

## Step 1: Access VPC control panel

The screenshot shows two windows side-by-side. On the left is the AWS VPC Console for the 'us-east-1' region. The main area displays 'Resources by Region' with various service counts: VPCs (1), Subnets (6), Route Tables (1), Internet Gateways (1), Security Groups (2), and others. A prominent orange button at the top center says 'Create VPC'. To the right of the VPC resources is the 'AWS Network Manager' section, which provides an overview of network management tools. On the far right of the VPC console is a note about creating a custom VPC and a detailed requirements section. On the right side of the image is a separate window titled 'Assignment1b\_UG\_v5.00.pdf' showing a network architecture diagram. The diagram illustrates a VPC with two private subnets: 'Private subnet 1 (10.0.3.0/24)' and 'Private subnet 2 (10.0.4.0/24)'. Each subnet contains an 'RDS instance' and a 'Test instance' respectively. Each instance is associated with a specific security group: 'DB-tier Security group' for RDS and 'Test-instance Security group' for the test instances. A legend at the bottom defines symbols for VPC, Subnet, Route Table, Internet Gateway, Security Group, and Endpoints.

## Step 2: Create a VPC

# Nguyen Gia Binh- 104219428 / SWH01067

VPC Console + [Alt+S]

us-east-1.console.amazonaws.com/vpcconsole/home?region=us-east-1#CreateVpc:createMode=vpcWithResources

N. Virginia vocabs/user2753113-104219428@student.swin.edu.au @ 3665-0929...

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

**VPC settings**

Resources to create Info  
Create only the VPC resource or the VPC and other networking resources.  
 VPC only  VPC and more

Name tag auto-generation Info  
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.  
 Auto-generate

IPv4 CIDR block Info  
Determine the starting IP and the size of your VPC using CIDR notation.  
10.0.0.0/16 65,536 IPs  
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info  
 No IPv6 CIDR block  
 Amazon-provided IPv6 CIDR block

Tenancy Info  
Default

Number of Availability Zones (AZs) Info  
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.  
1 2 3

**Preview**

VPC Show details  
Your AWS virtual network BNguyenVPC

Subnets (4)  
Subnets within this VPC

us-east-1a  
public1-us-east-1a  
private1-us-east-1a

us-east-1b  
public2-us-east-1b  
private2-us-east-1b

Route tables (3)  
Route network traffic to resources

rtb-public  
rtb-private1-us-east-1a  
rtb-private2-us-east-1b

Network connections (1)  
Connections to other network

Internet-Gateway-asrn1  
S3-bucket-asrn1b

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 8:59 AM 10/14/2023

VPC Console + [Alt+S]

us-east-1.console.amazonaws.com/vpcconsole/home?region=us-east-1#CreateVpc:createMode=vpcWithResources

N. Virginia vocabs/user2753113-104219428@student.swin.edu.au @ 3665-0929...

First availability zone us-east-1a

Second availability zone us-east-1b

Number of public subnets Info  
The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.  
0 2

Number of private subnets Info  
The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.  
0 2 4

Customize subnets CIDR blocks

Public subnet CIDR block in us-east-1a  
10.0.1.0/24 256 IPs

Public subnet CIDR block in us-east-1b  
10.0.2.0/24 256 IPs

Private subnet CIDR block in us-east-1a  
10.0.3.0/24 256 IPs

Private subnet CIDR block in us-east-1b  
10.0.4.0/24 256 IPs

NAT gateways (\$) Info  
Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway.

**Preview**

VPC Show details  
Your AWS virtual network BNguyenVPC

Subnets (4)  
Subnets within this VPC

us-east-1a  
public1-us-east-1a  
private1-us-east-1a

us-east-1b  
public2-us-east-1b  
private2-us-east-1b

Route tables (3)  
Route network traffic to resources

rtb-public  
rtb-private1-us-east-1a  
rtb-private2-us-east-1b

Network connections (1)  
Connections to other network

Internet-Gateway-asrn1  
S3-bucket-asrn1b

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 9:00 AM 10/14/2023

# Nguyen Gia Binh- 104219428 / SWH01067

VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateVpc:createMode=vpcWithResources

Services Search [Alt+S]

N. Virginia vocabs/user2753113-104219428@student.swin.edu.au @ 3665-0929...

Preview

VPC Show details Your AWS virtual network BNguyenVPC Subnets (4) Subnets within this VPC us-east-1a public1-us-east-1a private1-us-east-1a us-east-1b public2-us-east-1b private2-us-east-1b Route tables (3) Route network traffic to resources rtb-public rtb-private1-us-east-1a rtb-private2-us-east-1b Network connections (2) Connections to other networks Internet-Gateway-as1b S3-bucket-as1b

Public subnet CIDR block in us-east-1a 10.0.1.0/24 256 IPs

Public subnet CIDR block in us-east-1b 10.0.2.0/24 256 IPs

Private subnet CIDR block in us-east-1a 10.0.3.0/24 256 IPs

Private subnet CIDR block in us-east-1b 10.0.4.0/24 256 IPs

NAT gateways (\$ 0) Info Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway.

None In 1 AZ 1 per AZ

VPC endpoints Info Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None S3 Gateway

DNS options Info  Enable DNS hostnames  Enable DNS resolution

Additional tags

Create VPC

Your VPCs | VPC Management

CloudShell Feedback

Search 26° 10/14/2023

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#vpcs:

VPC dashboard ECG Global View New

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

CloudShell Feedback

Search 26° 10/14/2023

Your VPCs (1/2) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table
vpc-0ac460d15db259b83	Available	172.31.0.0/16	-	dopt-0d5e1208699117...	rtb-0167444c4c22b6a40	
BNguyenVPC	Available	10.0.0.0/16	-	dopt-0d5e1208699117...	-	

vpc-0ac460d15db259b83

Details Resource map New CIDRs Flow logs Tags Integrations

Details

CloudShell Feedback

Search 26° 10/14/2023

# Nguyen Gia Binh- 104219428 / SWH01067

VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#VpcDetails:VpcId=vpc-0a77c0d690062c346

N. Virginia N. Virginia vclabs/user2753115-104219428@student.swin.edu.au @ 3665-0929...

VPC dashboard Services Search [Alt+S]

EC2 Global View □ New

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 CloudShell Feedback

VPC > Your VPCs > vpc-0a77c0d690062c346 / BNguyenVPC

Details Info

VPC ID vpc-0a77c0d690062c346 State Available

Tenancy Default DHCP option set dopt-05fe1208699117669

Default VPC No IPv4 CIDR 10.0.0.0/16

Network Address Usage metrics Disabled Route 53 Resolver DNS Firewall rule groups Failed to load rule groups

DNS hostnames Enabled Main route table rtb-059ea74fcf49c85b5

IPv6 pool - Owner ID 366309293917

DNS resolution Enabled Main network ACL acl-01736c6b6a3d98d20

IPv6 CIDR (Network border group) -

Resource map New CIDRs Flow logs Tags Integrations

Resource map Info

VPC Show details Your AWS virtual network BNguyenVPC

Was the resource map helpful today? Give us feedback as often as possible. We are improving continually.

Subnets (4) Subnets within this VPC

- us-east-1a
- public1-us-east-1a
- private1-us-east-1a
- us-east-1b
- public2-us-east-1b
- private2-us-east-1b

Route tables (4) Route network traffic to resources

- rtb-private2-us-east-1b
- rtb-059ea74fcf49c85b5
- rtb-public
- rtb-private1-us-east-1a

Network connections (2) Connections to other networks

- Internet-Gateway-as1b
- S3-bucket-as1b

CloudShell Feedback

Search 26°

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 9:02 AM 10/14/2023

VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateVpcWizard:

N. Virginia vclabs/user2753115-104219428@student.swin.edu.au @ 3665-0929...

VPC > Your VPCs > Create VPC > Create VPC resources

Create VPC workflow

Success

Details

- Create VPC: vpc-0a77c0d690062c346
- Enable DNS hostnames
- Enable DNS resolution
- Verifying VPC creation: vpc-0a77c0d690062c346
- Create S3 endpoint: vpc-052c8bd2da486abd6
- Create subnet: subnet-0d74a915a109f4e8e
- Create subnet: subnet-0eef5dfac8f97c0c
- Create subnet: subnet-044c02d4bd4d4503c
- Create subnet: subnet-039d12a78c26dea2f9
- Create internet gateway: igw-0e2eedcd4ab2xthfgf
- Attach internet gateway to the VPC
- Create route table: rtb-0b0934941e4dfebfbb
- Create route
- Associate route table
- Associate route table
- Create route table: rtb-0bd3de39379b8c796
- Associate route table
- Create route table: rtb-03498489ec2b7f976
- Associate route table
- Verifying route table creation
- Associate S3 endpoint with private subnet route tables: vpc-052c8bd2da486abd6

View VPC

CloudShell Feedback

Search 26°

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 9:01 AM 10/14/2023

Step 3: Create security group. Access through the left navigation panel

### 3.1: TestInstanceSG

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

**Basic details**

Security group name [Info](#)  
TestinstanceSG  
Name cannot be edited after creation.

Description [Info](#)  
Test instance security group

VPC Info  
vpc-0a77c0d690062c346

**Inbound rules** [Info](#)

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
All traffic	All	All	Anywhere... <input type="text" value="0.0.0.0/0"/>	<input type="button" value="Delete"/>

Add rule

⚠️ Rules with source of 0.0.0.0/0 or ~ /0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

### 3.2: WebServerSG

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

**Basic details**

Security group name [Info](#)  
WebServerSG  
Name cannot be edited after creation.

Description [Info](#)  
web server security group

VPC Info  
vpc-0a77c0d690062c346

**Inbound rules** [Info](#)

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
HTTP	TCP	80	Anywhere... <input type="text" value="0.0.0.0/0"/>	<input type="button" value="Delete"/>
SSH	TCP	22	Anywhere... <input type="text" value="0.0.0.0/0"/>	<input type="button" value="Delete"/>
All ICMP - IPv4	ICMP	All	Custom <input type="text" value="sg-0315456b79134a99e"/>	<input type="button" value="Delete"/>

Add rule

⚠️ Rules with source of 0.0.0.0/0 or ~ /0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

### 3.3:DBServerSG

**Create security group Info**

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

**Basic details**

Security group name **Info**  
DBServerSG  
Name cannot be edited after creation.

Description **Info**  
database server security group

VPC Info  
vpc-0a77c0d690062c346

**Inbound rules Info**

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
MySQL/Aurora	TCP	3306	Custom	sg-0bc0ddae83173537

Add rule

**Outbound rules**

CloudShell Feedback 9:41 AM 10/14/2023

**VPC Console** us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSecurityGroup:

**Security Groups (6) Info**

Name	Security group ID	Security group name	VPC ID
sg-00c45605c471a0dab	default	vpc-0a77c0d690062c3	
sg-09697125f09a028b4	DBServerSG	vpc-0ac460d15db259t	
sg-007f1a0544c0c00b	TestInstanceSG	vpc-0ac460d15db259t	
sg-09e8294d75d8a370	default	vpc-0ac460d15db259t	
sg-08909aa6b5f861620	WebServer-SG	vpc-0ac460d15db259t	
sg-039805562b0ff9db9	WebServerSG	vpc-0ac460d15db259t	

Assignment1b\_UG\_v5.00.pdf C:/Users/DELL/Do... 3 / 7 9:41 AM 10/14/2023

**1.2 - Security groups**

Create the following security groups, each is associated with each tier shown in the architecture diagram:

CDS20019

School of Science, Computing and Engineering Technologies Swinburne University of Technology

Security group name	Protocols	Source
TestInstanceSG	All traffic	Anywhere
WebServerSG	HTTP (80), SSH (22)	Anywhere
	ICMP	TestInstanceSG
DBServerSG	MySQL (3306)	WebServerSG

**1.3 – EC2 virtual machine**

You will create two EC2 instances, a test instance and a bastion/web server instance.

**1.3.1 – Bastion/Web server instance**

Your web server must be deployed on an EC2 instance in Public Subnet 2. This EC2 instance should be configured similar to the EC2 created in Assignment 1A:

- Amazon Machine Image: *Amazon Linux 2 AMI (HVM), SSD Volume Type*
- Instance type: *t2.micro*
- Has Apache web server and other PHP packages installed (you can use the same bash script provided in Assignment 1A to bootstrap your EC2).

This instance will host the "Photo Album" web application, which was created in Assignment 1A –

## Step 4: Create 2 EC2 instance

### 4.1: Bastion/Web server EC2

**Launch an instance | EC2 | us-east-1 | VPC Console**

**Instance type**: t2.micro (Free tier eligible)

**Key pair (login)**: assignment1b

**Network settings**: VPC: vpc-0a77c0d690062c346 (BNGuyenVPC), Subnet: subnet-0ee55dfacaf897c0c, Auto-assign public IP: Disable

**Summary**: Number of instances: 1, Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI, Virtual server type: t2.micro, Firewall: WebServerSG, Storage: 1 volume(s) - 8 GiB.

**Launch instance**

**Name and tags**: Name: Bastion/Web server

**Application and OS Images (Amazon Machine Image)**: Search bar: Search our full catalog including 1000s of application and OS images, Quick Start: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE, Browse more AMIs: Including AMIs from AWS, Marketplace and the Community.

**Amazon Machine Image (AMI)**: Description: Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type, Architecture: 64-bit (x86), AMI ID: ami-0bb4c991fa89d4b9b, Verified provider.

**Summary**: Number of instances: 1, Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI, Virtual server type: t2.micro, Firewall: WebServerSG, Storage: 1 volume(s) - 8 GiB.

**Launch instance**

The screenshot shows the AWS CloudShell interface. A modal window titled 'Create key pair' is open, prompting for a key pair name ('assignment1b'), key pair type ('RSA'), and private key file format ('ppk'). A note at the bottom states: 'When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance.' A 'Create key pair' button is visible.

### 1.3 – EC2 virtual machine

You will create two EC2 instances, a test instance and a bastion/web server instance.

#### 1.3.1 – Bastion/Web server instance

Your web server must be deployed on an EC2 instance in Public Subnet 2. This EC2 instance should be configured similar to the EC2 created in Assignment 1A:

- Amazon Machine Image: **Amazon Linux 2 AMI (HVM), SSD Volume Type**
- Instance type: **t2.micro**
- Has Apache web server and other PHP packages installed (you can use the same bash script provided in Assignment 1A to bootstrap your EC2).

This instance will host the "Photo Album" web application, which was created in Assignment 1A – more details are in Section 2 of this specification document. This instance will also act as a bastion host for you to SSH into the Test instance, which resides in a private subnet.

**NOTE:** [your public dns] will change every time the webserver instance restarts. To avoid this behaviour and to ensure your Webserver URL remains persistent, add an Elastic IP Address to this instance by allocating an Elastic IP address in the same region under the Network and Security section in the left menu of the EC2 service page, then associate this new EIP to your Bastion/Web server instance. The public IP address of your instance should now automatically match this Elastic IP address.

