

COS20019 Cloud Computing Architecture – Assignment 2

Lorraine Becker
Student ID: 104584773
Tutorial: 08:30 Thursday
Swinburne University of Technology
Melbourne, Australia
104584773@student.swin.edu.au

Abstract - This report presents the process of developing a highly available Photo Album website onto AWS infrastructure.

Links:

- Album.php: <http://webserverelb-838355382.us-east-1.elb.amazonaws.com/photoalbum/album.php>

Keywords — cloud architecture, deployment, technology

INTRODUCTION

Assignment 2 of COS20019 Cloud Computing Architecture is focused on the development and deployment of a photo album website on Amazon Web Services (AWS) infrastructure. The cloud architecture outlined for this assignment is as follows:

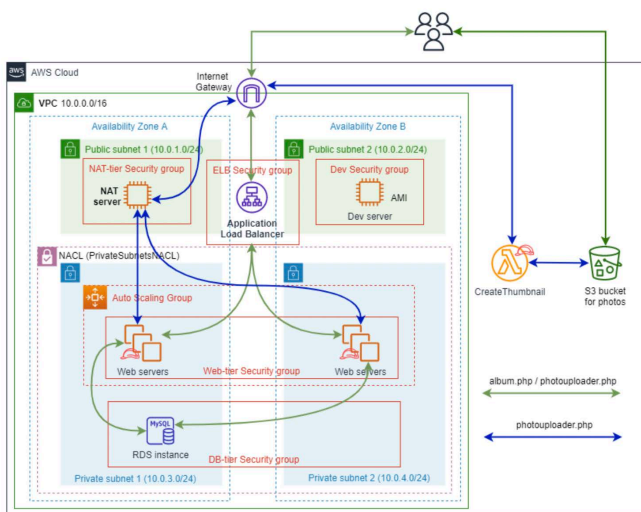


Figure 1 - Architecture diagram

Figure 1 System Architecture Diagram

VPC

A Virtual Private Cloud (VPC) was set up to host the services needed for this assignment. In the us-east-1 region, the VPC was configured exactly as shown in Figure 1. Public and private subnets were designed in both Availability Zones (AZ) as shown in Figure 1 so that their CIDRs corresponded.

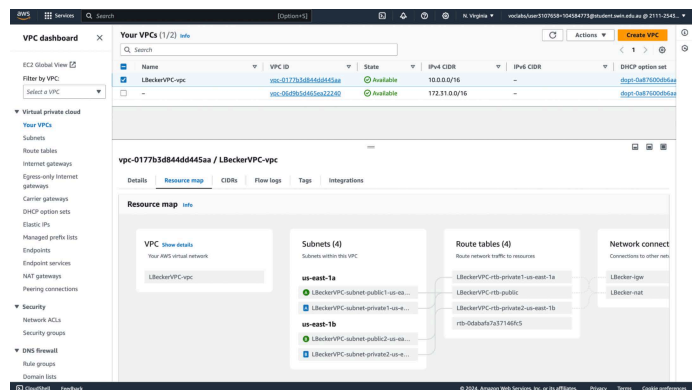


Fig 2 VPC Created

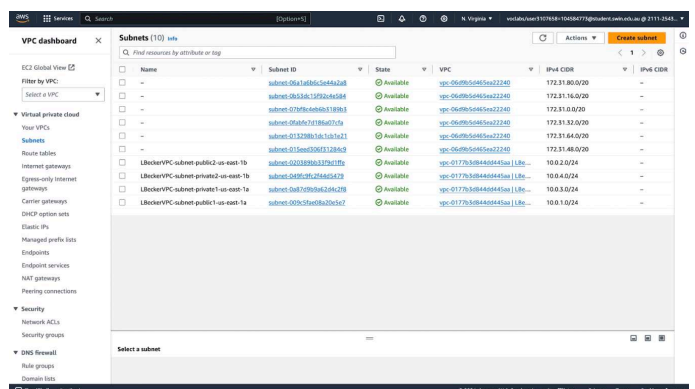


Fig 3 Subnets created for VPC

I. Public route table

The public route table directs traffic toward the internet gateway, facilitating connectivity to external networks. Within this framework, both public subnets are linked, as illustrated in Figure 4.

