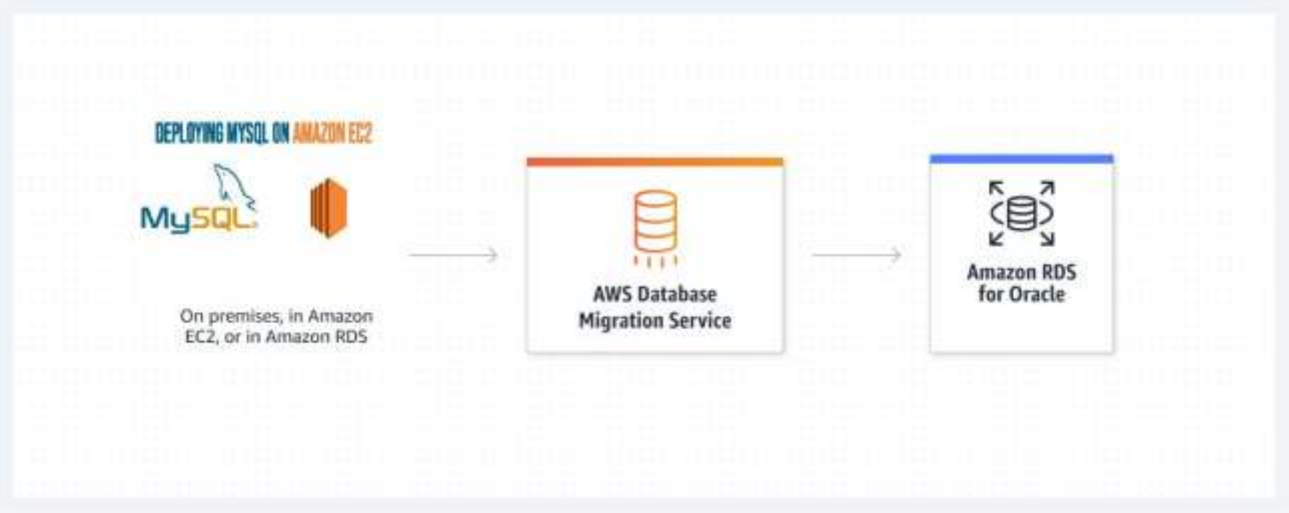
MIGRATING DATABASE FROM DB-ON EC2 TO AWS RDS USING DATABASE MIGRATION SERVICE


DEPLOYING MYSQL ON AMAZON EC2
MySQL
On premises, in Amazon EC2, or in Amazon RDS
AWS Database Migration Service
Amazon RDS for Oracle

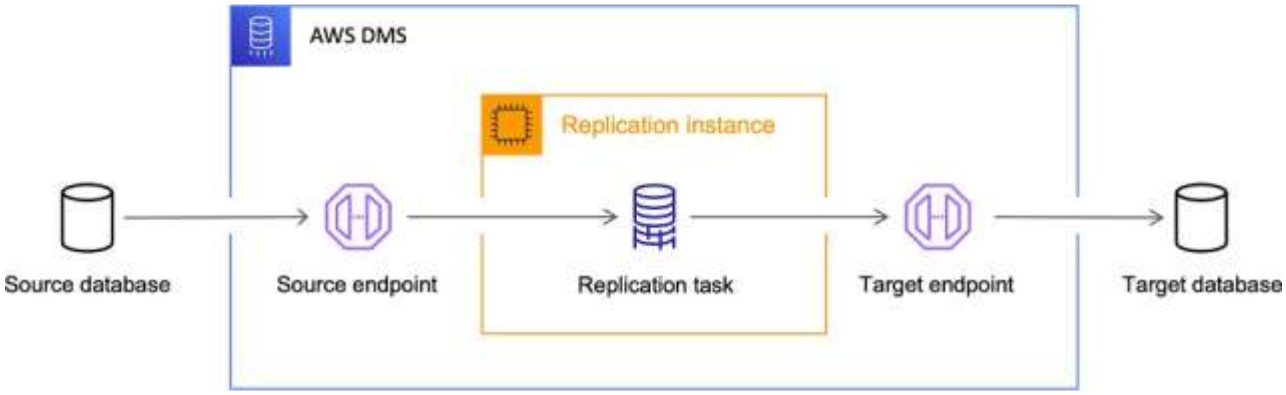
INTRODUCTION

AWS Database Migration Service (AWS DMS) is a cloud service that makes it possible to migrate relational databases, data warehouses, NoSQL databases, and other types of data stores. You can use AWS DMS to migrate your data into the AWS Cloud or between combinations of cloud and on-premises setups.

With AWS DMS, you can discover your source data stores, convert your source schemas, and migrate your data.

- To discover your source data infrastructure, you can use DMS Fleet Advisor. This service collects data from your on-premises database and analytic servers, and builds an inventory of servers, databases, and schemas that you can migrate to the AWS Cloud.
- To migrate to a different database engine, you can use DMS Schema Conversion. This service automatically assesses and converts your source schemas to a new target engine. Alternatively, you can download the AWS Schema Conversion Tool (AWS SCT) to your local PC to convert your source schemas.
- After you convert your source schemas and apply the converted code to your target database, you can use AWS DMS to migrate your data. You can perform one-time migrations or replicate ongoing changes to keep sources and targets in sync. Because AWS DMS is a part of the AWS Cloud, you get the cost efficiency, speed to market, security, and flexibility that AWS services offer.

At a basic level, AWS DMS is a server in the AWS Cloud that runs replication software. You create a source and target connection to tell AWS DMS where to extract data from and where to load it. Next, you schedule a task that runs on this server to move your data. AWS DMS creates the tables and associated primary keys if they don't exist on the target. You can create the target tables yourself if you prefer. Or you can use AWS Schema Conversion Tool (AWS SCT) to create some or all of the target tables, indexes, views, triggers, and so on.



OVERVIEW

AWS Database Migration Service (AWS DMS) is a service provided by Amazon Web Services that helps you migrate databases to AWS quickly and securely. It supports a wide range of source and target databases and is widely used for database migrations, continuous data replication, and database modernization.

Key Features of AWS DMS:

1. Database Migration:
   - One-time migration: Move data from one database to another, typically used for database migrations.
   - Ongoing replication: Sync data between the source and target databases in real-time. This allows for continuous replication as changes happen in the source database.
2. Supports Multiple Databases:
   - Source databases: Includes databases like Amazon RDS, Amazon Redshift, Oracle, MySQL, PostgreSQL, SQL Server, and others.
   - Target databases: Amazon RDS, Amazon Redshift, Amazon Aurora, Amazon DynamoDB, and various other database engines.
3. Data Transformation:
   - AWS DMS provides basic transformation capabilities, such as changing the format of data or applying simple transformations during the migration process.
4. Minimal Downtime:
   - DMS supports "cutover" migrations, allowing databases to be migrated with minimal downtime. This is particularly important for applications that require high availability.
5. Replication Instances:

   - The service uses replication instances to manage the migration. You can choose the size of the replication instance based on your workload requirements.

6. Monitoring and Reporting:
   - AWS DMS integrates with Amazon CloudWatch, allowing you to monitor the progress of your migration in real-time. You can view logs, metrics, and performance data to ensure a smooth migration.
7. Automated Failover:
   - DMS can automatically handle failover, which means that in case of issues, it will attempt to reroute replication traffic to available sources.
8. Data Validation:
   - The service includes an option for data validation after the migration, ensuring that the data is accurate and consistent between the source and target.
9. Support for Heterogeneous Migrations:
   - AWS DMS supports migrating data between different types of database engines (e.g., Oracle to MySQL or SQL Server to Amazon Aurora). This is known as heterogeneous database migration.

Use Cases for AWS DMS:

- Database Migration: Moving an on-premises database or another cloud database to AWS.
- Continuous Data Replication: Replicating data between databases for high availability, disaster recovery, or reporting purposes.
- Database Consolidation: Merging multiple databases into a single target database.
- Data Transformation: Migrate and transform data during the migration process.

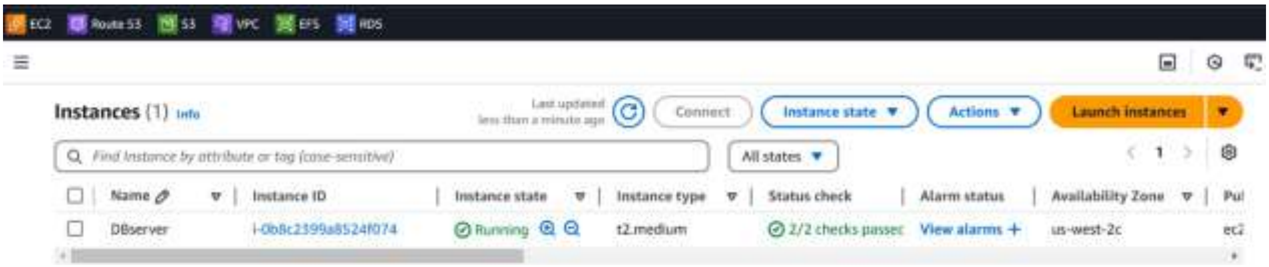
Workflow:

1. Setup Source and Target Database: Connect your source and target databases to AWS DMS.
2. Create a Replication Instance: Set up an instance that manages the data migration.
3. Create Migration Tasks: Configure the migration tasks, such as one-time migration or ongoing replication.
4. Monitor Progress: Track the migration process in real-time using AWS DMS monitoring tools.

SOURCE EC2 DB Creation

Prerequisite: Launch a new EC2 instance with Amazon Linux and run the following commands after connecting to the server.

Make sur that inbound rule to allow TCP traffic on port 3306 (default MySQL port) from your IP or a range of IPs (e.g., your home/office IP).



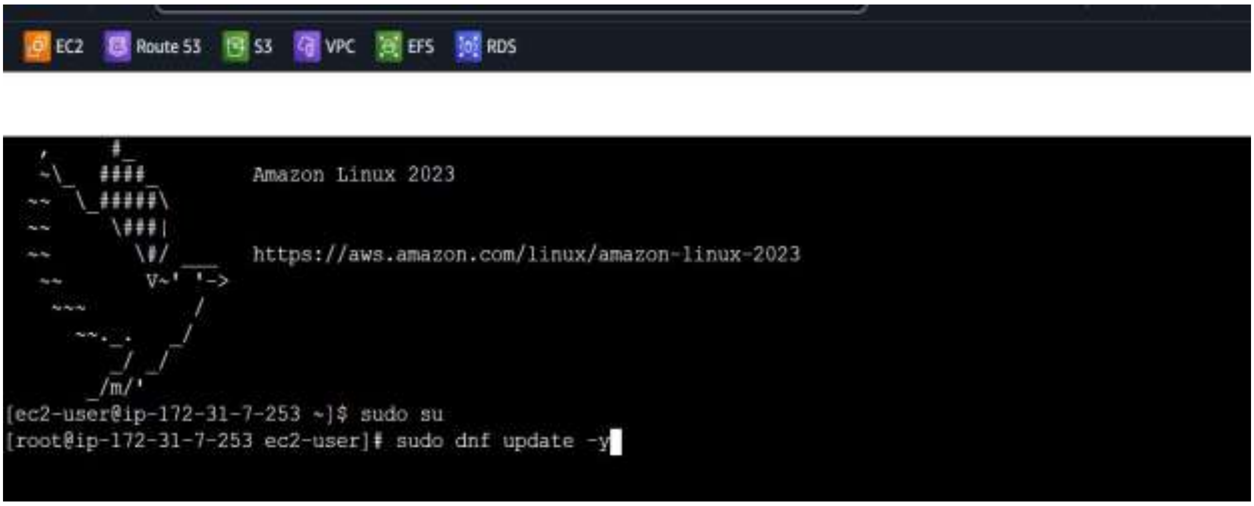
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pul
□	DBserver	i-0b8c2399a8524f074	Running	t2.medium	2/2 checks passec	View alarms +	us-west-2c	ec2

Install MySQL Server on Amazon Linux

1. Update the package index

sudo su -

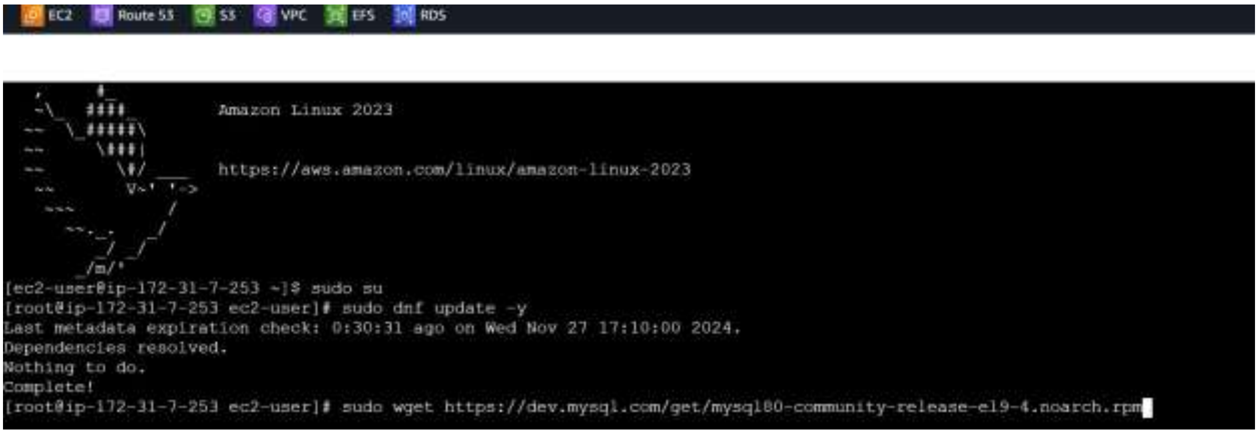
sudo dnf update -y



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-7-253 ~]$ sudo su
[root@ip-172-31-7-253 ec2-user]# sudo dnf update -y
```

2. Download the MySQL 8.0 community release package. This ensures you get the correct repository configuration for MySQL 8.0:

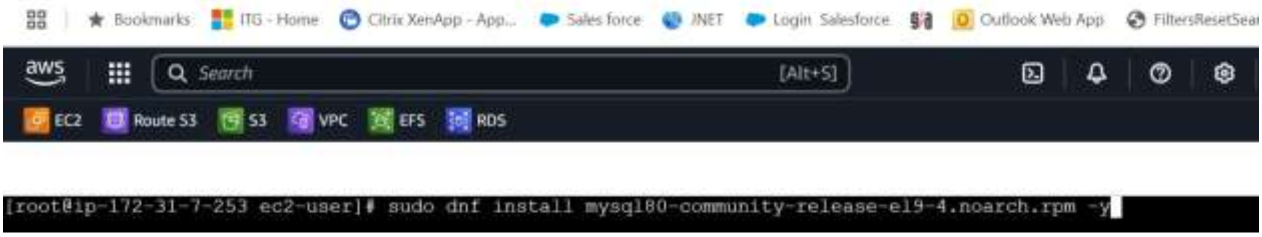
sudo wget https://dev.mysql.com/get/mysql80-community-release-el9-4.noarch.rpm



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-7-253 ~]$ sudo su
[root@ip-172-31-7-253 ec2-user]# sudo dnf update -y
Last metadata expiration check: 0:30:31 ago on Wed Nov 27 17:10:00 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-7-253 ec2-user]# sudo wget https://dev.mysql.com/get/mysql80-community-release-el9-4.noarch.rpm
```

3. Install the MySQL community release package. This adds the MySQL repository to your system:

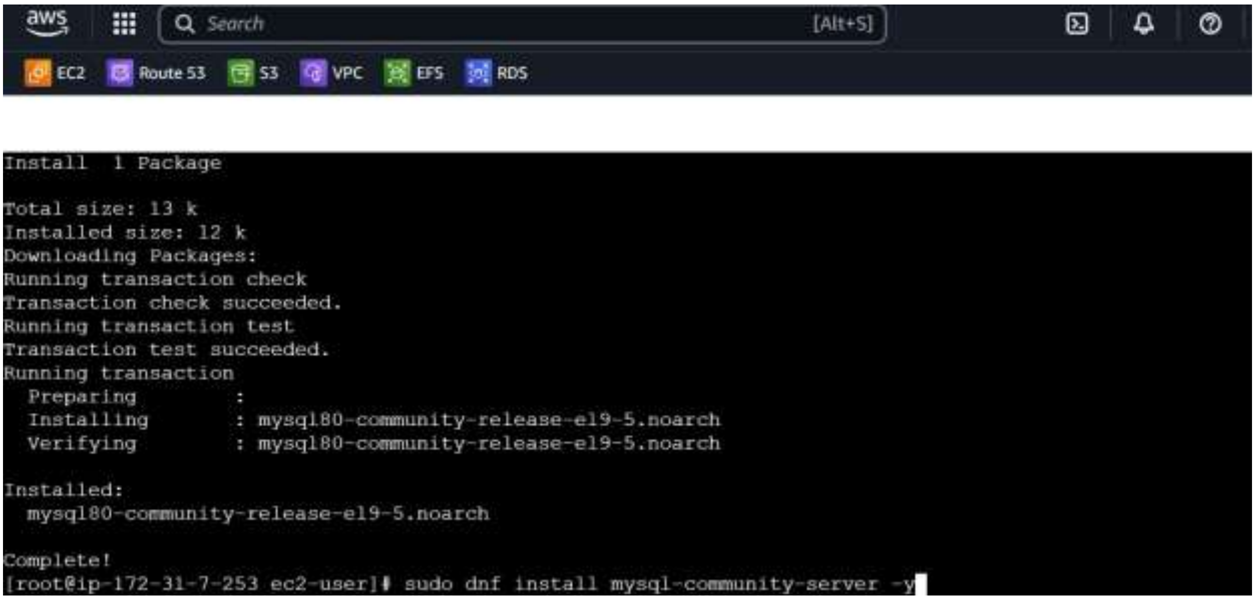
sudo dnf install mysql80-community-release-el9-4.noarch.rpm -y



```
[root@ip-172-31-7-253 ec2-user]# sudo dnf install mysql80-community-release-el9-4.noarch.rpm -y
```

4. Install MySQL server

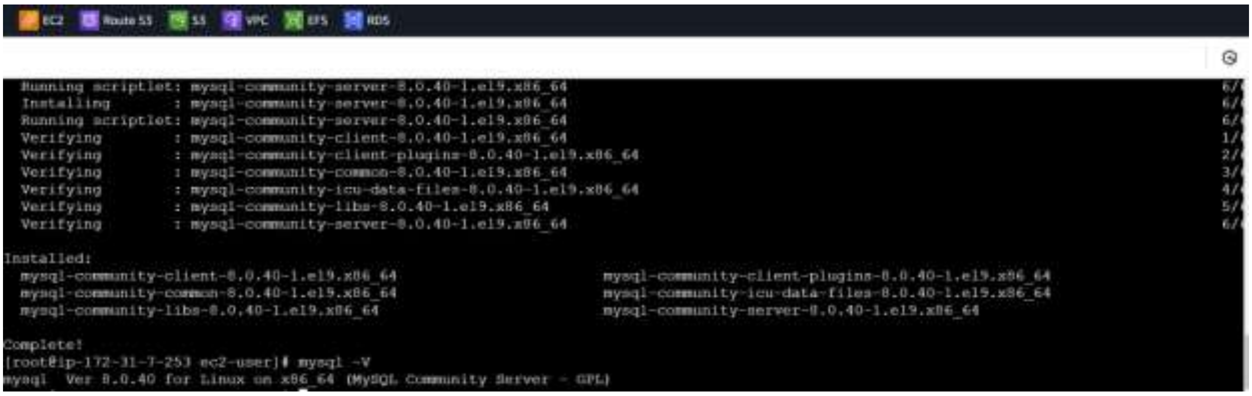
sudo dnf install mysql-community-server -y



```
Install  1 Package

Total size: 13 k
Installed size: 12 k
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing         :
  Installing        : mysql80-community-release-el9-5.noarch
  Verifying         : mysql80-community-release-el9-5.noarch

Installed:
  mysql80-community-release-el9-5.noarch

Complete!
[root@ip-172-31-7-253 ec2-user]# sudo dnf install mysql-community-server -y
```

5. Verify MySQL installation. Check the installed MySQL version:

mysql -V



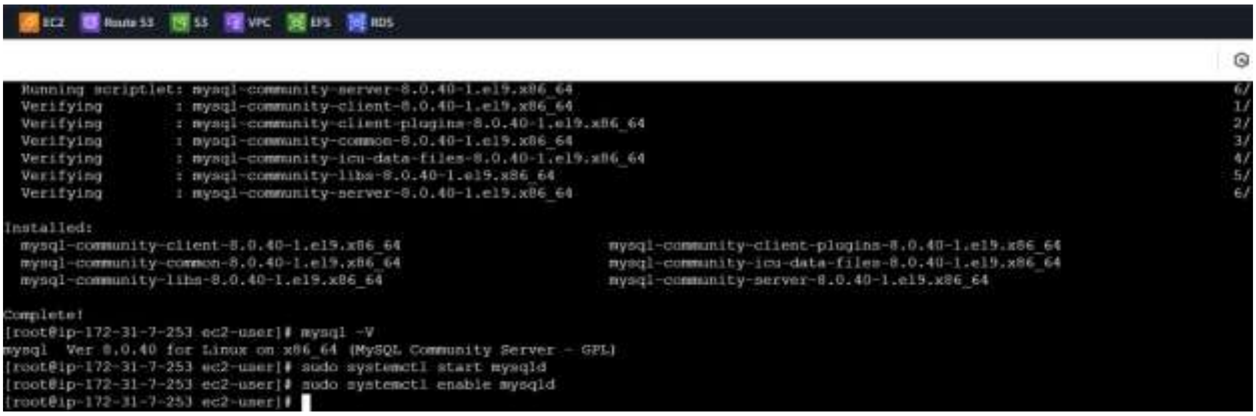
```
  Running scriptlet: mysql-community-server-8.0.40-1.el9.x86_64                                  6/
  Installing        : mysql-community-server-8.0.40-1.el9.x86_64                                  6/
  Running scriptlet: mysql-community-server-8.0.40-1.el9.x86_64                                  6/
  Verifying         : mysql-community-client-8.0.40-1.el9.x86_64                                  1/
  Verifying         : mysql-community-client-plugins-8.0.40-1.el9.x86_64                          2/
  Verifying         : mysql-community-common-8.0.40-1.el9.x86_64                                  3/
  Verifying         : mysql-community-icu-data-files-8.0.40-1.el9.x86_64                          4/
  Verifying         : mysql-community-libs-8.0.40-1.el9.x86_64                                    5/
  Verifying         : mysql-community-server-8.0.40-1.el9.x86_64                                  6/

Installed:
  mysql-community-client-8.0.40-1.el9.x86_64                 mysql-community-client-plugins-8.0.40-1.el9.x86_64
  mysql-community-common-8.0.40-1.el9.x86_64                 mysql-community-icu-data-files-8.0.40-1.el9.x86_64
  mysql-community-libs-8.0.40-1.el9.x86_64                   mysql-community-server-8.0.40-1.el9.x86_64

Complete!
[root@ip-172-31-7-253 ec2-user]# mysql -V
mysql  Ver 8.0.40 for Linux on x86_64 (MySQL Community Server - GPL)
```

6. Start and Enable MySQL Service

Start the MySQL service and configure it to run on boot:

sudo systemctl start mysqld

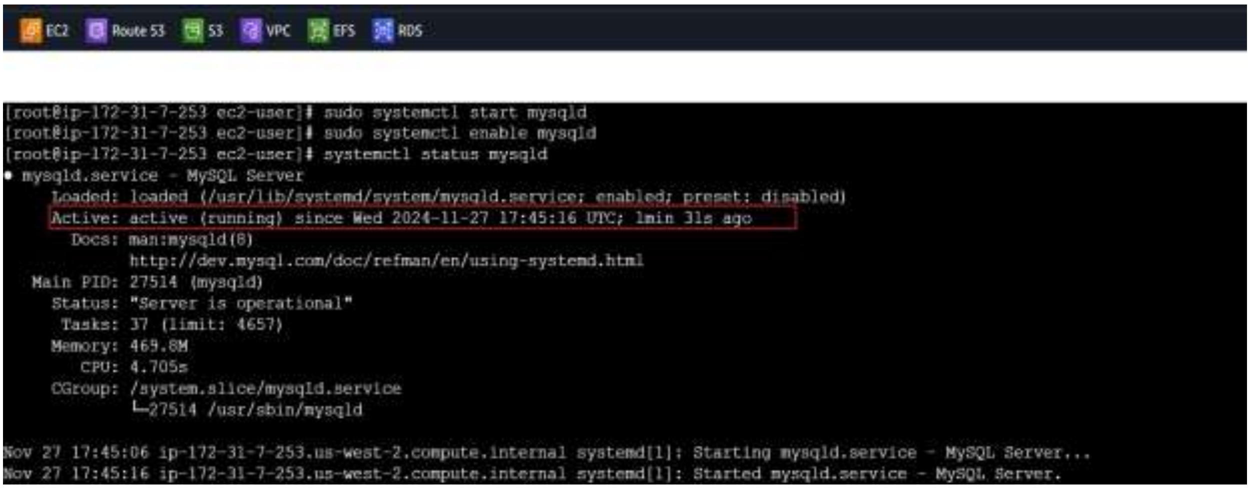
sudo systemctl enable mysqld



```
  Running scriptlet: mysql-community-server-8.0.40-1.el9.x86_64                                                     6/
  Verifying        : mysql-community-client-8.0.40-1.el9.x86_64                                                     1/
  Verifying        : mysql-community-client-plugins-8.0.40-1.el9.x86_64                                             2/
  Verifying        : mysql-community-common-8.0.40-1.el9.x86_64                                                     3/
  Verifying        : mysql-community-icu-data-files-8.0.40-1.el9.x86_64                                             4/
  Verifying        : mysql-community-libs-8.0.40-1.el9.x86_64                                                       5/
  Verifying        : mysql-community-server-8.0.40-1.el9.x86_64                                                     6/

Installed:
  mysql-community-client-8.0.40-1.el9.x86_64                        mysql-community-client-plugins-8.0.40-1.el9.x86_64
  mysql-community-common-8.0.40-1.el9.x86_64                        mysql-community-icu-data-files-8.0.40-1.el9.x86_64
  mysql-community-libs-8.0.40-1.el9.x86_64                          mysql-community-server-8.0.40-1.el9.x86_64

Complete!
[root@ip-172-31-7-253 ec2-user]# mysql -V
mysql  Ver 8.0.40 for Linux on x86_64 (MySQL Community Server - GPL)
[root@ip-172-31-7-253 ec2-user]# sudo systemctl start mysqld
[root@ip-172-31-7-253 ec2-user]# sudo systemctl enable mysqld
[root@ip-172-31-7-253 ec2-user]# 
```

Check MySQL service status

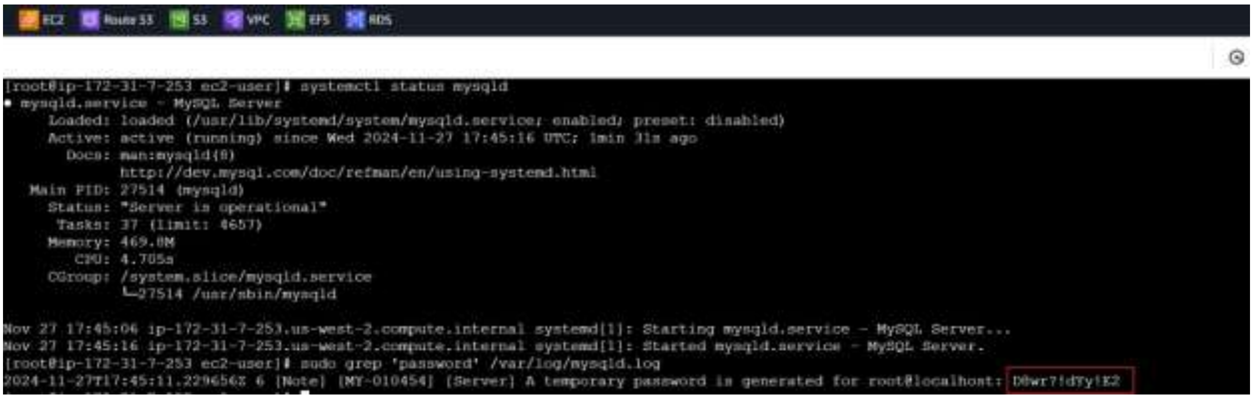
systemctl status mysqld



```
[root@ip-172-31-7-253 ec2-user]# sudo systemctl start mysqld
[root@ip-172-31-7-253 ec2-user]# sudo systemctl enable mysqld
[root@ip-172-31-7-253 ec2-user]# systemctl status mysqld
● mysqld.service - MySQL Server
     Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; preset: disabled)
     Active: active (running) since Wed 2024-11-27 17:45:16 UTC; 1min 31s ago
       Docs: man:mysqld(8)
             http://dev.mysql.com/doc/refman/en/using-systemd.html
   Main PID: 27514 (mysqld)
     Status: "Server is operational"
      Tasks: 37 (limit: 4657)
     Memory: 469.8M
        CPU: 4.705s
     CGroup: /system.slice/mysqld.service
             └─27514 /usr/sbin/mysqld

Nov 27 17:45:06 ip-172-31-7-253.us-west-2.compute.internal systemd[1]: Starting mysqld.service - MySQL Server...
Nov 27 17:45:16 ip-172-31-7-253.us-west-2.compute.internal systemd[1]: Started mysqld.service - MySQL Server.
```

7. Retrieve the Temporary Password. MySQL generates a temporary root password during installation:

sudo grep 'password' /var/log/mysqld.log

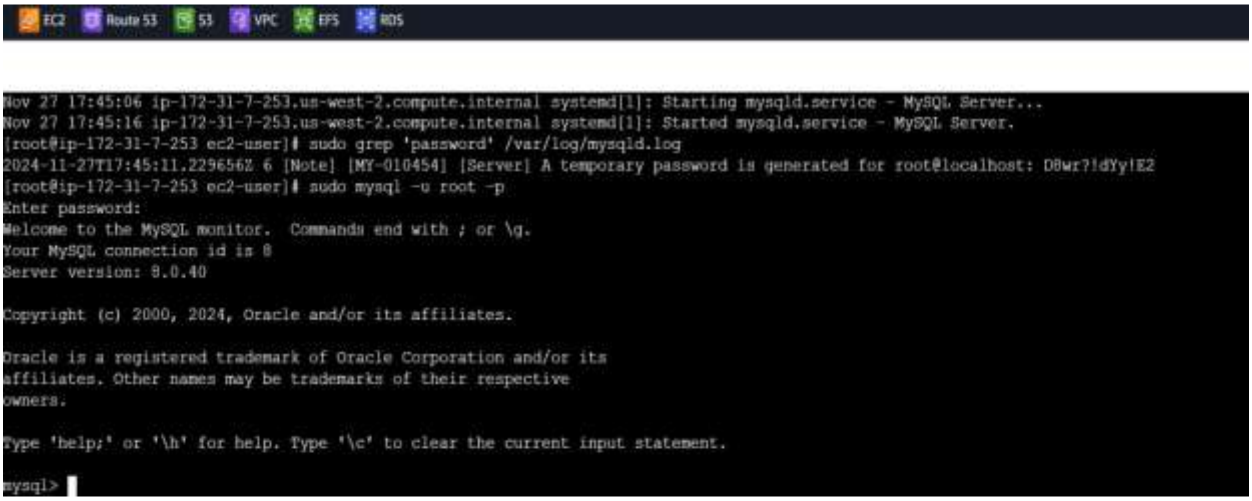


```
[root@ip-172-31-7-253 ec2-user]# systemctl status mysqld
● mysqld.service - MySQL Server
     Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; preset: disabled)
     Active: active (running) since Wed 2024-11-27 17:45:16 UTC; 1min 31s ago
       Docs: man:mysqld(8)
             http://dev.mysql.com/doc/refman/en/using-systemd.html
   Main PID: 27514 (mysqld)
     Status: "Server is operational"
      Tasks: 37 (limit: 4657)
     Memory: 469.0M
        CPU: 4.705s
     CGroup: /system.slice/mysqld.service
             └─27514 /usr/sbin/mysqld

Nov 27 17:45:06 ip-172-31-7-253.us-west-2.compute.internal systemd[1]: Starting mysqld.service - MySQL Server...
Nov 27 17:45:16 ip-172-31-7-253.us-west-2.compute.internal systemd[1]: Started mysqld.service - MySQL Server.
[root@ip-172-31-7-253 ec2-user]# sudo grep 'password' /var/log/mysqld.log
2024-11-27T17:45:11.229656Z 6 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: D0wr7!dYy!E2
```

8. Log in to MySQL Using the Temporary Password

sudo mysql -u root -p

Enter the temporary password when prompted.

