

# [Challenge] Build and Access an RDS Server

This lab is designed to reinforce the concept of leveraging an AWS-managed database instance for solving relational database needs.

**Amazon Relational Database Service (Amazon RDS)** makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, which allows you to focus on your applications and business. Amazon RDS provides you with six familiar database engines to choose from: Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB.

After completing this lab, you will be able to:

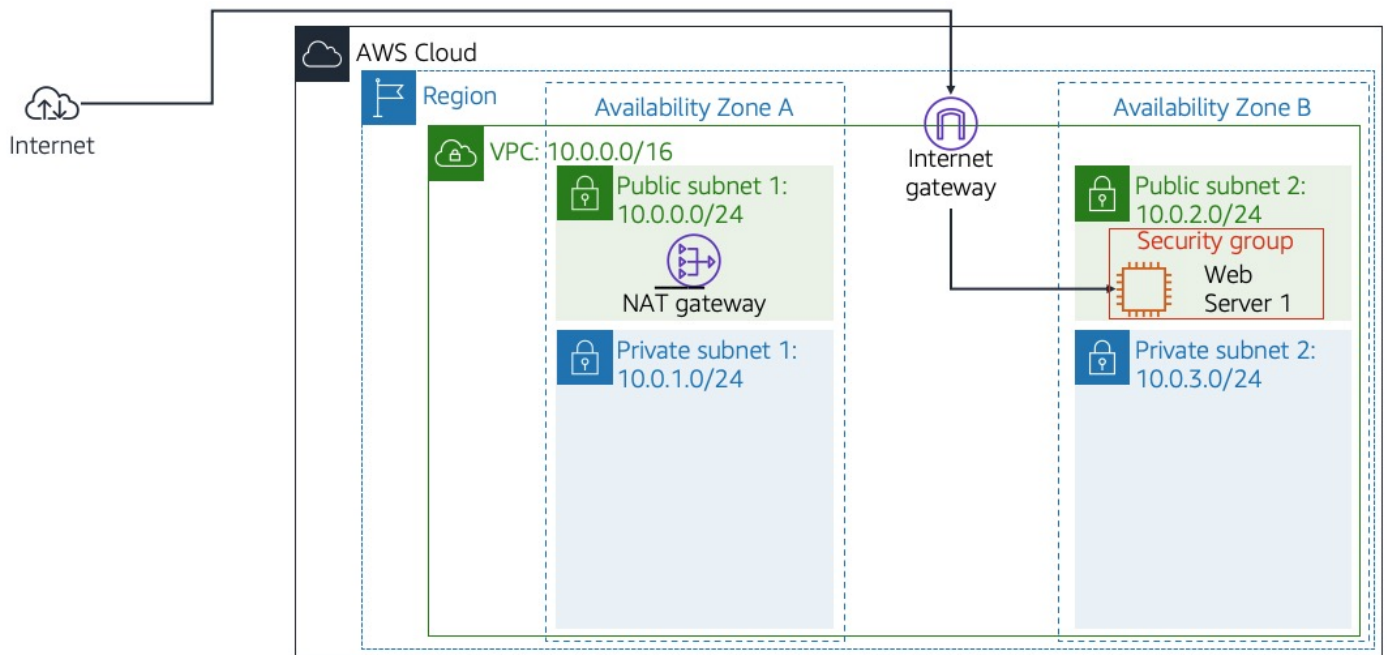
- Create an RDS instance
- Use the Amazon RDS Query Editor to query data.

