Swinburne University of Technology

COS20019 Cloud Computing Architecture

Assignment 1b

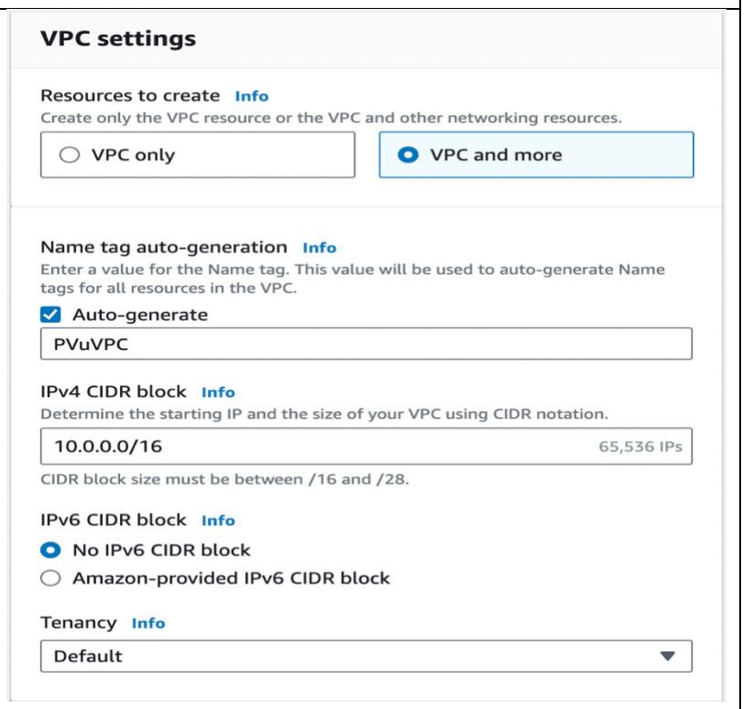
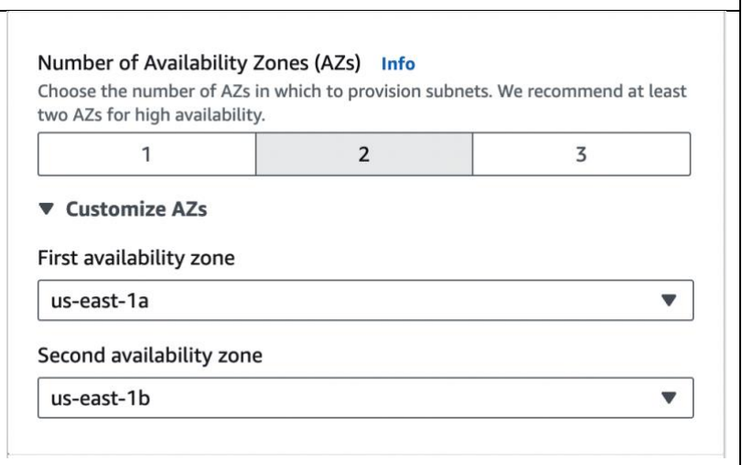
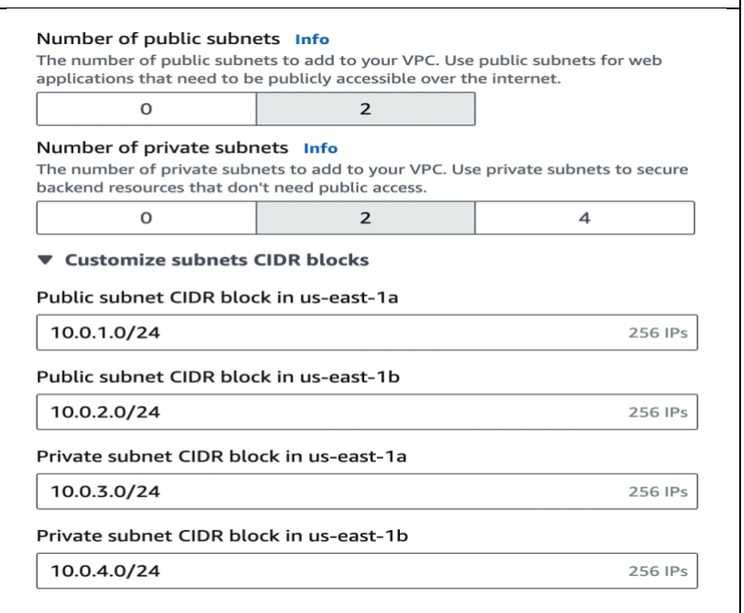
Saturday 14th October, 2023

Phan Vu
Student ID: 104222099

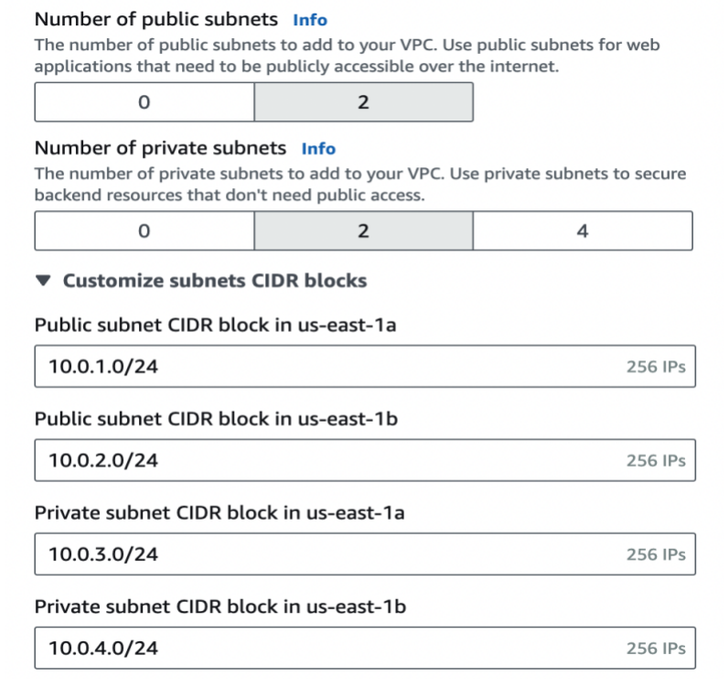
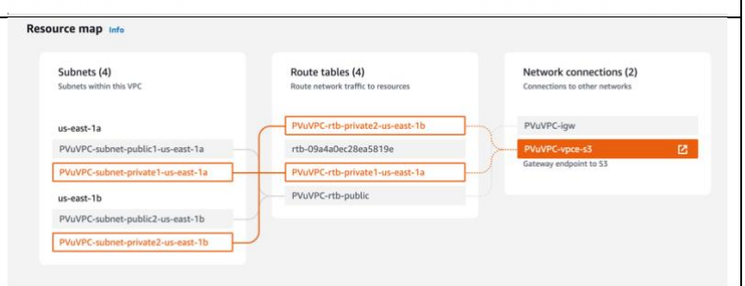
Infrastructure Requirements

URL: http://ec2-52-4-156-92.compute-1.amazonaws.com/COS20019/photoalbum/album.php

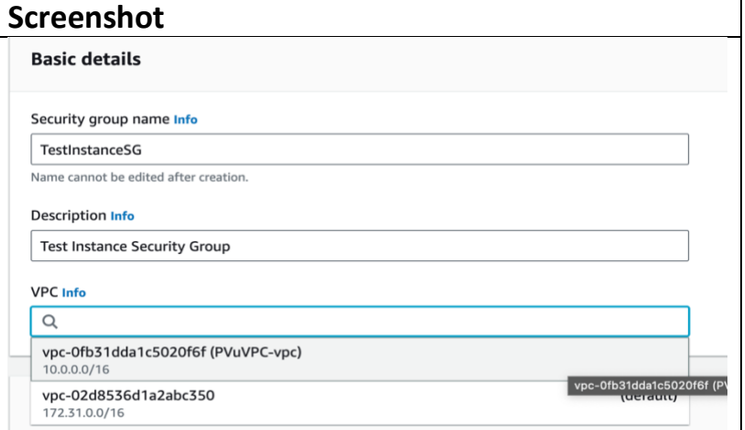
Marking scheme: VPC with 2 public and 2 private subnets

