

Developing a highly available Photo Album website

Name: Shaugato Paroi
University: Swinburne University Of Technology
Stuent ID: 103523487
Tutorial: Friday 12:30pm-
2:30pm
ELB to website: [Photo Album](#)
Link To phpMyAdmin: [phpmyadmin](#)
Date of Submission : 05/05/2024

I. INTRODUCTION

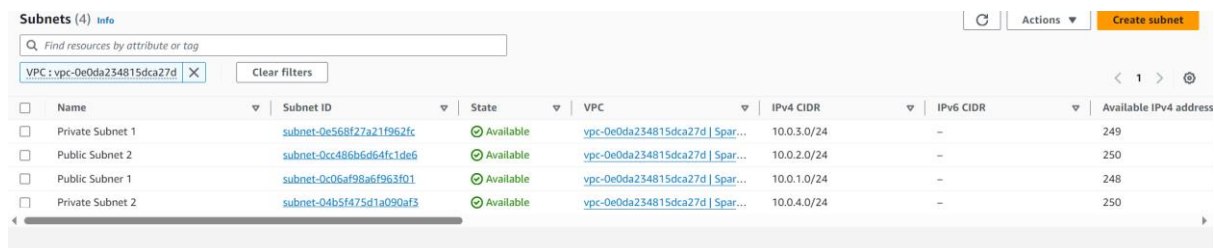
In this task, our task is to create a highly available website that will display the photos . This will extend from Assignment 1-b. In this task we will be using IAM to create EC2 and it will enable us to interact with other services like Lambda and S3. We will use as S3 bucket policy when creatin the S3 bucket in order to restrict public access. We will also uses custom AMI to launch our custom template Besides we will be using Elastic Load Balancer , Load Balancer . These will distribute all the incoming traffic in all the available zones by ensuring high availability throughout the available zone. Furthermore we will also be using a Nat Gateway , and this Nat Gateway will allow private instance to access the internet . We will also be creating NACL and security groups to allow access to only the necessary traffic to pass.

This report with describe the process how each of the steps can be implemented.

II. CREATING THE VPC

The first step is to create the VPC called “SParoiVPC” in the N.Virginia us-east-1

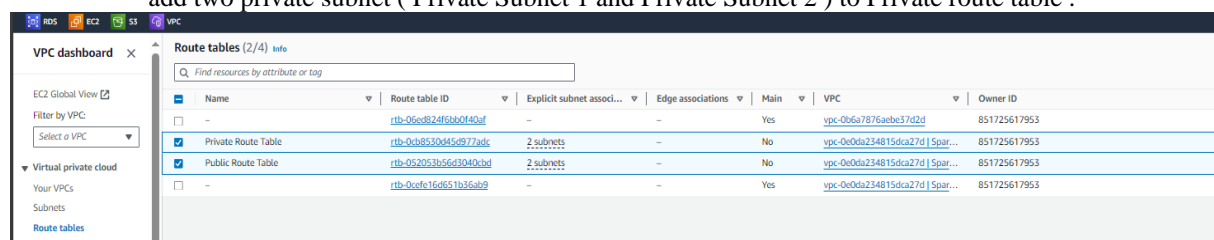
- 1.Go to VPC and create the VPC using “VPC only”. The CIDR block will be 10.0.0.0/16 as given in the assignment
- 2.Then we will create four subnets by going to the subnets: Public Subnet 1 - 10.0.1.0/24, Public subnet 2 - 10.0.2.0/24, Private Subnet 1 -10.0.3.0/24, Private Subnet 2 - 10.0.4.0/24



Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 address
Private Subnet 1	subnet-0e568f27a21f962fc	Available	vpc-0e0da234815dca27d Spar...	10.0.3.0/24	-	249
Public Subnet 2	subnet-0cc486b6d64fc1de6	Available	vpc-0e0da234815dca27d Spar...	10.0.2.0/24	-	250
Public Subnet 1	subnet-0c06af98a6f963f01	Available	vpc-0e0da234815dca27d Spar...	10.0.1.0/24	-	248
Private Subnet 2	subnet-04b5f475d1a090af3	Available	vpc-0e0da234815dca27d Spar...	10.0.4.0/24	-	250

Figure 1- Creating Four Subnets

3. Go to the Route table and create the two route table in two zones us-east-1a, and us-east-1b (Public Route Table, Private Route Table) . Then add two public subnet (Public Subnet 1 , Public subnet 2) to Public Route Table and add two private subnet (Private Subnet 1 and Private Subnet 2) to Private route table .



Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
-	rtb-06c824ff6b60f40af	-	-	Yes	vpc-0b6a7876aebc37d2d	851725617953
Private Route Table	rtb-0cb8530645d577ade	2 subnets	-	No	vpc-0e0da234815dca27d Spar...	851725617953
Public Route Table	rtb-052053b56d93040cd	2 subnets	-	No	vpc-0e0da234815dca27d Spar...	851725617953
-	rtb-0acfe16d651b36ab9	-	-	Yes	vpc-0e0da234815dca27d Spar...	851725617953

Figure 2 - Creating Route Tables

4. Go to the Internet Gateways and create an Internet Gateways called “Sparoi Internet Gateway” and attach it to the “SParoiVPC” Public route table which allows resources in public transports to internet

5. Go to the Nete Gateways and create Nat Gateway called “SPAROI NAT Gateway” with the following specifications
: vpc : SparoiVPC, subnet: Public Subnet 1

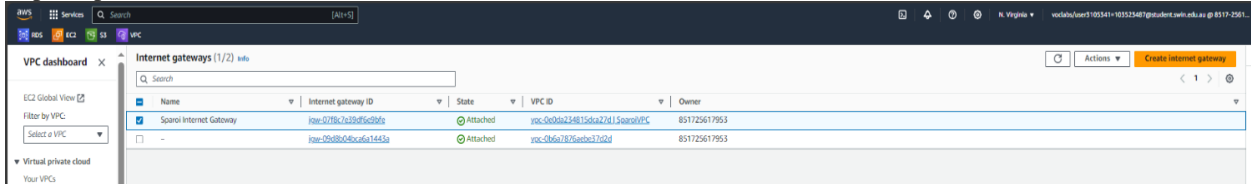


Figure 3 - Creating Internert Gateway

nat-0a7152959010f25d3 / SPAROI NAT Gateway

Details	Secondary IPv4 addresses	Monitoring	Tags
Details			
NAT gateway ID nat-0a7152959010f25d3	Connectivity type Public	State Available	State message -
NAT gateway ARN arn:aws:ec2:us-east-1:851725617953:natgateway/nat-0a7152959010f25d3	Primary public IPv4 address 54.174.168.116	Primary private IPv4 address 10.0.1.247	Primary network interface ID eni-0887fecb6195905b9
VPC vpc-0e0da234815dca27d / SparoiVPC	Subnet subnet-0c06af98a6f963f01 / Public Subnet 1	Created Wednesday, May 1, 2024 at 07:56:23 GMT+10	Deleted -

Figure 4 - SPAROI NAT Gateway details

vpc-0e0da234815dca27d / SparoiVPC

Details	Resource map	CIDRs	Flow logs	Tags	Integrations
Details					
VPC ID vpc-0e0da234815dca27d	State Available	DNS hostnames Disabled	DNS resolution Enabled		
Tenancy Default	DHCP option set dopt-0be513336d47656ce	Main route table rtb-0cfe16d651b36ab9	Main network ACL acl-09e8af41c129b24de		
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -		
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups Failed to load rule groups	Owner ID 851725617953			

