Assignment 2

Dang Khoa Le

Student ID: 103844421

Major: Bachelor of Software Engineering. Tutorial Wednesday, 06.30PM.

***Abstract*: This document presenting the demonstration for Assignment 2. The demonstration is based on AWS academy infrastructure.**

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

It has the following additional objectives:

1. Create IAM roles to enable EC2, Lambda, and S3 to interact with each other.

2. Restrict access to S3 using S3 bucket policy.

3. Create a lambda function.

4. Create a custom AMI.

5. Create a launch template based on your custom AMI.

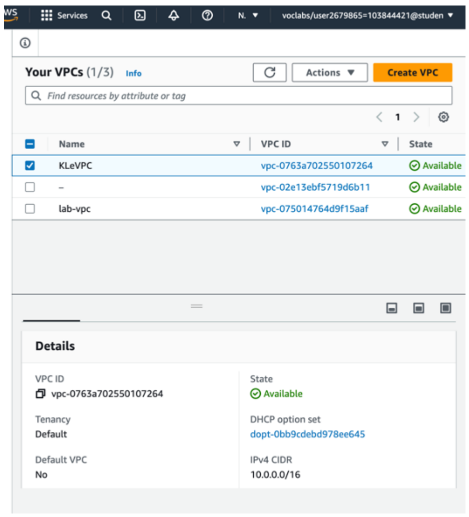
6. Create an auto scaling group across multiple Availability Zones with policies for scaling up and down.

7. Create an elastic load balancer to distribute service requests.

8. Access control and traffic limitations by using AWS NACLs.

I. CREATING A VPC:

The following images illustrate the configuration of the main VPC for this assignment (named as KLeVPC).



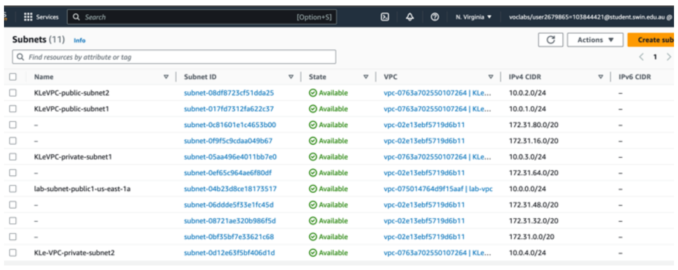
Your VPCs (1/3) Info

Name	VPC ID	State
KLeVPC	vpc-0763a702550107264	Available
–	vpc-02e13ebf5719d6b11	Available
lab-vpc	vpc-075014764d9f15aaf	Available

Details

VPC ID: vpc-0763a702550107264	State: Available
Tenancy: Default	DHCP option set: dopt-0bb9cdebd978ee645
Default VPC: No	IPv4 CIDR: 10.0.0.0/16

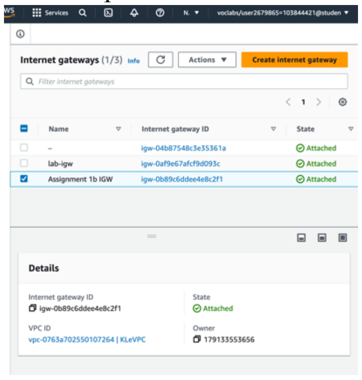
The VPC will be associated by 4 subnets (allocated in 2 subnet groups KLeVPC-rtb-private and KLeVPC-rtb-public). The 4 subnets are KLeVPC-public-subnet1 and KLeVPC-private-subnet1 (us-east1a), KLeVPC-public-subnet2 and KLeVPC-private-subnet2 (us-east1b), which have the IPv4 CIDR of 10.0.1.0/24, 10.0.3.0/24, 10.0.2.0/24, 10.0.4.0/24 by respectively.



Subnets (11) Info

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
KLeVPC-public-subnet2	subnet-08df8723cf51dda25	Available	vpc-0763a702550107264 | KLe...	10.0.2.0/24	–
KLeVPC-public-subnet1	subnet-017fd7312fa622c37	Available	vpc-0763a702550107264 | KLe...	10.0.1.0/24	–
–	subnet-0c81601e1c4653b00	Available	vpc-02e13ebf5719d6b11	172.31.80.0/20	–
–	subnet-0f9f5c9cdaa049b67	Available	vpc-02e13ebf5719d6b11	172.31.16.0/20	–
KLeVPC-private-subnet1	subnet-05aa496e4011bb7e0	Available	vpc-0763a702550107264 | KLe...	10.0.3.0/24	–
–	subnet-0ef65c964ae6f80df	Available	vpc-02e13ebf5719d6b11	172.31.64.0/20	–
lab-subnet-public1-us-east-1a	subnet-04b23d8ce18173517	Available	vpc-075014764d9f15aaf | lab-vpc	10.0.0.0/24	–
–	subnet-06ddde5f33e1fc45d	Available	vpc-02e13ebf5719d6b11	172.31.48.0/20	–
–	subnet-08721ae320b986f5d	Available	vpc-02e13ebf5719d6b11	172.31.32.0/20	–
–	subnet-0bf35bf7e33621c68	Available	vpc-02e13ebf5719d6b11	172.31.0.0/20	–
KLe-VPC-private-subnet2	subnet-0d12e63f5bf406d1d	Available	vpc-0763a702550107264 | KLe...	10.0.4.0/24	–

An Internet gateway was created before and named as 'Assignment 1b IGW' that attached to the main VPC. This Internet gateway will be associated to the public route table that linking with the 2 public subnets.



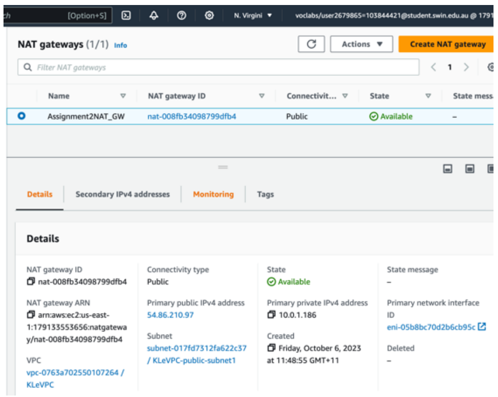
Internet gateways (1/3) Info

Name	Internet gateway ID	State
–	igw-04b87548c3e35361a	Attached
lab-igw	igw-0af9e67afcf9d093c	Attached
Assignment 1b IGW	igw-0b89c6ddee4e8c2f1	Attached

Details

Internet gateway ID: igw-0b89c6ddee4e8c2f1	State: Attached
VPC ID: vpc-0763a702550107264 | KLeVPC	Owner: 179133553656

A NAT-gateway also has been created with an attached Elastic IP address. This NAT gateway associates to the private route table that attached to the 2 private subnets. It allows private instances to be able to communicate with the public internet.



