AWS re:Start

CHALLENGE LAB

Build and Access an RDS Server

WEEK 7

Overview

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.

Your Challenge

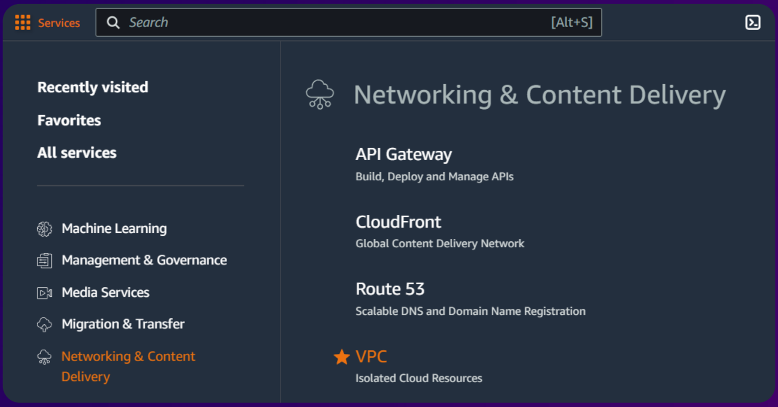
- Launch an Amazon RDS DB instance using either Amazon Aurora Provisioned DB or MySQL database engines.
- Connect (SSH) to the LinuxServer
- Install a MySQL client, and use it to connect to your db.
- Create a table RESTART with the following columns:
  - Student ID (Number),
  - Student Name,
  - Restart City,
  - Graduation Date (Date Time)
- Insert 10 sample rows into this table
- Select all rows from this table
- Create a table CLOUD_PRACTITIONER with the following columns:
  - Student ID (Number)
  - Certification Date (Date Time)
- Insert 5 sample rows into this table
- Select all rows from this table
- Perform an inner join between the 2 tables created above and display student ID, Student Name, Certification Date

Task 1

Create a Security Group for the RDS DB Instance

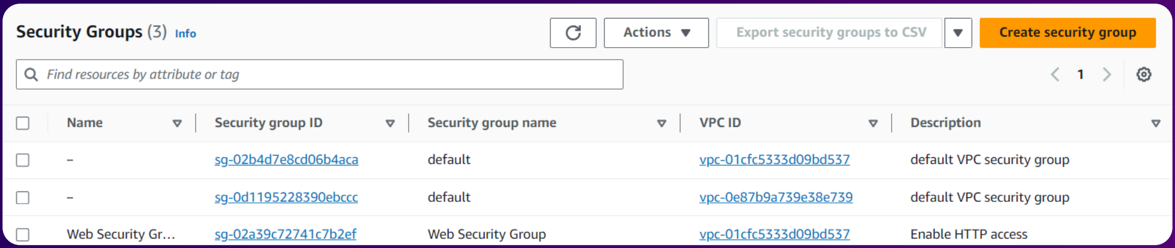
Step 1: Access the VPC management console

Open the AWS Management Console, and select VPC.



Step 2: Create security group

Navigate to the **Security Groups** section, and select Create security group.


Security Groups (3) Info
Actions
Export security groups to CSV
Create security group
Find resources by attribute or tag
1

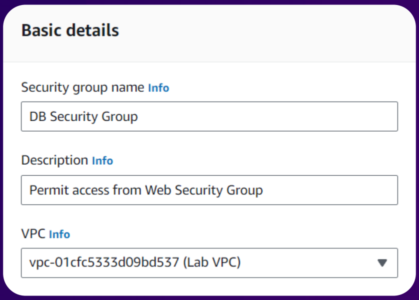
□	Name	Security group ID	Security group name	VPC ID	Description
□	–	sg-02b4d7e8cd06b4aca	default	vpc-01cfc5333d09bd537	default VPC security group
□	–	sg-0d1195228390ebccc	default	vpc-0e87b9a739e38e739	default VPC security group
□	Web Security Gr…	sg-02a39c72741c7b2ef	Web Security Group	vpc-01cfc5333d09bd537	Enable HTTP access

Task 1

Create a Security Group for the RDS DB Instance

Step 3: Basic details

In the **Basic details** section, configure the DB Security Group using the following settings.