Figure 5 - VPC details

vpc-0e0da234815dca27d / SparoiVPC

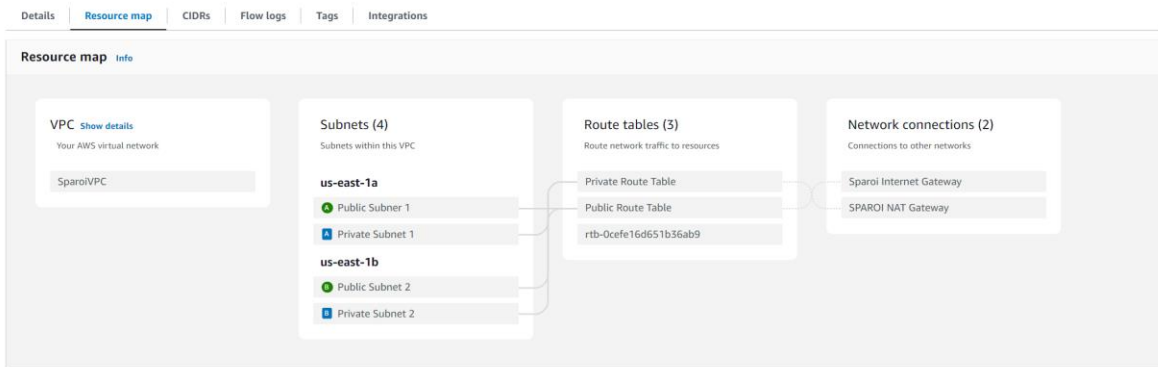


Figure 6 - SparoiVPC resource map

```

[ec2-user@ip-10-0-4-62 ~]$
[ec2-user@ip-10-0-4-62 ~]$
[ec2-user@ip-10-0-4-62 ~]$
[ec2-user@ip-10-0-4-62 ~]$
[ec2-user@ip-10-0-4-62 ~]$
[ec2-user@ip-10-0-4-62 ~]$
[ec2-user@ip-10-0-4-62 ~]$
[ec2-user@ip-10-0-4-62 ~]$ ping google.com
PING google.com (142.251.111.100) 56(84) bytes of data.
64 bytes from bk-in-f100.1e100.net (142.251.111.100): icmp_seq=1 ttl=5
64 bytes from bk-in-f100.1e100.net (142.251.111.100): icmp_seq=2 ttl=5
64 bytes from bk-in-f100.1e100.net (142.251.111.100): icmp_seq=3 ttl=5
64 bytes from bk-in-f100.1e100.net (142.251.111.100): icmp_seq=4 ttl=5
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 2.051/2.211/2.602/0.231 ms
[ec2-user@ip-10-0-4-62 ~]$ ping 10.0.2.79
PING 10.0.2.79 (10.0.2.79) 56(84) bytes of data.

```

Figure 7 - Nat Gateway is working

III. CREATING SECURITY GROUPS

As per the assignment requirement, we will create security groups by going to Security groups and on VPC dashboard.

