COS20019 – Assignment 2

Developing a highly available Photo Album website

Matthew Gibbons, 105335503, 11/5/25
Tutorial Class Monday 4:30 PM

I INTRODUCTION

Throughout Assignment 2, I extended upon the infrastructure I developed in Assignment 1b via implementing lambda functions and IAM roles. I also developed two web instances through the use of an auto-scaling group and a launch template containing an AMI that I created within a Dev instance. Furthermore I added security features by restricting S3 access using a security policy, used security groups for all instances/ELB'S, and controlled traffic through the use of NACL's.

Deployed Website:

http://photoloadbalancer-164214825.us-east-1.elb.amazonaws.com/photoalbum/album.php

II. Creating VPC

Firstly, I created the Virtual Private Cloud for the infrastructure and assigned its name as “”, region, and two availability zones, “Zone A” and “Zone B”, with a public and a private subnet within each of the zones. I then created a route table and an internet gateway and connected them.

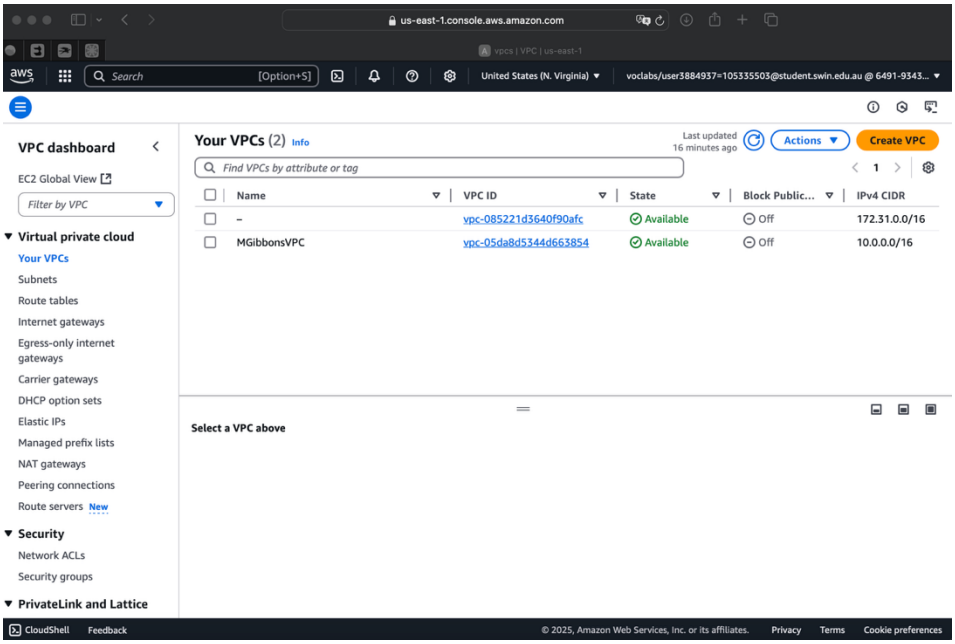

Figure 2 - VPC Creation

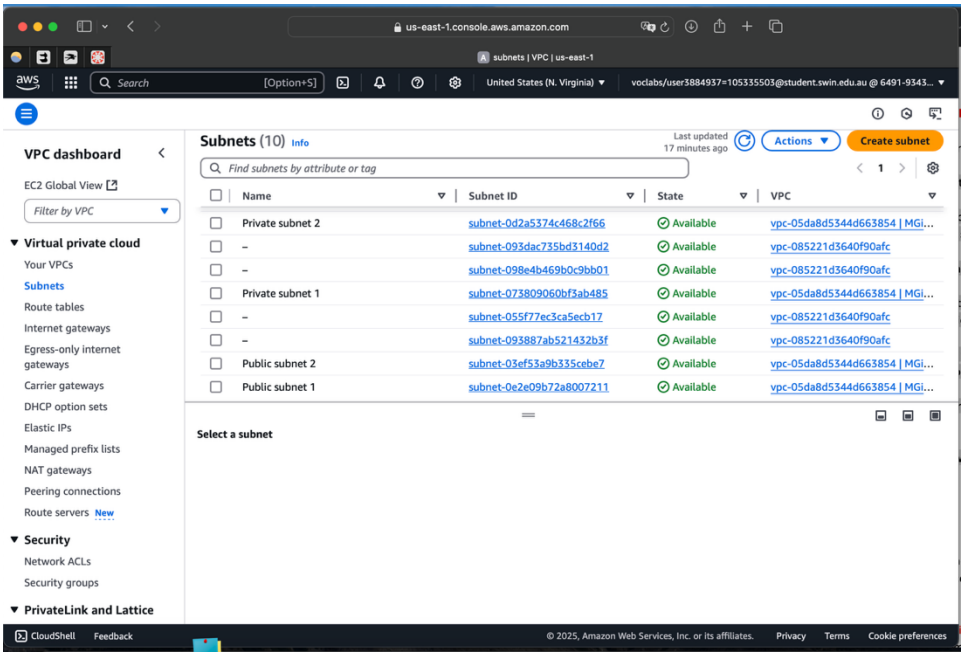

Figure 1 - All Subnets Created within VPC

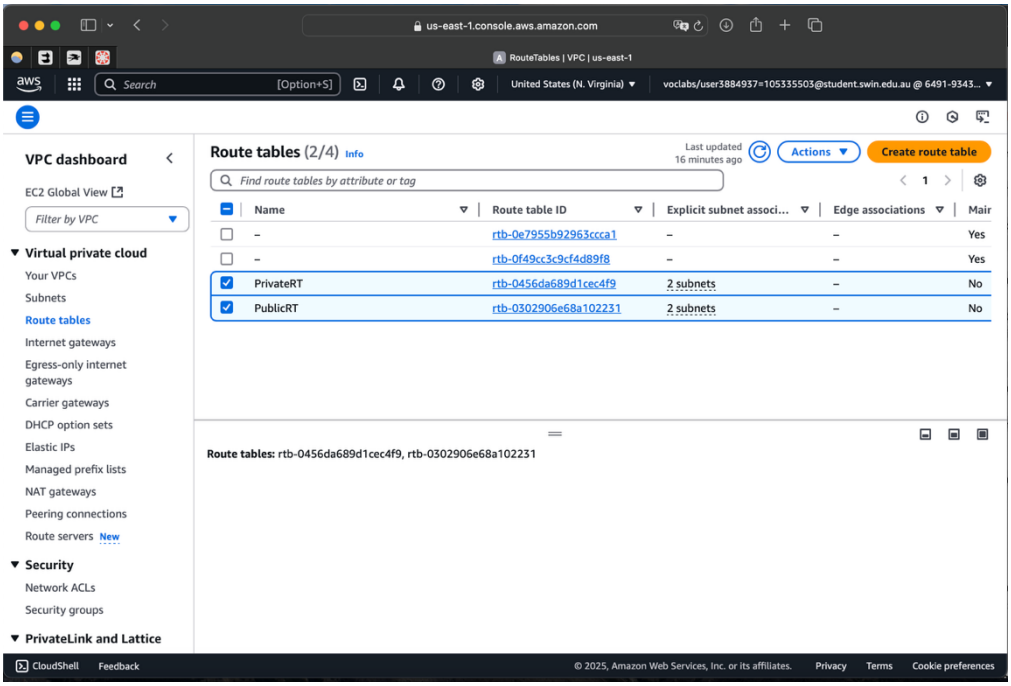

Figure 3 - Route Tables For Subnets

III. Creating NAT Gateway

I then created a NAT Gateway in the public subnet in Availability Zone 1a, which will be used later in order to enable the web instances in the private subnets to access the internet.

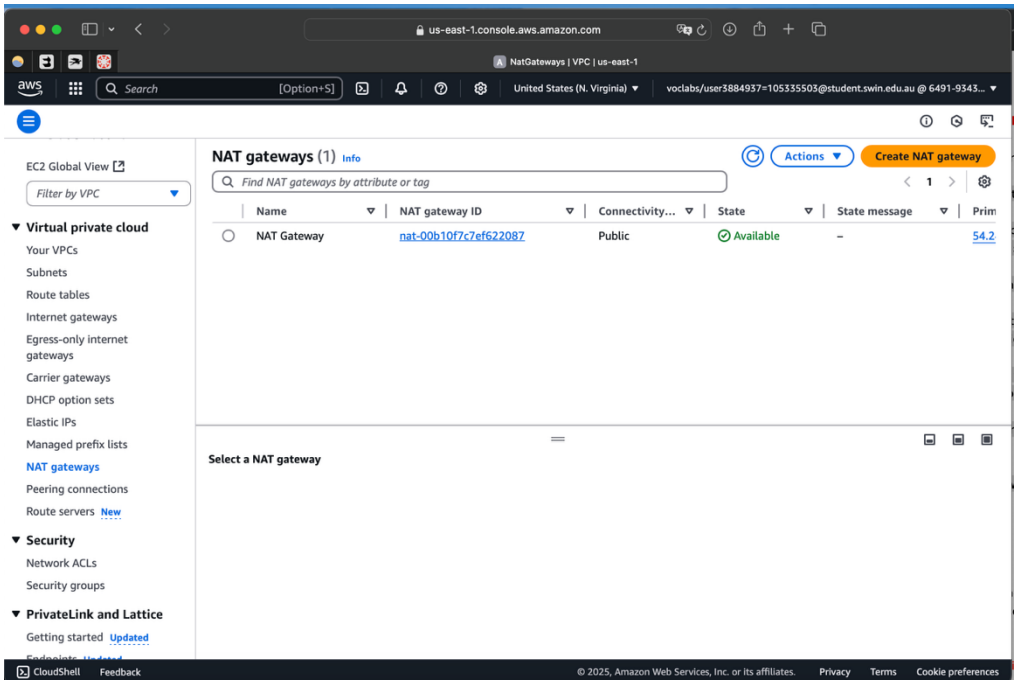

Figure 4 - The NAT Gateway

IV. Creating Elastic Load Balancer

I then created the Elastic Load Balancer and configured it to run health checks on the "/photoalbum/album.php" pathway in order to ensure the ELB is running health checks on all instances.

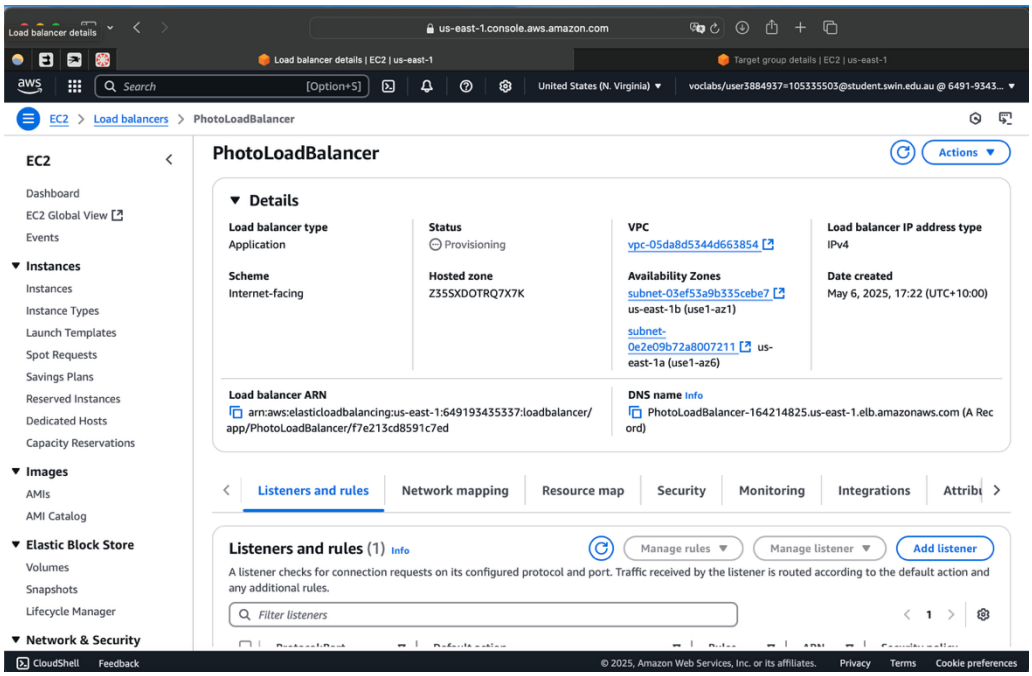

Figure 5 - Elastic Load Balancer for the PhotoAlbum site

V. Configuring S3 Bucket

I then utilised the same S3 bucket that was created in assignment 1b, however, I altered the policies of the bucket in order to only allow access from the web server and to deny access to all other HTTP traffic.