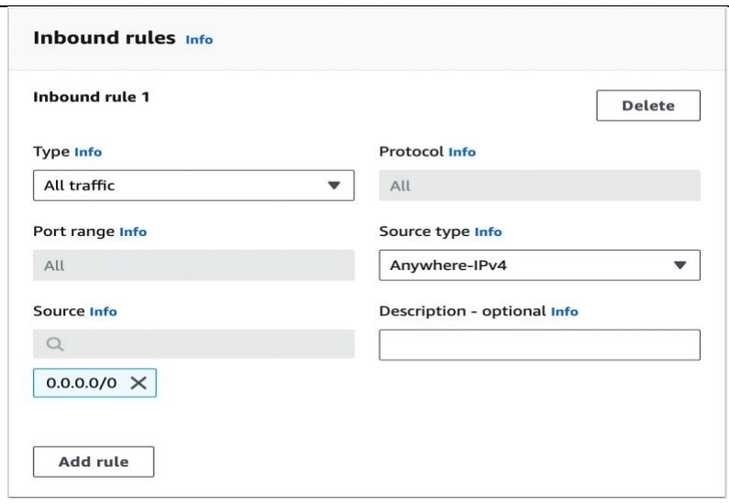
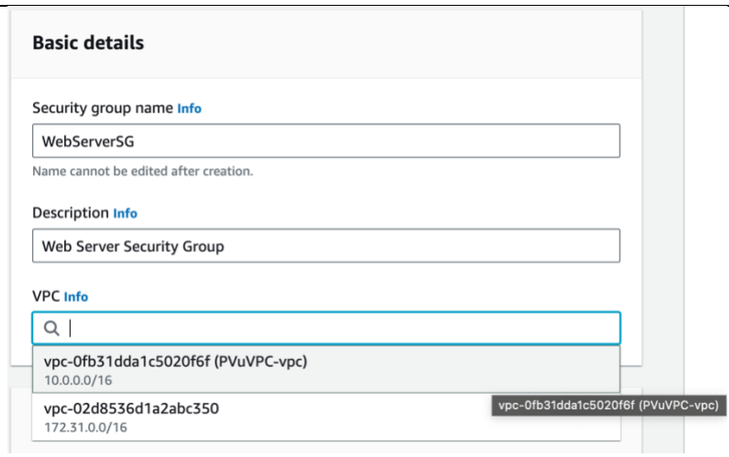
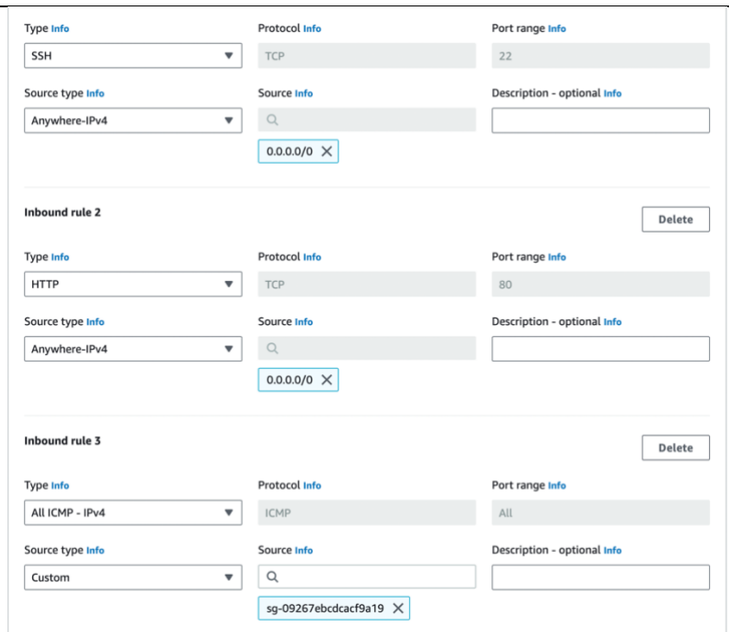
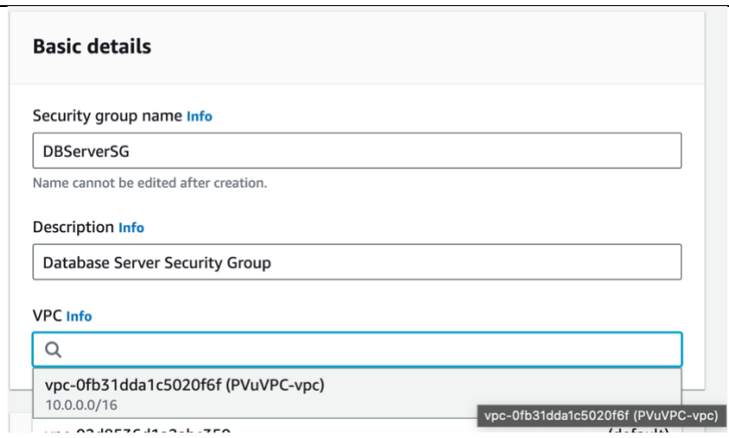
Step	Description	Screenshot
1	Create a VPC named **PVuVPC** in the **us-east-1** region with **10.0.0.0/16** IPv4 CIDR block.	
2	**Number of AZs** set to 2 with **us-east-1a** as first AZ and **us-east-1b** as AZ B.	
3	Both the number of public subnets and private subnets are set to **2**. Public subnet CIDR block is set with the following configuration that aligns with the VPC architecture diagram provided inthe assignment description.	

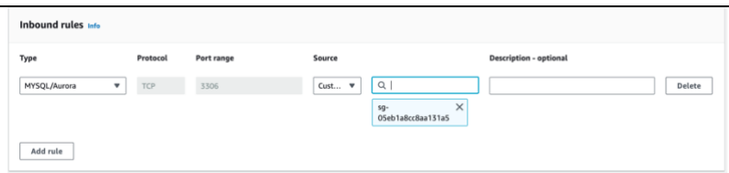
Marking scheme: Correct Public and Private Routing tables with correct subnet associations

Step	Description	Screenshot
1	Correct subnet association.	
2	Correct routing table.	

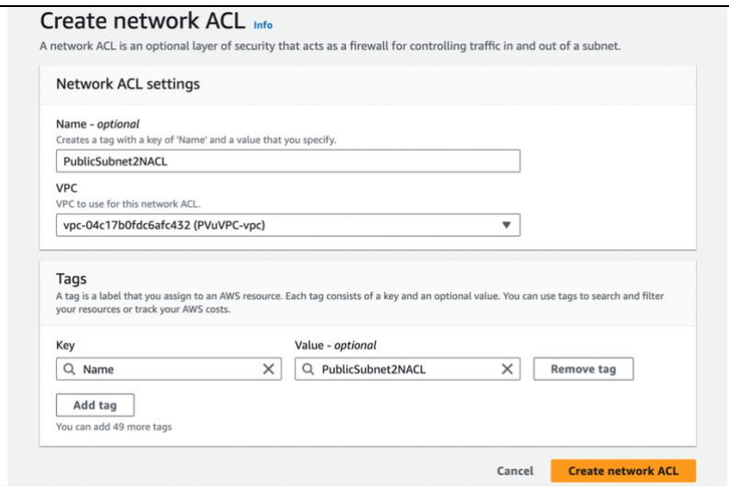
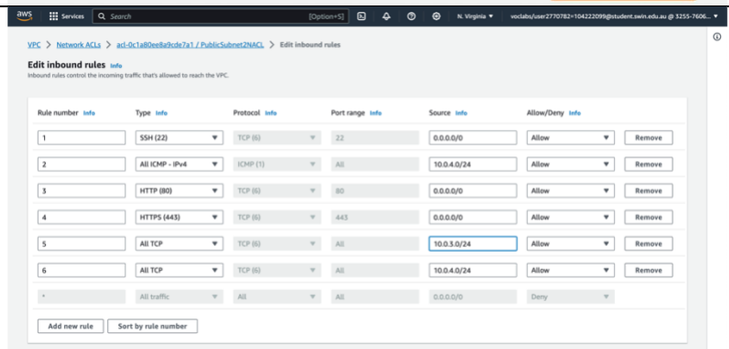
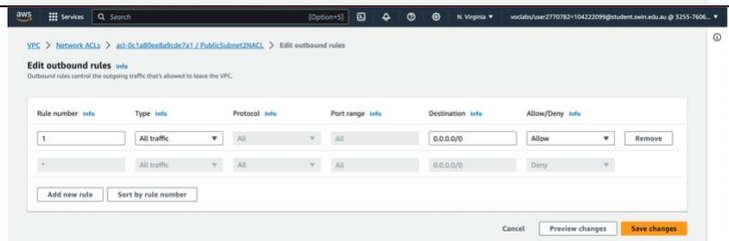
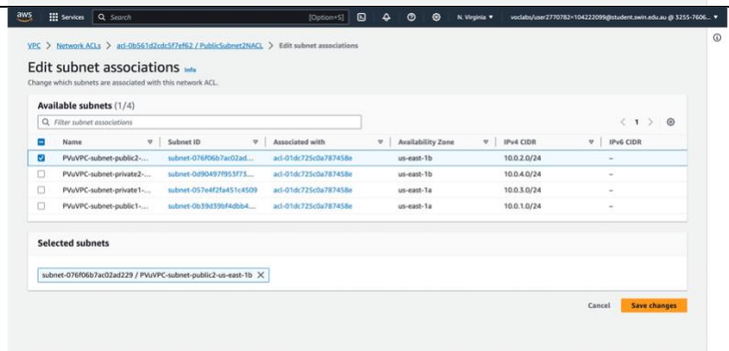
Marking scheme: Security groups properly configured and attached

