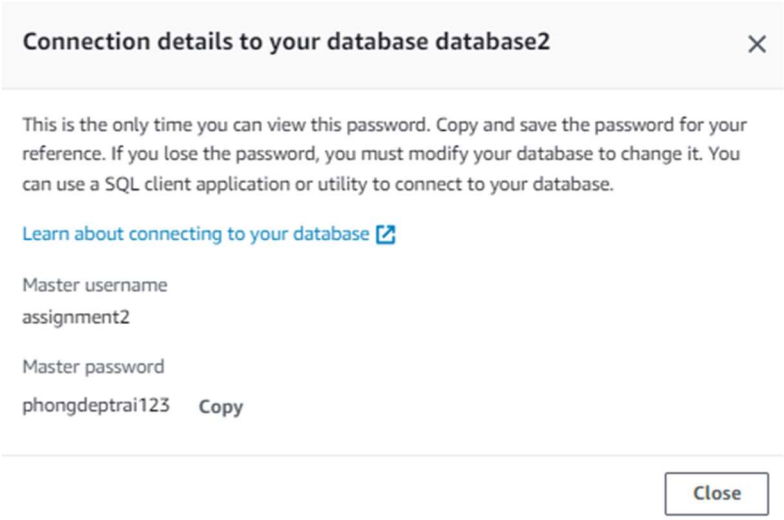
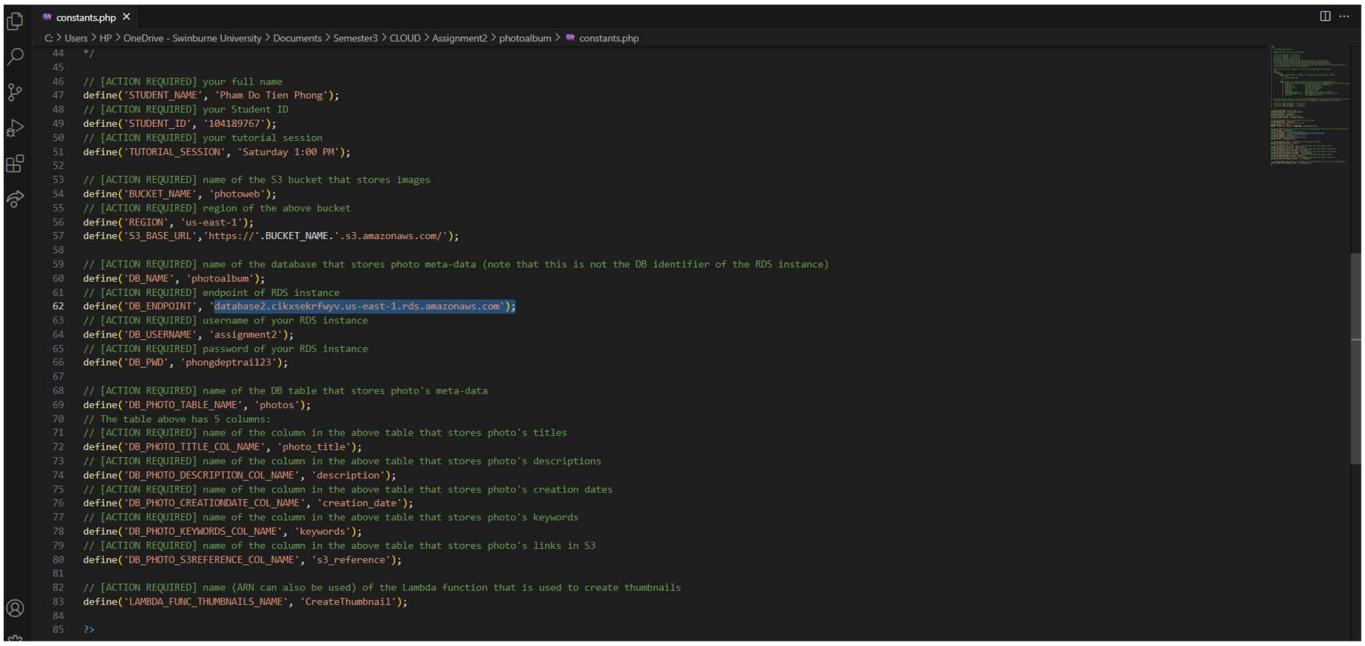


Figure 13: Connection details

_ The username is : assignment2

_ The password is : phongdeptrai123



```php
*/

// [ACTION REQUIRED] your full name
define('STUDENT_NAME', 'Pham Do Tien Phong');
// [ACTION REQUIRED] your Student ID
define('STUDENT_ID', '104189767');
// [ACTION REQUIRED] your tutorial session
define('TUTORIAL_SESSION', 'Saturday 1:00 PM');

// [ACTION REQUIRED] name of the S3 bucket that stores images
define('BUCKET_NAME', 'photoweb');
// [ACTION REQUIRED] region of the above bucket
define('REGION', 'us-east-1');
define('S3_BASE_URL','https://'.BUCKET_NAME.'.s3.amazonaws.com/');

// [ACTION REQUIRED] name of the database that stores photo meta-data (note that this is not the DB identifier of the RDS instance)
define('DB_NAME', 'photoalbum');
// [ACTION REQUIRED] endpoint of RDS instance
define('DB_ENDPOINT', 'database2.cikxsekrfwyv.us-east-1.rds.amazonaws.com');
// [ACTION REQUIRED] username of your RDS instance
define('DB_USERNAME', 'assignment2');
// [ACTION REQUIRED] password of your RDS instance
define('DB_PWD', 'phongdeptrai123');

// [ACTION REQUIRED] name of the DB table that stores photo's meta-data
define('DB_PHOTO_TABLE_NAME', 'photos');
// The table above has 5 columns:
// [ACTION REQUIRED] name of the column in the above table that stores photo's titles
define('DB_PHOTO_TITLE_COL_NAME', 'photo_title');
// [ACTION REQUIRED] name of the column in the above table that stores photo's descriptions
define('DB_PHOTO_DESCRIPTION_COL_NAME', 'description');
// [ACTION REQUIRED] name of the column in the above table that stores photo's creation dates
define('DB_PHOTO_CREATIONDATE_COL_NAME', 'creation_date');
// [ACTION REQUIRED] name of the column in the above table that stores photo's keywords
define('DB_PHOTO_KEYWORDS_COL_NAME', 'keywords');
// [ACTION REQUIRED] name of the column in the above table that stores photo's links in S3
define('DB_PHOTO_S3REFERENCE_COL_NAME', 's3_reference');

// [ACTION REQUIRED] name (ARN can also be used) of the Lambda function that is used to create thumbnails
define('LAMBDA_FUNC_THUMBNAILS_NAME', 'CreateThumbnail');

?>
```

