



Cloud Computing

**Assignment 2
Individual Report**

**Tutorial Class
Class 1 – 14 6:30PM – 8:30PM**

Xuan Tuan Minh Nguyen
103819212

Table Of Contents

TABLE OF CONTENTS	2
INTRODUCTION.....	3
1. <i>INTRODUCTION.....</i>	3
2. <i>REPORT'S OBJECTIVE.....</i>	3
THE IMPLEMENTATION OF THE PROJECT	3
1. <i>INFRASTRUCTURE DIAGRAM</i>	3
2. <i>THE IMPLEMENTATION</i>	4
2.1.1. <i>The VPC and NAT Gateway.....</i>	4
2.1.2. <i>S3 photo storage.....</i>	5
2.1.3. <i>Load balancing.....</i>	6
2.1.4. <i>Auto scaling.....</i>	7
2.1.5. <i>EC2 instance</i>	8
2.1.6. <i>CreateThumbnail Lambda function</i>	8
2.1.7. <i>Database with RDS.....</i>	9
2.1.8. <i>Security group and Network ACLs.....</i>	11
3. <i>RESULT.....</i>	12
LINK TO THE WEBSITE	14

Introduction

1. Introduction

Beside from offering a vast amount of spectacular services in order to provide a high capacity storing and transformative self-hosting virtual machines on a safe networking environment. Amazon Web Service, or AWS, also introduces a comprehensive range of services, such as Elastic Load Balancer (ELB), Auto Scaling Group (ASG), Lambda and Amazon Machine Image (AMI). That serves the purpose of automation and leverage to build a robust and highly scalable server.

2. Report's objective

This project will comprise the practice of integrating Elastic Load Balancer (ELB), Auto Scaling Group (ASG), Lambda function and Amazon Machine Image (AMI) to the EC2, VPC, S3 and RDS that has been demonstrated on the Assignment 1B to make the photo album program more adaptive and high accessing capacity.

The Implementation of the project

1. Infrastructure diagram

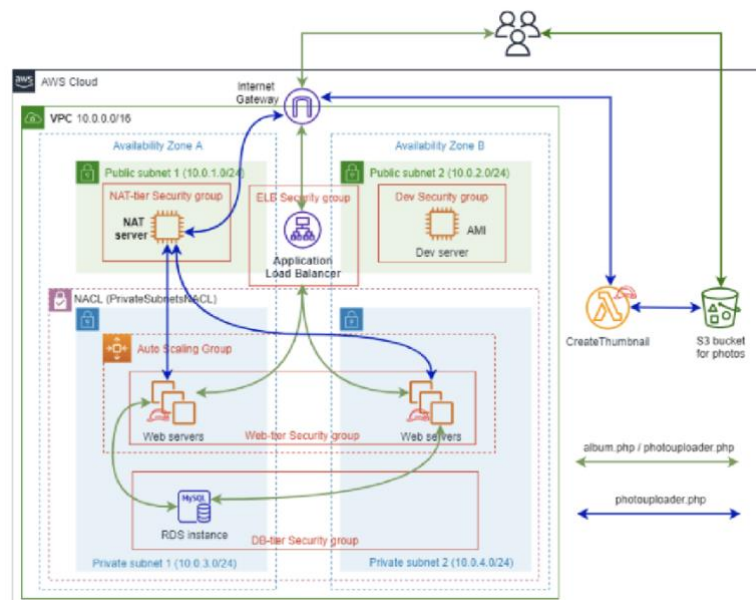


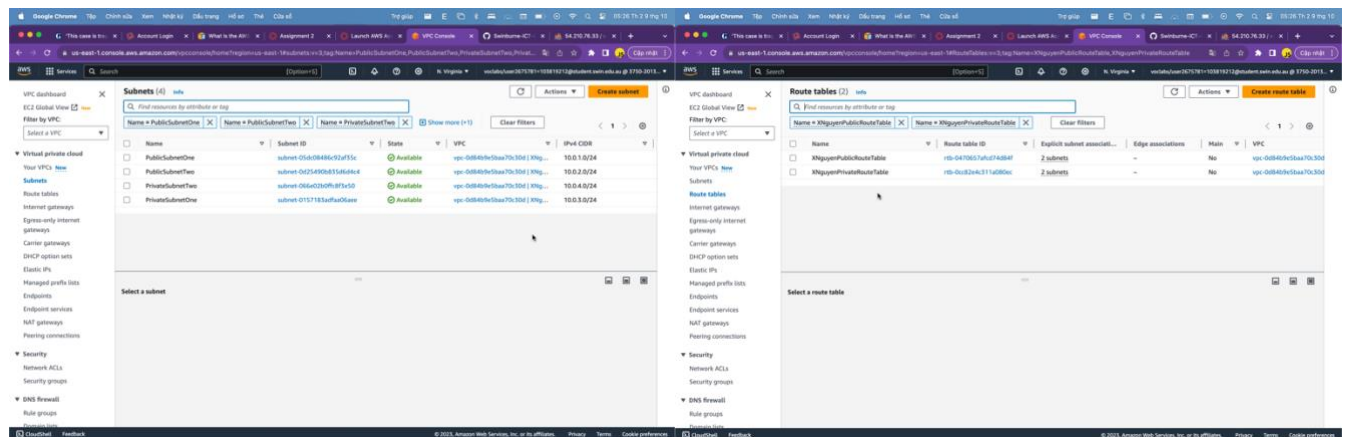
Figure 1 - Architecture diagram

The diagram above demonstrates the required infrastructures that are needed to create a robust, scalable and reliable network environment for the system. In general, this system is relatively the same as the previous assignment, which comprises of one VPC (10.0.0.0/16) and four subnets (two public subnets and two private subnets). However, the key difference in this assignment is the use of Network Address Translation (NAT), Elastic Load Balancer (ELB), Auto Scaling Group (ASG), Amazon Machine Image (AMI) and Lambda function that helps to create a high capacity and automatic system.