Step	Description	Screenshot
1	Create a new security group named **TestInstanceSG**.	

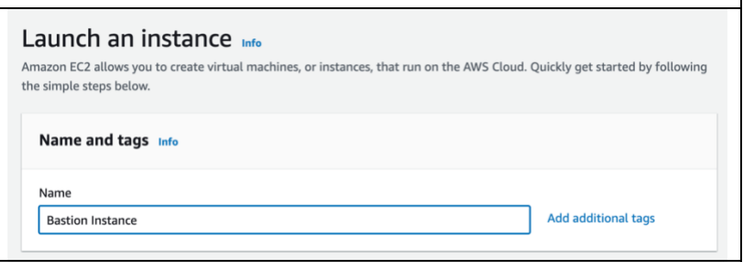
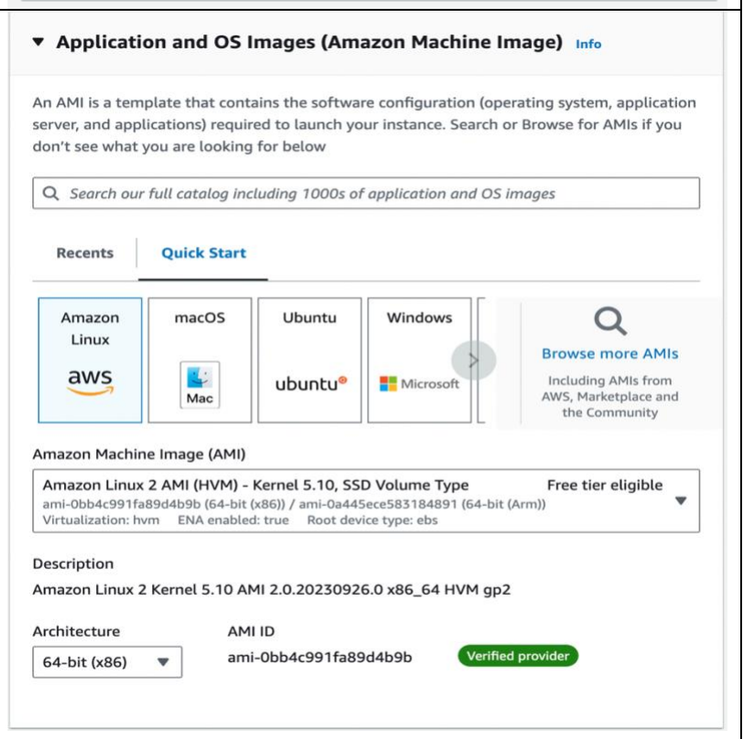
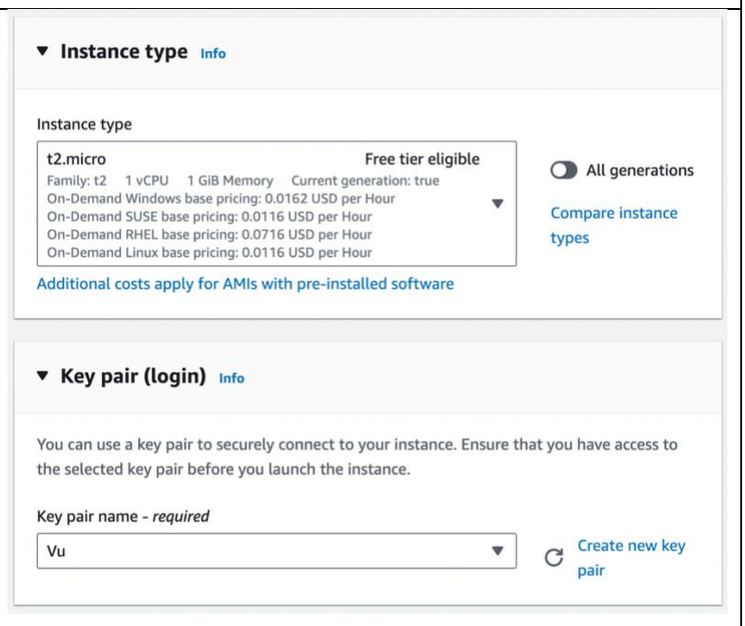
2	Inbound rule with type **All traffic** from **Anywhere**.	
3	Create a new security group named **WebServerSG.**	
4	Inbound rule with type **SSH (22)** and **HTTP (80)** from **Anywhere** and **All ICMP - IPv4** from security group **TestInstanceSG**.	
5	Create a new security group named **DBServerSG**.	

