USER MANUAL

10/10/2023

I. Introduction

This project will extend/modify the infrastructure and program developed in project phoot album.

II . Create a secure Virtual Private Cloud (VPC)

The VPC is named ‘AKaggdasVPC’ because we need to give it a name using the initial of our first name and Last name. The VPC has two availability zones each with a private and public subnet with suitable CIDR.

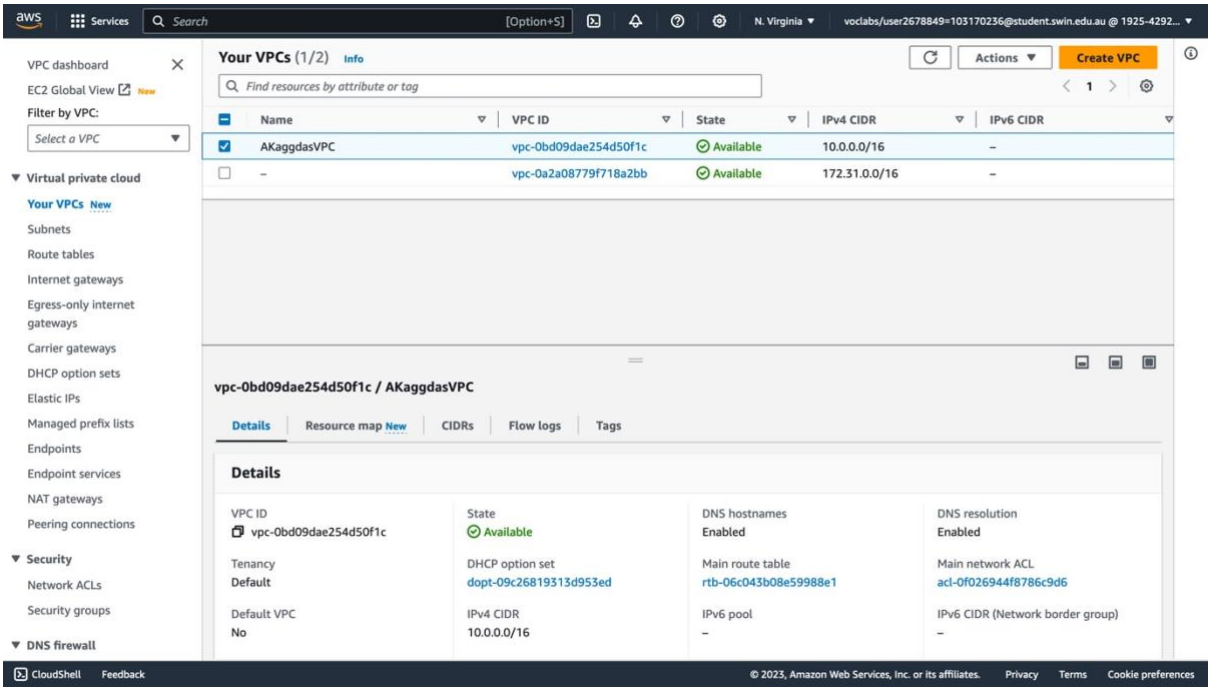

Figure 1 - Virtual Private Cloud

III. Route Tables

- **Public Route Table Configuration :**

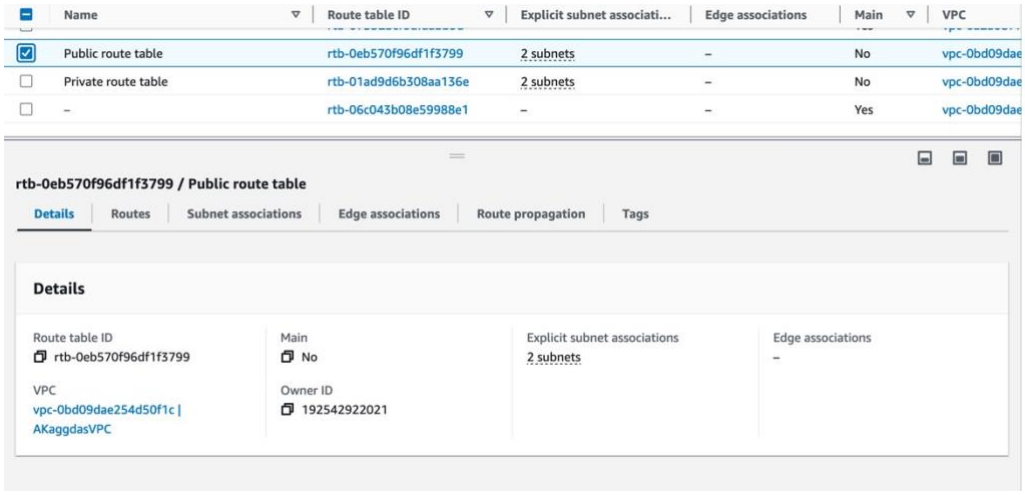

Figure 2 - Public Route table

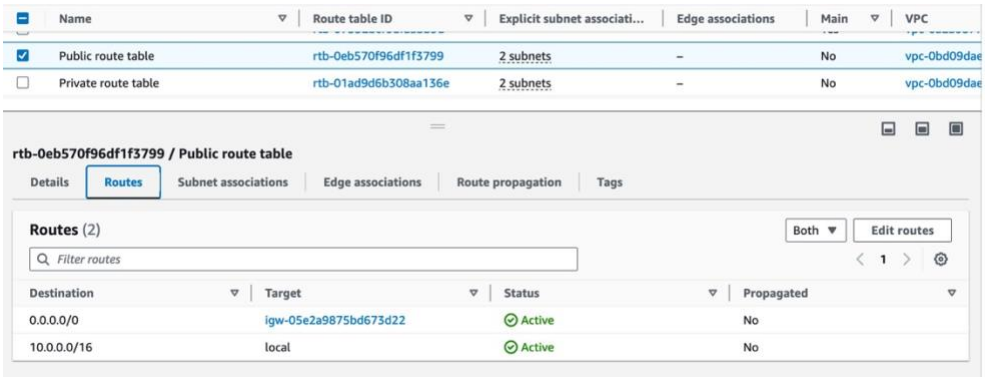

Figure 3 - Public Routes

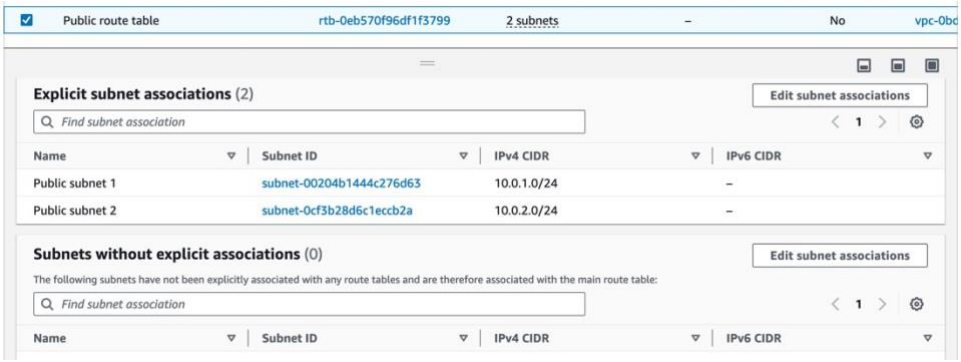

Figure 4 - Public Routes subnet associations

- **Private Route Table Configuration :**

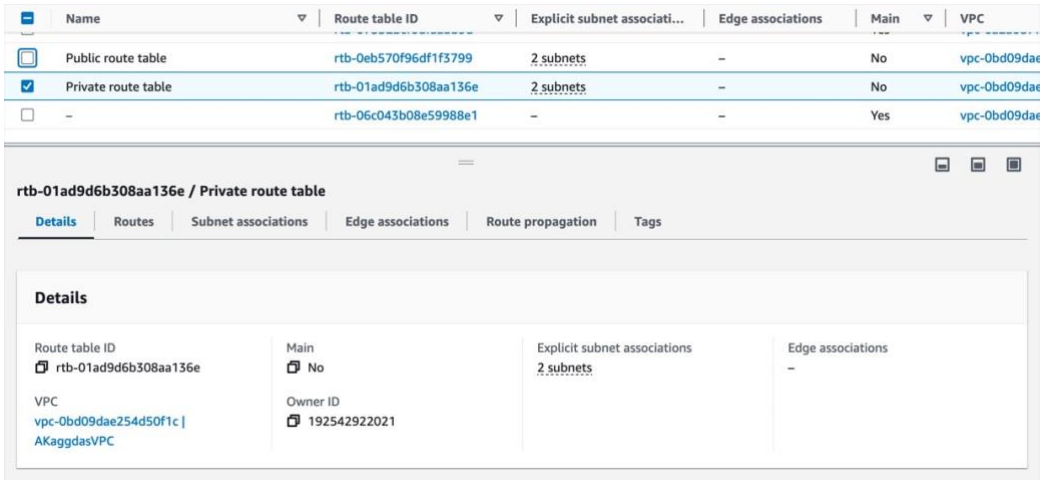

Figure 5 - Private Routes table

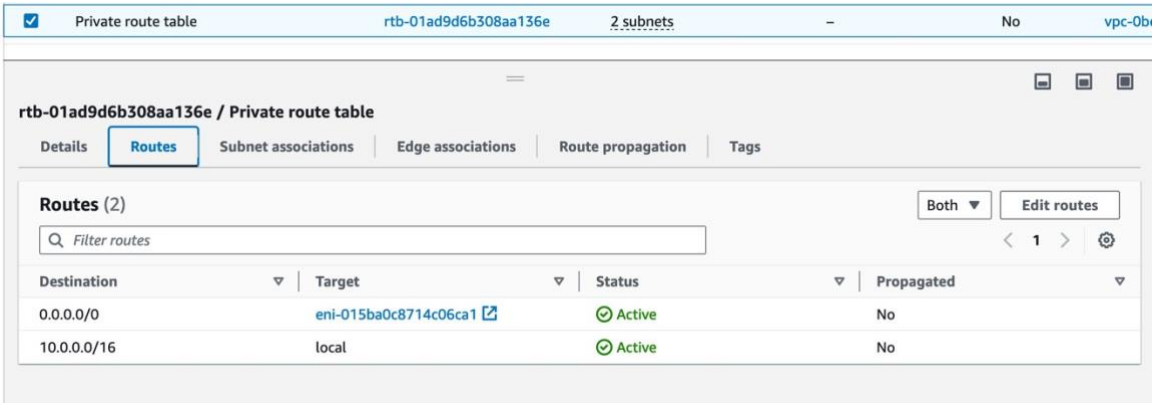

Figure 6 - Private Routes

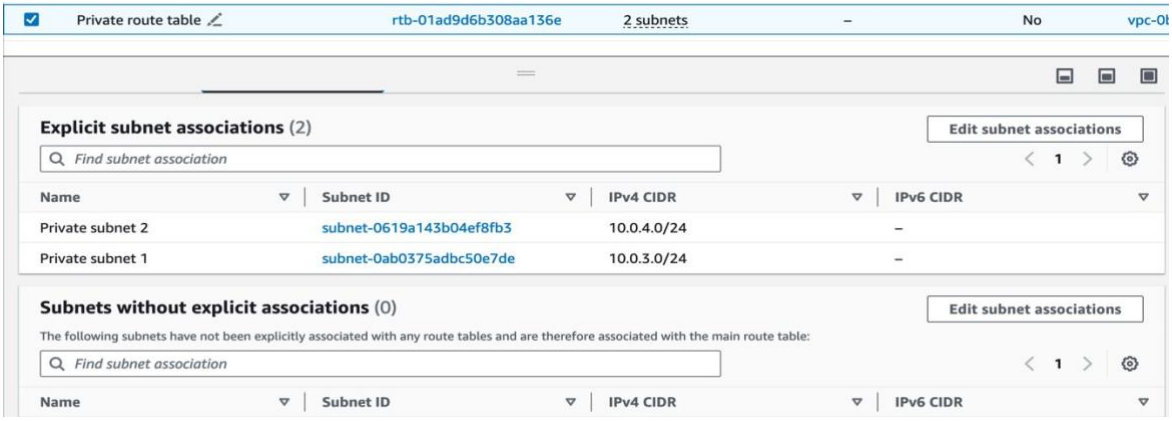

Figure 7 - Private Routes subnet associations

IV. Create Security groups

We create five security groups : 1] DevServerSG 2] DBServerSG. 3] ELBSG 4] NAT Server 5] Web Server .
We edit the Inbound rules for all the three and leave the outbound rules to default**.**