2. The implementation

2.1.1. The VPC and NAT Gateway

The VPC configuration is not as different as the Assignment 1B, which has four subnets on the VPC of 10.0.0.0/16 in two different Availability Zones (AZs). The public subnets will be direct to the public route table that is connect to the internet gateway. However, the private subnets will be reside on the private route table that routes to the NAT gateway, which helps to create a more safer network but still able to access to the components inside the private subnets.



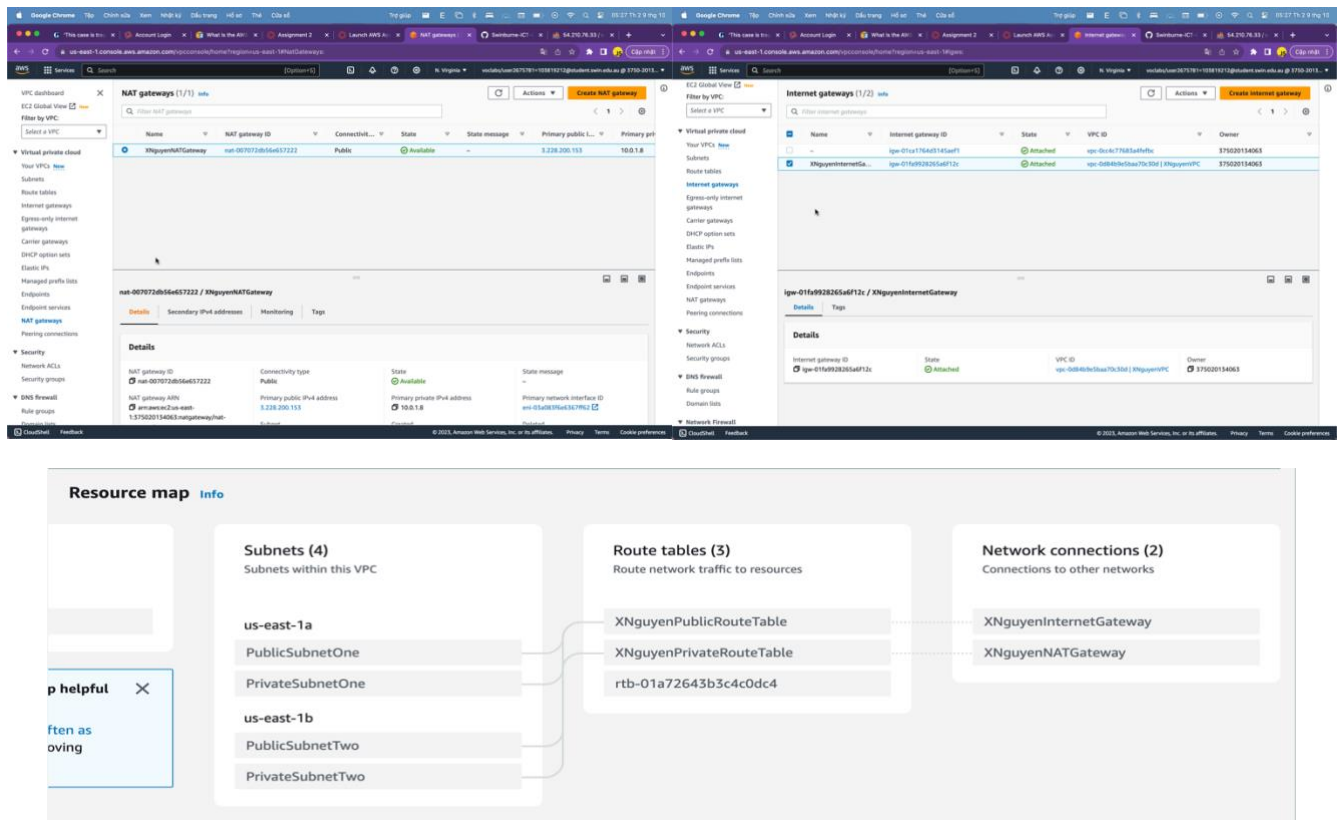
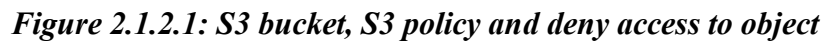


Figure 2.1.1.1: Subnets, Route tables, IGW, NAT and the Resource Map of the VPC

2.1.2. S3 photo storage

In order to create the storage for the objects, such as images, etc. S3 bucket is essentially required. In this stage, beside from using the same implementation from the previous assignment, I have added one more policy to the S3 bucket that allows triggering Lambda function to resize the upload pictures to the S3 bucket.



