Assignment 2

COS20019-Cloud Computing Architecture

Name: Yee Fung Lai

Student ID: 101225312

Tutorial Class:
Wednesday 6:30 p.m.

Submission Date:

9/10/2022

URL:http://assignment2loadbalancer-767262559.us-east-1.elb.amazonaws.com/photoalbum/album.php

I. Data Records

The data records currently stored in the relational database service (RDS) and Simple Storage Service (S3) are as shown below.

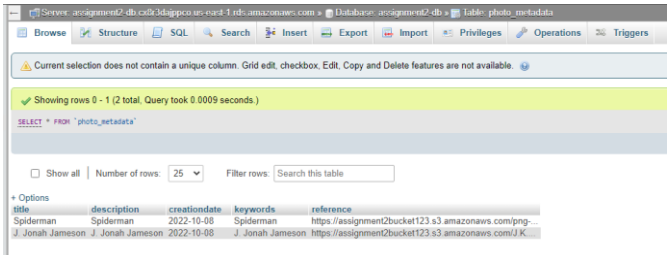


Figure 2 RDS screenshot from PhpMyAdmin

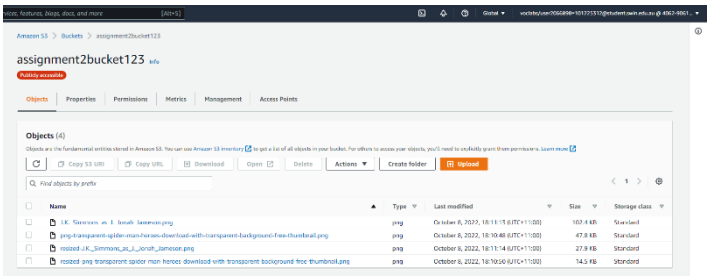
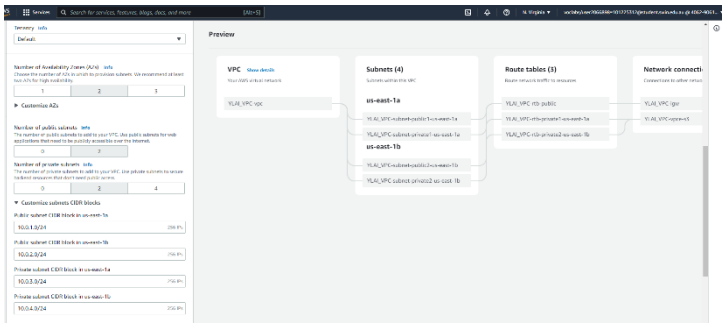


Figure 1 S3 screenshot from AWS Console

II. Deployment Steps

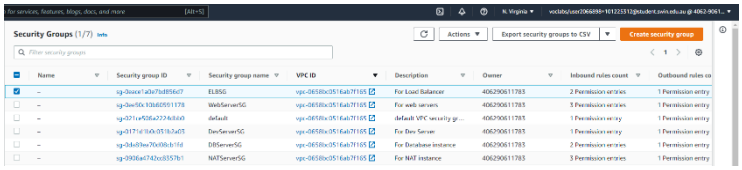
1. *Creating VPC*

The first step is to create the VPC which acts as a virtual network for all the components to be deployed in. The subnets are created along with the VPC in this page. There are 2 public and 2 private subnets split into two availability zones us-east A and B.



2. *Creating Security Groups*

The security groups needed for this network are first created with 'Allow all Traffic' inbound rules to make configuration easier.

