

**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](#_Toc147728695)

[Introduction 3](#_Toc147728696)

[*1.* *Introduction* 3](#_Toc147728697)

[*2.* *Report’s objective* 3](#_Toc147728698)

[The Implementation of the project 3](#_Toc147728699)

[*1.* *Infrastructure diagram* 3](#_Toc147728700)

[*2.* *The implementation* 4](#_Toc147728701)

[2.1.1. The VPC and NAT Gateway 4](#_Toc147728702)

[2.1.2. S3 photo storage 5](#_Toc147728703)

[2.1.3. Load balancing 6](#_Toc147728704)

[2.1.4. Auto scaling 7](#_Toc147728705)

[2.1.5. EC2 instance 8](#_Toc147728706)

[2.1.6. CreateThumbnail Lambda function 8](#_Toc147728707)

[2.1.7. Database with RDS 9](#_Toc147728708)

[2.1.8. Security group and Network ACLs 11](#_Toc147728709)

[*3.* *Result* 12](#_Toc147728710)

[Link to the website 14](#_Toc147728711)

# **Introduction**

## *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.

## *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**

## *Infrastructure diagram*

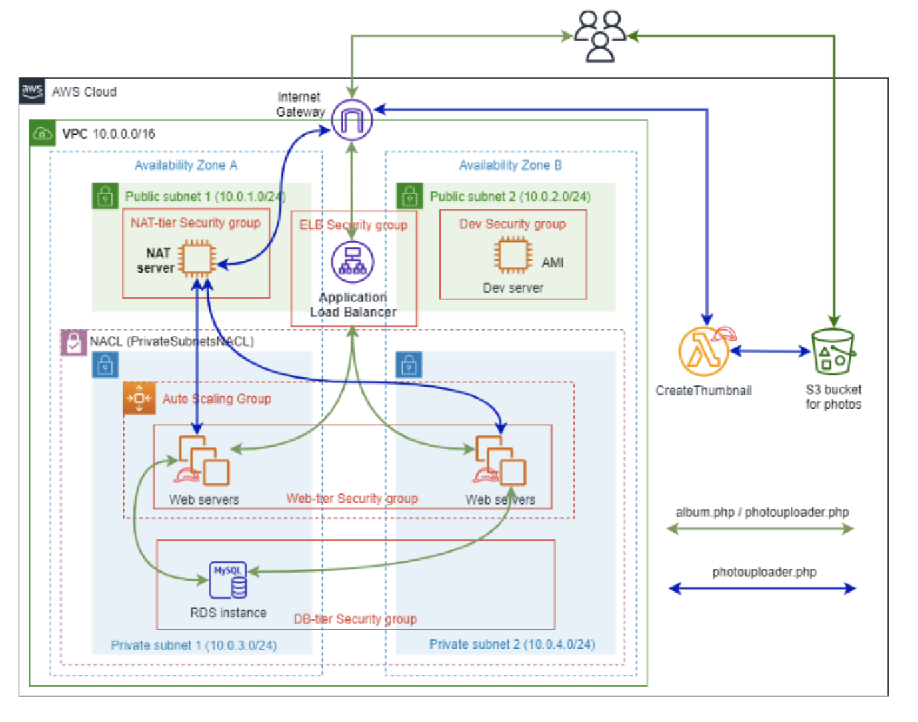


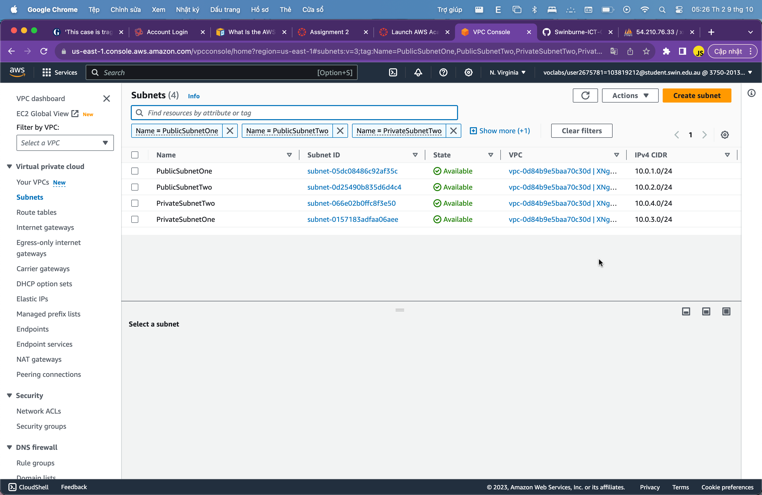
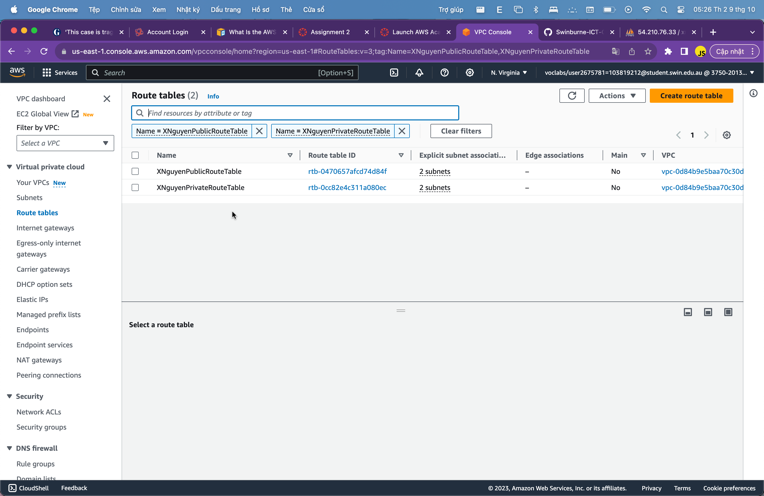
Figure 1 - Architecture diagram