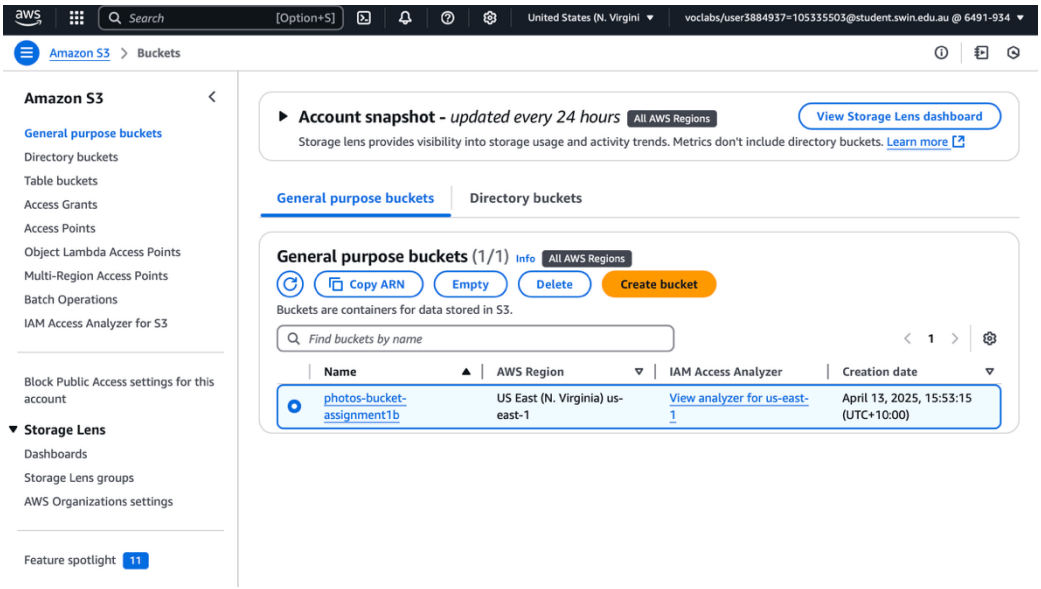

Figure 6 - S3 Bucket from assignemnt 1b

VI. Creating the Dev Instance

Next, I created the Dev instance, which will later be used to create the AMI for the Web instances. I also SSH into the dev instance and downloaded AWS SDK for PHP, Apache, PHP, and Mysql.