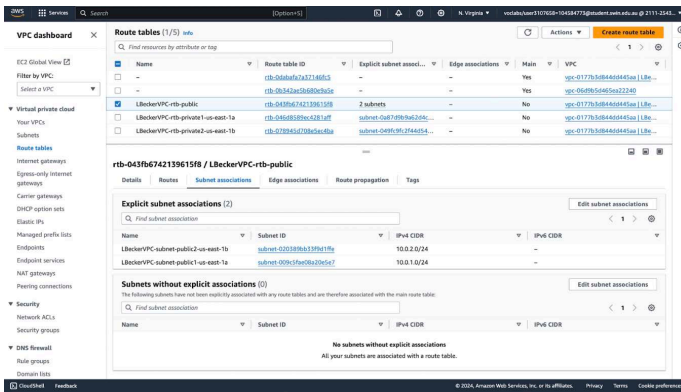


Fig 4 Subnet associations of public route table

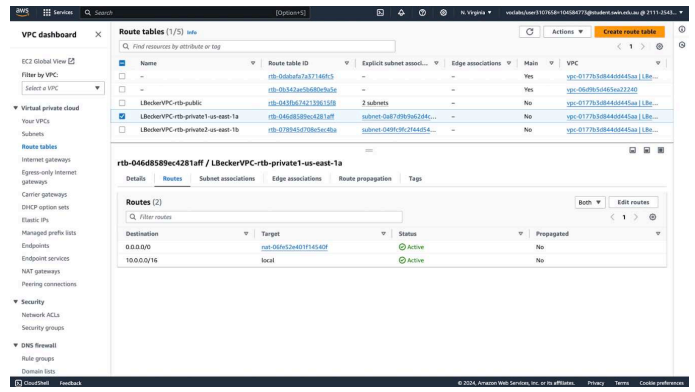


Fig 7 Routes of private route table 1

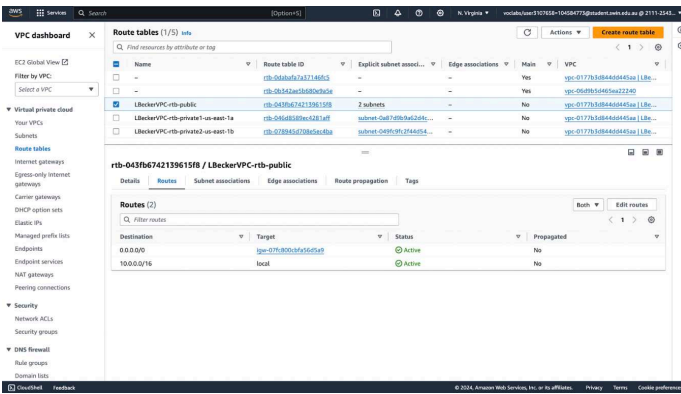


Fig 5 public route table routes

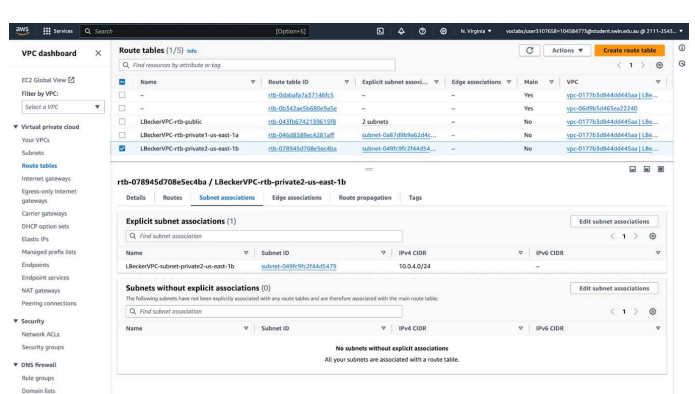


Fig 8 Subnet associations of private route table 2

I. Private route tables
Private route tables hold routes directing traffic to the NAT gateway, enabling communication between resources within private subnets and the Internet.

