Migrating PostgreSQL DB from on-premises to Amazon RDS with AWS DMS

Contents

Introduction

With AWS Database Migration Service (DMS), you can migrate on-premises PostgreSQL to Amazon RDS for PostgreSQL with minimal downtime. AWS DMS has some advantages, such as an easier setup process. It can also migrate tables that don't have a primary key, but supports only full-load, not change data capture (CDC), and validates the data between the source and target databases for tables which have primary key.

Precaution and Limitation:

At the time of writing this:

- *AWS DMS doesn't support PostgreSQL 15+.*
- *pglogical replication and administration requires superuser privileges.*
- *pglogical needs a primary key or unique key on the tables to capture the DMLs.*

Preparing the Source Environment

Prerequisites

Before getting started, install and configure the listed parameters in the postgresql.conf of source databases server. Allow the replication user in the pg_hba.conf file on the source databases.

1. **Download the pglogical rpm and install on the source databases.**

- apt install postgresql10-pglogical -- **Skip this Step on Windows.**

Configuring the source server

2. On the source server, in postgresql.conf, modify the following parameters:

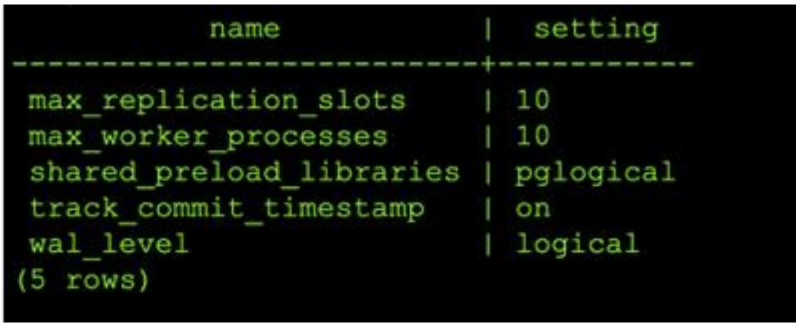
- ***max_worker_processes = 10***
- ***wal_level = 'logical'***
- ***max_wal_senders = 10***
- ***max_replication_slots = 10***
- ***wal_sender_timeout =60000***
- ***track_commit_timestamp = on***
- ***shared_preload_libraries = 'pglogical'*** -- **Skip this Step on Windows.**

3. Restart the PostgreSQL instances for these parameters to take effect.

4. To see the set parameters, run the following sql query on your source database server:

```
select name, setting from pg_settings where name in ('wal_level', 'track_commit_timestamp', 'max_worker_processes',
'max_replication_slots', 'max_wal_senders ', 'shared_preload_libraries');
```

The following screenshot shows the query output.



```
           name           | setting
--------------------------+-----------
 max_replication_slots    | 10
 max_worker_processes     | 10
 shared_preload_libraries | pglogical
 track_commit_timestamp   | on
 wal_level                | logical
(5 rows)
```

Note: *For now, you can skip step No 5. You can revisit this step once you setup DMS Replication Server.*

5. Add the IP address of the AWS DMS replication server to the pg_hba.conf configuration file and enable replication and socket connections. An example follows.

```
host    all    all    <RDS endpoint>/32        md5
host    replication    dms    <RDS endpoint>/32        md5
```

Setting up logical replication

6. On the source databases, create the pglogical extension with the following code:
create extension pglogical; -- **Skip this Step on Windows.**

Run the following command to verify that pglogical installed successfully:
select * FROM pg_catalog.pg_extension; -- **Skip this Step on Windows.**

***Note**: The pglogical schema and other objects are created under the pglogical schema, which is helpful in maintaining the information of the replication.*

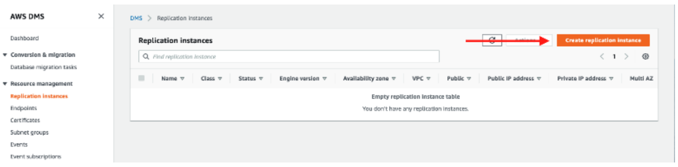
Setting-up AWS DMS Environment.

Prerequisites:

AWS RDS Posgresql Instance
VPC with Site-to-Site VPN or Direct Connect.

Creating Replication Instance in AWS Database Migration Services (AWS DMS)

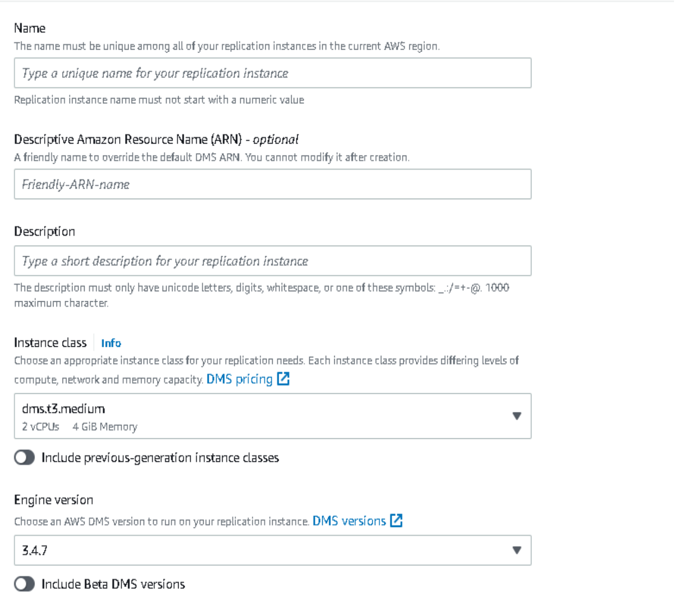
1 To create a replication instance, go to the Replication Instances section of the AWS DMS console. Choose Create replication instance to begin the replication instance creation wizard.



