

USER MANUAL

9/13/2023

I. Introduction

This Project requires creating and deploying a photo album website on a simple AWS architecture.

II . Create a secure Virtual Private Cloud (VPC)

Create a secure Virtual Private Cloud (VPC) .The VPC is named 'AKaggdas' because we need to give it a name using the initial of our first name and Last name. The VPC has two availability zones each with a private and public subnet with suitable CIDR .

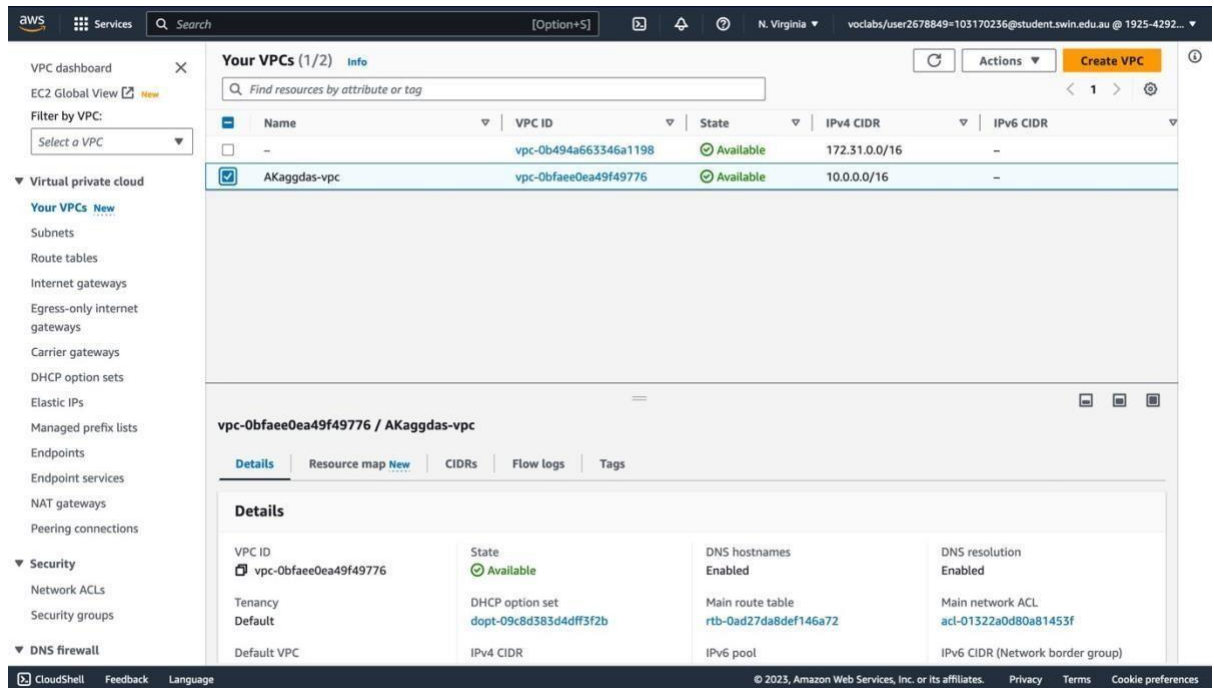


Figure 1 - Virtual Private Cloud

III. Route Tables

Public Route Table Configuration :

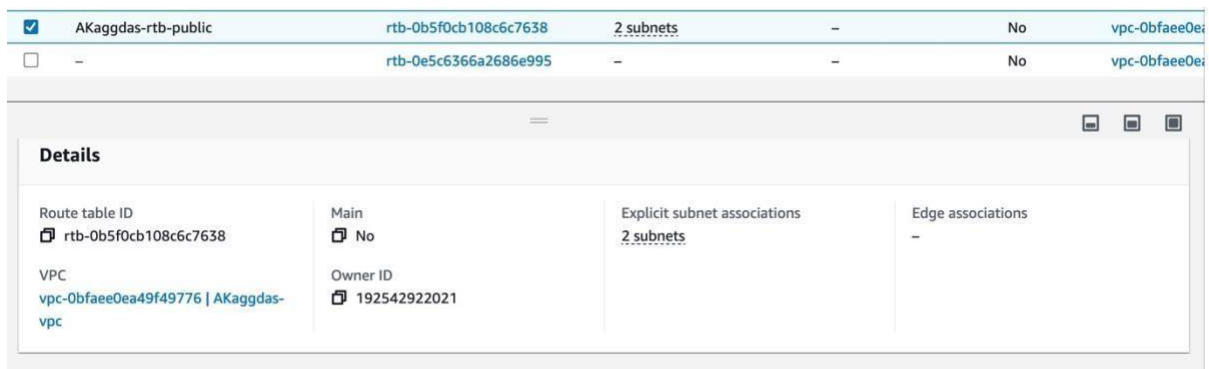


Figure 2 - 'AKaggdas' Public Route table

<input checked="" type="checkbox"/>	AKaggdas-rtb-public	rtb-0b5f0cb108c6c7638	2 subnets	-	No	vpc-0bfaee0
<input type="checkbox"/>	-	rtb-0e5c6366a2686e995	-	-	No	vpc-0bfaee0

Routes (2)

Both

<

1

>

⚙

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0608f3b4768674ad3	Active	No
10.0.0.0/16	local	Active	No

Figure 3 - 'AKaggdas' Public Routes

<input checked="" type="checkbox"/>	AKaggdas-rtb-public	rtb-0b5f0cb108c6c7638	2 subnets	-	No	vpc-0bfaee0
<input type="checkbox"/>	-	rtb-0e5c6366a2686e995	-	-	No	vpc-0bfaee0

Explicit subnet associations (2)

Edit subnet associations

<

1

>

⚙

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
AKaggdas-subnet-public1-us-east-1a	subnet-041b29723150d35c0	10.0.1.0/24	-
AKaggdas-subnet-public2-us-east-1b	subnet-00546cfae14df7214	10.0.2.0/24	-

