



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

COS 20019- Cloud Computing Architecture

Nguyen Manh Dung

20/5/2023

So this is all my step to finish Assignment 1b, belong with detailed explanation

1.1 VPC

I have created VPC which have my name and required instruction.

Assignment1b_UG_v5.0.pdf

1.1 - VPC:

- Name: [FirstNameInitial][LastName]VPC. For example, if your name is Bill Gates, your VPC would be named "BGatesVPC".
- Region: us-east-1
- Two availability zones each with a private and public subnet with suitable CIDR as shown in the diagram above.
- Associate public subnets with a public route table that routes to the Internet Gateway.

NOTE: due to some incompatibility issues, it is recommended to create your VPC manually (use the 'Create VPC' button in VPC tab). Please do NOT use the "Start VPC Wizard" button in AWS dashboard.

1.2 - Security groups

Create the following security groups, each is associated with each tier shown in the architecture diagram:

COS20019

VPC > Your VPCs > Create VPC > Create VPC resources

Create VPC workflow

Success

- Create VPC: vpc-009584008910217b0
- Enable DNS hostname
- Enable DNS resolution
- Verifying VPC creation: vpc-009584008910217b0
- Create S3 endpoint: vpc-00d2ae0dee50ea13
- Create subnet: subnet-03956294fb8be1fc
- Create subnet: subnet-03f96aaef18397e7e
- Create subnet: subnet-0229f300969d2bbce
- Create subnet: subnet-0ad3711f118882791
- Create internet gateway: iwg-0f5a5a446e1d63fd9b
- Attach internet gateway to the VPC
- Create route table: rtb-04b8579e1a625ed09
- Create route
- Associate route table
- Associate route table
- Create route table: rtb-018ac3abe945ff8f59
- Associate route table
- Create route table: rtb-0ea84746ce6a0fc13
- Associate route table
- Verifying route table creation
- Associate S3 endpoint with private subnet route tables: vpc-00d2ae0dee50ea13

View VPC

1.2 Create Security groups

5 security groups has been created which fits assignment requirement.

Assignment1b_UG_v5.0.pdf

1.2 - Security groups

Create the following security groups, each is associated with each tier shown in the architecture diagram:

COS20019

School of Science, Computing and Engineering Technologies	Swinburne University of Technology	
Security group name	Protocols	Source
TestinstanceSG	All traffic	Anywhere
WebServerSG	HTTP (80), SSH (22)	Anywhere
	ICMP	TestInstanceSG
DBServerSG	MySQL (3306)	WebServerSG

1.3 – EC2 virtual machine

You will create two EC2 instances, a test instance and a bastion/web server instance.

1.3.1 – Bastion/Web server instance

Your web server must be deployed on an EC2 instance in Public Subnet 2. This EC2 Instance should be configured similar to the EC2 created in Assignment 1A:

us-east-1.console.aws.amazon.com/vpc/home?region=

Basic details

Security group name: Info
TestinstanceSG

Description: Info
TestInstancesSG

VPC: Info

Inbound rule 1

Type	Protocol	Port range	Source
All traffic	All	All	Anywhere-IPv4

Outbound rules

Figure 1: TestInstanceSG security group

The screenshot shows two windows side-by-side. On the left is a Microsoft Edge browser displaying a PDF titled 'Assignment1b_UG_v5.00.pdf'. The PDF contains sections on security groups and EC2 virtual machines. On the right is a Microsoft Edge browser displaying the AWS VPC console. The 'Inbound rules' section is visible, showing two rules: one for SSH (Protocol TCP, Port range 22, Source Anywhere-IPv4) and one for HTTP (Protocol TCP, Port range 80, Source Anywhere-IPv4). The bottom status bar shows the date as 6/15/2023 and the time as 4:28 PM.

Figure 2: WebServerSG security group, with inbound rules below

The screenshot shows two windows side-by-side. On the left is a Microsoft Edge browser displaying a PDF titled 'Assignment1b_UG_v5.00.pdf'. The PDF contains sections on security groups and EC2 virtual machines. On the right is a Microsoft Edge browser displaying the AWS VPC console. The 'Inbound rules' section is visible, showing three rules: one for HTTP (Protocol TCP, Port range 80, Source Anywhere-IPv4), one for SSH (Protocol TCP, Port range 22, Source Anywhere-IPv4), and one for ICMP (Protocol ICMP, Port range All, Source Custom, Description optional). The bottom status bar shows the date as 6/16/2023 and the time as 10:11 PM.

