COS20019 - Cloud Computing Architecture
Assignment 2
Developing a highly available Photo Album website.

Student Name: Nur E Siam

Student ID: 103842784

I. Introduction

Amazon Web Services (AWS) offers a wide range of services that can be used to create a reliable and resilient photo album website. With the use of various AWS services for storage, web hosting, databases, caching, load balancing, and monitoring, we can develop an advanced and highly available photo album website with more features compared to previous assignments.

Foundation and Infrastructure

Before moving to the deployment phase, we first need to lay the foundation for our website, namely creating VPC with accurately configured subnets according to the given Architecture diagram.

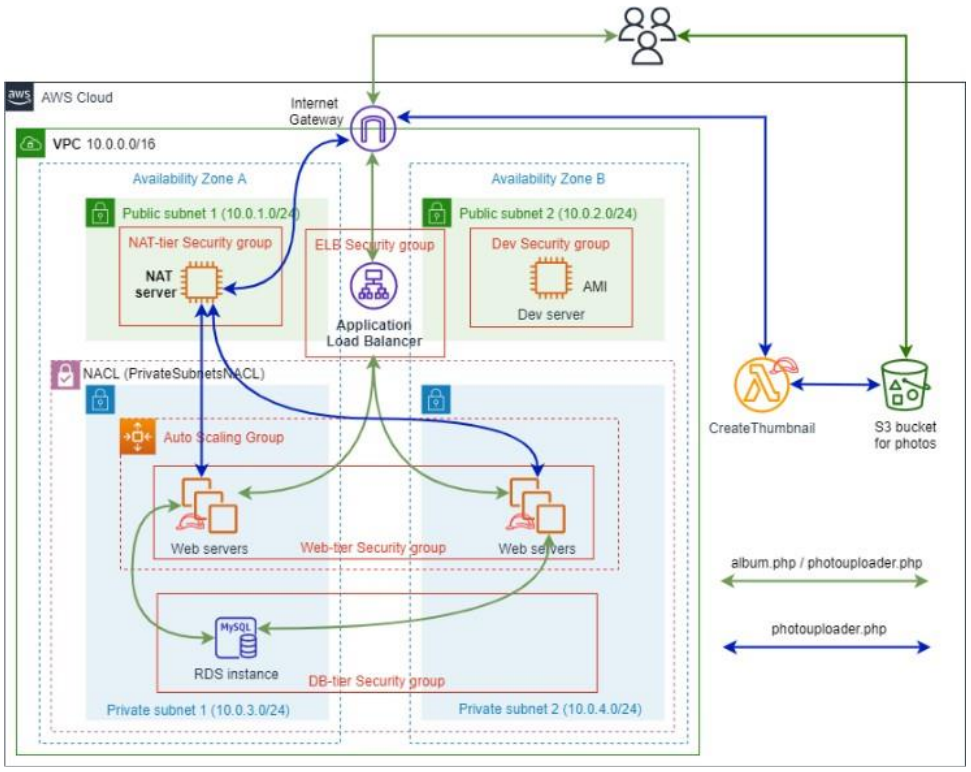
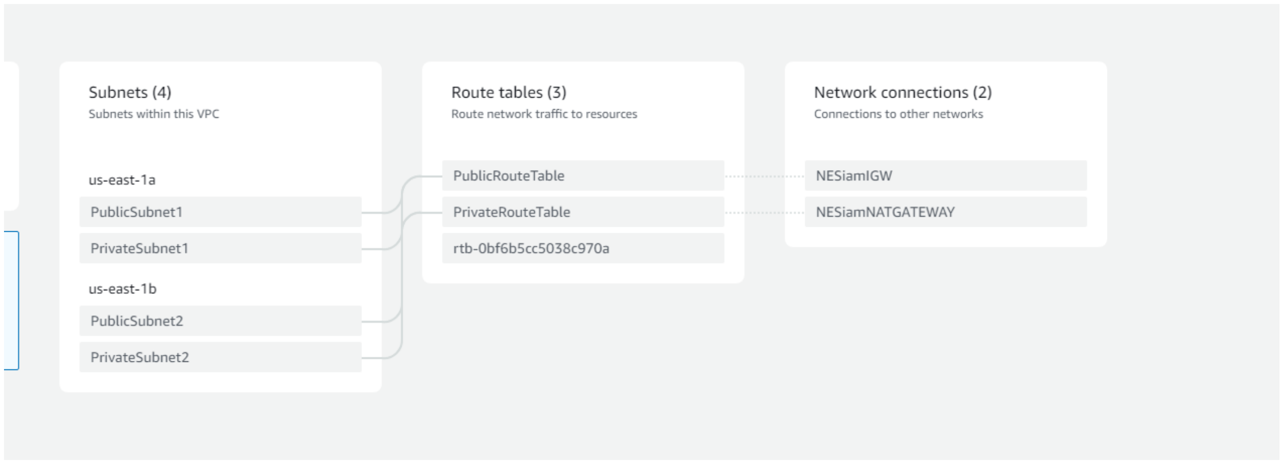


Figure 1: Architecture diagram.

This task is not so difficult as Assignment 1A and 1B has already introduced such objective. After configuring we might have a Resource map for our VPC and their according subnets as follows:


Subnets (4)

Subnets within this VPC

us-east-1a

PublicSubnet1

PrivateSubnet1

us-east-1b

PublicSubnet2

PrivateSubnet2

Route tables (3)

Route network traffic to resources

PublicRouteTable

PrivateRouteTable

rtb-0bf6b5cc5038c970a

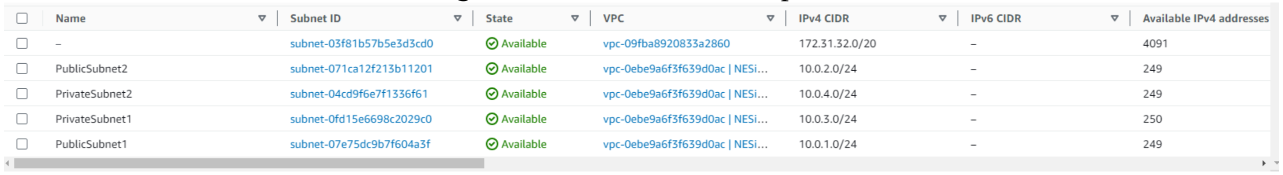
Network connections (2)

Connections to other networks

NESiamIGW

NESiamNATGATEWAY

Figure 2: VPC resource map.

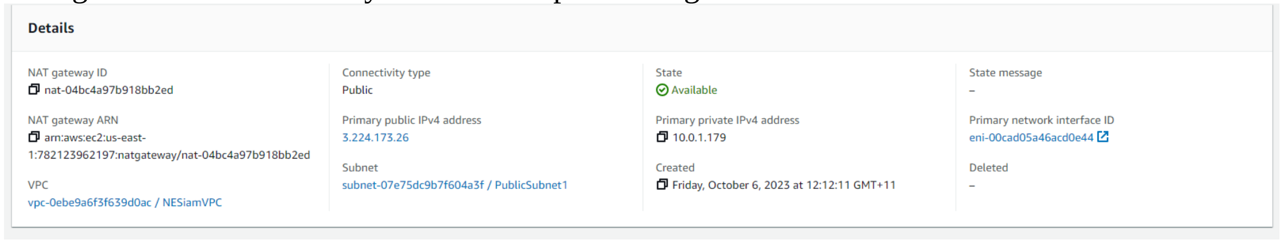


	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses
☐	–	subnet-03f81b57b5e3d3cd0	✓ Available	vpc-09fba8920833a2860	172.31.32.0/20	–	4091
☐	PublicSubnet2	subnet-071ca12f213b11201	✓ Available	vpc-0ebe9a6f3f639d0ac | NESi...	10.0.2.0/24	–	249
☐	PrivateSubnet2	subnet-04cd9f6e7f1336f61	✓ Available	vpc-0ebe9a6f3f639d0ac | NESi...	10.0.4.0/24	–	249
☐	PrivateSubnet1	subnet-0fd15e6698c2029c0	✓ Available	vpc-0ebe9a6f3f639d0ac | NESi...	10.0.3.0/24	–	250
☐	PublicSubnet1	subnet-07e75dc9b7f604a3f	✓ Available	vpc-0ebe9a6f3f639d0ac | NESi...	10.0.1.0/24	–	249

Figure 3: IPv4 CIDR for each subnet.

During the development phase, the Dev_Server_Public2 Subnet will have internet access for testing the website's functionality. Once the website is fully functional, we'll create an AMI for Dev_Server_Public2 and deploy it as needed.

The two Private subnets will host an autoscaling group and will be routed through a NAT Gateway, not a NAT instance.
The NAT Gateway will be hosted in the NAT_Server_Public1 subnet, which will be routed through the Internet Gateway. Here's a simplified diagram for reference:



Details

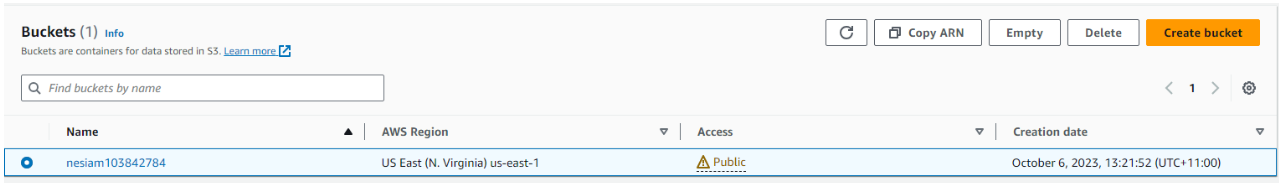
NAT gateway ID nat-04bc4a97b918bb2ed	Connectivity type Public	State ✓ Available	State message –
NAT gateway ARN arn:aws:ec2:us-east-1:782123962197:natgateway/nat-04bc4a97b918bb2ed	Primary public IPv4 address 3.224.173.26	Primary private IPv4 address 10.0.1.179	Primary network interface ID eni-00cad05a46acd0e44
VPC vpc-0ebe9a6f3f639d0ac / NESiamVPC	Subnet subnet-07e75dc9b7f604a3f / PublicSubnet1	Created Friday, October 6, 2023 at 12:12:11 GMT+11	Deleted –

Figure 4: NAT Gateway in NAT_Server_Public1

NAT will also conclude our first stage of this assignment, with the VPC fully constructed, we will move on to create entities that reside in these subnets.

II. Functionalities of the Website

As mentioned above, we will develop our website in public subnet 2 and only after making sure that every functionality is satisfied will we deploy it onto other subnets. Our first task would be to make sure that S3 bucket is working.



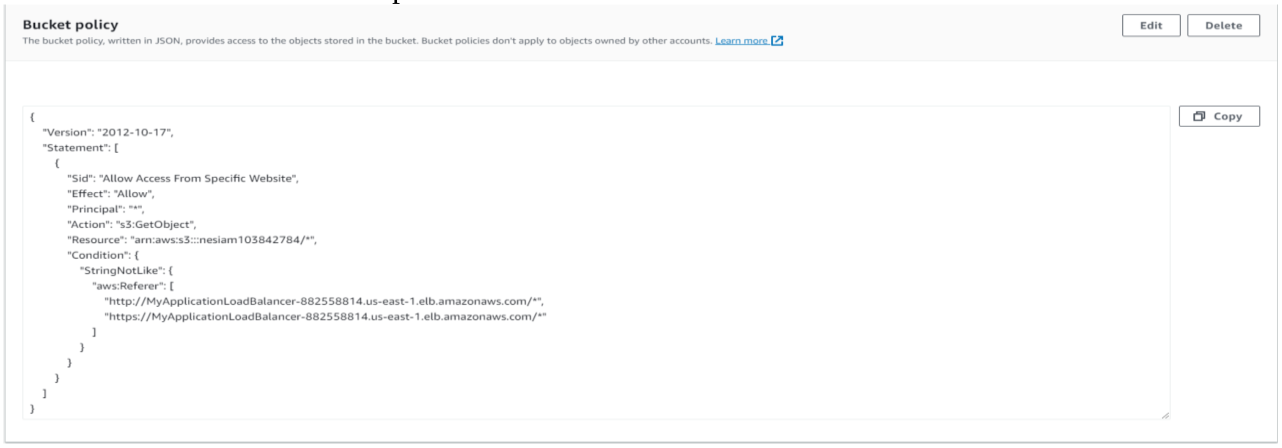
Buckets (1) Info

Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
●	nesiam103842784	US East (N. Virginia) us-east-1	⚠ Public	October 6, 2023, 13:21:52 (UTC+11:00)

Figure 5: S3 Bucket for Assignment 2.

According to the architecture diagram our S3 bucket will only allow access from and by the Application Load Balancer (ALB), therefore, a bucket policy will be provided so as to restrict permission from and to our bucket.