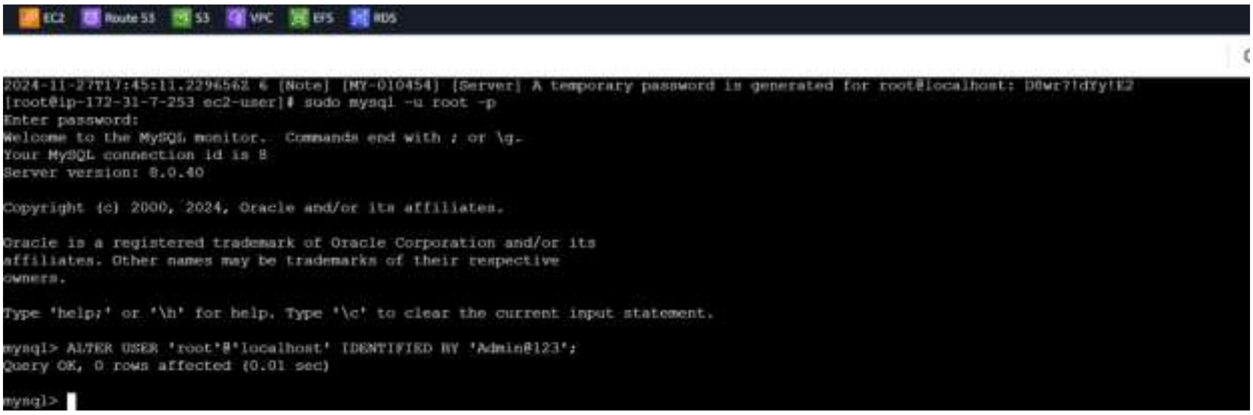


```
Nov 27 17:45:06 ip-172-31-7-253.us-west-2.compute.internal systemd[1]: Starting mysqld.service - MySQL Server...
Nov 27 17:45:16 ip-172-31-7-253.us-west-2.compute.internal systemd[1]: Started mysqld.service - MySQL Server.
[root@ip-172-31-7-253 ec2-user]# sudo grep 'password' /var/log/mysqld.log
2024-11-27T17:45:11.229656Z 6 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: D0wr7!dYy!E2
[root@ip-172-31-7-253 ec2-user]# sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.40

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> 
```

9. Change the Root Password

After logging in, set a new, strong root password:

ALTER USER 'root'@'localhost' IDENTIFIED BY 'Admin@123';



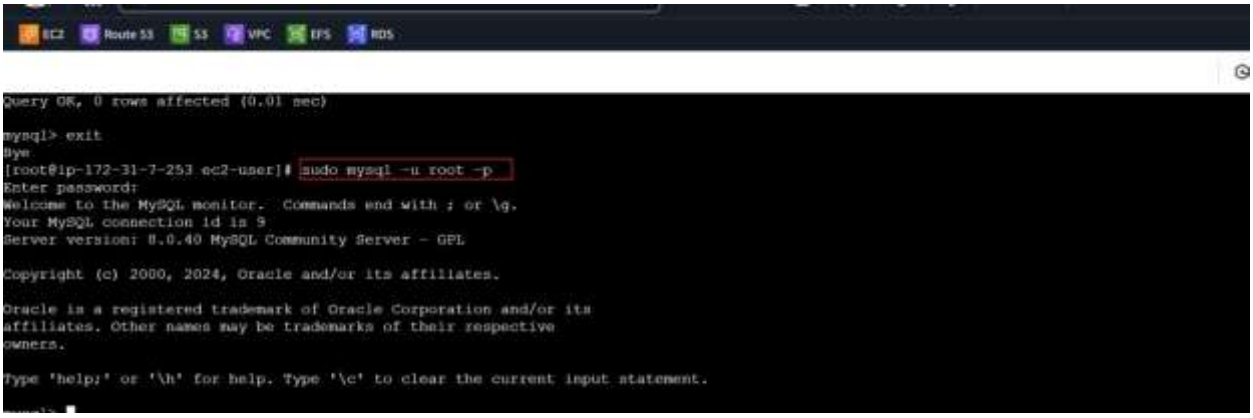
```
2024-11-27T17:45:11.229656Z 6 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: D0wr7!dYy!E2
[root@ip-172-31-7-253 ec2-user]# sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.40

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'Admin@123';
Query OK, 0 rows affected (0.01 sec)

mysql> 
```

10. Exit and Re-login with the New Password

exit

sudo mysql -u root -p

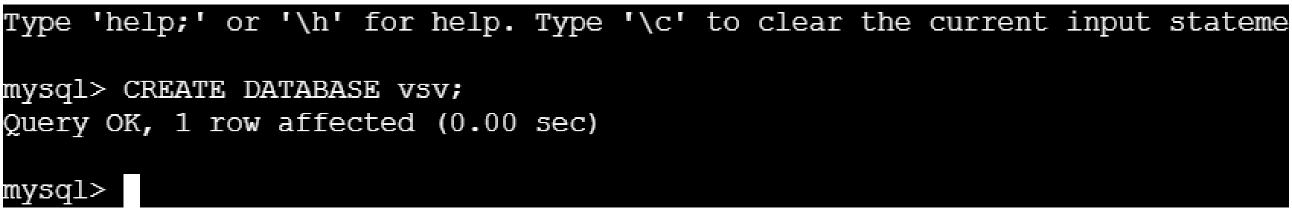
Enter the new password.



```
Query OK, 0 rows affected (0.01 sec)

mysql> exit
Bye
[root@ip-172-31-7-253 ec2-user]# sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.40 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
```

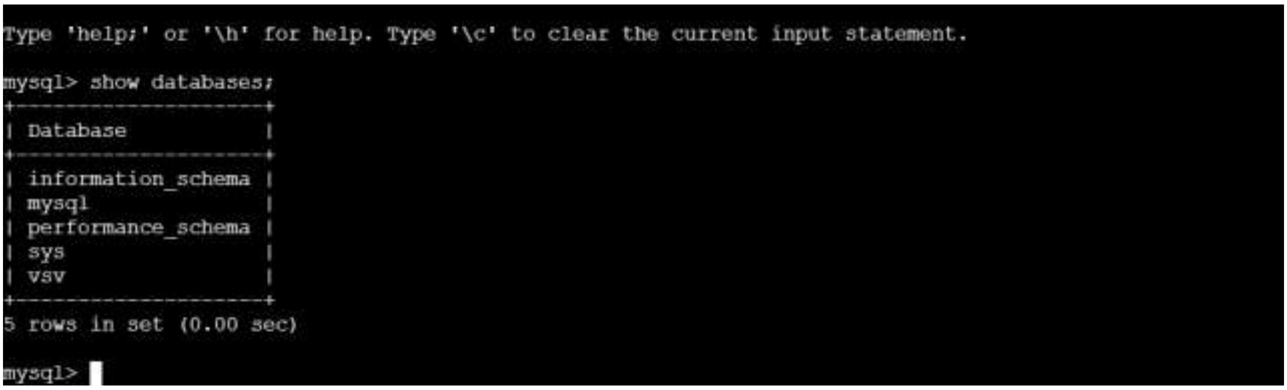
11. Create a Database

CREATE DATABASE vsv;



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

mysql> CREATE DATABASE vsv;
Query OK, 1 row affected (0.00 sec)

mysql>
```

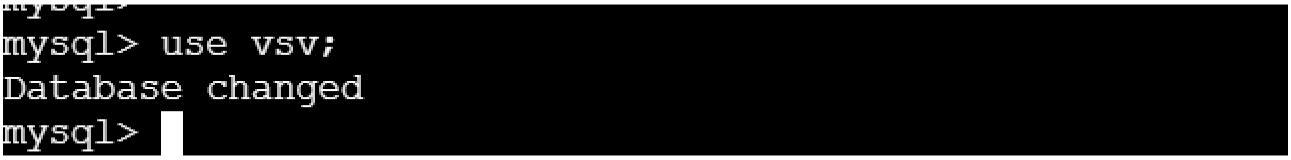
Show databases;



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

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| vsv                |
+--------------------+
5 rows in set (0.00 sec)

mysql>
```

- Change to your database

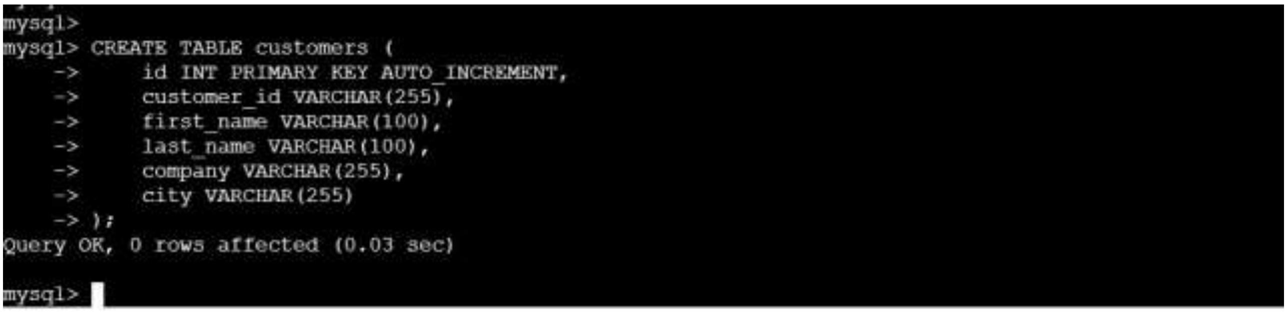
  Use vsv;



```
mysql>
mysql> use vsv;
Database changed
mysql>
```

- Crete a table with below query

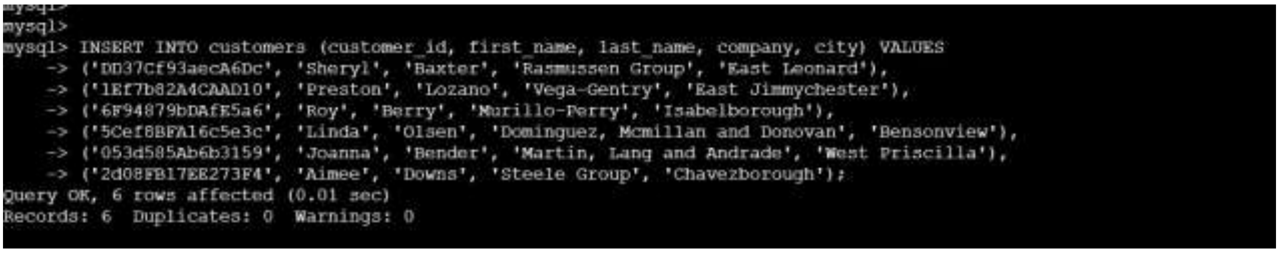
```
CREATE TABLE customers (
  id INT PRIMARY KEY AUTO_INCREMENT,
  customer_id VARCHAR(255),
  first_name VARCHAR(100),
  last_name VARCHAR(100),
  company VARCHAR(255),
  city VARCHAR(255)
);
```



```
mysql>
mysql> CREATE TABLE customers (
    ->     id INT PRIMARY KEY AUTO_INCREMENT,
    ->     customer_id VARCHAR(255),
    ->     first_name VARCHAR(100),
    ->     last_name VARCHAR(100),
    ->     company VARCHAR(255),
    ->     city VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql>
```

- Insert the few records into above table

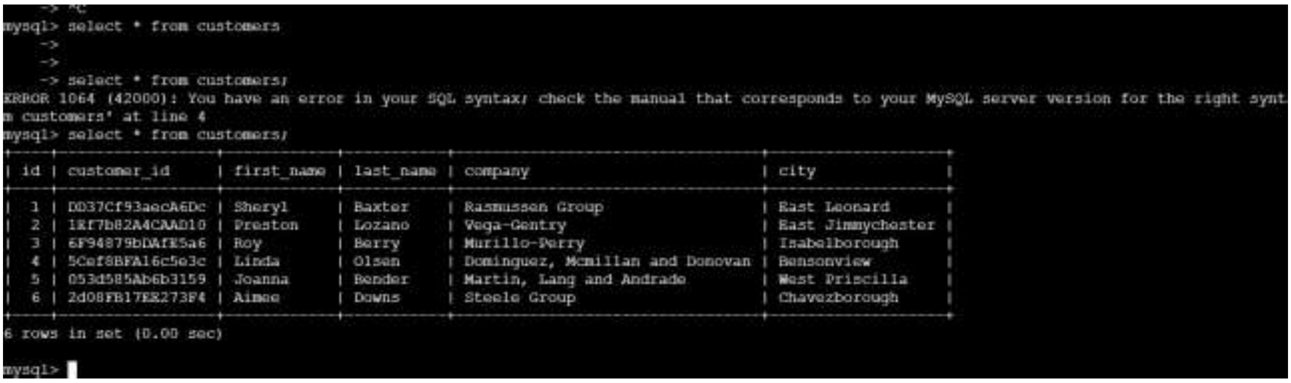
```
INSERT INTO customers (customer_id, first_name, last_name, company, city) VALUES
('DD37Cf93aecA6Dc', 'Sheryl', 'Baxter', 'Rasmussen Group', 'East Leonard'),
('1Ef7b82A4CAAD10', 'Preston', 'Lozano', 'Vega-Gentry', 'East Jimmychester'),
('6F94879bDAfE5a6', 'Roy', 'Berry', 'Murillo-Perry', 'Isabelborough'),
('5Cef8BFA16c5e3c', 'Linda', 'Olsen', 'Dominguez, Mcmillan and Donovan', 'Bensonview'),
('053d585Ab6b3159', 'Joanna', 'Bender', 'Martin, Lang and Andrade', 'West Priscilla'),
('2d08FB17EE273F4', 'Aimee', 'Downs', 'Steele Group', 'Chavezborough');
```



```
mysql>
mysql>
mysql> INSERT INTO customers (customer_id, first_name, last_name, company, city) VALUES
    -> ('DD37Cf93aecA6Dc', 'Sheryl', 'Baxter', 'Rasmussen Group', 'East Leonard'),
    -> ('1Ef7b82A4CAAD10', 'Preston', 'Lozano', 'Vega-Gentry', 'East Jimmychester'),
    -> ('6F94879bDAfE5a6', 'Roy', 'Berry', 'Murillo-Perry', 'Isabelborough'),
    -> ('5Cef8BFA16c5e3c', 'Linda', 'Olsen', 'Dominguez, Mcmillan and Donovan', 'Bensonview'),
    -> ('053d585Ab6b3159', 'Joanna', 'Bender', 'Martin, Lang and Andrade', 'West Priscilla'),
    -> ('2d08FB17EE273F4', 'Aimee', 'Downs', 'Steele Group', 'Chavezborough');
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

- Check the records in the tale

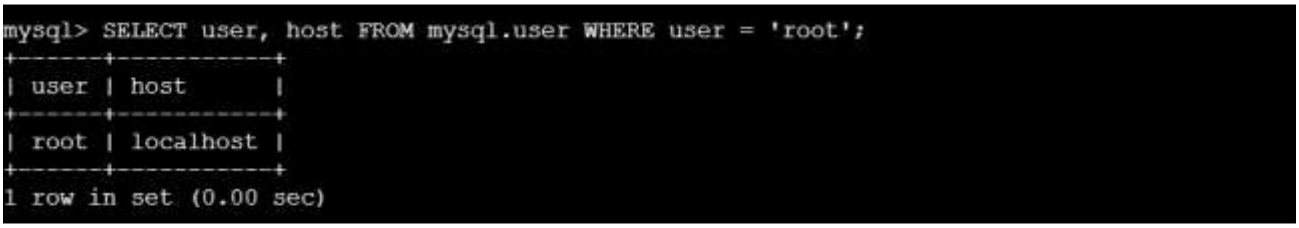
select * from vsv.customers;



```
    -> ^C
mysql> select * from customers
    ->
    ->
    -> select * from customers;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right synt
m customers' at line 4
mysql> select * from customers;
+----+-----------------+------------+-----------+---------------------------------+-------------------+
| id | customer_id     | first_name | last_name | company                         | city              |
+----+-----------------+------------+-----------+---------------------------------+-------------------+
|  1 | DD37Cf93aecA6Dc | Sheryl     | Baxter    | Rasmussen Group                 | East Leonard      |
|  2 | 1Ef7b82A4CAAD10 | Preston    | Lozano    | Vega-Gentry                     | East Jimmychester |
|  3 | 6F94879bDAfE5a6 | Roy        | Berry     | Murillo-Perry                   | Isabelborough     |
|  4 | 5Cef8BFA16c5e3c | Linda      | Olsen     | Dominguez, Mcmillan and Donovan | Bensonview        |
|  5 | 053d585Ab6b3159 | Joanna     | Bender    | Martin, Lang and Andrade        | West Priscilla    |
|  6 | 2d08FB17EE273F4 | Aimee      | Downs     | Steele Group                    | Chavezborough     |
+----+-----------------+------------+-----------+---------------------------------+-------------------+
6 rows in set (0.00 sec)

mysql> 
```

- Grant the permissions to you database
- Check Existing Users: First, check if the root user exists for '%' (i.e., any host) by running the following query:

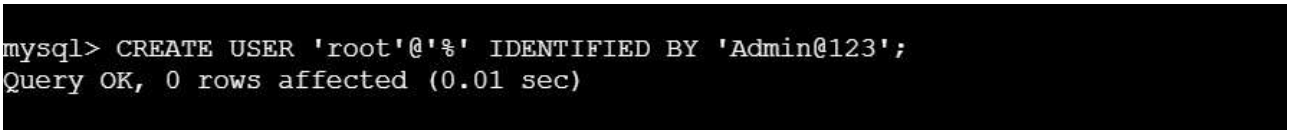
SELECT user, host FROM mysql.user WHERE user = 'root';



```
mysql> SELECT user, host FROM mysql.user WHERE user = 'root';
+------+-----------+
| user | host      |
+------+-----------+
| root | localhost |
+------+-----------+
1 row in set (0.00 sec)
```

- Create the User if Not Exists: If the user 'root'@'%' doesn't exist, create it using:

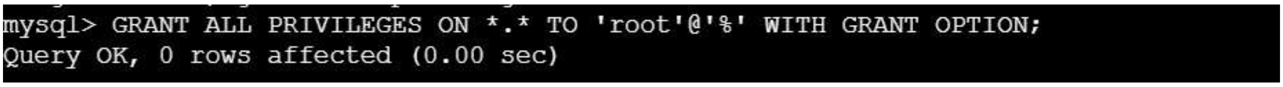
CREATE USER 'root'@'%' IDENTIFIED BY 'Admin@123';



```
mysql> CREATE USER 'root'@'%' IDENTIFIED BY 'Admin@123';
Query OK, 0 rows affected (0.01 sec)
```

- Grant Privileges: After creating the user, grant all privileges:

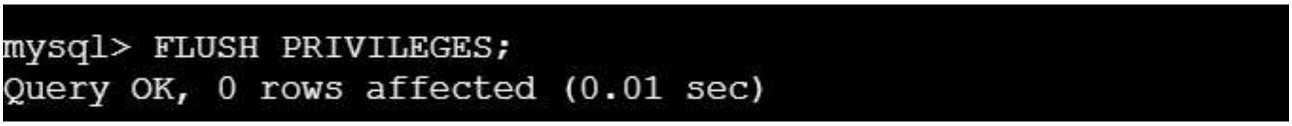
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' WITH GRANT OPTION;



```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)
```

- Apply the changes

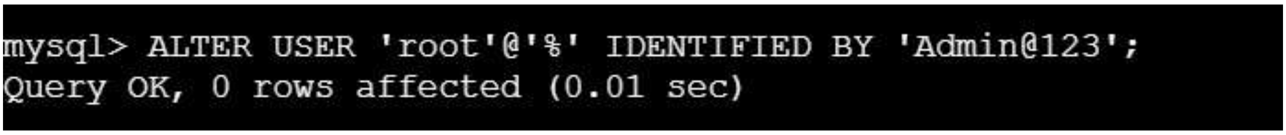
FLUSH PRIVILEGES;



```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

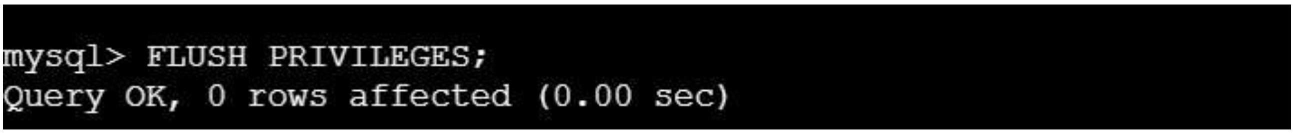
- is used to **change the password** of the root user in MySQL for connections from any host.