Basic details

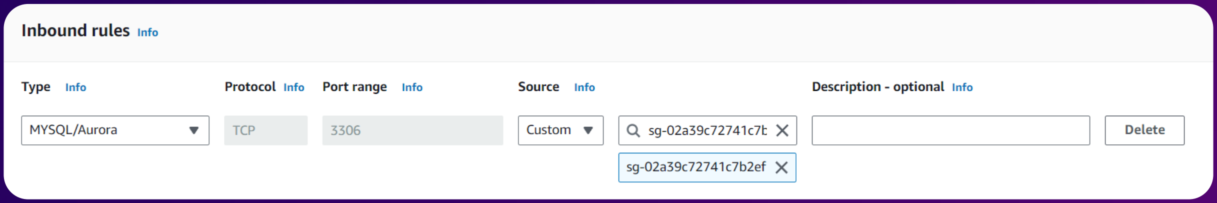
Security group name Info: DB Security Group

Description Info: Permit access from Web Security Group

VPC Info: vpc-01cfc5333d09bd537 (Lab VPC) ▼

Step 4: Inbound rules

In the **Inbound rules** section, configure the DB Security Group to permit inbound traffic on port 3306 from any EC2 instance that is associated with the Web Security Group.



Inbound rules Info

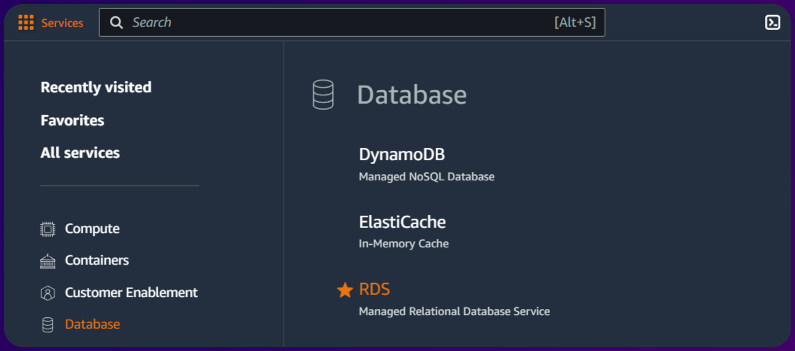
Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
MYSQL/Aurora ▼	TCP	3306	Custom ▼	sg-02a39c72741c7t ✕ sg-02a39c72741c7b2ef ✕		Delete

Task 2

Create a DB Subnet Group

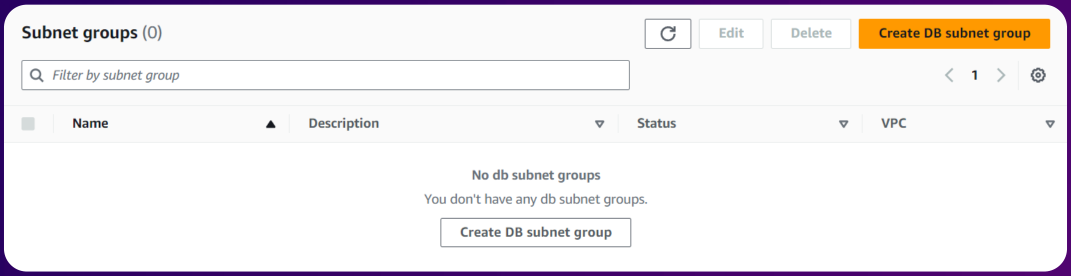
Step 1: Access the RDS database service

In the AWS Management Console, select RDS.



Step 2: Create DB subnet group

Navigate to the **Subnet groups** section, and select Create DB subnet group.

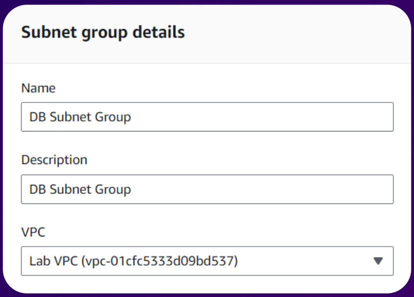


Task 2

Create a DB Subnet Group

Step 3: Subnet group details

In the **Subnet group details** section, configure the DB Subnet Group using the following settings.



Subnet group details

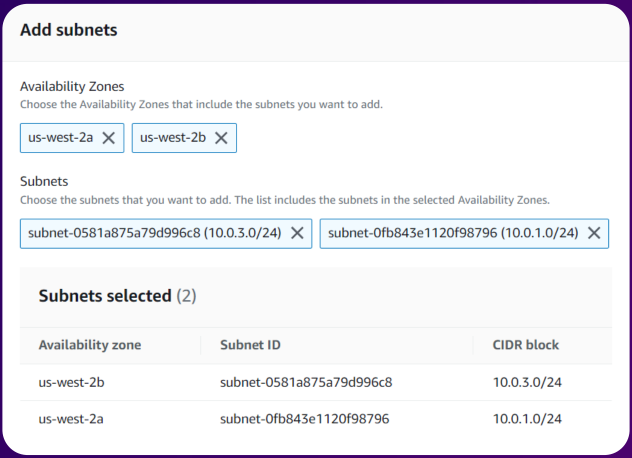
Name: DB Subnet Group

Description: DB Subnet Group

VPC: Lab VPC (vpc-01cfc5333d09bd537)

Step 4: Add subnets

In the **Add subnets** section, configure the following settings.