1. WebServerSG – Web server Security Group

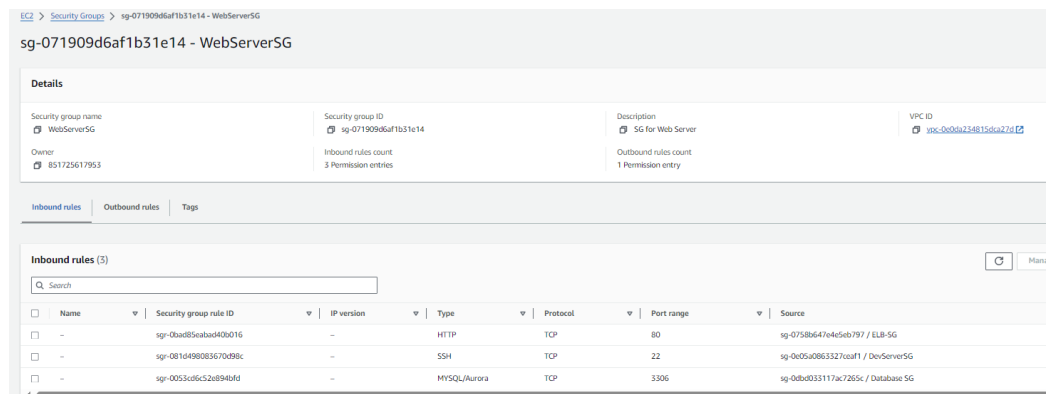


Figure 8 - WebServerSG details

2. DevServerSG – Dev Server Security Groups .

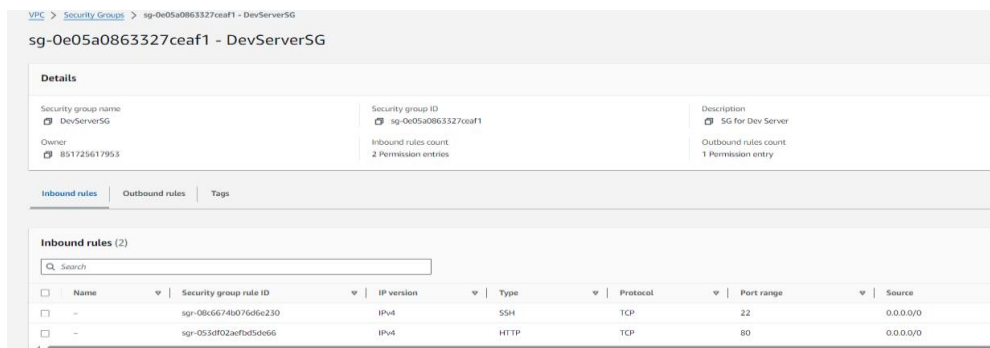


Figure 9 - DevServer Security Groups

3. DataBase SG – Data Base Server Security Groups

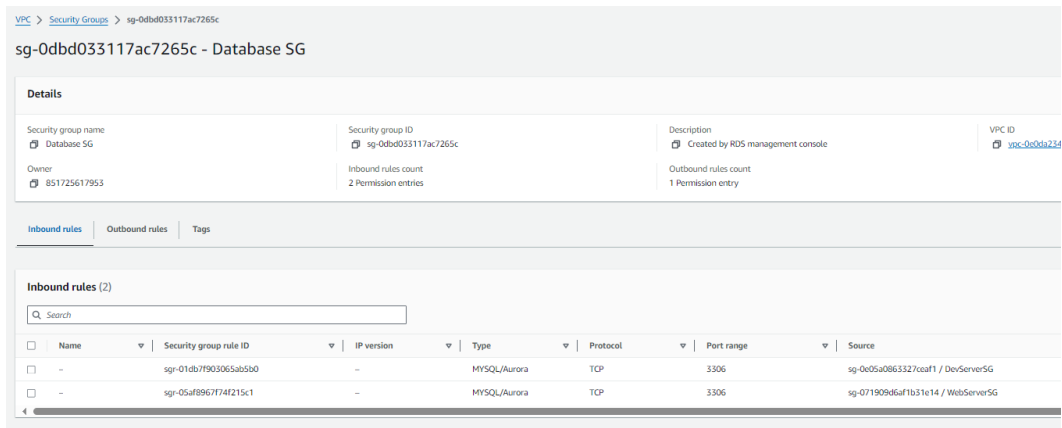


Figure 10 - DataBase Security Group

4. ELB-SG – Elastic Load Balancer Security Groups

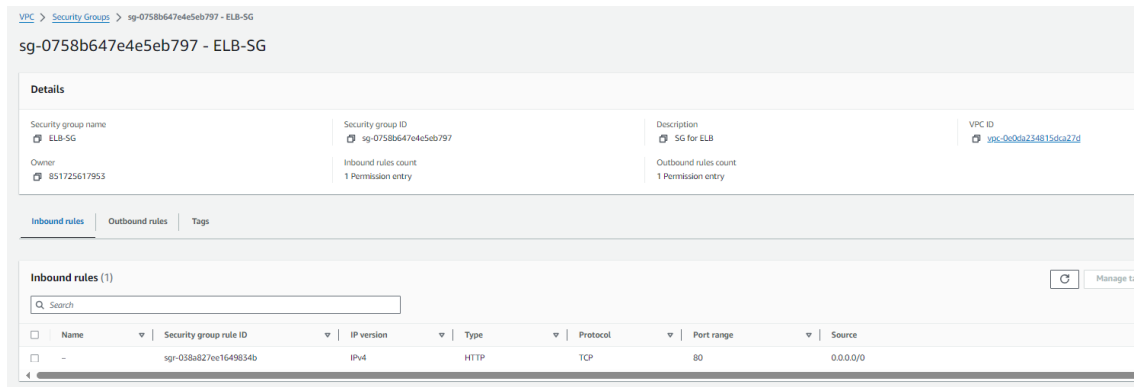


Figure 11 - ELB Security Groups

IV. CREATING DEV SERVER (EC2)

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

Dev Server: This will be used to create The AMIs “SparoiAMI” later and the AMIs will launch template for webserver. The Webserver will be created by The Auto Scaling Group in order to ensure High Availability for the phpMyAdmin website. DevServer will have phpMyAdmin installed, so web server will also have phpMyAdmin installed as webserver will be created by the “SparoiAMI” and SParoiAMI” is created based on the DevServer.

1. Go to EC2 → Instance → Launch Instance
2. Edit the field accordingly : Name - DevServer , Amazon Machine Image (Amazon Linux 2 AMI Kenal 5), Instance type - t2.micro, Key pair - Baston_web (ppk, by creating a new key pair)
3. Edit Network Setting Accordingly : VPC - SparoiVPC , Subnet - 10.0.2.0/24 (Public Subnet 2 as per the requirements) , Firewall - WebServerSG (choosing it by clicking Select Existing Security Groups)
4. As per the requirements , in order to install the Apache web server and other php package , we put the following codes on the user data section in Advanced details . Also select “LabInstanceProfile” in IAM setting. Then Launch instance.

```

#!/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
sed -i "s/upload_max_filesize = 2M/upload_max_filesize = 10M/g" /etc/php.ini
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

```

Figure 12- Apache Server and PHP installation Code

5. Allocate an Elastic Ip address and attach it to the DevServer. The elastic Ip address is **44.209.21.246**
6. Then I download AWS PHP SDK package into /var/www/html/photoalbum path inside DevServer instance and modify the constant.php according to my Database Setting , S3 bucket name and lambda arn

```

*      mydb.php                Interact with RDS DB
*      photo.php              Photo object class
*      photouploader.php      (executable) upload image to S3 and RDS
*      photouploadtemplate.html HTML template for the photo uploading function
*      utils.php              some supporting methods
*
*
* The values of the constant variables with "[ACTION REQUIRED]" in the comment must be updated. The current values are just examples.
* You need to replace the values of those constant variables with values specific to your setup.
*
* ===== READ THE ABOVE !!! =====
* ===== READ THE ABOVE !!! =====
* ===== READ THE ABOVE !!! =====
*/

// [ACTION REQUIRED] your full name
define('STUDENT_NAME', 'Shaogato Parol');
// [ACTION REQUIRED] your Student ID
define('STUDENT_ID', '189523487');
// [ACTION REQUIRED] your tutorial session
define('TUTORIAL_SESSION', 'Friday 12:30AM');

// [ACTION REQUIRED] name of the S3 bucket that stores images
define('BUCKET_NAME', 'sparoi-bucket');
// [ACTION REQUIRED] region of the above bucket
define('REGION', 'us-east-1');
define('S3_BASE_URL', 'https://sparoi-bucket.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', 'photoalbum');
// [ACTION REQUIRED] endpoint of RDS instance
define('DB_ENDPOINT', 'sparoi-db.cimgp6suwrl.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', 'albumphoto');
// 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');
?>

```

Figure 13 - Dev Server Constance.PHP in var/www/html/photoalbum/constance.php

/var/www/html/					
Name	Size	Changed	Rights	Owner	
..		2/05/2024 4:08:56 PM	rw-rw-r--	ec2-user	
aws		2/05/2024 4:24:57 PM	rw-rw-r--	ec2-user	
photoalbum		2/05/2024 4:22:32 PM	rw-rw-r--	ec2-user	
phpmyadmin		2/05/2024 4:26:07 PM	rw-r--r--	ec2-user	
aws.zip	5,810 KB	2/05/2024 5:09:51 AM	rw-rw-r--	ec2-user	
phpinfo.php	1 KB	2/05/2024 4:09:04 PM	rw-r--r--	root	
phpMyAdmin-5.2.1-en...	10,204 KB	14/12/2023 5:34:17 AM	rw-rw-r--	ec2-user	

Figure 14 - Dev Server Directory Will All Components

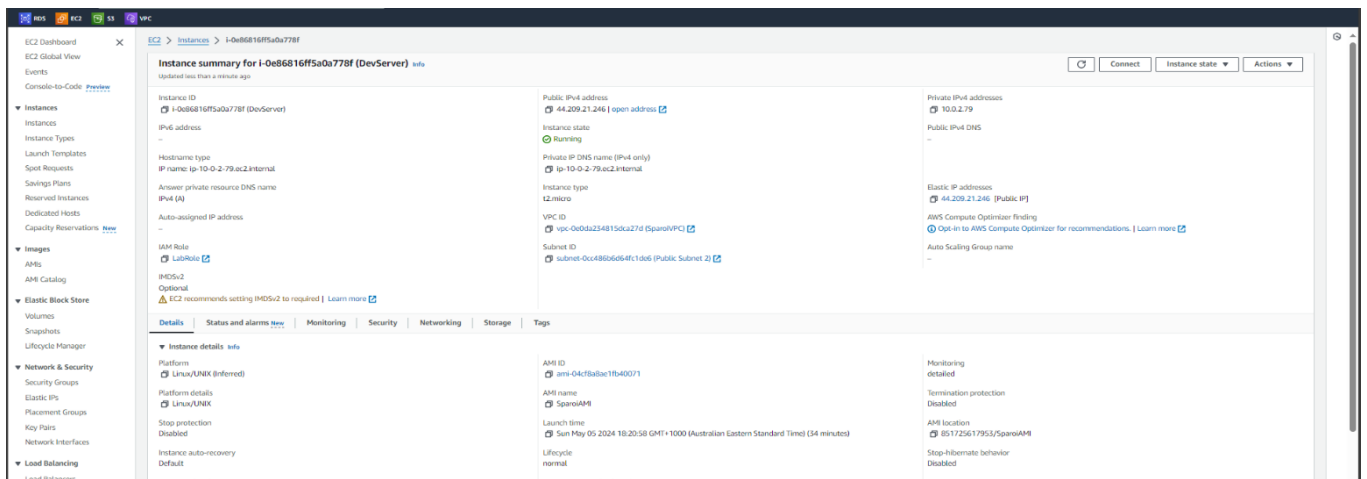


Figure 15 - DevServer Instance

After creating the DevServer instance successfully I created the AMIs “SparoiAMI, which will be used by Auto Saling group to deploy WebServer