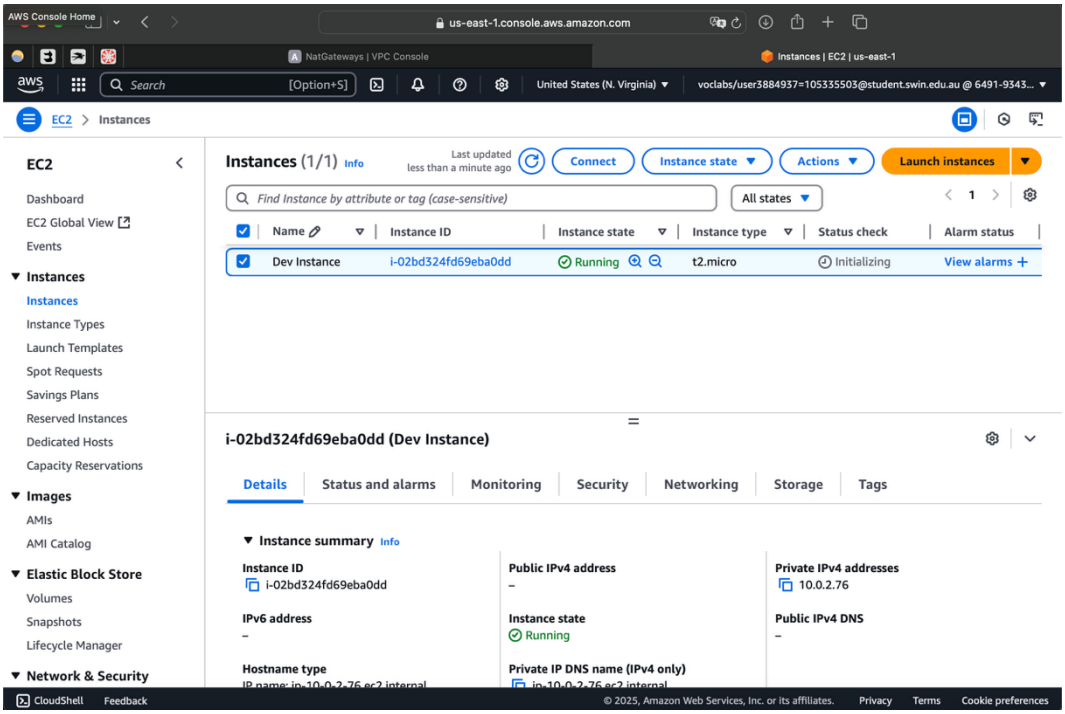

Figure 8 - Dev Instance I Created

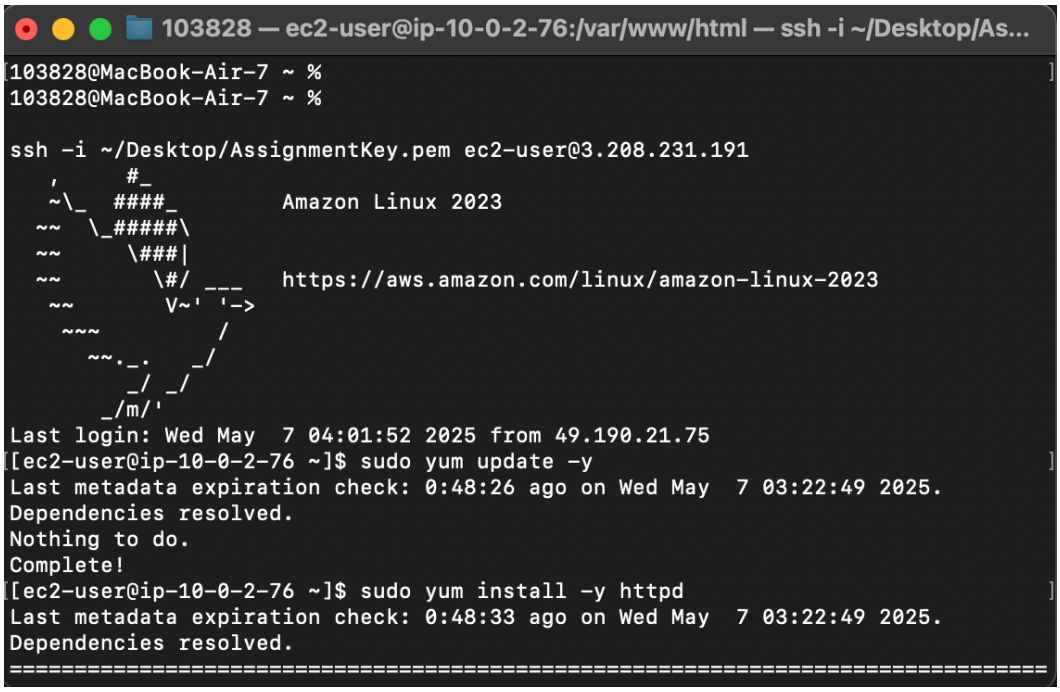

Figure 7 - SSH into Dev Server

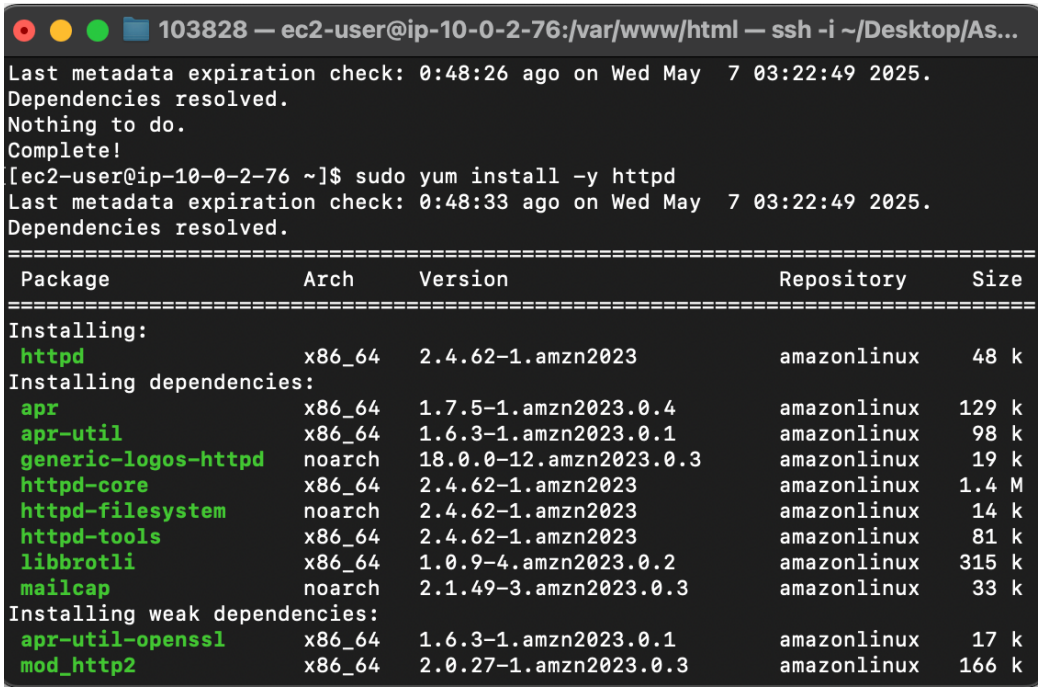

Figure 9 - Downloading Required Files

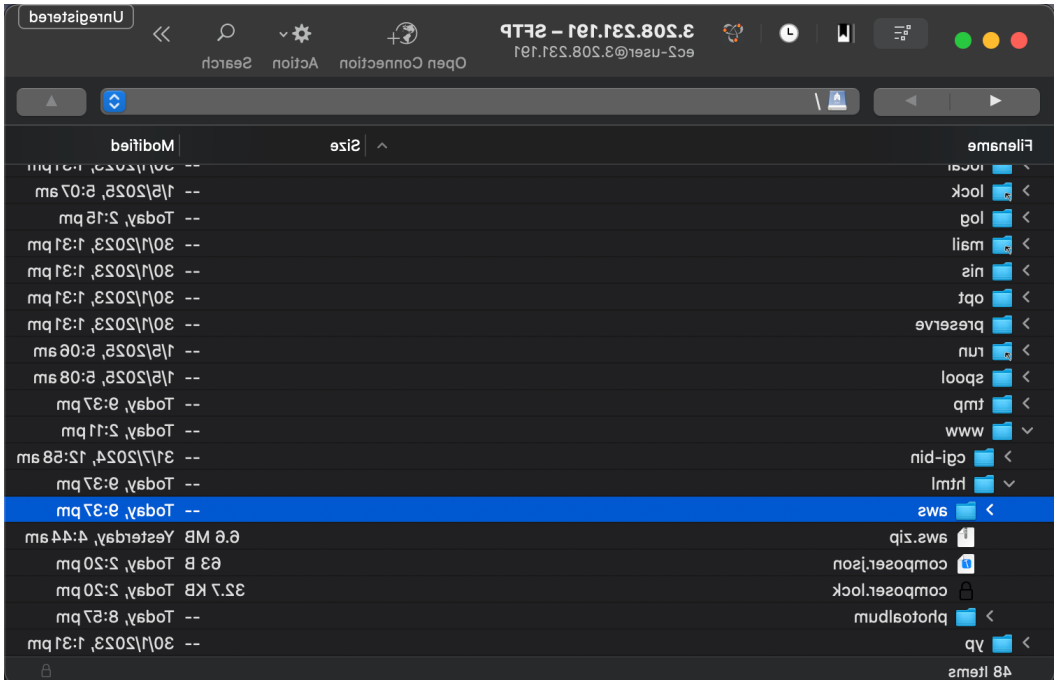

Figure 10 - Directory Structure on Cyberduck

VII. Configuring Security Groups

I then created the security groups needed for the Dev server, Web