

Creating and deploying Photo Album website onto a simple AWS infrastructure

Name: Shaugato Paroi
University: Swinburne University Of Technology
Stuent ID: 103523487
Tutorial: Friday 12:30pm-2:30pm
link to website: [Photo Album](#)
Date of Submission : 15/04/2024

I. INTRODUCTION

In this task, our task is to create a secure Virtual Private Cloud that will have 2 availability zones (Availability Zone A, and Availability Zone B), 2 Public subnets, and 2 private subnets. It will have 4 security groups and a Network Access Control Policy to provide security when accessing the services in this VPC. There will be a web server hosted on the public subnet, a database service (RDS instance), and a test instance that will use the two other private subnets. There will be a PHP website with MySQL database hosted on that RDS Instance. It will manage the photo metadata.

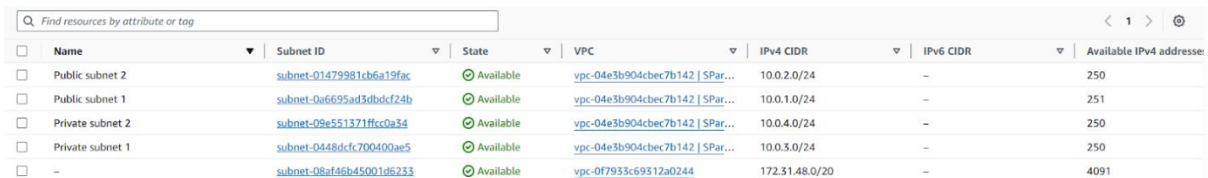
This report will describe how each object can be created and connected. This document will also how we can use Amazon S3 for storage purposes

II. CREATING THE VPC

A. Selecting a Template (Heading 2)

The first step is to create the VPC called “SParoi” in the N.Virginia us-east-1

1. Go to VPC and create the VPC using “VPC only”. The CIDR block will be 10.0.0.0/16 as given in the assignment
2. Then we will create four subnets by going to the subnets: Public Subnet 1 - 10.0.1.0/24, Public subnet 2 - 10.0.2.0/24, Private Subnet 1 -10.0.3.0/24, Private Subnet 2 - 10.0.4.0/24



	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses
<input type="checkbox"/>	Public subnet 2	subnet-01479981cb6a19fac	Available	vpc-04e3b904bec7b142 SParoi	10.0.2.0/24	–	250
<input type="checkbox"/>	Public subnet 1	subnet-0a6695ad3dbdcf24b	Available	vpc-04e3b904bec7b142 SParoi	10.0.1.0/24	–	251
<input type="checkbox"/>	Private subnet 2	subnet-09e551371ffcc0a34	Available	vpc-04e3b904bec7b142 SParoi	10.0.4.0/24	–	250
<input type="checkbox"/>	Private subnet 1	subnet-0448dcfc700400ae5	Available	vpc-04e3b904bec7b142 SParoi	10.0.3.0/24	–	250
<input type="checkbox"/>	–	subnet-08af46bd5001d6233	Available	vpc-0f7933c69312a0244	172.31.48.0/20	–	4091

Figure 1 - Creating the Four subnets

- Go to the Route table and create the two route table in two zones us-east-1a, and us-east-1b (Public Route Table, Private Route Table) . Then add two public subnet (Public Subnet 1 , Public subnet 2) to Public Route Table and add two private subnet (Private Subnet 1 and Private Subnet 2) to Private route table .

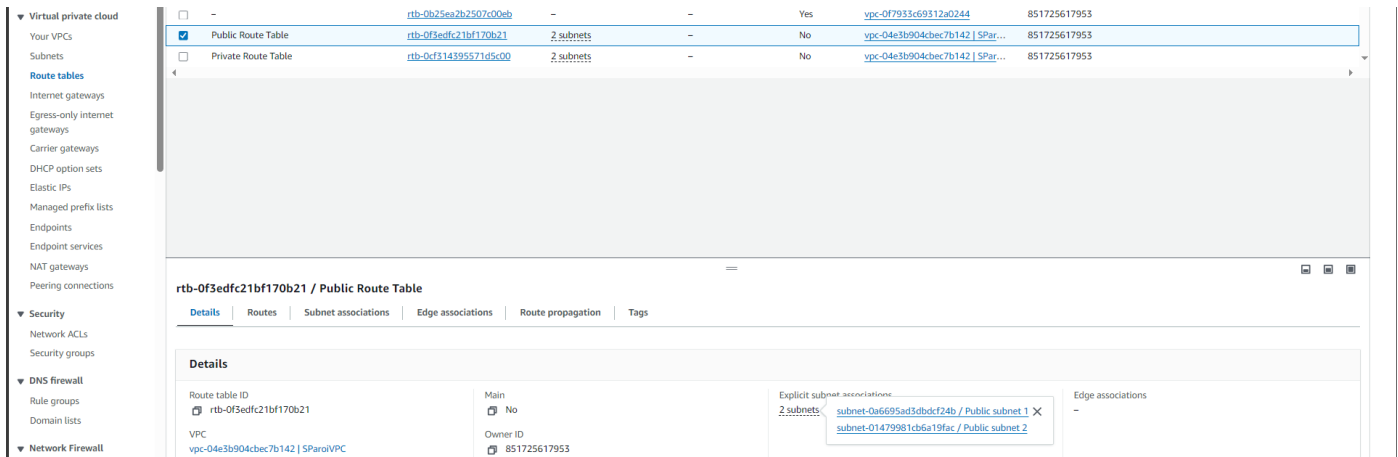


Figure 2 - Creating Route tables

- Go to the Internet Gateways and create an Internet Gateways called “My Internet Gateway” and attach it to the “SParoiVPC” Public route table which allows resources in public transports to internet.

