

COS20019 - Cloud Computing Architecture

Assignment 1 - Part B

Creating and deploying Photo Album website onto a simple

Name: Tran Kim Thu

Student ID: 104061810

1.1 Create VPC name and set up some features as requirement of assignment

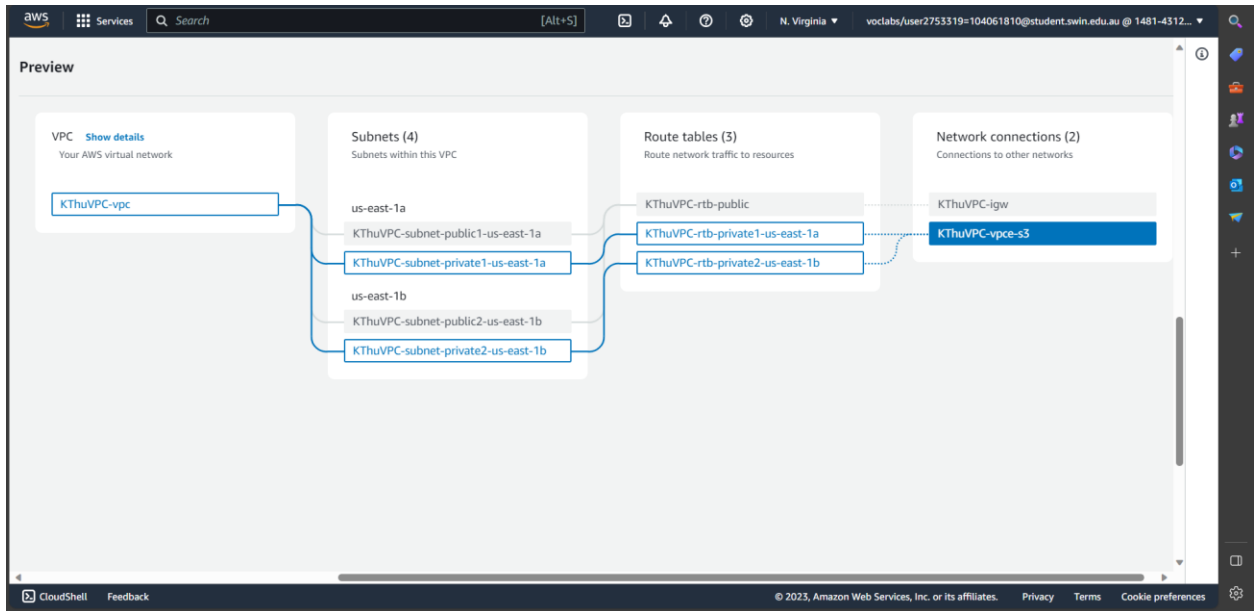
The image displays two screenshots of the AWS Management Console, specifically the 'VPC Settings' page, illustrating the configuration of a new VPC.

Top Screenshot: VPC Settings Overview

- Resources to create:** The 'VPC and more' option is selected.
- Name tag auto-generation:** The 'Auto-generate' checkbox is checked, and the name 'KThuVPC' is entered.
- IPv4 CIDR block:** The '10.0.0.0/16' block is selected, providing 65,536 IPs.
- IPv6 CIDR block:** The 'No IPv6 CIDR block' option is selected.
- Tenancy:** The 'Default' option is selected.
- Number of Availability Zones (AZs):** The 'Default' option is selected.
- Preview:** Shows the VPC 'KThuVPC-vpc' with four subnets: 'us-east-1a' (containing 'KThuVPC-subnet-public1-us-east-1a' and 'KThuVPC-subnet-private1-us-east-1a') and 'us-east-1b' (containing 'KThuVPC-subnet-public2-us-east-1b' and 'KThuVPC-subnet-private2-us-east-1b'). Three route tables, all named 'KThuVPC-rtb-p', are associated with the subnets.

Bottom Screenshot: Subnet Configuration

- Number of public subnets:** Set to 2.
- Number of private subnets:** Set to 4.
- Customize subnets CIDR blocks:**
 - Public subnet CIDR block in us-east-1a: 10.0.1.0/24 (256 IPs)
 - Public subnet CIDR block in us-east-1b: 10.0.2.0/24 (256 IPs)
 - Private subnet CIDR block in us-east-1a: 10.0.3.0/24 (256 IPs)
 - Private subnet CIDR block in us-east-1b: 10.0.4.0/24 (256 IPs)
- NAT gateways (\$):** The 'Choose the number of Availability Zones (AZs) in which to create NAT gateways' section is visible, with a note that there is a charge for each NAT gateway.
- Preview:** Similar to the top screenshot, showing the VPC 'KThuVPC-vpc' and its four subnets and three route tables.



1.2 – Create security groups and set up some features as requirement of assignment

The screenshot shows the 'Create security group' form in the AWS Management Console. The form is titled 'Create security group' and includes the following sections:

- Basic details:**
 - Security group name:** TestInstanceSG
 - Description:** Test Instance
 - VPC:** vpc-00013845d5ec0a845 (KThuVPC-vpc)
- Rules:** This section is currently empty, with a message stating 'This security group has no inbound rules.' and an 'Add rule' button.