Elastic Load Balancing (ELB) is used for dividing flows of users into different EC2 instances in order to reduce the overload. In this scenario, I have created an ELB with a target group listener of HTTP 80 to check the health of the instances, which will change the status to unhealthy if one of the machines are overloading or appearing problems. This helps to divide the workflow onto different available m

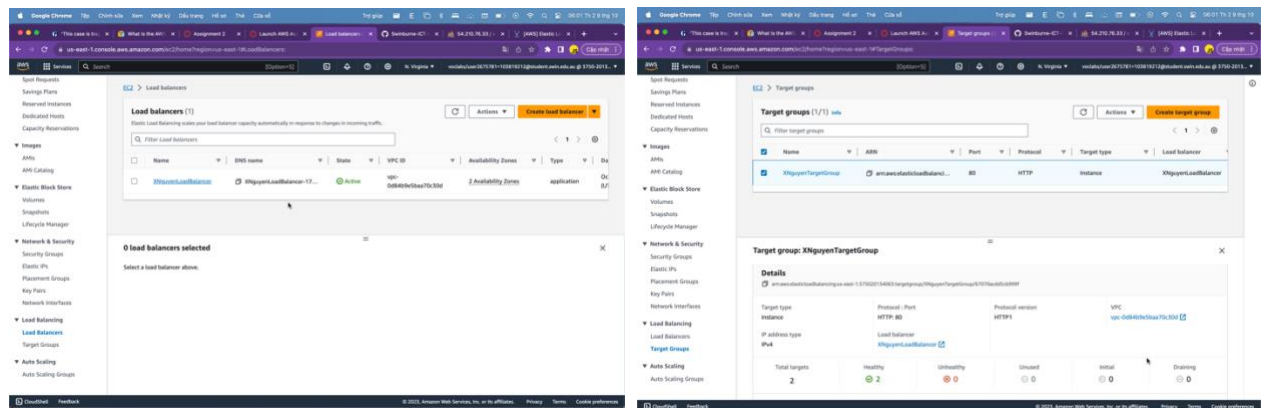


Figure 2.1.3.1: Load balancing and target group

2.1.4. Auto scaling

Beside from load balancing, auto scaling is one of the major components that helps creating multiple instances in special cases, such as when two instances are all overloaded. I have created an auto scaling group that keeps the amount of server is 2 and maximum at 3. Moreover, I have created a rule that triggers the auto scaling when the request to the server is up to 30 requests.

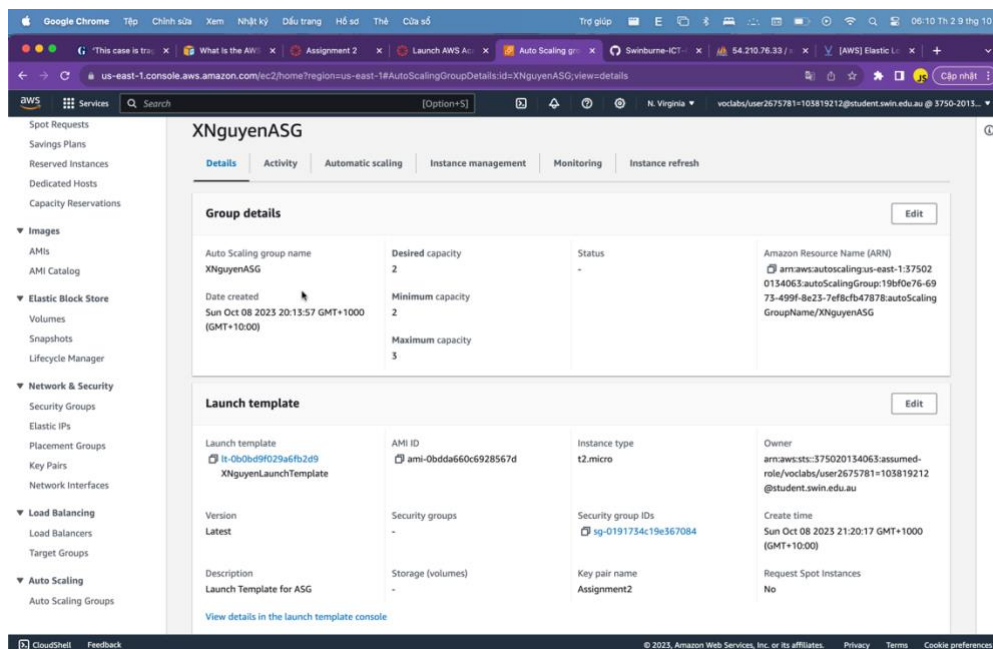


Figure 2.1.4.1: Auto scaling group

2.1.5. EC2 instance

In this stage, I have created a Developer EC2 instance, which will use as a base template for the AMI that will be use for the auto scaling group. The developer EC2 instance will use the LabRole IAM rule, which is already created from the Learner Lab and installed the Apache2 server and MySQL server.