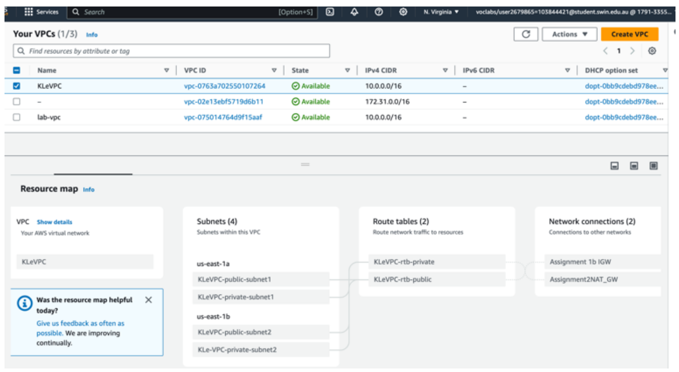
NAT gateways (1/1) Info

Name	NAT gateway ID	Connectivit...	State	State mess
Assignment2NAT_GW	nat-008fb34098799dfb4	Public	Available	–

Details

NAT gateway ID: nat-008fb34098799dfb4	Connectivity type: Public	State: Available	State message: –
NAT gateway ARN: arn:aws:ec2:us-east-1:179133553656:natgateway/nat-008fb34098799dfb4	Primary public IPv4 address: 54.86.210.97	Primary private IPv4 address: 10.0.1.186	Primary network interface ID: eni-05b8bc70d2b6cb95c
VPC: vpc-0763a702550107264 / KLeVPC	Subnet: subnet-017fd7312fa622c37 / KLeVPC-public-subnet1	Created: Friday, October 6, 2023 at 11:48:55 GMT+11	Deleted: –

Below is the completed Resource map for KLeVPC as indicating their components and how subnets, route tables and network connections (internet and NAT gateways) are integrated in this VPC.



Your VPCs (1/3) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set
KLeVPC	vpc-0763a702550107264	Available	10.0.0.0/16	–	dopt-0bb9cdebd978ee...
–	vpc-02e13ebf5719d6b11	Available	172.31.0.0/16	–	dopt-0bb9cdebd978ee...
lab-vpc	vpc-075014764d9f15aaf	Available	10.0.0.0/16	–	dopt-0bb9cdebd978ee...

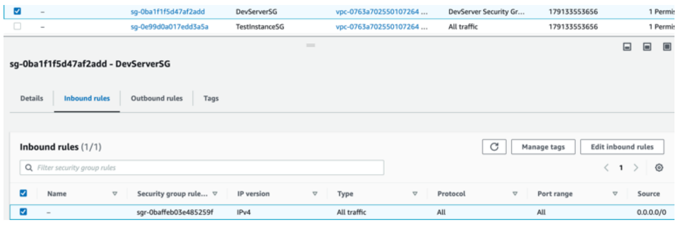
Resource map Info

VPC	Subnets (4)	Route tables (2)	Network connections (2)
KLeVPC	us-east-1a: KLeVPC-public-subnet1, KLeVPC-private-subnet1	KLeVPC-rtb-private	Assignment 1b IGW
	us-east-1b: KLeVPC-public-subnet2, KLe-VPC-private-subnet2	KLeVPC-rtb-public	Assignment2NAT_GW

II. Creating The Security Groups:

The following images illustrate the configuration of the 4 security groups for this assignment (with KLeVPC).

1. DevServerSG



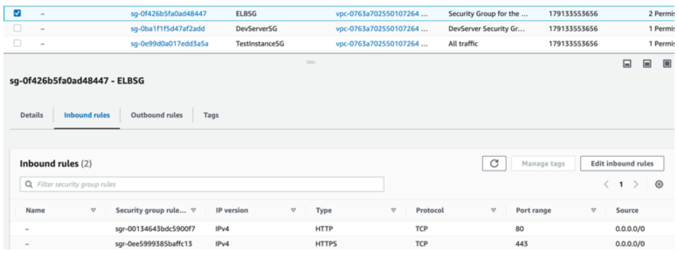
✓	–	sg-0ba1f1f5d47af2add	DevServerSG	vpc-0763a702550107264 ...	DevServer Security Gr...	179133553656	1 Permi
	–	sg-0e99d0a017edd3a5a	TestInstanceSG	vpc-0763a702550107264 ...	All traffic	179133553656	1 Permi

sg-0ba1f1f5d47af2add - DevServerSG

Inbound rules (1/1)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
–	sgr-0baffeb03e485259f	IPv4	All traffic	All	All	0.0.0.0/0

2. ELBSG



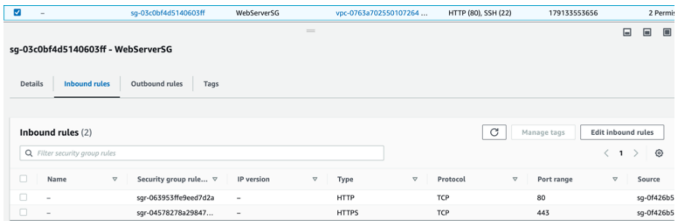
✓	–	sg-0f426b5fa0ad48447	ELBSG	vpc-0763a702550107264 ...	Security Group for the ...	179133553656	2 Permi
	–	sg-0ba1f1f5d47af2add	DevServerSG	vpc-0763a702550107264 ...	DevServer Security Gr...	179133553656	1 Permi
	–	sg-0e99d0a017edd3a5a	TestInstanceSG	vpc-0763a702550107264 ...	All traffic	179133553656	1 Permi

sg-0f426b5fa0ad48447 - ELBSG

Inbound rules (2)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
–	sgr-00134643bdc5900f7	IPv4	HTTP	TCP	80	0.0.0.0/0
–	sgr-0ee5999385baffc13	IPv4	HTTPS	TCP	443	0.0.0.0/0

3. WebServerSG



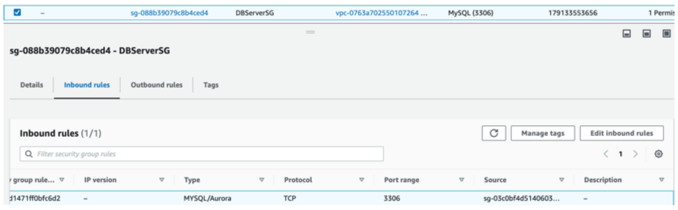
✓	–	sg-03c0bf4d5140603ff	WebServerSG	vpc-0763a702550107264 ...	HTTP (80), SSH (22)	179133553656	2 Permi

sg-03c0bf4d5140603ff - WebServerSG

Inbound rules (2)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
–	sgr-063953ffe9eed7d2a	–	HTTP	TCP	80	sg-0f426b5
–	sgr-04578278a29847...	–	HTTPS	TCP	443	sg-0f426b5

4. DBServerSG



✓	–	sg-088b39079c8b4ced4	DBServerSG	vpc-0763a702550107264 ...	MySQL (3306)	179133553656	1 Permi

sg-088b39079c8b4ced4 - DBServerSG

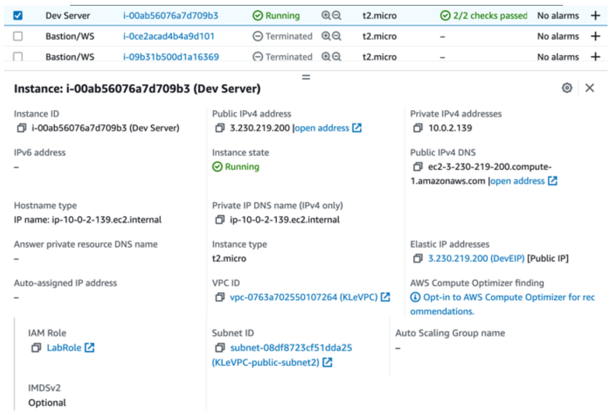
Inbound rules (1/1)

group rule...	IP version	Type	Protocol	Port range	Source	Description
d1471ff0bfc6d2	–	MYSQL/Aurora	TCP	3306	sg-03c0bf4d5140603...	–

These 4 security groups allow suitable traffics to satisfy the task functionality specifications as well as allows a proper security interfacing. Security groups including DevServerSG (for Dev Server EC2 instance, allows all traffics as this security group doesn't have to follow the least-privilege principle), ELBSG (for Elastic Load Balancer, allows all HTTP and HTTPS traffics from everywhere), WebServerSG (for the 2 Web Server instances within the private subnets, they allow incoming traffics from ELBSG via HTTP and HTTPS port), and DBServerSG (sourced by WebServerSG, for kle-database, allows MYSQL/Aurora traffics).