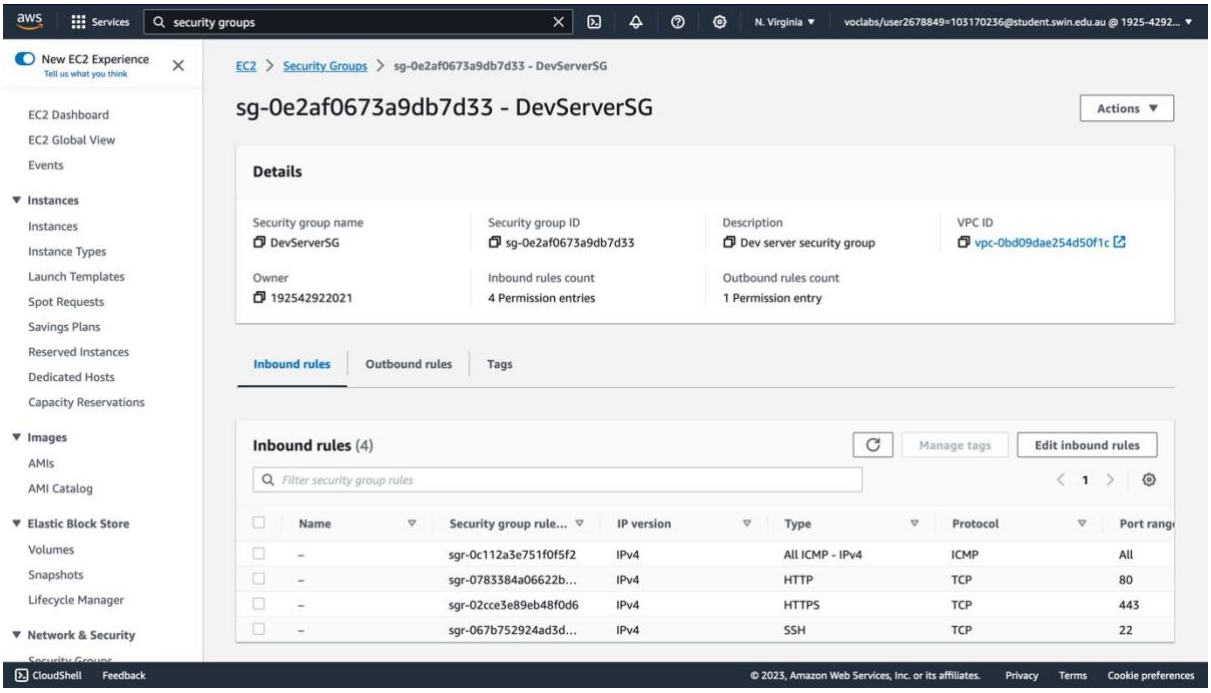

Figure 8 –Dev Server Security Group

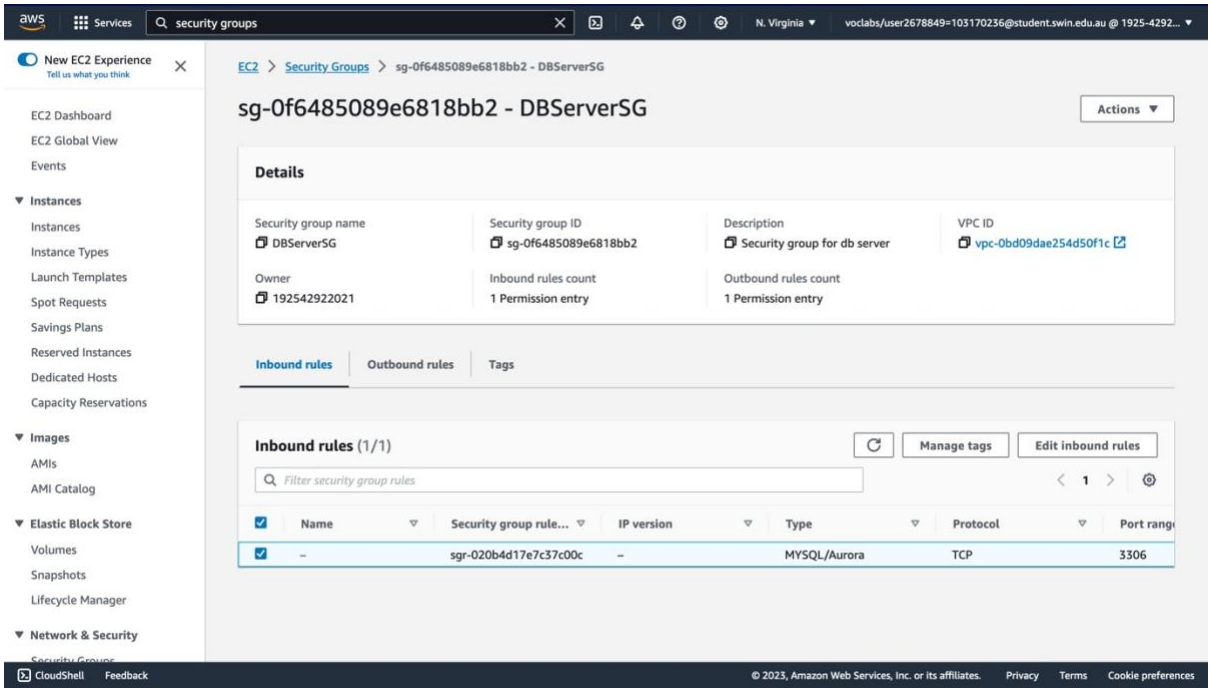

Figure 9 – DBServer Security Group

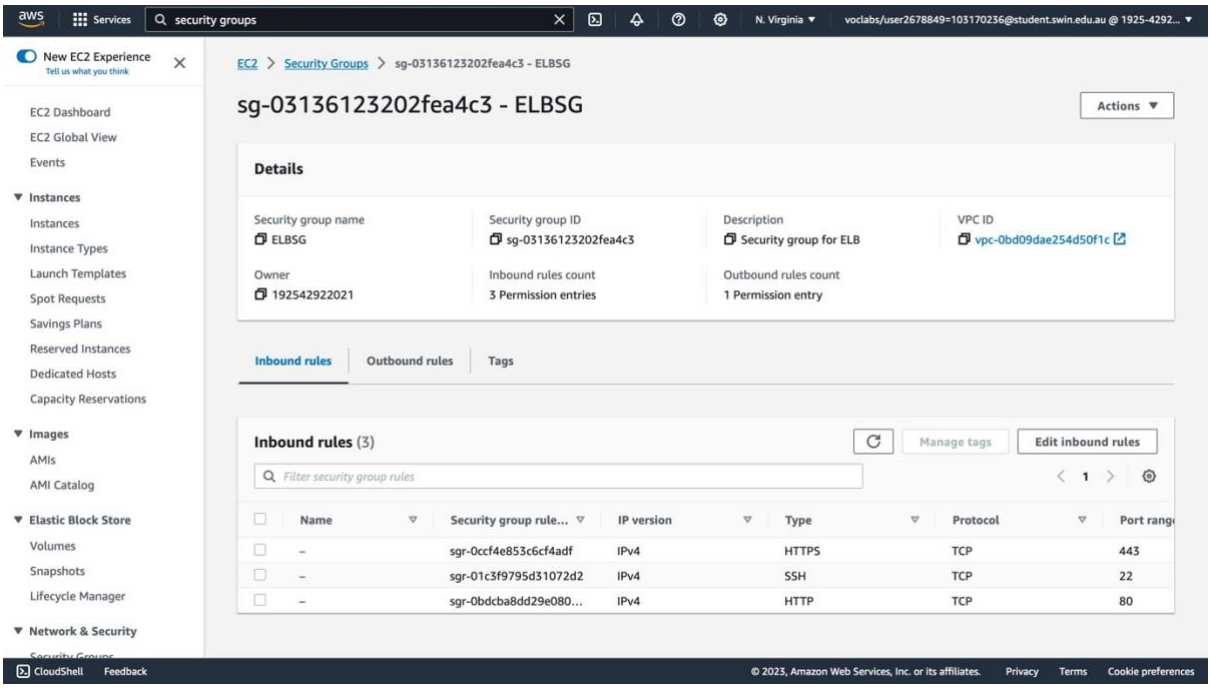

Figure 10 – ELBSG Security Group

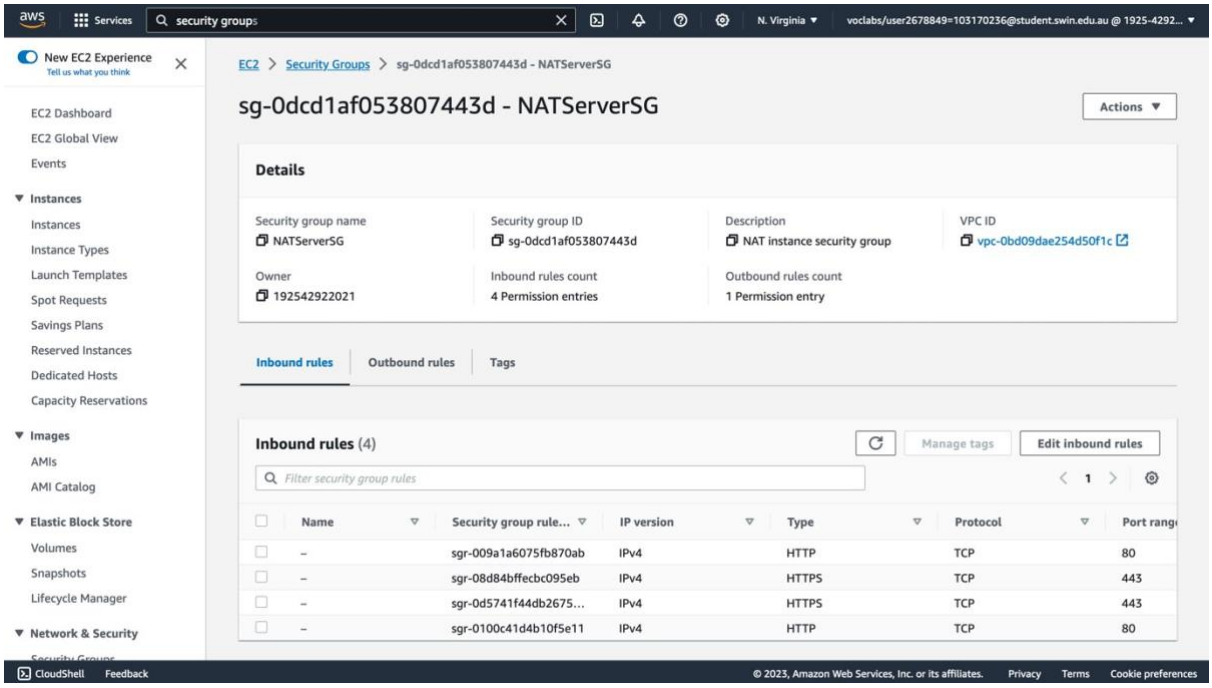

Figure 11: NAT Server Security Group

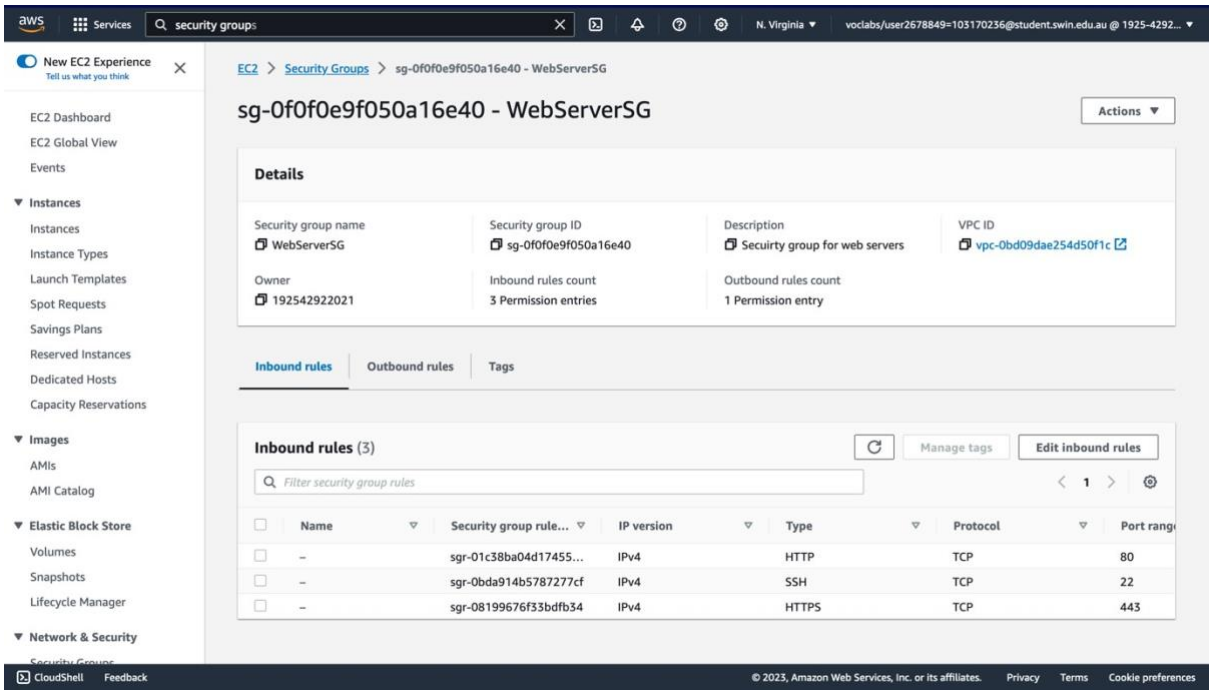

Figure 12 – WebServer Security Group

V. Create Key Pair

We create a key pair to associate with the EC2 instances .We name the key pair ‘assignment2.

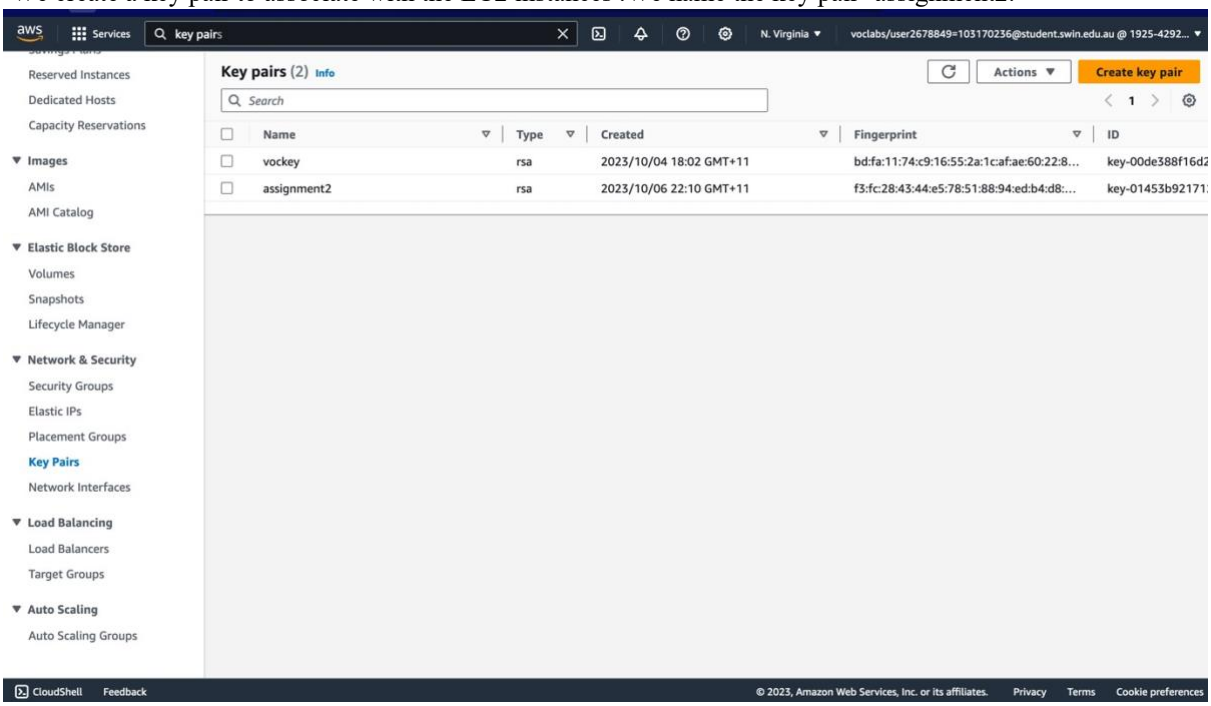

Figure 13 - Key Pair

VI. Create EC2 Instances