## Your Challenge

To finish the Challenge do the following:

5. Launch an Amazon RDS DB instance using either Amazon Aurora Provisioned DB or MySQL database engines. Make a note of the DB credentials, as it will be needed in next steps. Please note the following lab restrictions:
  - **DatabaseEngine:** Supported engines are Amazon Aurora or MySQL. Amazon Aurora serverless is not available.
  - **Template:** Choose Dev/Test or Free tier.
  - **Availability and durability:** Avoid creating a standby instance.
  - **DB instance size:** Choose Burstable classes - db.t2 and db.t3 instances of type db.t\*.micro to db.t\*.medium.
  - **Storage:** Choose General Purpose SSD (gp2) of a size up to 100 GB. Provisioned IOPS access is restricted.
  - **Amazon VPC:** Use the *Lab VPC*
  - **Security Group:** Include a security group that will allow the LinuxServer to connect to the RDS instance.
  - For MySQL, under **Additional configuration - Enable Enhanced monitoring** - Disable the option
  - **Purchasing Options:** On-Demand instances are allowed. Other purchasing options are disabled.
6. Click the **Details** followed by **Show**.
7. Click **Download PEM** (for Linux or macOS) or **Download PPK** (for Windows) depending on your local operating system.
8. Make a note of the **LinuxServer** address.
9. Connect (SSH) to the **LinuxServer** using the details you made a note of.
10. Install a MySQL client, and use it to connect to your db. Some helpful information is available [here](#)
11. Create a table RESTART with the following columns – **Capture screenshot for submission**
  - Student ID (Number),
  - Student Name,
  - Restart City,
  - Graduation Date (Date Time)
12. Insert 10 sample rows into this table – **Capture screenshot for submission**
13. Select all rows from this table – **Capture screenshot for submission**
14. Create a table CLOUD\_PRACTITIONER with the following columns – **Capture screenshot for submission**
  - Student ID (Number)
  - certification date (Date Time)
15. Insert 5 sample rows into this table – **Capture screenshot for submission**
16. Select all rows from this table – **Capture screenshot for submission**
17. Perform an inner join between the 2 tables created above and display **student ID, Student Name, Certification Date** – **Capture screenshot for submission**

Start with the following infrastructure:



### Step 1: Create a Security Group for the RDS DB Instance

สร้าง Security group เพื่ออนุญาตให้ Web server เข้าถึง RDS DB instance ได้, Security group นี้จะถูกนำไปใช้เมื่อเรียกใช้งาน DB instance

หมายเหตุ:

- การกำหนดค่านี้จะอนุญาตให้ทุก Web server ภายใน subnet ที่กำหนด สามารถเข้าถึง RDS DB instance ได้
- เพื่อความปลอดภัยที่ดีขึ้น ควรจำกัดการเข้าถึงเฉพาะกับ Web server ที่ต้องการเท่านั้น โดยการระบุ IP address ของเว็บ Web server แทนที่จะใช้ CIDR block ทั้งหมด

1) Network & Content Delivery > VPC > Security Groups

เพิ่ม Inbound rule ไปใน Security group เพื่ออนุญาตการร้องขอฐานข้อมูลขาเข้า เนื่องจาก Security group ยังไม่มี rule ใดๆในขณะนี้ เราจะเพิ่ม rule เพื่ออนุญาตการเข้าถึงจาก Web Security Group

หมายเหตุ:

- การกำหนดค่านี้จะอนุญาตให้เฉพาะ instance ที่อยู่ใน Web Security Group เท่านั้นที่จะสามารถเข้าถึง DB instance ได้
- วิธีนี้ปลอดภัยกว่าการอนุญาตการเข้าถึงจาก CIDR block ทั้งหมด

2) .. Inbound rules > Type: MySQL/Aurora (3306) > Source: sg Web Security Group

การกำหนดค่านี้ช่วยให้ DB Security group ของฐานข้อมูลอนุญาตทางด้านขาเข้า บนพอร์ต 3306 จาก EC2 instance ใดๆ ที่เชื่อมโยงกับ Web Security Group

หมายเหตุ:

- การกำหนดค่านี้ไม่ปลอดภัย เนื่องจากอนุญาตให้ EC2 instance ใดๆ ที่เชื่อมโยงกับ Web Security Group เข้าถึงฐานข้อมูลของคุณได้ ซึ่งรวมถึง instance ที่อาจไม่ได้ตั้งใจให้เข้าถึง
- เพื่อความปลอดภัยควรจำกัดการเข้าถึงฐานข้อมูลเฉพาะกับ instance ที่ต้องการเท่านั้น โดยการระบุ IP address ของ Web server ที่ต้องการ แทนที่จะใช้ Security group ทั้งหมด