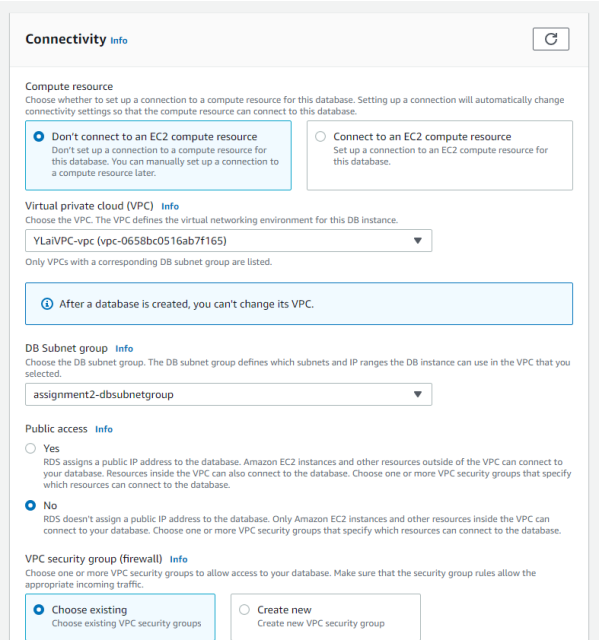


The inbound rules are configured last after all the instances and objects are assigned to make it easier for testing.

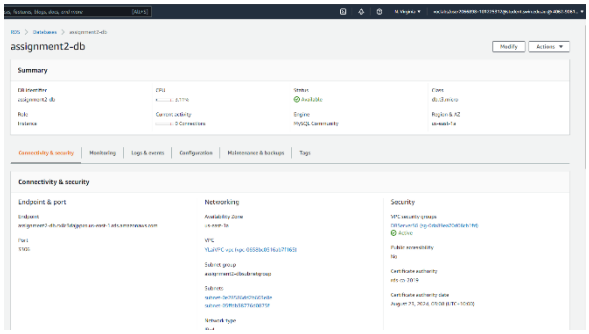
3. *Creating RDS and S3*

The storage services required for this assignment are created first to retrieve the endpoint and bucket names to be used in the configurations.

The RDS runs MySQL similar to assignment1b. It is placed in the subnet group that is in the private subnets of both the availability zones. It is also assigned to the DBServerSG security group.



Once the RDS is created, the endpoint is saved into a notepad to be used for setting up the EC2 instance connections.

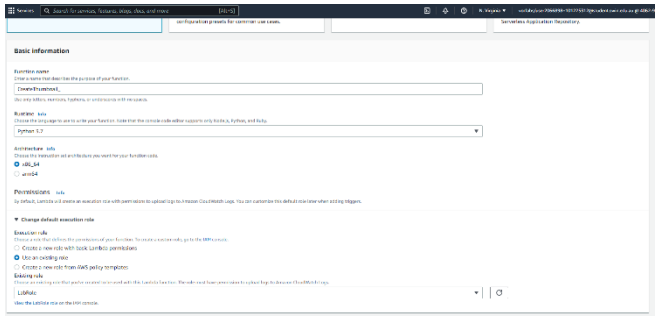


A standard S3 bucket is made as well to store the image files. Standard settings are used for setting up the bucket. The bucket ARN and name is also saved into a notepad.

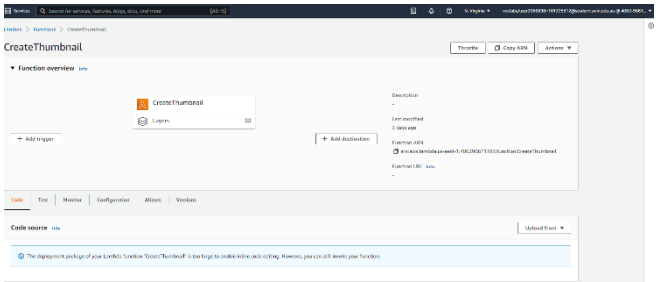
4. *Uploading AWS Lambda code*

Lambda is used to resize the uploaded photos in the RDS to then be uploaded to the S3 bucket

The function name is called CreateThumbnail and it is running on Python 3.7. The IAM role for the Lambda function is LabRole.

