

Assignment 1 – Part B Creating and deploying Photo Album website onto a simple AWS infrastructure

Cos20019 | Cloud computing architecture

Student name: Phuthai Hemathulintra

Student ID: 103804205

Lab Section: Wednesday 6:30 PM

Date of report: 14/09/2023

INTRODUCTION

The project of Photo Album website is implemented by AWS infrastructure. The AWS cloud service varidly provides developer service and existing resource to create from setting up VPC to enable the safe deployment of a web application to linking S3 bucket which store the photo objects. AWS networking has many different aspects implementing Network Access Control List (NACLs) and Security Groups, integrating Amazon RDS, and setting up routing tables. In order to show secure communication inside the VPC, SSH connectivity is also established between instances situated in public and private subnets.

The need to uphold security, scalability, and high availability standards—all of which are crucial in the deployment of contemporary cloud-based applications—is emphasized throughout this scenario. The project's comprehensive nature makes it an exemplary reference for AWS practitioners, providing insights into VPC architecture, network segmentation, and secure data management. Keywords: VPC, NACL, Security Group, Routing Table, SSH, RDS. By following the following implementation steps, this project can be pictured in a vivid demonstration. The following link is the path to photo album website. Website link: <http://ec2-44-218-82-104.compute-1.amazonaws.com/cos20019/photoalbum/album.php>

IMPLEMENTATION STEPS

Step1: Create VPC

- Create VPC select VPC only and allocate to 10.0.0.0/16

The screenshot shows the AWS VPC dashboard. A green success message at the top says "You successfully created vpc-008261e9c696b48c2 / PHemathulintraVPC". Below it, the VPC details are listed:

Details		Info	
VPC ID	vpc-008261e9c696b48c2	State	Available
Tenancy	Default	DNS hostnames	Disabled
Default VPC	No	DNS resolution	Enabled
Network Address Usage metrics	Route 53 Resolver DNS Firewall rule groups	Main route table	rtb-0b14edd450d49df1
Disabled	Failed to load rule groups	IPv6 pool	-
		Owner ID	474170150111

At the bottom, there are links for CloudShell, Feedback, and Language, along with a copyright notice: "© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

Figure 1 – create VPC

- Setting VPC, enable DNS resolution and hostnames

The screenshot shows the "Edit VPC settings" page for the previously created VPC. A modal window titled "Introducing the new edit VPC settings experience" provides information about the changes. The "DNS settings" section contains two checked checkboxes:

- Enable DNS resolution [Info](#)
- Enable DNS hostnames [Info](#)

At the bottom, there are links for CloudShell, Feedback, and Language, along with a copyright notice: "© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

Figure 2 – edit VPC DNS

Step2: Create subnets

Subnet	Ip	Availability zone
Public Subnet 1	10.0.1.0/24	Us-east-1a
Private Subnet 1	10.0.3.0/24	Us-east-1a
Public Subnet 2	10.0.2.0/24	Us-east-1b
Private Subnet 2	10.0.4.0/24	Us-east-1b

The screenshot shows a confirmation message: "You have successfully created 4 subnets: subnet-007091b9683040759, subnet-0258d646ebc3c001, subnet-0235e7928fb64fed0, subnet-0235e7928fb64fed0". Below it, a table lists the subnets:

Name	Subnet ID	Status	VPC	IPv4 CIDR
Public subnet 2	subnet-0235e7928fb64fed0	Available	vpc-008261e9c696b48c2	10.0.2.0/24
Private subnet 1	subnet-0258d646ebc3c001	Available	vpc-008261e9c696b48c2	10.0.3.0/24
Public subnet 1	subnet-007091b9683040759	Available	vpc-008261e9c696b48c2	10.0.1.0/24
Private subnet 2	subnet-0235e7928fb64fed0	Available	vpc-008261e9c696b48c2	10.0.4.0/24

At the bottom, there are links for CloudShell, Feedback, and Language, along with a copyright notice: "© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

Figure 3 – public-private subnets

Step3: Create an internet gateway and attach it to the VPC

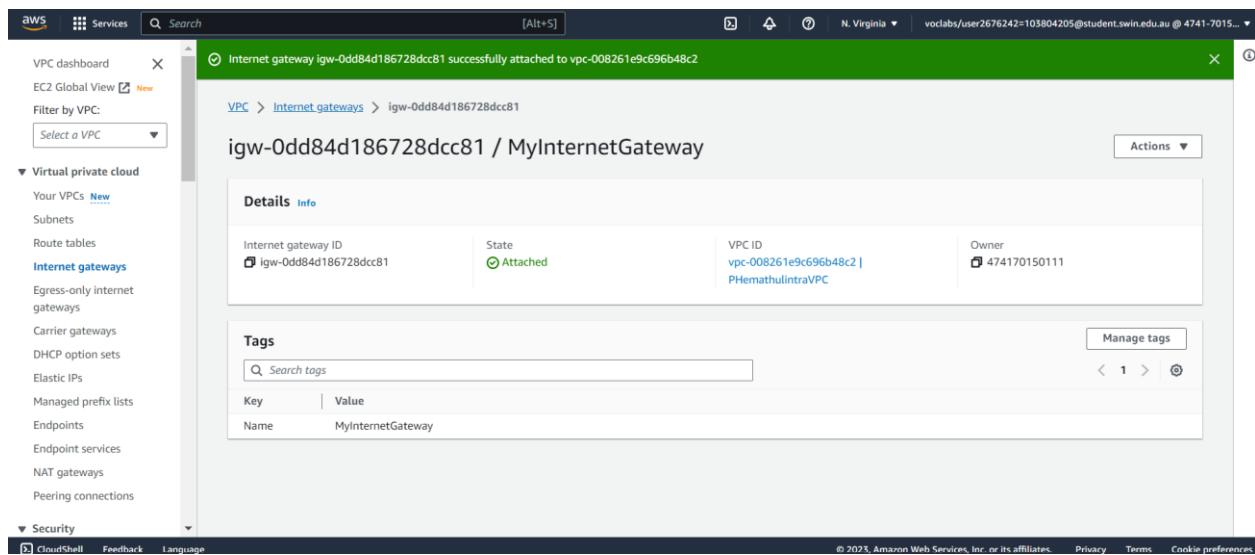


Figure 4 – create internet gateway and attach to VPC

Step4: Create 2 routing tables.

