

# **Assignment 2**

## **COS20019**

### **Developing a Highly available Photo Album website**

Student name: Tran Thien Thao Vy

Student ID: 104991221

Swinburne University of Technology

School of Science, Computing and Engineering Technologies

## **I. Introduction**

This paper describes an advanced AWS infrastructure designed to enhance the interactions between EC2, Lambda, and S3 services. Key objectives include the development of a Lambda function, the use of custom AMIs, the implementation of auto scaling with launch configurations, the application of elastic load balancers, and the enforcement of access control through AWS Network Access Control Lists (NACLs) and S3 bucket policies. By achieving these objectives, the infrastructure demonstrates significant improvements in security, scalability, and performance, thereby enabling more effective collaboration among EC2, Lambda, and S3 services within the AWS cloud environment.

Keywords — Cloud Computing, System Architecture, EC2, S3, VPC, Lambda.

## **II. Website Infrastructure**

### **1. Basic Infrastructure requirements:**

The VPC is as per Assignment 1b.

Instead of using the NAT instance, I deploy the NAT gateway as my Learner Lab environment also includes NAT gateway. It is way more convenient yet brings the same effect to the web functionality.

There are two AZs to configure. In specific, NAT is resided in the public-1 subnet, public-2 is for Dev Instance, we can use private-1 and two for auto-scaling webserver which will be employed in the later part of the assignment.

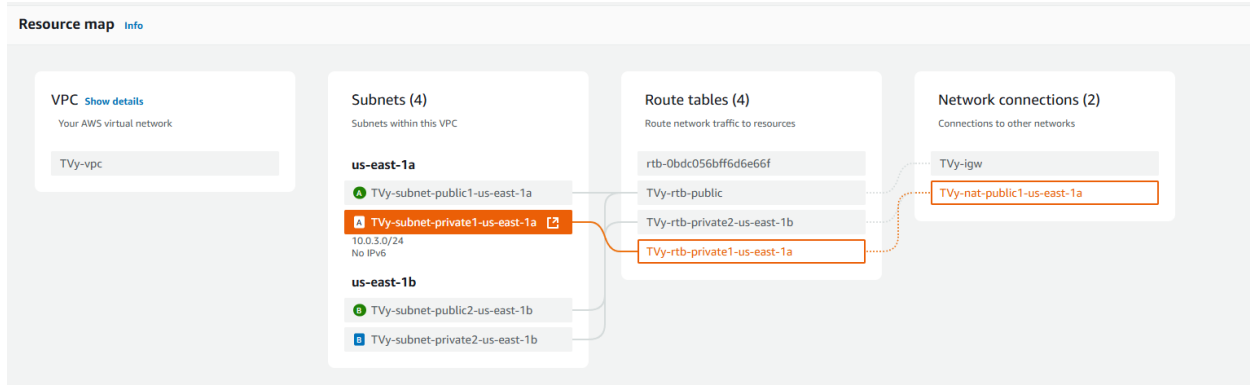


Figure 1: VPC resource map

Make sure that the NAT gateway will route to the Internet Gateway.

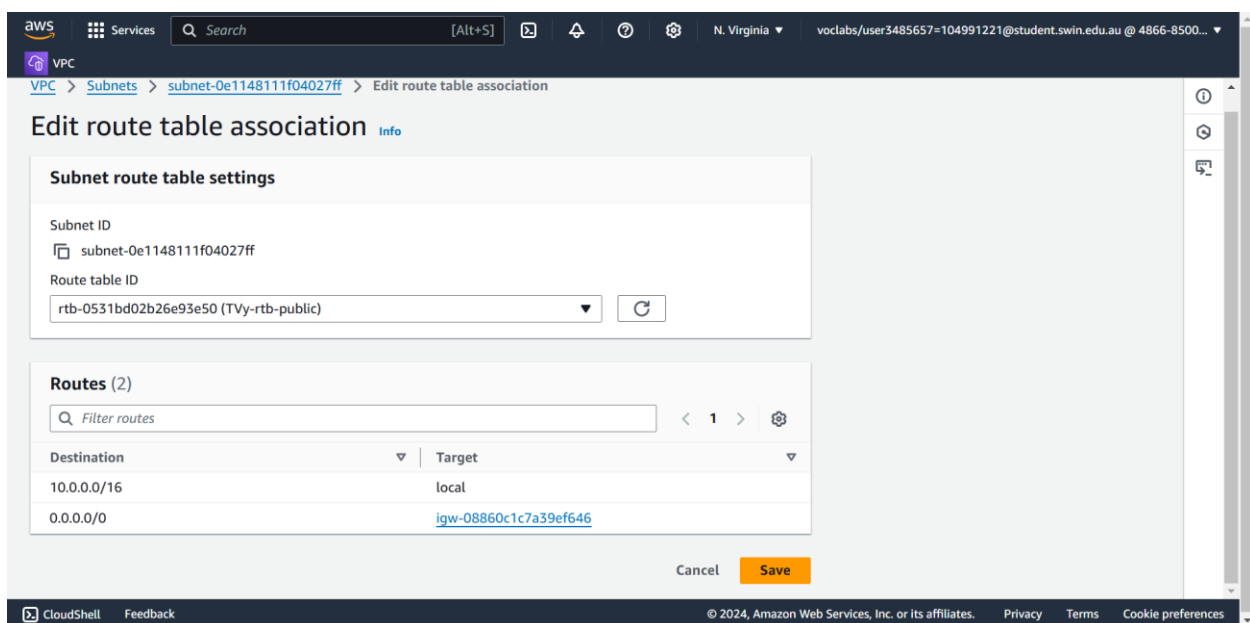


Figure 2: NAT subnet route table association

Subnets (4) Info

Find resources by attribute or tag

VPC: vpc-083893464d4fd7ea5

Name	Subnet ID	State	VPC	IPv4 CIDR
TVy-subnet-public1-us-east-1a	subnet-0e1148111f04027ff	Available	vpc-083893464d4fd7ea5   TVy-...	10.0.1.0/24
TVy-subnet-private2-us-east-1b	subnet-08f222976c1be8563	Available	vpc-083893464d4fd7ea5   TVy-...	10.0.4.0/24
TVy-subnet-public2-us-east-1b	subnet-0f9e0540dedcb225d	Available	vpc-083893464d4fd7ea5   TVy-...	10.0.2.0/24
TVy-subnet-private1-us-east-1a	subnet-00032c72c07254358	Available	vpc-083893464d4fd7ea5   TVy-...	10.0.3.0/24

Figure 3: subnets of TVy-VPC

Continue to configure the Dev Instance, which stores most AWS PHP SDK (configured by SSH), Apache web server (configured by user-data), source code of the website (configured by transferred through WinSCP).