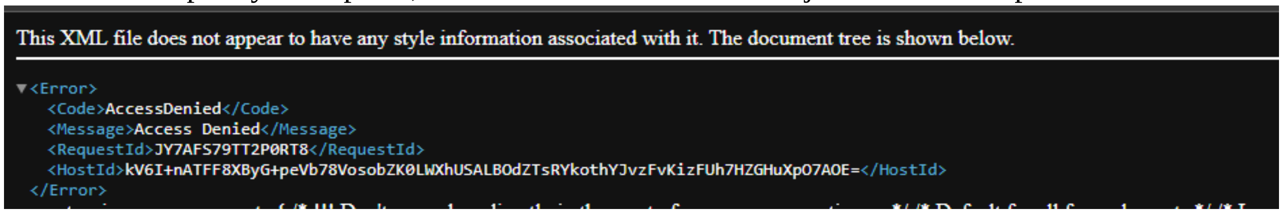
Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Allow Access From Specific Website",
         "Effect": "Allow",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::nesiam103842784/*",
         "Condition": {
            "StringNotLike": {
               "aws:Referer": [
                  "http://MyApplicationLoadBalancer-882558814.us-east-1.elb.amazonaws.com/*",
                  "https://MyApplicationLoadBalancer-882558814.us-east-1.elb.amazonaws.com/*"
               ]
            }
         }
      }
   ]
}
```

Figure 6: S3 Bucket Policy.

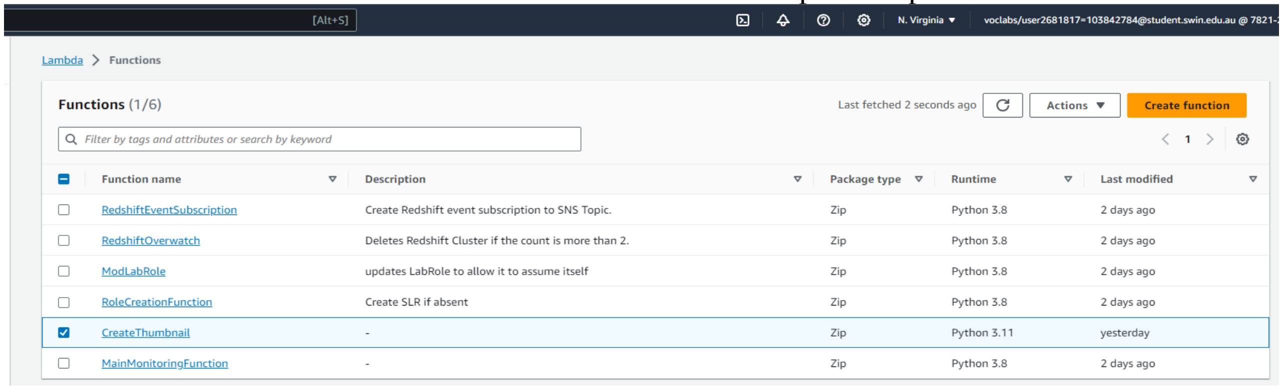
With the policy in its place, unauthorized access to our objects will not be possible.



```
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>JY7AFS79TT2P0RT8</RequestId>
   <HostId>kV6I+nATFF8XByG+peVb78VosobZK0LWXhUSALBOdZTsRYkothYJvzFvKizFUh7HZGHuXpO7AOE=</HostId>
 </Error>
```

Figure 7: Unauthorized access to the S3 Objects.

A Lambda function will also be needed to resize uploaded picture to S3.

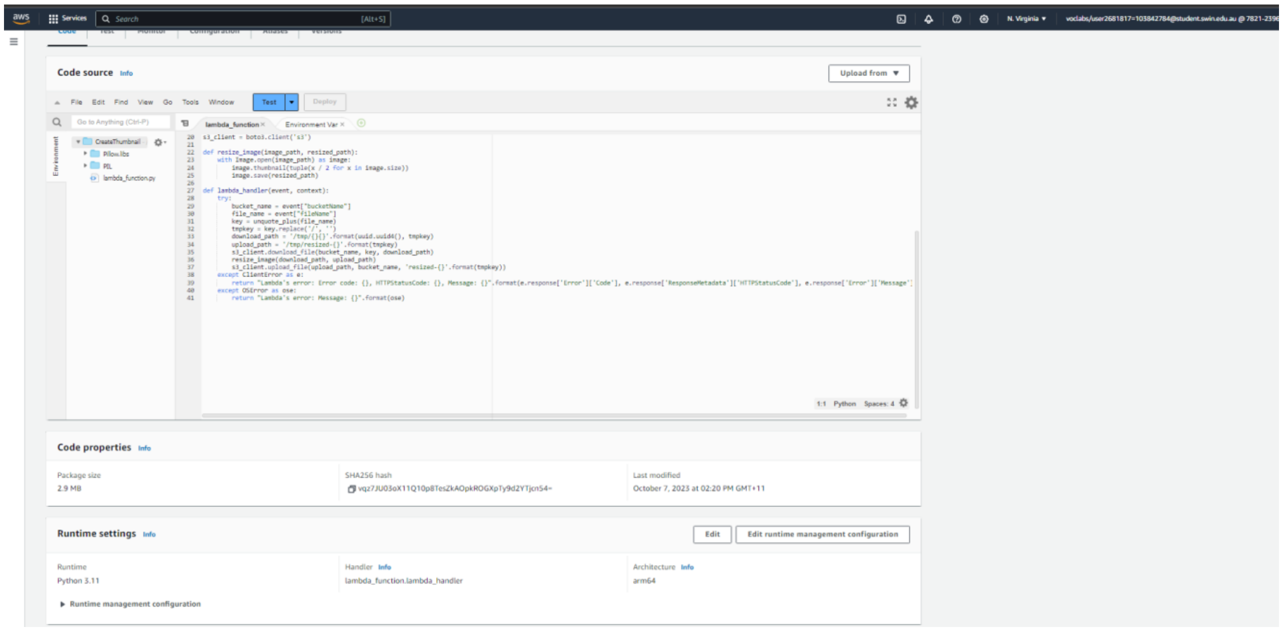


[Alt+S]
N. Virginia
voclabs/user2681817=103842784@student.swin.edu.au @ 7821-
Lambda > Functions
Last fetched 2 seconds ago
Actions
Create function
Filter by tags and attributes or search by keyword

Functions (1/6)

	Function name	Description	Package type	Runtime	Last modified
☐	RedshiftEventSubscription	Create Redshift event subscription to SNS Topic.	Zip	Python 3.8	2 days ago
☐	RedshiftOverwatch	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.8	2 days ago
☐	ModLabRole	updates LabRole to allow it to assume itself	Zip	Python 3.8	2 days ago
☐	RoleCreationFunction	Create SLR if absent	Zip	Python 3.8	2 days ago
✓	CreateThumbnail	-	Zip	Python 3.11	yesterday
☐	MainMonitoringFunction	-	Zip	Python 3.8	2 days ago

Figure 8: Lamda Function.



```
s3_client = boto3.client('s3')

def resize_image(image_path, resized_path):
    with Image.open(image_path) as image:
        image.thumbnail(tuple(x / 2 for x in image.size))
        image.save(resized_path)

def lambda_handler(event, context):
    try:
        bucket_name = event["bucketName"]
        file_name = event["fileName"]
        key = unquote_plus(file_name)
        tmpkey = key.replace('/', '')
        download_path = '/tmp/{}{}'.format(uuid.uuid4(), tmpkey)
        upload_path = '/tmp/resized-{}'.format(tmpkey)
        s3_client.download_file(bucket_name, key, download_path)
        resize_image(download_path, upload_path)
        s3_client.upload_file(upload_path, bucket_name, 'resized-{}'.format(tmpkey))
    except ClientError as e:
        return "Lambda's error: Error code: {}, HTTPStatusCode: {}, Message: {}".format(e.response['Error']['Code'], e.response['ResponseMetadata']['HTTPStatusCode'], e.response['Error']['Message']
    except OSError as ose:
        return "Lambda's error: Message: {}".format(ose)
```

Code properties

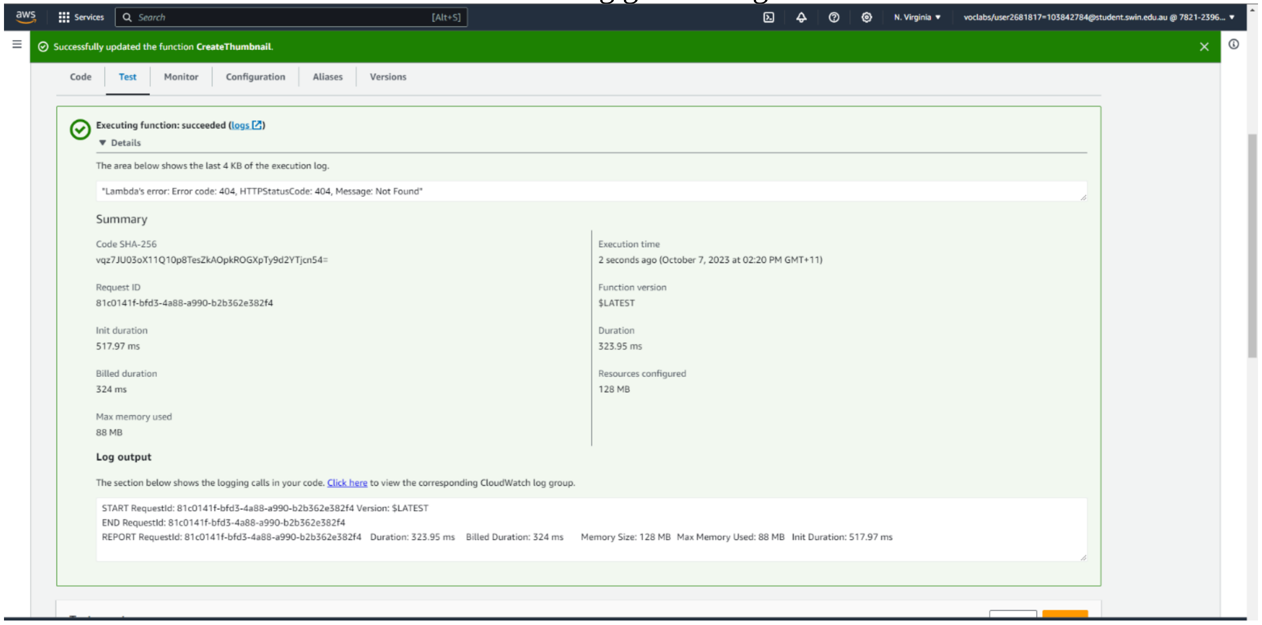
Package size	SHA256 hash	Last modified
2.9 MB	vqz7JU03oX11Q10p8TesZkAOpkROGXpTy9d2YTjcn54=	October 7, 2023 at 02:20 PM GMT+11

Runtime settings

Runtime	Handler	Architecture
Python 3.11	lambda_function.lambda_handler	arm64

Figure 9: Code structure and properties of the Lambda function.

As the package for creating the function is provided, we only need to test the package to make sure nothing goes wrong.



Successfully updated the function CreateThumbnail.

Executing function: succeeded (logs)

Details

The area below shows the last 4 KB of the execution log.

```
"Lambda's error: Error code: 404, HTTPStatusCode: 404, Message: Not Found"
```

Summary

Code SHA-256: vqz7JU03oX11Q10p8TesZkAOpkROGXpTy9d2YTjcn54=	Execution time: 2 seconds ago (October 7, 2023 at 02:20 PM GMT+11)
Request ID: 81c0141f-bfd3-4a88-a990-b2b362e382f4	Function version: $LATEST
Init duration: 517.97 ms	Duration: 323.95 ms
Billed duration: 324 ms	Resources configured: 128 MB
Max memory used: 88 MB	

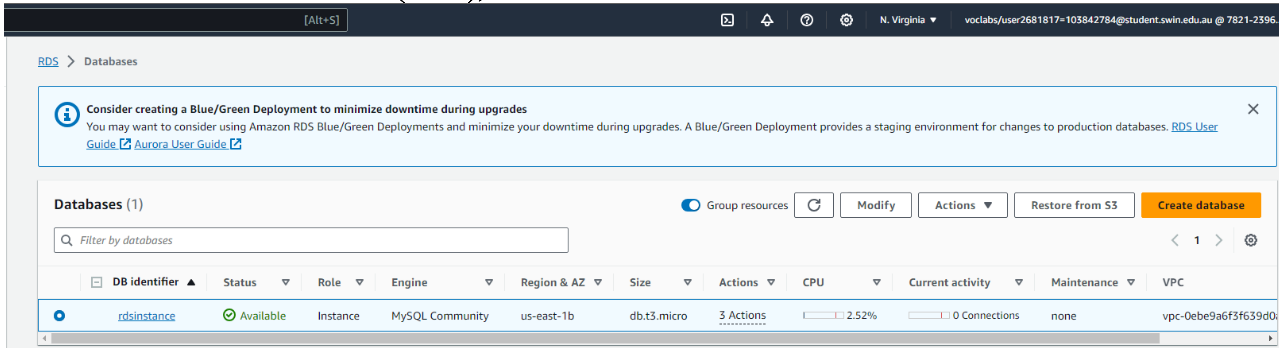
Log output

The section below shows the logging calls in your code. Click here to view the corresponding CloudWatch log group.

