

Assignment #2 VPC, EC2 and RDS

1

Assignment #2 VPC, EC2 and RDS

Darpankumar Jayantilal Patel,

8868275

Jaykumar Patel,

8916940

Keaton Lang,

8766111

INFO2350-25W-Sec1

Amarpreet Singh

2025-02-19

**Table of Contents**

<b>Table of Contents .....</b>	<b>2</b>
<b>Assignment #2 VPC, EC2 and RDS.....</b>	<b>3</b>
<b>Section 1: Practical Work .....</b>	<b>3</b>
<b>Task 1: Cloud Network Security (VPC Implementation) .....</b>	<b>3</b>
Screenshots.....	3
<b>Task 2: Create an EC2 Instance.....</b>	<b>6</b>
Screenshots.....	6
<b>Task 3: Implement a Customer Relationship Database in the Cloud.....</b>	<b>8</b>
Screenshots.....	8
<b>Section 2: Report Work.....</b>	<b>12</b>
<b>Part 1: Customer Relationship Database .....</b>	<b>12</b>
Describe the functionality of the database used to store customer details .....	12
Explain how it helps in business operations.....	12
<b>Part 2: Cloud Storage Comparison (S3 vs EFS).....</b>	<b>12</b>
When to use EFS.....	12
Comparison: When to use Amazon EFS vs. Amazon S3.....	12
Explanation of EFS Intelligent-tiering concepts .....	12
<b>Part 3: Network Security (VPC Implementation Notes).....</b>	<b>13</b>
Explain why VPC is essential for cloud security .....	13
Describe the IP addressing used in your setup .....	13
Justify security group settings for optimal access control .....	13

# Assignment #2 VPC, EC2 and RDS

3

## Assignment #2 VPC, EC2 and RDS

### Section 1: Practical Work

#### Task 1: Cloud Network Security (VPC Implementation)

##### Screenshots

→ Created a VPC (Virtual Private Cloud) in AWS with a suitable CIDR range.

The screenshot displays the AWS Management Console interface for a VPC. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/vpconsole/home?region=us-east-1#VpcDetails:Vpcid=vpc-07fad8e3df83b4305`. The page title is "vpc-07fad8e3df83b4305 / Assignment2\_Group10-vpc".

**VPC dashboard**

- EC2 Global View
- Filter by VPC
- Virtual private cloud**
  - Your VPCs
  - Subnets
  - Route tables
  - Internet gateways
  - Egress-only internet gateways
  - Carrier gateways
  - DHCP option sets
  - Elastic IPs
  - Managed prefix lists
  - NAT gateways
  - Peering connections
- Security

**Details**

<b>VPC ID</b> vpc-07fad8e3df83b4305	<b>State</b> Available	<b>Block Public Access</b> Off	<b>DNS hostnames</b> Enabled
<b>DNS resolution</b> Enabled	<b>Tenancy</b> default	<b>DHCP option set</b> dopt-004b30baf0b09be8d	<b>Main route table</b> rtb-0719bbd8d9367b5d0
<b>Main network ACL</b> acl-08532ee5efb20828a	<b>Default VPC</b> No	<b>IPv4 CIDR</b> 10.10.10.0/24	<b>IPv6 pool</b> -
<b>IPv6 CIDR (Network border group)</b> -	<b>Network Address Usage metrics</b> Disabled	<b>Route 53 Resolver DNS Firewall rule groups</b> Failed to load rule groups	<b>Owner ID</b> 746642308337

**Resource map** | CIDRs | Flow logs | Tags | Integrations

**Resource map**

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

9:41 PM 2025-02-18

# Assignment #2 VPC, EC2 and RDS

4

The screenshot shows the AWS VPC console interface. The main content area displays the details for a VPC named 'Assignment2\_Group10-vpc' with ID 'vpc-07fad8e3df83b4305'. The VPC is in an 'Available' state. Key details include: DNS resolution is enabled, Main network ACL is 'acl-08532ee5efb20828a', IPv4 CIDR is '10.10.10.0/24', and Block Public Access is turned off. The console also shows a navigation menu on the left with options like 'Subnets', 'Route tables', and 'Internet gateways'. A top navigation bar includes the AWS logo, search, and user information.

→ Added two subnets and named them subnet#01 and Subnet#02.

