

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.

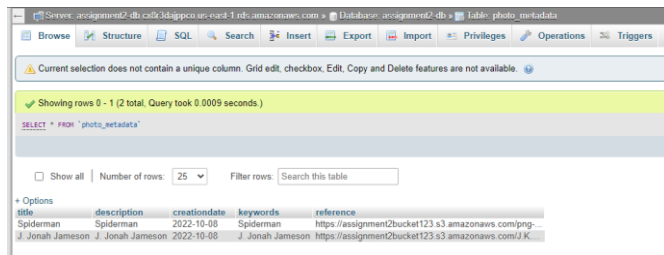


Figure 2 RDS screenshot from PhpMyAdmin

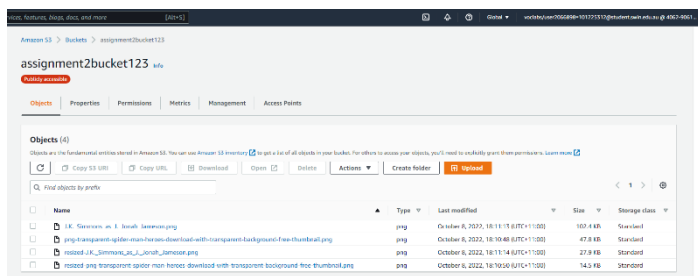
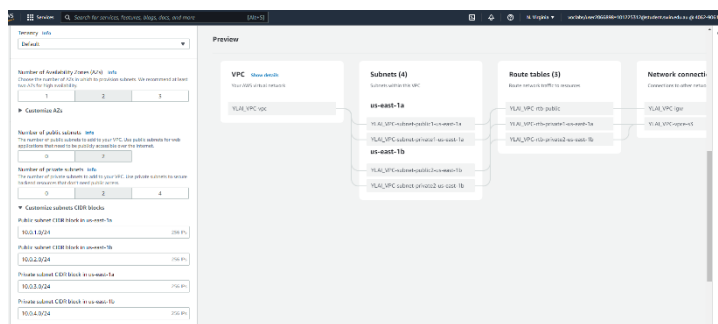


Figure 1 S3 screenshot from AWS Console

II. DEPLOYMENT STEPS

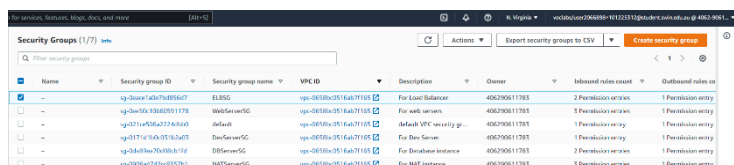
1. Creating VPC

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.

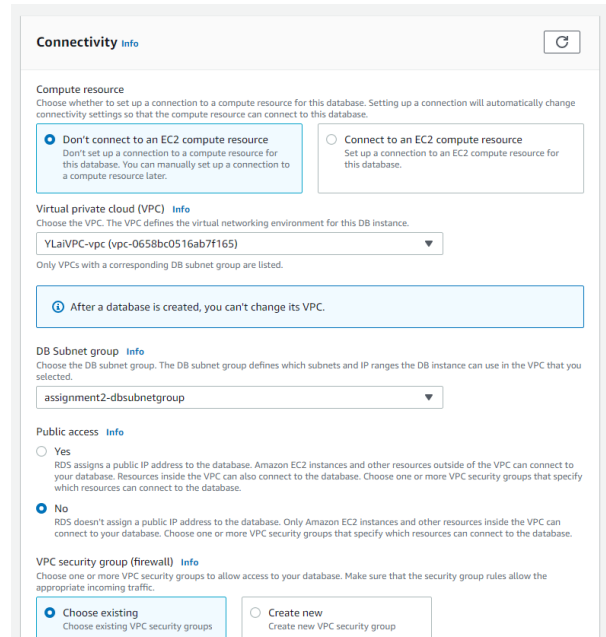


The inbound rules are configured last after all the instances and objects are assigned to make it easier for testing.

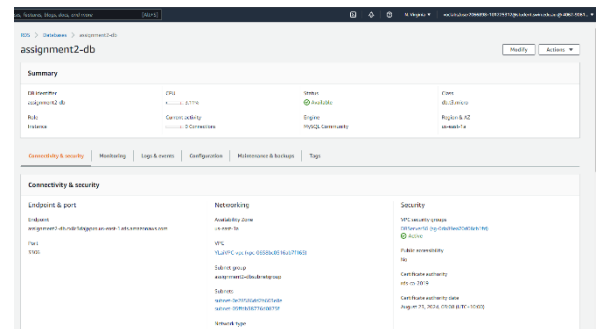
3. Creating RDS and S3

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.