```
START RequestId: 81c0141f-bfd3-4a88-a990-b2b362e382f4 Version: $LATEST
END RequestId: 81c0141f-bfd3-4a88-a990-b2b362e382f4
REPORT RequestId: 81c0141f-bfd3-4a88-a990-b2b362e382f4  Duration: 323.95 ms  Billed Duration: 324 ms   Memory Size: 128 MB  Max Memory Used: 88 MB  Init Duration: 517.97 ms
```

Figure 10: Lambda function test case.

With the test case successfully executed we will move onto the next part which is to create Relational Database Service (RDS), with RDS we can launch a database instance in minutes.



Consider creating a Blue/Green Deployment to minimize downtime during upgrades

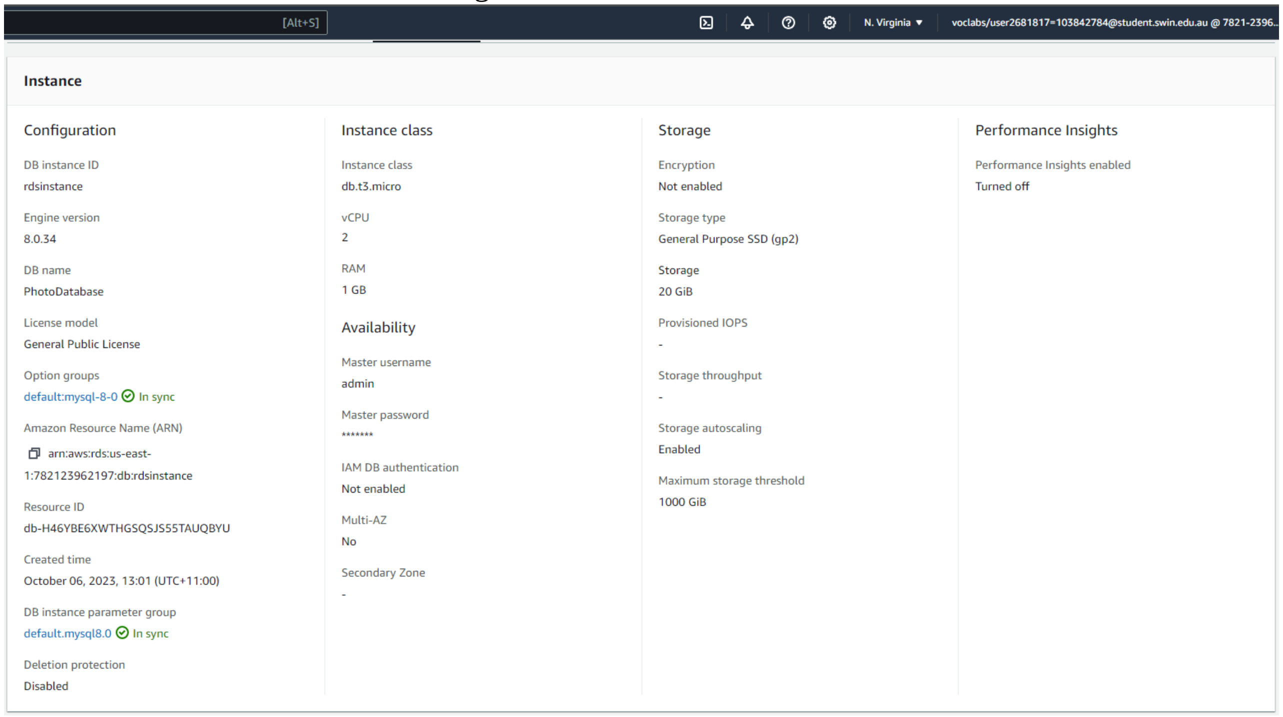
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide

Databases (1)

DB identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU	Current activity	Maintenance	VPC
rdsinstance	Available	Instance	MySQL Community	us-east-1b	db.t3.micro	3 Actions	2.52%	0 Connections	none	vpc-0ebe9a6f3f639d0

Figure 11: RDS successfully deployed.

Detailed configuration of the RDS are as follows:



[Alt+S]

N. Virginia ▾ voclabs/user2681817=103842784@student.swin.edu.au @ 7821-2396...

Instance

Configuration

DB instance ID
rdsinstance

Engine version
8.0.34

DB name
PhotoDatabase

License model
General Public License

Option groups
default:mysql-8-0 In sync

Amazon Resource Name (ARN)
arn:aws:rds:us-east-1:782123962197:db:rdsinstance

Resource ID
db-H46YBE6XWTHGSQSJS55TAUQBYU

Created time
October 06, 2023, 13:01 (UTC+11:00)

DB instance parameter group
default.mysql8.0 In sync

Deletion protection
Disabled

Instance class

Instance class
db.t3.micro

vCPU
2

RAM
1 GB

Availability

Master username
admin

Master password
****

IAM DB authentication
Not enabled

Multi-AZ
No

Secondary Zone
-

Storage

Encryption
Not enabled

Storage type
General Purpose SSD (gp2)

Storage
20 GiB

Provisioned IOPS
-

Storage throughput
-

Storage autoscaling
Enabled

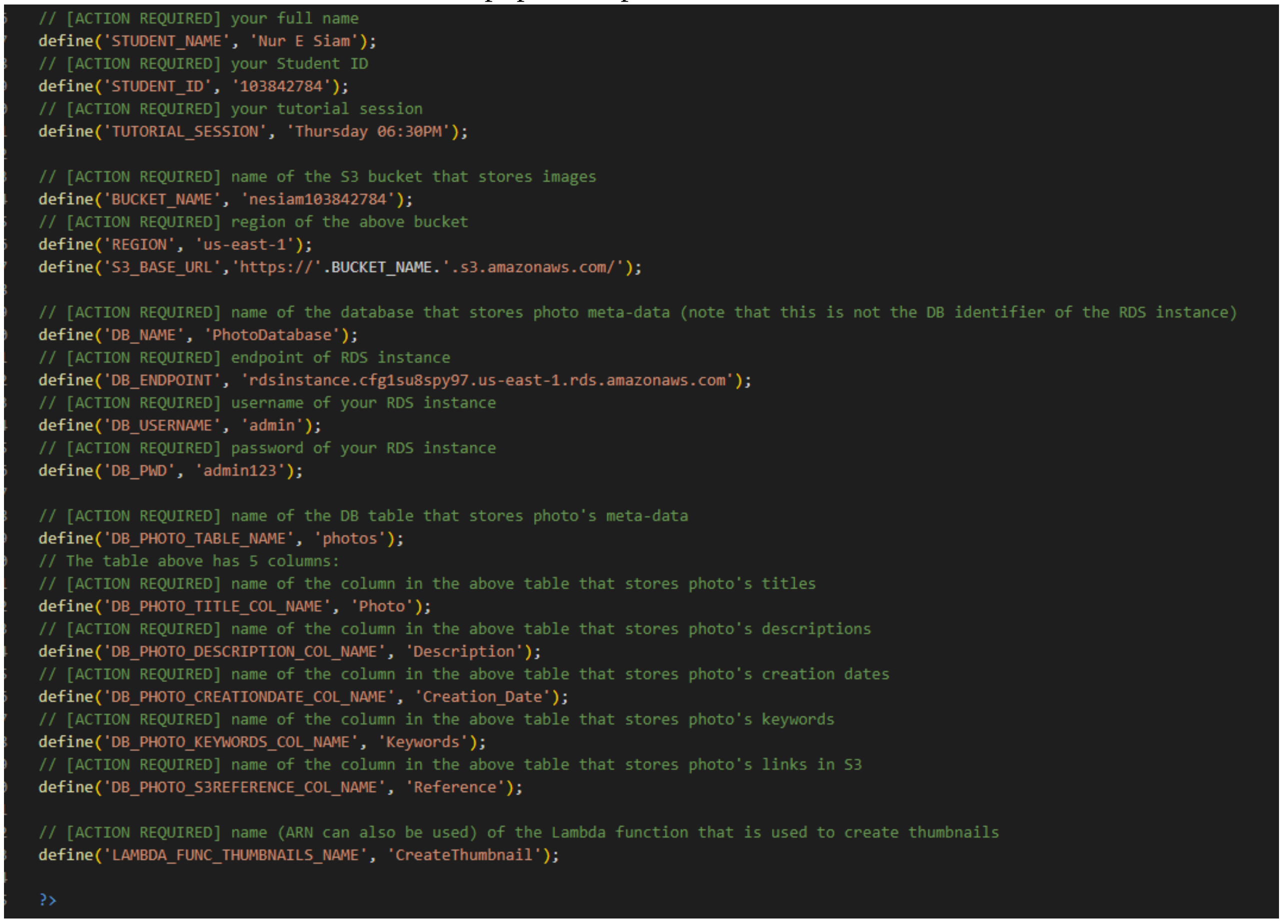
Maximum storage threshold
1000 GiB

Performance Insights

Performance Insights enabled
Turned off

Figure 12: RDS detailed configuration.

Before we can put our website onto the development server, we will need to configure all the missing part of the provided code for the website. To be specific, we will need to modify the constant.php in the provided ZIP file.



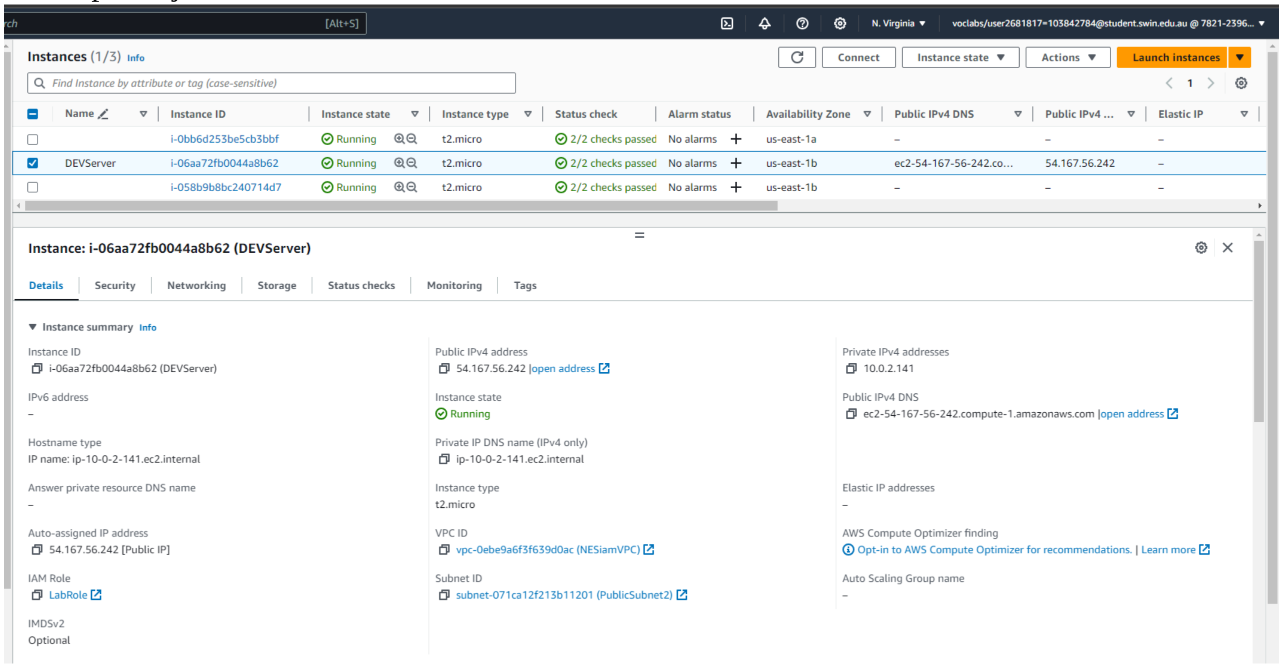
```php
// [ACTION REQUIRED] your full name
define('STUDENT_NAME', 'Nur E Siam');
// [ACTION REQUIRED] your Student ID
define('STUDENT_ID', '103842784');
// [ACTION REQUIRED] your tutorial session
define('TUTORIAL_SESSION', 'Thursday 06:30PM');

// [ACTION REQUIRED] name of the S3 bucket that stores images
define('BUCKET_NAME', 'nesiam103842784');
// [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', 'PhotoDatabase');
// [ACTION REQUIRED] endpoint of RDS instance
define('DB_ENDPOINT', 'rdsinstance.cfg1su8spy97.us-east-1.rds.amazonaws.com');
// [ACTION REQUIRED] username of your RDS instance
define('DB_USERNAME', 'admin');
// [ACTION REQUIRED] password of your RDS instance
define('DB_PWD', 'admin123');

// [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');
// [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', '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 13: Modified constant.php.

With all the components successfully configured, we can then move to creating an EC2 instance on the development server.

III. Development of the Website using EC2 and AMI

All the functionalities of our website will require a host to work, an EC2 instance would be an ideal environment for our website. Also an IAM role would also be necessary to be able to put objects into the S3 bucket and invoke the CreateThumbnail Lambda function.



[Alt+S]
N. Virginia
voclabs/user2681817=103842784@student.swin.edu.au @ 7821-2396...
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)

Instances (1/3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
	i-0bb6d253be5cb3bbf	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	–	–	–
DEVServer	i-06aa72fb0044a8b62	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-167-56-242.co...	54.167.56.242	–
	i-058b9b8bc240714d7	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	–	–	–

Instance: i-06aa72fb0044a8b62 (DEVServer)

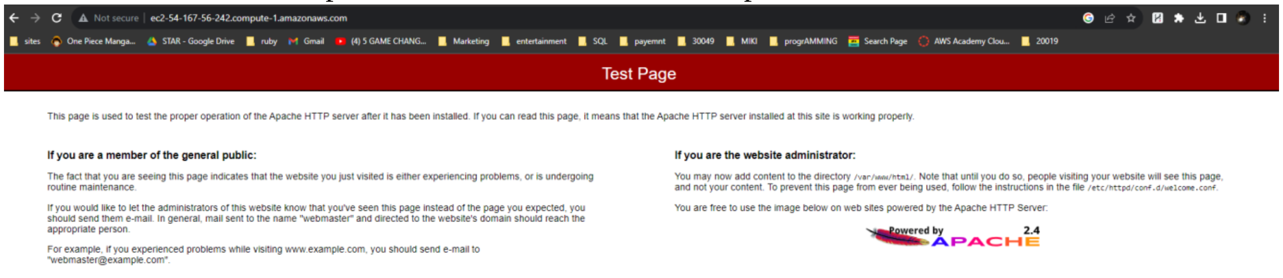
Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary Info

Instance ID: i-06aa72fb0044a8b62 (DEVServer)	Public IPv4 address: 54.167.56.242 |open address	Private IPv4 addresses: 10.0.2.141
IPv6 address: –	Instance state: Running	Public IPv4 DNS: ec2-54-167-56-242.compute-1.amazonaws.com |open address
Hostname type: IP name: ip-10-0-2-141.ec2.internal	Private IP DNS name (IPv4 only): ip-10-0-2-141.ec2.internal	
Answer private resource DNS name: –	Instance type: t2.micro	Elastic IP addresses: –
Auto-assigned IP address: 54.167.56.242 [Public IP]	VPC ID: vpc-0ebe9a6f3f639d0ac (NESiamVPC)	AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role: LabRole	Subnet ID: subnet-071ca12f213b11201 (PublicSubnet2)	Auto Scaling Group name: –
IMDSv2: Optional		

Figure 14:DevServerInstances with IAM role.