Add subnets

Availability Zones
Choose the Availability Zones that include the subnets you want to add.

us-west-2a ✕ us-west-2b ✕

Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

subnet-0581a875a79d996c8 (10.0.3.0/24) ✕ subnet-0fb843e1120f98796 (10.0.1.0/24) ✕

Subnets selected (2)

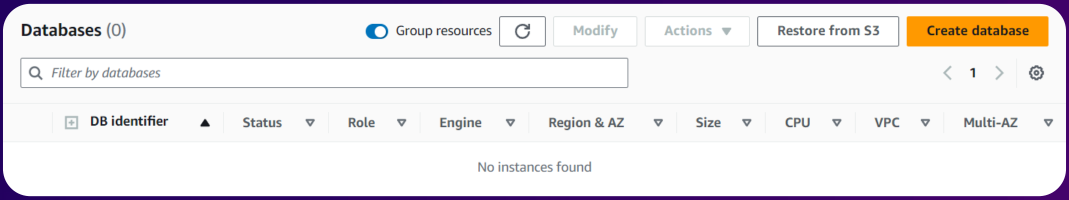
Availability zone	Subnet ID	CIDR block
us-west-2b	subnet-0581a875a79d996c8	10.0.3.0/24
us-west-2a	subnet-0fb843e1120f98796	10.0.1.0/24

Task 3

Launch an Amazon RDS DB instance

Step 1: Create database

Navigate to the Databases section, and select Create database.



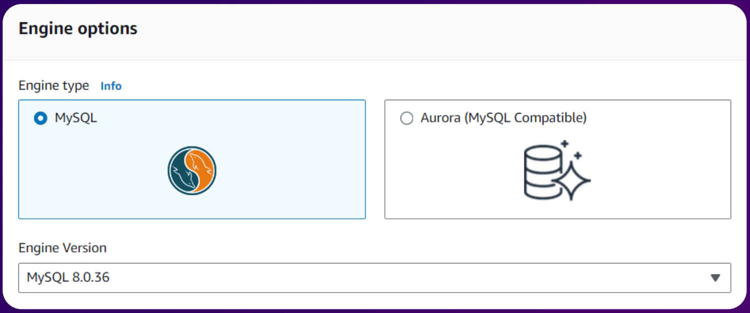
Databases (0)

DB identifier ▲	Status ▽	Role ▽	Engine ▽	Region & AZ ▽	Size ▽	CPU ▽	VPC ▽	Multi-AZ ▽

No instances found

Step 2: Engine options

In the **Engine options** section, for Engine type, choose MySQL, for Engine version, choose the latest version.



Engine options

Engine type Info

● MySQL ○ Aurora (MySQL Compatible)

Engine Version

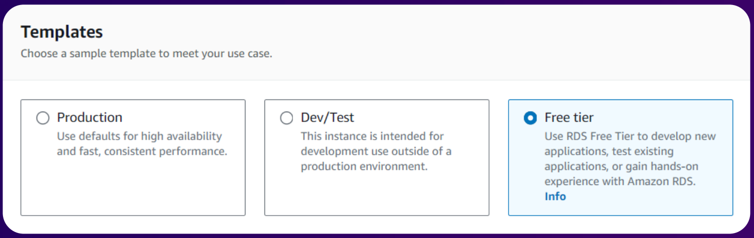
MySQL 8.0.36 ▼

Task 3

Launch an Amazon RDS DB instance

Step 3: Templates

In the **Templates** section, choose Free tier.



Templates

Choose a sample template to meet your use case.

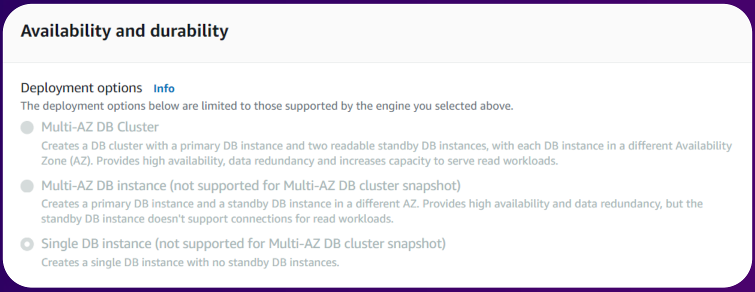
○ Production
Use defaults for high availability and fast, consistent performance.

○ Dev/Test
This instance is intended for development use outside of a production environment.

● Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
Info

Step 4: Availability and durability

Notice that when you selected the Free tier template, the Single DB instance option was selected and locked as the default deployment option in the **Availability and durability** section.