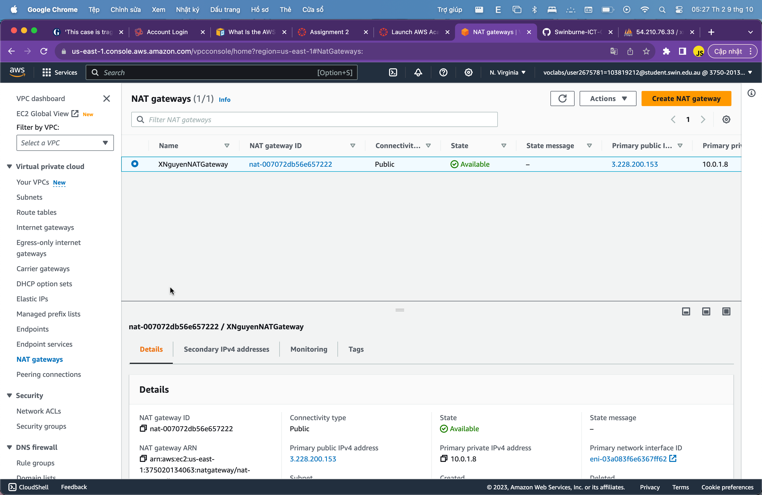
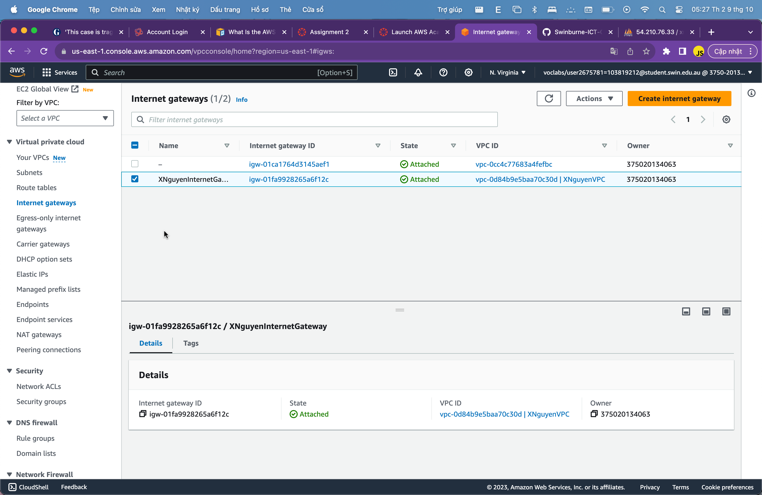
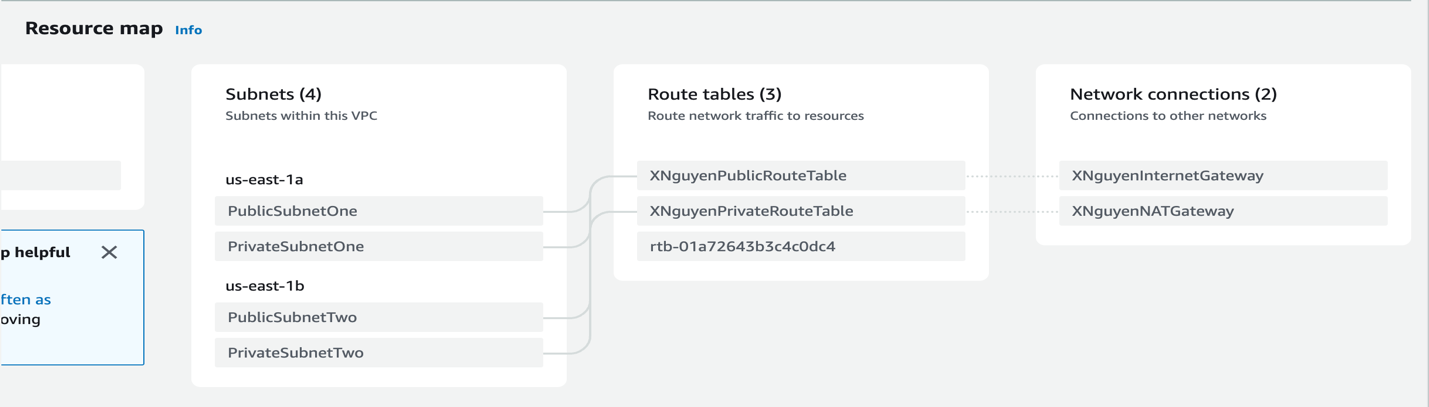
The diagram above demonstrates the required infrastructures that are needed to create a robust, scalable and reliable network environment for the system. In general, this system is relatively the same as the previous assignment, which comprises of one VPC (10.0.0.0/16) and four subnets (two public subnets 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.

