TASK-1: CREATING AN RDS INSTANCE AND UPLOADING THR DATA TO THE RDS

Amazon Relational Database Service (Amazon RDS) is a relational database that is easier to set up, operate, and scale in the Cloud. It is cost-efficient, have resizable capacity for relational database and manages common database administration tasks. DB instance is the basic building block of Amazon RDS where we create our own databases. A DB instance is an isolated database environment in the AWS Cloud. Multiple user-created databases can be stored in a single DB instance. We can create and modify the DB instance by using the AWS CLI (Command Line Interface), the Amazon RDS API, or the AWS Management Console.

Each DB instance requires a DB engine. Amazon RDS currently supports 5 engines which are MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server DB engines. Each DB engine has its own supported features and properties. Additionally, each DB engine has a set of parameters in a DB parameter group that control the behavior of the databases that it manages.

STEPS TO CREATE AN AMAZON RDS DB INSTANCE.

STEP-1: LOGIN TO THE AWS CONSOLE.

First of all, we will login to the AWS console and navigate to the RDS dashboard. There we will click on **Create Database.**

STEP-2: CREATING AN RDS DB INSTANCE.

- Choose Standard Database creation method.
- Select any one of the given engine types. In this case we will select MySQL Engine.
- Select the version of MySQL that you want to use.
- Under templates section, we have three options:
  1) Production
  2) Dev/Test

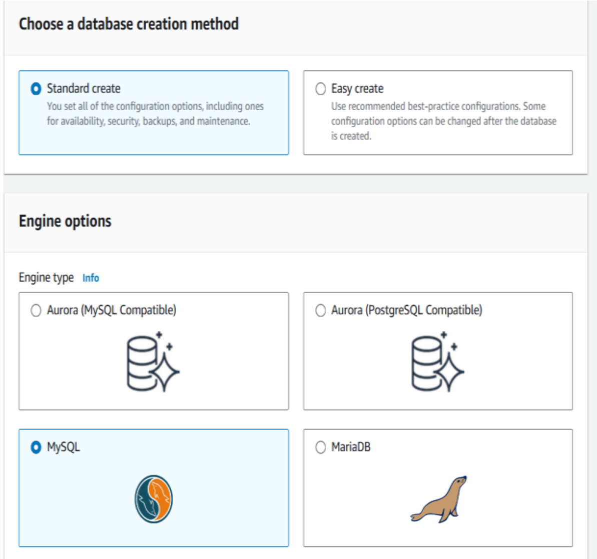
3) Free Tier

We will opt for free tier.

Under the settings section of the database, provide instance name, Database-1 and provide master username as admin and a password of your choice.

Select DB instance according to your needs. In free tier, by default, db.t2.micro will be chosen.

Rest leave everything as default, and click on create databases. See below images:



Choose a database creation method

● Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

○ Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

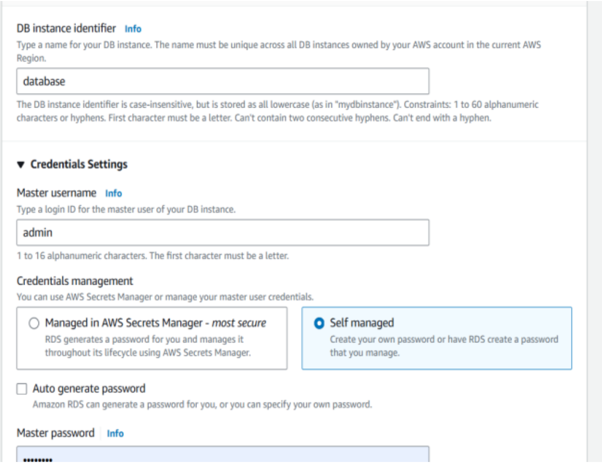
Engine type Info

○ Aurora (MySQL Compatible)

○ Aurora (PostgreSQL Compatible)

● MySQL

○ MariaDB



DB instance identifier Info
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

