



Assignment 1b Creating and deploying Photo Album website onto a simple AWS infrastructure.

COS 20019- Cloud Computing Architecture

Vu Duc Tran

15/04/2024

Tutorial time: Tuesday 10:30 – 12:30

Overview:

This assignment encompasses the creation of a robust Virtual Private Cloud (VPC), incorporating essential elements such as subnets, routing tables, and security groups to ensure a secure environment. Furthermore, we establish stringent control over access to and from the VPC via an internet gateway. Our next endeavor involves the customization of provided PHP code to develop a sophisticated website capable of storing metadata of photos uploaded to Amazon S3 into a MySQL database managed by Amazon RDS. Following this, we embark on the deployment and meticulous testing of this website on an Apache web server operating within an Amazon Elastic Compute Cloud (EC2) virtual machine instance. Concluding our efforts, we bolster the security measures by implementing a network Access Control List (ACL) on the public subnet housing our server. This report offers a detailed walkthrough of the entire process, providing coherent explanations of each step undertaken.

1. Infrastructure deployment

1.1 VPC

For the initial step, I was tasked with creating a VPC using my initials and setting it up in the (us-east-1 region). The VPC serves as a container housing both subnets and routing tables, connecting to the main page.

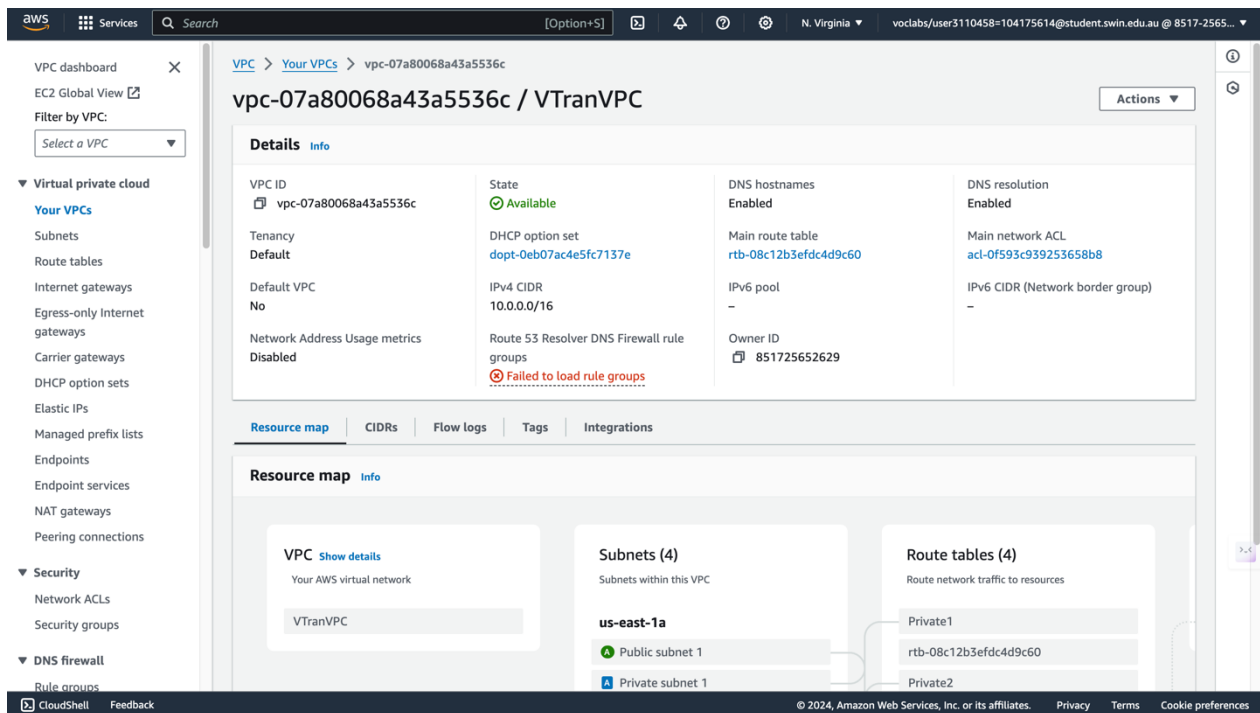


Figure 1

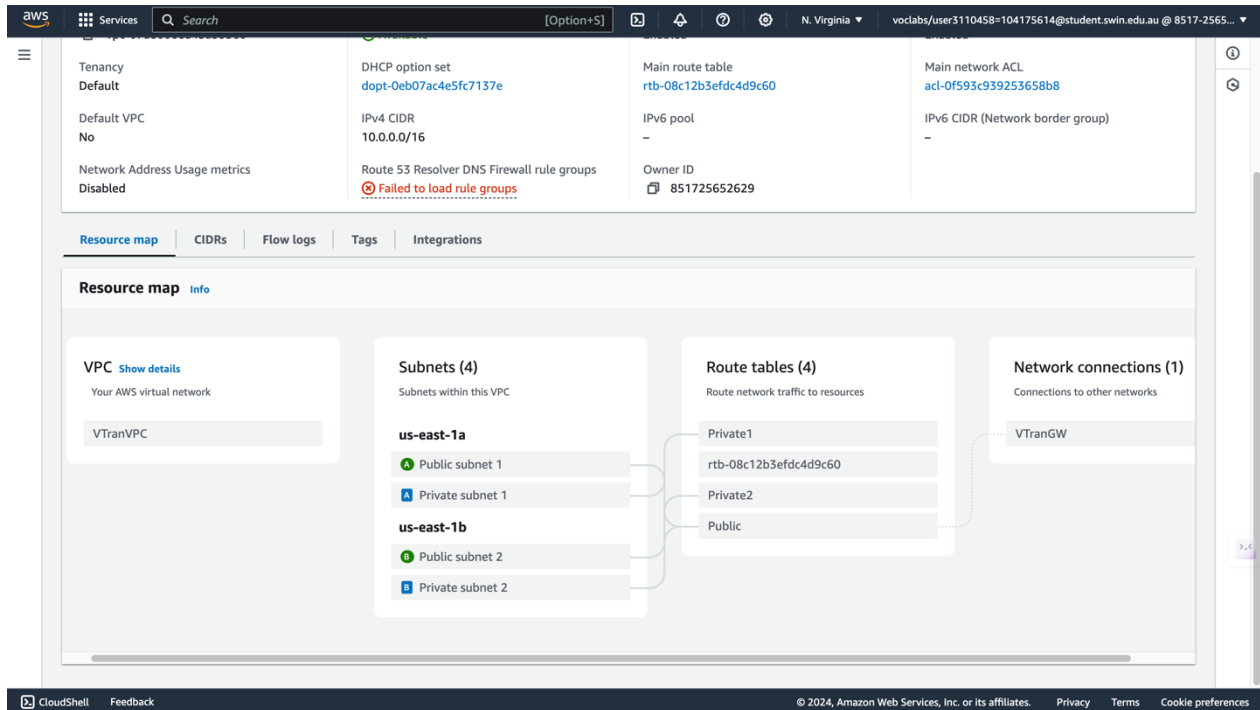


Figure 2

As depicted in figure 2, there are a total of four subnets: Public and Private Subnet 1, situated in the us-east 1a server, and Public and Private Subnet 2, respectively established in the us-east 1b server. The private subnets within this VPC are designated for testing purposes and are exclusively accessible via Public Subnet 2, which is linked with Public Subnet 1 in the public route table, constituting the primary connection to the internet.

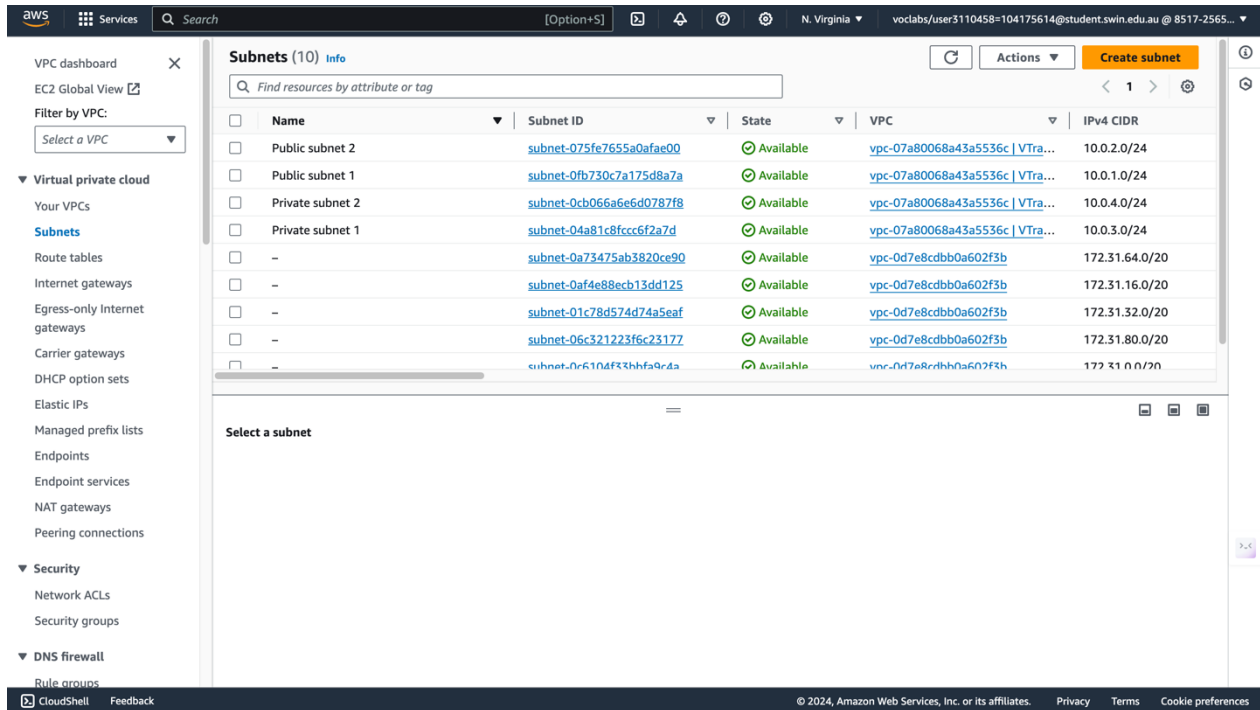


Figure 3

Additionally, as illustrated in figure 3, each subnet has been assigned a specific CIDR as outlined in the rubric.