## *The implementation*

### 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.

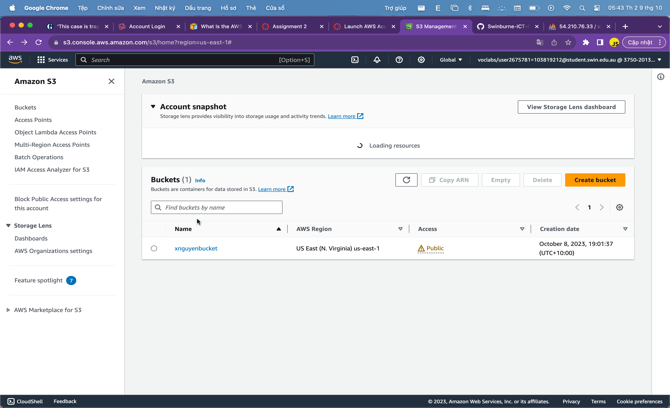
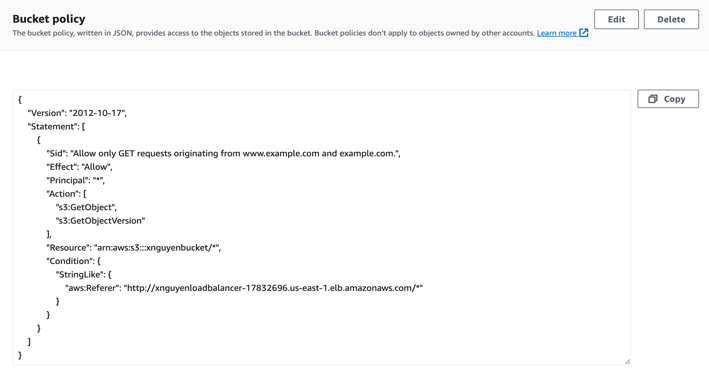
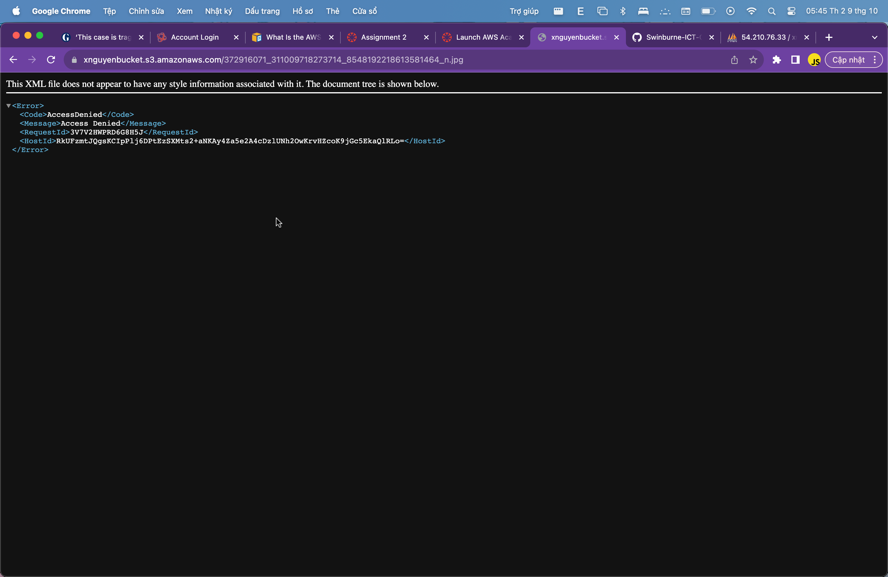




***Figure 2.1.1.1: Subnets, Route tables, IGW, NAT and the Resource Map of the VPC***

### S3 photo storage

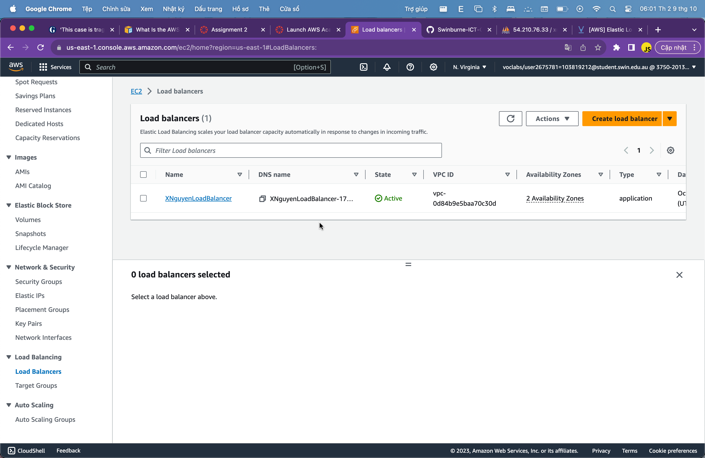
In order to create the storage for the objects, such as images, etc. S3 bucket is essentially required. In this stage, beside from using the same implementation from the previous assignment, I have added one more policy to the S3 bucket that allows triggering Lambda function to resize the upload pictures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