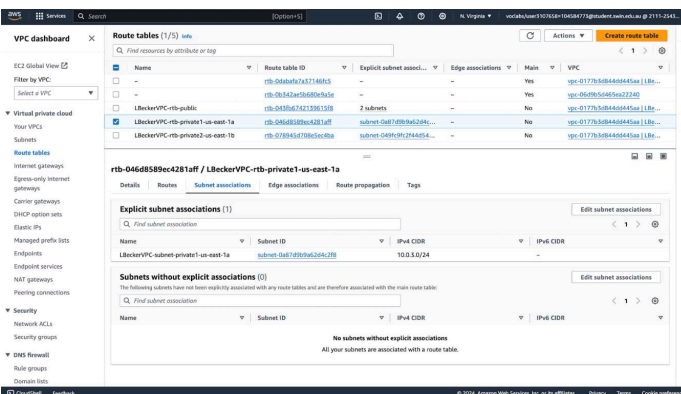


Fig 6 Subnet associations for private route table 1

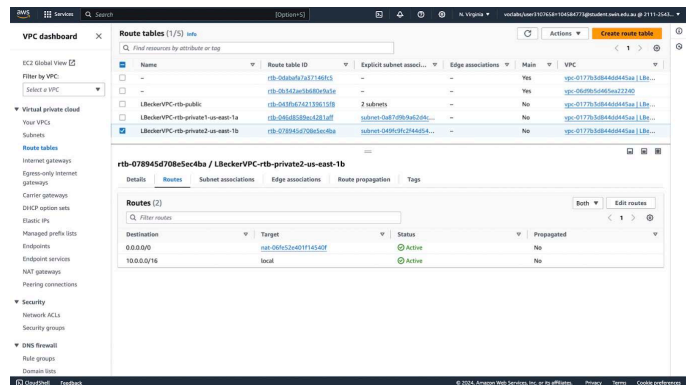


Fig 9 Routes of private route table 2

II. Security Groups

Following the guidelines outlined in Figure 1, security measures were established, each with specific rules. Notably, there was no need for a security group dedicated to NAT since a NAT gateway was employed instead of an EC2 instance for NAT functionality. The DevServerSG, for instance, permits inbound traffic from any source and allows unrestricted outbound connections.

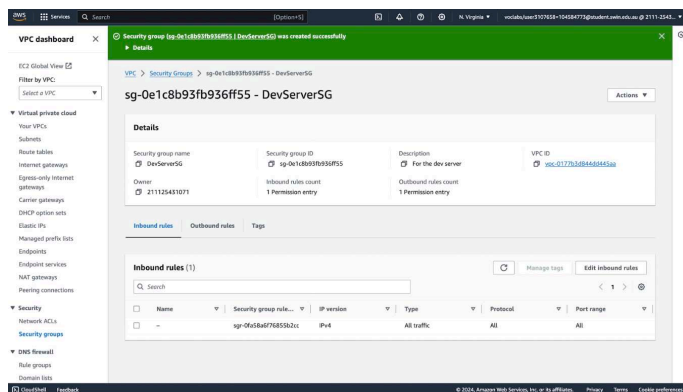


Figure 10 Dev Server security group

DBServerSG is configured with two inbound rules facilitating MySQL traffic from both DevServerSG, where the DevServerSG, and WebServerSG, hosting web servers within the auto-scaling group. The DevServerSG is in the rules for the application test before the creation of an AMI.

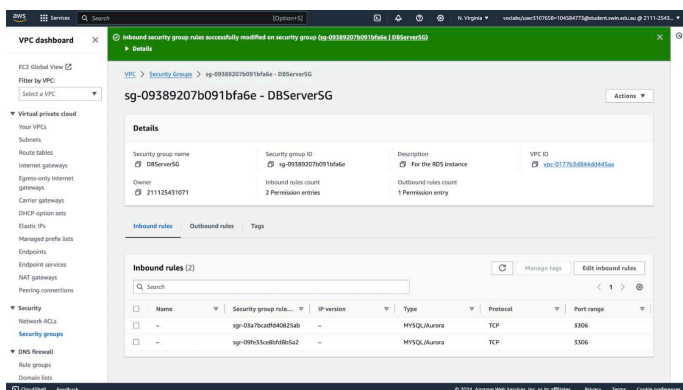


Fig 11 Security group of DBServerSG

WebServerSG permits incoming HTTP traffic originating from ELBSG. This configuration ensures that the web servers can effectively receive traffic forwarded by the load balancer.