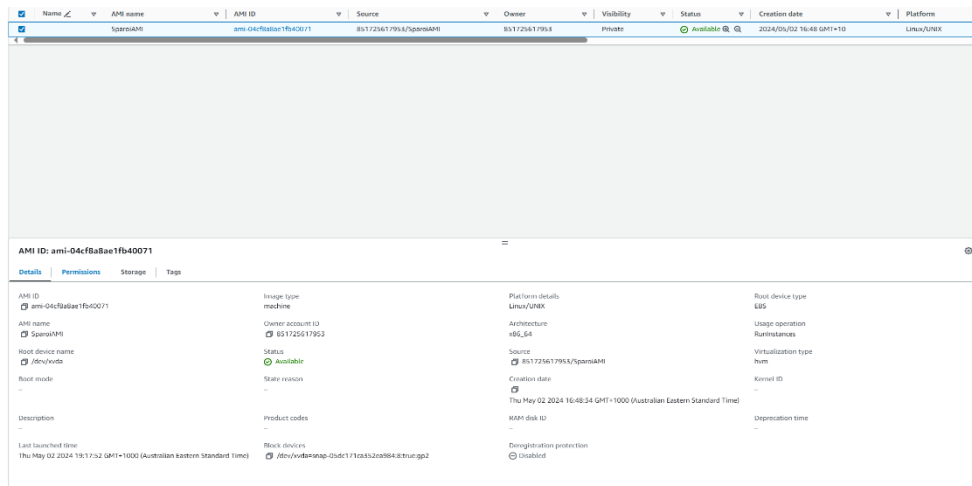


Figure 16 - SparoiAMI instance

V. CREATING TEMPLATE

Now I create a template “Sparoi-LT” using the t2.micro instance type, Dev_Server_SparoiKey and using the “LabInstanceProfile” IAM profile setting

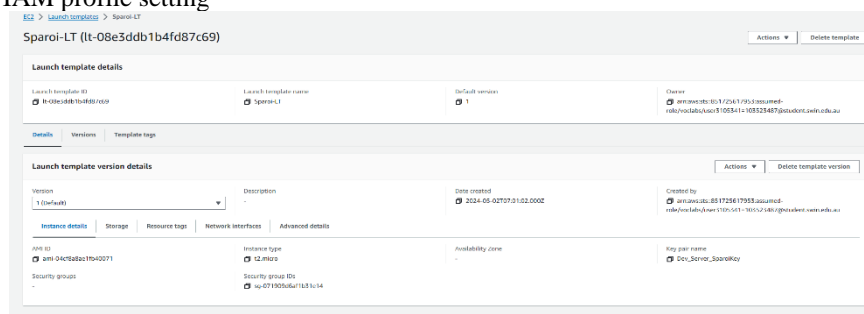


Figure 17 - Sparoi-LT instance

VI. CREATING RDS DATABASE INSTANCE

A. Creating Subnet Group For RDS instance

Before Creating the RDS Database , I create a Subnet Group called “sparoi subnet group” so that I can attach it later to the rds instance

- Go to the Subnet Group from RDS dashboard and click Create DB subnet Group

- Edit the field like the following : Name - sparoi subnet group, VPC - SParoiVPC, Availability Zone - us-east-1a, useast-1b , subnets - 10.0.3.0/24,10.0.4.0/24. Then Create.

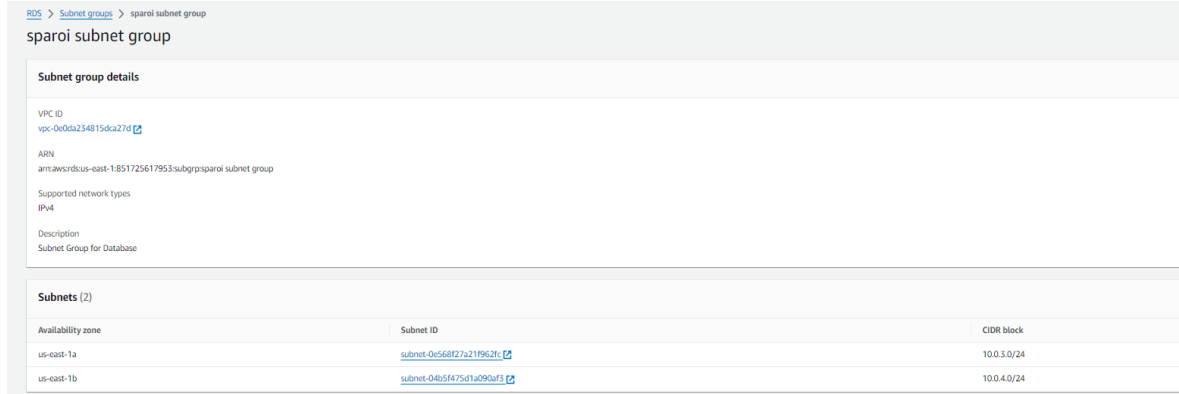


Figure 18 - creating sparoiin subnet group

B. Creating RDS Database Instance

1. From RDS service on Databases option, Click on Create Database
2. Edit the fields like the following : Database creation method : Standard Create, Engine - MySQL, Engine Version - MySQL 8.0.34, Templates - Free Tier, DB cluster identifier – sparoi-db , Credential Setting - Self-managed , Master Username - admin, master password - admin123 ,
3. On connectivity: Virtual Private Cloud - SparoiVPC, DB Subnet group - sparoi subnet group, VPC security groups - DataBase SG, Public Access: No. Then Create Database . Database can be accessible through DevServer or Elastic Load balancer by the url : phpmyadmin
4. For Installing phpmyadmin I follow the same steps as the Assignment 1-b