II. Create an Dev Server Instance:

These images illustrate the configuration for the Dev Server and its associations, this instance is needed to create the AMI images and also serve as the public allocation source for the phpMyAdmin and photo up/download functionalities at the later steps. They also has been attached to the IAM -LabRole.

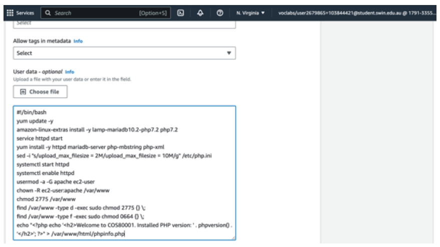


✓	Dev Server	i-00ab56076a7d709b3	Running	t2.micro	2/2 checks passed	No alarms	+
	Bastion/WS	i-0ce2acad4b4a9d101	Terminated	t2.micro	–	No alarms	+
	Bastion/WS	i-09b31b500d1a16369	Terminated	t2.micro	–	No alarms	+

Instance: i-00ab56076a7d709b3 (Dev Server)

Instance ID: i-00ab56076a7d709b3 (Dev Server)
Public IPv4 address: 3.230.219.200 |open address
Private IPv4 addresses: 10.0.2.139
IPv6 address: –
Instance state: Running
Public IPv4 DNS: ec2-3-230-219-200.compute-1.amazonaws.com |open address
Hostname type: IP name: ip-10-0-2-139.ec2.internal
Private IP DNS name (IPv4 only): ip-10-0-2-139.ec2.internal
Answer private resource DNS name: –
Instance type: t2.micro
Elastic IP addresses: 3.230.219.200 (DevEIP) [Public IP]
Auto-assigned IP address: –
VPC ID: vpc-0763a702550107264 (KLeVPC)
AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations.
IAM Role: LabRole
Subnet ID: subnet-08df8723cf51dda25 (KLeVPC-public-subnet2)
Auto Scaling Group name: –
IMDSv2: Optional

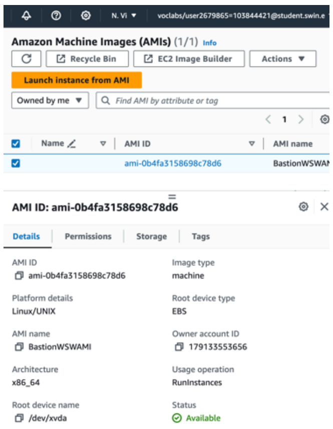
The Dev Server instance is allocated in KLeVPC's public subnet 2 (KLeVPC-public-subnet2), associated with the DevServerSG security groups and the DevEIP Elastic IP address (providing the fixed public IPv4 DNS for the instance). I also update the user data with the Apache http server and php, installed from the resource of Assignment 1b.



Allow tags in metadata: Select
User data - *optional*
Upload a file with your user data or enter it in the field.

```
#!/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
```

An AMI image also has been created from the Dev Server instance. The following images shows their configurations in detail.



Amazon Machine Images (AMIs) (1/1)

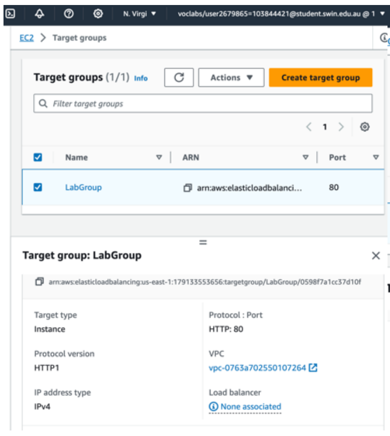
Name	AMI ID	AMI name
	ami-0b4fa3158698c78d6	BastionWSWAI

AMI ID: ami-0b4fa3158698c78d6

AMI ID: ami-0b4fa3158698c78d6
Image type: machine
Platform details: Linux/UNIX
Root device type: EBS
AMI name: BastionWSWAMI
Owner account ID: 179133553656
Architecture: x86_64
Usage operation: RunInstances
Root device name: /dev/xvda
Status: Available

III. Creating The Target Groups, Launch Template and Elastic Load Balancer:

The following images illustrate the configuration for the Target Group 'LabGroup'. The target group will use WebServerSG and the 2 public subnets, alongside with the HTTP port 80. It also serves the threshold value of 2 consecutive healthy and unhealthy checks.



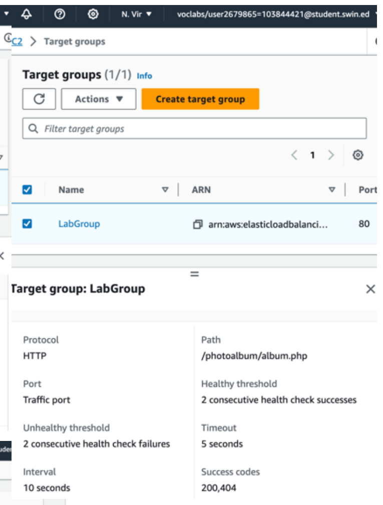
Target groups (1/1)

Name	ARN	Port
LabGroup	arn:aws:elasticloadbalanci...	80

Target group: LabGroup

arn:aws:elasticloadbalancing:us-east-1:179133553656:targetgroup/LabGroup/0598f7a1cc37d10f

Target type: Instance
Protocol : Port: HTTP: 80
Protocol version: HTTP1
VPC: vpc-0763a702550107264
IP address type: IPv4
Load balancer: None associated

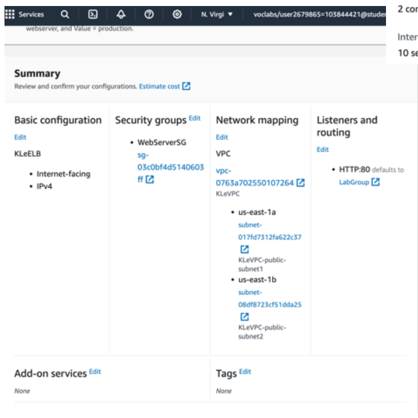


Target groups (1/1)

Name	ARN	Port
LabGroup	arn:aws:elasticloadbalanci...	80

Target group: LabGroup

Protocol: HTTP
Path: /photoalbum/album.php
Port: Traffic port
Healthy threshold: 2 consecutive health check successes
Unhealthy threshold: 2 consecutive health check failures
Timeout: 5 seconds
Interval: 10 seconds
Success codes: 200,404



webserver, and Value = production.