1) Public routing table

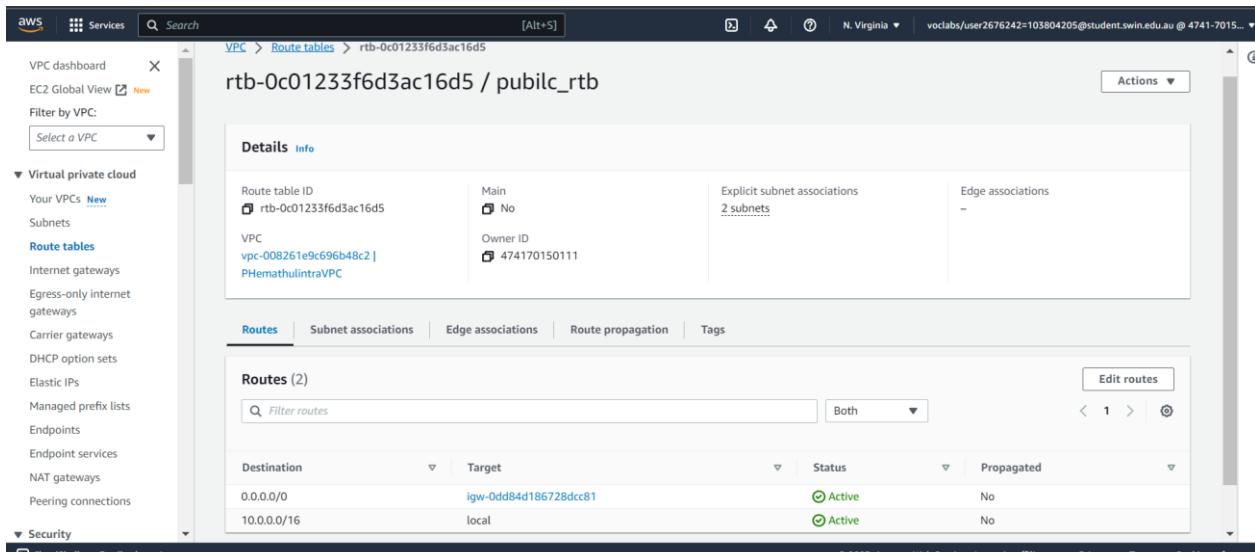


Figure 5 – create public route table

2) Private route table

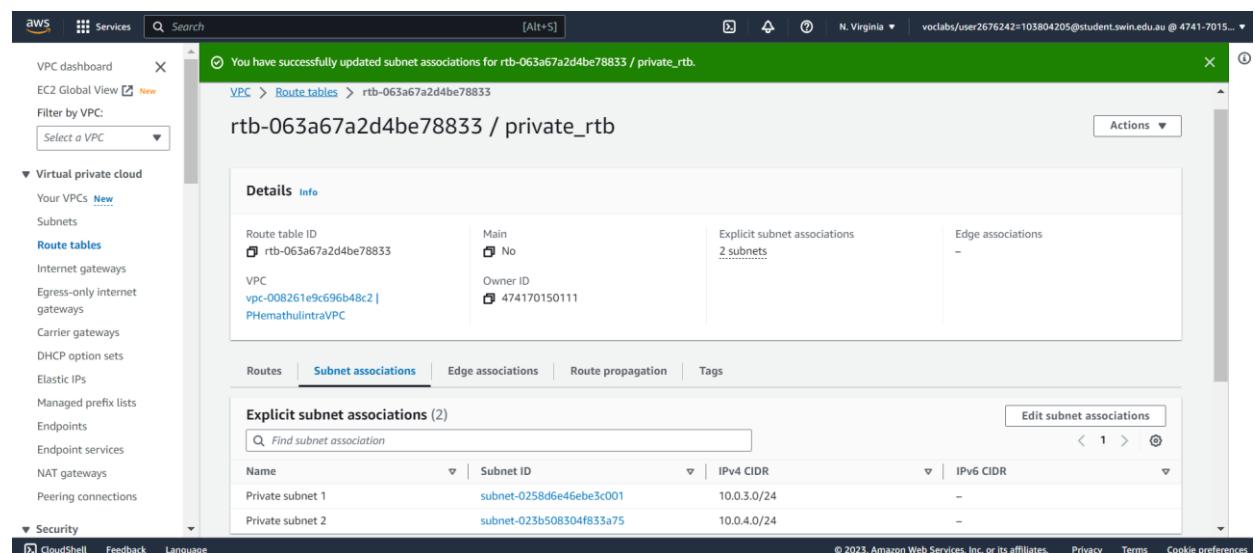


Figure 6 – create private route table

- Edit route of the Public route table to internet gateway and assign destination to 0.0.0.0/0

The screenshot shows the AWS VPC Route Tables page. The selected route table is 'rtb-0c01233f6d3ac16d5 / public_rtb'. The 'Routes' tab is active, displaying two entries:

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0dd84d186728dcc81	Active	No
10.0.0.16	local	Active	No

Figure 7 – edit route on public route table

- Associated appropriate subnets to the route tables

The screenshot shows the AWS VPC Route Tables page. The selected route table is 'public_rtb'. The 'Details' section shows it has 2 subnets associated.

Figure 8 – two route tables

- Check the VPC resource map

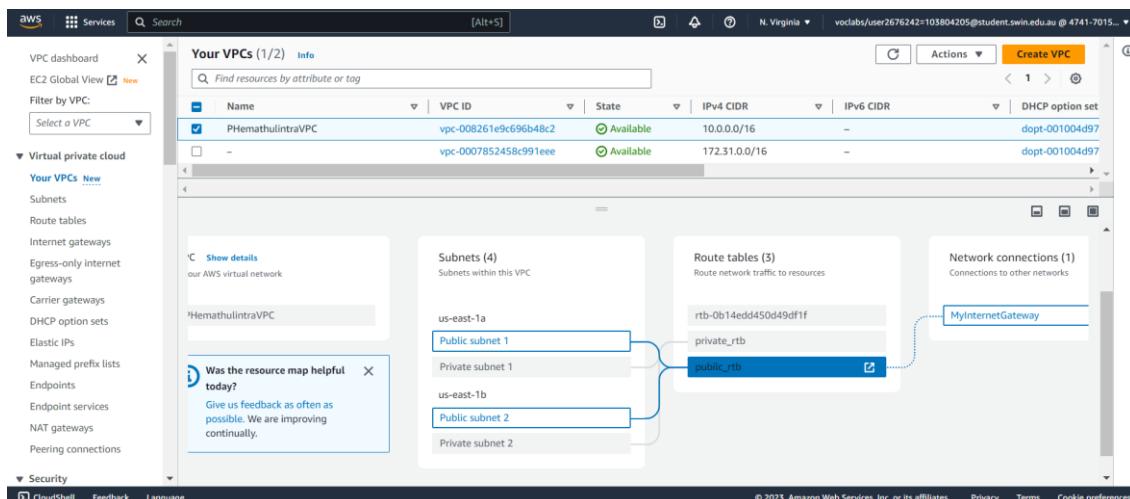


Figure 9 – view VPC resource map to ensure all subnet configurations

Step5: Create 3 security groups

1) TestInstanceSG

Security group name	Security group ID	Description	VPC ID
TestInstanceSG	sg-037fb1ad6f34a472f	allow all incoming traffic for testing	vpc-008261e9c696b48c2