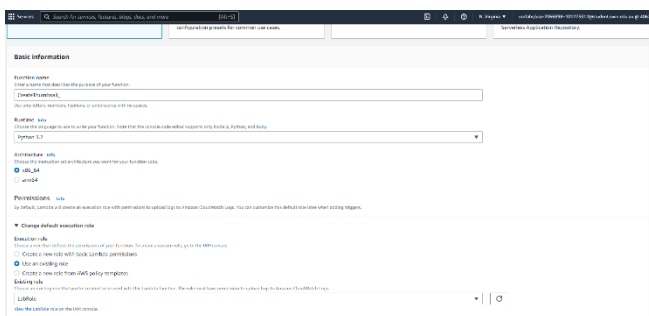


A standard S3 bucket is made as well to store the image files. Standard settings are used for setting up the bucket. The bucket ARN and name is also saved into a notepad.

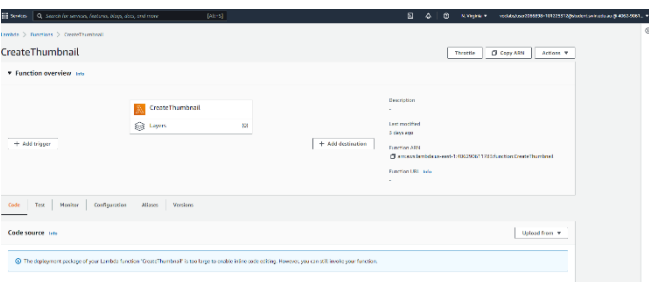
4. Uploading AWS Lambda code

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.

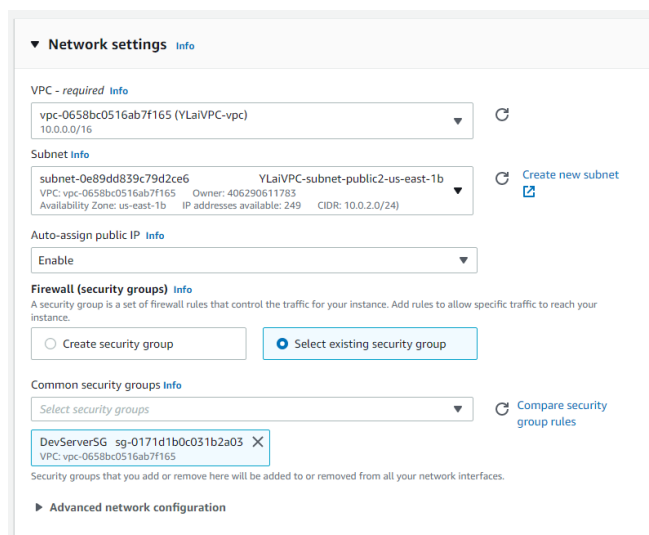


The code is then uploaded with the files provided with the assignment.



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.

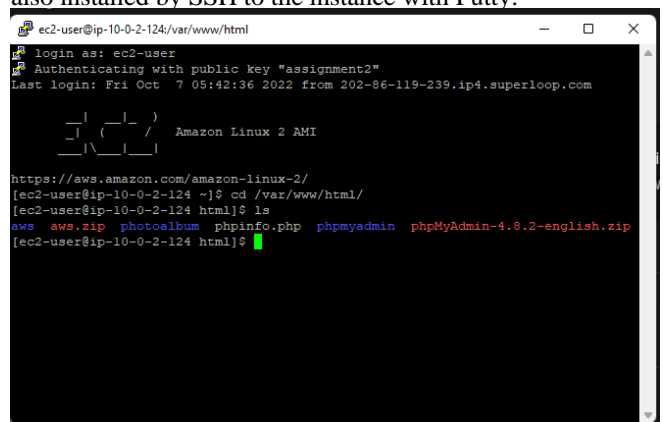


A setup script is also added in the User Data section to install the required components to run PHP.