Instance summary for i-0505c86b87f973cb1 (Dev Instance) <a href="#">Info</a>		
Updated less than a minute ago		
Instance ID i-0505c86b87f973cb1	Public IPv4 address 54.174.120.228   <a href="#">open address</a>	Private IPv4 addresses 10.0.1.64
	Instance state <span>Running</span>	Public IPv4 DNS ec2-54-174-120-228.compute-1.amazonaws.com   <a href="#">open address</a>
IPv6 address -	Private IP DNS name (IPv4 only) ip-10-0-1-64.ec2.internal	Elastic IP addresses 54.174.120.228 [Public IP]
Hostname type IP name: ip-10-0-1-64.ec2.internal	Instance type t2.micro	AWS Compute Optimizer finding <a href="#">Opt-in to AWS Compute Optimizer for recommendations.</a>   <a href="#">Learn more</a>
Answer private resource DNS name -	VPC ID vpc-083893464d4fd7ea5 (TVy-vpc)	Auto Scaling Group name -
Auto-assigned IP address -	Subnet ID subnet-0e114811f04027ff (TVy-subnet-public1-us-east-1a)	
IAM Role LabRole	Instance ARN arn:aws:ec2:us-east-1:486685003184:instance/i-0505c86b87f973cb1	
IMDSv2 Required		

Figure 4: Dev Instance configuration

Make sure the public DNS and Apache works by access to the browser through public DNS.

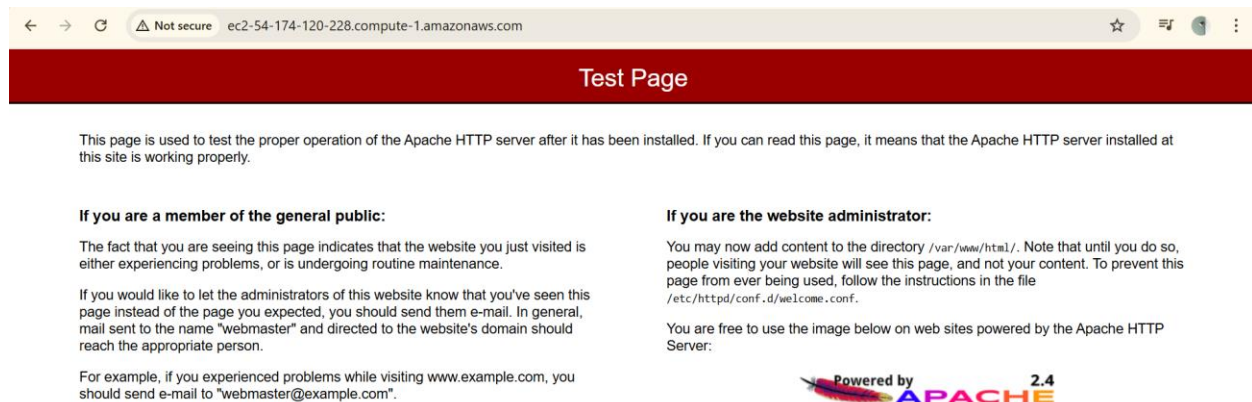
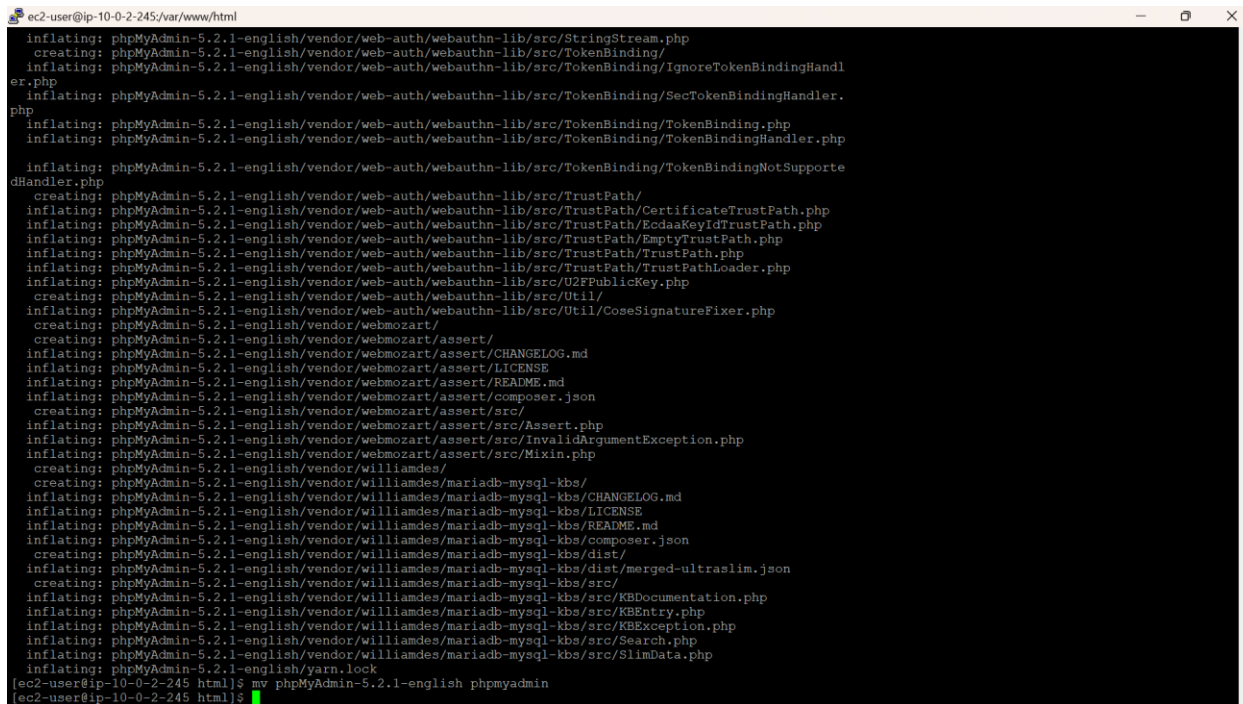


