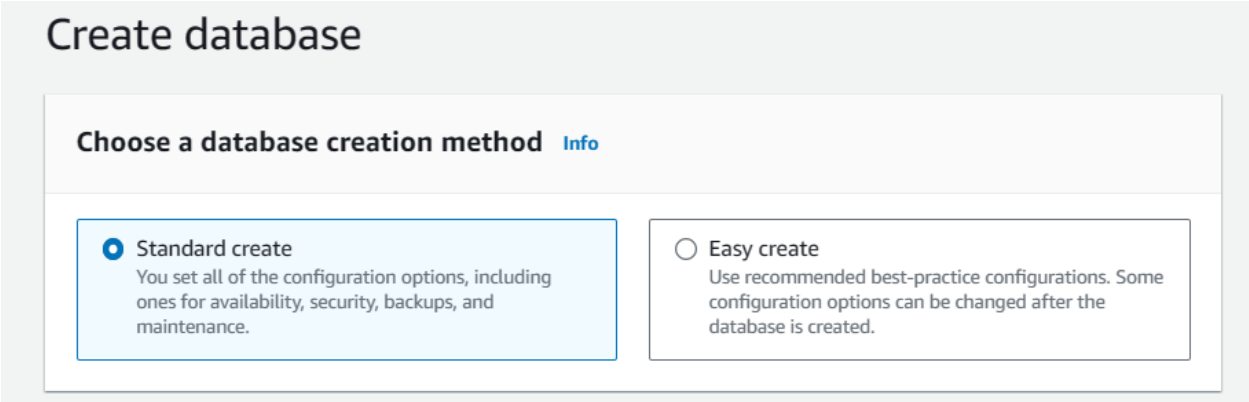
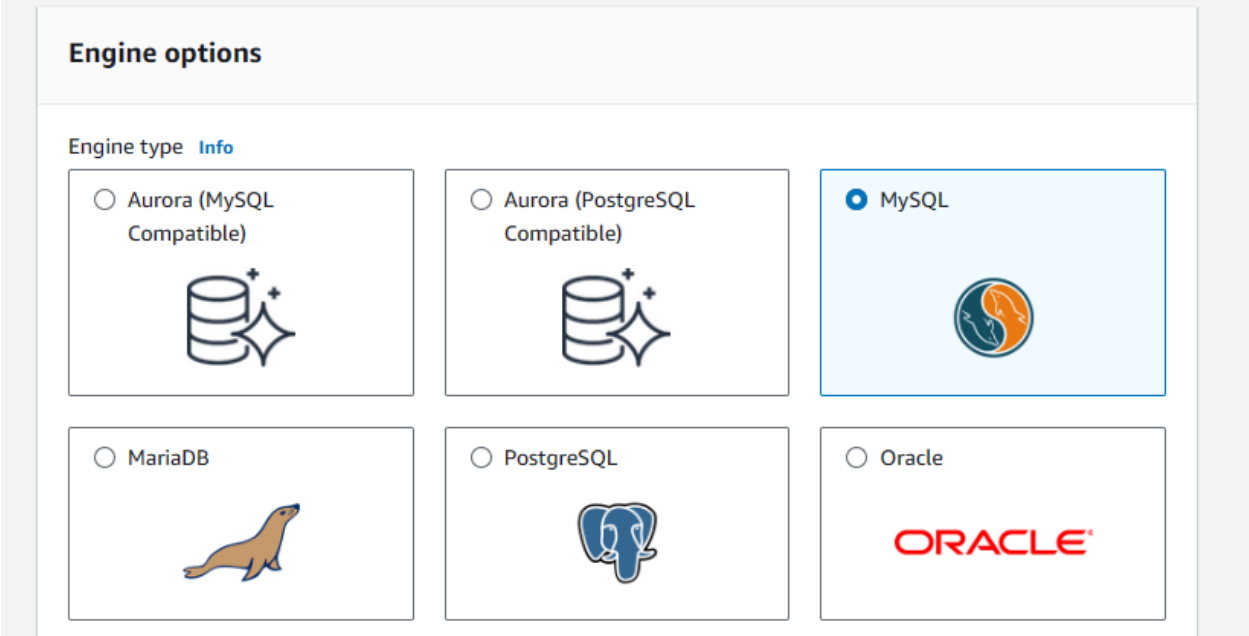
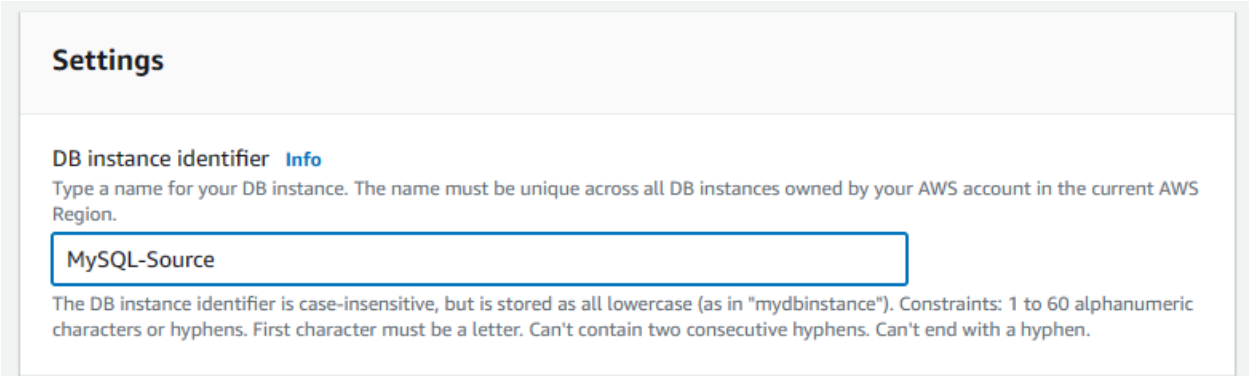
AWS DMS - Migrating MySQL RDS Database to PostgreSQL Database

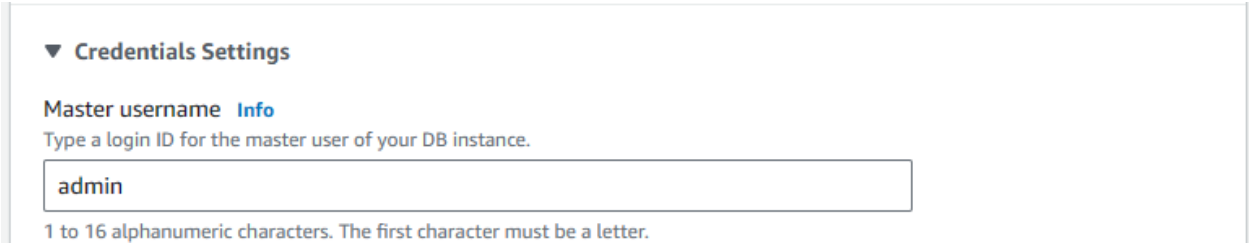
1. Create a Source Database using RDS.
   Engine: MySQL