Empty replication instance table
You don't have any replication instances.

2 In the Replication instance configuration section, give your replication instance a name and description. Then choose your instance class. The instance class you use depends on the size of your existing database and the amount of data flowing through it.

Then choose an engine version for AWS DMS. Finally, choose the amount of allocated storage for your replication instance.



Name
The name must be unique among all of your replication instances in the current AWS region.
Type a unique name for your replication instance
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.
Friendly-ARN-name

Description
Type a short description for your replication instance
The description must only have unicode letters, digits, whitespace, or one of these symbols: _.:/=+-@. 1000 maximum character.

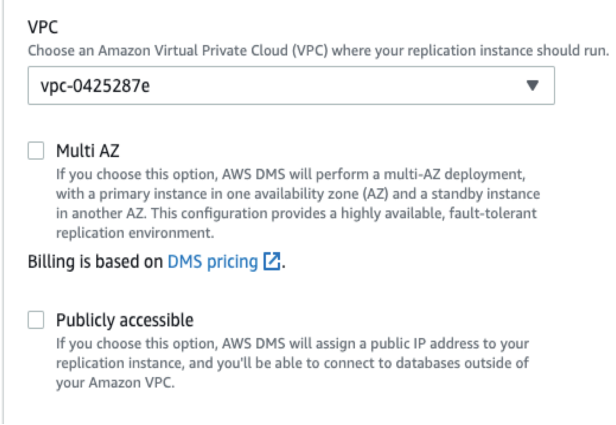
Instance class Info
Choose an appropriate instance class for your replication needs. Each instance class provides differing levels of compute, network and memory capacity. DMS pricing
dms.t3.medium
2 vCPUs 4 GiB Memory ▼
○ Include previous-generation instance classes

Engine version
Choose an AWS DMS version to run on your replication instance. DMS versions
3.4.7 ▼
○ Include Beta DMS versions

3 As you continue in the Replication instance configuration section, you need to choose a VPC for your replication instance. Choose the same VPC in which you provisioned your Amazon RDS database to ease network access for the replication instance.

You may choose to have a Multi-AZ setup for your replication instance for redundancy. If you will use AWS DMS to keep two databases in sync over a long period of time, you may want to use a Multi-AZ setup. If you are performing a one-time migration of your data from an existing database to a fully managed database in Amazon RDS, you likely don't need a Multi-AZ setup.

Finally, choose whether your replication instance should be publicly accessible. If your existing database is in the same VPC as your new database and your replication instance, you don't need your replication instance to be publicly accessible. If your existing database is not in the



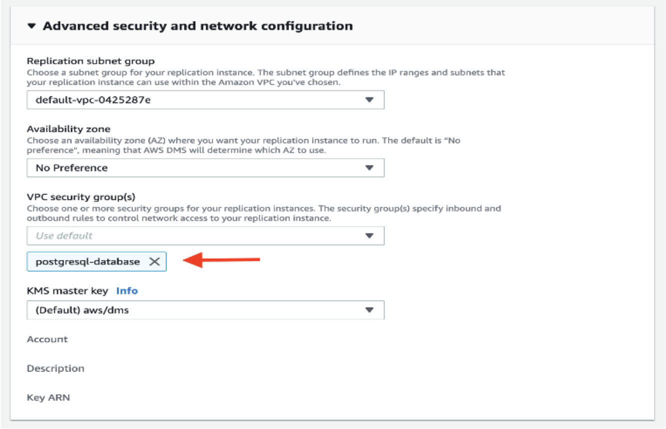
VPC
Choose an Amazon Virtual Private Cloud (VPC) where your replication instance should run.
vpc-0425287e ▼

□ Multi AZ
If you choose this option, AWS DMS will perform a multi-AZ deployment, with a primary instance in one availability zone (AZ) and a standby instance in another AZ. This configuration provides a highly available, fault-tolerant replication environment.
Billing is based on DMS pricing.

□ Publicly 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 Amazon VPC.

same VPC, you should use AWS Direct Connect or a VPN to allow connection from your VPC to your existing data center. You should avoid making your replication instance publicly accessible in most cases to avoid potential security issues.

4 Next, open the Advanced security and network configuration section. For the VPC security group(s) configuration, choose the same security group that you attached to your Amazon RDS database. This allows your replication instance to access your Amazon RDS database.