Summary

Review and confirm your configurations. Estimate cost

Basic configuration: KLeELB
- Internet-facing
- IPv4

Security groups:
- WebServerSG sg-03c0bf4d5140603ff

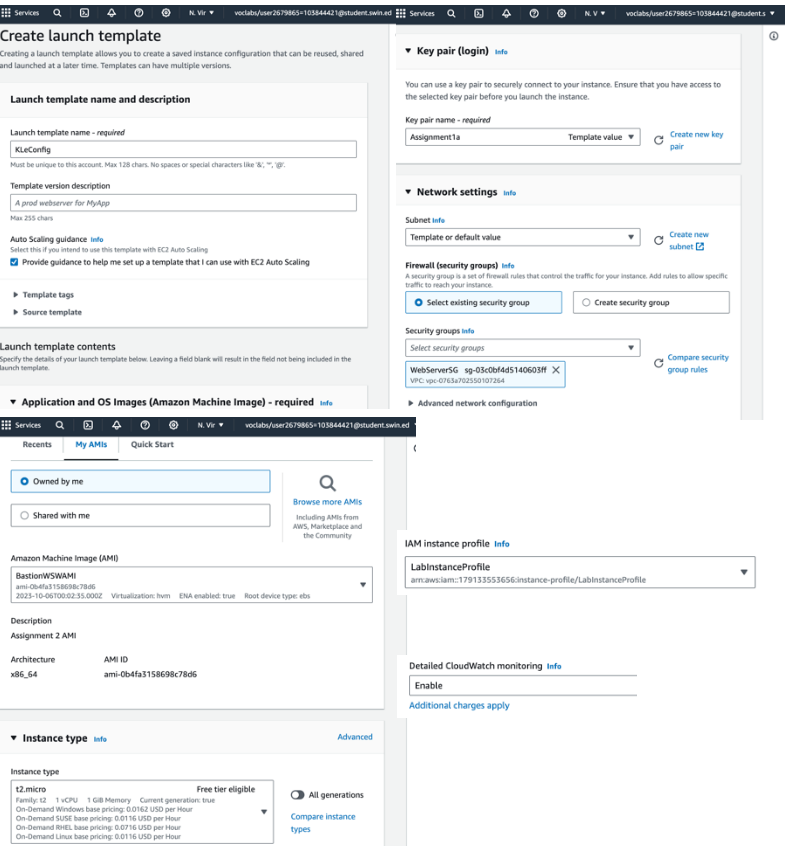
Network mapping: VPC vpc-0763a702550107264 KLeVPC
- us-east-1a subnet-017fd7312fa622c37 KLeVPC-public-subnet1
- us-east-1b subnet-08df8723cf51dda25 KLeVPC-public-subnet2

Listeners and routing:
- HTTP:80 defaults to LabGroup

Add-on services: None

Tags: None

Then, we launch template, named as KLeConfig, which will be used for the Auto Scaling Group in the upcoming step.


Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - required: KLeConfig

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description: A prod webserver for MyApp (Max 255 chars)

Auto Scaling guidance: Select this if you intend to use this template with EC2 Auto Scaling

☑ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

Template tags · Source template

Launch template contents: Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

Application and OS Images (Amazon Machine Image) - required

Recents · My AMIs · Quick Start

[✓ Owned by me] Shared with me

Amazon Machine Image (AMI): BastionWSWAMI, ami-0b4fa3158698c78d6, 2023-10-06T00:02:35.000Z, Virtualization: hvm, ENA enabled: true, Root device type: ebs

Description: Assignment 2 AMI

Architecture: x86_64 · AMI ID: ami-0b4fa3158698c78d6

Instance type: t2.micro (Free tier eligible). Family: t2, 1 vCPU, 1 GiB Memory, Current generation: true. On-Demand Windows base pricing: 0.0162 USD per Hour; On-Demand SUSE base pricing: 0.0116 USD per Hour; On-Demand RHEL base pricing: 0.0716 USD per Hour; On-Demand Linux base pricing: 0.0116 USD per Hour

Key pair (login): You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required: Assignment1a (Template value)

Network settings — Subnet: Template or default value

Firewall (security groups): A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[✓ Select existing security group] Create security group

Security groups: WebServerSG sg-03c0bf4d5140603ff, VPC: vpc-0763a702550107264

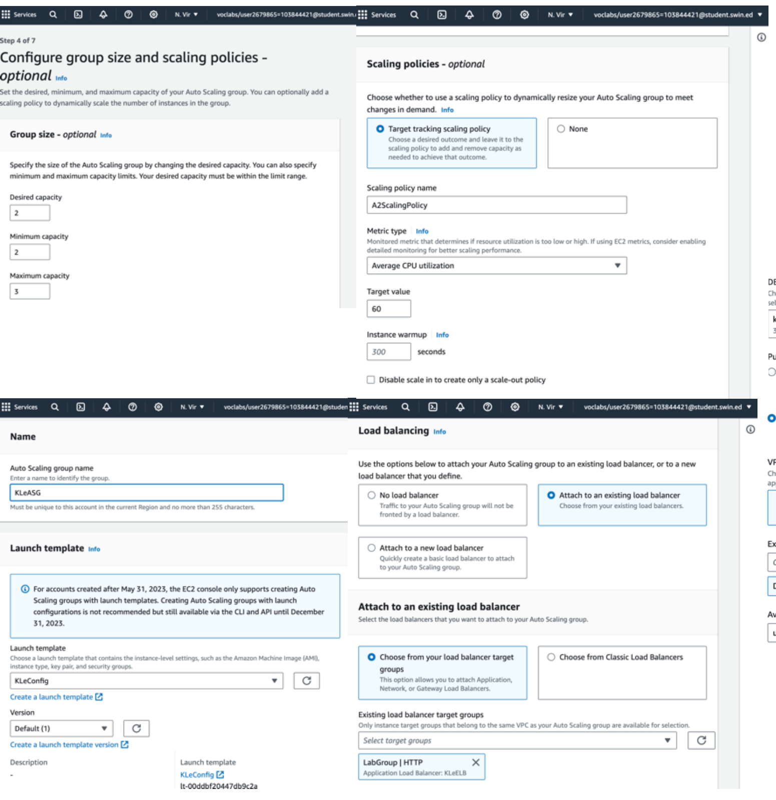
IAM instance profile: LabInstanceProfile, arn:aws:iam::179133553656:instance-profile/LabInstanceProfile

Detailed CloudWatch monitoring: Enable (Additional charges apply)

This template also uses the LabInstanceProfile (IAM LabRole) and allows CloudWatch monitoring, which can be used to keep track on any failure alarms of the 2 Web Server instances.

Nextly, we will create the Auto Scaling Group to initialise the 2 Bastion (Webserver instances) named as 'Bastion/WS'.

KLeASG will use the KLeConfig template, LabGroup target group, using desired and minimum capacity of 2 alongisde with the maximum of 3 in group size.


Step 4 of 7

Configure group size and scaling policies - optional

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

Group size - optional: Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity: 2 · Minimum capacity: 2 · Maximum capacity: 3

Scaling policies - optional: Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand.

[✓ Target tracking scaling policy] None

Scaling policy name: A2ScalingPolicy

Metric type: Average CPU utilization

Target value: 60

Instance warmup: 300 seconds

☐ Disable scale in to create only a scale-out policy

Name — Auto Scaling group name: KLeASG

Launch template: For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template: KLeConfig · Version: Default (1)

Description: - · Launch template: KLeConfig lt-00ddbf20447db9c2a

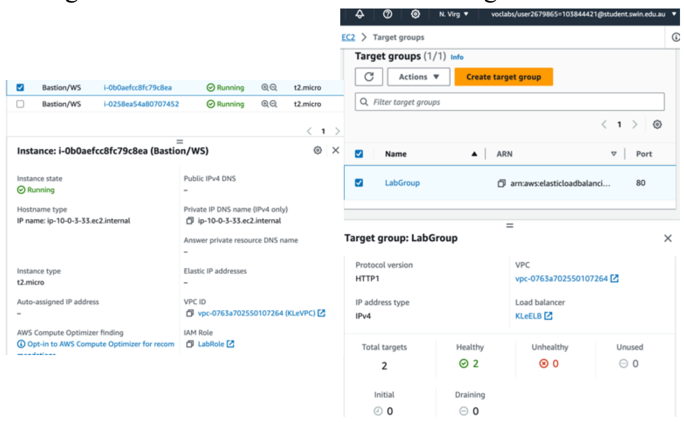
Load balancing: Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer [✓ Attach to an existing load balancer] Attach to a new load balancer

Attach to an existing load balancer: [✓ Choose from your load balancer target groups] Choose from Classic Load Balancers

Existing load balancer target groups: LabGroup | HTTP, Application Load Balancer: KLeELB

After finishing these steps, there will be 2 Bastion/WS instances automatically pops up on the EC2-Instacnes dashboard. After the 2 instances states showed to be completed, we have to check on the Target group to make sure that both instance shows up as healthy, which allows us to interact with the website via the ELB's DNS url at the later stages. We should also have to make sure that IAM - LabRole is assigned to the Bastion instances at this stage.