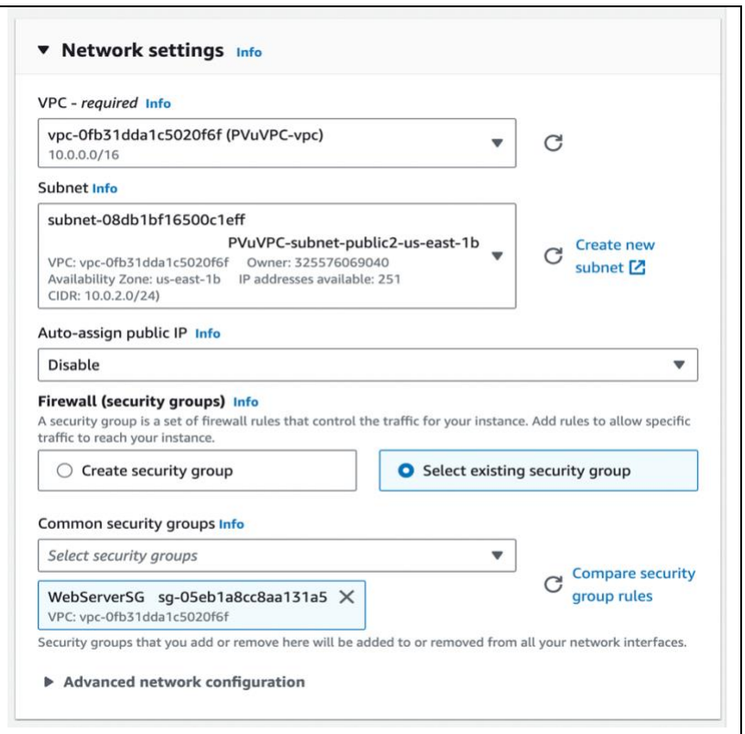
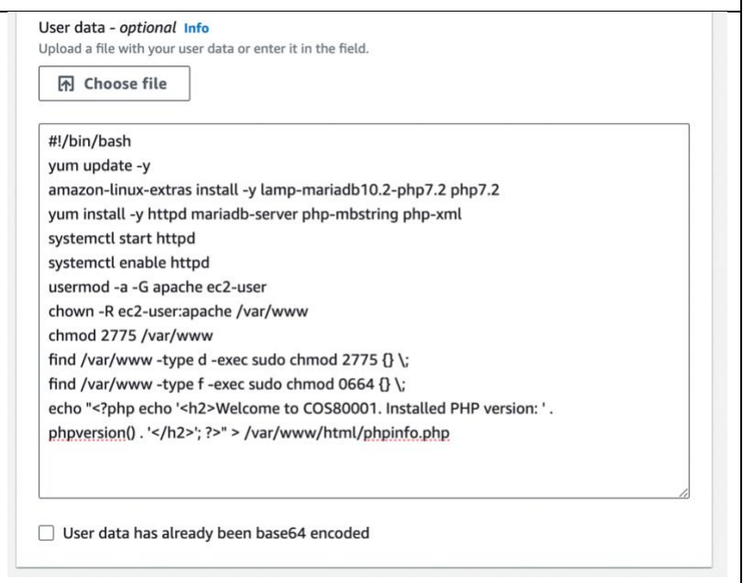
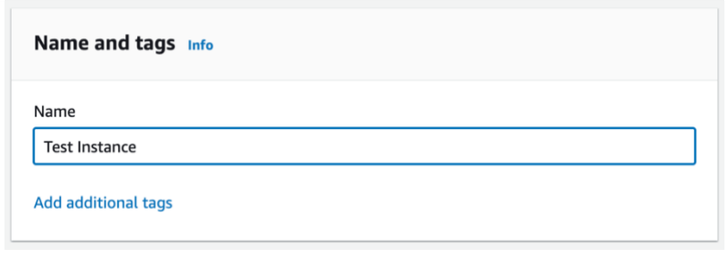
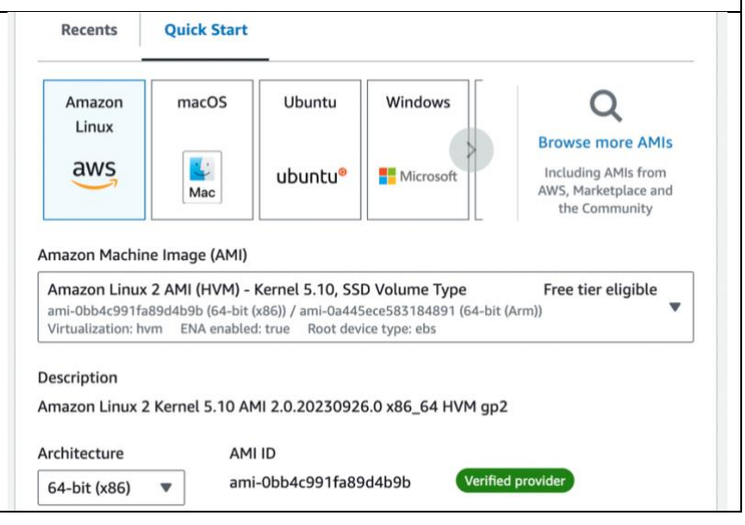
6	Inbound rule with **type MySQL (3306)** from security group **WebServerSG**.	

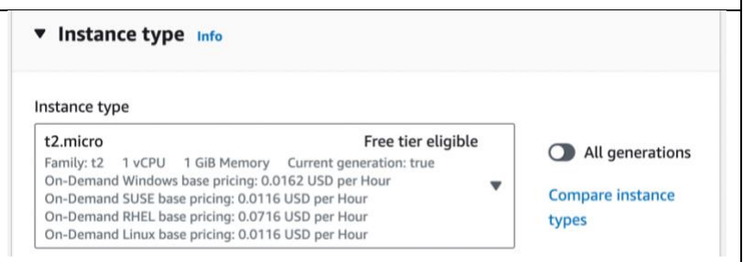
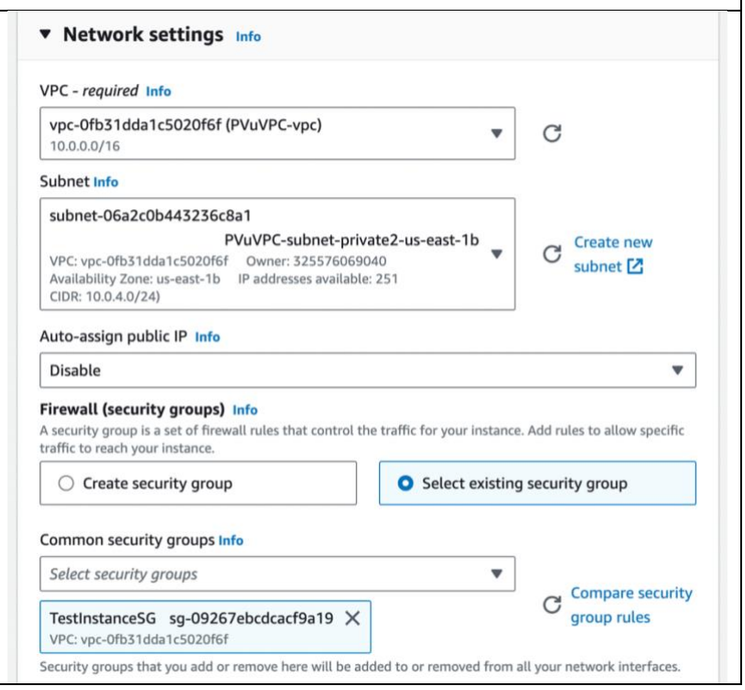
Marking scheme: Network ACL properly configured and attached

Step	Description	Screenshot
1	Create a new network ACL named **PublicSubnet2NACL**.	
2	Configuration of the inbound rules.	
3	Outbound rule set to **All traffic**.	
4	Associate the NACL to Public Subnet 2 (CIDR 10.0.2.0/24 and us-east-1b AZ).	
5	NACL successfully associated with the Public Subnet 2 (CIDR 10.0.2.0/24 and us-east-1b AZ).	

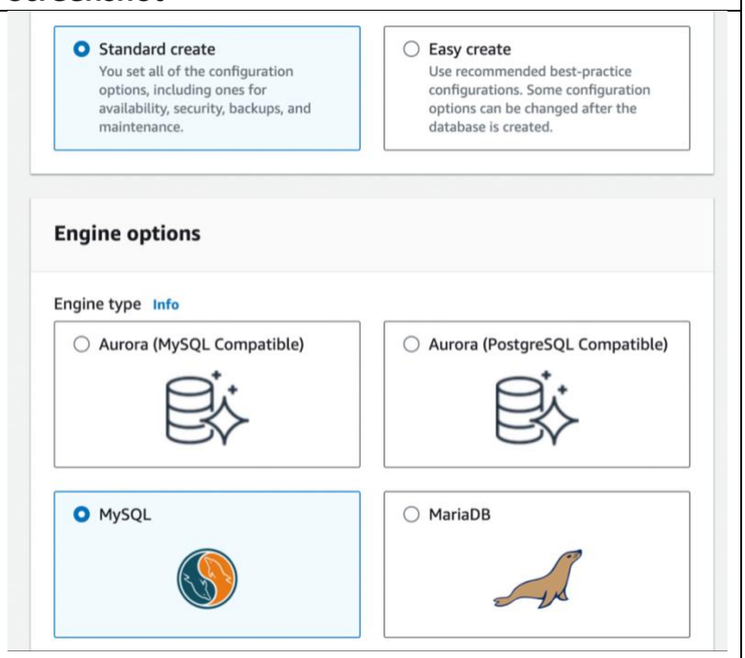
Marking scheme: Correct Web server and Test instances running in correct subnets