#### 1.3.2 – Test instance

This instance will be used for demonstration purposes only. It does not contribute to the functionality of Photo Album website. You will SSH into this instance and ping the web server (using "ping" command in Linux). Please take a screenshot(s) of the Linux terminal to demonstrate that:

- You are able to SSH into an instance in a private subnet (which is this Test instance). For instructions on how to connect to a private EC2 instance through a bastion host, you can refer to <https://aws.amazon.com/blogs/security/securing-connect-to-linux-instances-running-in-a-private-amazon-vpc/>
- You are able to establish a connection (ICMP ping) between this instance and the Bastion/Web server instance.

The screenshot shows the AWS VPC Console. A modal window for launching an instance is open. It specifies 1 instance, the 't2.micro' instance type, and a 1x 8 GiB gp2 root volume (Not encrypted). The 'Launch instance' button is visible. A note about the Free tier is displayed: '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 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.'

## Reuse the script from asm1a to setup

The screenshot shows the AWS VPC Console interface for launching a new EC2 instance. The user data field contains the following shell script:

```

#!/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
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 2770 {} \;
find /var/www -type f -exec sudo chmod 0640 {} \;
echo "<?php echo '<h2>Welcome to COS80001. Installed PHP version:' . phpversion() . '</h2>'; ?>" > /var/www/html/phpinfo.php

```

A tooltip for the free tier is displayed, stating: "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 GB of snapshots, and 100 GB of bandwidth to the internet."

**Instances | EC2 | us-east-1**

The screenshot shows the AWS EC2 Instances page. It displays one instance named "Bastion/Web s..." with the ID i-0264c88efc3959f0f. The instance is currently running.

Open left navigation panel and choose Elastic IP Address and choose Allocate Elastic IP Address

The screenshot shows the AWS VPC Console with the 'Allocate Elastic IP address' page open. The page has a search bar for 'Network Border Group' and a dropdown for 'Region' set to 'us-east-1'. Under 'Public IPv4 address pool', it shows 'Amazon's pool of IPv4 addresses'. There are sections for 'Public IPv4 address' and 'Customer-owned pool of IPv4 addresses'. Under 'Global static IP addresses', it mentions AWS Global Accelerator. A 'Create accelerator' button is present. The 'Tags - optional' section allows adding tags to the resource. At the bottom, there is a 'Cancel' button and a large orange 'Allocate' button.

The screenshot shows the AWS EC2 console with the 'Elastic IP address allocated successfully' page open. The page displays a table of 'Elastic IP addresses' with one entry: 'Name' (empty), 'Allocated IPv4 add...' (52.202.86.40), 'Type' (Public IP), and 'Allocated' (eipalloc-00000000). Below the table, there is a link to 'View IP address usage and recommendations to release unused IPs with Public IP insights'. The left sidebar shows the EC2 dashboard and various instance-related options like Instances, Launch Templates, and Network & Security. At the bottom, there is a 'CloudShell' feedback bar.

Select the created elastic IP and choose Action -> Allocate Elastic IP Address and add the Bastion/... EC2 instance to it

**Associate Elastic IP address**

Choose the instance or network interface to associate to this Elastic IP address (52.202.86.40)

**Elastic IP address:** 52.202.86.40

**Resource type**  
Choose the type of resource with which to associate the Elastic IP address.

- Instance
- Network interface

**Instance**  
Q i-0264c88efc3959f0f

**Private IP address**  
The private IP address with which to associate the Elastic IP address.  
Q Choose a private IP address

**Reassociation**  
Specify whether the Elastic IP address can be reassigned with a different resource if it already associated with a resource.  
 Allow this Elastic IP address to be reassociated

**Associate**

CloudShell Feedback Privacy Terms Cookie preferences © 2023, Amazon Web Services, Inc. or its affiliates.

Assignment1b\_UG\_v5.00.pdf 3 / 7 9:52 AM 10/14/2023

**1.3 – EC2 virtual machine**

You will create two EC2 instances, a test instance and a bastion/web server instance.

**1.3.1 – Bastion/Web server instance**

Your web server must be deployed on an EC2 instance in Public Subnet 2. This EC2 instance should be configured similar to the EC2 created in Assignment 1A:

- Amazon Machine Image: Amazon Linux 2 AMI (HVM), SSD Volume Type
- Instance type: t2.micro
- Has Apache web server and other PHP packages installed (you can use the same bash script provided in Assignment 1A to bootstrap your EC2).

This instance will host the "Photo Album" web application, which was created in Assignment 1A – more details are in Section 2 of this specification document. This instance will also act as a bastion host for you to SSH into the Test instance, which resides in a private subnet.

**NOTE:** [your.public.dns] will change every time the webserver instance restarts. To avoid this behaviour and to ensure your Websiter URL remains persistent, add an Elastic IP Address to this instance by allocating an Elastic IP address in the same region under the Network and Security section in the left menu of the EC2 service page, then associate this new EIP to your Bastion/Web server instance. The public IP address of your instance should now automatically match this Elastic IP address.

**1.3.2 – Test instance**

This instance will be used for demonstration purposes only. It does not contribute to the functionality of Photo Album website. You will SSH into this instance and ping the web server (using "ping" command in Linux). Please take a screenshot(s) of the Linux terminal to demonstrate that:

- You are able to SSH into an instance in a private subnet (which is this Test instance). For instructions on how to connect to a private EC2 instance through a bastion host, you can refer to <https://aws.amazon.com/blogs/security/securly-connect-to-linux-instances-running-in-a-private-amazon-vpc/>
- You are able to establish a connection (ICMP ping) between this instance and the Bastion/Web server instance.

The configuration of this instance is entirely your choice. This instance does not host the web application.

**Elastic IP address associated successfully.**

Elastic IP address 52.202.86.40 has been associated with instance i-0264c88efc3959f0f

**Elastic IP addresses (1/1)**

Name	Allocated IPv4 add...	Type	Allocat...
-	52.202.86.40	Public IP	eipalloc...

**Allocate Elastic IP address**

CloudShell Feedback Privacy Terms Cookie preferences us-east-1.console.aws.amazon.com/.../home?region=us-east-1 © 2023, Amazon Web Services, Inc. or its affiliates.

Assignment1b\_UG\_v5.00.pdf 3 / 7 9:52 AM 10/14/2023

**1.3 – EC2 virtual machine**

You will create two EC2 instances, a test instance and a bastion/web server instance.

**1.3.1 – Bastion/Web server instance**

Your web server must be deployed on an EC2 instance in Public Subnet 2. This EC2 instance should be configured similar to the EC2 created in Assignment 1A:

- Amazon Machine Image: Amazon Linux 2 AMI (HVM), SSD Volume Type
- Instance type: t2.micro
- Has Apache web server and other PHP packages installed (you can use the same bash script provided in Assignment 1A to bootstrap your EC2).

This instance will host the "Photo Album" web application, which was created in Assignment 1A – more details are in Section 2 of this specification document. This instance will also act as a bastion host for you to SSH into the Test instance, which resides in a private subnet.

**NOTE:** [your.public.dns] will change every time the webserver instance restarts. To avoid this behaviour and to ensure your Websiter URL remains persistent, add an Elastic IP Address to this instance by allocating an Elastic IP address in the same region under the Network and Security section in the left menu of the EC2 service page, then associate this new EIP to your Bastion/Web server instance. The public IP address of your instance should now automatically match this Elastic IP address.

**1.3.2 – Test instance**

This instance will be used for demonstration purposes only. It does not contribute to the functionality of Photo Album website. You will SSH into this instance and ping the web server (using "ping" command in Linux). Please take a screenshot(s) of the Linux terminal to demonstrate that:

- You are able to SSH into an instance in a private subnet (which is this Test instance). For instructions on how to connect to a private EC2 instance through a bastion host, you can refer to <https://aws.amazon.com/blogs/security/securly-connect-to-linux-instances-running-in-a-private-amazon-vpc/>
- You are able to establish a connection (ICMP ping) between this instance and the Bastion/Web server instance.

The configuration of this instance is entirely your choice. This instance does not host the web application.

4.2: Test instance. Since it is not specified how this is config I left it the same as Bastion but different security group and no script

**Name and tags**

Name: Test Instance

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

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

Recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux

Browse more AMIs: Including AMIs from AWS, Marketplace and the Community

**Description**

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type  
ami-0dbb4c991f89d945b9 (64-bit (x86)) / ami-04d45ec5e53104891 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

**Security group**

Security group name	Protocols	Source
TestInstanceSG	All traffic	Anywhere
WebServerSG	HTTP (80), SSH (22)	Anywhere
DBServerSG	ICMP	TestInstanceSG
DBServerSG	MySQL (3306)	WebServerSG

**1.3 – EC2 virtual machine**

You will create two EC2 instances, a test instance and a bastion/web server instance.

**1.3.1 – Bastion/Web server instance**

Your web server must be deployed on an EC2 instance in Public Subnet 2. This EC2 instance should be configured similar to the EC2 created in Assignment 1A:

- Amazon Machine Image: Amazon Linux 2 AMI (HVM), SSD Volume Type
- Instance type: t2.micro
- Has Apache web server and other PHP packages installed (you can use the same bash script provided in Assignment 1A to bootstrap your EC2).

This instance will host the "Photo Album" web application, which was created in Assignment 1A – more details are in Section 2 of this specification document. This instance will also act as a bastion host for you to SSH into the Test instance, which resides in a private subnet.

**Note:** [your/public.dns] will change every time the webserver instance restarts. To avoid this

**Instance type**

t2.micro

Family: t2 - 1 vCPU | 1 GiB Memory | Current generation: true  
Free tier eligible

On-Demand Windows base pricing: 0.0152 USD per hour  
On-Demand RHEL base pricing: 0.0116 USD per hour  
On-Demand SUSE base pricing: 0.0116 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: assignment1b

**Network settings**

VPC - required: vpc-0a77c0d690062c346 (BNGuyenVPC)  
10.0.0.16

Subnet Info: public-Subnet-1

**Security group**

Security group name	Protocols	Source
TestInstanceSG	All traffic	Anywhere
WebServerSG	HTTP (80), SSH (22)	Anywhere
DBServerSG	ICMP	TestInstanceSG
DBServerSG	MySQL (3306)	WebServerSG

**1.3 – EC2 virtual machine**

You will create two EC2 instances, a test instance and a bastion/web server instance.

**1.3.1 – Bastion/Web server instance**

Your web server must be deployed on an EC2 instance in Public Subnet 2. This EC2 instance should be configured similar to the EC2 created in Assignment 1A:

- Amazon Machine Image: Amazon Linux 2 AMI (HVM), SSD Volume Type
- Instance type: t2.micro
- Has Apache web server and other PHP packages installed (you can use the same bash script provided in Assignment 1A to bootstrap your EC2).

This instance will host the "Photo Album" web application, which was created in Assignment 1A – more details are in Section 2 of this specification document. This instance will also act as a bastion host for you to SSH into the Test instance, which resides in a private subnet.

**Note:** [your/public.dns] will change every time the webserver instance restarts. To avoid this

**Network settings**

- VPC - required: vpc-0a7c0d690062c346 (BNguyenVPC) 10.0.0.0/16
- Subnet: subnet-0ee55dfec8f97c0 public2-us-east-1b VPC: vpc-0a7c0d690062c346 Owner: 366309293917 Availability Zone: us-east-1b IP addresses available: 250 CIDR: 10.0.2.0/24
- Auto-assign public IP: Disable
- Firewall (security groups): Create security group (radio button selected)
- Common security groups: Select security groups (TestInstanceSG selected)
- Configure storage: Advanced

**Configure storage**

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates.

2WEI Riot Just Changed League's Entire Lore Exposing Logan 10:00 AM 10/14/2023

**Instances | EC2**

New EC2 Experience

EC2 Dashboard, EC2 Global View, Events

**Instances**

- Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations

**Images**

- AMIs, AMI Catalog

**Elastic Block Store**

- Volumes, Snapshots, Lifecycle Manager

**Network & Security**

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates.

2WEI Riot Just Changed League's Entire Lore Exposing Logan 10:04 AM 10/14/2023

**Assignment1**

Securely Conn File C:/Users/DELL/Do... COS2019

School of Science, Computing and Engineering Technologies Swinburne University of Technology

Security group name	Protocols	Source
TestInstanceSG	All traffic	Anywhere
WebServerSG	HTTP (80), SSH (22)	Anywhere
	ICMP	TestInstanceSG
DBServerSG	MySQL (3306)	WebServerSG

**1.3 – EC2 virtual machine**

You will create two EC2 instances, a test instance and a bastion/web server instance.

**1.3.1 – Bastion/Web server instance**

Your web server must be deployed on an EC2 instance in Public Subnet 2. This EC2 instance should be configured similar to the EC2 created in Assignment 1A:

- Amazon Machine Image: Amazon Linux 2 AMI (HVM), SSD Volume Type
- Instance type: t2.micro
- Has Apache web server and other PHP packages installed (you can use the same bash script provided in Assignment 1A to bootstrap your EC2).

This instance will host the "Photo Album" web application, which was created in Assignment 1A – more details are in Section 2 of this specification document. This instance will also act as a bastion host for you to SSH into the test instance, which resides in a private subnet.

**NOTE:** [your-public.dns] will change every time the webserver instance restarts. To avoid this behaviour and to ensure your Webserver URL remains persistent, add an Elastic IP Address to this instance by allocating an Elastic IP address in the same region under the Network and Security section in the left menu of the EC2 service page, then associate this new EIP to your Bastion/Web server instance. The public IP address of your instance should now automatically match this Elastic IP address.