3) Create security group

ใช้ Security group นี้เมื่อเรียกใช้งานฐานข้อมูล Amazon RDS

## Step 2: Create a DB Subnet Group

สร้างกลุ่ม DB subnet group เพื่อระบุให้ RDS รู้ว่า subnet ใดบ้างที่สามารถใช้สำหรับฐานข้อมูลได้, DB subnet แต่ละกลุ่มจะต้องมี subnet อยู่อย่างน้อยใน 2 Availability Zones (AZ) ที่แตกต่างกัน (เพื่อให้แน่ใจว่าฐานข้อมูลมีความยืดหยุ่นและทนทานต่อความผิดพลาด)

หมายเหตุ:

- การใช้ DB subnet group ช่วยจัดการความปลอดภัยและกำหนดเส้นทางเครือข่ายสำหรับ DB instance ได้
- 4) Databases RDs > Create DB Subnet Group
  - 5) Add subnets สำหรับ Availability Zone 1 และ 2
  - 6) Subnets สำหรับ Availability Zone 1 เลือก 10.0.1.0/24 และ Availability Zone 2 เลือก 10.0.3.0/24
  - 7) Create

สร้าง Private Subnet 1 (10.0.1.0/24) และ Private Subnet 2 (10.0.3.0/24), ใช้ DB subnet group นี้ เมื่อสร้างฐานข้อมูลในลำดับต่อไป

## Step 3: Create an Amazon RDS DB Instance

กำหนดค่าและเรียกใช้งาน MySQL DB instance ของ Amazon RDS แบบ Multi-AZ

การใช้งาน Multi-AZ ของ Amazon RDS ช่วยเพิ่มความพร้อมใช้งานและความทนทานให้กับ DB instance ทำให้เหมาะสำหรับการใช้งานฐานข้อมูลจริง (production workloads) เมื่อเตรียมการใช้งาน DB instance แบบ Multi-AZ, Amazon RDS จะสร้าง primary DB instance โดยอัตโนมัติ และทำการจำลองข้อมูลแบบทันที (synchronously) ไปยัง standby instance (instance สำรอง) ใน Availability Zone (AZ) ที่แตกต่างกัน

- 8) Databases > Create database > Standard create
- 9) Engine type > MySQL/Aurora

10) Templates > Dev/Test

11) Availability and durability > Multi-AZ DB Instance

12) Settings > DB instance identifier: lab-db, Master username: main, Master password: lab-password, Confirm password: lab-password

13) Instance configuration > DB instance class > Burstable classes (includes t classes) > db.t3.medium

**\*\* Burstable classes เป็นตัวเลือกที่ประหยัดค่าใช้จ่ายสำหรับ workloads ที่มีความต้องการ CPU ไม่สม่ำเสมอ \*\***

14) Storage type > General Purpose (SSD)

15) Connectivity > Virtual Private Cloud (VPC): Lab VPC

16) VPC security group > Choose existing

17) Existing VPC security groups > DB Security Group

18) Monitoring > Additional configuration > uncheck Enable Enhanced monitoring (เพื่อประหยัดค่าใช้จ่ายสำหรับ workloads พื้นฐานที่ไม่ต้องการการวิเคราะห์เชิงลึก)

