

Assignment 2 - Developing a highly available Photo Album website

Student name: Hai Nam Ngo

Student ID: 103488515

Tutorial session: Friday 04:30PM

URL of Album Page: <http://hngo-lb-609917414.us-east-1.elb.amazonaws.com/photoalbum/album.php>

I. INTRODUCTION

In this assignment, an AWS infrastructure was built based on the requirement. This entails incorporating LabRole and LabInstanceProfile IAM roles to facilitate secure interactions among EC2, Lambda, and S3 services. I will also implement restricted S3 access, craft a Lambda function, and create a custom AMI. Using this AMI, a new launch template and Auto Scaling Group will be developed, spanning multiple Availability Zones with scaling policies in place. Additionally, I will establish an Elastic Load Balancer and enforce stringent access and traffic controls via AWS NACLs to ensure secure and streamlined operations.

II. VPC

The VPC is like what has been created in Assignment 1b, the main VPC name is “HNgoVPC” with 2 availability zones and two subnets for each, that means four subnets in total as we can see below, correct configuration.

The public subnets will then connect to a public route table of the same VPC, which then connects to the Internet Gateway, making them become valid “public subnet”.

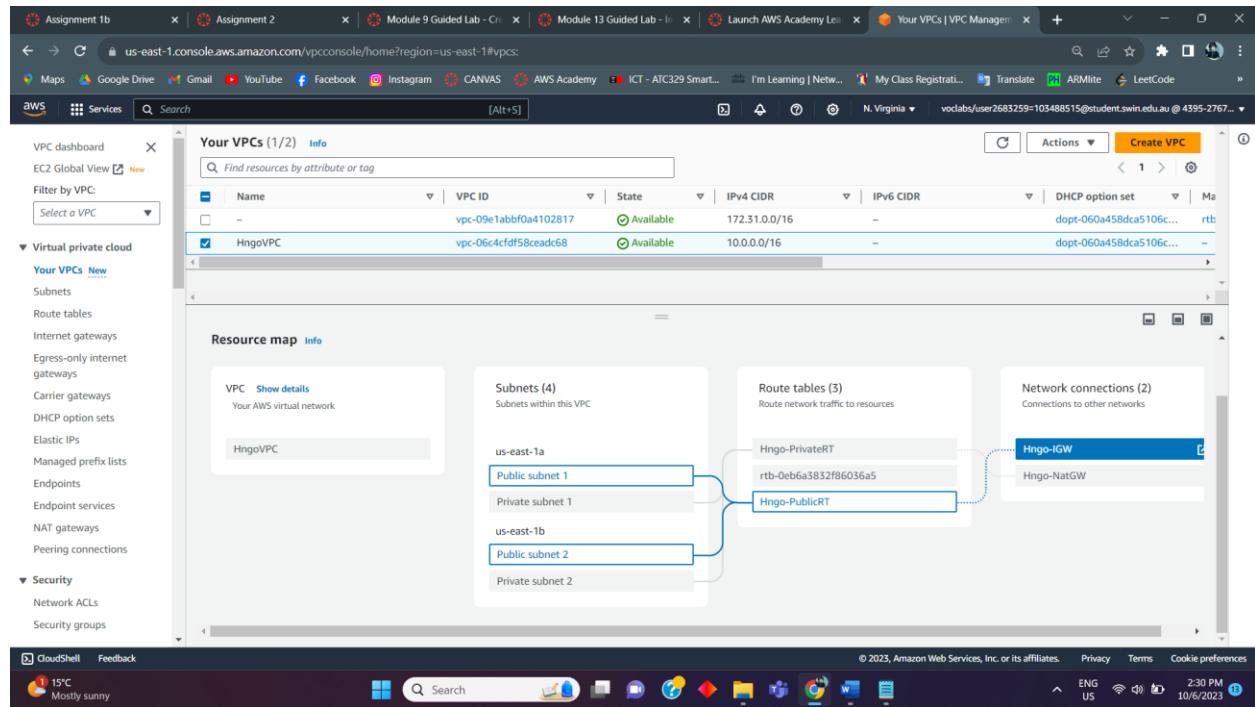


Figure 1 Resource map of HNgoVPC with subnets, route table, internet gateway and NAT gateway

On the other hand, private subnets will connect to a private route table, which is connected to a NAT Gateway. The way I created this Gateway will be illustrated in the next section.

III. NETWORK ADDRESS TRANSLATION (NAT)

The NAT Gateway service in AWS Console was chosen instead of creating a NAT Instance manually since NAT Gateway is easier to create. The configuration is making the Gateway available in the HNgoVPC and connect it with the subnets that I illustrate above.

The screenshot shows the AWS VPC dashboard with the 'NAT gateways' section selected. A single NAT gateway, 'Hngo-NatGW', is listed with the ID 'nat-0581324fb4fc63e1a'. The details panel shows the following information:

NAT gateway ID	Connectivity type	State	Primary public IP address	Primary private IP address	Primary network interface ID
nat-0581324fb4fc63e1a	Public	Available	23.23.213.149	10.0.1.73	eni-0bcebd7af87cac27

Other tabs in the details panel include Secondary IPv4 addresses, Monitoring, and Tags.

Figure 2 Creating Nat Gateway correctly using AWS Service

IV. EC2 INSTANCE

The EC2 Instance that needs to be created will be DevServer Instance, which is also allocated elastic IP address.