**1.3.2 – Test instance**

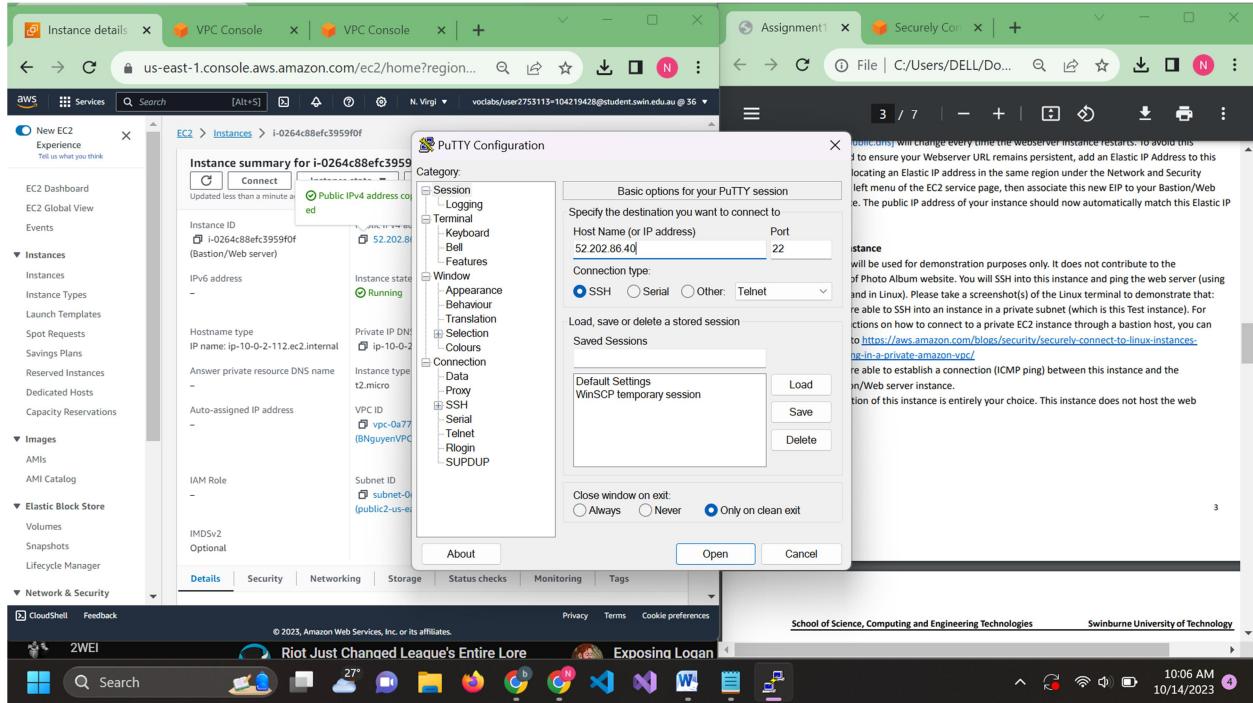
This instance will be used for demonstration purposes only. It does not contribute to the functionality of Photo Album website. You will SSH into this instance and ping the web server (using "ping" command in Linux). Please take a screenshot(s) of the Linux terminal to demonstrate that:

- You are able to SSH into an instance in a private subnet (which is this Test instance). For instructions on how to connect to a private EC2 instance through a bastion host, you can refer to <https://aws.amazon.com/blogs/security/securing-connect-to-linux-instances-running-in-a-private-amazon-vpc/>
- You are able to establish a connection (ICMP ping) between this instance and the Bastion/Web server instance.

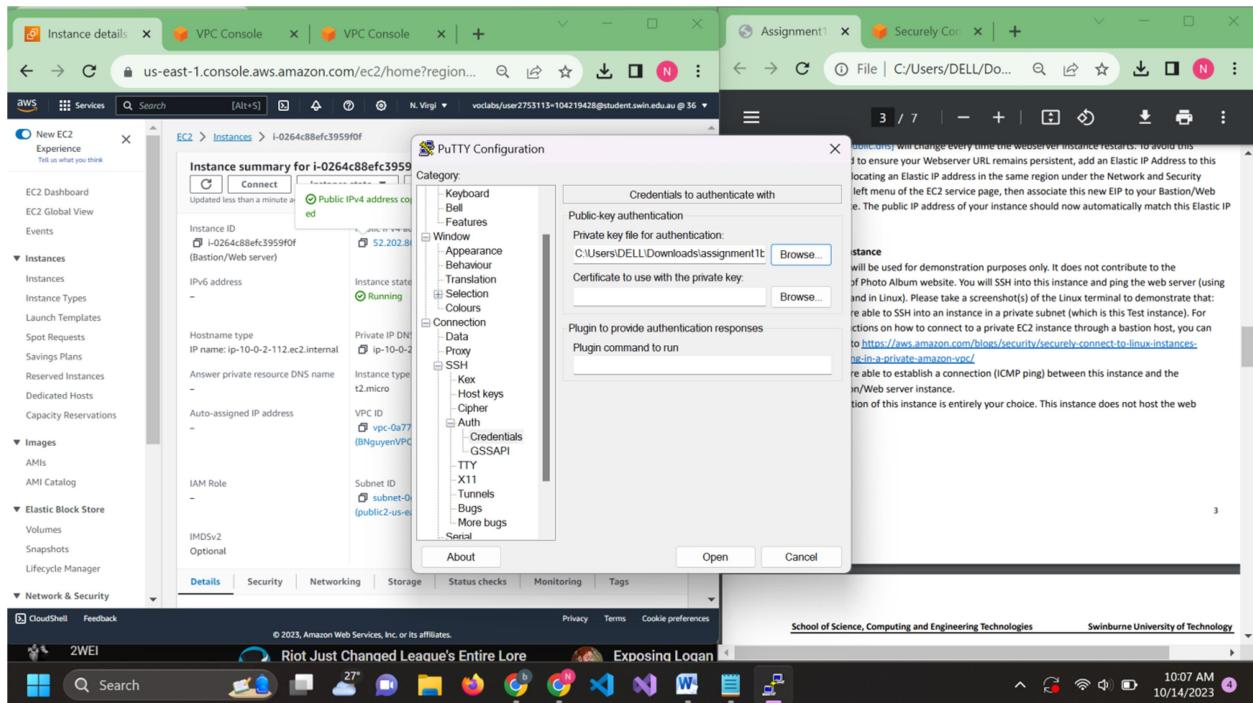
The configuration of this instance is entirely your choice. This instance does not host the web application.

#### 4.2.2: Test it using putty

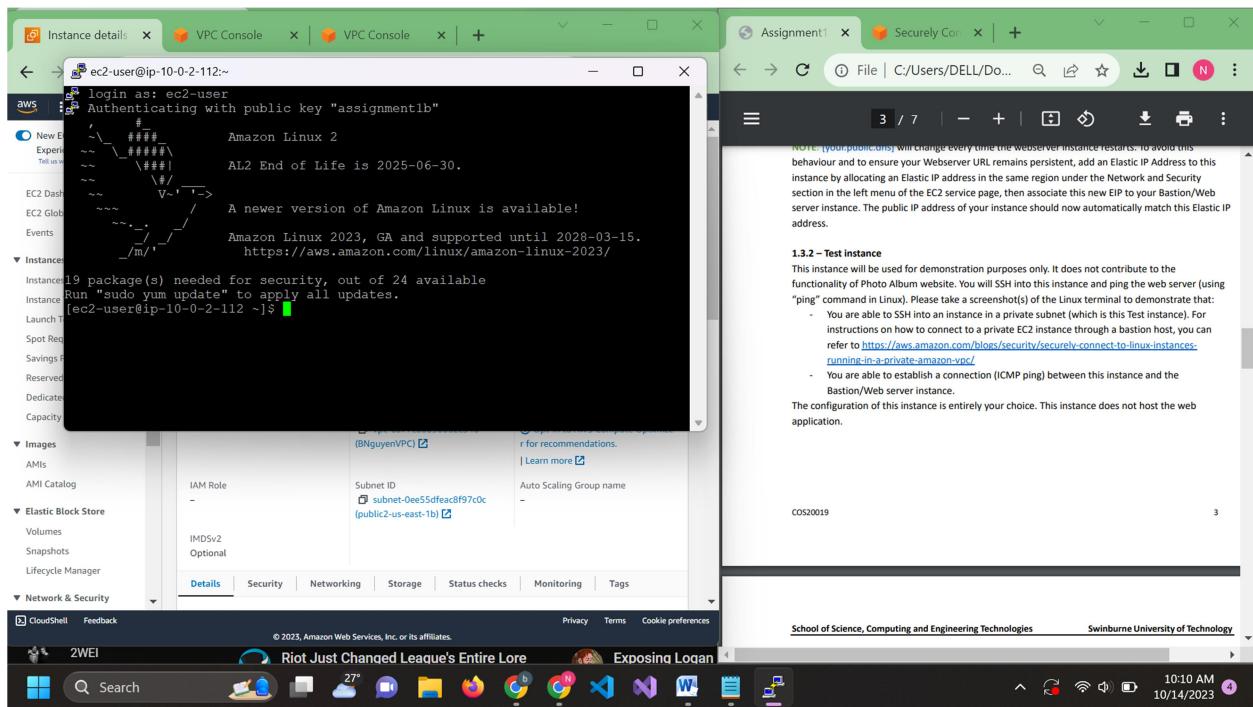
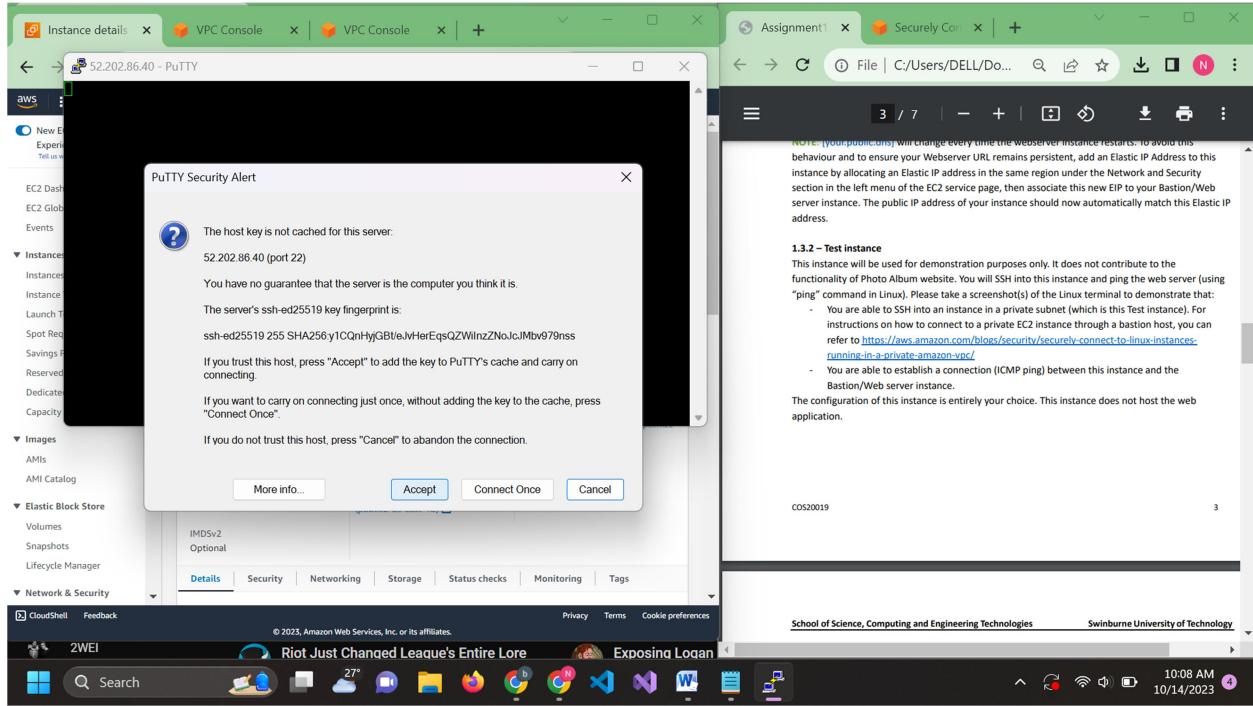
Copy Bastion public IPV4 and use it as hostname



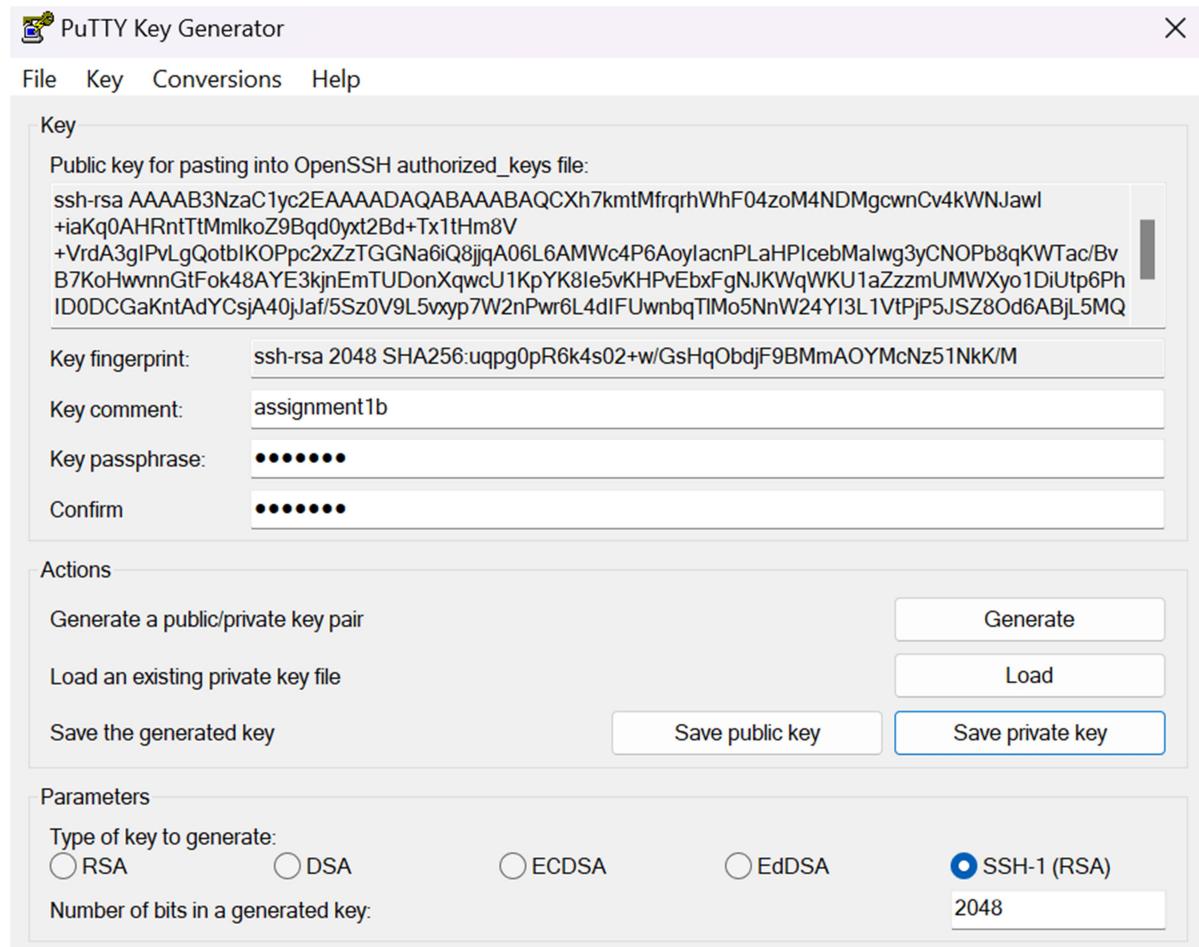
Use the keypair u created when creating Bastion