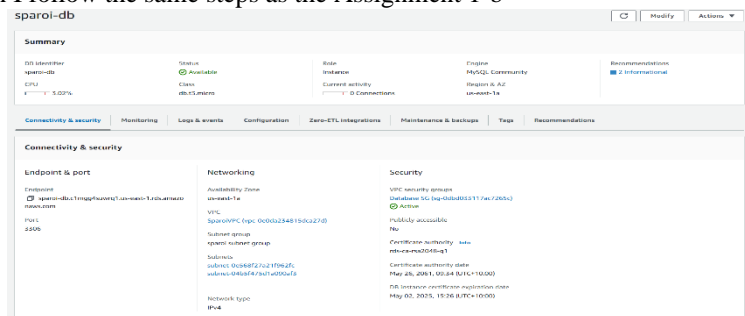


Figure 19 - RDS Instance Details

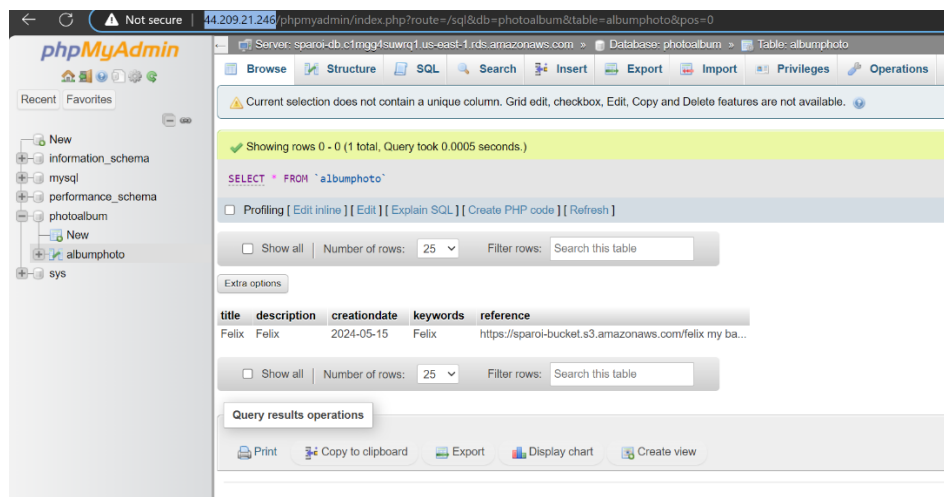


Figure 20 - Database Recors Access Through Dev Server Elastic IP

VII. CREATING TARGET GROUP

I create a Target group called Sparoi-TG by editing the specification like the following : Target Type = Instances, Target Group Name= Sparoi-TG, Protocol = HTTP, Health Check Path = /photoalbum/album.php

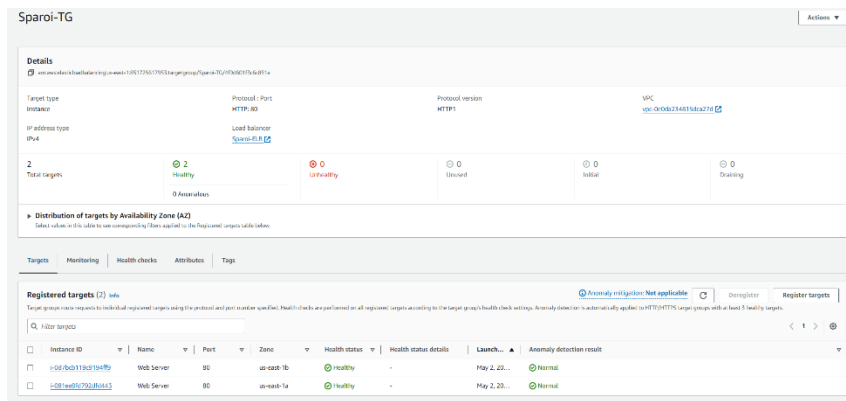


Figure 21 - Creating Sparoi-TG security groups

VIII. CREATING LOAD BALANCER

I create a Load Balancer “Sparoi-ELB”, by editing the specification like the following : Selecting Application Load Balancer, “Sparoi-ELB”, VPC = SparoiVPC, Mappings: us-east-1a (Public Subnet 1), us-east-1b (Public Subnet 2), Security groups : ELB-SG, Protocol (HTTP , port 80) default Action forward to Sparoi-TG.

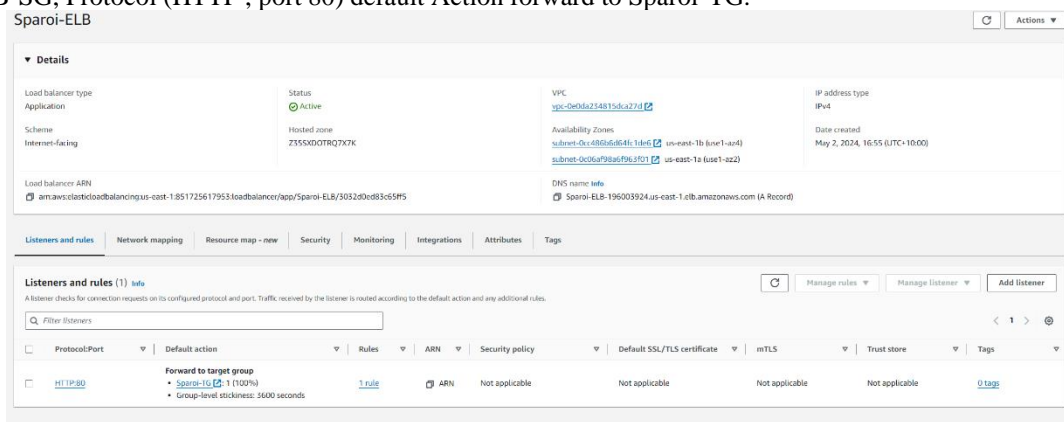


Figure 22 - Load Balancer Sparoi-ELB config

IX. CREATING AUTO SCALING GROUP

I will be creating a Auto Scaling Group Called “Sparoi-ASG” using the following specifications : Name : “Sparoi-ASG”, Launch Template = “Sparoi-LT”, VPC – SparoiVPC, Availability Zones: us-east-1a(10.0.3.0/24), us-east-1b(10.0.4.0/24), Existing Load Balancer, Target groups = Sparoi-TG, Enable Cloud Watch, Desired Capacity 2, Min Capacity 2 and Max Capacity 3, Enable Target Tracking Policy, Metric Type = Application Load Balancer request count per target , Target Group – Sparoi TG, Target Value = 30

Then create it .

After Creating it , in order to test it , I also terminate one of the Webserver instance , and then ensure extra webserver instance successfully created and also status shown as “healthy”.

