

AWS DMS - Migrating MySQL RDS Database to PostgreSQL Database

1. Create a Source Database using RDS.
Engine: MySQL

Choose username, password and initial database names as given in the slides below.
You can accommodate any changes as per your preferences.

Create database

Choose a database creation method [Info](#)

☒ **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



☐ PostgreSQL



☐ Oracle

ORACLE®

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.

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.

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

In the Connectivity part, attach a security group to your Database which Allows Inbound Traffic from MySQL/Aurora Traffic from 0.0.0.0/0 .

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name [Info](#)

Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

Inbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
MySQL/Aurora	TCP	3306	Anywhere... <input type="text" value="0.0.0.0/0"/>	<input type="text"/>

▼ Additional configuration
Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options

Initial database name [Info](#)

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#)

Option group [Info](#)

Once created. Now connect to your RDS Instance. You can create an EC2 Instance for this purpose or Use MySQL Workbench. You can download MySQL workbench from here:

<https://dev.mysql.com/downloads/installer/>

We have created a Ubuntu EC2 Instance.

Install MySQL on ubuntu using these commands:

```
sudo apt update  
sudo apt install mysql-server mysql-client -y
```

Once done, connect to your RDS instance as shown below:

```
[ec2-user@ip-172-31-34-188 ~]$ mysql -h mysql-source.cvwwkjmvcvgi.us-east-1.rds.amazonaws.com -u admin -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MySQL connection id is 16  
Server version: 8.0.33 Source distribution  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MySQL [(none)]> 
```

Make sure to run the command as sudo user in case you face any permission denied issue.

After connecting to database, use the command “ **show databases;** ”

It lists your databases. Now, use the sourcedb.

```
MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sourcedb |
| sys |
+-----+
5 rows in set (0.002 sec)
```

```
MySQL [(none)]> use sourcedb;
Database changed
```

There are no tables at the moment, We create in the next step.

```
MySQL [sourcedb]> show tables;
Empty set (0.002 sec)
```

You can get sample sql create table commands from the internet and insert some sample data to it as shown here:

```
MySQL [sourcedb]> CREATE TABLE employees (
->     employee_id INT PRIMARY KEY,
->     first_name VARCHAR(50),
->     last_name VARCHAR(50),
->     department VARCHAR(50),
->     hire_date DATE
-> );
Query OK, 0 rows affected (0.034 sec)
```

```
Database changed
MySQL [sourcedb]> INSERT INTO employees (employee_id, first_name, last_name, department, hire_date)
-> VALUES
->     (1, 'John', 'Doe', 'Engineering', '2020-01-15'),
->     (2, 'Jane', 'Smith', 'Marketing', '2019-08-10');
Query OK, 2 rows affected (0.005 sec)
Records: 2  Duplicates: 0  Warnings: 0
```


Now, we have created a table named “**employee**” with two rows of data inside.


```
MySQL [sourcedb]> select * from employees;
+-----+-----+-----+-----+-----+
| employee_id | first_name | last_name | department | hire_date |
+-----+-----+-----+-----+-----+
|          1 | John      | Doe       | Engineering | 2020-01-15 |
|          2 | Jane      | Smith     | Marketing   | 2019-08-10 |
+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```


Now we will migrate this table to our “**targetdb**”. Let's create our target database, with PostgreSQL as Engine.


Engine options


Engine type [Info](#)


☐ Aurora (MySQL Compatible)


☐ Aurora (PostgreSQL Compatible)


☐ MySQL


☐ MariaDB


☒ PostgreSQL


☐ Oracle


Make sure to choose **Engine version as 14.9 or any other version below 15 as PostgreSQL versions 15.0 and above do not support AWS DMS Service.**

▼ Hide filters

☒ Show versions that support the Multi-AZ DB cluster [Info](#)
 Create a Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

Engine Version

PostgreSQL 14.9-R1 ▼

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](#)

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.

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.

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

☐ **Manage master credentials in AWS Secrets Manager**

Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

 If you manage the master user credentials in Secrets Manager, some RDS features aren't supported.

[Learn more](#) 

☐ **Auto generate a password**

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

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm master password [Info](#)

Instance configuration

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

DB instance class [Info](#)

- ☐ Standard classes (includes m classes)
- ☐ Memory optimized classes (includes r and x classes)
- ☒ Burstable classes (includes t classes)

db.t3.micro

2 vCPUs 1 GiB RAM Network: 2,085 Mbps

☐ Include previous generation classes

Storage

Storage type [Info](#)

General Purpose SSD (gp2)

Baseline performance determined by volume size

Allocated storage [Info](#)

20

GiB

The minimum value is 20 GiB and the maximum value is 6,144 GiB

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) [Info](#)

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

Default VPC (vpc-018e1dad574748a56)

7 Subnets, 6 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

6 Subnets, 6 Availability Zones

Here, attach a security group to your PostgreSQL RDS instance, which **allows Inbound Traffic for PostgreSQL on port 5432 from anywhere 0.0.0.0/0**. Name it for example: **postgres-sg**.

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 resources can connect to the database.

Existing VPC security groups

Choose one or more options

postgres-sg 

Create a target database here with the name “**targetdb**”.

Create the RDS Instance.

▼ Additional configuration
Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options

Initial database name [Info](#)

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#)

After creation of RDS instance,

Now use the command below to install PostgreSQL on the same Ubuntu instance we created.

`sudo apt install postgresql -y`

Use the command shown below for connecting with your postgres database:

```
[ec2-user@ip-172-31-34-188 ~]$ psql -h postgresql-target.cvwwkjmmcvgi.us-east-1.rds.amazonaws.com -U postgres -l
Password for user postgres:
      List of databases
  Name      | Owner   | Encoding | Collate | Ctype   | ICU Locale | Locale Provider | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----
 postgres   | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |             | libc            |
 rdsadmin   | rdsadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 |             | libc            | rdsadmin=CTC/rdsadmin+
            |          |          |             |             |             |                 | rdstopmgr=Tc/rdsadmin
 targetdb   | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |             | libc            |
 template0  | rdsadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 |             | libc            | =c/rdsadmin +
            |          |          |             |             |             |                 | rdsadmin=CTC/rdsadmin
 template1  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |             | libc            | =c/postgres +
            |          |          |             |             |             |                 | postgres=CTC/postgres
(5 rows)
```

```
[ec2-user@ip-172-31-34-188 ~]$ psql -h postgresql-target.cvwwkjmmcvgi.us-east-1.rds.amazonaws.com -U postgres -d targetdb
Password for user postgres:
psql (15.0, server 14.9)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, compression: off)
Type "help" for help.
targetdb=>
```

There are no tables at the moment in our Postgres database.

```
targetdb=> \dt
Did not find any relations.
targetdb=> []
```

Now, we will create a migration job using DMS to migrate a table from "**sourcedb**" to "**targetdb**".

Start with creating a replication instance. Choose the configurations shown below.

Replication instances (0)

Find replication instance

Create replication instance

	Name	Status	VPC	Class	Engine version	Availability zone	Network type	Public	Public IP address	Private IP address
--	------	--------	-----	-------	----------------	-------------------	--------------	--------	-------------------	--------------------

Create replication instance [Info](#)

Settings

Name

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

ReplicationInstance

Replication instance name must not start with a numeric value

Instance configuration [Info](#)

Instance class [Info](#)

dms.t3.medium

2 vCPUs 4 GiB Memory

☐ Include previous-generation instance classes

Engine version

Choose an AWS DMS version to run on your replication instance. For more details, See the [AWS DMS release notes](#). To see how long each DMS version will be supported, check the [AWS DMS support lifecycle policy](#).

3.5.1

☐ 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

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-018e1dad574748a56) ▼



[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-subnetgroup ▼

☐ **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.

Once created, wait for 20-25 minutes for it to be available.

DMS > [Replication instances](#) > replicationinstance

replicationinstance Delete Reboot Modify

Summary			
Status 🟢 Available	Class dms.t3.medium	Engine version 3.5.1	Associated migration tasks -

Now, let's create two endpoints for our “**sourcedb**” and “**targetdb**”.

Endpoints (0)

Find endpoint

< 1 >

Name

Type

Status

Engine

Server name

Port

Migration Hub Mapping

ARN

Certificate ARN

Empty endpoint table

You don't have any endpoints.

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.

Endpoint configuration

Endpoint identifier [Info](#)

A label for the endpoint to help you identify it.

source-endpoint

Provide access information manually.

Source engine

The type of database engine this endpoint is connected to. [Learn more](#)

MySQL

Access to endpoint database [Info](#)

☐ AWS Secrets Manager

☒ Provide access information manually

Server name

The name of the data server for the data provider.

mysql-source.cvwwwkjmvcgi.us-east-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