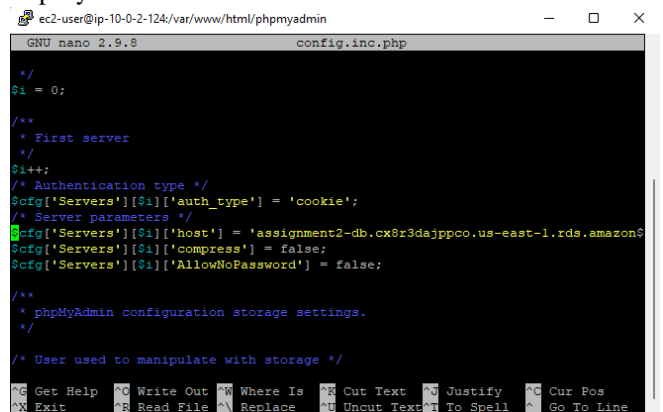
User data Info

```
#!/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.



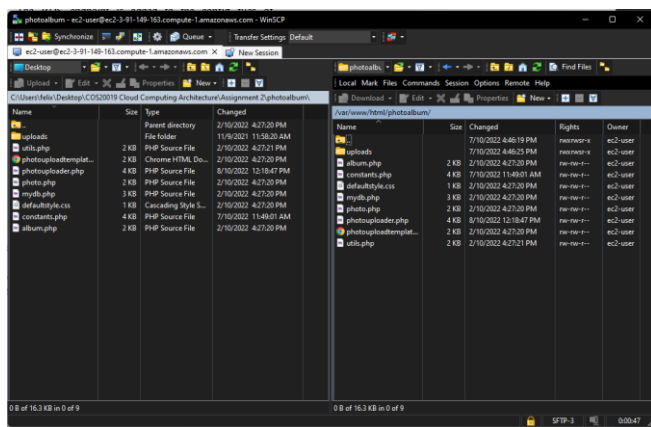
With the information from the previously created RDS, S3 and Lambda, the fields constants.php are filled out.