***Figure 2.1.2.1: S3 bucket, S3 policy and deny access to object***

### Load balancing

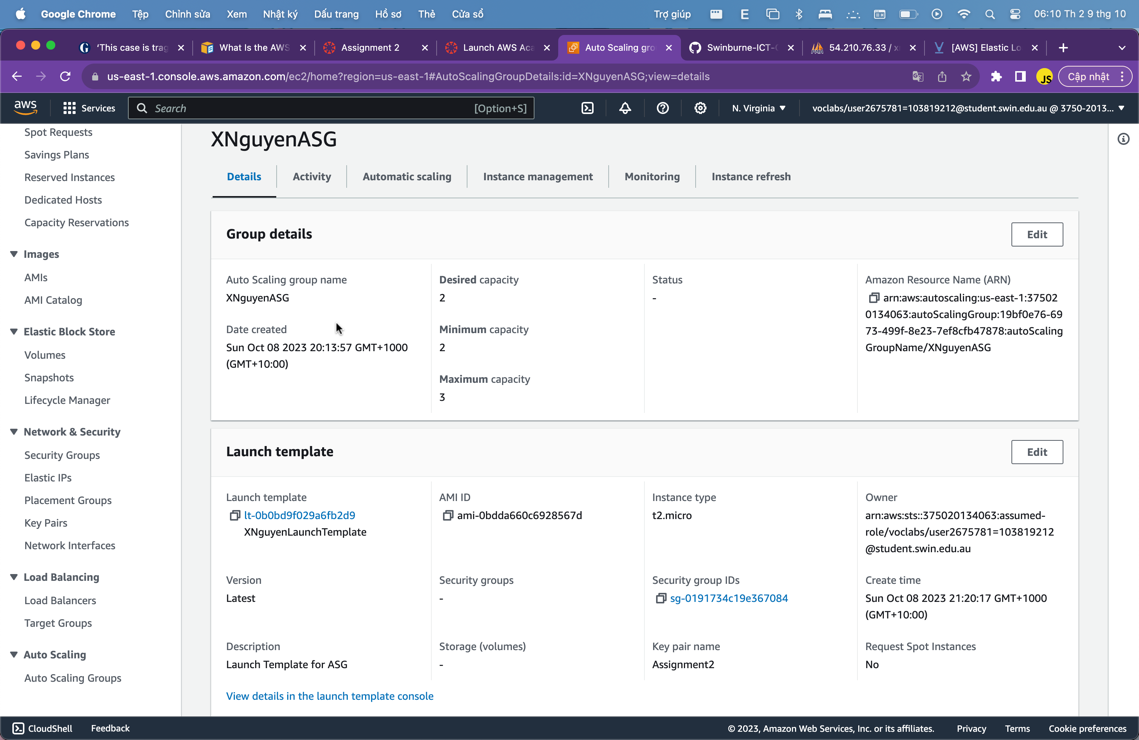
Elastic Load Balancing (ELB) is use for diving flows of users into different EC2 instances in order to reduce the overload. In this scenario, I have created a ELB with a target group listener of HTTP 80 to check the health of the instances, which will change the status to unhealthy if one of the machines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

A screenshot of a computer

Description automatically generated

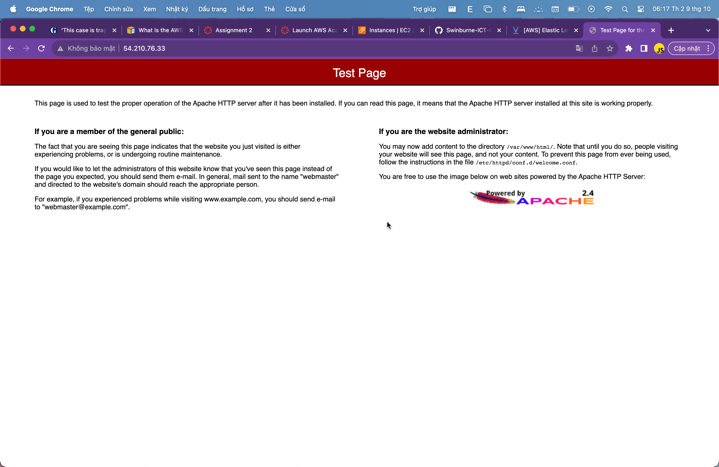
***Figure 2.1.3.1: Load balancing and target group***

### Auto scaling

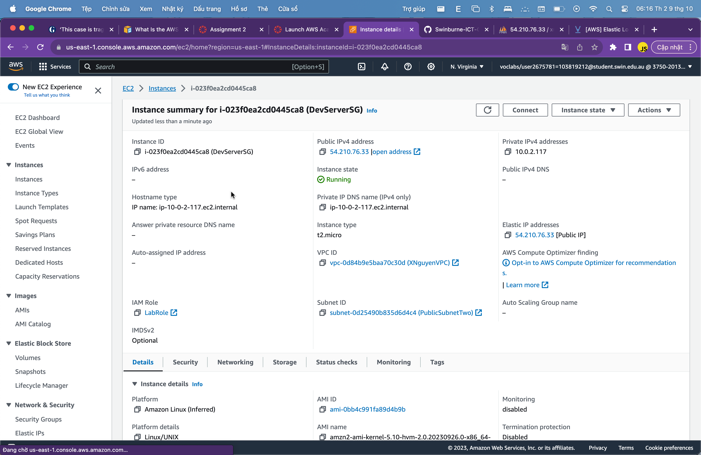
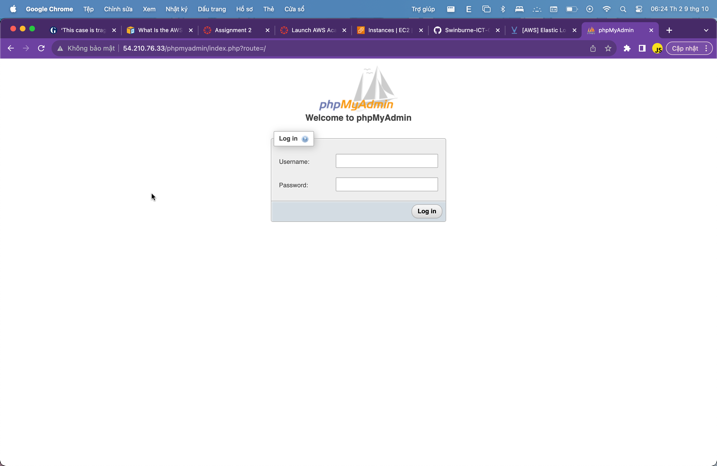
Beside from load balancing, auto scaling is one of the major components that helps creating multiple instances in special cases, such as when two instances are all overloaded. I have created an auto scaling group that keeps the amount of server is 2 and maximum at 3. Moreover, I have created a rule that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