Test the Endpoint connection and only proceed when it's successful.

▼ Test endpoint connection (optional)

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

VPC

vpc-07c7da1dfdd90422f

Replication instance

A replication instance performs the database migration

replicationinstance

**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

Run test

Endpoint identifier	Replication instance	Status	Message
source-endpoint	replicationinstance	successful	

Now, create an endpoint for “**targetdb**”.

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.

Endpoint configuration

Endpoint identifier [Info](#)

A label for the endpoint to help you identify it.

Target engine

The type of database engine this endpoint is connected to. [Learn more](#) 

Access to endpoint database [Info](#)

☐ AWS Secrets Manager

☒ Provide access information manually

Server name

The name of the data server for the data provider.

Port

The port the database runs on for this endpoint.

User name [Info](#)

Password [Info](#)

Secure Socket Layer (SSL) mode [Info](#)

The type of Secure Socket Layer enforcement

Database name

▼ Test endpoint connection (optional)

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

VPC

Replication instance

A replication instance performs the database migration

Endpoint identifier	Replication instance	Status	Message
target-endpoint	replicationinstance	successful	

Database migration tasks (0)



Actions ▼

Quick view and compare

Create task

< 1 >

Create

Now, both of our Endpoints have successfully been created. We are ready to run a migration job now.

Go to Migration Tasks and create one.

Choose appropriate Replication instance, endpoints, VPCs, Task Settings etc. as shown below:

Task configuration

Task identifier

database-migration-task

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

replicationinstance - vpc-018e1dad574748a56 ▼

Source database endpoint

source-endpoint ▼

Target database endpoint

target-endpoint ▼

Migration type | [Info](#)

Migrate existing data ▼

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.

Target table preparation mode [Info](#)

☐ Do nothing

☒ Drop tables on target

☐ Truncate

LOB column settings [Info](#)

☐ Don't include LOB columns

☐ Full LOB mode

☒ Limited LOB mode

Maximum LOB size (KB)

32

▼ Selection rules

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

Add new selection rule

Add new selection rule

Schema

Enter a schema

▼

Source name

Use the % character as a wildcard

%

Source table name

Use the % character as a wildcard

%

Action

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

Include

▼

Source filters

Info

Add column filter

Migration task startup configuration

Start migration task

- ☒ Automatically on create
- Available only if the premigration assessment is not enabled.

Create the task.

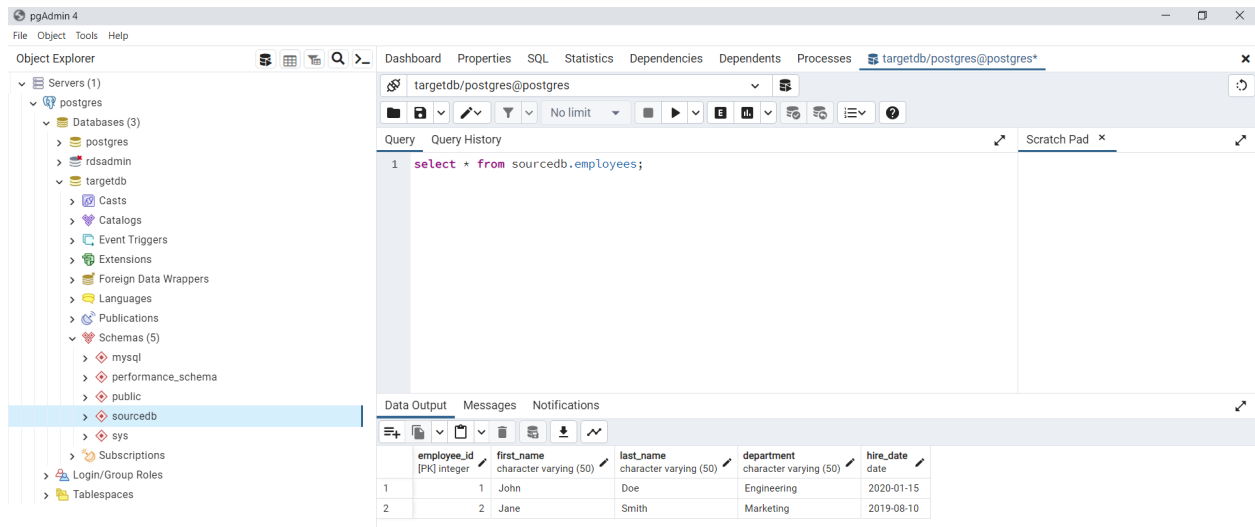
Wait for it to be completed.

Database migration tasks (1)										
<div><div>Find database migration tasks</div><div>< 1 > ⚙</div></div>										
<input type="checkbox"/>	Identifier	Status	Progress	Type	Source	Target	Replication instance	Started	Stopped	
<input type="checkbox"/>	database-migration-task	Load complete	100%	Full load	source-endpoint	target-endpoint	replicationinstance	September 3, 2023 at 21:53:08 (UTC+05:30)	September 3, 2023	

After completion, you can view the migrated table under **table statistics**.

Table statistics (157)											
Total rows include loaded source table rows from Inserts, Deletes, Updates, DDLs, and Full load rows.											
<div><div>sour</div><div>< 1 > ⚙</div></div>											
<input checked="" type="checkbox"/>	Schema name	Table	Load state	Elapsed load time	Inserts	Deletes	Updates	DDLs	Applied inserts	Applied deletes	Applied updates
<input checked="" type="checkbox"/>	sourcedb	employees	Table completed	1 s	0	0	0	0	0	0	0

Additionally, you can use PgAdmin to connect to your PostgreSQL RDS instance and view the table.



This completes AWS DMS hands-on.

Clean-up. Start with Deleting Migration Task, Endpoints, Replication Instances. Then delete your RDS instances and the EC2 Instance.