The screenshot shows the AWS VPC console interface displaying a list of subnets. A green notification banner at the top states 'You have successfully deleted subnet-0ace86a8e2294566a'. The subnets list includes 'Subnet#01' (ID: subnet-05bed0428d0d8760d) and 'Subnet#02' (ID: subnet-09d1122ee670db0da), both in an 'Available' state. The details for Subnet#01 are expanded, showing its ARN, IPv4 CIDR of '10.10.10.0/28', Availability Zone of 'us-east-1a', and associated route table 'rtb-051a85d6a23f24568'. The console also shows a navigation menu on the left and a top navigation bar with the AWS logo and search.

# Assignment #2 VPC, EC2 and RDS

5

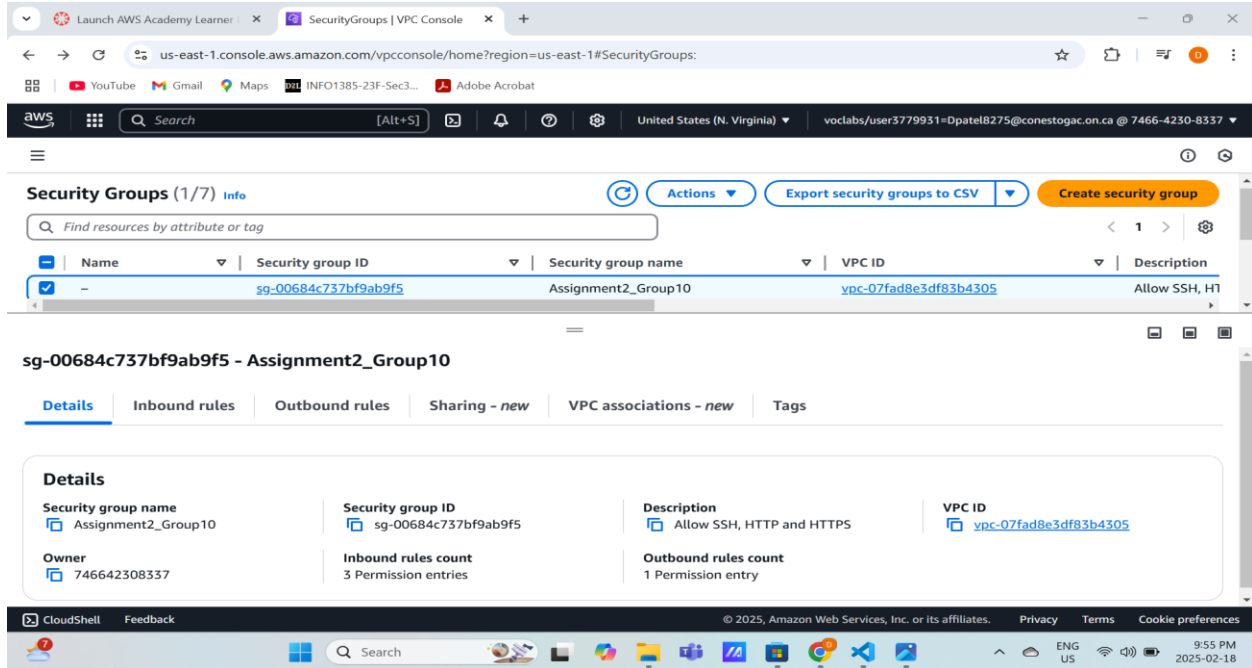
The screenshot shows the AWS VPC console interface. At the top, a green notification bar states: "You have successfully deleted subnet-0ace86a8e2294566a". Below this, the "Subnets (1/11)" section is active, displaying a table of subnets. The table has columns for Name, Subnet ID, State, and VPC. Two subnets are listed: Subnet#01 (subnet-05bed0428d0d8760d) and Subnet#02 (subnet-09d1122ee670db0da), both in an "Available" state. Below the table, the details for Subnet#02 are expanded, showing fields such as Subnet ID, Subnet ARN, State, IPv4 CIDR (10.10.10.16/28), Availability Zone (us-east-1b), Route table (rtb-0719bbd8d9367b5d0), Network ACL (acl-08532ee5efb20828a), and Block Public Access (Off).

→ Created a security group allowing only SSH and HTTP/HTTPS traffic.

The screenshot shows the AWS Security Groups console. The breadcrumb navigation indicates the path: VPC > Security Groups > sg-00684c737bf9ab9f5 - Assignment2\_Group10. The "Details" section for the security group "Assignment2\_Group10" is displayed, showing its ID (sg-00684c737bf9ab9f5), description ("Allow SSH, HTTP and HTTPS"), and VPC ID (vpc-07fad8e3df83b4305). Below the details, the "Inbound rules (3)" section is active, showing a table of inbound rules. The table has columns for Name, Security group rule ID, IP version, Type, Protocol, and Port range. Three rules are listed: one for SSH (TCP, port 22), one for HTTPS (TCP, port 443), and one for HTTP (TCP, port 80).