We create two EC2 instances, a ‘Dev server’ instance and a ’Nat’ instance. The Nat instance is created in order for the instances launched by the auto scaling group to communicate to resources outside the VPC(the internet). The ‘Dev server’ instance is created with all necessary configurations and an AMI of this instance created to launch the auto scaling group.

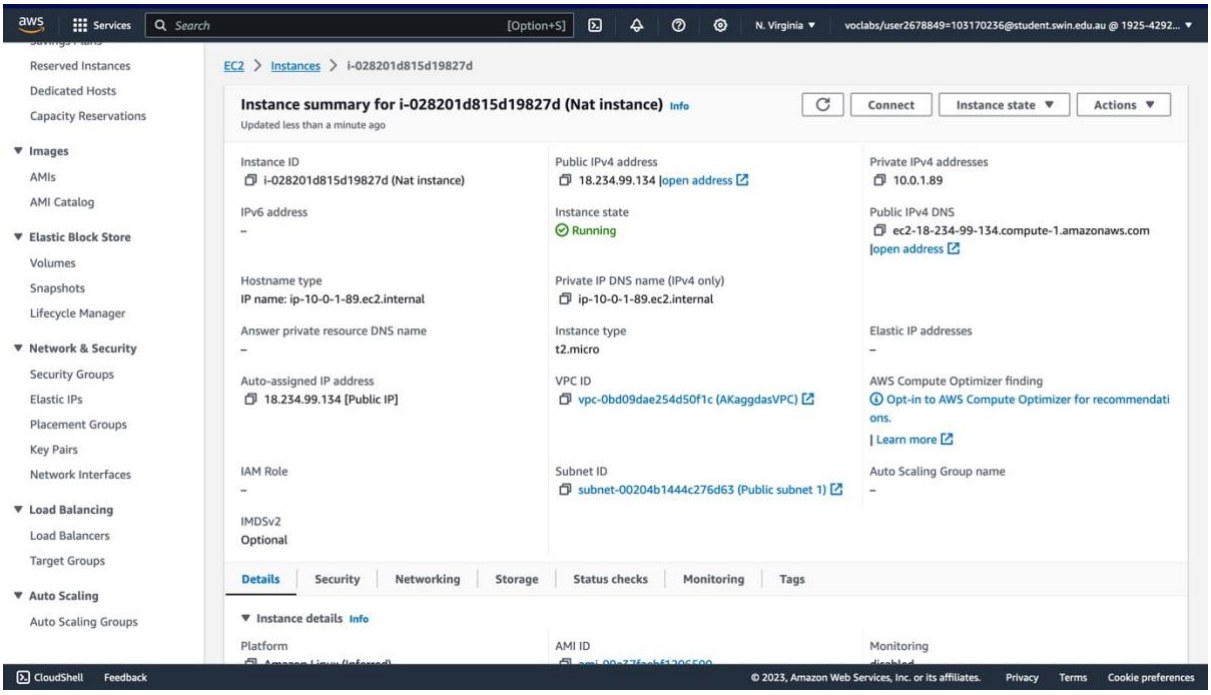

Figure 14 : Summary of Nat Instance

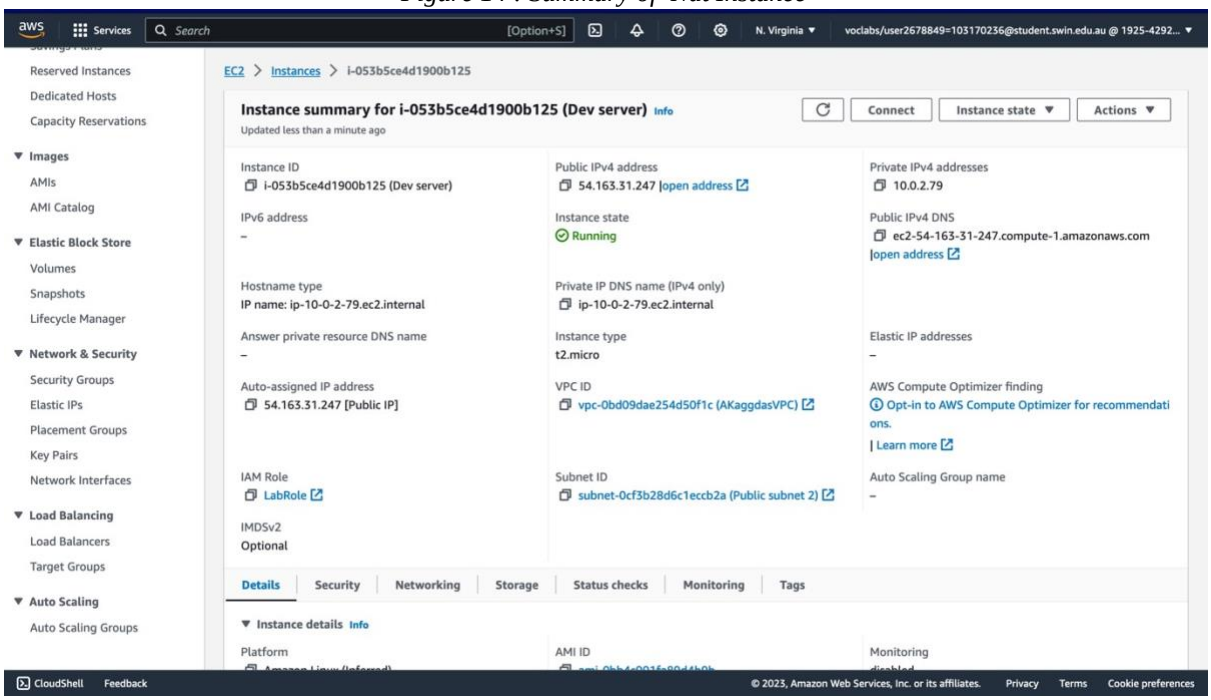

Figure 15: Summary of Dev server Instance

VII. Create an Amazon RDS DB Instance

We name the RDS ‘photoalbum’.

