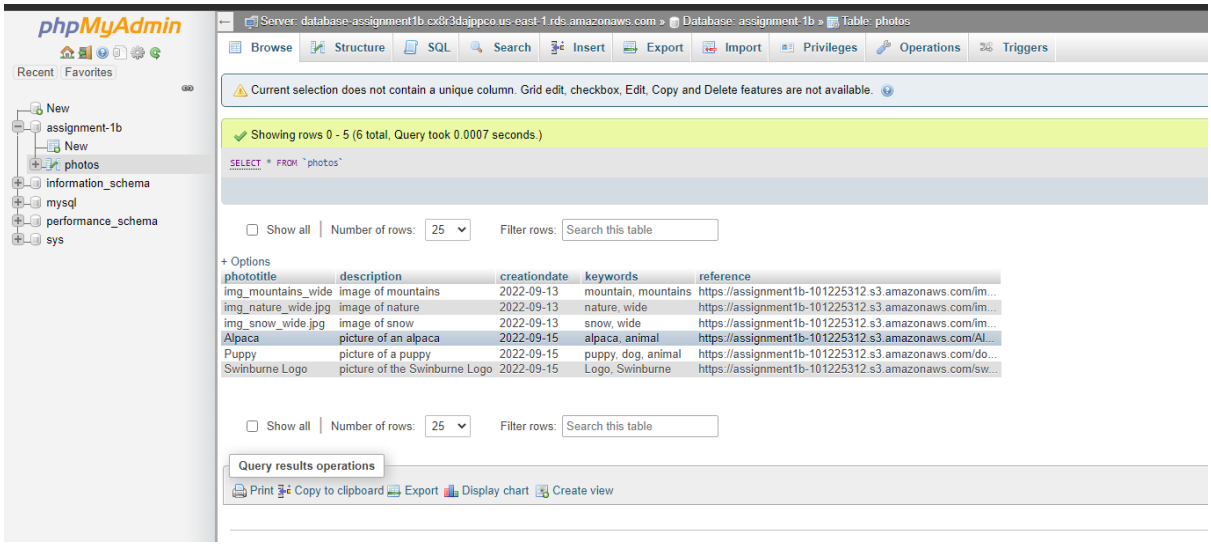
COS20019 Cloud Computing Architecture Assignment 1B

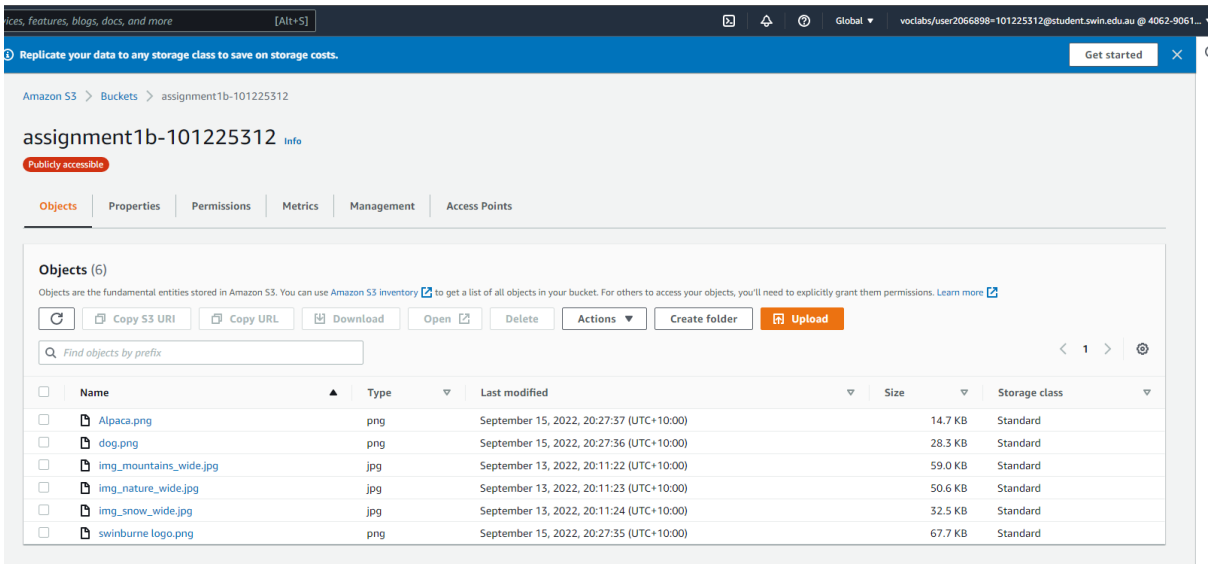
Name: Lai Yee Fung
ID: 101225312
Tutorial Class: Wednesday 6:30p.m.
BA513

URL: http://ec2-35-172-168-37.compute-1.amazonaws.com/cos80001/photoalbum/album.php