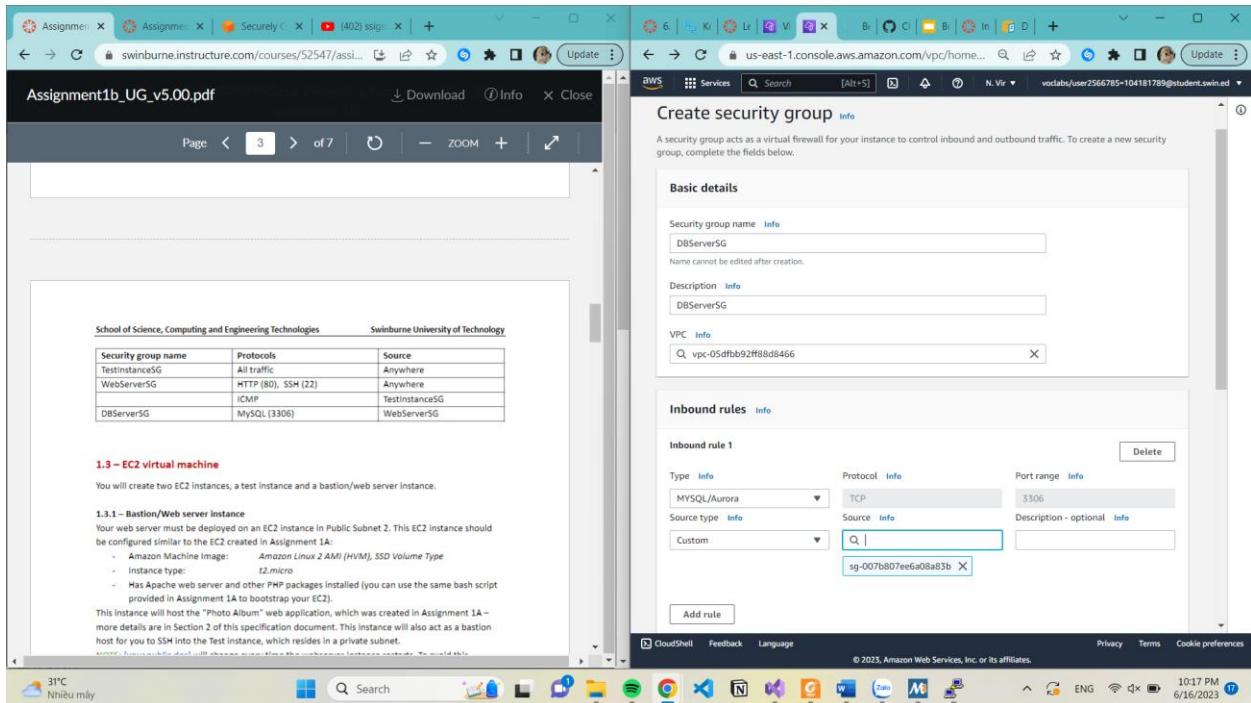


Figure: DBServerSG

1.3 EC2 virtual instance

I have to create two instance which stands for Bastion/Web sever and Test.

Assignment1b_UG_v5.00.pdf

1.3 – EC2 virtual machine

You will create two EC2 instances, a test instance and a bastion/web server instance.

1.3.1 – Bastion/Web server instance

Your web server must be deployed on an EC2 Instance in Public Subnet 2. This EC2 Instance should be configured similar to the EC2 created in Assignment 1A:

- Amazon Machine Image: Amazon Linux 2 AMI (HVM), SSD Volume Type
- Instance type: t2.micro
- Has Apache web server and other PHP packages installed (you can use the same bash script provided in Assignment 1A to bootstrap your EC2).

This instance will host the "Photo Album" web application, which was created in Assignment 1A – more details are in Section 2 of this specification document. This instance will also act as a bastion host for you to SSH into the Test instance, which resides in a private subnet.