1.2 Create Security groups

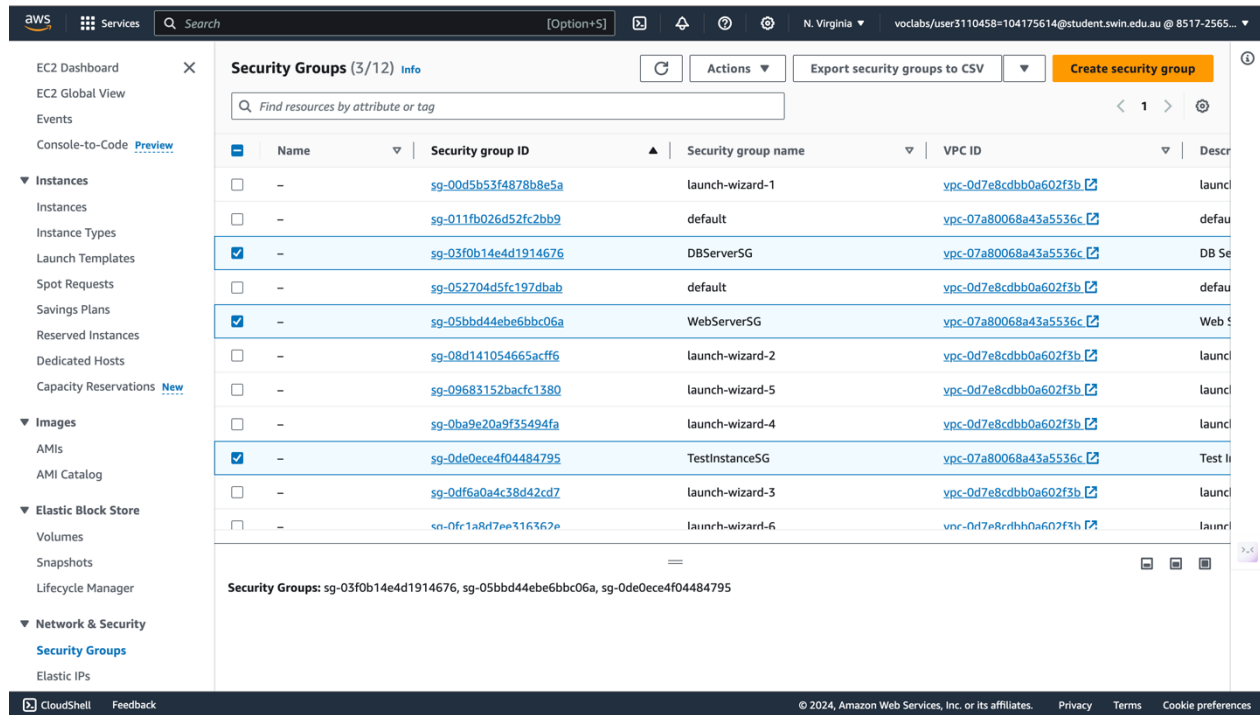


Figure 4

As depicted in figure 4 above, there are three security groups, each associated with different subnets serving distinct purposes. The WebServerSg security group is assigned to the Web Server instance, permitting only SSH and HTTP traffic, along with ICMP IPv4 traffic as displayed in figure 5 below. Similarly, the TestInstanceSG allows all traffic and is designated for backend testing on the Test Instance. The ICMP security configuration, illustrated in figure 4, permits ICMP traffic solely from the TestServerSG, establishing a connection between the Web server instance and the Test server instance. Lastly, the DBServerSG exclusively allows MySQL traffic from WebServerSg, ensuring that access to the database is restricted to the Web server.