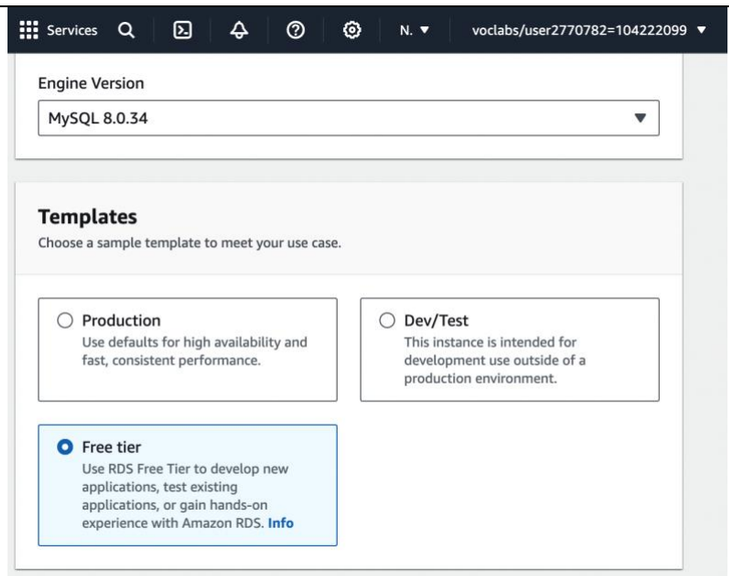
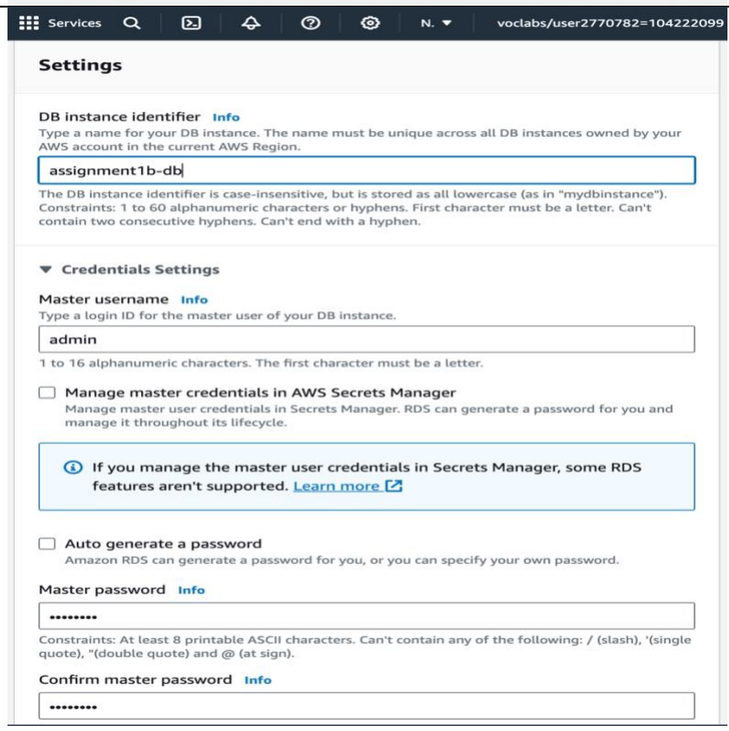
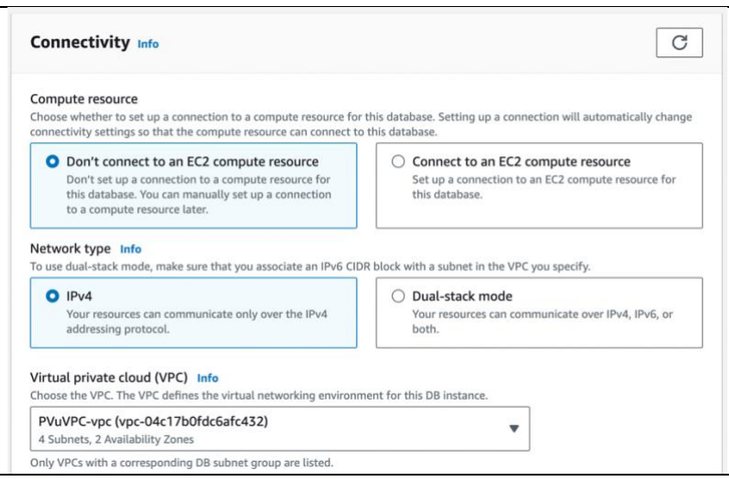
Step	Description	Screenshot
1	Create a web server instance named **Bastion Instance**.	
2	Use **Amazon Linux 2 AMI (HVM), SSD Volume Type** for OS image.	
3	Choose **t2.micro** for **Instance type.**	

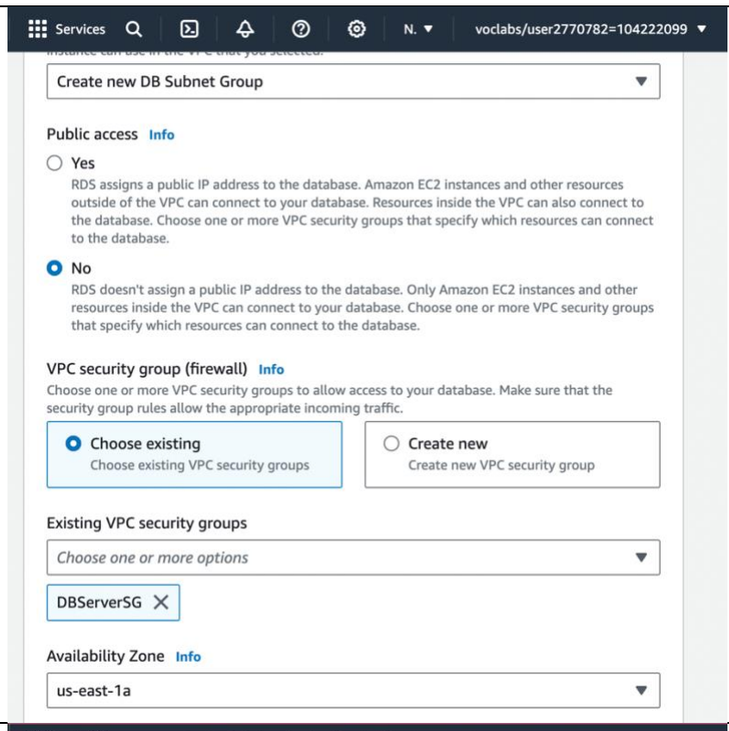
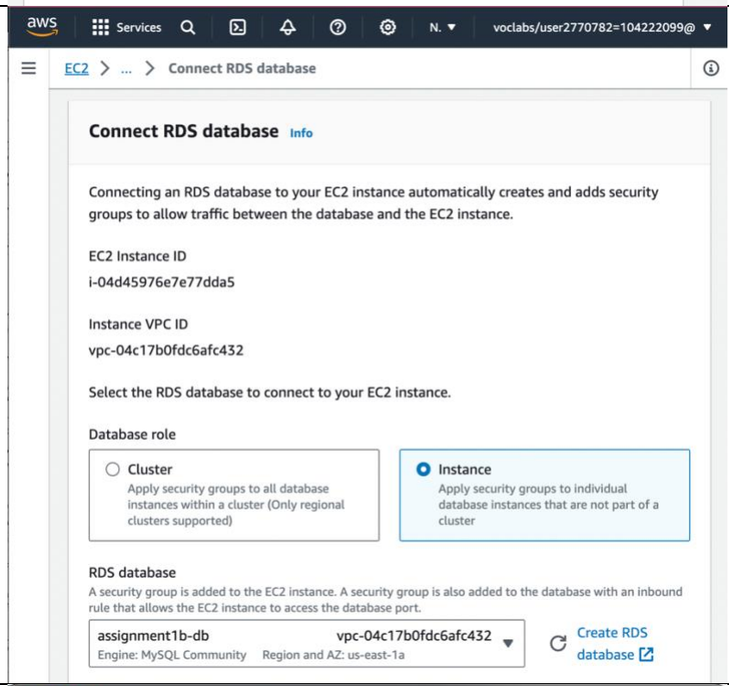
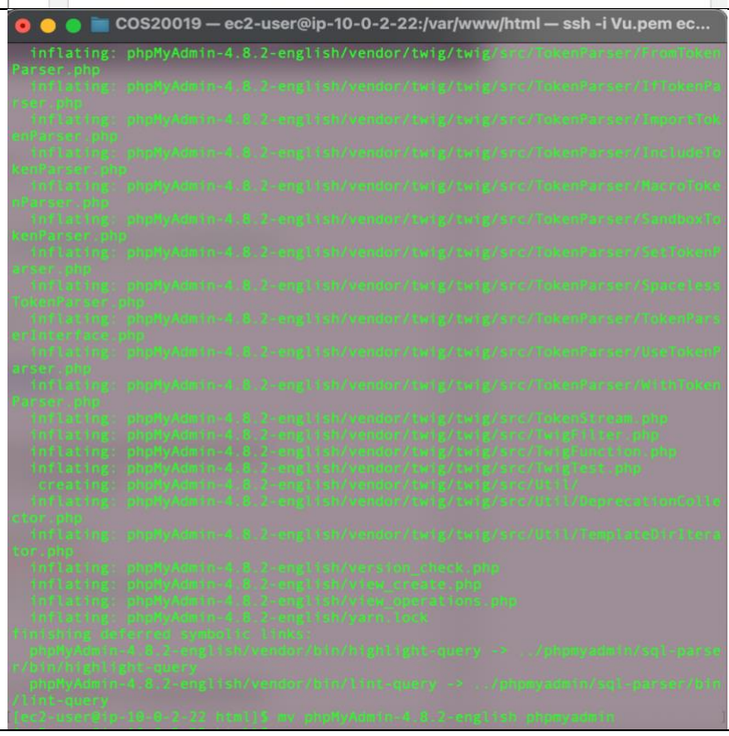
4	Choose **PVuVPC** as the VPC for the instance. Choose **PVuVPC-subnet-public2-us-east-1b (CIDR 10.0.2.0/24)** as subnet association. Use **WebServerSG** security group.	
5	Bash script to install Apache web server and other PHP packages to the **Bastion Instance.**	
6	Create a web server instance named **Test Instance**.	
7	Use **Amazon Linux 2 AMI (HVM), SSD Volume Type** for OS image.	