ALTER USER 'root'@'%' IDENTIFIED BY 'Admin@123';



```
mysql> ALTER USER 'root'@'%' IDENTIFIED BY 'Admin@123';
Query OK, 0 rows affected (0.01 sec)
```

FLUSH PRIVILEGES;

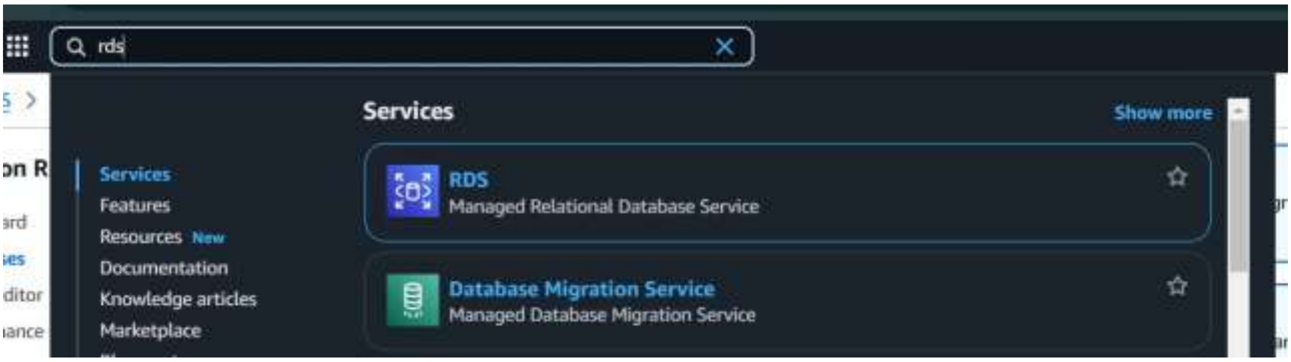


```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

Create MySQL Database FOR TARGET on AWS (RDS)

At the top of the AWS console, you'll see a **Search bar**. Type **"RDS"** into the search field.

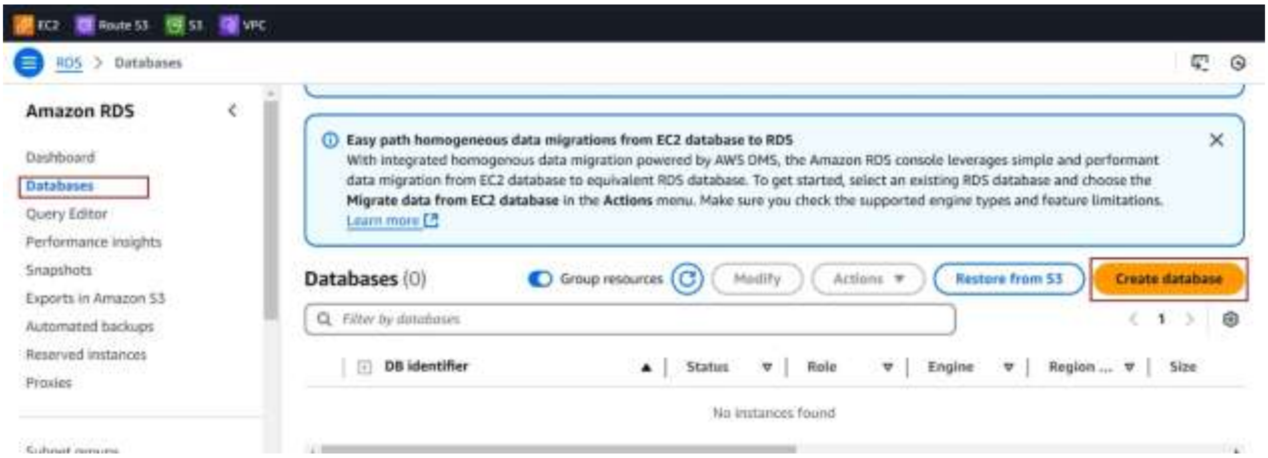
Click on **RDS** from the dropdown list that appears



rds
Services
Show more
Services
Features
Resources New
Documentation
Knowledge articles
Marketplace
RDS
Managed Relational Database Service
Database Migration Service
Managed Database Migration Service

On the **left-side pane**, under the **RDS Dashboard**, you'll see several options.

Click on **Databases,** then Click on "Create Database".

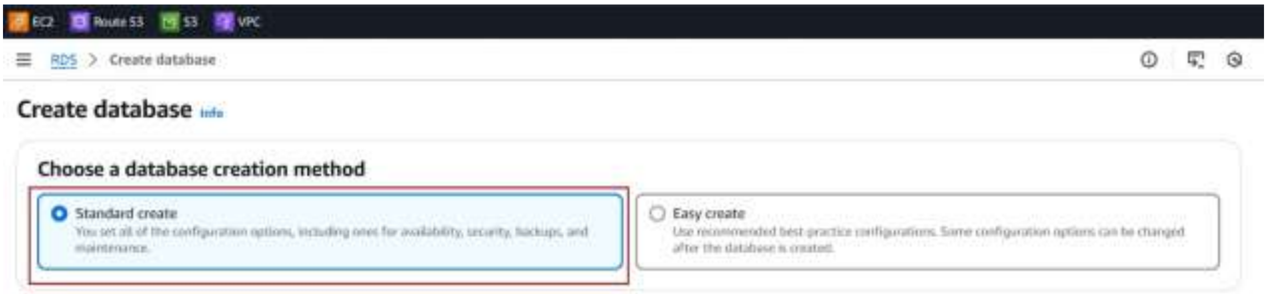


Easy path homogeneous data migrations from EC2 database to RDS

With integrated homogenous data migration powered by AWS DMS, the Amazon RDS console leverages simple and performant data migration from EC2 database to equivalent RDS database. To get started, select an existing RDS database and choose the **Migrate data from EC2 database** in the **Actions** menu. Make sure you check the supported engine types and feature limitations. Learn more

No instances found

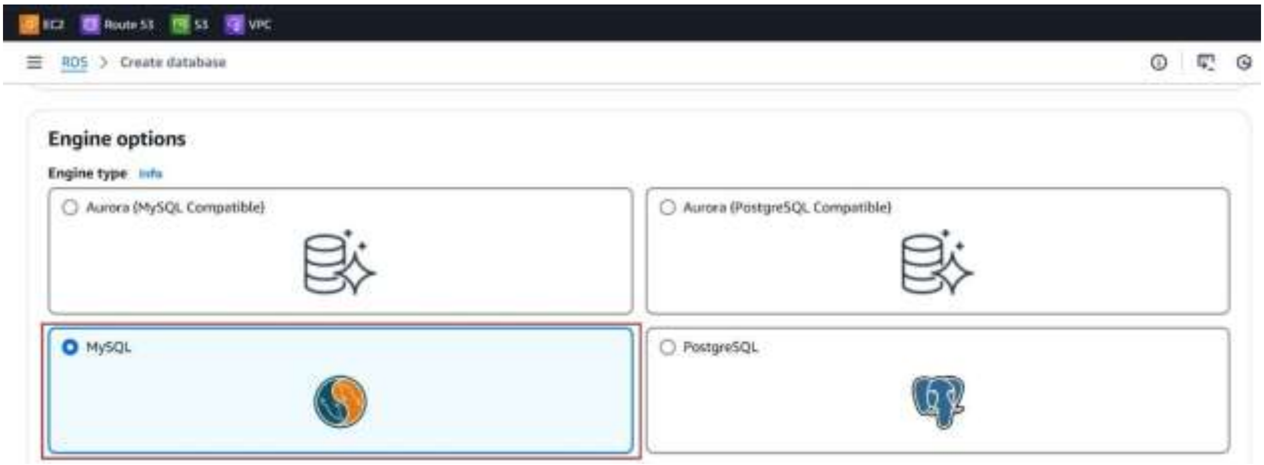
Choose **Standard Create** for detailed configurations



● 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 type: Select **MySQL**.



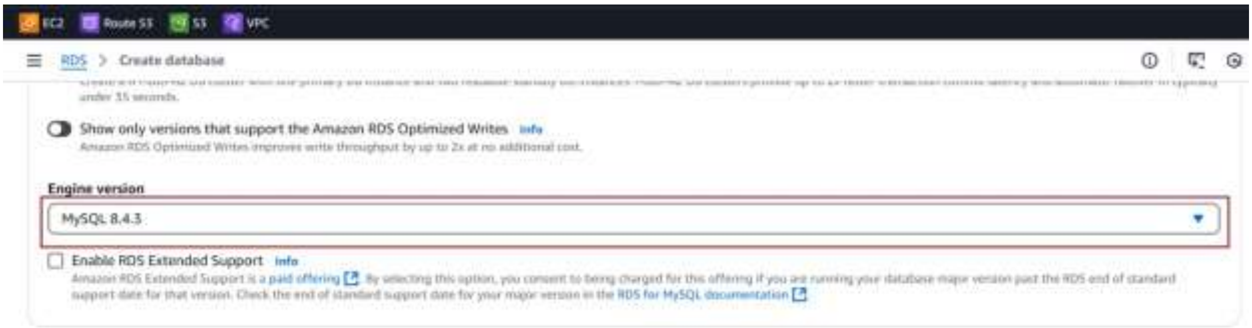
○ Aurora (MySQL Compatible)

○ Aurora (PostgreSQL Compatible)

● MySQL

○ PostgreSQL

Version: Choose the MySQL version you prefer



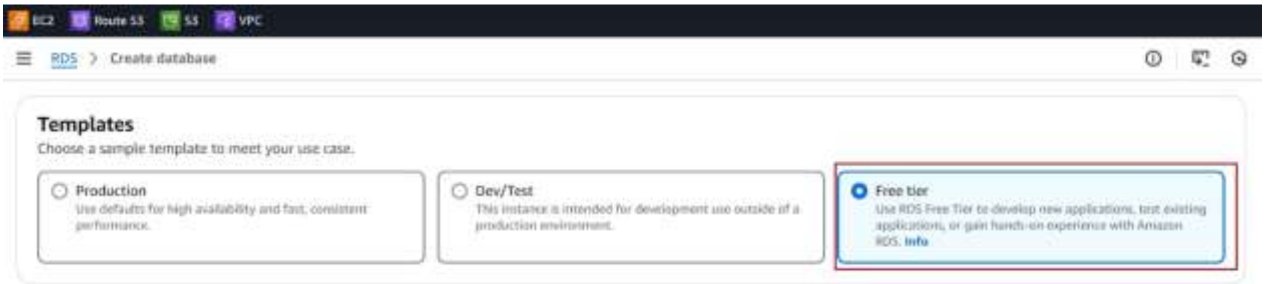
under 35 seconds.

○ Show only versions that support the Amazon RDS Optimized Writes Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine version
MySQL 8.4.3

□ Enable RDS Extended Support Info
Amazon RDS Extended Support is a paid offering. By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the RDS for MySQL documentation

Choose **a Free tier template** for learning or experimenting with RDS at no cost



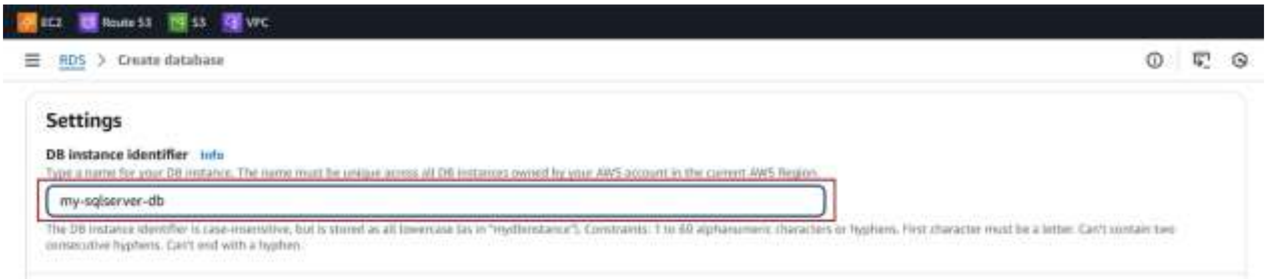
Templates
Choose a sample template to meet your use case.

○ 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

DB instance identifier: Enter a unique name for your database (e.g., my-sqlserver-db).

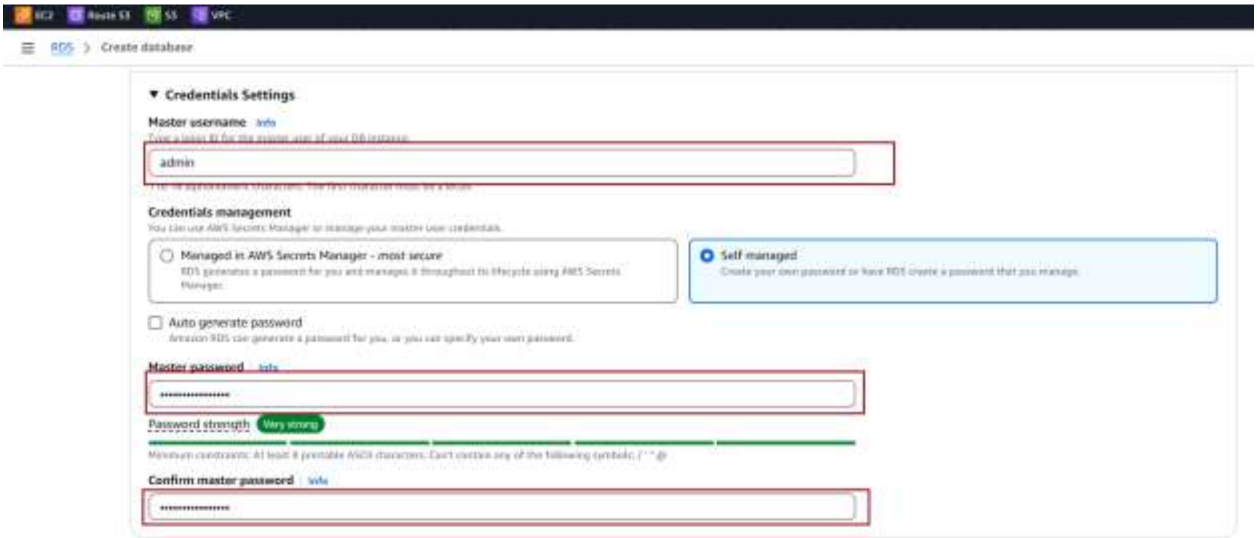


Settings

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.
my-sqlserver-db
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.

Master username: Set the admin username (e.g., admin).

Master password: Create and confirm a strong password.



EC2 Route 53 S3 VPC

RDS > Create database

▼ 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

•••••••••••••••

Password strength Very strong

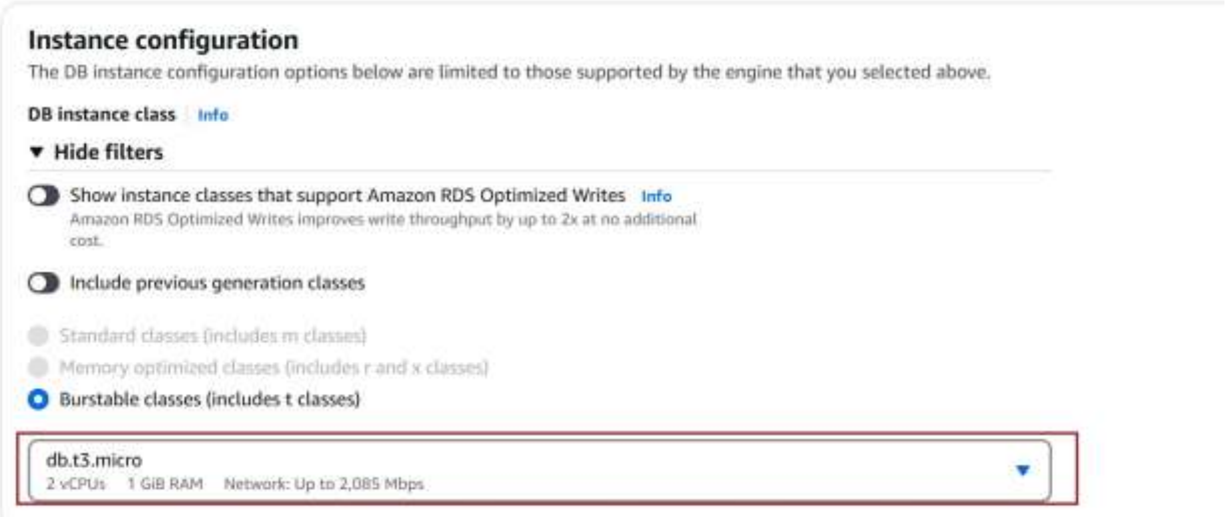
Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

Confirm master password Info

•••••••••••••••

DB instance class: Choose the instance type based on workload needs:

- For testing: **db.t3.micro** (free tier eligible).



Instance configuration

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

DB instance class Info

▼ Hide filters

○ Show instance classes that support Amazon RDS Optimized Writes Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

○ Include previous generation classes

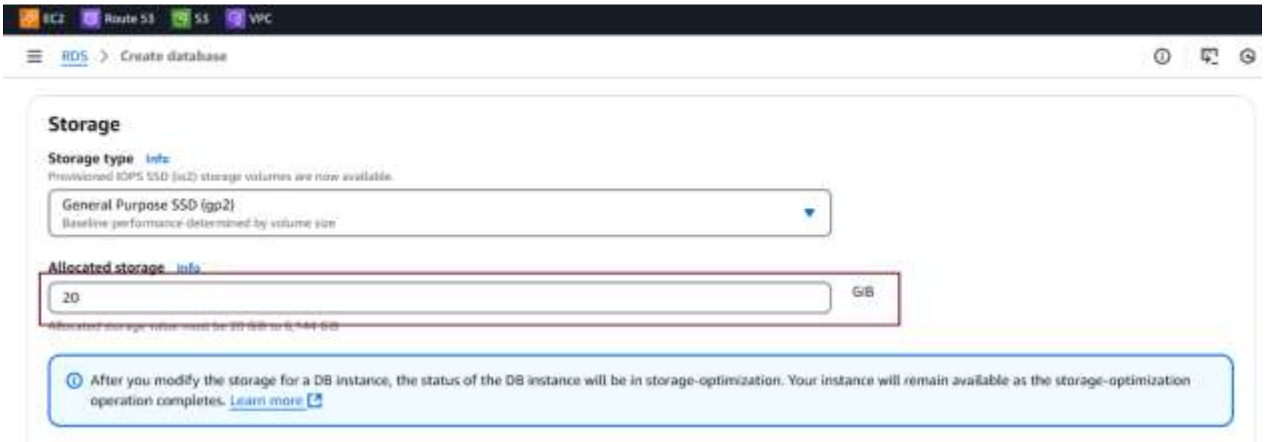
○ Standard classes (includes m classes)

○ Memory optimized classes (includes r and x classes)

● Burstable classes (includes t classes)

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

Allocated storage: Set the storage size (minimum 20 GiB).



EC2 Route 53 S3 VPC

RDS > Create database

Storage

Storage type Info

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

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

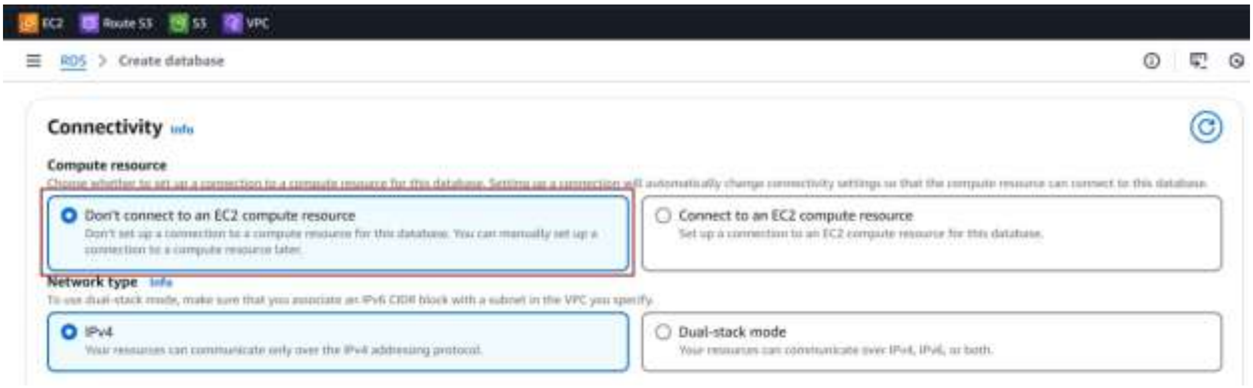
Allocated storage Info

20 GiB

Allocated storage value must be 20 GiB to 6,144 GiB

ⓘ After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes. Learn more

Choose Don’t connect to an EC2 compute resource option under **Compute resource**



Connectivity Info

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

● Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

○ Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Network type Info

To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

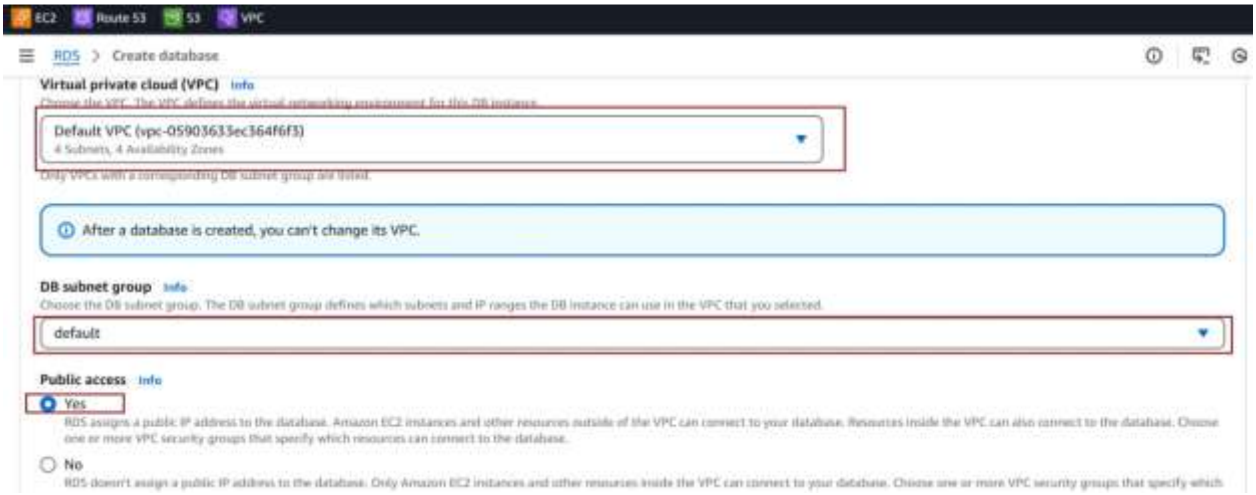
● IPv4
Your resources can communicate only over the IPv4 addressing protocol.

○ Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.

Virtual Private Cloud (VPC): Select an existing VPC or create a new one.

DB Subnet group: Choose a subnet group within the selected VPC.

Select **Yes for Public access**



Virtual private cloud (VPC) Info

Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-05903633ec364f6f3)
4 Subnets, 4 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

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.

default

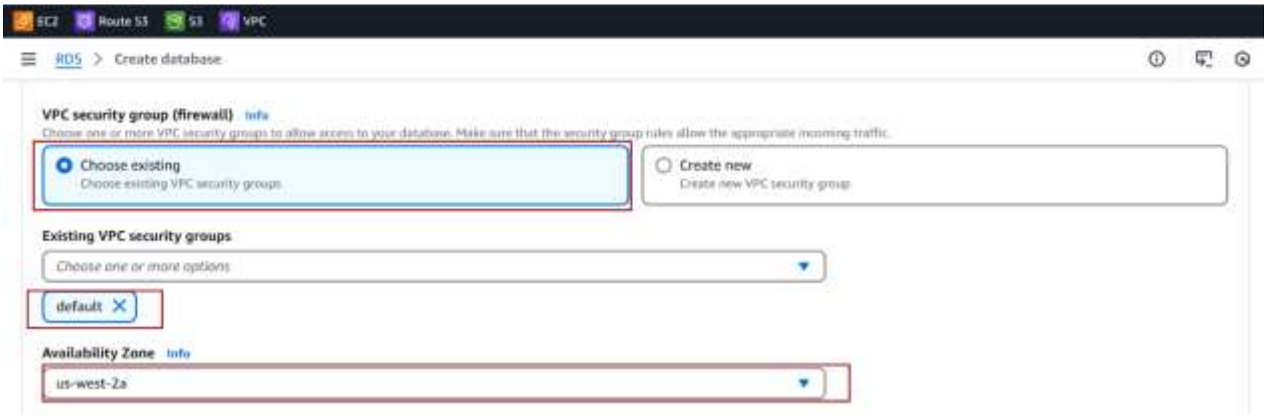
Public access Info

● Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

○ No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which

VPC security group:

- Ensure inbound traffic on **port 3306** (default MySQL port) is allowed
- choose Availability Zone if required.



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

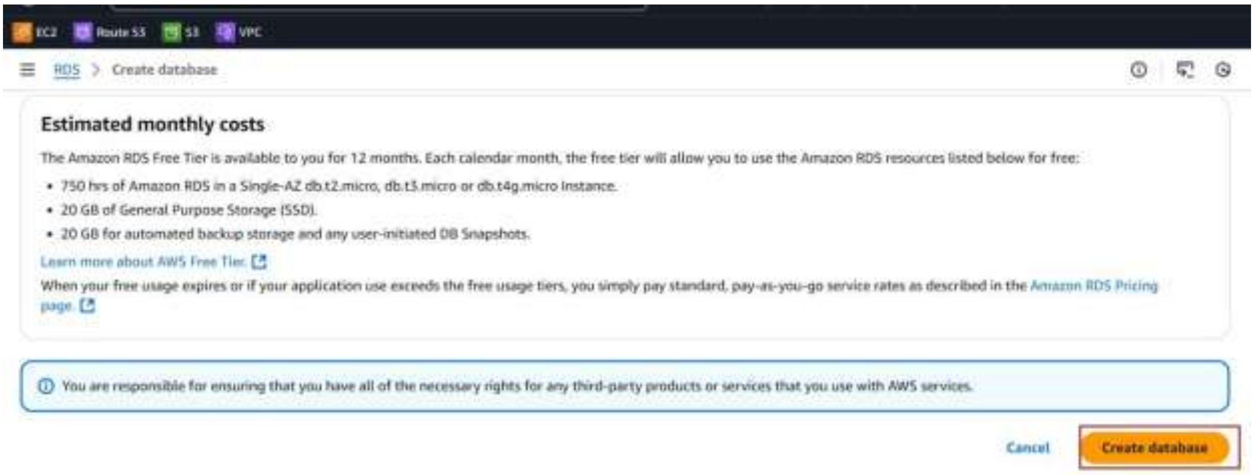
Choose one or more options

default ✕

Availability Zone Info

us-west-2a

Click **Create database**



Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

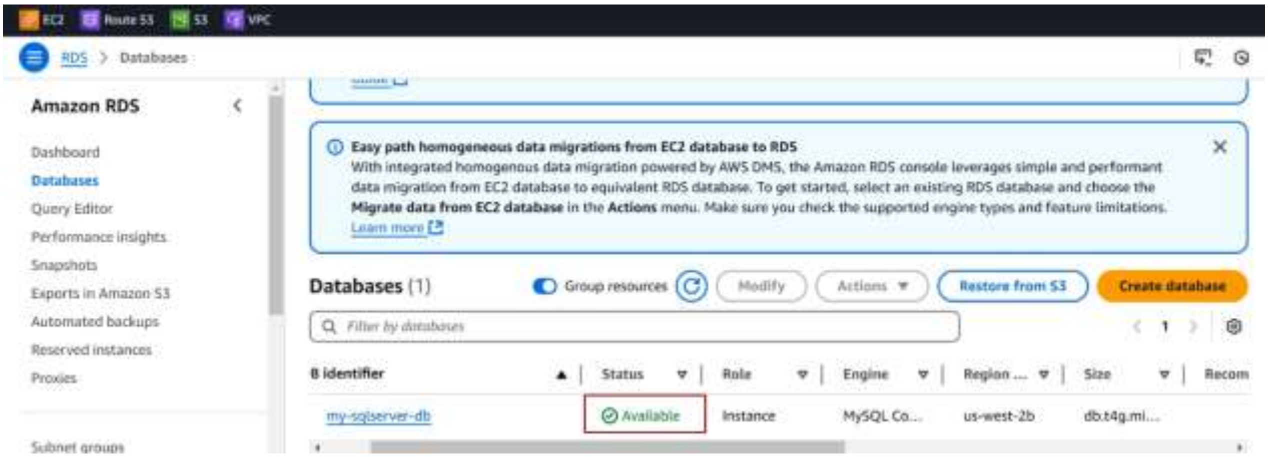
- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

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.

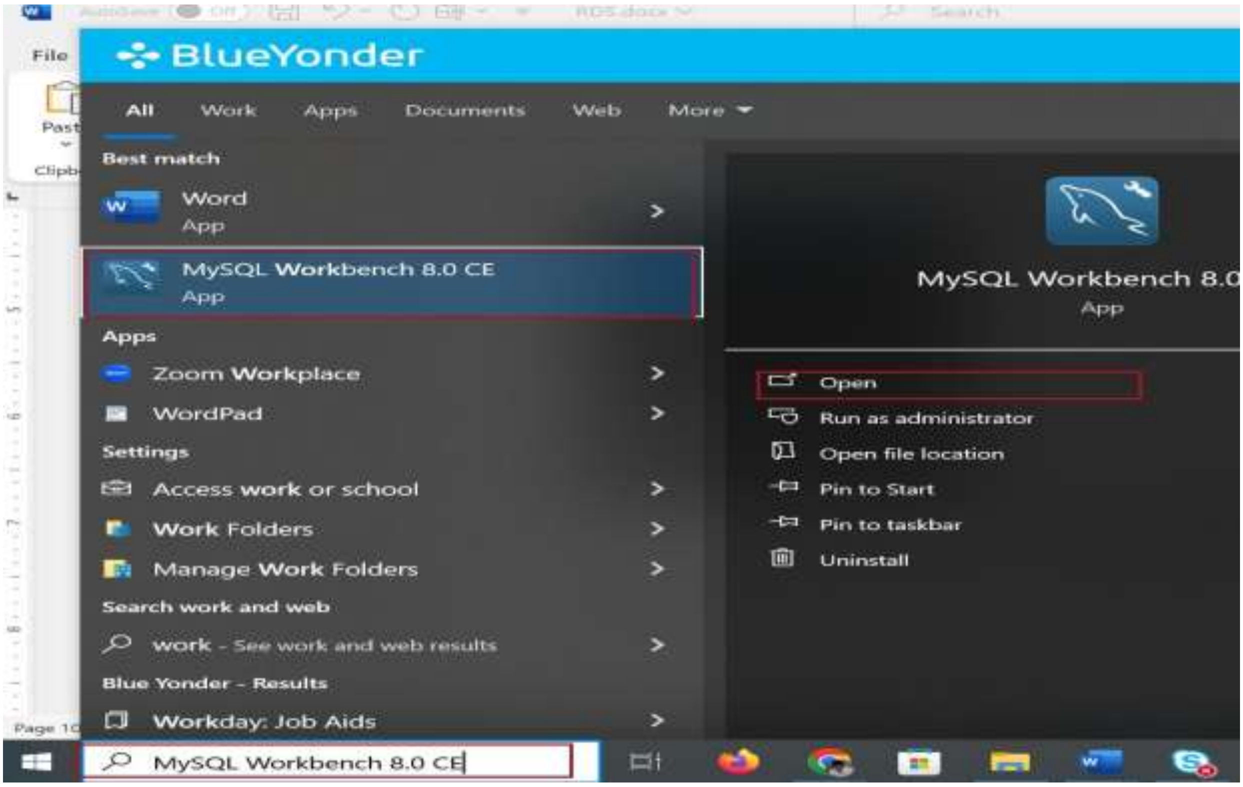
It may take a few minutes for the database status to change to **Available**



ⓘ Easy path homogeneous data migrations from EC2 database to RDS
With integrated homogenous data migration powered by AWS DMS, the Amazon RDS console leverages simple and performant data migration from EC2 database to equivalent RDS database. To get started, select an existing RDS database and choose the **Migrate data from EC2 database** in the **Actions** menu. Make sure you check the supported engine types and feature limitations.
Learn more

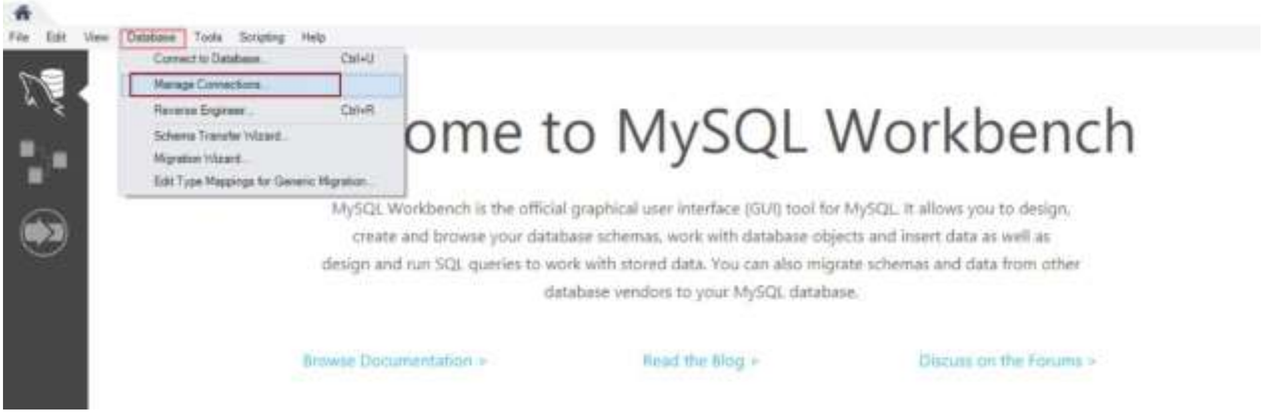
B identifier	Status	Role	Engine	Region ...	Size	Recom