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

admin

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

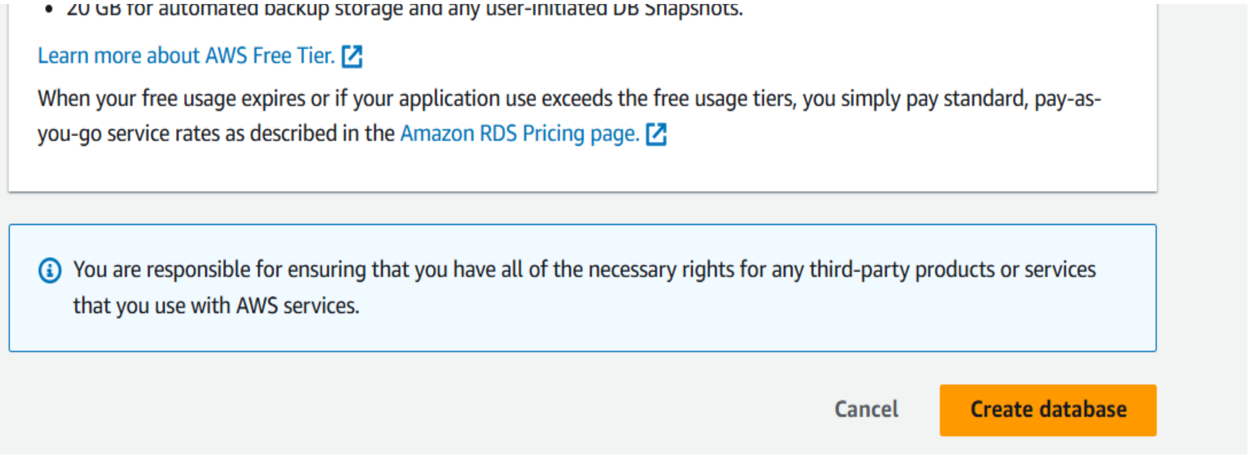
○ Managed in AWS Secrets Manager - *most secure*
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

● Self managed
Create your own password or have RDS create a password that you manage.

□ Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info

••••••••



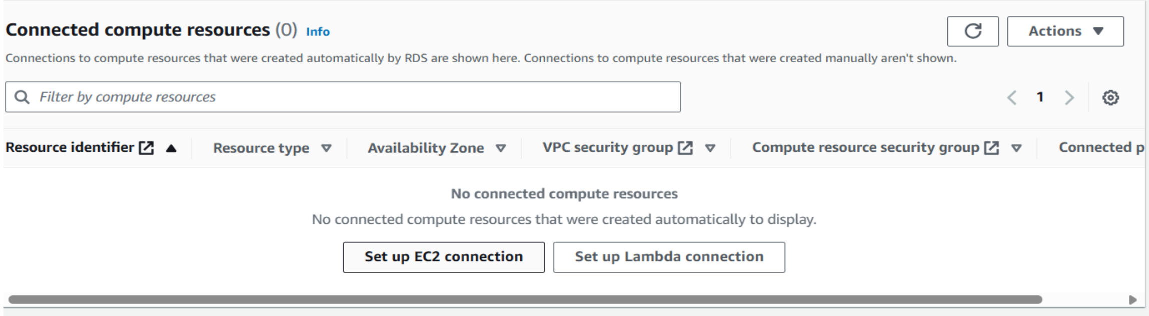
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

Learn more about AWS Free Tier.

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the Amazon RDS Pricing page.

You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

CONNECTING RDS WITH EC2 :



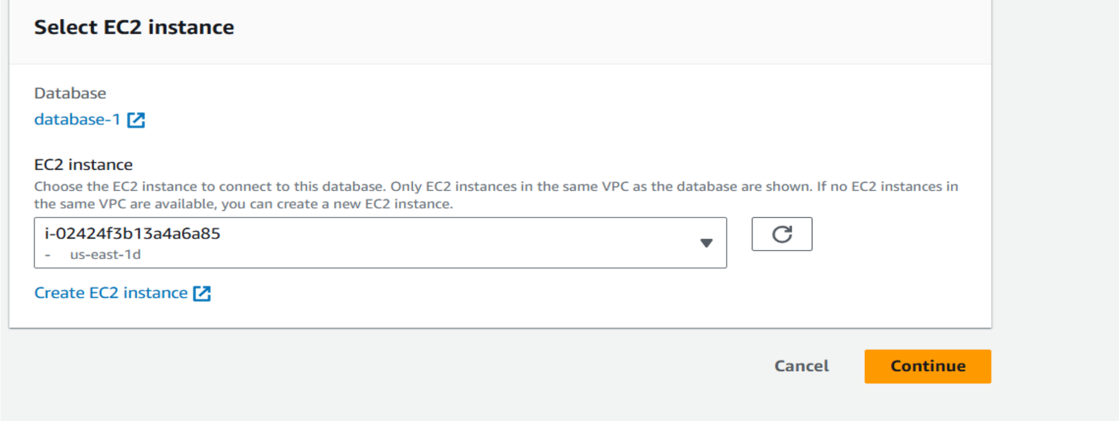
Connected compute resources (0) Info

Connections to compute resources that were created automatically by RDS are shown here. Connections to compute resources that were created manually aren't shown.

Resource identifier ▲	Resource type	Availability Zone	VPC security group	Compute resource security group	Connected p

No connected compute resources

No connected compute resources that were created automatically to display.