▼ Advanced security and network configuration

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 Amazon VPC you've chosen.
default-vpc-0425287e

Availability zone
Choose an availability zone (AZ) where you want your replication instance to run. The default is "No preference", meaning that AWS DMS will determine which AZ to use.
No Preference

VPC security group(s)
Choose one or more security groups for your replication instances. The security group(s) specify inbound and outbound rules to control network access to your replication instance.
Use default
postgresql-database ✕

KMS master key Info
(Default) aws/dms

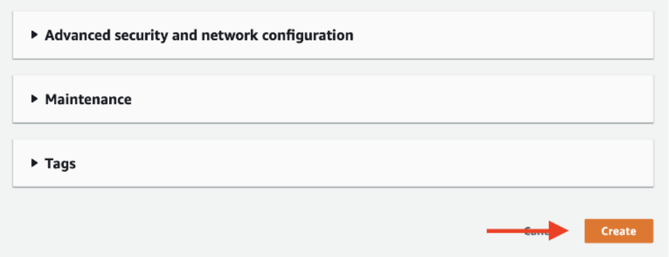
Account

Description

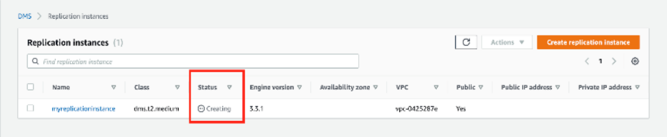
Key ARN

5 You may also edit the maintenance and tags settings.

When you're ready, choose Create to create your replication instance in AWS DMS.



6 After you choose Create, AWS provisions your replication instance. It shows a Status of the replication instance is Creating while AWS provisions and initializes your instance.

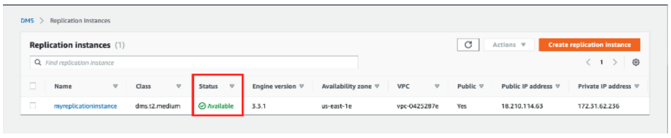


DMS > Replication instances

Replication instances (1)

Name	Class	Status	Engine version	Availability zone	VPC	Public	Public IP address	Private IP address
myreplicationinstance	dms.t2.medium	Creating	3.3.1		vpc-0425287e	Yes		

7 When your replication instance is ready to go, its Status is Available.



DMS > Replication instances

Replication instances (1)

Name	Class	Status	Engine version	Availability zone	VPC	Public	Public IP address	Private IP address
myreplicationinstance	dms.t2.medium	Available	3.3.1	us-east-1e	vpc-0425287e	Yes	18.210.114.63	172.31.62.236

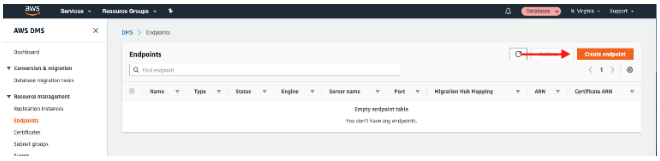
8 Add DMS Server IP address to sourcer database server hba.conf file – go Step 5 of **Preparing the Source Environment**

Creating Endpoints in AWS DMS

A replication task is a job to migrate data from one database to another by using AWS DMS. Before creating a replication task, you must register endpoints for your source and target databases. An endpoint describes the connection address, credentials, and other information required to connect to a database.

1 First, create the endpoint for your Source database.

Navigate to the Endpoints section of the AWS DMS console. Choose Create endpoint to create a new endpoint.



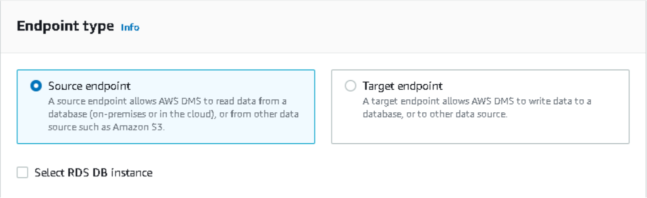
Name	Type	Status	Engine	Server name	Port	Migration Hub Mapping	ARN	Certificate ARN

Empty endpoint table
You don't have any endpoints.

2 In the endpoint creation wizard, choose to create a Source endpoint.

For on-premises Instance
Uncheck the check box to **Select RDS DB instance**

For RDS Instance:
Choose the check box to Select RDS DB instance and choose your newly created Amazon RDS database in the dropdown.



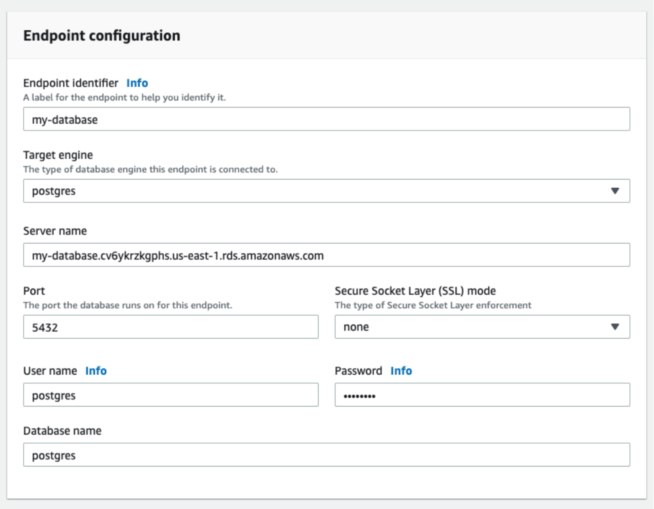
Endpoint type Info

● 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 source.