my-sqlserver-db	Available	Instance	MySQL Co...	us-west-2b	db.t4g.mi...	

Open Workbench from the Start Menu or desktop shortcut.



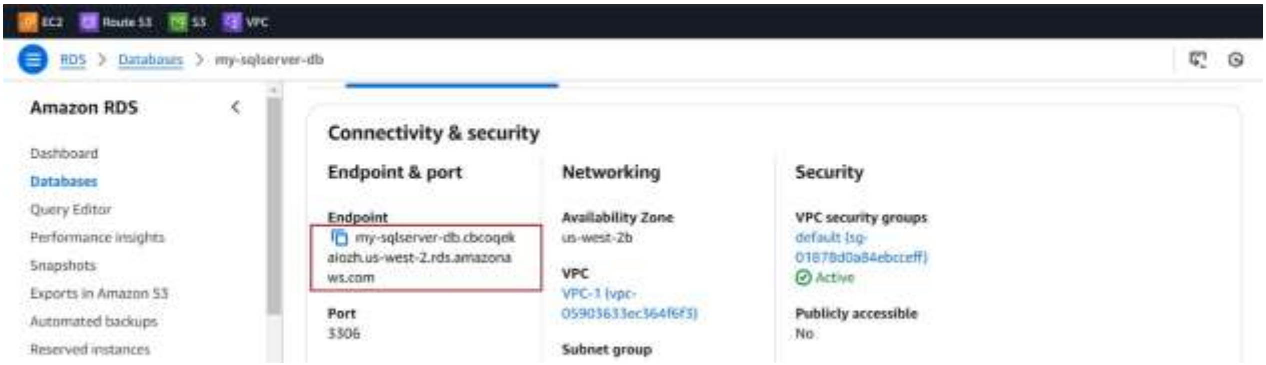
Connect MySQL Workbench to Your AWS Database

Select Database tab and click on Manage Connections



MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

Copy the Endpoint to use in the connection creation



Connectivity & security

Endpoint & port

Endpoint: my-sqlserver-db.cbcoqekaiozh.us-west-2.rds.amazonaws.com

Port: 3306

Networking

Availability Zone: us-west-2b

VPC: VPC-1 (vpc-05903633ec364f6f3)

Subnet group

Security

VPC security groups: default (sg-01878d0a84ebcceff) Active

Publicly accessible: No

Connection Name: Give it a name (e.g., **AWS MySQL**).

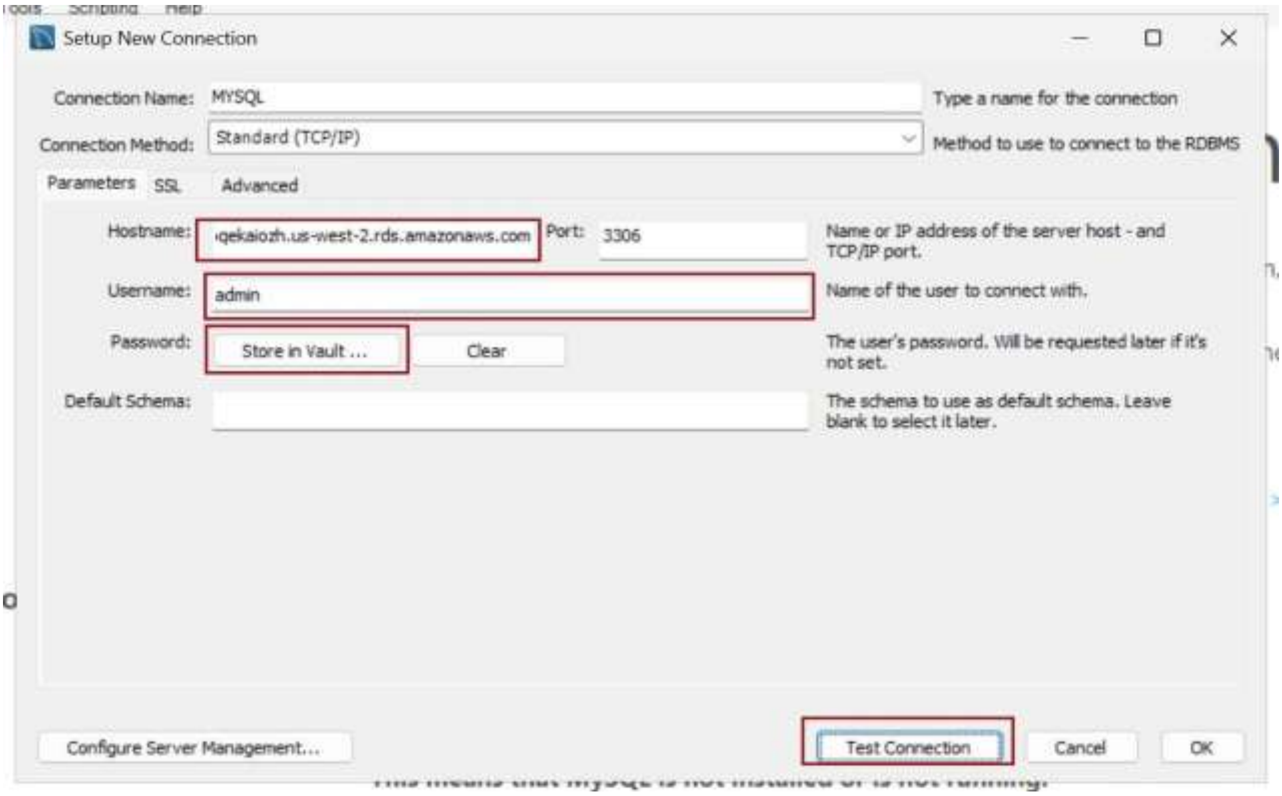
Hostname: Enter your RDS endpoint.

Port: Default is **3306**.

Username: The master username you created (e.g., **admin**).

Password: Click **Store in Vault** and enter the master password

Click **Test Connection**.



Connection Name: MYSQL — Type a name for the connection

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Hostname: qekaiozh.us-west-2.rds.amazonaws.com Port: 3306 — Name or IP address of the server host - and TCP/IP port.

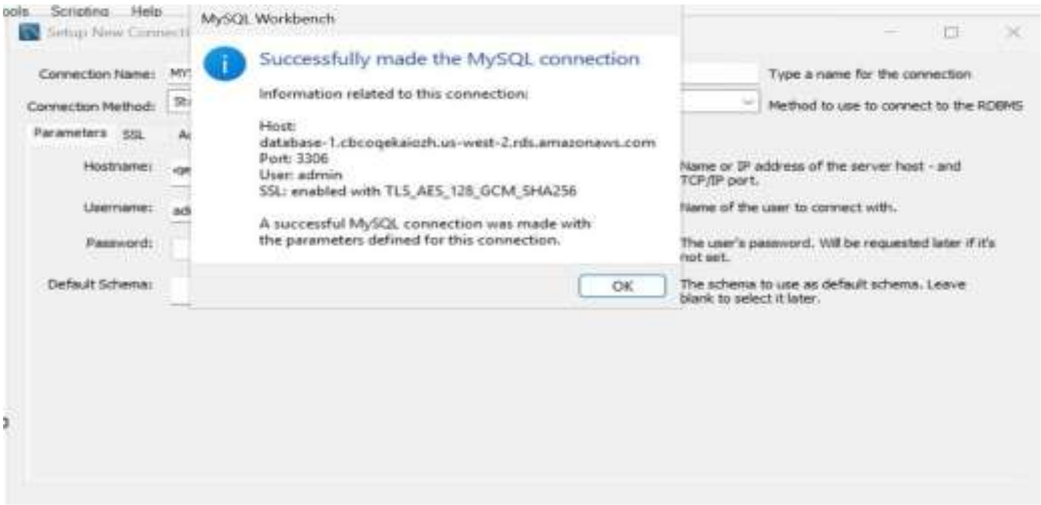
Username: admin — Name of the user to connect with.

Password: — The user's password. Will be requested later if it's not set.

Default Schema: — The schema to use as default schema. Leave blank to select it later.

If successful, you'll see a message: **Connection parameters are correct**.

Click **OK** to save the connection.



MySQL Workbench

Successfully made the MySQL connection

Information related to this connection:

Host:
database-1.cbcoqekaiozh.us-west-2.rds.amazonaws.com
Port: 3306
User: admin
SSL: enabled with TLS_AES_128_GCM_SHA256

A successful MySQL connection was made with the parameters defined for this connection.

Connection Name: Type a name for the connection

Connection Method: Method to use to connect to the RDBMS

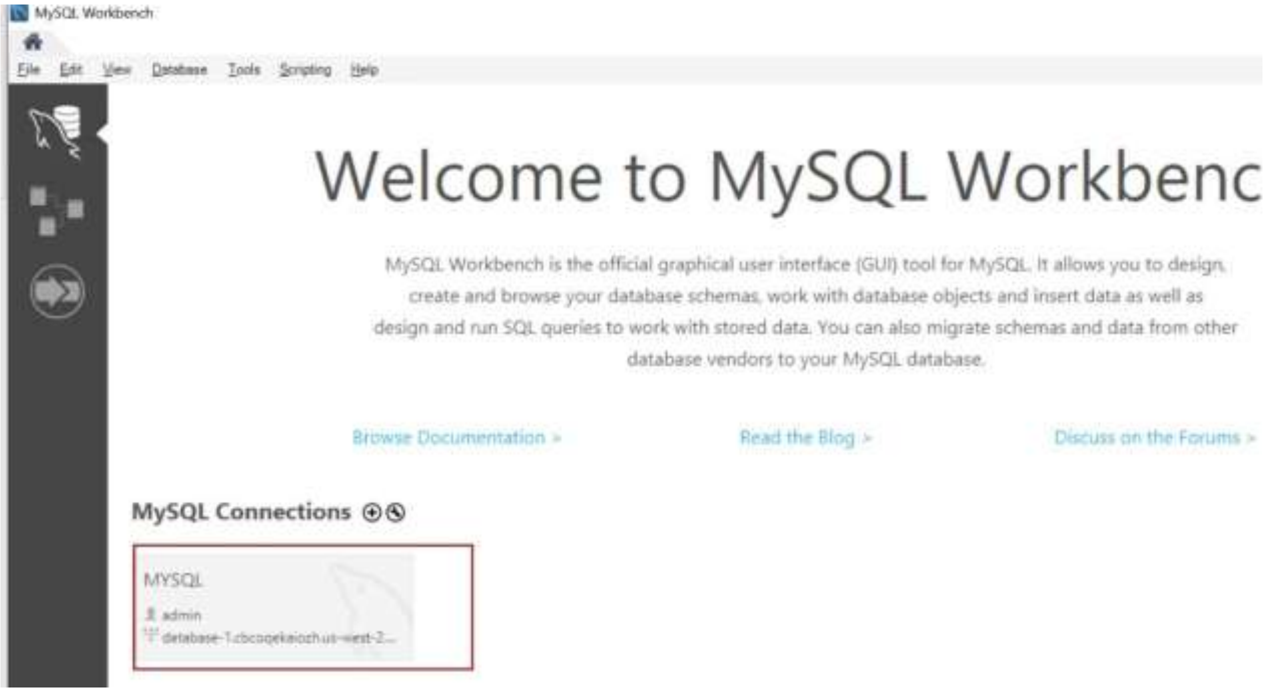
Hostname: Name or IP address of the server host - and TCP/IP port.

Username: Name of the user to connect with.

Password: The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

Double-click your saved connection in MySQL Workbench to connect to the MySQL RDS instance.



Welcome to MySQL Workbenc

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

MySQL Connections

MYSQL
admin
database-1.cbcoqekaiozh.us-west-2...

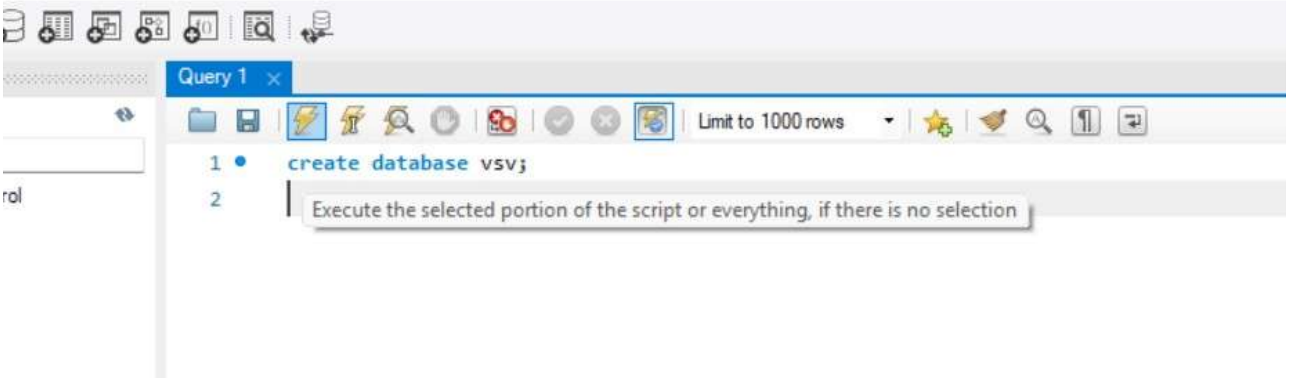
Create a New Database in MySQL Workbench

We will create empty database

Type the following SQL command to create a new database:

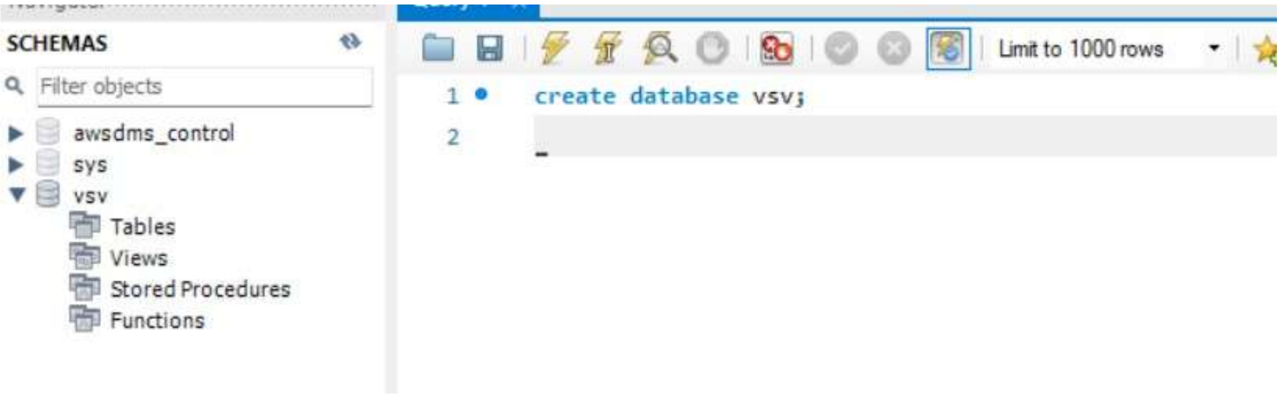
```
CREATE DATABASE vsv;
```

Click the **Execute** button (lightning bolt icon).



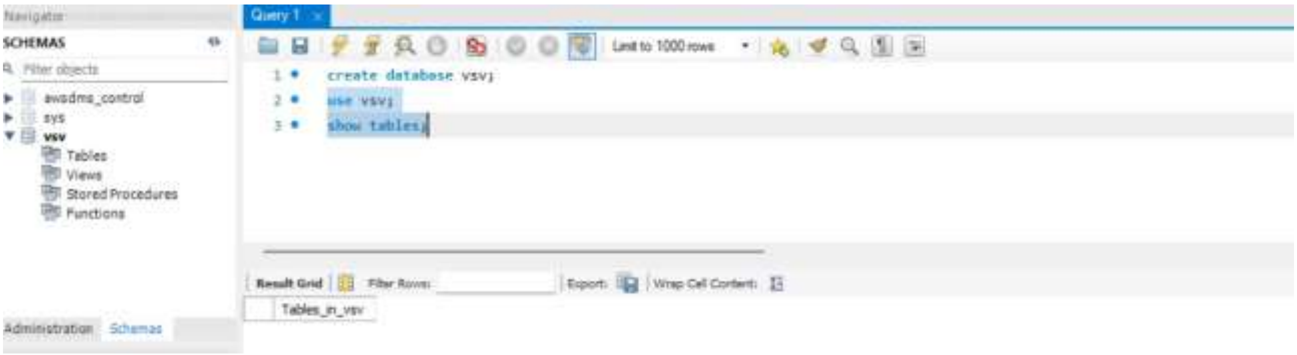
```
create database vsv;
```

Execute the selected portion of the script or everything, if there is no selection



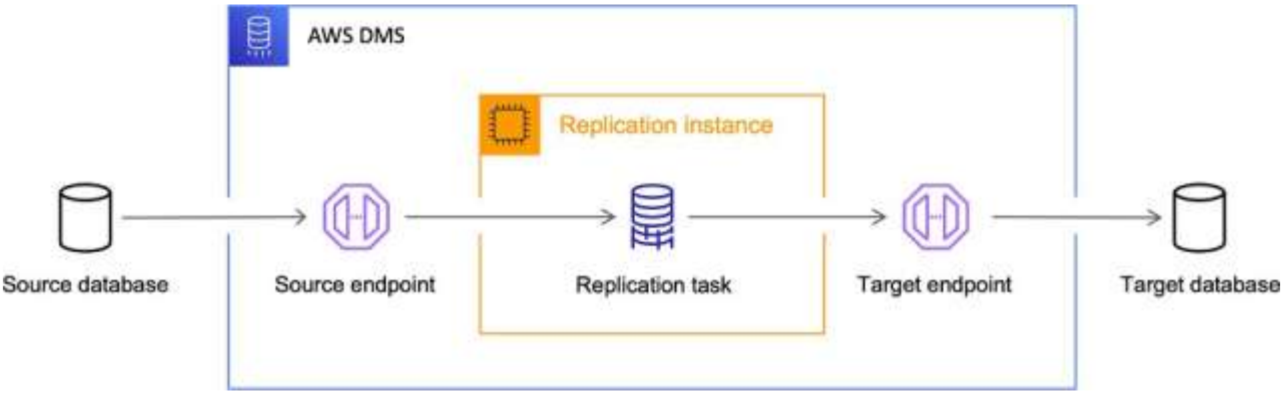
```
create database vsv;
```

- the database is created when we fetch the table nothing is there



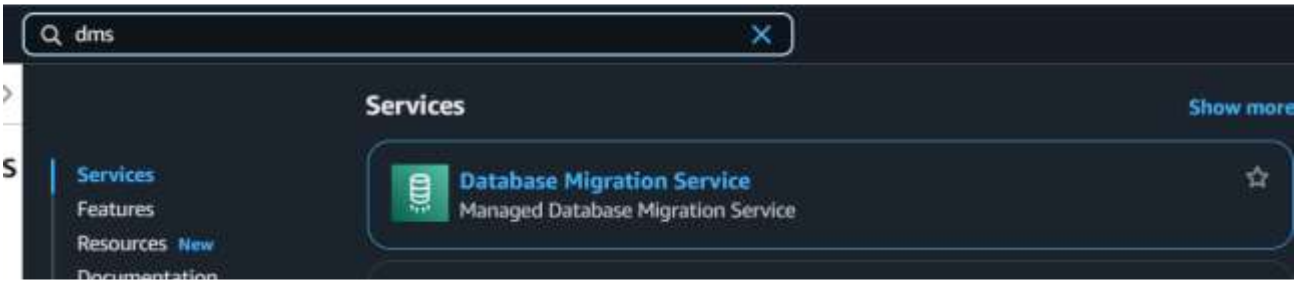
```
create database vsv;
use vsv;
show tables;
```

Tables_in_vsv

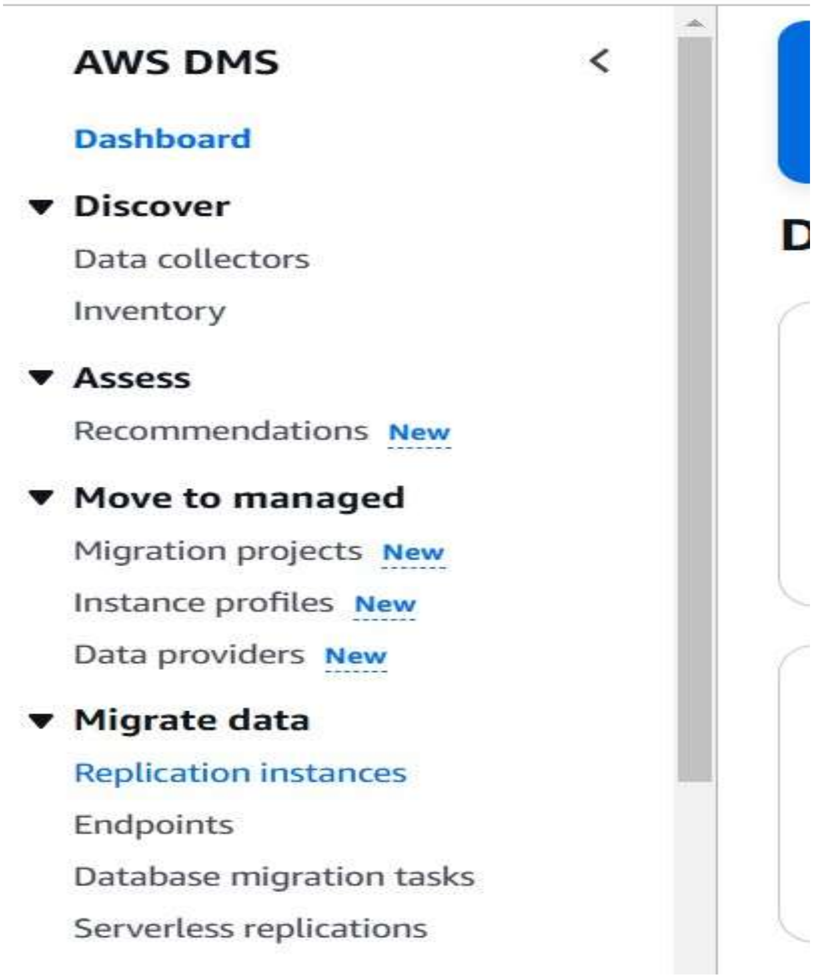


SETUP THE DATABASE MIGRATION SERVICE

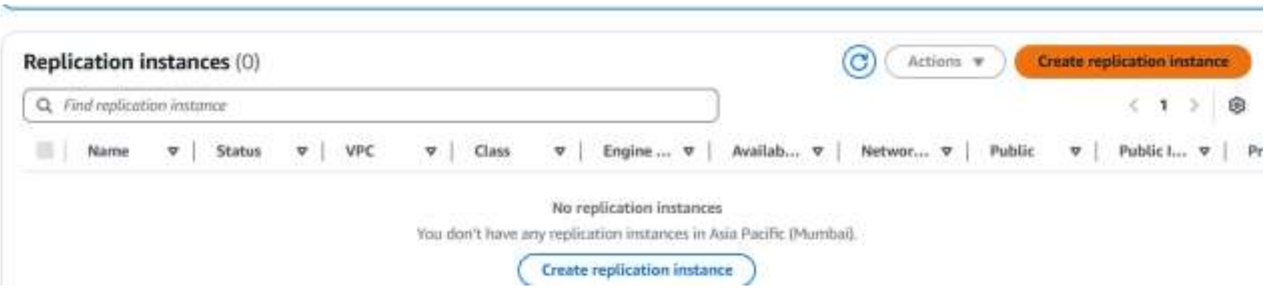
- open the database migration service



- click on replication instance in migrate data block



- Click on create replication instance

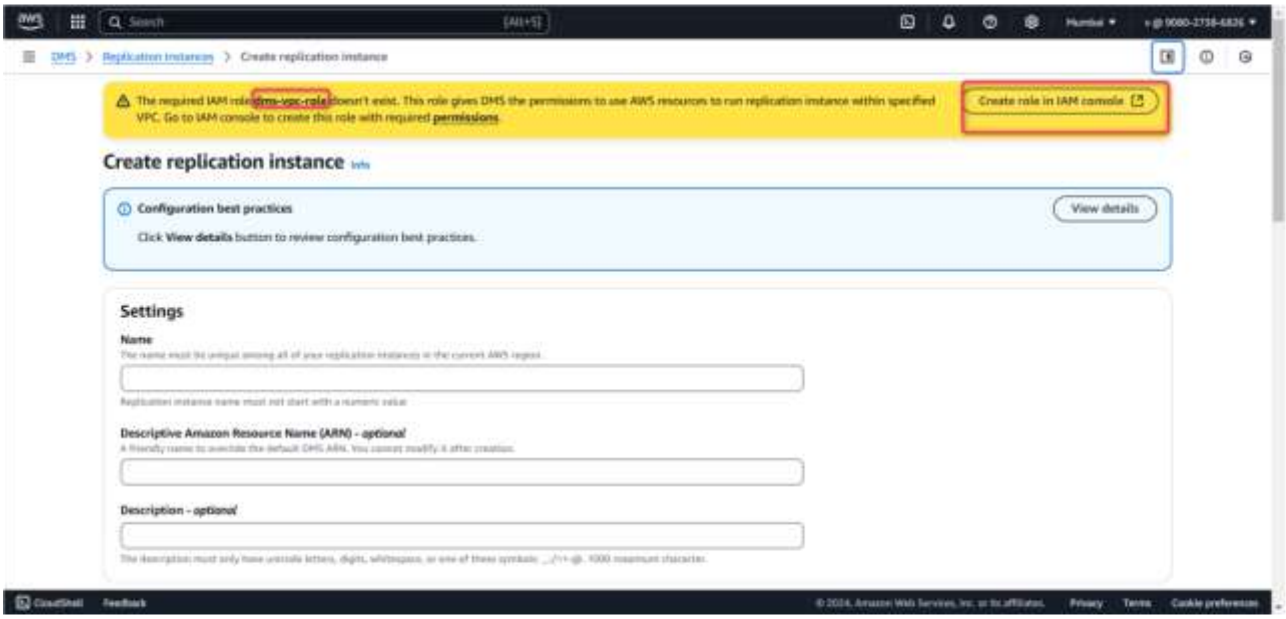

Replication instances (0)
Actions
Create replication instance
Find replication instance
1
Name
Status
VPC
Class
Engine ...
Availab...
Networ...
Public
Public I...
Pr
Create replication instance

No replication instances

You don't have any replication instances in Asia Pacific (Mumbai).

- In replication instance creation time it will ask role.

- Click on create role in IAM console
- While creating the role give name dms-vpc-role it is mandatory we give this name only it replication instance fetch the role details



DMS > Replication instances > Create replication instance

⚠ The required IAM role dms-vpc-role doesn't exist. This role gives DMS the permissions to use AWS resources to run replication instance within specified VPC. Go to IAM console to create this role with required permissions.

Create replication instance Info

ⓘ Configuration best practices

Click **View details** button to review configuration best practices.

Settings

Name

The name must be unique among all of your replication instances in the current AWS region.

Replication instance name must not start with a numeric value.

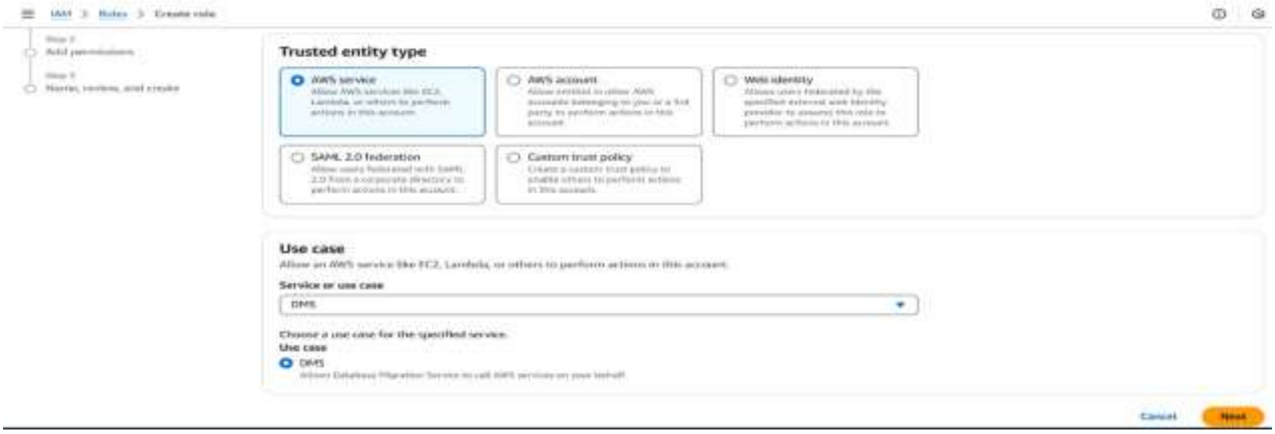
Descriptive Amazon Resource Name (ARN) - *optional*

A friendly name to override the default DMS ARN. You cannot modify it after creation.

Description - *optional*

The description must only have unicode letters, digits, whitespace, or one of these symbols: _.:/=+-@. 1000 maximum character.

- Select aws service
- Use case is DMS then click on next



IAM > Roles > Create role

Trusted entity type

● AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

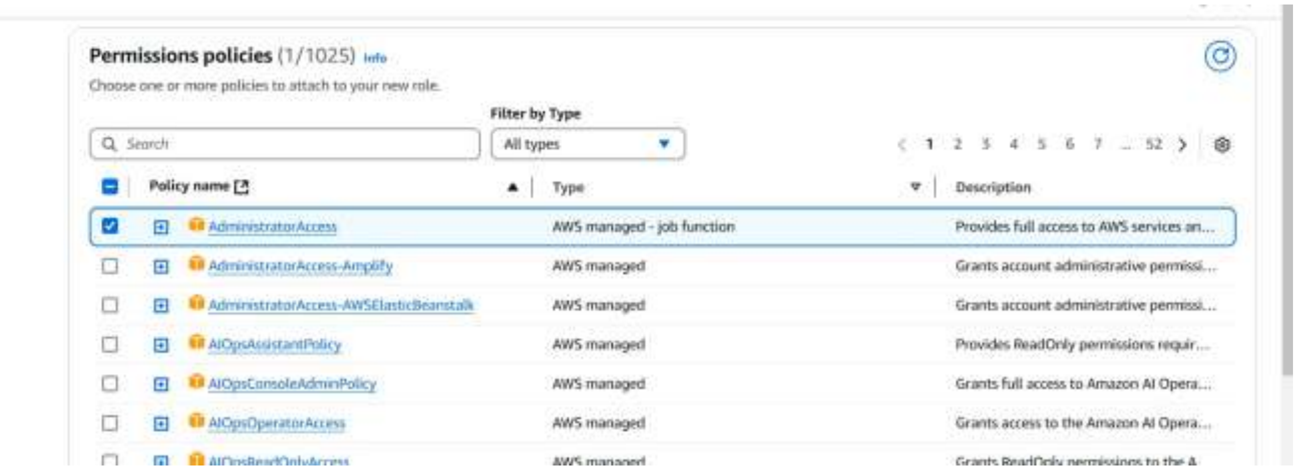
Service or use case: DMS

Choose a use case for the specified service.

Use case

● DMS
Allows Database Migration Service to call AWS services on your behalf.

- Select required permissions or administrator access and click next



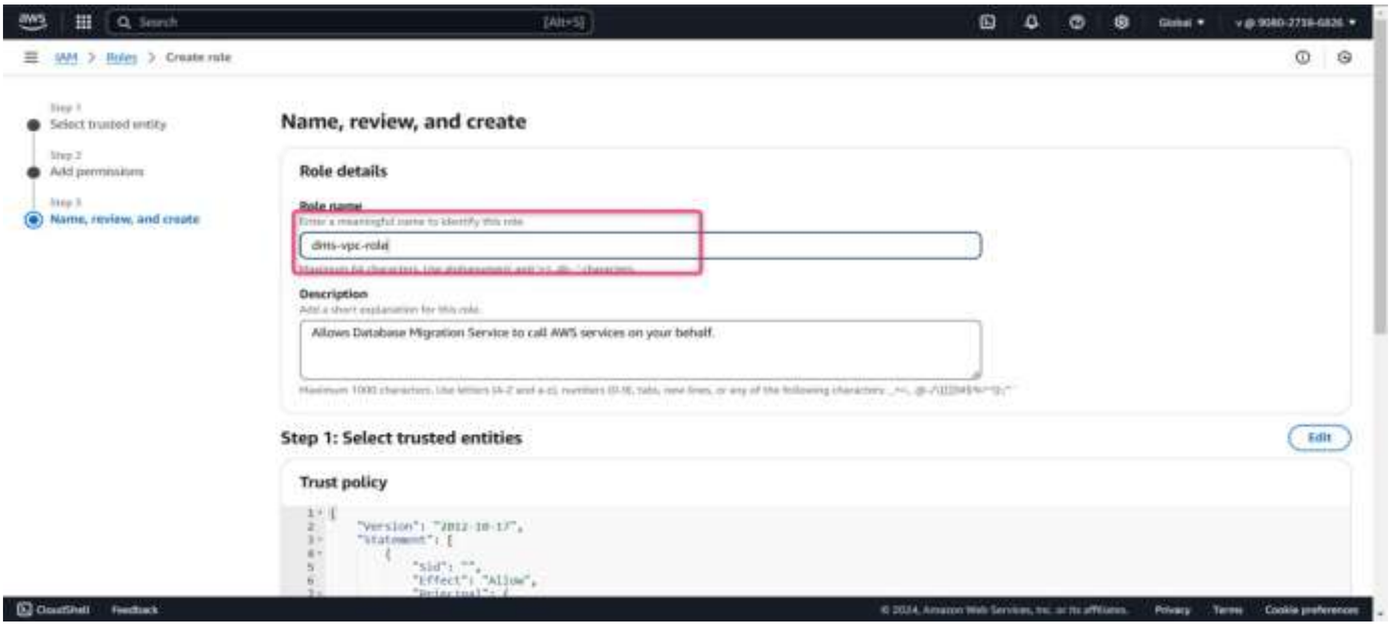
Permissions policies (1/1025) Info

Choose one or more policies to attach to your new role.

Filter by Type: All types

☑	Policy name	Type	Description
☑	AdministratorAccess	AWS managed - job function	Provides full access to AWS services an...
□	AdministratorAccess-Amplify	AWS managed	Grants account administrative permissi...
□	AdministratorAccess-AWSElasticBeanstalk	AWS managed	Grants account administrative permissi...
□	AIOpsAssistantPolicy	AWS managed	Provides ReadOnly permissions requir...
□	AIOpsConsoleAdminPolicy	AWS managed	Grants full access to Amazon AI Opera...
□	AIOpsOperatorAccess	AWS managed	Grants access to the Amazon AI Opera...
□	AIOpsReadOnlyAccess	AWS managed	Grants ReadOnly permissions to the A...

- Give name dms-vpc-role only
- Then click on create role



IAM > Roles > Create role

Name, review, and create

Role details

Role name: dms-vpc-role

Description: Allows Database Migration Service to call AWS services on your behalf.

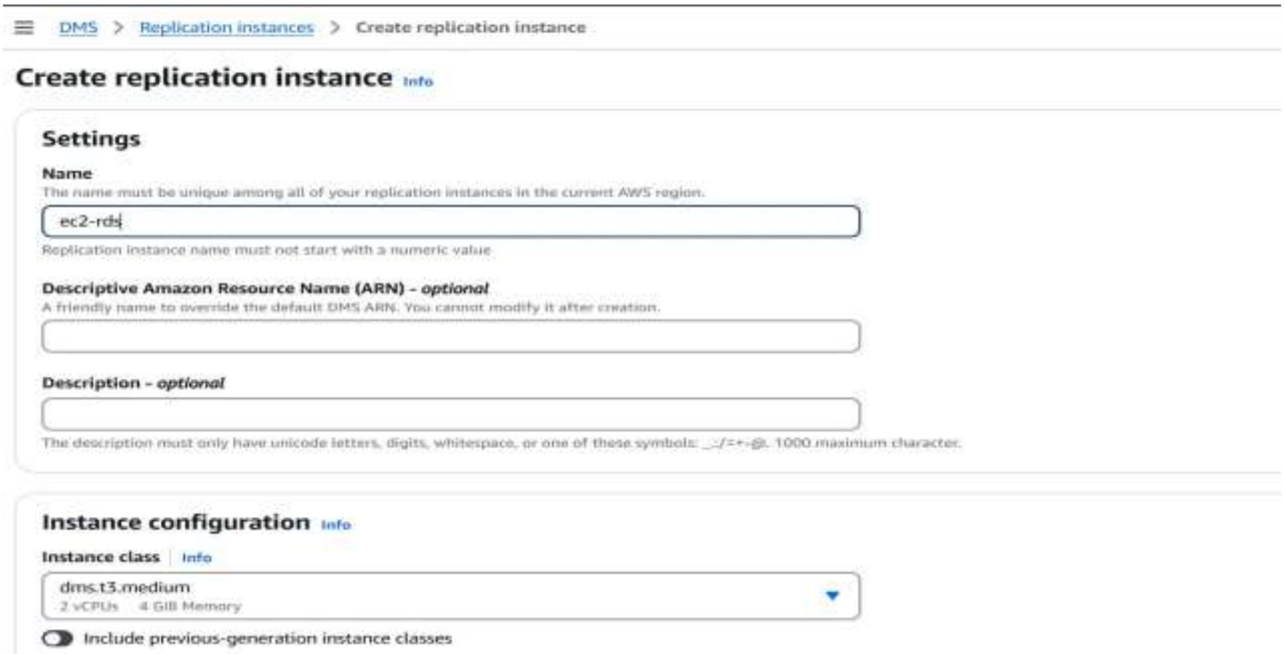
Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
```