Availability and durability

Deployment options Info

The deployment options below are limited to those supported by the engine you selected above.

○ Multi-AZ DB Cluster
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.

○ Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.

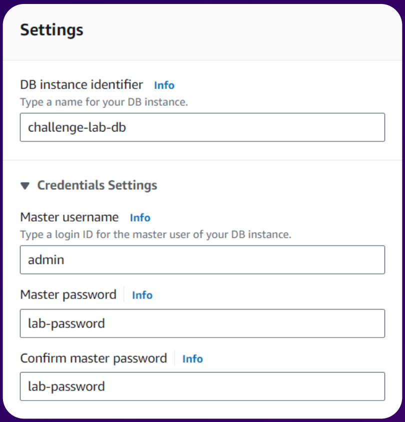
● Single DB instance (not supported for Multi-AZ DB cluster snapshot)
Creates a single DB instance with no standby DB instances.

Task 3

Launch an Amazon RDS DB instance

Step 5: Settings

In the **Settings** section, configure the following parameters.



Settings

DB instance identifier Info
Type a name for your DB instance.
challenge-lab-db

▼ Credentials Settings

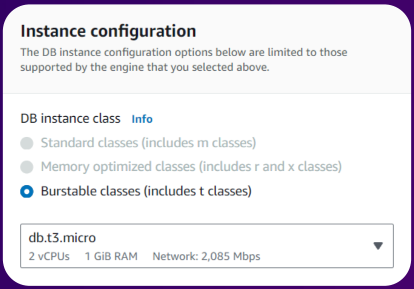
Master username Info
Type a login ID for the master user of your DB instance.
admin

Master password Info
lab-password

Confirm master password Info
lab-password

Step 6: Instance configuration

In the **Instance configuration** section, for DB instance class, configure the following settings.



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

DB instance class Info
○ Standard classes (includes m classes)
○ Memory optimized classes (includes r and x classes)
● Burstable classes (includes t classes)

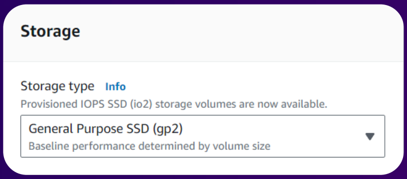
db.t3.micro ▾
2 vCPUs 1 GiB RAM Network: 2,085 Mbps

Task 3

Launch an Amazon RDS DB instance

Step 7: Storage

In the **Storage** section, for Storage type, select General Purpose SSD (gp2).



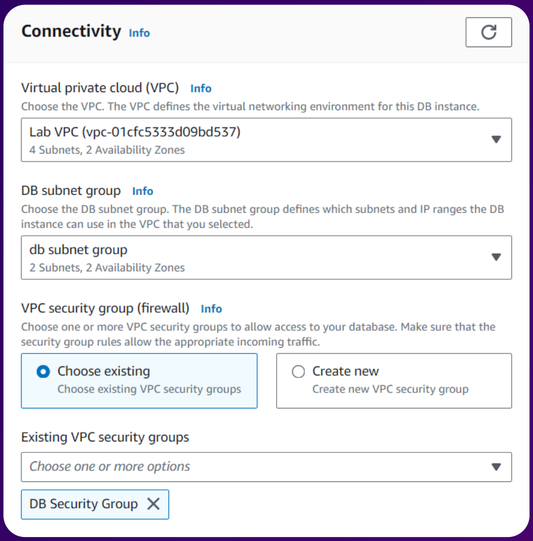
Storage

Storage type Info
Provisioned IOPS SSD (io2) storage volumes are now available.

General Purpose SSD (gp2) ▼
Baseline performance determined by volume size

Step 8: Connectivity

In the **Connectivity** section, configure the following settings.



Connectivity Info

Virtual private cloud (VPC) Info
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Lab VPC (vpc-01cfc5333d09bd537) ▼
4 Subnets, 2 Availability Zones

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.

db subnet group ▼
2 Subnets, 2 Availability Zones

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 ▼

DB Security Group ✕

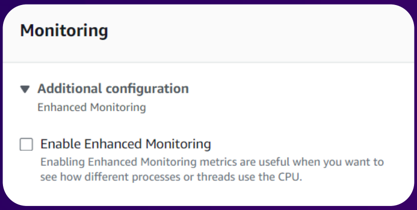
aws re/start

Task 3

Launch an Amazon RDS DB instance

Step 9: Monitoring

In the **Monitoring** section, for Additional configuration, uncheck Enable Enhanced Monitoring.