NOTE: [your.publicdns] will change every time the webserver instance restarts. To avoid this behaviour and to ensure your Webserver URL remains persistent, add an Elastic IP Address to this instance by allocating an Elastic IP Address in the same region under the Network and Security section in the left menu of the EC2 service page, then associate this new EIP to your Bastion/Web server instance. The public IP address of your instance should now automatically match this Elastic IP address.

1.3.2 – Test instance

This instance will be used for demonstration purposes only. It does not contribute to the functionality of Photo Album website. You will SSH into this instance and ping the web server (using "ping" command in Linux). Please take a screenshot(s) of the Linux terminal to demonstrate that:

- You are able to SSH into an instance in a private subnet (which is this Test instance). For instructions on how to connect to a private EC2 instance through a bastion host, you can refer to [running-in-a-private-amazon-vpc/](https://aws.amazon.com/blogs/security/securing-connect-to-linux-instances-running-in-a-private-amazon-vpc/)

Instances (1/5) Info

Name	Instance ID	Instance state	Instance type	Status check
Mdungweb	i-0ef0b3ed4def84332	Running	t2.micro	2/2 checks passed

- I have to allocate elastic IP address to avoid my IP to change everytime instance restart.

Assignment1b_UG_v5.00.pdf

1.3 – EC2 virtual machine

You will create two EC2 instances, a test instance and a bastion/web server instance.

1.3.1 – Bastion/Web server instance

Your web server must be deployed on an EC2 Instance in Public Subnet 2. This EC2 Instance should be configured similar to the EC2 created in Assignment 1A:

- Amazon Machine Image: Amazon Linux 2 AMI (HVM), SSD Volume Type
- Instance type: t2.micro
- Has Apache web server and other PHP packages installed (you can use the same bash script provided in Assignment 1A to bootstrap your EC2).

This instance will host the "Photo Album" web application, which was created in Assignment 1A – more details are in Section 2 of this specification document. This instance will also act as a bastion host for you to SSH into the Test instance, which resides in a private subnet.

NOTE: [your.publicdns] will change every time the webserver instance restarts. To avoid this behaviour and to ensure your Webserver URL remains persistent, add an Elastic IP Address to this instance by allocating an Elastic IP Address in the same region under the Network and Security section in the left menu of the EC2 service page, then associate this new EIP to your Bastion/Web server instance. The public IP address of your instance should now automatically match this Elastic IP address.

1.3.2 – Test instance

This instance will be used for demonstration purposes only. It does not contribute to the functionality of Photo Album website. You will SSH into this instance and ping the web server (using "ping" command in Linux). Please take a screenshot(s) of the Linux terminal to demonstrate that:

- You are able to SSH into an instance in a private subnet (which is this Test instance). For instructions on how to connect to a private EC2 instance through a bastion host, you can refer to [running-in-a-private-amazon-vpc/](https://aws.amazon.com/blogs/security/securing-connect-to-linux-instances-running-in-a-private-amazon-vpc/)