Servers, Load balancer, and Database. After creating and editing the respective AWS utilised and associated them with their security group.

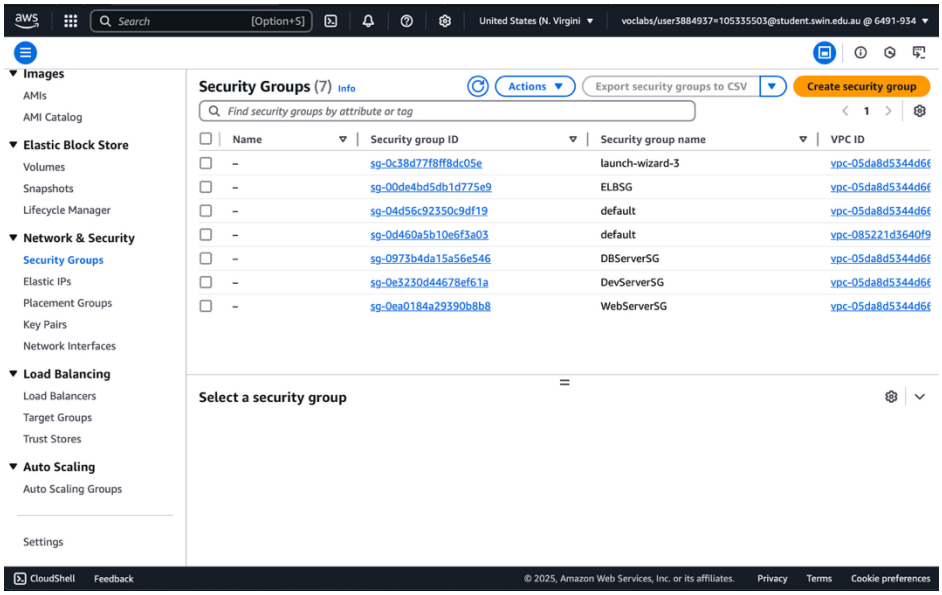

Figure 11 - All Security Groups

VIII. Creating a Lambda Function

I then created the Lambda Function that will be used to resize the images, I also configured the Lambda Function with one of the pre-made IAM Roles in order to improve security.