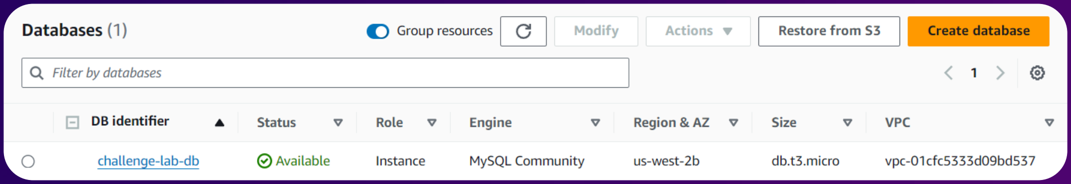
Monitoring

▼ Additional configuration
Enhanced Monitoring

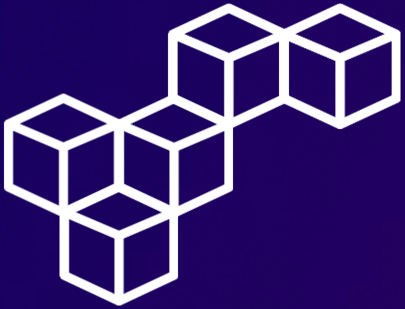
□ Enable Enhanced Monitoring
Enabling Enhanced Monitoring metrics are useful when you want to see how different processes or threads use the CPU.

Step 10: Review database creation

Verify the availability of the challenge-lab-db database and take note of its endpoint in the **Connectivity & Security** section.



	DB identifier ▲	Status	Role	Engine	Region & AZ	Size	VPC
○	challenge-lab-db	✓ Available	Instance	MySQL Community	us-west-2b	db.t3.micro	vpc-01cfc5333d09bd537



Task 4

Use SSH to connect to the Linux Server

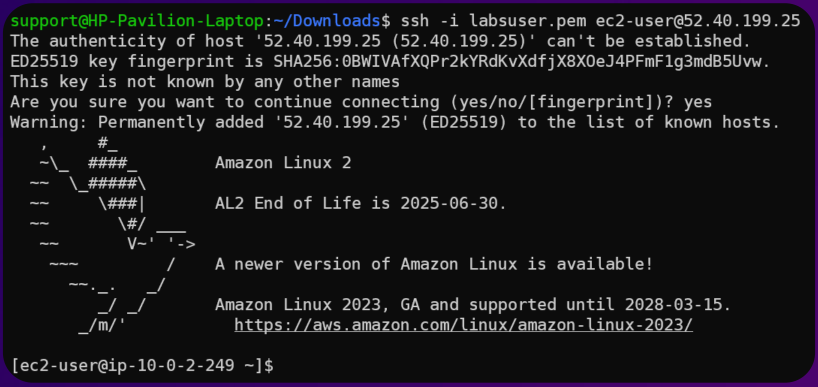
Initial Preparations

In the AWS Management Console, select the LinuxServer EC2 instance and make note of the **Public IPv4 address**.

Download the **private key file** labsuser.pem. Change to the Downloads directory and modify the permissions on the key to be read-only (r--------).

Connect to the Linux Server using SSH

Establish a connection to the LinuxServer EC2 instance using the ssh command, the key and the instance’s public IPv4 address.



```
support@HP-Pavilion-Laptop:~/Downloads$ ssh -i labsuser.pem ec2-user@52.40.199.25
The authenticity of host '52.40.199.25 (52.40.199.25)' can't be established.
ED25519 key fingerprint is SHA256:0BWIVAfXQPr2kYRdKvXdfjX8XOeJ4PFmF1g3mdB5Uvw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.40.199.25' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       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.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-2-249 ~]$
```