Figure 5: Test page of Dev instance

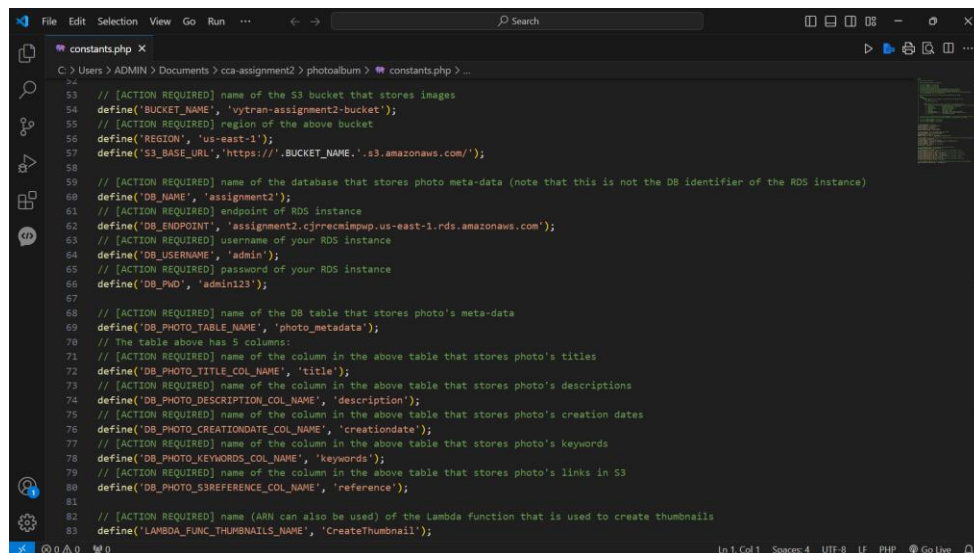
As things are settled, we can continue to configure the database section. The first thing is to SSH to Dev Instance and download “phpmyadmin” by following the same instruction as per assignment 1a.



```
ec2-user@ip-10-0-2-245:var/www/html
inflating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/StringStream.php
creating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/TokenBinding/
inflating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/TokenBinding/IgnoreTokenBindingHandler.php
inflating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/TokenBinding/SecTokenBindingHandler.php
inflating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/TokenBinding/TokenBinding.php
inflating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/TokenBinding/TokenBindingHandler.php
inflating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/TokenBinding/TokenBindingNotSupportedHandler.php
creating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/TrustPath/
inflating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/TrustPath/CertificateTrustPath.php
inflating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/TrustPath/EcdsaKeyIdTrustPath.php
inflating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/TrustPath/EmptyTrustPath.php
inflating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/TrustPath/TrustPath.php
inflating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/TrustPath/TrustPathLoader.php
inflating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/U2FPublicKey.php
creating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/Util/
inflating: phpMyAdmin-5.2.1-english/vendor/web-auth/webauthn-lib/src/Util/CoseSignatureFixer.php
creating: phpMyAdmin-5.2.1-english/vendor/webmozart/assert/
inflating: phpMyAdmin-5.2.1-english/vendor/webmozart/assert/CHANGELOG.md
inflating: phpMyAdmin-5.2.1-english/vendor/webmozart/assert/LICENSE
inflating: phpMyAdmin-5.2.1-english/vendor/webmozart/assert/README.md
inflating: phpMyAdmin-5.2.1-english/vendor/webmozart/assert/composer.json
creating: phpMyAdmin-5.2.1-english/vendor/webmozart/assert/src/
inflating: phpMyAdmin-5.2.1-english/vendor/webmozart/assert/src/Assert.php
inflating: phpMyAdmin-5.2.1-english/vendor/webmozart/assert/src/InvalidArgumentException.php
inflating: phpMyAdmin-5.2.1-english/vendor/webmozart/assert/src/Mixin.php
creating: phpMyAdmin-5.2.1-english/vendor/williamdes/mariadb-mysql-kbs/
inflating: phpMyAdmin-5.2.1-english/vendor/williamdes/mariadb-mysql-kbs/CHANGELOG.md
inflating: phpMyAdmin-5.2.1-english/vendor/williamdes/mariadb-mysql-kbs/LICENSE
inflating: phpMyAdmin-5.2.1-english/vendor/williamdes/mariadb-mysql-kbs/README.md
inflating: phpMyAdmin-5.2.1-english/vendor/williamdes/mariadb-mysql-kbs/composer.json
creating: phpMyAdmin-5.2.1-english/vendor/williamdes/mariadb-mysql-kbs/dist/
inflating: phpMyAdmin-5.2.1-english/vendor/williamdes/mariadb-mysql-kbs/dist/merged-ultraslim.json
creating: phpMyAdmin-5.2.1-english/vendor/williamdes/mariadb-mysql-kbs/src/
inflating: phpMyAdmin-5.2.1-english/vendor/williamdes/mariadb-mysql-kbs/src/KBDocumentation.php
inflating: phpMyAdmin-5.2.1-english/vendor/williamdes/mariadb-mysql-kbs/src/KBEntry.php
inflating: phpMyAdmin-5.2.1-english/vendor/williamdes/mariadb-mysql-kbs/src/KBException.php
inflating: phpMyAdmin-5.2.1-english/vendor/williamdes/mariadb-mysql-kbs/src/Search.php
inflating: phpMyAdmin-5.2.1-english/vendor/williamdes/mariadb-mysql-kbs/src/SlimData.php
inflating: phpMyAdmin-5.2.1-english/yarn.lock
[ec2-user@ip-10-0-2-245 html]$ mv phpMyAdmin-5.2.1-english phpmyadmin
[ec2-user@ip-10-0-2-245 html]$
```

Figure 6: PuTTY screen of phpmyadmin download

After that, we need to specialize some data in constants.file.



```
File Edit Selection View Go Run ... Search
C:\Users\ADMIN\Documents> cca-assignment2 > photoalbum > constants.php > ...
24
25 // [ACTION REQUIRED] name of the S3 bucket that stores images
26 define('BUCKET_NAME', 'vytran-assignment2-bucket');
27
28 // [ACTION REQUIRED] region of the above bucket
29 define('REGION', 'us-east-1');
30
31 define('S3_BASE_URL', 'https://'.BUCKET_NAME.'.s3.amazonaws.com/');
32
33
34 // [ACTION REQUIRED] name of the database that stores photo meta-data (note that this is not the DB identifier of the RDS instance)
35 define('DB_NAME', 'assignment2');
36
37 // [ACTION REQUIRED] endpoint of RDS instance
38 define('DB_ENDPOINT', 'assignment2.cjrcemimpp.us-east-1.rds.amazonaws.com');
39
40 // [ACTION REQUIRED] username of your RDS instance
41 define('DB_USERNAME', 'admin');
42
43 // [ACTION REQUIRED] password of your RDS instance
44 define('DB_PWD', 'admin123');
45
46
47 // [ACTION REQUIRED] name of the DB table that stores photo's meta-data
48 define('DB_PHOTO_TABLE_NAME', 'photo_metadata');
49
50 // The table above has 5 columns:
51 // [ACTION REQUIRED] name of the column in the above table that stores photo's titles
52 define('DB_PHOTO_TITLE_COL_NAME', 'title');
53
54 // [ACTION REQUIRED] name of the column in the above table that stores photo's descriptions
55 define('DB_PHOTO_DESCRIPTION_COL_NAME', 'description');
56
57 // [ACTION REQUIRED] name of the column in the above table that stores photo's creation dates
58 define('DB_PHOTO_CREATIONDATE_COL_NAME', 'creationdate');
59
60 // [ACTION REQUIRED] name of the column in the above table that stores photo's keywords
61 define('DB_PHOTO_KEYWORDS_COL_NAME', 'keywords');
62
63 // [ACTION REQUIRED] name of the column in the above table that stores photo's links in S3
64 define('DB_PHOTO_S3REFERENCE_COL_NAME', 'reference');
65
66
67 // [ACTION REQUIRED] name (ARN can also be used) of the Lambda function that is used to create thumbnails
68 define('LAMBDA_FUNC_THUMBNAILS_NAME', 'CreateThumbnail');
```

Log into phpMyAdmin page through Dev Instance EC2 DNS and create the photos' metadata table.