After deploying the EC2 instance, we can check if it is accessible by allocating for it an Elastic Ips and access the website via its public IPv4 DNS:



Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

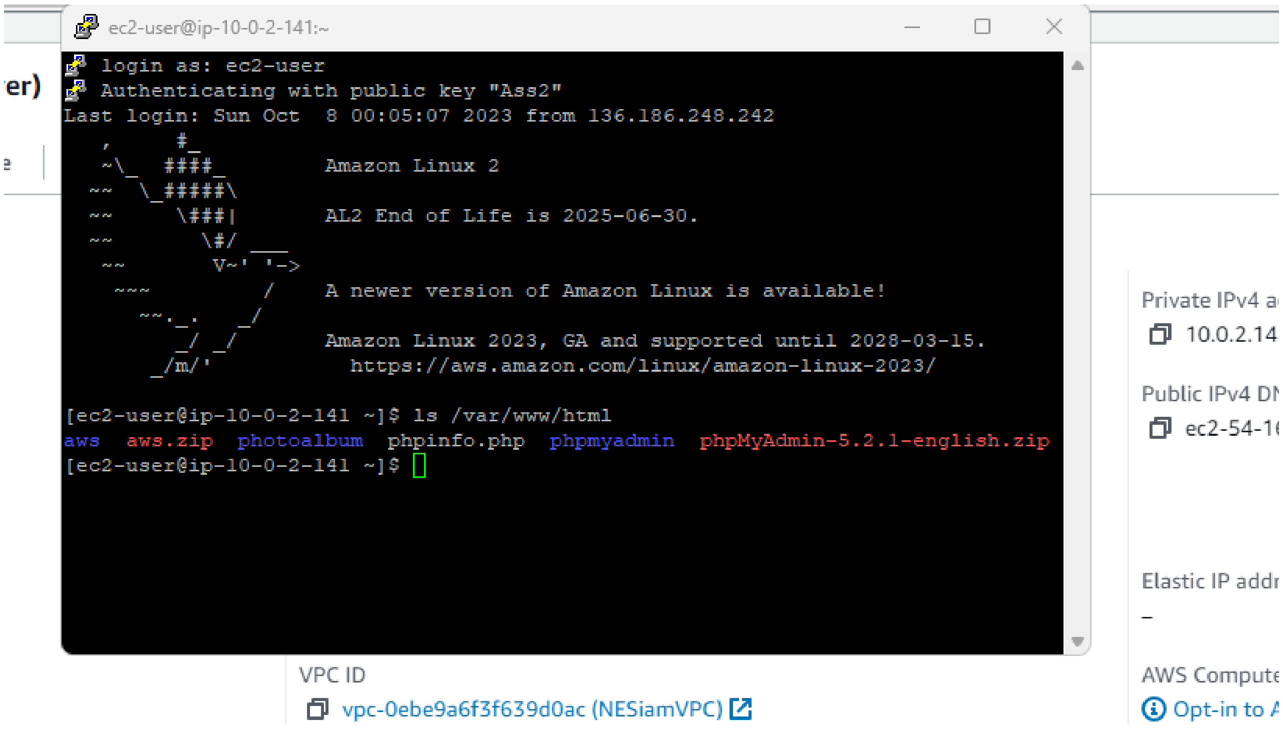
If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:

Figure 15: EC2 Instance successfully deployed.

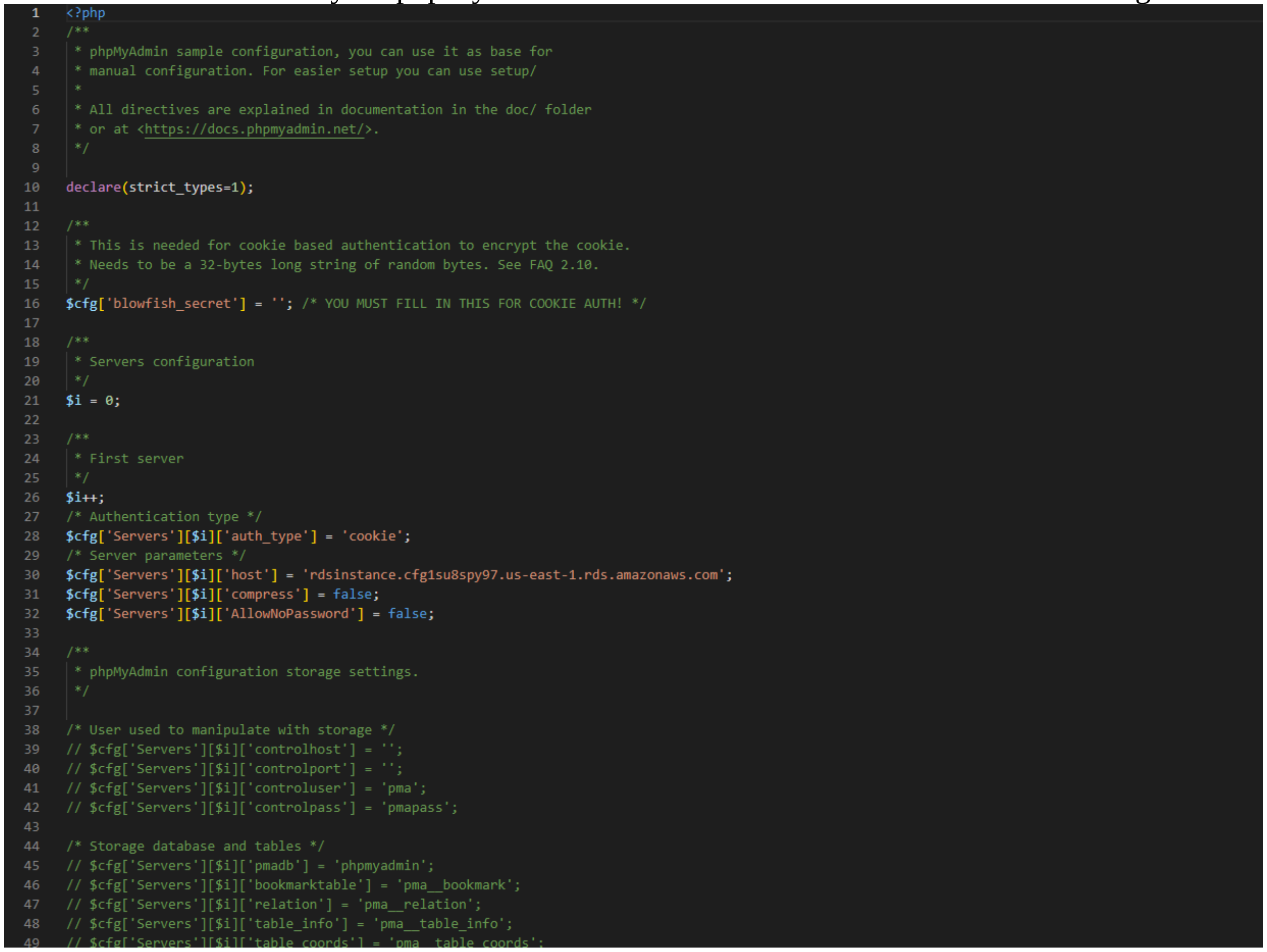
We will then connect to the instance via SSH to configure the database using the same Public IPv4 DNS.



```
login as: ec2-user
Authenticating with public key "Ass2"
Last login: Sun Oct  8 00:05:07 2023 from 136.186.248.242
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       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/

[ec2-user@ip-10-0-2-141 ~]$ ls /var/www/html
aws  aws.zip  photoalbum  phpinfo.php  phpmyadmin  phpMyAdmin-5.2.1-english.zip
[ec2-user@ip-10-0-2-141 ~]$
```

Figure 16: DevServer SSH terminal.

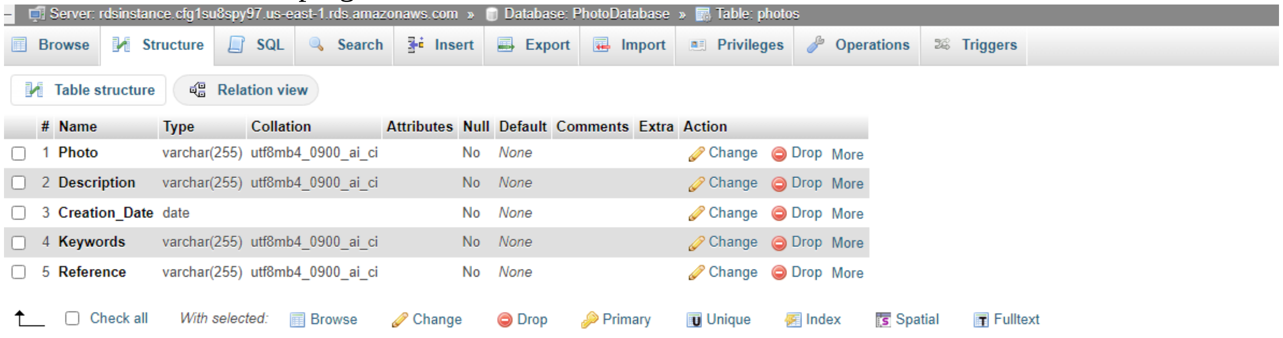
To set up phpMyAdmin for creating database metadata and monitoring the database's performance, you need to make a minor modification in the config.inc.php file. This change will establish a connection between your phpMyAdmin console and the RDS instance created in stage 3.



```php
<?php
/**
 * phpMyAdmin sample configuration, you can use it as base for
 * manual configuration. For easier setup you can use setup/
 *
 * All directives are explained in documentation in the doc/ folder
 * or at <https://docs.phpmyadmin.net/>.
 */

declare(strict_types=1);

/**
 * This is needed for cookie based authentication to encrypt the cookie.
 * Needs to be a 32-bytes long string of random bytes. See FAQ 2.10.
 */
$cfg['blowfish_secret'] = ''; /* YOU MUST FILL IN THIS FOR COOKIE AUTH! */

/**
 * Servers configuration
 */
$i = 0;

/**
 * First server
 */
$i++;
/* Authentication type */
$cfg['Servers'][$i]['auth_type'] = 'cookie';
/* Server parameters */
$cfg['Servers'][$i]['host'] = 'rdsinstance.cfg1su8spy97.us-east-1.rds.amazonaws.com';
$cfg['Servers'][$i]['compress'] = false;
$cfg['Servers'][$i]['AllowNoPassword'] = false;

/**
 * phpMyAdmin configuration storage settings.
 */

/* User used to manipulate with storage */
// $cfg['Servers'][$i]['controlhost'] = '';
// $cfg['Servers'][$i]['controlport'] = '';
// $cfg['Servers'][$i]['controluser'] = 'pma';
// $cfg['Servers'][$i]['controlpass'] = 'pmapass';

/* Storage database and tables */
// $cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';
// $cfg['Servers'][$i]['bookmarktable'] = 'pma__bookmark';
// $cfg['Servers'][$i]['relation'] = 'pma__relation';
// $cfg['Servers'][$i]['table_info'] = 'pma__table_info';
// $cfg['Servers'][$i]['table_coords'] = 'pma__table_coords';
```

Figure 17: config.inc.php

After changing the value of ‘*localhost*’ in *config.inc.php* we can access *phpMyAdmin* on our developing server website to create meta-data for our database.

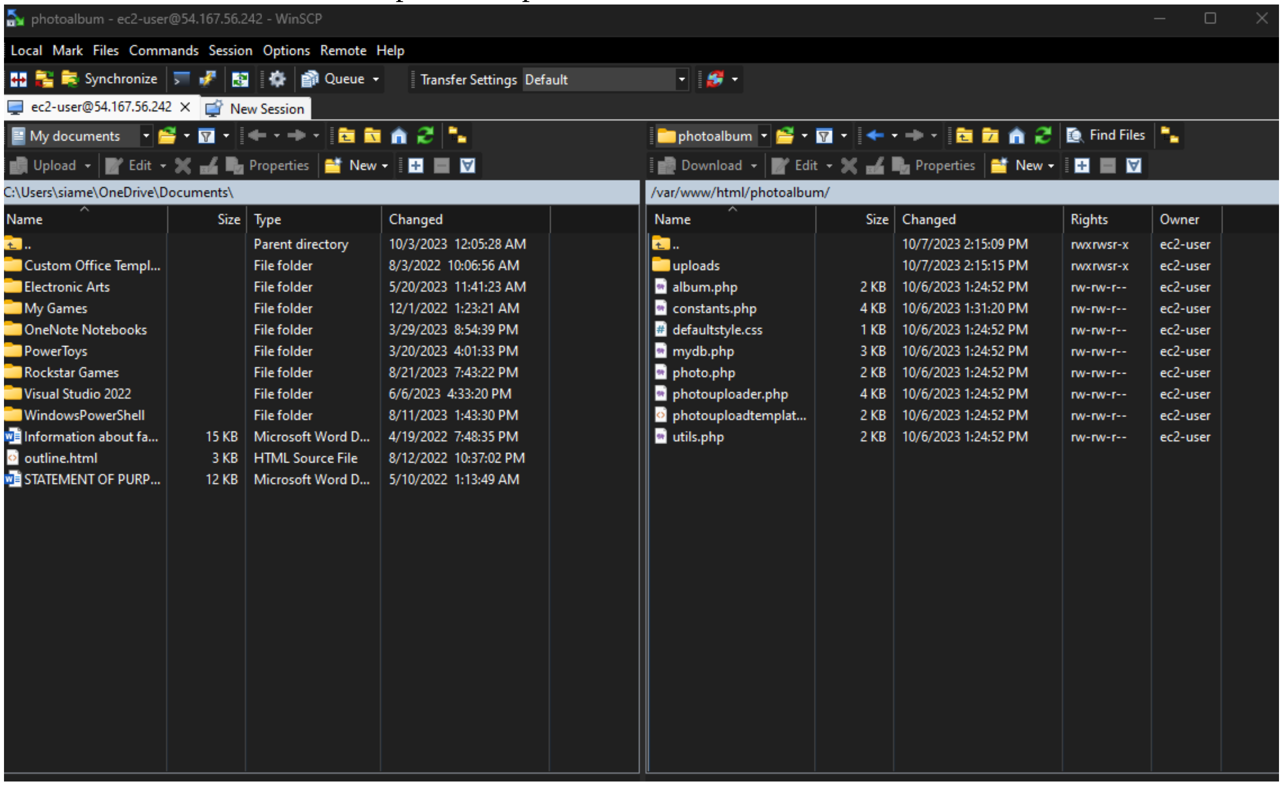

Server: rdsinstance.cfg1su8spy97.us-east-1.rds.amazonaws.com » Database: PhotoDatabase » Table: photos
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Photo	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	Creation_Date	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

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Figure 18: Meta-data for the database.