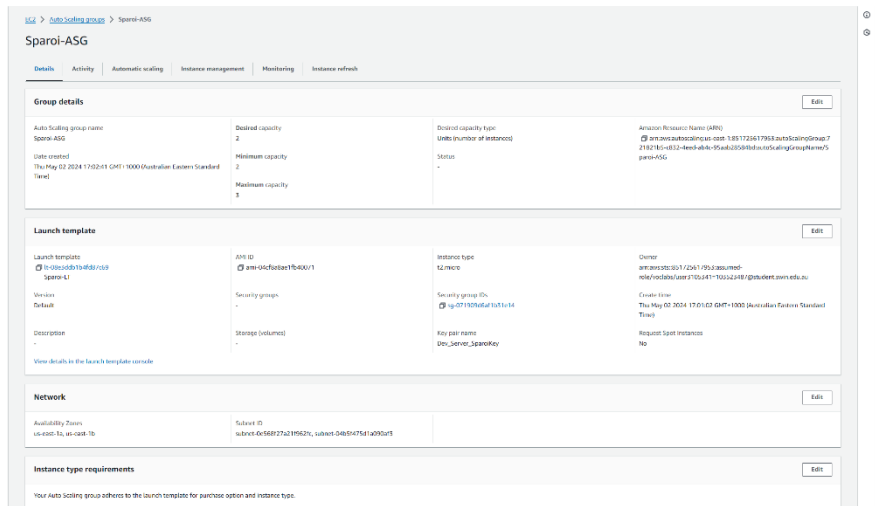


Figure 23 - Sparoi-ASG auto Scaling Group

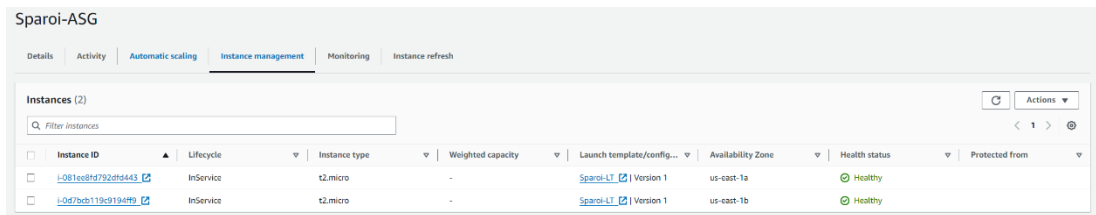


Figure 24 - Intances Launce By-ASG are shown Healthy

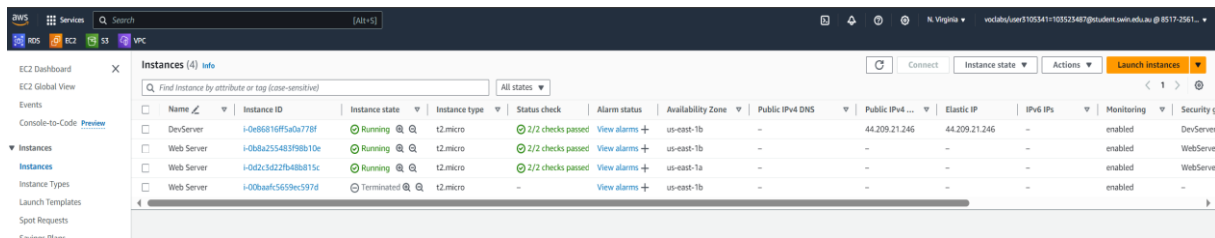


Figure 25 - Terminating One Webserver And Successfully Created Another Wevserver

X. CREATING AND SETTING UP S3 BUCKET

For creating a S3 bucket called “sparoi-bucket” I took the following steps:

1. Go to Amazon S3 service , click on Buckets then click “create bucket”
2. Edit the field like this : Bucket type - General , Bucket name – sparoi-bucket, enable public access by unchecking “ Block all public address “ , region - US East (N. Virginia) us-east-1. Then create bucket

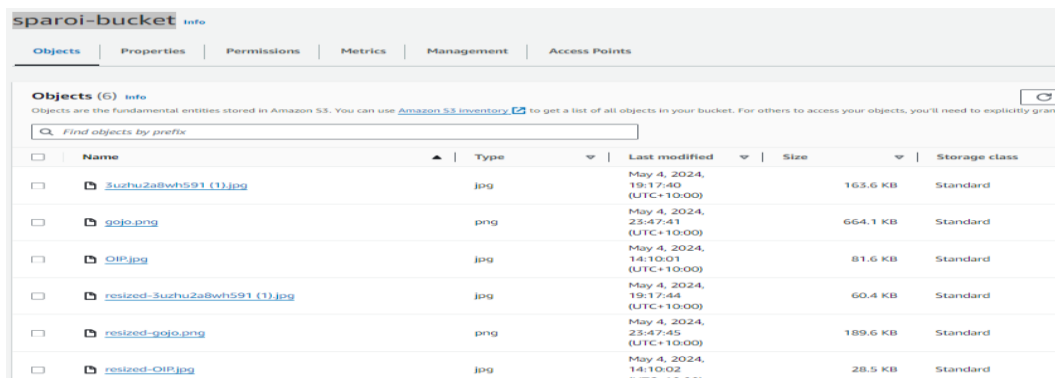


Figure 26 - Creating the sparoi-bucket

3. After creating the bucket I added a “bucket policy” which will restrict the access of the bucket and will only allow request for elastic load balancer

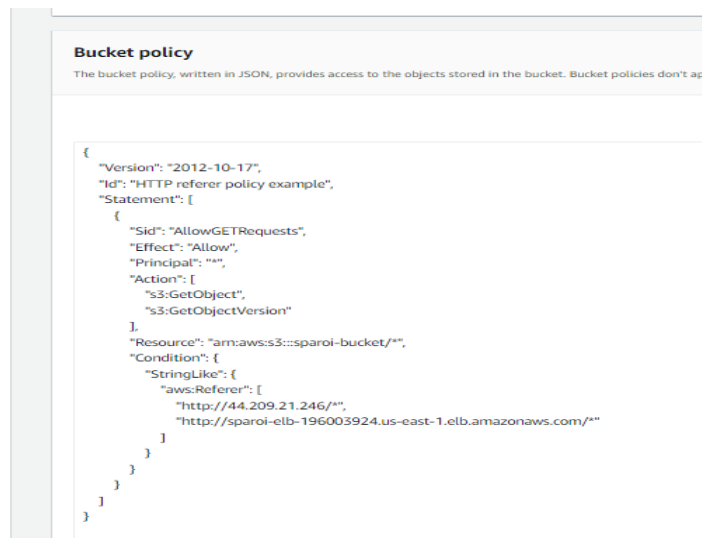


Figure 27 - Creating the Bucket Policy

XI. CREATING AND SETTING UP LAMBDA FUNCTION

We Create a called “CreateThumbnail” by going to “Lambda” Service . I click create fucting and then edit the following details: Function name = “CreateThumbnail”, Runtime = Python 3.11, Architecture = x86_64, Existing role = Labrole. After creating the function I uploaded the provided “lambda-deployment-package-0.1.zip” to modify the function.