Elastic IP addresses > Allocate Elastic IP address

Allocate Elastic IP address

Elastic IP address settings

Network Border Group

Q us-east-1

Public IPv4 address pool

Amazon's pool of IPv4 addresses

Customer owned pool of IPv4 addresses

Global static IP addresses

AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network. Learn more

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

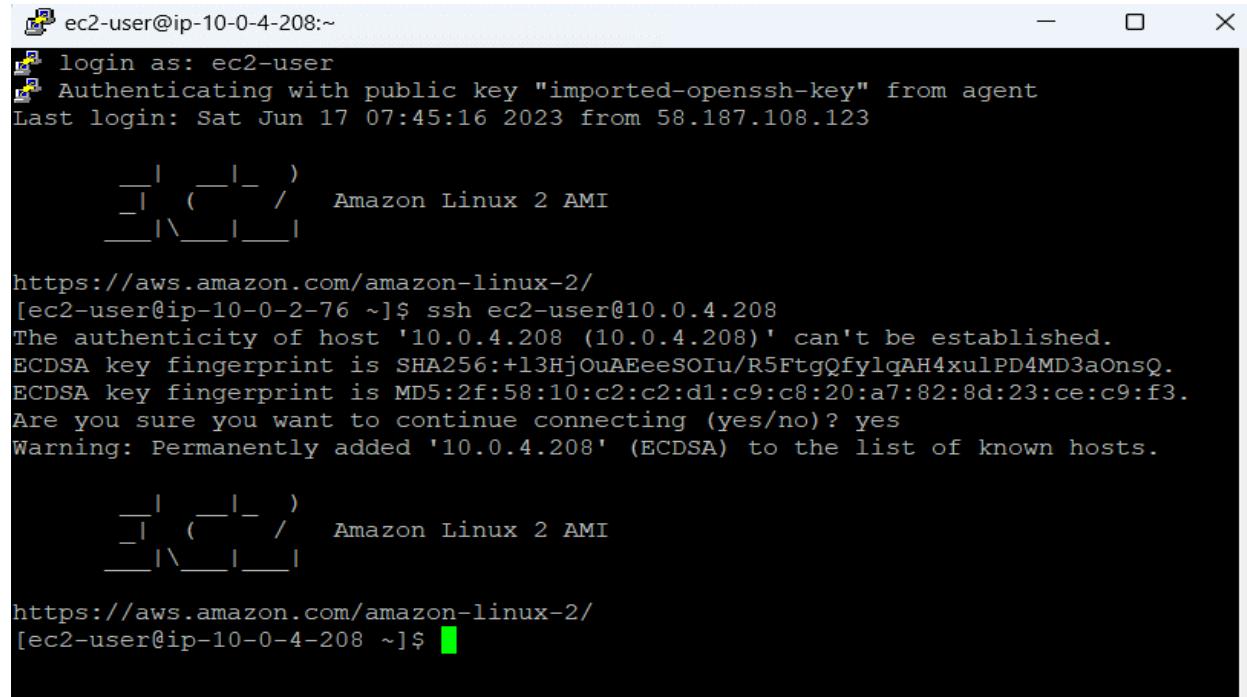
Add new tag

You can add up to 50 more tags

Figure: Test instance sucessful created

Figure: Above screenshotted is the step to convert the key pair that i created in previous step from .pem to .pk

1.3.2 This step is to access SSH of Test instance



```

ec2-user@ip-10-0-4-208:~$ login as: ec2-user
Authenticating with public key "imported-openssh-key" from agent
Last login: Sat Jun 17 07:45:16 2023 from 58.187.108.123

[|_| ( _ / ) Amazon Linux 2 AMI
[|_| \_ |__|]

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-2-76 ~]$ ssh ec2-user@10.0.4.208
The authenticity of host '10.0.4.208 (10.0.4.208)' can't be established.
ECDSA key fingerprint is SHA256:+13HjOuAEeeSOIu/R5FtgQfylqAH4xulPD4MD3aOnsQ.
ECDSA key fingerprint is MD5:2f:58:10:c2:c2:d1:c9:c8:20:a7:82:8d:23:ce:c9:f3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.4.208' (ECDSA) to the list of known hosts.

[|_| ( _ / ) Amazon Linux 2 AMI
[|_| \_ |__|]

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-4-208 ~]$ 

