



Assignment 2

COS 20019- Cloud Computing Architecture

Nguyen Manh Dung

Student ID: 104181789

Tutorial Class :Saturday 13:00PM – 17:00PM

12/7/2023

I. Introduction

This assignment will extend/modify the infrastructure and program you developed in Assignment 1b.

The essay seeks to accomplish a number of goals. In order to ease communication and ensure safe data transmission between EC2 (Elastic Compute Cloud), Lambda, and S3 (Simple Storage Service), we will first look at the creation of IAM (Identity and Access Management) roles. Second, we'll go through how to build S3 bucket policies to limit access to particular people or groups and manage S3 access. The development of a lambda function that may automatically run code in response to particular triggers is the focus of our third subject.

The essay seeks to accomplish a number of goals. In order to ease communication and ensure safe data transmission between EC2 (Elastic Compute Cloud), Lambda, and S3 (Simple Storage Service), we will first look at the creation of IAM (Identity and Access Management) roles. Second, we'll go through how to build S3 bucket policies to limit access to particular people or groups and manage S3 access. The development of a lambda function that may automatically run code in response to particular triggers is the focus of our third subject.

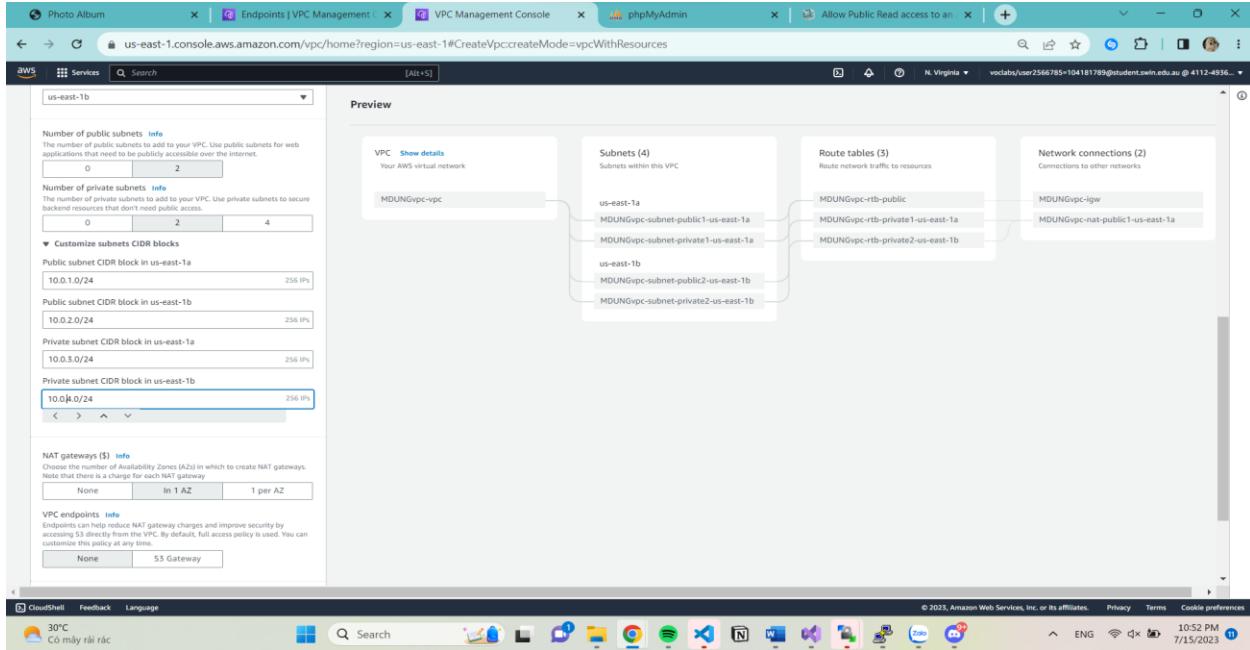
Seventh, we'll build an elastic load balancer with good workload distribution to govern system access. Finally, we'll look at network control and traffic limitations using Amazon NACLs (Network Access Control Lists), which permit or reject traffic depending on IP address, port, or protocol.

II. Application Deployment

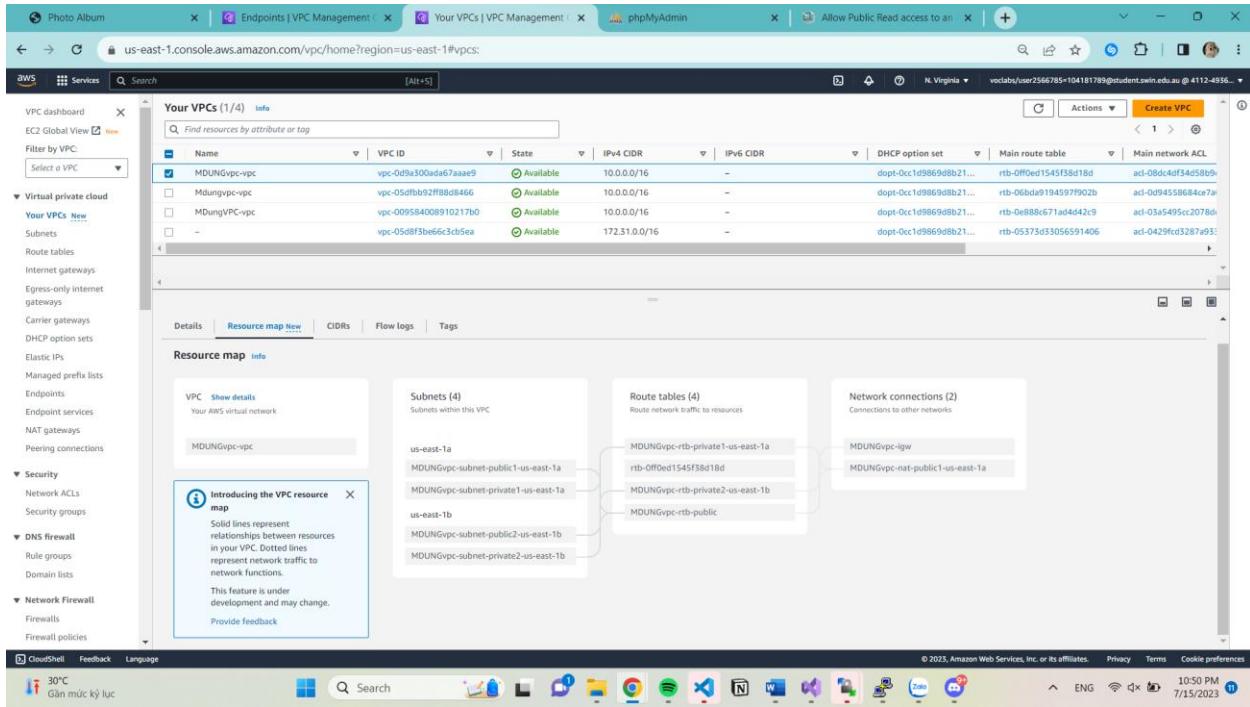
1. Infrastructure

- a. VPC configured with 2AZs both with public and private subnets. Public and private route tables route to IGW and NAT, respectively.

Created VPC



Below screenshots illustrate VPC diagram



2. Security Group

Create five security groups (I've configure inbound and outbound rules with same configuration)

- **ELBSG:** for the ELB created

The screenshot shows the AWS CloudFormation console with a tab for 'EC2 Management' open. A new security group is being created with the following details:

- Basic details:**
 - Security group name: ELBSG
 - Description: for the ELB created
 - VPC: vpc-0ef9a300ada67aae9
- Inbound rules:** One rule is defined:
 - Type: HTTP
 - Protocol: TCP
 - Port range: 80
 - Source: Anywhere-IPv4
 - Destination: 0.0.0.0/0
- Outbound rules:** One rule is defined:
 - Type: HTTP
 - Protocol: TCP
 - Port range: 80
 - Destination: Anywhere-IPv4
 - Destination: 0.0.0.0/0

The browser status bar at the bottom indicates the date and time as 7/16/2023 12:01 AM.