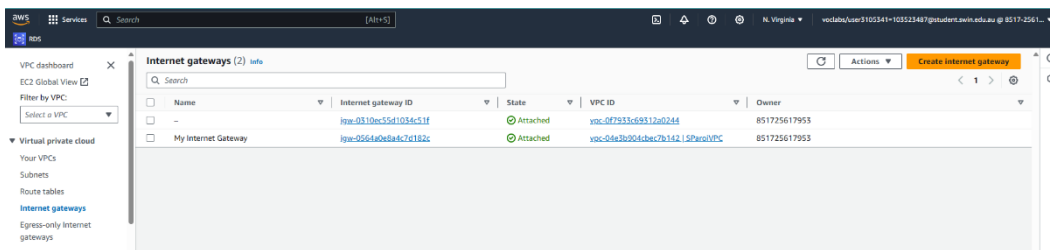


Figure 3 -Creating Internet Gateway

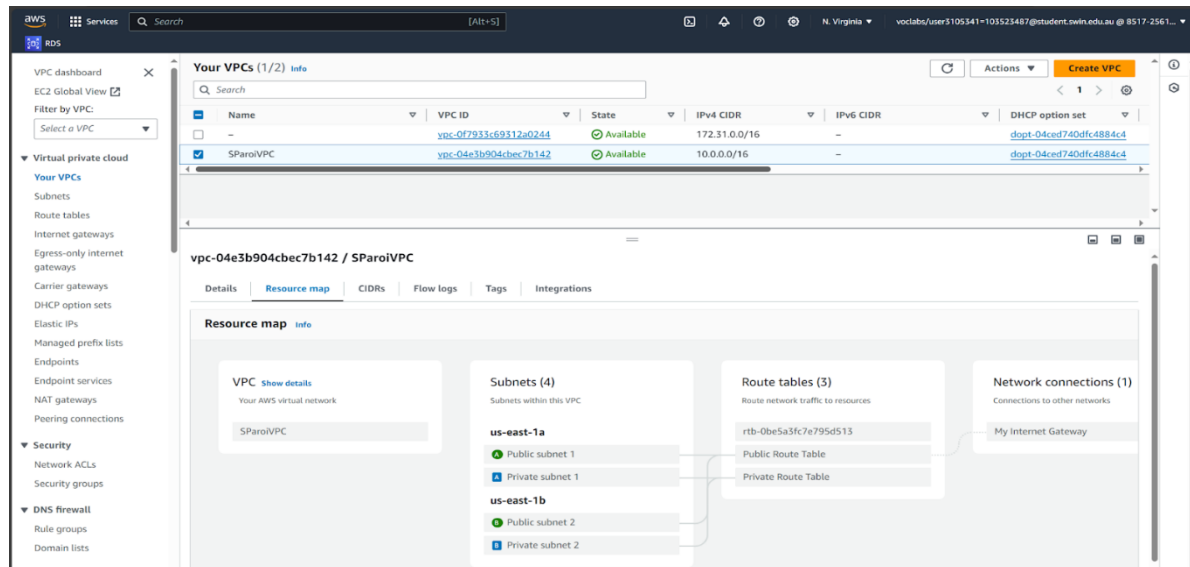


Figure 4 - The VPC

III. CREATING SECURITY GROUPS

As per the assignment requirement, we will create 3 security groups by going to Security groups and on VPC dashboard.

1. TestInstanceSG (Allowing All traffic)
2. WebServerSG (Allowing HTTP, SSH from **anywhere**, allowing ICMP from only TestInstanceSG)
3. DBServerSG (Allowing MySQL/Aqora from WebServerSG)

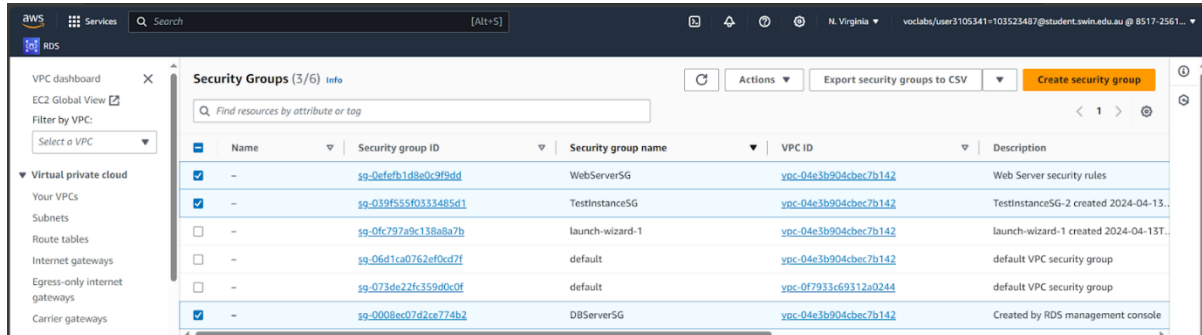


Figure 5 - Creating Security Groups

IV. CREATING EC2 INSTANCES

A. Bastion/Web Instance:

1. Go to EC2 → Instance → Launch Instance
2. Edit the field accordingly : Name - Bastion/Web , Amazon Machine Image (Amazon Linux 2 AMI Kernel 5), Instance type - t2.micro, Key pair - Bastion_web (ppk, by creating a new key pair)
3. Edit Network Setting Accordingly : VPC - SPaoviVPC , Subnet - 10.0.2.0/24 (Public Subnet 2 as per the requirements) , Firewall - WebServerSG (choosing it by clicking Select Existing Security Groups)
4. As per the requirements , in order to install the Apache web server and other php package , we put the following codes on the user data section in Advanced details . Then Launch instance.

```
#/bin/bash
yum update -y
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
service httpd start
yum install -y httpd mariadb-server php-mbstring php-xml
sed -i "s/upload_max_filesize = 2M/upload_max_filesize = 10M/g" /etc/php.ini
systemctl start httpd
systemctl enable httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec sudo chmod 2775 {} \;
find /var/www -type f -exec sudo chmod 0664 {} \;
echo "<?php echo '<h2>Welcome to COS80001. Installed PHP version: ' . phpversion() . '</h2>'; ?>" > /var/www/html/phpinfo.php
```

Figure 6 - Apache server and PHP installation Code

5. Allocate an Elastic Ip address and attach it the EC2 Bastion/Web as per the requirement. The elastic Ip address is **18.233.199.68**