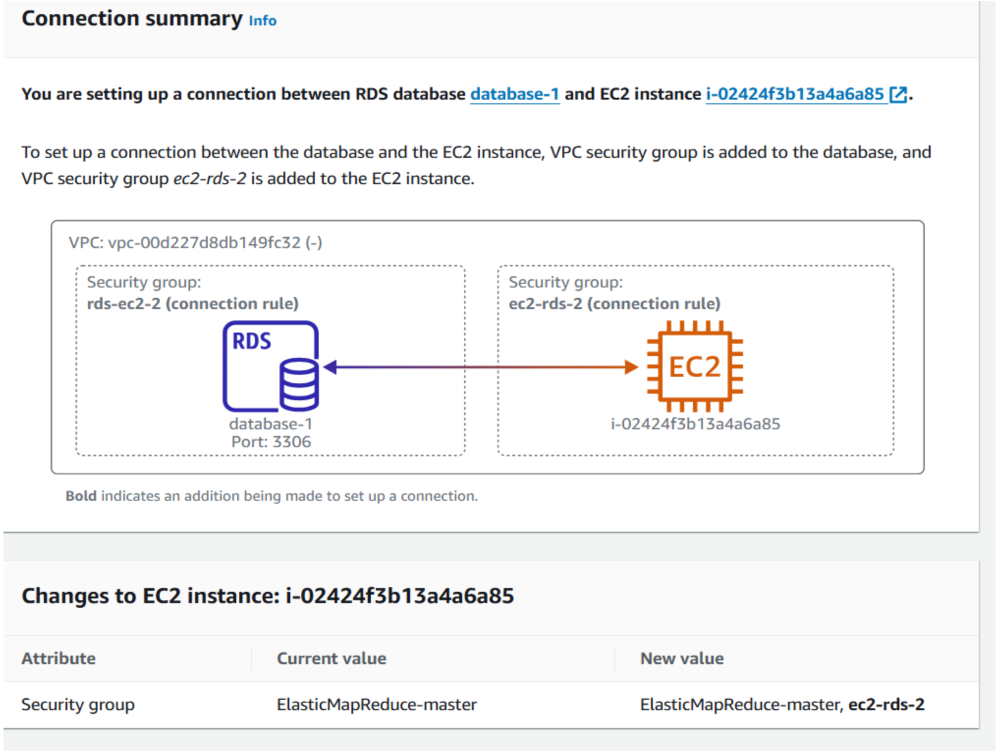


Select EC2 instance

Database
database-1

EC2 instance
Choose the EC2 instance to connect to this database. Only EC2 instances in the same VPC as the database are shown. If no EC2 instances in the same VPC are available, you can create a new EC2 instance.

i-02424f3b13a4a6a85
- us-east-1d



Connection summary Info

You are setting up a connection between RDS database database-1 and EC2 instance i-02424f3b13a4a6a85.

To set up a connection between the database and the EC2 instance, VPC security group is added to the database, and VPC security group *ec2-rds-2* is added to the EC2 instance.

VPC: vpc-00d227d8db149fc32 (-)
Security group:
rds-ec2-2 (connection rule)
RDS
database-1
Port: 3306
Security group:
ec2-rds-2 (connection rule)
EC2
i-02424f3b13a4a6a85

Bold indicates an addition being made to set up a connection.

Changes to EC2 instance: i-02424f3b13a4a6a85

Attribute	Current value	New value
Security group	ElasticMapReduce-master	ElasticMapReduce-master, **ec2-rds-2**

STEPS TO LINK RDS DATABASE WITH MySQL WORKBENCH.

- Open MySQL Workbench and click on databases icon
- There we will name the connection, and choose the connection method.
- In Parameters section, we will enter the hostname as it is mentioned in RDS instance and Port as 3306.
- Next, we will enter user name and password.
- Click OK button and MySQL Workbench is connected to the database instance of RDS.

UPLOADING DATA INTO RDS:

- On MySQL workbench, we write down SQL Queries to upload the data.

```
CREATE DATABASE assignment
USE assignment

CREATE TABLE TaxiTrips(
  VendorID INT,
  tpep_pickup_datetime DATETIME,
  tpep_dropoff_datetime DATETIME,
  Passenger_count INT,
  Trip_distance FLOAT,
  RateCodeID INT,
  Store_and_fwd_flag CHAR(1),
  PULocationID INT,
  DOLocationID INT,
  Payment_type INT,
  Fare_amount FLOAT,
  Extra FLOAT,
  MTA_tax FLOAT,
  Improvement_surcharge FLOAT,
  Tip_amount FLOAT,
  Tolls_amount FLOAT,
  Total_amount FLOAT,
   Airport_fee FLOAT,
   PRIMARY KEY ( tpep_pickup_datetime, tpep_dropoff_datetime)
);
LOAD DATA LOCAL INFILE "C:/Users/Admin/Downloads/yellow_tripdata_2017-01.csv"
INTO TABLE TaxiTrips
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES;

LOAD DATA LOCAL INFILE "C:/Users/Admin/Downloads/yellow_tripdata_2017-02.csv"
INTO TABLE TaxiTrips
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES;

SELECT * FROM TaxiTrips;
```