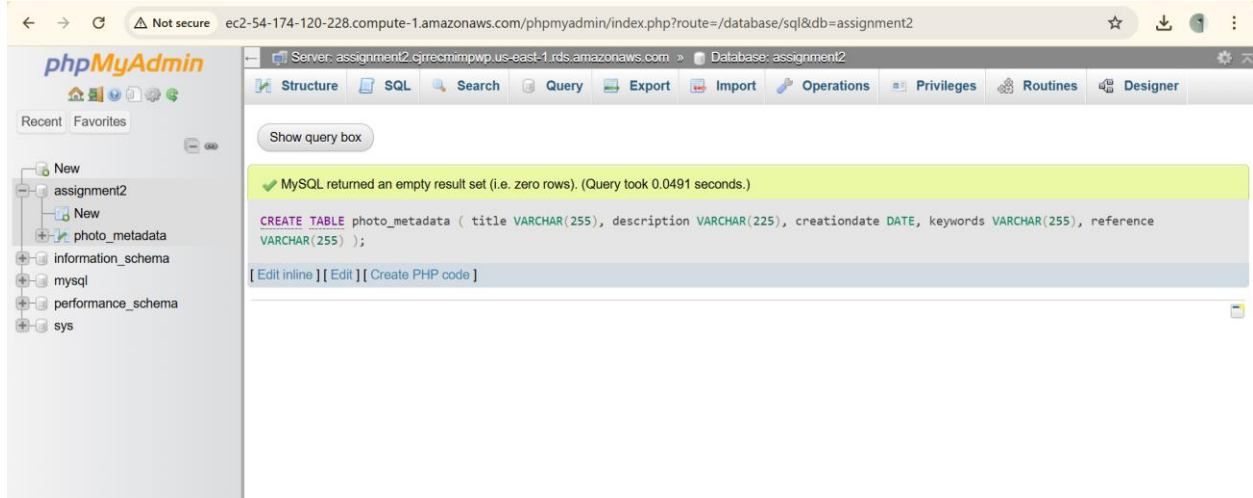


Figure 7: phpMyAdmin screen

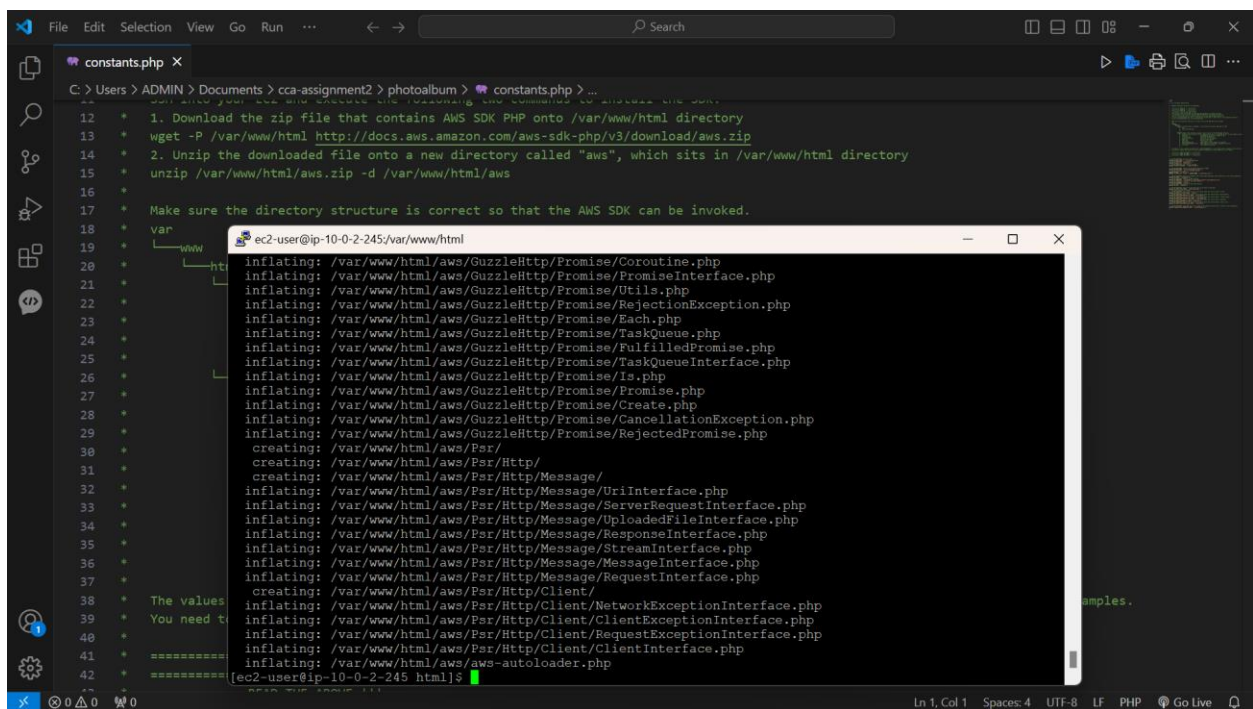


Figure 8: AWS SDK package through SSH

For the website to work, we need to download AWS SDK package through SSH into the Dev Instance. Then transfer the files to WinSCP.

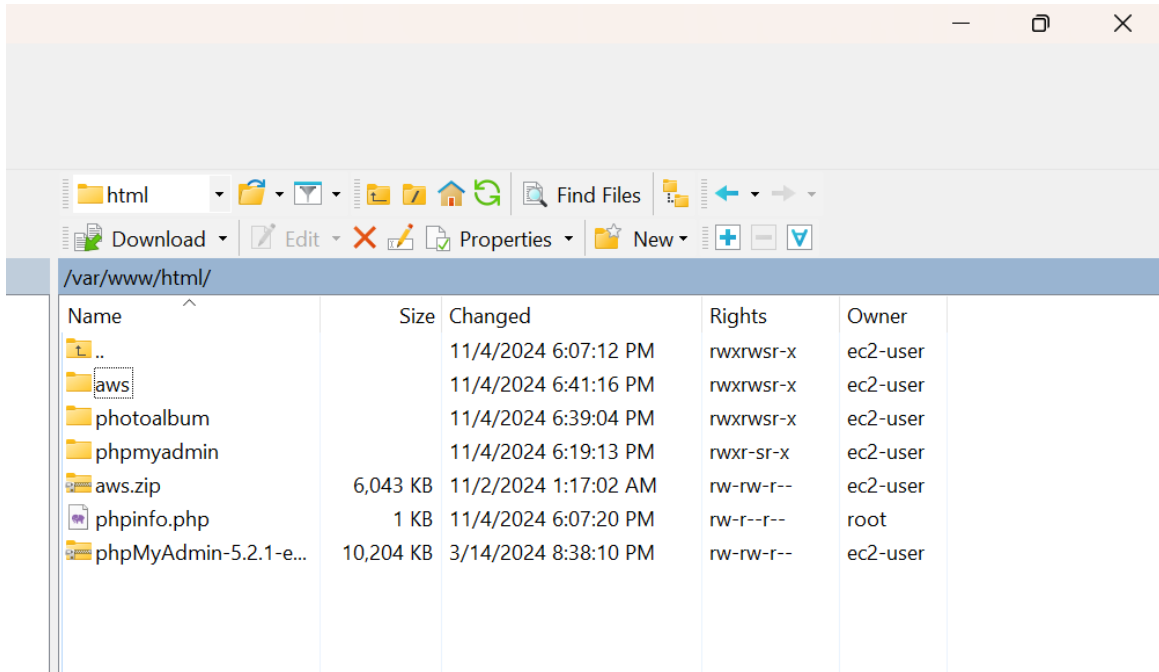


Figure 9: WinSCP directories in the correct arrangement.

After the photoalbum php files and AWS SDK files and phpMyAdmin are settled in WinSCP. We can connect to the website through Dev Instance public DNS (make sure the instance is associated with an elastic IP address for further use).