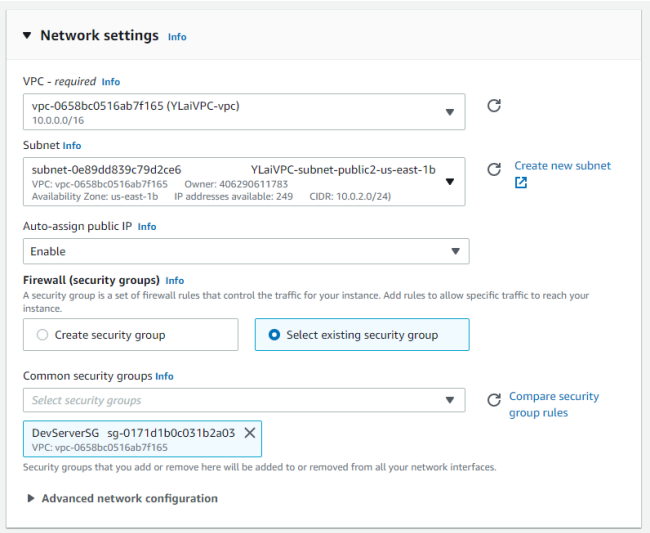


The code is then uploaded with the files provided with the assignment.



5. *Setting up Dev Instance*

The Dev instance is first created as it is used to make an AMI. Which will be used by the Auto Scaling. The configurations used are the similar to the ones set up in lab tutorials. It is assigned to the public subnet in availability zone B in the DevServerSG security group.

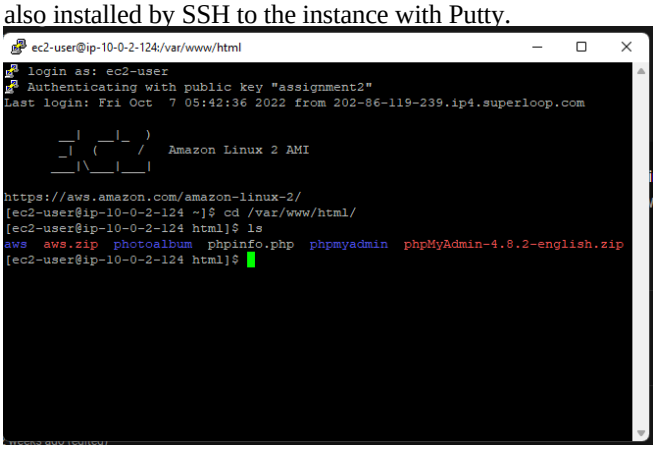


A setup script is also added in the User Data section to install the required components to run PHP.

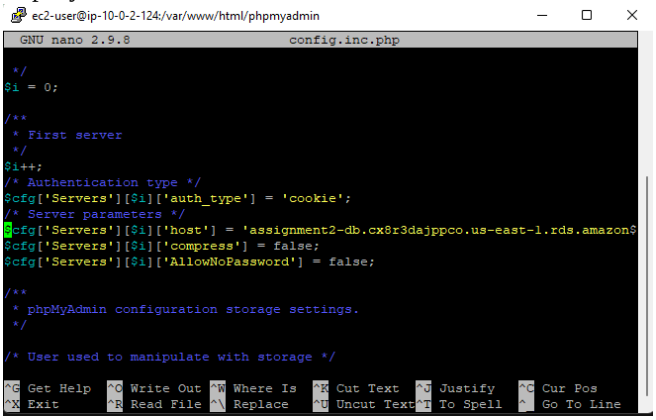
User data Info

```
#!/bin/bash
yum update -y
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
service httpd start
yum install -y httpd mariadb-server php-mbstring php-xml
systemctl start httpd
systemctl enable httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec sudo chmod 2775 {} \;
find /var/www -type f -exec sudo chmod 0664 {} \;
echo "<?php echo '<h2>Welcome to COS80001. Installed PHP version: ' .
phpversion() . '</h2>'; ?>" > /var/www/html/phpinfo.php
```