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

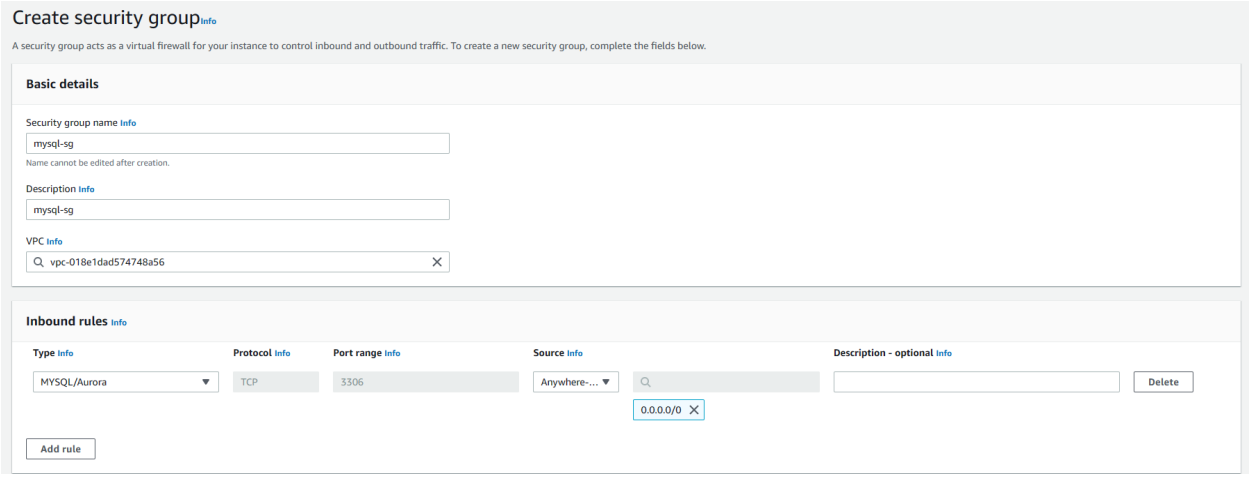


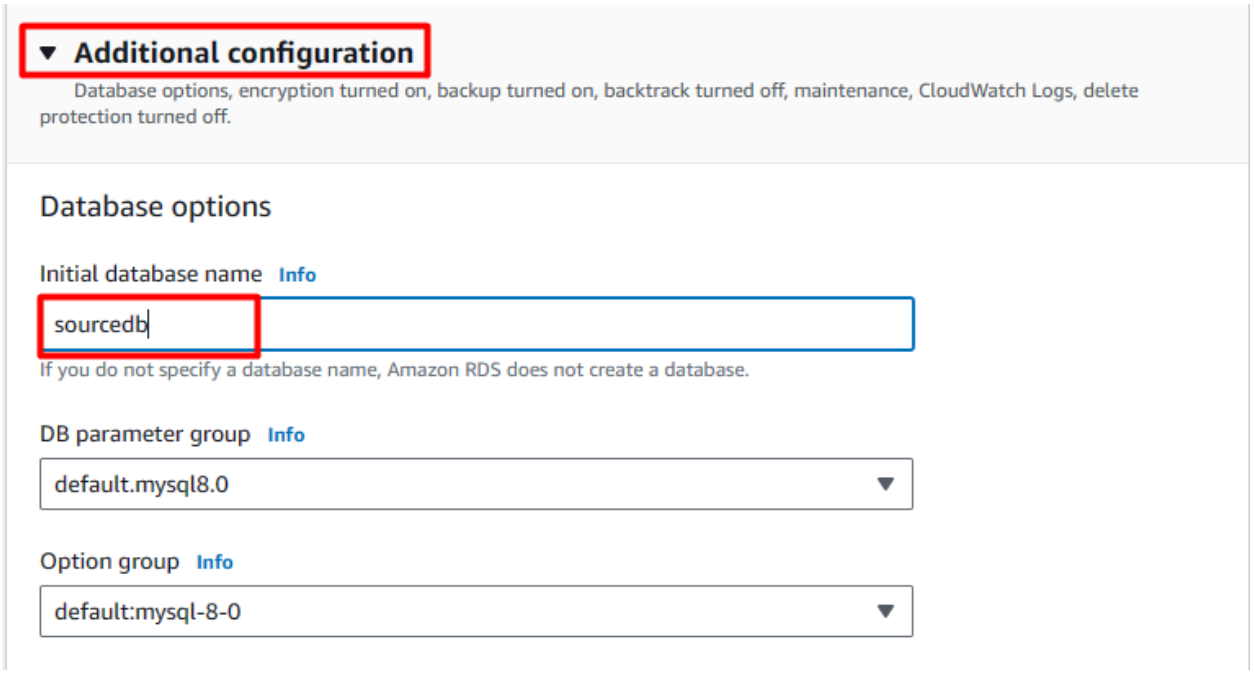






In the Connectivity part, attach a security group to your Database which Allows Inbound Traffic from MySQL/Aurora Traffic from 0.0.0.0/0 .





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.cvwwkjmmcvgi.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:

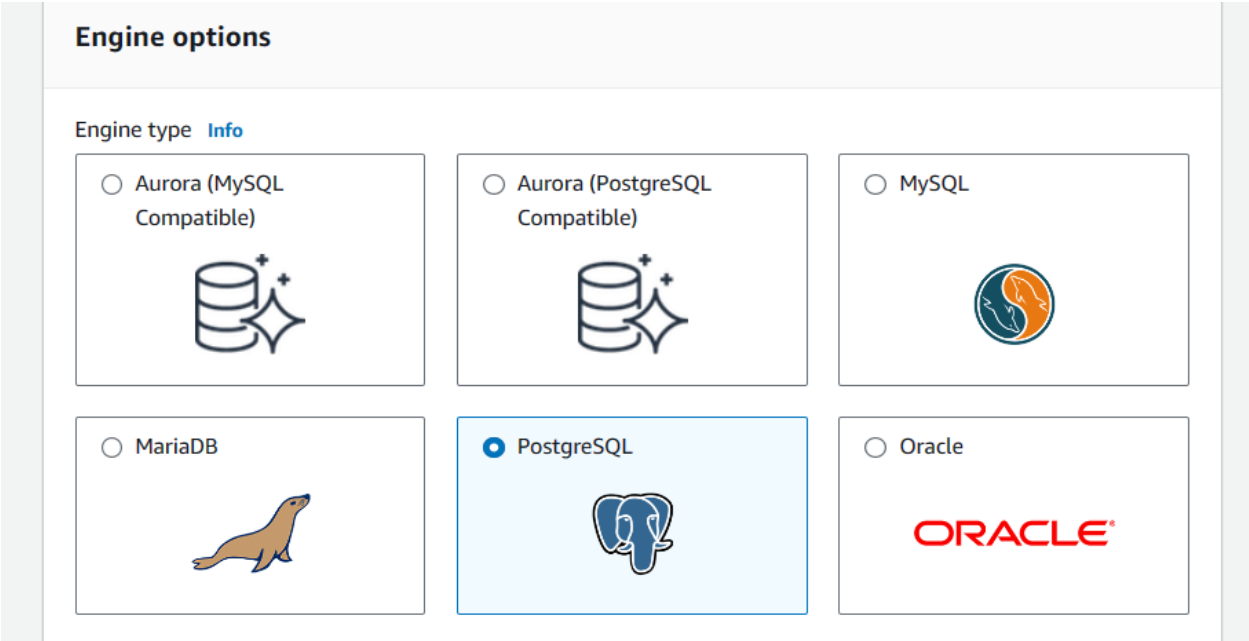
```
MySQL [sourcedb]> CREATE TABLE employees (
    ->     employee_id INT PRIMARY KEY,
    ->     first_name VARCHAR(50),
    ->     last_name VARCHAR(50),
    ->     department VARCHAR(50),
    ->     hire_date DATE
    -> );
Query OK, 0 rows affected (0.034 sec)
```

```
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
```