The screenshot shows the AWS Management Console with the VPC Management Console tab active. A success message at the top states: "Security group (sg-01d95624c30f9d1bc | ELBSG) was created successfully". The main pane displays the details of the newly created security group, sg-01d95624c30f9d1bc - ELBSG. The "Details" section includes fields for Security group name (ELBSG), Security group ID (sg-01d95624c30f9d1bc), Description (for the ELB created), Owner (411249364900), Inbound rules count (1 Permission entry), and Outbound rules count (1 Permission entry). Below this, the "Inbound rules" tab is selected, showing one rule: Name sgr-07aa15e1375791c..., IP version IPv4, Type HTTP, Protocol TCP, Port range 80, Source 0.0.0.0/0, and Description -. There is also a "Run Reachability Analyzer" button.

This screenshot shows the same security group configuration page, but the "Outbound rules" tab is now selected. It displays one outbound rule: Name sgr-07aa15e1375791c..., IP version IPv4, Type HTTP, Protocol TCP, Port range 80, Destination 0.0.0.0/0, and Description -. There is also a "Run Reachability Analyzer" button.

- **WebServerSG:** for all the web servers in private subnets
(Source is ELB security group)

The screenshot shows two consecutive screenshots of the AWS Management Console interface.

Screenshot 1: Creating a Security Group

- The URL is <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroupsort=desc:tag:Name>.
- The page title is "Basic details".
- Security group name:** WebServerSG
- Description:** for all the web server in public subnets
- VPC:** vpc-0d9a300ada67aaae9
- Inbound rules:** An inbound rule is being configured for port 80 (HTTP) from a custom source (sg-01d95624c30f9d1bc).
- Outbound rules:** An outbound rule is being configured for port 80 (HTTP) to a custom destination (sg-01d95624c30f9d1bc).

Screenshot 2: Security Group Created

- The URL is https://us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#SecurityGroup:group_id=sg-0df7cb4c23f2464e4.
- The page title is "Security group (sg-0df7cb4c23f2464e4 | WebServerSG) was created successfully".
- Details:**
 - Security group name:** WebServerSG
 - Security group ID:** sg-0df7cb4c23f2464e4
 - Description:** for all the web servers in private subnets
 - VPC ID:** vpc-0d9a300ada67aaae9
 - Owner:** 411249564900
 - Inbound rules count:** 1 Permission entry
 - Outbound rules count:** 1 Permission entry
- Inbound rules:** One inbound rule is listed: sgr-0257591bc5c22df0c (HTTP, TCP, port 80, source sg-01d95624c30f9d1bc).

The screenshot shows the AWS Management Console interface for managing security groups. The specific page is for the security group 'sg-0df7cb4c23f2464e4' named 'WebServerSG'. The 'Outbound rules' tab is selected. A single outbound rule is listed:

Name	Security group rule...	IP version	Type	Protocol	Port range	Destination	Description
-	sgr-0aa3d1a0175c70ae1	IPv4	All traffic	All	All	0.0.0.0/0	-

At the bottom right of the page, there are links for 'Privacy', 'Terms', and 'Cookie preferences'.

- **DBServerSG:** for the RDS instance (source is DevServerSG and WebServerSG)

Basic details

Security group name: **DBServerSG**
Description: for the RDS instance
VPC: vpc-0df9a300ada67aae9

Inbound rules

Type	Protocol	Port range	Source	Description - optional
MySQL/Aurora	TCP	3306	Custom sg-0df7cb4c23f2464e4	
MySQL/Aurora	TCP	3306	Custom sg-024b955e419e793c7	

Outbound rules

Type	Protocol	Port range	Destination	Description - optional
MySQL/Aurora	TCP	3306	Custom sg-024b955e419e793c7	
Custom TCP	TCP	0	Custom sg-024b955e419e793c7	

Details

Security group name: **DBServerSG**
Owner: 411249364900
VPC ID: vpc-0df9a300ada67aae9

Inbound rules (2)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-01768d4284a5decda	-	MySQL/Aurora	TCP	3306	sg-0df7cb4c23f2464e...	-
-	sgr-075765d8d1bb10...	-	MySQL/Aurora	TCP	3306	sg-024b955e419e793...	-

- **NATServerSG:** for the NAT server

Basic details

Security group name: **NATServerSG**
Description: **for the NAT server**

Inbound rules

Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	3306	Anywhere-IPv4	0.0.0.0/0

Outbound rules

Type	Protocol	Port range	Destination	Description - optional
Custom TCP	TCP	0	Anywhere-IPv4	0.0.0.0/0

Details

Security group name: **NATServerSG**
Owner: **411249564900**
Inbound rules count: **1 Permission entry**
Outbound rules count: **1 Permission entry**

Inbound rules (1/1)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-091b969da5145...	IPv4	Custom TCP	TCP	0	0.0.0.0/0	-

The screenshot shows the AWS Management Console interface for managing security groups. The specific page is for the security group 'sg-08b658c786feab4c3 - DBServerSG'. The 'Outbound rules' tab is selected, showing one rule listed in a table.

Name	Security group rule...	IP version	Type	Protocol	Port range	Destination	Description
-	sgr-04d52850a940bef92	IPv4	All traffic	All	All	0.0.0.0/0	-

At the bottom right of the page, there is a footer with links: © 2023, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

- **DevServerSG:** for the Dev server(all traffic)

The screenshot shows two consecutive screenshots of the AWS Management Console interface.

Screenshot 1: Security Group Creation

This screenshot shows the 'Basic details' section of a new security group. The security group name is 'DevServerSG'. The description is 'for the NAT server'. The VPC dropdown shows 'vpc-0d9a300ada67aaae9'. The 'Inbound rules' section is visible, showing a single rule allowing all traffic from anywhere on port 0.0.0.0/0. The 'Outbound rules' section is also visible, showing a single rule allowing all traffic to anywhere on port 0.0.0.0/0.

Screenshot 2: Security Group Details

This screenshot shows the details of the security group 'sg-024b955e419e793c7 - DevServerSG'. It lists the security group name, ID, owner (411249364900), and descriptions. It shows 1 inbound rule and 1 outbound rule. The 'Inbound rules' tab is selected, displaying a table with one rule: 'sgr-0642ecb9d408e44f2' (IPv4, All traffic, All, 0.0.0.0/0). The 'Outbound rules' tab is also visible.

The screenshot shows the 'Outbound rules' tab for a security group named 'sg-024b955e419e793c7 - DevServerSG'. There is one rule listed:

Name	Security group rule...	IP version	Type	Protocol	Port range	Destination	Description
-	sgr-0acde09b851fd218f	IPv4	All traffic	All	All	0.0.0.0/0	-

3. Network ACLs

Create NACL

The screenshot shows the 'Create network ACL' page. It includes fields for 'Name' (PrivateSubnetsNACL), 'VPC' (vpc-0d9a300ad67aaaef (MDUNGvpc-vpc)), and 'Tags' (Key: Name, Value: PrivateSubnetsNACL). The 'Create network ACL' button is highlighted.