Test Instance

VPC info

Q vpc-00013845d5ec0a845

Inbound rules

Type info

Protocol info

Port range info

Source info

Description - optional info

All traffic

All

All

Anyw...

0.0.0.0/0

Delete

Add rule

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Outbound rules

Type info

Protocol info

Port range info

Destination info

Description - optional info

All traffic

All

All

Custom

Delete

CloudShell

Feedback

© 2023, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

VPC > Security Groups > Create security group

Create security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name info

WebServerSG

Name cannot be edited after creation.

Description info

Web Server

VPC info

Q vpc-00013845d5ec0a845

Inbound rules

This security group has no inbound rules.

Add rule

CloudShell

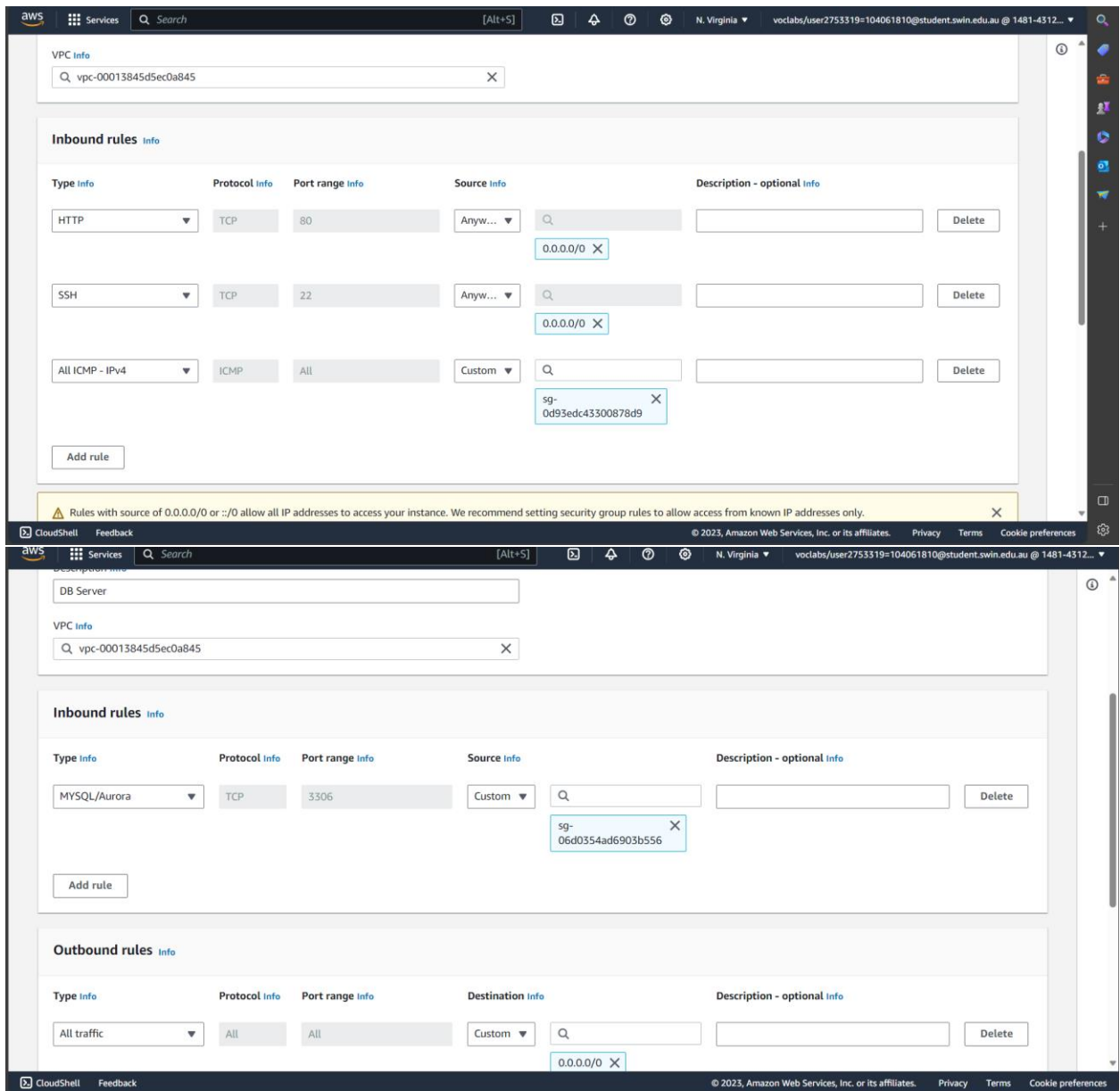
Feedback

© 2023, Amazon Web Services, Inc. or its affiliates.