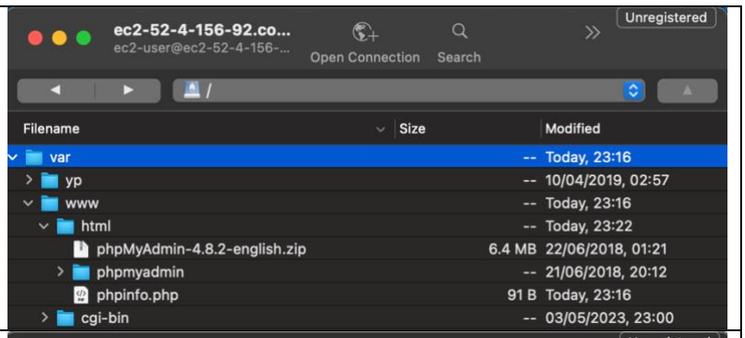
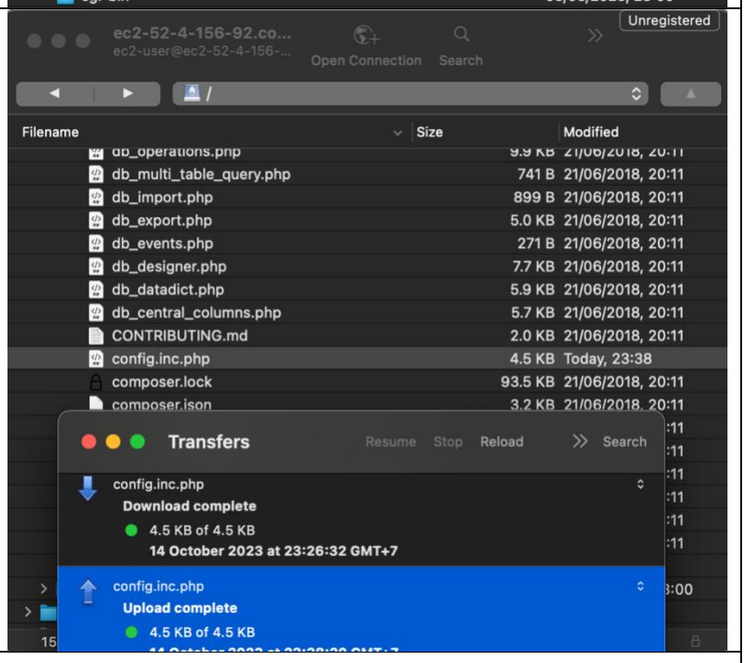
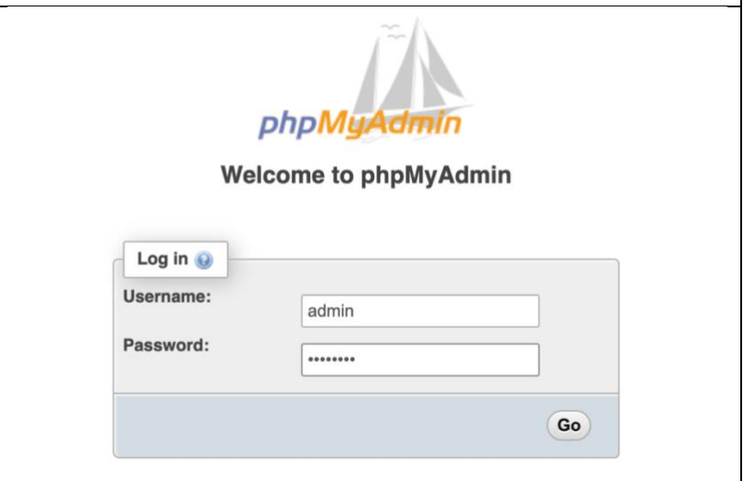
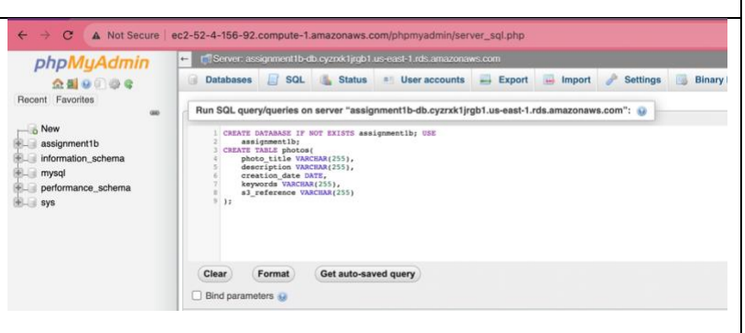
8	Choose **t2.micro** for **Instance type.**	
9	Choose **PVuVPC** as the VPC for the instance. Choose **PVuVPC-subnet-private2-us-east-1b (CIDR 10.0.4.0/24)** as subnet association. Use **TestInstanceSG** security group.	

Marking scheme: Database schema as specified

Step	Description	Screenshot
1	Create new RDS with the engine type option of **MySQL**.	

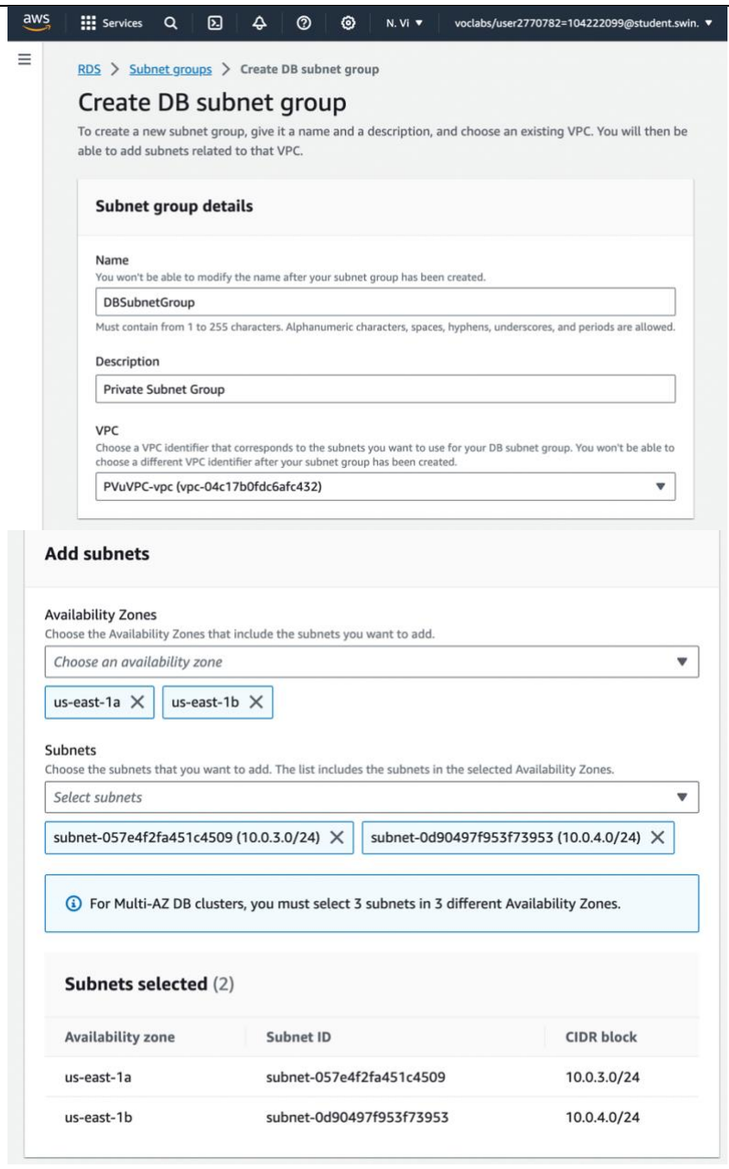
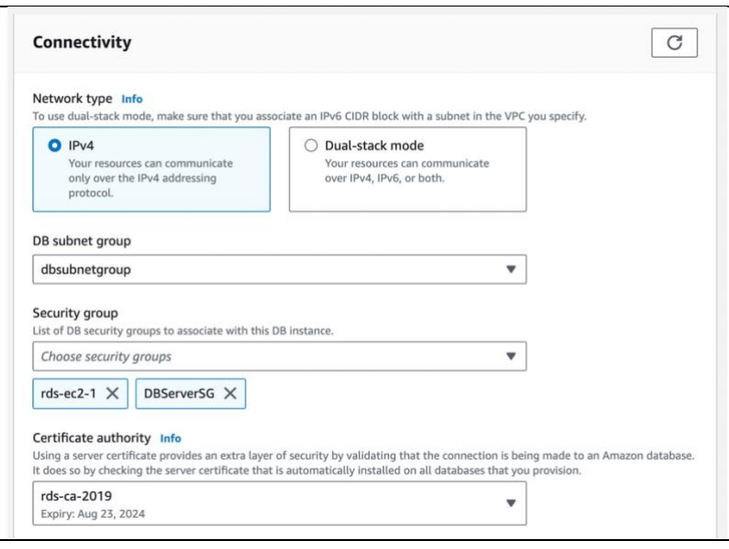
2	Choose **MySQL 8.0.34** as DB engine version. Use the **Free tier** template for the RDS.	
3	Assign the DB instance identifier with the name **assignment1b-db**. Modify credentials settings for master account.	
4	In this case, opt to not connect to an EC2 immediately. Choose **PVuVPC** for the DB instance.	