Edit subnet associations

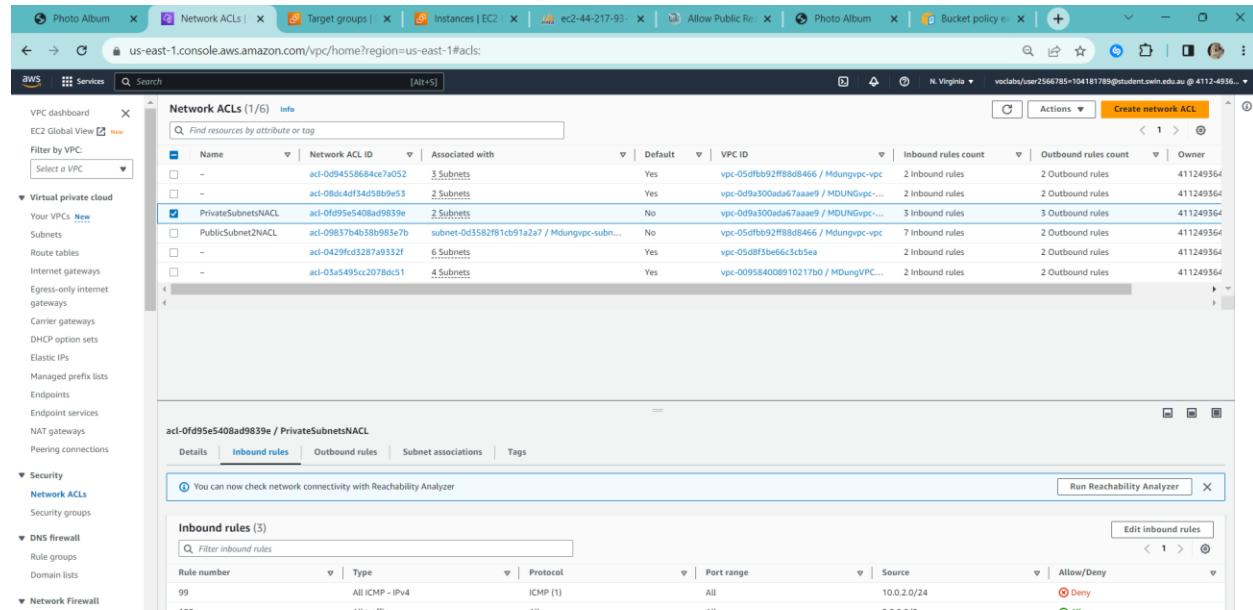
The screenshot shows the AWS VPC Management Console with the URL us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#EditNetworkAclSubnetAssociations:networkAclId=acl-0fd95e5408ad9839e. The page title is "Edit subnet associations".
Available subnets (2/4)
Change which subnets are associated with this network ACL.

Name	Subnet ID	Associated with	Availability Zone	IPv4 CIDR	IPv6 CIDR
<input checked="" type="checkbox"/> MDUNGvpc-subnet-private2-us-east-1b	subnet-04175915b21ab10c8	acl-08dc4df34d58b9e53	us-east-1b	10.0.4.0/24	-
<input type="checkbox"/> MDUNGvpc-subnet-public2-us-east-1b	subnet-09484043054499069	acl-08dc4df34d58b9e53	us-east-1b	10.0.2.0/24	-
<input type="checkbox"/> MDUNGvpc-subnet-public1-us-east-1a	subnet-07ebbd01099eb074	acl-08dc4df34d58b9e53	us-east-1a	10.0.1.0/24	-
<input checked="" type="checkbox"/> MDUNGvpc-subnet-private1-us-east-1a	subnet-0b4769787bd3f3d8d	acl-08dc4df34d58b9e53	us-east-1a	10.0.3.0/24	-

Selected subnets
subnet-0b4769787bd3f3d8d / MDUNGvpc-subnet-private1-us-east-1a | subnet-04175915b21ab10c8 / MDUNGvpc-subnet-private2-us-east-1b

Cancel **Save changes**

Edit rule



The screenshot shows the AWS VPC Network ACLs page. The left sidebar navigation includes: VPC dashboard, EC2 Global View, Filter by VPC (Select a VPC), Virtual private cloud (Your VPCs New Subnets Route tables Internet gateways Egress-only internet gateways Carrier gateways DHCP option sets Elastic IPs Managed prefix lists Endpoints Endpoint services NAT gateways Peering connections), Security (Network ACLs, Security groups), DNS firewall (Rule groups, Domain lists), Network Firewall (Firewalls, Firewall policies), CloudShell, Feedback, Language (29°C, Có máy rái rác), and a browser toolbar with various icons.

Inbound rules (3)

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

Outbound rules (3)

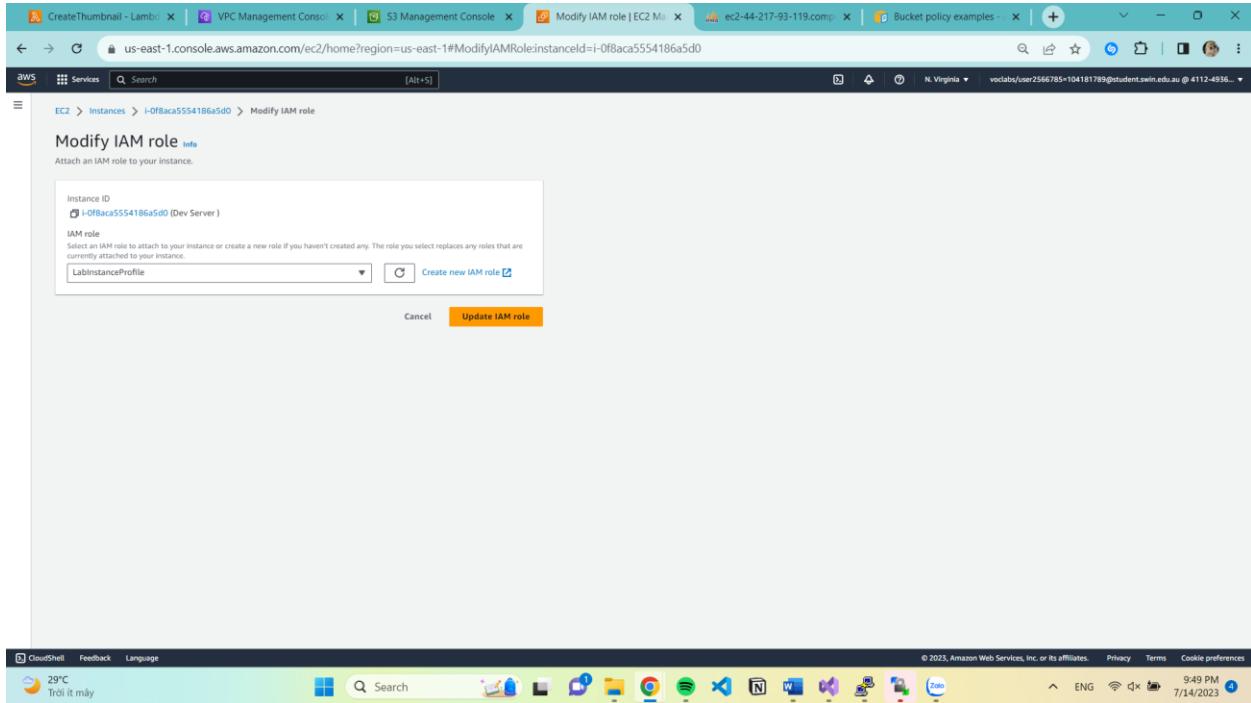
Rule number	Type	Protocol	Port range	Destination	Allow/Deny
99	All ICMP - IPv4	ICMP (1)	All	10.0.2.0/24	Deny
100	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