Privacy

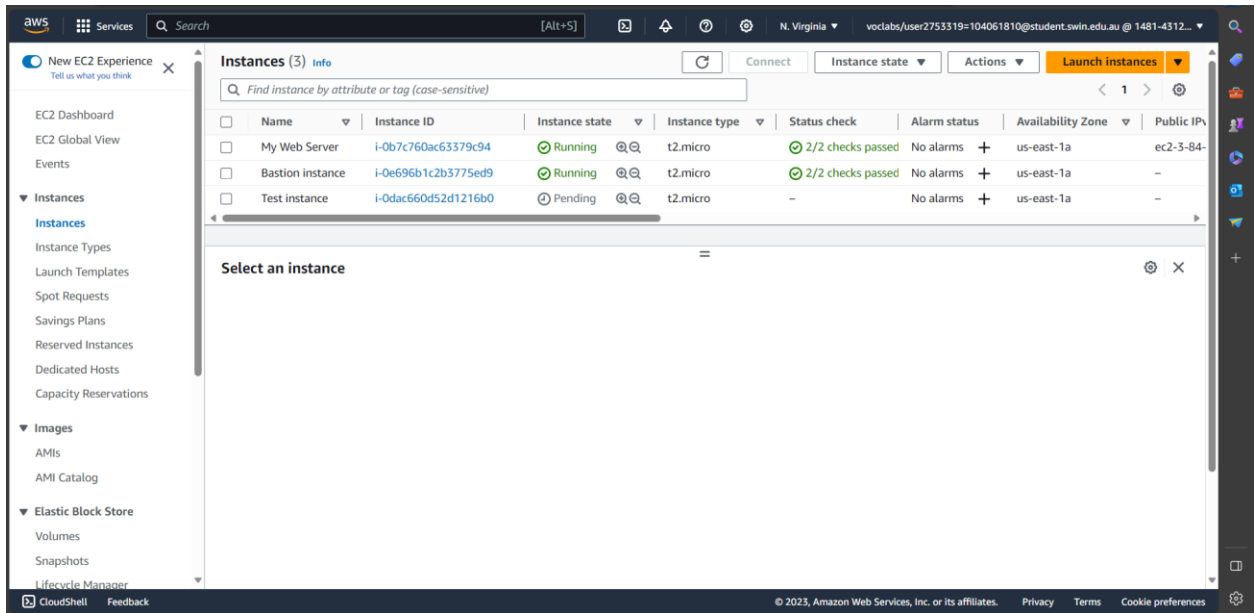
Terms

Cookie preferences



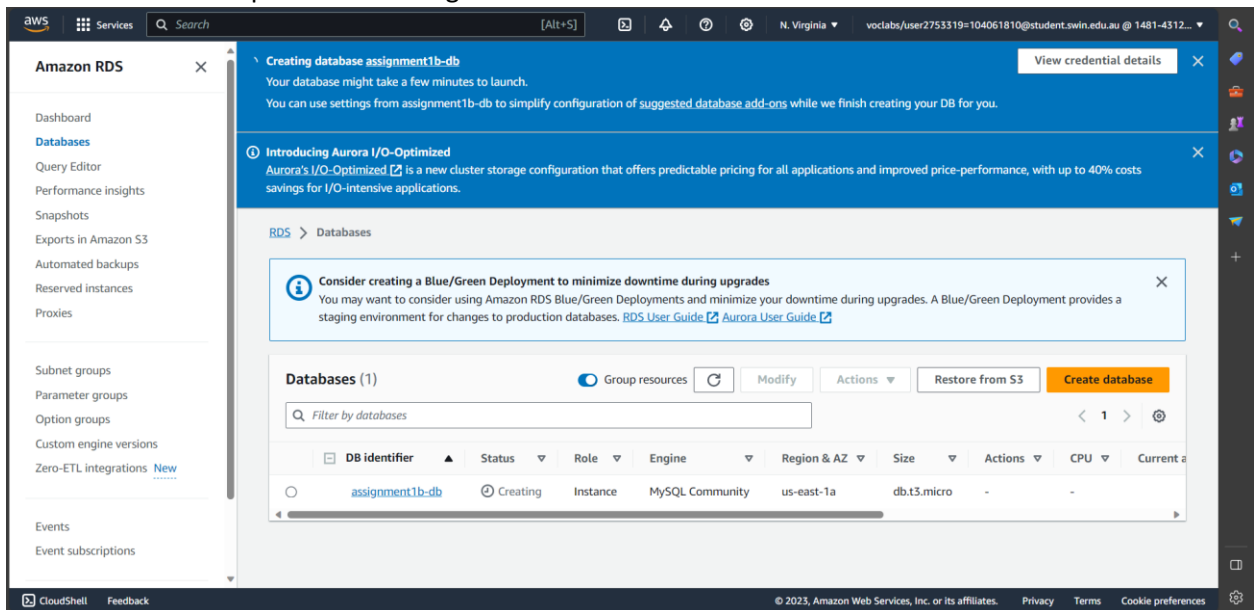
1.3 – EC2 virtual machine

create two EC2 instances(a test instance and a bastion instance) and set up some features as requirments of assignment.



1.4 – RDS database instance

Create RDS DB as requirements of assignment



install phpMyAdmin (a web-based MySQL administration tool) on EC2 web server instance and manage database through phpMyAdmin's UI

The image is a composite of three screenshots illustrating the installation of phpMyAdmin on an EC2 instance.