# Assignment #2 VPC, EC2 and RDS

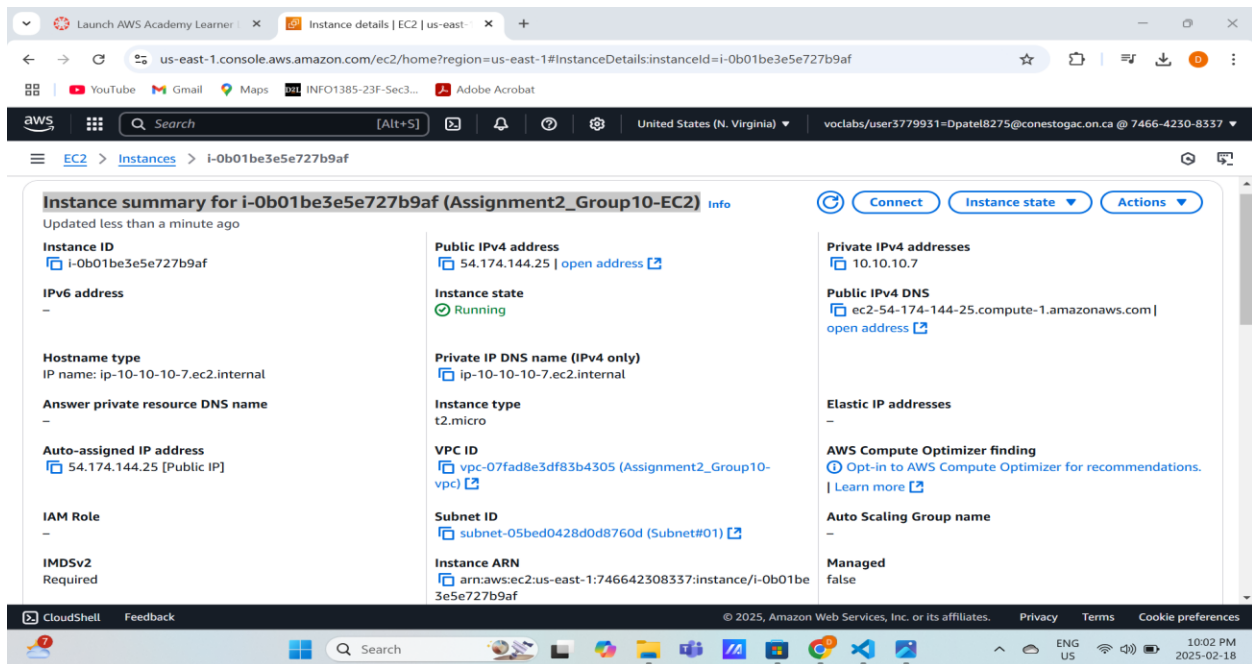
6



## Task 2: Create an EC2 Instance

### Screenshots

→ Launched an EC2 instance inside the VPC created in Task 1.



→ Connected to the instance via SSH.