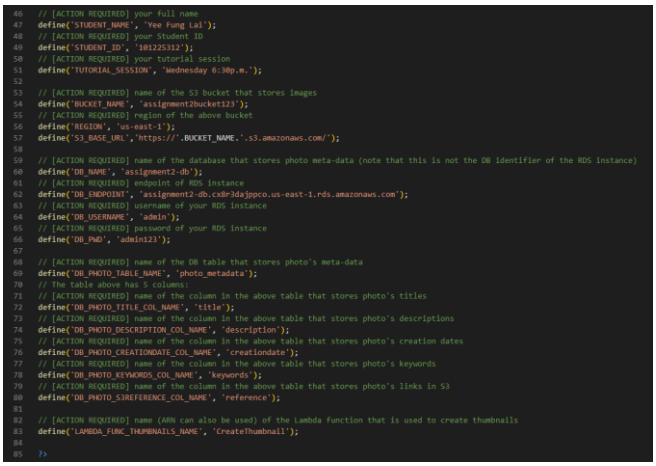
Required files like PhpMyAdmin and AWS SDK PHP is also installed by SSH to the instance with Putty.



The RDS endpoint is added to the config files of PhpMyAdmin to connect to the RDS.

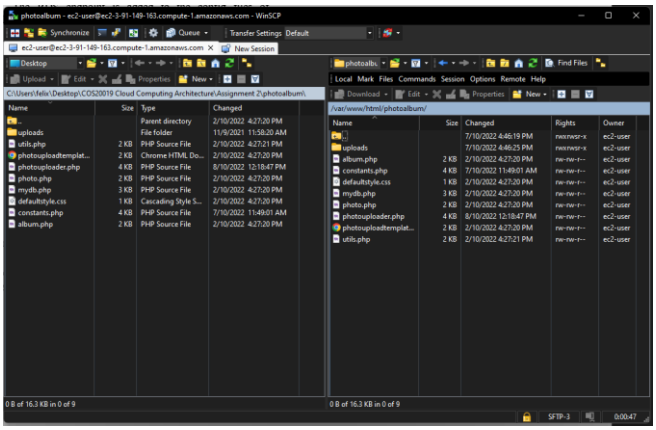


With the information from the previously created RDS, S3 and Lambda, the fields constants.php are filled out.

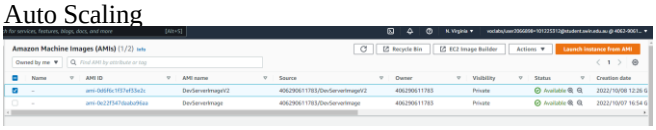


```
// [ACTION REQUIRED] your full name
define('STUDENT_NAME', 'Yee Fung Lai');
// [ACTION REQUIRED] your Student ID
define('STUDENT_ID', '101225312');
// [ACTION REQUIRED] your tutorial session
define('TUTORIAL_SESSION', 'Wednesday 6:30p.m.');

// [ACTION REQUIRED] name of the S3 bucket that stores images
define('BUCKET_NAME', 'assignment2bucket123');
// [ACTION REQUIRED] region of the above bucket
define('REGION', 'us-east-1');
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', 'assignment2-db');
// [ACTION REQUIRED] endpoint of RDS instance
define('DB_ENDPOINT', 'assignment2-db.cx8r3da3jppco.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', 'photo_metadata');
// 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', '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', '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');

// [ACTION REQUIRED] name (ARN can also be used) of the Lambda function that is used to create thumbnails
define('LAMBDA_FUNC_THUMBNAILS_NAME', 'CreateThumbnail');

?>
```

All the php files are then transferred to the Dev Instance with WinSCP. At this point the Dev instance has all the required components.