The screenshot shows the AWS EC2 Instances page. A new instance named 'DevServer' is listed with the instance ID 'i-0e0690b54931bae28'. The instance is currently 'Running'. The details panel shows the following configuration:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
DevServer	i-0e0690b54931bae28	Running	t2.micro	Initializing	No alarms	us-east-1b	-	18.214.237.4

Other tabs in the details panel include Security, Networking, Storage, Status checks, Monitoring, and Tags.

Figure 3 DevServer Instance is created correctly

It will be available in public subnet 2, have IAM Role “LabInstanceProfile” attached to it, and DevServerSG is also attached to this instance, however, the specification of the rule will be illustrated in the security group section.

V. DATABASE WITH RDS

Figure 4 hngords is created correctly

The hngords database is created with HNGoVPC, the subnet group that was created above, and DBServerSG (which will be specified in the security group section).

Next step, DevServer Instance will be used to connect to putty, which will allow us to download aws file and phpMyAdmin file, which are all the information and resources needed to build the website, combined with this database.

Figure 5 Access putty.exe by using DevServer Instance

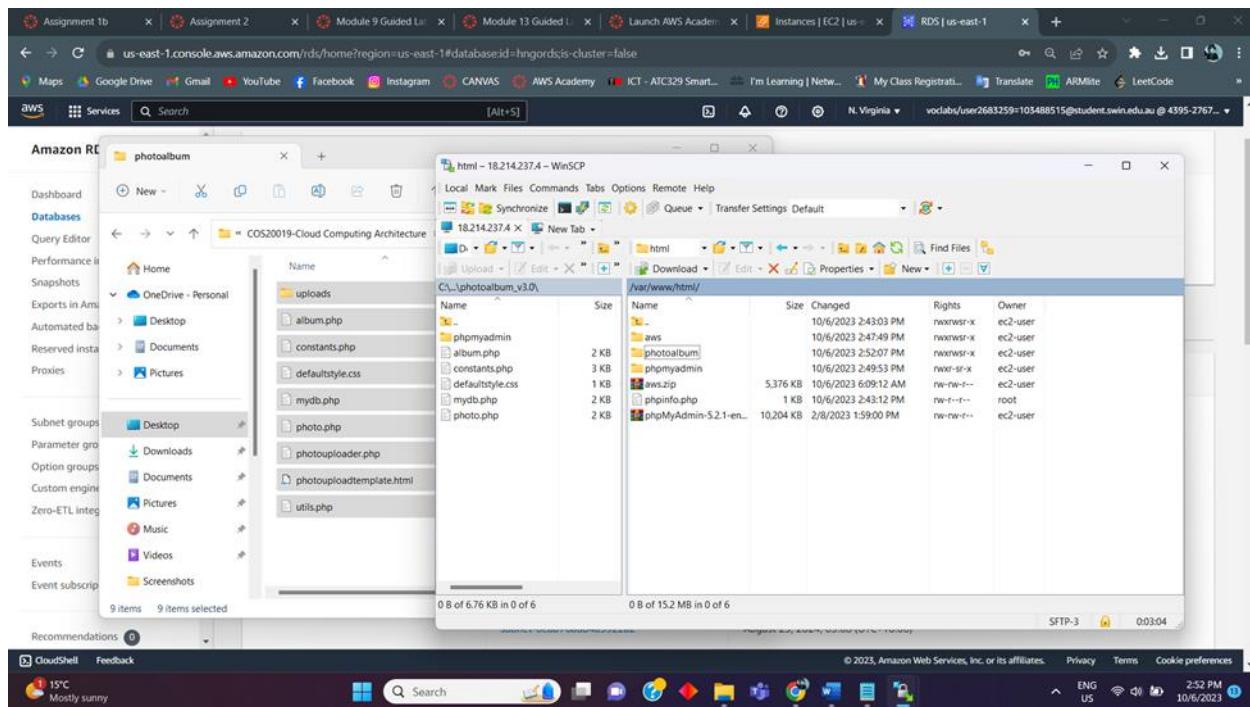


Figure 6 Access to WinSCP

Another file which is called “photoalbum” will be created to contain every file for the album website.