Figure 10: Website screen through ec2 public DNS access.

← → ↻ ⚠ Not secure ec2-54-174-120-228.compute-1.amazonaws.com/photoalbum/album.php ☆ ⬇ 🌐 ⋮

**Student name:** Tran Thien Thao Vy

**Student ID:** 104991221

**Tutorial session:** Friday 07:15AM

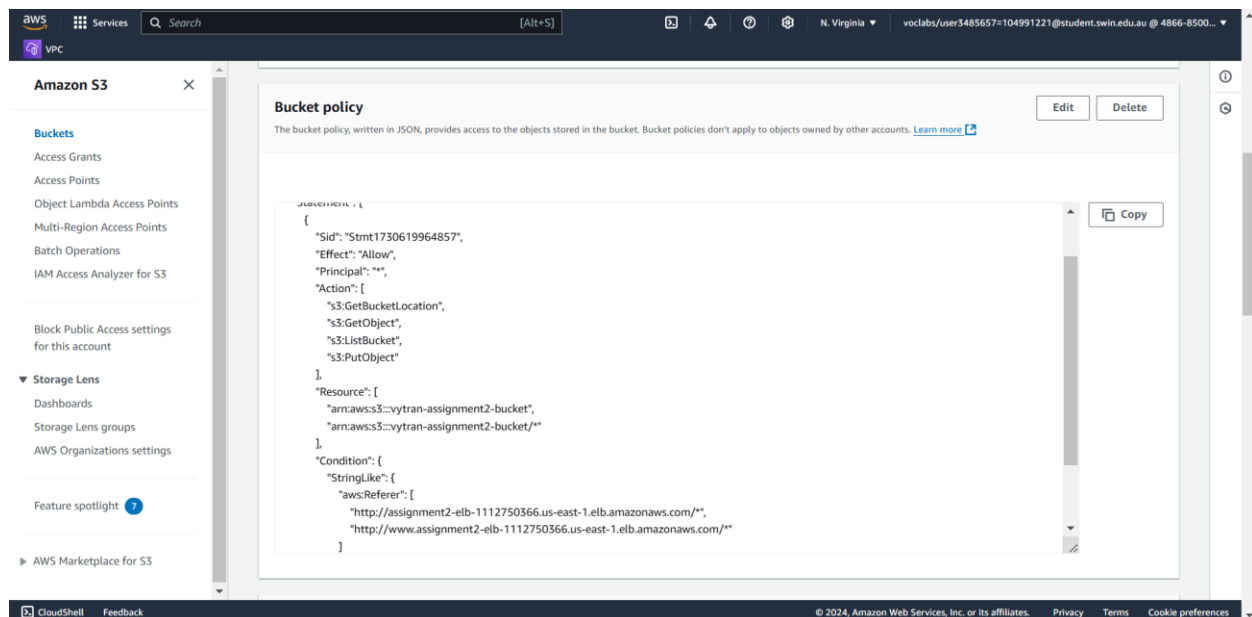
**Uploaded photos:**

[Upload more photos](#)

Photo	Name	Description	Creation date	Keywords
	milk tea	drink	2024-11-04	milk, tea

Figure 11: Meta-data polulated.

S3 is used for photo storage. It is built the same way as Assignment 1b, however we need to make sure that bucket policy and permissions are modified correctly.



The screenshot shows the AWS Management Console interface for the 'Bucket policy' of an S3 bucket named 'vytran-assignment2-bucket'. The left sidebar shows the 'Amazon S3' service with various options like Buckets, Access Grants, and Storage Lens. The main panel displays the bucket policy in JSON format, which grants 'Allow' permissions for actions like 's3:GetBucketLocation', 's3:GetObject', 's3:ListBucket', and 's3:PutObject' to the principal '\*'. The condition restricts access to requests from the specific ELB endpoints mentioned in the text.

```

{
  "Sid": "Stmt1730619964857",
  "Effect": "Allow",
  "Principal": "*",
  "Action": [
    "s3:GetBucketLocation",
    "s3:GetObject",
    "s3:ListBucket",
    "s3:PutObject"
  ],
  "Resource": [
    "arn:aws:s3:::vytran-assignment2-bucket",
    "arn:aws:s3:::vytran-assignment2-bucket/*"
  ],
  "Condition": {
    "StringLike": {
      "aws:Referer": [
        "http://assignment2-elb-1112750366.us-east-1.elb.amazonaws.com/*",
        "http://www.assignment2-elb-1112750366.us-east-1.elb.amazonaws.com/*"
      ]
    }
  }
}

```

Figure 12: S3 bucket policy

After configuring Lambda function by uploading the zip file and configure its IAM role, we can run a test to check if S3 and Lambda are working well.