Top Screenshot: A web browser window showing the SWINBURNE course page for "Supporting materials". It lists "Web server source code: photoalbum_v3.0.zip" and "Install phpMyAdmin on a Linux EC2 instance: Install phpMyAdmin on EC2.pdf". Below this, a terminal window shows the installation process for phpMyAdmin 4.8.2 on a Linux EC2 instance. The terminal output includes the following commands and their results:

```
ec2-user@ip-10-0-2-182:/var/www/html$ cd /var/www/html
ec2-user@ip-10-0-2-182:/var/www/html$ curl -O https://raw.githubusercontent.com/phpmyadmin/phpmyadmin/4.8.2/phpmyadmin
ec2-user@ip-10-0-2-182:/var/www/html$ mv phpmyadmin phpmyadmin-4.8.2
ec2-user@ip-10-0-2-182:/var/www/html$ mv phpmyadmin-4.8.2 phpmyadmin
ec2-user@ip-10-0-2-182:/var/www/html$ ls -la
total 12
drwxr-xr-x 2 root root 4096 Oct 15 11:58 .
drwxr-xr-x 1 root root 4096 Oct 15 11:58 ..
-rw-r--r-- 1 root root 6232 Oct 15 11:58 phpmyadmin
-rw-r--r-- 1 root root 1024 Oct 15 11:58 phpinfo.php
```

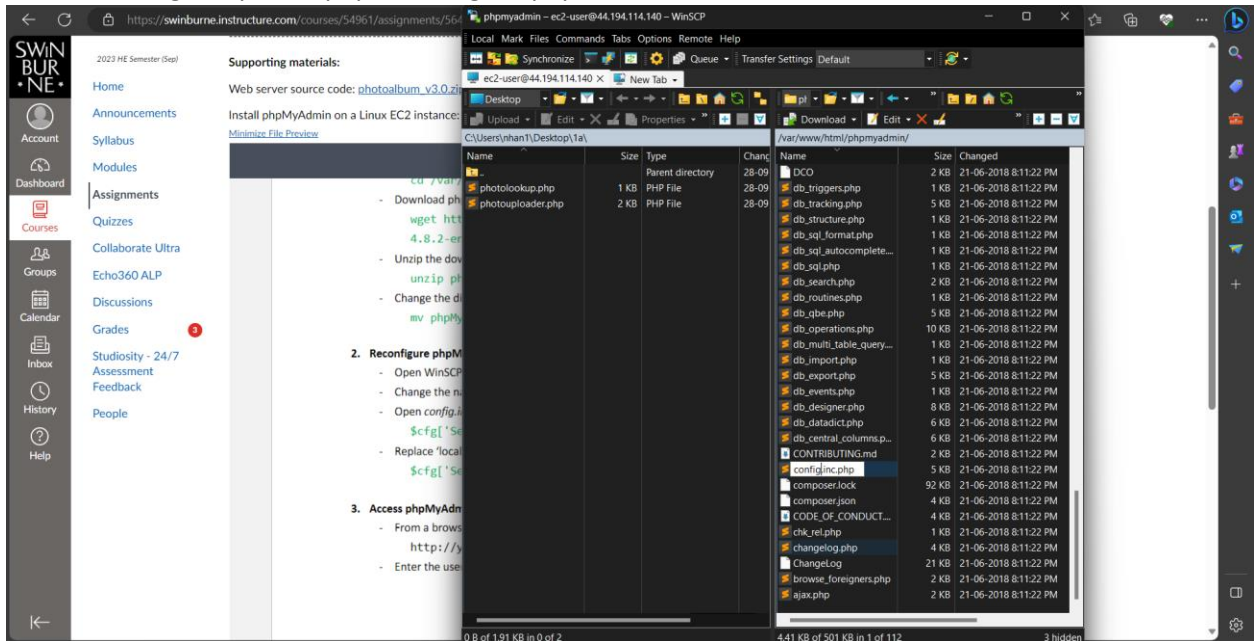
Middle Screenshot: A terminal window showing the installation progress of phpMyAdmin 4.8.2. The output includes the following commands and their results:

```
ec2-user@ip-10-0-2-182:/var/www/html$ cd /var/www/html
ec2-user@ip-10-0-2-182:/var/www/html$ curl -O https://raw.githubusercontent.com/phpmyadmin/phpmyadmin/4.8.2/phpmyadmin
ec2-user@ip-10-0-2-182:/var/www/html$ mv phpmyadmin phpmyadmin-4.8.2
ec2-user@ip-10-0-2-182:/var/www/html$ mv phpmyadmin-4.8.2 phpmyadmin
ec2-user@ip-10-0-2-182:/var/www/html$ ls -la
total 12
drwxr-xr-x 2 root root 4096 Oct 15 11:58 .
drwxr-xr-x 1 root root 4096 Oct 15 11:58 ..
-rw-r--r-- 1 root root 6232 Oct 15 11:58 phpmyadmin
-rw-r--r-- 1 root root 1024 Oct 15 11:58 phpinfo.php
```