Figure 14: Code of constant.php

This is the code I used in constant.php , the most important change is the enpoint of RDS instance to the endpoint of my database (*database2*)

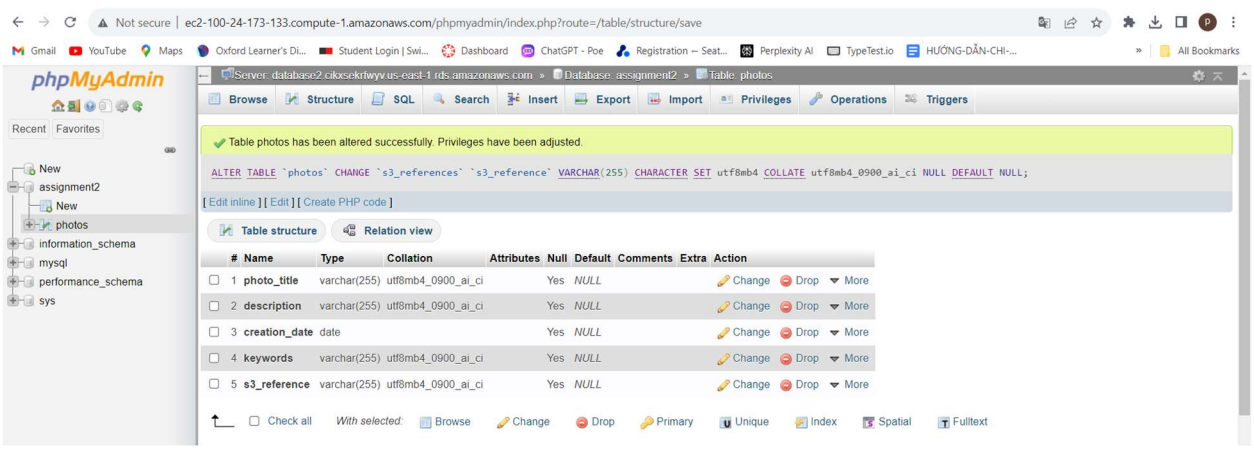


Figure 15: Create table in MySQL

I create MySQL table with this code

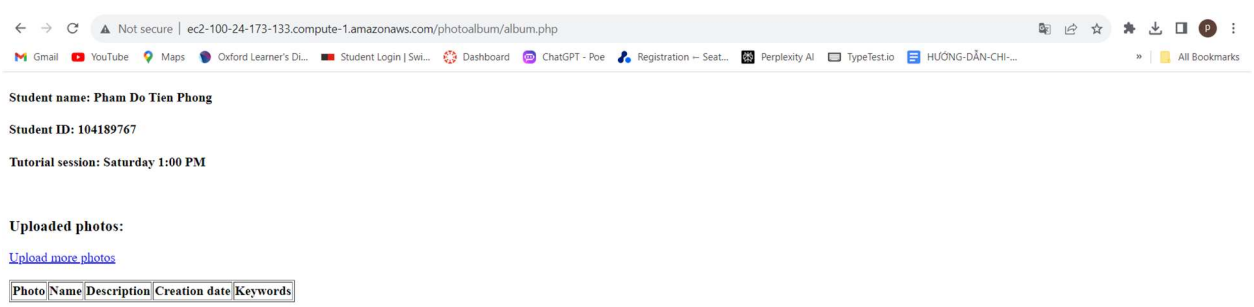


Figure 16: Default display of album.php

This is the default display of album.php

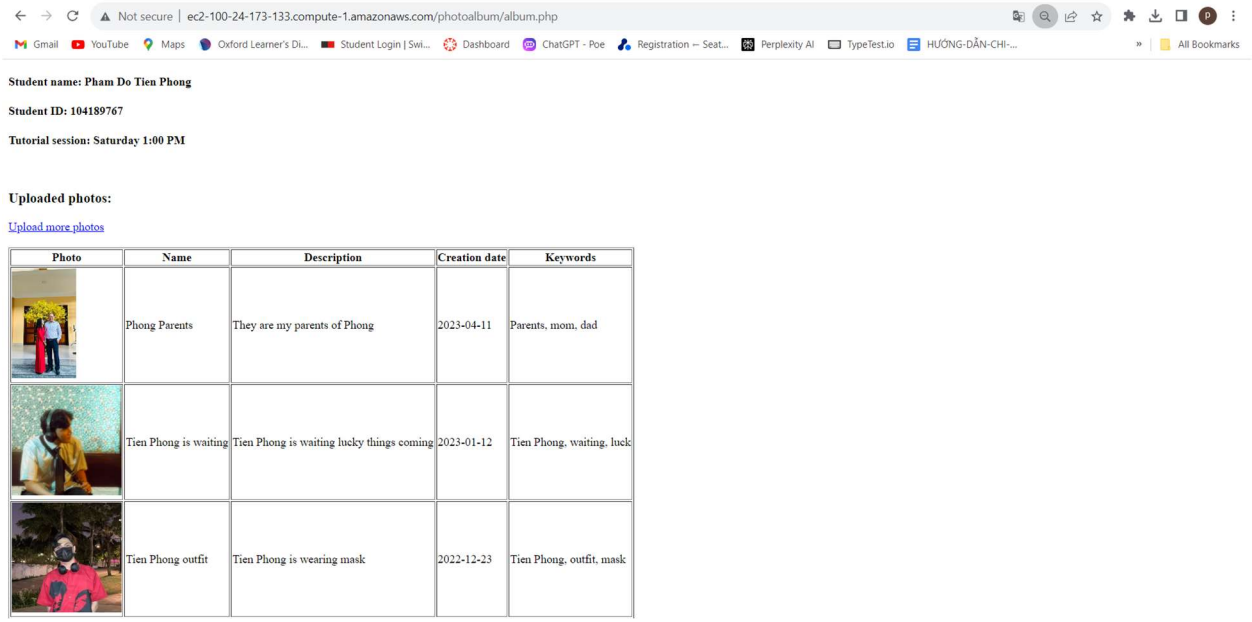


Figure 17: display of album.php after adding

and then , I add some photos for this album