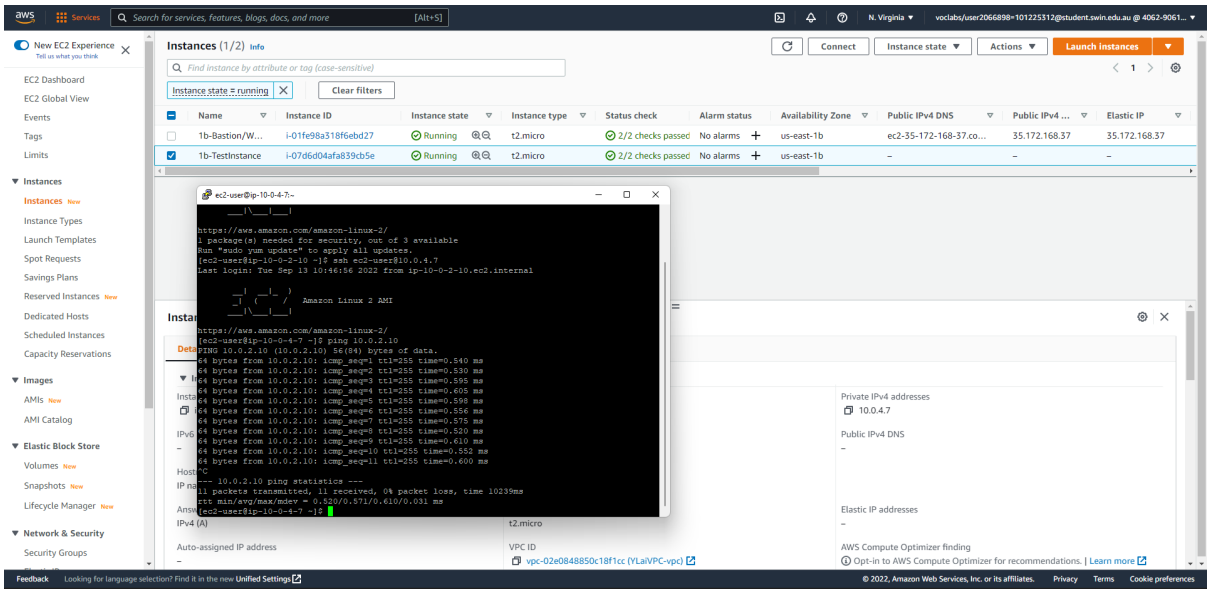
Screenshot of database



Screenshot of S3 bucket



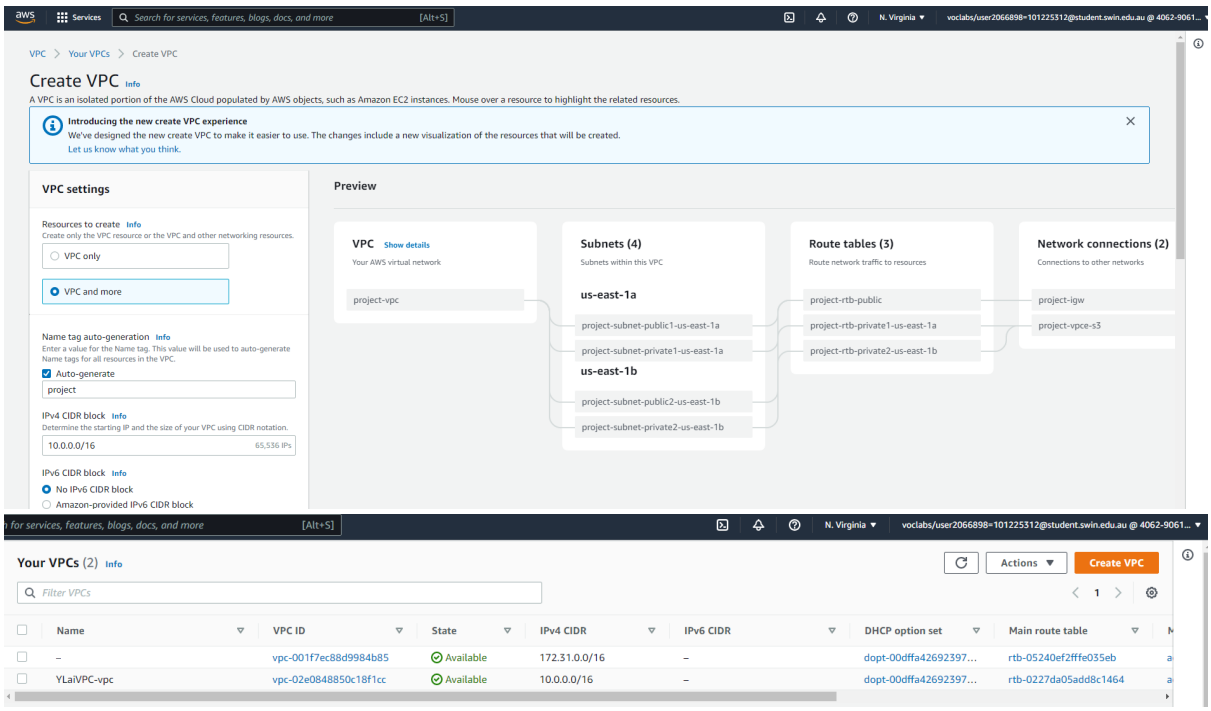
Screenshot of terminal ping from Test instance to Web server EC2 instance



This is done by accessing the private Test instance from the public Bastion/Web Server instance with Pageant agent forwarding.

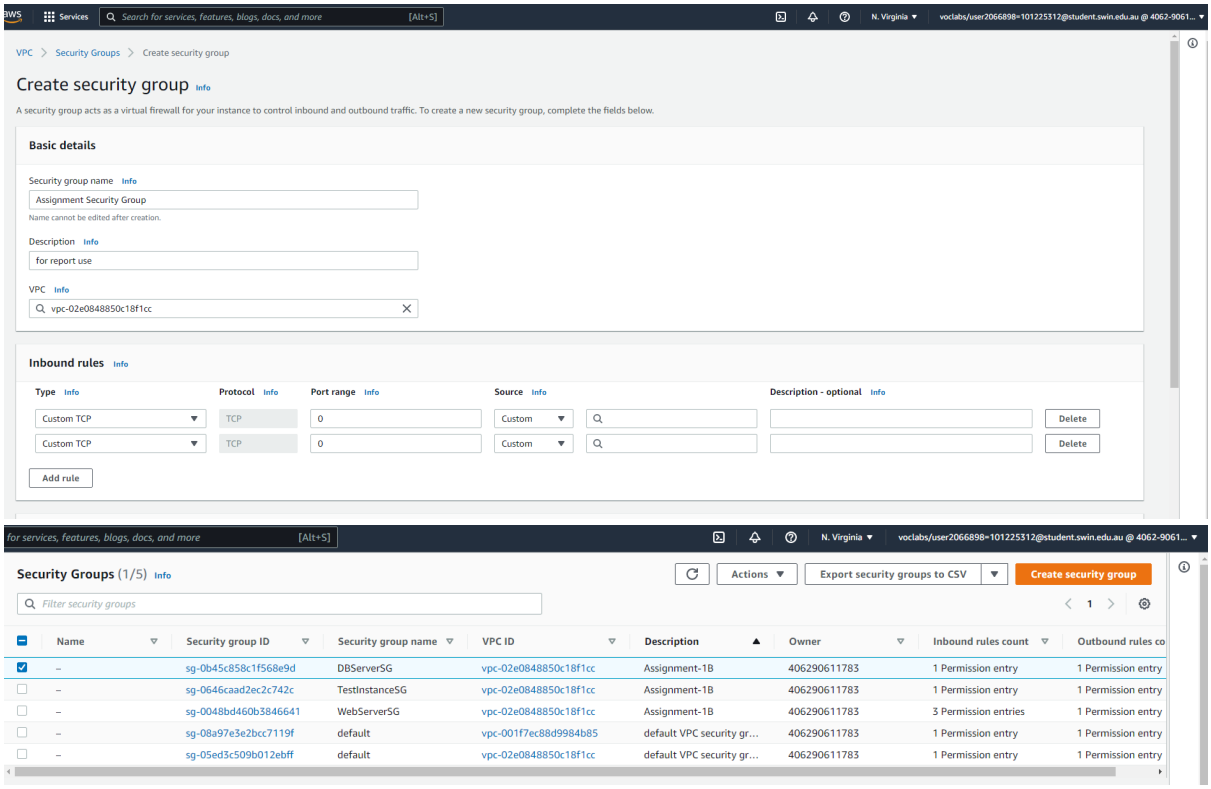
Step by step:

1. Creating VPC



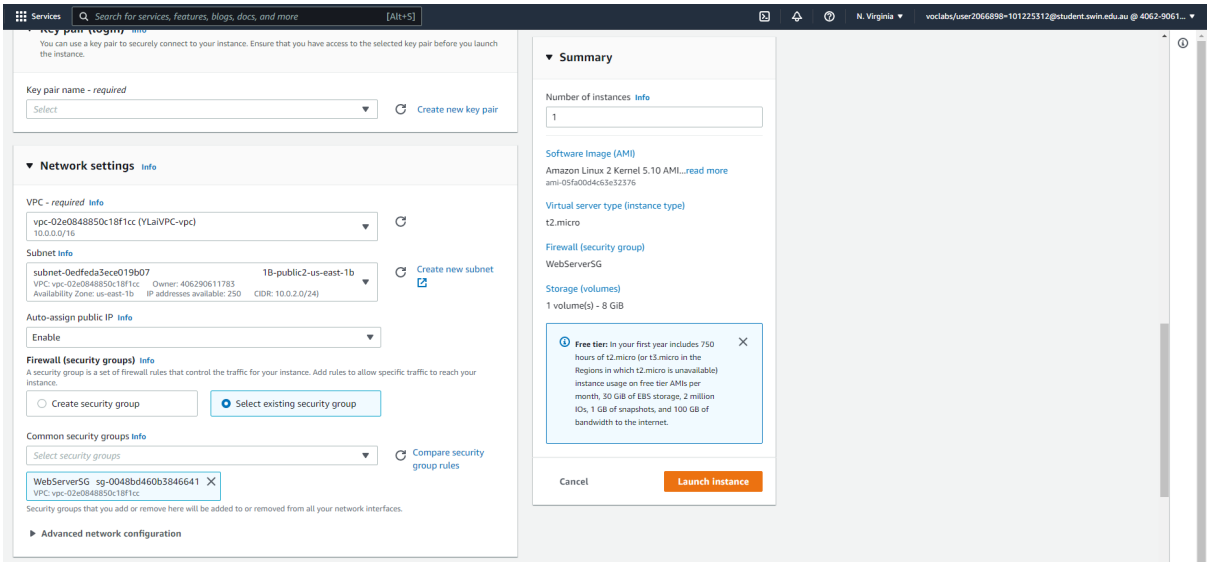
I first created the VPC for the assignment. In the creation page, I also created the 4 subnets (2 public and 2 private) required for the assignment with 2 in each availability zone. The CIDR blocks was also configured in this page.

2. Creating Security Groups

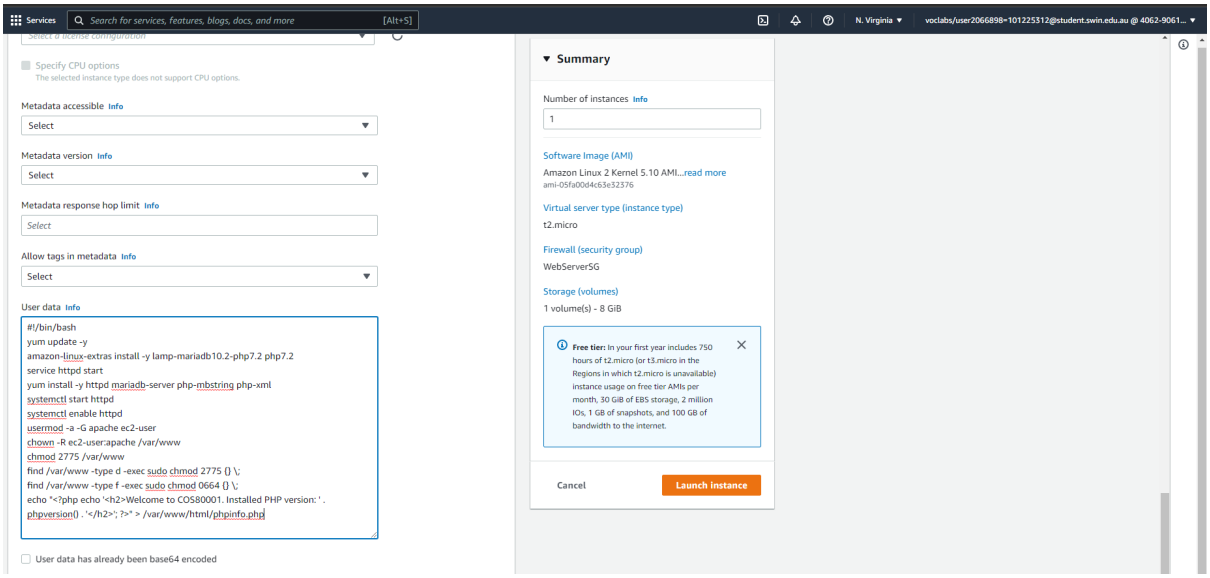


I created 3 security groups to be used in the VPC. The inbound rules are set in the Inbound rules section by choosing the type of connection and the source to match the requirements of the assignment. Security groups limit the connections allowed to go into the object it is assigned to.