REPLICATION INSTANCE CREATION

- Open creation of replication instance page give name
- Select instance class t3.medium



Create replication instance Info

Settings

Name
The name must be unique among all of your replication instances in the current AWS region.
ec2-rds
Replication instance name must not start with a numeric value

Descriptive Amazon Resource Name (ARN) - *optional*
A friendly name to override the default DMS ARN. You cannot modify it after creation.

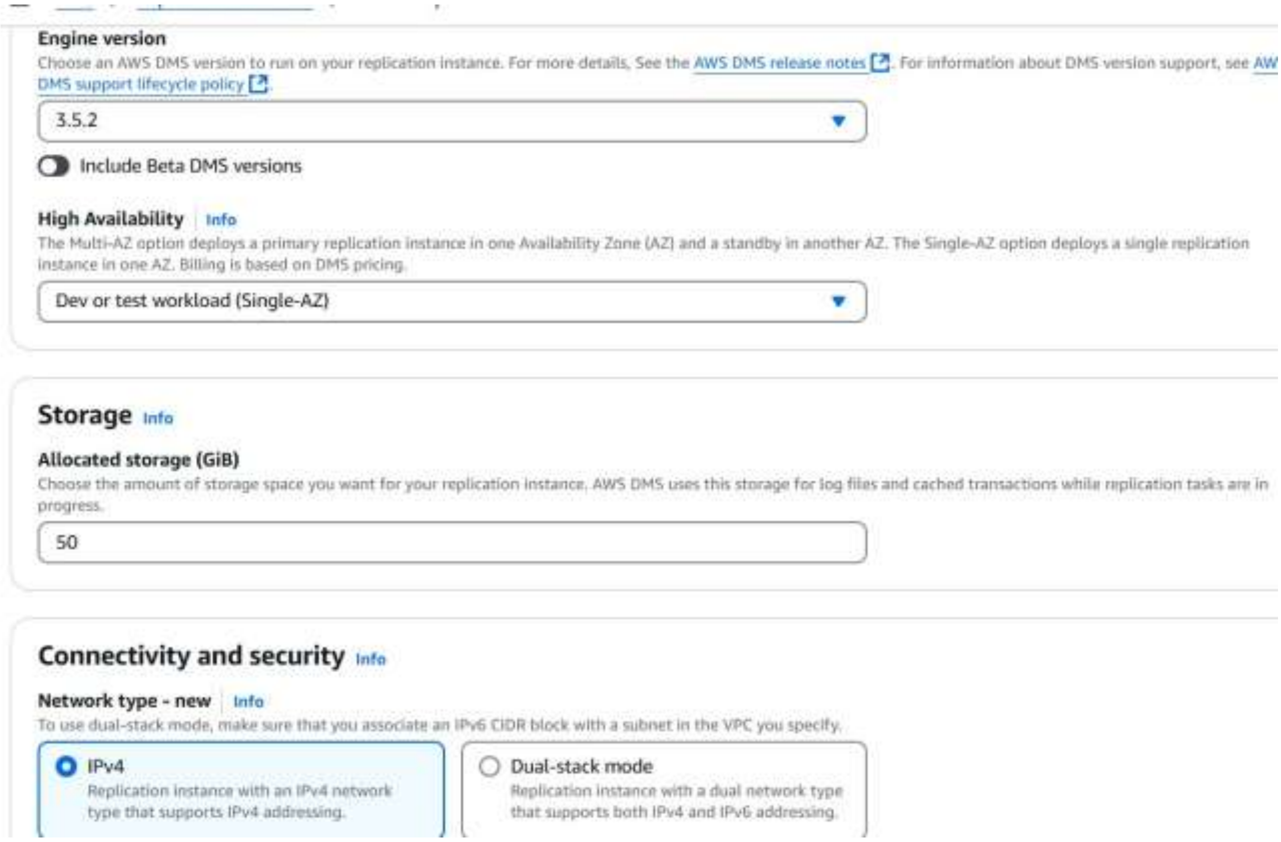
Description - *optional*
The description must only have unicode letters, digits, whitespace, or one of these symbols: _.:/=+-@. 1000 maximum character.

Instance configuration Info

Instance class | Info
dms.t3.medium
2 vCPUs 4 GiB Memory

○ Include previous-generation instance classes

- Select replication instance engine version [note this version not related to sql engine verion this is only for replication instance only]
- Select the high availability is dev or test
- Take the 50gb storage it is enough for our operation



Engine version
Choose an AWS DMS version to run on your replication instance. For more details, See the AWS DMS release notes. For information about DMS version support, see AW DMS support lifecycle policy.
3.5.2

○ Include Beta DMS versions

High Availability | Info
The Multi-AZ option deploys a primary replication instance in one Availability Zone (AZ) and a standby in another AZ. The Single-AZ option deploys a single replication instance in one AZ. Billing is based on DMS pricing.
Dev or test workload (Single-AZ)

Storage Info

Allocated storage (GiB)
Choose the amount of storage space you want for your replication instance. AWS DMS uses this storage for log files and cached transactions while replication tasks are in progress.
50

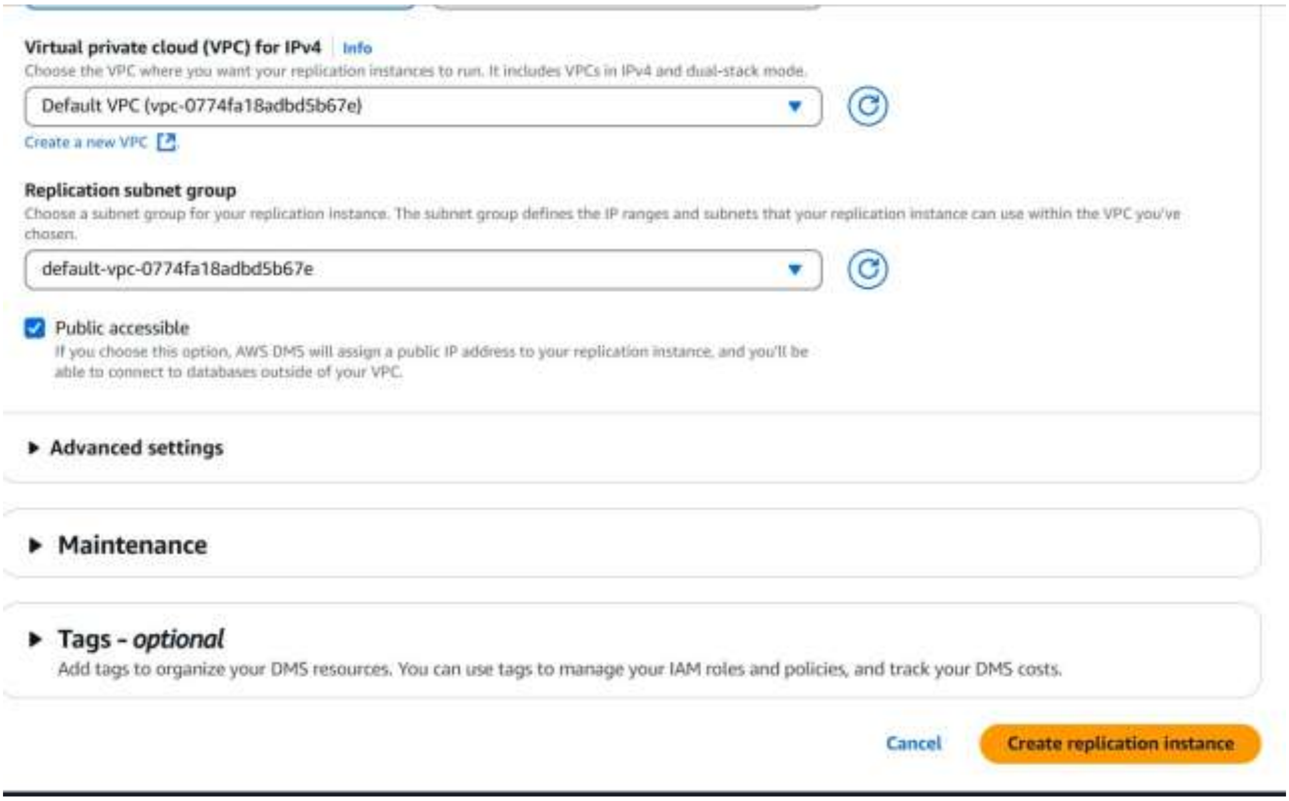
Connectivity and security Info

Network type - new | Info
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

● IPv4
Replication instance with an IPv4 network type that supports IPv4 addressing.

○ Dual-stack mode
Replication instance with a dual network type that supports both IPv4 and IPv6 addressing.

- Select your vpc and subnet group
- Then create replication instance



Virtual private cloud (VPC) for IPv4 Info

Choose the VPC where you want your replication instances to run. It includes VPCs in IPv4 and dual-stack mode.

Default VPC (vpc-0774fa18adbd5b67e)

Create a new VPC

Replication subnet group

Choose a subnet group for your replication instance. The subnet group defines the IP ranges and subnets that your replication instance can use within the VPC you've chosen.

default-vpc-0774fa18adbd5b67e

☑ Public accessible

If you choose this option, AWS DMS will assign a public IP address to your replication instance, and you'll be able to connect to databases outside of your VPC.

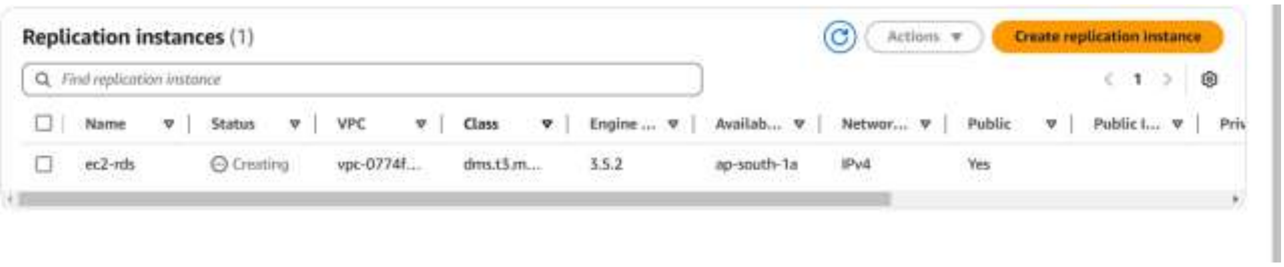
▶ Advanced settings

▶ Maintenance

▶ Tags - *optional*

Add tags to organize your DMS resources. You can use tags to manage your IAM roles and policies, and track your DMS costs.

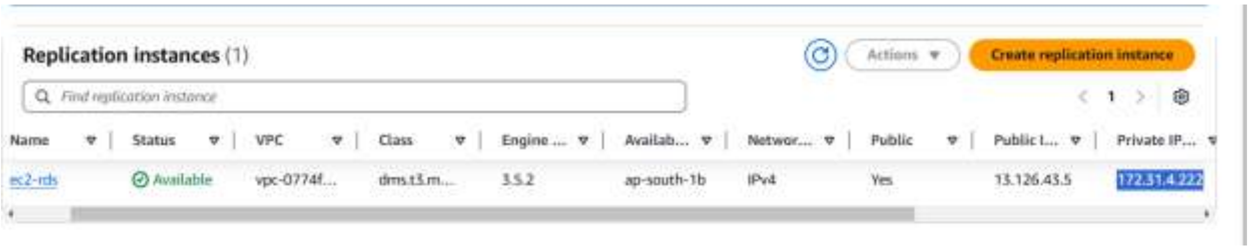
- It is on creating state
- It will take 15min to complete the instance creation



Replication instances (1)

□	Name	Status	VPC	Class	Engine ...	Availab...	Networ...	Public	Public I...	Priv
□	ec2-rds	Creating	vpc-0774f...	dms.t3.m...	3.5.2	ap-south-1a	IPv4	Yes		

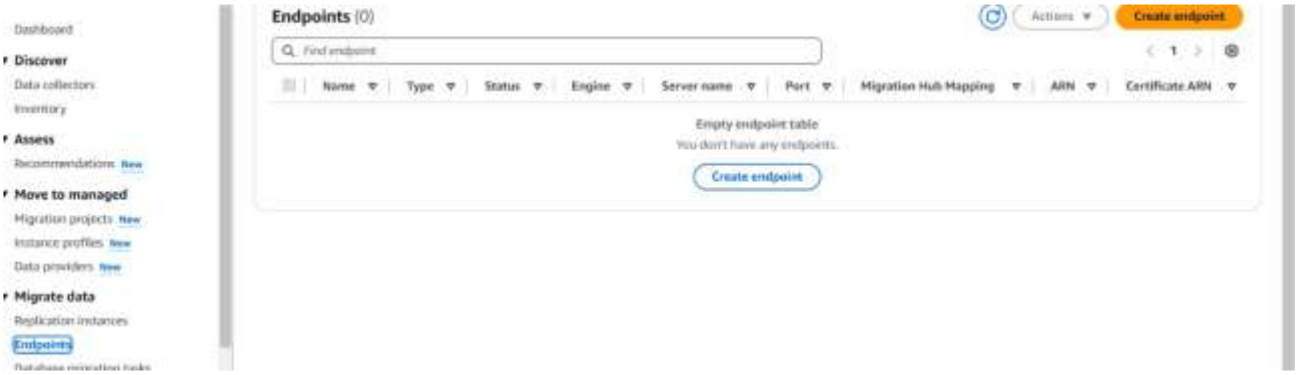
- Instance is created



Replication instances (1)

Name	Status	VPC	Class	Engine ...	Availab...	Networ...	Public	Public I...	Private IP...
ec2-rds	Available	vpc-0774f...	dms.t3.m...	3.5.2	ap-south-1b	IPv4	Yes	13.126.43.5	172.31.4.222

- open your dms service and click on endpoint
- click on create endpoint

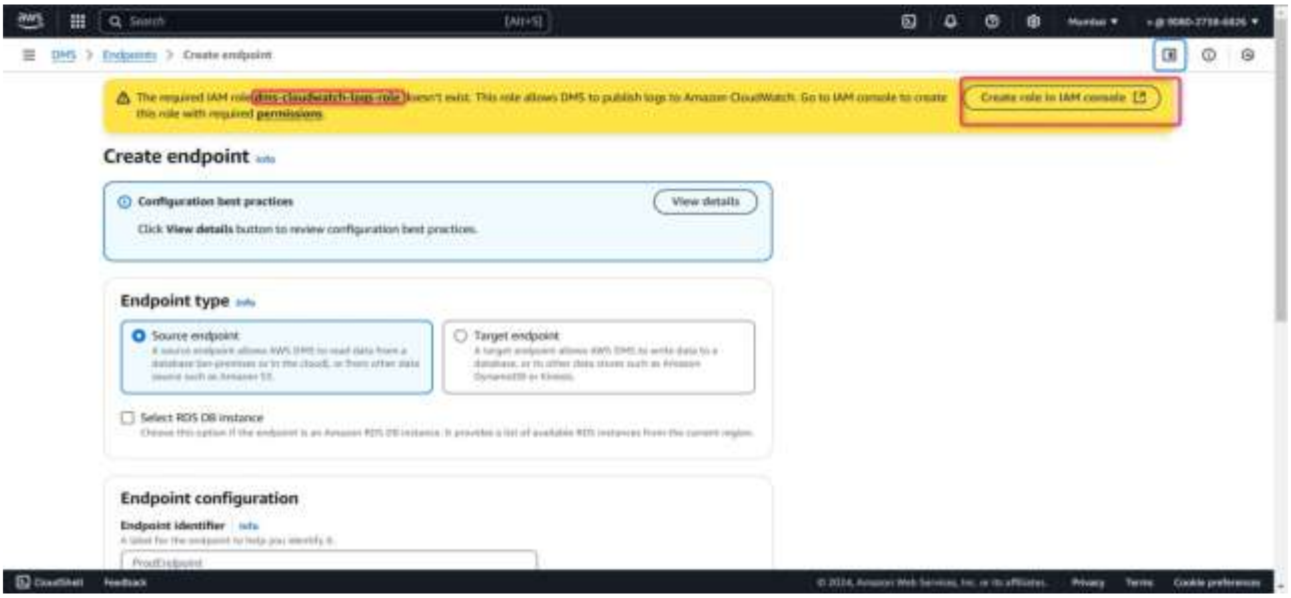


Endpoints (0)

Empty endpoint table

You don't have any endpoints.

- In endpoint creation time it will ask role.
- Click on create role in IAM console
- While creating the role give name dms-cloudwatch-logs-role it is mandatory we give this name only it endpoint fetch the role details



The required IAM role dms-cloudwatch-logs-role doesn't exist. This role allows DMS to publish logs to Amazon CloudWatch. Go to IAM console to create this role with required permissions.

Create endpoint

Configuration best practices

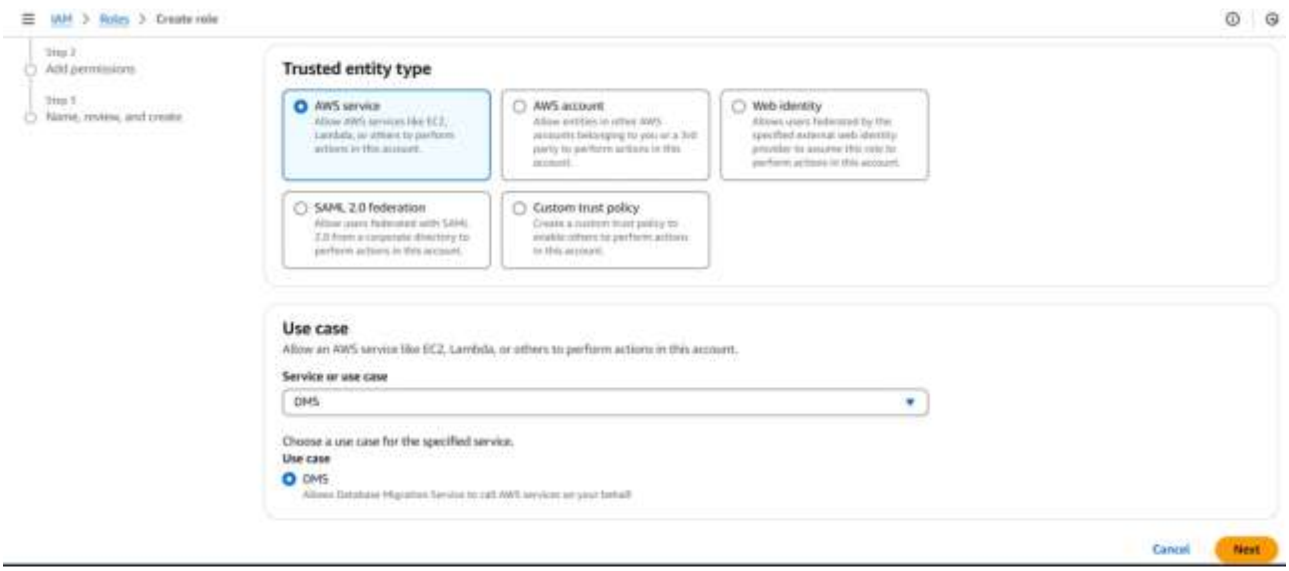
Click **View details** button to review configuration best practices.