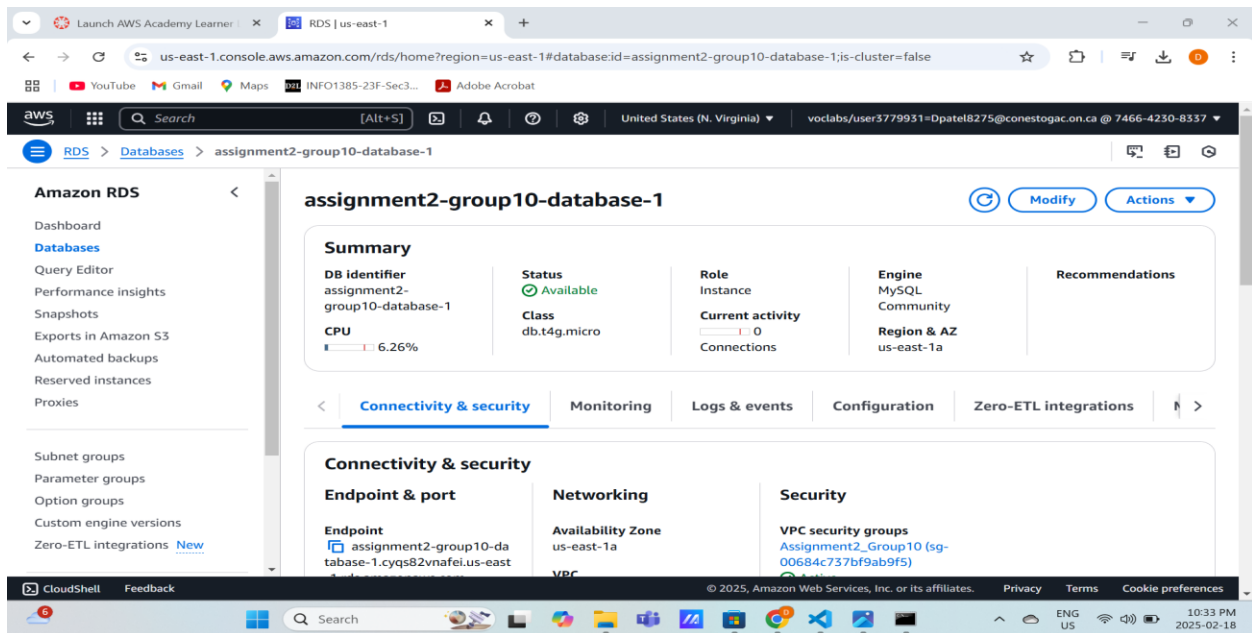
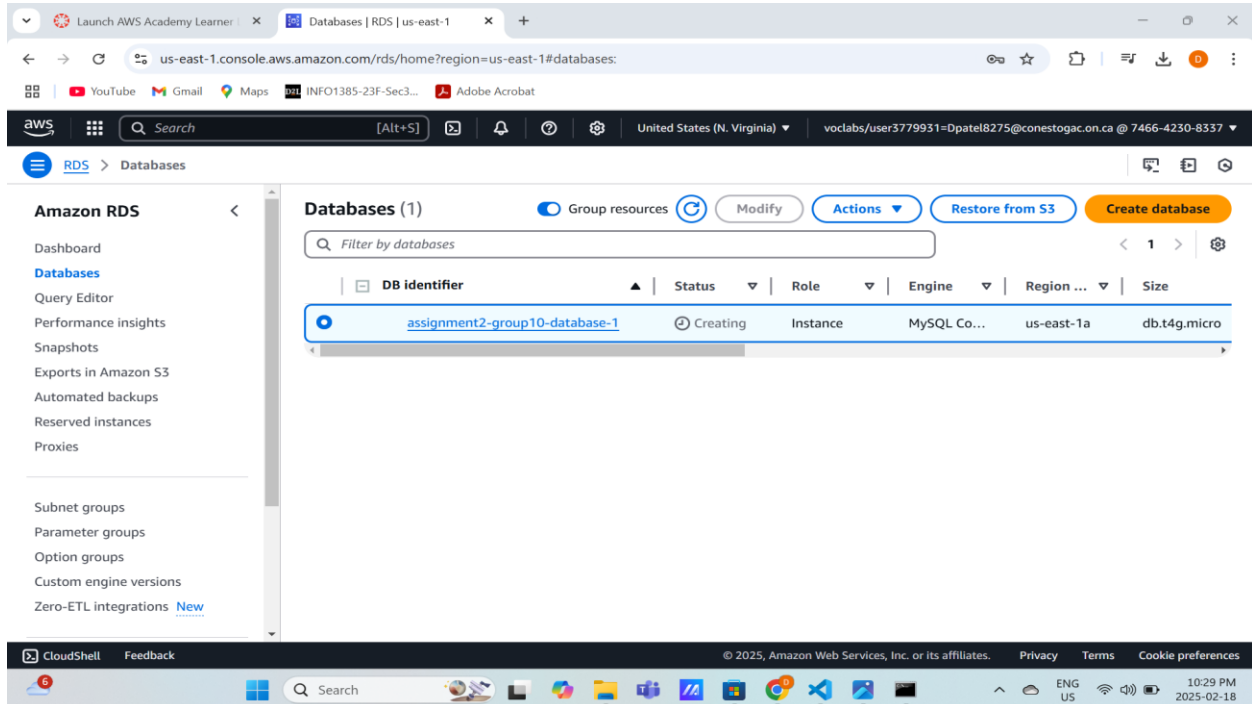
# Assignment #2 VPC, EC2 and RDS