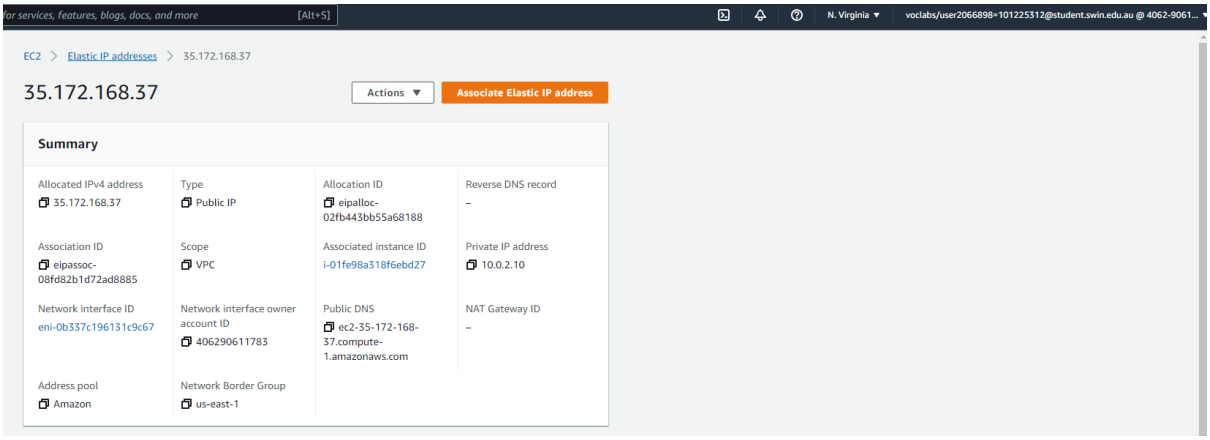
3. Creating EC2 Instances



I created 2 EC2 instances for this assignment (Bastion/WebServer and Test Instance). The VPC, subnet, and security groups are configured according to the requirements of the assignment in the network settings section.

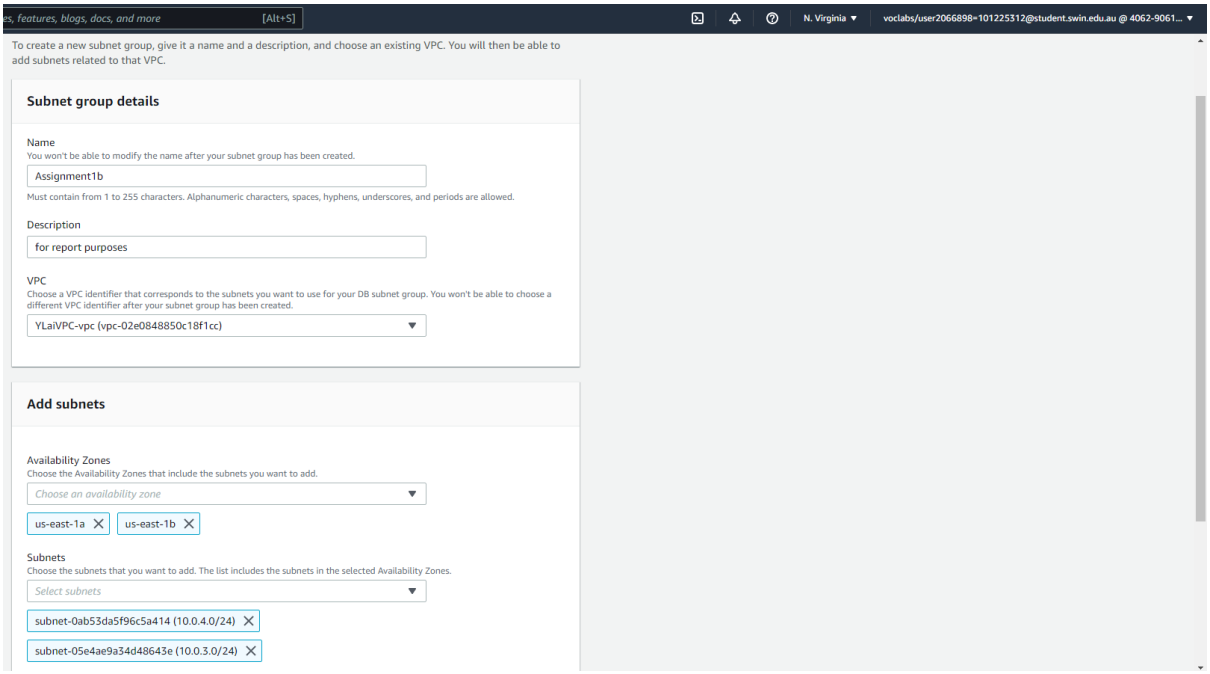


For the Bastion/Web Server instance, Apache http server and php is installed with a script in the User Data located under Advanced Details section.