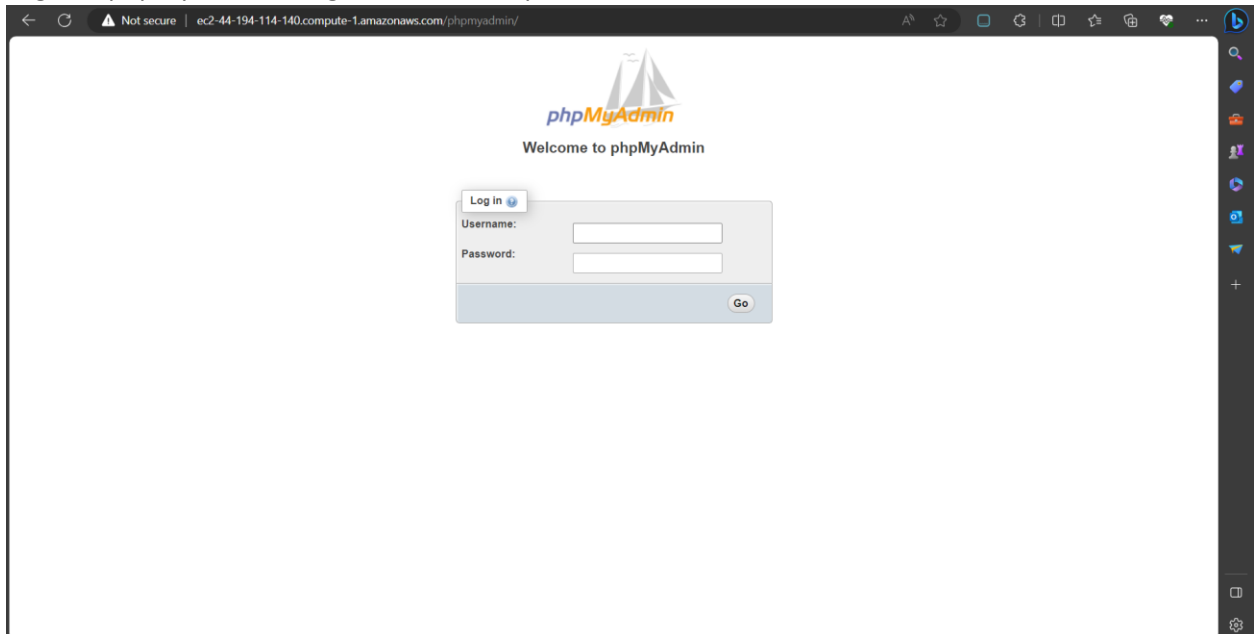
Bottom Screenshot: A file manager window showing the installed files on the EC2 instance. The files listed are:

Name	Size	Type	Changed
..	-	Parent directory	28-09
photoalbum.php	1 KB	PHP File	28-09
photoalbum.php	2 KB	PHP File	28-09
phpmyadmin	-	Directory	28-09
phpmyadmin-4.8.2-en...	6,232 KB	Directory	22-06-2018 1:21:25 AM
phpinfo.php	1 KB	PHP File	15-10-2023 11:58:24 AM

Rename config.sample.inc.php to config.inc.php



Login to phpMyAdmin using username and password created before



Create the database as well as a new table in phpMyadmin using the following SQL commands.

The image shows two screenshots. The top screenshot is from phpMyAdmin, displaying SQL commands to create a database and a table. The bottom screenshot is from the AWS Management Console, showing the configuration of Network ACLs for a VPC.

phpMyAdmin SQL Commands:

```
1 CREATE DATABASE IF NOT EXISTS assignment1b;
2 USE assignment1b;
3 CREATE TABLE photos (
4   photo_title VARCHAR(255),
5   description VARCHAR(255),
6   creation_date DATE,
7   keywords VARCHAR(255),
8   s3_reference VARCHAR(255)
9 );
```

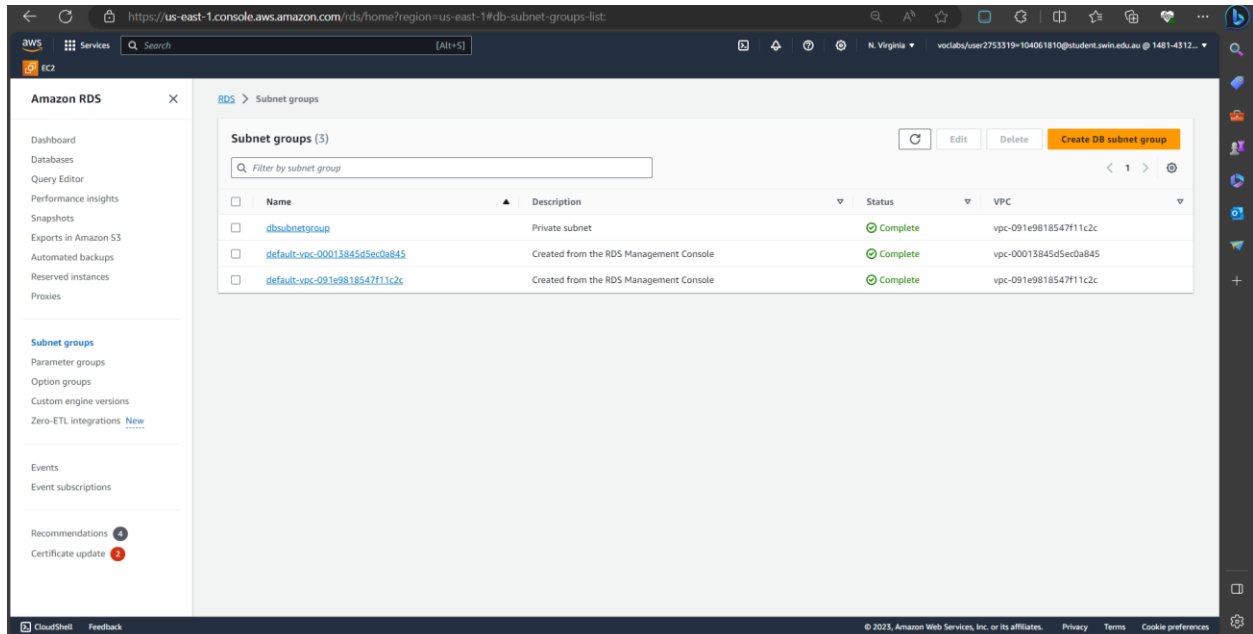
AWS Management Console - Network ACLs (1/5)