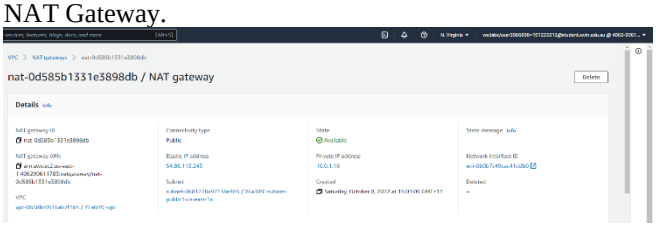


An AMI is made from the Dev instance to be used in the Auto Scaling

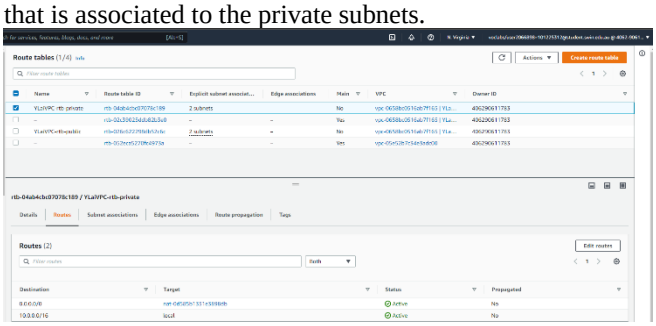


6. *NAT Gateway*

A Network Address Translation (NAT) Gateway to connect instances in private subnet to services outside of the VPC while not allowing the services on the outside connect inside. The configuration is relatively simple as there are not many fields to complete. An elastic IP is also attached to the NAT Gateway.

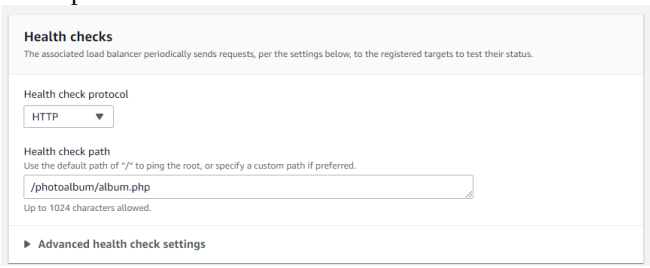


The NAT gateway is then attached to the route table that is associated to the private subnets.

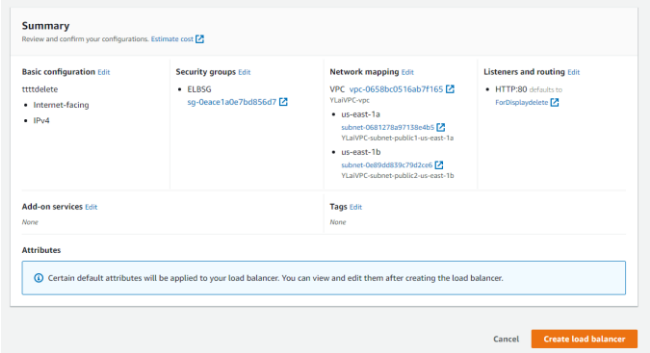


7. *Elastic Load Balancer*

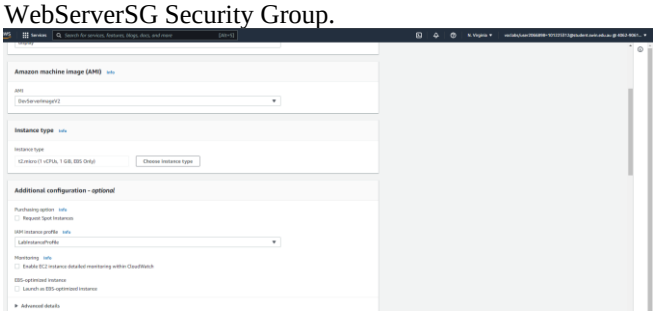
The Elastic Load Balancer is used to balance the load between the available running instances. First the target group is made as it is used in the load balancer. There is not much to configure other than setting the VPC and the health check path.