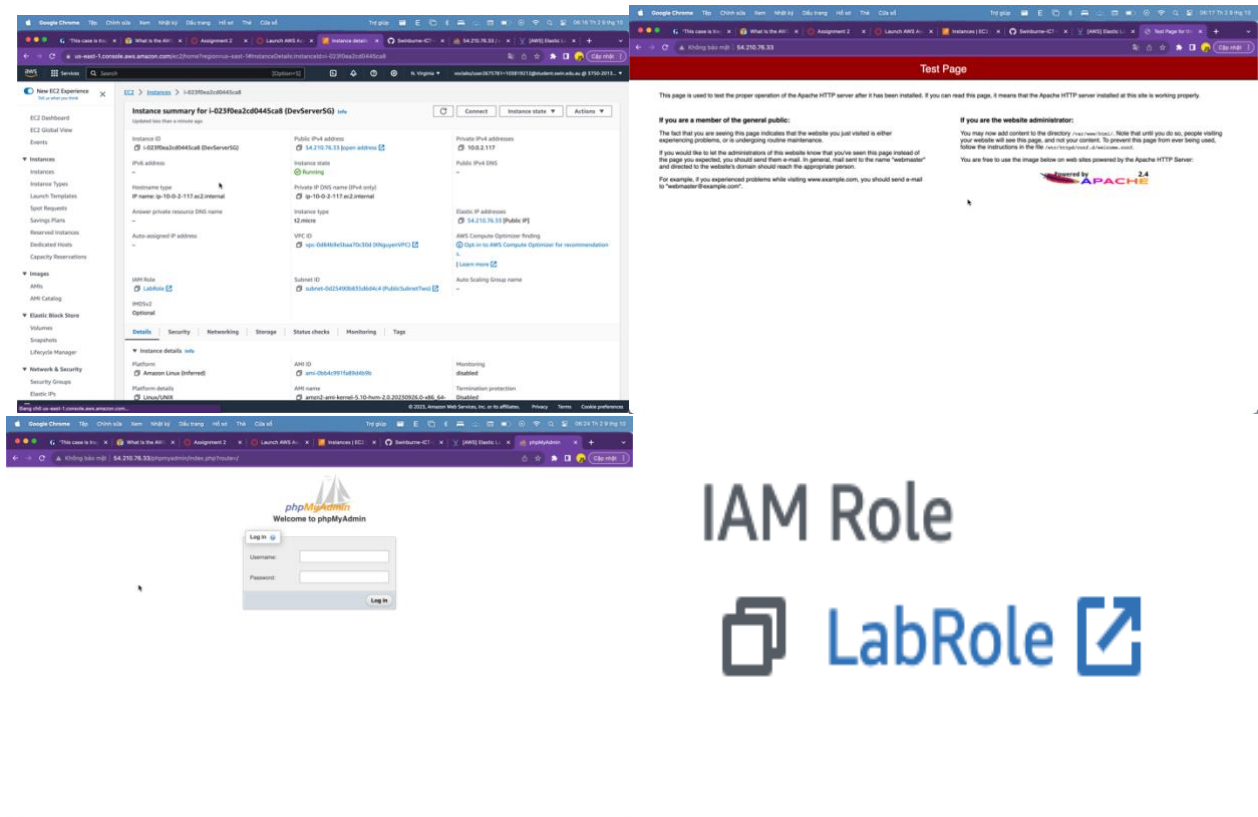


Figure 2.1.5.1: EC2 instance, EC2 IAM role, EC2 Apache and EC2 MySQL

2.1.6. CreateThumbnail Lambda function

Lambda function is a key component in automation, it helps automating the process using programming code. In this scenario, the Lambda function will serve the purpose of resizing the image to the perfect size that fits to the frame using Python 3.11.

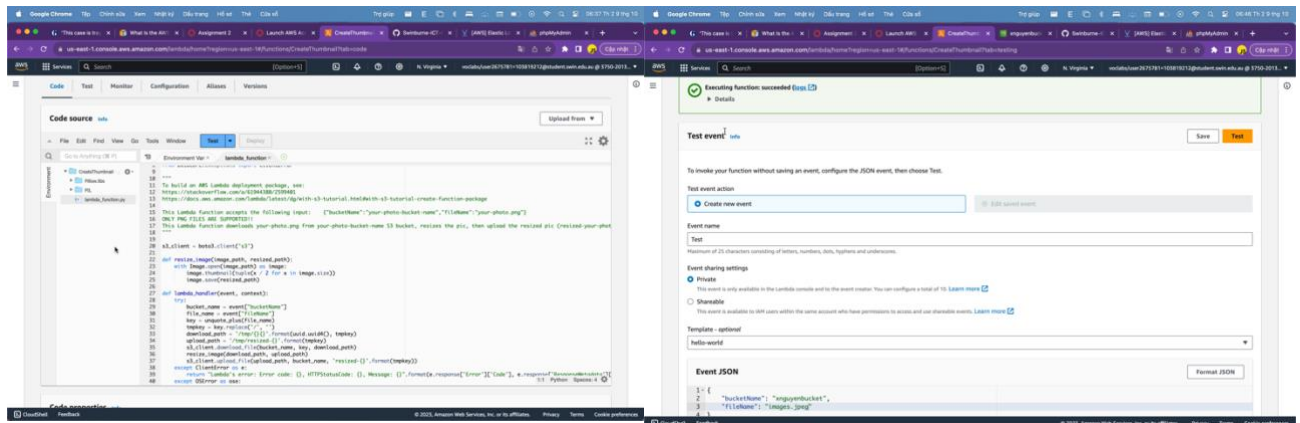


Figure 2.1.6.1: Lambda function and the test case

2.1.7. Database with RDS

There are not much difference from the previous assignment. However, since the web servers are allocated into the private subnet, thus further configurations are required but it is not essential to this assignment so I will pass it.