5	**Public access** set to **No**. Use **DBServerSG** for **VPC security group**. AZ set to **us-east-1a** (AZ 1 according to the provided diagram).	
6	Connect EC2 **Bastion Instance** to the created RDS instance.	
7	Installing PHP to the EC2 instance.	

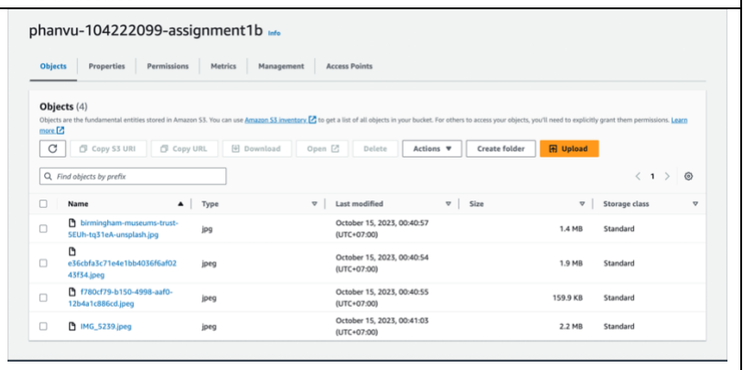
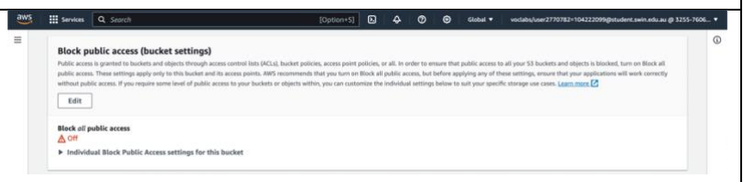
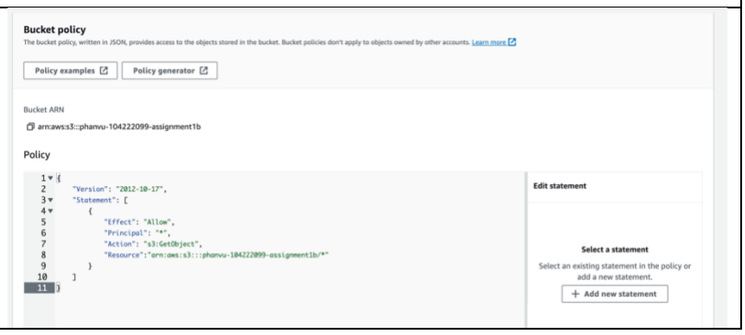
8	phpMyAdmin successfully installed in the **Bastion Instance.**	
9	Rename **config.sample.inc.php** to **config.inc.php**.	
10	Replace **'localhost'** in **$cfg['Servers'][$i]['host'] = 'localhost';** with: **'assignment1b-db.cyzrxk1jrgb1.us-east-1.rds.amazonaws.com'**	
11	Login to phpMyAdmin using created master credentials in the RDS creation step.	
12	Create the database as well as a new table in phpMyadmin using the following SQL commands.	

13	Database and table successfully created.	

Marking scheme: Database running in correct subnets

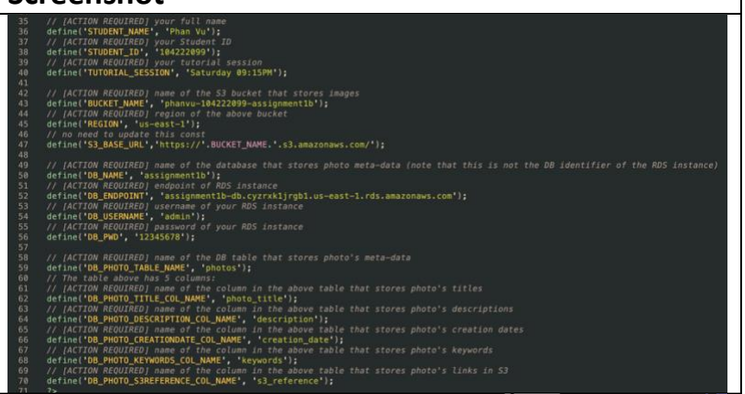
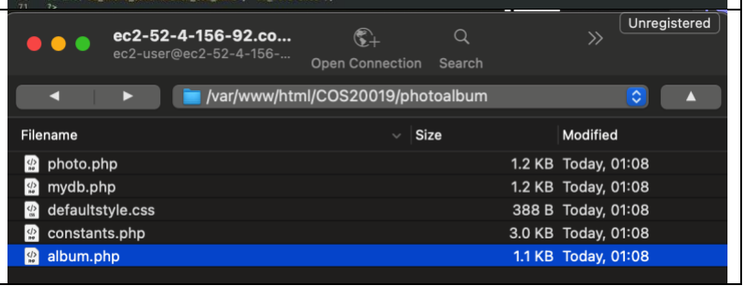
Step	Description	Screenshot
1	Create a new subnet group named **DBSubnetGroup** in **PVuVPC.** Configuration: - **AZ**: Choose **us-east-1a** and **us-east-1b**. - Subnets: Choose private subnets (CIDR 10.0.3.0/24 and 10.0.4.0/24).	
2	Connect the RDS instance with the created subnet group.	

Marking scheme: S3 objects publicly accessible, using proper access policy

Step	Description	Screenshot
1	Create a new bucket named luutuanhoang-104180391-assignment1b Manually add photos into the bucket.	
2	Block all public access set to Off.	
3	The following policy is used to enable public access to all available objects in this S3 bucket.	