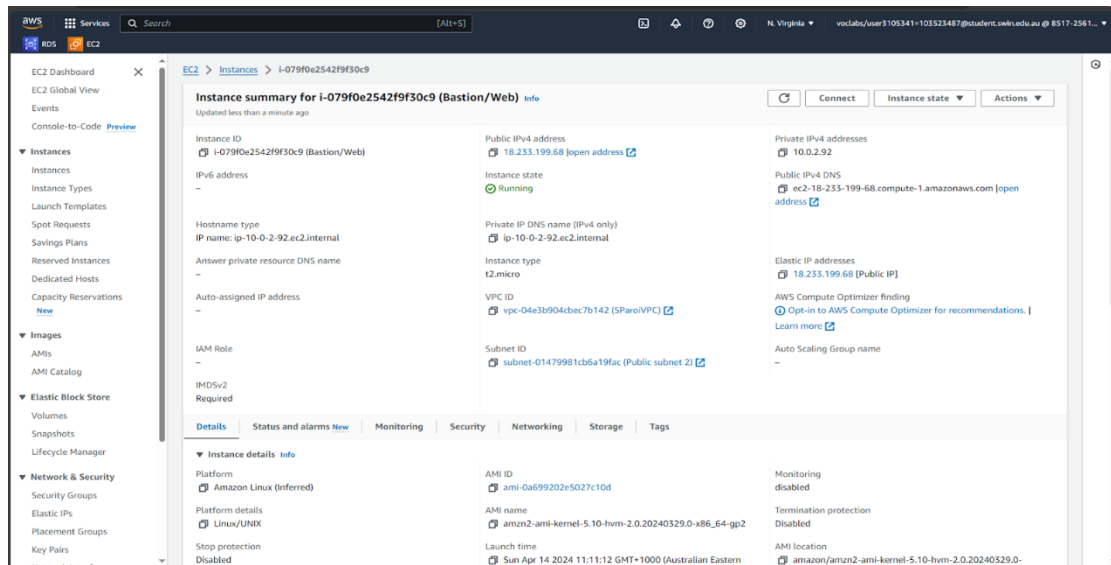


Figure 7 - Bastion/Web Instance

B. Test Instance

- We create the test instance the same way we create the Bastion/Web instance
- We use Test_intance as key pair, 10.0.4.0/24 (Private subnet 2 as per the requirements) as subnet , TestInstanceSG for security groups here

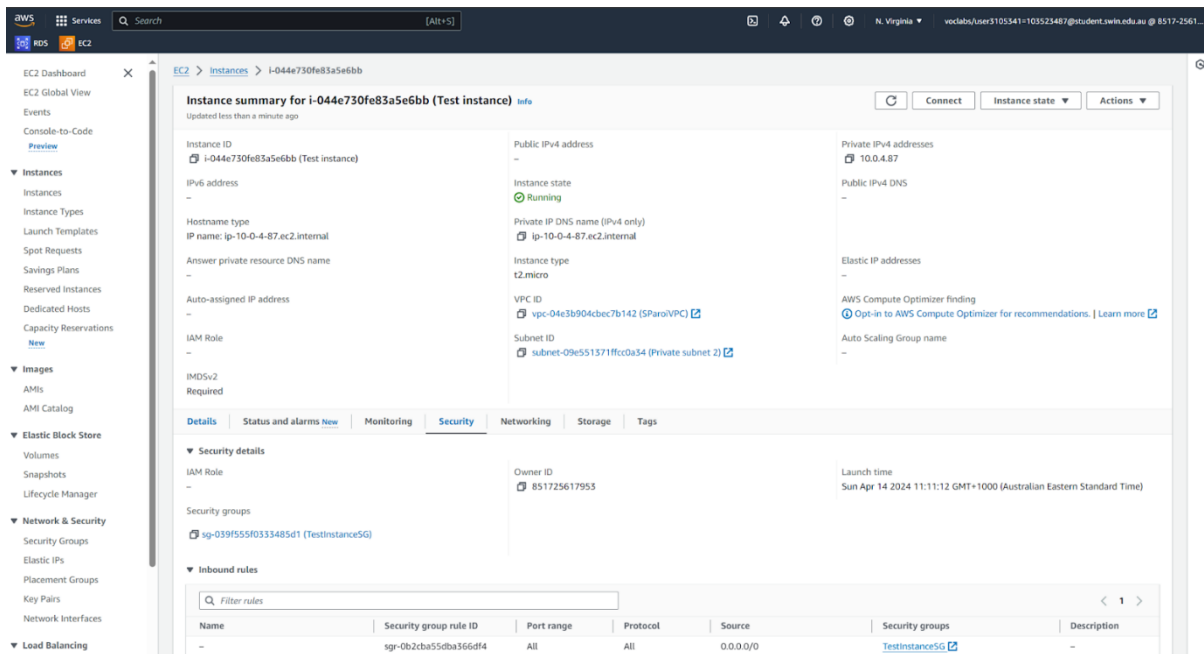


Figure 8- Test Instance

C. SSH into Test Instance From Bastion/Web Instance

By following the steps from <https://aws.amazon.com/blogs/security/securely-connect-to-linux-instances-running-in-a-private-vpc/> [1] we establish a SSH connection to Test Instance form Bastion/Web instance:

```
ec2-user@ip-10-0-2-92~  
login as: ec2-user  
Authenticating with public key "Bastion_Web"  
  
#  
~\##### 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-10-0-2-92 ~]$  
[ec2-user@ip-10-0-2-92 ~]$  
[ec2-user@ip-10-0-2-92 ~]$  
[ec2-user@ip-10-0-2-92 ~]$  
[ec2-user@ip-10-0-2-92 ~]$  
[ec2-user@ip-10-0-2-92 ~]$ ssh user@10.0.4.87  
The authenticity of host '10.0.4.87 (10.0.4.87)' can't be established.  
ECDSA key fingerprint is SHA256:RK+KZ/7SL8ZSweuXNko6IBX4mwncrlwOBH25wl74loc.  
ECDSA key fingerprint is MD5:82:97:ce:26:07:39:59:b6:27:40:ec:3c:78:85:c3:c8.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '10.0.4.87' (ECDSA) to the list of known hosts.  
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).  
[ec2-user@ip-10-0-2-92 ~]$  
[ec2-user@ip-10-0-2-92 ~]$
```

Figure 10 - SSH into Test Instance (10.0.4.87) from Boston/Web instance

D. ICMP ping request to Bastion/Web instance from Test Instance

Now as we are already in the Test instance by establishing the ssh request , we ping to Bastion/Web from Test instance for testing purpose as per the requirements.

```

ec2-user@ip-10-0-0-2-92:~
└─/m/' https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-0-2-92 ~]$
[ec2-user@ip-10-0-0-2-92 ~]$
[ec2-user@ip-10-0-0-2-92 ~]$
[ec2-user@ip-10-0-0-2-92 ~]$
[ec2-user@ip-10-0-0-2-92 ~]$
[ec2-user@ip-10-0-0-2-92 ~]$
[ec2-user@ip-10-0-0-2-92 ~]$ ssh user@10.0.4.87
The authenticity of host '10.0.4.87 (10.0.4.87)' can't be established.
ECDSA key fingerprint is SHA256:RK+KZ/75L8ZSweuXNko6IBX4nmwr1oWBHz5wL741oc.
ECDSA key fingerprint is MD5:82:97:ce:26:07:39:59:b6:27:40:ec:c3c:78:85:c3:c8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.4.87' (ECDSA) to the list of known hosts.
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-0-0-2-92 ~]$
[ec2-user@ip-10-0-0-2-92 ~]$ ping 18.233.199.68
PING 18.233.199.68 (18.233.199.68) 56(84) bytes of data.
64 bytes from 18.233.199.68: icmp_seq=1 ttl=254 time=0.492 ms
64 bytes from 18.233.199.68: icmp_seq=2 ttl=254 time=0.504 ms
64 bytes from 18.233.199.68: icmp_seq=3 ttl=254 time=0.511 ms
64 bytes from 18.233.199.68: icmp_seq=4 ttl=254 time=0.515 ms
64 bytes from 18.233.199.68: icmp_seq=5 ttl=254 time=0.480 ms
64 bytes from 18.233.199.68: icmp_seq=6 ttl=254 time=0.553 ms
^C
--- 18.233.199.68 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5077ms
rtt min/avg/max/mdev = 0.480/0.509/0.553/0.026 ms
[ec2-user@ip-10-0-0-2-92 ~]$