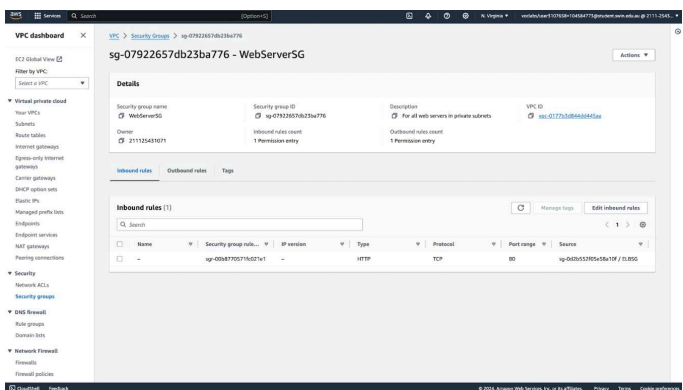


Fig 12 Security group of WebServerSG

HTTP traffic is allowed from anywhere for the ELBSG security group

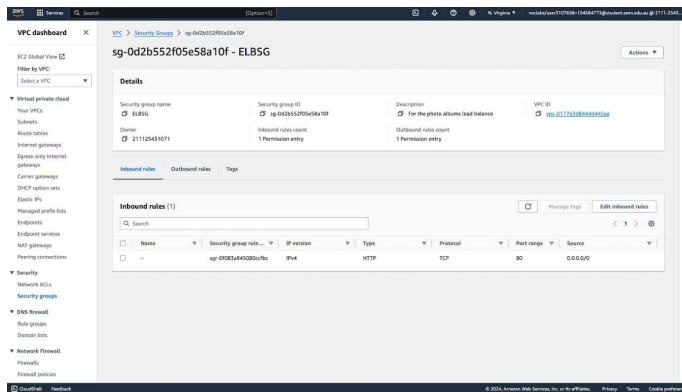


Fig 13 Security group of ELBSG

RDS

An instance of Amazon Relational Database Service (RDS) was established under the name "photoalbum-db". This RDS instance was created for the photo album, containing a table responsible for storing metadata of the photos. To facilitate access for the photo album application, this RDS instance is incorporated into the DBServerSG security group. Furthermore, the subnet group for the RDS instance is configured to utilise the two private subnets within the designated VPC, as per the instructions.

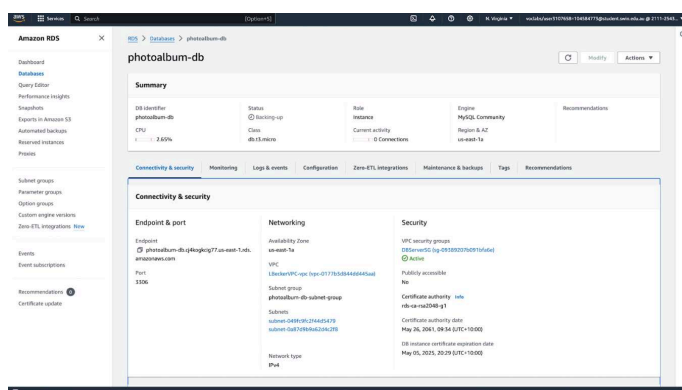


Fig 14 Photo Album RDS Instance

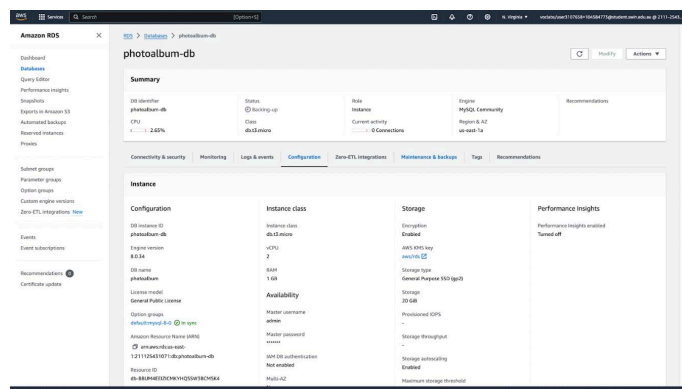


Fig 15 Photo Album RDS Instance Configuration

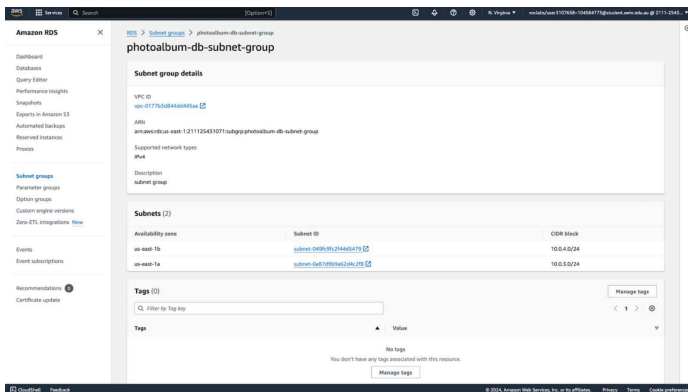


Fig 16 Subnet group of Photo Album RDS Instance

EC2

The dev server instance was positioned within public subnet 2. This server plays a role in executing the initial configuration of the photo album application, AWS SDK, phpMyAdmin, and their associated dependencies. Additionally, it was utilised to establish the photos table within the RDS instance, essential for the functioning of the photo album application. To enable the execution of the CreateThumbnail Lambda function, the LabRole IAM role was assigned to the development server.