Endpoint type

Endpoint configuration

Endpoint identifier

- Select aws service
- Use case is DMS then click on next



IAM > Roles > Create role

Step 2
Add permissions

Step 3
Name, review, and create

Trusted entity type

● AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

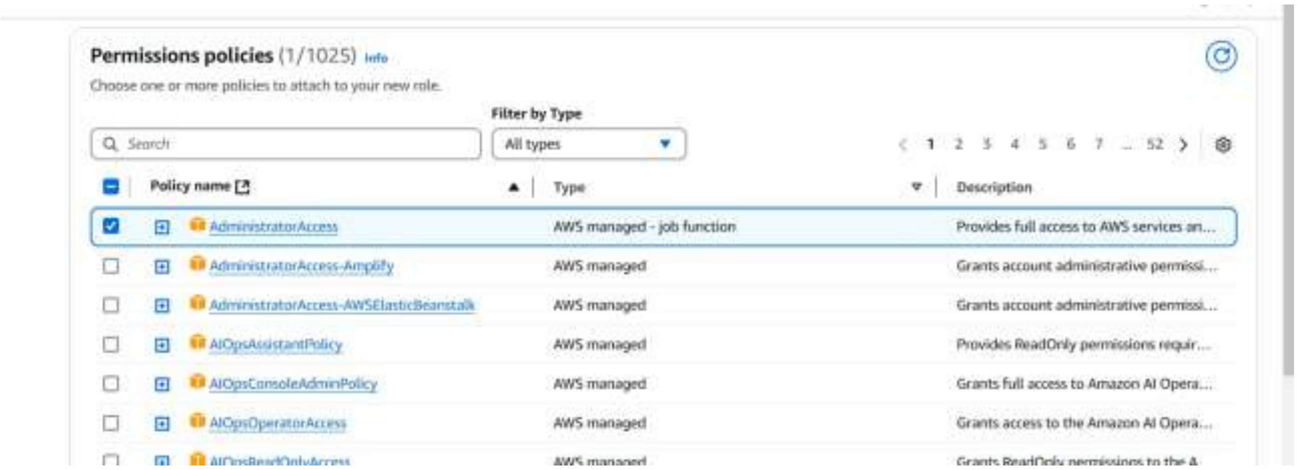
Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case: DMS

Choose a use case for the specified service.
Use case
● DMS
Allows Database Migration Service to call AWS services on your behalf

- Select required permissions or administrator access and click next



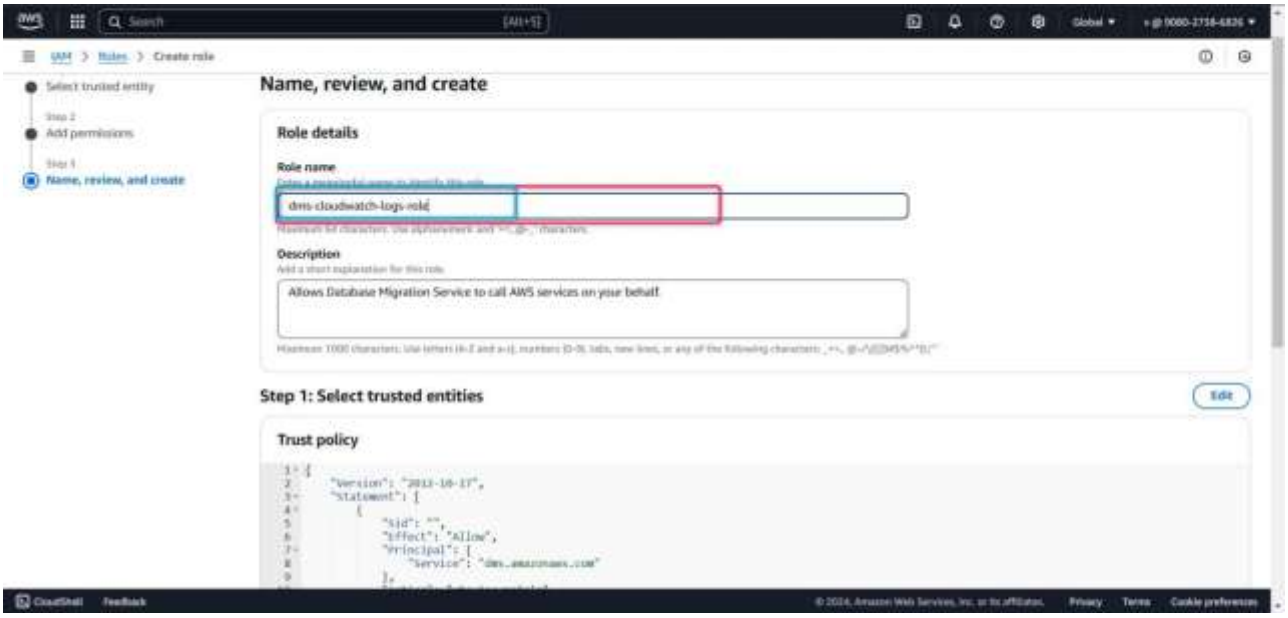
Permissions policies (1/1025) Info

Choose one or more policies to attach to your new role.

Filter by Type: All types

	Policy name	Type	Description
☑	AdministratorAccess	AWS managed - job function	Provides full access to AWS services an...
□	AdministratorAccess-Amplify	AWS managed	Grants account administrative permissi...
□	AdministratorAccess-AWSElasticBeanstalk	AWS managed	Grants account administrative permissi...
□	AIOpsAssistantPolicy	AWS managed	Provides ReadOnly permissions requir...
□	AIOpsConsoleAdminPolicy	AWS managed	Grants full access to Amazon AI Opera...
□	AIOpsOperatorAccess	AWS managed	Grants access to the Amazon AI Opera...
□	AIOpsReadOnlyAccess	AWS managed	Grants ReadOnly permissions to the A...

- Give name dms-cloudwatch-log-role only
- Then click on create role



IAM > Roles > Create role

Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

Name, review, and create

Role details

Role name: dms-cloudwatch-logs-role

Description: Allows Database Migration Service to call AWS services on your behalf