```

Figure: Access instance

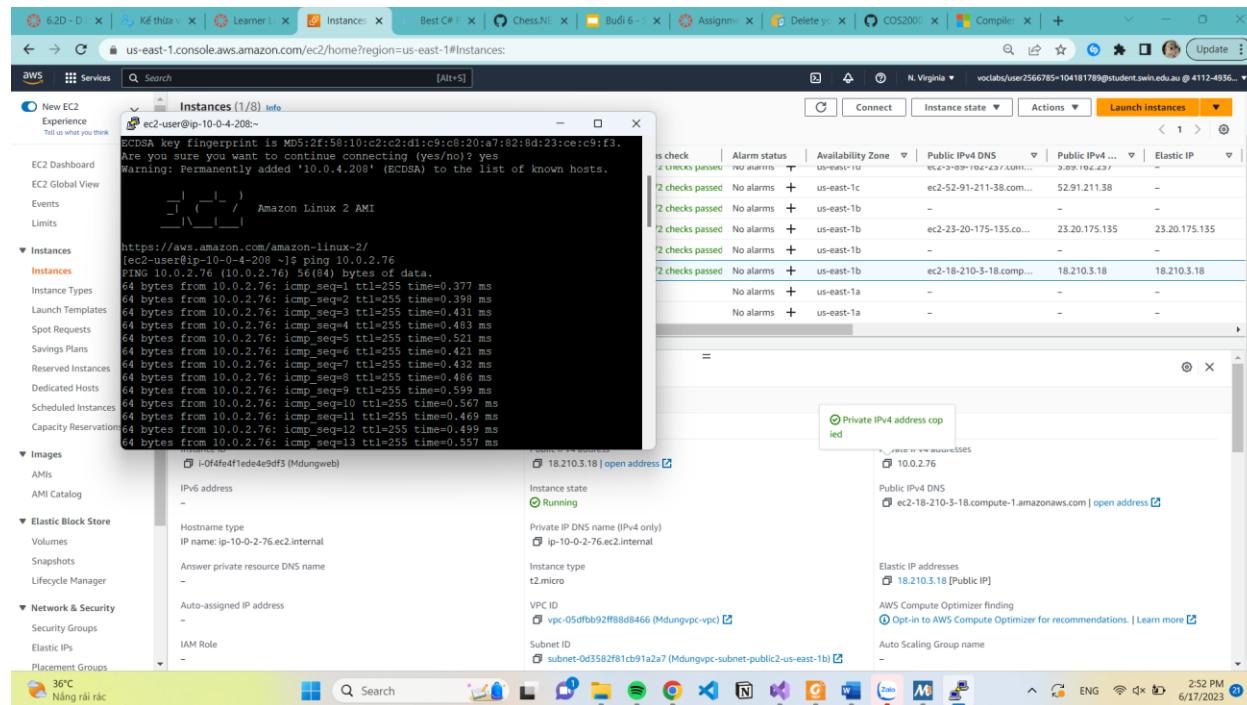