***Figure 2.1.4.1: Auto scaling group***

### 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.

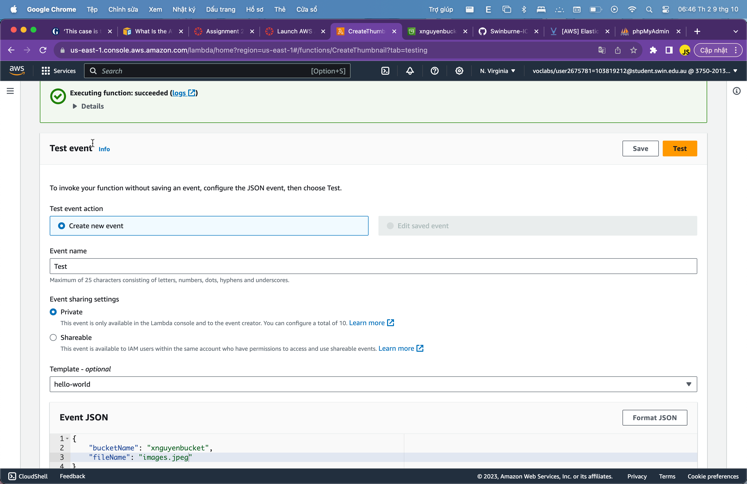
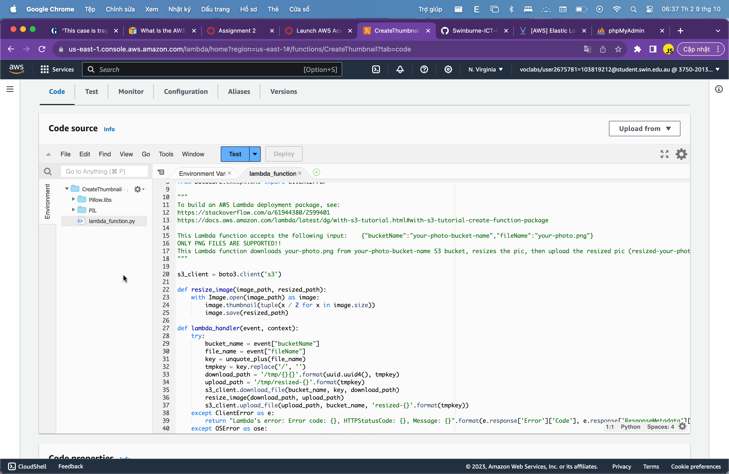
A screenshot of a computer

Description automatically generated

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

### CreateThumbnail Lambda function

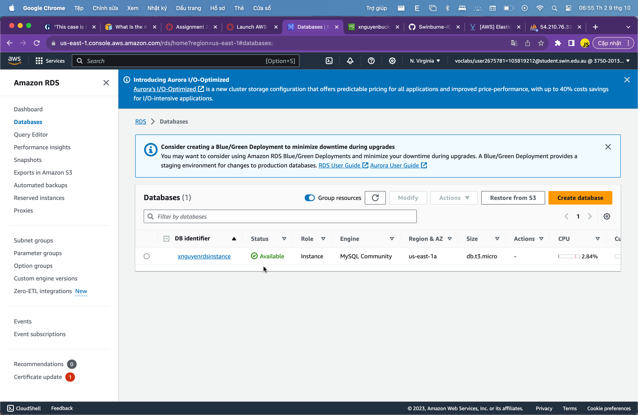
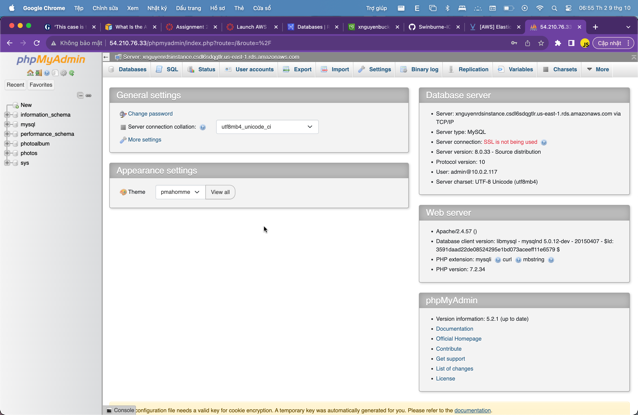
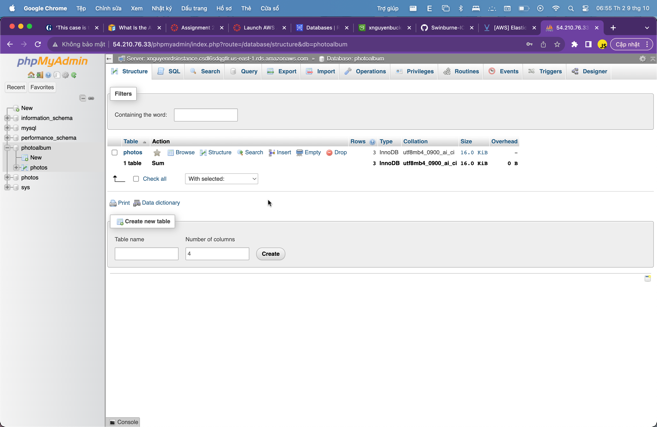
Lambda function is a key component in automation, it helps automating the process using programming code. In this scenario, the Lambda function will serve the purpose of resizing the image to the perfect size that fits to the frame using Python 3.11.