Then the Load balancer is set up in the public subnets in both availability zones with ELBSG security group

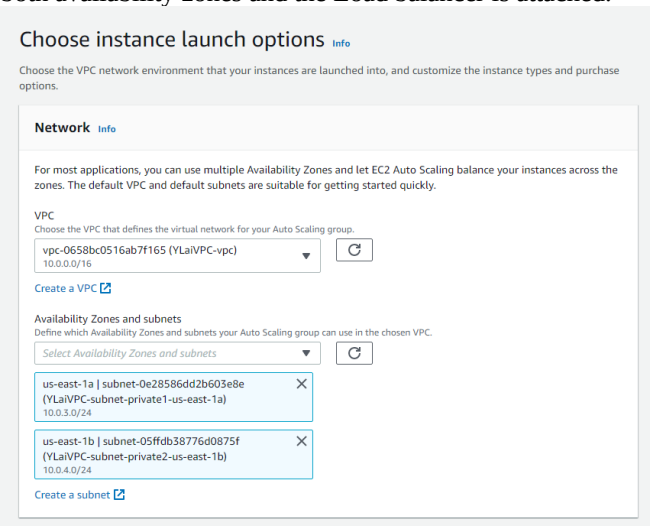


The Launch configuration is made with the AMI from Dev Instance with the IAM instance profile set to LabInstanceProfile. The instances will be launched into WebServerSG Security Group.

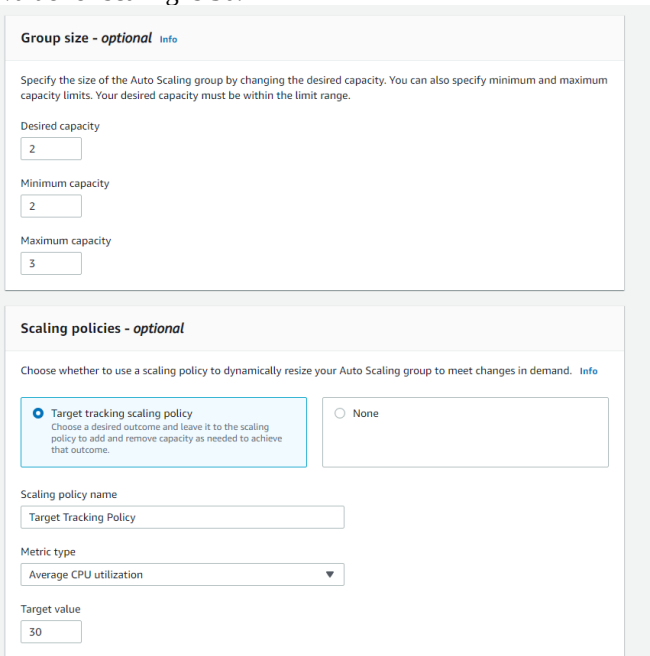


From the Launch Configuration page, an Auto Scaling Group is created from the created Launch Configuration. The instances will be launched into the private subnets of

both availability zones and the Load balancer is attached.



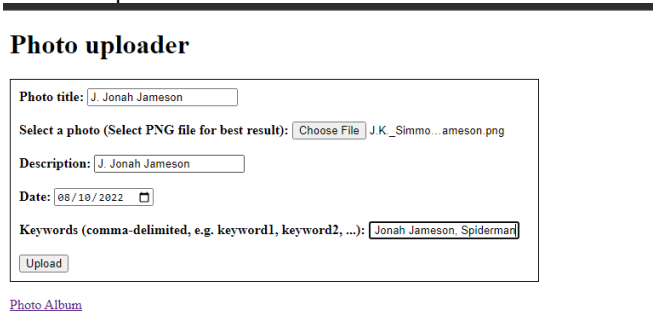
The groups size is configured to 2,2,3 and the target value for scaling is 30.