□ Select RDS DB instance

3 This fills in most of the Endpoint configuration details for you. You need to enter your password near the bottom of the section.



Endpoint configuration

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

Target engine
The type of database engine this endpoint is connected to.
postgres

Server name
my-database.cv6ykrzkgphs.us-east-1.rds.amazonaws.com

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

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

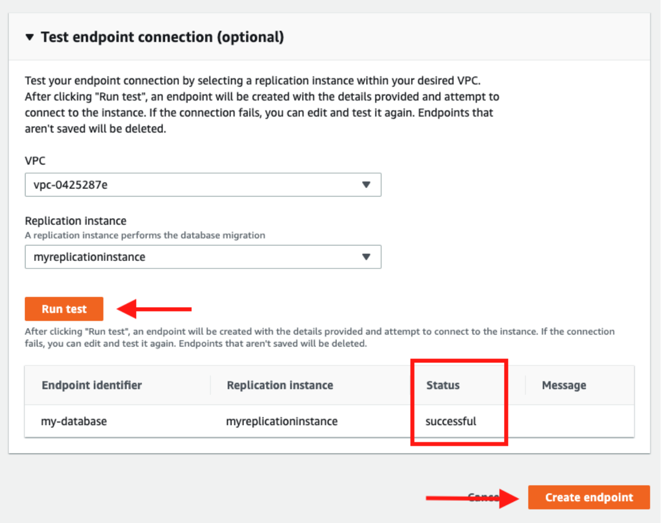
User name Info
postgres

Password Info
••••••••

Database name
postgres

4 Before you save your endpoint, test the connection to ensure that it was configured correctly. To test it, open the Test endpoint connection section.

Choose the replication instance you want to use, and then choose Run test. After a few seconds, you should see a Status of successful. This indicates that you configured your security group and endpoint correctly. To save your endpoint, choose Create endpoint



Test endpoint connection (optional)

Test your endpoint connection by selecting a replication instance within your desired VPC. After clicking "Run test", an endpoint will be created with the details provided and attempt to connect to the instance. If the connection fails, you can edit and test it again. Endpoints that aren't saved will be deleted.

VPC
vpc-0425287e

Replication instance
A replication instance performs the database migration
myreplicationinstance

After clicking "Run test", an endpoint will be created with the details provided and attempt to connect to the instance. If the connection fails, you can edit and test it again. Endpoints that aren't saved will be deleted.

Endpoint identifier	Replication instance	Status	Message
my-database	myreplicationinstance	successful	

Follow these same steps again to create an endpoint for your Target database. Unlike the Source database, you need to fill out the connection endpoint, port, and credentials yourself.

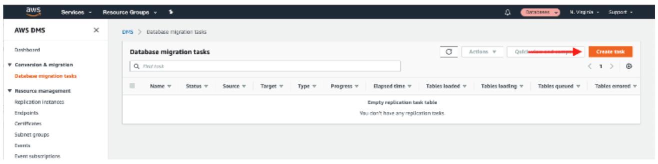
You also need to ensure that your replication instance has network access to your source database.

Before moving forward, you should have two endpoints configured: one for your source database and one for your target database. Make sure that you have tested both endpoints and can successfully connect to both databases. Then move on to the next module.

Creating Replication Task

A replication task is responsible for migrating data from a source database to a target database. In your case, you are moving data from an existing database to your newly created database in Amazon RDS.

1 To get started, navigate to the Database migration tasks section of the AWS DMS console. Choose Create task to create a new replication task.



Database migration tasks

Name	Status	Source	Target	Type	Progress	Elapsed time	Tables loaded	Tables loading	Tables queued	Tables errored

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

2 In the Task configuration section, set up the parameters of your replication task. Give your task a name and choose the replication instance you created in an earlier module. Then choose the source endpoint for your existing database and your target endpoint for your fully managed database in Amazon RDS.

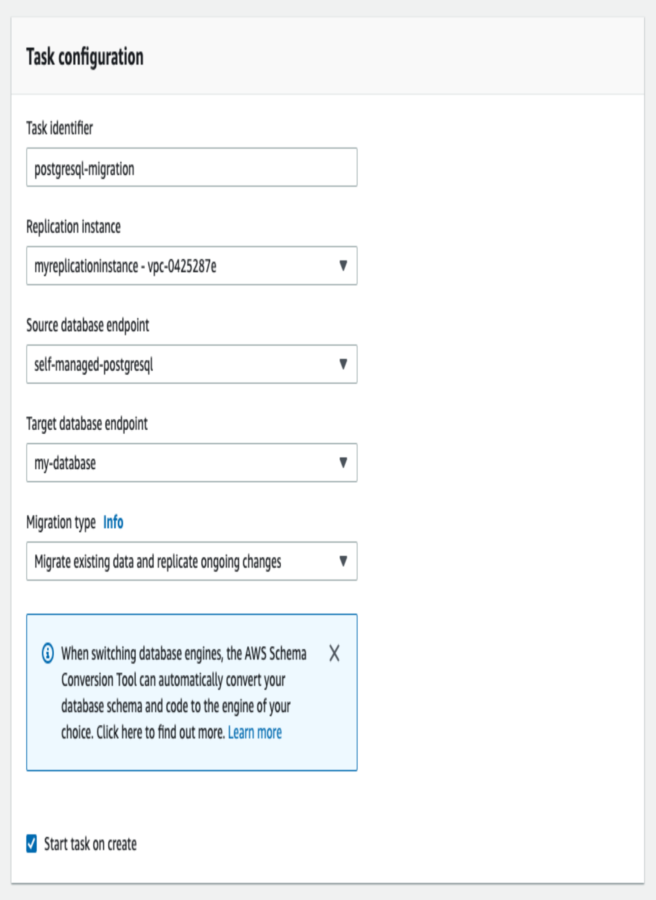
You need to choose a migration type. There are two migration types:

1. Migrate existing data, which performs a one-time process to copy data from your source database to your target database.