Figure 4 - 'AKaggdas' Public Route table Subnet Associations

Private Route Table Configuration :

<input checked="" type="checkbox"/>	AKaggdas-rtb-private	rtb-0ad27da8def146a72	2 subnets	-	Yes	vpc-0bfaee0
<input type="checkbox"/>	AKaggdas-rtb-public	rtb-0b5f0cb108c6c7638	2 subnets	-	No	vpc-0bfaee0
<input type="checkbox"/>	-	rtb-0e5c6366a2686e995	-	-	No	vpc-0bfaee0

Details

Route table ID
rtb-0ad27da8def146a72

Main
Yes

Explicit subnet associations
2 subnets

Edge associations
-

VPC
vpc-0bfaee0ea49f49776 | AKaggdas-vpc

Owner ID
192542922021

Figure 5 - 'AKaggdas' Private Route table

<input checked="" type="checkbox"/>	AKaggdas-rtb-private	rtb-0ad27da8def146a72	2 subnets	-	Yes	vpc-0bfaee0e
<input type="checkbox"/>	AKaggdas-rtb-public	rtb-0b5f0cb108c6c7638	2 subnets	-	No	vpc-0bfaee0e
<input type="checkbox"/>	-	rtb-0e5c6366a2686e995	-	-	No	vpc-0bfaee0e

Routes (1)

Both
< 1 >

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

Figure 6 - 'AKaggdas' Private Routes

<input checked="" type="checkbox"/>	AKaggdas-rtb-private	rtb-0ad27da8def146a72	2 subnets	-	Yes	vpc-0bfaee0e
<input type="checkbox"/>	AKaggdas-rtb-public	rtb-0b5f0cb108c6c7638	2 subnets	-	No	vpc-0bfaee0e
<input type="checkbox"/>	-	rtb-0e5c6366a2686e995	-	-	No	vpc-0bfaee0e

Find subnet association

< 1 >

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
AKaggdas-subnet-private1-us-east-1a	subnet-0a6e4ab5914747030	10.0.3.0/24	-
AKaggdas-subnet-private2-us-east-1b	subnet-0e95d2e0477ec2118	10.0.4.0/24	-

Figure 7 - 'AKaggdas' Private Route table Subnet Associations

IV. Create Security groups

We create three security groups : 1] TestInstanceSG 2] WebServerSG 3] DBServerSG .We edit the **Inbound rules** for all the three and leave the outbound rules to default.

The screenshot shows the AWS Management Console interface for the 'TestInstanceSG' security group. The left sidebar contains navigation links for various AWS services. The main content area displays the details of the security group, including its name, ID, description, and VPC ID. The 'Inbound rules' tab is active, showing a table with one rule that allows all traffic from all IP addresses.

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-02d8959733a7e0...	IPv4	All traffic	All	All