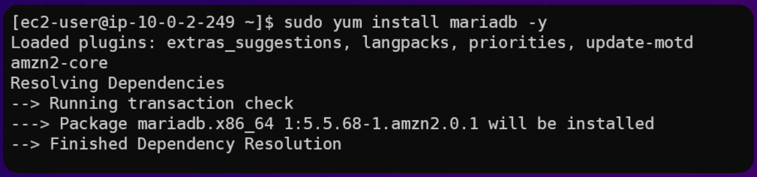
aws re/start

Task 5

Configure the EC2 instance to connect to the DB

Step 1: Install the DB client

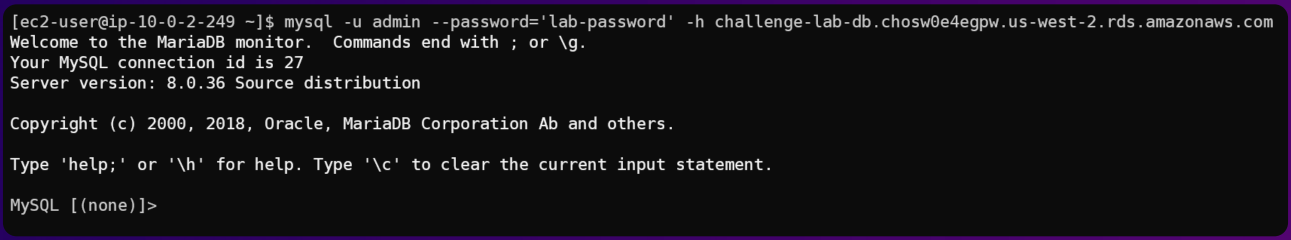
Run the command sudo yum install mariadb –y to install the MariaDB client.



```
[ec2-user@ip-10-0-2-249 ~]$ sudo yum install mariadb -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
---> Package mariadb.x86_64 1:5.5.68-1.amzn2.0.1 will be installed
--> Finished Dependency Resolution
```

Step 2: Connect to the database

After installing a MySQL client, run the following command to connect to the database.



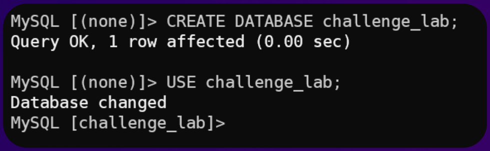
```
[ec2-user@ip-10-0-2-249 ~]$ mysql -u admin --password='lab-password' -h challenge-lab-db.chosw0e4egpw.us-west-2.rds.amazonaws.com
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 27
Server version: 8.0.36 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

Task 6

Interact With Your DB

Step 1: Create a database

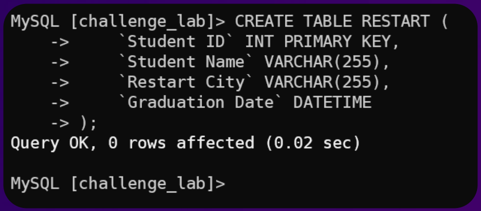
CREATE the **challenge_lab** database and switch to it.



```
MySQL [(none)]> CREATE DATABASE challenge_lab;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> USE challenge_lab;
Database changed
MySQL [challenge_lab]>
```

Step 2: Create the RESTART table

CREATE a table **RESTART** with the following columns: Student ID (Number), Student Name, Restart City, Graduation Date (Date Time).



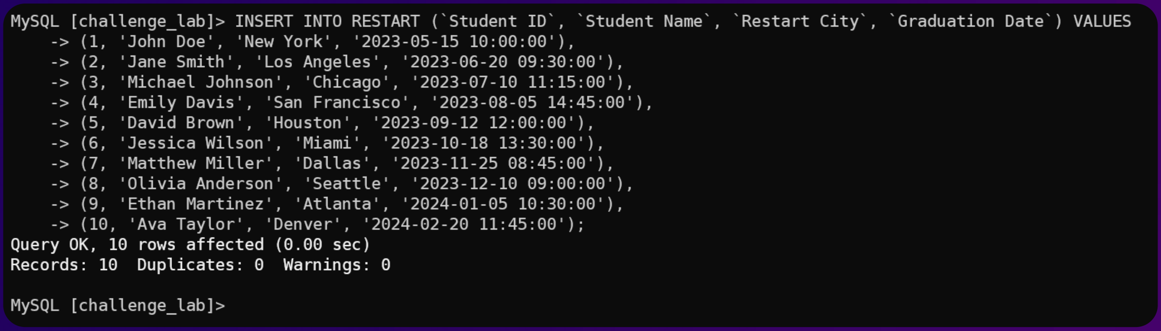
```
MySQL [challenge_lab]> CREATE TABLE RESTART (
    ->     `Student ID` INT PRIMARY KEY,
    ->     `Student Name` VARCHAR(255),
    ->     `Restart City` VARCHAR(255),
    ->     `Graduation Date` DATETIME
    -> );
Query OK, 0 rows affected (0.02 sec)

MySQL [challenge_lab]>
```

Task 6

Interact With Your DB

Step 3: Insert sample rows

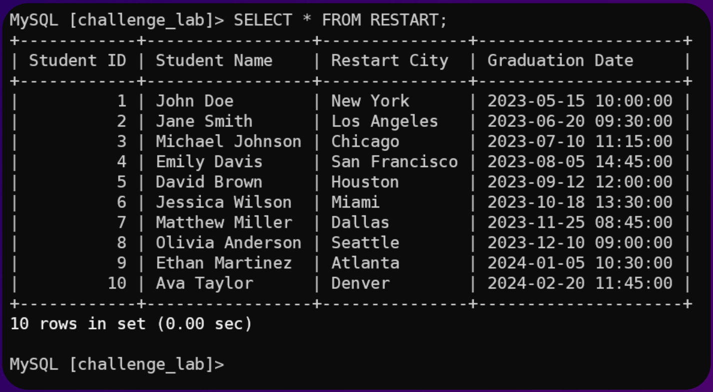
INSERT 10 sample rows into the RESTART table.