Inbound rules (1/1)

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-08a6464f0ad12097	IPv4	All traffic	All	All

Figure 10 – create security group1

2) WebServerSG

Security group name	Security group ID	Description	VPC ID
WebServerSG	sg-0843f6ff9c2925695	allow http and ssh and permit only incoming ICMP from TestInstanceSG	vpc-008261e9c696b48c2

Inbound rules (3)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-01646beaa515560f	IPv4	HTTP	TCP	80	0.0.0.0/0	-
-	sgr-0eeec04327e749146c	-	All ICMP - I...	ICMP	All	sg-037fb1ad6f34a472f / TestInstanceSG	-
-	sgr-0ca3fc0cbf3a7ddcc	IPv4	SSH	TCP	22	0.0.0.0/0	-

Figure 11 – create security group2

3) DBServerSG

Security group name	Security group ID	Description	VPC ID
DBServerSG	sg-08309ddf096d09f21	permit MySQL access from WebServerSG	vpc-008261e9c696b48c2

Inbound rules (1/1)

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sg-08c946ed91e17eca	-	MySQL/Aurora	TCP	3306

Figure 12 – create security group3

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inb...
-	sg-08309ddf096d09f21	DBServerSG	vpc-008261e9c696b48c2	permit MySQL access f...	474170150111	1 P...
-	sg-0843f6ff9c2925695	WebServerSG	vpc-008261e9c696b48c2	allow http and ssh and...	474170150111	3 P...
-	sg-063c3112fd45e0142	default	vpc-0007852458c991ee	default VPC security gr...	474170150111	1 P...
-	sg-063c30d37e792231e	default	vpc-008261e9c696b48c2	default VPC security gr...	474170150111	1 P...
-	sg-037fb1ad6f34a472f	TestInstanceSG	vpc-008261e9c696b48c2	allow all incoming tra...	474170150111	1 P...

Inbound rules (3)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
-	sgr-051646beaa515560f	IPv4	HTTP	TCP	80	0.0.0.0/0
-	sgr-0eeec04327e749146c	-	All ICMP - IPv4	ICMP	All	sg-037fb1ad6f34a472f / TestInstanceSG
-	sgr-0ca3fc0cbf3a7ddcc	IPv4	SSH	TCP	22	0.0.0.0/0

Figure 13 – three security groups

Step 6: create resource for use when launching web server

1) a key pair (.pem file)

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name: MyKP_websvr

Key pair type: **RSA** (Info) ED25519

Private key file format: **.pem** (For use with OpenSSH)

.ppk (For use with PuTTY)

Tags - optional

Add new tag

You can add up to 50 more tags.

Cancel **Create key pair**

Figure 14 – create a key pair for web server

2) public Elastic IP

Allocated IPv4 address	Type	Allocation ID	Reverse DNS record
44.218.82.104	Public IP	eipalloc-0f7ac3acc48b5df3a	-
Association ID	Scope	Associated instance ID	Private IP address
-	VPC	-	-
Network interface ID	Network interface owner account ID	Public DNS	NAT Gateway ID
-	-	-	-
Address pool	Network Border Group		
Amazon	us-east-1		

Summary

Tags (0)

No tags associated with this resource

Figure 15 – create static IP

Step 7: launch the instances

1) Bastion/Web server

Name: Bastion/Web server

Application and OS Images (Amazon Machine Image) **Info**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux

Browse more AMIs

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-0f84a9675b22ea52 (64-bit (x86)) / ami-04249813e163e2cb8 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Number of instances **Info**
1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-0f84a9675b22ea52

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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)