Figure 8 - Test Instance Security Group

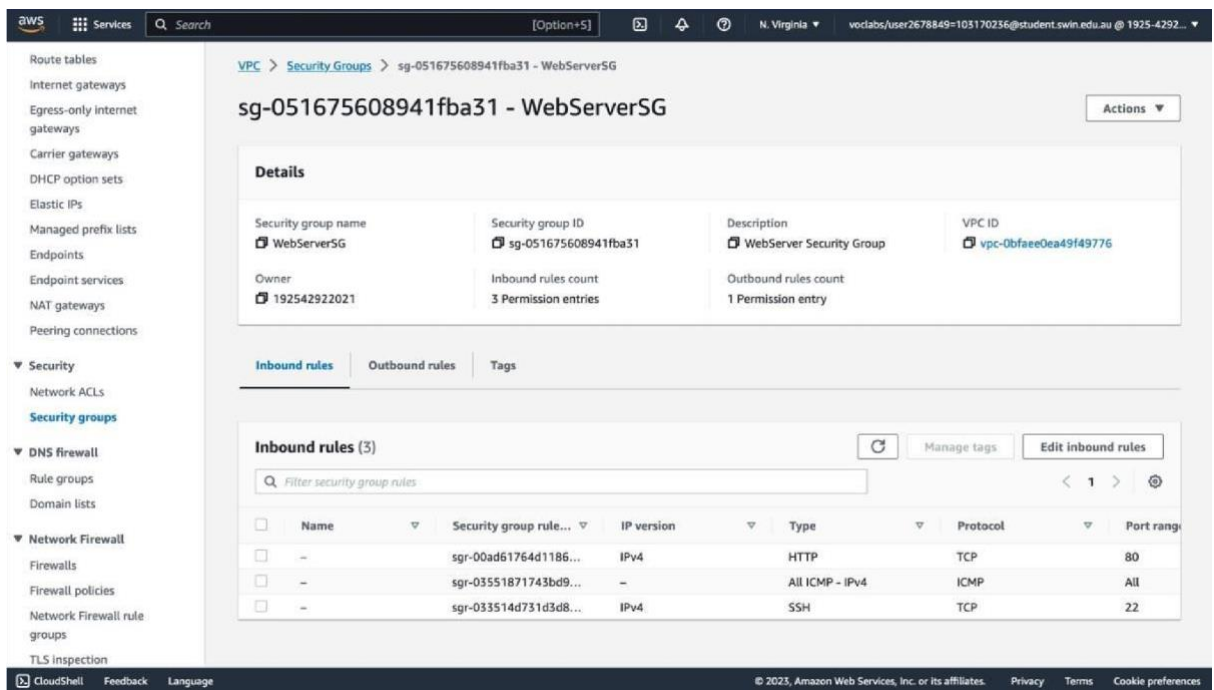


Figure 9 - Web Server Security Group

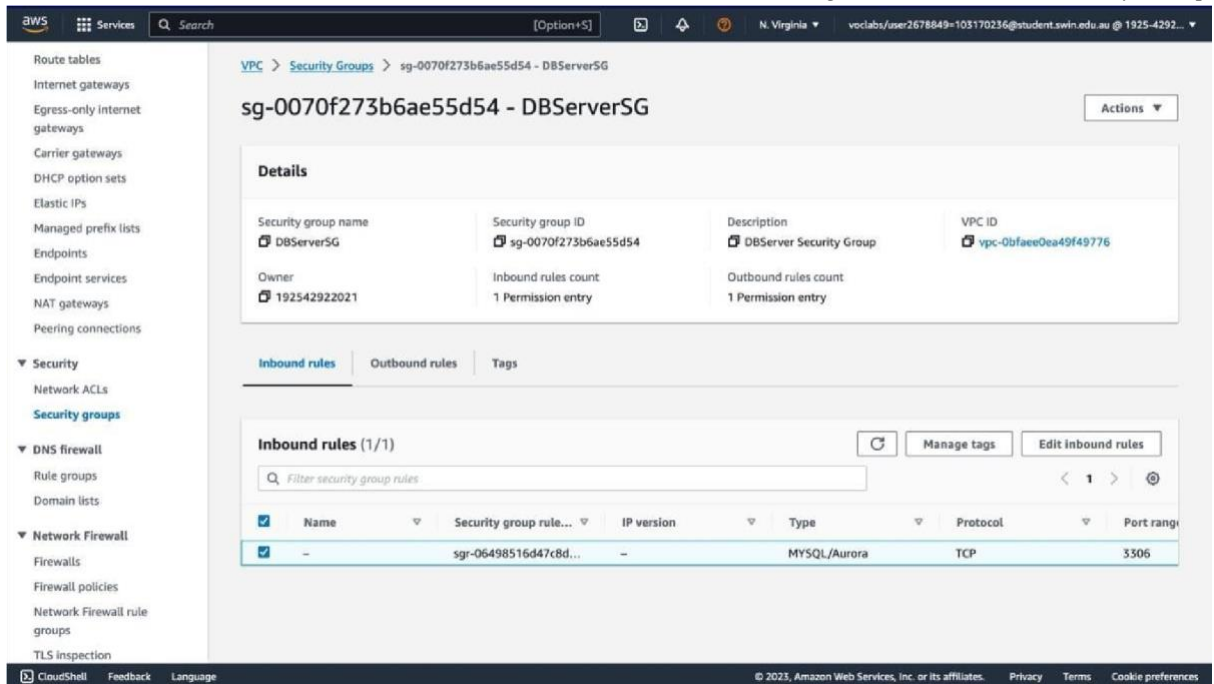


Figure 10 – Data base Server Security Group

V. Create Key Pair

We create a key pair to associate with the EC2 instances .

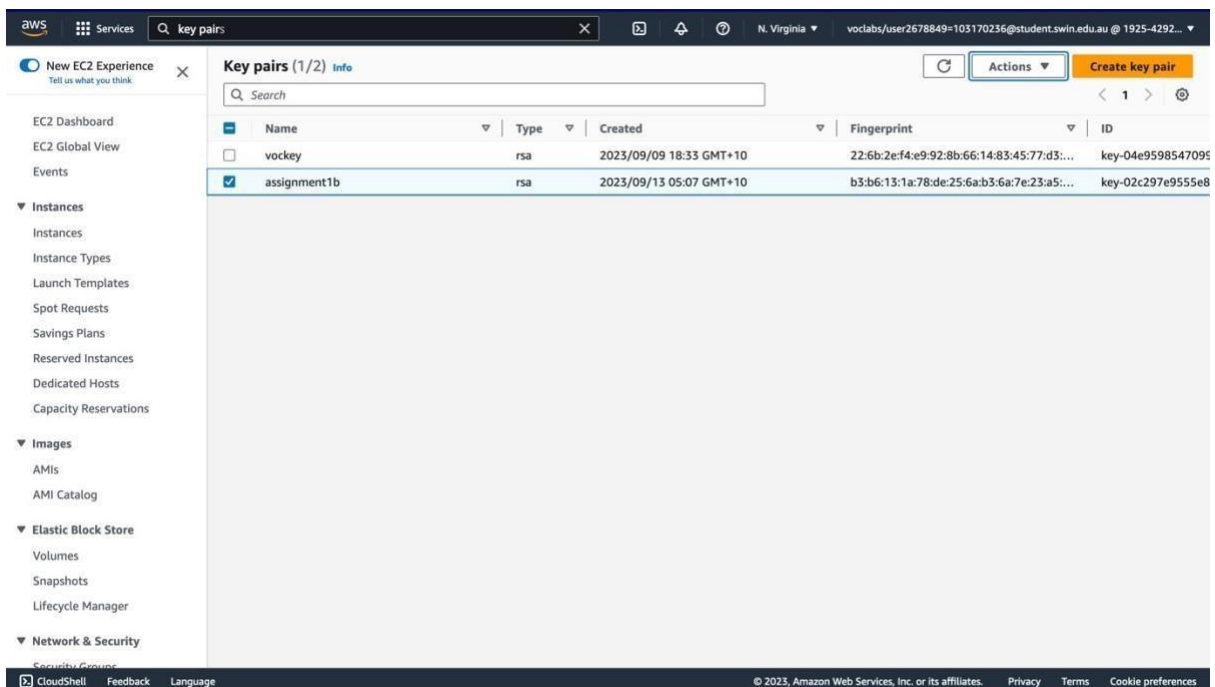


Figure 11 - Key Pair

VI. Create EC2 Instances

Web server Instance:

We create two EC2 instances, a 'test' instance and a 'web server' instance. We first create the Web Server EC2 instance. The instance type for Web Server Instance is *t2.micro* and Amazon Machine Image is *Amazon Linux 2 AMI (HVM), SSD Volume Type*. When we edit the network settings we select 'AKaggdas-vpc'. Now we expand Advanced settings copy the content provided to us in the 'Install_PHP_AWS.rtf' and paste it in the user data box. We launch the instance. We then add an Elastic IP Address to this instance by allocating an Elastic IP address in the same region. The web server instance is in the public subnet.