4. S3 Photo Storage

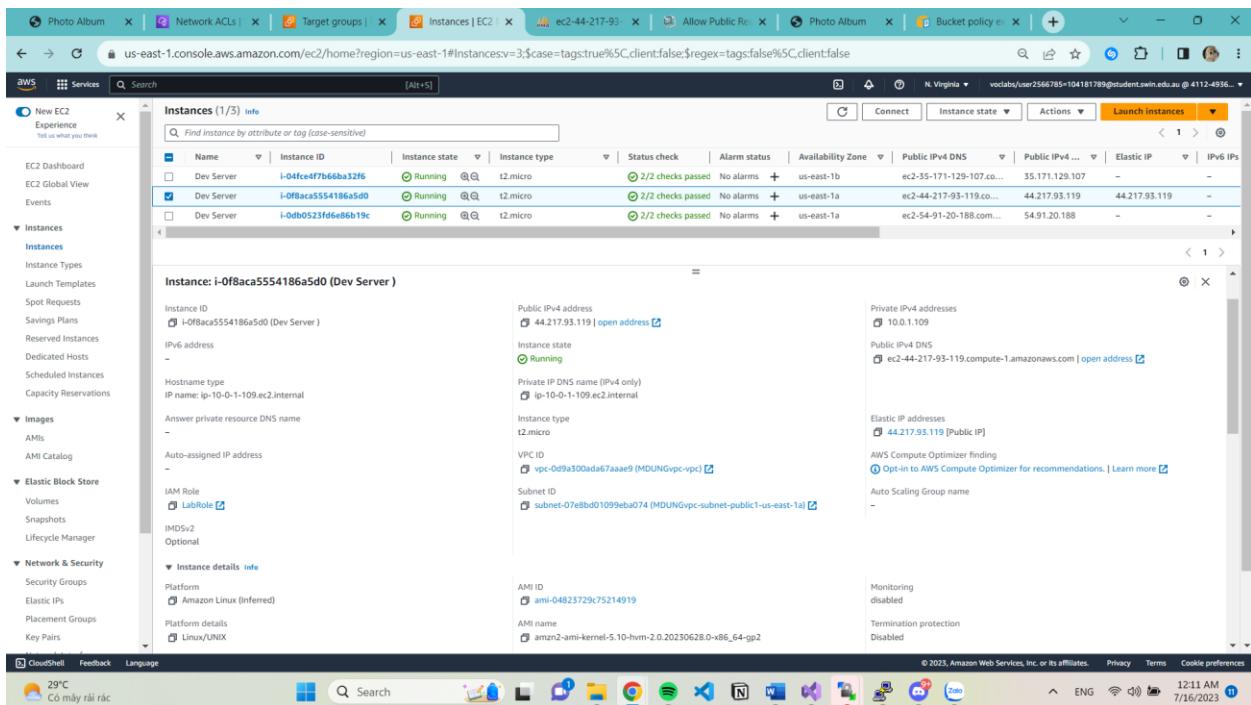
Create Bucket

The screenshot shows the AWS S3 'Create bucket' interface. In the 'General configuration' section, the 'Bucket name' is set to 'mdungasm2bucket'. The 'AWS Region' is selected as 'US East (N. Virginia) us-east-1'. Under 'Object Ownership', the 'Object Ownership' dropdown is set to 'Bucket owner enforced'. The 'Block Public Access settings for this bucket' section is expanded, showing several options under 'Block all public access' and a warning message: 'Turning off block all public access might result in this bucket and the objects within becoming public. AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.' At the bottom, there is an acknowledgement checkbox: 'I acknowledge that the current settings might result in this bucket and the objects within becoming public.'

Modify IAM Role



DevServer IAM Role



Lambda IAM Role

The screenshot shows the AWS Lambda console interface. The main area displays the 'CreateThumbnail' function. The 'Configuration' tab is active. On the left, a sidebar lists various configuration options: General configuration, Triggers, Permissions (which is currently selected), Destinations, Function URL, Environment variables, Tags, VPC, Monitoring and operations tools, and Concurrency. The 'Permissions' section shows the 'Execution role' set to 'LabRole'. Below it, the 'Resource-based policy statements' section contains a single policy entry:

```

User: arn:aws:sts::411249364900:assumed-role/volabs/user2566785+104181789@student.swin.edu.au is not authorized to perform: iam:GetPolicy on resource: policy
arn:aws:iam::411249364900:policy/c84300a1795577142224211w411249364900-VoClabPolicy1-1GVT29JQ0G4P with an explicit deny in an identity-based policy
  
```

The status bar at the bottom indicates the user is from 'volabs/user2566785+104181789@student.swin.edu.au @ 4112-4936...'.

ASM2 Launch Template

Create launch template

The screenshot shows the AWS EC2 console 'Create launch template' wizard. The 'Summary' step is displayed, listing the following configuration details:

- Software Image (AMI)
- Virtual server type (instance type)
- Firewall (security group)
- Storage (volumes)

A note about the 'Free tier' is shown, stating: 'In your first year includes > 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.'

The 'Create launch template' button is highlighted in orange. The status bar at the bottom indicates the user is from 'volabs/user2566785+104181789@student.swin.edu.au @ 4112-4936...'.

Amazon Machine Image (AMI)

- WebServerAMI
 - ami-0db56227d508e6879
 - 2023-07-14T16:33:42.000Z
 - Virtualization: hvm
 - ENI enabled: true
 - Root device type: ebs

Instance type

t2.micro

Key pair (login)

asm2kp

Network settings

Summary

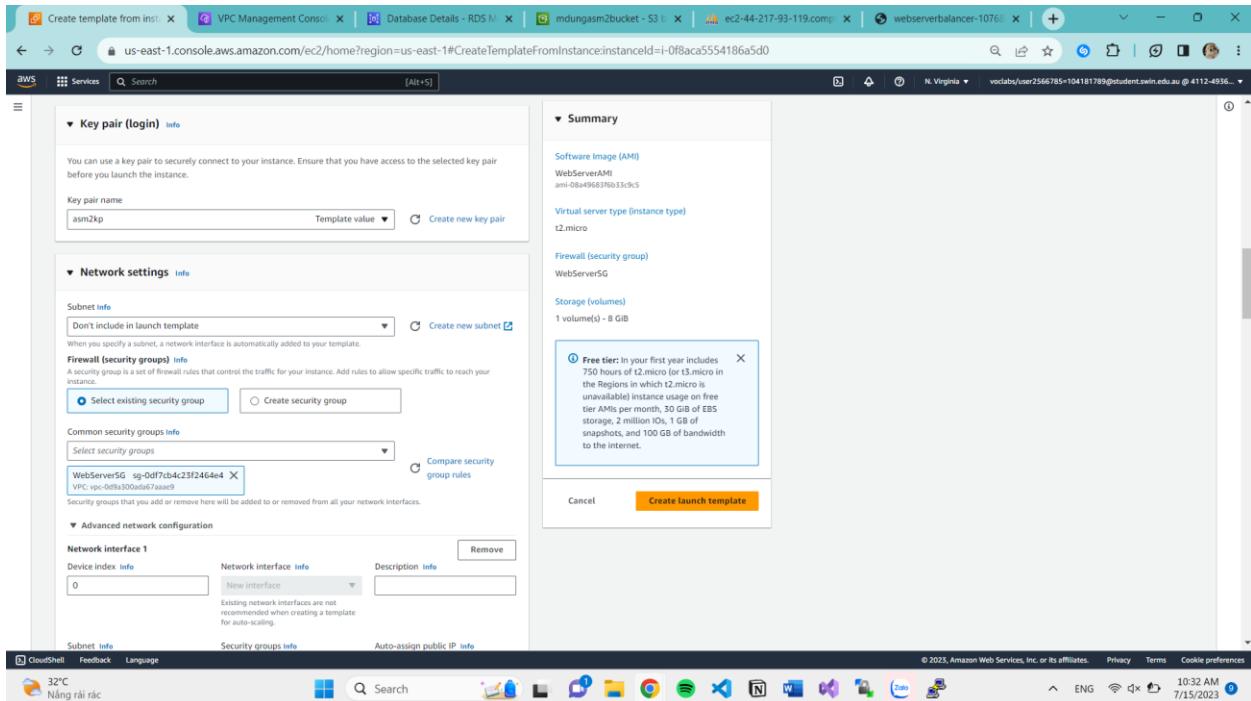
Software Image (AMI): WebServerAMI
ami-0db56227d508e6879

Virtual server type (instance type): t2.micro

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOPS, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Create launch template



5. ASG configured and working correctly

The screenshot shows the AWS CloudWatch Metrics console with the URL <https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#AutoScalingGroupDetails:id=WebServerScalingGroup;view=details>. The page displays the 'WebServerScalingGroup' configuration under the 'Automatic scaling' tab. Key details include:

- Auto Scaling group name:** WebServerScalingGroup
- Desired capacity:** 2
- Status:** Updating capacity
- Amazon Resource Name (ARN):** arn:aws:autoscaling:us-east-1:411249364900:autoScalingGroup:83cb3d04-4fed-4b73-8fe5-3374473b9d48:autoScalingGroupName/WebServerScalingGroup
- Launch template:** WebServerTemplate (AMI ID: ami-09924a776fe62d1e1, Instance type: t2.micro)
- Network:** Subnet ID: subnet-00000000, Security group: sg-024b55e419e793c7
- Request Spot Instances:** No

The screenshot shows the AWS CloudWatch Metrics console with the URL <https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#AutoScalingGroupDetails:id=WebServerScalingGroup;view=scaling>. The page displays the 'WebServerScalingGroup' configuration under the 'Automatic scaling' tab. Key details include:

- Scaling policies:** One scaling policy named 'Target Tracking Policy' is listed.
- Dynamic scaling policies:** One policy named 'Target tracking scaling' is listed.
- Predictive scaling policies:** One policy named 'Evaluation based on 2 days' is listed.