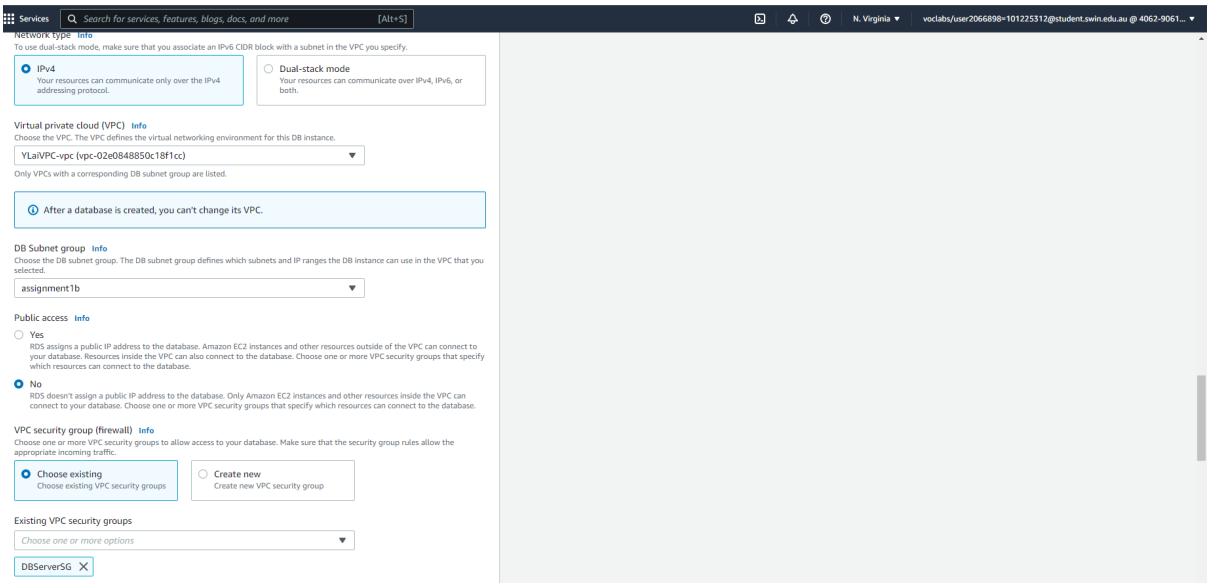


Elastic IP was also created and assigned to the Bastion/Web Server instance to make sure that the public IPv4 address, and DNS remains the same even after stopping the instance.

4. Creating Database Subnet and Database

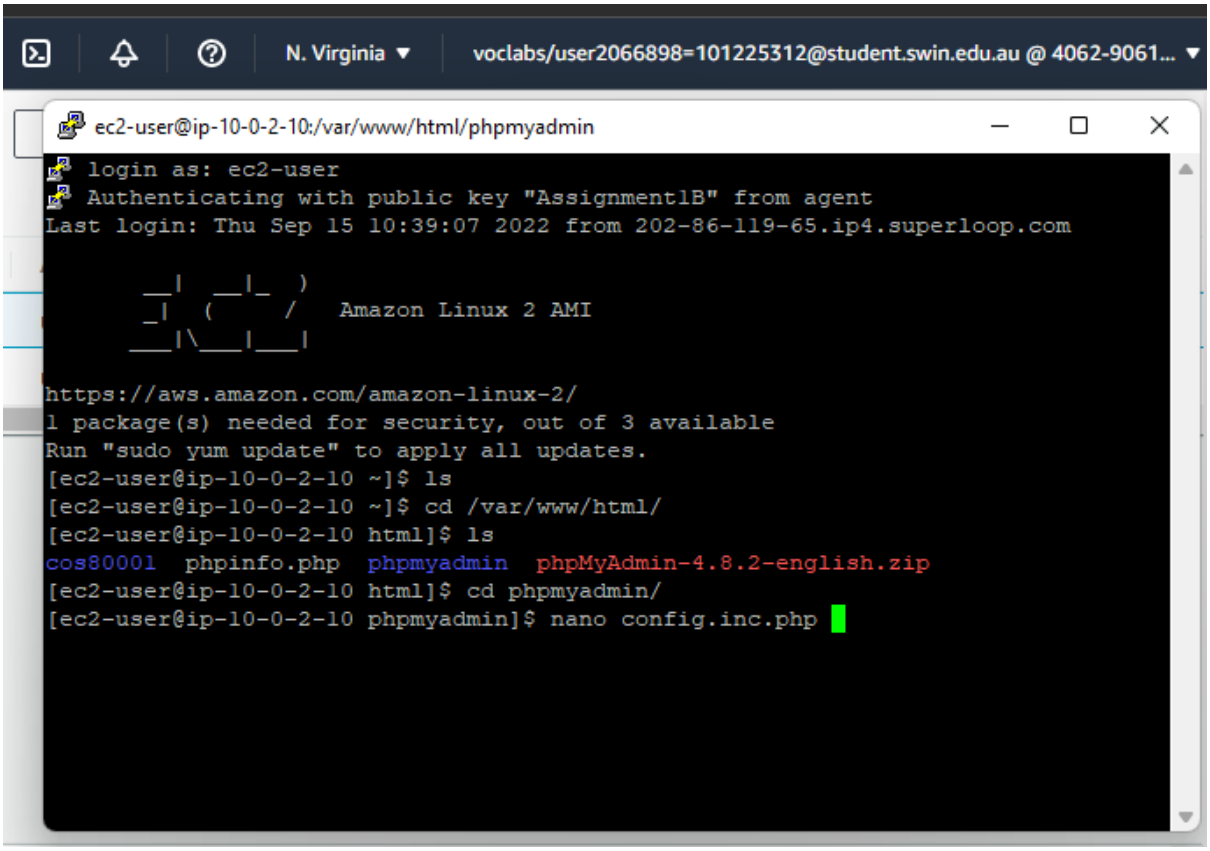


First the subnet group for the database is created for the VPC and subnets.