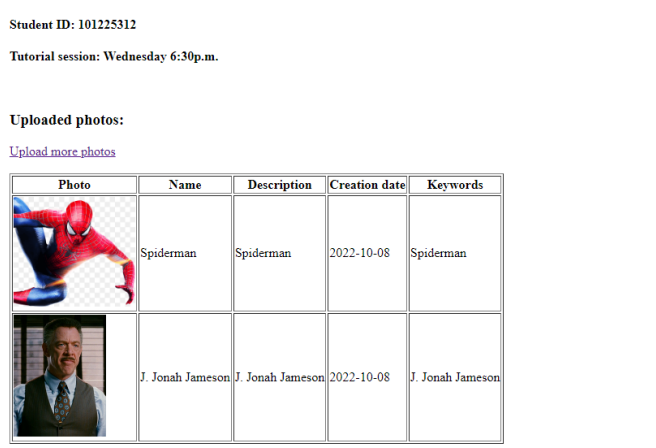
Once created, the Auto Scaling Group will show the current running number of instances.

8. *Usage*

At this point, the website is usable. By using the DNS of the load balancer, the web page can be accessed. With the photo uploader page, some photos and details are inserted and then uploaded.

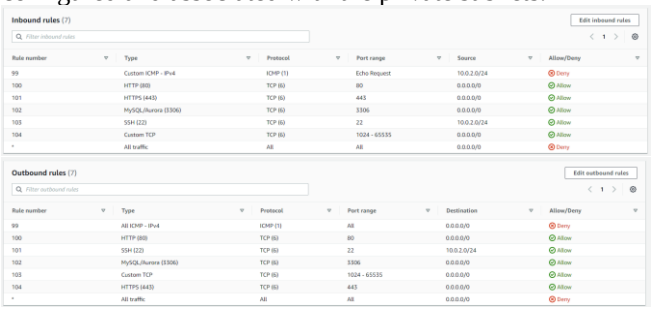


Once uploaded, the page will be redirected to the album.php which will show the details of the uploaded items.



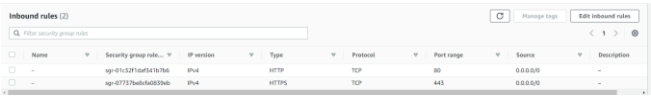
9. *Network ACL and Security Group configurations*

The Network ACL named PrivateSubnetsNACL is configured and associated with the private subnets.

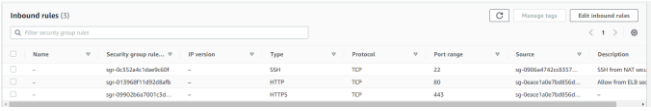


The Security group inbound rules are configured to match the diagram provided in the assignment outline.

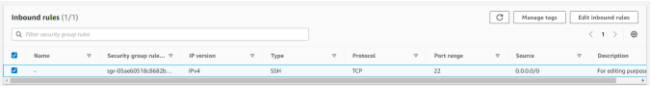
ELBSG



WebServerSG

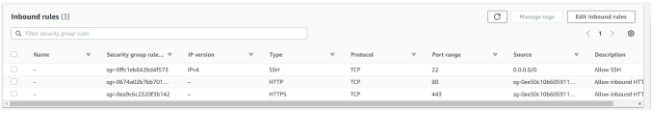


DevServerSG



DBServerSG

NATServerSG



III. Issues Faced

One of the main issues faced throughout the assignment is the implementation of a NAT instance. The NAT instance would work fine with no issues whatsoever. However, every time the AWS console is restarted, an issue would pop up where the private instances could not ping out. After some trial and errors, it was found that the lines:

sudo sysctl -w net.ipv4.ip_forward=1
sudo /sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo service iptables save

Must be rerun to re-establish a connection. If the lines are not run, the instances inside the private subnet will not be able to ping out. Thus, I have decided to opt for a NAT gateway instead to make things easier.

Aside from that, there wasn't any major issues except for some typos causing the whole structure to stop working. Most of the sections can be referenced to previous lab tutorials for guidance.