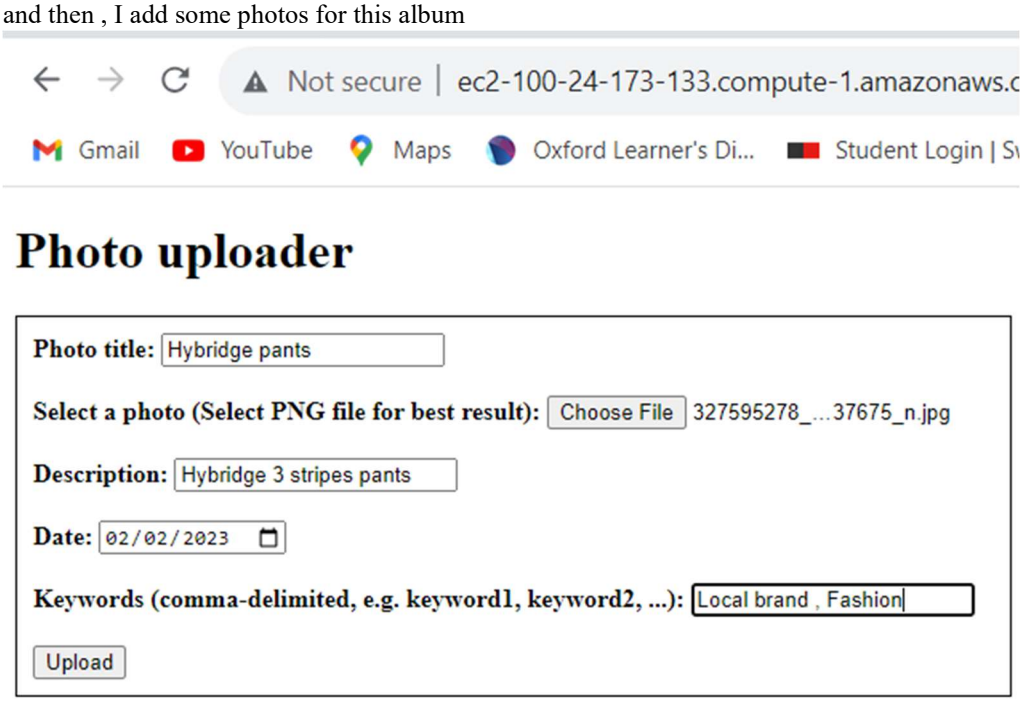
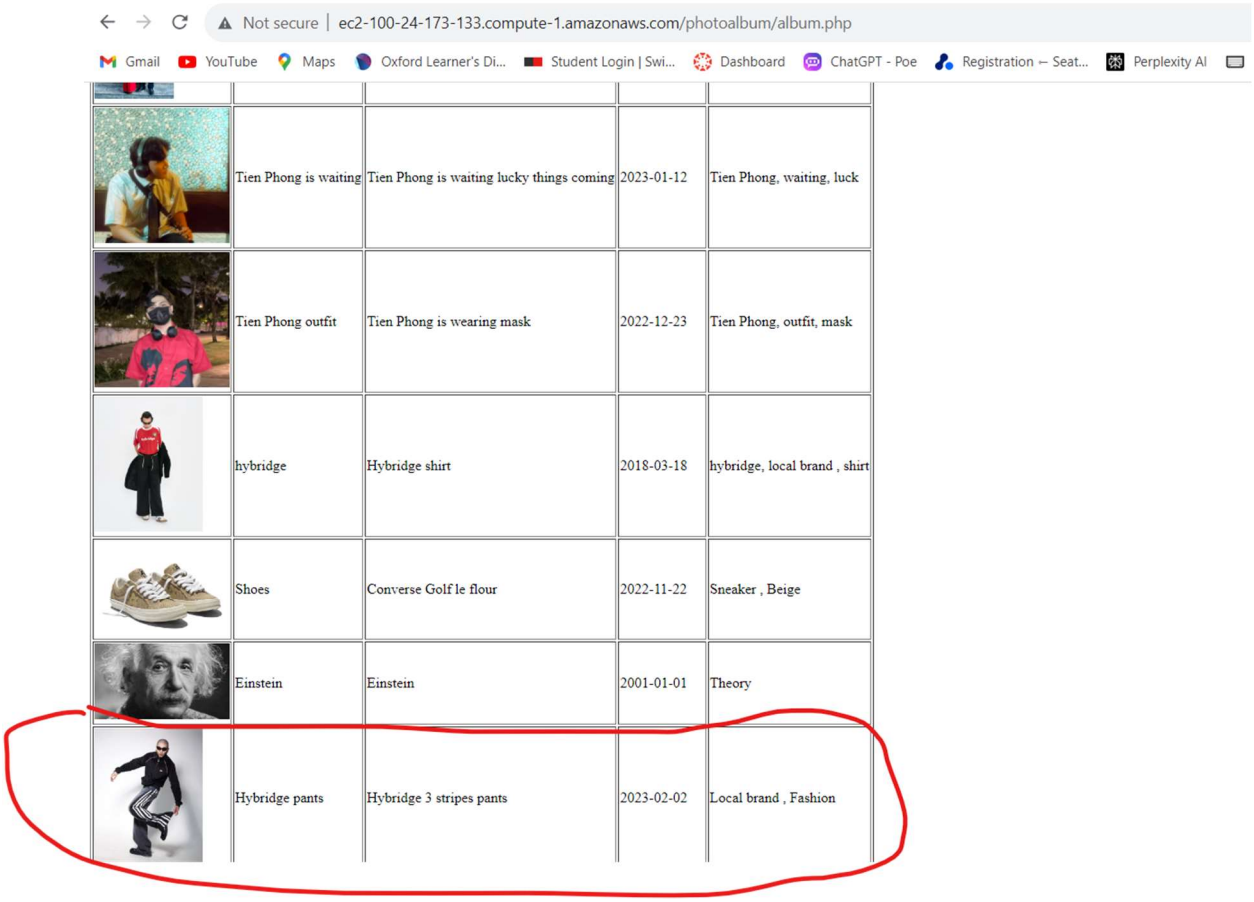


Figure 18: display of photouploader.php

I upload photo by photouploader.php



	Tien Phong is waiting	Tien Phong is waiting lucky things coming	2023-01-12	Tien Phong, waiting, luck
	Tien Phong outfit	Tien Phong is wearing mask	2022-12-23	Tien Phong, outfit, mask
	hybridge	Hybridge shirt	2018-03-18	hybridge, local brand , shirt
	Shoes	Converse Golf le flour	2022-11-22	Sneaker , Beige
	Einstein	Einstein	2001-01-01	Theory
	Hybridge pants	Hybridge 3 stripes pants	2023-02-02	Local brand , Fashion