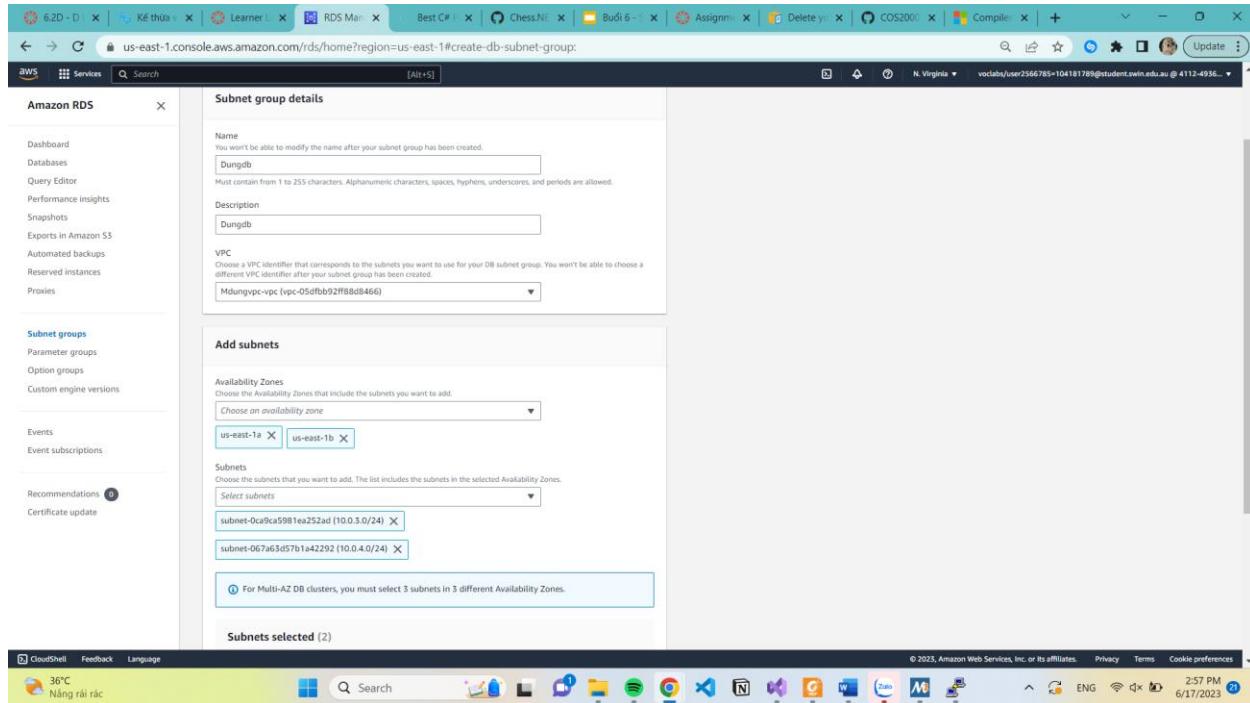
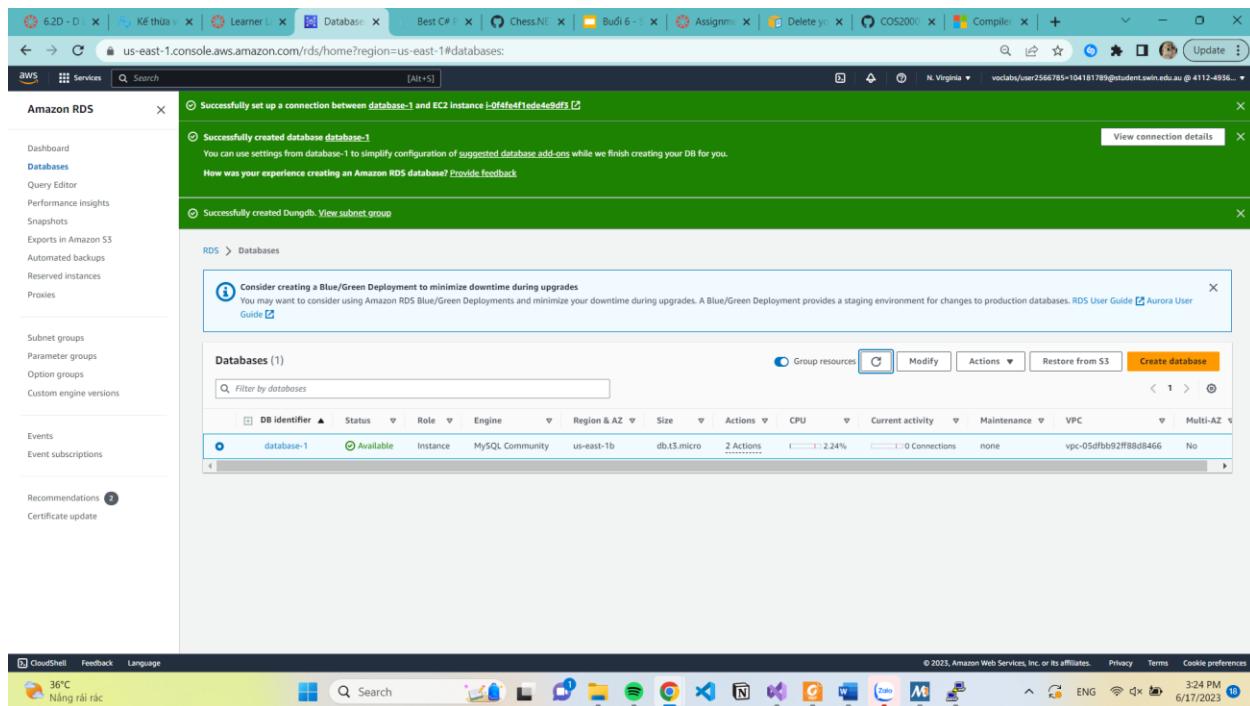