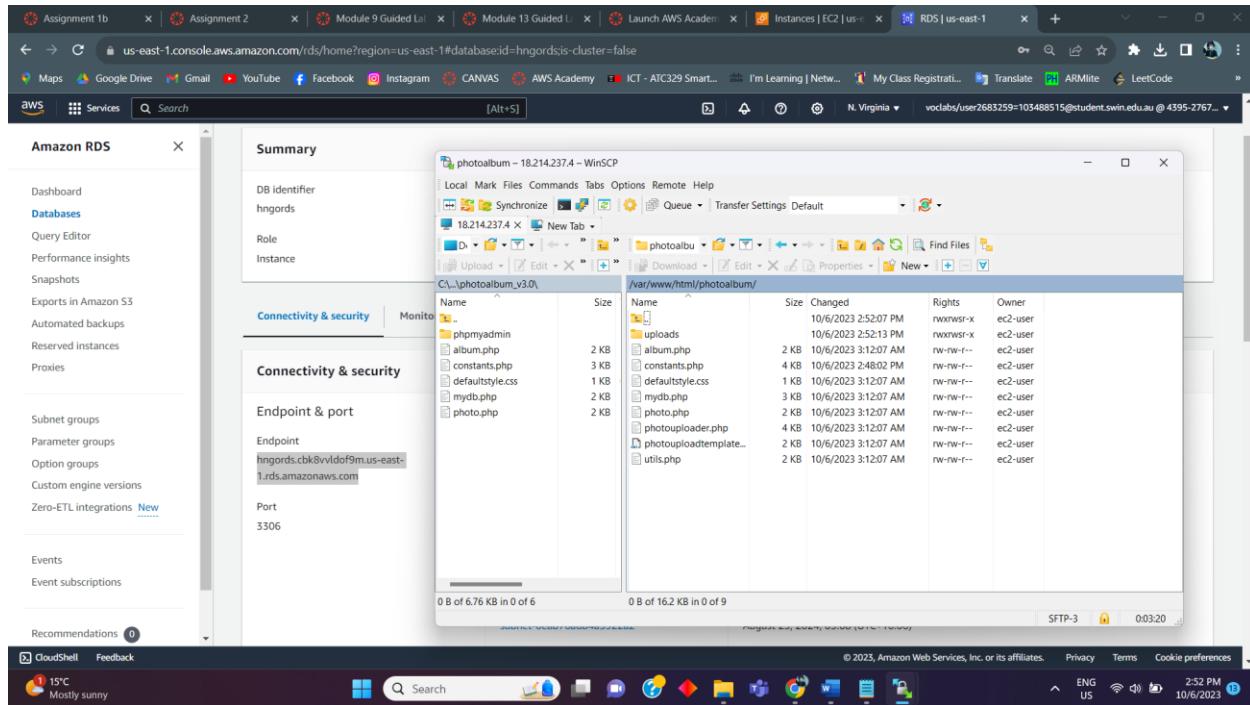


Figure 7 Files in the "photoalbum"

Next, using WinSCP, we access the resources that were downloaded earlier. We will modify the “config.sample.inc.php” to “config.inc.php”, and change the host server to the endpoint of RDS

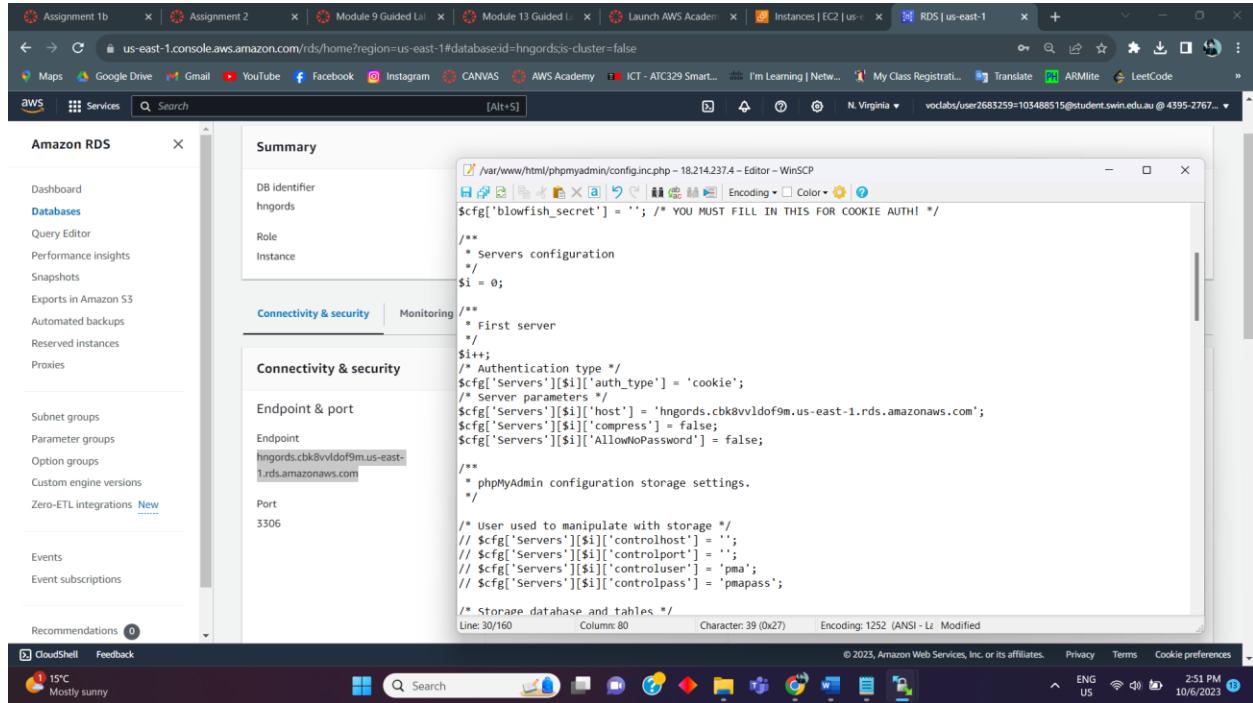


Figure 8 Modify the information to the endpoint of RDS

Finally, is accessing the phpMyAdmin to create a database and a table in that database, which contain all information needed for the photo album website.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	PhotoTitle	varchar(255)	utf8mb4_0900_ai_ci		No	None		Change Drop More	
2	Description	varchar(255)	utf8mb4_0900_ai_ci		No	None		Change Drop More	
3	CreationDate	date			No	None		Change Drop More	
4	Keywords	varchar(255)	utf8mb4_0900_ai_ci		No	None		Change Drop More	
5	Reference	varchar(255)	utf8mb4_0900_ai_ci		No	None		Change Drop More	

Figure 9 "photos" table inside the "photos" database

VI. LOAD BALANCING

I created a target group and a load balancer to support the auto scaling to create a highly available environment.

Target group will be named “Hngo-TG” and it will be linked with the load balancer later.