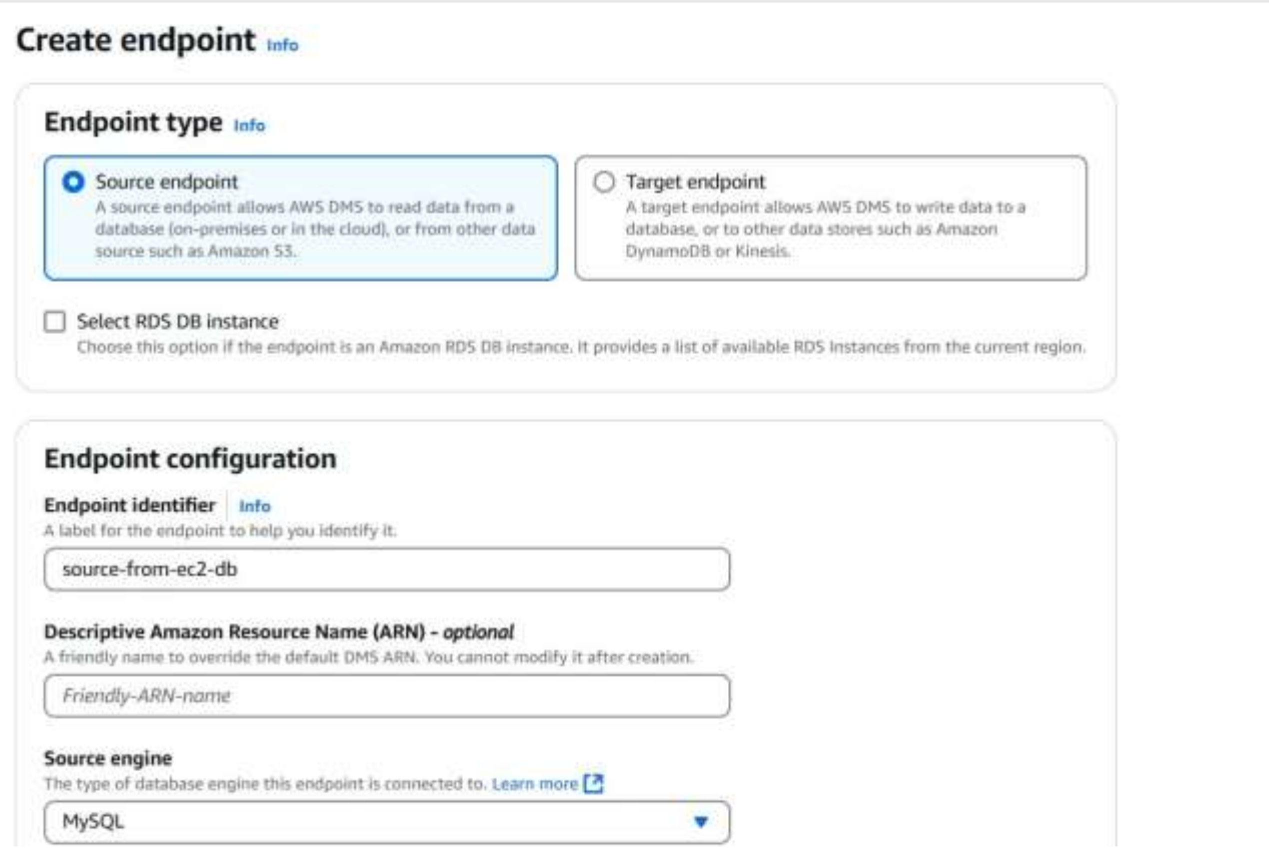
Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "dms.amazonaws.com"
            },
```

SOURCE ENDPOINT CREATION

- Open the endpoint page in dms service
- Select the endpoint type is source
- Enter the identifier named
- Select the your engine type my end is sql



Create endpoint Info

Endpoint type Info

● Source endpoint
A source endpoint allows AWS DMS to read data from a database (on-premises or in the cloud), or from other data source such as Amazon S3.

○ Target endpoint
A target endpoint allows AWS DMS to write data to a database, or to other data stores such as Amazon DynamoDB or Kinesis.

□ Select RDS DB instance
Choose this option if the endpoint is an Amazon RDS DB instance. It provides a list of available RDS instances from the current region.

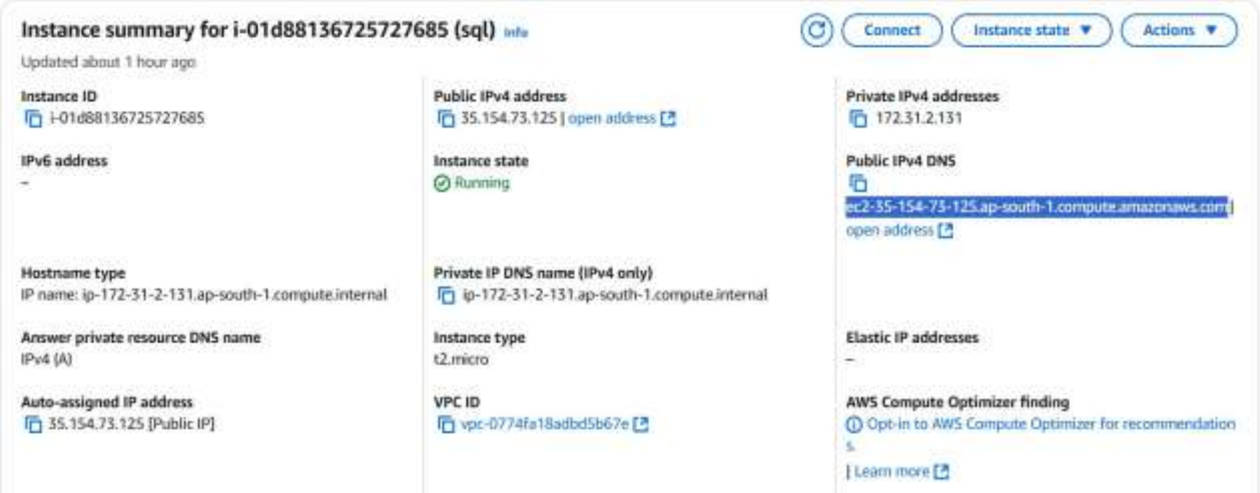
Endpoint configuration

Endpoint identifier Info
A label for the endpoint to help you identify it.
source-from-ec2-db

Descriptive Amazon Resource Name (ARN) - *optional*
A friendly name to override the default DMS ARN. You cannot modify it after creation.
Friendly-ARN-name

Source engine
The type of database engine this endpoint is connected to. Learn more
MySQL

- Open the ec2-db instance page in ec2 console and copy the public ipv4 dns name OR private ipv4 dns name



Instance summary for i-01d88136725727685 (sql) Info
Updated about 1 hour ago

Instance ID
i-01d88136725727685

IPv6 address
–

Hostname type
IP name: ip-172-31-2-131.ap-south-1.compute.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
35.154.73.125 [Public IP]

Public IPv4 address
35.154.73.125 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-2-131.ap-south-1.compute.internal

Instance type
t2.micro

VPC ID
vpc-0774fa18adbd5b67e

Private IPv4 addresses
172.31.2.131

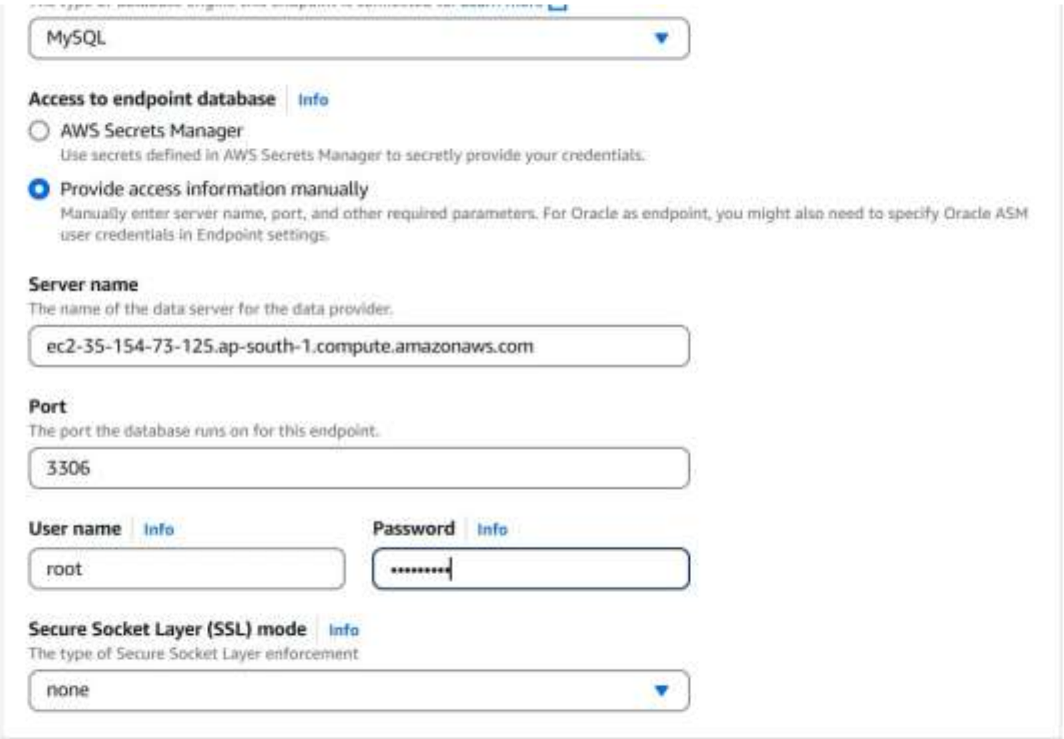
Public IPv4 DNS
ec2-35-154-73-125.ap-south-1.compute.amazonaws.com | open address

Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

- Select the access to end point database is provide access information manullay

- Paste the server name as your ec2-db instance public dns name or private dns name
- Enter the port is 3306 for sql
- You have created one user for reapplication enter that User name my end is root
- Enter the Password of your user “”Admin@123” this is the password for ec2-db
- We are created a user with the name of root and password is Admin@123



The type of database engine this endpoint is connected to. Learn more

MySQL

Access to endpoint database Info

○ AWS Secrets Manager
Use secrets defined in AWS Secrets Manager to secretly provide your credentials.

● Provide access information manually
Manually enter server name, port, and other required parameters. For Oracle as endpoint, you might also need to specify Oracle ASM user credentials in Endpoint settings.

Server name
The name of the data server for the data provider.
ec2-35-154-73-125.ap-south-1.compute.amazonaws.com

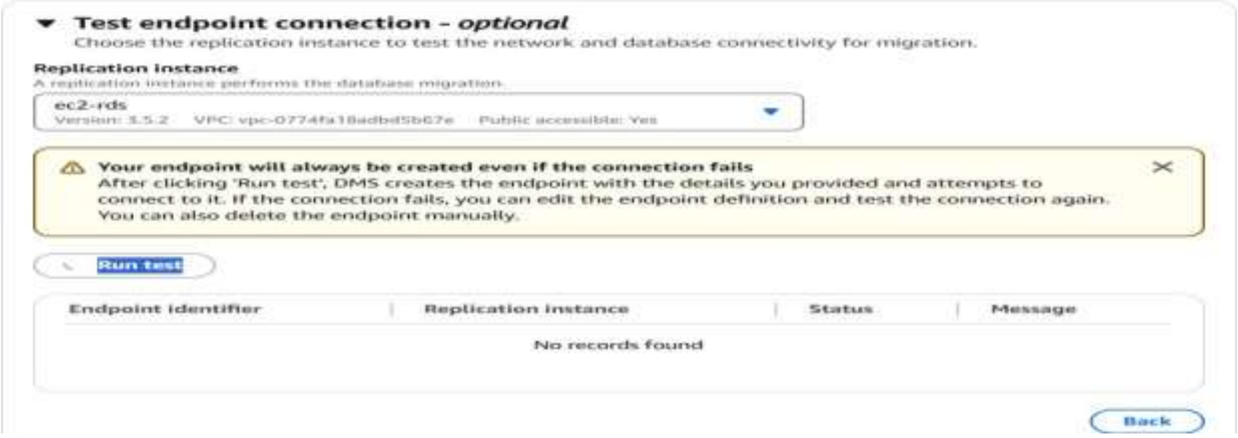
Port
The port the database runs on for this endpoint.
3306

User name Info
root

Password Info
•••••••••

Secure Socket Layer (SSL) mode Info
The type of Secure Socket Layer enforcement
none

- Scroll down test the endpoint connection
- Click on run test



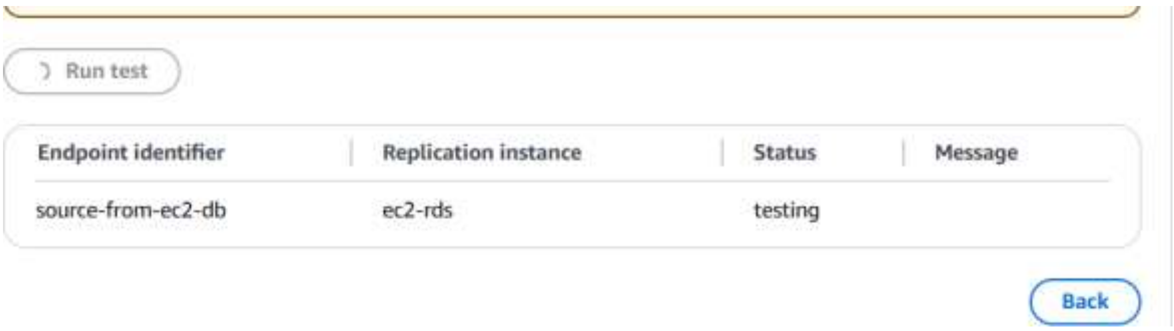
Test endpoint connection - *optional*
Choose the replication instance to test the network and database connectivity for migration.

Replication instance
A replication instance performs the database migration.
ec2-rds
Version: 3.5.2 VPC: vpc-0774fa18adbd5b67e Public accessible: Yes

Your endpoint will always be created even if the connection fails
After clicking 'Run test', DMS creates the endpoint with the details you provided and attempts to connect to it. If the connection fails, you can edit the endpoint definition and test the connection again. You can also delete the endpoint manually.

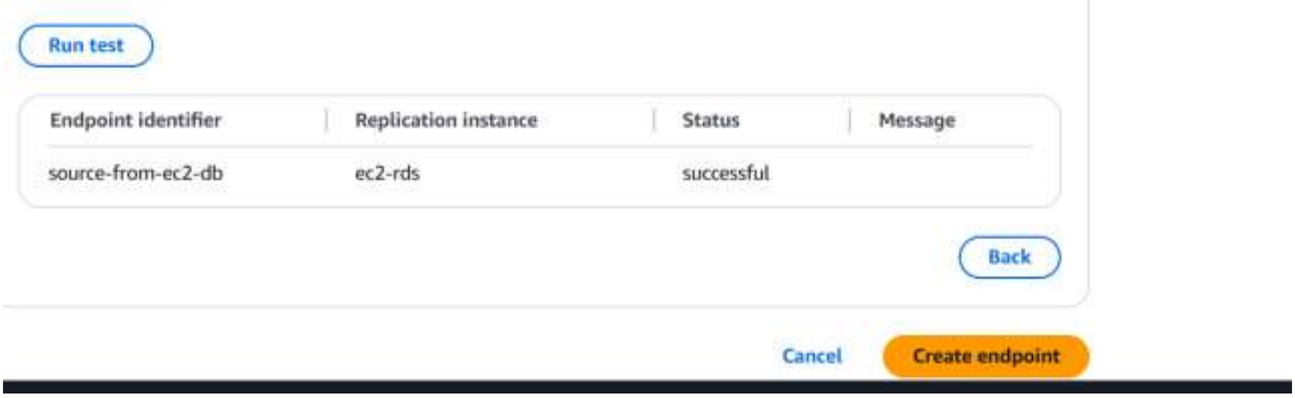
Endpoint identifier	Replication instance	Status	Message
No records found			

- It is in testing state



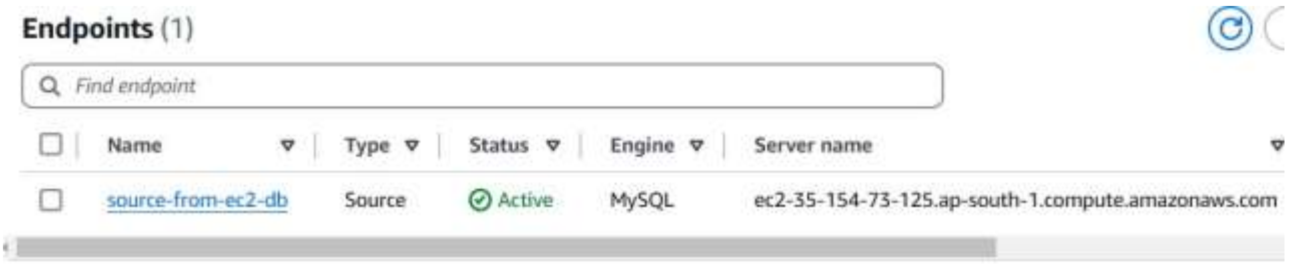
Endpoint identifier	Replication instance	Status	Message
source-from-ec2-db	ec2-rds	testing	

- Status is successful it mans your ec2-db is connected successfully
- Click on create endpoint



Endpoint identifier	Replication instance	Status	Message
source-from-ec2-db	ec2-rds	successful	

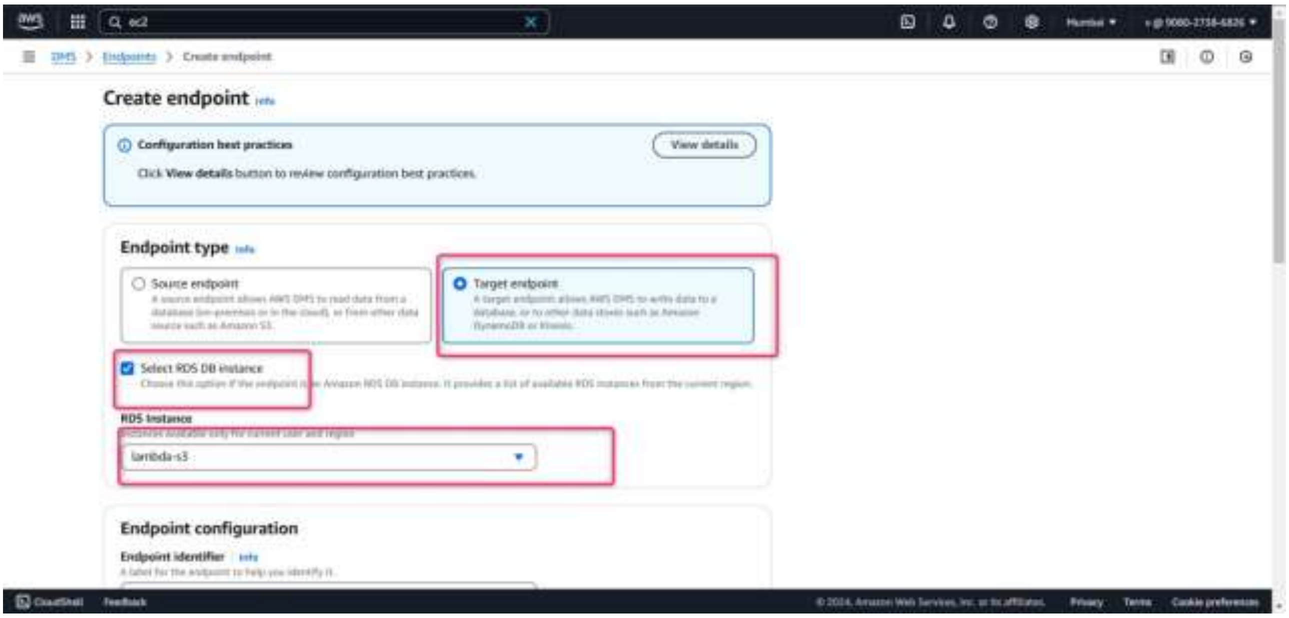
- Source endpoint is created successfully
- Next create another endpoint that target
- Click on create endpoint



☐	Name	Type	Status	Engine	Server name
☐	source-from-ec2-db	Source	✓ Active	MySQL	ec2-35-154-73-125.ap-south-1.compute.amazonaws.com

TARGET ENDPOINT CREATION

- Select endpoint type is target endpoint
- Click on Select rds db instance if we select this check box it will fetch the details automatically
- Select your rds instance check box



DMS > Endpoints > Create endpoint

Create endpoint Info

Configuration best practices

Click **View details** button to review configuration best practices.

Endpoint type Info

○ Source endpoint
A source endpoint allows AWS DMS to read data from a database (on-premises or in the cloud), or from other data source such as Amazon S3.

● Target endpoint
A target endpoint allows AWS DMS to write data to a database, or to other data stores such as Amazon DynamoDB or Kinesis.

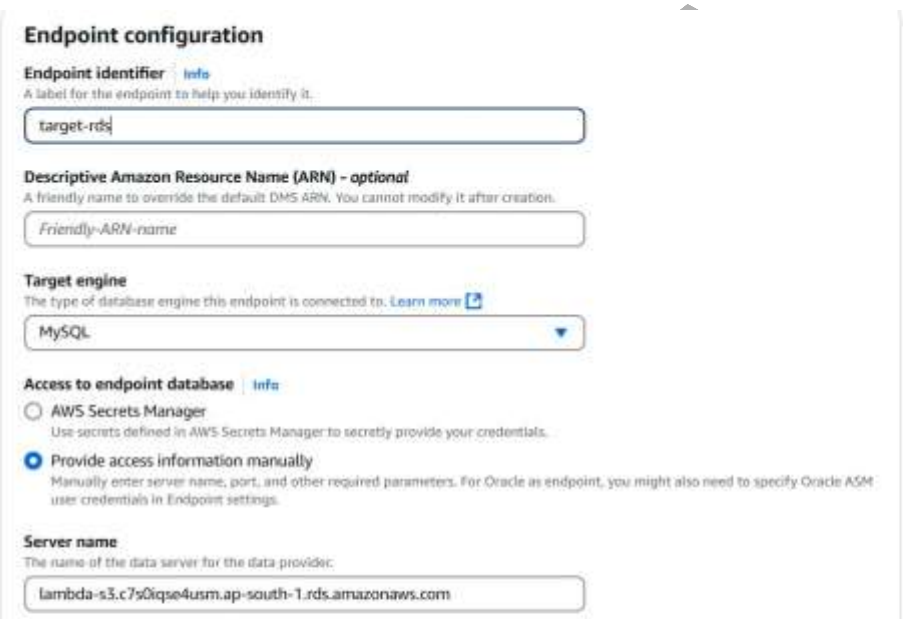
☑ Select RDS DB instance
Choose this option if the endpoint is an Amazon RDS DB instance. It provides a list of available RDS instances from the current region.

RDS Instance
Instances available only for current user and region
lambda-s3

Endpoint configuration

Endpoint identifier Info
A label for the endpoint to help you identify it.

- Enter the identifier name
- Select your engine type my end is sql
- Select the access to end point database is provide access information manullay



Endpoint configuration

Endpoint identifier Info
A label for the endpoint to help you identify it.
target-rds

Descriptive Amazon Resource Name (ARN) - *optional*
A friendly name to override the default DMS ARN. You cannot modify it after creation.
Friendly-ARN-name

Target engine
The type of database engine this endpoint is connected to. Learn more
MySQL

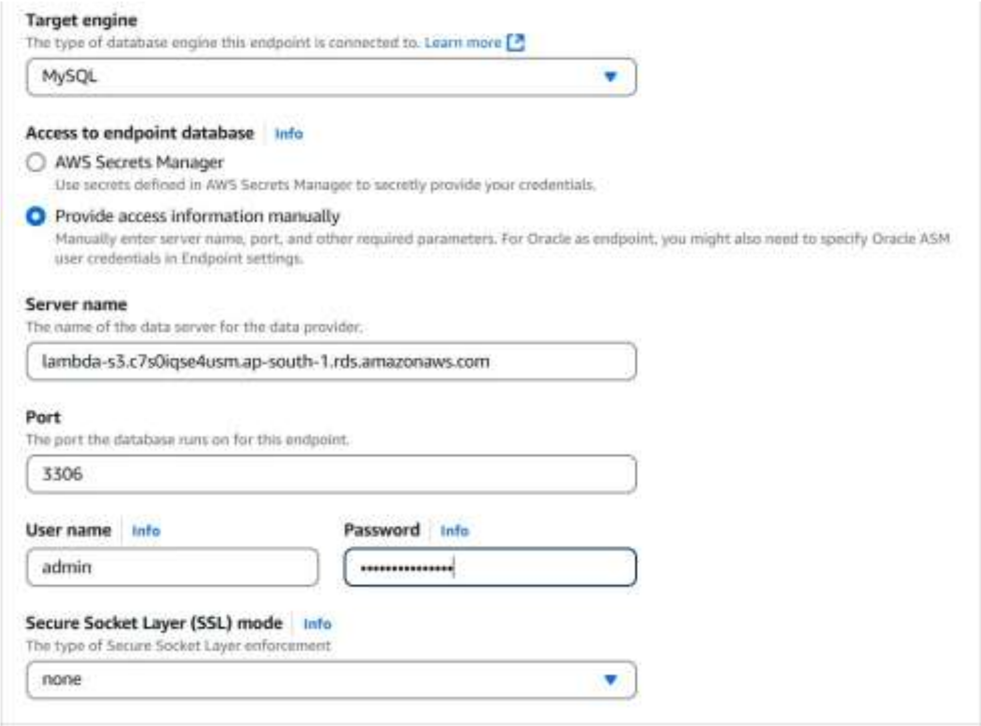
Access to endpoint database Info

○ AWS Secrets Manager
Use secrets defined in AWS Secrets Manager to secretly provide your credentials.

● Provide access information manually
Manually enter server name, port, and other required parameters. For Oracle as endpoint, you might also need to specify Oracle ASM user credentials in Endpoint settings.

Server name
The name of the data server for the data provider.
lambda-s3.c7s0iqse4usm.ap-south-1.rds.amazonaws.com

- The server name is your rds endpoint it will automatically fetched
- Enter your user name and password



Target engine

The type of database engine this endpoint is connected to. Learn more

MySQL ▼

Access to endpoint database Info

○ AWS Secrets Manager

Use secrets defined in AWS Secrets Manager to secretly provide your credentials.

● Provide access information manually

Manually enter server name, port, and other required parameters. For Oracle as endpoint, you might also need to specify Oracle ASM user credentials in Endpoint settings.

Server name

The name of the data server for the data provider.

lambda-s3.c7s0iqse4usm.ap-south-1.rds.amazonaws.com

Port

The port the database runs on for this endpoint.

3306

User name Info

admin

Password Info

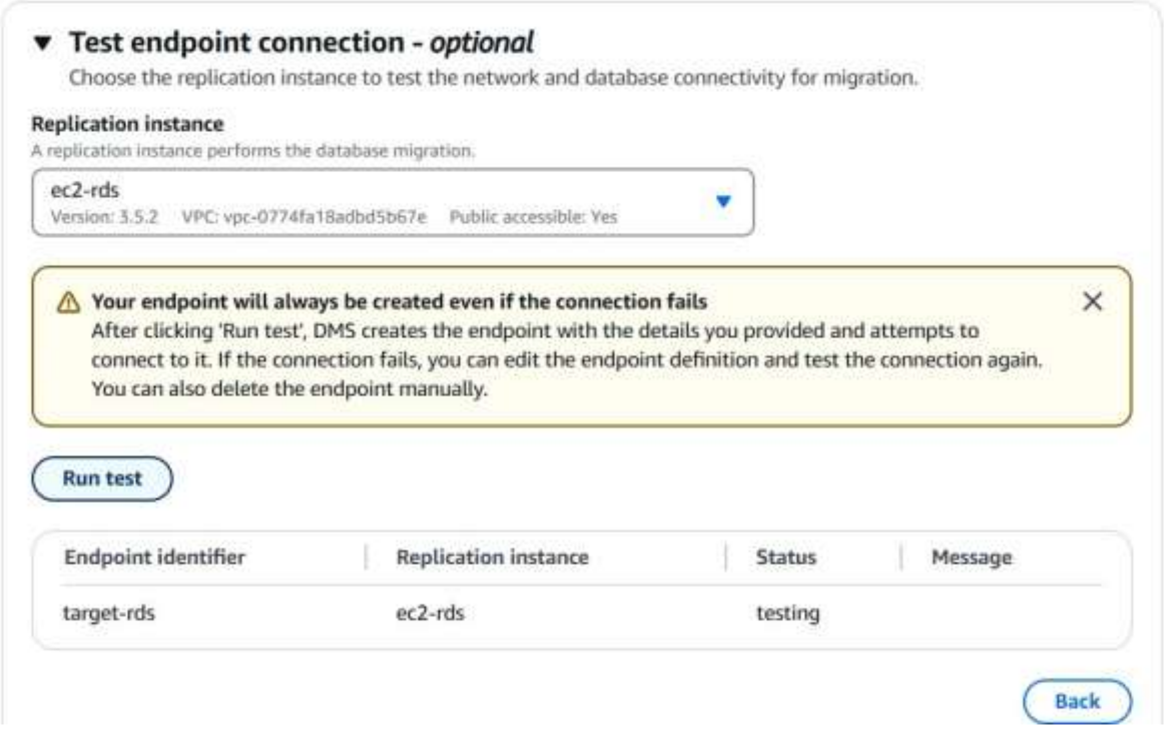
••••••••••••••••

Secure Socket Layer (SSL) mode Info

The type of Secure Socket Layer enforcement

none ▼

- Scroll down to test the endpoint connection
- Click on run test
- It is testing state



▼ **Test endpoint connection - *optional***

Choose the replication instance to test the network and database connectivity for migration.

Replication instance

A replication instance performs the database migration.

ec2-rds ▼
Version: 3.5.2 VPC: vpc-0774fa18adbd5b67e Public accessible: Yes

⚠ **Your endpoint will always be created even if the connection fails**

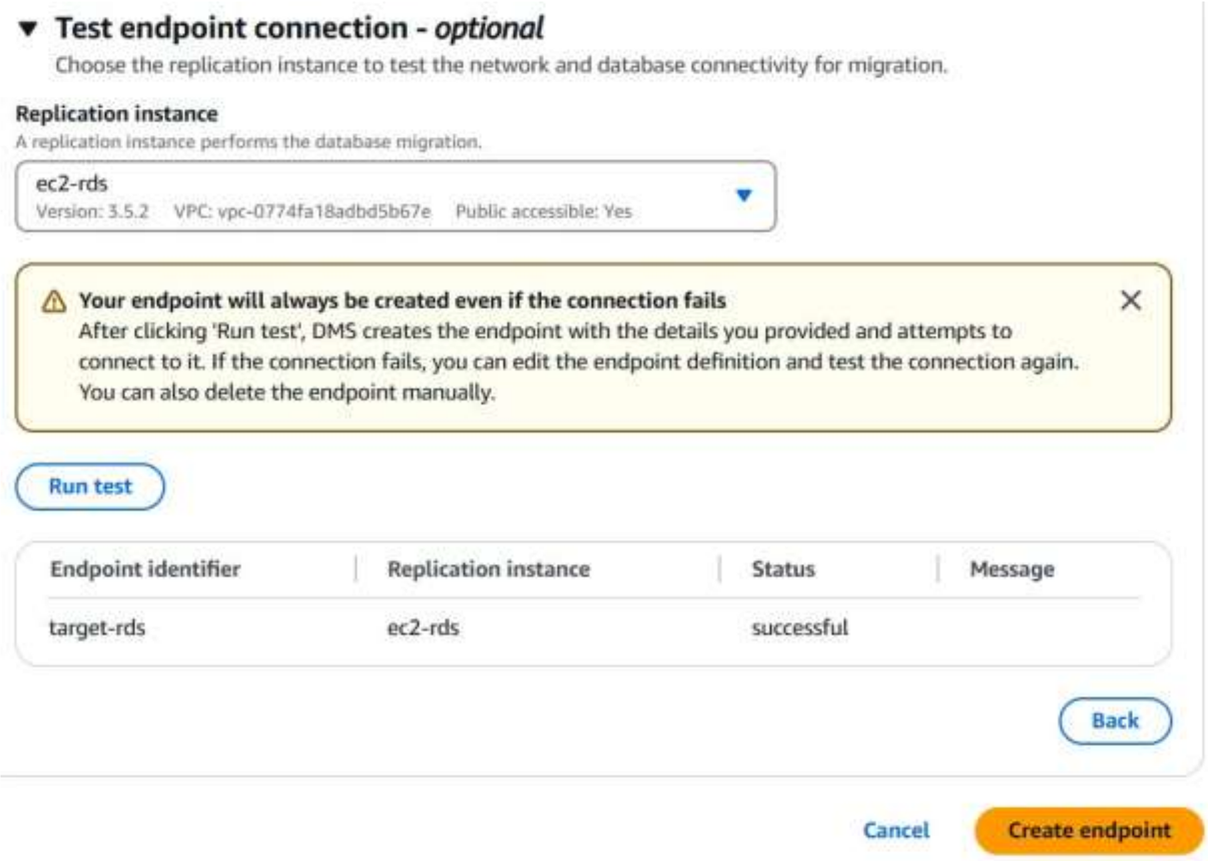
After clicking 'Run test', DMS creates the endpoint with the details you provided and attempts to connect to it. If the connection fails, you can edit the endpoint definition and test the connection again. You can also delete the endpoint manually.

Run test

Endpoint identifier	Replication instance	Status	Message
target-rds	ec2-rds	testing	

Back

- Status is successful it means your rds db is connected successfully
- Click on create endpoint



▼ **Test endpoint connection - *optional***

Choose the replication instance to test the network and database connectivity for migration.

Replication instance

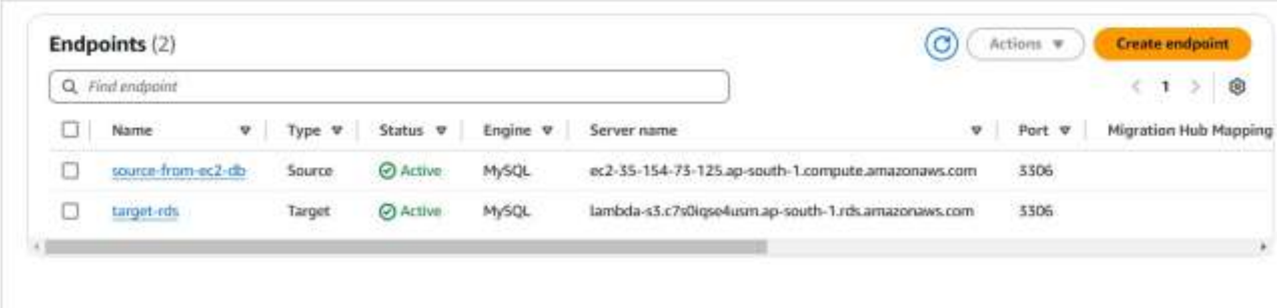
A replication instance performs the database migration.

ec2-rds
Version: 3.5.2 VPC: vpc-0774fa18adbd5b67e Public accessible: Yes

⚠ **Your endpoint will always be created even if the connection fails**
After clicking 'Run test', DMS creates the endpoint with the details you provided and attempts to connect to it. If the connection fails, you can edit the endpoint definition and test the connection again. You can also delete the endpoint manually.

Endpoint identifier	Replication instance	Status	Message
target-rds	ec2-rds	successful	

- These are our both end points

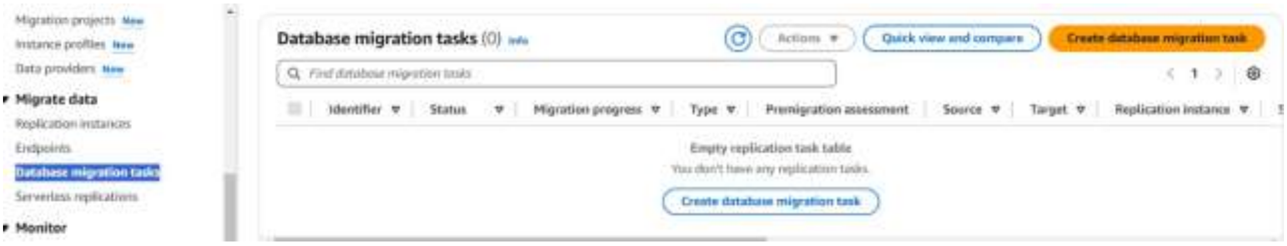


Endpoints (2)

☐	Name	Type	Status	Engine	Server name	Port	Migration Hub Mapping
☐	source-from-ec2-db	Source	✓ Active	MySQL	ec2-35-154-73-125.ap-south-1.compute.amazonaws.com	3306	
☐	target-rds	Target	✓ Active	MySQL	lambda-s3.c7s0iqse4usm.ap-south-1.rds.amazonaws.com	3306	

DATABASE MIGRATION TASK CREATION

- Click on create database migration task on migrate data section

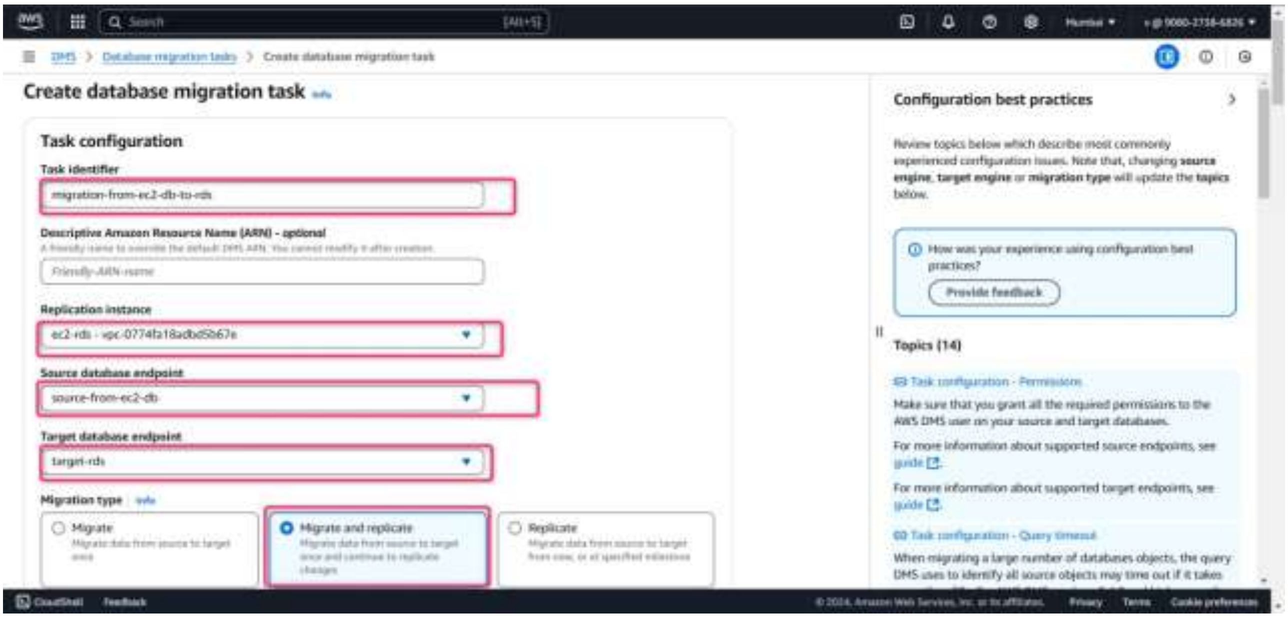


Database migration tasks (0) Info

Empty replication task table
You don't have any replication tasks.

- Enter your identifier name
- Select your replication instance
- Select your source and target endpoints

- Select migration type is migrate and replicate if we select migrate and replicate only ec2-db new records are automatically updated into target database
- Here database migration task is continuously ruined



DMS > Database migration tasks > Create database migration task

Create database migration task

Task configuration

Task identifier: migration-from-ec2-db-to-rds

Descriptive Amazon Resource Name (ARN) - optional
A friendly name to override the default DMS ARN. You cannot modify it after creation.
Friendly-ARN-name

Replication instance: ec2-rds - vpc-0774fa18adbd5b67e

Source database endpoint: source-from-ec2-db

Target database endpoint: target-rds

Migration type
○ Migrate — Migrate data from source to target once
● Migrate and replicate — Migrate data from source to target once and continue to replicate changes
○ Replicate — Migrate data from source to target from now, or at specified milestones

Configuration best practices

Review topics below which describe most commonly experienced configuration issues. Note that, changing **source engine**, **target engine** or **migration type** will update the **topics** below.

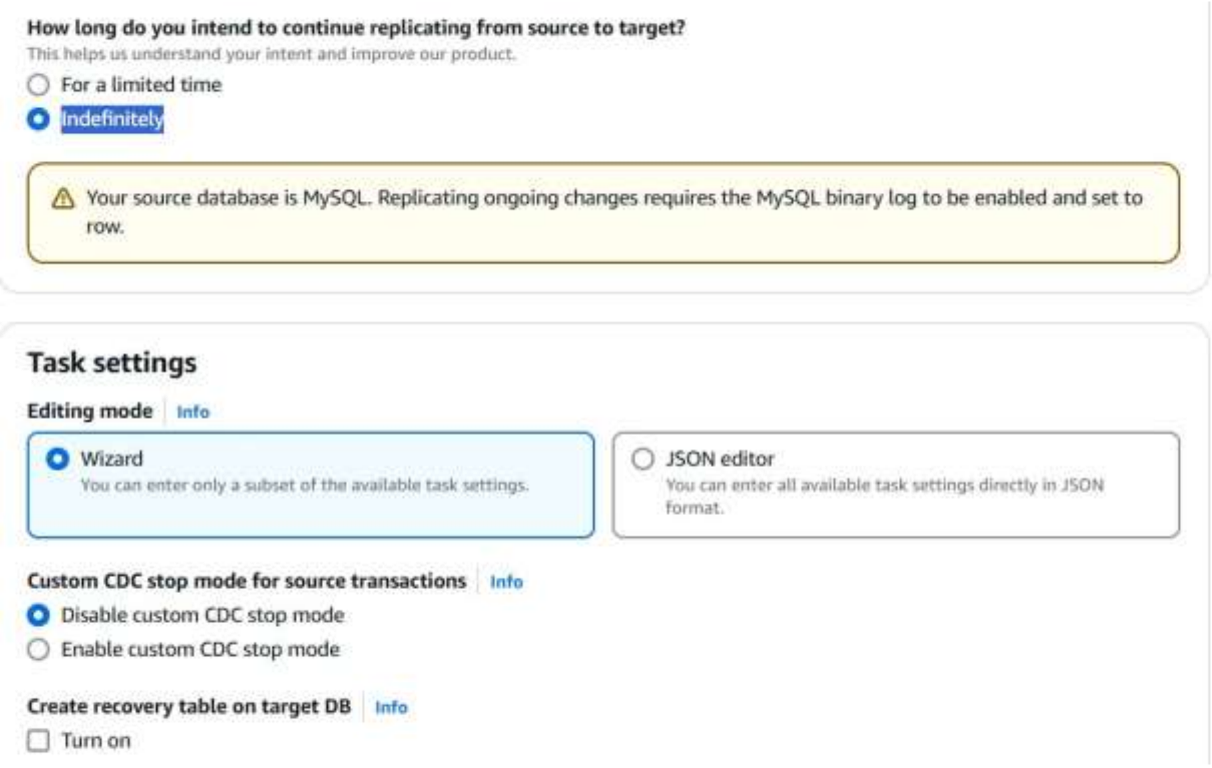
How was your experience using configuration best practices?

Topics (14)

Task configuration - Permissions
Make sure that you grant all the required permissions to the AWS DMS user on your source and target databases.
For more information about supported source endpoints, see guide.
For more information about supported target endpoints, see guide.

Task configuration - Query timeout
When migrating a large number of databases objects, the query DMS uses to identify all source objects may time out if it takes

- Select the below options as usual



How long do you intend to continue replicating from source to target?
This helps us understand your intent and improve our product.
○ For a limited time
● Indefinitely

⚠ Your source database is MySQL. Replicating ongoing changes requires the MySQL binary log to be enabled and set to row.

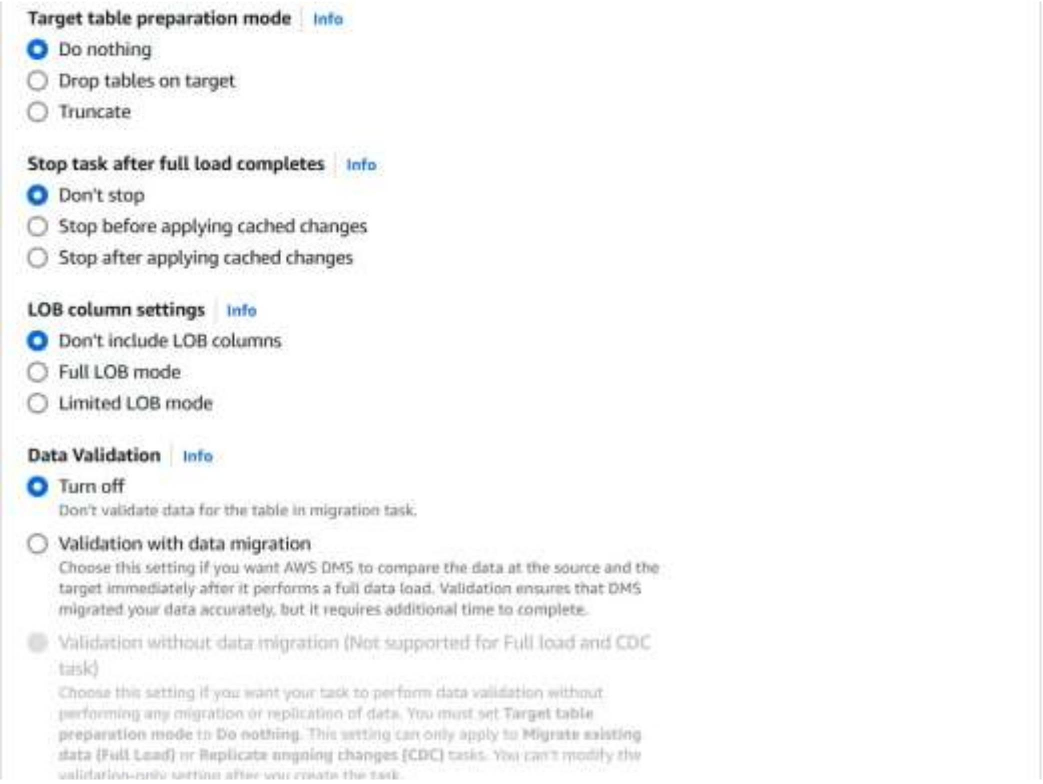
Task settings

Editing mode | Info
● Wizard — You can enter only a subset of the available task settings.
○ JSON editor — You can enter all available task settings directly in JSON format.

Custom CDC stop mode for source transactions | Info
● Disable custom CDC stop mode
○ Enable custom CDC stop mode

Create recovery table on target DB | Info
□ Turn on

- Select the below options as usual



Target table preparation mode Info

● Do nothing

○ Drop tables on target

○ Truncate

Stop task after full load completes Info

● Don't stop

○ Stop before applying cached changes

○ Stop after applying cached changes

LOB column settings Info

● Don't include LOB columns

○ Full LOB mode

○ Limited LOB mode

Data Validation Info

● Turn off

Don't validate data for the table in migration task.

○ Validation with data migration

Choose this setting if you want AWS DMS to compare the data at the source and the target immediately after it performs a full data load. Validation ensures that DMS migrated your data accurately, but it requires additional time to complete.

○ Validation without data migration (Not supported for Full load and CDC task)

Choose this setting if you want your task to perform data validation without performing any migration or replication of data. You must set **Target table preparation mode** to **Do nothing**. This setting can only apply to **Migrate existing data (Full Load)** or **Replicate ongoing changes (CDC)** tasks. You can't modify the validation-only setting after you create the task.

- In table mappings
- Click on Selection rules
- Then click on add new selection rule
- If we have multiple tables we need to add multiple selections rules

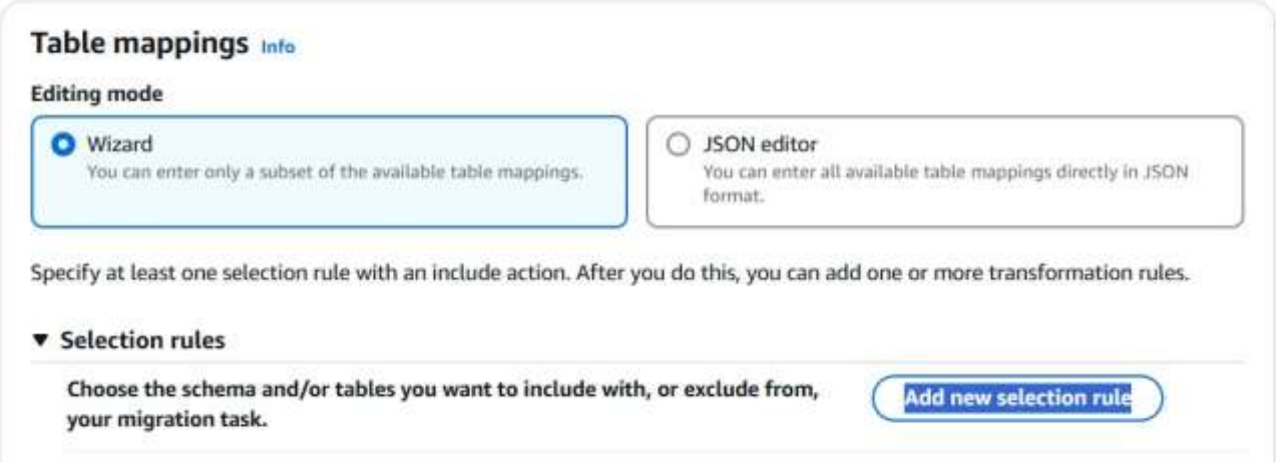


Table mappings Info

Editing mode

● Wizard
You can enter only a subset of the available table mappings.

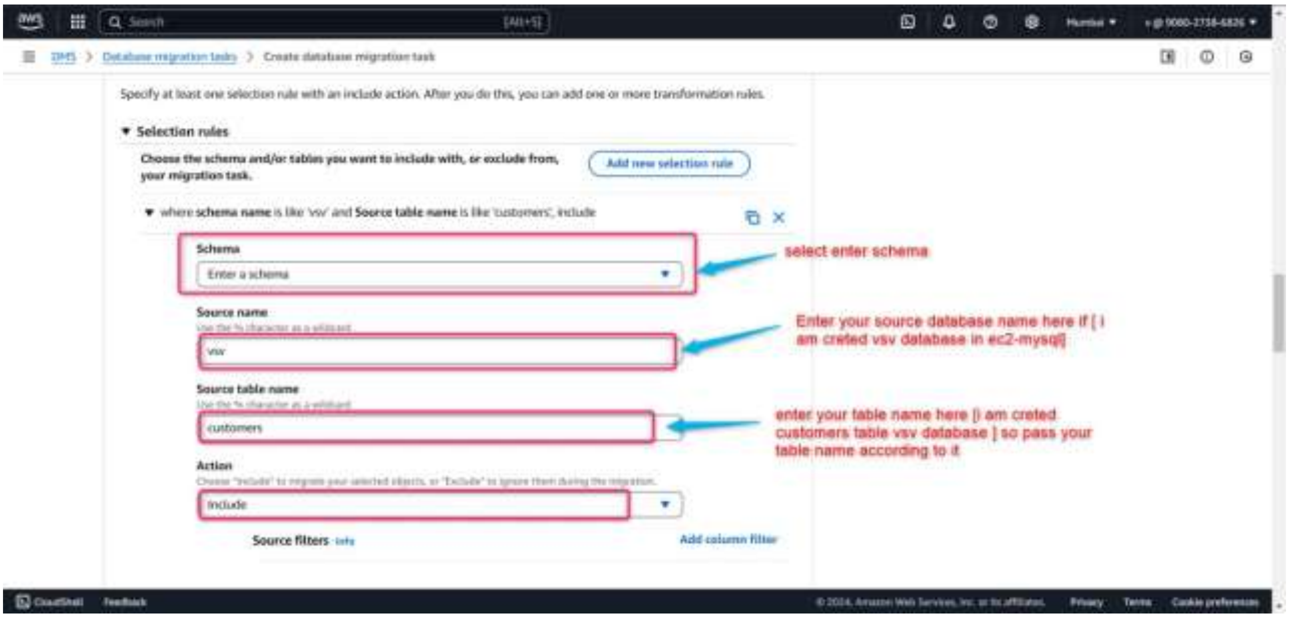
○ JSON editor
You can enter all available table mappings directly in JSON format.

Specify at least one selection rule with an include action. After you do this, you can add one or more transformation rules.

▼ **Selection rules**

Choose the schema and/or tables you want to include with, or exclude from, your migration task.

- In schema section select enter schema
- In source name enter your source database name my end is vsv
- Enter source table name enter your table name if u want to migrate my end is customers



DMS > Database migration tasks > Create database migration task

Specify at least one selection rule with an include action. After you do this, you can add one or more transformation rules.

▼ Selection rules

Choose the schema and/or tables you want to include with, or exclude from, your migration task.

▼ where schema name is like 'vsv' and Source table name is like 'customers', include

Schema: Enter a schema

select enter schema

Source name (Use the % character as a wildcard): vsv

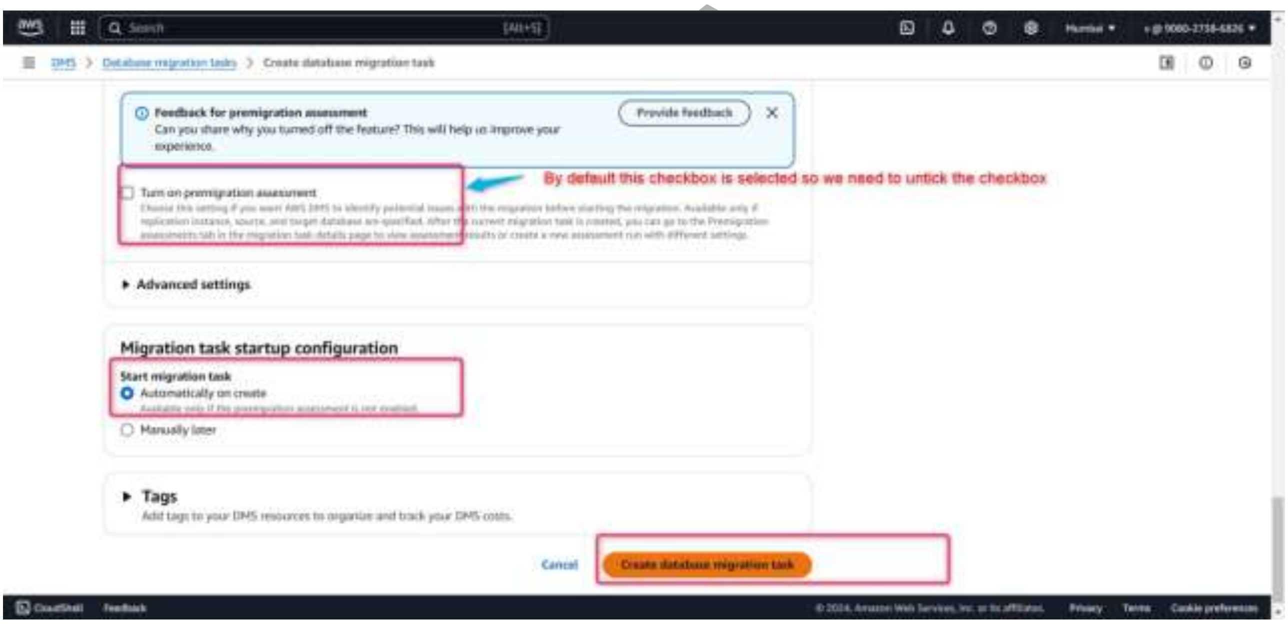
Enter your source database name here if [i am creted vsv database in ec2-mysql]

Source table name (Use the % character as a wildcard): customers

enter your table name here [i am creted customers table vsv database] so pass your table name according to it

Action (Choose "Include" to migrate your selected objects, or "Exclude" to ignore them during the migration.): Include

- Deselect turn on premigration assessment
- Select migration task startup configuration is automatically on create
- Click on create database migration task



DMS > Database migration tasks > Create database migration task

Feedback for premigration assessment
Can you share why you turned off the feature? This will help us improve your experience.

□ Turn on premigration assessment
Choose this setting if you want AWS DMS to identify potential issues with the migration before starting the migration. Available only if replication instance, source, and target database are specified. After the current migration task is created, you can go to the Premigration assessments tab in the migration task details page to view assessment results or create a new assessment run with different settings.

By default this checkbox is selected so we need to untick the checkbox

▸ Advanced settings

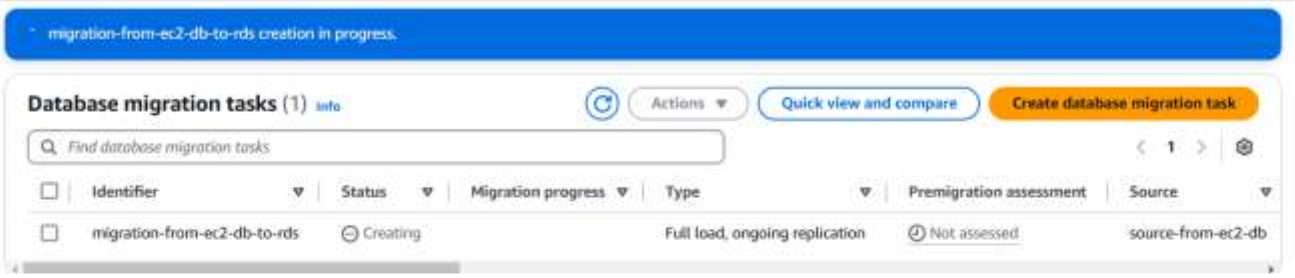
Migration task startup configuration

Start migration task
● Automatically on create
Available only if the premigration assessment is not enabled.
○ Manually later

▸ Tags
Add tags to your DMS resources to organize and track your DMS costs.

Provide feedback
Cancel
Create database migration task
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

- It is creating state

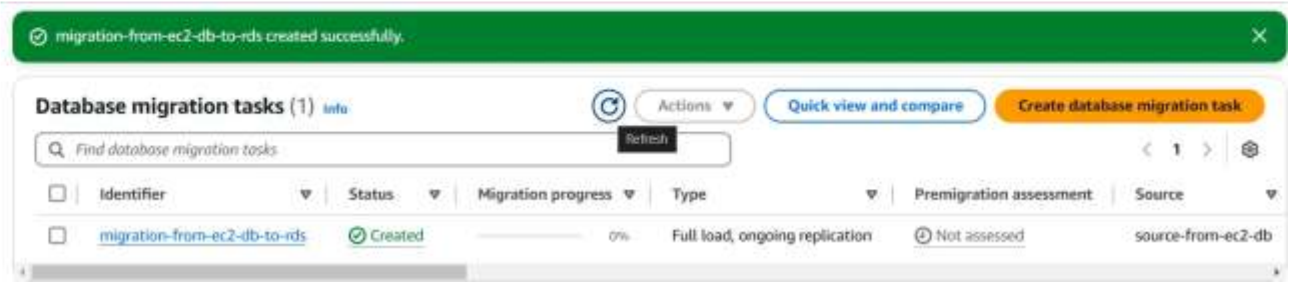


migration-from-ec2-db-to-rds creation in progress.

Database migration tasks (1) Info

Identifier	Status	Migration progress	Type	Premigration assessment	Source
migration-from-ec2-db-to-rds	Creating		Full load, ongoing replication	Not assessed	source-from-ec2-db

- Status created

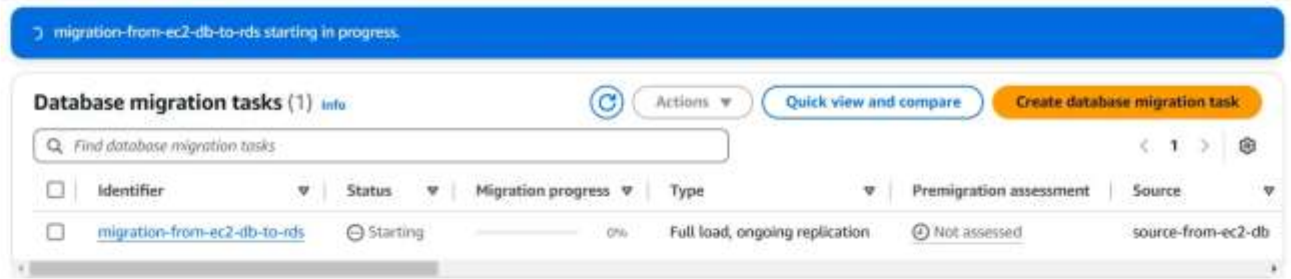


migration-from-ec2-db-to-rds created successfully.

Database migration tasks (1) Info

☐	Identifier	Status	Migration progress	Type	Premigration assessment	Source
☐	migration-from-ec2-db-to-rds	Created	0%	Full load, ongoing replication	Not assessed	source-from-ec2-db

- Status starting

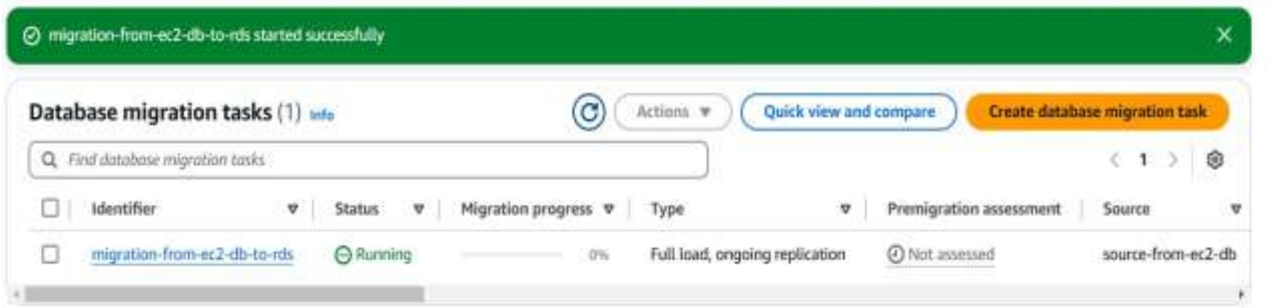


migration-from-ec2-db-to-rds starting in progress.

Database migration tasks (1) Info

☐	Identifier	Status	Migration progress	Type	Premigration assessment	Source
☐	migration-from-ec2-db-to-rds	Starting	0%	Full load, ongoing replication	Not assessed	source-from-ec2-db

- Status running

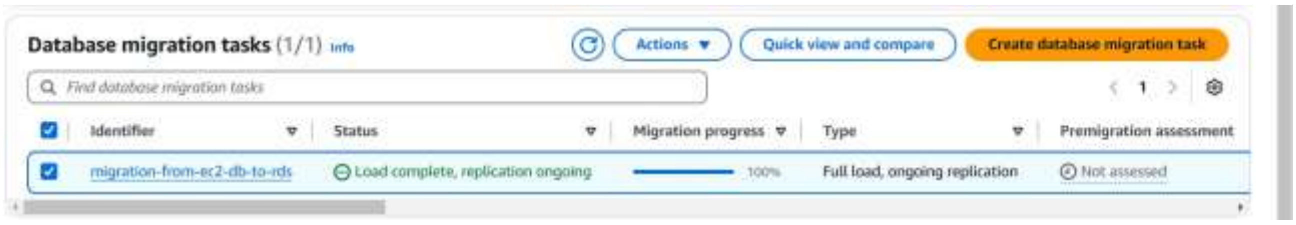


migration-from-ec2-db-to-rds started successfully

Database migration tasks (1) Info

☐	Identifier	Status	Migration progress	Type	Premigration assessment	Source
☐	migration-from-ec2-db-to-rds	Running	0%	Full load, ongoing replication	Not assessed	source-from-ec2-db

- Status load completed it means data migrated successfully



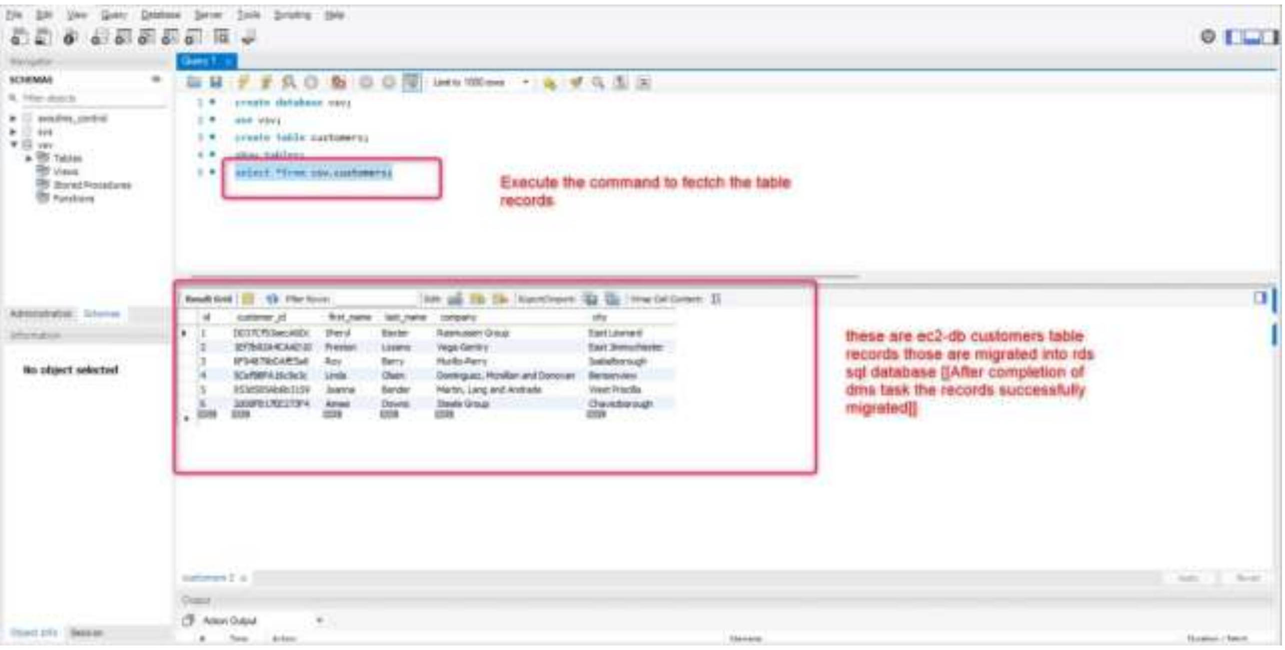
Database migration tasks (1/1) Info

☑	Identifier	Status	Migration progress	Type	Premigration assessment
☑	migration-from-ec2-db-to-rds	Load complete, replication ongoing	100%	Full load, ongoing replication	Not assessed

- Connect to your rds database from sql workbench or ec2 server
- Enter the below query to fetch the table records