	Name	Instance ID	State		Type
☑	Bastion/WS	i-0b0aefcc8fc79c8ea	Running		t2.micro
☐	Bastion/WS	i-0258ea54a80707452	Running		t2.micro

Instance: i-0b0aefcc8fc79c8ea (Bastion/WS)

Instance state: Running · Public IPv4 DNS: –

Hostname type: IP name: ip-10-0-3-33.ec2.internal · Private IP DNS name (IPv4 only): ip-10-0-3-33.ec2.internal

Instance type: t2.micro · Elastic IP addresses: –

Auto-assigned IP address: – · VPC ID: vpc-0763a702550107264 (KLeVPC)

IAM Role: LabRole

EC2 > Target groups — Target groups (1/1)

Name	ARN	Port
LabGroup	arn:aws:elasticloadbalanci...	80

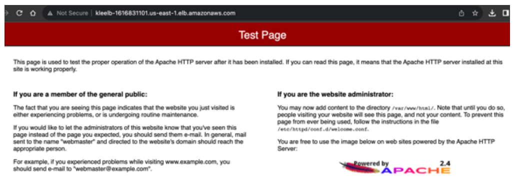
Target group: LabGroup

Protocol version: HTTP1 · VPC: vpc-0763a702550107264

IP address type: IPv4 · Load balancer: KLeELB

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	2	0	0	0	0

We can also try to access the ELB's DNS at this stage to ensure they operates properly, which is accessible via: KLeELB-1616831101.us-east-1.elb.amazonaws.com


kleelb-1616831101.us-east-1.elb.amazonaws.com

Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

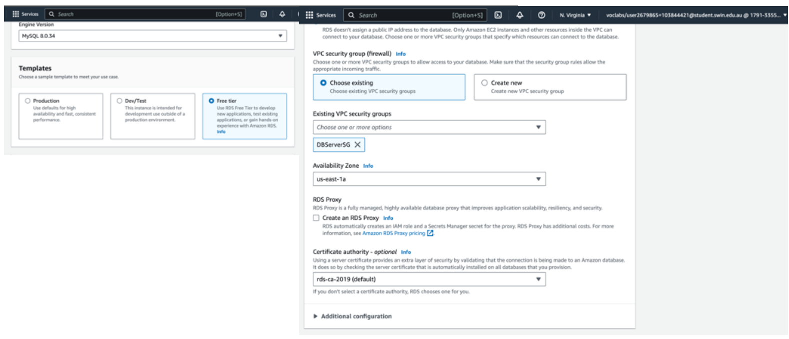
If you are the website administrator:

You may now add content to the directory /var/www/html/. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf.

You are free to use the image below on web sites powered by the Apache HTTP Server:

IV. *Creating The RDS Database:*

The following images illustrate the configuration for the RDS database and its associations.


Engine Version: MySQL 8.0.34

Templates: Production [✓ Free tier] Dev/Test

VPC security group (firewall): [✓ Choose existing] Create new

Existing VPC security groups: DBServerSG

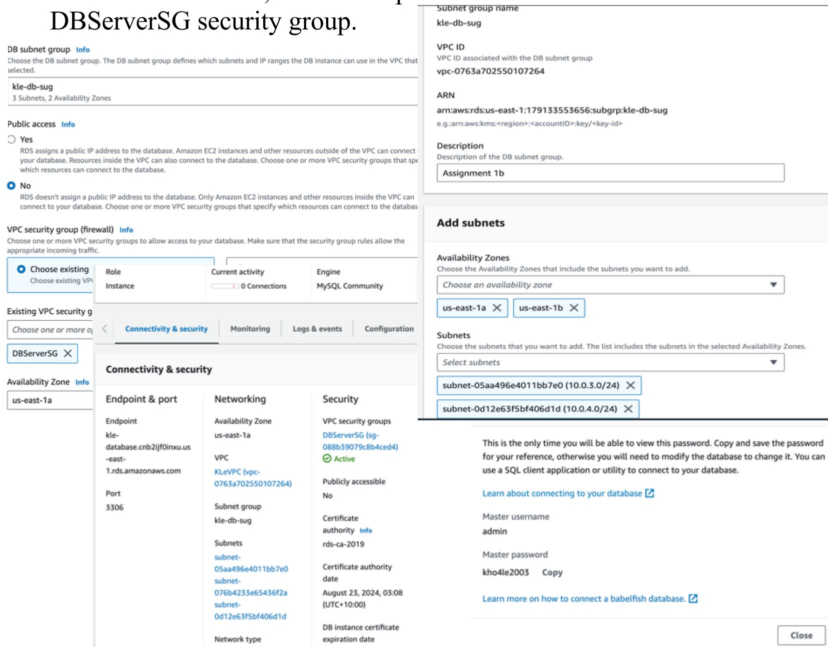
Availability Zone: us-east-1a

RDS Proxy: ☐ Create an RDS Proxy

Certificate authority - optional: rds-ca-2019 (default)

Additional configuration

The RDS database is named as kle-database, below is its detailed connectivity and endpoint. The database is associated with the KLeVPC, KLeVPC-private-subnet1 and use the DBServerSG security group.


DB subnet group: kle-db-sug, 3 Subnets, 2 Availability Zones

Public access: Yes [✓ No]

RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall): Choose existing · Existing VPC security groups: DBServerSG · Availability Zone: us-east-1a

Role: Instance · Current activity: 0 Connections · Engine: MySQL Community

Connectivity & security · Monitoring · Logs & events · Configuration

Connectivity & security

Endpoint & port	Networking	Security
Endpoint: kle-database.cnb2ijf0inxu.us-east-1.rds.amazonaws.com	Availability Zone: us-east-1a	VPC security groups: DBServerSG (sg-088b39079c8b4ced4) Active
Port: 3306	VPC: KLeVPC (vpc-0763a702550107264)	Publicly accessible: No
	Subnet group: kle-db-sug	Certificate authority: rds-ca-2019
	Subnets: subnet-05aa496e4011bb7e0, subnet-076b4233e65436f2a, subnet-0d12e63f5bf406d1d	Certificate authority date: August 23, 2024, 03:08 (UTC+10:00)
	Network type	DB instance certificate expiration date

Subnet group name: kle-db-sug

VPC ID: vpc-0763a702550107264

ARN: arn:aws:rds:us-east-1:179133553656:subgrp:kle-db-sug

Description: Assignment 1b

Add subnets — Availability Zones: us-east-1a, us-east-1b

Subnets: subnet-05aa496e4011bb7e0 (10.0.3.0/24), subnet-0d12e63f5bf406d1d (10.0.4.0/24)

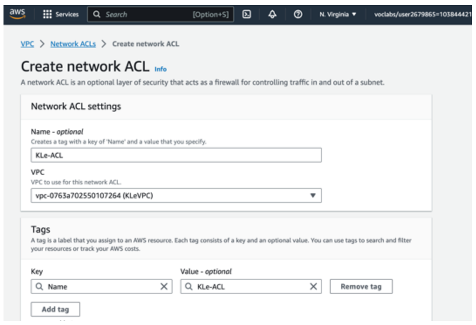
This is the only time you will be able to view this password. Copy and save the password for your reference, otherwise you will need to modify the database to change it. You can use a SQL client application or utility to connect to your database.