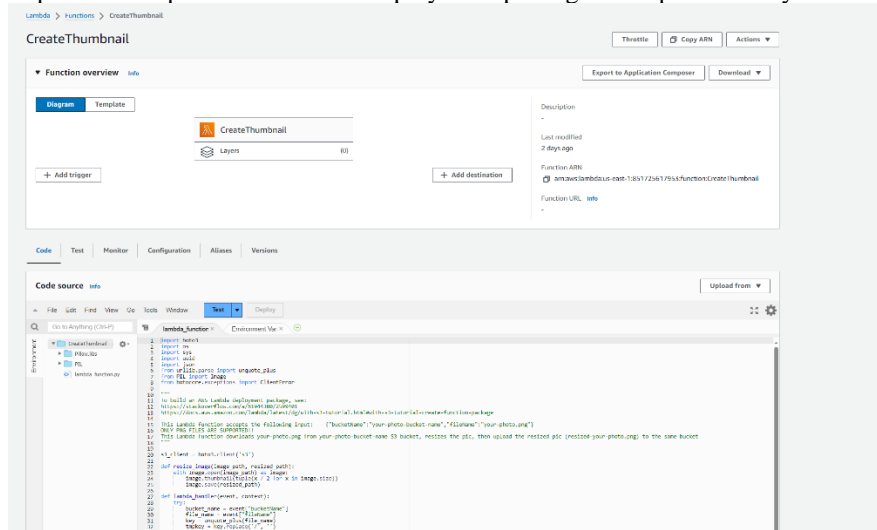


Figure 28 - Modifying Lambda Function and Code

XII. USING IAM ROLE



Figure 29 - Using Labrole in DevServer

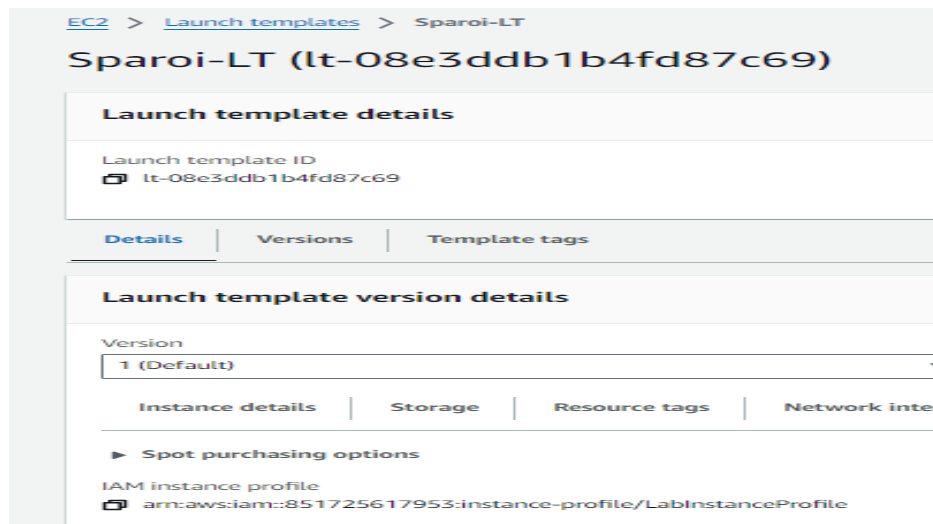


Figure 30 - Using LabInstanceRole in Launce Template

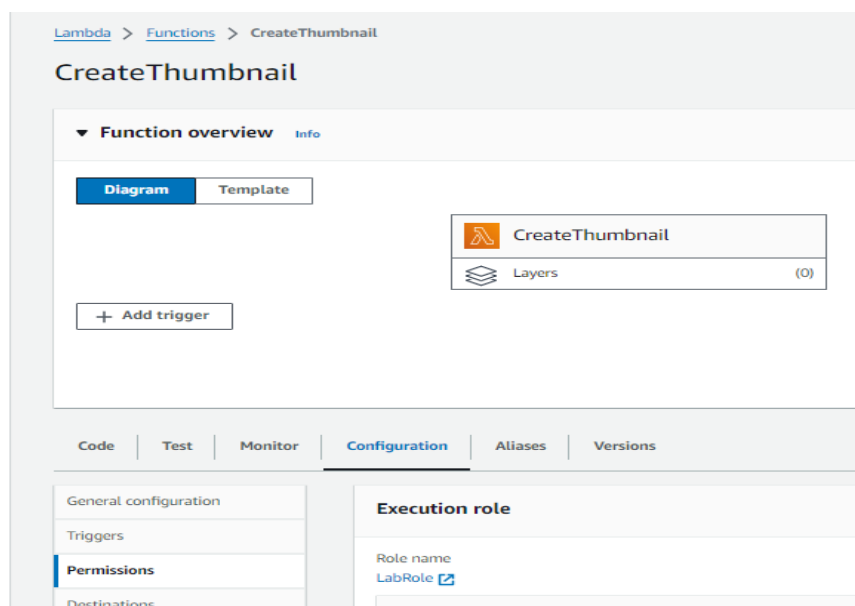


Figure 31 - Using LabRole for creating Thumbnail Function

XIII. CREATING NACL

I create a PrivateSubnetNACL by doing the following things : Click On “Create Network ACL, name = “PrivateSubnetNACL”, VPC = SparoiVPC, Create it. After successfully creating it , go to the Subnet association of that specific NACL and then clic Edit Subnet Association . And select the Private Subnet 1 (10.0.3.0/24), Private Subnet 2 (10.0.4.0/24). Then save it. After that I modify the inbound and outbound rules in order to implement the least principle formula

Network ACLs (1/3) Info

Find resources by attribute or tag

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules count	Outbound rules count	Owner
PrivateSubnetNACL	acl-0f604e0c0b6d911fc	2 Subnets	No	vpc-0e0da234815dca27d / SearaVPC	5 Inbound rules	5 Outbound rules	851725617953
-	acl-0e62a9d9f8e19f4e2	6 Subnets	Yes	vpc-0b6a7876aeb37d2d1	2 Inbound rules	2 Outbound rules	851725617953
-	acl-09baf41c129b24de	2 Subnets	Yes	vpc-0e0da234815dca27d / SearaVPC	2 Inbound rules	2 Outbound rules	851725617953

acl-0f604e0c0b6d911fc / PrivateSubnetNACL

Details | Inbound rules | Outbound rules | **Subnet associations** | Tags

Subnet associations (2)

Filter subnet associations

Name	Subnet ID	Associated with	Availability Zone	IPv4 CIDR	IPv6 CIDR
Private Subnet 1	subnet-0c568f27221f962fc	acl-0f604e0c0b6d911fc / PrivateSubnetNACL	us-east-1a	10.0.3.0/24	-
Private Subnet 2	subnet-04b39475d1490d4f5	acl-0f604e0c0b6d911fc / PrivateSubnetNACL	us-east-1b	10.0.4.0/24	-

Figure 32 – Adding Both Private Subnet on PrivateSubnetNACL

Network ACLs (1/3) Info

Find resources by attribute or tag

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules count	Outbound rules count	Owner
PrivateSubnetNACL	acl-0f604e0c0b6d911fc	2 Subnets	No	vpc-0e0da234815dca27d / SearaVPC	3 Inbound rules	3 Outbound rules	851725617953
-	acl-0e62a9d9f8e19f4e2	6 Subnets	Yes	vpc-0b6a7876aeb37d2d1	2 Inbound rules	2 Outbound rules	851725617953
-	acl-09baf41c129b24de	2 Subnets	Yes	vpc-0e0da234815dca27d / SearaVPC	2 Inbound rules	2 Outbound rules	851725617953

acl-0f604e0c0b6d911fc / PrivateSubnetNACL

Details | **Inbound rules** | Outbound rules | Subnet associations | Tags

Inbound rules (3)

Filter inbound rules

Rule number	Type	Protocol	Port range	Source	Allow/Deny
1	All ICMP - IPv4	ICMP (1)	All	10.0.2.0/24	Deny
2	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Figure 33 - NACL Inbound Rules

Network ACLs (1/3) Info

Find resources by attribute or tag

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules count	Outbound rules count	Owner
PrivateSubnetNACL	acl-0f604e0c0b6d911fc	2 Subnets	No	vpc-0e0da234815dca27d / SearaVPC	3 Inbound rules	3 Outbound rules	851725617953
-	acl-0e62a9d9f8e19f4e2	6 Subnets	Yes	vpc-0b6a7876aeb37d2d1	2 Inbound rules	2 Outbound rules	851725617953
-	acl-09baf41c129b24de	2 Subnets	Yes	vpc-0e0da234815dca27d / SearaVPC	2 Inbound rules	2 Outbound rules	851725617953

acl-0f604e0c0b6d911fc / PrivateSubnetNACL

Details | Inbound rules | **Outbound rules** | Subnet associations | Tags

Outbound rules (3)

Filter outbound rules

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
1	All ICMP - IPv4	ICMP (1)	All	10.0.2.0/24	Deny
2	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Figure 34 - NACL Outbound rules