```
select *from vsv.customers;
```

- The ec2-db previous coustomers table is migrated successfully



```
create database vsv;
use vsv;
create table customers;
show tables;
select *from vsv.customers;
```

Execute the command to fectch the table records

id	customer_id	first_name	last_name	company	city
1	DD37Cf93aecA6Dc	Sheryl	Baxter	Rasmussen Group	East Leonard
2	1Ef7b82A4CAAD10	Preston	Lozano	Vega-Gentry	East Jimmychester
3	6F94879bDAfE5a6	Roy	Berry	Murillo-Perry	Isabelborough
4	5Cef8BFA16c5e3c	Linda	Olsen	Dominguez, Mcmillan and Donovan	Bensonview
5	053d585Ab6b3159	Joanna	Bender	Martin, Lang and Andrade	West Priscilla
6	2d08FB17EE273F4	Aimee	Downs	Steele Group	Chavezborough
NULL	NULL	NULL	NULL	NULL	NULL

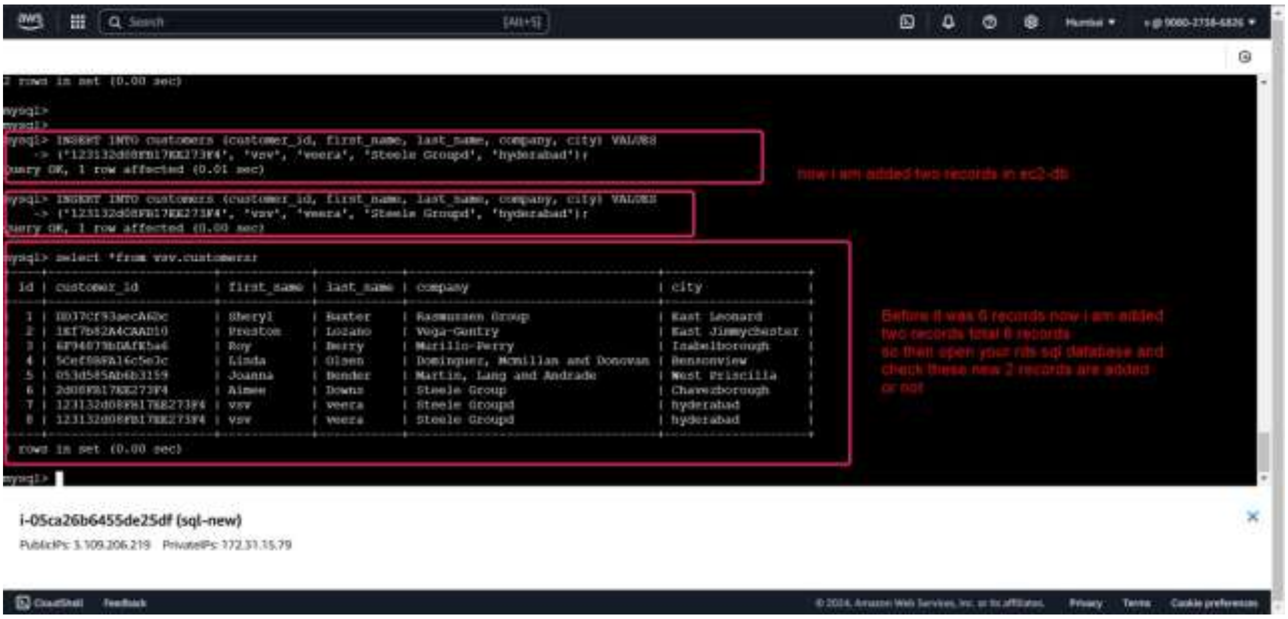
these are ec2-db customers table records those are migrated into rds sql database [[After completion of dms task the records successfully migrated]]

- Now connect to ec2-db add few records

INSERT INTO customers (customer_id, first_name, last_name, company, city) VALUES ('123132d08FB17EE273F4', 'vsv', 'veera', 'Steele Groupd', 'hyderabad');

- Enter below query in ec2-db total we have 8 records last two records are new records

select *from vsv.customers;



```
2 rows in set (0.00 sec)

mysql>
mysql>
mysql> INSERT INTO customers (customer_id, first_name, last_name, company, city) VALUES
    -> ('123132d08FB17EE273F4', 'vsv', 'veera', 'Steele Groupd', 'hyderabad');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO customers (customer_id, first_name, last_name, company, city) VALUES
    -> ('123132d08FB17EE273F4', 'vsv', 'veera', 'Steele Groupd', 'hyderabad');
Query OK, 1 row affected (0.00 sec)

mysql> select *from vsv.customers;
+----+----------------------+------------+-----------+---------------------------------+-------------------+
| id | customer_id          | first_name | last_name | company                         | city              |
+----+----------------------+------------+-----------+---------------------------------+-------------------+
|  1 | DD37Cf93aecA6Dc      | Sheryl     | Baxter    | Rasmussen Group                 | East Leonard      |
|  2 | 1Ef7b82A4CAAD10      | Preston    | Lozano    | Vega-Gentry                     | East Jimmychester |
|  3 | 6F94879bDAfE5a6      | Roy        | Berry     | Murillo-Perry                   | Isabelborough     |
|  4 | 5Cef8BFA16c5e3c      | Linda      | Olsen     | Dominguez, Mcmillan and Donovan | Bensonview        |
|  5 | 053d585Ab6b3159      | Joanna     | Bender    | Martin, Lang and Andrade        | West Priscilla    |
|  6 | 2d08FB17EE273F4      | Aimee      | Downs     | Steele Group                    | Chavezborough     |
|  7 | 123132d08FB17EE273F4 | vsv        | veera     | Steele Groupd                   | hyderabad         |
|  8 | 123132d08FB17EE273F4 | vsv        | veera     | Steele Groupd                   | hyderabad         |
+----+----------------------+------------+-----------+---------------------------------+-------------------+
8 rows in set (0.00 sec)

mysql>
```

now i am added two records in ec2-db

Before it was 6 records now i am added two records total 8 records so then open your rds sql database and check these new 2 records are added or not

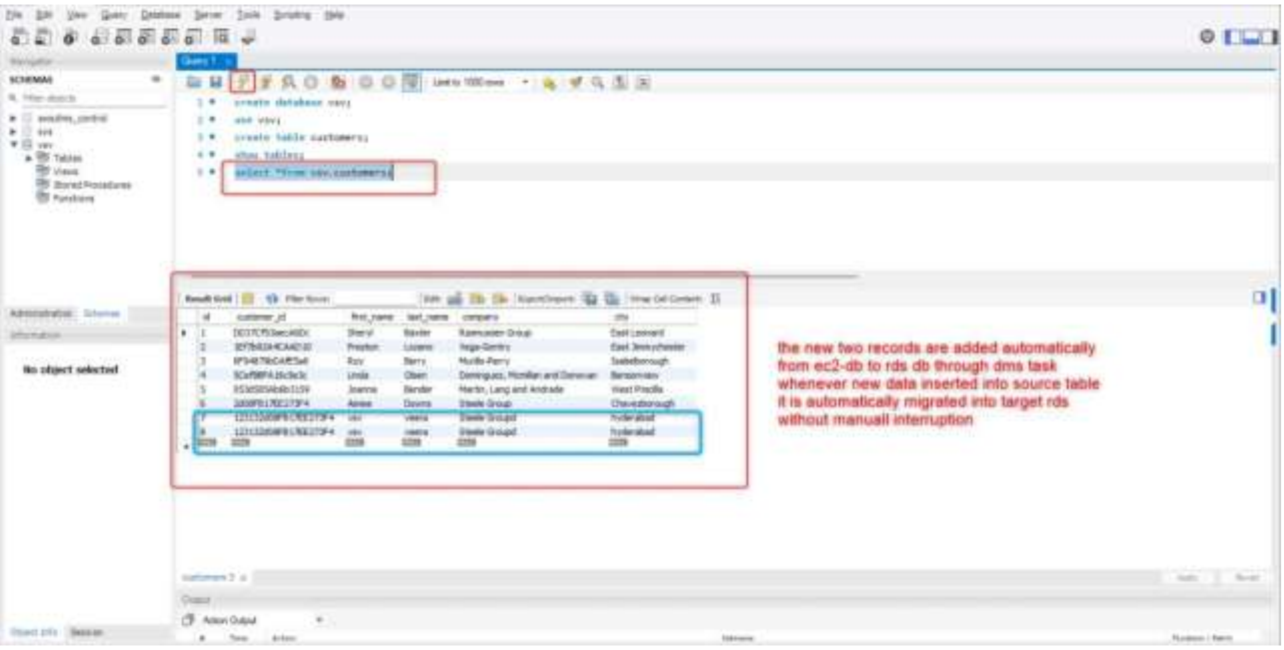
i-05ca26b6455de25df (sql-new)

PublicIPs: 3.109.206.219 PrivateIPs: 172.31.15.79

- Connect to rds database again check the new 2 records are added or not in rds database
- Enter the below command to fetch the records from table

select *from vsv.customers;

- New 2 records are added automatically into rds database through dms task



```
create database vsv;
use vsv;
create table customers;
show tables;
select *from vsv.customers;
```

id	customer_id	first_name	last_name	company	city
1	DD37Cf93aecA6Dc	Sheryl	Baxter	Rasmussen Group	East Leonard
2	1Ef7b82A4CAAD10	Preston	Lozano	Vega-Gentry	East Jimmychester
3	6F94879bDAfE5a6	Roy	Berry	Murillo-Perry	Isabelborough
4	5Cef8BFA16c5e3c	Linda	Olsen	Dominguez, Mcmillan and Donovan	Bensonview
5	053d585Ab6b3159	Joanna	Bender	Martin, Lang and Andrade	West Priscilla
6	2d08FB17EE273F4	Aimee	Downs	Steele Group	Chavezborough
7	12112d08FB17EE273F4	vsv	veena	Steele Groupd	hyderabad
8	12112d08FB17EE273F4	vsv	veena	Steele Groupd	hyderabad
NULL	NULL	NULL	NULL	NULL	NULL

the new two records are added automatically from ec2-db to rds db through dms task whenever new data inserted into source table it is automatically migrated into target rds without manuall interruption

RESUME POINTS

1. Worked on migration of a MySQL database from an EC2 instance to Amazon RDS, ensuring minimal downtime and no data loss.

2. Utilized AWS Database Migration Service (DMS) to streamline the migration process, facilitating real-time data replication with minimal disruption.

3. Configured AWS DMS tasks to handle both full data load and ongoing replication, ensuring seamless transition to the new RDS environment.

4. Monitored and optimized the performance of the RDS instance post-migration, tuning parameters such as storage, instance type, and IOPS.

5. Automated the backup and recovery processes for RDS to ensure high availability and disaster recovery readiness.

6. Ensured proper security configurations in the RDS instance, including setting up IAM roles, encryption, and VPC settings to comply with security best practices.

7. Leveraged CloudWatch for monitoring and troubleshooting during and after the migration, ensuring performance and availability goals were met.

ADVANTAGES MIGRATING DATABASE TO RDS

Improved Scalability:

RDS allows for easier scaling of the database instance with minimal downtime. You can scale up or scale down based on performance needs without manually managing hardware resources.

Managed Service:

Amazon RDS is a fully managed service, meaning AWS handles the administrative tasks such as backups, patching, and updates, freeing up time and resources for other important tasks.

High Availability:

By migrating to RDS, you can take advantage of built-in features like Multi-AZ deployments, which provide high availability and automatic failover in case of an instance failure.

Automated Backups and Snapshots:

RDS offers automated daily backups and the ability to create manual snapshots, making it easier to restore your database to a specific point in time if needed.

Performance Optimization:

RDS provides options to optimize database performance, such as enabling read replicas for improved read scalability and adjusting instance types and IOPS to meet your workload requirements.

Security and Compliance:

AWS RDS includes built-in security features like data encryption at rest and in transit, VPC integration, and IAM roles for secure access management, helping meet compliance requirements.

Reduced Operational Overhead:

With RDS, AWS handles most of the database maintenance tasks (e.g., patching, backups, and monitoring), reducing operational overhead and allowing your team to focus on higher-value tasks.

Cost Efficiency:

RDS provides flexible pricing models (on-demand, reserved instances) that can be more cost-effective compared to running MySQL on EC2, especially when factoring in the management overhead of maintaining the EC2 instance.

Simplified Disaster Recovery:

RDS's built-in support for automated backups and cross-region replication ensures that your database can be quickly recovered in case of a disaster, reducing downtime and potential data loss.