Master username: admin

Master password: kho4le2003

In advanced to this RDS database, I also have created a subnet group (kle-db-sug) of 2 private subnet (KLeVPC-private-subnet1 and KLeVPC-private-subnet2), this subnet group also use 2 availability zones us-east-1a and us-east-1b, thus it is compatible with the Web Server.

V. Create Network ACL:


aws
Services
Search
[Option+S]
N. Virginia
voclabs/user2679865=103844421
VPC > Network ACLs > Create network ACL

Create network ACL Info

A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.

Network ACL settings

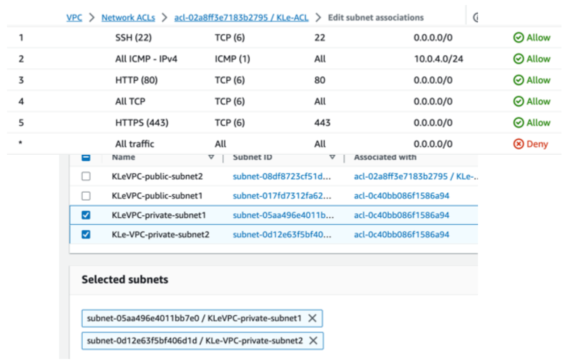
Name - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
KLe-ACL

VPC
VPC to use for this network ACL.
vpc-0763a702550107264 (KLeVPC)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*
Name	KLe-ACL

The following images illustrate the configuration for the Network ACL (KLe-ACL) and its inbound rules (Allows SSH, HTTP, HTTPS and ICMP traffics). This NACL will be associated to the 2 private subnets as indicated in Assignment 2 specifications.


VPC > Network ACLs > acl-02a8ff3e7183b2795 / KLe-ACL > Edit subnet associations

1	SSH (22)	TCP (6)	22	0.0.0.0/0	Allow
2	All ICMP - IPv4	ICMP (1)	All	10.0.4.0/24	Allow
3	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
4	All TCP	TCP (6)	All	0.0.0.0/0	Allow
5	HTTPS (443)	TCP (6)	443	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

	Name	Subnet ID	Associated with
☐	KLeVPC-public-subnet2	subnet-08df8723cf51d...	acl-02a8ff3e7183b2795 / KLe-...
☐	KLeVPC-public-subnet1	subnet-017fd7312fa62...	acl-0c40bb086f1586a94
☑	KLeVPC-private-subnet1	subnet-05aa496e4011b...	acl-0c40bb086f1586a94
☑	KLe-VPC-private-subnet2	subnet-0d12e63f5bf40...	acl-0c40bb086f1586a94

Selected subnets

subnet-05aa496e4011bb7e0 / KLeVPC-private-subnet1
subnet-0d12e63f5bf406d1d / KLe-VPC-private-subnet2

VI. Installing phpMyAdmin:

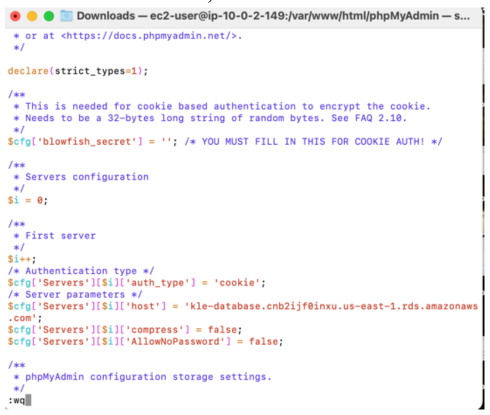
The following images illustrate how I installed phpMyAdmin to the Dev Server.

From the command line of my local terminal, I have to download the phpmyadmin file, then unzip and move them appropriately to the specified directory. Then, logging to my EC2 instance by using the SFTP method and my key pair (pre-named as Assignment1a) on Cyberduck, then moving to the var/www/html/phpmyadmin, locating the file config.sample.inc.php and change the name to, config.inc.php. After that, open the file and edit the line on my terminal to the appropriate RDS endpoint (nano file editor):

$cfg['Servers'][$i]['host'] = 'localhost';

Into:

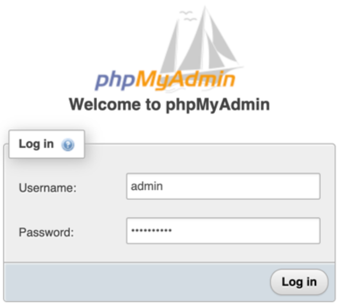
$cfg['Servers'][$i]['host'] = ‘kle-database.cnb2ijf0inxu.us-east-1.rds.amazonaws.com’;


Downloads — ec2-user@ip-10-0-2-149:/var/www/html/phpMyAdmin — s...

```
 * or at <https://docs.phpmyadmin.net/>.
 */

declare(strict_types=1);

/**
 * This is needed for cookie based authentication to encrypt the cookie.
 * Needs to be a 32-bytes long string of random bytes. See FAQ 2.10.
 */
$cfg['blowfish_secret'] = ''; /* YOU MUST FILL IN THIS FOR COOKIE AUTH! */

/**
 * Servers configuration
 */
$i = 0;

/**
 * First server
 */
$i++;
/* Authentication type */
$cfg['Servers'][$i]['auth_type'] = 'cookie';
/* Server parameters */
$cfg['Servers'][$i]['host'] = 'kle-database.cnb2ijf0inxu.us-east-1.rds.amazonaws
.com';
$cfg['Servers'][$i]['compress'] = false;
$cfg['Servers'][$i]['AllowNoPassword'] = false;

/**
 * phpMyAdmin configuration storage settings.
 */
:wq
```

Then I am able to enter the username and password of my DB to login to phpMyAdmin via my Elastic Load Balancer’s Public DNS:

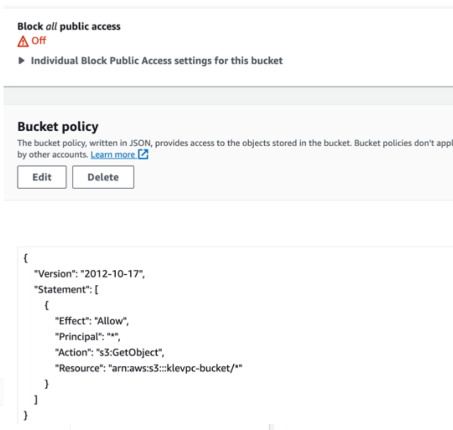
http://KLeELB-1616831101.us-east-1.elb.amazonaws.com/phpmyadmin/



Username:	admin
Password:	••••••••••

VII. Creating S3 Bucket:

The following images illustrate the configuration for the S3 Bucket named ‘klevpc-bucket’.