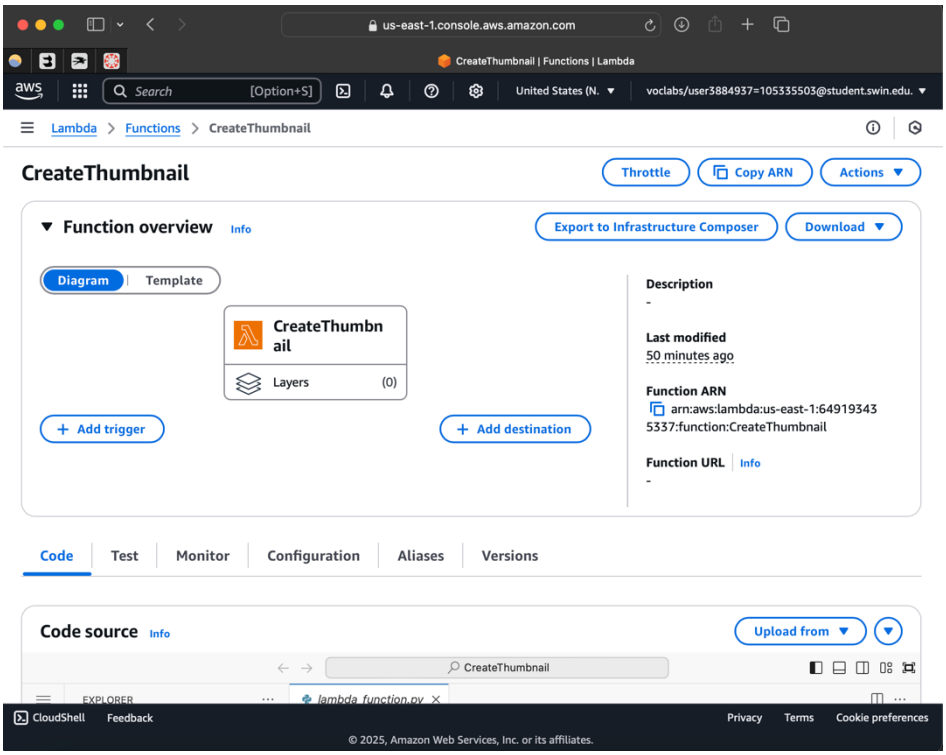

Figure 12 - Create Thumbnail Lambda Function

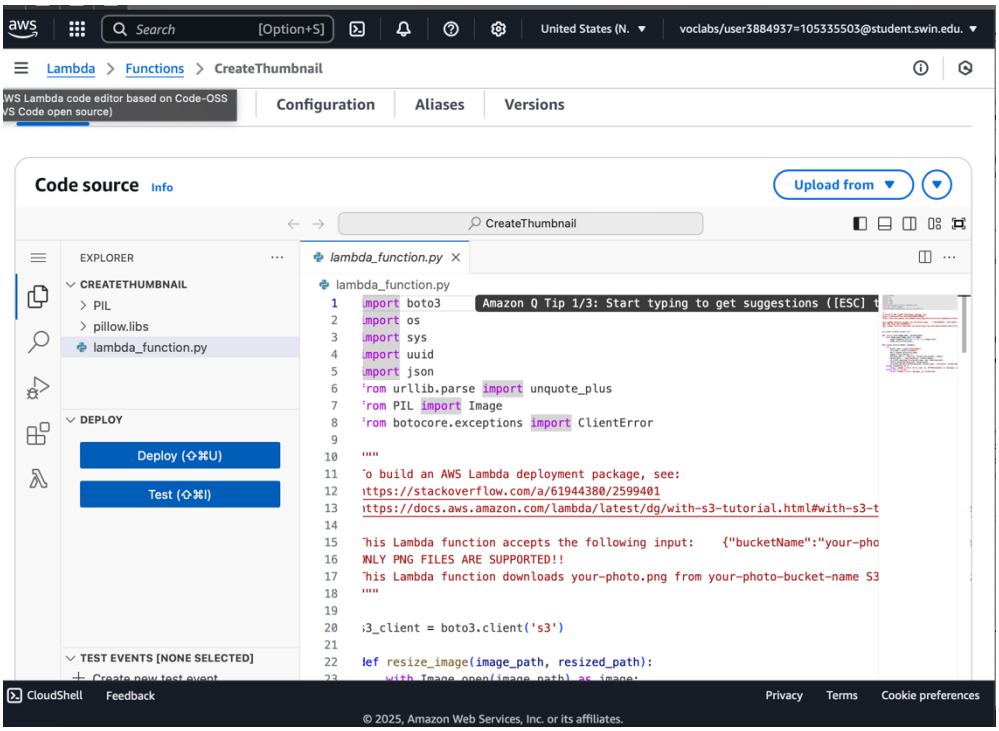

Figure 13 - Code Source for Lambda Function

IX. Creating RDS Database

I then created the RDS Database, which will be used to store the image metadata. I then configured the Database's structure via phpMyAdmin, after SSH into the Web server.

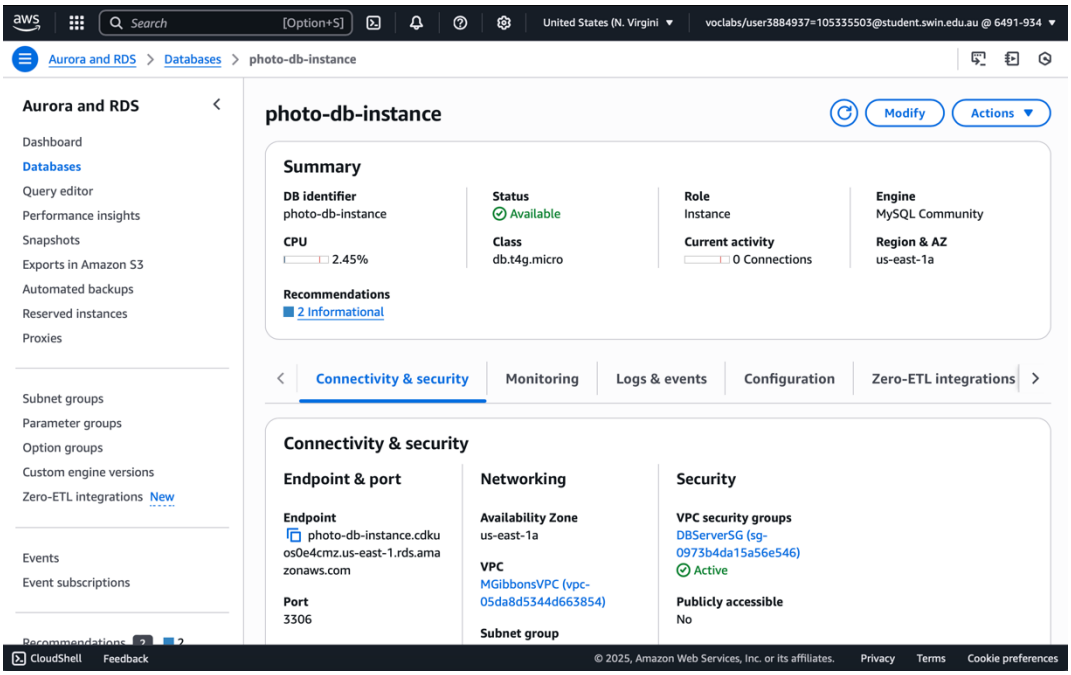

Figure 14 - Photo RDS Database

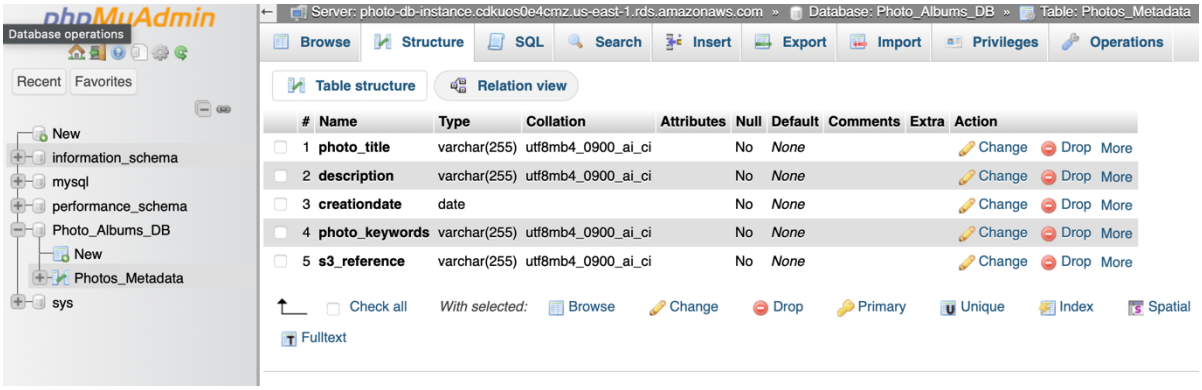

Figure 15 - Database Structure in phpMyAdmin

X. Creating AMI In Web Instance

I then utilised the Web Instance in order to create an image, which I will use later to create the Launch Template for the Auto Scaling group.