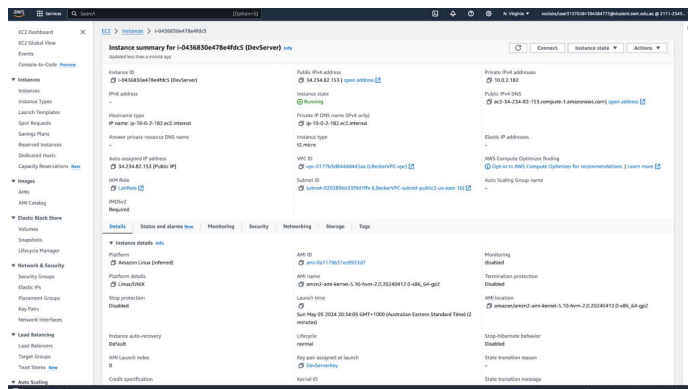


Fig 17 Instance of DevServer

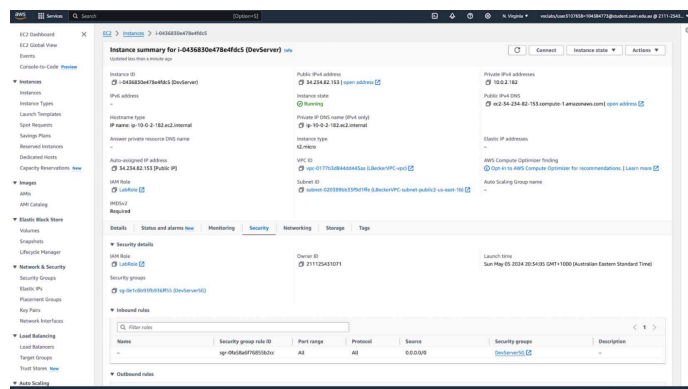


Fig 18 Dev Server Instance Security

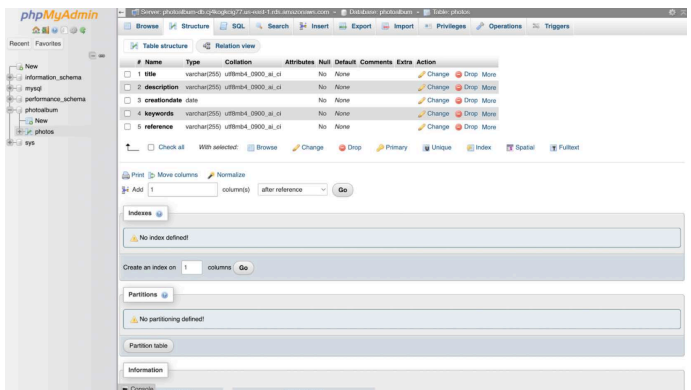


Fig 19 Photos Table Structure

S3 BUCKET

In the US East (N. Virginia) region, an S3 bucket named "lbecker-photoalbum-bucket" was established. Its purpose was to serve as a repository for the photos placed on the photo album website.

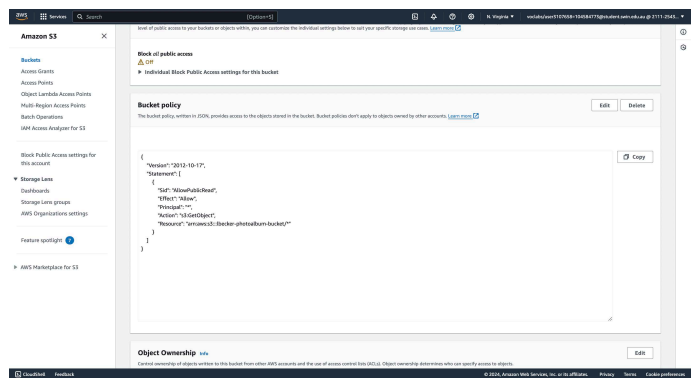


Fig 20 S3 Bucket Policy

LAMBDA

A Python 3.11 Lambda function named "CreateThumbnail" has been made. This function operates on arm64 architecture and its task is to involve fetching a file from a designated S3 bucket and filename, then resizing it. Finally, it re-uploads the resized image file back to the S3 bucket, prefixed with "resized-".

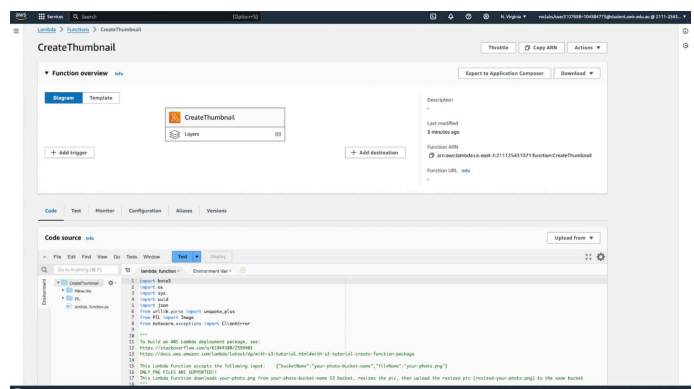


Fig 21 Function Code for CreateThumbnail

TARGET GROUP

WebServerTargetGroup is a target group created to use by the Elastic Load Balancer. The endpoint “/photoalbum/album.php” is given to conduct health checks for. This is important for verifying if a target is healthy.