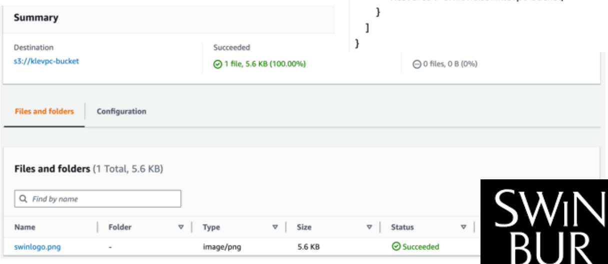

Block *all* public access
Off
Individual Block Public Access settings for this bucket

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't appl by other accounts. Learn more

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::klevpc-bucket/*"
    }
  ]
}
```

A Swinburne logo is uploaded onto the bucket as above while the bucket’s access policy also has been modified to be publicly accessible.


Summary

Destination	Succeeded	
s3://klevpc-bucket	1 file, 5.6 KB (100.00%)	0 files, 0 B (0%)

Files and folders (1 Total, 5.6 KB)

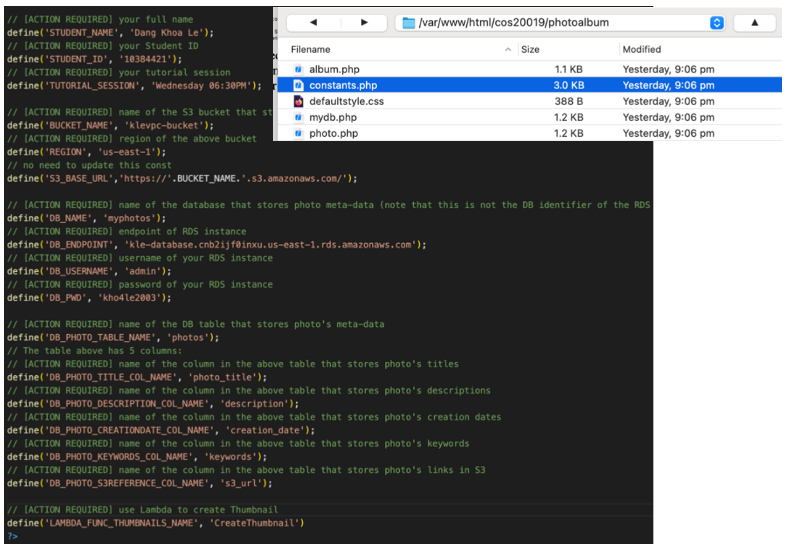
Name	Folder	Type	Size	Status
swinlogo.png	-	image/png	5.6 KB	Succeeded

The Swinburne logo object, also has enabled to be shared with presigned URL: https://klevpc-bucket.s3.amazonaws.com/swinlogo.png

VIII. Modifying phpMyAdmin Page:

The following images illustrate the modification for the http web-page phpMyAdmin.

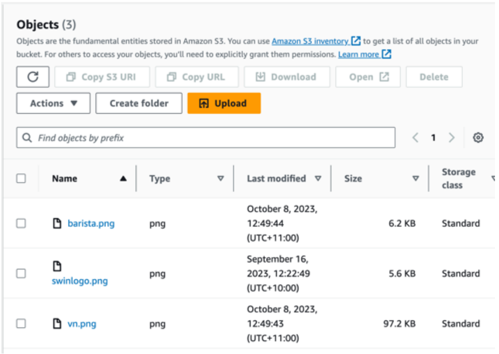
Below are my modification for the file php.constants (edited by Visual Studio Code) and the new table storing my new images named as myphotos.



Filename	Size	Modified
album.php	1.1 KB	Yesterday, 9:06 pm
constants.php	3.0 KB	Yesterday, 9:06 pm
defaultstyle.css	388 B	Yesterday, 9:06 pm
mydb.php	1.2 KB	Yesterday, 9:06 pm
photo.php	1.2 KB	Yesterday, 9:06 pm

```
// [ACTION REQUIRED] your full name
define('STUDENT_NAME', 'Dang Khoa Le');
// [ACTION REQUIRED] your Student ID
define('STUDENT_ID', '10384421');
// [ACTION REQUIRED] your tutorial session
define('TUTORIAL_SESSION', 'Wednesday 06:30PM');

// [ACTION REQUIRED] name of the S3 bucket that st
define('BUCKET_NAME', 'klevpc-bucket');
// [ACTION REQUIRED] region of the above bucket
define('REGION', 'us-east-1');
// no need to update this const
define('S3_BASE_URL','https://'.BUCKET_NAME.'.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
define('DB_NAME', 'myphotos');
// [ACTION REQUIRED] endpoint of RDS instance
define('DB_ENDPOINT', 'kle-database.cnb2ijf0inxu.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', 'kho4le2003');

// [ACTION REQUIRED] name of the DB table that stores photo's meta-data
define('DB_PHOTO_TABLE_NAME', 'photos');
// 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', 'photo_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', 'creation_date');
// [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', 's3_url');

// [ACTION REQUIRED] use Lambda to create Thumbnail
define('LAMBDA_FUNC_THUMBNAILS_NAME', 'CreateThumbnail')
?>
```

Eventually, the S3 bucket will then store 3 images in total which they will be resized by Lambda.



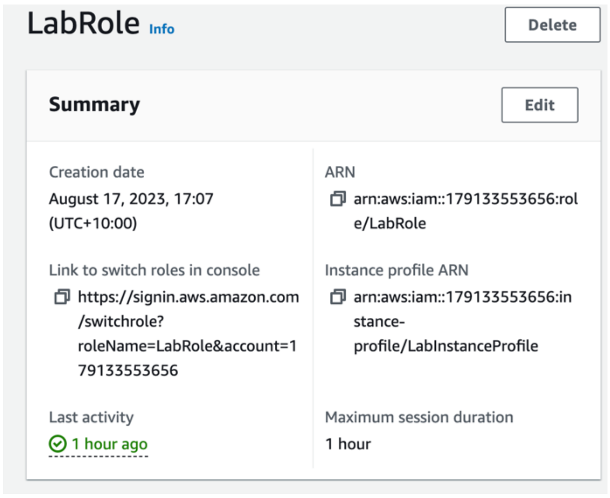
Objects (3)

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

Name	Type	Last modified	Size	Storage class
barista.png	png	October 8, 2023, 12:49:44 (UTC+11:00)	6.2 KB	Standard
swinlogo.png	png	September 16, 2023, 12:22:49 (UTC+10:00)	5.6 KB	Standard
vn.png	png	October 8, 2023, 12:49:43 (UTC+11:00)	97.2 KB	Standard

IX. IAM Role and Lambda Configuration:

An IAM Role created at default (LabRole) has been previously assigned to allow EC2, Lambda and S3 bucket interaction functioned properly.



LabRole Info

Summary

Creation date
August 17, 2023, 17:07 (UTC+10:00)

ARN
arn:aws:iam::179133553656:role/LabRole

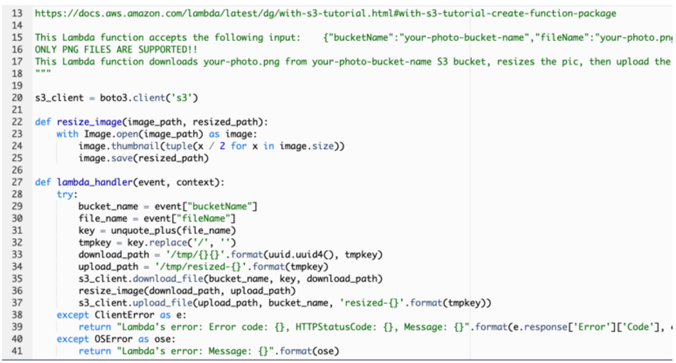
Link to switch roles in console
https://signin.aws.amazon.com/switchrole?roleName=LabRole&account=179133553656

Instance profile ARN
arn:aws:iam::179133553656:instance-profile/LabInstanceProfile

Last activity
1 hour ago

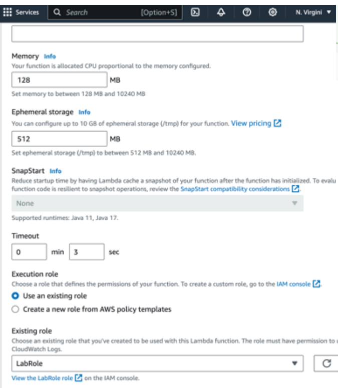
Maximum session duration
1 hour

Lambda named "CreateThumbnail" is configured as below, which use LabRole to integrate with EC2 and S3 bucket. Photos uploaded from the website is resized into appropriate (smaller) size by lambda function. The lambda contains codebase - package provided by default AWS (of this unit) which contains the library and full code resources to resize images and up/download images to S3 (png source based images preferred).