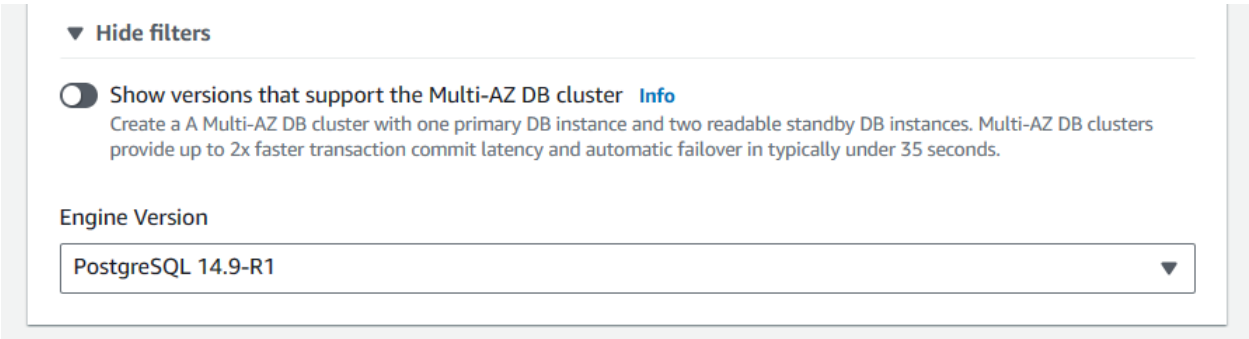
Now, we have created a table named "**employee**" with two rows of data inside.

```
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.



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****.*



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.

PosgreSQL-Target

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.

postgres

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

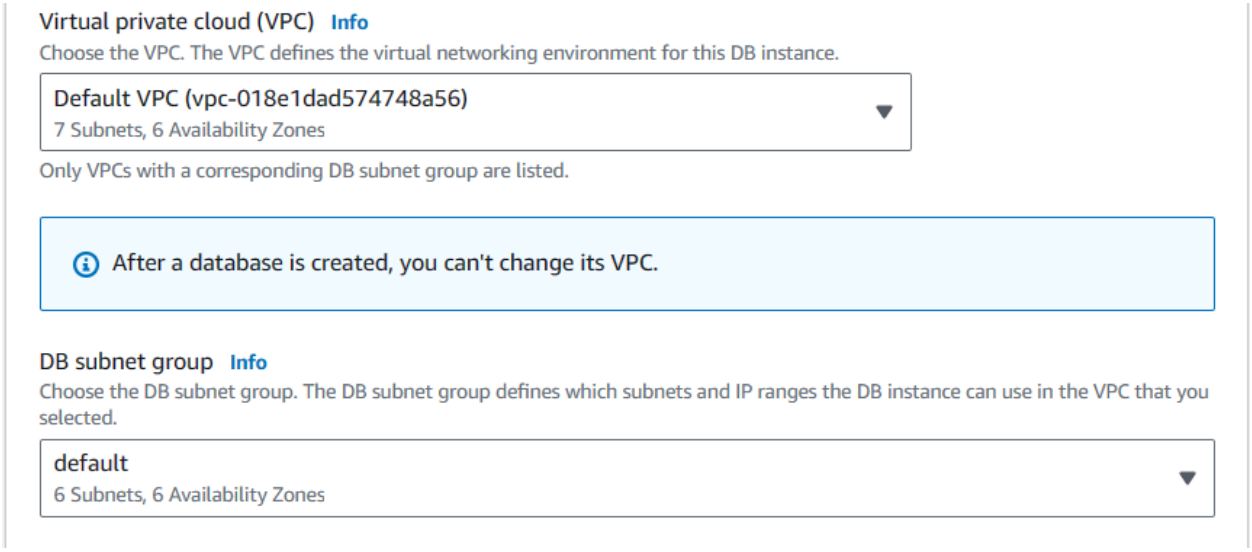
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

- 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.

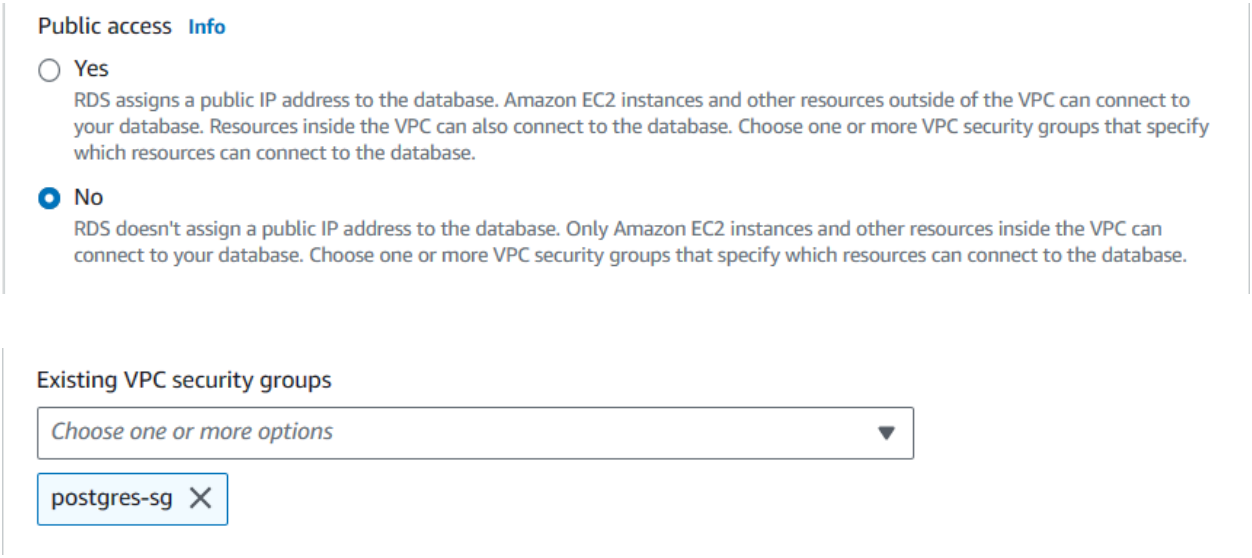
Network type Info

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.

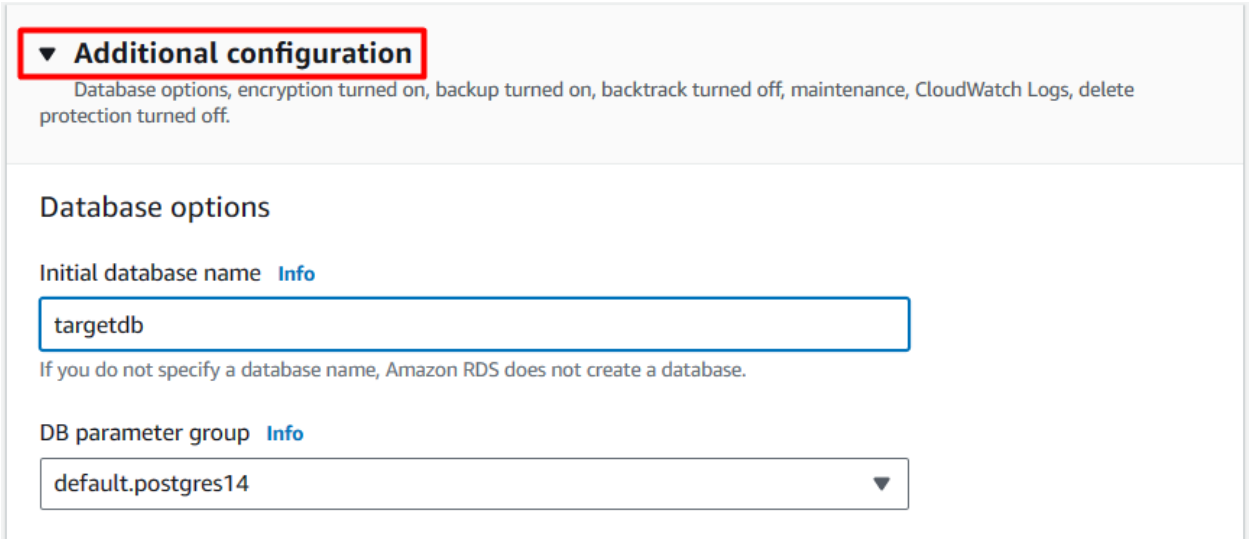


Here, attach a security group to your PostgreSQL RDS instance, which **allows Inbound Traffic for PostgreSQL on port 5432 from anywhere 0.0.0.0/0**. Name it for example: **postgres-sg**.



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

Create the RDS Instance.



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 posgresql-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)
```