The screenshot shows the AWS EC2 Target Groups page. On the left, there's a navigation sidebar with sections like Instances, Images, Elastic Block Store, and Network & Security. The main content area displays a table titled 'Target groups (1/1) Info' with one entry: 'Hngo-TG'. The table columns include Name, ARN, Port, Protocol, Target type, Load balancer, and VPC ID. Below the table, a detailed view of the 'Hngo-TG' target group is shown, with tabs for Details, Targets, Monitoring, Health checks, Attributes, and Tags. The 'Details' tab is selected, showing the ARN, Target type (Instance), Protocol (HTTP: 80), IP address type (IPv4), and VPC information (vpc-06c4cfdf58ceadc68). The status bar at the bottom indicates the date and time as 10/8/2023.

Figure 10 Hngo-TG (Target group) is created successfully

Next, the load balancer mentioned earlier will be created right after that. This load balancer will be placed in public subnets in HNGoVPC. And the most important thing is the health check path must be correctly configured as “/photoalbum/gallery.php”.

The screenshot shows the AWS EC2 Load Balancers page. The left sidebar includes sections for Images, Elastic Block Store, Network & Security, and Load Balancing. Under Load Balancing, the 'Load Balancers' section is selected. A table titled 'Load balancers (1/1)' shows one entry: 'Hngo-LB'. The table columns are Name, DNS name, State, VPC ID, Availability Zones, Type, and Date created. The 'Hngo-LB' row has a green 'Active' status. Below the table, a detailed view of the 'Hngo-LB' load balancer is provided, with tabs for Details, Load balancer ARN, and DNS name info. The 'Details' tab is active, showing the load balancer type (Application), status (Active), scheme (Internet-facing), and various VPC and subnet details. The status bar at the bottom shows the date and time as 10/8/2023.

Figure 11 Hngo-LB (Load balancer) created successfully

As mentioned above, the load balancer will forward to target group Hngo-TG.

The screenshot shows the AWS Management Console with the EC2 service selected. Under the 'Load Balancers' section, a single load balancer named 'Hngo-LB' is listed. The 'Listeners and rules' tab is active, showing one rule for port 80 forwarding to target group 'Hngo-TG' at 100% weight. The 'Security' tab is also visible, showing the attached security group 'ELBSG'.

Figure 12 Hngo-TG linked with Hngo-LB

The ELBSG (Security Group for Load Balancer) is also applied correctly. Details for the configuration will be illustrated in the security group section.

The screenshot shows the AWS Management Console with the EC2 service selected. Under the 'Security Groups' section, the security group 'ELBSG' is selected. The 'Details' tab is active, showing the security group ID 'sg-0ee0952c4bacd0aa6' and name 'ELBSG', with a description 'Access for Load Balancer'. The 'Listeners and rules' tab is also visible.

VII. AUTO SCALING

In this stage, we need to create an AMI first to use for the auto scaling group.

The screenshot shows the AWS EC2 console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Images:visibility=owned-by-me>. The left sidebar shows navigation options like Dedicated Hosts, Capacity Reservations, Images (AMIs), Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area displays the 'Amazon Machine Images (AMIs) (1/1) Info' page. It lists a single AMI entry: 'Web Server AMI' with ID 'ami-0bd942f3c712dbfec'. The details pane for this AMI shows it is a 'Web Server AMI' created from '439527679533/Web Server AMI'. The status is 'Available'. The bottom of the screen shows the AWS navigation bar with CloudShell, Feedback, and various icons.

Figure 13 Web Server AMI is created by using DevServer Instance

After that, we will come to create the main part, auto scaling group with the requirement rules.

The screenshot shows the AWS EC2 console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroupsId=Hngo-ASG;view=details>. The left sidebar shows the same navigation options as Figure 13. The main content area displays the 'Auto Scaling groups (1/1) Info' page. It lists one Auto Scaling group named 'Hngo-ASG' using the 'Hngo-LT' launch template. The details pane shows the group name is 'Hngo-ASG', the desired capacity is 2, and the minimum and maximum capacities are both 3. The bottom of the screen shows the AWS navigation bar.

Figure 14 The minimum number of servers is 2. The maximum number of servers is 3

The screenshot shows the AWS EC2 Auto Scaling Groups console. On the left, there's a sidebar with various AWS services like Dedicated Hosts, Capacity Reservations, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The 'Auto Scaling Groups' section is selected. In the main area, it shows 'Auto Scaling groups (1/1) info'. There's a table with one row for 'Hngo-ASG' using 'Hngo-LT | Version Default' with 2 instances. Below this, under 'Auto Scaling group: Hngo-ASG', there's a 'Target Tracking Policy' section with a target of 30 and other configuration details.

Figure 15 Target tracking scaling policy to keep the request count per target of the ELB

The Auto Scaling Group is placed in private subnet 1 and private subnet 2.

VIII. S3 PHOTO STORAGE

In this stage, I have created a s3 bucket also named “hngords”, with a bucket policy that restricts access to a specific HTTP referrer, which is my album website.

The screenshot shows the AWS S3 Buckets console. On the left, there's a sidebar with 'Buckets', 'Storage Lens', and 'Feature spotlight'. The main area shows the 'hngords' bucket under 'Amazon S3 > Buckets > hngords'. The bucket is 'Publicly accessible'. The 'Objects' tab is selected, showing '0 objects'. There are buttons for 'Upload' and 'Actions'. The screenshot also shows the AWS navigation bar and various service links.