Figure 19: photo uploaded

This photo is uploaded in album.php

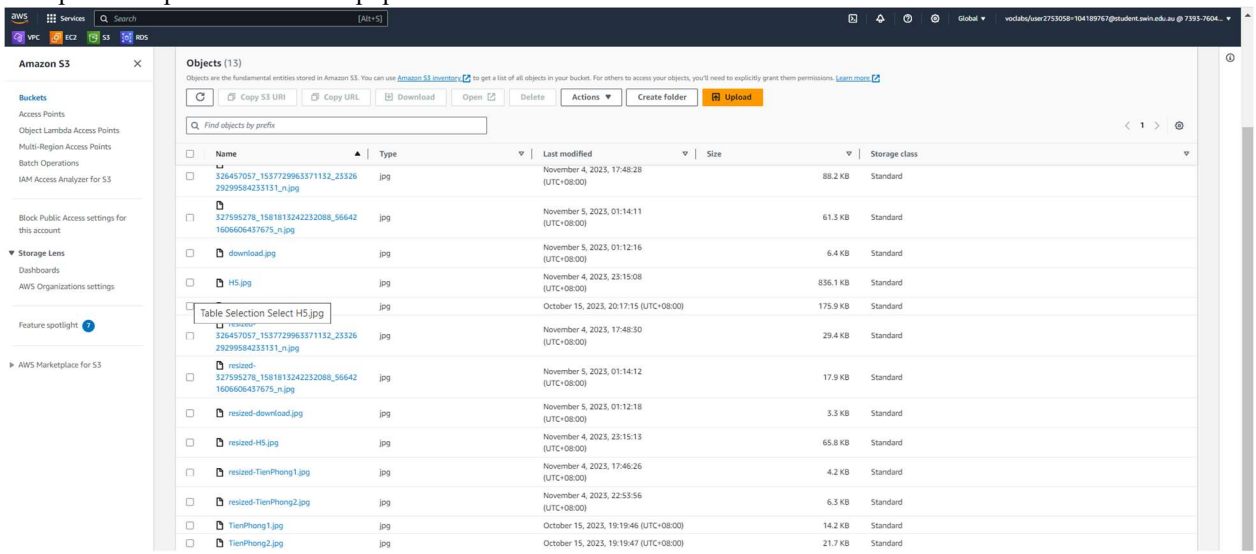


Figure 20: S3 bucket display

And then , the S3 bucket will have these images files

5.Create Load Balacing

and then, I have to create Elastic Load Balancing

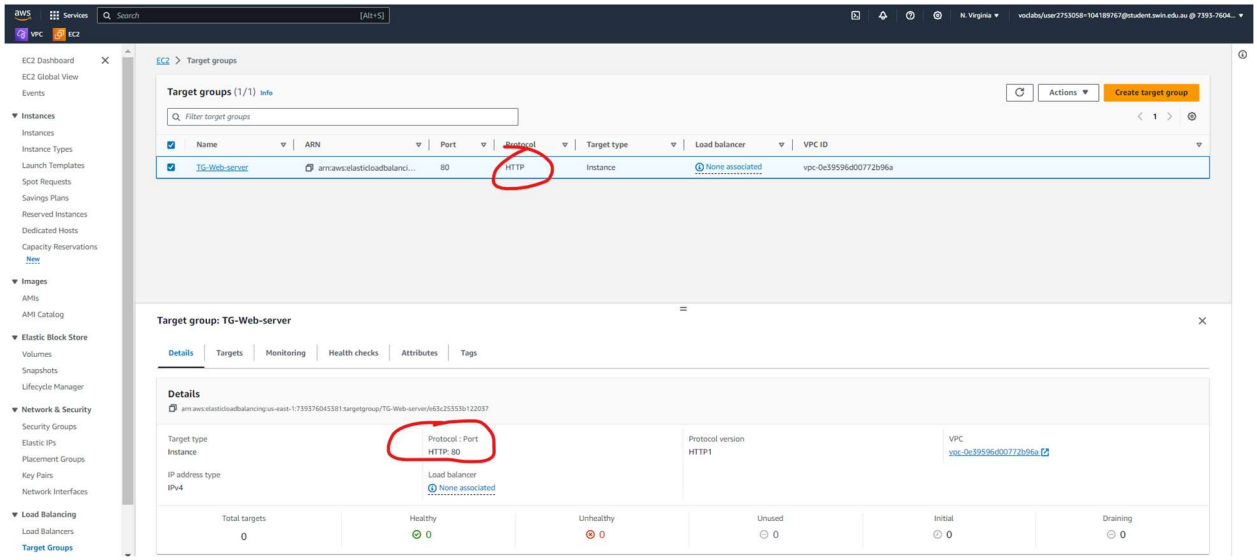


Figure 21: Information of target groups

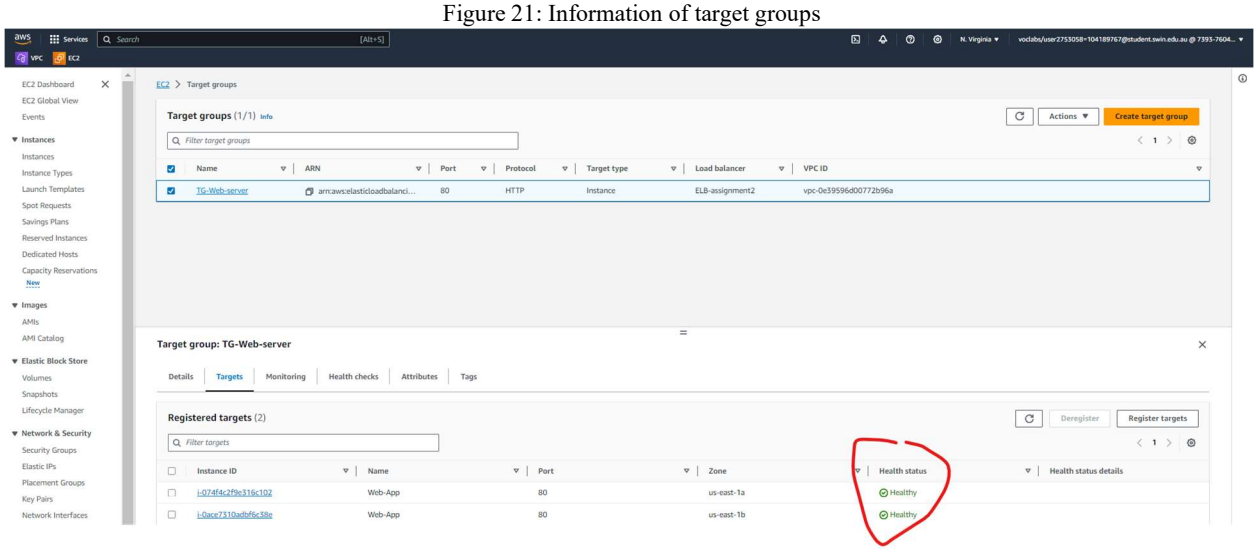


Figure 22: Target groups in Healthy status

I create target group with the path */photoalbum/album.php* and HTTP protocol with port 80, making sure all of registered targets in Healthy status