```

Figure 11 - Ping from Test Instance (10.0.4.87) to Boston/Web instance (18.233.199.68)

V. CREATING RDS DATABASE INSTANCE

A. Creating Subnet Group For RDS instance

Before Creating the RDS Database , I create a Subnet Group called “sparoi-db-subnet-group” so that I can attach it later to the rds instance

1. Go to the Subnet Group from RDS dashboard and click Create DB subnet Group

- Edit the field like the following : Name - sparoi-db-subnet-group, VPC - SParoiVPC, Availability Zone - us-east-1a, us-east-1b , subnets - 10.0.3.0/24,10.0.4.0/24 (Private subnet 1, Private subnet 2 , Selecting both private subnet as the requirement was to keep in private subnet) . Then Create.

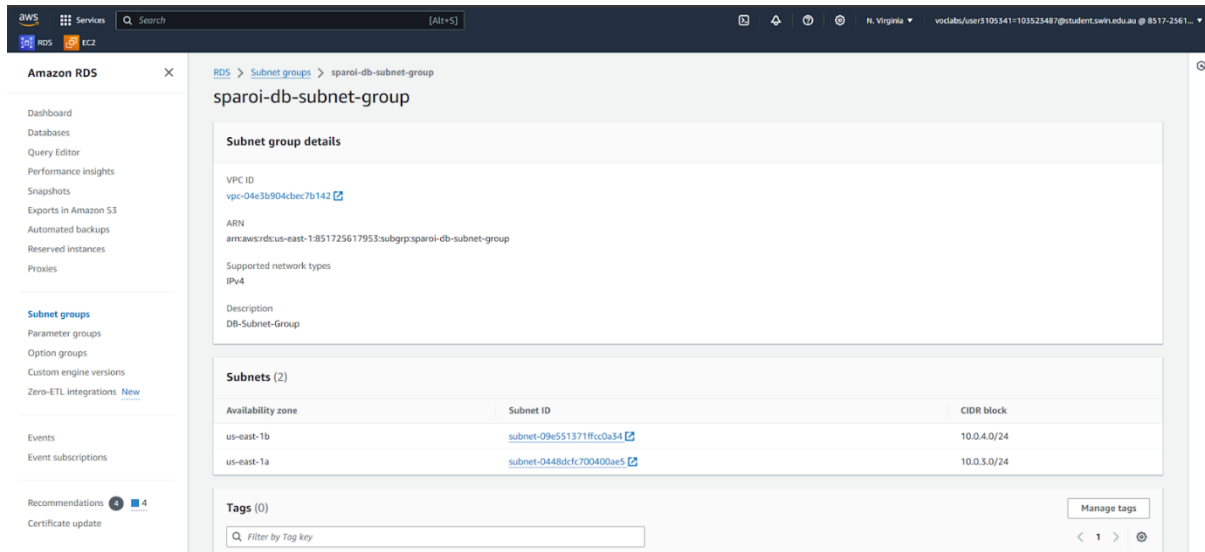


Figure 11 - Creating SParoi DB Subnet Group

B. Creating RDS Database Instance

- From RDS service on Databases option, Click on Create Database
- Edit the fields like the following : Database creation method : Standard Create, Engine - MySQL, Engine Version - MySQL 8.0.34, Templates - Free Tier, DB cluster identifier – “database-1”, Credential Setting - Self-managed , Master Username - admin, master password - admin123
- On connectivity: Virtual Private Cloud - SParoiVPC, DB Subnet group - sparoi-db-subnet-group, VPC security groups - DBServerSG . Then Create Database

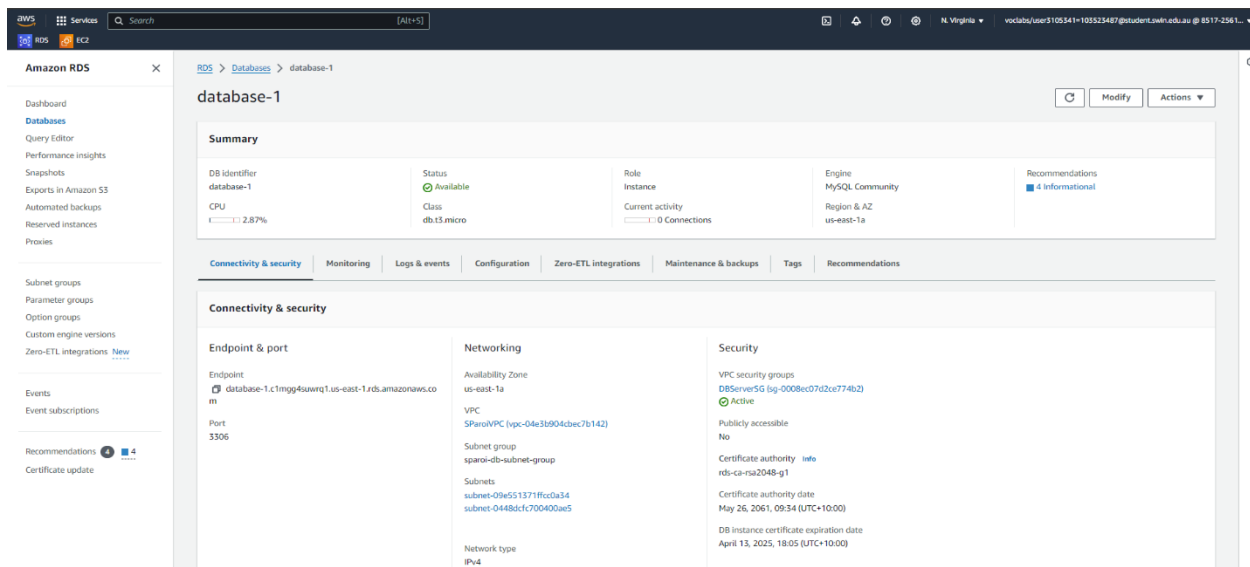


Figure 12 - Created 'database-1' RDS instance

VI. INSTALLING PHPMYADMIN

For Installing Phpmyadmin I took the following steps

1. SSH to Bastion/Web using putty , then using `cd /var/www/html` navigate to the html directory
2. Download the phpmyadmin file using `wget` command and unzip it .
3. After that I open WinSCP and connect it to Bastion/Web instance and then got to the `var/www/html/phpmyadmin` . Then I changed “`config.sample.inc.php`” to “`config.inc.php`”