8

## Task 3: Implement a Customer Relationship Database in the Cloud

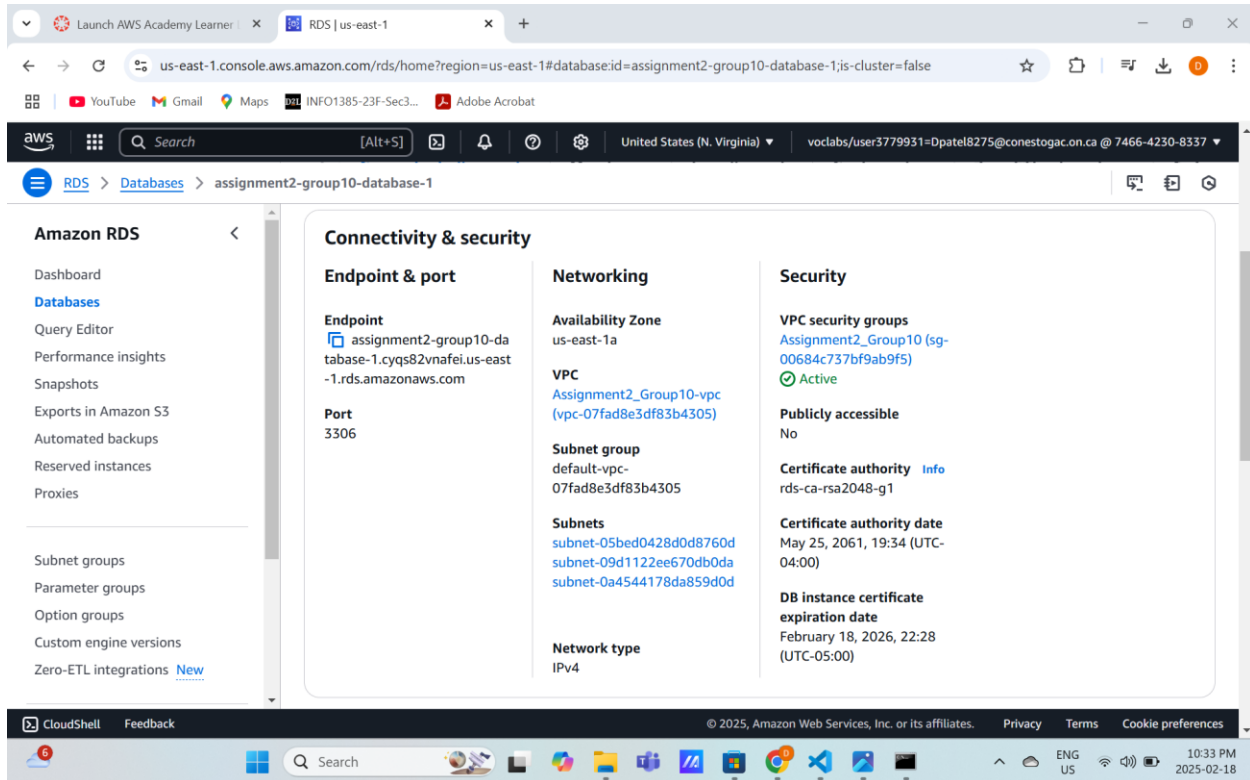
### Screenshots

→ Created a Relationship Database using AWS RDS – MySQL

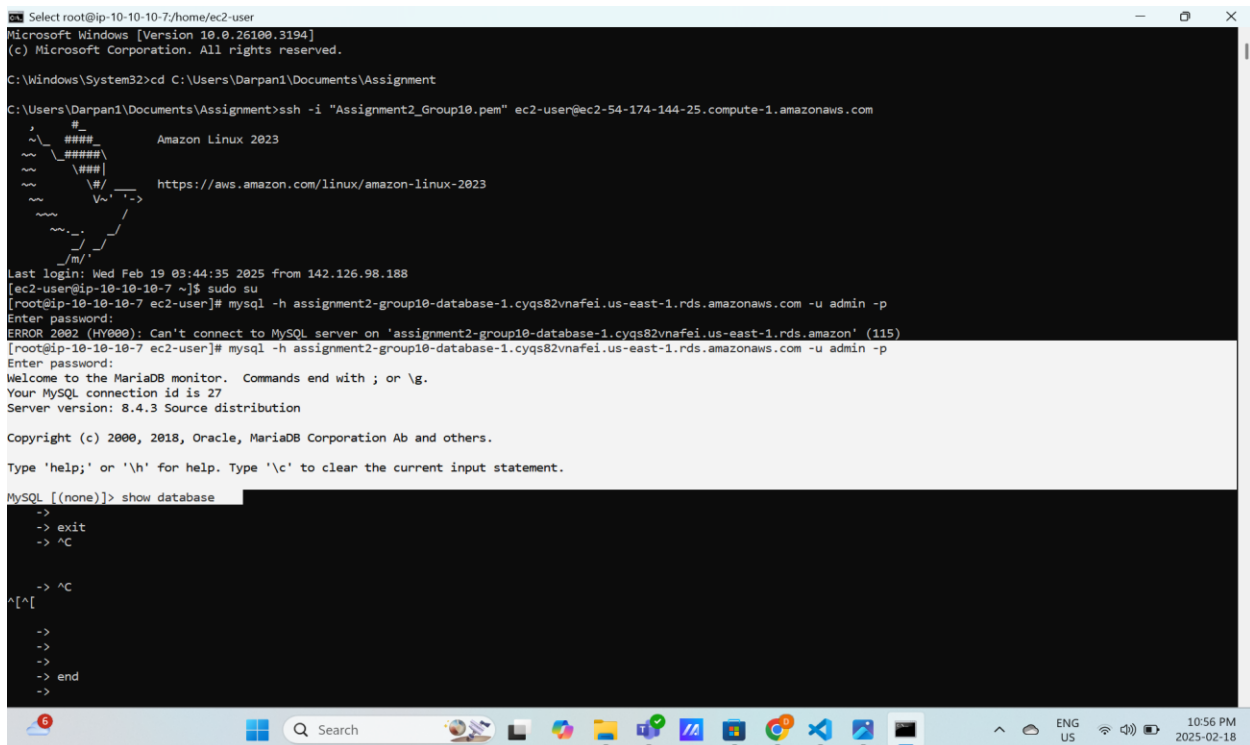


# Assignment #2 VPC, EC2 and RDS

9



→ Connected to the MySQL.



→ screenshots of the database setup and Implementation of two tables:

## Assignment #2 VPC, EC2 and RDS

10

```
Select root@ip-10-10-10-7/home/ec2-user
MySQL [(none)]> CREATE TABLE Orders (
  -> OrderID INT PRIMARY KEY,
  -> CustomerID INT,
  -> OrderDate DATE,
  -> Amount DECIMAL(10,2),
  -> FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
  -> );
ERROR 1046 (3D000): No database selected
MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.015 sec)

MySQL [(none)]> create database Group10_db
  ->
  -> /c;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '/c' at line 3
MySQL [(none)]> create database Group10_db;
Query OK, 1 row affected (0.023 sec)

MySQL [(none)]> use Group10_db;
Database changed
MySQL [Group10_db]> CREATE TABLE Customers (
  -> CustomerID INT AUTO_INCREMENT PRIMARY KEY,
  -> Name VARCHAR(100),
  -> Email VARCHAR(100) UNIQUE,
  -> Phone VARCHAR(20),
  -> Address TEXT
  -> );
MySQL [Group10_db]> CREATE TABLE Orders (
  -> OrderID INT AUTO_INCREMENT PRIMARY KEY,
  -> CustomerID INT,
  -> OrderDate DATE,
  -> Amount DECIMAL(10,2),
  -> FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
  -> );
Query OK, 0 rows affected (0.073 sec)

MySQL [Group10_db]>
```

→ Inserted a minimum of 5 records into each table.