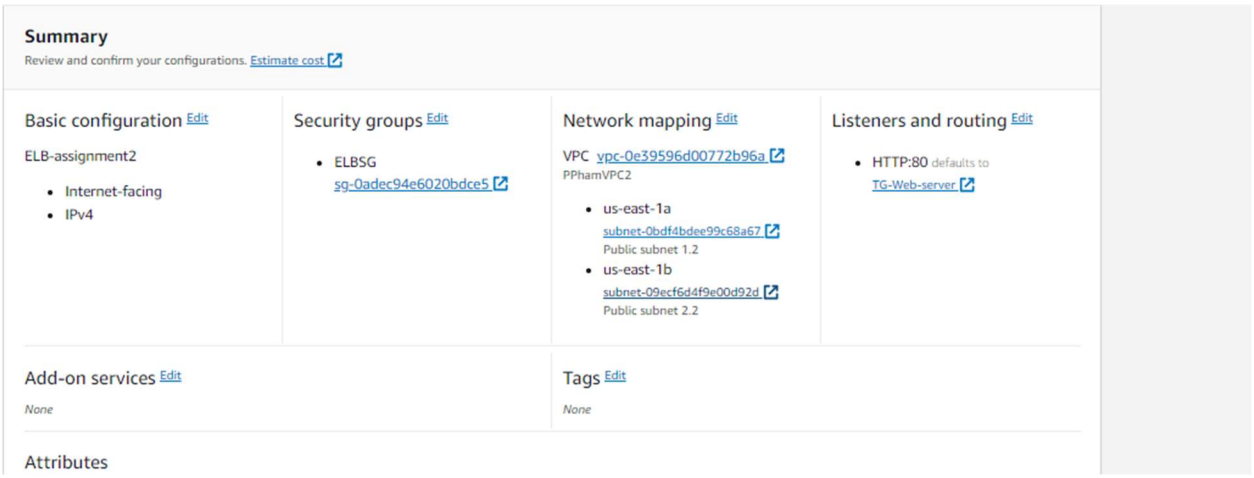


Figure 23: Information of Load Balancers

I create Load balancers in public subnets.

6.Create Auto Scaling Group

Creating ASG is the next step

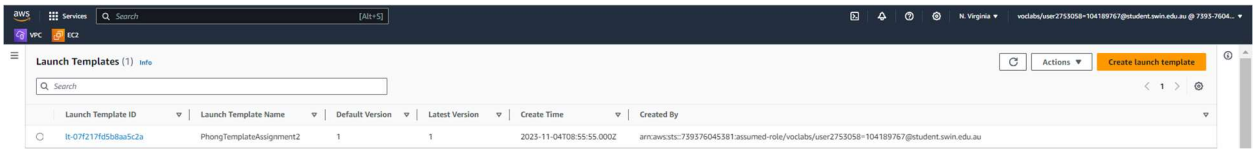


Figure 24: Template

I create a template to use for Auto scalling group (*PhongTemplateAssignment2*)

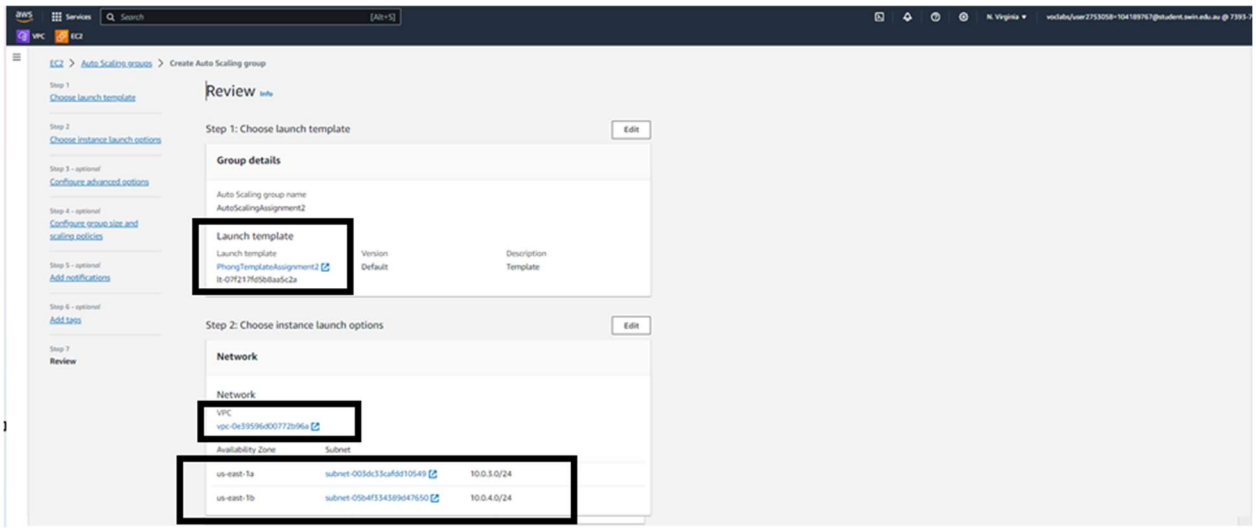


Figure 25: Information of Step 1 & Step 2

Step 1 : I launch template I create above

Step 2 : I choose the VPC I using in this assignment (*PPhamVPC2*) and the 2 Availability Zones storing 2 Private subnets