6. ELB configured and working correctly

The screenshot shows the AWS Cloud Console interface for managing a Load Balancer. The left sidebar navigation includes services like Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area displays the 'WebServerBalancer' configuration. Key details include:

- Load balancer type:** Application
- Status:** Active
- Scheme:** Internet-facing
- Hosted zone:** Z35SKD0TRQ7X7K
- VPC:** vpc-0d9a300eda67aaae9
- IP address type:** IPv4
- Availability Zones:** us-east-1b (use1-az1), subnet-09484d43054c99061; us-east-1b (use1-az2), subnet-07e8bd01099eba074
- Date created:** July 15, 2023, 10:57 (UTC+07:00)
- DNS name:** WebServerBalancer-333406670.us-east-1.elb.amazonaws.com (A Record)

The 'Listeners and rules' tab is active, showing one rule for port 80:

ProtocolPort	Default action	Rules	ARN	Security policy	Default SSL cert	Tags
HTTP:80	Forward to target group	• WebServerGroup: 1 (100%) • Group-level stickiness: Off	1 rule	Not applicable	Not applicable	0 tags

Health check

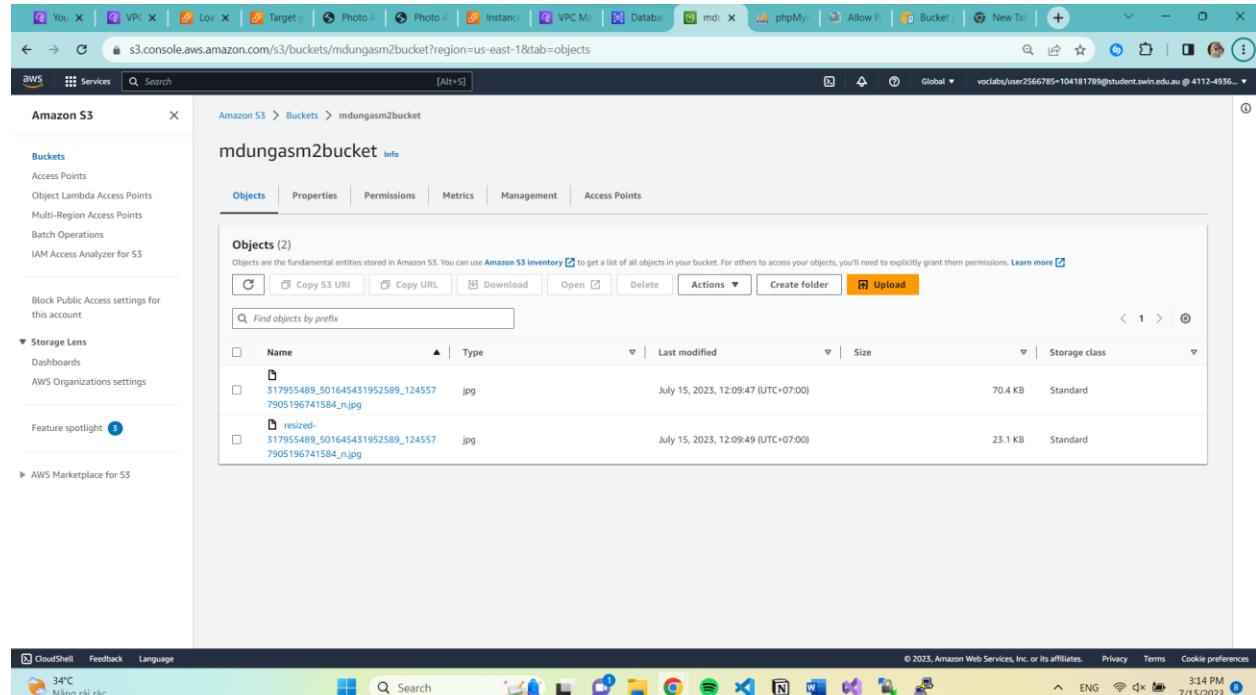
The screenshot shows the AWS Cloud Console interface for managing a Target Group. The left sidebar navigation includes services like Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area displays the 'WebServerGroup' configuration. Key details include:

- Target type:** Instance
- Protocol:** HTTP: 80
- Protocol version:** HTTP1
- VPC:** vpc-0d9a300eda67aaae9
- Total targets:** 2
- Healthy:** 2
- Unhealthy:** 0
- Unused:** 0
- Initial:** 0
- Draining:** 0

The 'Targets' tab is active, showing the registered targets:

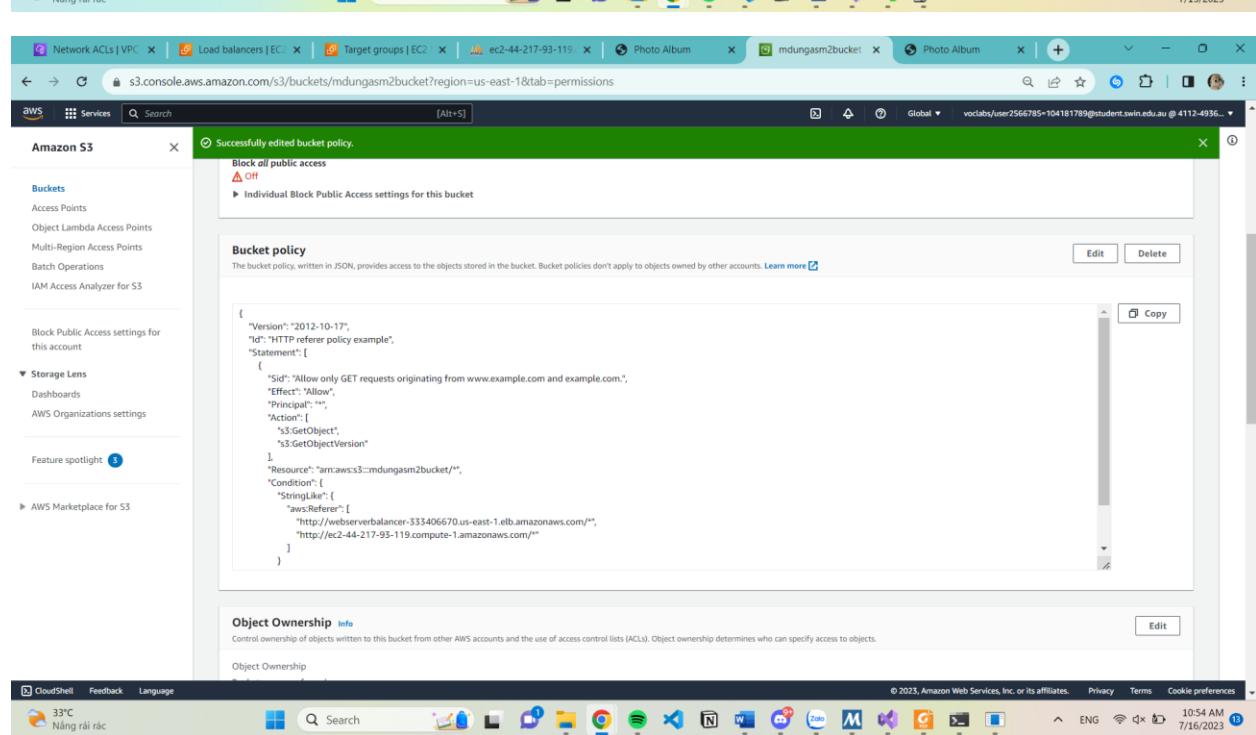
Instance ID	Name	Port	Zone	Health status	Health status details
i-0db0523fd6e86b19c	Dev Server	80	us-east-1a	healthy	
i-04fce4f7b66ba52f6	Dev Server	80	us-east-1b	healthy	