2. Replicate ongoing changes, which copies all ongoing operations from your source database to your target database.

If you are migrating your application from using a self-managed database to using a fully managed database, you want to use both types. The first type copies all data in your database, and the second type ensures that all additional updates are replicated to your new database until you switch your application to use the new database.

For the migration type, choose Migrate existing data and replicate ongoing changes. Note that this requires you to have logical replication enabled on your source database.



Task configuration

Task identifier: postgresql-migration

Replication instance: myreplicationinstance - vpc-0425287e

Source database endpoint: self-managed-postgresql

Target database endpoint: my-database

Migration type Info: Migrate existing data and replicate ongoing changes

When switching database engines, the AWS Schema Conversion Tool can automatically convert your database schema and code to the engine of your choice. Click here to find out more. Learn more

☑ Start task on create

3 In the Table mappings section, choose the tables you want to copy over. Enter the name of the schemas and tables you want to copy. You can use % as a wildcard character to copy multiple tables or schemas.

In this walkthrough, you migrate your tables exactly as they exist in the source database. If you have more complex requirements, AWS DMS supports filters and transformations of your source data before writing to your target database.

When you are ready, choose Create task to start your replication task.

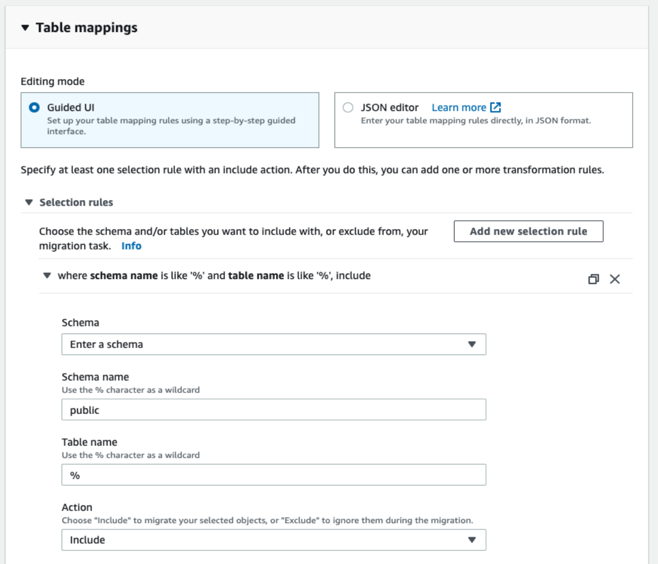


Table mappings

Editing mode

● Guided UI — Set up your table mapping rules using a step-by-step guided interface.

○ JSON editor Learn more — Enter your table mapping rules 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. Info

where **schema name** is like '%' and **table name** is like '%', include

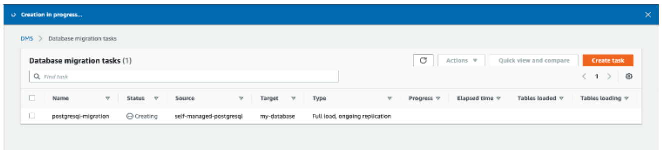
Schema: Enter a schema

Schema name — Use the % character as a wildcard: public

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

4 After you create your task, your task is shown in the Database migration tasks section with a Status of Creating.



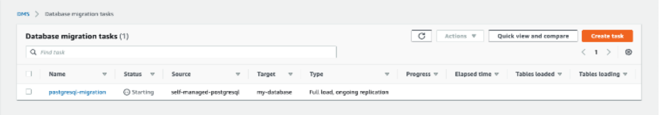
Creation in progress...

DMS > Database migration tasks

Database migration tasks (1)

Name	Status	Source	Target	Type	Progress	Elapsed time	Tables loaded	Tables loading
postgresql-migration	Creating	self-managed-postgresql	my-database	Full load, ongoing replication				

5 After the task is initialized, its Status is Starting.

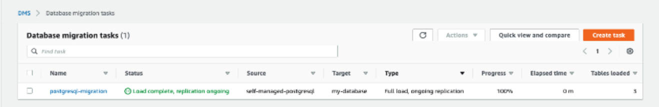


DMS > Database migration tasks

Database migration tasks (1)

Name	Status	Source	Target	Type	Progress	Elapsed time	Tables loaded	Tables loading
postgresql-migration	Starting	self-managed-postgresql	my-database	Full load, ongoing replication				

6 After the migration of existing data is complete, it shows a Status of Load complete, replication ongoing. Any updates to your source database at this point are copied to your target database.