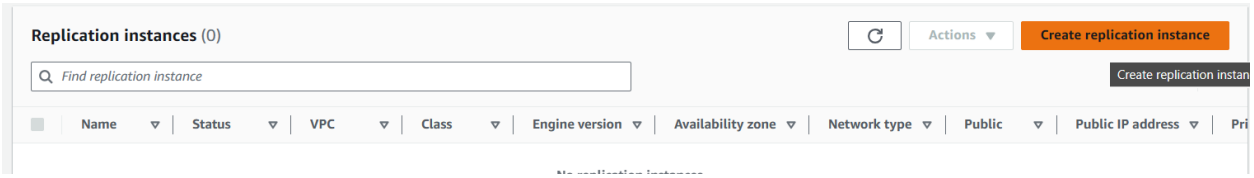
```
[ec2-user@ip-172-31-34-188 ~]$ psql -h posgresql-target.cvwwkjmmcvgi.us-east-1.rds.amazonaws.com -U postgres -d targetdb
Password for user postgres:
psql (15.0, server 14.9)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, compression: off)
Type "help" for help.

targetdb=>
```

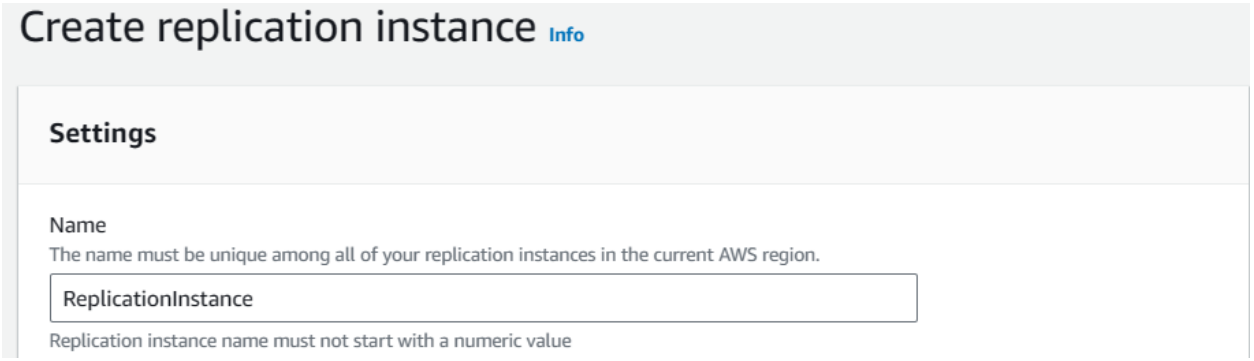
There are no tables at the moment in our Postgres database.