Figure 16 S3 bucket is created successfully

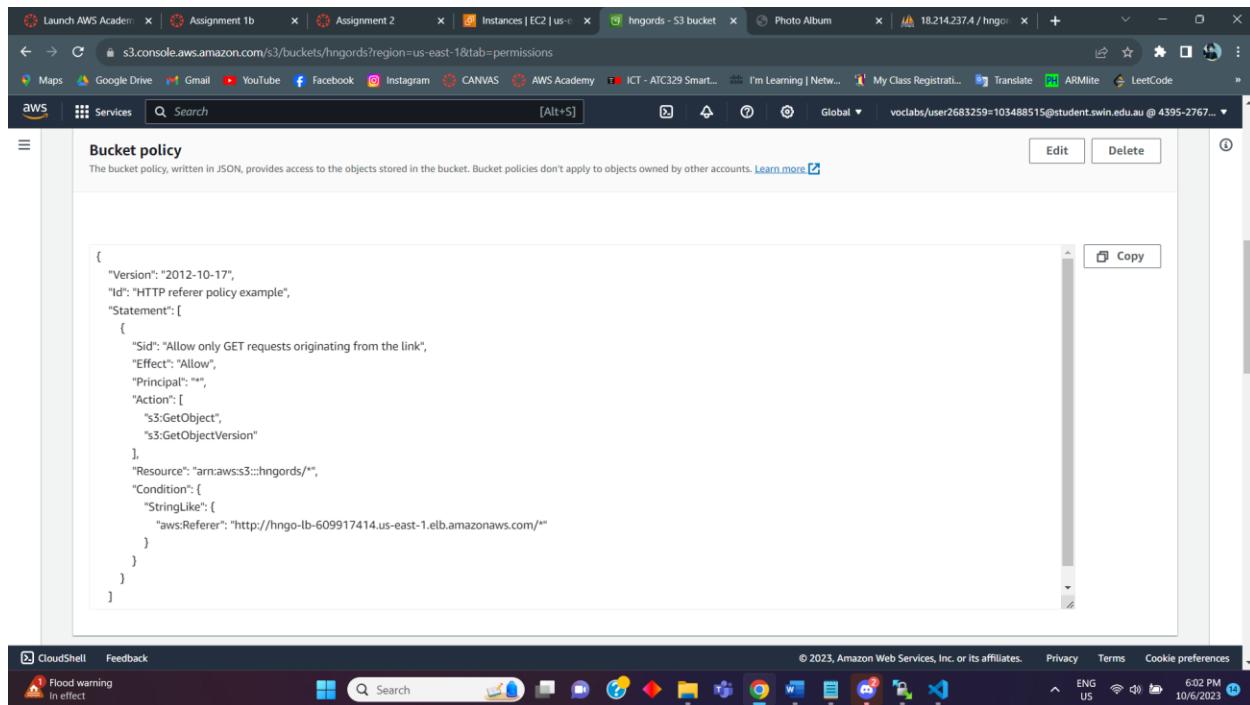


Figure 17 S3 Bucket Policy details

IX. CREATETHUMBNAIL LAMBDA FUNCTION

In this stage, I have created a Lambda function called “CreateThumbnail”. A zip file was uploaded to this lambda function.

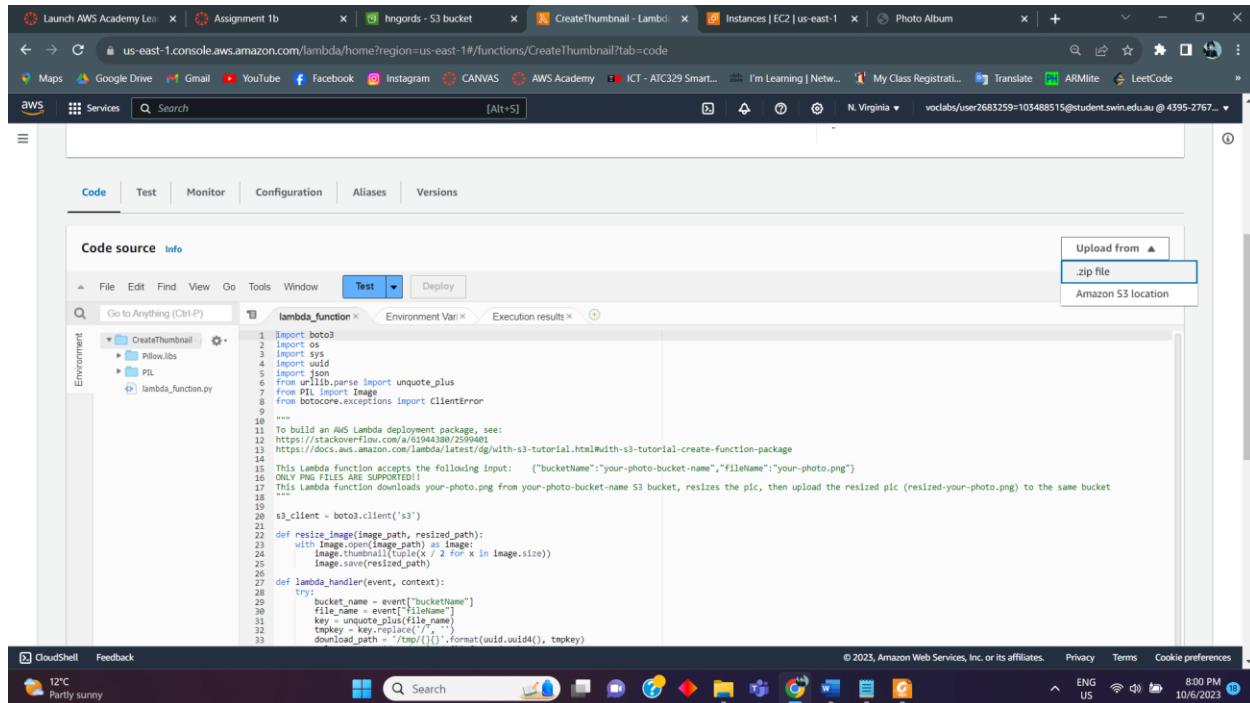


Figure 18 Source code is uploaded successfully

A test function will also be created to test the function in the testing section below.