## Accept

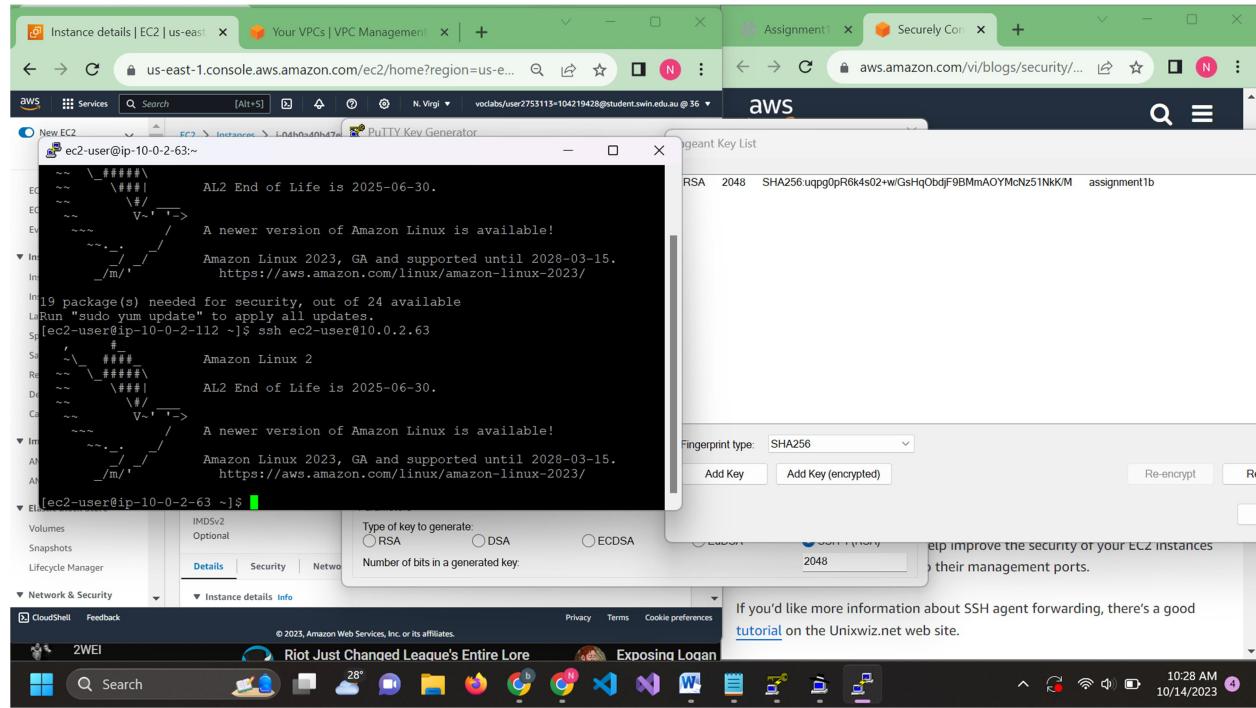


Go into puttyGen convert the created key

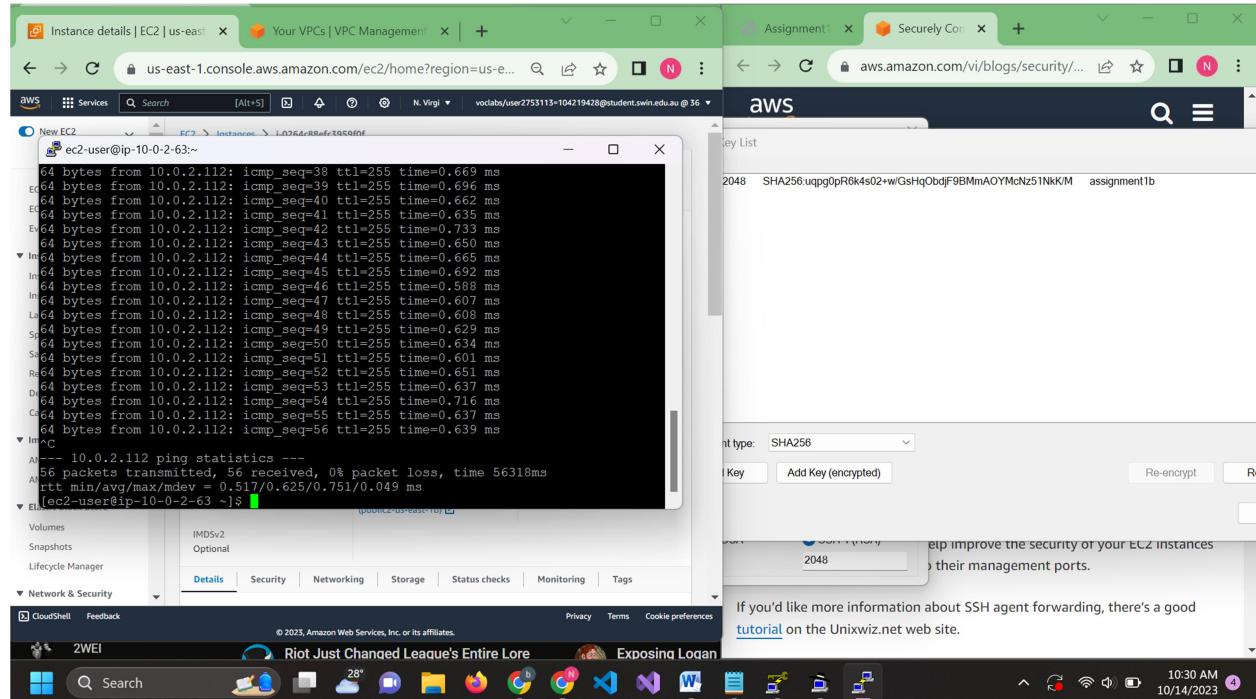


Nguyen Gia Binh- 104219428 / SWH01067

Go into putty and enter ssh [ec2-user@10.0.2.63](ssh://ec2-user@10.0.2.63) (10.0.2.63 is the Test instance private ipv4)



Ping Bastion using its private IPv4



Step 5: Create RDS database instance

The screenshot shows two side-by-side windows. On the left is the AWS RDS 'Choose a database creation method' page, where 'Standard create' is selected. It lists engine options: Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), MySQL, MariaDB, PostgreSQL, and Oracle. On the right is a PDF titled 'Assignment1' from Swinburne University of Technology, page 4/7. The PDF contains instructions for creating an RDS database instance, specifying MySQL 8.0.25 with Free tier, No public access, and private subnets. It also includes notes about security and network ACLs.

This screenshot shows the same AWS RDS creation interface and PDF assignment sheet as the previous one. The engine version is now set to MySQL 8.0.28. The 'Free Tier' template is selected. The PDF assignment sheet remains the same, providing configuration details for the RDS instance.

Password: lickmya707

The screenshot shows two side-by-side windows. On the left is the AWS RDS MySQL setup page, where a master user 'admin' is being created with a password. On the right is a Microsoft Word document titled 'Assignment1' containing instructions for setting up an RDS database instance.

**Credentials Settings**

- Master username: **Info** admin
- Master password: **Info** .....  
Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), (single quote), (double quote) and @ (at sign).
- Confirm master password: **Info** .....

**Instance configuration**

The DB instance configuration options below are limited to those supported by the engine that you selected above.

**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 resources or 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 cross-region.

**1.4 – RDS database instance**

Your RDS instance must have the following configs:

- DB engine version: MySQL 8.0.25
- Template: Free tier
- Public access: No
- Resides in private subnets.

**NOTE:** Your RDS instance needs to be in a private subnet. Only WebServerSG security group can access it. However, you need to be able to access your database over the internet so that you can set it up and maintain it. There are several ways to do this. The easiest way is to install phpMyAdmin (a web-based MySQL administration tool) on your EC2 web server instance and manage your database through phpMyAdmin's UI. Instructions on how to do this are in [Install phpMyAdmin on EC2.pdf](#).

Create a database in your RDS instance with a table called **photos** that stores meta-data about the photos stored in the S3 bucket. This table should have the following columns:

- Photo title (varchar(255) type)
- Description (varchar(255) type)
- Creation date (date type)
- Keywords (varchar(255) type)
- Reference to the photo object in S3 (varchar(255) type)

**1.5 – Network ACL**

To add an additional layer of security to your web server, you have been asked to design and deploy a Network ACL. You can find more information about Network ACLs in the [AWS Documentation](#).

The screenshot shows the same AWS RDS MySQL setup page and assignment document as the previous one, but with a focus on the 'Subnet group' configuration section. A note says 'After a database is created, you can't change its VPC'.

**DB subnet group: Info**

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

default

**Public access: Info**

Yes: 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 the database. Choose one or more VPC security groups that specify which resources can connect to the database.

**VPC security group (firewall): Info**

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

Choose existing: Choose existing VPC security groups  
Create new: Create new VPC security group

**Existing VPC security groups**

Choose one or more options  
default

**Availability Zone: Info**

No preference

**RDS Proxy**

RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and performance.

**1.4 – RDS database instance**

Your RDS instance must have the following configs:

- DB engine version: MySQL 8.0.25
- Template: Free tier
- Public access: No
- Resides in private subnets.

**NOTE:** Your RDS instance needs to be in a private subnet. Only WebServerSG security group can access it. However, you need to be able to access your database over the internet so that you can set it up and maintain it. There are several ways to do this. The easiest way is to install phpMyAdmin (a web-based MySQL administration tool) on your EC2 web server instance and manage your database through phpMyAdmin's UI. Instructions on how to do this are in [Install phpMyAdmin on EC2.pdf](#).

Create a database in your RDS instance with a table called **photos** that stores meta-data about the photos stored in the S3 bucket. This table should have the following columns:

- Photo title (varchar(255) type)
- Description (varchar(255) type)
- Creation date (date type)
- Keywords (varchar(255) type)
- Reference to the photo object in S3 (varchar(255) type)

**1.5 – Network ACL**

To add an additional layer of security to your web server, you have been asked to design and deploy a Network ACL. You can find more information about Network ACLs in the [AWS Documentation](#).

So I forgot to config subnet and put it in the database so let create the subnet group first

Nguyen Gia Binh- 104219428 / SWH01067

The screenshot shows the 'Create DB subnet group' page in the AWS RDS console. In the 'Subnet group details' section, the name is set to 'assignment1bdb-SG' and the description is 'database security group'. A VPC dropdown is set to 'BNguyenVPC (vpc-0a77c0d690062c346)'. In the 'Add subnets' section, under 'Availability Zones', multiple zones are selected: us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, and us-east-1f. The status bar at the bottom indicates it's 10:43 AM on 10/14/2023.

Select all the correspond private subnet

The screenshot shows the 'Add subnets' step in the RDS setup. Under 'Availability Zones', the same six zones are selected. In the 'Subnets' section, two specific subnets are chosen: 'subnet-044c02db4d845030c (10.0.3.0/24)' and 'subnet-039d2a28c26dea2f9 (10.0.4.0/24)'. A note states: 'For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.' The 'Subnets selected (2)' table lists the selected subnets. The status bar at the bottom indicates it's 10:44 AM on 10/14/2023.

Config the database again but change the connectivity section

# Nguyen Gia Binh- 104219428 / SWH01067

The screenshot shows the AWS RDS console in the us-east-1 region. A new database instance is being created with the following specifications:

- VPC:** BNguyenVPC (vpc-0a77c0d690062c346)
- DB subnet group:** assignment1bdb-sg (2 Subnets, 2 Availability Zones)
- Public access:** Yes
- VPC security group (firewall):** Choose existing (DBServerSG)
- Availability Zone:** us-east-1a

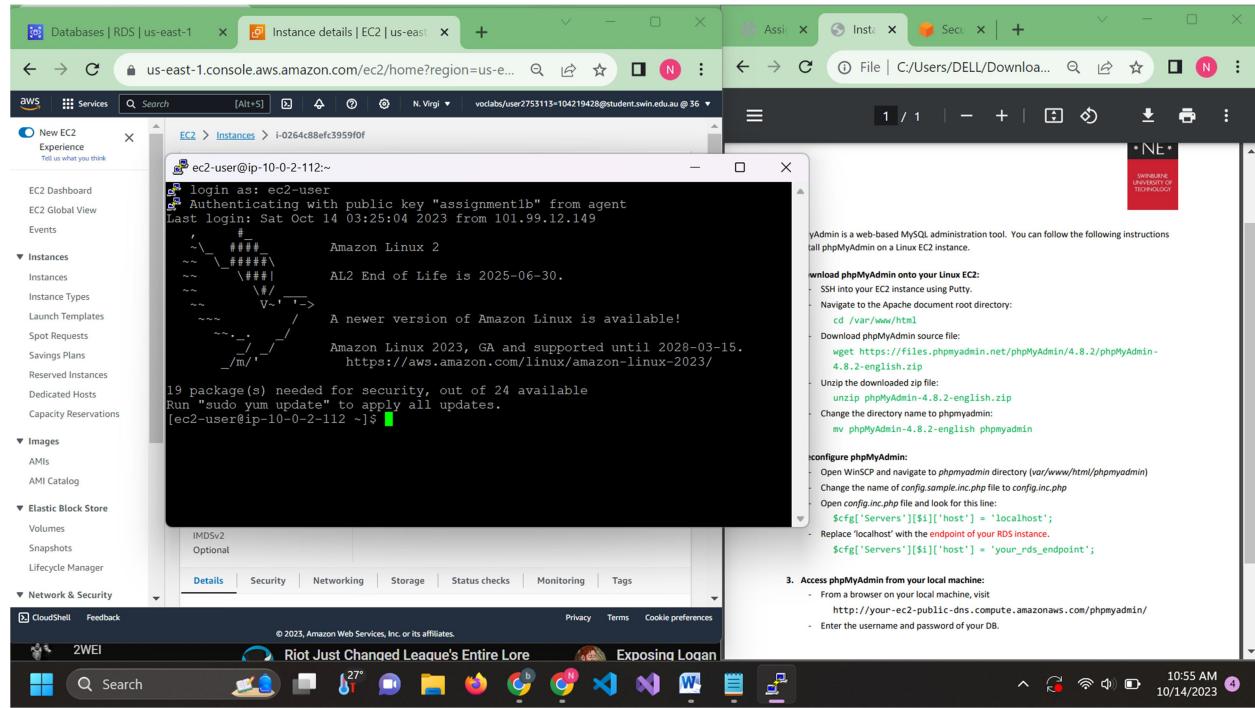
A modal window on the right provides information about MySQL, including its popularity and various features.