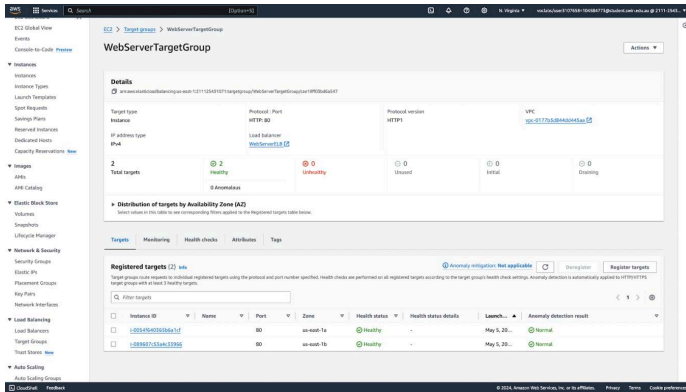


Figure 29 Target Group Targets

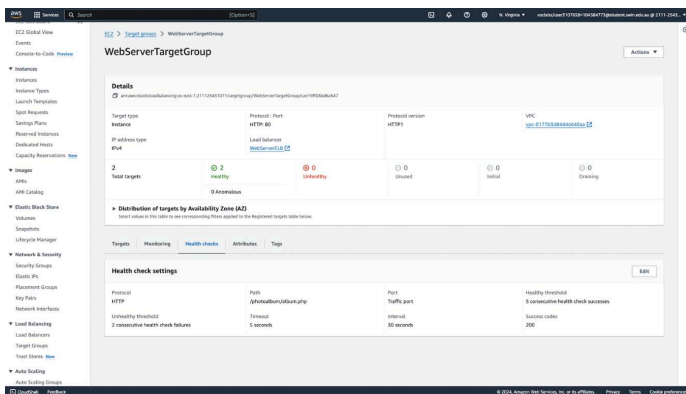


Fig 30 Target Group Health Check

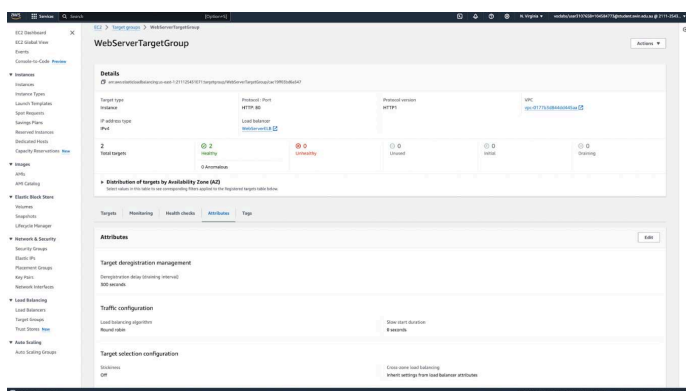


Fig 31 Target Group Attributes of WebServerTargetGroup

AUTO SCALING GROUP

The auto-scaling group named "WebServerASG" was set up to independently control the number of server instances. Employing the WebServerLT launch template, this group is configured with a minimum and desired capacity of 2 instances and a maximum limit of 3. Finally, WebServerASG launches instances into the two private subnets

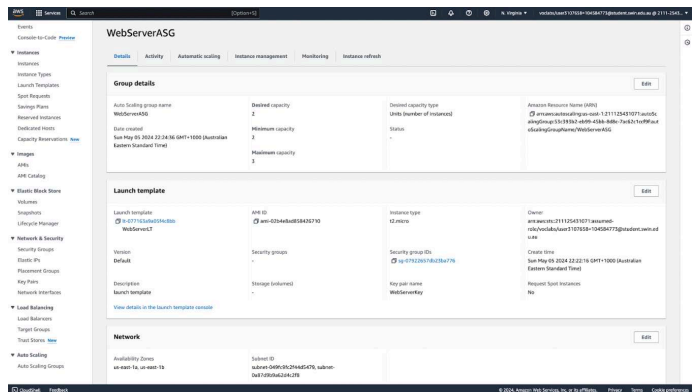


Fig 32 WebServerASG

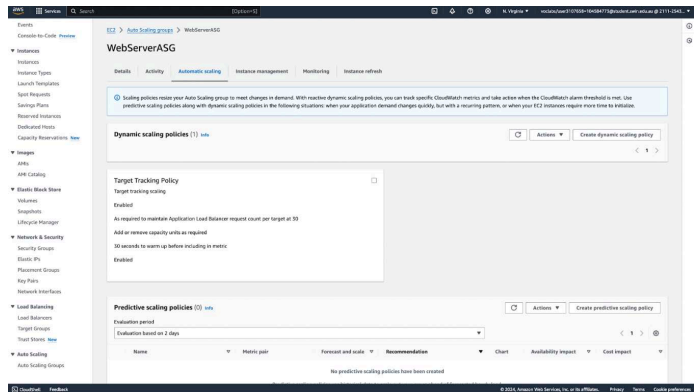


Fig 33 WebServerASG Target Tracking Policy

LOAD BALANCER

Within the load balancer named "WebServerELB," in its configuration for listeners and rules, it actively monitors port 80 and redirects incoming traffic to the designated WebServerTargetGroup.