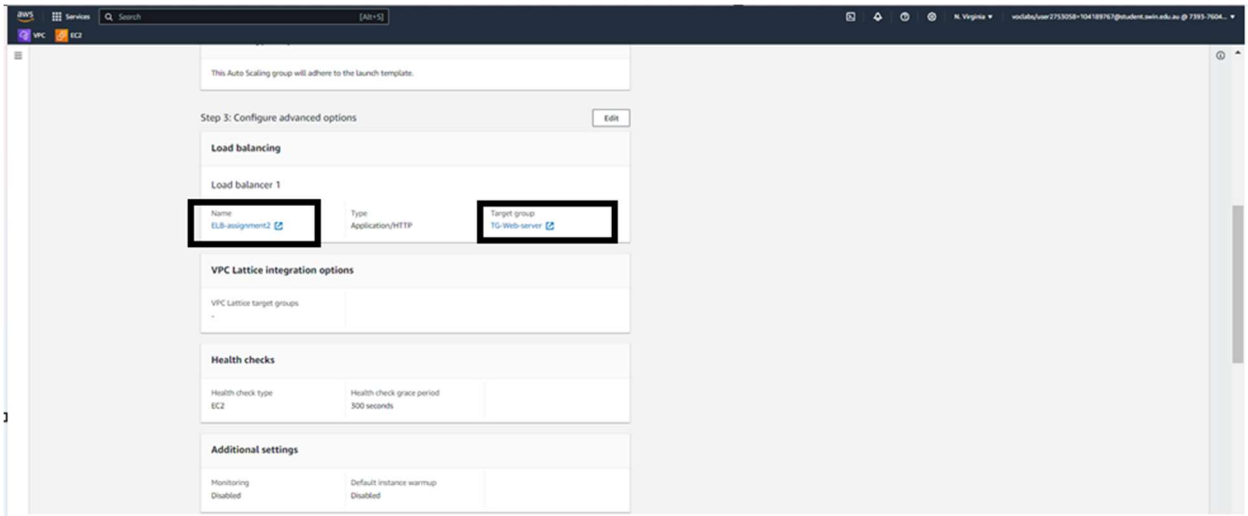


Figure 26: Information of Step 3

Step 3 : Using Load Balancing I created in the Step 5 (*ELB-assignment2*) and the target group *(TG-Web-server*)

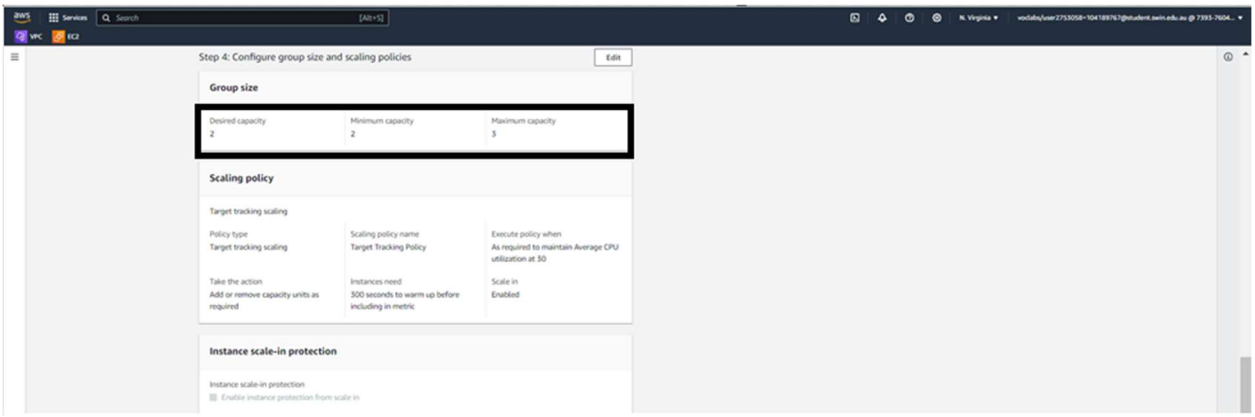


Figure 27: Information of Step 4

Step 4 : I set group size for and scaling policies

. Desired capacity : 2

. Maximum quality : 2

. Maximum capacity : 3

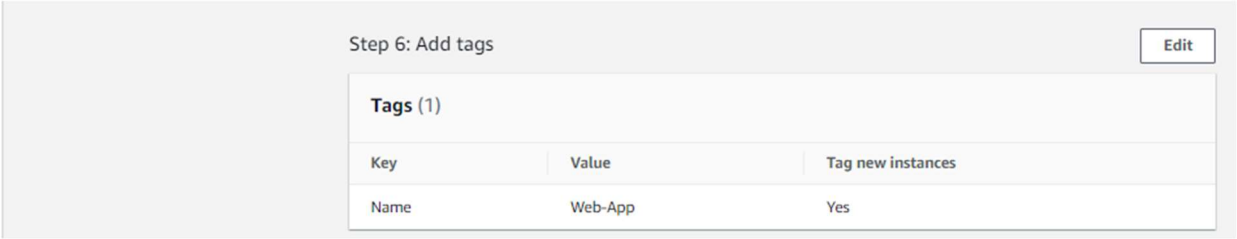


Figure 28: Information of Step 6

Step 5 (additional)

Step 6: Add tags with (*Key : Name , Value : Web – App*)

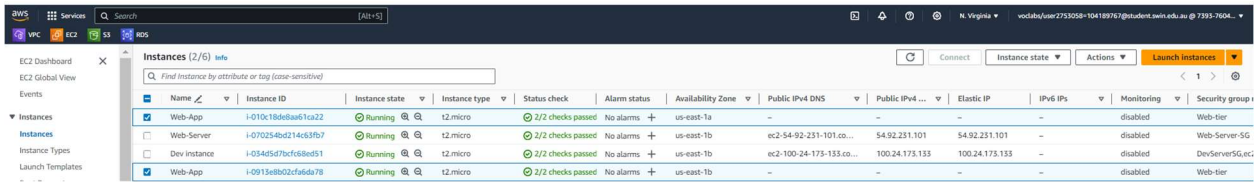


Figure 29: Instances created by ASG

There are 2 Web-App created by Auto Scaling Group

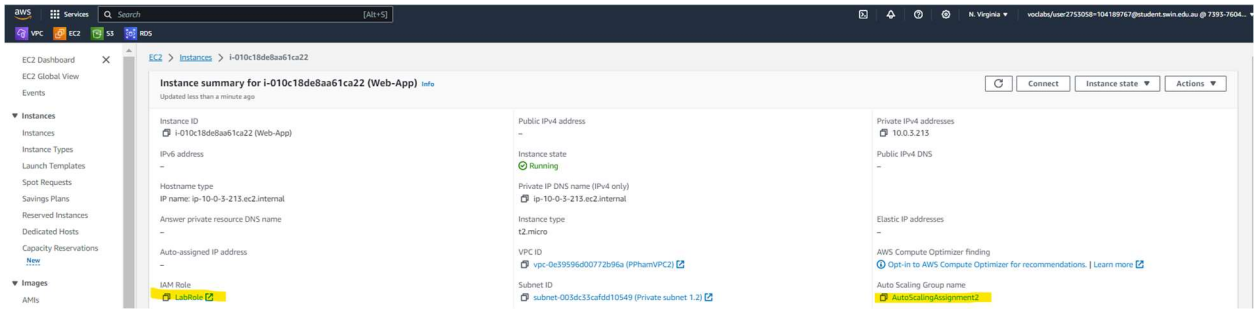


Figure 30: Information of instances created by ASG

Web-App is the instance created by Auto Scaling Group (*AutoScalingAssignment2*) with IAM Role (*LabRole*)