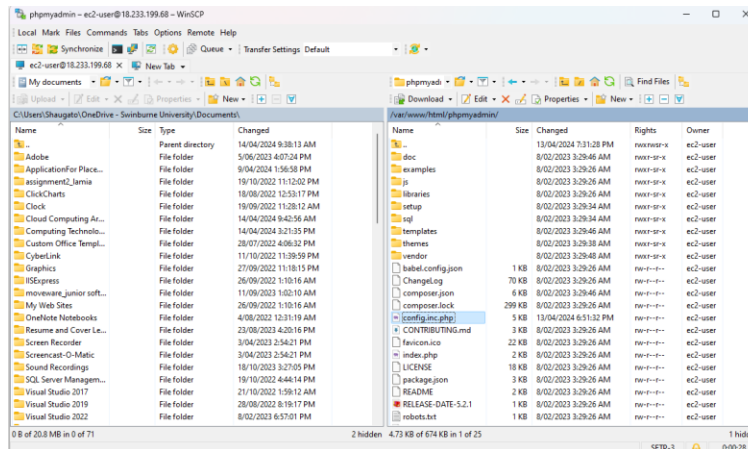


Figure 13 - Changing “`config.sample.inc.php`” to “`config.inc.php`”

4. Then inside “`config.inc.php`” changing the line `$cfg['Servers'][$i]['host'] = 'localhost';` to `$cfg['Servers'][$i]['host'] = 'database-1.c1mgg4suwrq1.us-east-1.rds.amazonaws.com';`
Here 'database-1.c1mgg4suwrq1.us-east-1.rds.amazonaws.com' is my RDS database instance endpoint

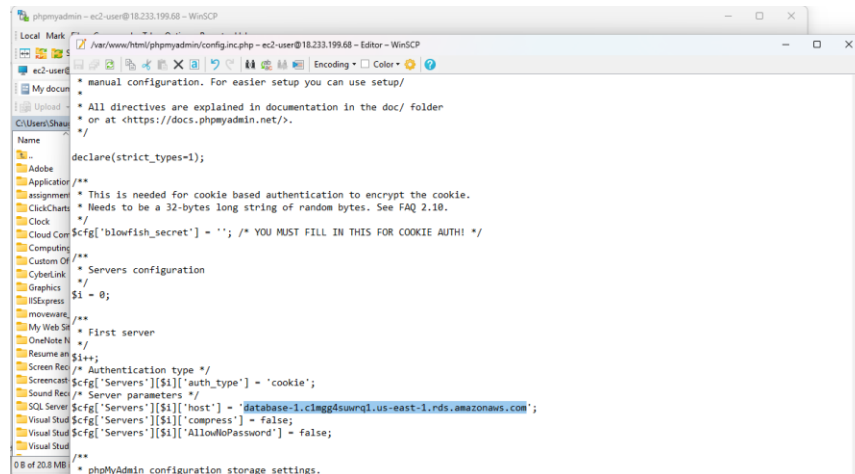


Figure 14 - hanging config in “`config.sample.inc.php`”

VII. CREATING DATABASE

As I already configure phpmyadmin on Bastion/web instance , I am able to login to the phpmyadmin server using the Bastion/Web instance ip 18.233.199.68 .

The Url: phpMyAdmin

Username: admin

Password: admin123

After that I created a new database there called “photos” . Then in that database I created a table called “photos” (So for a note my database and table both are named “photos”) as well. I created the “photos” table as per the requirement:

1. Photo_title - varchar (255)
2. Description - varchar (255)
3. Creation_data - date
4. Keywords - varchar(255)
5. Reference_to_s3 - varchar (255)

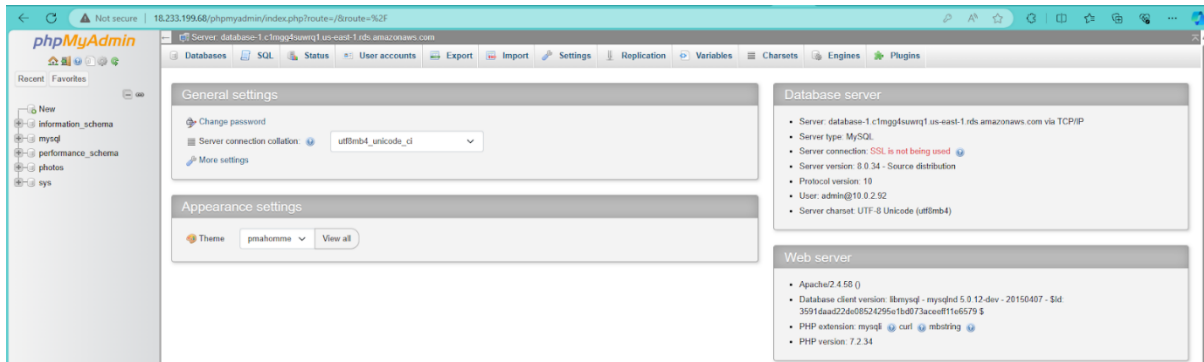


Figure 15 - phpmyadmin website

After that I created a new database there called “photos” . Then in that database I created a table called photos as well. I created the “photos” table as per the requirement:

1. Photo_title - varchar (255)
2. Description - varchar (255)
3. Creation_data - date
4. Keywords - varchar(255)
5. Reference_to_s3 - varchar (255)