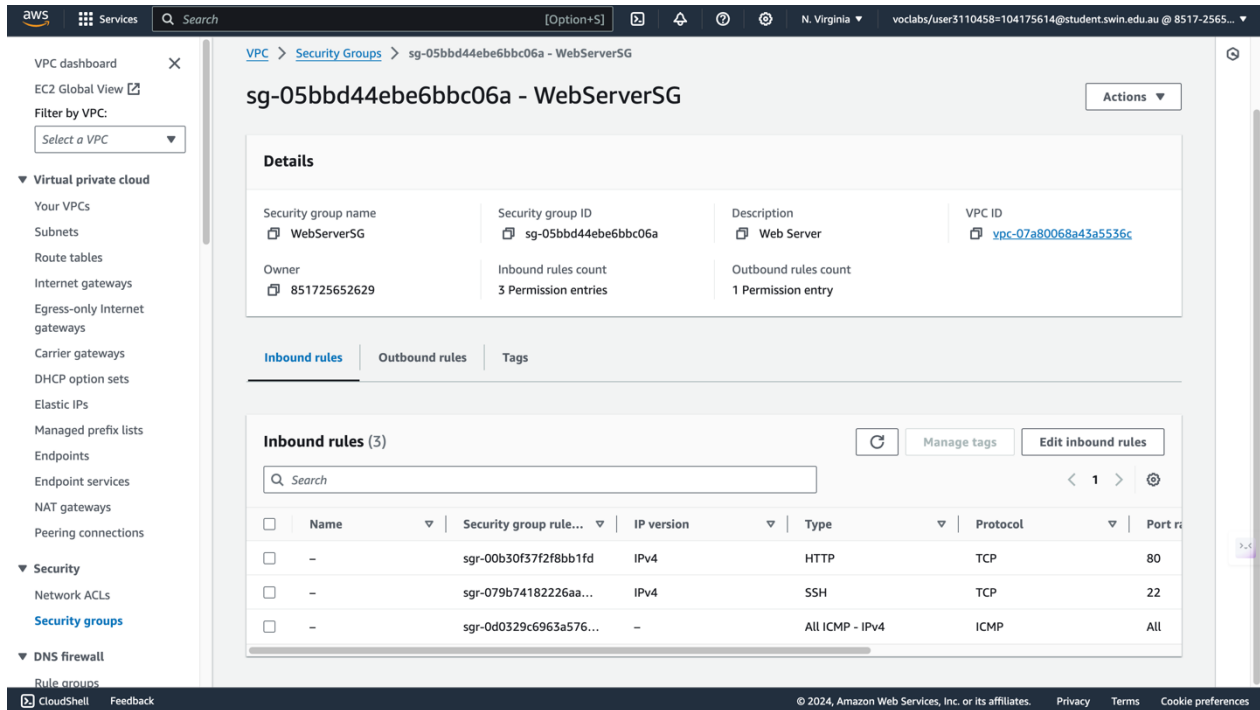


Figure 5

1.3 EC2 virtual instance

1.3.1 Bastion/Web server instance

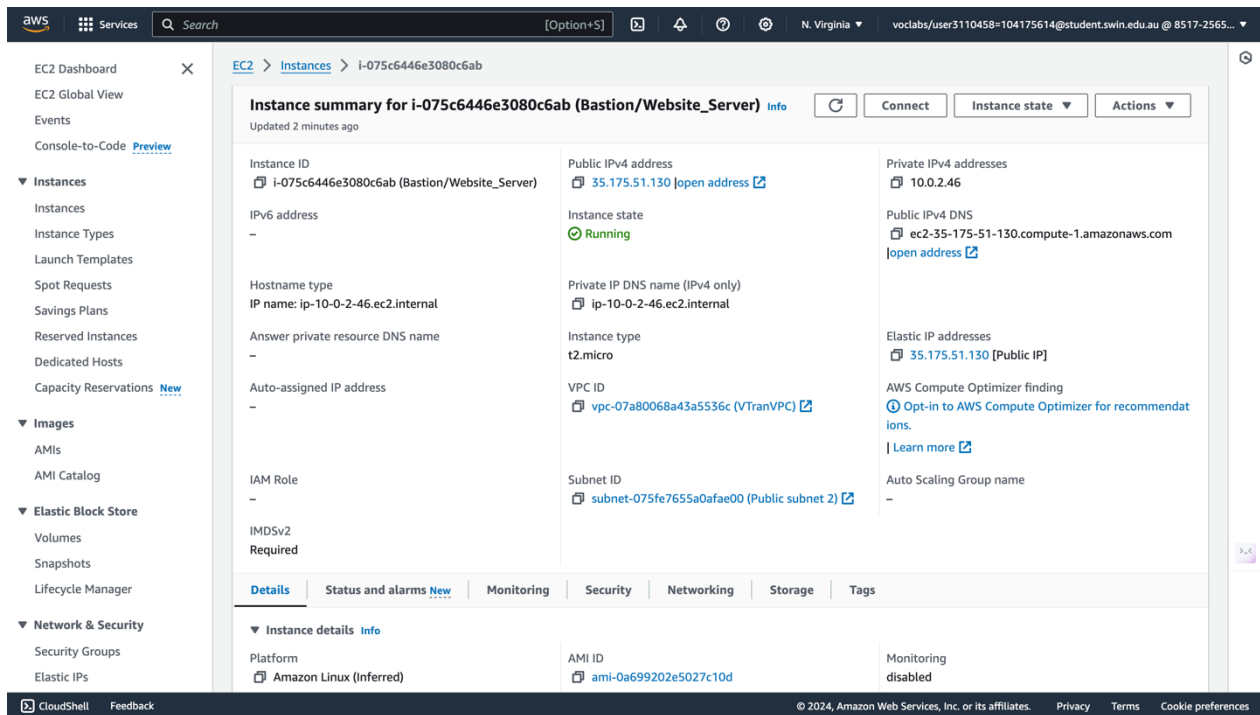


Figure 6

This instance will serve as the hosting platform for the "Photo Album" web application and will also function as a bastion host for SSH access to the Test