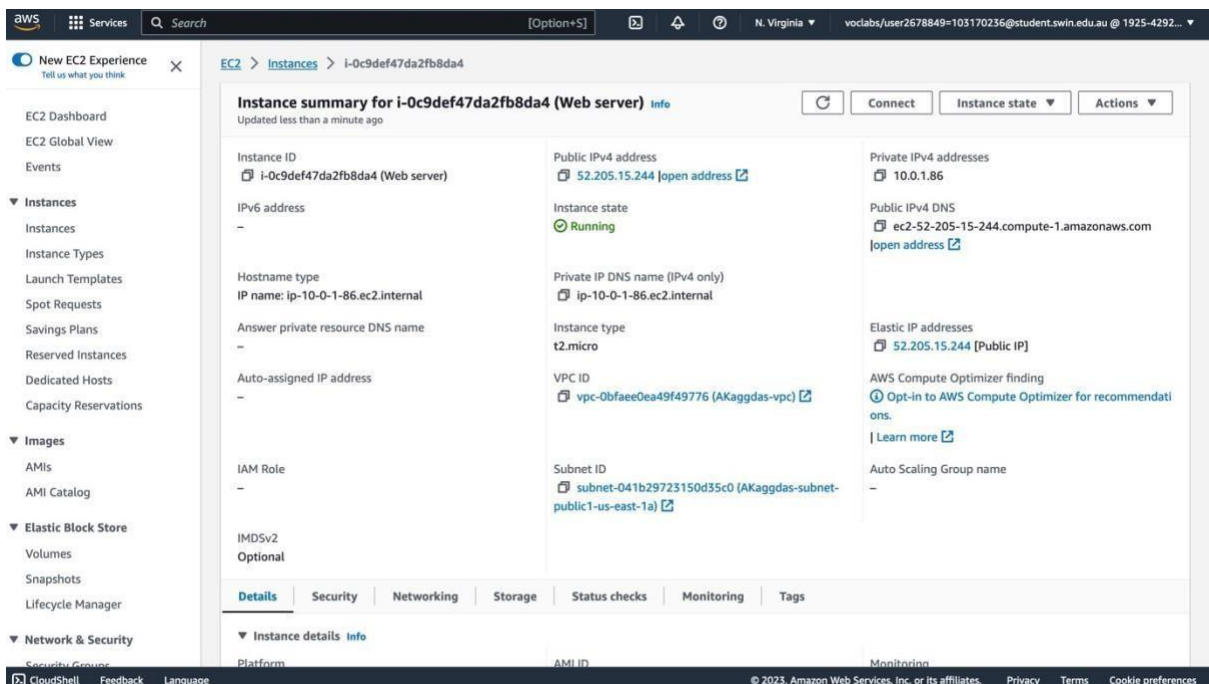


Figure 12 – Summary of Web Server Instance

Test instance :

The configuration for the test instance is the same as for the web server instance, except that the test instance does not have a public IP address and is located in a private subnet. This instance is used for demonstration purposes only. It does not contribute to the functionality of Photo Album website.

The screenshot displays the AWS Management Console interface for an EC2 instance. The left sidebar shows the navigation menu with categories like EC2 Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main content area shows the 'Instance summary for i-0144216cf75e822be (Test instance)'. The instance is in a 'Running' state. Key details include: Instance ID: i-0144216cf75e822be (Test instance), Public IPv4 address: -, Private IPv4 addresses: 10.0.4.194, Instance state: Running, Private IP DNS name (IPv4 only): ip-10-0-4-194.ec2.internal, Instance type: t2.micro, VPC ID: vpc-0bfaee0ea49f49776 (AKaggdas-vpc), Subnet ID: subnet-0e95d2e0477ec2118 (AKaggdas-subnet-private2-us-east-1b), IAM Role: -, IMDSv2: Optional. The console also shows tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. The bottom of the console shows the 'Instance details' section with Platform: Amazon Linux (Inferred) and AMI ID: ami-0f84a9675b22ea32.

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0144216cf75e822be (Test instance)	-	10.0.4.194

Instance state	Private IP DNS name (IPv4 only)	Public IPv4 DNS
Running	ip-10-0-4-194.ec2.internal	-

Instance type	Elastic IP addresses	AWS Compute Optimizer finding
t2.micro	-	Opt-in to AWS Compute Optimizer for recommendations. Learn more

VPC ID	Subnet ID	Auto Scaling Group name
vpc-0bfaee0ea49f49776 (AKaggdas-vpc)	subnet-0e95d2e0477ec2118 (AKaggdas-subnet-private2-us-east-1b)	-

IAM Role	Subnet ID	Monitoring
-	subnet-0e95d2e0477ec2118 (AKaggdas-subnet-private2-us-east-1b)	disabled

Platform	AMI ID	Monitoring
Amazon Linux (Inferred)	ami-0f84a9675b22ea32	disabled

Figure 13 – Summary of Test Instance

We successfully access the test instance in the private subnet by using the commands displayed in the figure below.

```
aishwaryakaggdas@Aishwaryas-MacBook-Air Desktop % ssh-agent bash

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
bash-3.2$ ssh-add -L
-L: No such file or directory
bash-3.2$ ssh-add -K assignment1b.pem
Identity added: assignment1b.pem (assignment1b.pem)
bash-3.2$ ssh-add -L
-L: No such file or directory
bash-3.2$ ssh-add -L assignment1b.pem
-L: No such file or directory
Identity added: assignment1b.pem (assignment1b.pem)
bash-3.2$ ssh -A ec2-user@52.205.15.244
Last login: Fri Sep 15 12:38:27 2023 from 106.216.202.4

    __|  __|  )
    _| (  _| /   Amazon Linux 2 AMI
   ___| \____|___|

https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 36 available
Run "sudo yum update" to apply all updates.
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-10-0-1-86 ~]$ ssh ec2-user@10.0.4.194

    __|  __|  )
    _| (  _| /   Amazon Linux 2 AMI
   ___| \____|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-4-194 ~]$
```