The screenshot shows the AWS RDS console in the us-east-1 region after a database instance has been successfully created. The success message indicates:

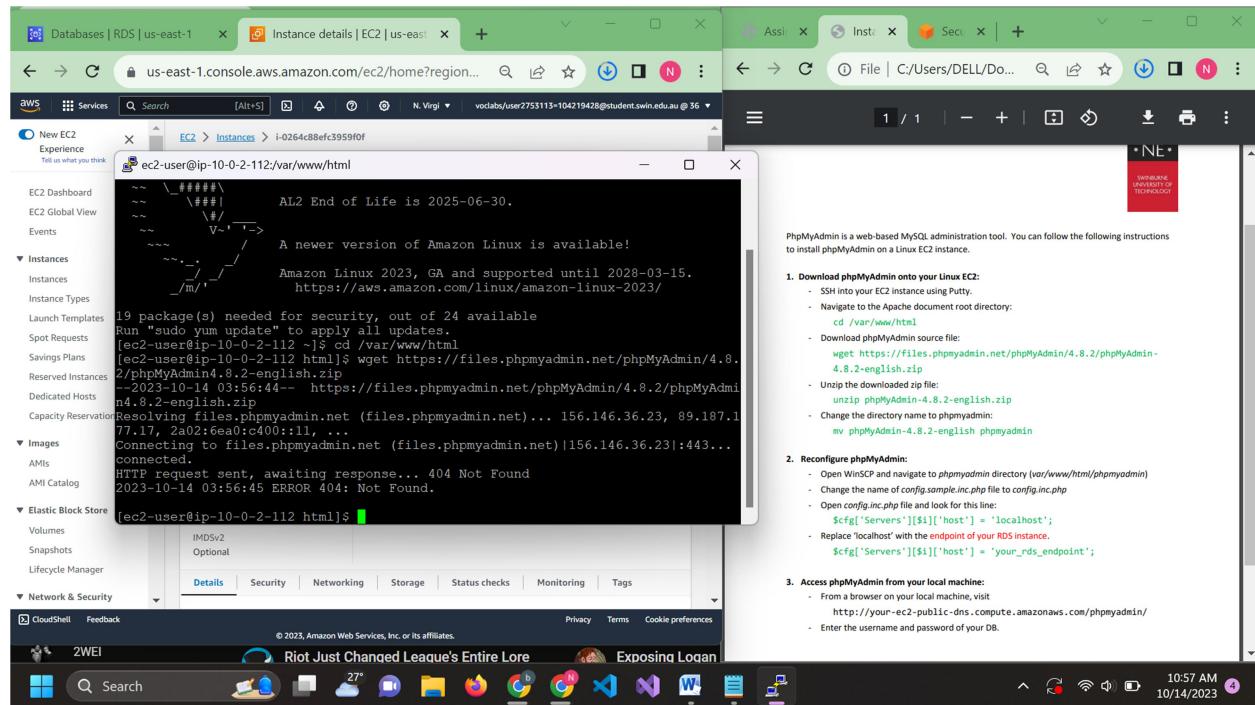
- Successfully deleted DB instance assignment1bdb
- Successfully created assignment1bdb-SG. View subnet group
- Creating database assignment1b-db. Your database might take a few minutes to launch. You can use settings from assignment1b-db to simplify configuration of suggested database add-ons while we finish creating your DB for you.
- Successfully created database assignment1bdb
- Introducing Aurora I/O-Optimized. Aurora's I/O-Optimized is a new cluster storage configuration that offers predictable pricing for all applications and improved price-performance, with up to 40% costs savings for I/O-intensive applications.

The main interface shows the newly created database "assignment1b-db" in the "Databases" list, which is currently "Creating".

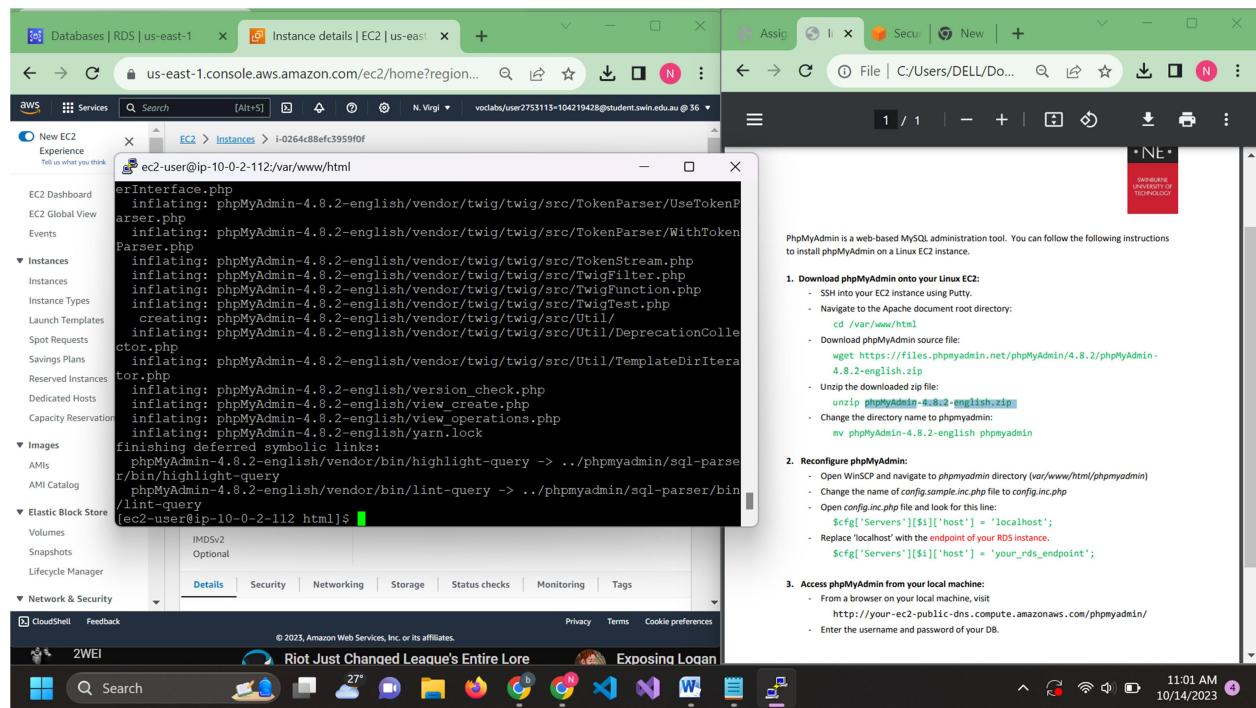
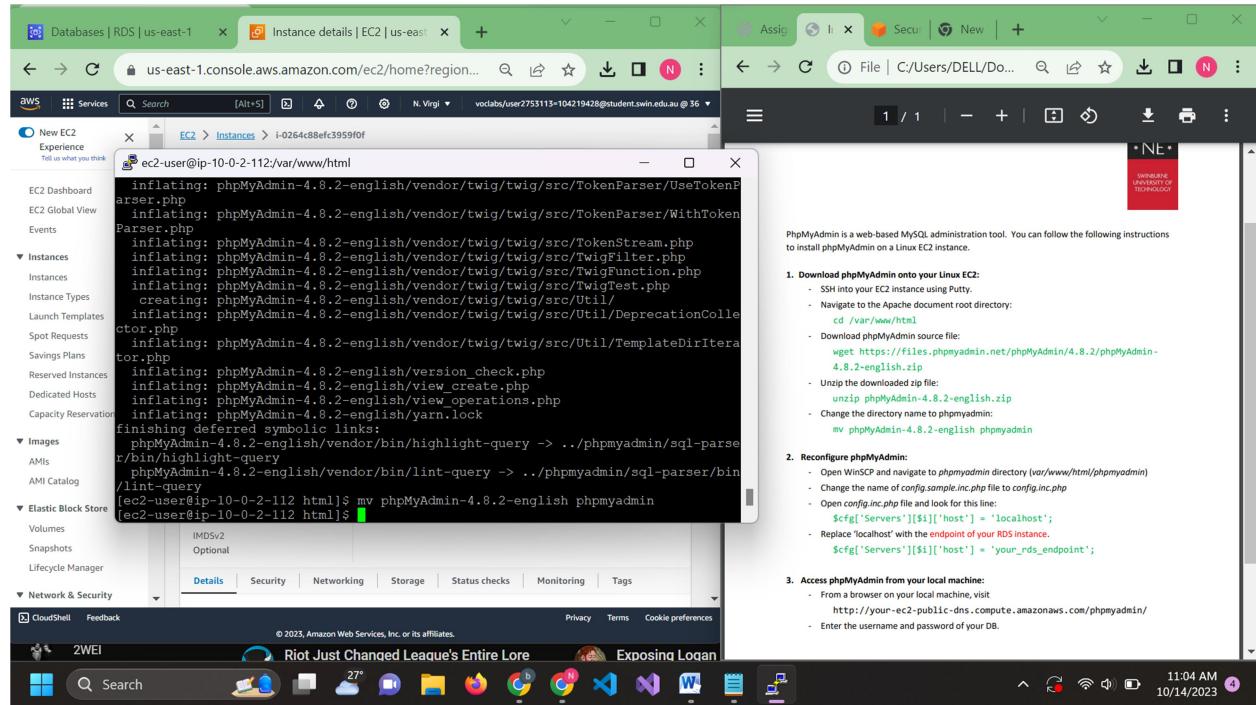
Access Putty again this time use Bastion public ipv4



To fix this u need to copy the directory from the pdf into a text editor so the “-“ is not missing(I forgot to take the screenshot of that)

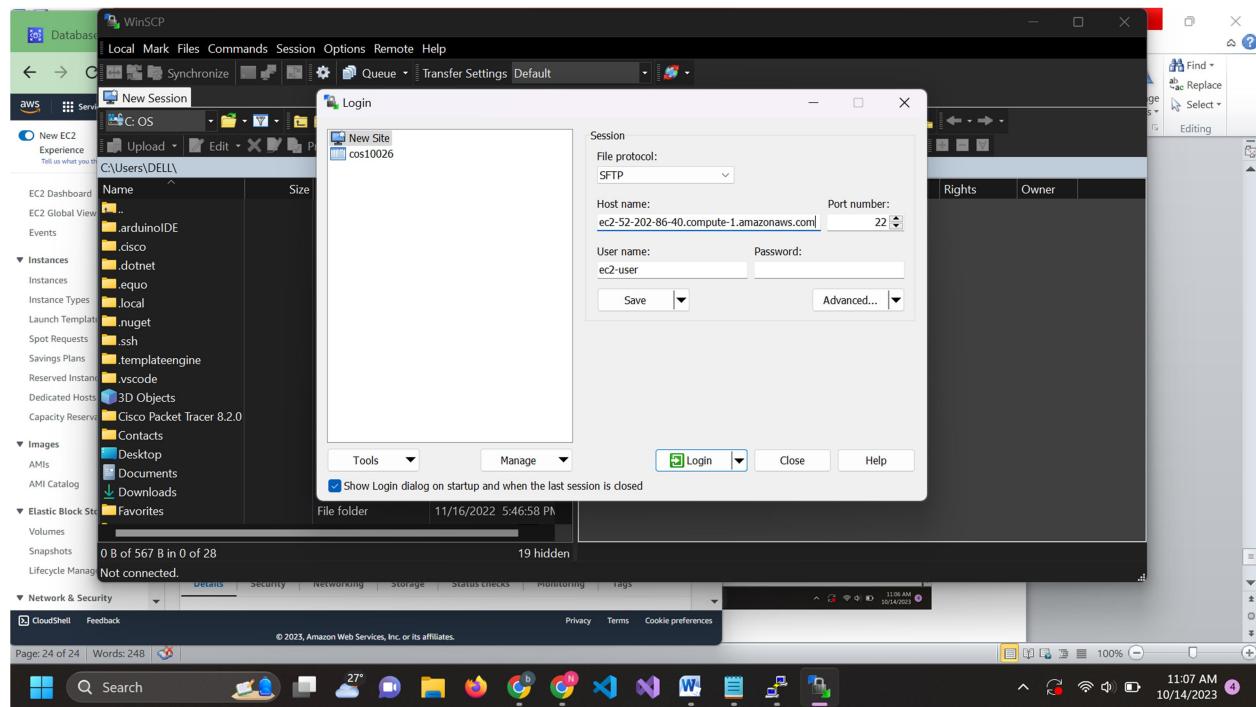
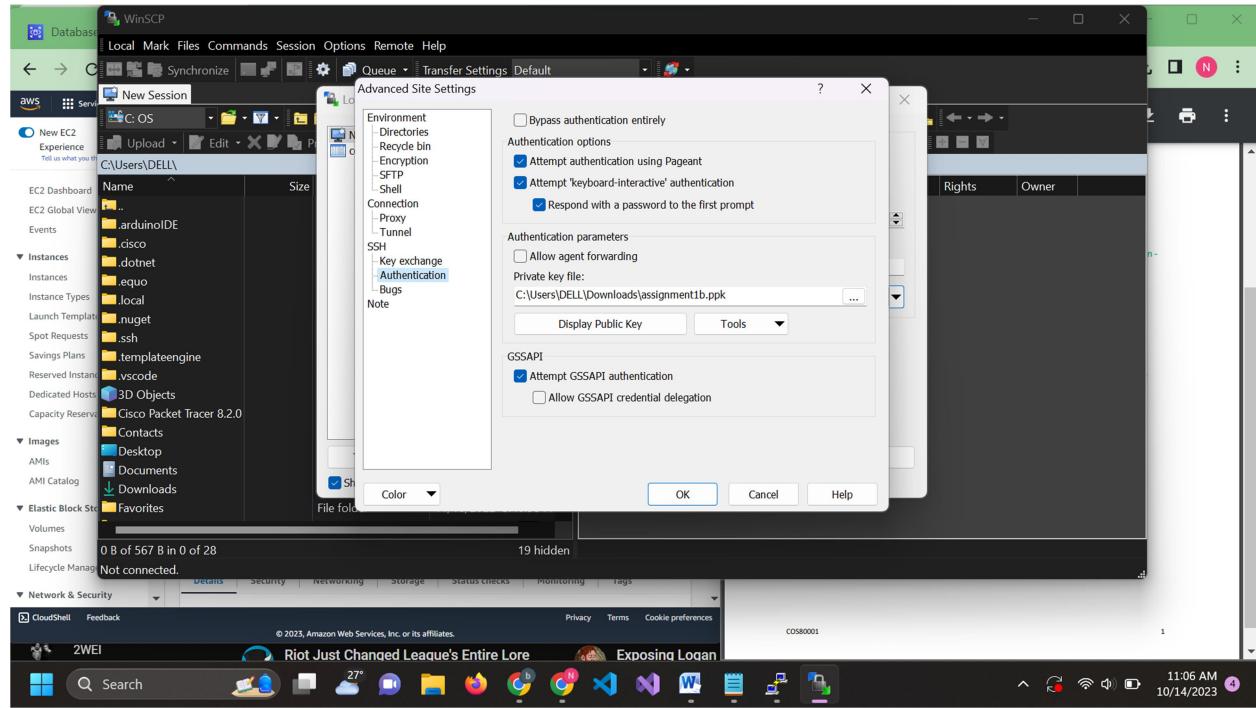