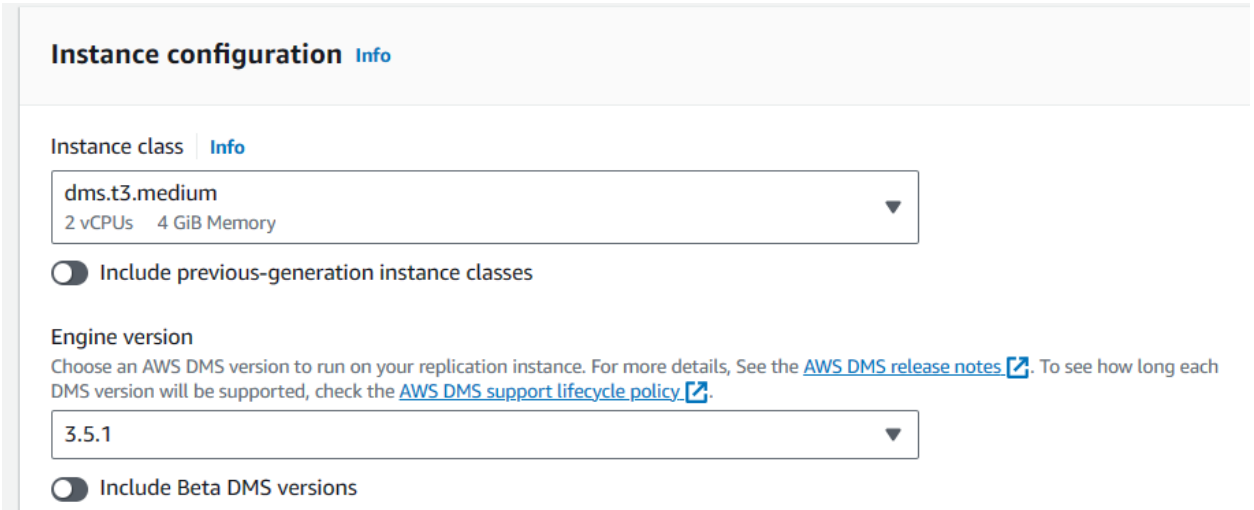
```
targetdb=> \dt
Did not find any relations.
targetdb=>
```

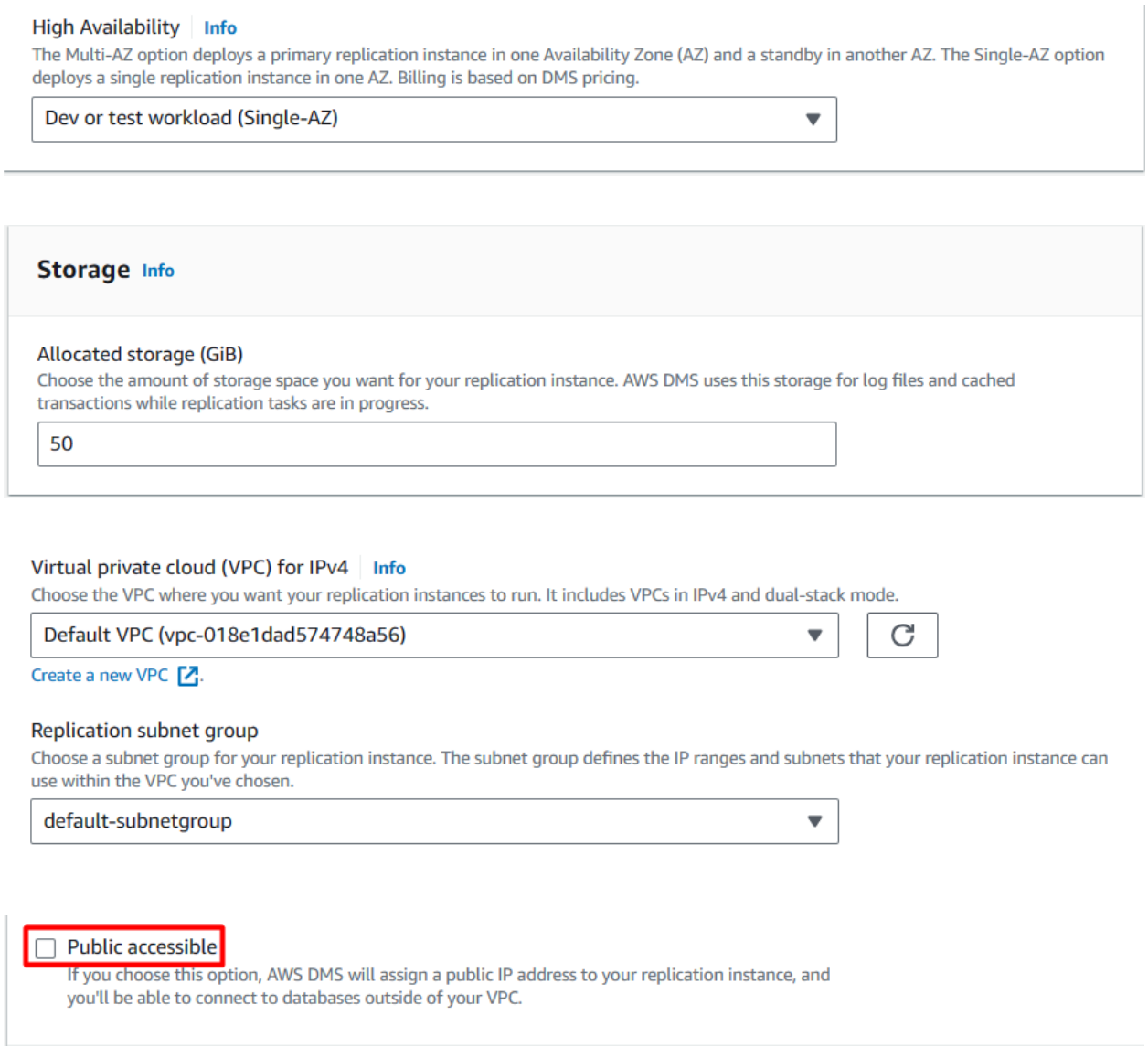
Now, we will create a migration job using DMS to migrate a table from ”**sourcedb**” to “**targetdb**”.