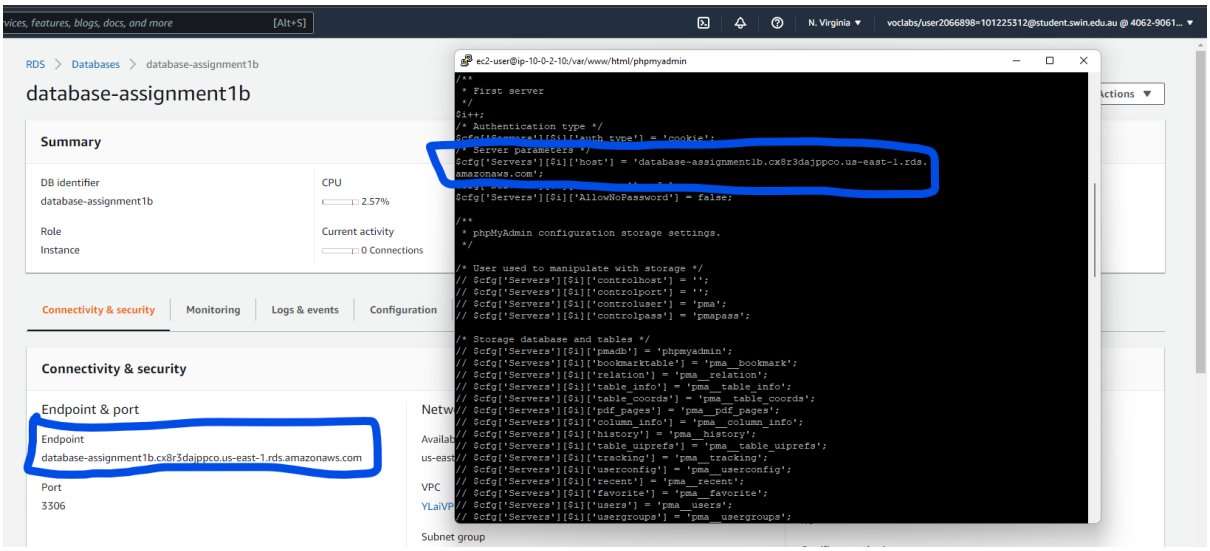


A SQL database is created in the assignment VPC an into the DB subnet group. It is also using the security group DBServerSG made in the first step.

5. Installing phpMyAdmin in Bastion/WebServer

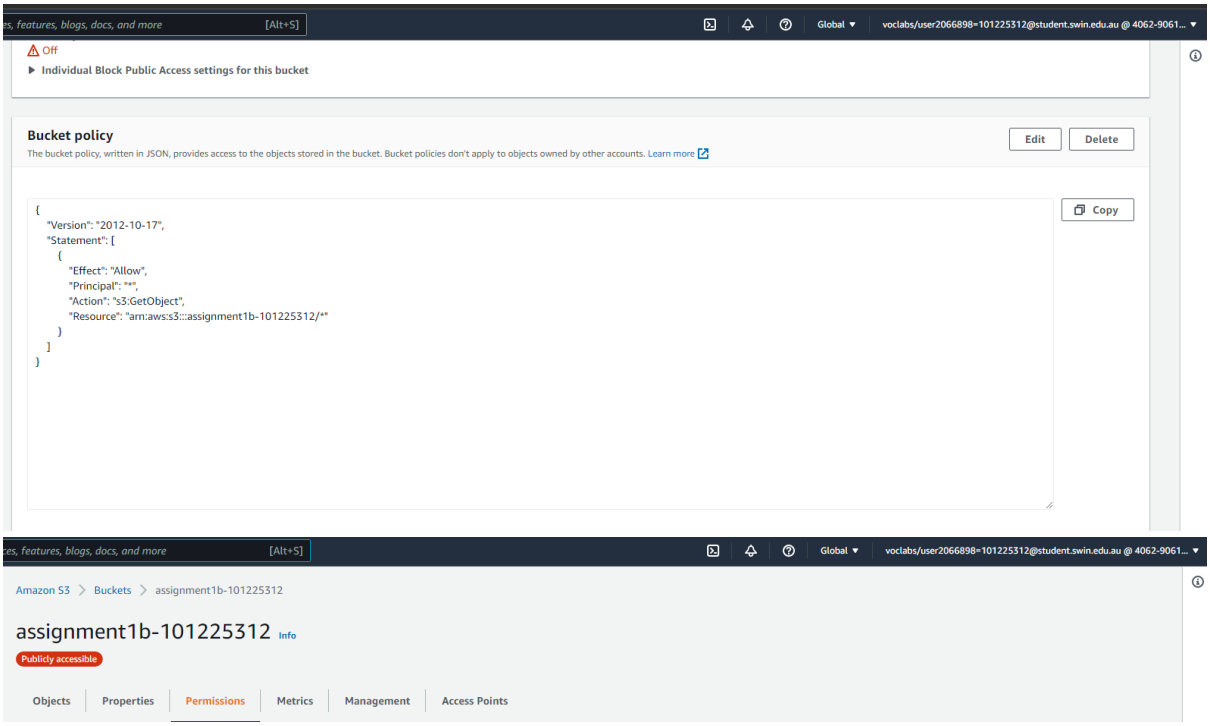


I connected to the Bastion/Web Server instance with putty using the public IPv4 DNS and key pair. Using the link provided, I installed phpMyAdmin to /var/www/html/ directory. Then I had to reconfigure the localhost field with my database endpoint using the text editor "nano".

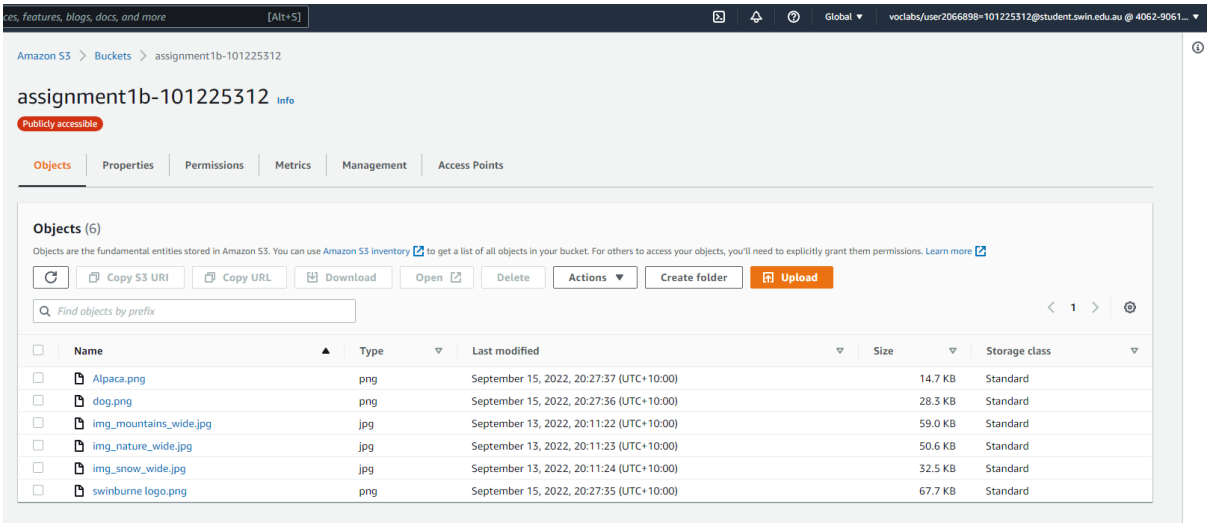


With this, phpMyAdmin webpage accessed with the Bastion/Webserver will be able to use the database created.