Cancel **Launch instance** Review commands

Figure 14– launch web server: select image

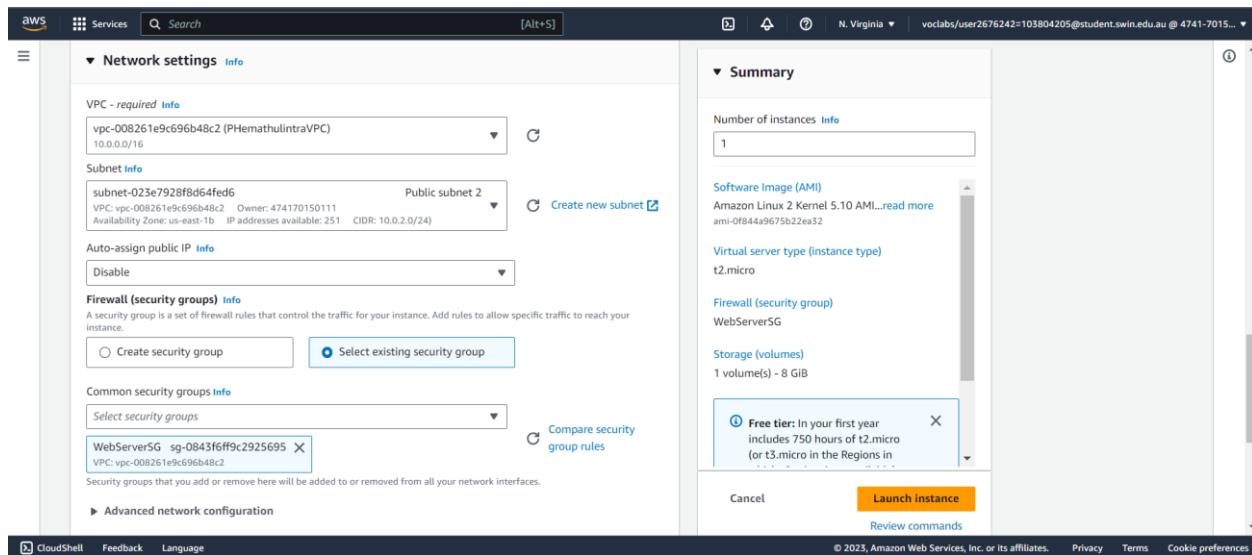


Figure 15 – launch web server: edit network settings

- attach Elastic IP to Bastion/Web instance

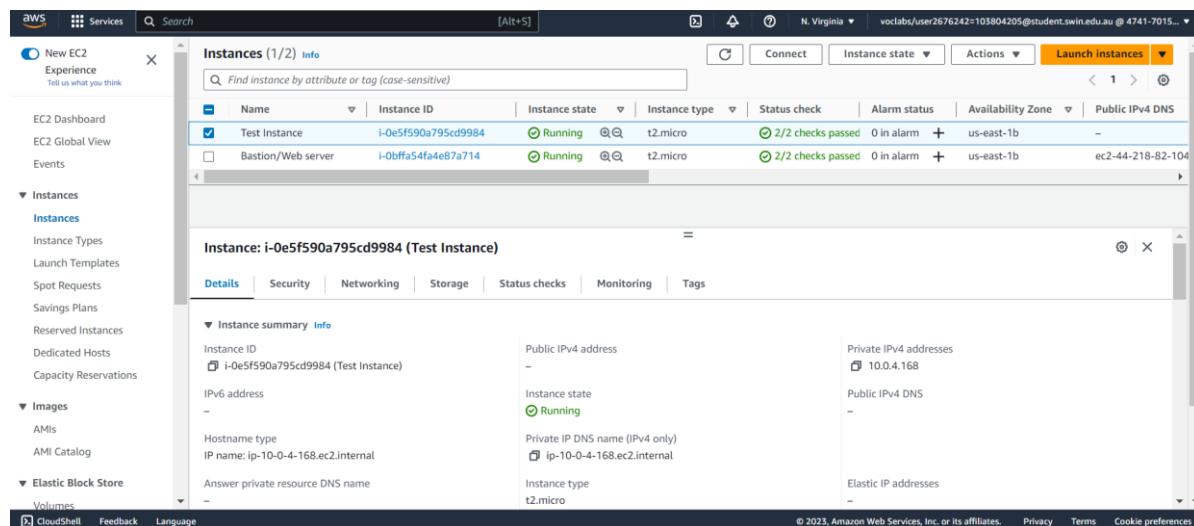


Figure 16 – attach static IP to web server

- check that PHP is installed

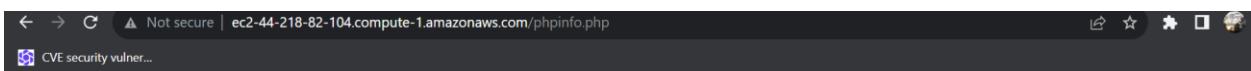


Figure 17 – access phpinfo to test php functionality

2) Test Instance

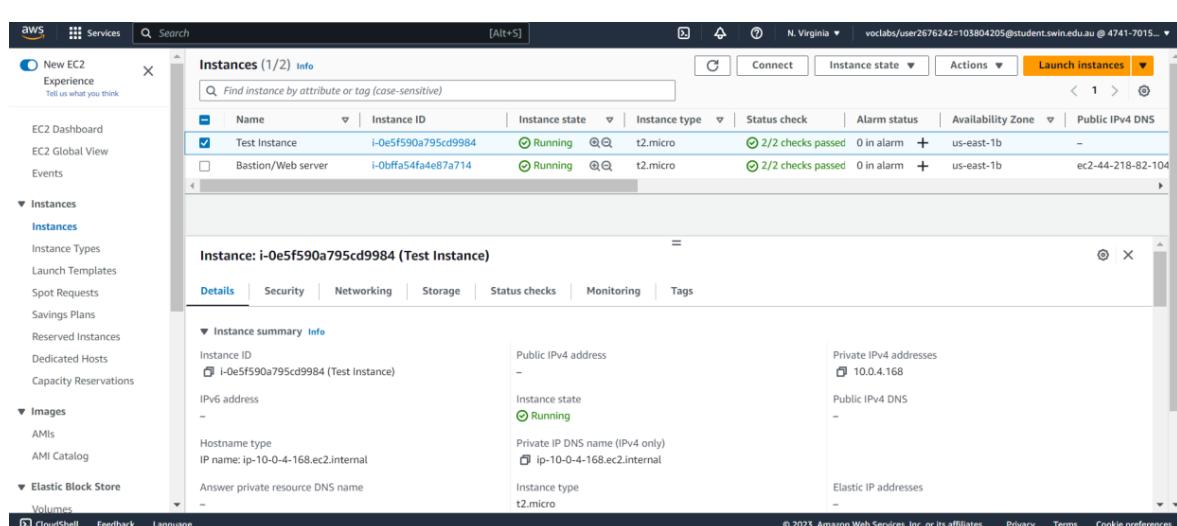


Figure 18 – test instance is in private subnet 2

Step 8: initiate SSH via variant of putty components

- 1) Open puttygen. Upload pem file. Create passphrase. Select SSH-1(RSA) and save private key.

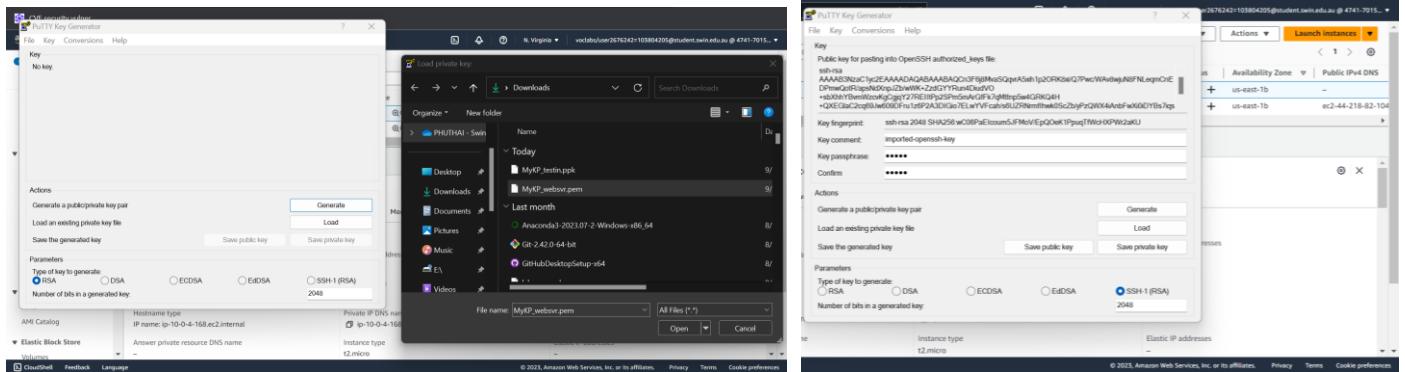


Figure 19-20 – puttygen key conversion

- 2) Open pageant and add key. Use the one that is converted to ppk. Enter passphrase and close pegeant.