## Unzip the file

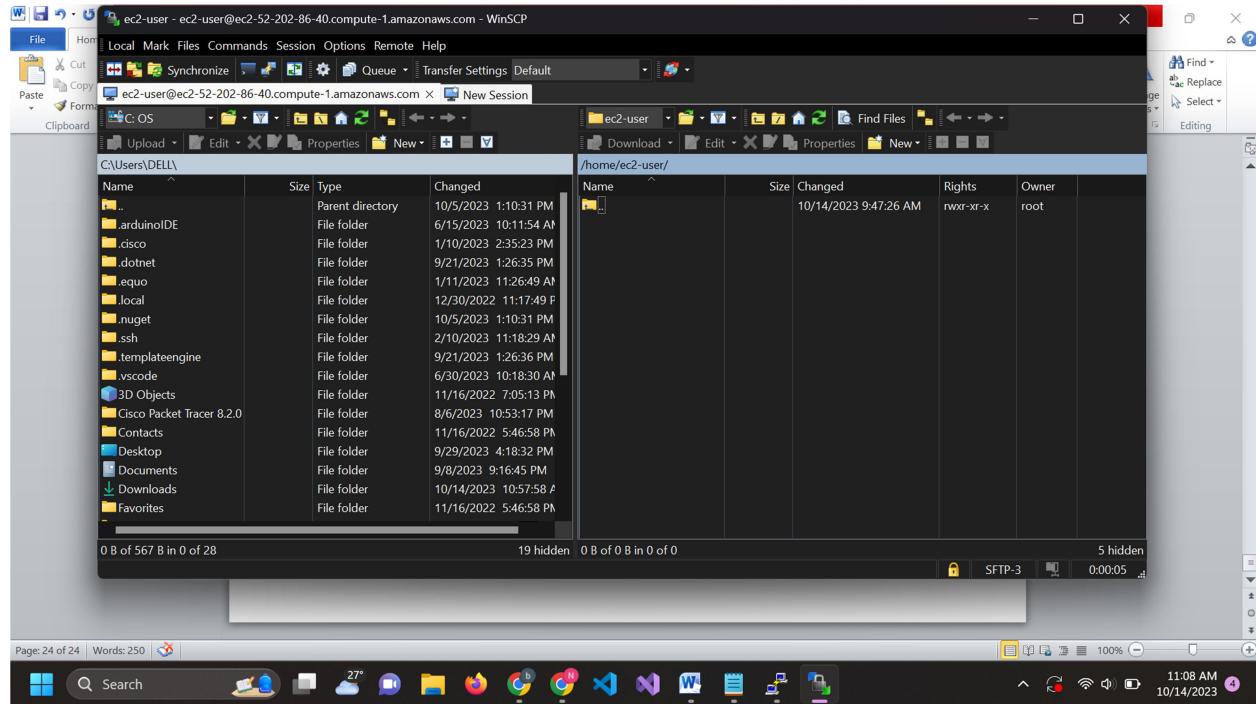


Nguyen Gia Bin- 104219428 / SWH01067

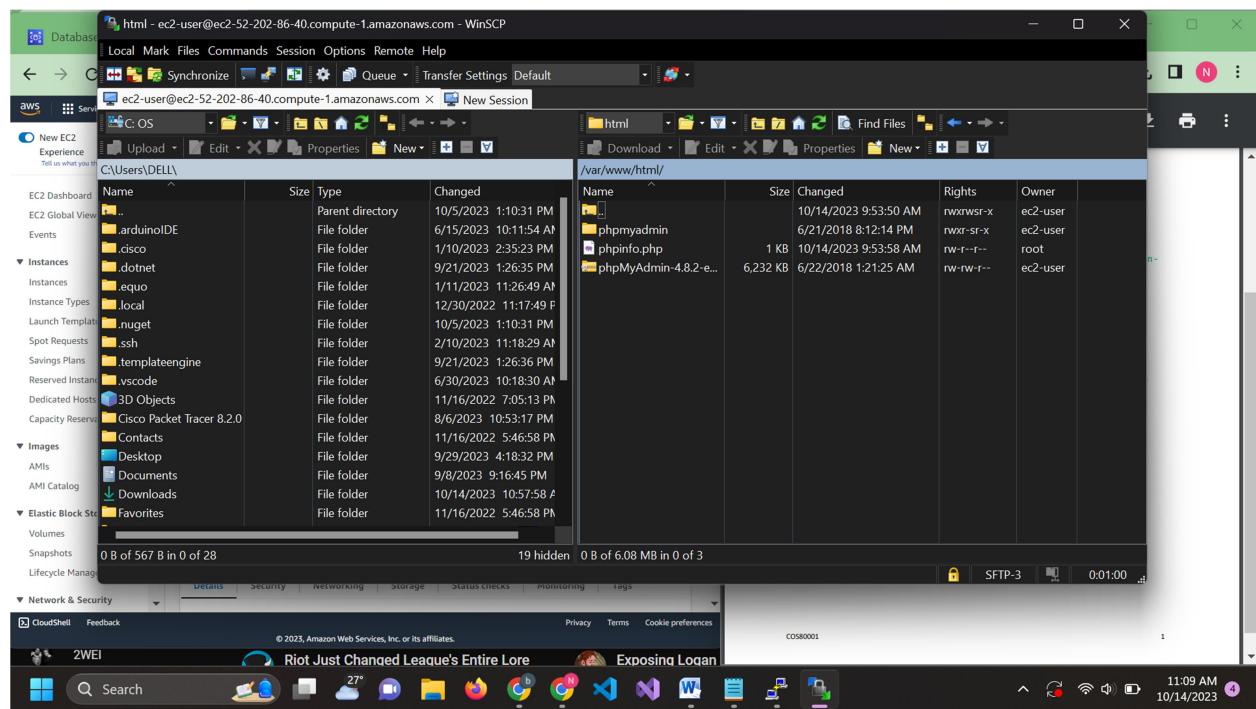
Go into WinSCP to access the Bastion using public DNS i think



Nguyen Gia Bin- 104219428 / SWH01067

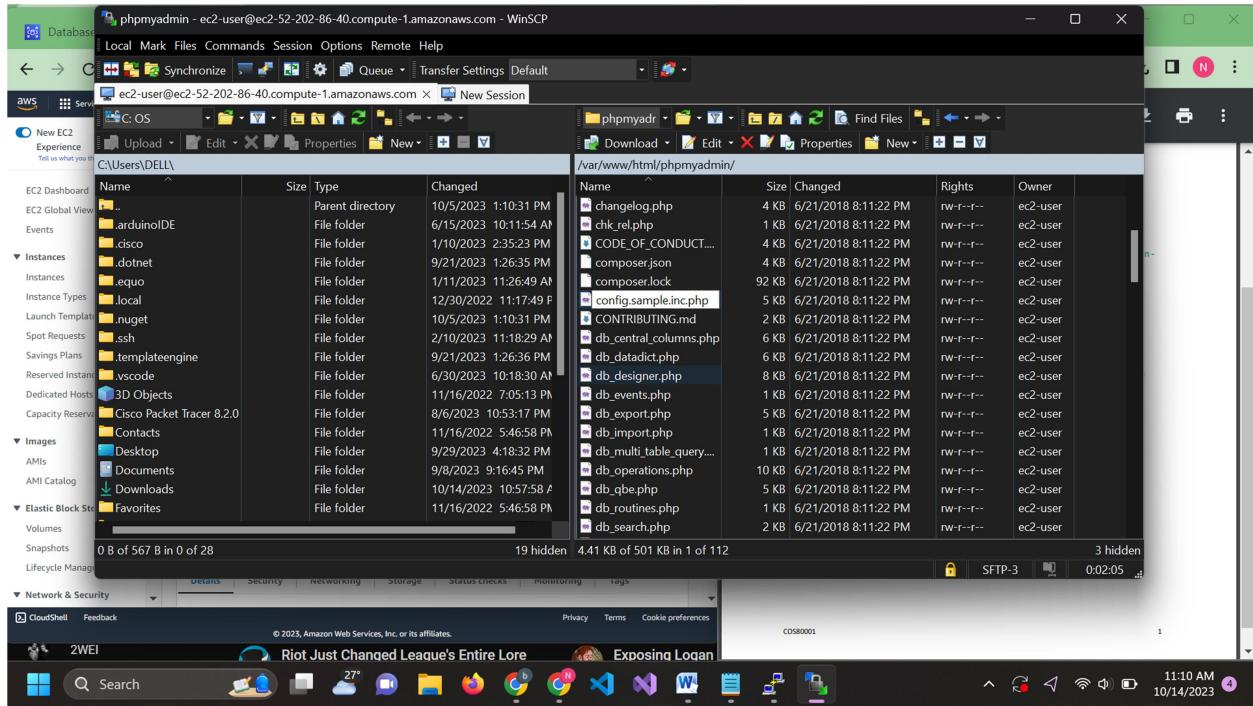


Access this directory

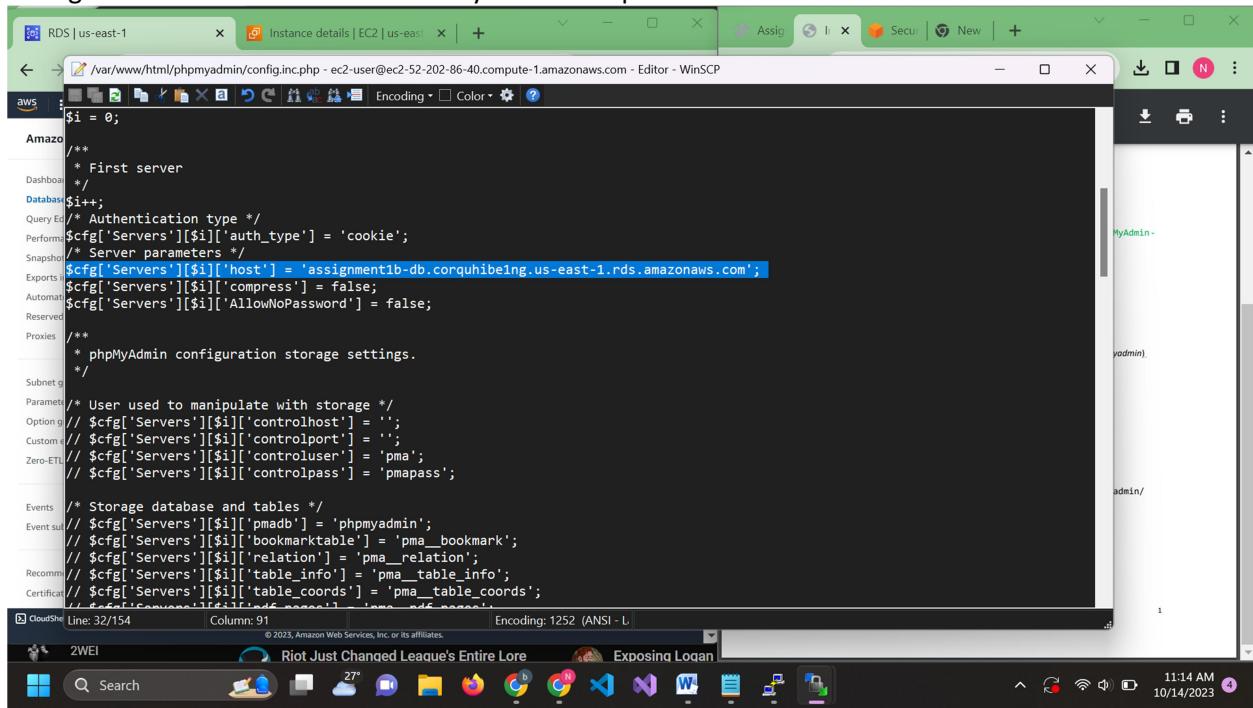


Change config.sample.inc.php to config.inc.php

Nguyen Gia Binh- 104219428 / SWH01067



Changr the code line from localhost to your rds endpoint



Enter using your public dns to test if it working(I'm not surw why I don't have to enter password for my rds

The screenshot shows two browser windows side-by-side. The left window is the AWS EC2 Instance Details page for an instance named 'i-0264c88efc3959f0f'. It displays the instance's public IP address (52.202.86.40), private IP address (10.0.2.112), and its state as 'Running'. The right window is a 'Test Page' from an EC2 instance, showing the Apache welcome message: '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.' Below this message are sections for 'If you are a member of the general public:' and 'If you are the website administrator:', both containing instructions and links.

## Endpoint link with /phpmyadmin

The screenshot shows two browser windows. The left window is the AWS RDS Connectivity & Security page for a database endpoint named 'assignment1bdb'. It lists details such as the VPC (BNguyenVPC), Subnet group (assignment1bdb-sg), and Network type (IPv4). The right window is the 'Welcome to phpMyAdmin' login page, which includes fields for 'Username' (admin) and 'Password', and a 'Go' button.

The screenshot shows the AWS RDS console with the following details:

- Endpoint & port:** assignment1b-db.corquhibeing.us-east-1.rds.amazonaws.com, Port 3306.
- Networking:** VPC security groups: DBServerSG (sg-093112a3cf136927a), Active.
- Security:** Publicly accessible: No, Certificate authority: rds-ca-2019, Certificate authority date: August 23, 2024, 00:08 (UTC+07:00), DB instance certificate expiration date: August 23, 2024, 00:08 (UTC+07:00).

The screenshot shows the phpMyAdmin interface with the following details:

- Database:** photo
- Table:** photo
- Query results:**
  - MySQL returned an empty result set (i.e. zero rows). (Query took 0.0012 seconds.)
  - SQL Query: `SELECT * FROM `photo`;`
  - Operations: Profiling [Edit inline], [Edit], [Explain SQL], [Create PHP code], [Refresh]

The table above is created using SQL command

Step 6: Create S3 bucket

**General configuration**

**Bucket name:** asm1b  
Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

**AWS Region:** US East (N. Virginia) us-east-1

**Copy settings from existing bucket - optional**  
Only the bucket settings in the following configuration are copied.  
[Choose bucket](#)

**Object Ownership** [Info](#)  
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

**ACLs disabled (recommended)**  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

**ACLs enabled**  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

**Object Ownership**  
Bucket owner enforced

**Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

**Warning:** 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.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

The screenshot shows the AWS S3 Management Console interface. At the top, there are three tabs: 'RDS | us-east-1', 'Instance details | EC2 | us-east-1', and 'S3 Management Console'. Below the tabs, the URL is s3.console.aws.amazon.com/s3/buckets?region=us-east-1. The main content area is titled 'Amazon S3 > Buckets'. It features an 'Account snapshot' section with a link to 'View Storage Lens dashboard'. Below this is a table titled 'Buckets (1) Info' with one entry: 'asm1bphoto' located in 'US East (N. Virginia) us-east-1' with 'Access' set to 'Objects can be public' and 'Creation date' as 'October 14, 2023, 11:40:15 (UTC+07:00)'. There are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. A search bar at the bottom allows 'Find buckets by name'. The bottom of the screen shows the Windows taskbar with various pinned icons like File Explorer, Edge, and File Manager.