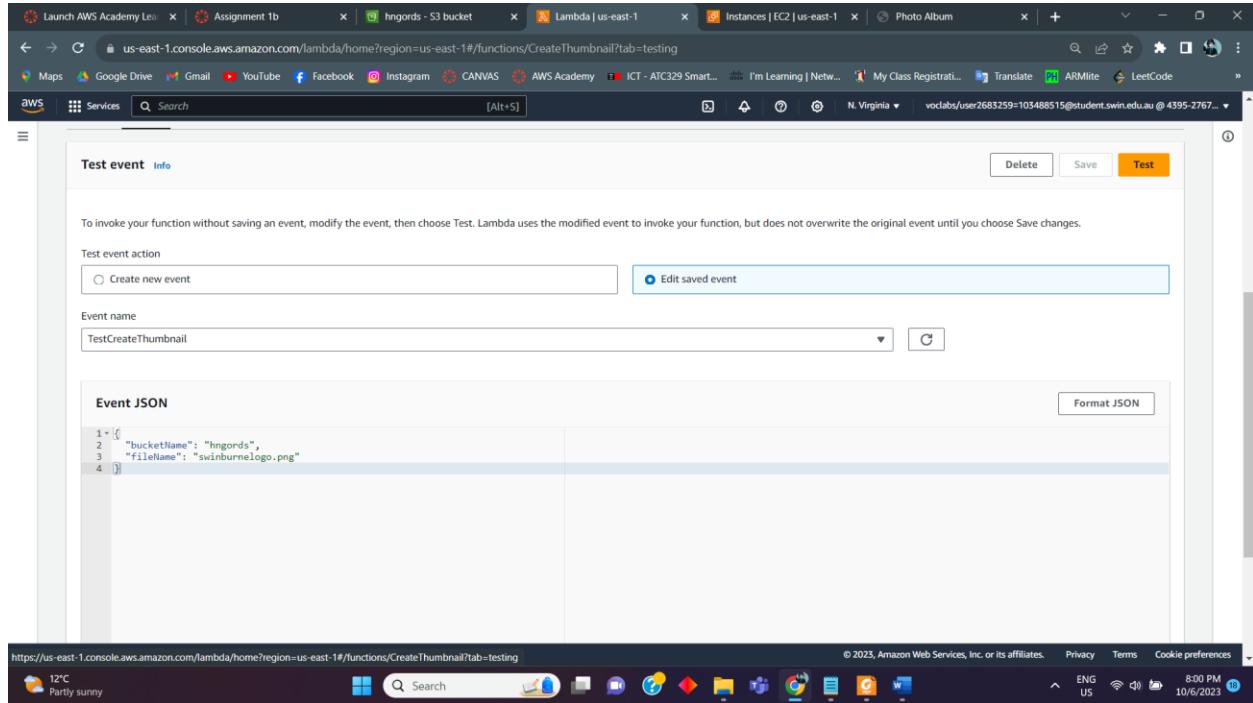


Figure 19 Testing function for s3 and a filename siwnburnelogo.png (please click the test button to see details)

X. SECURITY GROUP

At this stage, the infrastructure has almost been completed, the thing that needs to be tightened will be security group. There are four security groups in total since I used NAT Gateway, so NAT Security group is not necessary. ELBSG, WebServerSG, DBServerSG must follow the least-privilege principle