Figure 14 - SSH into an instance in a private subnet

We are able to establish a connection (ICMP ping) between this instance and the Bastion/Web server instance by pinging the public IP address of the web server.

```
https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 36 available
Run "sudo yum update" to apply all updates.
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-10-0-1-86 ~]$ ping 10.0.4.194
PING 10.0.4.194 (10.0.4.194) 56(84) bytes of data.
 64 bytes from 10.0.4.194: icmp_seq=1 ttl=255 time=0.799 ms
 64 bytes from 10.0.4.194: icmp_seq=2 ttl=255 time=0.713 ms
 64 bytes from 10.0.4.194: icmp_seq=3 ttl=255 time=0.770 ms
 64 bytes from 10.0.4.194: icmp_seq=4 ttl=255 time=0.752 ms
 64 bytes from 10.0.4.194: icmp_seq=5 ttl=255 time=0.790 ms
 64 bytes from 10.0.4.194: icmp_seq=6 ttl=255 time=0.742 ms
 64 bytes from 10.0.4.194: icmp_seq=7 ttl=255 time=0.709 ms
 64 bytes from 10.0.4.194: icmp_seq=8 ttl=255 time=0.807 ms
 64 bytes from 10.0.4.194: icmp_seq=9 ttl=255 time=0.706 ms
 64 bytes from 10.0.4.194: icmp_seq=10 ttl=255 time=0.717 ms
 64 bytes from 10.0.4.194: icmp_seq=11 ttl=255 time=0.773 ms
 64 bytes from 10.0.4.194: icmp_seq=12 ttl=255 time=0.777 ms
```

Figure 15 – Ping the web server

VII. Create an Amazon RDS DB Instance

This RDS instance has the following configs:

- DB engine version: *MySQL 8.0.34*
- Template: *Free tier*
- Public access: *No*


```

https://aws.amazon.com/amazon-linux-2/
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-10-0-1-86 ~]$ cd /var/www/html
[ec2-user@ip-10-0-1-86 html]$ wget https://files.phpmyadmin.net/phpMyAdmin/5.2.1/phpMyAdmin-5.2.1-english.zip
--2023-09-12 19:17:12-- https://files.phpmyadmin.net/phpMyAdmin/5.2.1/phpMyAdmin-5.2.1-english.zip
Resolving files.phpmyadmin.net (files.phpmyadmin.net)... 89.187.177.16, 156.144.36.23, 2a02:0ea8:c400::11, ...
Connecting to files.phpmyadmin.net (files.phpmyadmin.net)[89.187.177.16]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18448276 (18.0M) [application/zip]
Saving to: 'phpMyAdmin-5.2.1-english.zip'

100%[=====] 18,448,276 --.-K/s in 0.1s

2023-09-12 19:17:12 (95.1 MB/s) - 'phpMyAdmin-5.2.1-english.zip' saved [18448276/18448276]

[ec2-user@ip-10-0-1-86 html]$ unzip phpMyAdmin-5.2.1-english.zip
Archive:  phpMyAdmin-5.2.1-english.zip
  creating: phpMyAdmin-5.2.1-english/
  extracting: phpMyAdmin-5.2.1-english/.rsrcsrc.json
  inflating: phpMyAdmin-5.2.1-english/CONTRIBUTING.md
  inflating: phpMyAdmin-5.2.1-english/ChangeLog
  inflating: phpMyAdmin-5.2.1-english/LICENSE
  inflating: phpMyAdmin-5.2.1-english/README
  extracting: phpMyAdmin-5.2.1-english/RELEASE-DATE-5.2.1
  extracting: phpMyAdmin-5.2.1-english/babel.config.json
  inflating: phpMyAdmin-5.2.1-english/composer.json
  inflating: phpMyAdmin-5.2.1-english/composer.lock
  inflating: phpMyAdmin-5.2.1-english/config.sample.inc.php
  creating: phpMyAdmin-5.2.1-english/doc/
  creating: phpMyAdmin-5.2.1-english/doc/html/
  inflating: phpMyAdmin-5.2.1-english/doc/html/_images/chart.png
  inflating: phpMyAdmin-5.2.1-english/doc/html/_images/column_chart.png
  inflating: phpMyAdmin-5.2.1-english/doc/html/_images/line_chart.png
  inflating: phpMyAdmin-5.2.1-english/doc/html/_images/pie_chart.png
  extracting: phpMyAdmin-5.2.1-english/doc/html/_images/pma-relations-links.png
  inflating: phpMyAdmin-5.2.1-english/doc/html/_images/pma-relations-relation-link.png
  inflating: phpMyAdmin-5.2.1-english/doc/html/_images/pma-relations-relation-name.png
  inflating: phpMyAdmin-5.2.1-english/doc/html/_images/pma-relations-relation-view-link.png
  extracting: phpMyAdmin-5.2.1-english/doc/html/_images/query_result_operations.png
  inflating: phpMyAdmin-5.2.1-english/doc/html/_images/scatter_chart.png
  inflating: phpMyAdmin-5.2.1-english/doc/html/_images/spline_chart.png
  inflating: phpMyAdmin-5.2.1-english/doc/html/_images/timeline_chart.png
  inflating: phpMyAdmin-5.2.1-english/doc/html/_images/usergroups.png
  creating: phpMyAdmin-5.2.1-english/doc/html/_sources/
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/bookmarks.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/charts.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/config.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/copyright.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/credits.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/developers.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/faq.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/glossary.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/import_export.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/index.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/intro.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/other.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/privileges.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/references.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/require.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/security.rst.txt
  inflating: phpMyAdmin-5.2.1-english/doc/html/_sources/settings.rst.txt