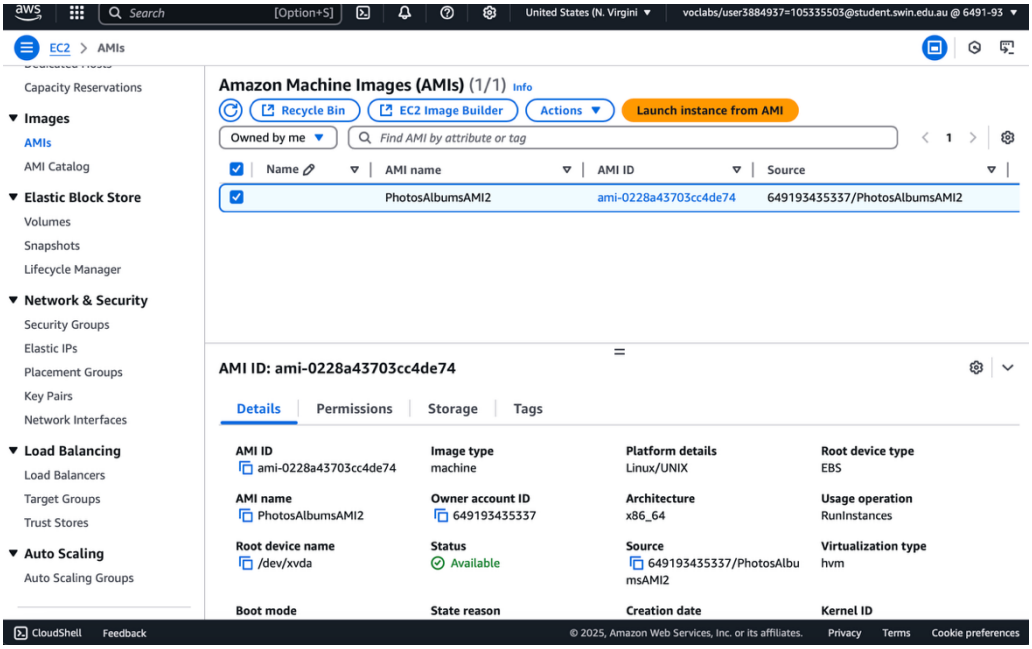

Figure 16 - AMI of Web Server

XI. Creating the Auto scaling group & Launch template

I then used the AMI I created in order to create the launch template. I then used the Launch template to create an Auto scaling group in order to create two Web Instances in the private subnets.