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

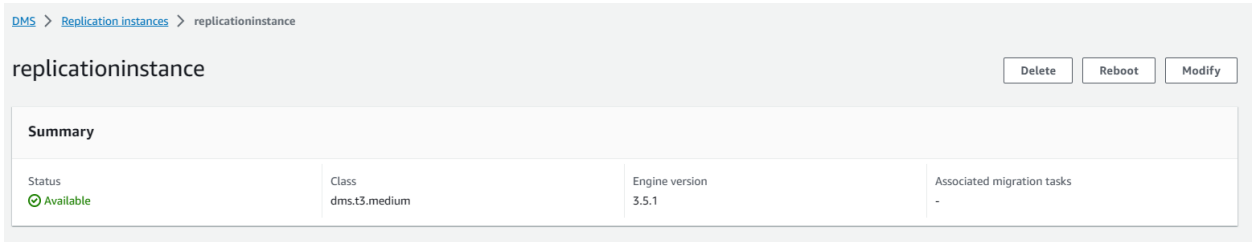




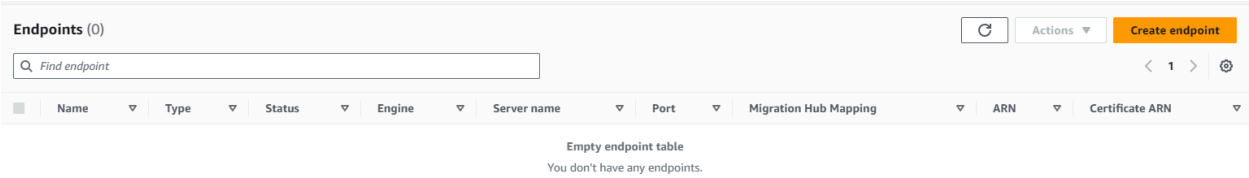


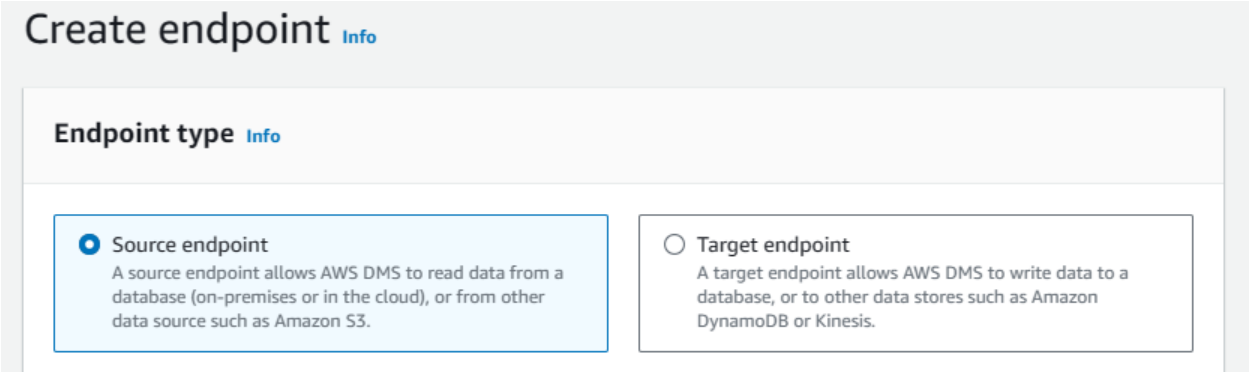


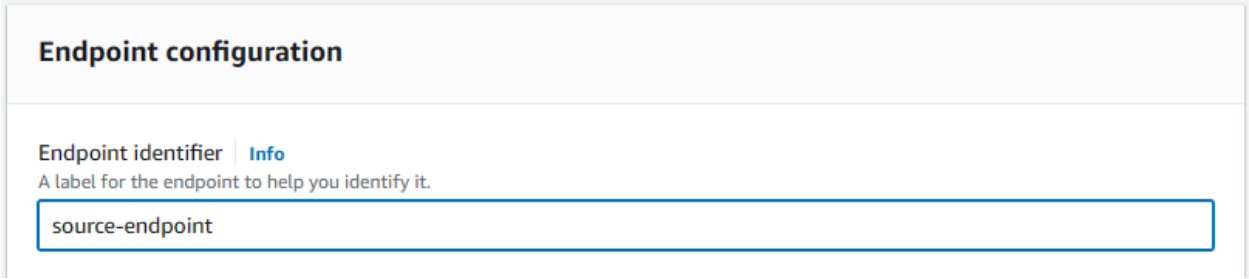
Once created, wait for 20-25 minutes for it to be available.



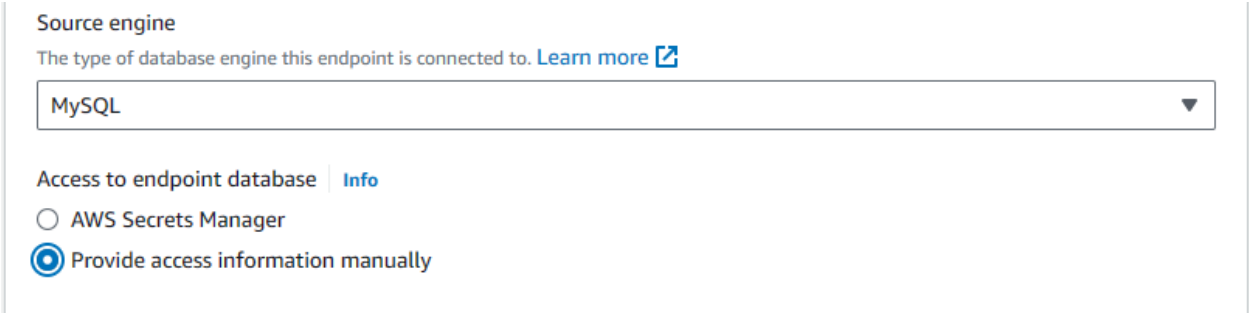
Now, let's create two endpoints for our “**sourcedb**” and “**targetdb**”.







Provide access information manually.



Server name
The name of the data server for the data provider.
mysql-source.cvwwkjmmcvgi.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

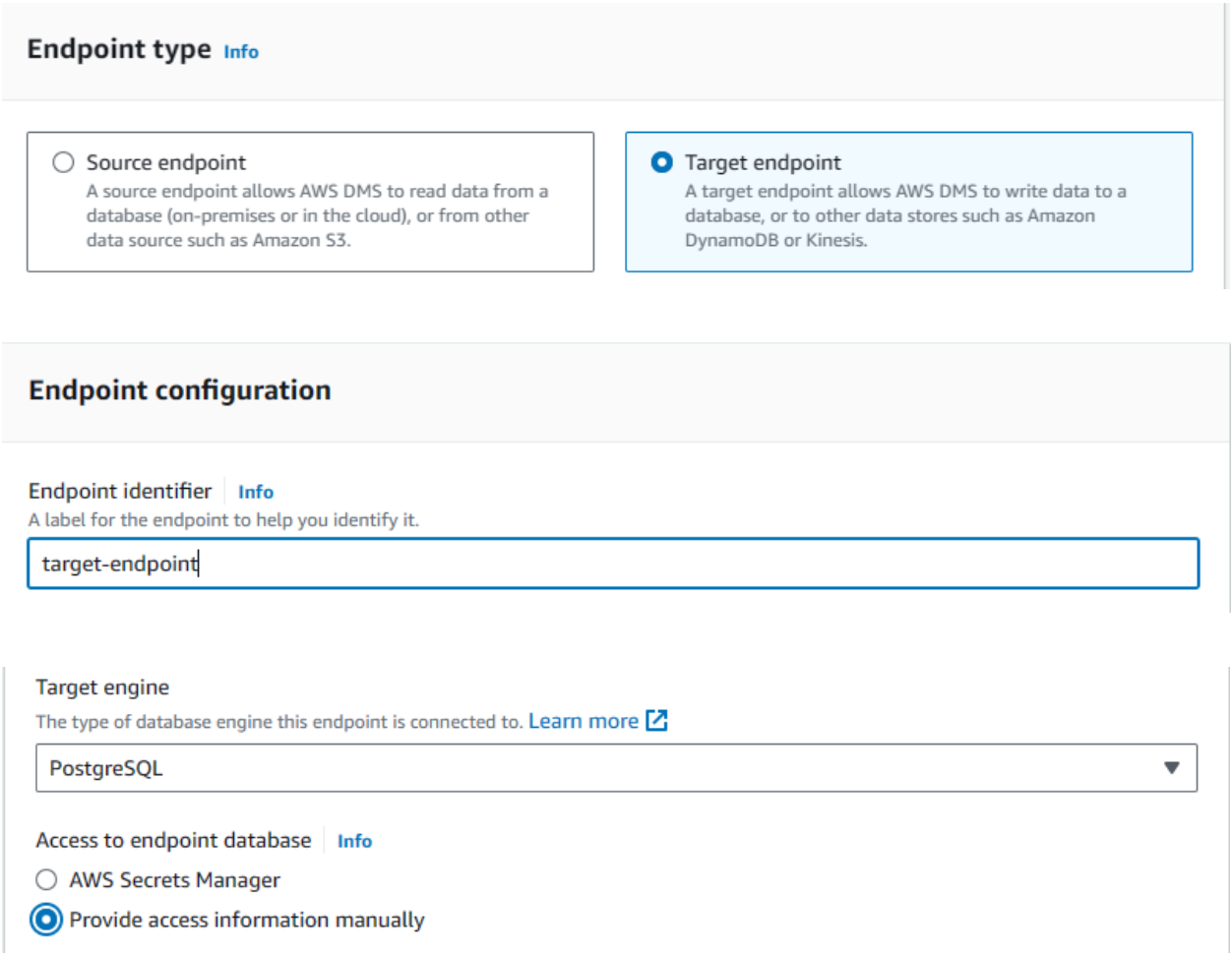
Your endpoint will always be created even if the connection fails
After clicking 'Run test', DMS creates the endpoint with the details you provided and attempts to connect to it. If the connection fails, you can edit the endpoint definition and test the connection again. You can also delete the endpoint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