Figure: Establish a connection (ICMP ping) between Test instance and Web server instance

1.4 Create RDS database instance



Create database



Create a database in your RDS instance with a table called photos that stores meta-data about the photos stored in the S3 bucket.

1.5 Create Network ACL

This step is to add an additional layer of security to your web server called Network ACL, below is all the step to deploy it, with all requirement be filled.

The left side of the screenshot shows a web browser window with a URL like swinburne.instructure.com/courses/52547/assignments/1. The content discusses creating a database in RDS and deploying a Network ACL (NACL) to limit traffic to a specific subnet. It includes a list of required columns for the database table and network rules for the NACL.

The right side of the screenshot shows the AWS Management Console, specifically the VPC > Network ACLs > acl-09837b4b38b983e7b / PublicSubnet2NACL > Edit subnet associations page. It lists available subnets and their associations with the NACL. A specific subnet, subnet-0d3582f81cb91a2a7, is selected and highlighted.

This screenshot is identical to the one above, showing the assignment instructions on the left and the AWS VPC Network ACLs configuration page on the right. The focus is on setting up inbound rules for the NACL.

The screenshot shows two side-by-side browser windows. The left window displays assignment instructions from swinburne.instructure.com/courses/52547/assignments/1. It includes a list of database columns for a 'photos' table and requirements for Network ACL (NACL) rules. The right window shows the AWS Management Console VPC service, specifically the inbound rule configuration for a subnet. Two rules are listed: one for ICMP (Protocol: All ICMP - IPv4, Port range: ICMP (1), Source: 10.0.4.0/24, Action: Allow) and one for HTTP (Protocol: TCP (80), Port range: 80, Source: 0.0.0.0/0, Action: Allow).

This screenshot shows two browser windows similar to the previous one, but for a different assignment (Assignment 2). The left window shows assignment instructions for creating a 'photos' table and NACL rules. The right window shows the AWS VPC inbound rule configuration for a subnet, listing two rules: one for HTTPS (Protocol: HTTPS (443), Port range: 443, Source: 0.0.0.0/0, Action: Allow) and one for All TCP (Protocol: All TCP, Port range: All, Source: 10.0.3.0/24, Action: Allow).

Create a database in your RDS instance with a table called **photos** that stores meta-data about the photos stored in the S3 bucket. This table should have the following columns:

- Photo title (*varchar(255)* type)
- Description (*varchar(255)* type)
- Creation date (*date* type)
- Keywords (*varchar(255)* type)
- Reference to the photo object in S3 (*varchar(255)* type)

1.5 – Network ACL

To add an additional layer of security to your web server, you have been asked to design and deploy a Network ACL (named "PublicSubnet2NACL") that limits ICMP and other necessary traffic to the corresponding subnet (Public Subnet 2). This NACL must follow the least-privilege principle. In other words, irrelevant traffic from irrelevant sources must not be allowed. To be specific, the NACL:

- must ALLOW SSH(22) traffic from anywhere so that you can access the WebServer instance.
- must ALLOW ICMP traffic only from the subnet that contains the Test Instance.
- must ALLOW other necessary traffic so that the Photo Album website is fully functional for users from anywhere.

2. Functional requirements of Photo Album website

Your Photo Album website must have the following functional requirements.

2.1 – Photo storage

Create an S3 bucket to store your photos. Manually upload some photos onto S3 bucket that you just created and ensure they have been successfully uploaded.

¹ Ideally, SSH(22) traffic should only be allowed from your home network's public IPv4 address range since common users do not need to access the web server. But for simplicity, you can allow SSH from anywhere in this assignment.

Inbound rule configuration (AWS VPC console):

- Rule number: **Info**
- Type: **Info** (All TCP)
- Protocol: **Info** (TCP (6))
- Port range: **Info** (All)
- Source: **Info** (10.0.4.0/24=)
- Allow/Deny: **Info** (Allow)

Create a database in your RDS instance with a table called **photos** that stores meta-data about the photos stored in the S3 bucket. This table should have the following columns:

- Photo title (*varchar(255)* type)
- Description (*varchar(255)* type)
- Creation date (*date* type)
- Keywords (*varchar(255)* type)
- Reference to the photo object in S3 (*varchar(255)* type)

1.5 – Network ACL

To add an additional layer of security to your web server, you have been asked to design and deploy a Network ACL (named "PublicSubnet2NACL") that limits ICMP and other necessary traffic to the corresponding subnet (Public Subnet 2). This NACL must follow the least-privilege principle. In other words, irrelevant traffic from irrelevant sources must not be allowed. To be specific, the NACL:

- must ALLOW SSH(22) traffic from anywhere so that you can access the WebServer instance.
- must ALLOW ICMP traffic only from the subnet that contains the Test Instance.
- must ALLOW other necessary traffic so that the Photo Album website is fully functional for users from anywhere.

2. Functional requirements of Photo Album website

Your Photo Album website must have the following functional requirements.