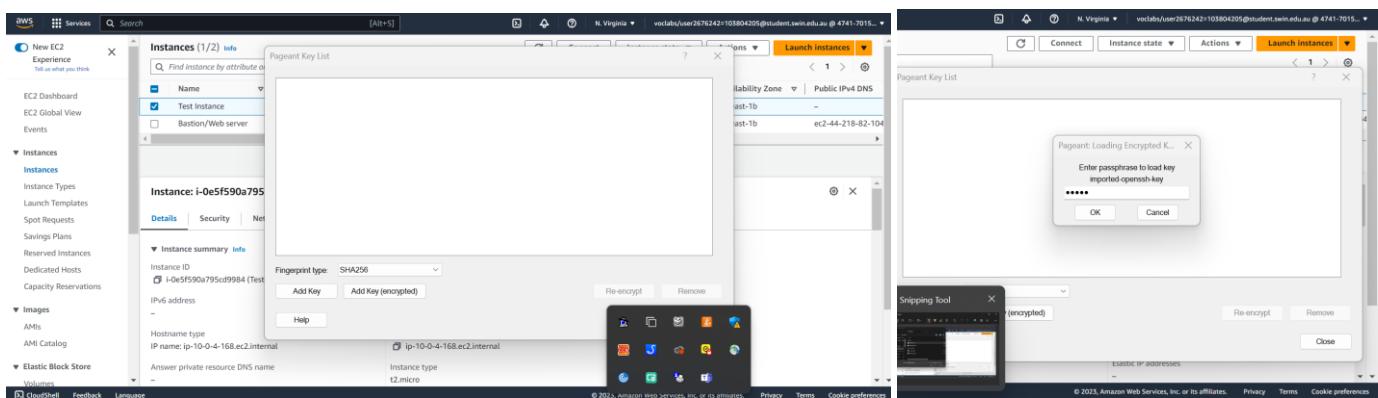


Figure 21-22 – pageant configurations

- 3) Open putty. In host field, enter public IPv4 of Bastion. Select Allow agent forwarding. Click open.