Run test

Run test

Endpoint identifier	Replication instance	Status	Message
source-endpoint	replicationinstance	successful	

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



Server name
The name of the data server for the data provider.
posgresql-target.cvwwkjmmcvgi.us-east-1.rds.amazonaws.com

Port
The port the database runs on for this endpoint.
5432

User name Info
postgres

Password Info
••••••••

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

Database name
targetdb

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

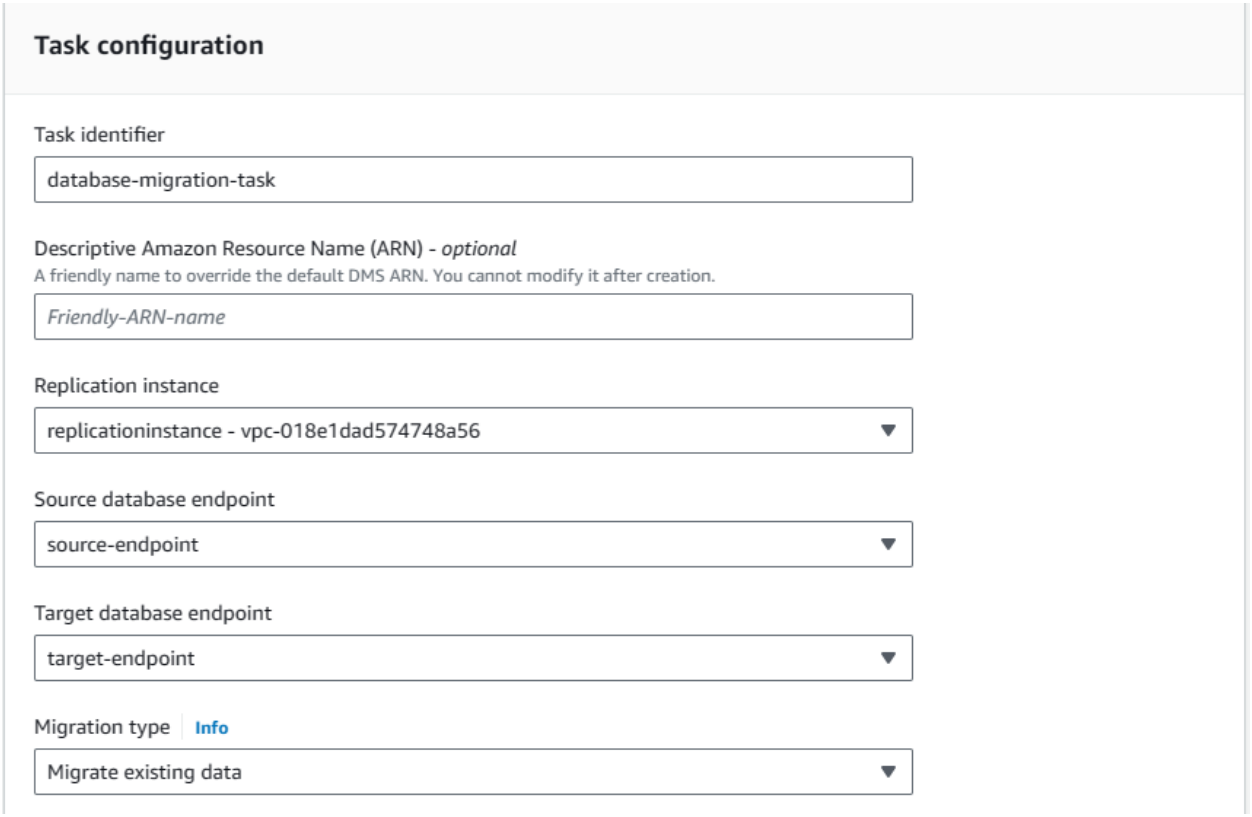
Endpoint identifier	Replication instance	Status	Message
target-endpoint	replicationinstance	successful	

Database migration tasks (0)
Find database migration tasks
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 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)