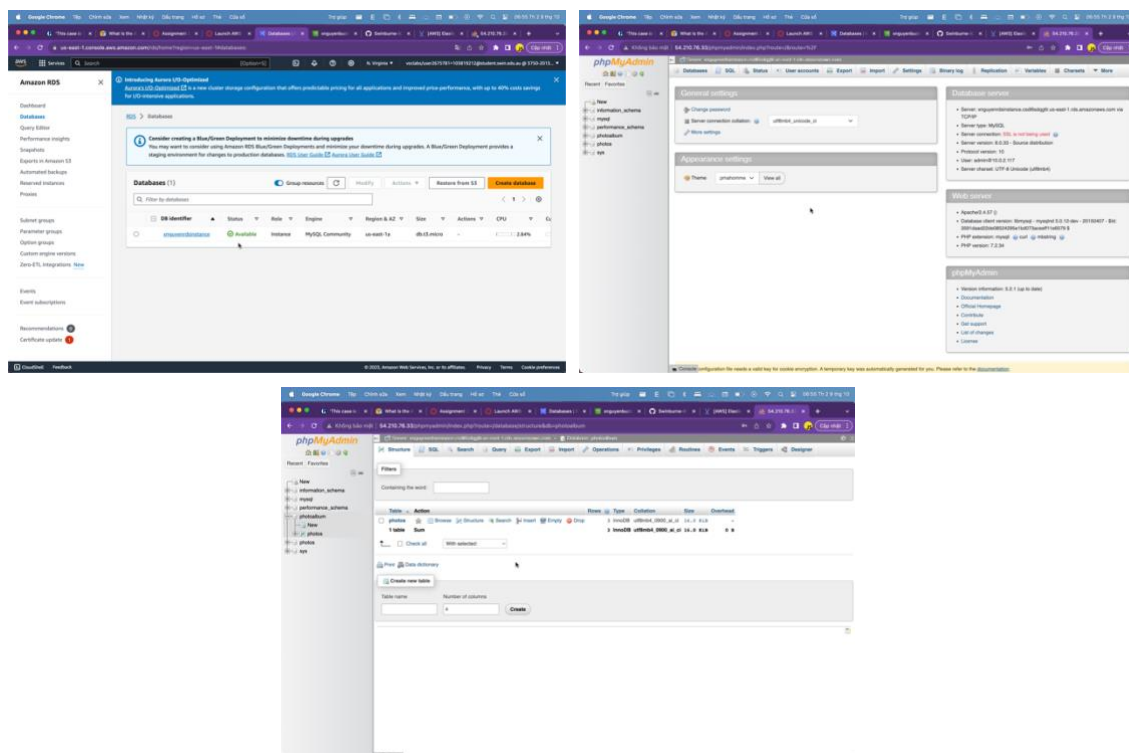
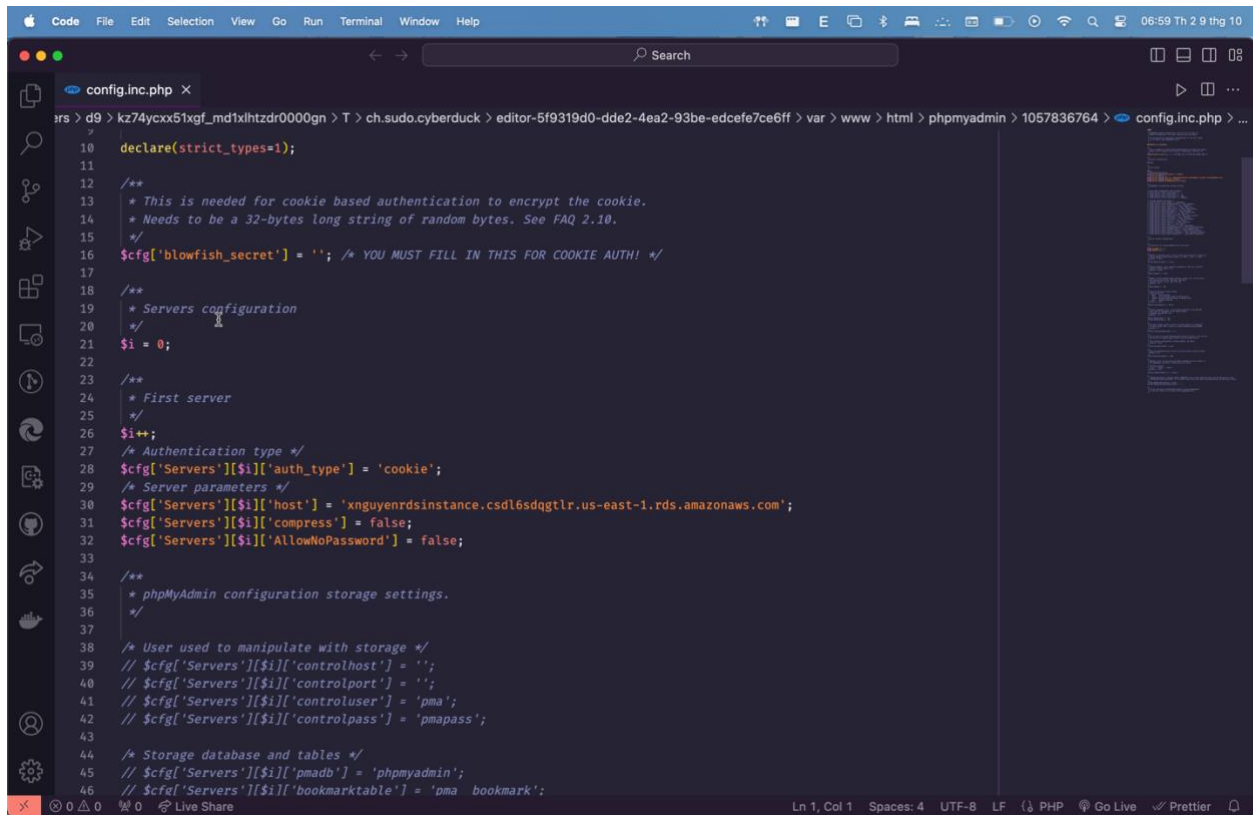