7.S3 bucket

In this stage, I used the old bucket instead of creating new bucket

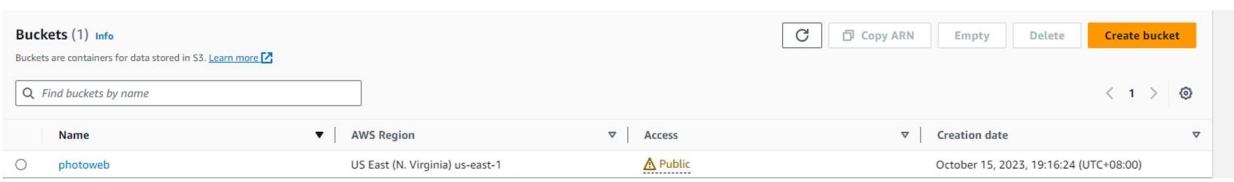


Figure 31: S3 bucket

photoweb is my bucket

```
1  {
2      "Version": "2012-10-17",
3      "Id": "assignmentbucket2",
4      "Statement": [
5          {
6              "Sid": "PublicRead",
7              "Effect": "Allow",
8              "Principal": "*",
9              "Action": [
10                 "s3:GetObject",
11                 "s3:GetBucketLocation",
12                 "s3:ListBucket",
13                 "s3:PutObject"
14             ],
15             "Resource": [
16                 "arn:aws:s3:::photoweb/*",
17                 "arn:aws:s3:::photoweb"
18             ],
19             "Condition": {
20                 "StringLike": {
21                     "aws:Referer": [
22                         "http://elb-assignment2-434376215.us-east-1.elb.amazonaws.com/*",
23                         "http://ec2-100-24-173-133.compute-1.amazonaws.com/*"
24                     ]
25                 }
26             }
```

Actions : access , get, put, list

resource name : photoweb

Allow access : ELB and EC2 DNS

Figure 32: S3 bucket policy

The thing I change is the S3 bucket policy , in this assignment , S3 bucket will not allow to access from public ; however, It only allows access from and by Application Load Balancer (ALB), it allows only the ELB to access, get, put, list object.

8.Lambda Function

The final stage is that create Lambda function

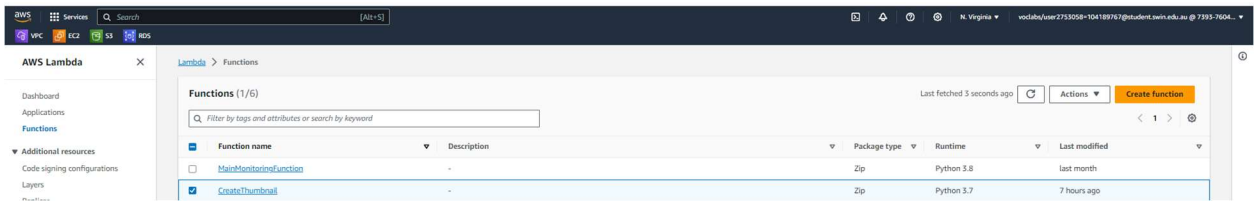


Figure 33: Create Lambda Function

I create a Lambda function (*CreateThumbnail*)