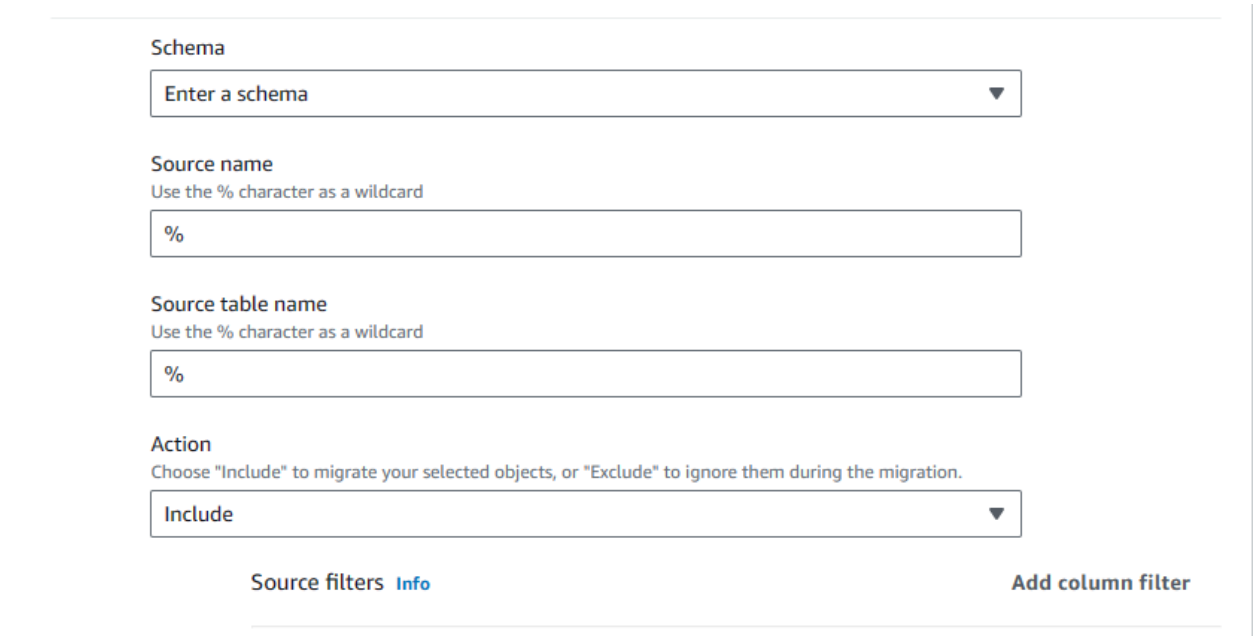
32

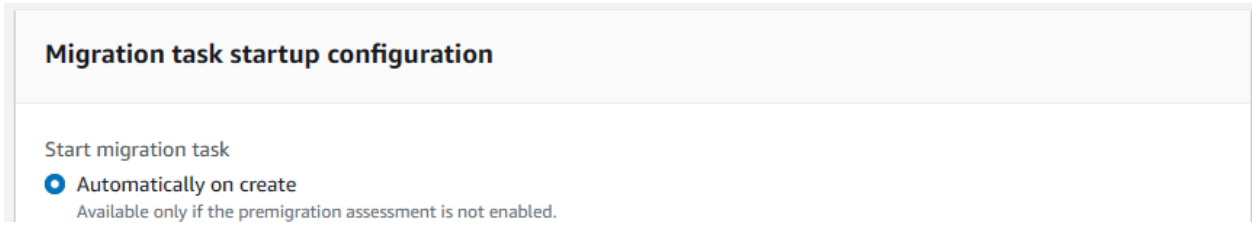
▼ **Selection rules**

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

Add new selection rule

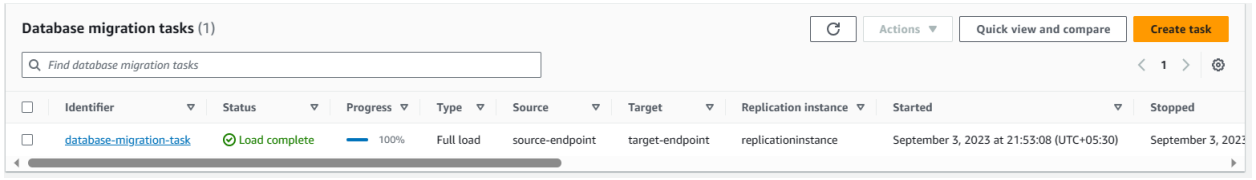
Add new selection rule



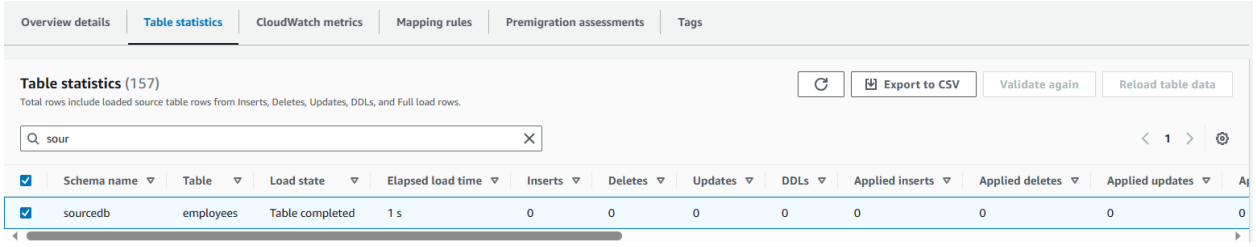


Create the task.

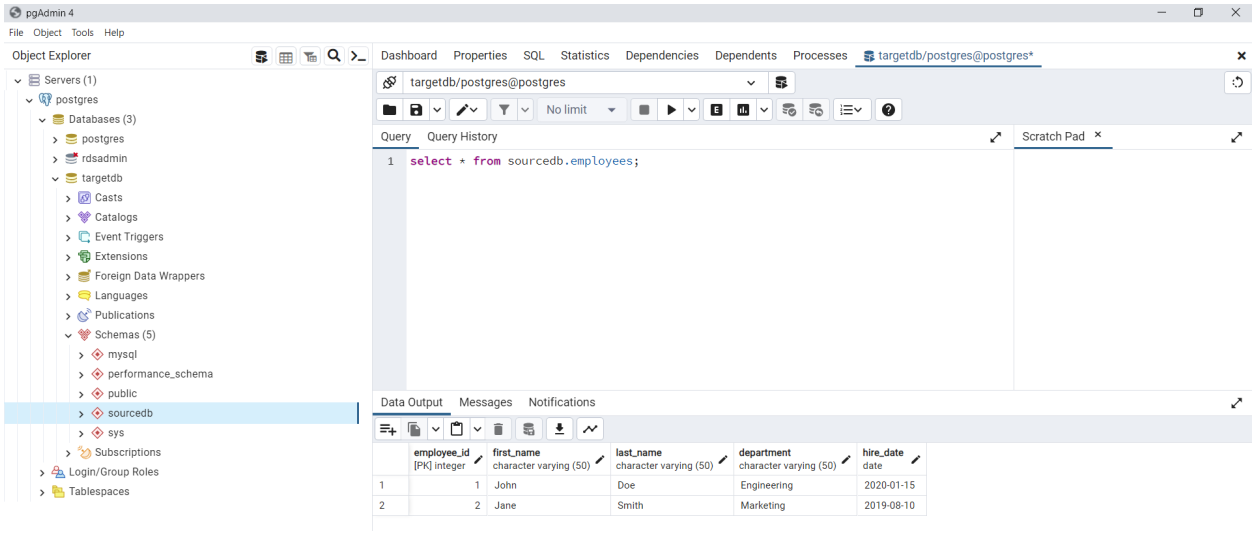
Wait for it to be completed.



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



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.