```

Figure 18 - unzip phpMyAdmin-5.2.1-english.zip

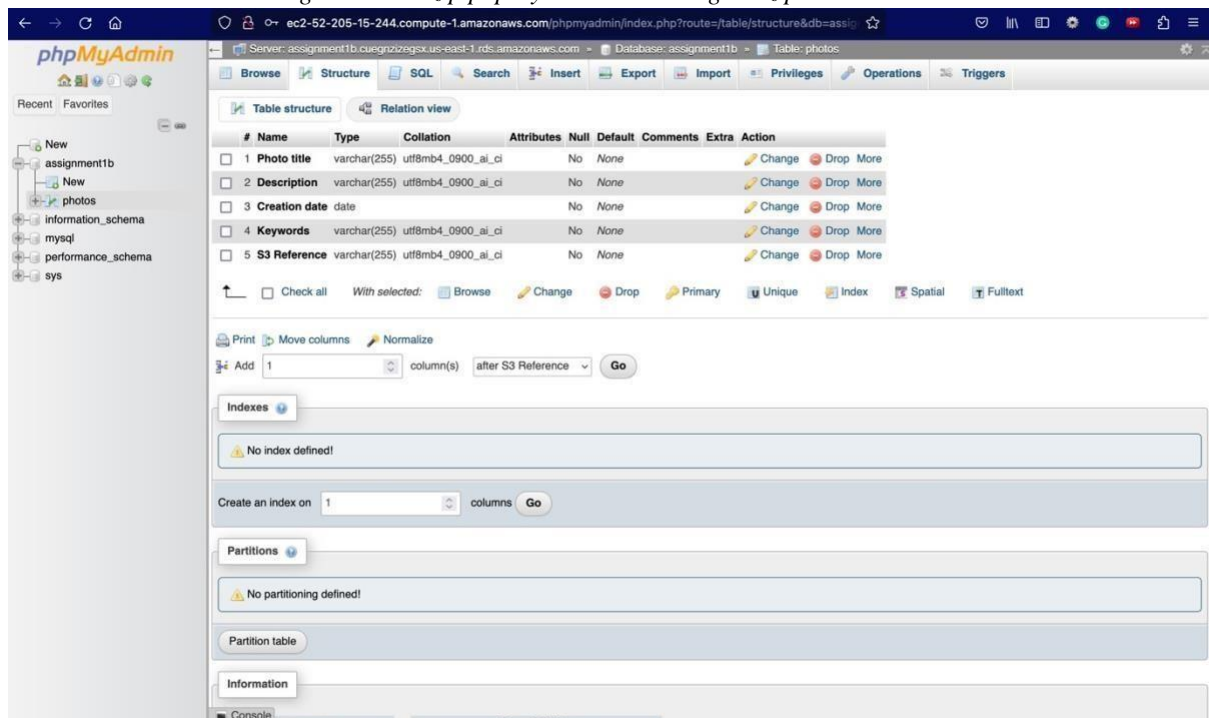


Figure 19- Access phpMyAdmin from your local machine and create Appropriate table on your RDS Database:

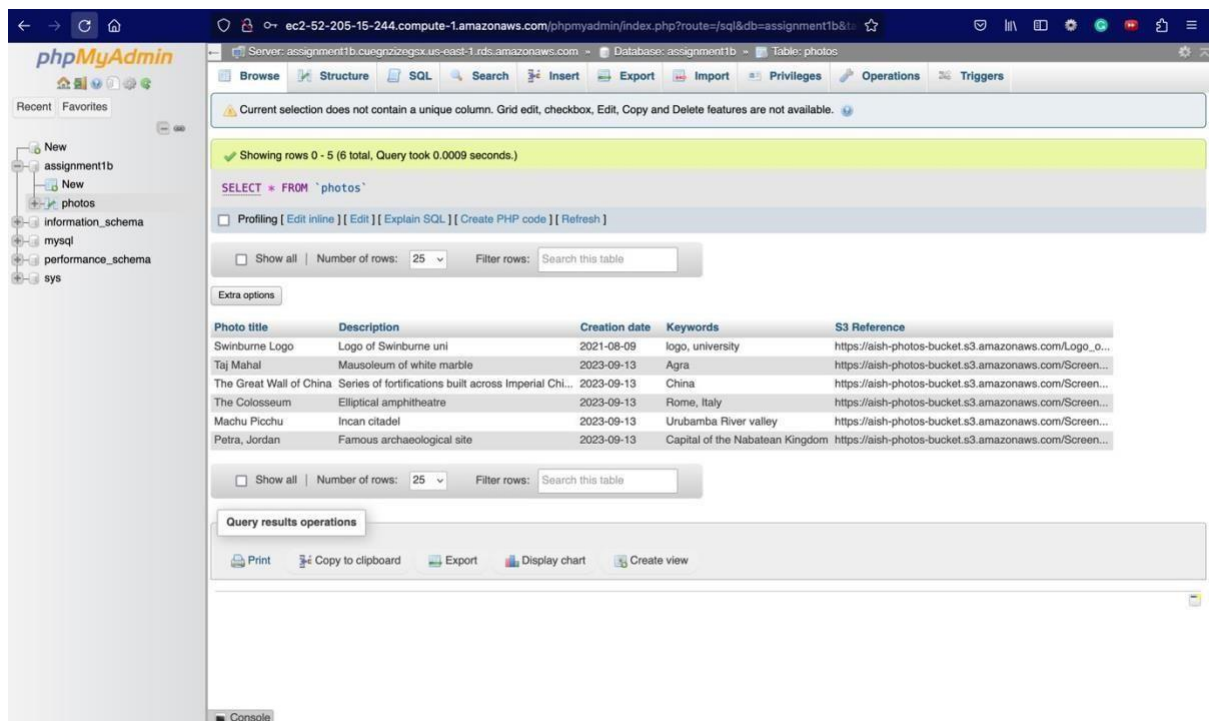


Figure 20- Populate 'photos' table with a few records

IX. Reconfigure phpMyAdmin

We Open Filezilla and navigate to phpmyadmin directory (var/www/html/phpmyadmin). We change the name of config.sample.inc.php file to config.inc.php. We open config.inc.php file and look for this line:

```
$cfg['Servers'][$i]['host'] = 'localhost';
```