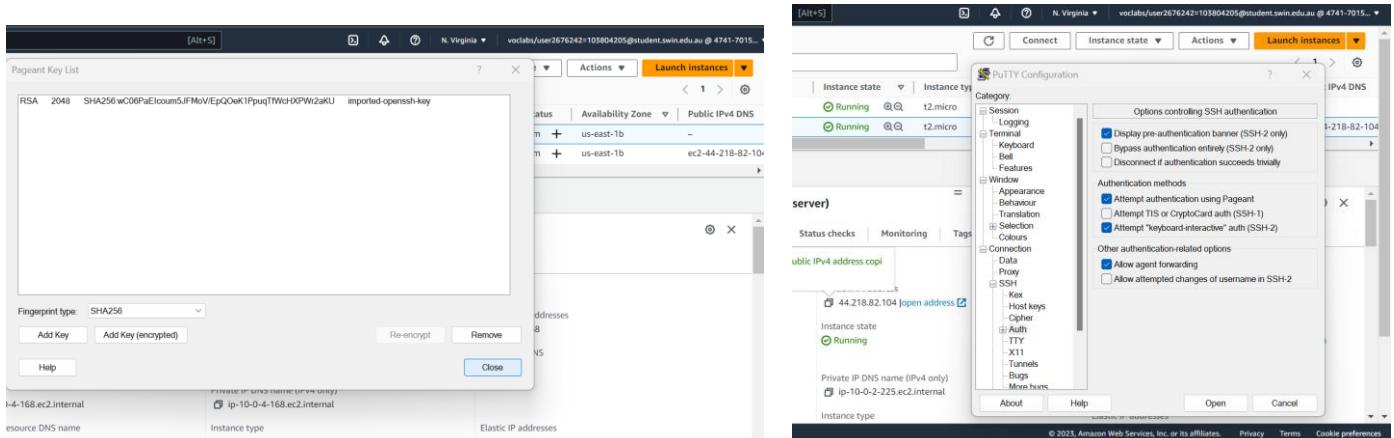


Figure 23-24 – putty configuration

- 4) Login as ec2-user and SSH to Test Instance using its private IP address

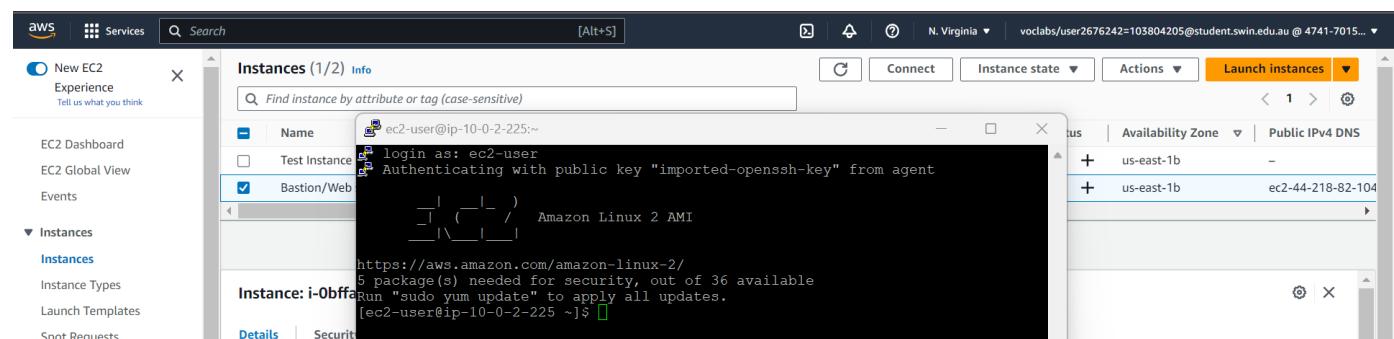


Figure 25 – connect to ec2 web server and login

5) Ping from Test instance to private IP address of Bastion.

```

Authenticating with public key "imported-openssh-key" from agent
|_ |_) / Amazon Linux 2 AMI
| \_|_
https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 36 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-2-225 ~]$ ssh ec2-user@10.0.4.168
-bash: ec2-user@10.0.4.168: command not found
[ec2-user@ip-10-0-2-225 ~]$ ssh ec2-user@10.0.4.168
The authenticity of host '10.0.4.168 (10.0.4.168)' can't be established.
ECDSA key fingerprint is SHA256:pwYOGCXDJWXMeIDbgwRJSJVHQSAmuSKBGjaWLN6Yrk.
ECDSA key fingerprint is MD5:fa:44:01:0b:53:b0:f2:d3:f5:c4:71:96:61:cf:38:d6.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.4.168' (ECDSA) to the list of known hosts.
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-0-2-225 ~]$ ping 10.0.2.225
PING 10.0.2.225 (10.0.2.225) 56(84) bytes of data.
64 bytes from 10.0.2.225: icmp_seq=1 ttl=255 time=0.018 ms
64 bytes from 10.0.2.225: icmp_seq=2 ttl=255 time=0.033 ms
64 bytes from 10.0.2.225: icmp_seq=3 ttl=255 time=0.038 ms
64 bytes from 10.0.2.225: icmp_seq=4 ttl=255 time=0.033 ms
64 bytes from 10.0.2.225: icmp_seq=5 ttl=255 time=0.035 ms
64 bytes from 10.0.2.225: icmp_seq=6 ttl=255 time=0.033 ms
^C
--- 10.0.2.225 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5128ms
rtt min/avg/max/mdev = 0.018/0.031/0.038/0.009 ms
[ec2-user@ip-10-0-2-225 ~]$

```

Figure 26 – shh to ec2 test instance

Step 9: configure RDS

- In RDS, create a subnet group

Create DB subnet group

DB Subnet group details

Name: db-prm-subnet-group

Description: private subnet group for DB

VPC: **Premastered VPC** (vpc-008201c69048462)

Add subnets

Add subnets

Availability Zones: us-east-1a, us-east-1b

Subnets:

- subnet-023608349833a75 (10.0.4.0/24)
- subnet-0258d6e46c3001 (10.0.5.0/24)

For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Subnets selected (2)

Availability zone	Subnet ID	CIDR block
us-east-1b	subnet-023608349833a75	10.0.4.0/24
us-east-1a	subnet-0258d6e46c3001	10.0.5.0/24

Figure 27 – create subnet group for DB, Figure 28 – subnet group for DB: select both zones and subnets

Create database

Choose a database creation method

Standard create

Easy create

Engine type: MySQL

Aurora (MySQL Compatible)

Aurora (PostgreSQL Compatible)

MySQL

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute and storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas across regions.

-Username = admin, Password = Pa55w.rd

- Create RDS

Connectivity via

Compute resource

Choose whether you can connect to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource

Connect to an EC2 compute resource

This database can now connect to an EC2 compute instance for this database. You can manually set up a connection to this database.

EC2 instance: ec2-10-0-2-214

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

Network type: IP

You can use standard mode, make sure that you associate an IPv4 CIDR block with a subnet in the VPC you specify.

Public access

No

RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

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 (MySQL) info

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow appropriate incoming traffic.

Choose existing

Choose existing VPC security groups

Create new

Create new VPC security group

Additional VPC security group

Choose one or more options

DBServerSG

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute and storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas across regions.

Figure 28 – create RDS: connect to web server, Figure 29 – create RDS: use the RDS security group

Figure 30 – create RDS: mark out automation and encryption, Figure 31 – find RDS endpoint

- My endpoint = my-rds-db-instance.c8moalxq8aty.us-east-1.rds.amazonaws.com

Step 10: Install phpMyAdmin on EC2

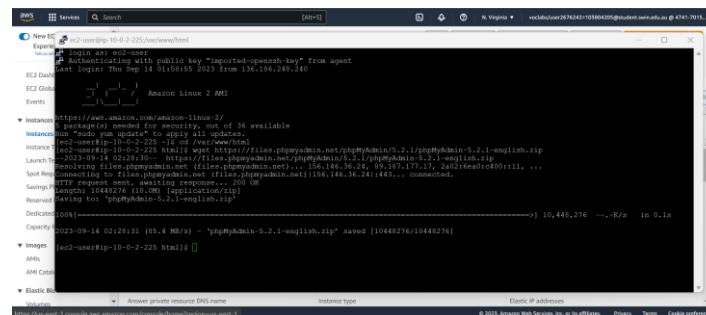


Figure 32 – Install phpMyAdmin: download

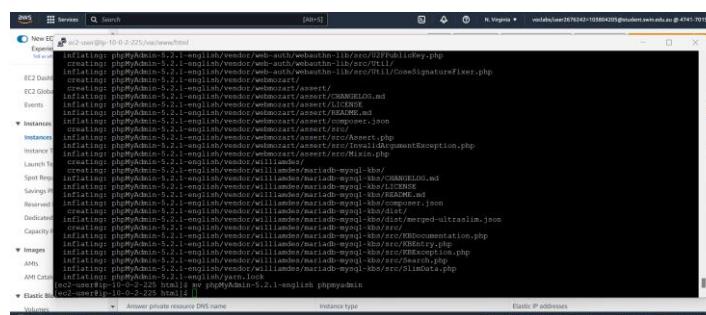


Figure 33 – Install phpMyAdmin: extract and change file name

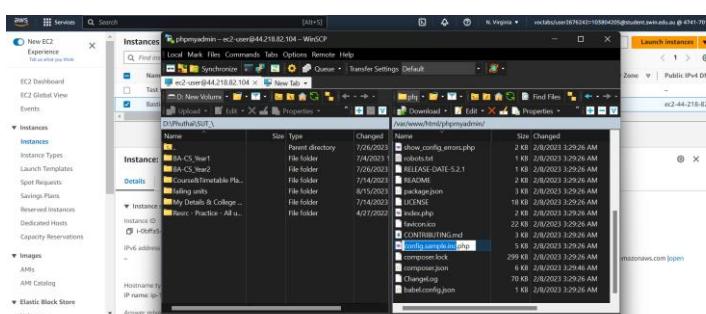


Figure 34 – WINSCP: change file name

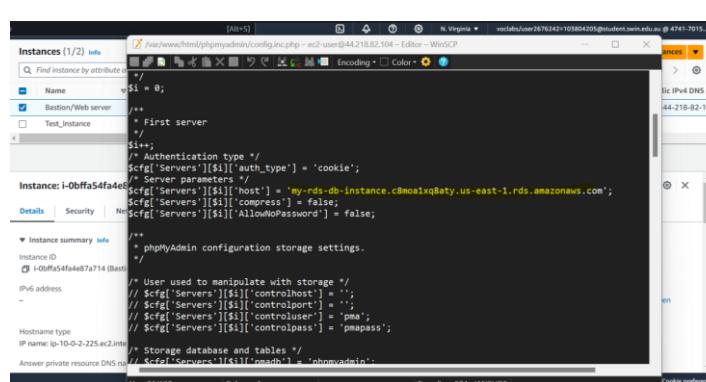


Figure 35 – WINSCP: insert RDS endpoint

Step 11: go to [http://\[publicDNSIPv4\]/phpmyadmin](http://[publicDNSIPv4]/phpmyadmin)

- Create a table ‘photos’. Add 5 columns to the table.