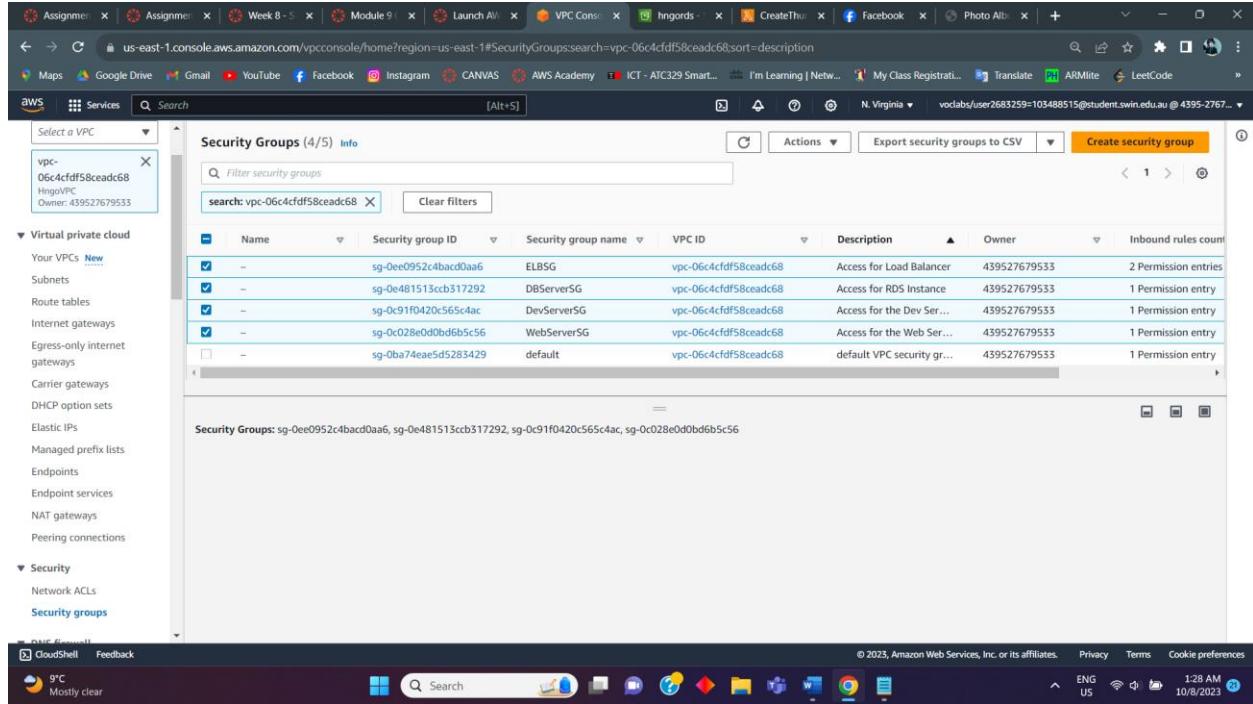


Figure 20 4 Security Groups in HNGoVPC

ELBSG allows HTTP and HTTPS from anywhere. WebServer will take HTTP traffic from ELB, and DBServer will take SQL traffic from the WebServer, making it become a 3-tier security. The DevServerSG allows all traffic from anywhere.

XI. NACL

The screenshot shows the AWS Network ACL Management console. On the left, there's a navigation sidebar with options like 'Virtual private cloud' and 'Security'. The main area displays a table of Network ACLs:

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules count
ac-022f016700a0c6288	2 Subnets		Yes	vpc-06c4cfdf58ceacd68 / HngoVPC	2 Inbound rules
PrivateSubnetsNACL	ac-060f415c954a7c187	2 Subnets	No	vpc-06c4cfdf58ceacd68 / HngoVPC	3 Inbound rules
ac-07cdff0147f2bc59	6 Subnets		Yes	vpc-09e1abb0fa4102817	2 Inbound rules

Below the table, there's a tab for 'Inbound rules' which lists three rules:

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	Allow
110	All ICMP - IPv4	ICMP (1)	All	10.0.2.0/24	Deny
*	All traffic	All	All	0.0.0.0/0	Deny

Figure 21 NACL in private subnet 1 and private subnet 2, allow everything except ICMP from/to DevServer

XII. TESTING

The screenshot shows a web browser window displaying a photo album website. The URL is <http://hngo-lb-609917414.us-east-1.elb.amazonaws.com/photoalbum/album.php>. The page content includes:

- Student name: Hai Nam Ngo
- Student ID: 103488515
- Tutorial session: Friday 04:30PM
- Uploaded photos:

There is a table showing five uploaded photos:

Photo	Name	Description	Creation date	Keywords
	Unimelb	logo for unimelb	2023-10-06	logo
	RMIT	Logo for RMIT	2023-10-06	logo.rmit
	Monash	logo for Monash Uni	2023-10-06	logo.uni.dream
	Vietnam	national flag of VN	2023-10-07	mycountry.home
	Australia	flag of Australia	2023-10-07	flag.national.study

Figure 22 The PhotoAlbum website should be accessible

The swinburnelogo.png file is uploaded directly to S3 bucket to test the lambda function, for more details please come to Lambda Function to specify.

Name	Type	Last modified	Size	Storage class
monashlogo.png	png	October 6, 2023, 20:23:19 (UTC+11:00)	18.2 KB	Standard
resized-monashlogo.png	png	October 6, 2023, 20:23:20 (UTC+11:00)	8.0 KB	Standard
resized-rmitlogo.png	png	October 6, 2023, 20:22:31 (UTC+11:00)	32.2 KB	Standard
resized-swinburnelogo.png	png	October 6, 2023, 20:20:35 (UTC+11:00)	41.2 KB	Standard
resized-unimelblogo.png	png	October 6, 2023, 20:21:59 (UTC+11:00)	586.8 KB	Standard
resized-vietnamflag.png	png	October 7, 2023, 02:04:18 (UTC+11:00)	422.0 B	Standard
rmitlogo.png	png	October 6, 2023, 20:22:29 (UTC+11:00)	57.4 KB	Standard

Figure 23 The S3 bucket to see if photos are uploaded and their resized versions are created

PhotoTitle	Description	CreationDate	Keywords	Reference
Unimelb	logo for unimelb	2023-10-06	logo	https://hngords.s3.amazonaws.com/unimelblogo.png
RMIT	Logo for RMIT	2023-10-06	logo.rmit	https://hngords.s3.amazonaws.com/rmitlogo.png
Monash	logo for Monash Uni	2023-10-06	logo.uni.dream	https://hngords.s3.amazonaws.com/monashlogo.png
Vietnam	national flag of VN	2023-10-07	mycountry/home	https://hngords.s3.amazonaws.com/vietnamflag.png
Australia	flag of Australia	2023-10-07	flag.national.study	https://hngords.s3.amazonaws.com/aus.png

Figure 24 Check the database to see if their meta-data is recorded

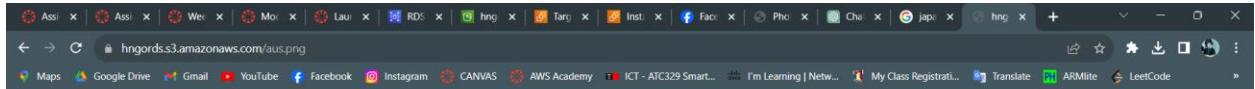
The PhotoAlbum website is accessible through the load balancer only, cannot access to the same way as assignment 1b: <http://18.214.237.4/photoalbum/album.php> (not accessible).