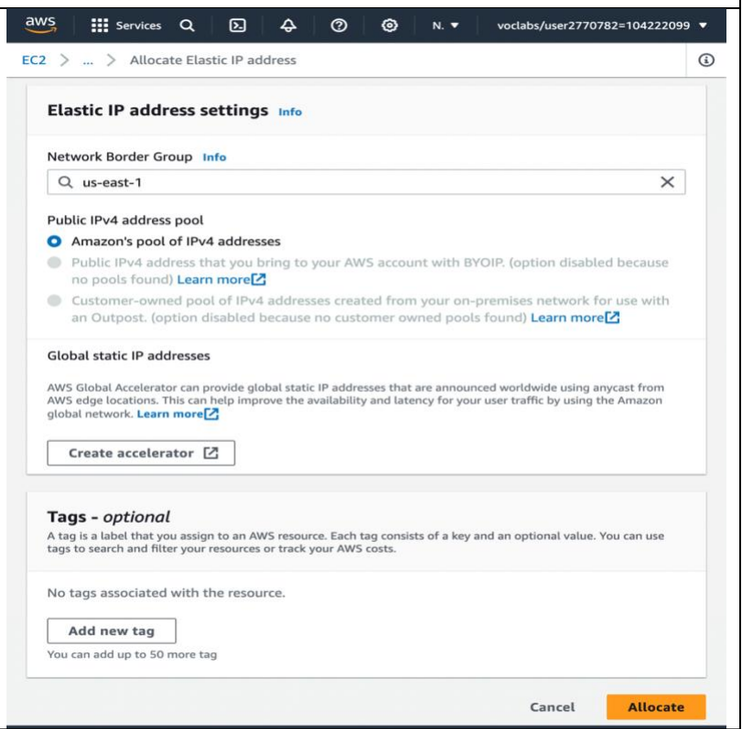
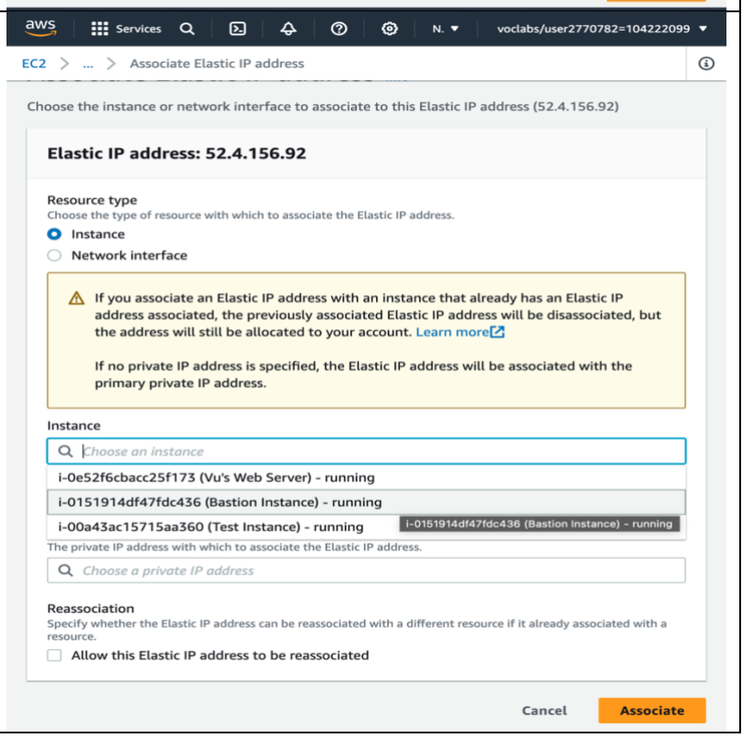
Functional Requirements

Marking scheme: album.php page displayed from EC2 Web server

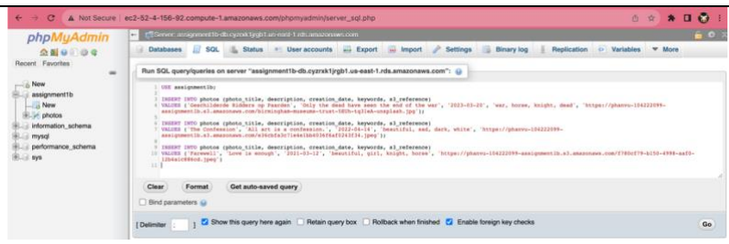
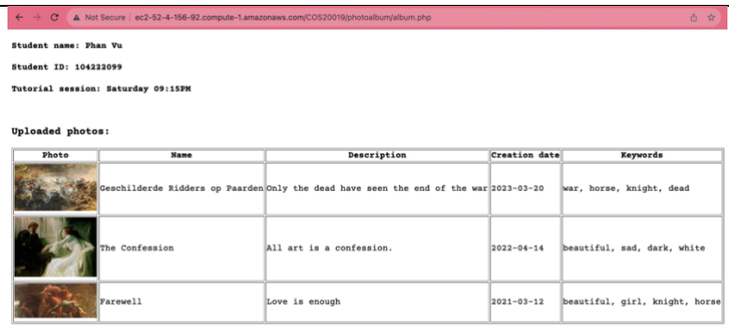
Step	Description	Screenshot
1	Modify constants in **constants.php**.	
2	Transfer website source code to **Bastion Instance**. **Directory:** var/www/html/ COS20019/photoalbum/	

3	**album.php** page successfully displayed from EC2 Web server (**Bastion Instance**).	

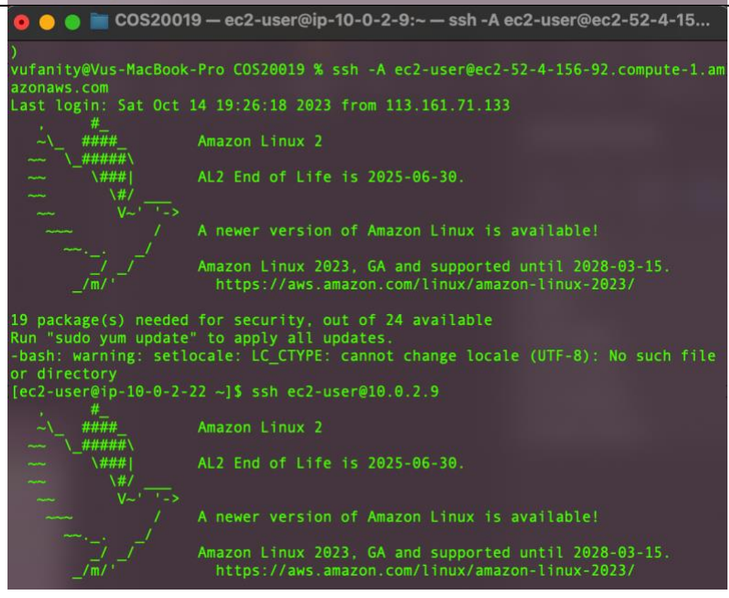
Marking scheme: Provided URL is persistent (Elastic IP Association)

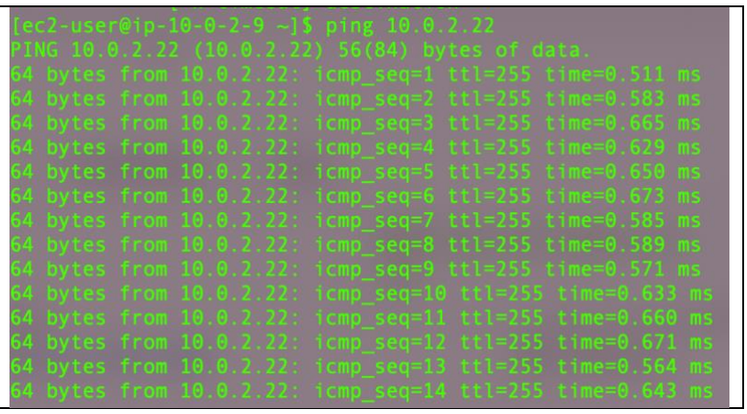
Step	Description	Screenshot
1	Allocate the new Elastic IP address from **us-east-1 Network Border Group.**	
2	Associated created Elastic IP address to the **Bastion Instance**. **Bastion Instance** associated with the Elastic IP address. Elastic IP address**: 52.4.156.92** Website URL: **ec2-52-4-156-92.compute1.amazonaws.com /cos20019/photoalbum/ album.php**	

Marking scheme: Photos loaded from S3 with matching metadata from RDS

Step	Description	Screenshot
1	Insert metadata into the database using SQL commands.	
2	Data records in the database.	
3	The website is able to list all the photos (stored in the S3 bucket) along with their meta-data (stored in the database).	

Marking scheme: Web server instance reachable from Test instance via ICMP

Step	Description	Screenshot
1	SSH to Bastion Instance.	
2	SSH from Bastion Instance to Test Instance.	

3	Successfully ping the Bastion Instance from the Test instance.	

P/s: Sorry for all of the inconveniences that I made. Thank you.

END.