Replace 'localhost' with the endpoint of your RDS instance.

```
$cfg['Servers'][$i]['host'] = 'assignment1b.cuegnzizexs.us-east-1.rds.amazonaws.com';
```

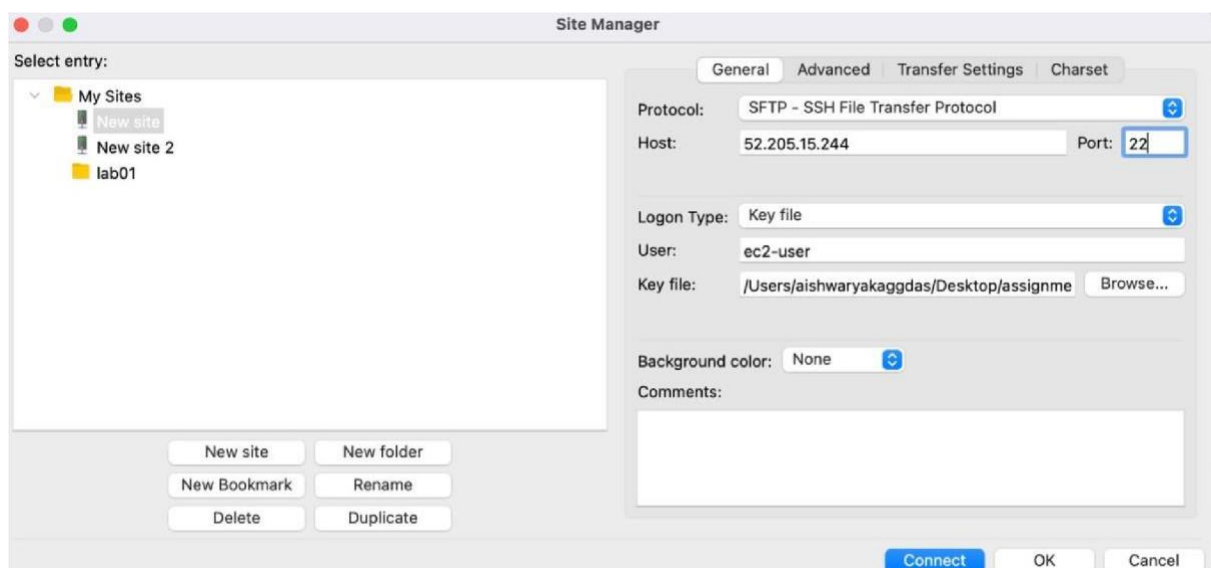


Figure 21 – Connect to FileZilla

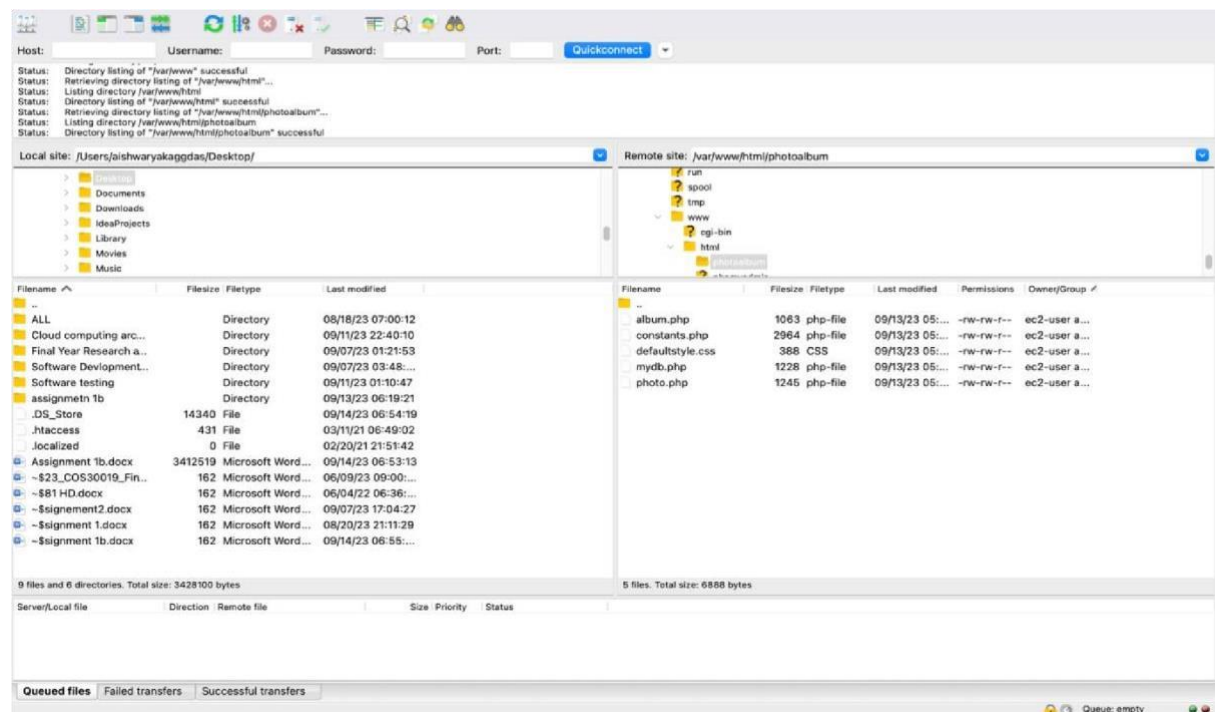


Figure 22- Navigate to phpmyadmin directory

X . Functional requirements of Photo Album website

The Photo Album website has the following functional requirements.