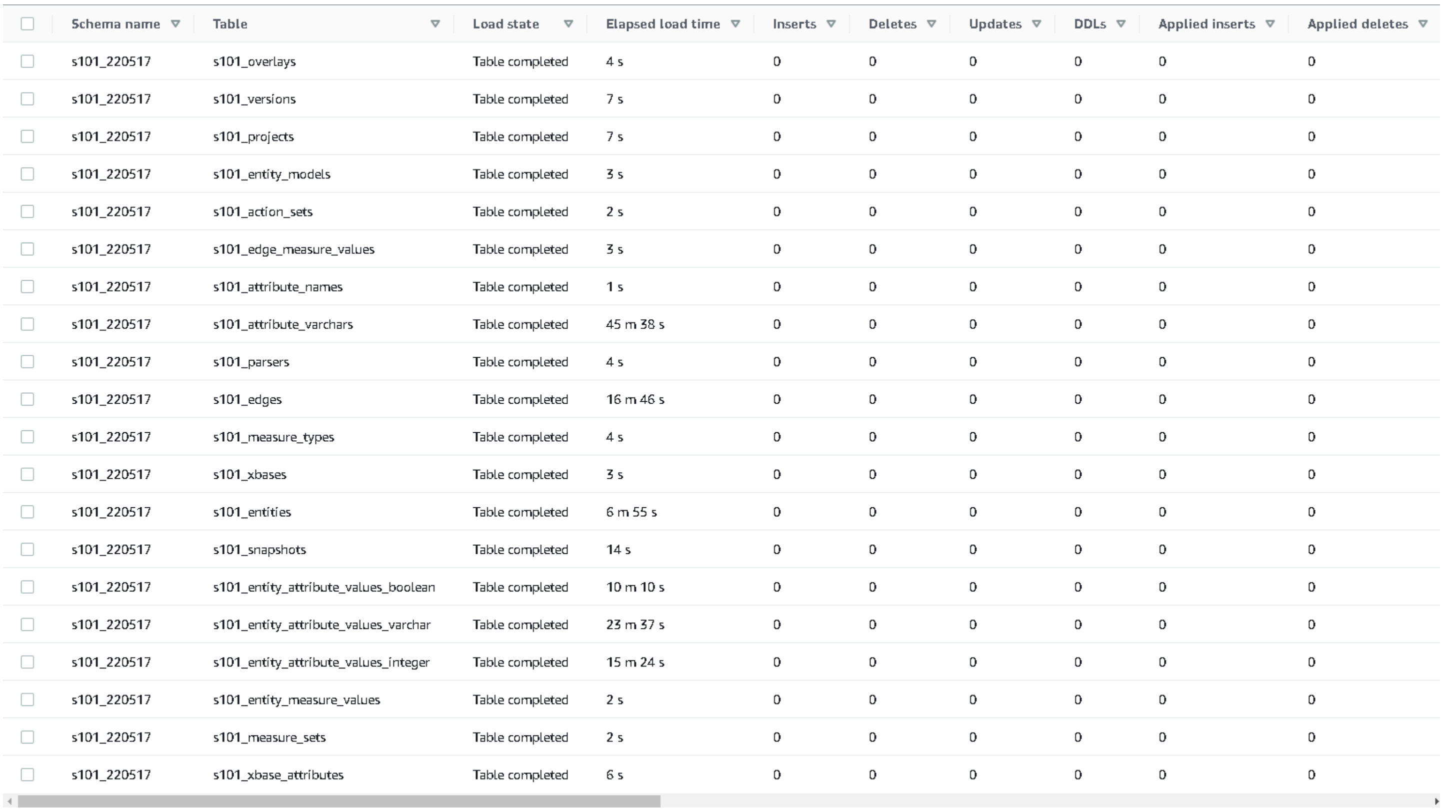
DMS > Database migration tasks

Database migration tasks (1)

Name	Status	Source	Target	Type	Progress	Elapsed time	Tables loaded
postgresql-migration	Load complete, replication ongoing	self-managed-postgresql	my-database	Full load, ongoing replication	100%	0 m	5