```
MySQL [challenge_lab]> INSERT INTO RESTART (`Student ID`, `Student Name`, `Restart City`, `Graduation Date`) VALUES
    -> (1, 'John Doe', 'New York', '2023-05-15 10:00:00'),
    -> (2, 'Jane Smith', 'Los Angeles', '2023-06-20 09:30:00'),
    -> (3, 'Michael Johnson', 'Chicago', '2023-07-10 11:15:00'),
    -> (4, 'Emily Davis', 'San Francisco', '2023-08-05 14:45:00'),
    -> (5, 'David Brown', 'Houston', '2023-09-12 12:00:00'),
    -> (6, 'Jessica Wilson', 'Miami', '2023-10-18 13:30:00'),
    -> (7, 'Matthew Miller', 'Dallas', '2023-11-25 08:45:00'),
    -> (8, 'Olivia Anderson', 'Seattle', '2023-12-10 09:00:00'),
    -> (9, 'Ethan Martinez', 'Atlanta', '2024-01-05 10:30:00'),
    -> (10, 'Ava Taylor', 'Denver', '2024-02-20 11:45:00');
Query OK, 10 rows affected (0.00 sec)
Records: 10  Duplicates: 0  Warnings: 0

MySQL [challenge_lab]>
```

Step 4: Select all rows

SELECT all rows from the RESTART table.



```
MySQL [challenge_lab]> SELECT * FROM RESTART;
+------------+-----------------+---------------+---------------------+
| Student ID | Student Name    | Restart City  | Graduation Date     |
+------------+-----------------+---------------+---------------------+
|          1 | John Doe        | New York      | 2023-05-15 10:00:00 |
|          2 | Jane Smith      | Los Angeles   | 2023-06-20 09:30:00 |
|          3 | Michael Johnson | Chicago       | 2023-07-10 11:15:00 |
|          4 | Emily Davis     | San Francisco | 2023-08-05 14:45:00 |
|          5 | David Brown     | Houston       | 2023-09-12 12:00:00 |
|          6 | Jessica Wilson  | Miami         | 2023-10-18 13:30:00 |
|          7 | Matthew Miller  | Dallas        | 2023-11-25 08:45:00 |
|          8 | Olivia Anderson | Seattle       | 2023-12-10 09:00:00 |
|          9 | Ethan Martinez  | Atlanta       | 2024-01-05 10:30:00 |
|         10 | Ava Taylor      | Denver        | 2024-02-20 11:45:00 |
+------------+-----------------+---------------+---------------------+
10 rows in set (0.00 sec)

MySQL [challenge_lab]>
```

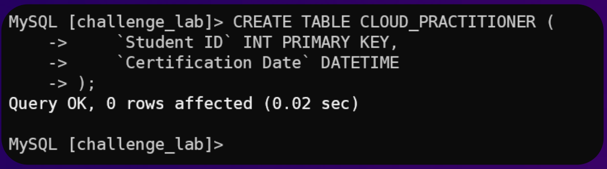
aws re/start

Task 6

Interact With Your DB

Step 5: Create the CLOUD_PRACTITIONER table

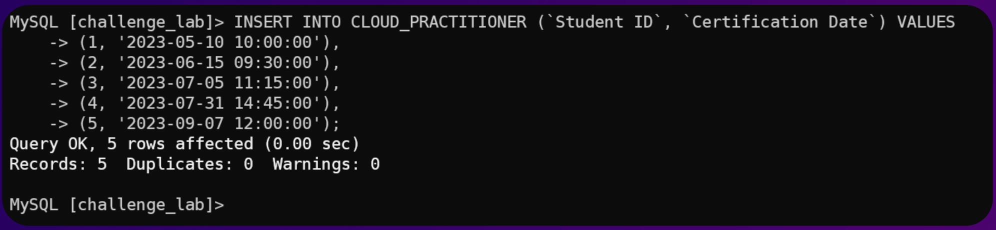
CREATE a table CLOUD_PRACTITIONER with the following columns: Student ID (Number), Certification Date (Date Time).