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules count	Outbound rules count
-	acl-0204bdc03dfe9fdad	3 Subnets	Yes	vpc-00013845d5ec0a845 / KThuVPC-vpc	2 Inbound rules	2 Outbound rules
PublicSubnet2NACL	acl-09b7ad0760d04055d	subnet-082ef953b9796688f / KThuVPC-subnet...	No	vpc-00013845d5ec0a845 / KThuVPC-vpc	7 Inbound rules	2 Outbound rules
-	acl-0c80840665eef63ea	4 Subnets	Yes	vpc-091e9818547f11c2c / kthuVPC-vpc	2 Inbound rules	2 Outbound rules
-	acl-0c334ea5287017855	6 Subnets	Yes	vpc-0ccb59cb62228bb59	2 Inbound rules	2 Outbound rules
publicsubnet2NACL	acl-034028c544e7737c0	-	No	vpc-091e9818547f11c2c / kthuVPC-vpc	7 Inbound rules	2 Outbound rules

Network ACL Rule Details:

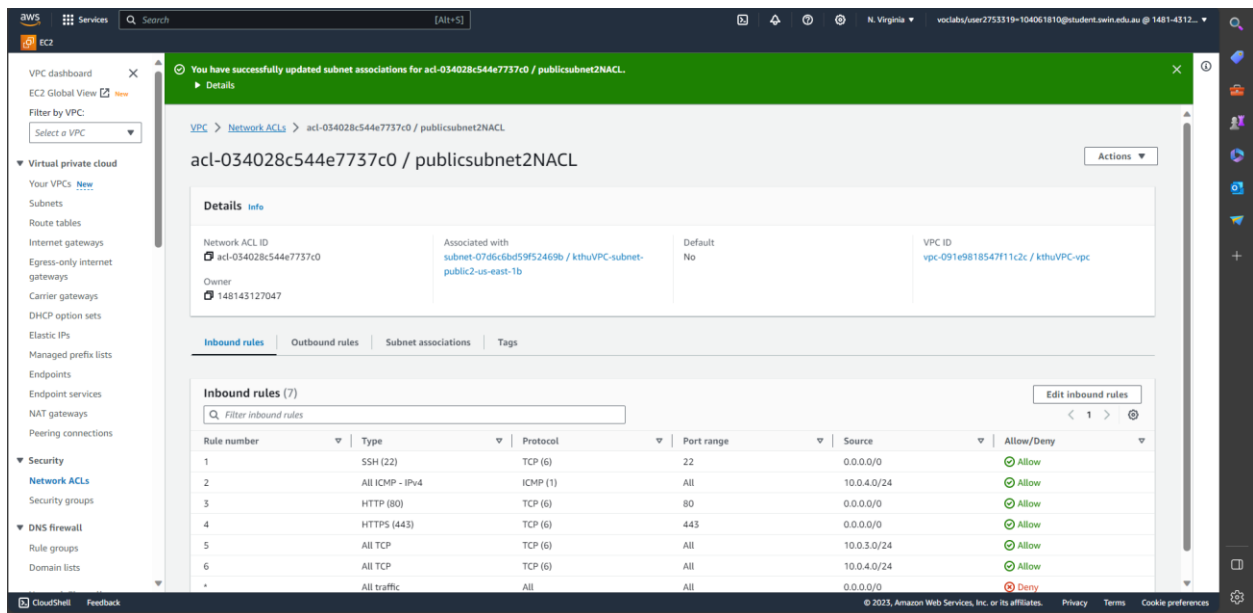
Rule number	Type	Protocol	Port range	Source	Allow/Deny
1	SSH (22)	TCP (6)	22	0.0.0.0/0	Allow
2	All ICMP - IPv4	ICMP (1)	All	10.0.4.0/24	Allow
3	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
4	HTTPS (443)	TCP (6)	443	0.0.0.0/0	Allow
5	All TCP	TCP (6)	All	10.0.3.0/24	Allow
6	All TCP	TCP (6)	All	10.0.4.0/24	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Create a new subnet group named DBSubnetGroup in my VPC.