We can then upload the provided codes to test our website.


photoalbum - ec2-user@54.167.56.242 - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize Queue Transfer Settings Default
ec2-user@54.167.56.242 New Session
My documents Upload Edit Properties New
photoalbum Download Edit Properties New Find Files

C:\Users\siame\OneDrive\Documents\

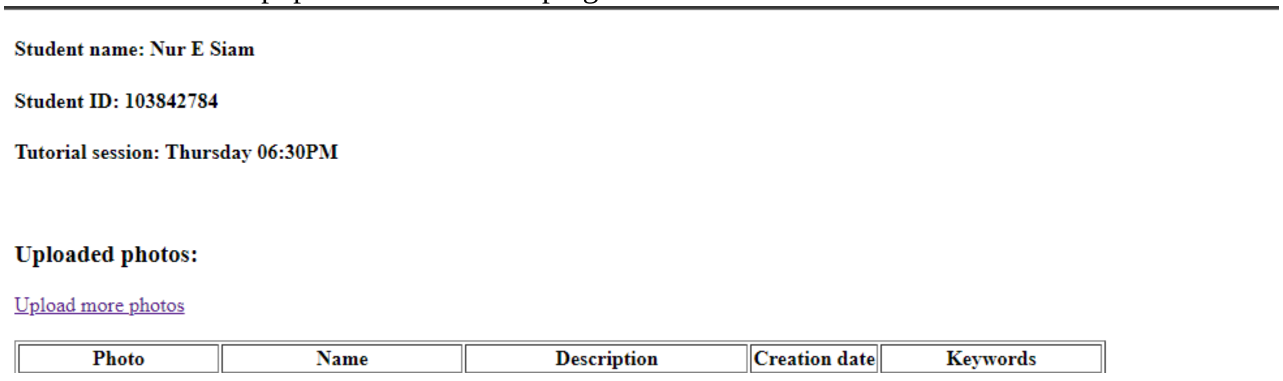
Name	Size	Type	Changed
..		Parent directory	10/3/2023 12:05:28 AM
Custom Office Templ...		File folder	8/3/2022 10:06:56 AM
Electronic Arts		File folder	5/20/2023 11:41:23 AM
My Games		File folder	12/1/2022 1:23:21 AM
OneNote Notebooks		File folder	3/29/2023 8:54:39 PM
PowerToys		File folder	3/20/2023 4:01:33 PM
Rockstar Games		File folder	8/21/2023 7:43:22 PM
Visual Studio 2022		File folder	6/6/2023 4:33:20 PM
WindowsPowerShell		File folder	8/11/2023 1:43:30 PM
Information about fa...	15 KB	Microsoft Word D...	4/19/2022 7:48:35 PM
outline.html	3 KB	HTML Source File	8/12/2022 10:37:02 PM
STATEMENT OF PURP...	12 KB	Microsoft Word D...	5/10/2022 1:13:49 AM

/var/www/html/photoalbum/

Name	Size	Changed	Rights	Owner
..		10/7/2023 2:15:09 PM	rwxrwsr-x	ec2-user
uploads		10/7/2023 2:15:15 PM	rwxrwsr-x	ec2-user
album.php	2 KB	10/6/2023 1:24:52 PM	rw-rw-r--	ec2-user
constants.php	4 KB	10/6/2023 1:31:20 PM	rw-rw-r--	ec2-user
defaultstyle.css	1 KB	10/6/2023 1:24:52 PM	rw-rw-r--	ec2-user
mydb.php	3 KB	10/6/2023 1:24:52 PM	rw-rw-r--	ec2-user
photo.php	2 KB	10/6/2023 1:24:52 PM	rw-rw-r--	ec2-user
photouploader.php	4 KB	10/6/2023 1:24:52 PM	rw-rw-r--	ec2-user
photouploadtemplat...	2 KB	10/6/2023 1:24:52 PM	rw-rw-r--	ec2-user
utils.php	2 KB	10/6/2023 1:24:52 PM	rw-rw-r--	ec2-user

Figure 19: Uploaded codes for the website.

Our album.php site on the developing server website would look like this:



Student name: Nur E Siam

Student ID: 103842784

Tutorial session: Thursday 06:30PM

Uploaded photos:

Upload more photos

Photo	Name	Description	Creation date	Keywords

Figure 20: album.php

We can proceed to upload some photos on the website.

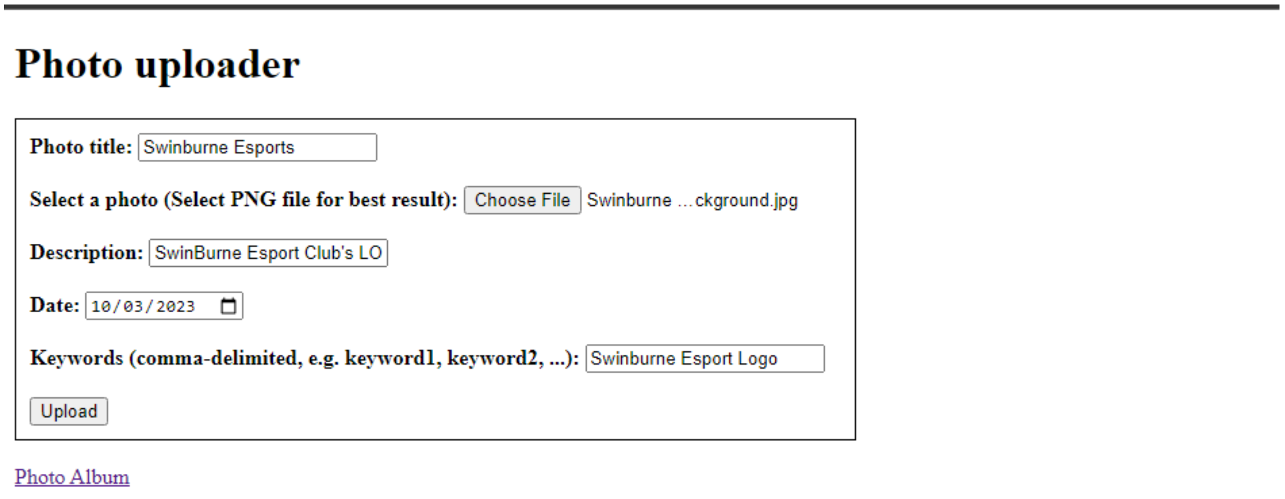


Photo uploader

Photo title: Swinburne Esports

Select a photo (Select PNG file for best result): Choose File Swinburne ...ckground.jpg

Description: SwinBurne Esport Club's LO

Date: 10/03/2023

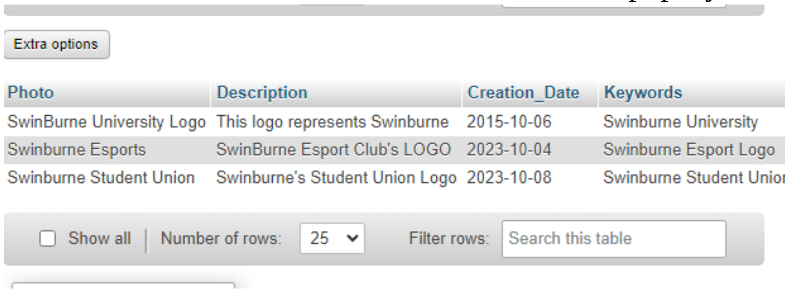
Keywords (comma-delimited, e.g. keyword1, keyword2, ...): Swinburne Esport Logo

Upload

Photo Album

Figure 21: photouploader.php.

we can see that S3 and the meta data in phpMyAdmin works as expected.

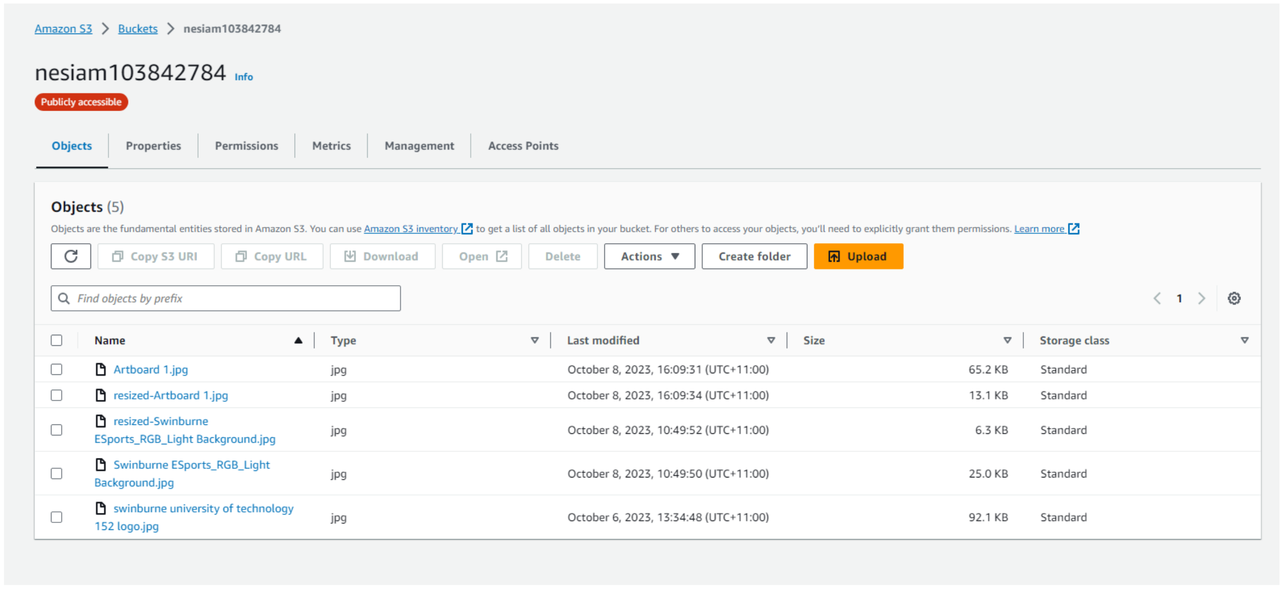


Extra options

Photo	Description	Creation_Date	Keywords	Reference
SwinBurne University Logo	This logo represents Swinburne	2015-10-06	Swinburne University	https://nesiam103842784.s3.amazonaws.com/swinburne...
Swinburne Esports	SwinBurne Esport Club's LOGO	2023-10-04	Swinburne Esport Logo	https://nesiam103842784.s3.amazonaws.com/Swinburne...
Swinburne Student Union	Swinburne's Student Union Logo	2023-10-08	Swinburne Student Union	https://nesiam103842784.s3.amazonaws.com/Artboard ...

Show all | Number of rows: 25 | Filter rows: Search this table

Figure 22: Meta-data in phpMyAdmin.



Amazon S3 > Buckets > nesiam103842784

nesiam103842784 Info

Publicly accessible

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (5)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

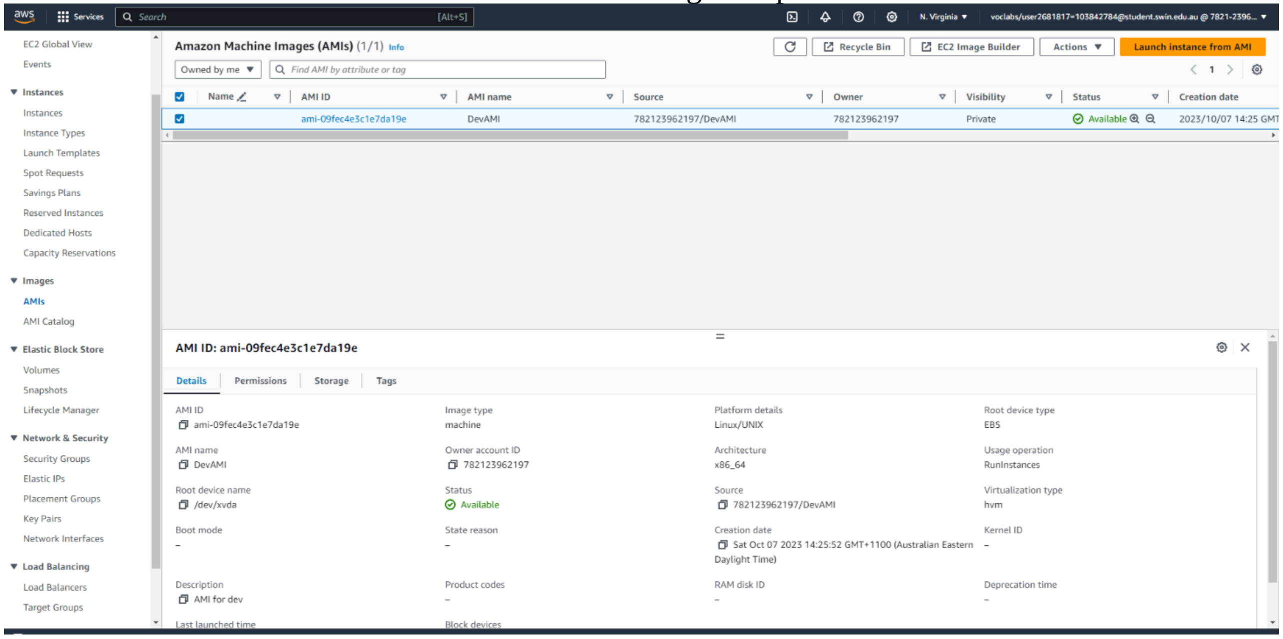
Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
Artboard 1.jpg	jpg	October 8, 2023, 16:09:31 (UTC+11:00)	65.2 KB	Standard
resized-Artboard 1.jpg	jpg	October 8, 2023, 16:09:34 (UTC+11:00)	13.1 KB	Standard
resized-Swinburne ESports_RGB_Light Background.jpg	jpg	October 8, 2023, 10:49:52 (UTC+11:00)	6.3 KB	Standard
Swinburne ESports_RGB_Light Background.jpg	jpg	October 8, 2023, 10:49:50 (UTC+11:00)	25.0 KB	Standard
swinburne university of technology 152 logo.jpg	jpg	October 6, 2023, 13:34:48 (UTC+11:00)	92.1 KB	Standard

Figure 23: Resized objects in S3 Bucket.