```
MySQL [challenge_lab]> CREATE TABLE CLOUD_PRACTITIONER (
    ->     `Student ID` INT PRIMARY KEY,
    ->     `Certification Date` DATETIME
    -> );
Query OK, 0 rows affected (0.02 sec)

MySQL [challenge_lab]>
```

Step 6: Insert sample rows

INSERT 5 sample rows into the CLOUD_PRACTITIONER table.



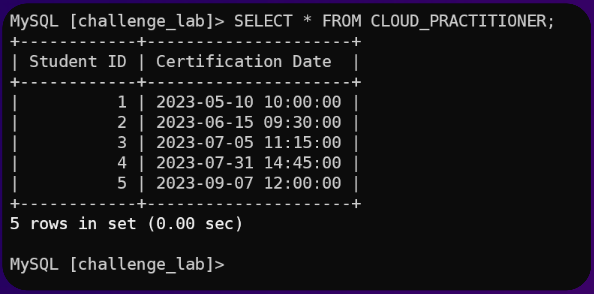
```
MySQL [challenge_lab]> INSERT INTO CLOUD_PRACTITIONER (`Student ID`, `Certification Date`) VALUES
    -> (1, '2023-05-10 10:00:00'),
    -> (2, '2023-06-15 09:30:00'),
    -> (3, '2023-07-05 11:15:00'),
    -> (4, '2023-07-31 14:45:00'),
    -> (5, '2023-09-07 12:00:00');
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0

MySQL [challenge_lab]>
```

Task 6

Interact With Your DB

Step 7: Select all rows

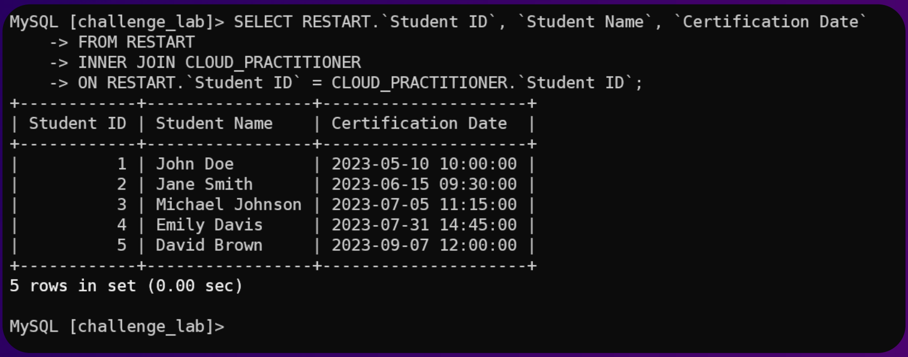
SELECT all rows from the CLOUD_PRACTITIONER table.



```
MySQL [challenge_lab]> SELECT * FROM CLOUD_PRACTITIONER;
+------------+---------------------+
| Student ID | Certification Date  |
+------------+---------------------+
|          1 | 2023-05-10 10:00:00 |
|          2 | 2023-06-15 09:30:00 |
|          3 | 2023-07-05 11:15:00 |
|          4 | 2023-07-31 14:45:00 |
|          5 | 2023-09-07 12:00:00 |
+------------+---------------------+
5 rows in set (0.00 sec)

MySQL [challenge_lab]>
```

Step 8: Perform an inner join

Perform an INNER JOIN between the 2 tables created above and display Student ID, Student Name, Certification Date.



```
MySQL [challenge_lab]> SELECT RESTART.`Student ID`, `Student Name`, `Certification Date`
    -> FROM RESTART
    -> INNER JOIN CLOUD_PRACTITIONER
    -> ON RESTART.`Student ID` = CLOUD_PRACTITIONER.`Student ID`;
+------------+-----------------+---------------------+
| Student ID | Student Name    | Certification Date  |
+------------+-----------------+---------------------+
|          1 | John Doe        | 2023-05-10 10:00:00 |
|          2 | Jane Smith      | 2023-06-15 09:30:00 |
|          3 | Michael Johnson | 2023-07-05 11:15:00 |
|          4 | Emily Davis     | 2023-07-31 14:45:00 |
|          5 | David Brown     | 2023-09-07 12:00:00 |
+------------+-----------------+---------------------+
5 rows in set (0.00 sec)

MySQL [challenge_lab]>
```

aws re/start

Conclusions

Amazon Relational Databases

Amazon Relational Databases offer scalable and reliable solutions for managing structured data, catering to diverse business needs.

Amazon RDS DB Instances

Amazon RDS DB Instances provide flexible configurations and high availability options, ensuring continuous access to databases.

Permitting connections to a DB instance

Permitting connections to a DB instance allows seamless communication between applications and databases, facilitating real-time data interactions.

DB Subnet Groups

DB Subnet Groups enable secure networking configurations, ensuring data privacy and compliance with regulatory requirements.

Interacting with a Database

Interacting with a database through applications or query tools enables data retrieval, updates, and analysis, empowering informed decision-making and efficient data management.

aws re/start

Cristhian Becerra

cristhian-becerra-espinoza

+51 951 634 354

cristhianbecerra99@gmail.com

Lima, Peru

aws re/start