For testing Purpose , I ssh from DevServer (Connected Dev server via WinSCP using the elastic IP 44.209.21.246) to WevServer 10.0.3.121 (any web server) and Ensure it work. And then I tried to ping from DevServer to Wevserver , then wevserver to devserver and ensure it doesn't work

```
[ec2-user@ip-10-0-3-47 ~]$ ping 10.0.2.79
PING 10.0.2.79 (10.0.2.79) 56(84) bytes of data.
^C
--- 10.0.2.79 ping statistics ---
9 packets transmitted, 0 received, 100% packet loss, time 8187ms
```

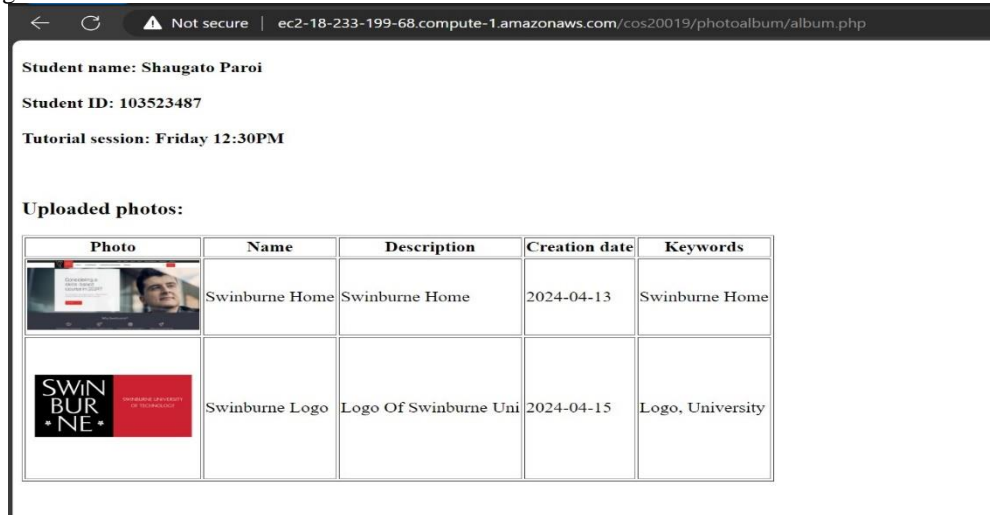
Figure 35 - Ping Blocked Inbound (10.0.2.79 is the private ip of DevServer)

```
[ec2-user@ip-10-0-2-79 ~]$ ping 10.0.3.47
PING 10.0.3.47 (10.0.3.47) 56(84) bytes of data.
^C
--- 10.0.3.47 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3071ms
[ec2-user@ip-10-0-2-79 ~]$ S
```

Figure 36 - Ping Blocked OutBound (Here 10.0.3.47 is the Private Ip address of webserver)

XIV. TESTING WEBSITE FUNCTIONALITY

For Testing Website the website can be accessible through the [Photo Album \(Photo Album \(sparoi-elb-196003924.us-east-1.elb.amazonaws.com\)\)](#) link . From there we can click “upload more photos” to upload photos . This will load photouploader.php ([Photo Album \(sparoi-elb-196003924.us-east-1.elb.amazonaws.com\)](#)) . This photo object are stroed in the Sparoi-bucket (S3) . And the Lambda function will resize the photo Accordingly. I also make sure to configure Security groups accordingly so that Photo album Website can only be accessible from load Balancer only which means it won't be accessible through Devserver.



Student name: Shaugato Paroi
 Student ID: 103523487
 Tutorial session: Friday 12:30PM

Uploaded photos:



Photo	Name	Description	Creation date	Keywords
	Swinburne Home	Swinburne Home	2024-04-13	Swinburne Home
	Swinburne Logo	Logo Of Swinburne Uni	2024-04-15	Logo, University

Figure 37 - Accessing album.php through Elb Address



Photo uploader

Photo title:

Select a photo (Select PNG file for best result): No file chosen

Description:

Date:

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

[Photo Album](#)

Figure 38- Photouploader.php can also be accessible via ELB address

Amazon S3 > Buckets > sparoi-bucket

sparoi-bucket [Info](#)

[Objects](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (6) [Info](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	3uzhu2a8wh591 (1).jpg	jpg	May 4, 2024, 19:17:40 (UTC+10:00)	163.6 KB	Standard
<input type="checkbox"/>	gojo.png	png	May 6, 2024, 00:44:43 (UTC+10:00)	664.1 KB	Standard
<input type="checkbox"/>	OIP.jpg	jpg	May 4, 2024, 14:10:01 (UTC+10:00)	81.6 KB	Standard
<input type="checkbox"/>	resized-3uzhu2a8wh591 (1).jpg	jpg	May 4, 2024, 19:17:44 (UTC+10:00)	60.4 KB	Standard
<input type="checkbox"/>	resized-gojo.png	png	May 6, 2024, 00:44:48 (UTC+10:00)	189.6 KB	Standard
<input type="checkbox"/>	resized-OIP.jpg	jpg	May 4, 2024, 14:10:02 (UTC+10:00)	28.5 KB	Standard

Figure 39 - Photos have been resized via Lambda function



Figure 40 - S3 bucket is not publicly accessible

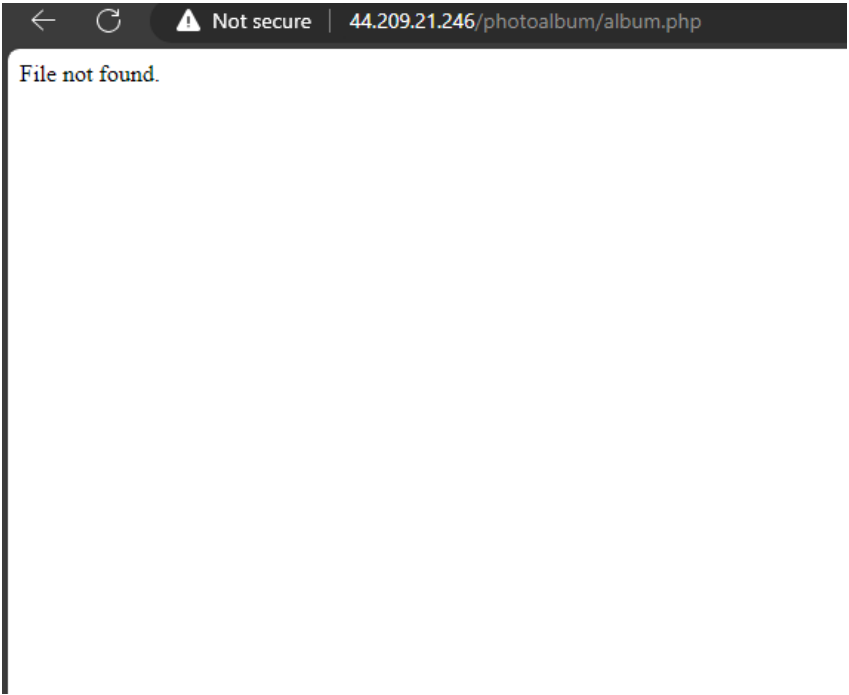


Figure 41 - PhotoAlbum can not be Accessible through Devserver's Elastic IP

REFERENCES