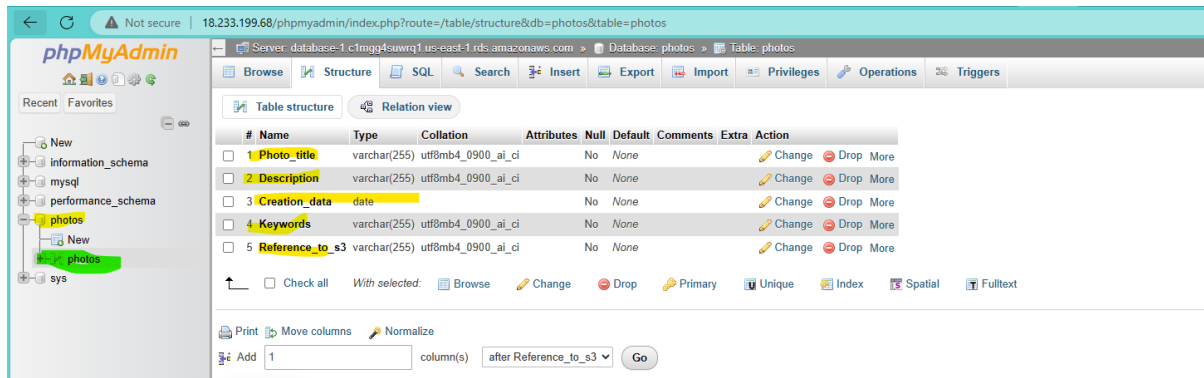


Figure 16 - Figure 16 - Creating “photos” table in “photos” database

VIII. CREATING AND SETTING UP S3 BUCKET:

For creating a S3 bucket called “sparois3bucket” i took the following steps:

1. Go to Amazon S3 service , click on Buckets then click “create bucket”
2. Edit the field like this : Bucket type - General , Bucket name - sparois3bucket, enable public access by unchecking “Block all public address “ , region - US East (N. Virginia) us-east-1. Then create bucket

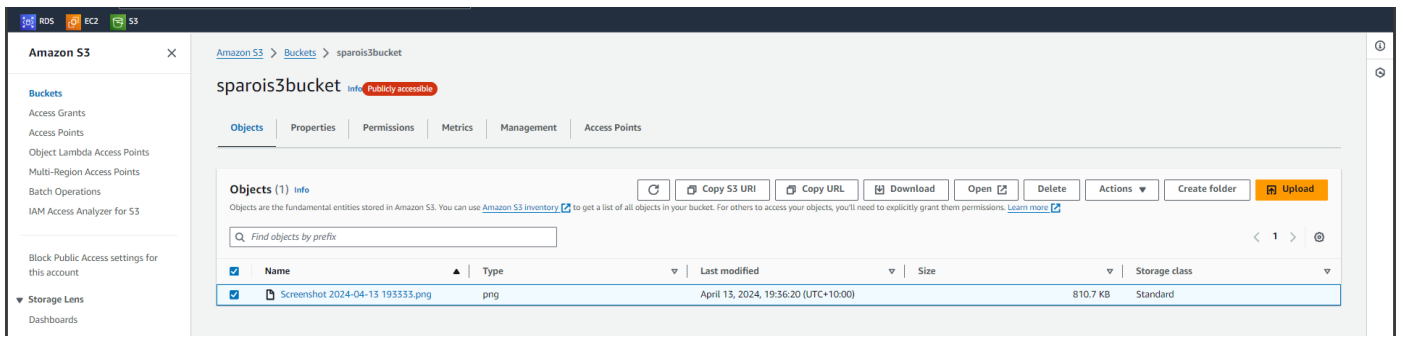


Figure 17 - Created the bucket

- After creating the bucket I added a “bucket policy” by selecting the “sparois3bucket” and going in the permission section [2]

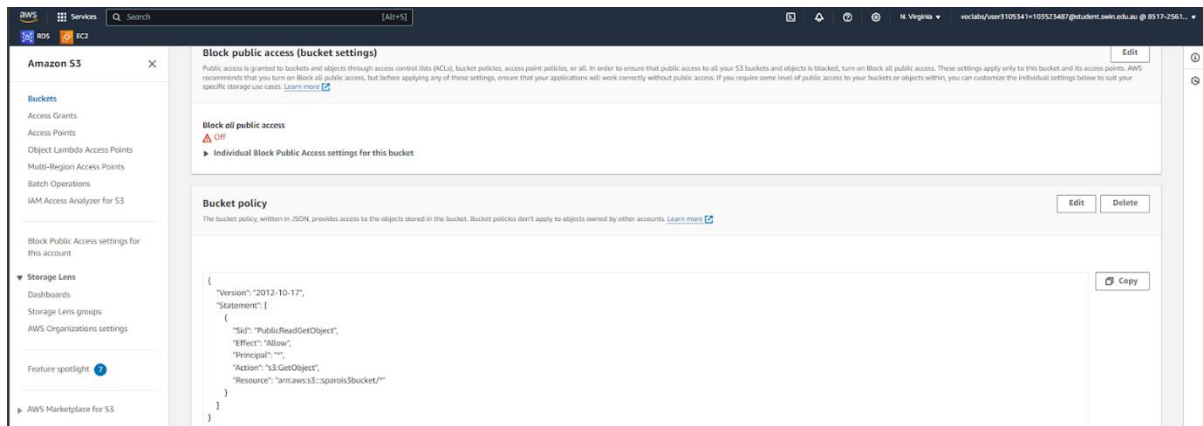


Figure 18 - Creating the Bucket Policy

- After that I took a screen shot of swinburne image and added it to the “sparois3bucket” as a object , so that it can be accessible from the web server created in Bastion/Web instance . I tested the object accessibility by using it URL <https://sparois3bucket.s3.amazonaws.com/Screenshot+2024-04-13+193333.png> to access it.

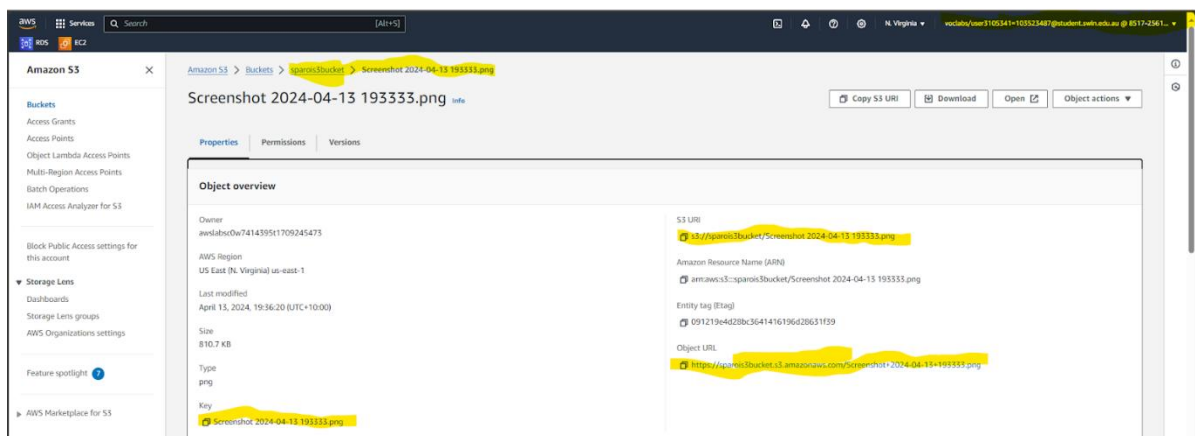


Figure 19 -Creating the Bucket Object (Photo Meta Data)