instance located in a private subnet. Additionally, to address the issue of AWS's dynamic allocation of public IP addresses with each new session, I have assigned an elastic public IP address to ensure consistency in IP address access.

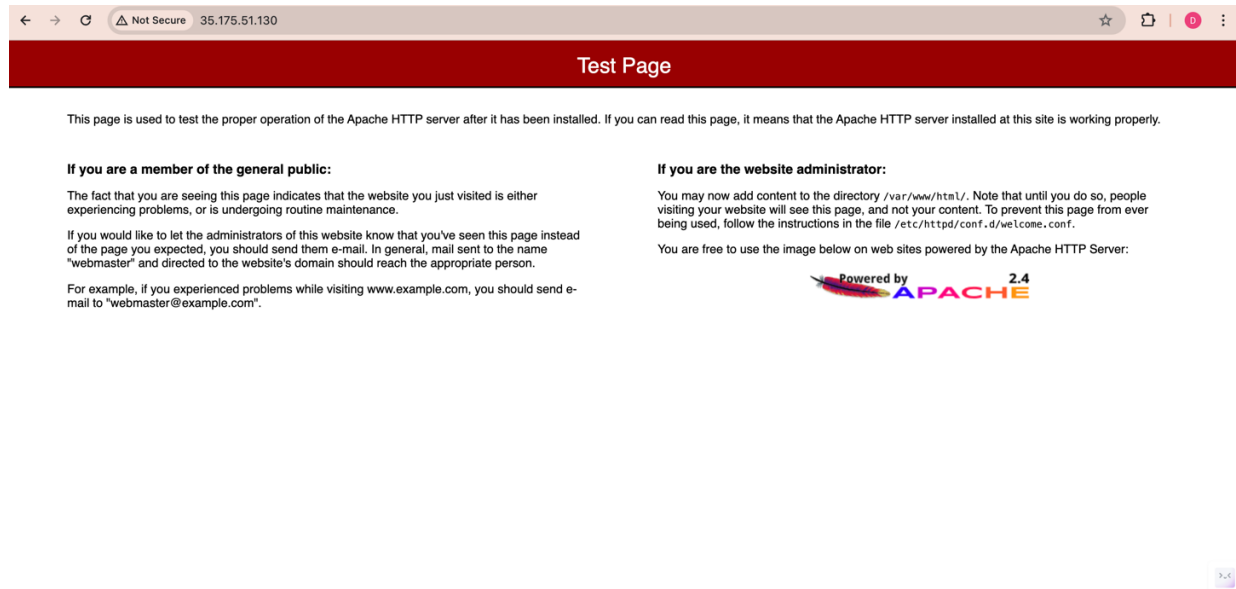
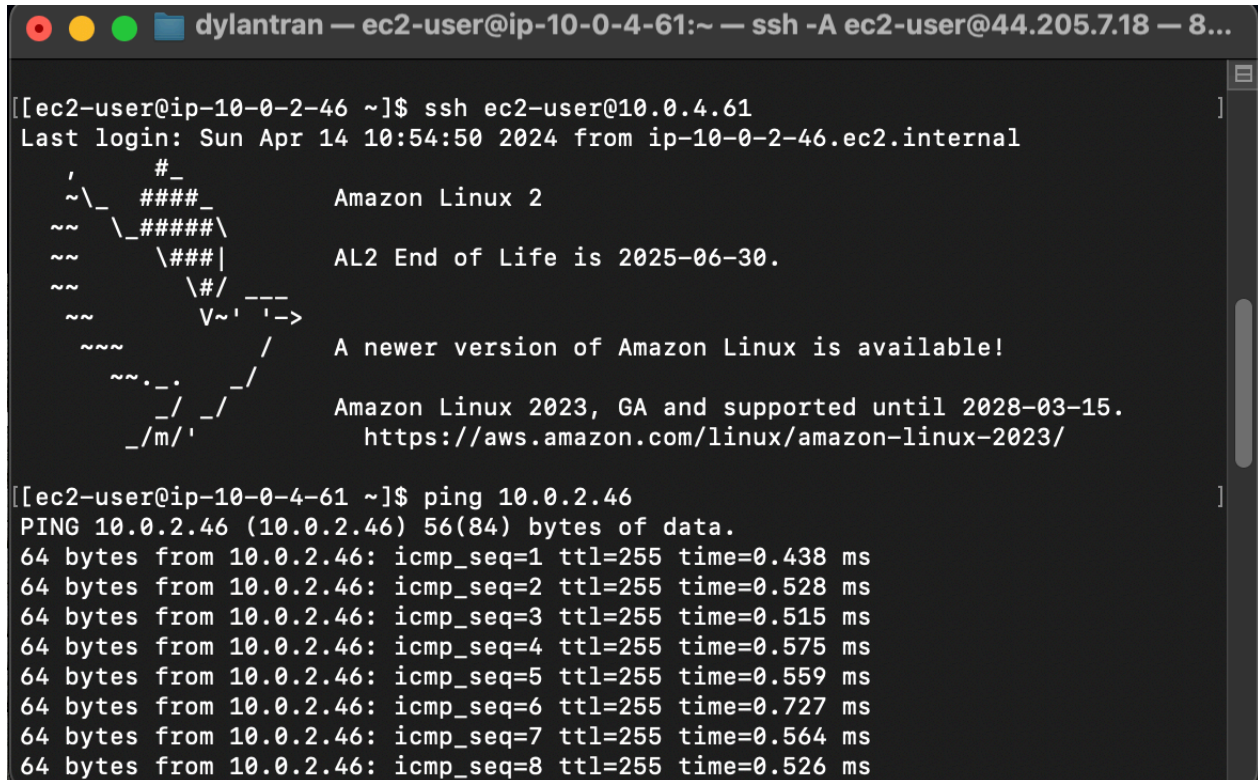