```

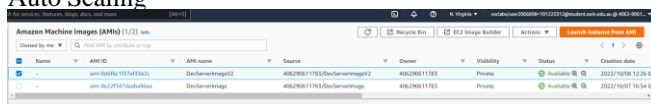
46 // [ACTION REQUIRED] your full name
47 define("STUDENT_NAME", "Yee Fung Lai");
48 // [ACTION REQUIRED] id of the instance
49 define("STUDENT_ID", "18122512");
50 // [ACTION REQUIRED] your tutorial session
51 define("TUTORIAL_SESSION", "Wednesday 6:30p.m.");
52
53 // [ACTION REQUIRED] name of the S3 bucket that stores images
54 define("BUCKET_NAME", "assignment2ucket123");
55 // [ACTION REQUIRED] region of the above bucket
56 define("REGION", "us-east-1");
57 define("S3_BASE_URL", "https://"+BUCKET_NAME+".s3.amazonaws.com/");
58
59 // [ACTION REQUIRED] name of the database that stores photo meta-data (note that this is not the DB identifier of the RDS instance)
60 define("DB_NAME", "assignment2-db");
61 // [ACTION REQUIRED] endpoint of RDS instance
62 define("DB_ENDPOINT", "assignment2-db.cdr3da3pccs.us-east-1-rds.amazonaws.com");
63 // [ACTION REQUIRED] username of your RDS instance
64 define("DB_USERNAME", "admin");
65 // [ACTION REQUIRED] password of your RDS instance
66 define("DB_PWD", "admin123");
67
68 // [ACTION REQUIRED] name of the DB table that stores photo's meta-data
69 define("DB_PHOTO_TABLE_NAME", "photo_metadata");
70 // The table above has 5 columns:
71 // [ACTION REQUIRED] name of the column in the above table that stores photo's titles
72 define("DB_PHOTO_TITLE_COL_NAME", "title");
73 // [ACTION REQUIRED] name of the column in the above table that stores photo's descriptions
74 define("DB_PHOTO_DESCRIPTION_COL_NAME", "description");
75 // [ACTION REQUIRED] name of the column in the above table that stores photo's creation dates
76 define("DB_PHOTO_CREATIONDATE_COL_NAME", "creationdate");
77 // [ACTION REQUIRED] name of the column in the above table that stores photo's keywords
78 define("DB_PHOTO_KEYWORDS_COL_NAME", "keywords");
79 // [ACTION REQUIRED] name of the column in the above table that stores photo's links in S3
80 define("DB_PHOTO_SREFERENCE_COL_NAME", "reference");
81
82 // [ACTION REQUIRED] name (ARN can also be used) of the Lambda Function that is used to create thumbnails
83 define("LAMBDA_FUNC_THUMBNAILS_NAME", "CreateThumbnail");
84
85 }

```

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

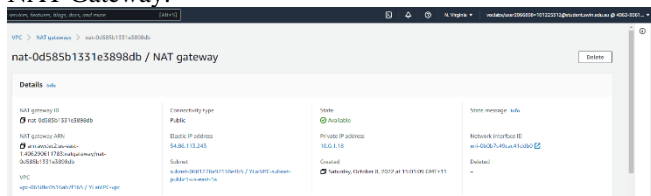


An AMI is made from the Dev instance to be used in the Auto Scaling

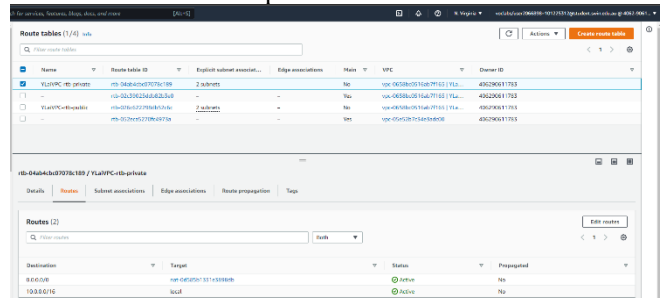


6. NAT Gateway

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.

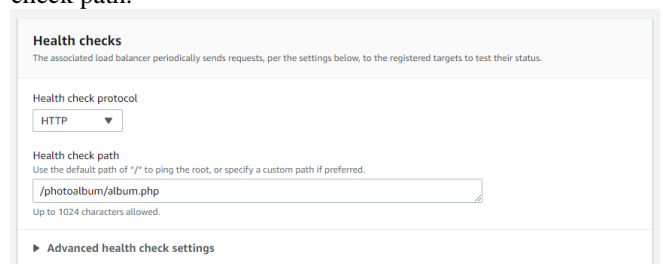


The NAT gateway is then attached to the route table that is associated to the private subnets.

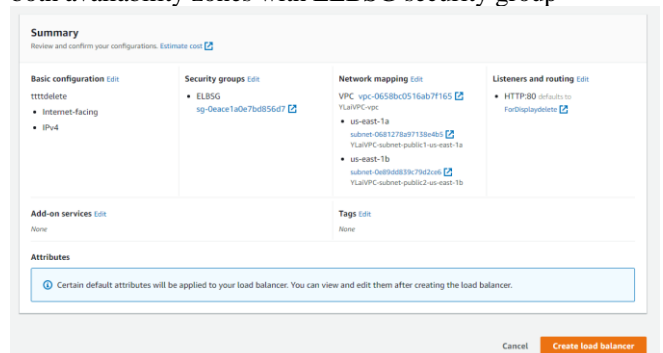


7. Elastic Load Balancer

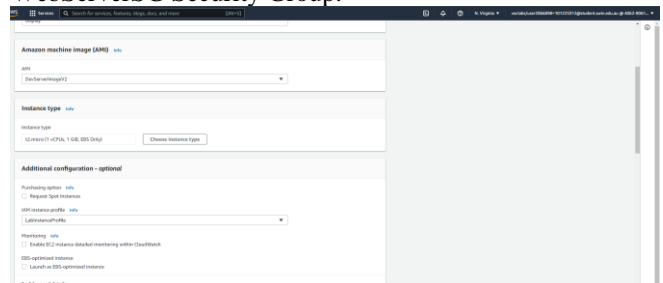
The Elastic Load Balancer is used to balance the load between the available running instances. First the target group is made as it is used in the load balancer. There is not much to configure other than setting the VPC and the health check path.



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.

Choose instance launch options [Info](#)

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0658bc0516ab7f165 (YLaIVPC-vpc)
10.0.0.0/16

[Create a VPC](#)

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

us-east-1a | subnet-0e28586dd2b603e8e
(YLaIVPC-subnet-private1-us-east-1a)
10.0.3.0/24

us-east-1b | subnet-05ffdb38776d0875f
(YLaIVPC-subnet-private2-us-east-1b)
10.0.4.0/24

[Create a subnet](#)

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

Group size - optional [Info](#)

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity
2

Minimum capacity
2

Maximum capacity
3

Scaling policies - optional

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

☒ Target tracking scaling policy
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