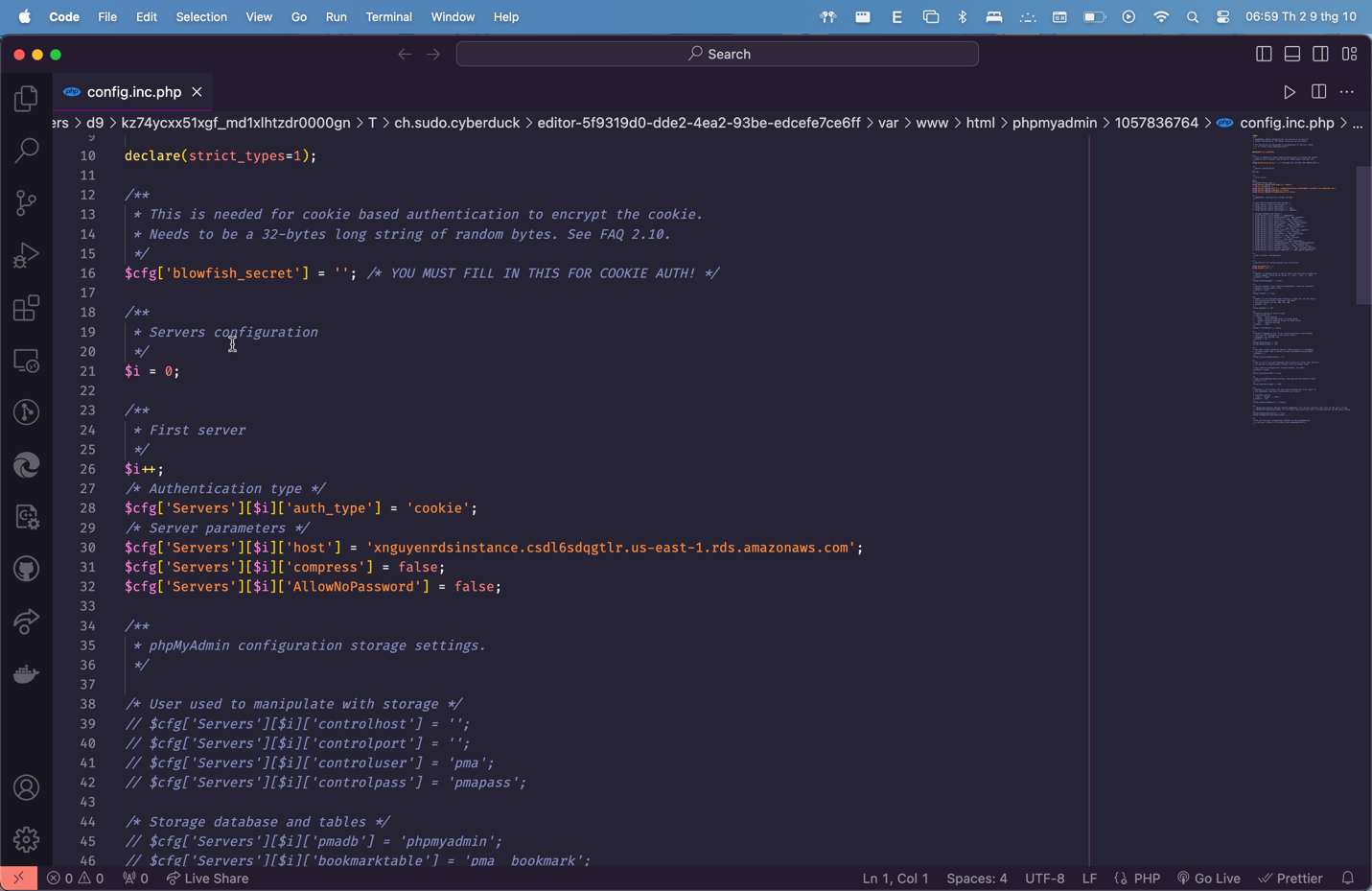
***Figure 2.1.6.1: Lambda function and the test case***

### Database with RDS

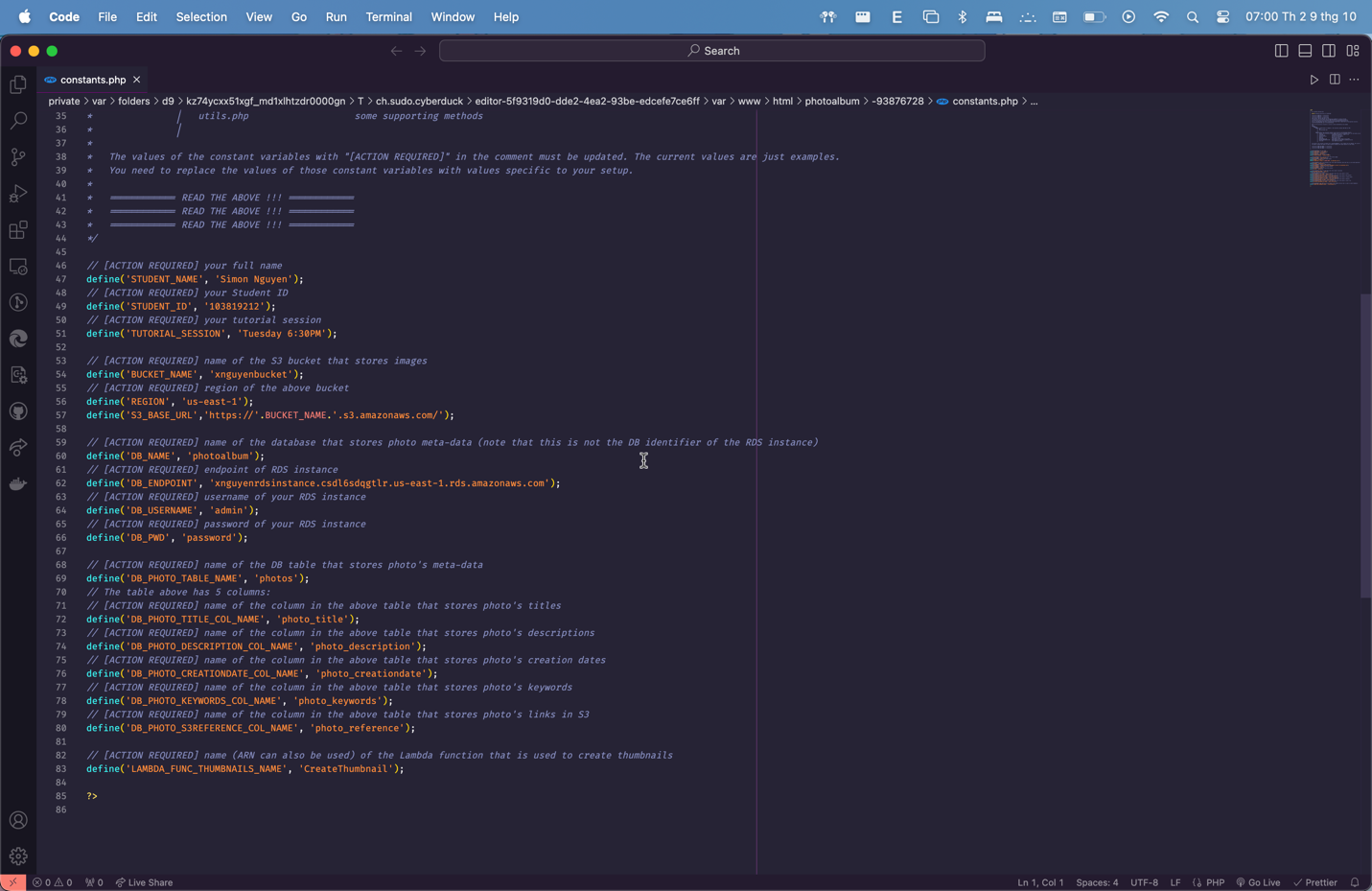
There are not much difference from the previous assignment. However, since the web servers are allocated into the private subnet, thus further configurations are required but it is not essential to this assignment so I will pass it.