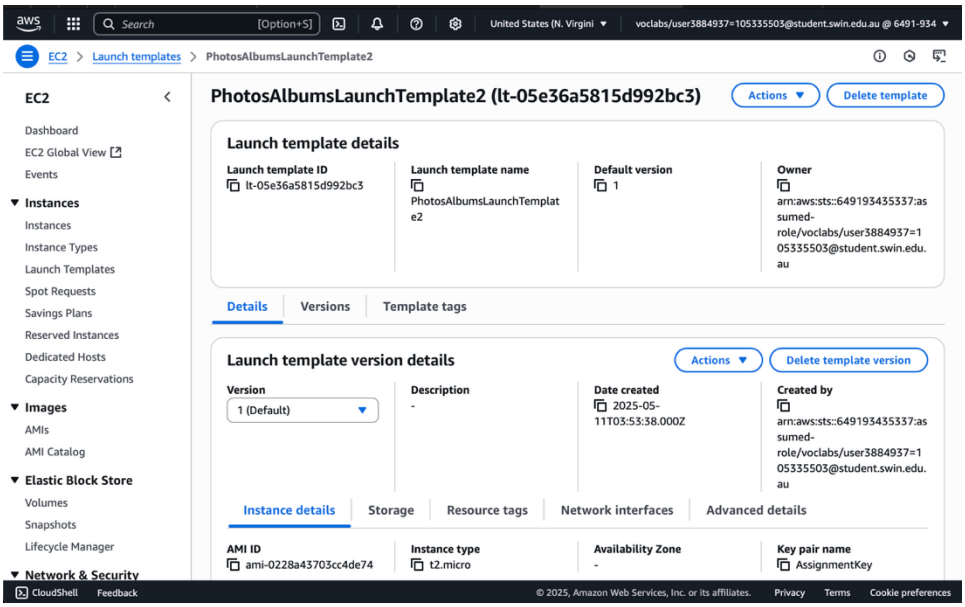

Figure 17 - Launch Template

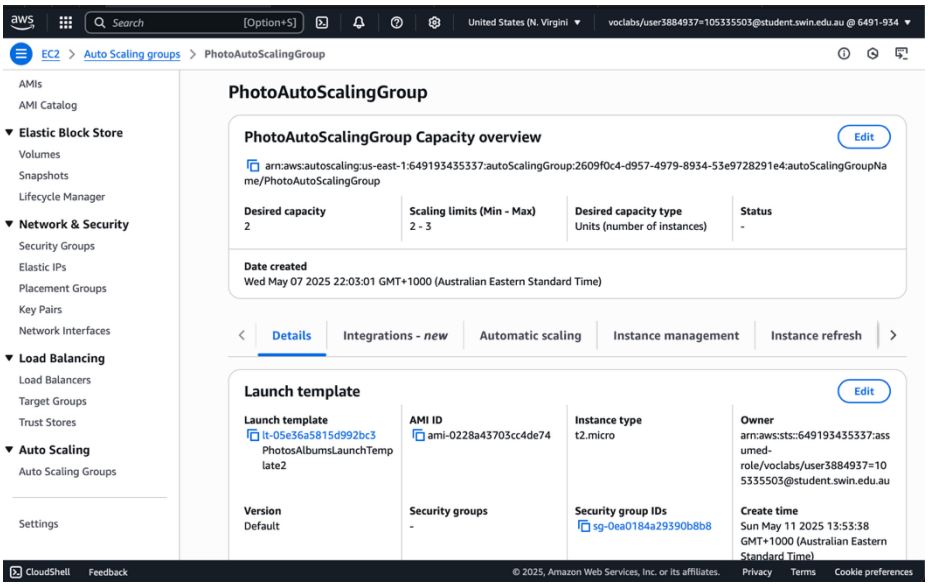

Figure 18 - Auto Scaling Group

XII. Configuring NACL's

I then created NACL policies in order to add an additional security feature via blocking ICMP traffic to and from the Dev Server.

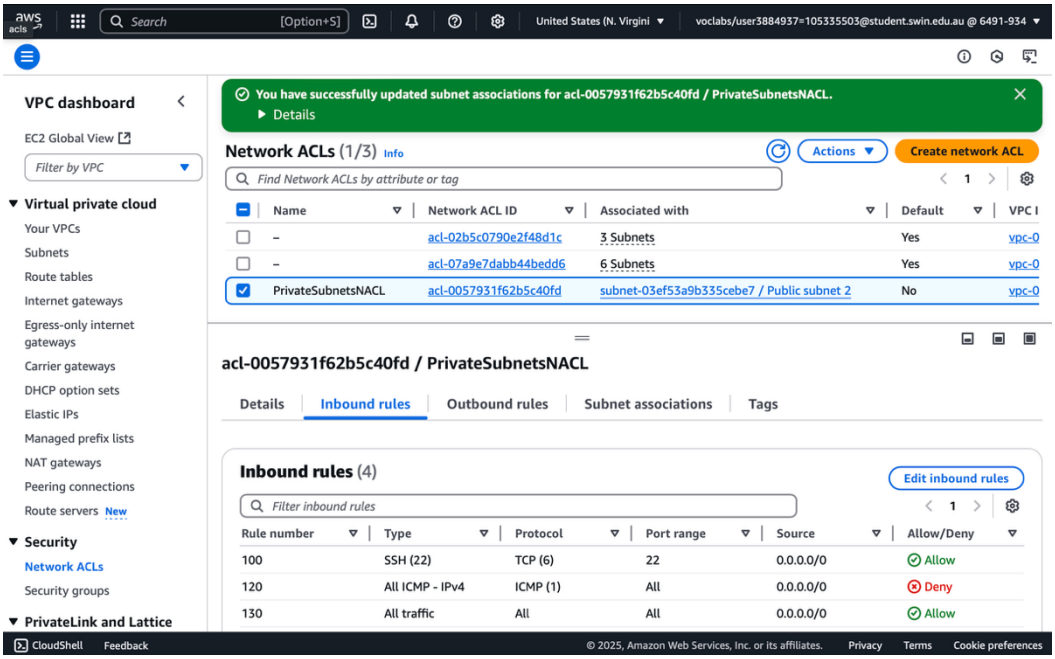

Figure 19 - NACL Policies

XIII. Testing Website

Lastly I Tested the websites functionality, determining whether it could succesfuly enable me to upload a png with its metadata and store, resize the image into a thumbnail and print the thumbnail and metadat in the table on the album.php page.

Student name: Matthew Gibbons

Student ID: 105335503

Tutorial session: Monday 16:30AM

Uploaded photos:

Upload more photos

Photo	Name	Description	Creation date	Keywords
	Truck	orange truck	2025-05-11	truck, orange, cool
	Truck2	red truck	2025-05-11	truck, red, ok

Figure 20 - Album.php Page in Browser

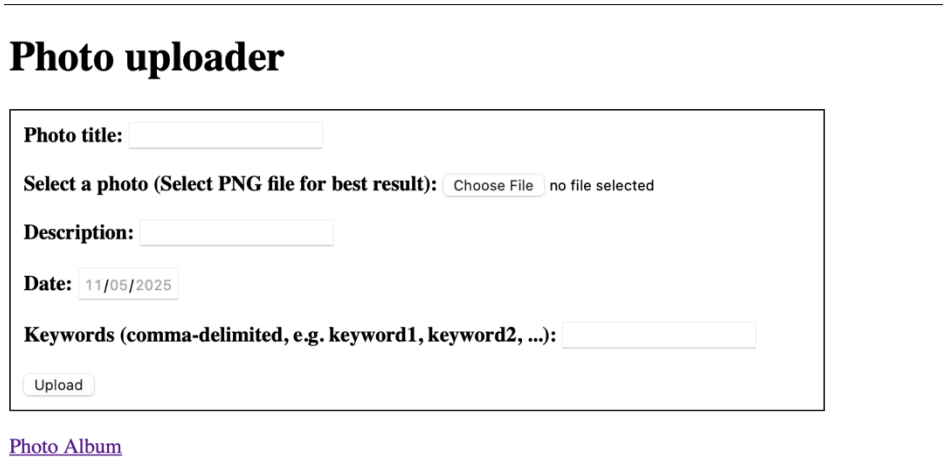

Figure 21 - photouploader.php page in Browser

XIV. Learner Lab Credit Usage

Throughout the assignment, I was cautious not to spend unnecessary credits on my Learner Lab account. In order to limit my credit usage, I ensured no unnecessary instances were running. I also removed/terminated resources used from assignment 1 or assignment 1b.

XV. Conclusion

Ultimately, I was successfully able to complete and fulfil the functional and infrastructure requirements of Assignment 2. Although I had difficulties with the use of the auto scaling group and launch template, I was able to overcome these difficulties by reviewing instructions from previous labs. I am proud of my achievements throughout this assignment and am significantly more confident in my ability to create highly available websites.