```
Select root@ip-10-10-10-7/home/ec2-user
MySQL [Group10_db]> INSERT INTO Customers (Name, Email, Phone, Address) VALUES
  -> ('John Doe', 'john@example.com', '123-456-7890', '123 Elm Street'),
  -> ('Jane Smith', 'jane@example.com', '987-654-3210', '456 Maple Avenue'),
  -> ('Alice Brown', 'alice@example.com', '555-555-5555', '789 Oak Lane'),
  -> ('Bob White', 'bob@example.com', '666-666-6666', '159 Pine Road'),
  -> ('Charlie Black', 'charlie@example.com', '777-777-7777', '753 Birch Street');
CustomerID, OrderDate, Amount) VALUES
(1, '2025-02-15', 199.99)
Records: 5 Duplicates: 0 Warnings: 0

MySQL [Group10_db]>
MySQL [Group10_db]> INSERT INTO Orders (CustomerID, OrderDate, Amount) VALUES
  -> (1, '2025-02-15', 199.99),
  -> (2, '2025-02-16', 299.49),
  -> (3, '2025-02-17', 89.99),
  -> (4, '2025-02-18', 149.99),
  -> (5, '2025-02-19', 349.00);
Query OK, 5 rows affected (0.007 sec)
Records: 5 Duplicates: 0 Warnings: 0

MySQL [Group10_db]> SELECT * FROM Customers;
+-----+-----+-----+-----+-----+
| CustomerID | Name | Email | Phone | Address |
+-----+-----+-----+-----+-----+
| 1 | John Doe | john@example.com | 123-456-7890 | 123 Elm Street |
| 2 | Jane Smith | jane@example.com | 987-654-3210 | 456 Maple Avenue |
| 3 | Alice Brown | alice@example.com | 555-555-5555 | 789 Oak Lane |
| 4 | Bob White | bob@example.com | 666-666-6666 | 159 Pine Road |
| 5 | Charlie Black | charlie@example.com | 777-777-7777 | 753 Birch Street |
+-----+-----+-----+-----+-----+
5 rows in set (0.002 sec)

MySQL [Group10_db]>
```