The website's functionality has been successfully configured and tested. We are now ready to proceed with creating an Amazon Machine Image (AMI) of the DevServer and deploying it within our Auto Scaling Group.



Amazon Machine Images (AMIs) (1/1) Info

Name	AMI ID	AMI name	Source	Owner	Visibility	Status	Creation date
	ami-09fec4e3c1e7da19e	DevAMI	782123962197/DevAMI	782123962197	Private	Available	2023/10/07 14:25 GMT

AMI ID: ami-09fec4e3c1e7da19e

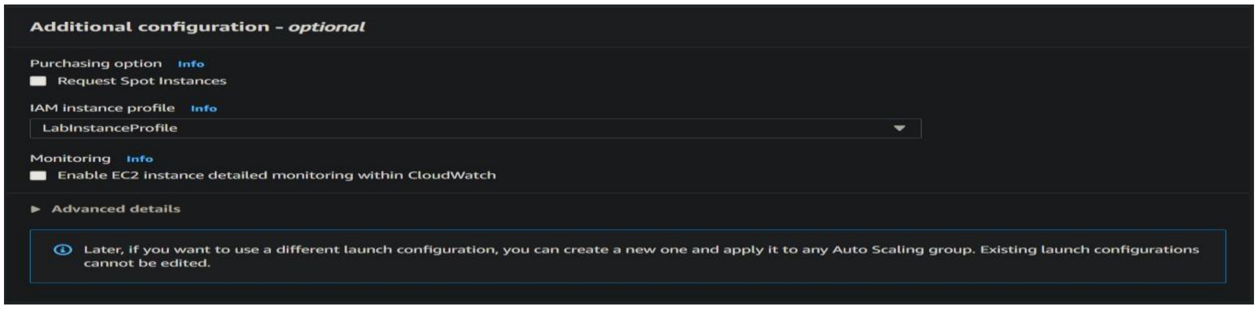
Details | Permissions | Storage | Tags

AMI ID ami-09fec4e3c1e7da19e	Image type machine	Platform details Linux/UNIX	Root device type EBS
AMI name DevAMI	Owner account ID 782123962197	Architecture x86_64	Usage operation RunInstances
Root device name /dev/xvda	Status Available	Source 782123962197/DevAMI	Virtualization type hvm
Boot mode –	State reason –	Creation date Sat Oct 07 2023 14:25:52 GMT+1100 (Australian Eastern Daylight Time)	Kernel ID –
Description AMI for dev	Product codes –	RAM disk ID –	Deprecation time –
Last launched time	Block devices		

Figure 24: AMI of DevServer.

IV. Elastic Load Balancer and Auto Scaling Group

Now that the development process is complete, our next steps involve enhancing the website's high availability and scalability by utilizing Elastic Load Balancer and Auto Scaling Group. we need to ensure that the IAM is applied correctly using the provided profile.



Additional configuration - *optional*

Purchasing option Info
☐ Request Spot Instances

IAM instance profile Info
LabInstanceProfile

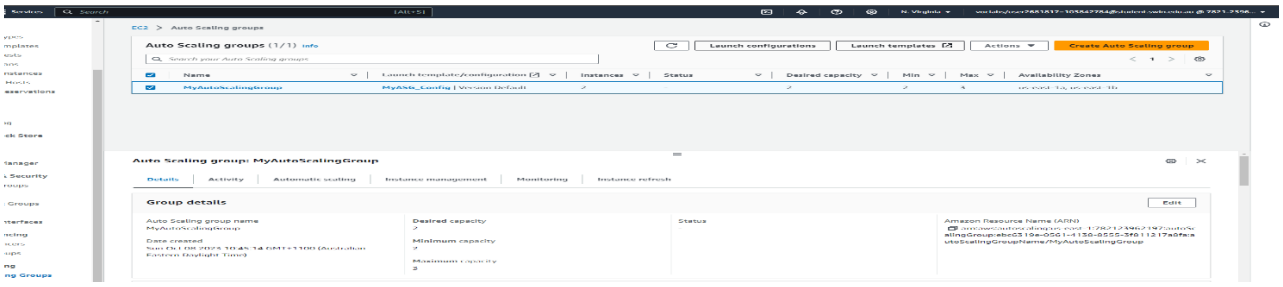
Monitoring Info
☐ Enable EC2 instance detailed monitoring within CloudWatch

▸ Advanced details

Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

Figure 25: IAM configuration

After that we can create an Auto Scaling Group that will control the Web Server Instance.



EC2 > Auto Scaling groups

Auto Scaling groups (1/1) Info

	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
✓	MyAutoScalingGroup	MyASG_Config | Version Default	2	-	2	2	4	us-east-1a, us-east-1b

Auto Scaling group: MyAutoScalingGroup

Details | Activity | Automatic scaling | Instance management | Monitoring | Instance refresh

Group details

Auto Scaling group name
MyAutoScalingGroup

Date created
Sun Oct 08 2023 10:45:14 GMT+1100 (Australian Eastern Daylight Time)

Desired capacity
2

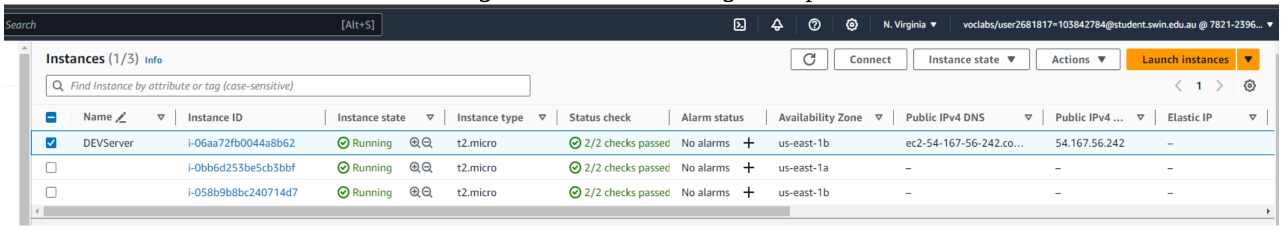
Minimum capacity
2

Maximum capacity
3

Status
-

Amazon Resource Name (ARN)
arn:aws:autoscaling:us-east-1:782123962197:autoScalingGroup:ebc63 19e-0561-4138-8555-3f0112 17a0fa:autoScalingGroupName/MyAutoScalingGroup

Figure 26: Auto Scaling Group.

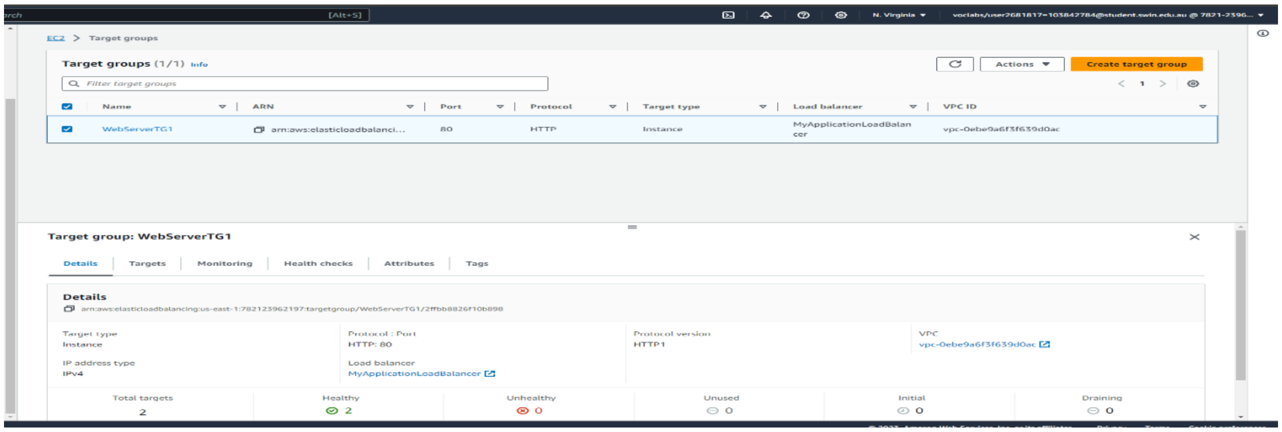


Instances (1/3) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
✓	DEVServer	i-06aa72fb0044a8b62	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-167-56-242.co...	54.167.56.242	–
☐		i-0bb6d253be5cb3bbf	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	–	–	–
☐		i-058b9b8bc240714d7	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	–	–	–

Figure 27: Two Instance creating from Auto Scaling Group.

A Target group is needed to launch the load balancer.



EC2 > Target groups

Target groups (1/1) Info

	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
✓	WebServerTG1	arn:aws:elasticloadbalanci...	80	HTTP	Instance	MyApplicationLoadBalancer	vpc-0ebe9a6f3f639d0ac

Target group: WebServerTG1

Details | Targets | Monitoring | Health checks | Attributes | Tags

Details
arn:aws:elasticloadbalancing:us-east-1:782123962197:targetgroup/WebServerTG1/2ffbb8826f10b898

Target type
Instance

IP address type
IPv4

Protocol : Port
HTTP: 80

Load balancer
MyApplicationLoadBalancer

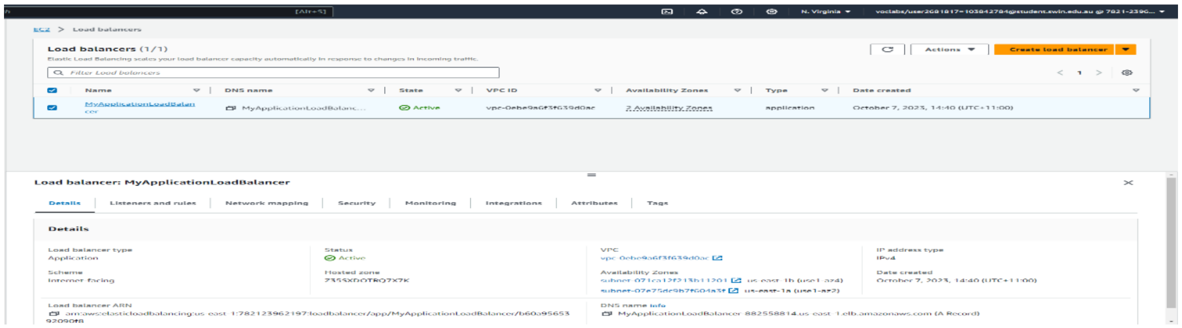
Protocol version
HTTP1

VPC
vpc-0ebe9a6f3f639d0ac

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	2	0	0	0	0

Figure 28: Target group attached to the Load Balancer.

Our Load Balancer will look like this.


EC2 > Load balancers

Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
MyApplicationLoadBalancer	MyApplicationLoadBalanc...	Active	vpc-0ebe9a6f3f639d0ac	2 Availability Zones	application	October 7, 2023, 14:40 (UTC+11:00)

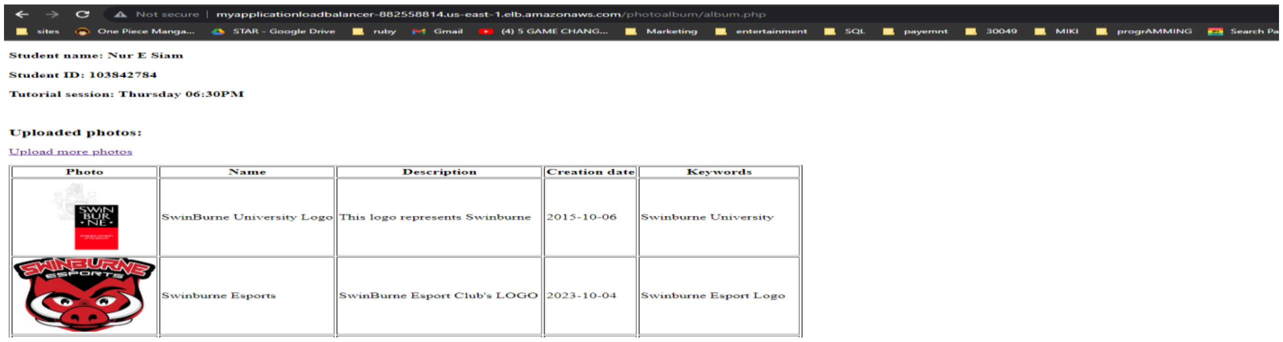
Load balancer: MyApplicationLoadBalancer

Details

Load balancer type: Application
Status: Active
VPC: vpc-0ebe9a6f3f639d0ac
IP address type: IPv4
Scheme: Internet-facing
Hosted zone: Z35SXDOTRQ7X7K
Availability Zones: subnet-071ca12f213b11201 us-east-1b (use1-az4); subnet-07e75dc9b7f604a3f us-east-1a (use1-az2)
Date created: October 7, 2023, 14:40 (UTC+11:00)
Load balancer ARN: arn:aws:elasticloadbalancing:us-east-1:782123962197:loadbalancer/app/MyApplicationLoadBalancer/b60a9565392090f8
DNS name: MyApplicationLoadBalancer-882558814.us-east-1.elb.amazonaws.com (A Record)

Figure 29: Load Balancer.

We can then use the DNS name provided by the Load Balancer to access our Web Server Instances created by the Auto Scaling Group.


Not secure | myapplicationloadbalancer-882558814.us-east-1.elb.amazonaws.com/photoalbum/album.php

Student name: Nur E Siam
Student ID: 103842784
Tutorial session: Thursday 06:30PM

Uploaded photos:
Upload more photos

Photo	Name	Description	Creation date	Keywords
	SwinBurne University Logo	This logo represents Swinburne	2015-10-06	Swinburne University
	Swinburne Esports	SwinBurne Esport Club's LOGO	2023-10-04	Swinburne Esport Logo

Figure 30: Album.php accessed from the Load Balancer.

As the S3 Bucket will only allow access from the Load Balancer that we just created, the picture of a logo will be presented. We can also upload more logos to test the functionality of the website.