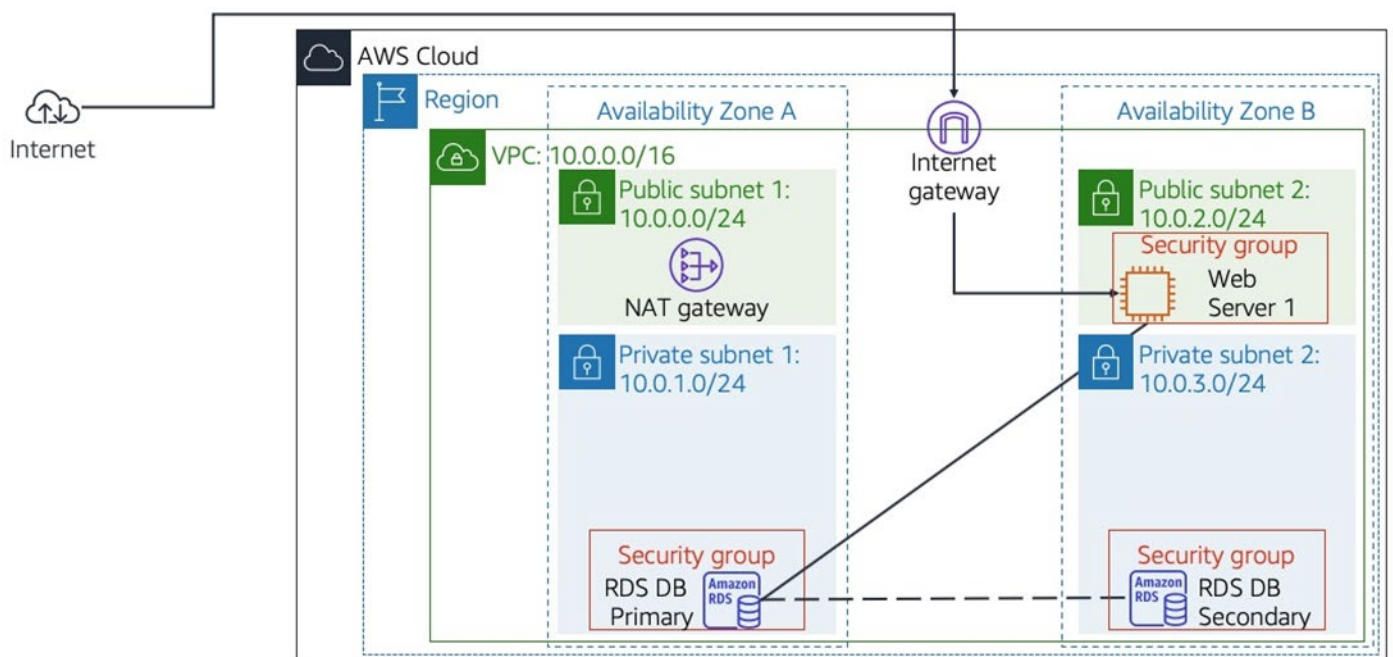
19) Additional configuration > Initial database name: lab, Backup: uncheck Enable automated backups (เพื่อประหยัดค่าใช้จ่ายสำหรับ workloads พื้นฐานที่ไม่สำคัญมาก)

การปิดการสำรองข้อมูล โดยปกติจะไม่แนะนำ แต่จะทำให้การ deploy ฐานข้อมูล (กระบวนการติดตั้ง กำหนดค่า และทำให้ฐานข้อมูลพร้อมใช้งานสำหรับผู้ใช้) สำหรับ lab นี้เร็วขึ้น

20) Create database

ฐานข้อมูลถูกเรียกใช้งานอยู่ในขณะนี้

At the end of #3, this is the infrastructure:



#### Step 4: Use SSH to connect to an Amazon Linux EC2 instance

เชื่อมต่อกับ Amazon Linux EC2 instance โดยใช้ SSH Utility เพื่อดำเนินการต่อ

21) CLI > `ssh -i labsuser.pem ec2-user@<public-ip>` (-i ใช้สำหรับระบุ private key)

#### Step 5: Connecting to a DB instance running the MySQL database engine

To install the MySQL command-line client on Amazon Linux 2, run the following command:

22) `sudo yum install mariadb`

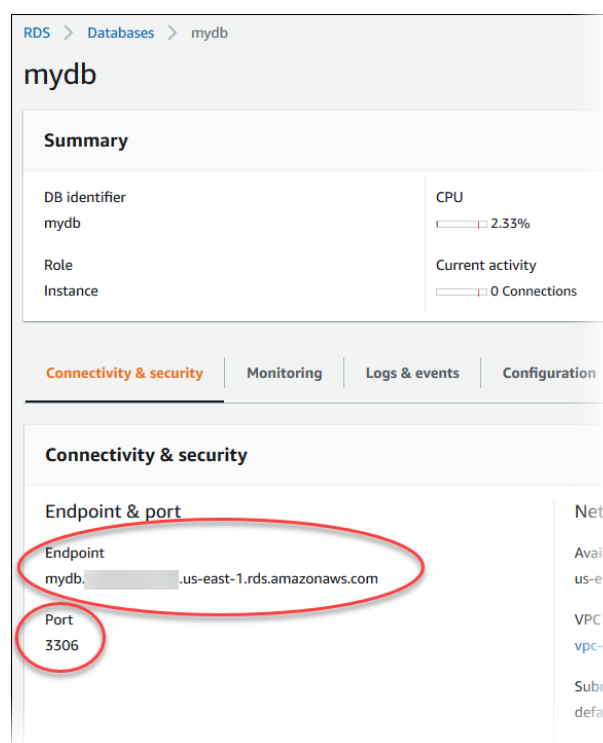
To check the version of your MySQL command-line client, run the following command:

- `mysql --version`

To read the MySQL documentation for your current client version, run the following command:

- `man mysql`

#### Finding the connection information for a MySQL DB instance



#### Connecting from the MySQL command-line client (unencrypted)

**สำคัญ!** อนุญาตให้ใช้การเชื่อมต่อ MySQL แบบไม่เข้ารหัส เฉพาะกรณีที่ client และ server อยู่ใน VPC เดียวกัน และ เครือข่ายมีความปลอดภัยสูง

23) `mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com -P 3306 -u mymasteruser -p`

- mysql: โปรแกรมคำสั่ง MySQL

- -h: ระบุชื่อ DNS (endpoint) ของ DB instance
- **mysql-instance1.123456789012.us-east-1.rds.amazonaws.com**: ชื่อ DNS ของ DB instance
- -P: ระบุพอร์ตของ DB instance
- 3306: พอร์ตของ DB instance (โดยปกติสำหรับ MySQL)
- -u: ระบุชื่อผู้ใช้งานข้อมูล
- **mymasteruser**: ชื่อผู้ใช้งานข้อมูล
- -p: แจ้งให้ป้อนรหัสผ่าน

\*\* DNS (Domain Name System) คือ ระบบที่ใช้สำหรับแปลงชื่อโดเมนของเว็บไซต์หรืออุปกรณ์บนเครือข่าย เป็นที่อยู่ IP (IP address) ซึ่งเป็นตัวระบุที่อยู่ของเซิร์ฟเวอร์หรืออุปกรณ์บนเครือข่าย \*\*

\*\* endpoint คือ จุดเชื่อมต่อระหว่างผู้ใช้หรือแอปพลิเคชันกับฐานข้อมูล เปรียบเสมือนประตูที่ช่วยให้เข้าถึงข้อมูลภายในฐานข้อมูลได้ \*\*

## Lab Results

Create a table RESTART:

```
MySQL [lab]> CREATE TABLE RESTART(
  -> Student_ID INT NOT NULL,
  -> Student_Name VARCHAR(20),
  -> Restart_City VARCHAR(20),
  -> Graduation_Date DATETIME
  -> );
Query OK, 0 rows affected (0.05 sec)
```

```
MySQL [lab]> SHOW tables;
+-----+
| Tables_in_lab |
+-----+
| RESTART        |
+-----+
1 row in set (0.00 sec)
```

Insert 19 sample rows into this table:

```
MySQL [lab]> INSERT INTO RESTART (Student_ID, Student_Name, Restart_City, Graduation_Date)
  -> VALUES
  -> (1, 'Get Tanakit', 'Bangkok', '2024-04-01 00:00:00'),
  -> (2, 'Mo Salah', 'Cairo', '2024-04-02 00:00:00'),
  -> (3, 'Darwin Nunez', 'Montevideo', '2024-04-03 00:00:00'),
  -> (4, 'Andrea Pirlo', 'Milan', '2024-04-04 00:00:00'),
  -> (5, 'Oliver Kahn', 'Munich', '2024-04-05 00:00:00'),
  -> (6, 'Steven Gerrard', 'Liverpool', '2024-04-06 00:00:00'),
  -> (7, 'Xabi Alonso', 'Madrid', '2024-04-07 00:00:00'),
  -> (8, 'Wataru Endo', 'Tokyo', '2024-04-08 00:00:00'),
  -> (9, 'Jack Ma', 'Beijing', '2024-04-09 00:00:00'),
  -> (10, 'Lebron James', 'Los Angeles', '2024-04-10 00:00:00');
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

Select all rows from this table:

```
MySQL [lab]> SELECT * FROM RESTART;
```

Student_ID	Student_Name	Restart_City	Graduation_Date
1	Get Tanakit	Bangkok	2024-04-01 00:00:00
2	Mo Salah	Cairo	2024-04-02 00:00:00
3	Darwin Nunez	Montevideo	2024-04-03 00:00:00
4	Andrea Pirlo	Milan	2024-04-04 00:00:00
5	Oliver Kahn	Munich	2024-04-05 00:00:00
6	Steven Gerrard	Liverpool	2024-04-06 00:00:00
7	Xabi Alonso	Madrid	2024-04-07 00:00:00
8	Wataru Endo	Tokyo	2024-04-08 00:00:00
9	Jack Ma	Beijing	2024-04-09 00:00:00
10	Lebron James	Los Angeles	2024-04-10 00:00:00

```
10 rows in set (0.00 sec)
```

Create a table CLOUD\_PRACTITIONER:

```
MySQL [lab]> CREATE TABLE CLOUD_PRACTITIONER (  
  -> Student_ID INT(8) NOT NULL,  
  -> Certificate_Date DATETIME  
  -> );  
Query OK, 0 rows affected, 1 warning (0.08 sec)
```

```
MySQL [lab]> DESCRIBE CLOUD_PRACTITIONER;
```

Field	Type	Null	Key	Default	Extra
Student_ID	int	NO		NULL	
Certificate_Date	datetime	YES		NULL	

```
2 rows in set (0.01 sec)
```

Insert 5 sample rows into this table:

Select all rows from this table:

```
MySQL [lab]> INSERT INTO CLOUD_PRACTITIONER (Student_ID, Certificate_Date)
-> VALUES
-> (1, '2024-04-11 00:00:00'),
-> (2, '2024-04-12 00:00:00'),
-> (3, '2024-04-13 00:00:00'),
-> (4, '2024-04-14 00:00:00'),
-> (5, '2024-04-15 00:00:00');
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
MySQL [lab]> SELECT * FROM CLOUD_PRATITIONER;
ERROR 1146 (42S02): Table 'lab.CLOUD_PRATITIONER' doesn't exist
MySQL [lab]> SELECT * FROM CLOUD_PRACTITIONER;
```

```
+-----+-----+
| Student_ID | Certificate_Date |
+-----+-----+
| 1 | 2024-04-11 00:00:00 |
| 2 | 2024-04-12 00:00:00 |
| 3 | 2024-04-13 00:00:00 |
| 4 | 2024-04-14 00:00:00 |
| 5 | 2024-04-15 00:00:00 |
+-----+-----+
5 rows in set (0.00 sec)
```

Perform an inner join between the 2 tables create above and display:

```
MySQL [lab]> SELECT RE.Student_ID, RE.Student_Name, CL.Certificate_date
-> FROM RESTART RE
-> JOIN CLOUD_PRACTITIONER CL ON RE.Student_ID = CL.Student_ID;
+-----+-----+-----+
| Student_ID | Student_Name | Certificate_date |
+-----+-----+-----+
| 1 | Get Tanakit | 2024-04-11 00:00:00 |
| 2 | Mo Salah | 2024-04-12 00:00:00 |
| 3 | Darwin Nunez | 2024-04-13 00:00:00 |
| 4 | Andrea Pirlo | 2024-04-14 00:00:00 |
| 5 | Oliver Kahn | 2024-04-15 00:00:00 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```