Figure 7

1.3.2 This step is to access SSH of Test instance



```

dylantran — ec2-user@ip-10-0-4-61:~ — ssh -A ec2-user@44.205.7.18 — 8...

[[ec2-user@ip-10-0-2-46 ~]$ ssh ec2-user@10.0.4.61
Last login: Sun Apr 14 10:54:50 2024 from ip-10-0-2-46.ec2.internal

  _
 _\#####_      Amazon Linux 2
~~\#####\
~~\###|      AL2 End of Life is 2025-06-30.
~~\#/
~~V~'--->
~~~
~~~.~.~
~~/_/_/_/
~/m/'

A newer version of Amazon Linux is available!

Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/

[[ec2-user@ip-10-0-4-61 ~]$ ping 10.0.2.46
PING 10.0.2.46 (10.0.2.46) 56(84) bytes of data.
64 bytes from 10.0.2.46: icmp_seq=1 ttl=255 time=0.438 ms
64 bytes from 10.0.2.46: icmp_seq=2 ttl=255 time=0.528 ms
64 bytes from 10.0.2.46: icmp_seq=3 ttl=255 time=0.515 ms
64 bytes from 10.0.2.46: icmp_seq=4 ttl=255 time=0.575 ms
64 bytes from 10.0.2.46: icmp_seq=5 ttl=255 time=0.559 ms
64 bytes from 10.0.2.46: icmp_seq=6 ttl=255 time=0.727 ms
64 bytes from 10.0.2.46: icmp_seq=7 ttl=255 time=0.564 ms
64 bytes from 10.0.2.46: icmp_seq=8 ttl=255 time=0.526 ms

```

Figure 8

Illustrated in figure 8, it depicts the test instance server initiating a ping request to the private IP address of the Bastion/WebServer. Accessing the test server necessitated SSH access into the Bastion/WebServer, followed by another SSH connection into the test server, and subsequently initiating a ping to the private IP address of the WebServer utilizing ICMP.

1.3 Create RDS database instance

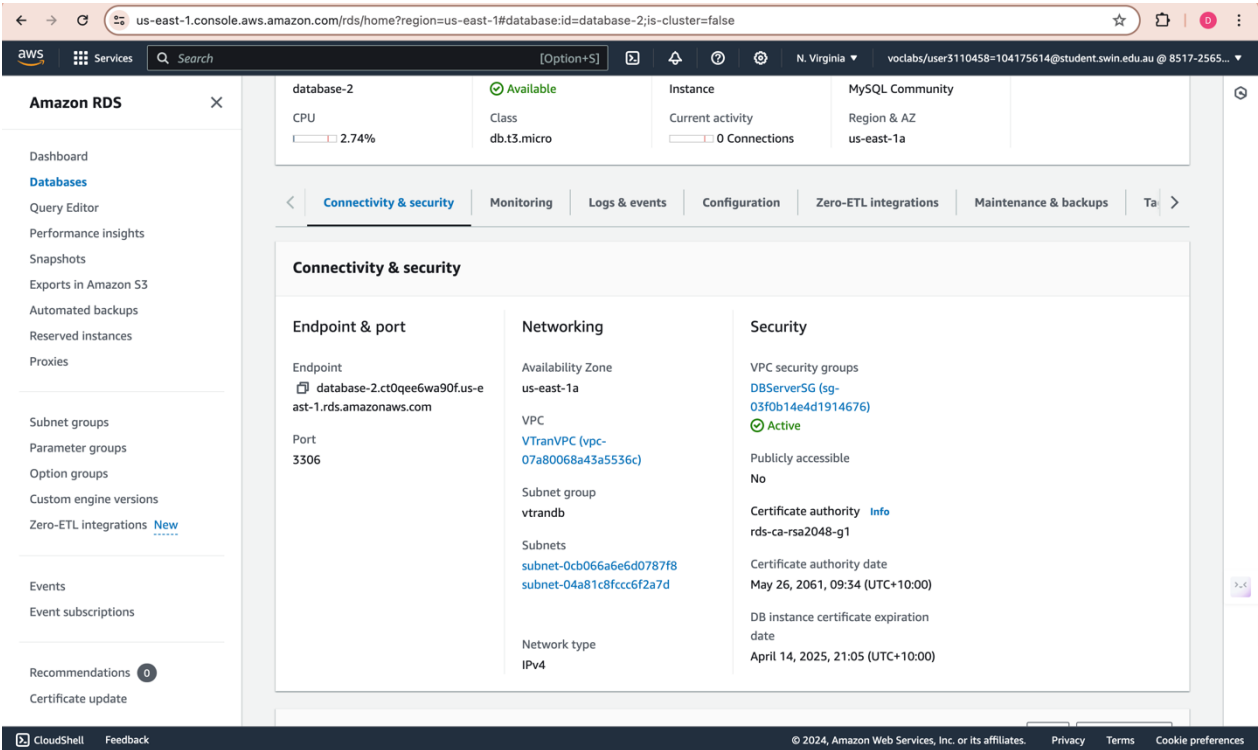


Figure 9

We established a database accessible exclusively from the Bastion/WebServer, playing a crucial role in storing and managing metadata linked to photos stored in an S3 Bucket. Within AWS, this database features a table named "photos," containing fields such as Photo title (varchar(255) type), Description (varchar(255) type), Creation date (date type), Keywords (varchar(255) type), and a reference to the photo object in S3 (varchar(255) type). This table structure is visually represented in figure 10

below, displaying the layout of the table. The reference to the photo serves as a placeholder for the link to the photos stored in the bucket.