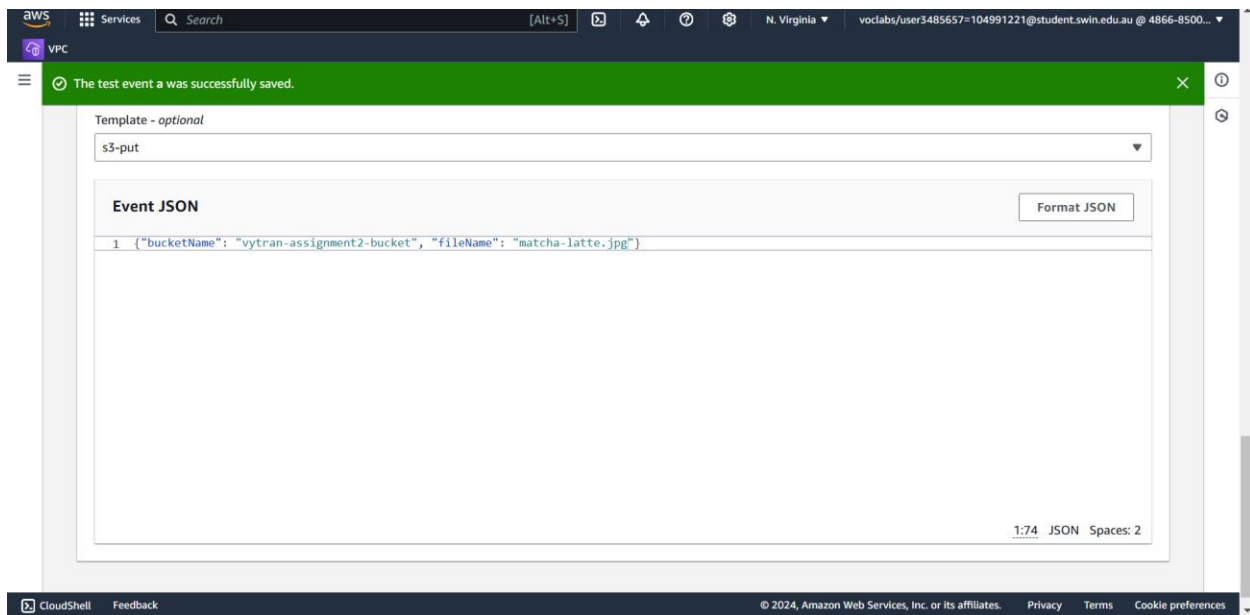


Figure 13: Lambda code to test a case

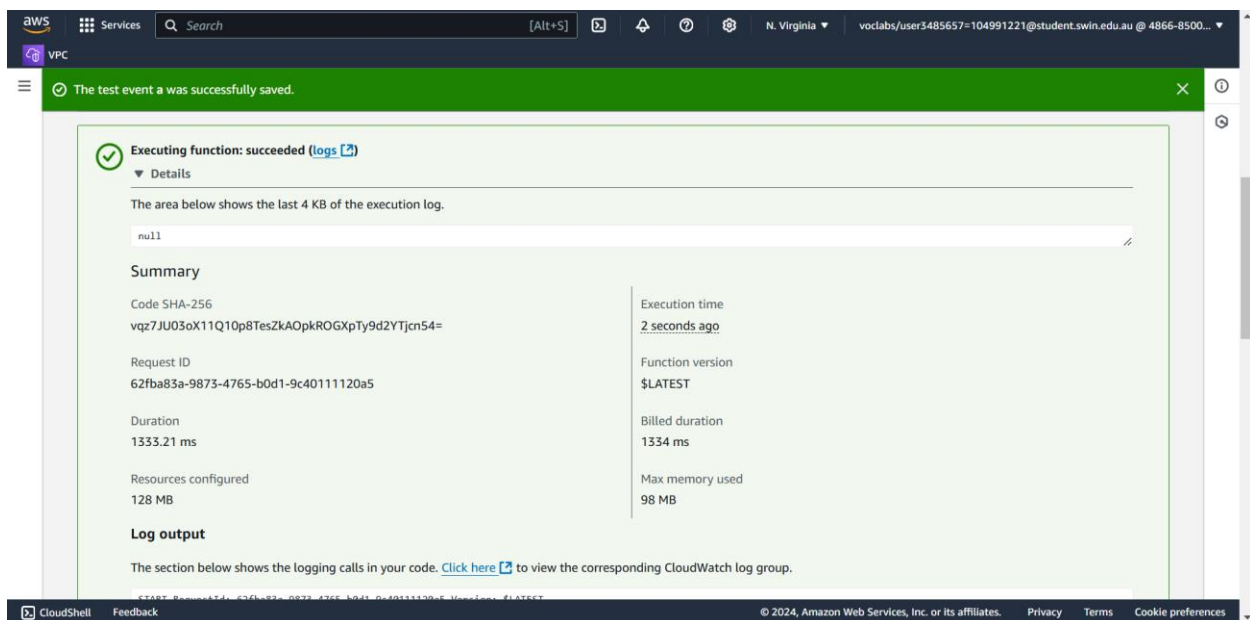


Figure 14: test succeeded.

Therefore, the meta-data of the “milk tea” object photo will be a resized jpg.



<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">chocolate-cookies.jpg</a>	jpg	November 3, 2024, 16:41:06 (UTC+07:00)	60.8 KB	Standard
<input type="checkbox"/>	<a href="#">drink-1.png</a>	png	November 3, 2024, 16:55:12 (UTC+07:00)	18.8 KB	Standard
<input type="checkbox"/>	<a href="#">matcha-latte.jpg</a>	jpg	November 3, 2024, 15:33:54 (UTC+07:00)	35.0 KB	Standard
<input type="checkbox"/>	<a href="#">milk-tea.jpg</a>	jpg	November 4, 2024, 18:50:21 (UTC+07:00)	52.0 KB	Standard
<input type="checkbox"/>	<a href="#">resized-chocolate-cookies.jpg</a>	jpg	November 3, 2024, 16:41:08 (UTC+07:00)	15.8 KB	Standard
<input type="checkbox"/>	<a href="#">resized-drink-1.png</a>	png	November 3, 2024, 16:55:14 (UTC+07:00)	6.6 KB	Standard
<input type="checkbox"/>	<a href="#">resized-matcha-latte.jpg</a>	jpg	November 3, 2024, 15:49:46 (UTC+07:00)	10.0 KB	Standard
<input type="checkbox"/>	<a href="#">resized-milk-tea.jpg</a>	jpg	November 4, 2024, 18:50:23 (UTC+07:00)	14.4 KB	Standard

Figure 15: S3 objects.

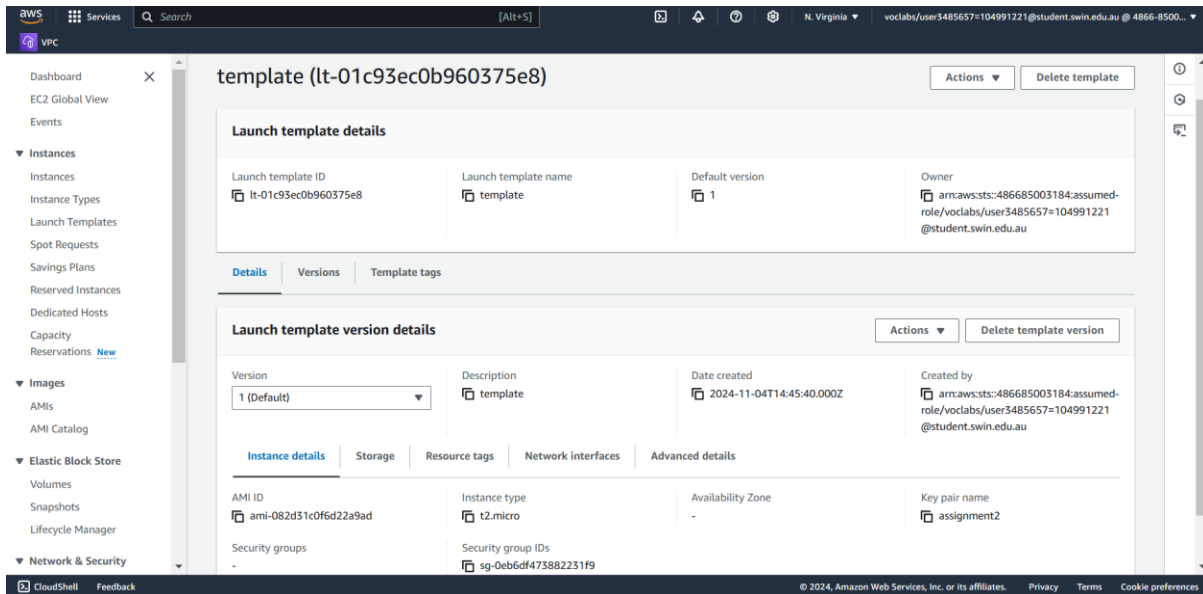
After the Dev Instance configuration, which works properly, we can continue create the custom AMI, which can be used for the web servers.

<input type="checkbox"/>	Name	AMI name	AMI ID	Source	Owner	Visibility
<input type="checkbox"/>	web-ami		ami-082d31c0f6d22a9ad	486685003184/web-ami	486685003184	Private

Figure 16: custom AMI for web servers.

## 2. Developing a highly available website by configuring a load balancer and an auto scaling group.

We firstly need to launch a template for the web servers, in which the template has to employ the customized AMI.



Details of the launched template.

After that, we need to configure a load balancer and an auto scaling group as well.