6. Creating and configuring S3 bucket

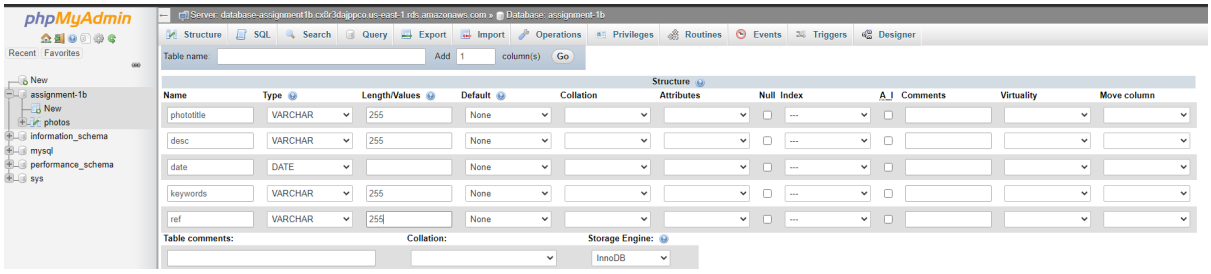


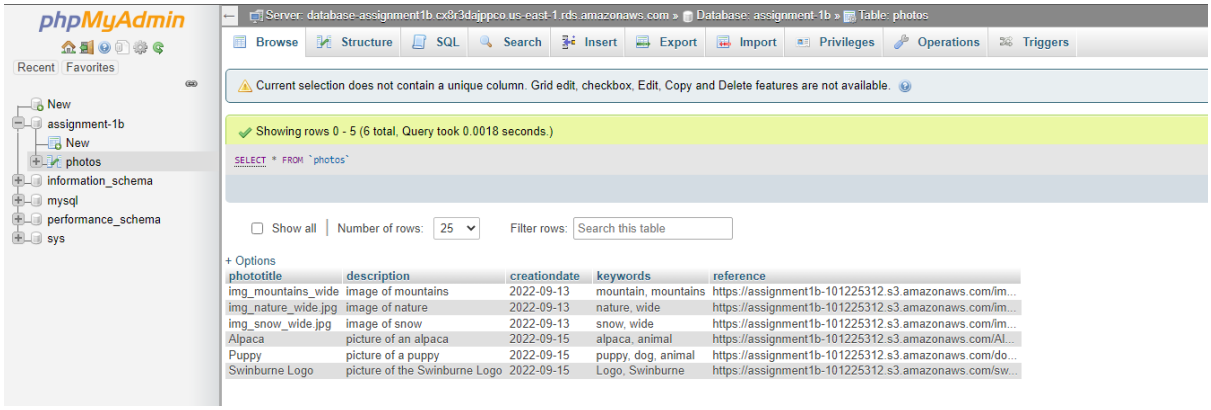
An S3 bucket was created to store the images. I looked up online and placed this statement in the bucket policy to allow public access to the files of the bucket as seen in the red bar that says “Publicly accessible” under the bucket name.



I also uploaded several photos to be used in the database.

7. Creating and populating database





I created a new relational database and a table with the following attributes. After that, I manually populated the table with data using the images I uploaded to the S3 bucket as reference.

8. Importing php source code

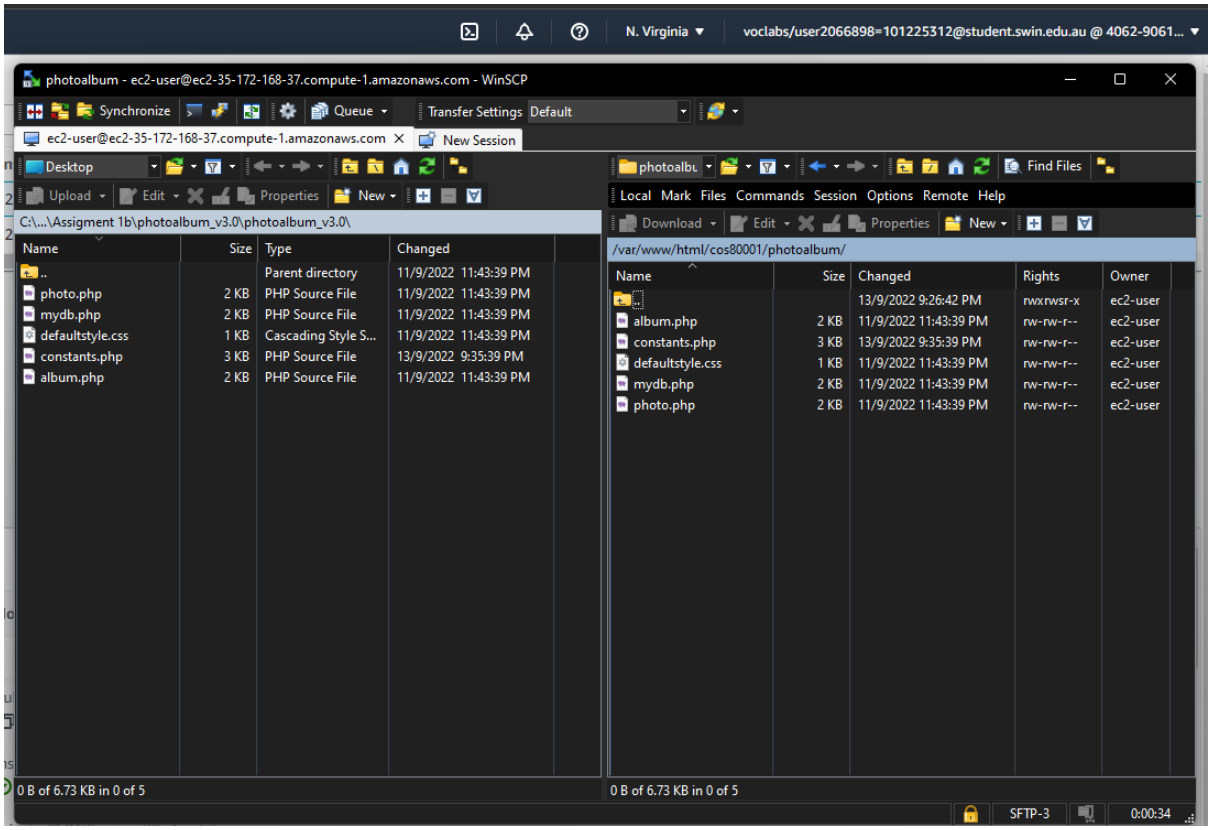
```php
// [ACTION REQUIRED] your full name
define('STUDENT_NAME', 'Lai Yee Fung');
// [ACTION REQUIRED] your Student ID
define('STUDENT_ID', '101225312');
// [ACTION REQUIRED] your tutorial session
define('TUTORIAL_SESSION', 'Wednesday 6:30p.m.');

// [ACTION REQUIRED] name of the S3 bucket that stores images
define('BUCKET_NAME', 'assignment1b-101225312');
// [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 of the RDS instance)
define('DB_NAME', 'assignment-1b');
// [ACTION REQUIRED] endpoint of RDS instance
define('DB_ENDPOINT', 'database-assignment1b.cx8r3dajppco.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', 'assignment1b');

// [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', 'phototitle');
// [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', 'creationdate');
// [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');
?>
```