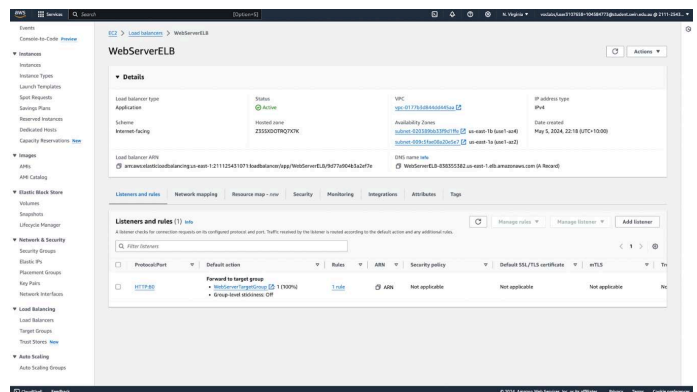


Fig 34 WebServerELB Listeners and rules

IAM ROLES

With regards to the regulations of the Learner Lab, the creation of new roles is restricted, thus necessitating the utilisation of the provided role.

The LabRole offers permissions to both the Lambda function and the Dev/Web servers. This ensures that the application fully operates as expected.

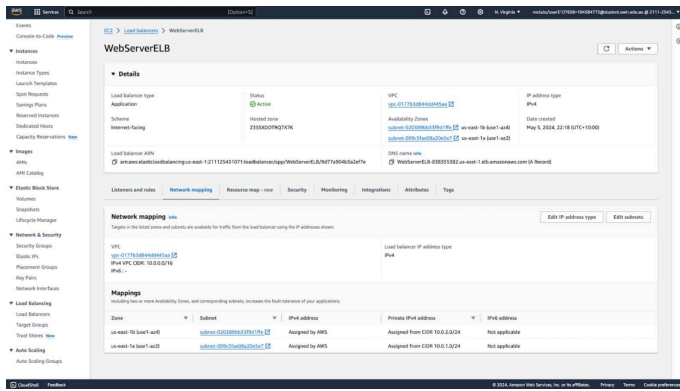


Fig 35 WebServerELB Network mapping

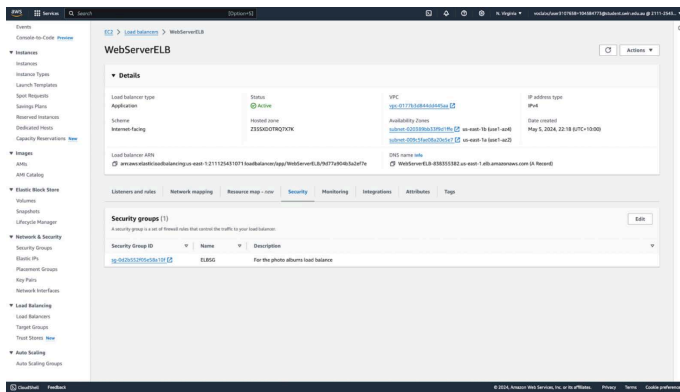


Fig 36 WebServerELB Security

NETWORK ACL

“PrivateSubnetsNACL” is the name of the network ACL that I created. Its purpose is to restrict ICMP traffic exchange between the Dev Server. Furthermore, the network ACL facilitates inbound HTTP traffic originating from the pair of public subnets designated for use by the load balancer.