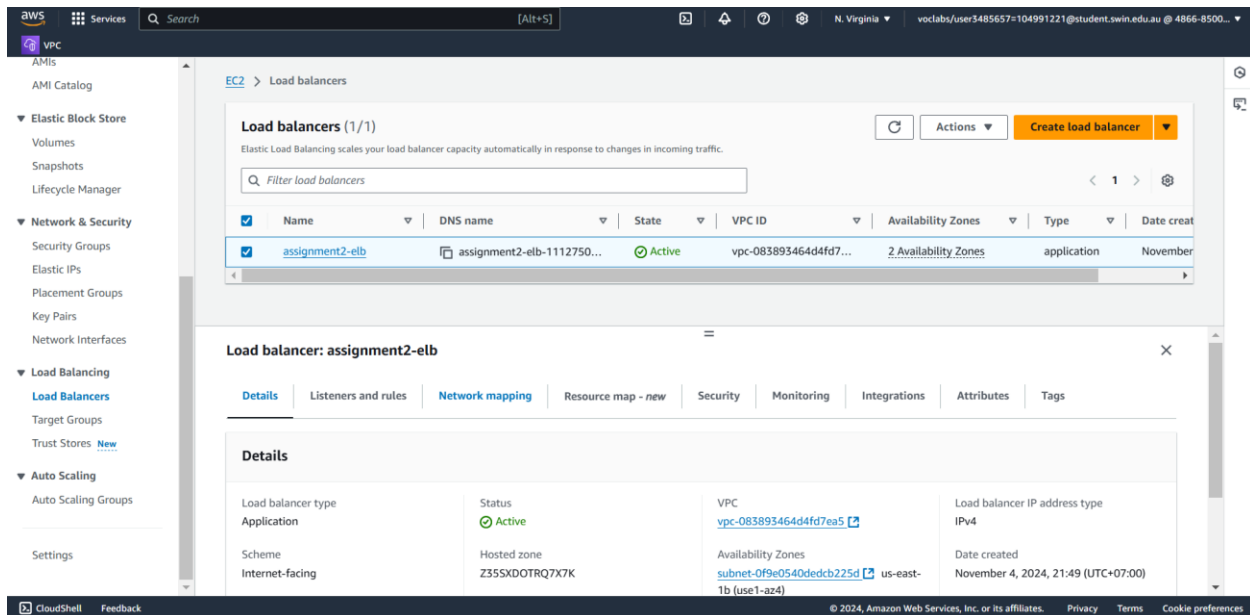


Figure 18: load balancer for web servers.

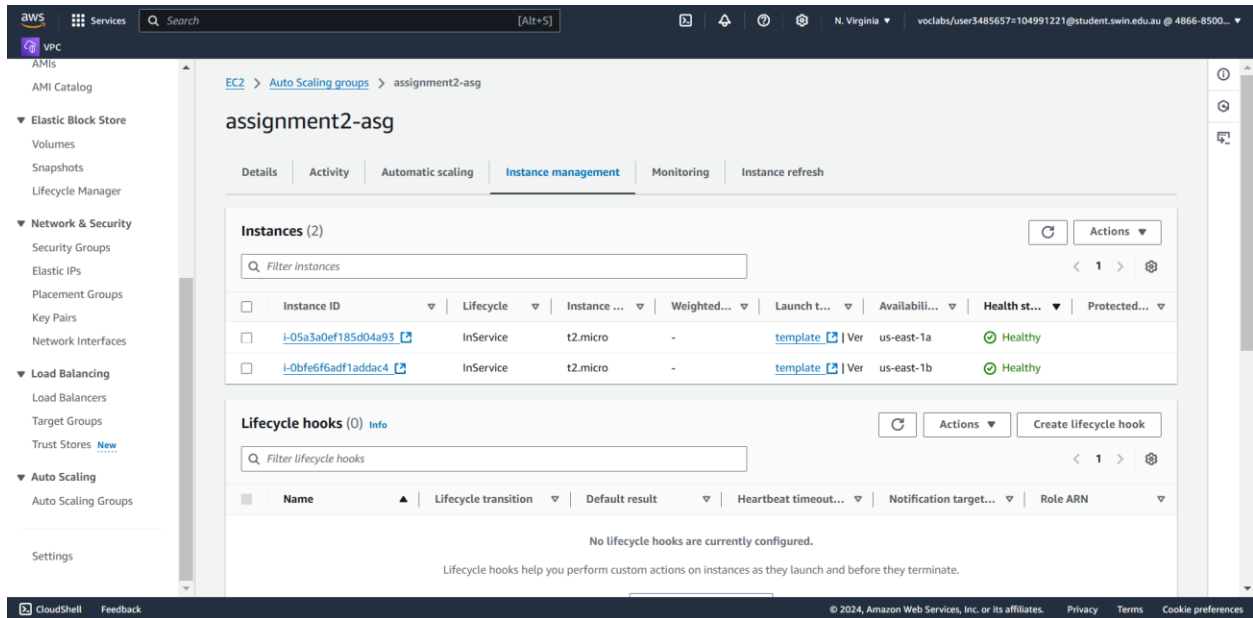


Figure 19: Make sure the registered targets are healthy.

After all configurations, which are supposing-ly correct, we can access the website through the ELB public ARN.

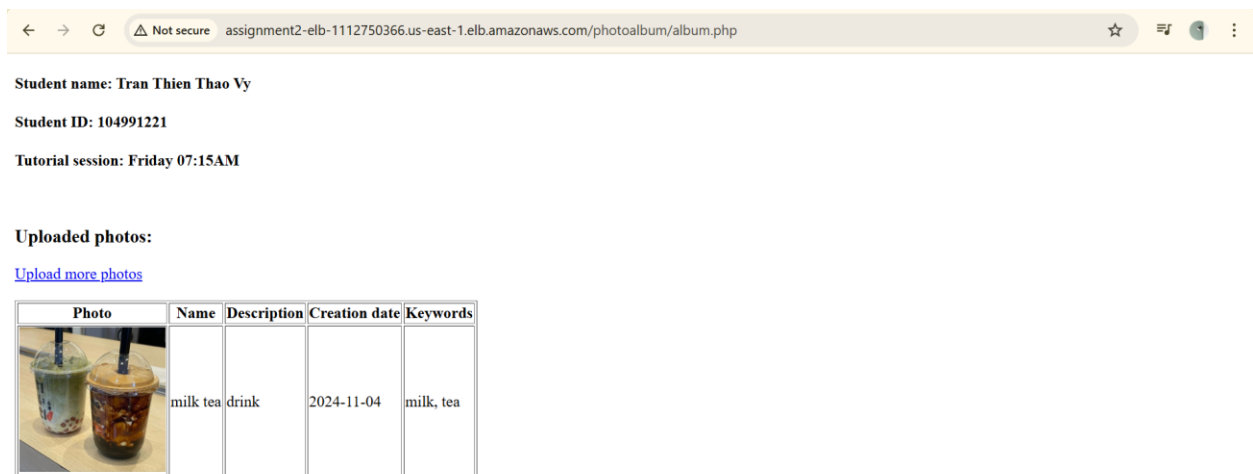


Figure 20: album.php Webpage

We can upload photos and photos' meta-data in the uploader page.