I edited the php source code to match the fields used in the database as well as the login and password for the database.



Using WinSCP, I imported all the php source files into the Bastion/Web Server instance. I also had to manually create the cos80001 directory. To check if it worked, I just had to go into the album page.

Student name: Lai Yee Fung

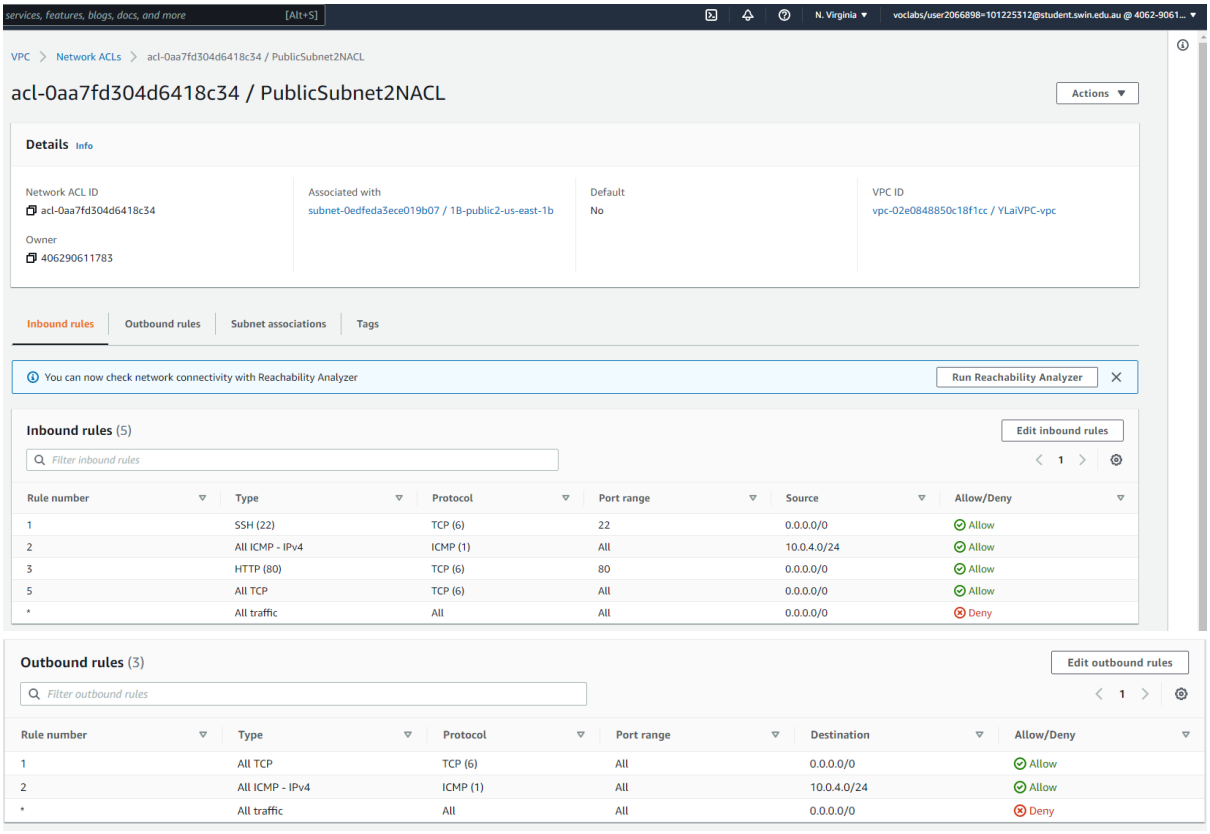
Student ID: 101225312

Tutorial session: Wednesday 6:30p.m.

Uploaded photos:

Photo	Name	Description	Creation date	Keywords
	img_mountains_wide	image of mountains	2022-09-13	mountain, mountains
	img_nature_wide.jpg	image of nature	2022-09-13	nature, wide
	img_snow_wide.jpg	image of snow	2022-09-13	snow, wide
	Alpaca	picture of an alpaca	2022-09-15	alpaca, animal
	Puppy	picture of a puppy	2022-09-15	puppy, dog, animal
	Swinburne Logo	picture of the Swinburne Logo	2022-09-15	Logo, Swinburne

9. Configuring NACL



NACL inbound and outbound rules are set as another layer of security above security groups. An issue I ran into here was with which rules I needed to set for the album page to work. As stated in the requirements, SSH and ICMP is allowed but the other types are not specified.

I tested with the ICMP inbound and outbound to private subnet 2 which worked after several tries by pinging in linux terminal through Putty. As for the other types, I figured it out through trial and error by testing whether the album page would load if I denied access to a certain connection type.

Overall, most of the parts were not too hard for this assignment as it had been done before in previous lab sessions. As for the new parts I had to do some minor searching from previous lab materials or on the web to find the solution.