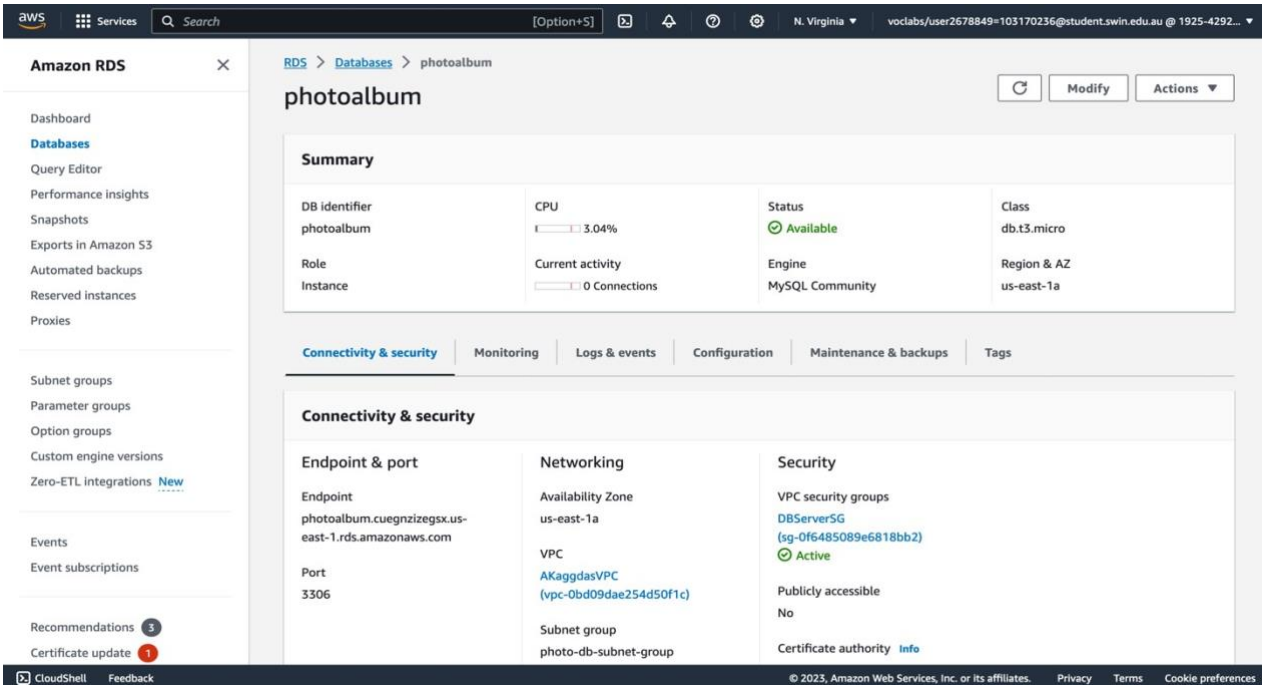

Figure 16- Summary of photoalbum RDS

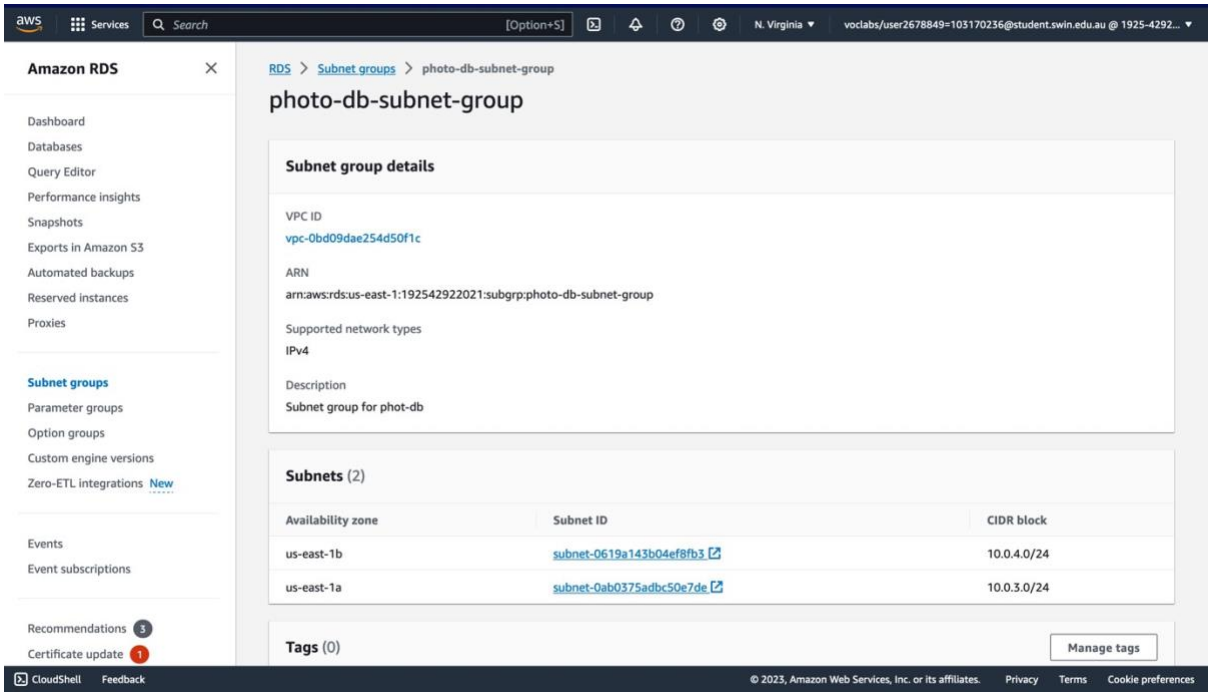

Figure 17- photo-db-subnet-group

VIII. S3 photo storage

We create an S3 bucket and name it ‘aish-photos-bucket’ to store our photos.

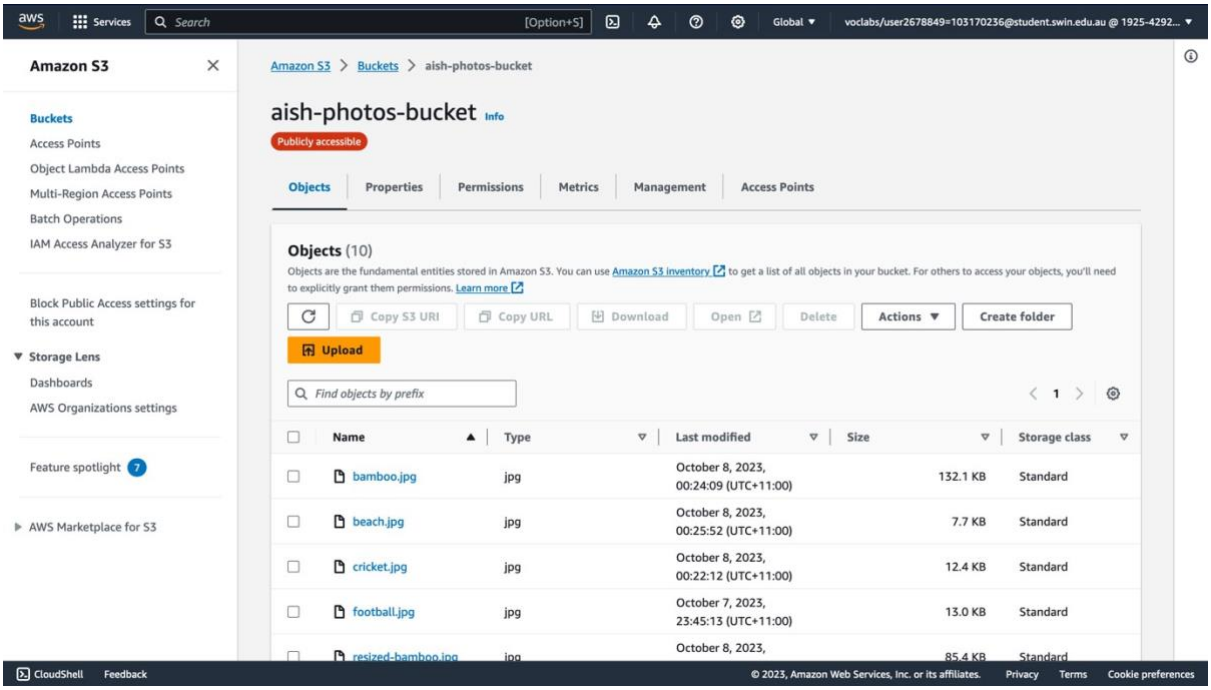

Figure 18 – aish-photos bucket Objects

Below is the screenshot of the bucket policy attached to our bucket, essentially this bucket only allows our website with the load balancer dns to retrieve the images from the bucket and display them to the user.

