

# Oracle RDS to Aurora PostgreSQL migration lab using DMS and SCT:

The following instructions help you launch a CloudFormation stack which sets up the resources required for this lab. Once the resources are launched you are guided through the migration of a sample database from Oracle or SQLServer to Aurora PostgreSQL or MySQL using the Database Migration Service (DMS) and the Schema Conversion Tool (SCT). The resources launched by the CloudFormation stack include:

1. An Oracle RDS source within a VPC with all networking resources configured with a sample database installed from <https://github.com/aws-labs/aws-database-migrationsamples>
2. An Aurora target within same VPC as #1 with all networking resources configured
3. A Windows Host with the Schema Conversion Tool pre-installed within same VPC as #1
4. A Database Migration Service Instance within same VPC as #1
5. The CloudFormation script will provision the necessary security groups with access to database resources from #3 & #4 and allow RDP access to #3

## High level steps:

1. Log into your AWS account
2. Launch the CloudFormation stack
3. Log in to the Windows Host and install JDBC Drivers
4. Use the SCT to convert source Oracle schema objects to target Aurora PostgreSQL
5. Create a source endpoint (the source database is launched using the CloudFormation stack)
6. Create a target endpoint (the target database is launched using the CloudFormation stack)
7. Create a task to migrate the data from the source database to the target database 8. Verify the data has been migrated to the target

## Create an EC2 Key Pair if Needed

You will need an EC2 key pair in order to log into the SCT instance launched as part of your stack. If you have access to an existing one in the region where your assets are launched, you can skip this step and use it, (however you should check to be sure.) If you don't have access to an existing key pair, you will need to create one, (you can remove it once the lab is complete if you want to.) To create a new key pair:

1. Open the EC2 service console
2. In the left hand gutter under "Network & Security" click "Key Pairs"
3. Supply a name and click "Create"
4. Download or otherwise save the .pem file **Lab Instructions:**
1. Log in to your AWS console and go to the CloudFormation landing page - <https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks?filter=active>

- Click create stack, select **'Specify an Amazon S3 template URL'** and launch the CloudFormation stack from this url: <https://s3-us-west-2.amazonaws.com/mtt-reinvent-2018/reInvent-2018-DMS-Workshop-v1.4.template>
- Here is a screenshot of the first page –

## Create stack

**Select Template**  
Specify Details  
Options  
Review

**Select Template**  
Select the template that describes the stack that you want to create. A stack is a group of related resources that you manage as a single unit.

**Design a template** Use AWS CloudFormation Designer to create or modify an existing template. [Learn more.](#)  
Design template

**Choose a template** A template is a JSON/YAML-formatted text file that describes your stack's resources and their properties. [Learn more.](#)  
☐ Select a sample template  
  
☐ Upload a template to Amazon S3  
Choose File No file chosen  
☒ Specify an Amazon S3 template URL  

<https://s3-us-west-2.amazonaws.com/dmsworkshop/Oracle> [View/Edit template in Designer](#)

- Hit next to bring up the stack details page

**Specify Details**  
Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Stack name

**Parameters**

**Source RDS Database Configuration**  

|                      |  |   |
|----------------------|--|---|
| SourceDBUsername     | <input type="text" value="dbmaster"/>    | Enter database Admin username for the Source DB |
| SourceDBPassword     | <input type="text" value="srcadmin123"/> | Enter password for the admin user: dbmaster     |
| SourceDBStorage      | <input type="text" value="50"/>          | Enter storage for Source DB in GB               |
| SourceDBInstanceType | <input type="text" value="db.r4.large"/> | Source RDS DB Instance type                     |
| SourceDBEngineType   | <input type="text" value="oracle-se2"/>  | Source DB Engine type                           |

**Target Aurora Database Configuration**  

|                    |  |  |
|--------------------|--|--|
| AuroraDBUsername   | <input type="text" value="auradmin"/>          | Enter database Admin username for Aurora RDS |
| AuroraDBPassword   | <input type="text" value="auroradmin123"/>     | Enter password for Aurora RDS Admin user     |
| AuroraInstanceType | <input type="text" value="db.r4.large"/>       | Aurora DB Instance type                      |
| AuroraEngineType   | <input type="text" value="aurora-postgresql"/> | Aurora DB Engine type                        |

**Schema Conversion Tool Host Configuration**  

|                     |   |   |
|---------------------|---|---|
| SCTHostKeyName      | <input type="text" value="Search"/>     | Name of an existing EC2 KeyPair to enable Remote Desktop access to the instance |
| SCTHostInstanceType | <input type="text" value="c4.2xlarge"/> | SCT EC2 Instance type   |