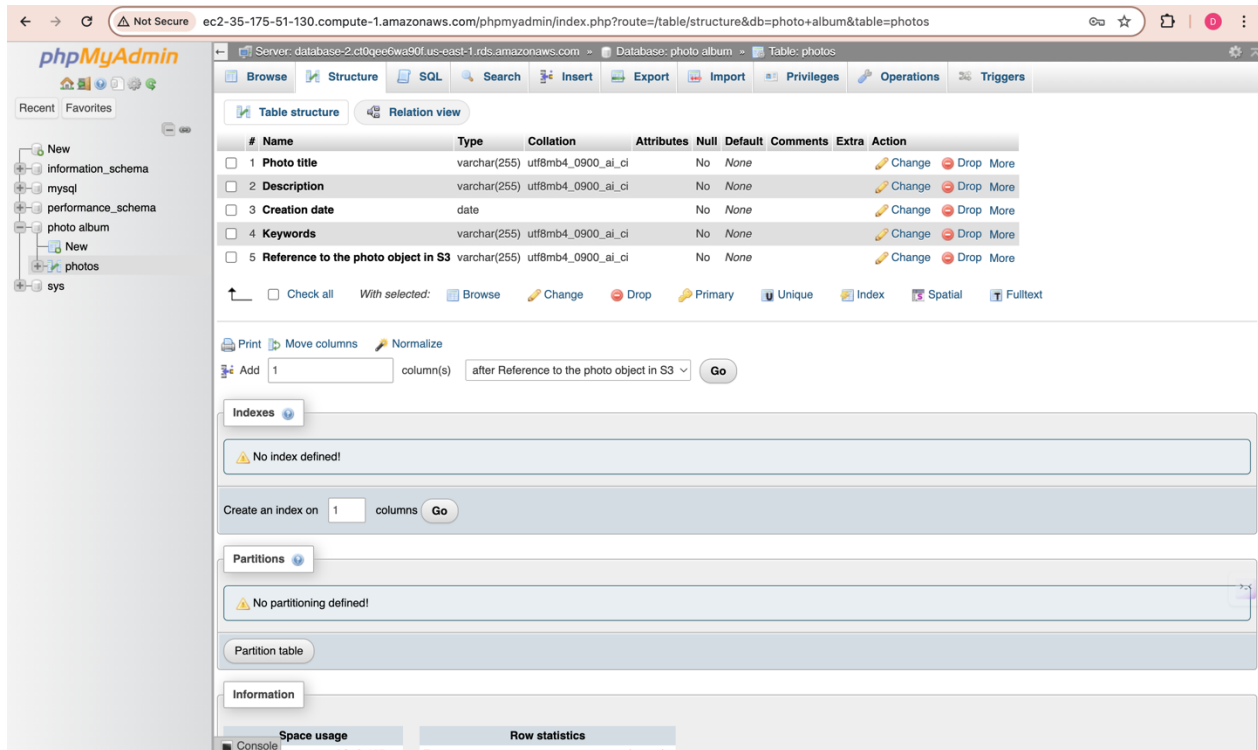


Figure 10

1.5 Create Network ACL

To enhance the security of the web server, I devised and implemented a Network ACL named "PublicSubnet2NACL," strategically restricting ICMP and other essential traffic solely to the associated subnet (Public Subnet 2).

Figures 11 and 12 below illustrate the inbound and outbound rules specified in the assignment criteria, outlining the types of data permitted to enter and exit the system.

The screenshot shows the AWS Management Console interface for Network ACLs. The left sidebar lists various services, with 'Virtual private cloud' expanded. The main content area shows 'Network ACLs (1/3)' for a specific VPC. The 'PublicSubnet2NACL' is selected, and the 'Inbound rules' tab is active, displaying a list of 5 rules.

Name	Network ACL ID	Associated with	Default	VPC ID
-	acl-0f593c939253658b8	4 Subnets	Yes	vpc-07a80068a43a5536c / VTranVPC
<input checked="" type="checkbox"/> PublicSubnet2NACL	acl-0933c9c043aa09a70	-	No	vpc-07a80068a43a5536c / VTranVPC
-	acl-0a5254252555cc6b6	6 Subnets	Yes	vpc-0d7e8cddb0a602f3b

acl-0933c9c043aa09a70 / PublicSubnet2NACL

Details | **Inbound rules** | Outbound rules | Subnet associations | Tags

Inbound rules (5)

Rule number	Type	Protocol	Port range	Source	Allow/Deny
1	SSH (22)	TCP (6)	22	0.0.0.0/0	Allow
2	All ICMP - IPv4	ICMP (1)	All	10.0.4.0/24	Allow
3	Custom TCP	TCP (6)	0	0.0.0.0/0	Allow
4	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Figure 11