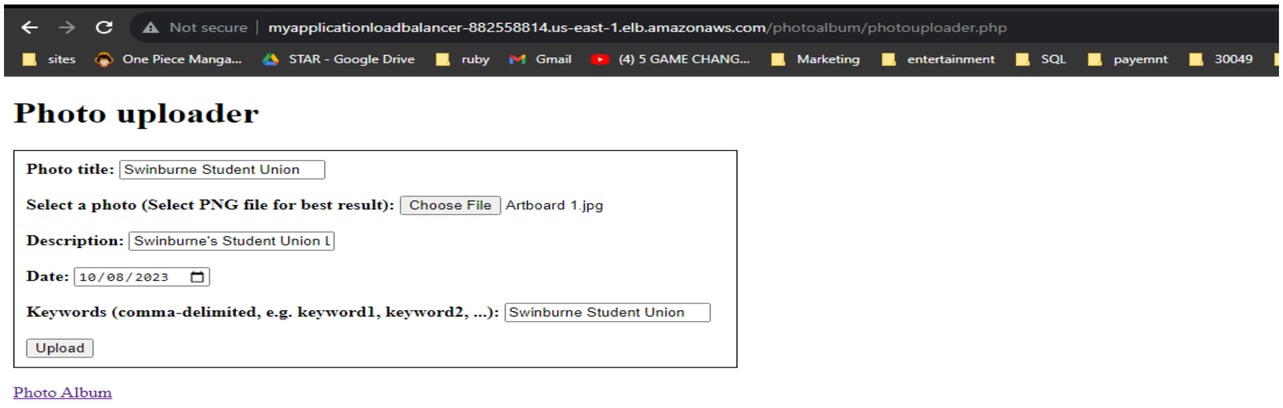
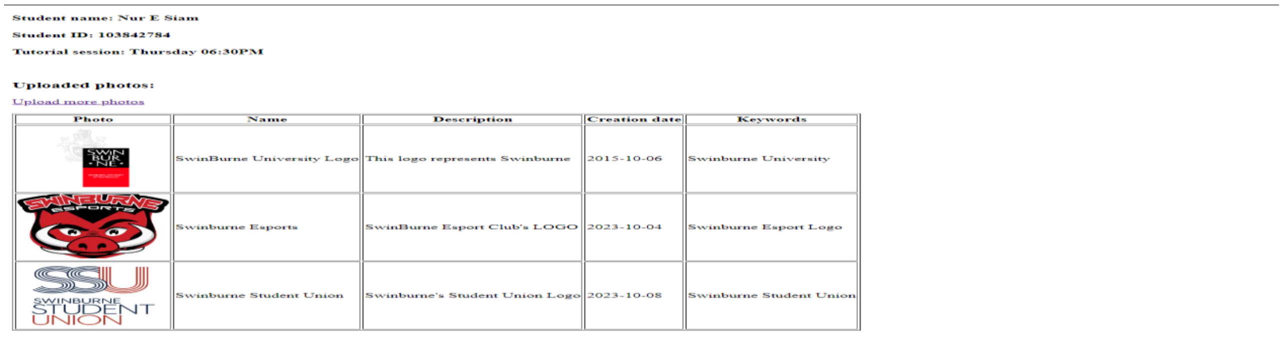

Not secure | myapplicationloadbalancer-882558814.us-east-1.elb.amazonaws.com/photoalbum/photouploader.php

Photo uploader

Photo title: Swinburne Student Union
Select a photo (Select PNG file for best result): Choose File Artboard 1.jpg
Description: Swinburne's Student Union L
Date: 10/08/2023
Keywords (comma-delimited, e.g. keyword1, keyword2, ...): Swinburne Student Union
Upload

Photo Album

Figure 31: photouploader.php accessed from the Load Balancer.


Student name: Nur E Siam
Student ID: 103842784
Tutorial session: Thursday 06:30PM

Uploaded photos:
Upload more photos

Photo	Name	Description	Creation date	Keywords
	SwinBurne University Logo	This logo represents Swinburne	2015-10-06	Swinburne University
	Swinburne Esports	SwinBurne Esport Club's LOGO	2023-10-04	Swinburne Esport Logo
	Swinburne Student Union	Swinburne's Student Union Logo	2023-10-08	Swinburne Student Union

Figure 32: Functionalities test from Load Balancer.

V. Security Group and Network ACL

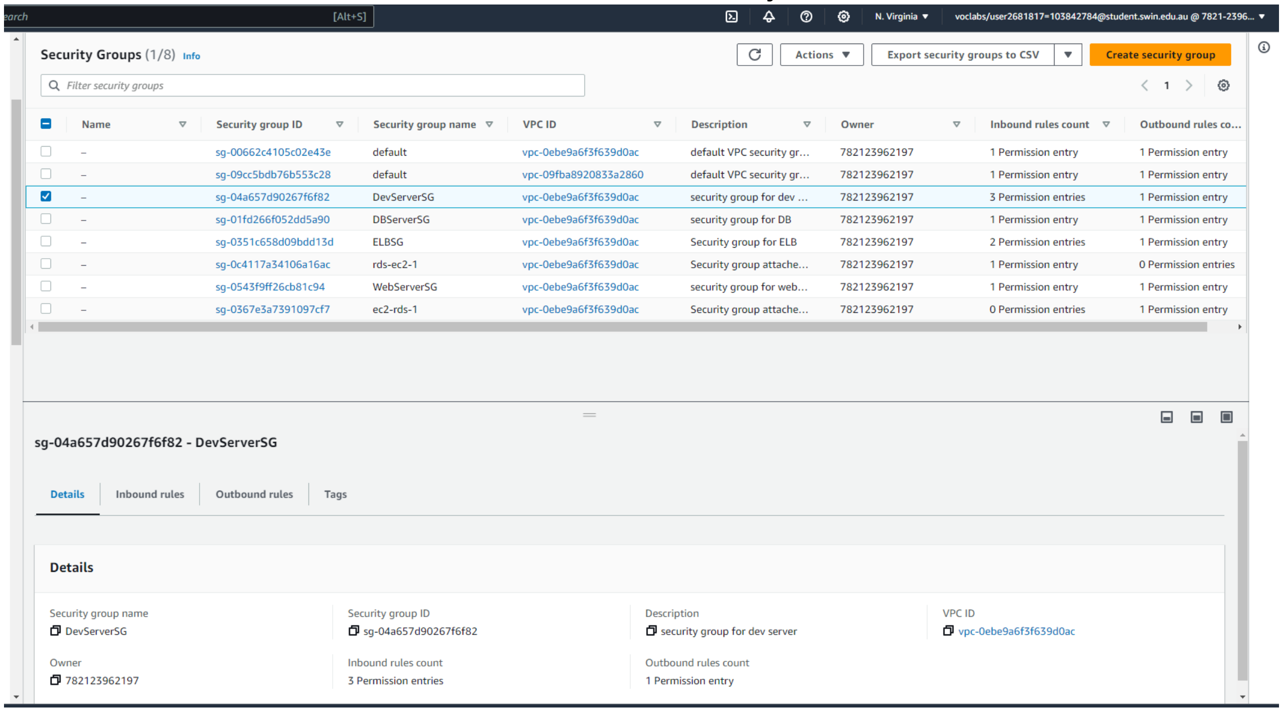
After making sure that all of our Web Server Instances are working properly, we can then proceed to ensure Security and Accessibility to and from our Web Server Instances.

Web Server Security Group should only accept Inbound from Elastic Load Balancer and Outbound to the NAT gateway.

DevServer Security Group can accept Inbound and Outbound from All Traffics.

Database Server Security Group can accept all Inbound and Outbound traffics from the Webserver and the Devserver.

The Elastic Load Balancer Security Group can accept all Inbound and Outbound Traffics from the Internet Gateway.



Security Groups (1/8) Info

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules co...
–	sg-00662c4105c02e43e	default	vpc-0ebe9a6f3f639d0ac	default VPC security gr...	782123962197	1 Permission entry	1 Permission entry
–	sg-09cc5bdb76b553c28	default	vpc-09fba8920833a2860	default VPC security gr...	782123962197	1 Permission entry	1 Permission entry
–	sg-04a657d90267f6f82	DevServerSG	vpc-0ebe9a6f3f639d0ac	security group for dev ...	782123962197	3 Permission entries	1 Permission entry
–	sg-01fd266f052dd5a90	DBServerSG	vpc-0ebe9a6f3f639d0ac	security group for DB	782123962197	1 Permission entry	1 Permission entry
–	sg-0351c658d09bdd13d	ELBSG	vpc-0ebe9a6f3f639d0ac	Security group for ELB	782123962197	2 Permission entries	1 Permission entry
–	sg-0c4117a34106a16ac	rds-ec2-1	vpc-0ebe9a6f3f639d0ac	Security group attache...	782123962197	1 Permission entry	0 Permission entries
–	sg-0543f9ff26cb81c94	WebServerSG	vpc-0ebe9a6f3f639d0ac	security group for web...	782123962197	1 Permission entry	1 Permission entry
–	sg-0367e3a7391097cf7	ec2-rds-1	vpc-0ebe9a6f3f639d0ac	Security group attache...	782123962197	0 Permission entries	1 Permission entry

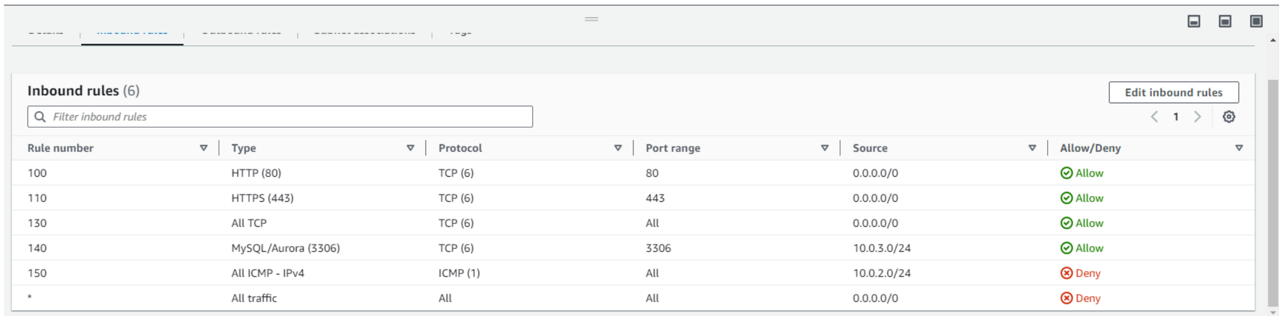
sg-04a657d90267f6f82 - DevServerSG

Details

Security group name: DevServerSG	Security group ID: sg-04a657d90267f6f82	Description: security group for dev server	VPC ID: vpc-0ebe9a6f3f639d0ac
Owner: 782123962197	Inbound rules count: 3 Permission entries	Outbound rules count: 1 Permission entry	

Figure 33: Security Groups

Network ACL is going to be the last part of our configuration, we will create an ACL that will restrict DevServer from sending ICMP packet to the WebServer.
The Inbound and Outbound rules for our Network ACL will look like this.



Inbound rules (6)

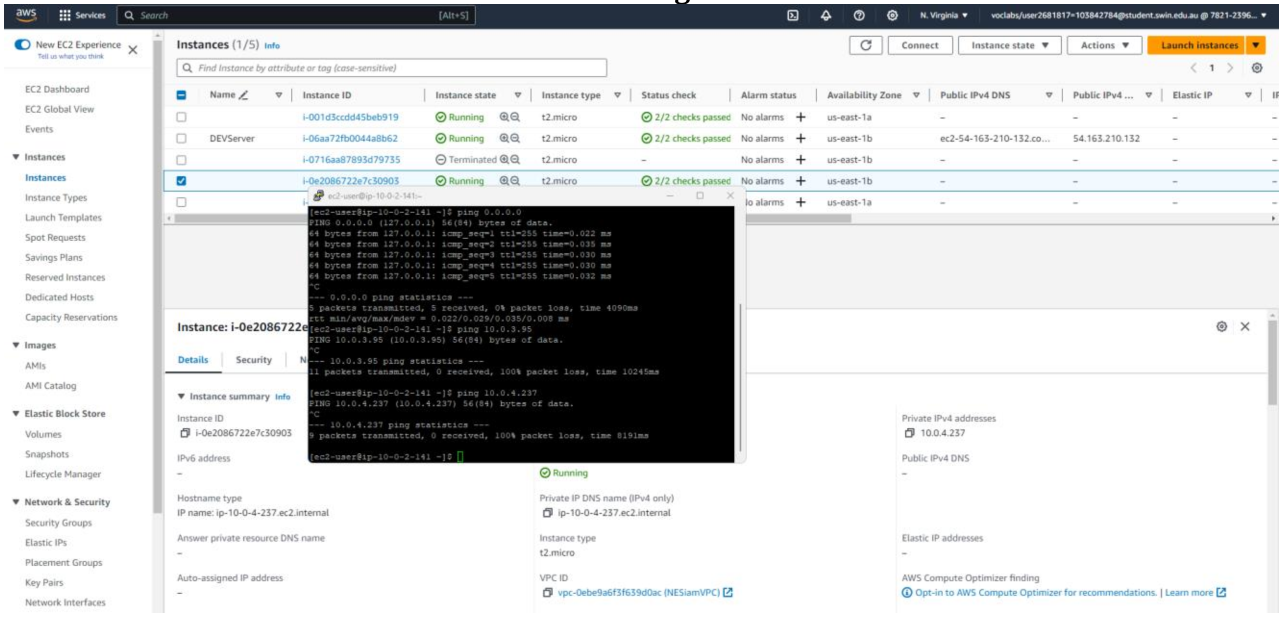
Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
110	HTTPS (443)	TCP (6)	443	0.0.0.0/0	Allow
130	All TCP	TCP (6)	All	0.0.0.0/0	Allow
140	MySQL/Aurora (3306)	TCP (6)	3306	10.0.3.0/24	Allow
150	All ICMP - IPv4	ICMP (1)	All	10.0.2.0/24	Deny
*	All traffic	All	All	0.0.0.0/0	Deny

Figure 34: Network ACLs.