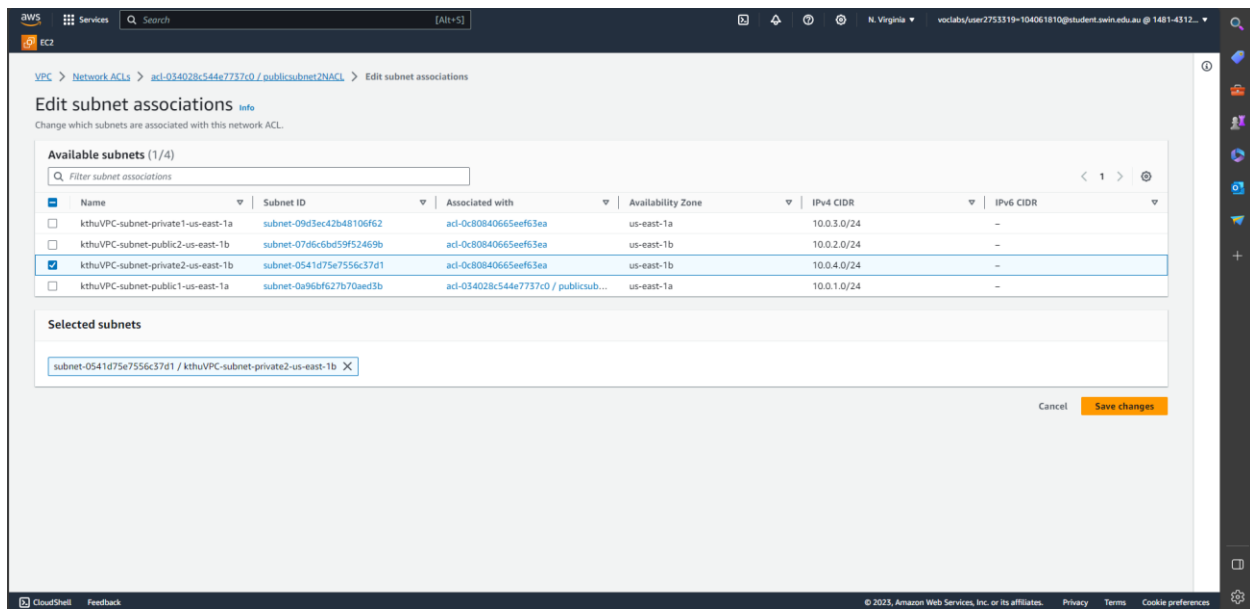
The screenshot shows the Amazon RDS console with the 'Subnet groups' page selected. The page displays a list of three subnet groups. The first group, 'dbsubnetgroup', is a private subnet. The other two are default VPC subnets created from the RDS Management Console. A 'Create DB subnet group' button is visible in the top right corner.

Name	Description	Status	VPC
dbsubnetgroup	Private subnet	Complete	vpc-091e9818547f11c2c
default-vpc-00013845d5ec0a845	Created from the RDS Management Console	Complete	vpc-00013845d5ec0a845
default-vpc-091e9818547f11c2c	Created from the RDS Management Console	Complete	vpc-091e9818547f11c2c

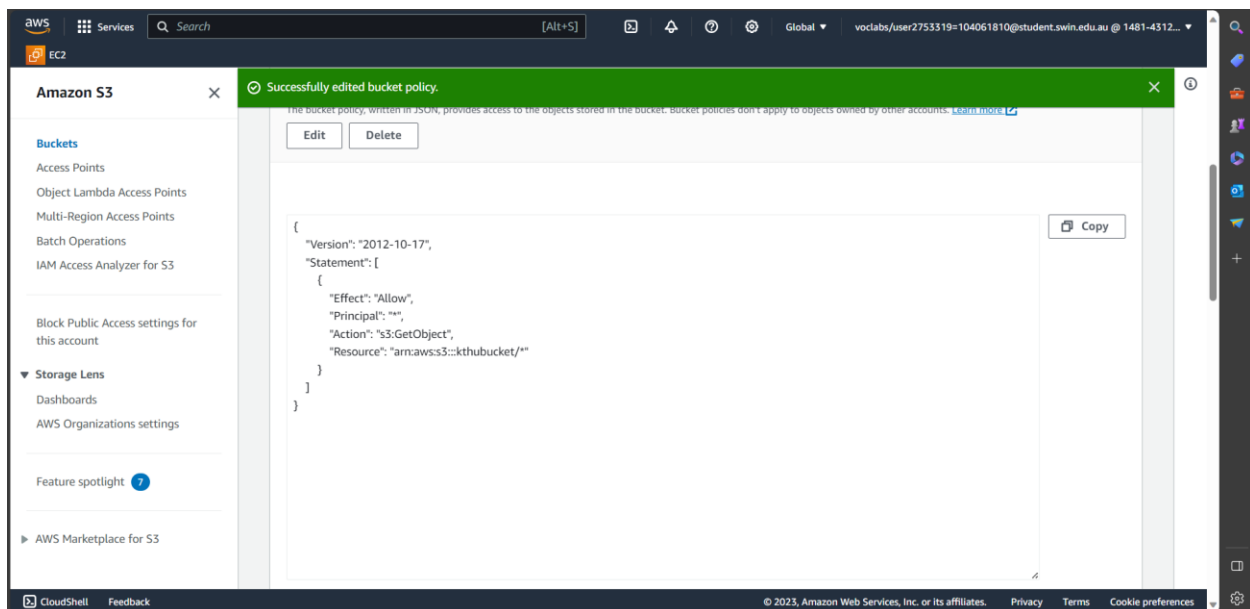


The screenshot shows the Amazon VPC console with the 'Network ACLs' page selected. The details for the Network ACL 'acl-034028c544e7737c0 / publicsubnet2NACL' are displayed. The details section shows the Network ACL ID, Associated with (subnet-0705c6bd59f52469b / kthuVPC-subnet-public2-us-east-1b), Default (No), and VPC ID (vpc-091e9818547f11c2c / kthuVPC-vpc). The 'Inbound rules' tab is selected, showing a list of seven rules. The rules are numbered 1 through 7, with the last rule being 'All traffic' which is denied.