7. Photos stored in S3 are correctly accessible. S3 bucket policy is correct.



The screenshot shows the AWS S3 console with the 'Objects' tab selected. The bucket 'mdungasm2bucket' contains two files:

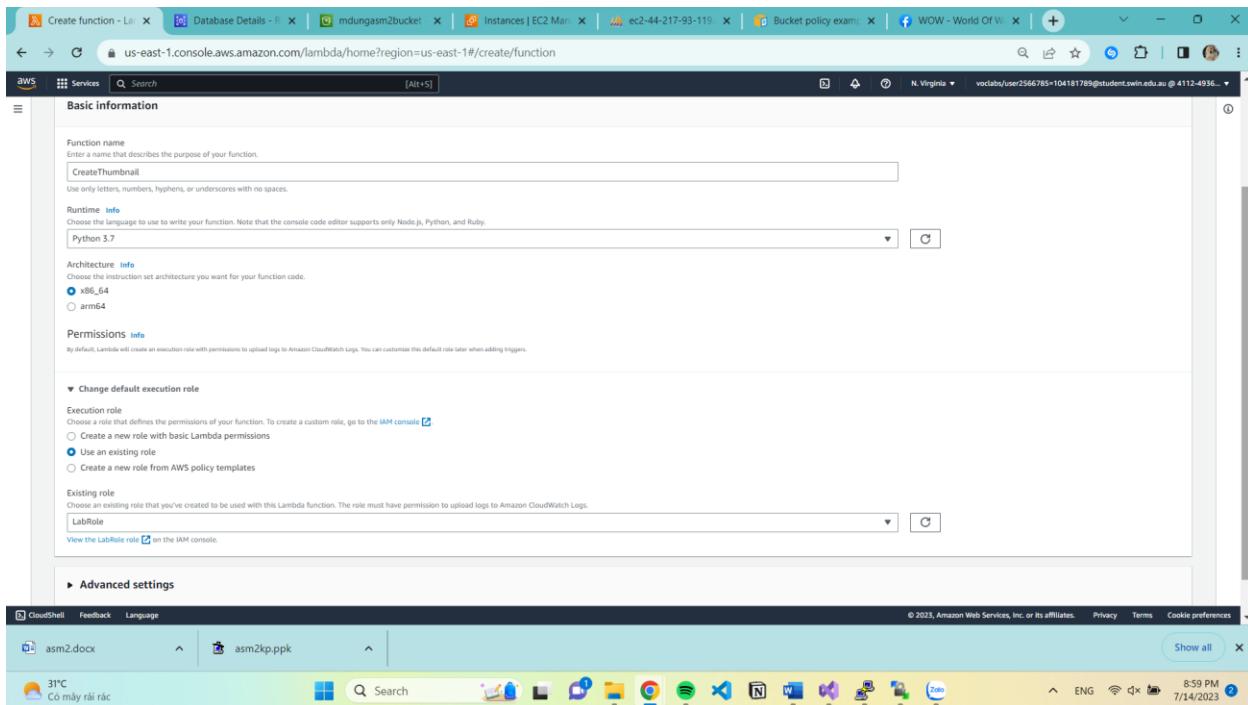
Name	Type	Last modified	Size	Storage class
317955489_501645431952589_124557.jpg	jpg	July 15, 2023, 12:09:47 (UTC+07:00)	70.4 KB	Standard
317955489_501645431952589_124557_resized-.jpg	jpg	July 15, 2023, 12:09:49 (UTC+07:00)	23.1 KB	Standard



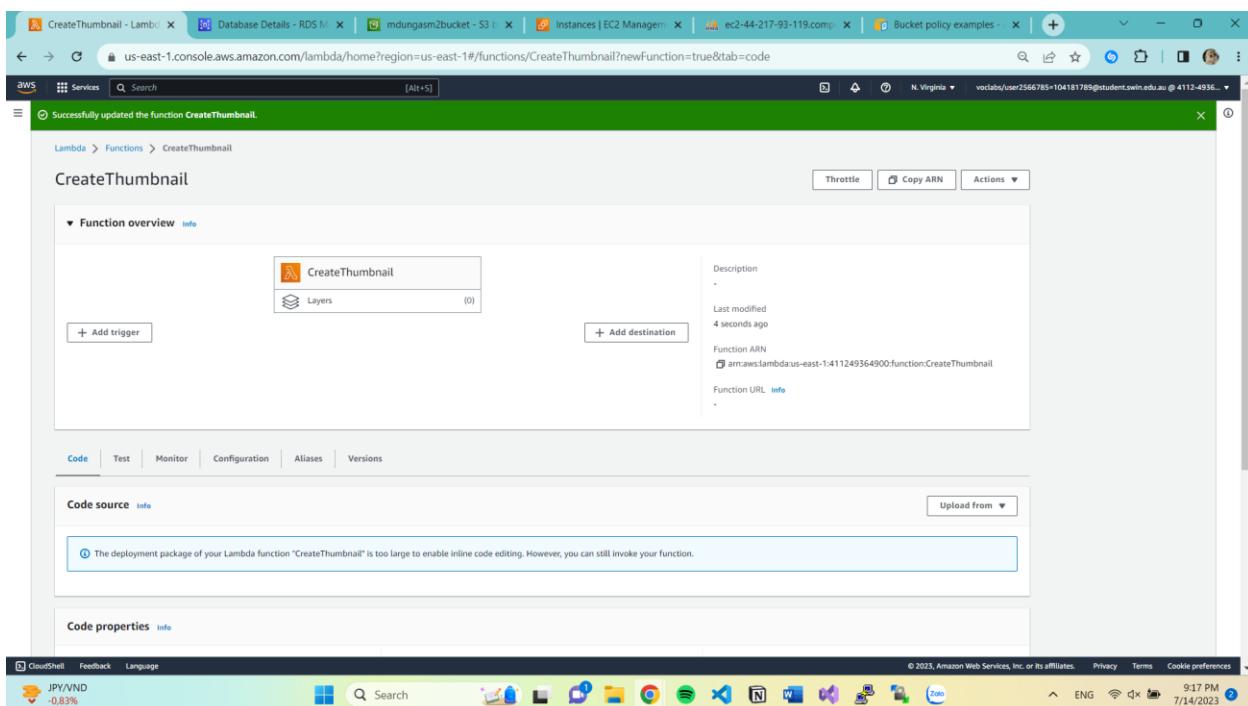
The screenshot shows the 'Bucket policy' section of the AWS S3 console. A green success message says 'Successfully edited bucket policy.' Below it, the 'Block all public access' setting is shown as 'Off'. The 'Bucket policy' JSON code is displayed:

```
{
  "Version": "2012-10-17",
  "Id": "HTTP referrer policy example",
  "Statement": [
    {
      "Sid": "Allow only GET requests originating from www.example.com and example.com",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "S3.GetObject",
        "S3.GetObjectVersion"
      ],
      "Resource": "arn:aws:s3:::mdungasm2bucket/*",
      "Condition": {
        "StringLike": {
          "aws:Referrer": [
            "http://webservicebalancer-333406670.us-east-1.elb.amazonaws.com/*",
            "http://ec2-44-217-93-119.compute-1.amazonaws.com/*"
          ]
        }
      }
    }
  ]
}
```

8. Lambda configured and working correctly.



Upload from zip file



Lambda Role

The screenshot shows the AWS Lambda console with the 'CreateThumbnail' function selected. The 'Configuration' tab is active. The 'Execution role' section shows 'LabRole' assigned. A warning message in the resource summary indicates that the user lacks permission to perform the 'iam:GetPolicy' action on the specified policy.