The screenshot shows the phpMyAdmin interface for a database named 'myPhotoAI_db'. A new table named 'photos' is being created with the following structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	PhotoTitle	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More
2	Description	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More
3	CreationDate	date			No	None			Change Drop More
4	Keywords	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More
5	Reference	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More

Below the table structure, there is a section for 'Indexes' with a note: 'No index defined!'.

Figure 36 – create a table , 5 columns with input types

Step 12: create NACL

The screenshot shows the AWS VPC dashboard. A new Network ACL named 'PublicSubnet2NACL' has been successfully created. The details page for this NACL shows it is associated with a single subnet and has no inbound or outbound rules defined.

Figure 37 – create Network ACL

- Inbound rules and outbound rules should be the same and attach it to public subnet 2

The screenshot shows the 'Edit inbound rules' section for the 'PublicSubnet2NACL'. It lists five rules allowing SSH, HTTP, HTTPS, ICMP, and All TCP traffic from 10.0.4.0/24 to all ports. The NACL is associated with a single subnet.

Figure 38 – NACL: edit inbound rules

The screenshot shows the 'Edit outbound rules' section for the 'PublicSubnet2NACL'. It lists one rule allowing all traffic from all ports to all ports. The NACL is associated with a single subnet.

Figure 39 – edit outbound rules

Step 13: create S3 bucket

The left screenshot shows the 'Create bucket' page with a 'Bucket name' field containing 'my-s3-bucket-for-photos'. The right screenshot shows the 'Edit Block public access (bucket settings)' page with several policy options listed under 'Block public access'.

Figure 40 – create S3 bucket, Figure 41 – S3: edit public access

- Specify public access and bucket policy for bucket policy

The screenshot shows the 'Permissions' tab selected in the navigation bar. Below it, the 'Bucket policy' section displays a JSON policy document:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Public-Allow-ReadPermission",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "S3.GetObject",
      "Resource": "arn:aws:s3:::my-s3-bucket-for-photos/*"
    }
  ]
}
```

Figure 42 – S3: edit new bucket policy

The screenshot shows the 'Upload successful' page with two files uploaded: 'swinlogo1.png' and 'swinlogo2.png'. Both files are marked as 'Successful'.

Figure 43 – upload photos successfully

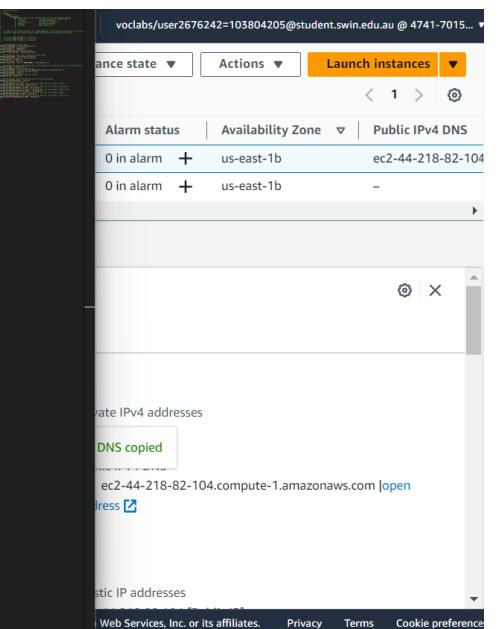
Step 14: In phpMyAdmin, insert new roles with photos' URLs in photos table

The screenshot shows the phpMyAdmin interface with the 'photos' table selected. The table has columns: PhotoTitle, Description, CreationDate, Keywords, and Reference. Two rows have been inserted:

PhotoTitle	Description	CreationDate	Keywords	Reference
swinlogo1	new Swinburne logo for online	2023-09-14	SL-N-001	https://my-s3-bucket-for-photos.s3.amazonaws.com/s...
swinlogo2	old Swinburne logo	2023-09-14	SL-O-001	https://my-s3-bucket-for-photos.s3.amazonaws.com/s...

Figure 44 – insert row with URL of each S3 photo

Step 15: In photoalbumv3, enter correct code based on resource in AWS