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

******

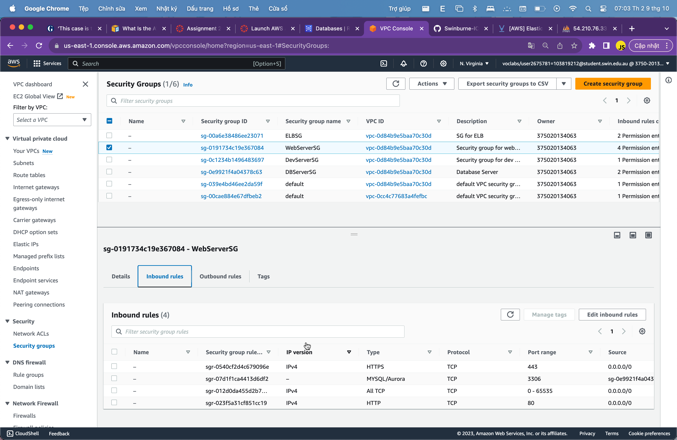
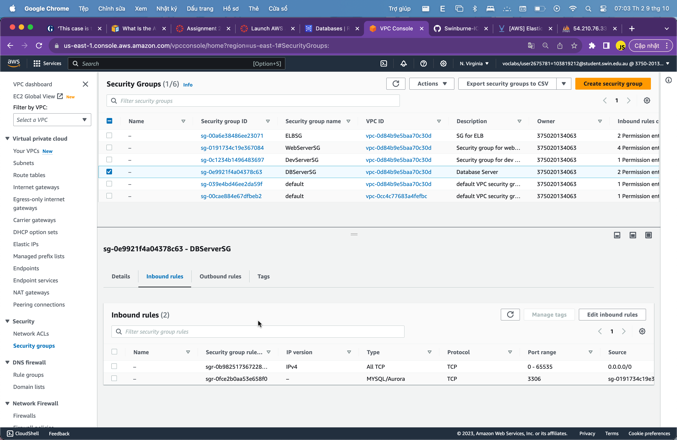
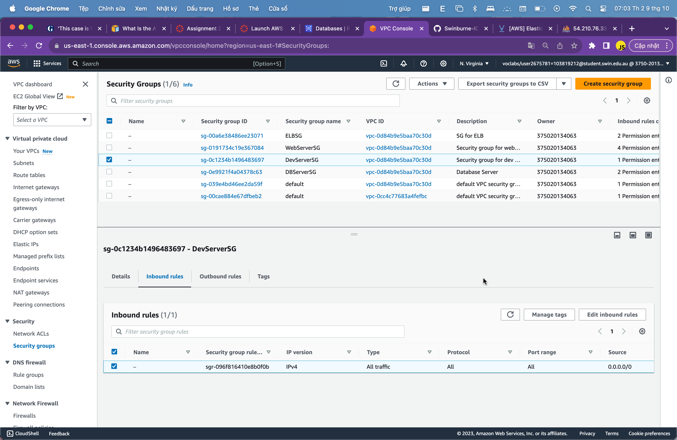
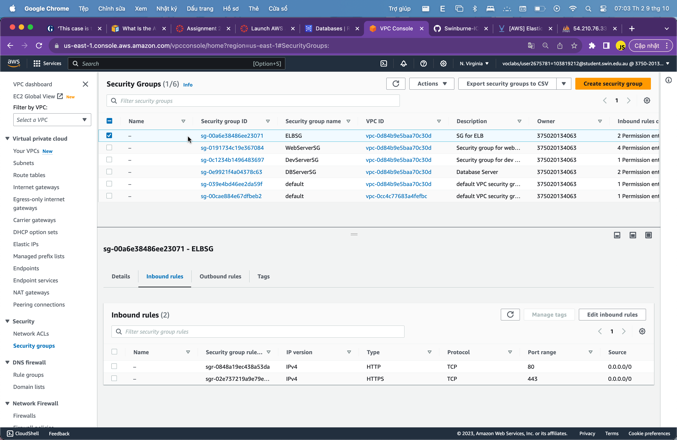
***Figure 2.1.7.2: File config.inc.php***