9. RDS configured and working correctly

Create RDS Database

The screenshot shows the AWS RDS Management Console with the 'Create DB subnet group' wizard open. The 'Subnet group details' step is shown, where a new subnet group named 'dbasm2' is being created under the 'MDUNGvpc-vpc' VPC. The 'Add subnets' step shows two subnets from the 'us-east-1a' availability zone selected: 'us-east-1a' and 'us-east-1b'. The 'Subnets' step is partially visible below.

Databases - RDS Management | Subnet Groups - RDS Manager | VPC Management Console | New Tab | us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databases:

Successfully created database database-asm2

You can use settings from database-asm2 to simplify configuration of suggested database add-ons while we finish creating your DB for you.

How was your experience creating an Amazon RDS database? [Provide feedback](#)

RDS > Databases

Databases (2)

DB identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU	Current activity	Maintenance	VPC	Multi-AZ
database-1	Available	Instance	MySQL Community	us-east-1b	db.t3.micro	2 Actions	2.23%	0 Connections	available	vpc-05dfb92ff8bd8466	No
database-asm2	Backing-up	Instance	MySQL Community	us-east-1a	db.t3.micro	-	33.42%	0 Connections	none	vpc-0d9a300ada67aae9	No

CloudShell Feedback Language

36°C Nắng rải rác

Search

CloudShell Feedback Language

35°C Nhiều mây

CloudShell Feedback Language

III. Functional requirements

10. Website accessible via ELB

You | Lô | Lô | Target | Photo | Photo | Create | VPC M | Database | mdung | phpMy | Allow P | Bucket | New Tab | us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancer:loadBalancerArn=arn:aws:elasticloadbalancing:us-east-1:411249364900:loadbalancer/app/Web... | CloudShell Feedback Language

Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Scheduled Instances Capacity Reservations

Images AMIs AMI Catalog

Elastic Block Store Volumes Snapshots Lifecycle Manager

Network & Security Security Groups Elastic IPs Placement Groups Key Pairs Network Interfaces

Load Balancing Load Balancers Target Groups

Auto Scaling Auto Scaling Groups

CloudShell Feedback Language

35°C Nhiều mây

CloudShell Feedback Language

CloudShell Feedback Language

The screenshot shows a web browser window with multiple tabs open at the top. The active tab displays a photo album page titled "Photo Album". The page includes the following information:

Student name: Nguyen Manh Dung
Student ID: 104181789
Tutorial session: Saturday 09:15AM

Uploaded photos:

Upload more photos

Photo	Name	Description	Creation date	Keywords
	my son a		2023-07-05	q

11. Photos and their meta-data displayed on album.php page

The screenshot shows a web browser window with multiple tabs open at the top. The active tab displays a photo album page titled "Photo Album". The page includes the following information:

Student name: Nguyen Manh Dung
Student ID: 104181789
Tutorial session: Saturday 09:15AM

Uploaded photos:

Upload more photos

Photo	Name	Description	Creation date	Keywords
	my son a		2023-07-05	q
	nhen	spi	2023-06-28	wow
	co	vñ	2023-07-15	vnam

12. Photos and their meta-data can be uploaded to the S3 bucket and RDS database, respectively

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with 'Amazon S3' selected under 'Services'. Under 'Buckets', 'mdungasm2bucket' is listed. The main area shows the 'Objects' tab for the bucket. There are four objects listed:

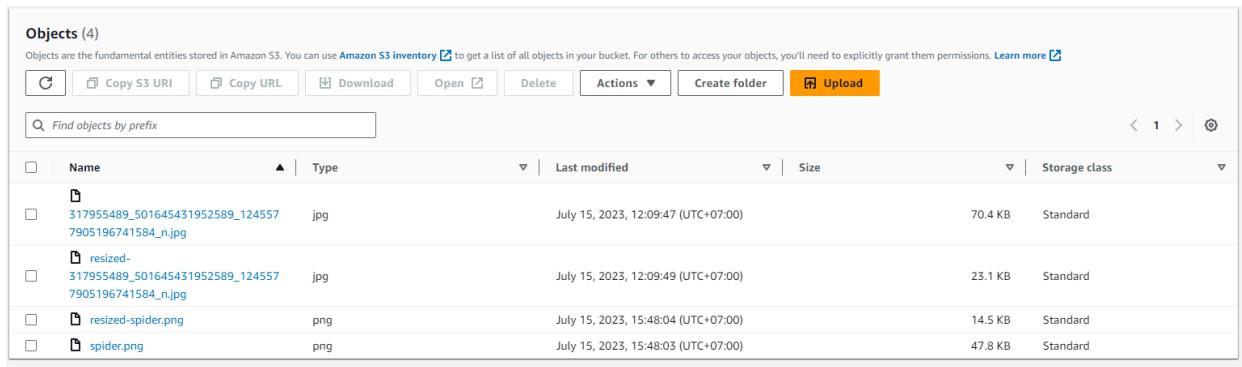
Name	Type	Last modified	Size	Storage class
317955489_501645431952589_124557_7905196741584_n.jpg	jpg	July 15, 2023, 12:09:47 (UTC+07:00)	70.4 KB	Standard
resized-317955489_501645431952589_124557_7905196741584_n.jpg	jpg	July 15, 2023, 12:09:49 (UTC+07:00)	23.1 KB	Standard
resized-spider.png	png	July 15, 2023, 15:48:04 (UTC+07:00)	14.5 KB	Standard
spider.png	png	July 15, 2023, 15:48:03 (UTC+07:00)	47.8 KB	Standard

The photographs are shown not only in the album .php but also in the S3 bucket and phpmyadmin.php once they have been uploaded using the photouploader.php website.

The screenshot shows the phpMyAdmin interface. On the left, the database structure is visible with 'New', 'asm2', 'asm2db', 'information_schema', 'mysql', 'performance_schema', and 'sys' databases. The 'asm2' database is selected, and the 'asm2table' table is shown. The table has the following data:

title	description	creationdate	keywords	reference
my son a		2023-07-05	q	https://mdungasm2bucket.s3.amazonaws.com/317955489_501645431952589_124557_7905196741584_n.jpg
nhien spi		2023-06-28	wow	https://mdungasm2bucket.s3.amazonaws.com/spider.png

13. Photos are resized by the Lambda function.



The screenshot shows an Amazon S3 bucket named 'Objects (4)'. It lists four items:

Name	Type	Last modified	Size	Storage class
317955489_501645431952589_124557_7905196741584_n.jpg	jpg	July 15, 2023, 12:09:47 (UTC+07:00)	70.4 KB	Standard
resized-317955489_501645431952589_124557_7905196741584_n.jpg	jpg	July 15, 2023, 12:09:49 (UTC+07:00)	23.1 KB	Standard
resized-spider.png	png	July 15, 2023, 15:48:04 (UTC+07:00)	14.5 KB	Standard
spider.png	png	July 15, 2023, 15:48:03 (UTC+07:00)	47.8 KB	Standard