```

57 // [ACTION REQUIRED] your Student ID
58 define('STUDENT_ID', '103804205');
59 // [ACTION REQUIRED] your tutorial session
60 define('TUTORIAL_SESSION', 'Wednesday 06:30PM');
61
62 // [ACTION REQUIRED] name of the S3 bucket that stores images
63 define('BUCKET_NAME', 'my-s3-bucket-for-photos');
64 // [ACTION REQUIRED] region of the above bucket
65 define('REGION', 'us-east-1');
66 // no need to update this const
67 define('S3_BASE_URL', 'https://.BUCKET_NAME.'.s3.amazonaws.com/');
68
69 // [ACTION REQUIRED] name of the database that stores photo meta-data (note that this is not the DB identifier)
70 define('DB_NAME', 'myPhotoAL_db');
71 // [ACTION REQUIRED] endpoint of RDS instance
72 define('DB_ENDPOINT', 'my-rds-db-instance.c8moaxlqx8aty.us-east-1.rds.amazonaws.com');
73 // [ACTION REQUIRED] username of your RDS instance
74 define('DB_USERNAME', 'admin');
75 // [ACTION REQUIRED] password of your RDS instance
76 define('DB_PWD', 'Pa55w.rd');
77
78 // [ACTION REQUIRED] name of the DB table that stores photo's meta-data
79 define('DB_PHOTO_TABLE_NAME', 'photos');
80 // The table above has 5 columns:
81 // [ACTION REQUIRED] name of the column in the above table that stores photo's titles
82 define('DB_PHOTO_TITLE_COL_NAME', 'PhotoTitle');
83 // [ACTION REQUIRED] name of the column in the above table that stores photo's descriptions
84 define('DB_PHOTO_DESCRIPTION_COL_NAME', 'Description');
85 // [ACTION REQUIRED] name of the column in the above table that stores photo's creation dates
86 define('DB_PHOTO_CREATIONDATE_COL_NAME', 'CreationDate');
87 // [ACTION REQUIRED] name of the column in the above table that stores photo's keywords
88 define('DB_PHOTO_KEYWORDS_COL_NAME', 'Keywords');
89 // [ACTION REQUIRED] name of the column in the above table that stores photo's links in S3
90 define('DB_PHOTO_S3REFERENCE_COL_NAME', 'Reference');
91
92 //>

```

Figure 45 – edit constant.php with identities from previous configuration accordingly

Step 16: In WINSCP, create path of directory where photoalbum is hosting the website

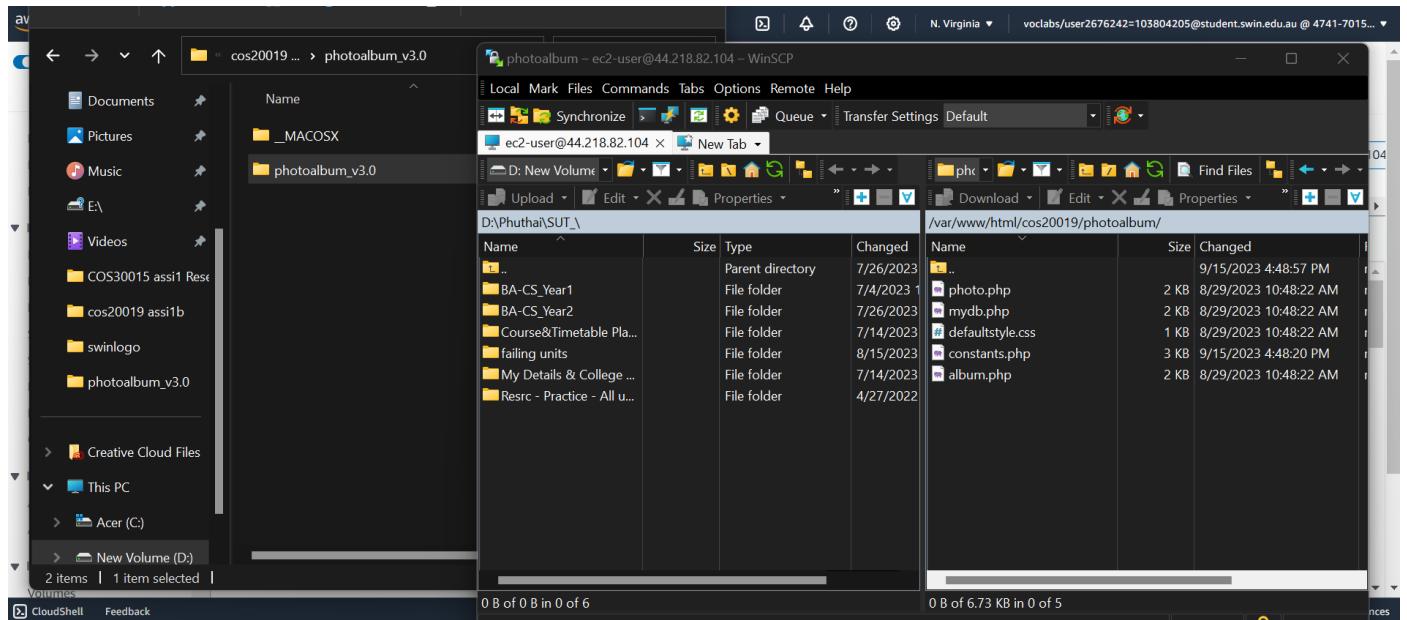


Figure 46 – WINSCP: uploading constant.php with other php files in photoalbum folder in the well-organized directory.

Step17: access to your photo album website which its database is in cloud

Website link: <http://ec2-44-218-82-104.compute-1.amazonaws.com/cos20019/photoalbum/album.php>

Not secure | ec2-44-218-82-104.compute-1.amazonaws.com/cos20019/photoalbum/album.php

CVE security vulner...

Student name: Phuthai Hemathulintra

Student ID: 103804205

Tutorial session: Wednesday 06:30PM

Uploaded photos:

Photo	Name	Description	Creation date	Keywords
	swinlogo1	new Swinburne logo for online	2023-09-14	SL-N-001
	swinlogo2	old Swinburne logo	2023-09-14	SL-O-001

Figure 47 – two photos were uploaded with description

Step 18: Testing

- add more photos

Student name: Phuthai Hemathulintra

Student ID: 103804205

Tutorial session: Wednesday 06:30PM

Uploaded photos:

Photo	Name	Description	Creation date	Keywords
 SWINBURNE UNIVERSITY OF TECHNOLOGY	swinlogo1	new Swinburne logo for online	2023-09-14	SL-N-001
	swinlogo2	old Swinburne logo	2023-09-14	SL-O-001
	swinlogo3	razor	0000-00-00	SL-N-002
	swinlogo4	old logo with red background	0000-00-00	SL-O-002
	swinlogo5	Swinburne student Union logo	2023-09-15	SL-N-003
	swinlogo6	Swinburne online logo	2023-09-15	SL-N-004
	swinlogo7	Swinburne simple logo	2023-09-15	SL-N-005

Figure 48 – more photos were uploaded with description successfully

- ping to web server again to test NACL

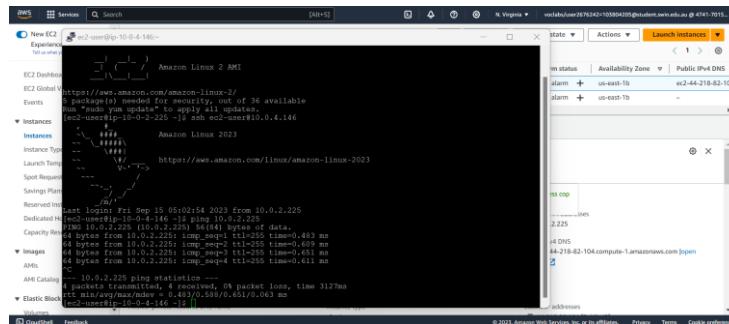


Figure 48 – more photos were uploaded with description successfully