Figure 2.1.7.1: RDS Instance and access to RDS Instance via Dev Server

```
10 declare(strict_types=1);
11
12 /**
13  * This is needed for cookie based authentication to encrypt the cookie.
14  * Needs to be a 32-bytes long string of random bytes. See FAQ 2.10.
15  */
16 $cfg['blowfish_secret'] = ''; /* YOU MUST FILL IN THIS FOR COOKIE AUTH! */
17
18 /**
19  * Servers configuration
20  */
21 $i = 0;
22
23 /**
24  * First server
25  */
26 $i++;
27 /* Authentication type */
28 $cfg['Servers'][$i]['auth_type'] = 'cookie';
29 /* Server parameters */
30 $cfg['Servers'][$i]['host'] = 'xnguyenrdsinstance.csd16sdqgtr.us-east-1.rds.amazonaws.com';
31 $cfg['Servers'][$i]['compress'] = false;
32 $cfg['Servers'][$i]['AllowNoPassword'] = false;
33
34 /**
35  * phpMyAdmin configuration storage settings.
36  */
37
38 /* User used to manipulate with storage */
39 // $cfg['Servers'][$i]['controlhost'] = '';
40 // $cfg['Servers'][$i]['controlport'] = '';
41 // $cfg['Servers'][$i]['controluser'] = 'pma';
42 // $cfg['Servers'][$i]['controlpass'] = 'pmapass';
43
44 /* Storage database and tables */
45 // $cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';
46 // $cfg['Servers'][$i]['bookmarktable'] = 'pma_bookmark';
```

Figure 2.1.7.2: File config.inc.php

```

private > var > folders > d9 > k274ycx51vgf_md1xhtzdr0000gn > T > ch.sudo.cyberduck > editor-5f9319d0-dde2-4ea2-93be-edcfe7ce6ff > var > www > html > photoalbum > -93876728 > constants.php > ...
35 *
36 *
37 *
38 * The values of the constant variables with "[ACTION REQUIRED]" in the comment must be updated. The current values are just examples.
39 * You need to replace the values of those constant variables with values specific to your setup.
40 *
41 * ===== READ THE ABOVE !!! =====
42 * ===== READ THE ABOVE !!! =====
43 * ===== READ THE ABOVE !!! =====
44 *
45
46 // [ACTION REQUIRED] your full name
47 define('STUDENT_NAME', 'Siam Nguyen');
48 // [ACTION REQUIRED] your Student ID
49 define('STUDENT_ID', '103819212');
50 // [ACTION REQUIRED] your tutorial session
51 define('TUTORIAL_SESSION', 'Tuesday 6:30PM');
52
53 // [ACTION REQUIRED] name of the S3 bucket that stores images
54 define('BUCKET_NAME', 'xnguyenbucket');
55 // [ACTION REQUIRED] region of the above bucket
56 define('REGION', 'us-east-1');
57 define('S3_BASE_URL', 'https://'.BUCKET_NAME.'.s3.amazonaws.com/');
58
59 // [ACTION REQUIRED] name of the database that stores photo meta-data (note that this is not the DB identifier of the RDS instance)
60 define('DB_NAME', 'photoalbum');
61 // [ACTION REQUIRED] endpoint of RDS instance
62 define('DB_ENDPOINT', 'xnguyendbinstance.cadltdgglr.us-east-1.rds.amazonaws.com');
63 // [ACTION REQUIRED] username of your RDS instance
64 define('DB_USERNAME', 'admin');
65 // [ACTION REQUIRED] password of your RDS instance
66 define('DB_PWD', 'password');
67
68 // [ACTION REQUIRED] name of the DB table that stores photo's meta-data
69 define('DB_PHOTO_TABLE_NAME', 'photos');
70 // The table above has 5 columns:
71 // [ACTION REQUIRED] name of the column in the above table that stores photo's titles
72 define('DB_PHOTO_TITLE_COL_NAME', 'photo_title');
73 // [ACTION REQUIRED] name of the column in the above table that stores photo's descriptions
74 define('DB_PHOTO_DESCRIPTION_COL_NAME', 'photo_description');
75 // [ACTION REQUIRED] name of the column in the above table that stores photo's creation dates
76 define('DB_PHOTO_CREATIONDATE_COL_NAME', 'photo_creationdate');
77 // [ACTION REQUIRED] name of the column in the above table that stores photo's keywords
78 define('DB_PHOTO_KEYWORDS_COL_NAME', 'photo_keywords');
79 // [ACTION REQUIRED] name of the column in the above table that stores photo's links in S3
80 define('DB_PHOTO_SREFERENCE_COL_NAME', 'photo_reference');
81
82 // [ACTION REQUIRED] name (ARN can also be used) of the Lambda function that is used to create thumbnails
83 define('LAMBDA_FUNC_THUMBNAILS_NAME', 'CreateThumbnail');
84
85
86
7>

```

Figure 2.1.7.3: File constants.php

2.1.8. Security group and Network ACLs

The first security group, which is for developer server will accepts any inbound traffics. Secondly, the security group for ELB will allows HTTP and HTTPS traffic from the NAT gateway. Thirdly, the web server security group only allows for HTTP and HTTPS. Finally, the security group for database will accepts inbound traffic from the security group of web server. On the other hand, the network ACL rule will block ICMP traffic from bi-direction, which is in and out from private security group.