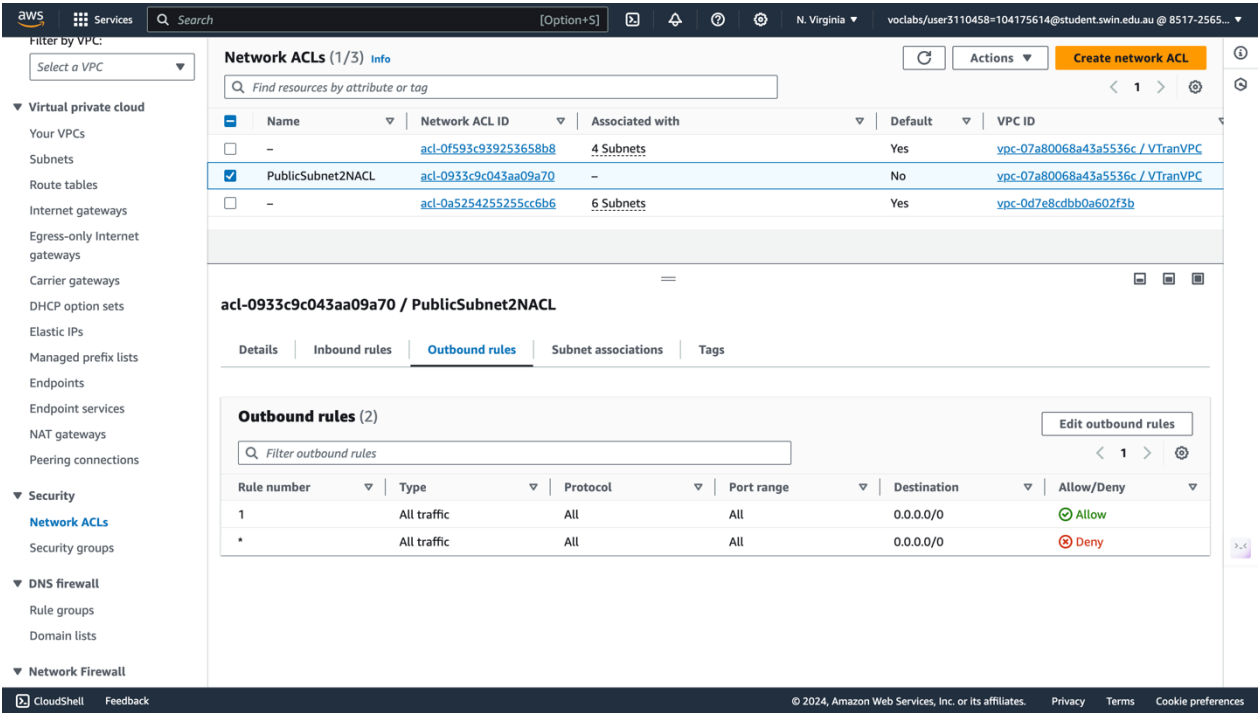


Figure 12

2. Functional requirements of Photo Album website.

2.1 Photo storage

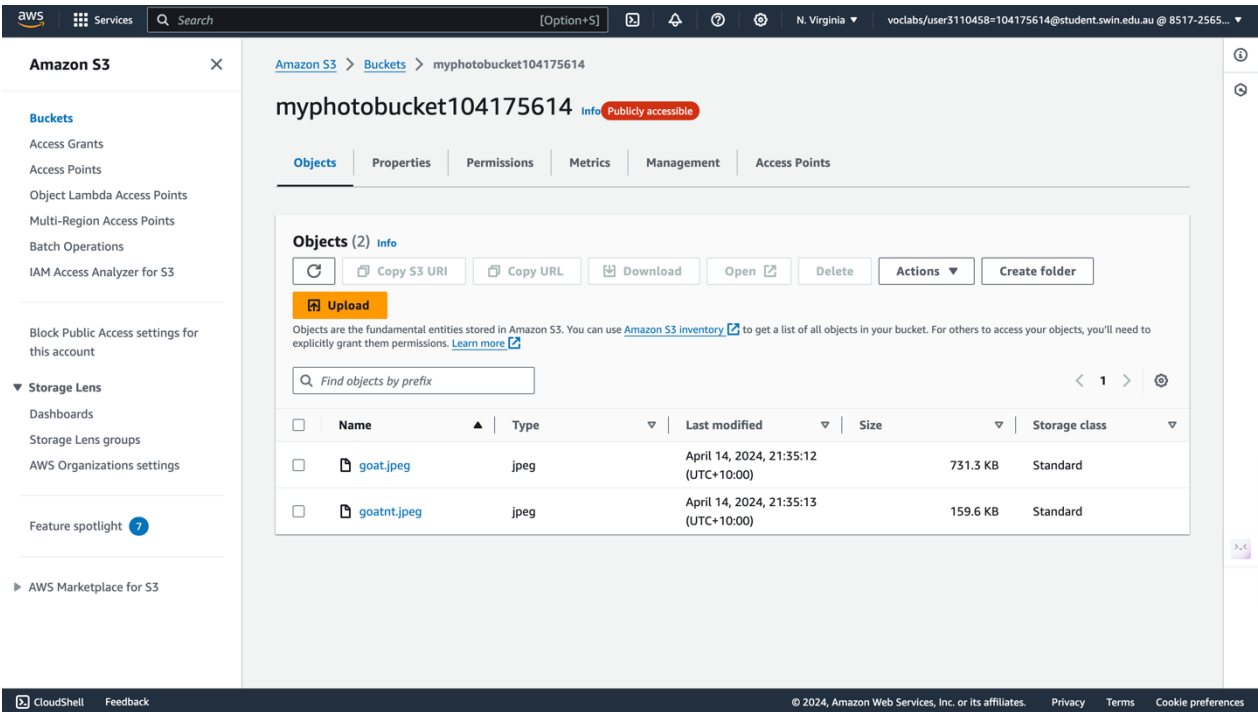
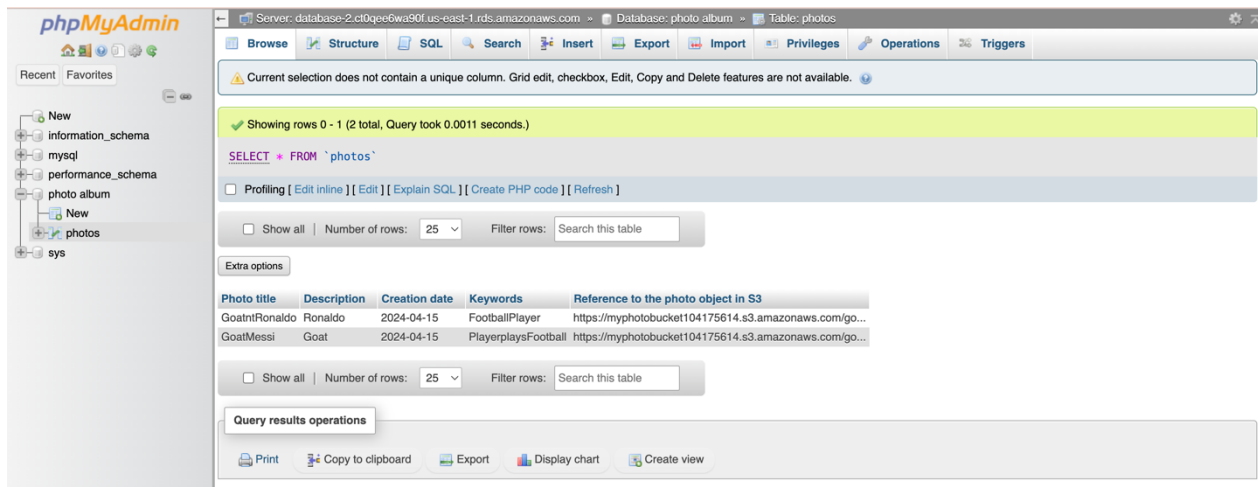


Figure 13

2.2 Photo meta-data in RDS database



Server: database-2-ci0qee6wa90f.us-east-1.rds.amazonaws.com - Database: photo_album - Table: photos

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 1 (2 total, Query took 0.0011 seconds.)