VI. Testing

We will conduct several tests to verify that the website's functionality has been properly configured. While some tests have already been performed earlier, this section will cover the remaining tests.

Test 1: Ping Test



Instances (1/5)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☐		i-001d3ccdd45beb919	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	–	–	–
☐	DEVServer	i-06aa72fb0044a8b62	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-163-210-132.co...	54.163.210.132	–
☐		i-0716aa87893d79735	Terminated	t2.micro	–	No alarms	us-east-1b	–	–	–
☑		i-0e2086722e7c30903	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	–	–	–
☐						No alarms	us-east-1a	–	–	–

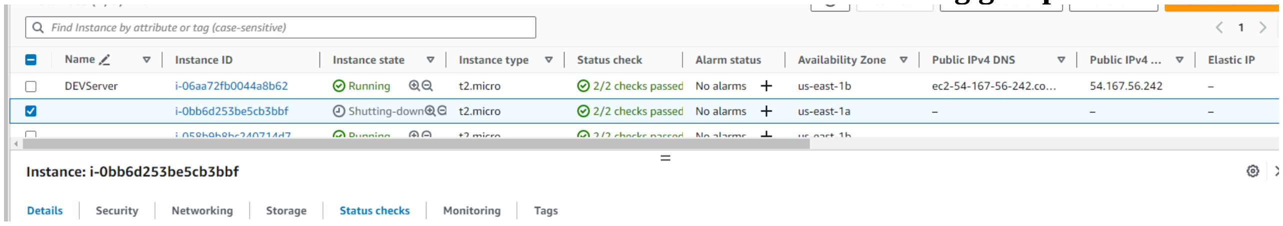
```
[ec2-user@ip-10-0-2-141 ~]$ ping 0.0.0.0
PING 0.0.0.0 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=255 time=0.022 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=255 time=0.035 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=255 time=0.030 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=255 time=0.030 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=255 time=0.032 ms
^C
--- 0.0.0.0 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4090ms
rtt min/avg/max/mdev = 0.022/0.029/0.035/0.008 ms
[ec2-user@ip-10-0-2-141 ~]$ ping 10.0.3.95
PING 10.0.3.95 (10.0.3.95) 56(84) bytes of data.
^C
--- 10.0.3.95 ping statistics ---
11 packets transmitted, 0 received, 100% packet loss, time 10245ms

[ec2-user@ip-10-0-2-141 ~]$ ping 10.0.4.237
PING 10.0.4.237 (10.0.4.237) 56(84) bytes of data.
^C
--- 10.0.4.237 ping statistics ---
9 packets transmitted, 0 received, 100% packet loss, time 8191ms

[ec2-user@ip-10-0-2-141 ~]$
```

Instance: i-0e2086722e7c30903

Instance ID: i-0e2086722e7c30903
Private IPv4 addresses: 10.0.4.237
Instance state: Running
Hostname type: IP name: ip-10-0-4-237.ec2.internal
Private IP DNS name (IPv4 only): ip-10-0-4-237.ec2.internal
Instance type: t2.micro
VPC ID: vpc-0ebe9a6f3f639d0ac (NESiamVPC)
AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Figure 35: All ICMP Testing.

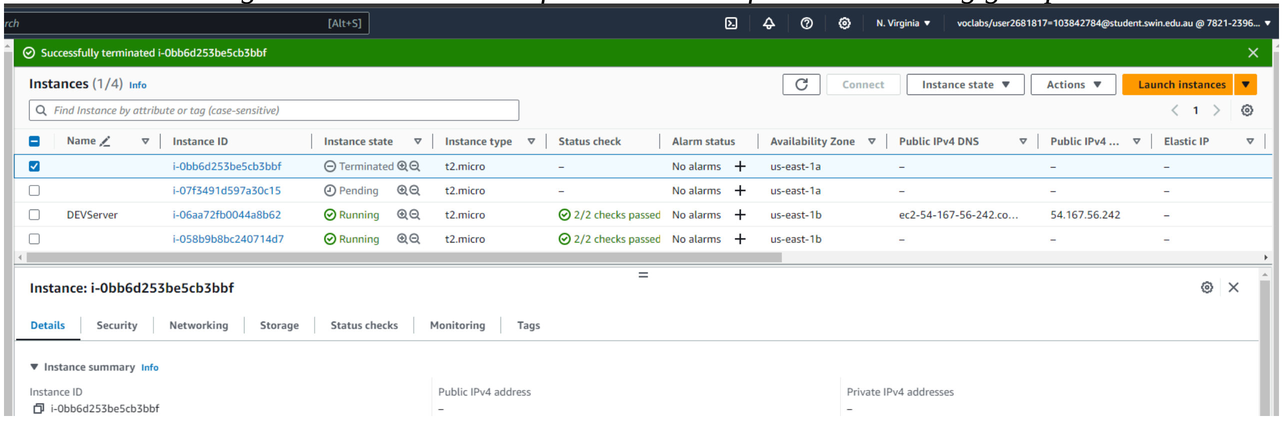
Test 2: Termination of Auto instance and Auto Scaling group



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☐	DEVServer	i-06aa72fb0044a8b62	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-167-56-242.co...	54.167.56.242	–
☑		i-0bb6d253be5cb3bbf	Shutting-down	t2.micro	2/2 checks passed	No alarms	us-east-1a	–	–	–

Instance: i-0bb6d253be5cb3bbf

Figure 36: Termination of a webinstance from Auto scaling group.



Successfully terminated i-0bb6d253be5cb3bbf

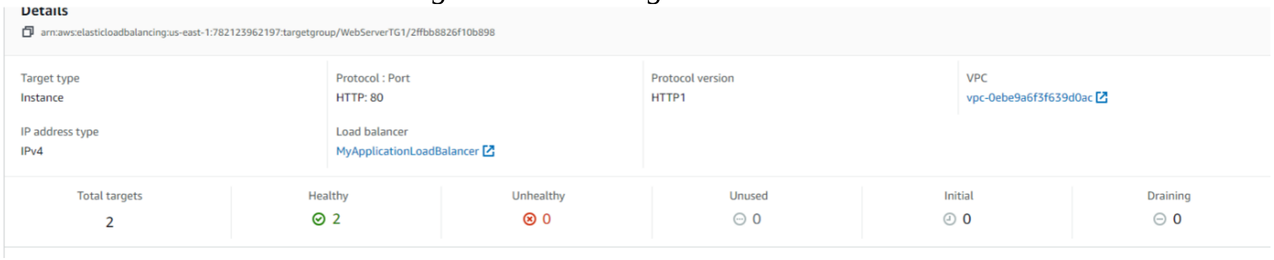
Instances (1/4)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☑		i-0bb6d253be5cb3bbf	Terminated	t2.micro	–	No alarms	us-east-1a	–	–	–
☐		i-07f3491d597a30c15	Pending	t2.micro	–	No alarms	us-east-1a	–	–	–
☐	DEVServer	i-06aa72fb0044a8b62	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-167-56-242.co...	54.167.56.242	–
☐		i-058b9b8bc240714d7	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	–	–	–

Instance: i-0bb6d253be5cb3bbf

Instance ID: i-0bb6d253be5cb3bbf
Public IPv4 address: –
Private IPv4 addresses: –

Figure 37: Creating a webinstance.



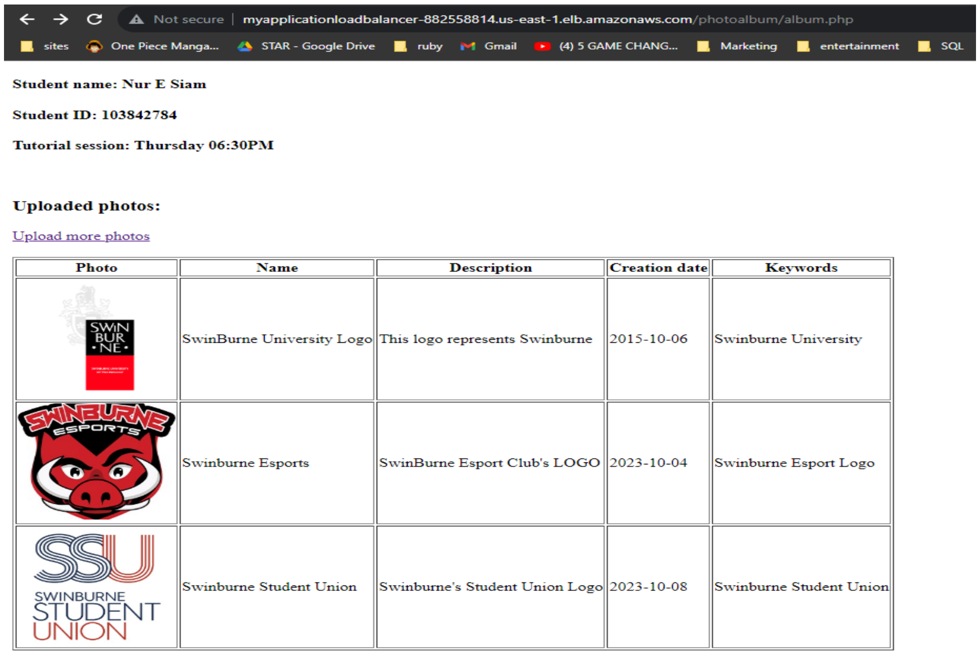
Details

arn:aws:elasticloadbalancing:us-east-1:782123962197:targetgroup/WebServerTG1/2ffbb8826f10b898

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-0ebe9a6f3f639d0ac
IP address type	**Load balancer**		
IPv4	MyApplicationLoadBalancer		

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	2	0	0	0	0

Figure 38: Both instances are Healthy.



Not secure | myapplicationloadbalancer-882558814.us-east-1.elb.amazonaws.com/photoalbum/album.php

Student name: Nur E Siam

Student ID: 103842784

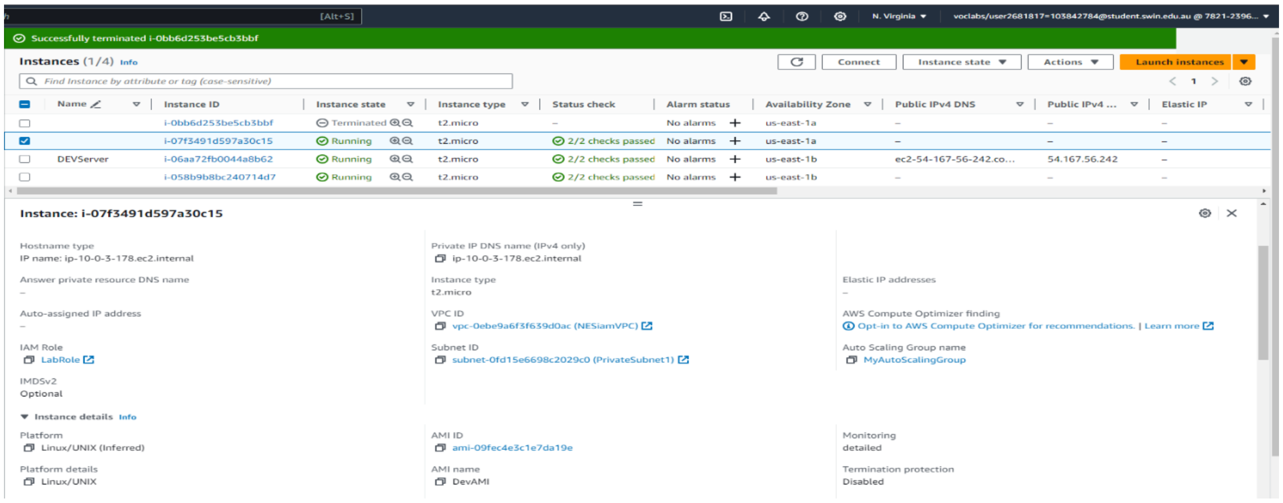
Tutorial session: Thursday 06:30PM

Uploaded photos:

Upload more photos

Photo	Name	Description	Creation date	Keywords
	SwinBurne University Logo	This logo represents Swinburne	2015-10-06	Swinburne University
	Swinburne Esports	SwinBurne Esport Club's LOGO	2023-10-04	Swinburne Esport Logo
	Swinburne Student Union	Swinburne's Student Union Logo	2023-10-08	Swinburne Student Union

Figure 39: The LB DNS is still accessible after the termination and initialization of new Web instance.



Successfully terminated i-0bb6d253be5cb3bbf

Instances (1/4)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
	i-0bb6d253be5cb3bbf	Terminated	t2.micro	–	No alarms	us-east-1a	–	–	–
	i-07f3491d597a30c15	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	–	–	–
DEVServer	i-06aa72fb0044a8b62	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-167-56-242.co...	54.167.56.242	–
	i-058b9b8bc240714d7	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	–	–	–

Instance: i-07f3491d597a30c15

Hostname type: IP name: ip-10-0-3-178.ec2.internal

Answer private resource DNS name: –

Auto-assigned IP address: –

IAM Role: LabRole

IMDSv2: Optional

Private IP DNS name (IPv4 only): ip-10-0-3-178.ec2.internal

Instance type: t2.micro

VPC ID: vpc-0ebe9a6f3f639d0ac (NESiamVPC)

Subnet ID: subnet-0fd15e6698c2029c0 (PrivateSubnet1)

Elastic IP addresses: –

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name: MyAutoScalingGroup

Instance details

Platform: Linux/UNIX (Inferred)

Platform details: Linux/UNIX

AMI ID: ami-09fec4e3c1e7da19e

AMI name: DevAMI

Monitoring: detailed

Termination protection: Disabled

Figure 40: The new Web instance have the same Security and IAM role.

References

Link to the ELB album.php: http://myapplicationloadbalancer-882558814.us-east-1.elb.amazonaws.com/photoalbum/album.php

Link to the ELB photouploader.php: http://myapplicationloadbalancer-882558814.us-east-1.elb.amazonaws.com/photoalbum/photouploader.php

Link to the DevServer: http://ec2-54-167-56-242.compute-1.amazonaws.com/