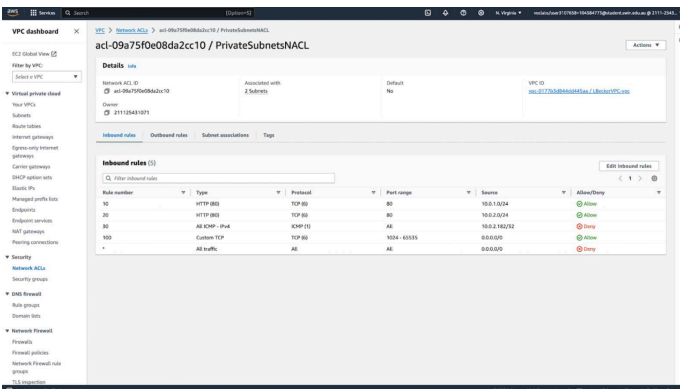


Fig 37 Inbound Rules of PrivateSubnetsNACL

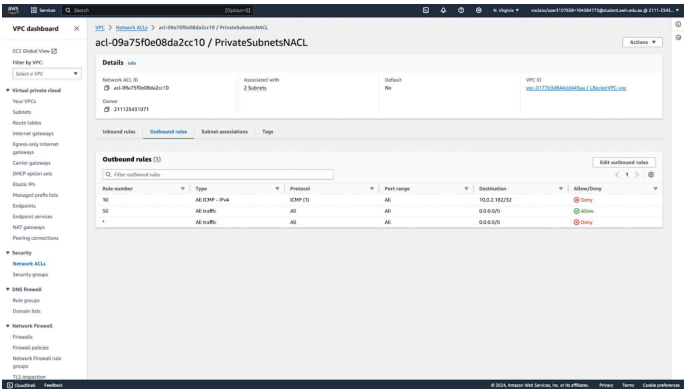


Fig 38 Outbound Rules of PrivateSubnetsNACL

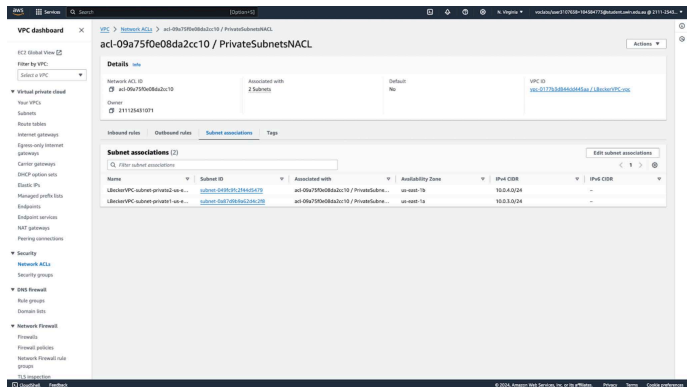


Fig 39 Subnet Associations of PrivateSubnetsNACL

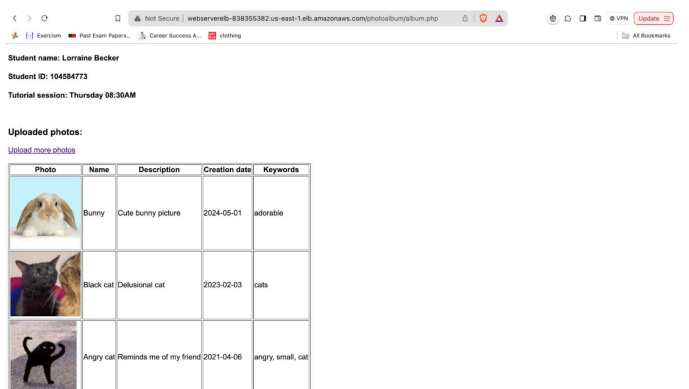


Fig 39 Images uploaded onto album.php

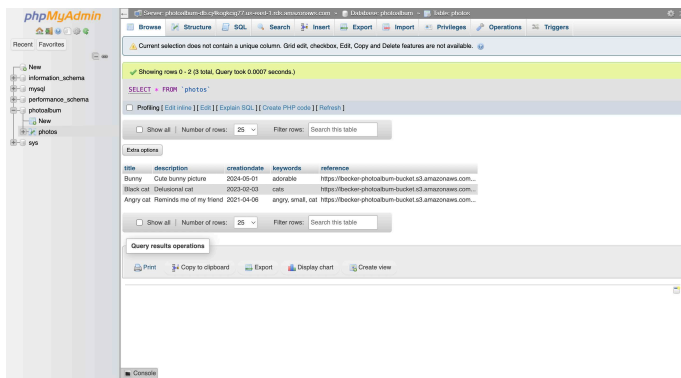


Fig 41 The images in phpMyAdmin

CONCLUSION

AWS delivers robust and scalable infrastructure and functionalities for constructing cloud-based systems. Leveraging AWS services, the Photo Album site was deployed on AWS infrastructure. AWS EC2 was utilized for hosting the Web Server, RDS managed the photo metadata, and S3 stored the image files. Lambda played a role as a in resizing images upon upload to the photo album. Moreover, Load balancing was made through the integration of a the following: AMI, Launch Template, Target Group, Auto Scaling Group, and Elastic Application Load Balancer.

[1] Swinburne University, "COS20019 - Assignment 2," Swinburne University, 2024. [Online]. Available: <https://swinburne.instructure.com/courses/56945/assignments/587215>

[2] Amazon Web Services, "NAT Instances", Amazon Virtual Private Cloud, 2024. [Online]. Available: https://docs.aws.amazon.com/vpc/latest/userguide/VPC_NAT_Instance.html