```
https://docs.aws.amazon.com/lambda/latest/dg/with-s3-tutorial.html#with-s3-tutorial-create-function-package

This Lambda function accepts the following input:    {"bucketName":"your-photo-bucket-name","fileName":"your-photo.pn
ONLY PNG FILES ARE SUPPORTED!!
This Lambda function downloads your-photo.png from your-photo-bucket-name S3 bucket, resizes the pic, then upload the
"""

s3_client = boto3.client('s3')

def resize_image(image_path, resized_path):
    with Image.open(image_path) as image:
        image.thumbnail(tuple(x / 2 for x in image.size))
        image.save(resized_path)

def lambda_handler(event, context):
    try:
        bucket_name = event["bucketName"]
        file_name = event["fileName"]
        key = unquote_plus(file_name)
        tmpkey = key.replace('/', '')
        download_path = '/tmp/{}{}'.format(uuid.uuid4(), tmpkey)
        upload_path = '/tmp/resized-{}'.format(tmpkey)
        s3_client.download_file(bucket_name, key, download_path)
        resize_image(download_path, upload_path)
        s3_client.upload_file(upload_path, bucket_name, 'resized-{}'.format(tmpkey))
    except ClientError as e:
        return "Lambda's error: Error code: {}, HTTPStatusCode: {}, Message: {}".format(e.response['Error']['Code'],
    except OSError as ose:
        return "Lambda's error: Message: {}".format(ose)
```

A test code also has been implemented and returns successful.



Memory Info
Your function is allocated CPU proportional to the memory configured.
128 MB
Set memory to between 128 MB and 10240 MB

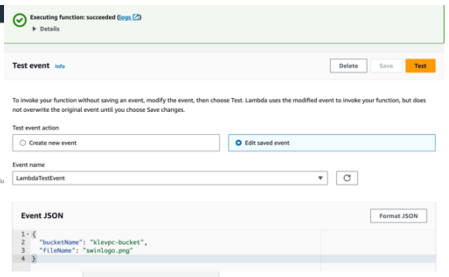
Ephemeral storage Info
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing
512 MB
Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

SnapStart Info
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evalu function code is resilient to snapshot operations, review the SnapStart compatibility considerations.
None
Supported runtimes: Java 11, Java 17.

Timeout
0 min 3 sec

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
[✓ Use an existing role]
Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
LabRole
View the LabRole role on the IAM console.



Executing function: succeeded (logs)
Details

Test event Info

To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.

Test event action
Create new event [✓ Edit saved event]

Event name
LambdaTestEvent

Event JSON

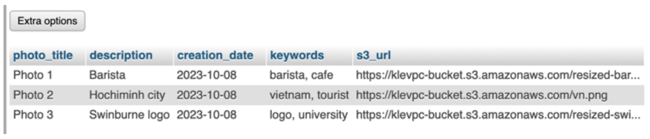
```
{
  "bucketName": "klevpc-bucket",
  "fileName": "swinlogo.png"
}
```

The resized swinlogo, barista and vn png photos by Lambda can be accessible via these url:

https://klevpc-bucket.s3.amazonaws.com/resized-swinlogo.png
https://klevpc-bucket.s3.amazonaws.com/resized-barista.png
https://klevpc-bucket.s3.amazonaws.com/vn.png

X. Accessing The Photo:

In similar to Assignment 1b, we have to navigate to phpmyadmin and construct a proper table for the photo demonstration. Make sure that they are appropriately formatted and named, while using png is preferable with the integration of the Lambda code.

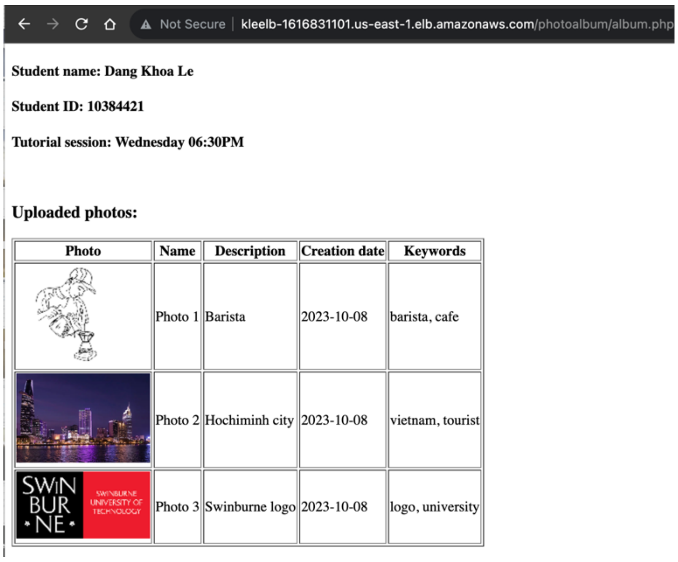


photo_title	description	creation_date	keywords	s3_url
Photo 1	Barista	2023-10-08	barista, cafe	https://klevpc-bucket.s3.amazonaws.com/resized-bar...
Photo 2	Hochiminh city	2023-10-08	vietnam, tourist	https://klevpc-bucket.s3.amazonaws.com/vn.png
Photo 3	Swinburne logo	2023-10-08	logo, university	https://klevpc-bucket.s3.amazonaws.com/resized-swi...

Extra options

Apparently, I can access all of my uploaded photos by the following http link:

http://kleelb-1616831101.us-east-1.elb.amazonaws.com/cos20019/photoalbum/album.php



Not Secure | kleelb-1616831101.us-east-1.elb.amazonaws.com/photoalbum/album.php

Student name: Dang Khoa Le

Student ID: 10384421

Tutorial session: Wednesday 06:30PM

Uploaded photos:

Photo	Name	Description	Creation date	Keywords
	Photo 1	Barista	2023-10-08	barista, cafe
	Photo 2	Hochiminh city	2023-10-08	vietnam, tourist
	Photo 3	Swinburne logo	2023-10-08	logo, university

(Various figures in this project may simultaneously demonstrate irrelevant topic as they are used as testing, please ignore).

XI. Possible Systematic Errors:

Target Group (LabGroup) could return with unhealthy health check for the 2 Web server instances when the port is not properly configured as 80 (HTTP) or the path is not corrected to be '/albumphoto/album.php', also make sure to set the success code to be 200 (add 404 if the health check still fails).

Make sure that the Elastic Load Balancer (KLeELB) uses the AMI created by Dev Server, or else the ELB's DNS would not being able to be accessed via web browser.

CloudWatch monitor can also being implemented to check for any possible failures during Assignment 2's Set-up.