→ Display all records from both tables.

# Assignment #2 VPC, EC2 and RDS

11

```
Select root@ip-10-10-10-7/home/ec2-user

MySQL [Group10_db]> INSERT INTO Customers (Name, Email, Phone, Address) VALUES
-> ('John Doe', 'john@example.com', '123-456-7890', '123 Elm Street'),
-> ('Jane Smith', 'jane@example.com', '987-654-3210', '456 Maple Avenue'),
-> ('Alice Brown', 'alice@example.com', '555-555-5555', '789 Oak Lane'),
-> ('Bob White', 'bob@example.com', '666-666-6666', '159 Pine Road'),
-> ('Charlie Black', 'charlie@example.com', '777-777-7777', '753 Birch Street');
CustomerID, OrderDate, Amount) VALUES
(1, '2025-02-15', 199.99)
Records: 5 Duplicates: 0 Warnings: 0

MySQL [Group10_db]>
MySQL [Group10_db]> INSERT INTO Orders (CustomerID, OrderDate, Amount) VALUES
-> (1, '2025-02-15', 199.99),
-> (2, '2025-02-16', 299.49),
-> (3, '2025-02-17', 89.99),
-> (4, '2025-02-18', 149.99),
-> (5, '2025-02-19', 349.00);
Query OK, 5 rows affected (0.007 sec)
Records: 5 Duplicates: 0 Warnings: 0

MySQL [Group10_db]> SELECT * FROM Customers;
+-----+-----+-----+-----+-----+
| CustomerID | Name      | Email           | Phone      | Address      |
+-----+-----+-----+-----+-----+
| 1          | John Doe  | john@example.com | 123-456-7890 | 123 Elm Street |
| 2          | Jane Smith | jane@example.com | 987-654-3210 | 456 Maple Avenue |
| 3          | Alice Brown | alice@example.com | 555-555-5555 | 789 Oak Lane |
| 4          | Bob White  | bob@example.com | 666-666-6666 | 159 Pine Road |
| 5          | Charlie Black | charlie@example.com | 777-777-7777 | 753 Birch Street |
+-----+-----+-----+-----+-----+
5 rows in set (0.002 sec)

MySQL [Group10_db]> _
```

```
Select root@ip-10-10-10-7/home/ec2-user

MySQL [Group10_db]> INSERT INTO Customers (Name, Email, Phone, Address) VALUES
-> ('John Doe', 'john@example.com', '123-456-7890', '123 Elm Street'),
-> ('Jane Smith', 'jane@example.com', '987-654-3210', '456 Maple Avenue'),
-> ('Alice Brown', 'alice@example.com', '555-555-5555', '789 Oak Lane'),
-> ('Bob White', 'bob@example.com', '666-666-6666', '159 Pine Road'),
-> ('Charlie Black', 'charlie@example.com', '777-777-7777', '753 Birch Street');
CustomerID, OrderDate, Amount) VALUES
(1, '2025-02-15', 199.99)
Records: 5 Duplicates: 0 Warnings: 0

MySQL [Group10_db]>
MySQL [Group10_db]> INSERT INTO Orders (CustomerID, OrderDate, Amount) VALUES
-> (1, '2025-02-15', 199.99),
-> (2, '2025-02-16', 299.49),
-> (3, '2025-02-17', 89.99),
-> (4, '2025-02-18', 149.99),
-> (5, '2025-02-19', 349.00);
Query OK, 5 rows affected (0.007 sec)
Records: 5 Duplicates: 0 Warnings: 0

MySQL [Group10_db]> SELECT * FROM Customers;
+-----+-----+-----+-----+-----+
| CustomerID | Name      | Email           | Phone      | Address      |
+-----+-----+-----+-----+-----+
| 1          | John Doe  | john@example.com | 123-456-7890 | 123 Elm Street |
| 2          | Jane Smith | jane@example.com | 987-654-3210 | 456 Maple Avenue |
| 3          | Alice Brown | alice@example.com | 555-555-5555 | 789 Oak Lane |
| 4          | Bob White  | bob@example.com | 666-666-6666 | 159 Pine Road |
| 5          | Charlie Black | charlie@example.com | 777-777-7777 | 753 Birch Street |
+-----+-----+-----+-----+-----+
5 rows in set (0.002 sec)

MySQL [Group10_db]> SELECT * FROM Orders;
+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | Amount |
+-----+-----+-----+-----+
| 1       | 1          | 2025-02-15 | 199.99 |
| 2       | 2          | 2025-02-16 | 299.49 |
| 3       | 3          | 2025-02-17 | 89.99  |
| 4       | 4          | 2025-02-18 | 149.99 |
| 5       | 5          | 2025-02-19 | 349.00 |
+-----+-----+-----+-----+
5 rows in set (0.004 sec)

MySQL [Group10_db]>
```