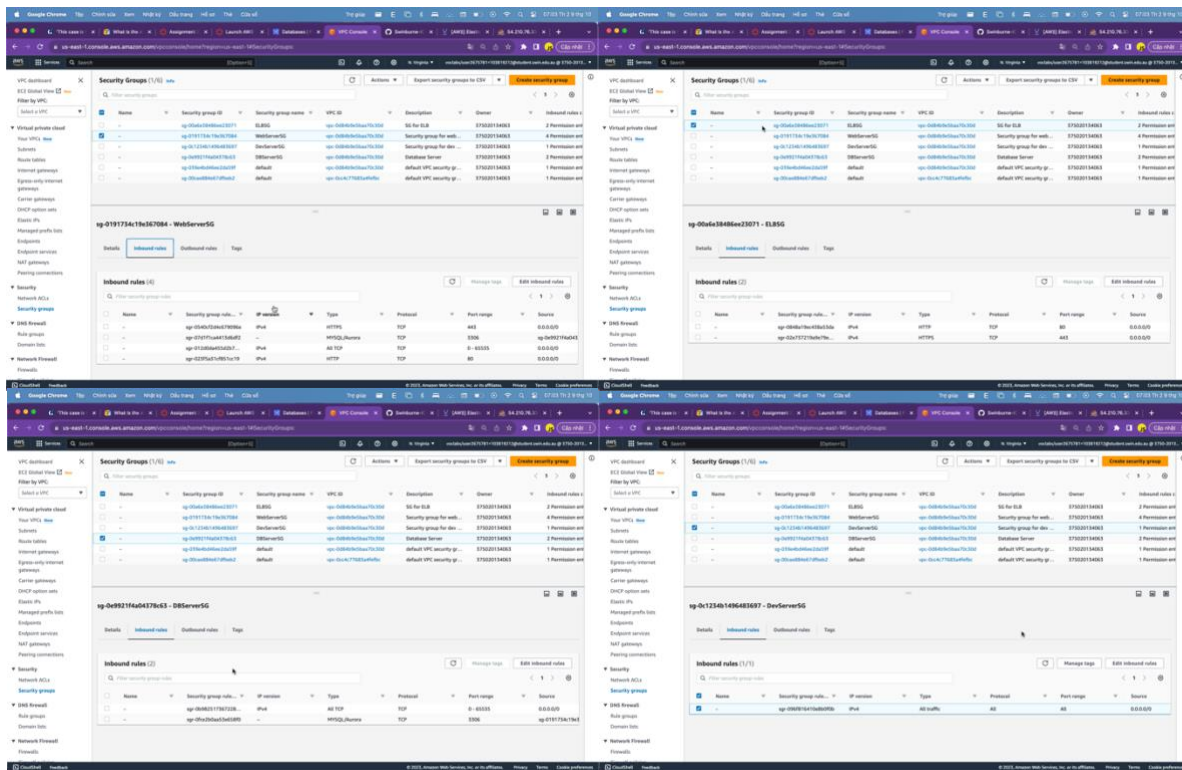


Figure 2.1.8.1: Rules of security group

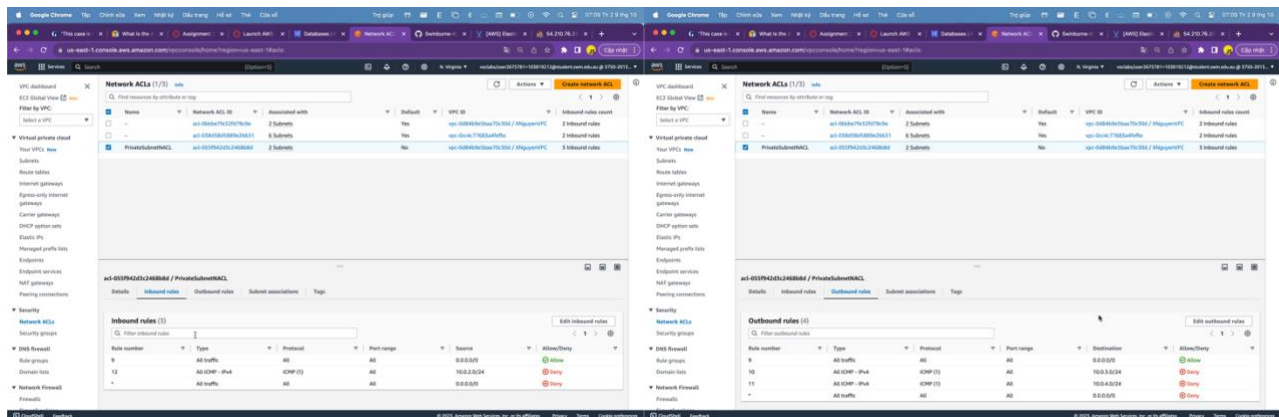


Figure 2.1.8.2: Rules of Network ACL

3. Result

```

ec2-user@ip-10-0-2-117:~
~ ~ \_#####\
~ ~ \###|      AL2 End of Life is 2025-06-30.
~ ~ \#/  ---
~ ~ V~'  '->
~ ~ ~
~ ~ ~ /      A newer version of Amazon Linux is available!
~ ~ ~ _./  _/
~ ~ ~ _/  _/      Amazon Linux 2023, GA and supported until 2028-03-15.
~ ~ ~ _/m/'      https://aws.amazon.com/linux/amazon-linux-2023/

-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file
or directory
[ec2-user@ip-10-0-2-117 ~]$ ping 0.0.0.0
PING 0.0.0.0 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=255 time=0.019 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=255 time=0.030 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=255 time=0.030 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=255 time=0.030 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=255 time=0.029 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=255 time=0.029 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=255 time=0.028 ms
^C
--- 0.0.0.0 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6150ms
rtt min/avg/max/mdev = 0.019/0.027/0.030/0.007 ms
[ec2-user@ip-10-0-2-117 ~]$