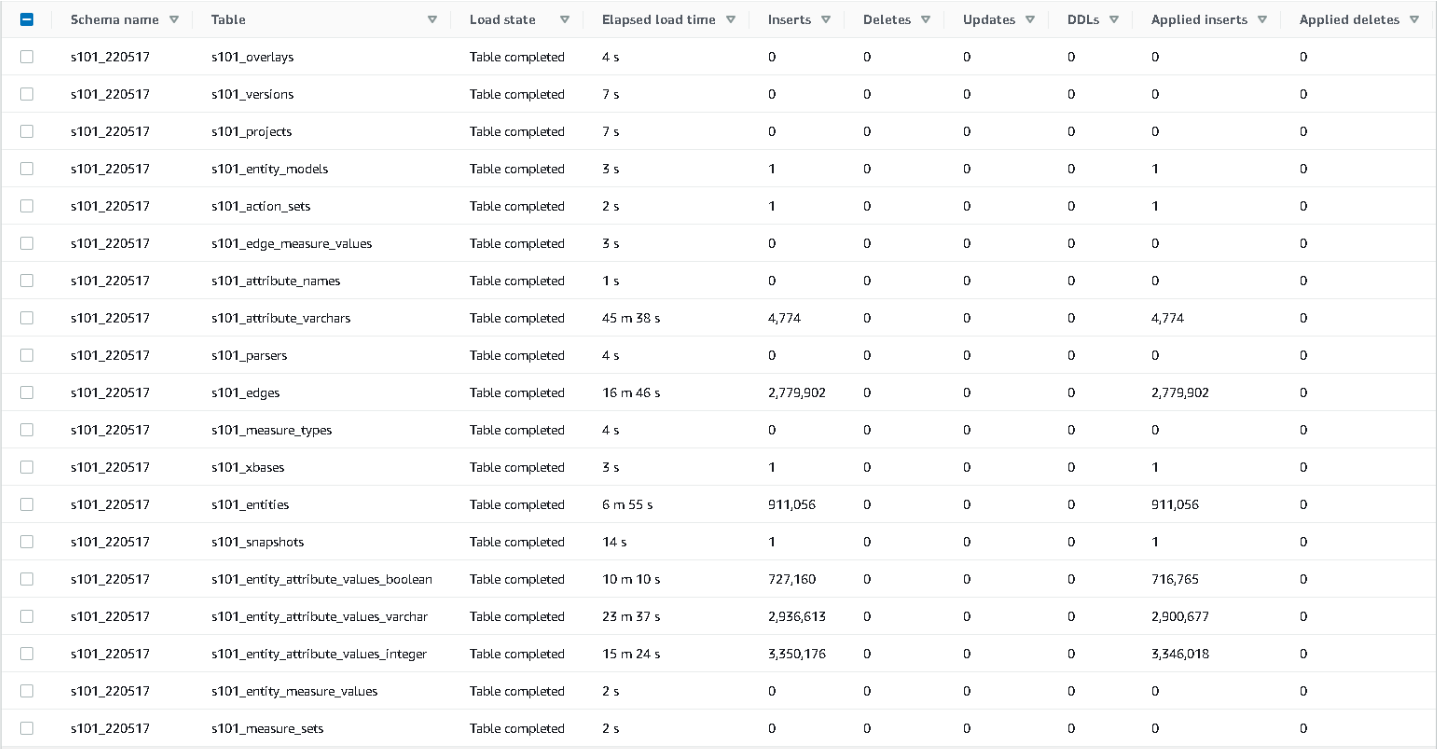
Sample timeline

Full Load for a 14 GB database.



☐	Schema name	Table	Load state	Elapsed load time	Inserts	Deletes	Updates	DDLs	Applied inserts	Applied deletes
☐	s101_220517	s101_overlays	Table completed	4 s	0	0	0	0	0	0
☐	s101_220517	s101_versions	Table completed	7 s	0	0	0	0	0	0
☐	s101_220517	s101_projects	Table completed	7 s	0	0	0	0	0	0
☐	s101_220517	s101_entity_models	Table completed	3 s	0	0	0	0	0	0
☐	s101_220517	s101_action_sets	Table completed	2 s	0	0	0	0	0	0
☐	s101_220517	s101_edge_measure_values	Table completed	3 s	0	0	0	0	0	0
☐	s101_220517	s101_attribute_names	Table completed	1 s	0	0	0	0	0	0
☐	s101_220517	s101_attribute_varchars	Table completed	45 m 38 s	0	0	0	0	0	0
☐	s101_220517	s101_parsers	Table completed	4 s	0	0	0	0	0	0
☐	s101_220517	s101_edges	Table completed	16 m 46 s	0	0	0	0	0	0
☐	s101_220517	s101_measure_types	Table completed	4 s	0	0	0	0	0	0
☐	s101_220517	s101_xbases	Table completed	3 s	0	0	0	0	0	0
☐	s101_220517	s101_entities	Table completed	6 m 55 s	0	0	0	0	0	0
☐	s101_220517	s101_snapshots	Table completed	14 s	0	0	0	0	0	0
☐	s101_220517	s101_entity_attribute_values_boolean	Table completed	10 m 10 s	0	0	0	0	0	0
☐	s101_220517	s101_entity_attribute_values_varchar	Table completed	23 m 37 s	0	0	0	0	0	0
☐	s101_220517	s101_entity_attribute_values_integer	Table completed	15 m 24 s	0	0	0	0	0	0
☐	s101_220517	s101_entity_measure_values	Table completed	2 s	0	0	0	0	0	0
☐	s101_220517	s101_measure_sets	Table completed	2 s	0	0	0	0	0	0
☐	s101_220517	s101_xbase_attributes	Table completed	6 s	0	0	0	0	0	0