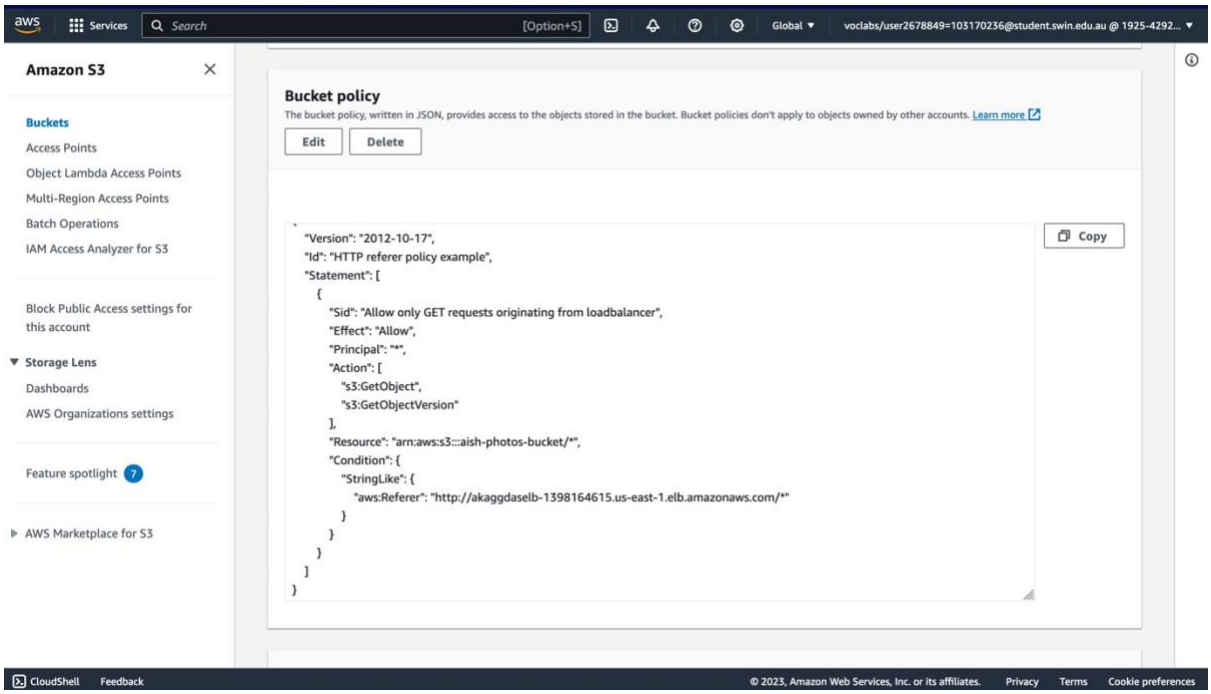

Figure 19 – aish-photos bucket policy

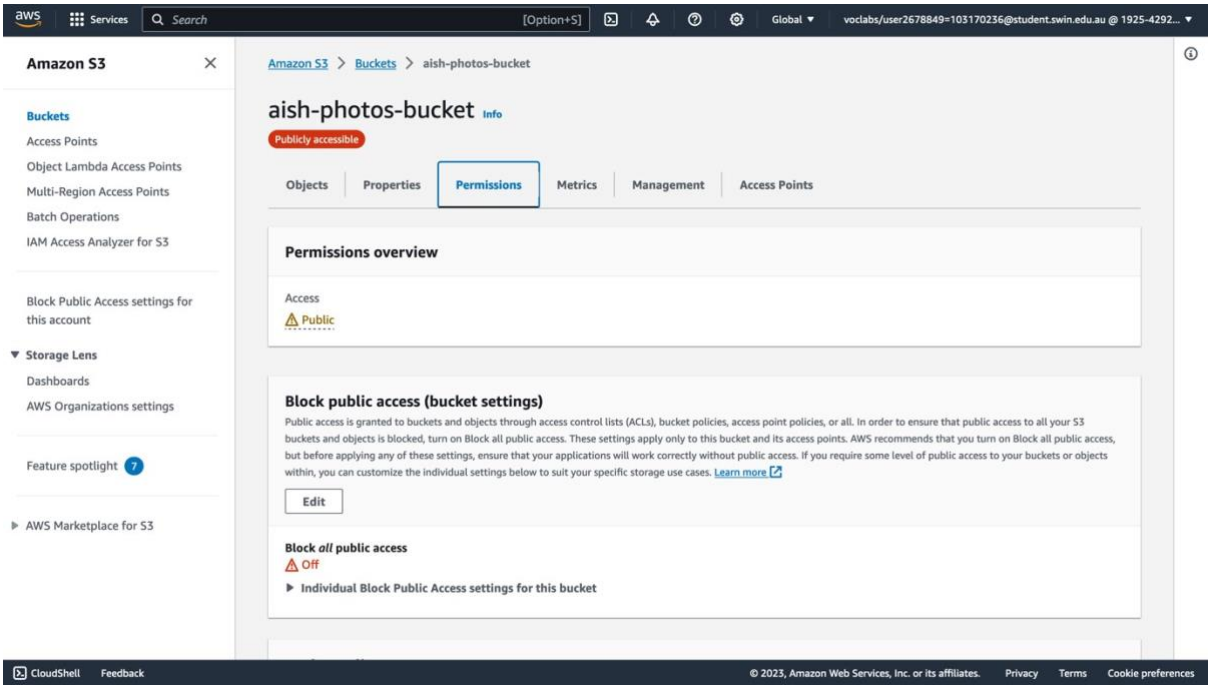

Figure 20 – aish-photos permissions

IX. Auto Scaling Group and Load Balancer

We create an AMI for Auto Scaling and name it DevServer AMI.

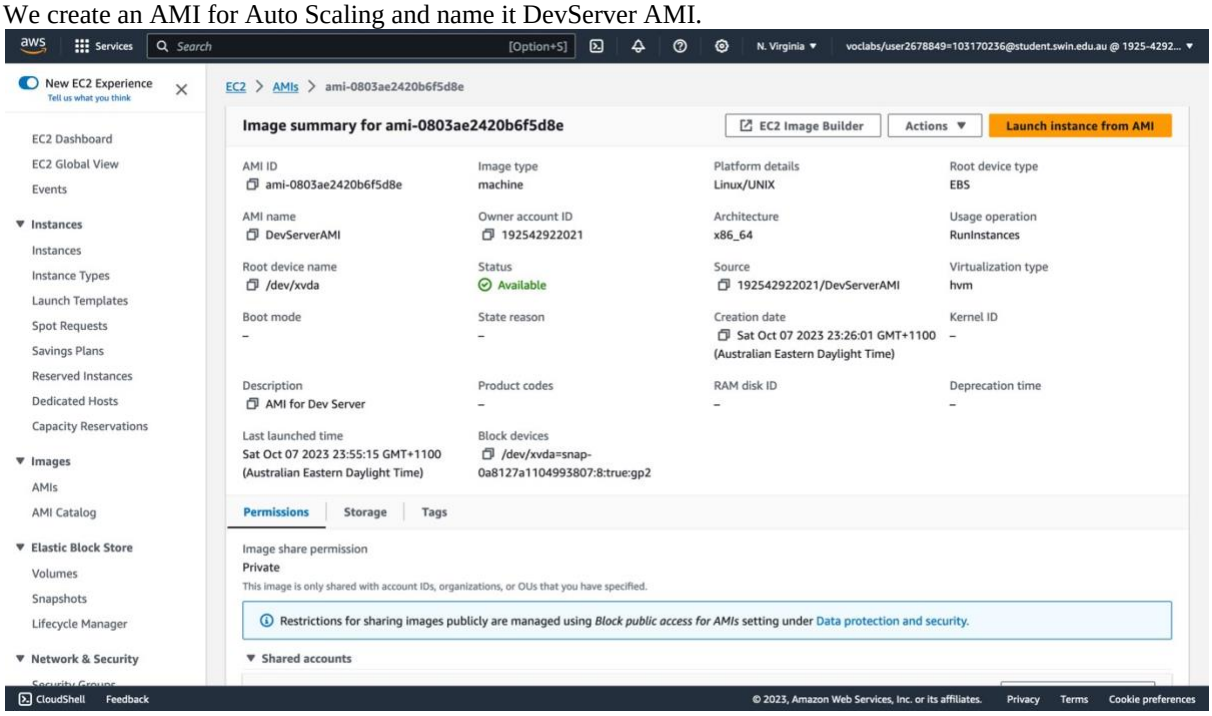

Figure 21 – DevServer AMI

We create target group. *Targets* are the individual instances that will respond to requests from the Load Balancer.

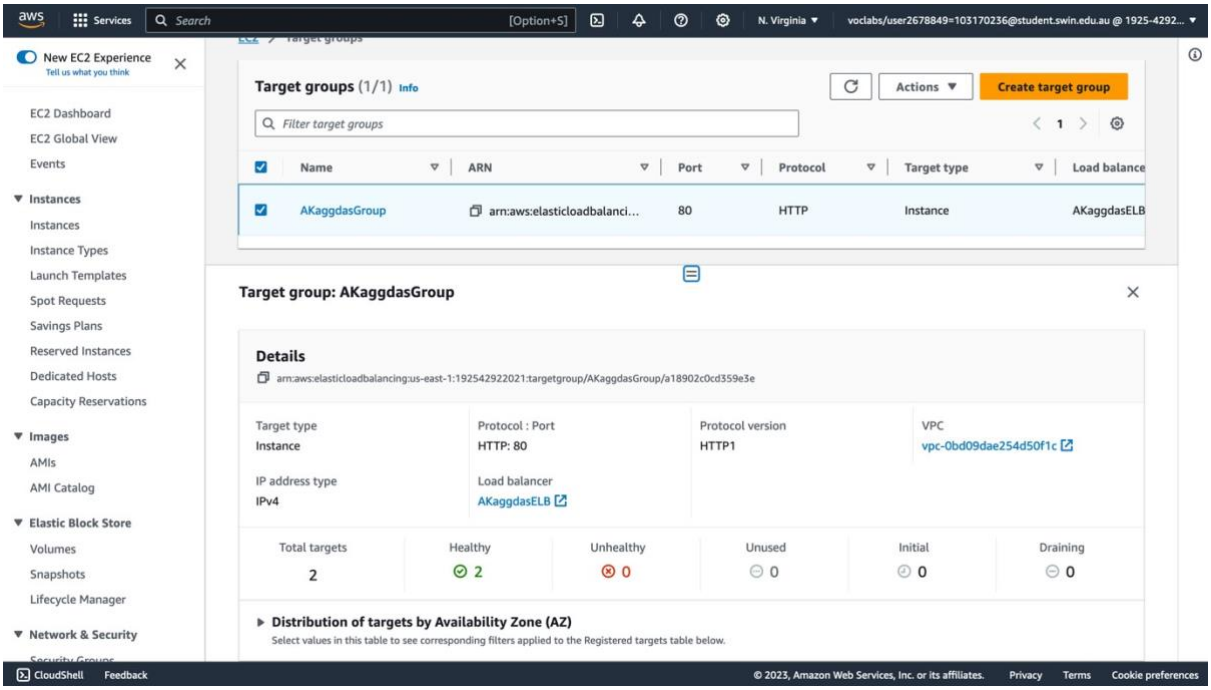

Figure 22: AKaggdasGroup Target Group

We name the load balancer 'AkaggdasELB'.