Submission Checklist

Student Name: Nguyen Manh Dung

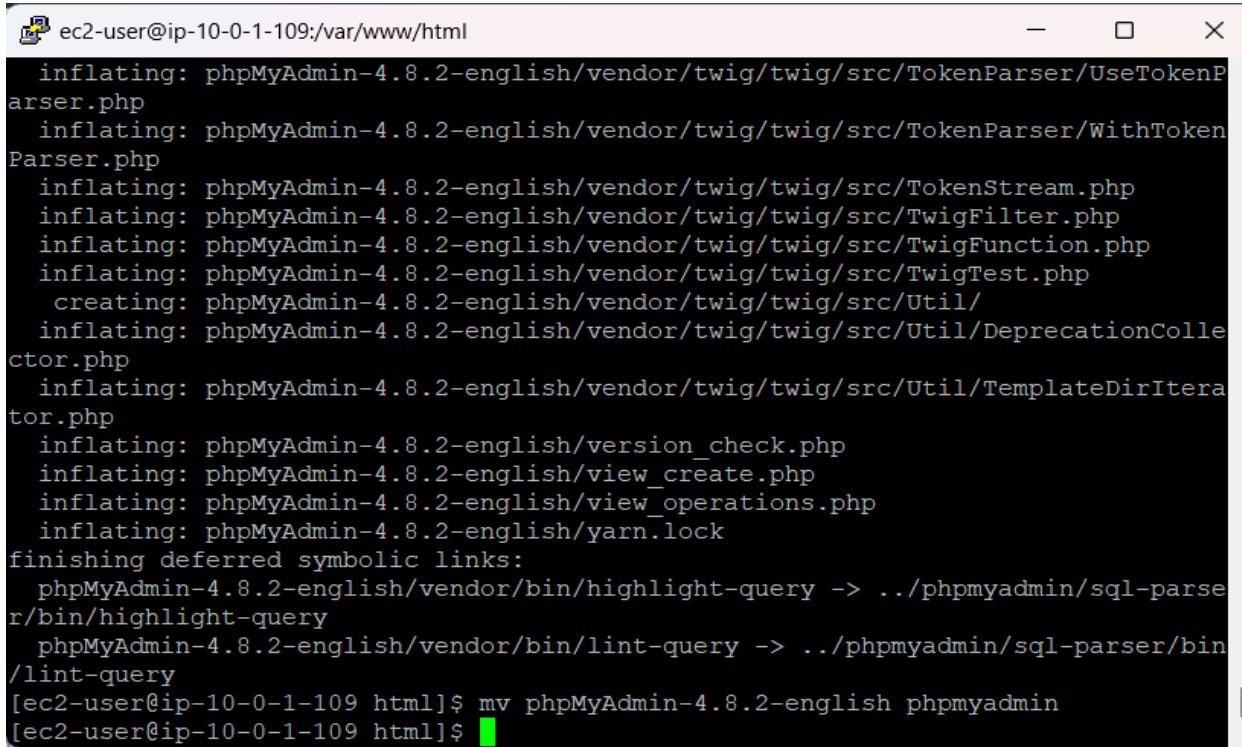
Student Id: 104181789.....

Tutorial time: 13PM-17PM Saturday.....

Date of submission: 15/07.....

IV. Some configuration needed

Download myphpAdmin



```
ec2-user@ip-10-0-1-109:/var/www/html
inflating: phpMyAdmin-4.8.2-english/vendor/twig/twig/src/TokenParser/UseTokenParser.php
inflating: phpMyAdmin-4.8.2-english/vendor/twig/twig/src/TokenParser/WithTokenParser.php
inflating: phpMyAdmin-4.8.2-english/vendor/twig/twig/src/TokenStream.php
inflating: phpMyAdmin-4.8.2-english/vendor/twig/twig/src/TwigFilter.php
inflating: phpMyAdmin-4.8.2-english/vendor/twig/twig/src/TwigFunction.php
inflating: phpMyAdmin-4.8.2-english/vendor/twig/twig/src/TwigTest.php
creating: phpMyAdmin-4.8.2-english/vendor/twig/twig/src/Util/
inflating: phpMyAdmin-4.8.2-english/vendor/twig/twig/src/Util/DeprecationCollector.php
inflating: phpMyAdmin-4.8.2-english/vendor/twig/twig/src/Util/TemplateDirIterator.php
inflating: phpMyAdmin-4.8.2-english/version_check.php
inflating: phpMyAdmin-4.8.2-english/view_create.php
inflating: phpMyAdmin-4.8.2-english/view_operations.php
inflating: phpMyAdmin-4.8.2-english/yarn.lock
finishing deferred symbolic links:
  phpMyAdmin-4.8.2-english/vendor/bin/highlight-query -> ../../phpmyadmin/sql-parser/bin/highlight-query
  phpMyAdmin-4.8.2-english/vendor/bin/lint-query -> ../../phpmyadmin/sql-parser/bin/lint-query
[ec2-user@ip-10-0-1-109 html]$ mv phpMyAdmin-4.8.2-english phpmyadmin
[ec2-user@ip-10-0-1-109 html]$
```

Create a Database table

The screenshot shows the phpMyAdmin interface for managing a MySQL database. The left sidebar lists databases: New, asm2db, information_schema, mysql, performance_schema, and sys. Under the asm2db database, there is a folder named 'asm2table' which contains a single table named 'asm2table'. The main panel displays the 'Structure' tab for the 'asm2table' table. The table has five columns: title (VARCHAR, 255), description (VARCHAR, 255), creationdate (DATE), keywords (VARCHAR, 255), and reference (VARCHAR, 255). The 'Default' dropdown for all columns is set to 'None'. The 'Collation' dropdown is also set to 'None' for all columns. The 'Attributes' column contains checkboxes for 'Null', 'Auto_increment', and 'Comments'. The 'Comments' column is empty for all rows. The 'Virtuality' column is set to 'M' for all rows. At the bottom right of the structure table, there are 'Preview SQL' and 'Save' buttons. Below the structure table, there is a 'Console' section which is currently empty. The bottom of the screen shows the Windows taskbar with various pinned icons and system status.

Download aws

The screenshot shows a terminal window titled "constants.php" with the following content:

```
C: > Users > Admin > Downloads > COS20019 Cloud > asm2 > photoalbum > constants.php > ...
4
5  ec2-user@ip-10-0-1-109: ~
6  -sdk-php/v3/download/aws.zip
7  --2023-07-15 03:23:56-- http://docs.aws.amazon.com/aws-sdk-php/v3/download/aws.
8  zip
9  Resolving docs.aws.amazon.com (docs.aws.amazon.com)... 108.159.227.99, 108.159.2
10 27.3, 108.159.227.51
11  Connecting to docs.aws.amazon.com (docs.aws.amazon.com)|108.159.227.99|:80... co
12  nected.
13  HTTP request sent, awaiting response... 301 Moved Permanently
14  Location: https://docs.aws.amazon.com/aws-sdk-php/v3/download/aws.zip [following
15  ]
16  --2023-07-15 03:23:56-- https://docs.aws.amazon.com/aws-sdk-php/v3/download/aws
17  .zip
18  Connecting to docs.aws.amazon.com (docs.aws.amazon.com)|108.159.227.99|:443...
19  HTTP request sent, awaiting response... 200 OK
20  Length: 5411810 (5.2M) [application/zip]
21  Saving to: '/var/www/html/aws.zip'
22
23  [ec2-user@ip-10-0-1-109 ~]$ aws-autoloader.php
24
25
26  photoalbum (this directory contains source files of the PhotoAlbum website)
27  uploads (this directory stores images before they are uploaded to S3, for more deets see photouploader.php)
28  album.php          (executable) display all images in DB
29  constants.php       Constants defined here
30  defaultstyle.css   CSS style for the website
31  mydb.php           Interact with RDS DB
32  photo.php          Photo object class
33  photouploader.php  (executable) upload image to S3 and RDS
```

The terminal shows the command "aws-autoloader.php" being run, which extracts the AWS SDK for PHP v3 into the "photoalbum" directory. A progress bar indicates the download speed is 12.4MB/s. A tooltip provides information about the AWS SDK for PHP.

At the bottom right, there is a message about deprecated extensions and a link to "Show Deprecated Extensions".