******

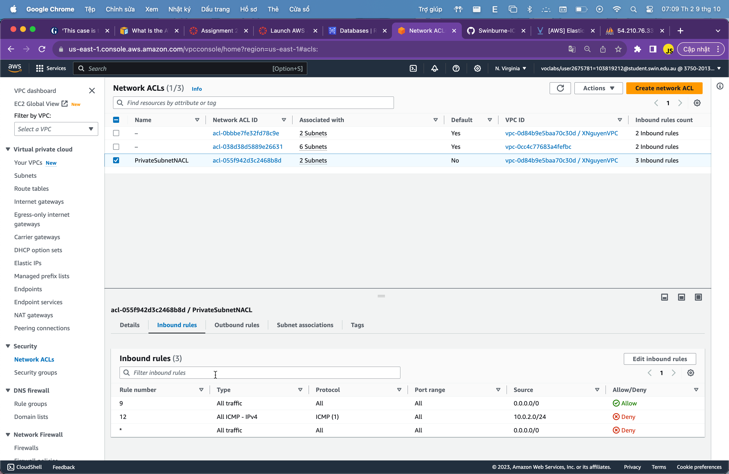
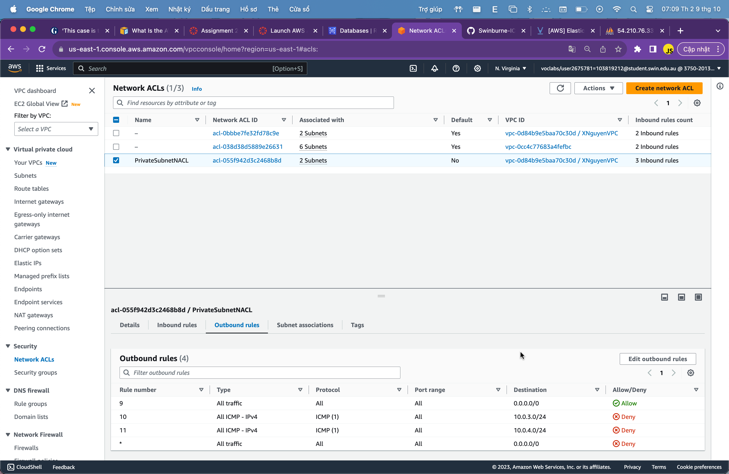
***Figure 2.1.7.3: File constants.php***

### 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.

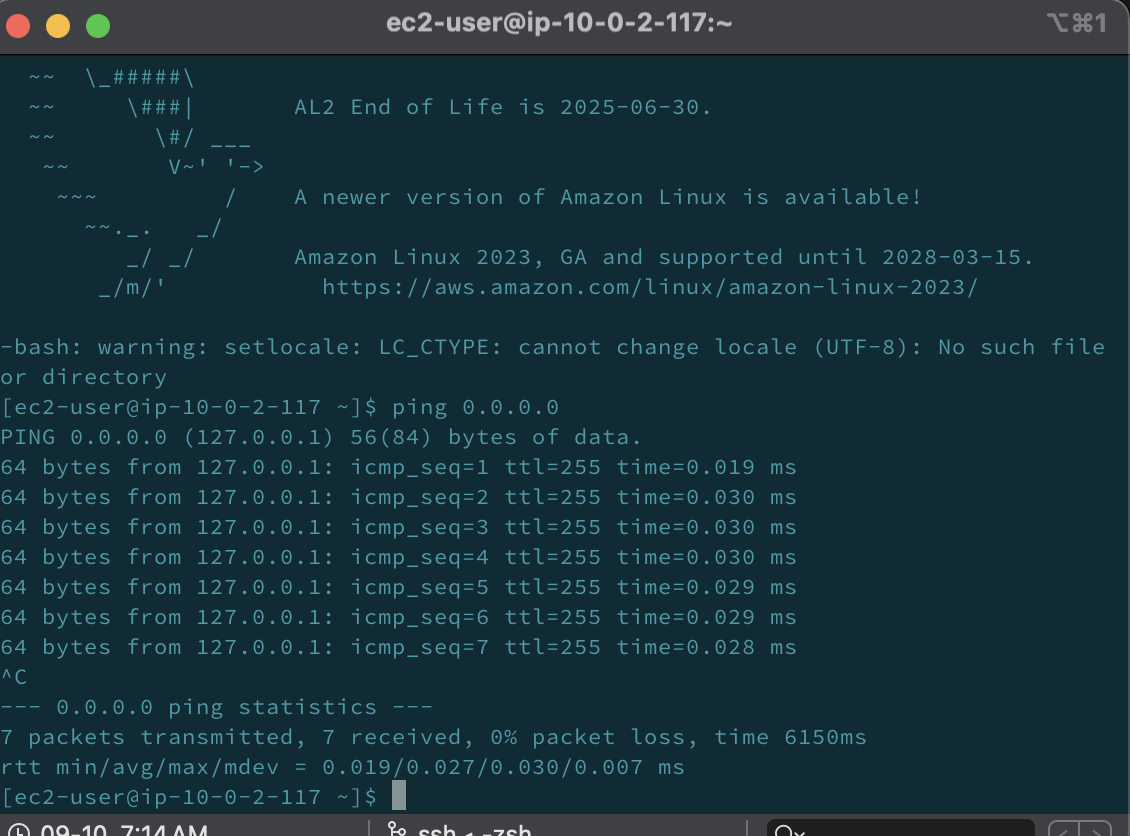


***Figure 2.1.8.1: Rules of security group***

******

***Figure 2.1.8.2: Rules of Network ACL***

## *Result*



***Figure 3.1: Ping developer server***

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

***Figure 3.2: album.php***

A screenshot of a computer

Description automatically generated

***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>