## Step 7: Create network ACLs

The screenshot shows two windows side-by-side. On the left is the 'Create network ACL' wizard in the AWS VPC console. It has a step counter '4 / 7'. Step 1: 'Network ACL settings' shows a 'Name - optional' field with 'PublicSubnet2NACL' and a 'VPC' dropdown with 'vpc-0a77c0d690062c346 (BNguyenVPC)'. Step 2: 'Tags' shows a key-value pair 'Name' with 'PublicSubnet2NACL'. Step 3: 'Next Step' button. Step 4: 'Create network ACL' button. On the right is a document page titled '1.5 – Network ACL' with the following content:

- Creation date (date type)
- Keywords (varchar(255) type)
- Reference to the photo object in S3 (varchar(255) type)

**1.5 – Network ACL**

To add an additional layer of security to your web server, you have been asked to design and deploy a Network ACL (named "PublicSubnet2NACL") that limits ICMP and other necessary traffic to the corresponding subnet (Public Subnet 2). This NACL must follow the least-privilege principle. In other words, irrelevant traffic from irrelevant sources must not be allowed. To be specific, the NACL:

- must ALLOW SSH(22) traffic from anywhere so that you can access the WebServer instance.
- must ALLOW ICMP traffic only from the subnet that contains the Test instance.
- must ALLOW other necessary traffic so that the Photo Album website is fully functional for users from anywhere.

**2. Functional requirements of Photo Album website**

Your Photo Album website must have the following functional requirements.

**2.1 – Photo storage**

Create an S3 bucket to store your photos. Manually upload some photos onto S3 bucket that you just created and ensure they have been successfully uploaded.

<sup>1</sup> Ideally, SSH(22) traffic should only be allowed from your home network's public IPv4 address range since common users do not need to access the web server. But for simplicity, you can allow SSH from anywhere in this assignment.

The bottom of the screen shows the Windows taskbar with various pinned icons like File Explorer, Edge, and File Manager.

The screenshot shows a dual-monitor setup. The left monitor displays the AWS VPC Network ACLs console, specifically the 'Network ACLs (3)' page. A modal window is open, stating 'You successfully created acl-074deaf274c6da33e / PublicSubnet2NACL.' The main table lists three Network ACLs: 'acl-01736c6b6a3d98d20' (4 Subnets), 'acl-0b81796d203c78a83' (6 Subnets), and 'PublicSubnet2NACL' (acl-074deaf274c6da33e) which is currently selected. The right monitor displays a Microsoft Word document with the title '1.5 – Network ACL'. The document contains the following requirements:

- Creation date: (date type)
- Keywords (varchar(255) type)
- Reference to the photo object in S3 (varchar(255) type)

**1.5 – Network ACL**

To add an additional layer of security to your web server, you have been asked to design and deploy a Network ACL (named "PublicSubnet2NACL") that limits ICMP and other necessary traffic to the corresponding subnet (Public Subnet 2). This NACL must follow the least-privilege principle. In other words, irrelevant traffic from irrelevant sources must not be allowed. To be specific, the NACL:

- must ALLOW SSH(22) traffic from anywhere so that you can access the WebServer instance.
- must ALLOW ICMP traffic only from the subnet that contains the Test instance.
- must ALLOW other necessary traffic so that the Photo Album website is fully functional for users from anywhere.

**2. Functional requirements of Photo Album website**

Your Photo Album website must have the following functional requirements.

**2.1 – Photo storage**

Create an S3 bucket to store your photos. Manually upload some photos onto S3 bucket that you just created and ensure they have been successfully uploaded.

<sup>1</sup>Ideally, SSH(22) traffic should only be allowed from your home network's public IPv4 address range since common users do not need to access the web server. But for simplicity, you can allow SSH from anywhere in this assignment.

Choose the Network acl you created and go into action to edit inbound/outbound and subnet association so it can be access from anywhere

### 7.1: Inbound

The screenshot shows the 'Edit inbound rules' interface for the 'PublicSubnet2NACL'. The table lists the following inbound rules:

Rule number	Type	Protocol	Port range	Source	Allow/Deny
3	SSH (22)	TCP (6)	22	0.0.0.0/0	Allow
4	All ICMP - IPv4	ICMP (1)	All	0.0.0.0/0	Allow
1	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
2	HTTPS (443)	TCP (6)	443	0.0.0.0/0	Allow
5	All TCP	TCP (6)	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Buttons at the bottom include 'Add new rule', 'Sort by rule number', 'Cancel', 'Preview changes', and 'Save changes'.

## 7.2: Outbound

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
1	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
2	HTTPS (443)	TCP (6)	443	0.0.0.0/0	Allow
3	SSH (22)	TCP (6)	22	0.0.0.0/0	Allow
4	All ICMP - IPv4	ICMP (1)	All	0.0.0.0/0	Allow
5	All TCP	TCP (6)	All	0.0.0.0/0	Allow
	All traffic		All	0.0.0.0/0	Deny

**Add new rule** **Sort by rule number**

**Cancel** **Preview changes** **Save changes**

## 7.3: Subnet

Name	Subnet ID	Associated with	Availability Zone	IPv4 CIDR	IPv6 CIDR
public1-us-east-1a	subnet-0d74a915a109f4c8e	acl-01736c6b6a3d98d20	us-east-1a	10.0.1.0/24	-
public2-us-east-1b	subnet-0ee55dfeac8f97c0c	acl-01736c6b6a3d98d20	us-east-1b	10.0.2.0/24	-
private1-us-east-1a	subnet-044c02db4d845030c	acl-01736c6b6a3d98d20	us-east-1a	10.0.3.0/24	-
private2-us-east-1b	subnet-039d2a28c26dea2f9	acl-01736c6b6a3d98d20	us-east-1b	10.0.4.0/24	-

**Available subnets (1/4)**

**Selected subnets**

**Cancel** **Save changes**

# Nguyen Gia Binh- 104219428 / SWH01067

The screenshot shows the AWS VPC Management Console with the Network ACLs page open. A success message at the top states: "You have successfully updated subnet associations for acl-074def274c6da33e / PublicSubnet2NACL." The main table lists three Network ACLs:

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules count	Outbound rules count
ad-01736c6b6a3d98d20	3 Subnets	Yes	vpc-0a77c0d690062c346 / BNguyenVPC	2 Inbound rules	2 Outbound rules	
ad-0b81796d203c78a83	6 Subnets	Yes	vpc-0ac460d15db259b83	2 Inbound rules	2 Outbound rules	
<b>PublicSubnet2NACL</b>	<b>acl-074def274c6da33e</b>	<b>subnet-039d2a28c26dea2f9 / private2-us-east-1b</b>	<b>No</b>	<b>vpc-0a77c0d690062c346 / BNguyenVPC</b>	<b>6 Inbound rules</b>	<b>6 Outbound rules</b>

Below the table, a detailed view of the selected Network ACL (acl-074def274c6da33e) is shown, including its details, inbound rules, outbound rules, subnet associations, and tags.

## Step 8: upload photo into your S3 bucket

The screenshot shows the AWS S3 console with the "Upload" interface open. A file named "asm1bphoto.jpg" is being uploaded. The destination is set to "s3://asm1bphoto". To the right, a browser window displays a photo album website titled "School of Science, Computing and Engineering Technologies, Swinburne University of Technology". The website lists three photos from the "COS20019" folder, all of which are publicly available. The photos are "dark-cosmic-jhin-sp...", "jhin-dark-cosmic-lo...", and "jhin-empyrean-lol...". The website also includes sections for "Photo meta-data in RDS Database" and "Photo Album website functionality".

## 8.2: Edit bucket policy

The screenshot shows two browser tabs side-by-side. The left tab is the AWS S3 console under the 'Bucket policy' section, displaying a JSON policy document. The right tab is a blog post titled 'Policy' with a sample JSON code for an S3 bucket policy:

```

1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Effect": "Allow",
6        "Principal": "*",
7        "Action": "s3:GetObject",
8        "Resource": "arn:aws:s3:::my-bucket/*"
9      }
10   ]
11 }

```

Below the code, a note says: "Save the changes you've made to the bucket's policy and your bucket will have public read access enabled."

The right tab also contains a numbered list and a bullet point related to CORS settings.

## Step 9: Put your photo into the database to test it

### 9.1 copy photo url

<https://asm1bphoto.s3.amazonaws.com/dark-cosmic-jhin-splash-art-lol-4K-87.jpg>

The screenshot shows two browser tabs side-by-side. The left tab is the AWS S3 console under the 'Properties' tab for an object named 'dark-cosmic-jhin-splash-art-lol-4K-87.jpg'. The right tab is a school website (Swinburne University of Technology) showing a logo and some meta-data information.

The left tab displays the following object properties:

Property	Value
Owner	awsblsc0w6337101t1695136062
S3 URI	<a href="https://asm1bphoto/dark-cosmic-jhin-splash-art-lol-4K-87.jpg">https://asm1bphoto/dark-cosmic-jhin-splash-art-lol-4K-87.jpg</a>
AWS Region	US East (N. Virginia) us-east-1
Amazon Resource Name (ARN)	<a href="arn:aws:s3:::asm1bphoto/dark-cosmic-jhin-splash-art-lol-4K-87.jpg">arn:aws:s3:::asm1bphoto/dark-cosmic-jhin-splash-art-lol-4K-87.jpg</a>
Last modified	October 14, 2023, 11:57:06 (UTC+0:00)
Entity tag (Etag)	<a href="#">a1d0eecef63b831d8534ff785ea967ec</a>
Size	250.7 KB
Type	jpg
Object URL	<a href="https://asm1bphoto.s3.amazonaws.com/dark-cosmic-jhin-splash-art-lol-4K-87.jpg">https://asm1bphoto.s3.amazonaws.com/dark-cosmic-jhin-splash-art-lol-4K-87.jpg</a>
Key	dark-cosmic-jhin-splash-art-lol-4K-87.jpg

The right tab shows a logo for 'School of Science, Computing and Engineering Technologies, Swinburne University of Technology'. It includes a note about making objects publicly available and a section on 'Photo meta-data in RDS Database' with a table of meta-data fields:

Field	Description
Photo title	Swinburne Logo
Description	Logo of Swinburne uni
Creation date	2021-08-09
Keywords	logo, university
Object URL in S3	<a href="https://photo-bucket.s3.amazonaws.com/swinburnelogo.jpg">https://photo-bucket.s3.amazonaws.com/swinburnelogo.jpg</a>

Insert it into the table in your database using sql

# Nguyen Gia Bin- 104219428 / SWH01067

The screenshot shows the phpMyAdmin interface for a database named 'photo'. In the 'Run SQL query/queries on table photo.photo:' panel, the following SQL code is entered:

```
1 INSERT INTO photo (title, description, date, keywords, refference)
2 VALUES ('Jhin', 'Jhin waalpaper', '2023-10-14', 'Jhin, darkstar,wallpaper',
'https://asm1bphoto.s3.amazonaws.com/dark-cosmic-jhin-splash-art-lol-4K-87.jpg')
```

The 'Columns' panel on the right lists the table structure:

title	description	date	keywords	refference
-------	-------------	------	----------	------------

Below the SQL panel are several buttons: SELECT\*, SELECT, INSERT, UPDATE, DELETE, Clear, Format, Get auto-saved query, Bind parameters, Delimiter (set to ;), Show this query here again (checked), Retain query box, Rollback when finished, Enable foreign key checks, and Go.

The screenshot shows the phpMyAdmin interface after the SQL query has been executed. A message at the top states: "Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)".

The SQL query listed is:

```
SELECT * FROM `photo`
```

The results table shows one row:

title	description	date	keywords	refference
Jhin	Jhin waalpaper	2023-10-14	Jhin, darkstar,wallpaper	https://asm1bphoto.s3.amazonaws.com/dark-cosmic-jh...

Below the results are 'Query results operations' buttons: Print, Copy to clipboard, Export, Display chart, Create view.