```

Figure 3.1: Ping developer server

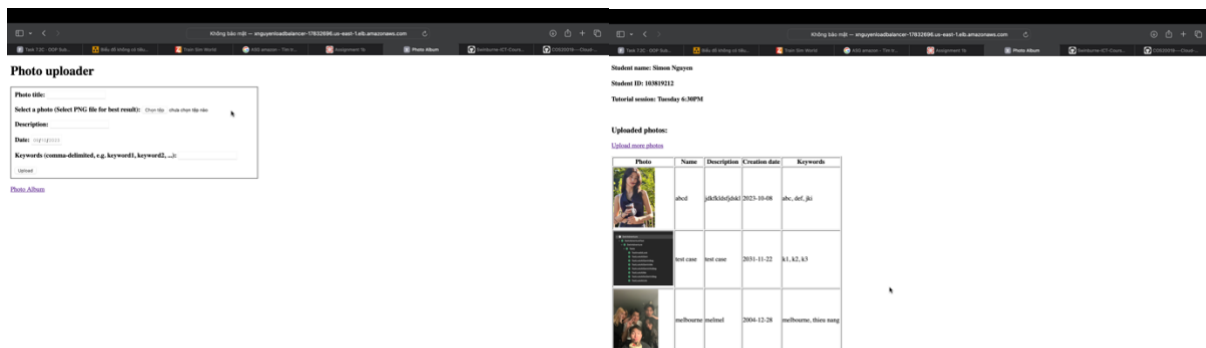


Figure 3.2: album.php

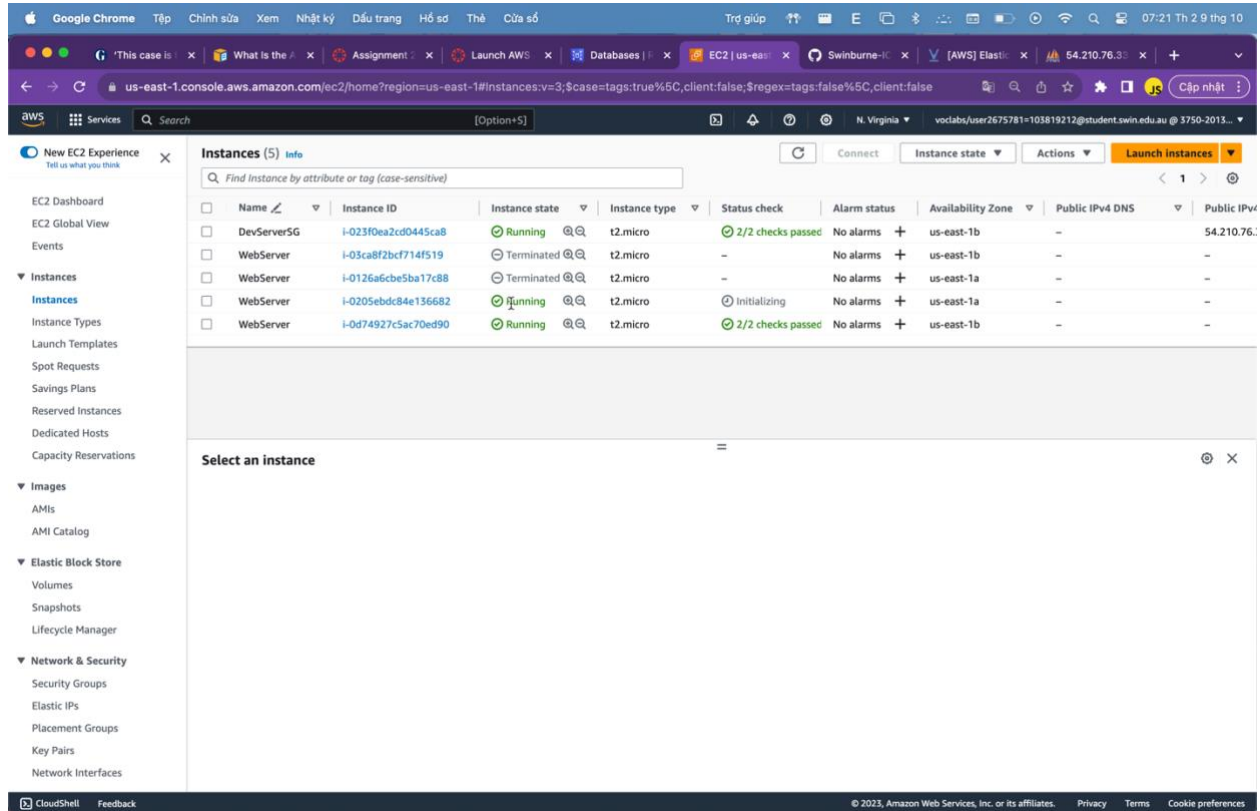


Figure 3.3: Auto scaling group in action

Link to the website

Link to developer server: <http://54.210.76.33>

Link to ELB photouploader.php: <http://xnguyenloadbalancer-17832696.us-east-1.elb.amazonaws.com/photoalbum/photouploader.php>

Link to ELB album.php: <http://xnguyenloadbalancer-17832696.us-east-1.elb.amazonaws.com/photoalbum/album.php>