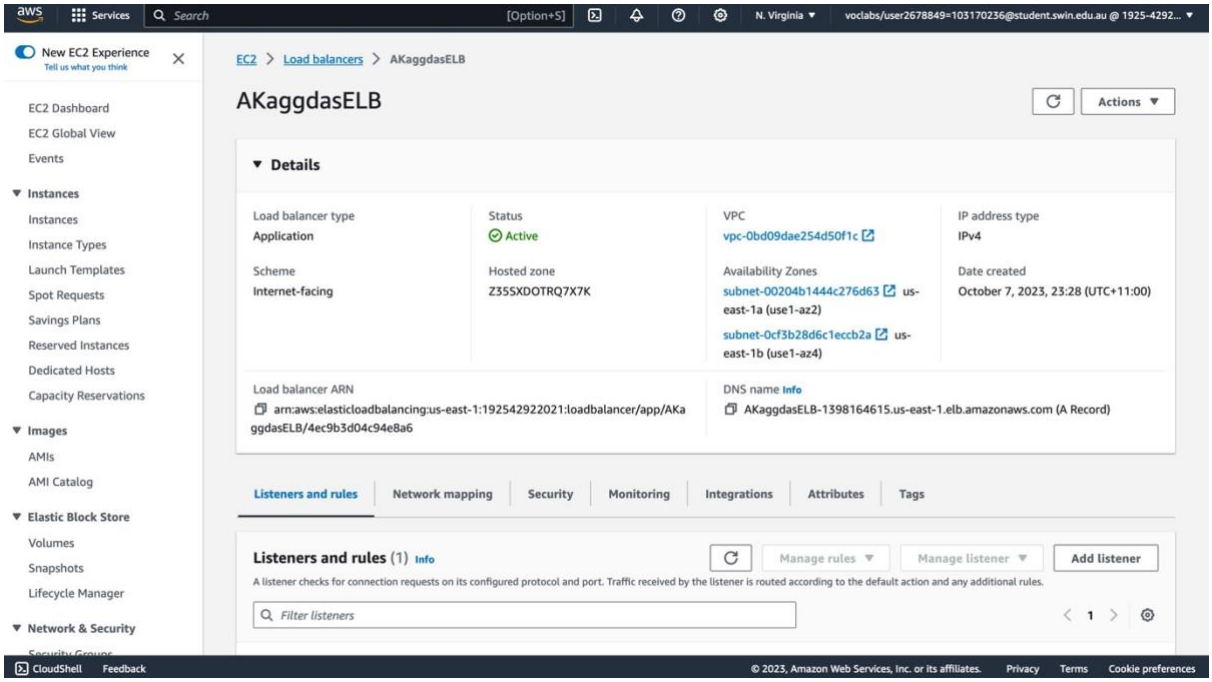

Figure 23 – AKaggdasELB load balancer

We will create a *launch template* for the Auto Scaling group and name it 'AKaggdasConfig '.

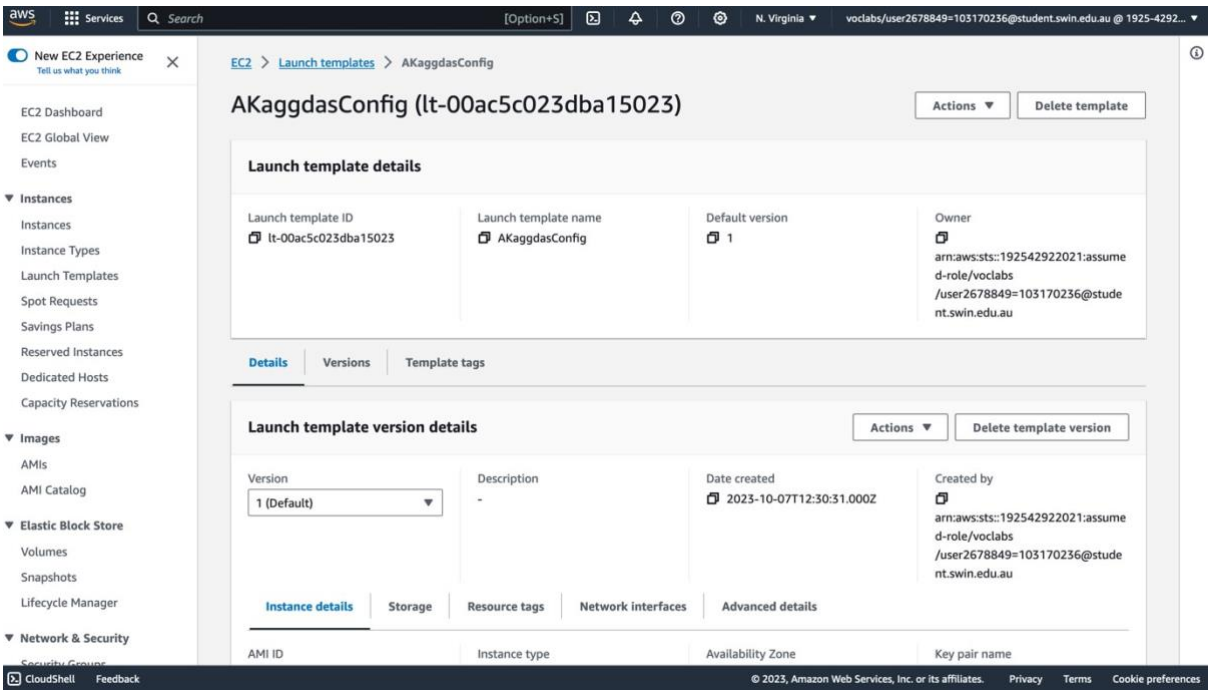

Figure 24 – AKaggdasConfig Launch Template

We name the auto scaling group ' AKaggdas'.

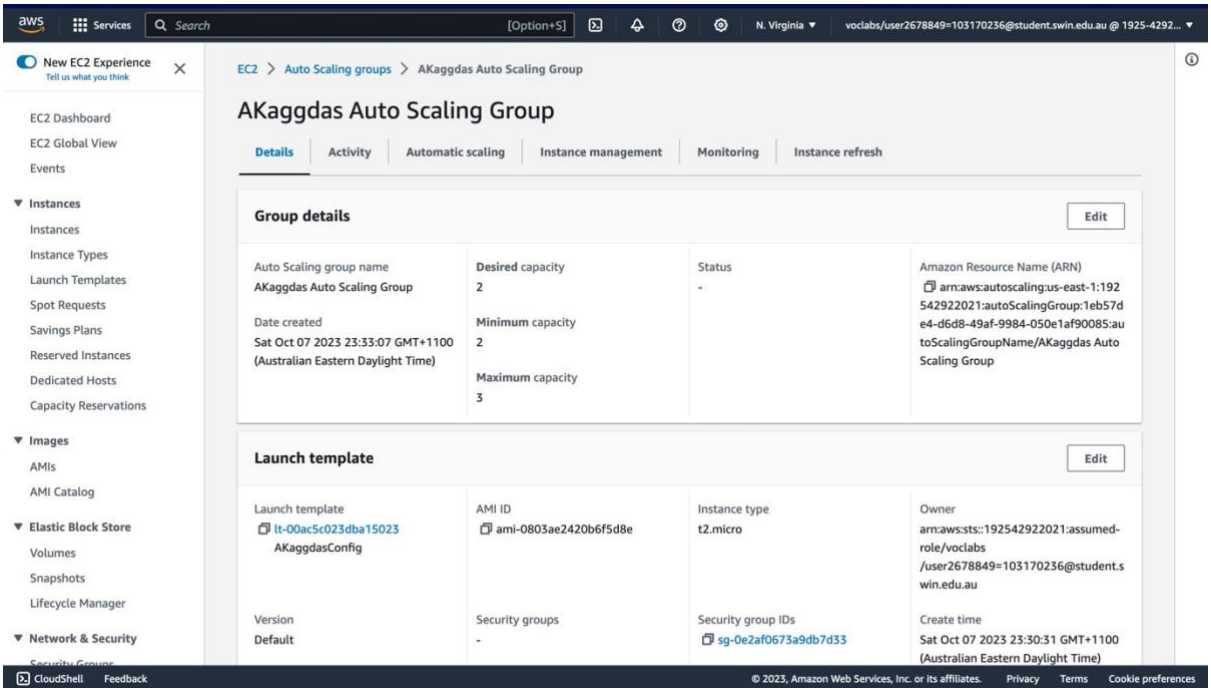

Figure 25 – AKaggdas Auto Scaling group

The auto scaling group launches web server instances, with a default of two and a maximum of three. Even if the web server instance is terminated, they will automatically be launched by autoscaling group.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Web servers	i-08a45c4a6cdabc440	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a
Dev server	i-053b5ce4d1900b125	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b
Web servers	i-01b66fb344563dd5b	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b
Nat instance	i-028201d815d19827d	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a

Figure 26: Web server instances

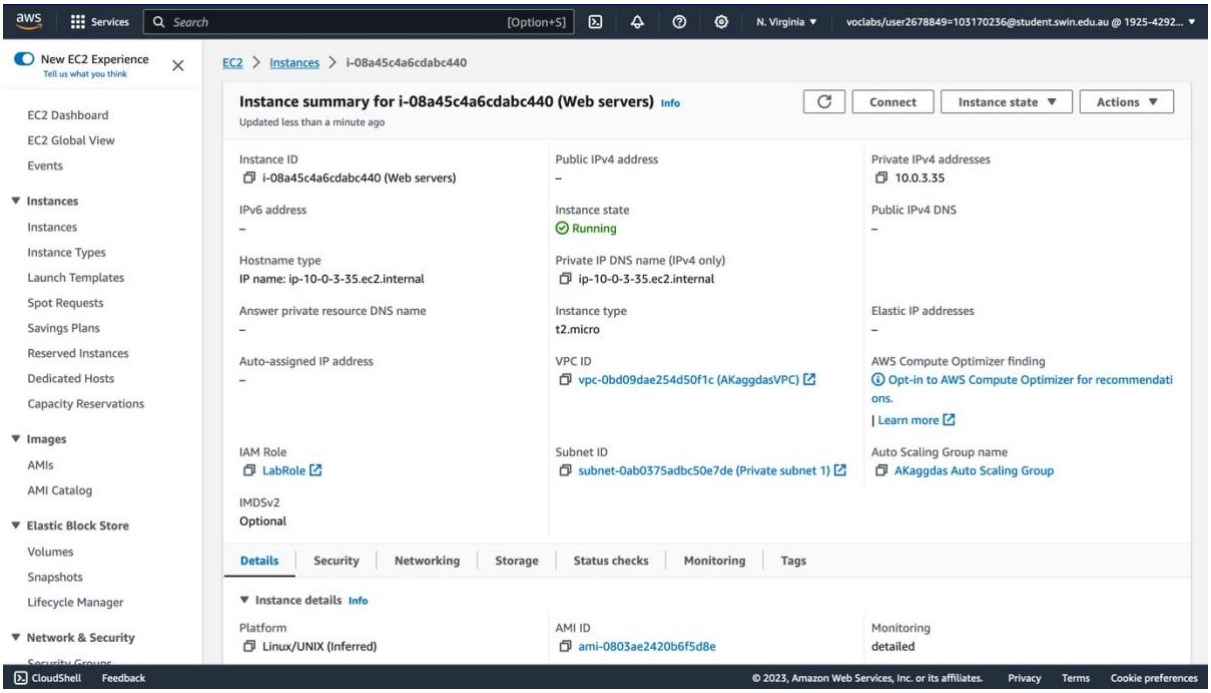

Figure 27: Summary of Web server instance

XI. IAM

The IAM role ‘LabRole’ is attached to the lambda function and automatically to the web servers when launched.