IX. CONFIGURING THE WEBSITE FUNCTIONALITY

- After setting up the S3 bucket , I set up the website functionality so that it can list all the photos that are stored in the S3 bucket . I downloaded the photoalbum_v3.0.zip and unzipped it . Then I put it on `/var/www/html/cos20017/` in Bastion/Web server instance that is connected via WinSCP.
- Then I go to the photoalbum and open the constan.php according to the assignment instruction .

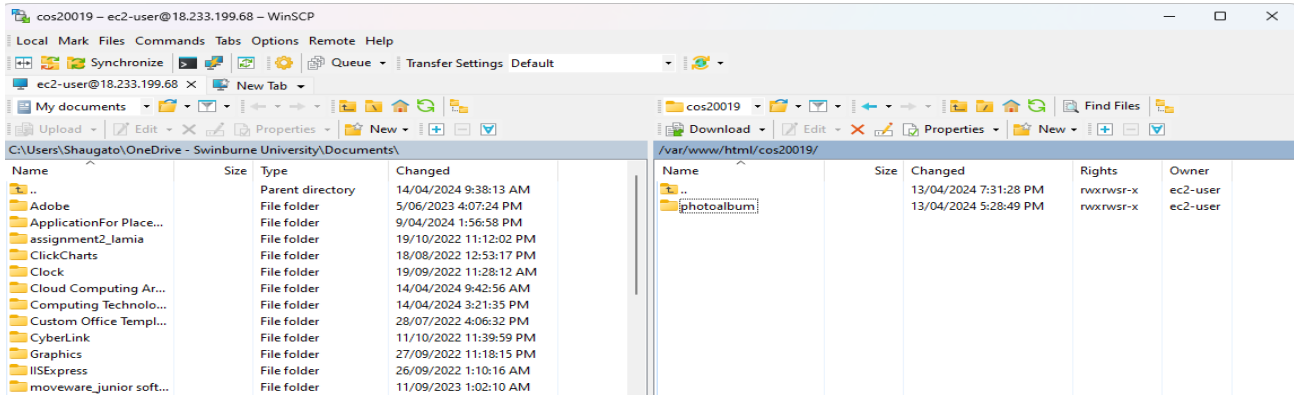


Figure 20 - Saving photoalbum on var/www/html/cos20019/ directory

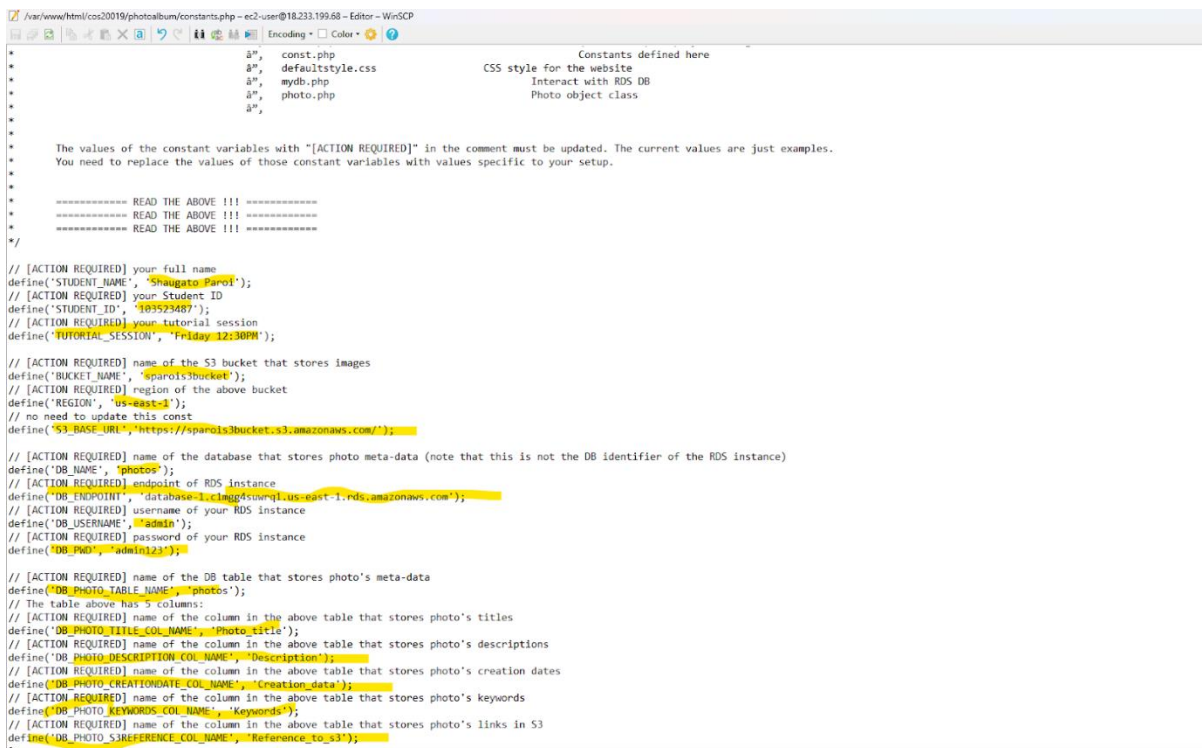


Figure 21 - Modifying constan.php accordingly

X. CREATING NETWORK ACL

Now I create a Network ACL called “PublicSubnet2NACL to limit the ICMP and other necessary traffic for accessing Public Subnet 2. I created it using the least privilege principle.