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	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
4	HTTPS (443)	TCP (6)	443	0.0.0.0/0	Allow
5	All TCP	TCP (6)	All	10.0.3.0/24	Allow
6	All TCP	TCP (6)	All	10.0.4.0/24	Allow
7	All traffic	All	All	0.0.0.0/0	Deny



Create an S3 bucket to store photos. Manually upload some photos onto S3 bucket that just created



Amazon S3

Successfully edited Block Public Access settings for this bucket.

Permissions overview

Access

Bucket and objects not public

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access

Off

► Individual Block Public Access settings for this 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](#)

Upload succeeded

View details below.

Summary

Destination	Succeeded	Failed
s3://kthubucket	3 files, 1.1 MB (100.00%)	0 files, 0 B (0%)

Files and folders | Configuration

Files and folders (3 Total, 1.1 MB)

Find by name

Name	Folder	Type	Size	Status	Error
sunset-569093_1280.jpg	-	image/jpeg	446.6 KB	Succeeded	-
eiffel-tower-3349075_1280.jpg	-	image/jpeg	284.8 KB	Succeeded	-
funes-4984899_1280.jpg	-	image/jpeg	388.7 KB	Succeeded	-

Insert metadata into the database using SQL commands and Data records in the database.

phpMyAdmin interface showing the 'photos' table in the 'assignment1b' database. The table contains 3 rows, filtered by 'photo_title' between 'ITALY...' and 'PARIS...'. The query results are displayed in a table format.

photo_title	description	creation_date	keywords	s3_reference
Italy	forest	2018-01-01	italian	https://kthubucket.s3.amazonaws.com/funes_4084899...
New York	sunset	2017-01-01	sun	https://kthubucket.s3.amazonaws.com/sunset.569093...
Paris	eiffel	2022-01-01	paris	https://kthubucket.s3.amazonaws.com/eiffel-tower-3...

Visual Studio Code editor showing the 'photoalbum' project. The file explorer on the left lists files like 'album.php', 'constants.php', 'defaultstyle.css', 'mydb.php', and 'photo.php'. The right pane shows the contents of 'photo.php'.

```
<?php
require_once('mydb.php');
require_once('constants.php');
require_once('album.php');

// Get the photo title from the URL
$photo_title = $_GET['photo_title'];

// Get the photo description from the database
$query = "SELECT * FROM photos WHERE photo_title = '$photo_title'";
$result = mysqli_query($conn, $query);
$row = mysqli_fetch_assoc($result);

// Get the photo keywords from the database
$query = "SELECT * FROM keywords WHERE photo_title = '$photo_title'";
$result = mysqli_query($conn, $query);
$row = mysqli_fetch_assoc($result);




// Get the photo s3 reference from the database
$query = "SELECT * FROM s3_reference WHERE photo_title = '$photo_title'";
$result = mysqli_query($conn, $query);
$row = mysqli_fetch_assoc($result);

// Display the photo information
echo "<div class='photo'>";
echo "<div class='photo_title'>";
echo $photo_title;
echo "</div>";
echo "<div class='photo_description'>";
echo $row['description'];
echo "</div>";
echo "<div class='photo_keywords'>";
echo $row['keywords'];
echo "</div>";
echo "<div class='photo_s3_reference'>";
echo $row['s3_reference'];
echo "</div>";
echo "</div>";
```

This my photos that uploaded on my web server.

Student name: Tran Kim Thu
Student ID: 104061810
Tutorial session: Saturday 9:00 AM

Uploaded photos:

Photo	Name	Description	Creation date	Keywords
	Paris	eiffel	2022-01-01	paris
	Italy	forest	2018-01-01	italian
	New York	sunset	2017-01-01	sun

Ping from TestInstance to WebServer

New EC2 Experience

EC2 Dashboard

EC2 Global View

Events

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups

Elastic IPs

CloudShell

Feedback

Instances (1/3) info

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input checked="" type="checkbox"/> bastion instance	i-000e53ffa0c0151bb	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-44-194-114-140.co...	44.194.114.140	44.194.114.140
<input type="checkbox"/> test instance	i-060f92b85a2fb5448	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	-	-	-
<input type="checkbox"/> My Web Server	i-0b7c760ac63379c94	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-210-131-226.co...	54.210.131.226	-

ec2-user@ip-10-0-4-111:~

```
~
_/_/_/_/
/  V-  \
/_/_/_/_/

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/

19 package(s) needed for security, out of 24 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-2-182 ~]$ ssh ec2-user@10.0.4.111
Last login: Sun Oct 15 16:00:29 2023 from ip-10-0-2-182.ec2.internal

#
_/_/_/_/
/  V-  \
/_/_/_/_/

Amazon Linux 2
#####
AL2 End of Life is 2025-06-30.

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-10-0-4-111 ~]$ ping 10.0.2.182
PING 10.0.2.182 (10.0.2.182) 56(84) bytes of data.

```

Private IPv4 address copied

10.0.2.182

Public IPv4 DNS

ec2-44-194-114-140.compute-1.amazonaws.com [Open address](#)

Elastic IP addresses

44.194.114.140 [Public IP]

AWS Compute Optimizer finding

© 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)