## **Section 2: Report Work**

### **Part 1: Customer Relationship Database**

#### Describe the functionality of the database used to store customer details

Customer Relationship Database is a type of database that stores customer information. It is also able to manage their data, and any transactions made. The functionality of the database allows you to track all purchase history, communication between customers and any data regarding the sales of products. During this assignment, we created two tables, one for Customers and one for Orders. The customer table is essential for storing information regarding the customer, such as their name, email, phone number and address. The order table is in charge of tracking all orders from customers. The use of both these tables allows for a more efficiently run business, improving both customer service and performance in sales.

#### Explain how it helps in business operations

The use of a customer relationship database can help business operations in multiple ways. As mentioned earlier, this database can help improve customer service by providing a more personalized experience and targeted marketing. The database also allows businesses to easily track and manage every customer's order. This allows business to easily locate any purchase history in case of an issue regarding a product. It can also show which customers shop more regularly and which customers are new. Lastly, the use of a customer relationship database can overall help a business stay organized and make customer information easily manageable. This can overall improve the efficiency of a business.

### **Part 2: Cloud Storage Comparison (S3 vs EFS)**

#### When to use EFS

EFS should be used when there are multiple services or EC2 instances that require shared access to a file system. For example, services that need a common set of files and are running on multiple instances should use Amazon EFS. Large data analysis that requires multiple access should also use Amazon EFS.

#### Comparison: When to use Amazon EFS vs. Amazon S3

Amazon S3 is mainly used for hosting websites, data analytics, storage and backups while Amazon EFS should be used when there are multiple services or EC2 instances that require shared access to a file system. Amazon EFS is more commonly used when sharing content and when collaborating with others, making it more useful for work that requires a lot of reading and writing.

#### Explanation of EFS Intelligent-tiering concepts

EFS Intelligent-tiering uses access patterns to move any files between two storage classes, frequent access and infrequent access. Frequent access tier is for files that are accessed often

and required high-performance access. Files that are accessed less often than get moved to the infrequent access tier. Moving files between these two storage classes help businesses with cost efficiency and does not have any effect on performance.

### **Part 3: Network Security (VPC Implementation Notes)**

#### Explain why VPC is essential for cloud security

A Virtual Private Cloud (VPC) is crucial for cloud security because it enables businesses to create a safe, segregated network environment inside the cloud, guaranteeing that private information and apps are shielded from outside attacks. Users can divide their network into subnets, set up routing rules, and implement security measures like firewalls and network access control lists (NACLs) with a VPC. A crucial line of defense in a multi-cloud or hybrid-cloud infrastructure, this isolation stops unwanted access, reduces the attack surface, and guarantees that data traffic moves in a safe environment.

#### Describe the IP addressing used in your setup

Private IP addresses are assigned using the CIDR block 10.10.10.0/24 in my VPC configuration. While internal services like databases are kept segregated in a private network (10.10.16/28), internet-facing resources like web servers with elastic IPs are housed in a public subnet (10.10.10.0/28). Private instances can safely access the internet over a NAT gateway in the public network. Scalability, security, and effective resource allocation are guaranteed by this structured IP addressing.

#### Justify security group settings for optimal access control

My security group configuration limits access to only those who need it. While only internal traffic can access private servers and databases, trustworthy IPs can access public-facing servers via HTTP, HTTPS, and SSH. Outbound rules only permit contact with services that are absolutely necessary. This strategy reduces exposure and guarantees safe, regulated network access.