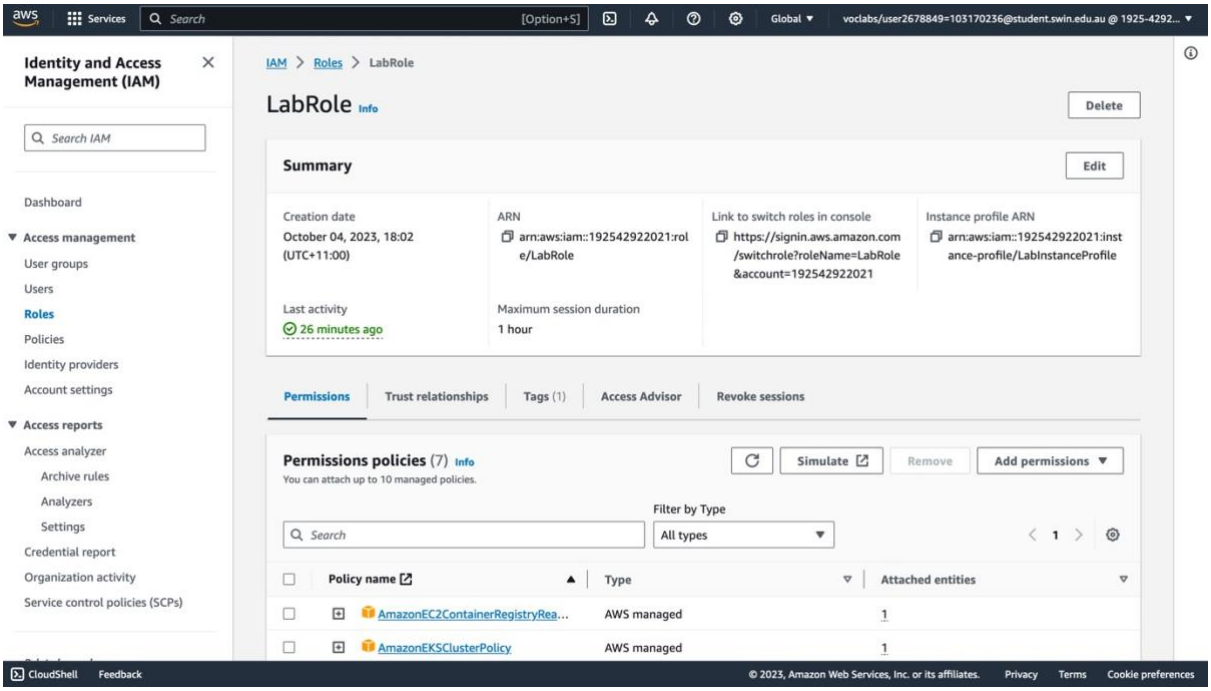

Figure 28: 'Lab Role ' IAM

XII. Lambda

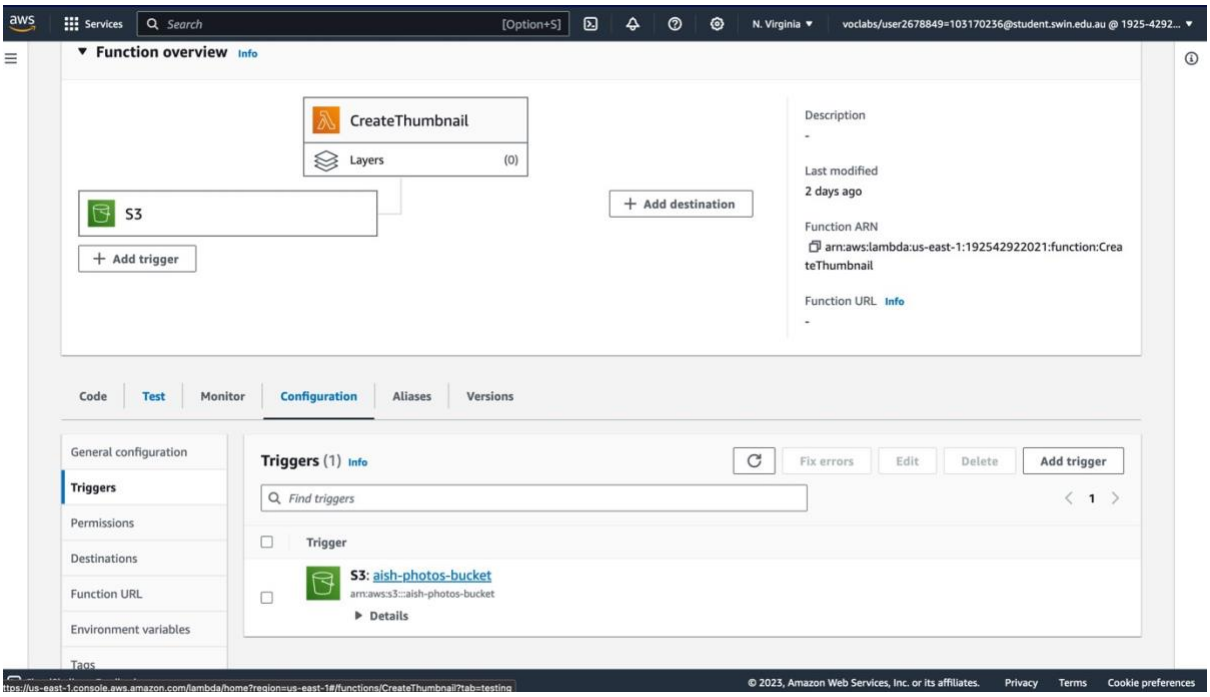

Figure 29: Function overview of Lambda 'CreateThumbnail'

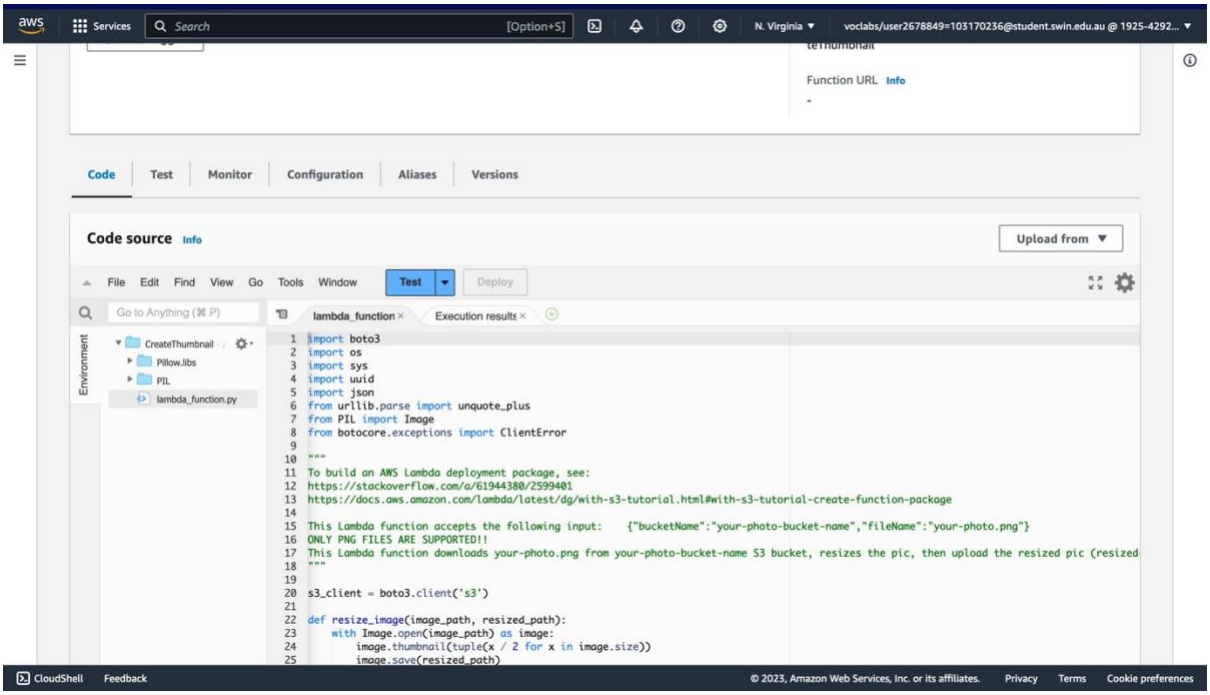

Figure 30: Lambda Function code

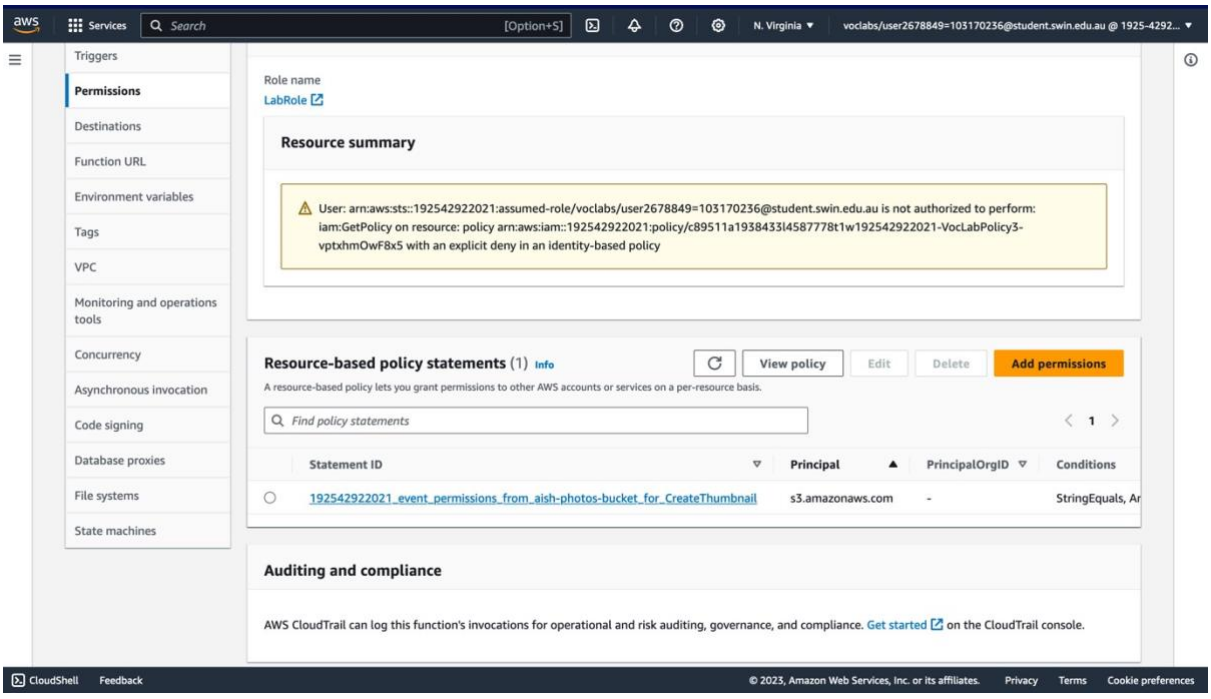

Figure 31: Lambda Permissions

XIV. Testing

- Check the S3 bucket to see if photos are actually uploaded and if their resized versions are created