The screenshot shows the AWS Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroups:id=Hngo-ASG&view=activity>. The left sidebar navigation includes services like Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Load Balancers, Target Groups, and Auto Scaling (selected). The main content area displays the Auto Scaling groups (1/1) info table with one entry: Hngo-ASG (Hngo-LT | Version Default), which has 3 instances, a status of - (green), a desired capacity of 2, and a min/max of 2/3. Below this is the Auto Scaling group: Hngo-ASG page, specifically the Activity history (30) section. It lists three events: 1. A successful launch of a new EC2 instance (i-0eb0ef5d772b40bf4) at 2023-10-07T14:54:04Z in response to an unhealthy instance needing to be replaced. 2. A connection draining in progress for instance i-075c99212a9caf39, indicating it has been terminated or stopped. 3. A successful launch of a new EC2 instance (i-0aacf987e053ad2d0) at 2023-10-07T09:09:30Z in response to an unhealthy instance needing to be replaced.

Figure 25 ASG launched a new WebServers Instance after user terminated a WebServers Instance earlier

The screenshot shows the AWS Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#TargetGroups>. The left sidebar navigation includes services like Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Load Balancers, Target Groups (selected), and Auto Scaling. The main content area displays the Target groups (1/1) info table with one entry: Hngo-TG (arn:aws:elasticloadbalancing:us-east-1:06c4cfdf58cead), which has a port of 80, protocol of HTTP, target type of Instance, load balancer of Hngo-LB, and VPC ID of vpc-06c4cfdf58cead. Below this is the Target group: Hngo-TG page, specifically the Registered targets (2) section. It lists two targets: i-0aacf987e053ad2d0 (WebServers, port 80, zone us-east-1b, health status healthy) and i-0eb0ef5d772b40bf4 (WebServers, port 80, zone us-east-1a, health status healthy).

Figure 26 All EC2 targets are healthy



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
<RequestId>NSJ5A5K03WIKX55</RequestId>
<HostId>jmWYQrS03ZFj3lZtZknDvP1ShyWoqzFBH4bmNYNKztxjKC42HDnrSe4xegvbPSip8I8Y=</HostId>
</Error>
```



Figure 27 Test direct access to S3 photos, which should not be publicly accessible

Access putty.exe by using DevServer and I ping one of the WebServers Instance to see if they can ping each other or not, and the result shows that it is not possible to ping between those instances, which means that the rule is working.

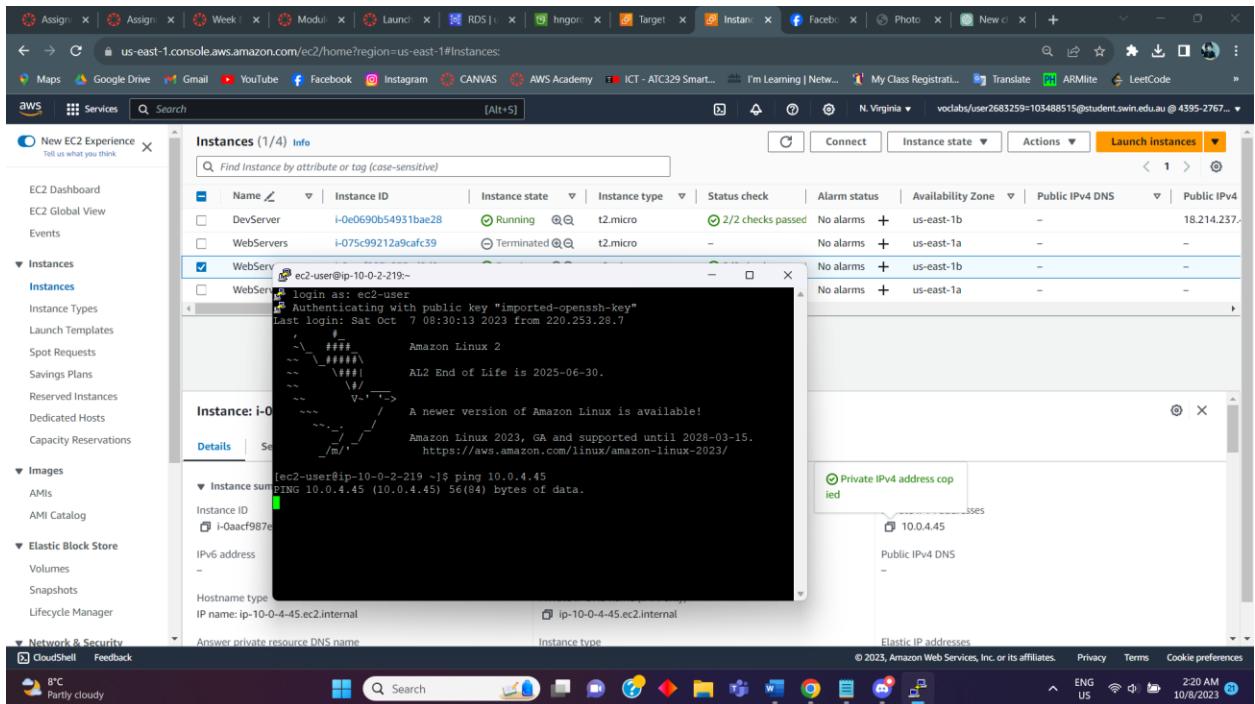


Figure 28 Test the Network ACL bidirectional functionality by sending ICMP traffic between the WebServers and DevServer.