2.1 – Photo storage

Create an S3 bucket to store your photos. Manually upload some photos onto S3 bucket that you just created and ensure they have been successfully uploaded.

¹ Ideally, SSH(22) traffic should only be allowed from your home network's public IPv4 address range since common users do not need to access the web server. But for simplicity, you can allow SSH from anywhere in this assignment.

Edit outbound rules (AWS VPC console):

- Outbound rule 1:
 - Rule number: **Info**
 - Type: **Info** (All traffic)
 - Protocol: **Info** (All)
 - Port range: **Info** (All)
 - Destination: **Info** (0.0.0.0/0)
 - Allow/Deny: **Info** (Allow)
- Outbound rule 2:
 - Rule number: **Info**
 - Type: **Info** (All traffic)
 - Protocol: **Info** (All)
 - Port range: **Info** (All)
 - Destination: **Info**

2. Functional requirements of Photo Album website.

2.1 Photo storage

Create an S3 bucket to store photos

The screenshot displays two browser windows side-by-side. The left window is a course assignment page from Swinburne Instructure, showing a warning message about SSH traffic and a note about bucket naming rules. The right window is the 'Create bucket' page on the AWS S3 console. It shows the 'General configuration' section where the bucket name is set to 'Mdungbucket', the AWS Region is 'US East (N. Virginia) us-east-1', and the 'Copy settings from existing bucket - optional' dropdown is set to 'Choose bucket'. Below this, the 'Object Ownership' section shows that 'ACLs disabled (recommended)' is selected, indicating that all objects in the bucket are owned by the account. The status bar at the bottom of the browser indicates the date and time as 6/17/2023 at 4:07 PM.

2.2 Photo meta-data in RDS database

The screenshot shows the 'Table structure' view in phpMyAdmin for the 'photos' table in the 'asm1db' database. The table has five columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Photo title	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 to the photo object in S3	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More

Below the table structure, there is a 'Indexes' section with a note that 'No index defined!' and a 'Create an index on' input field. At the bottom, there is an 'Information' section showing 'Space usage' and 'Row statistics' for the table.

2.3 Photo Album website functionally

I have to create 2 column in the database also modify code in constant.php

Photo title	Description	Creation date	Keywords	Reference to the photo object in S3
My photo	photo	2023-06-17	photo	https://mdungbucket.s3.amazonaws.com/IMG_6429.CR2
My dog	doggo	2023-06-17	dog	https://mdungbucket.s3.amazonaws.com/317955489_501


```

// [ACTION REQUIRED] your Student ID
define('STUDENT_ID', '104181789');
// [ACTION REQUIRED] your tutorial session
define('TUTORIAL_SESSION', 'Saturday 04:32PM');

// [ACTION REQUIRED] name of the S3 bucket that stores images
define('BUCKET_NAME', 'mdungbucket');
// [ACTION REQUIRED] region of the above bucket
define('REGION', 'us-east-1');
// no need to update this const
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
define('DB_NAME', 'asmfdb');
// [ACTION REQUIRED] endpoint of RDS instance
define('DB_ENDPOINT', 'database-1.c1bb2grj0bz.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_title');
// [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_to_the_photo_object_in_S3');
?

```

The screenshot shows the MySQL Workbench interface with two rows of data being inserted into the 'photos' table. The columns are title, description, creationdate, keywords, and reference.

Column	Type	Function	Null	Value
title	varchar(255)		My photos	
description	varchar(255)		photo	
creationdate	date		2023-06-17	
keywords	varchar(255)		photo	
reference	varchar(255)		https://edungbucket.s3.amazonaws.com/portrait.jpg	

Column	Type	Function	Null	Value
title	varchar(255)		my dog	
description	varchar(255)		doggo	
creationdate	date		2023-06-14	
keywords	varchar(255)		dog	
reference	varchar(255)		3554489_501645431952589_12455779051967415_n.jpg	

This is my Photo Album website

The screenshot shows a web browser displaying the photoalbum website. The page header includes student information: Student name: Nguyen Manh Dung, Student ID: 104181789, and Tutorial session: Saturday 04:32PM.

Uploaded photos:

Photo	Name	Description	Creation date	Keywords
	My photos	photo	2023-06-17	photo
	my dog	doggo	2023-06-14	dog