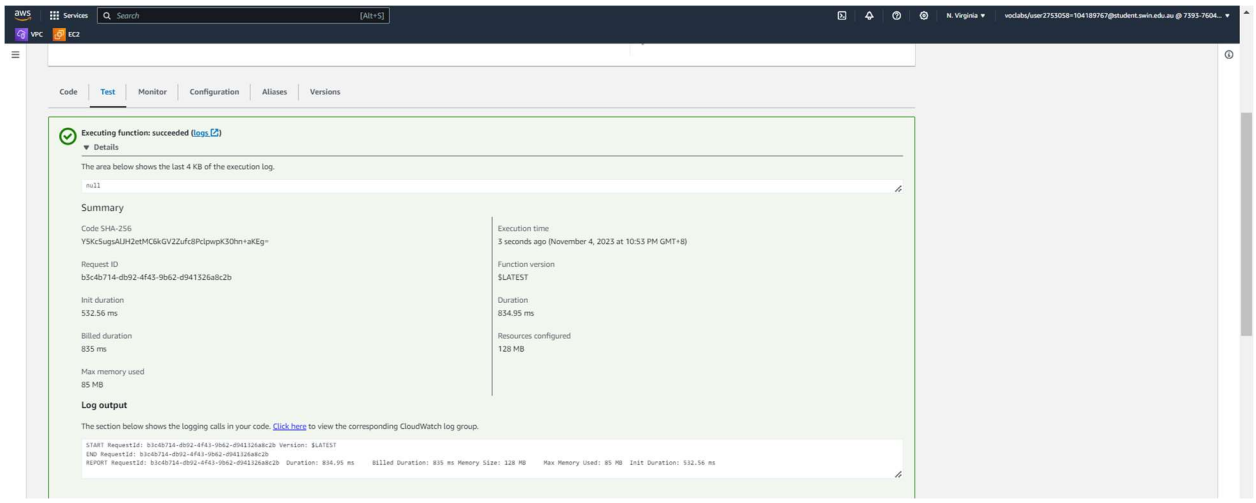


Figure 34: Test Lambda Function successfully

The Lamda function test case is successful

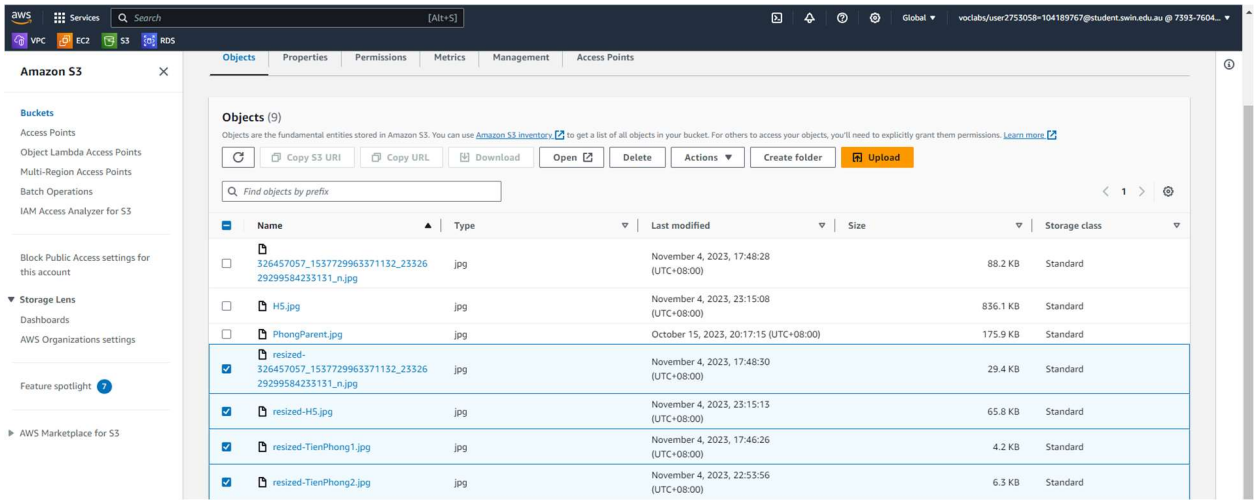


Figure 35: Resized photo created

And after a successfully testcase, It will resize uploaded picture to S3.

9.Testing

The PhotoAlbum website is accessible through the load balancer only

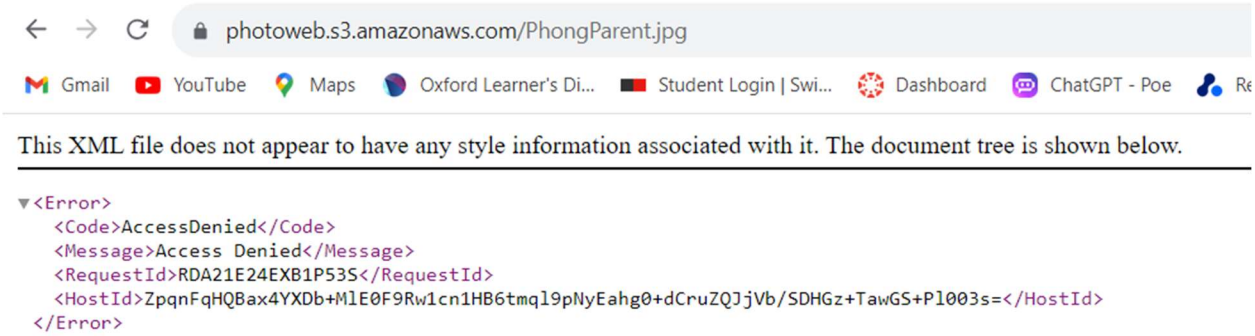


Figure 36: Test accessible through ELB only

I can not access PhotoAlbum website with other links different from ELB

Test the Network ACL bidierectional functionality by sending ICMP traffic between the web servers and Dev server.

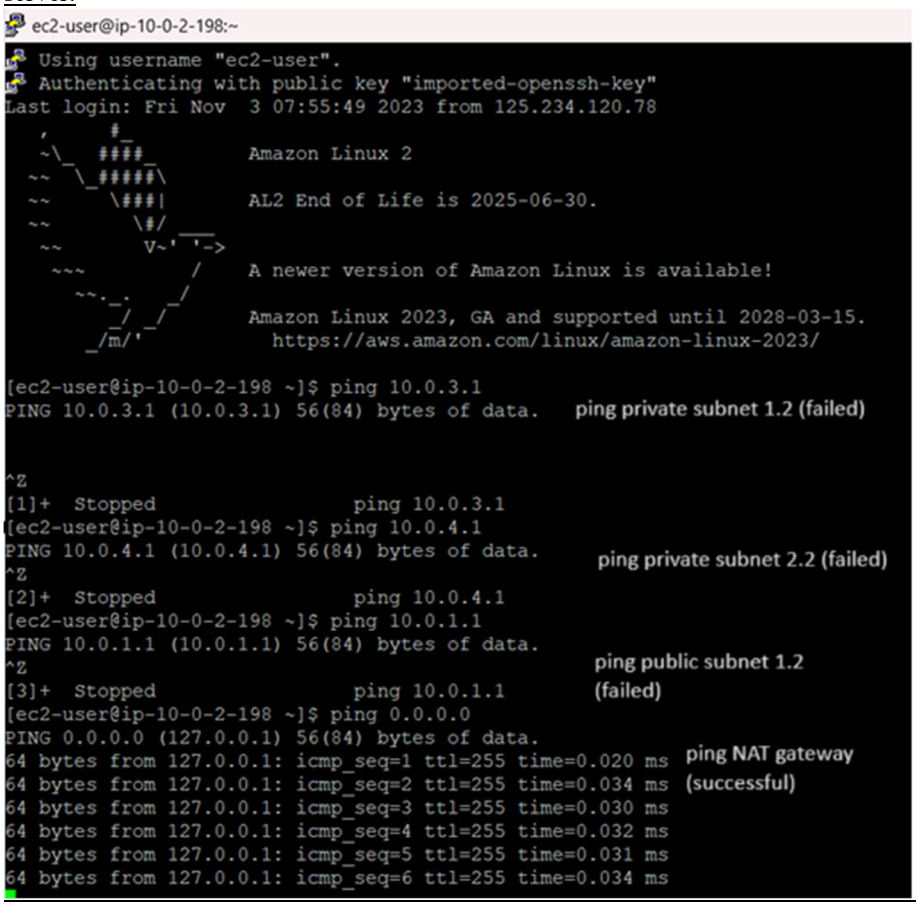


Figure 37: Test sending ICMP

I can ping the NAT gateway, which located at 0.0.0.0, but the ICMP sent to private subnet 1.2(10.0.3.1) and private subnet 2.2(10.0.4.1) is not reachable. Therefore, our Network ACL is working.