Edit constant.php

# Nguyen Gia Binh- 104219428 / SWH01067

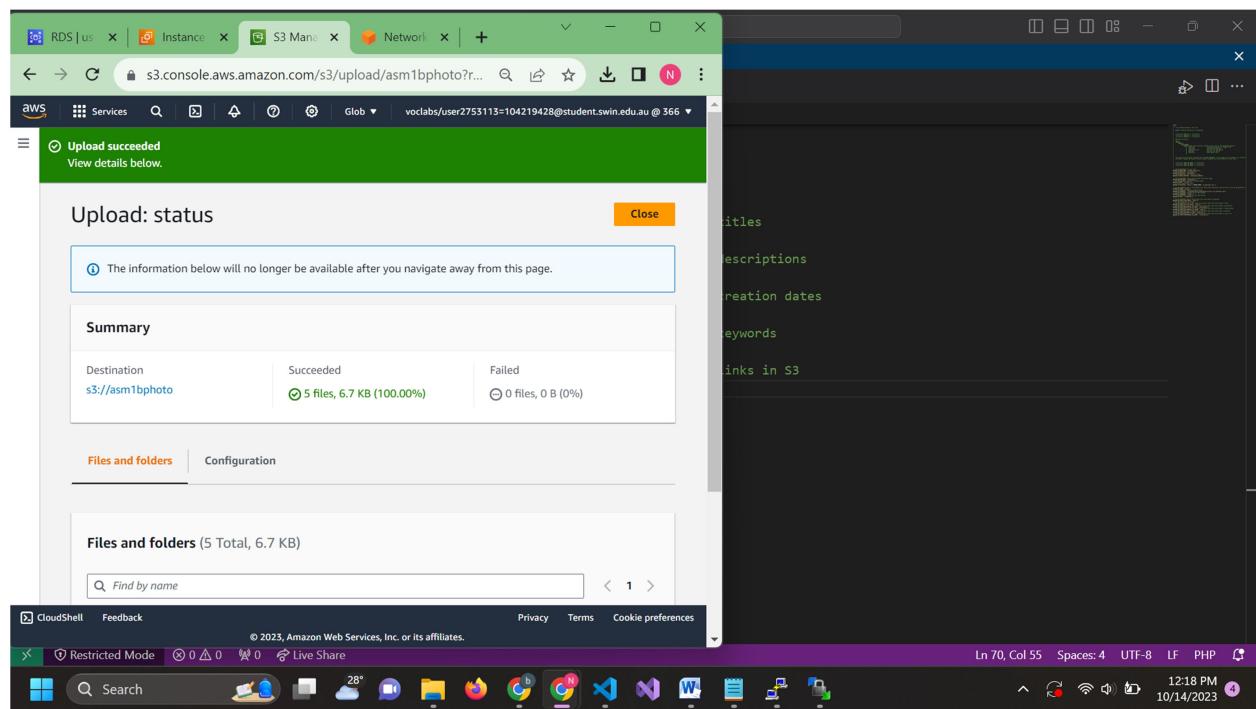
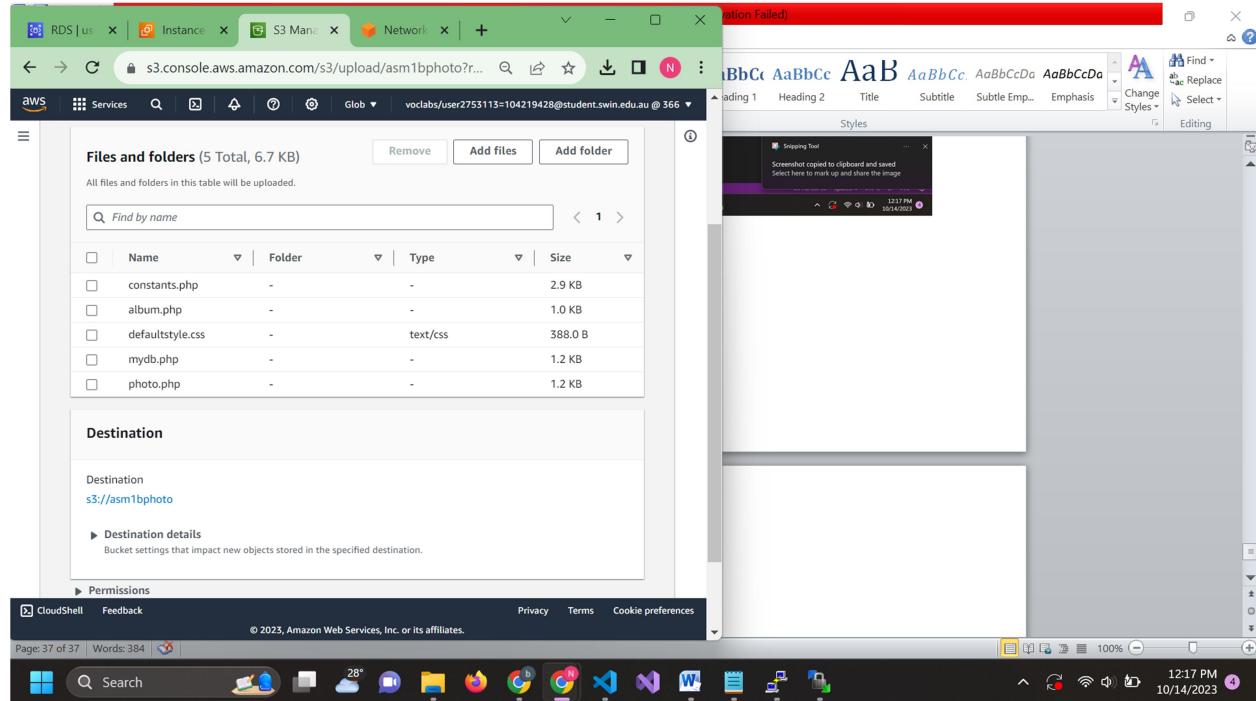
A screenshot of a code editor window titled "constants.php". The code defines various variables for a photo album application. It includes constants for student name ('Nguyen Gia Binh'), student ID ('104219428'), tutorial session ('Saturday 12:00AM'), S3 bucket name ('asm1bphoto'), region ('us-east-1'), DB name ('photo'), DB endpoint ('assignment1b-db.corquhibe1.us-east-1.rds.amazonaws.com'), DB username ('admin'), DB password ('lickmya707'), and DB table name ('photo'). The code is annotated with comments indicating required values.

```
33  */
34
35 // [ACTION REQUIRED] your full name
36 define('STUDENT_NAME', 'Nguyen Gia Binh');
37 // [ACTION REQUIRED] your Student ID
38 define('STUDENT_ID', '104219428');
39 // [ACTION REQUIRED] your tutorial session
40 define('TUTORIAL_SESSION', 'Saturday 12:00AM');
41
42 // [ACTION REQUIRED] name of the S3 bucket that stores images
43 define('BUCKET_NAME', 'asm1bphoto');
44 // [ACTION REQUIRED] region of the above bucket
45 define('REGION', 'us-east-1');
46 // no need to update this const
47 define('S3_BASE_URL','https://'.BUCKET_NAME.'.s3.amazonaws.com/');
48
49 // [ACTION REQUIRED] name of the database that stores photo meta-data (note that this is not the DB identifier of the RDS instance)
50 define('DB_NAME', 'photo');
51 // [ACTION REQUIRED] endpoint of RDS instance
52 define('DB_ENDPOINT', 'assignment1b-db.corquhibe1.us-east-1.rds.amazonaws.com');
53 // [ACTION REQUIRED] username of your RDS instance
54 define('DB_USERNAME', 'admin');
55 // [ACTION REQUIRED] password of your RDS instance
56 define('DB_PWD', 'lickmya707');
57
58 // [ACTION REQUIRED] name of the DB table that stores photo's meta-data
59 define('DB_PHOTO_TABLE_NAME', 'photo');
60 // The table above has 5 columns:
```

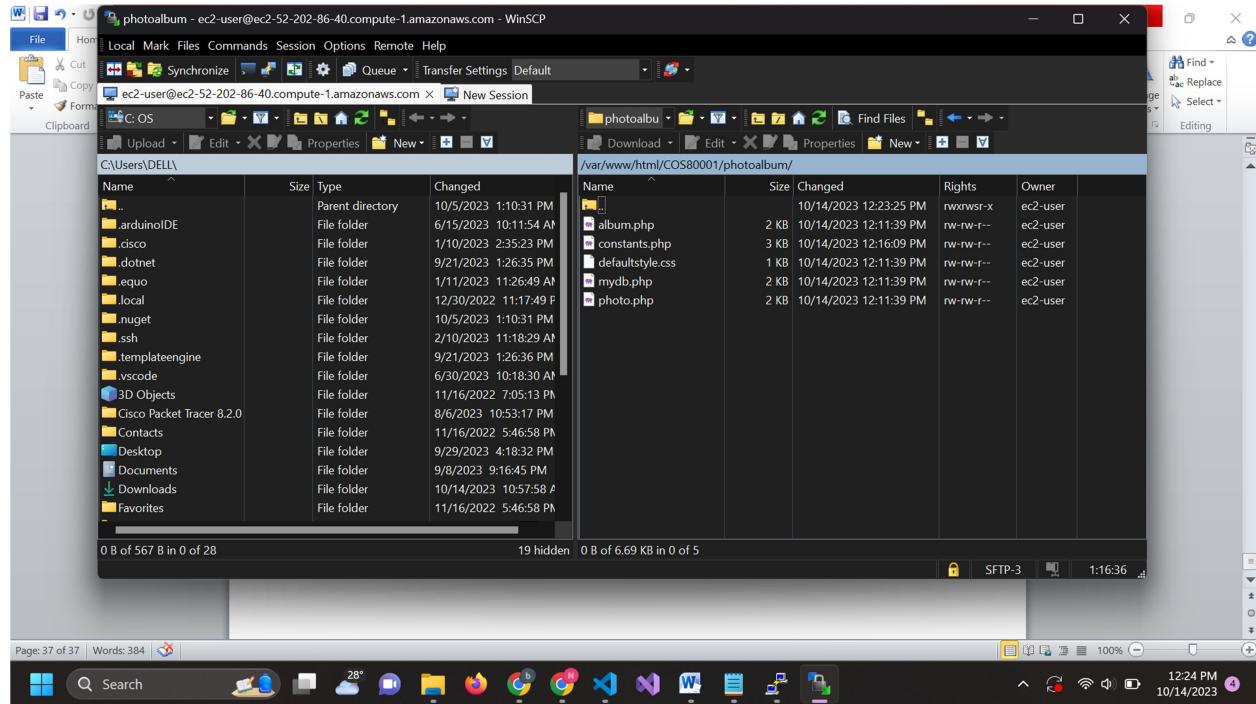
A screenshot of a code editor window titled "constants.php". The code defines variables for a photo album application, including DB password ('lickmya707'), DB table name ('photo'), column names for titles ('title'), descriptions ('description'), creation dates ('date'), keywords ('keywords'), and links ('reference'), and a column for S3 references ('s3reference\_col\_name'). A Snipping Tool window is overlaid on the bottom right, showing a screenshot of the code editor and the message "Screenshot copied to clipboard and saved".

```
56 define('DB_PWD', 'lickmya707');
57
58 // [ACTION REQUIRED] name of the DB table that stores photo's meta-data
59 define('DB_PHOTO_TABLE_NAME', 'photo');
60 // The table above has 5 columns:
61 // [ACTION REQUIRED] name of the column in the above table that stores photo's titles
62 define('DB_PHOTO_TITLE_COL_NAME', 'title');
63 // [ACTION REQUIRED] name of the column in the above table that stores photo's descriptions
64 define('DB_PHOTO_DESCRIPTION_COL_NAME', 'description');
65 // [ACTION REQUIRED] name of the column in the above table that stores photo's creation dates
66 define('DB_PHOTO_CREATIONDATE_COL_NAME', 'date');
67 // [ACTION REQUIRED] name of the column in the above table that stores photo's keywords
68 define('DB_PHOTO_KEYWORDS_COL_NAME', 'keywords');
69 // [ACTION REQUIRED] name of the column in the above table that stores photo's links in S3
70 define('DB_PHOTO_S3REFERENCE_COL_NAME', 's3reference');
```

## Upload file to your bucket

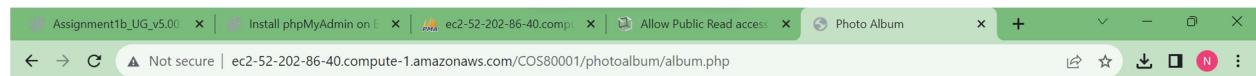


Upload the files into the directory in WinSCP



Access it in your local brower using Bastion public DNS

<http://ec2-52-202-86-40.compute-1.amazonaws.com/COS80001/photoalbum/album.php>

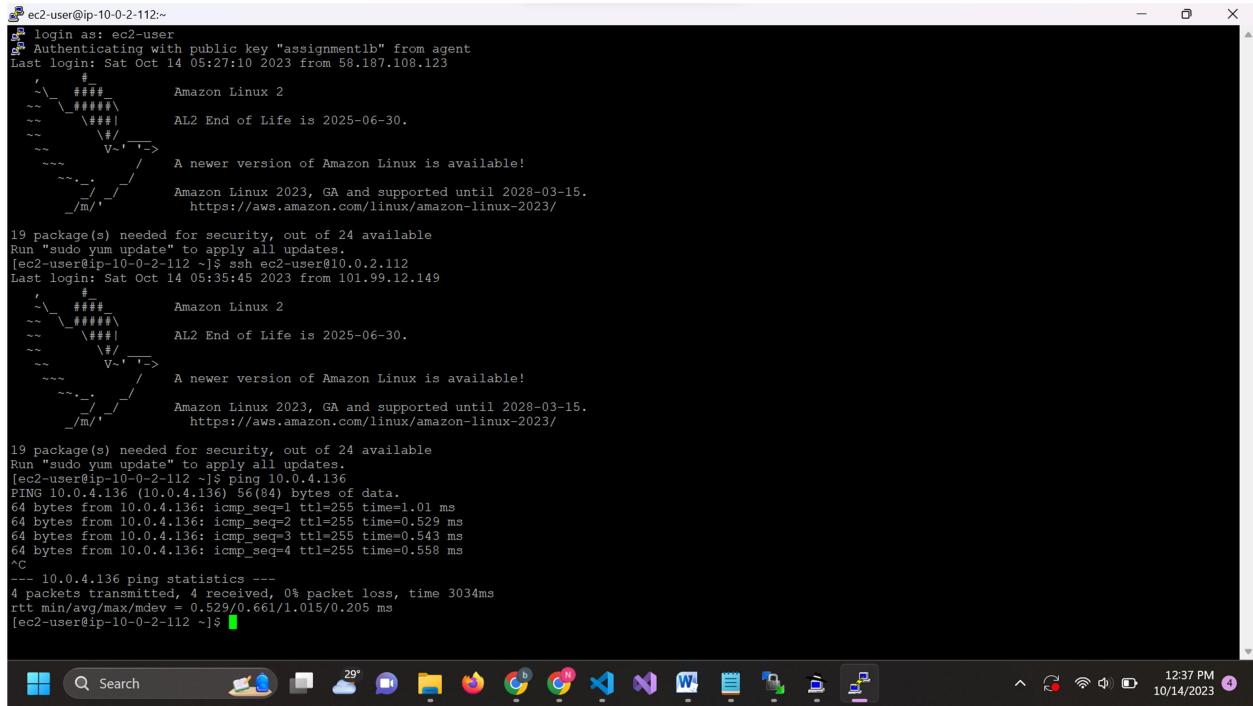


#### Uploaded photos:

Photo	Name	Description	Creation date	Keywords
	Jhin	Jhin waalpaper	2023-10-14	Jhin, darkstar,wallpaper



Step 10 : Ping Test from Bastion (login using Bastion public elastic IP, ssh Bastion same Elastic IP and ping Testinstance using it privateIpv4)



The screenshot shows a terminal window titled "ec2-user@ip-10-0-2-112:~". The session starts with a login message and a welcome banner for Amazon Linux 2. It then displays a notice about an available update. The user runs a "ping" command to test connectivity to another instance at 10.0.4.136. The ping statistics show 4 packets transmitted with 0% loss and a round-trip time of 3034ms. The terminal window has a dark background with light-colored text. The bottom of the screen shows the Windows taskbar with various pinned icons.

```
ec2-user@ip-10-0-2-112:~  
Authenticating with public key "assignmentelb" from agent  
Last login: Sat Oct 14 05:27:10 2023 from 58.187.108.123  
      #  
      #####  
      \###/  
      \##| AL2 End of Life is 2025-06-30.  
      \|/  
      V~'-->  
      A newer version of Amazon Linux is available!  
      .--.  
      /--/  
      Amazon Linux 2023, GA and supported until 2028-03-15.  
      https://aws.amazon.com/linux/amazon-linux-2023/  
  
19 package(s) needed for security, out of 24 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-10-0-2-112 ~]$ ssh ec2-user@10.0.2.112  
Last login: Sat Oct 14 05:35:45 2023 from 101.99.12.149  
      #  
      #####  
      \###/  
      \##| AL2 End of Life is 2025-06-30.  
      \|/  
      V~'-->  
      A newer version of Amazon Linux is available!  
      .--.  
      /--/  
      Amazon Linux 2023, GA and supported until 2028-03-15.  
      https://aws.amazon.com/linux/amazon-linux-2023/  
  
19 package(s) needed for security, out of 24 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-10-0-2-112 ~]$ ping 10.0.4.136  
PING 10.0.4.136 (10.0.4.136) 56(84) bytes of data.  
64 bytes from 10.0.4.136: icmp_seq=1 ttl=255 time=1.01 ms  
64 bytes from 10.0.4.136: icmp_seq=2 ttl=255 time=0.529 ms  
64 bytes from 10.0.4.136: icmp_seq=3 ttl=255 time=0.543 ms  
64 bytes from 10.0.4.136: icmp_seq=4 ttl=255 time=0.558 ms  
^C  
--- 10.0.4.136 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3034ms  
rtt min/avg/max/mdev = 0.529/0.661/1.015/0.205 ms  
[ec2-user@ip-10-0-2-112 ~]$
```