A. Inbound Rules:

1. Allowing Ping request (ICMP) from only 10.0.4.0/24
2. Allowing HTTP request from anywhere 0.0.0.0/0
3. Allowing SSH request from anywhere 0.0.0.0/24
4. Allowing MySQL/Aurora request from only 10.0.3.0/24
5. Allowing Custom TCP for port 1024-65535 by following the least principle rule

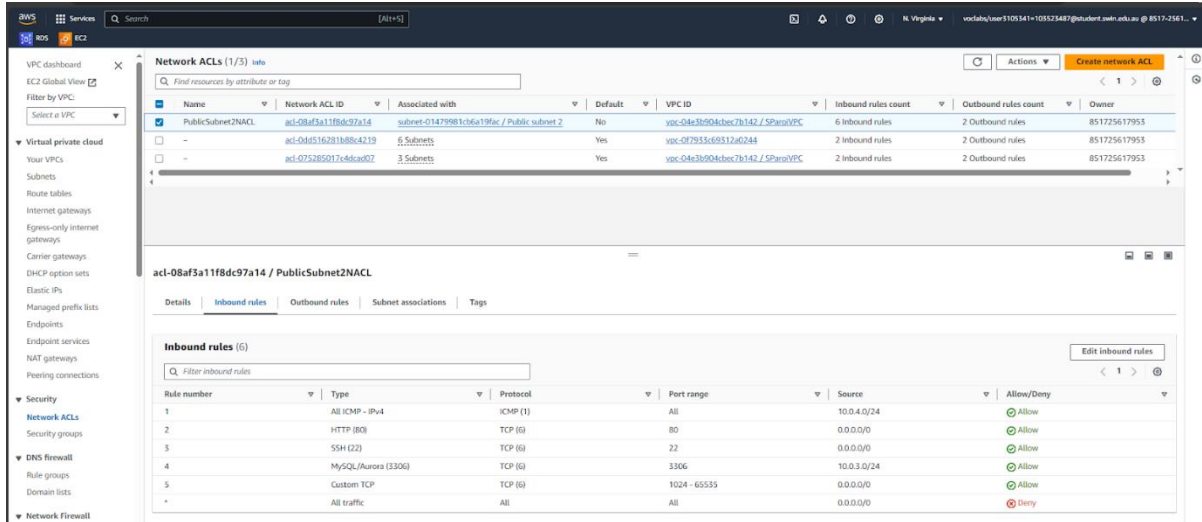


Figure 22 - Configuring Inbound rules

B. Outbound Rules:

1. Allowing all traffic 0.0.0.0/0

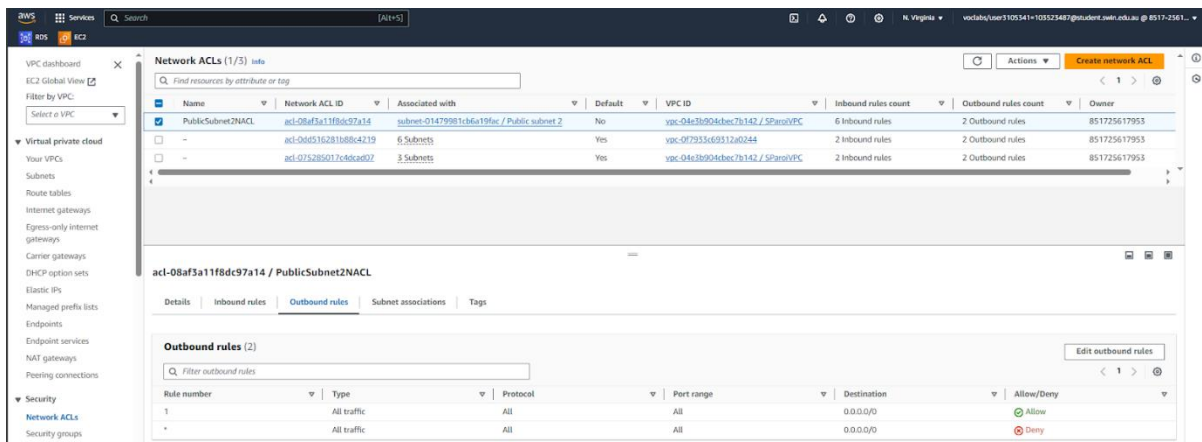


Figure 23- Configuring outbound rules -

XI. TESTING

I already added a object (photo meta data) on my7 S3 bucket . Now I populate the photo metadata on the ” photos” table on the “photos” database . Then I tested it goi to <http://ec2-18-233-199-68.compute-1.amazonaws.com/cos20019/photoalbum/album.php> this website

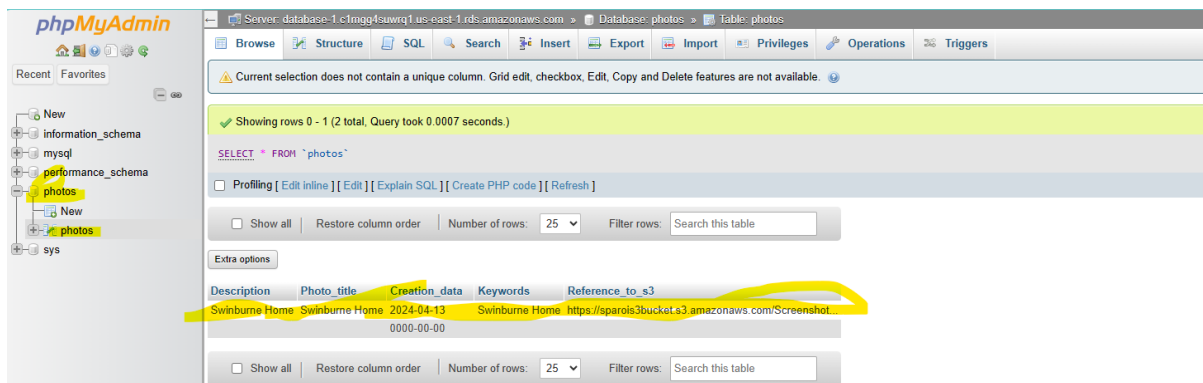


Figure 24 - Populating Photo Meta Data

Not secure | ec2-18-233-199-68.compute-1.amazonaws.com/cos20019/photoalbum/album.php

Student name: Shaugato Paroi

Student ID: 103523487

Tutorial session: Friday 12:30PM

Uploaded photos:



Photo	Name	Description	Creation date	Keywords
	Swinburne Home	Swinburne Home	2024-04-13	Swinburne Home
	Swinburne Logo	Logo Of Swinburne Uni	2024-04-15	Logo, University

Figure 25 - Successfully Able to view the website

REFERENCES

- [1] <https://aws.amazon.com/blogs/security/securely-connect-to-linux-instances-running-in-a-private-amazon-vpc/>
- [2] AWS, "Setting permissions for website access - Amazon Simple Storage Service," docs.aws.amazon.com, Sep. 15, 2023. <https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteAccessPermissionsReqd.html>