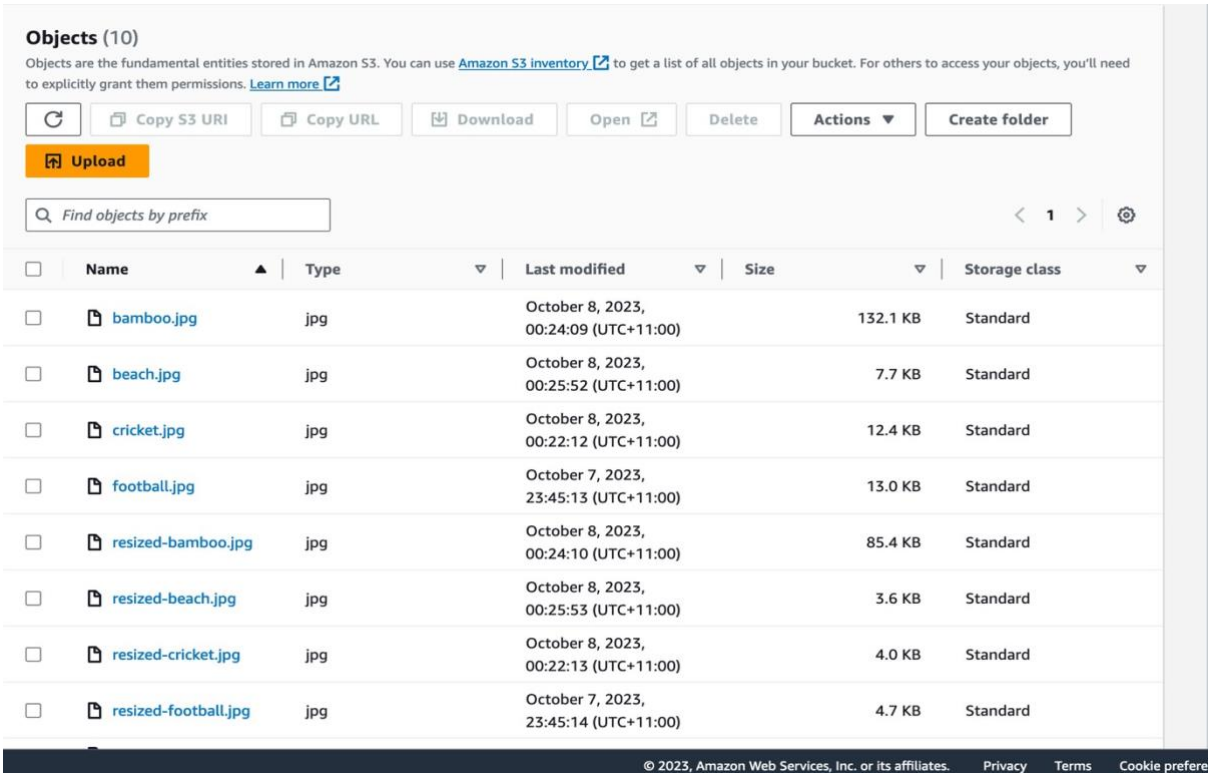

Figure 32 : S3 objects

- Check the database to see if their meta-data is recorded.

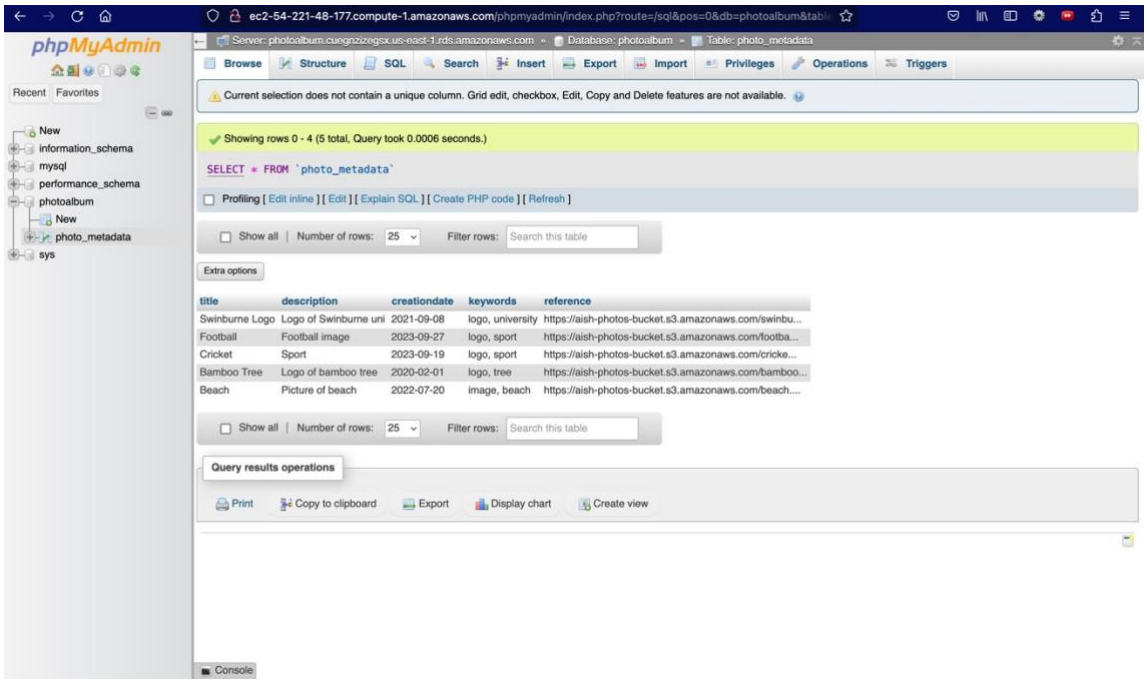

Figure 33: photo_metdata

- The PhotoAlbum website is accessible through the load balancer only.

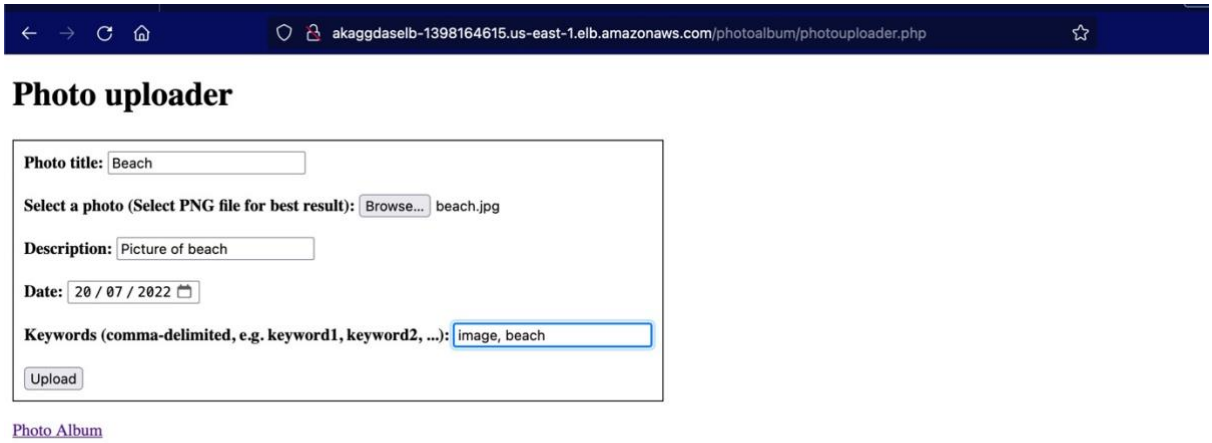

Figure 34: photouplaoder.php

URL of photouplaoder.php：*http://akaggdaselb-1398164615.us-east-1.elb.amazonaws.com/photoalbum/photouploader.php*

URL of album.php：*http://akaggdaselb-1398164615.us-east-1.elb.amazonaws.com/photoalbum/album.php*

▪

Figure 35:album.php

Terminate servers then check to see if replacement EC2 instances are automatically deployed by the ASG. Thoroughly test the functionality of the website again once new instances have been launched .

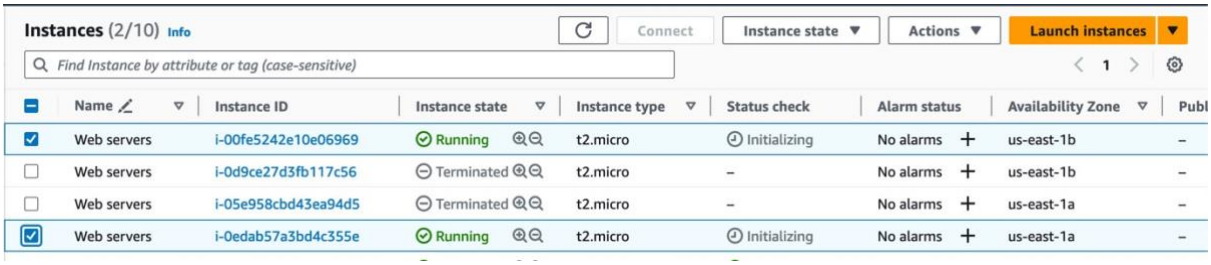

Figure 36:Web server instances terminated and launched again

▪ All EC2 targets are healthy.

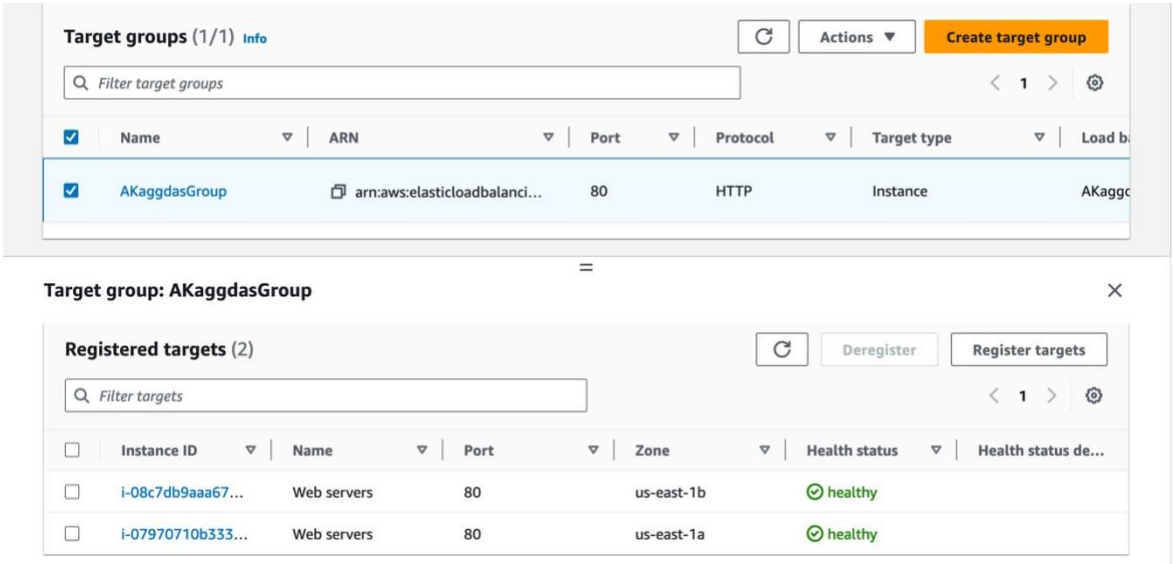

Figure 37 :'AKaggdasGroup' target

▪

Test direct access to your S3 photos, which should not be publicly accessible.

Figure 38 :Direct access to S3 photos denied