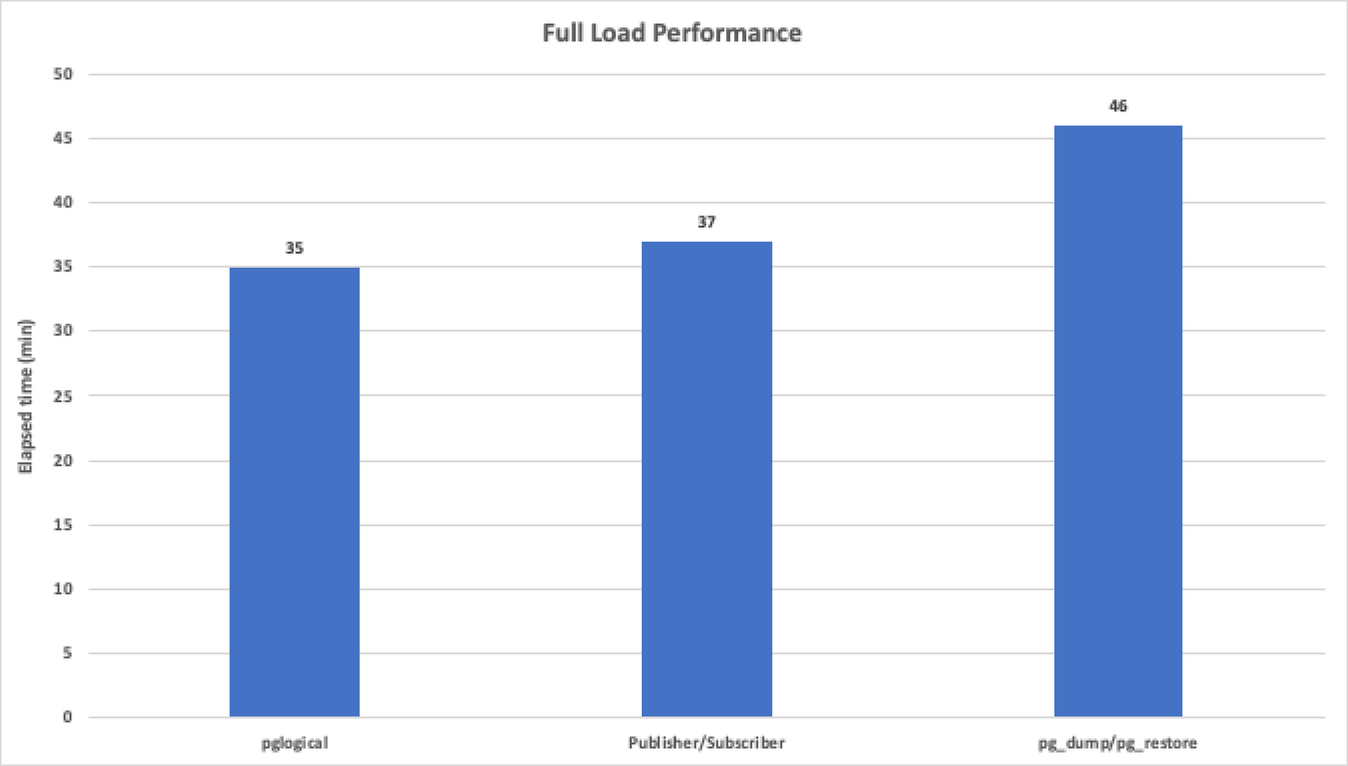
Incremental Replication for around additional 7 GB of Data



⊟	Schema name	Table	Load state	Elapsed load time	Inserts	Deletes	Updates	DDLs	Applied inserts	Applied deletes
☐	s101_220517	s101_overlays	Table completed	4 s	0	0	0	0	0	0
☐	s101_220517	s101_versions	Table completed	7 s	0	0	0	0	0	0
☐	s101_220517	s101_projects	Table completed	7 s	0	0	0	0	0	0
☐	s101_220517	s101_entity_models	Table completed	3 s	1	0	0	0	1	0
☐	s101_220517	s101_action_sets	Table completed	2 s	1	0	0	0	1	0
☐	s101_220517	s101_edge_measure_values	Table completed	3 s	0	0	0	0	0	0
☐	s101_220517	s101_attribute_names	Table completed	1 s	0	0	0	0	0	0
☐	s101_220517	s101_attribute_varchars	Table completed	45 m 38 s	4,774	0	0	0	4,774	0
☐	s101_220517	s101_parsers	Table completed	4 s	0	0	0	0	0	0
☐	s101_220517	s101_edges	Table completed	16 m 46 s	2,779,902	0	0	0	2,779,902	0
☐	s101_220517	s101_measure_types	Table completed	4 s	0	0	0	0	0	0
☐	s101_220517	s101_xbases	Table completed	3 s	1	0	0	0	1	0
☐	s101_220517	s101_entities	Table completed	6 m 55 s	911,056	0	0	0	911,056	0
☐	s101_220517	s101_snapshots	Table completed	14 s	1	0	0	0	1	0
☐	s101_220517	s101_entity_attribute_values_boolean	Table completed	10 m 10 s	727,160	0	0	0	716,765	0
☐	s101_220517	s101_entity_attribute_values_varchar	Table completed	23 m 37 s	2,936,613	0	0	0	2,900,677	0
☐	s101_220517	s101_entity_attribute_values_integer	Table completed	15 m 24 s	3,350,176	0	0	0	3,346,018	0
☐	s101_220517	s101_entity_measure_values	Table completed	2 s	0	0	0	0	0	0
☐	s101_220517	s101_measure_sets	Table completed	2 s	0	0	0	0	0	0

Performance Comparison

Performance of pg_dump and pg_restore, publisher and subscriber, and pglogical in a full load migration.



Source: https://docs.aws.amazon.com/dms/latest/sbs/chap-manageddatabases.postgresql-rds-postgresql-performance-comparison.html