☐ None

Scaling policy name
Target Tracking Policy

Metric type
Average CPU utilization

Target value
30

Auto Scaling groups							
Auto Scaling groups (1) Info							
	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max
	Assignment2_ASG	Assignment2_LaunchConfig	2	-	2	2	3

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.

Photo uploader

Photo title:

Select a photo (Select PNG file for best result): J.K_Simmo...ameson.png

Description:

Date:

Keywords (comma-delimited, e.g. keyword1, keyword2, ...):

[Photo Album](#)

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



Student name: Yee Fung Lai

Student ID: 101225312

Tutorial session: Wednesday 6:30p.m.

Uploaded photos:

[Upload more photos](#)

Photo	Name	Description	Creation date	Keywords
	Spiderman	Spiderman	2022-10-08	Spiderman
	J. Jonah Jameson	J. Jonah Jameson	2022-10-08	J. Jonah Jameson

9. Network ACL and Security Group configurations

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

configured and associated with the private subnets.

Inbound rules (7)

Filter inbound rules

< 1 > ?

Rule number	Type	Protocol	Port range	Source	Allow/Deny
99	Custom ICMP - IPv4	ICMP (1)	Echo Request	10.0.2.0/24	Deny
100	HTTP (80)	TCP (8)	80	0.0.0.0/0	Allow
101	HTTPS (443)	TCP (8)	443	0.0.0.0/0	Allow
102	MySQL/Amazon (3306)	TCP (8)	3306	0.0.0.0/0	Allow
103	SSH (22)	TCP (8)	22	10.0.2.0/24	Allow
104	Custom TCP	TCP (8)	1024 - 65535	0.0.0.0/0	Deny
+	All traffic	All	All	0.0.0.0/0	Deny

Outbound rules (7)

Filter outbound rules






< 1 > ?

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
99	All ICMP - IPv4	ICMP (1)	All	0.0.0.0/0	Deny
100	HTTP (80)	TCP (8)	80	0.0.0.0/0	Allow
101	SSH (22)	TCP (8)	22	10.0.2.0/24	Allow
102	MySQL/Amazon (3306)	TCP (8)	3306	0.0.0.0/0	Allow
103	Custom TCP	TCP (8)	1024 - 65535	0.0.0.0/0	Allow
104	HTTPS (443)	TCP (8)	443	0.0.0.0/0	Allow
+	All traffic	All	All	0.0.0.0/0	Deny

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

Inbound rules (2)										Manage tags	Edit inbound rules
<input type="text" value="Filter security group rules"/>											
Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description				
-	sg-01321a1b4e1b1b1b1	IPv4	HTTP	TCP	80	0.0.0.0/0					
-	sg-01775b6dc083db0	IPv4	HTTPS	TCP	443	0.0.0.0/0					

WebServerSG

Inbound rules (3)									Manage tags	Edit inbound rules
<input type="text" value="Filter security group rules"/>								   		
<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description		
<input type="checkbox"/>	-	sg-01321a1b4e1b1b1b1	-	SSH	TCP	22	sg-0086a4742a3b337...	SSH from NAT ser...		
<input type="checkbox"/>	-	sg-013960f11b028af8b	-	HTTP	TCP	80	sg-0086a4742a3b337...	Allow from ELB se...		
<input type="checkbox"/>	-	sg-00902b6a7001c34...	-	HTTPS	TCP	443	sg-0086a4742a3b337...			

DevServerSG

Inbound rules (1/1)								
Filter security group rules								
Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description	
-	sgp-01a6d051b6a62b...	IPv4	SSH	TCP	22	0.0.0.0/0	For editing purpose	

DBServerSG

Inbound rules (2)								
Filter security group rules								
Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description	
-	sgp-00ee096ca5c31577	-	MySQL/Aurora	TCP	3306	sg-0171a750a057b62a...	For editing purpose	
-	sgp-00a4f7037790176296	-	MySQL/Aurora	TCP	3306	sg-0ee05a10b6d05911...	Allow from Internet	

NATServerSG

Inbound rules (3)								
Filter security group rules								
Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description	
-	sgp-08f1a6423b76a05753	IPv4	SSH	TCP	22	0.0.0.0/0	Allow SSH	
-	sgp-08f746d2b76b701...	-	HTTP	TCP	80	sg-0ee05a10b6d05911...	Allow inbound HT	
-	sgp-0b0a94a2320f3b142	-	HTTPS	TCP	443	sg-0ee05a10b6d05911...	Allow inbound HT	

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.