`SELECT * FROM `photos``

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

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

Extra options

Photo title	Description	Creation date	Keywords	Reference to the photo object in S3
GoatntRonaldo	Ronaldo	2024-04-15	FootballPlayer	https://myphotobucket104175614.s3.amazonaws.com/go...
GoatMessi	Goat	2024-04-15	PlayerplaysFootball	https://myphotobucket104175614.s3.amazonaws.com/go...

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

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

Figure 14

2.3 Photo Album website functionally

I modified the code in constant.php file as following (figure 15).

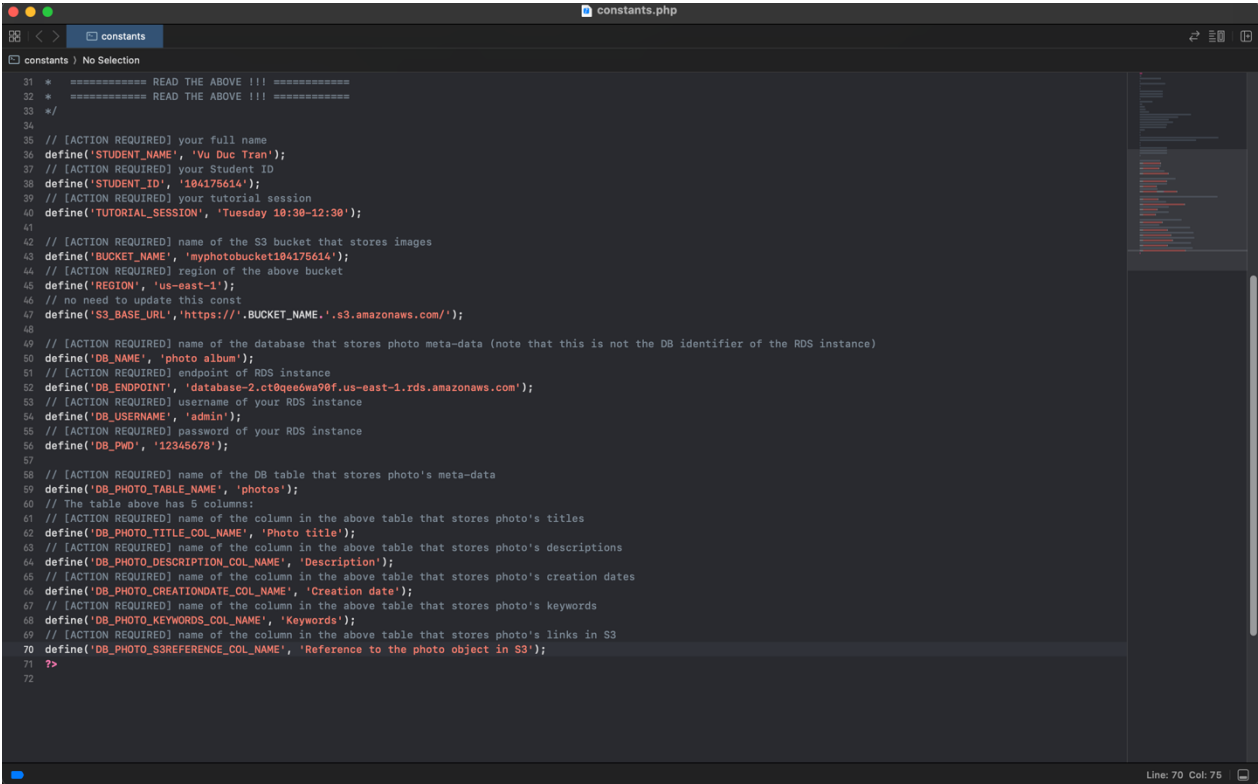


Figure 15

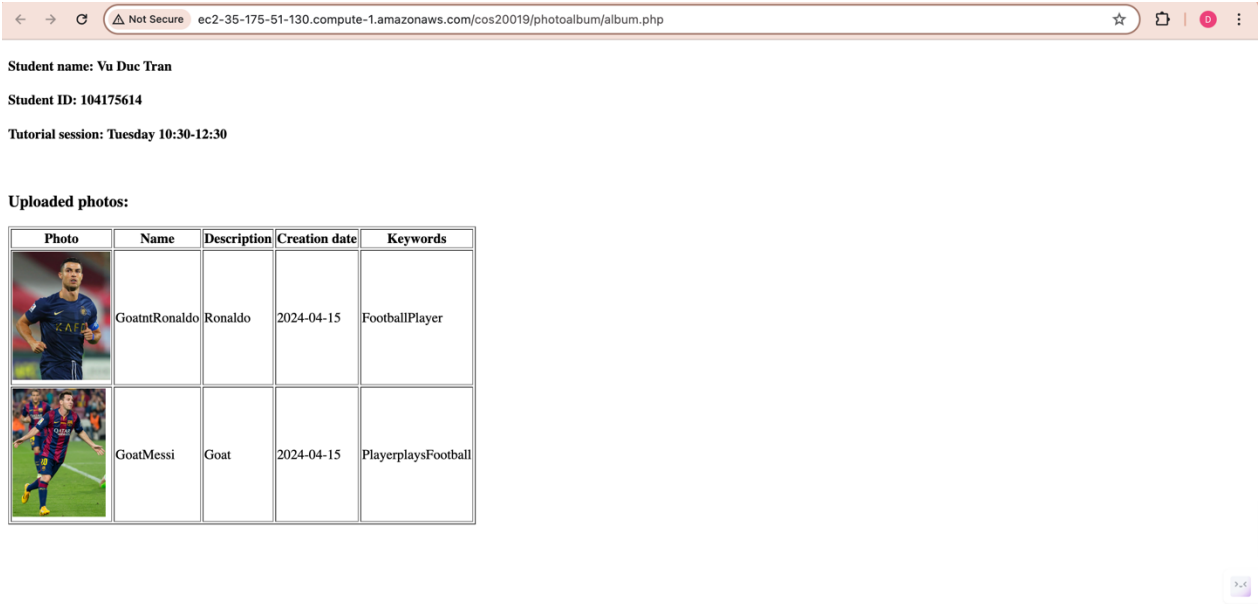


Figure 16

I created an S3 bucket to store our photos and manually uploaded several photos into the bucket to ensure successful storage. All objects within the bucket are made publicly accessible using a policy, as depicted in figure 16, granting public access to all objects within the bucket.