← → ↻ ⚠ Not secure assignment2-elb-1112750366.us-east-1.elb.amazonaws.com/photoalbum/photouploader.php ☆ ⬇ 🌐 ⋮

## Photo uploader

Photo title:

Select a photo (Select PNG file for best result):  chocolate-cake.jpg

Description:

Date:

Keywords (comma-delimited, e.g. keyword1, keyword2, ...):

[Photo Album](#)

Figure 21: photouploader.php webpage

After uploading some photos, here is the webpage.

← → ↻ ⚠ Not secure assignment2-elb-1112750366.us-east-1.elb.amazonaws.com/photoalbum/album.php ☆ ⬇ 🌐 ⋮

Student name: Tran Thien Thao Vy

Student ID: 104991221

Tutorial session: Friday 07:15AM

Uploaded photos:

[Upload more photos](#)



Photo	Name	Description	Creation date	Keywords
	milk tea	drink	2024-11-04	milk, tea
	chocolate cake	sweet	2024-11-04	chocolate, cake

Figure 21: album.php webpage









<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	 <a href="#">drink-1.png</a>	png	November 3, 2024, 16:55:12 (UTC+07:00)	18.8 KB	Standard
<input type="checkbox"/>	 <a href="#">matcha-latte.jpg</a>	jpg	November 3, 2024, 15:33:54 (UTC+07:00)	35.0 KB	Standard
<input type="checkbox"/>	 <a href="#">milk-tea.jpg</a>	jpg	November 4, 2024, 18:50:21 (UTC+07:00)	52.0 KB	Standard
<input type="checkbox"/>	 <a href="#">resized-chocolate-cake.jpg</a>	jpg	November 4, 2024, 21:59:35 (UTC+07:00)	16.4 KB	Standard
<input type="checkbox"/>	 <a href="#">resized-chocolate-cookies.jpg</a>	jpg	November 3, 2024, 16:41:08 (UTC+07:00)	15.8 KB	Standard
<input type="checkbox"/>	 <a href="#">resized-drink-1.png</a>	png	November 3, 2024, 16:55:14 (UTC+07:00)	6.6 KB	Standard
<input type="checkbox"/>	 <a href="#">resized-matcha-latte.jpg</a>	jpg	November 3, 2024, 15:49:46 (UTC+07:00)	10.0 KB	Standard
<input type="checkbox"/>	 <a href="#">resized-milk-tea.jpg</a>	jpg	November 4, 2024, 18:50:23 (UTC+07:00)	14.4 KB	Standard

Figure 22: S3 bucket objects

The jpg photo data are resized after being forwarded to S3 by the Lambda function from the EC2 instance.

Security Groups (5) <a href="#">Info</a>					<a href="#">Actions</a>	<a href="#">Export security groups to CSV</a>
<input type="text" value="Find resources by attribute or tag"/>						
VPC ID = <a href="#">vpc-083893464d4fd7ea5</a> <input type="button" value="X"/>					<input type="button" value="Clear filters"/>	
<input type="checkbox"/>	Name	Security group ID	Security group name	VPC ID		
<input type="checkbox"/>	-	<a href="#">sg-0eb6df473882231f9</a>	WebTier-SG	<a href="#">vpc-083893464d4fd7ea5</a>		
<input type="checkbox"/>	-	<a href="#">sg-09b5f49075b01aeee</a>	ELB-SG	<a href="#">vpc-083893464d4fd7ea5</a>		
<input type="checkbox"/>	-	<a href="#">sg-0739426ac037c9189</a>	DevServer-SG	<a href="#">vpc-083893464d4fd7ea5</a>		
<input type="checkbox"/>	-	<a href="#">sg-03063fbf033a19f5c</a>	default	<a href="#">vpc-083893464d4fd7ea5</a>		
<input type="checkbox"/>	-	<a href="#">sg-030aa3387891bb429</a>	DBServer-SG	<a href="#">vpc-083893464d4fd7ea5</a>		

Rules of security groups are configure as following:

- DevServer-SG: Inbounds from all traffic and outbounds to all traffic.
- WebTier-SG: Inbounds from ELB-SG and outbound to the NAT gateway.
  - ELB-SG: Inbounds from Internet gateway and outbounds to IGW.
- DBServer-SG: Inbounds from/outbounds to WebTier and DevServer.

We can test the high availability of the instances by terminating some instances and the other instances will be initialized and the website will continue to work:

Successfully initiated termination (deletion) of i-0f1bc9858c7da7a06

Instances (1/7) Info

Find Instance by attribute or tag (case-sensitive)

Instance state = running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	Bastion/Web s...	i-0448603706198257e	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1b	ec2-3-223-28-194.com
<input type="checkbox"/>	Test Instance	i-00bcaf002027e1244	Running	t2.nano	2/2 checks passec	View alarms +	us-east-1b	ec2-107-21-120-10.co.
<input type="checkbox"/>	Dev Instance	i-0639be31cfd95f46b	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1b	ec2-54-174-120-228.cc
<input checked="" type="checkbox"/>		i-0f1bc9858c7da7a06	Shutting-d...	t2.micro	2/2 checks passec	View alarms +	us-east-1b	-
<input type="checkbox"/>		i-0bfe6f6adf1addac4	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1b	-
<input type="checkbox"/>		i-0153d29ed522d229e	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	-
<input type="checkbox"/>	Assignment 1a...	i-036b4b8da91ca4da	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1c	ec2-34-229-46-93.com

i-0f1bc9858c7da7a06

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Inst...

Instances (9) Info

Find Instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	Bastion/Web s...	i-0448603706198257e	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1b	ec2-3-223-28-194.com
<input type="checkbox"/>	Test Instance	i-00bcaf002027e1244	Running	t2.nano	2/2 checks passec	View alarms +	us-east-1b	ec2-107-21-120-10.co.
<input type="checkbox"/>	Dev Instance	i-0639be31cfd95f46b	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1b	ec2-54-174-120-228.cc
<input type="checkbox"/>		i-0f1bc9858c7da7a06	Terminated	t2.micro	-	View alarms +	us-east-1b	-
<input type="checkbox"/>		i-00e49b30770e3318d	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1b	-
<input type="checkbox"/>		i-0bfe6f6adf1addac4	Terminated	t2.micro	-	View alarms +	us-east-1b	-
<input type="checkbox"/>		i-05a3a0ef185d04a93	Terminated	t2.micro	-	View alarms +	us-east-1a	-
<input type="checkbox"/>		i-0153d29ed522d229e	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	-
<input type="checkbox"/>	Assignment 1a...	i-036b4b8da91ca4da	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1c	ec2-34-229-46-93.com

Select an instance

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Inst...



Student name: Tran Thien Thao Vy

Student ID: 104991221

Tutorial session: Friday 07:15AM

Uploaded photos:

[Upload more photos](#)

Photo	Name	Description	Creation date	Keywords
	milk tea	drink	2024-11-04	milk, tea
	chocolate cake	sweet	2024-11-04	chocolate, cake

### III. Additional information for marking

Album.php ELB link: <http://assignment2-elb-1112750366.us-east-1.elb.amazonaws.com/photoalbum/album.php>

Dev Instance EC2 link: <http://ec2-54-174-120-228.compute-1.amazonaws.com/>

PhpMyAdmin link: <http://ec2-54-174-120-228.compute-1.amazonaws.com/phpmyadmin>