2.1 – Photo storage

We create an S3 bucket and name it ‘aish-photos-bucket’ to store our photos. We manually upload some photos onto S3 bucket and ensure they have been successfully uploaded.

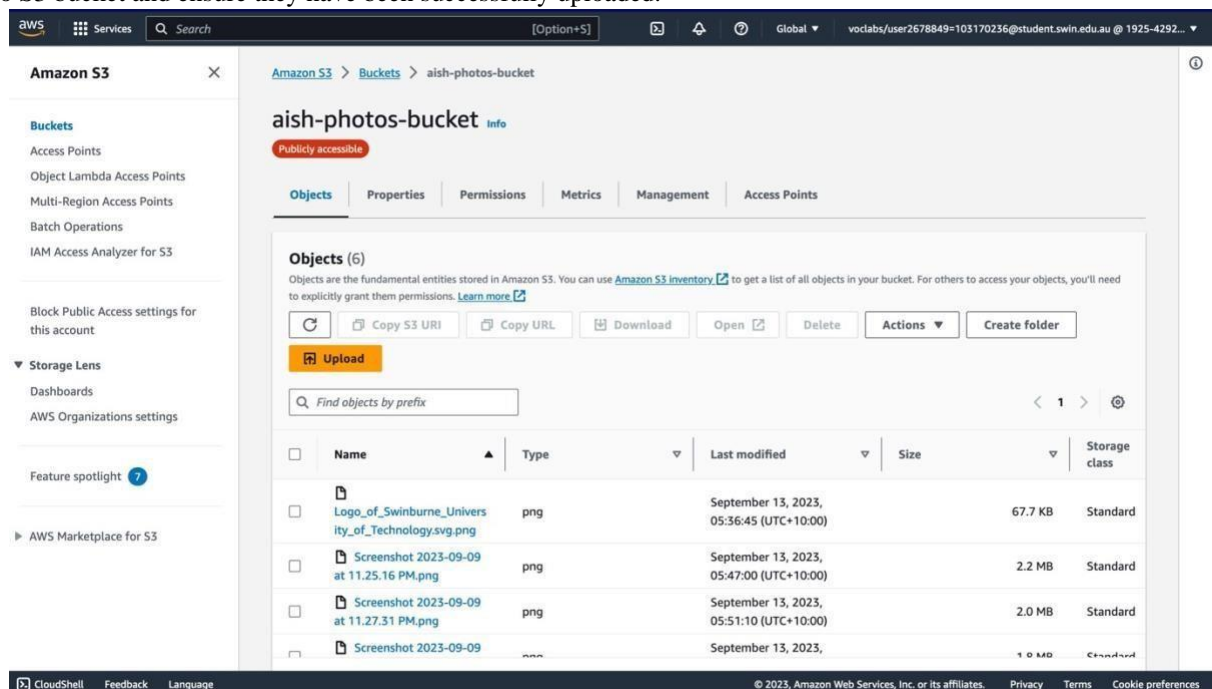


Figure 23 – aish-photos bucket

2.2 – Photo meta-data in RDS Database

We populate the table with a few records the first being

- Photo title: *Swinburne Logo*
- Description: *Logo of Swinburne uni*
- Creation date: *2021-08-09* □ Keywords: *logo, university*
- Object URL in S3:
https://aishphotosbucket.s3.amazonaws.com/Logo_of_Swinburne_University_of_Technology.svg.png

Photo	Name	Description	Creation date	Keywords
	Swinburne Logo	Logo of Swinburne uni	2021-08-09	logo, university

Figure 24 – First record of the table

2.3 – Photo Album website functionality

The website is able to list all the photos (stored in the S3 bucket) along with their meta-data (stored in the database).

Photo	Name	Description	Creation date	Keywords
	Swinburne Logo	Logo of Swinburne uni	2021-08-09	logo, university
	Taj Mahal	Mausoleum of white marble	2023-09-13	Agra
	The Great Wall of China	Series of fortifications built across Imperial China	2023-09-13	China
	The Colosseum	Elliptical amphitheatre	2023-09-13	Rome, Italy
	Machu Picchu	Incan citadel	2023-09-13	Urubamba River valley
	Petra, Jordan	Famous archaeological site	2023-09-13	Capital of the Nabatean Kingdom

Figure 25 – album.php

URL of the album.php :

<http://ec2-52-205-15-244.compute-1.amazonaws.com/photoalbum/album.php>

XI. Problems And Achievements

Problems:

The objects in the S3 bucket were not publicly accessible, which was one of the challenges I faced. To solve the problem, I researched and discovered that we require to attach a public bucket policy to make the objects accessible to everyone. The link that I used to resolve this issue is provided below.

URL : <https://saturncloud.io/blog/how-to-access-images-from-amazon-s3-by-url/>

The web server needed an additional layer of protection, which was my second problem. I was unable to design and deploy a network ACL .I attempted but failed to complete this task. It was the most difficult part of the project.

Achievements :

- Create VPC with 2 public and 2 private subnets
- Correct Public and Private Routing tables with correct subnet associations □ Security groups properly configured and attached.
- Correct Web server and Test instances running in correct subnets
- Database schema as specified
- Database running in correct subnets
- S3 objects publicly accessible, using proper access policy

Functional Requirements:

- album.php page displayed from EC2 Web server
- Provided URL is persistent (Elastic IP Association)
- Photos loaded from S3 with matching metadata from RDS
- Web server instance reachable from Test instance via ICMP