**DMS Instance Configuration**  

|                    |   |                                      |
|--------------------|---|--------------------------------------|
| DMSInstanceType    | <input type="text" value="dms.c4.large"/> | DMS Instance type                    |
| DMSInstanceStorage | <input type="text" value="50"/>           | Enter storage for DMS Instance in GB |
| DMSInstanceVersion | <input type="text" value="2.4.3"/>        | DMS Engine Version to use            |

**Other parameters**  

|                     |                                      |   |
|---------------------|--------------------------------------|---|
| CreateRoleResources | <input type="text" value="true"/>    | Should the template attempt to create the required Roles for DMS  |
| LaptopIPAddress     | <input type="text" value="0.0.0.0"/> | Enter the IP Address of the machine you will access the Workshop From. You can use <a href="http://idehelp.amazonaws.com">http://idehelp.amazonaws.com</a> to obtain it |

Cancel

Previous

Next

- Do the following for the entries on this page
  - Stack name** - give a friendly name such as dmslab for the stack

- b. **sourceDBInstanceType** – Pick any instance type. Maybe pick something different than your neighbor to spread things around a bit.
  - c. **SourceDBEngineType** – choose oracle-se2
  - d. **SCTHostKeyName** – Select one of your existing EC2 KeyPairs to enable access to the SCT Host. **You select a value from the list!** If you do not have KeyPairs already please see the instructions above on how to create one.
  - e. **CreateRoleResources** – This should be true, **unless you have run DMS before in this region and the required IAM roles have already been created**. Setting this to true when the roles already exist will cause the Stack Creation to fail, and you will need to re-run the stack creation with this set to false. If you aren't sure, try it with "true" first. If it fails, try again with "false."
  - f. **LaptopIPAddress** – the IP address you are running the workshop from, this will limit access to the SCT Host only from your laptop. If you want to limit access to your SCT host enter your laptop IP address here.
  - g. Finally Hit next
6. On the next page, simply hit next

Capabilities

**i** The following resource(s) require capabilities: [AWS::IAM::Role]  
This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#).

☐ I acknowledge that AWS CloudFormation might create IAM resources with custom names.

Options

Tags

You can specify tags (key-value pairs) for resources in your stack. You can add up to 50 unique key-value pairs for each stack. [Learn more](#).

| Key (127 characters maximum) | Value (255 characters maximum) |
|------------------------------|--------------------------------|
| 1                            |                                |

Permissions

You can choose an IAM role that CloudFormation uses to create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses the permissions defined in your account. [Learn more](#).

IAM Role:   
Enter role arn:

▼ Rollback Triggers

Rollback triggers enable you to have AWS CloudFormation monitor the state of your application during stack creation and updating, and to rollback that operation if the application breaches the threshold of any of the alarms you've specified. [Learn more](#)

Monitoring Time:  Minutes  
Minimum value of 0. Maximum value of 100.

| Type                     | ARN (Amazon Resource Name) |
|--------------------------|----------------------------|
| 1 AWS::CloudWatch::Alarm |                            |

Available triggers remaining: 5

► Advanced

You can set additional options for your stack, like notification options and a stack policy. [Learn more](#).

Cancel Previous Next

7. On the next page, check the check box to allow CloudFormation to create IAM resources with custom names. Then click next!

\*Leave everything else default. Once the CloudFormation stack launch is complete, you should see 2 RDS instances in your account in the region you created the stack. Approximately, it will **take 15 to 20 minutes** for creating the Stack.

8. When the stack creation completes, look at the Outputs section. It will provide you with the names etc. of the resources needed to complete the workshop.

| Overview             | Outputs  | Resources   | Events | Template | Parameters | Tags | Stack Policy | Change Sets | Rollback Triggers |
|----------------------|--|-------------|--------|----------|------------|------|--------------|-------------|-------------------|
| Key                  | Value  | Description |        |          |            |      |              |             |                   |
| DMSInstanceName      | dmslab-DMS-Instance  | None Yet    |        |          |            |      |              |             |                   |
| TargetDBUserPassword | auroradmin123  | None Yet    |        |          |            |      |              |             |                   |
| SCTHostPublicDNS     | ec2-34-221-254-187.us-west-2.compute.amazonaws.com                       | None Yet    |        |          |            |      |              |             |                   |
| Regionname           | us-west-2  |             |        |          |            |      |              |             |                   |
| SourceDBUser         | dbmaster   | None Yet    |        |          |            |      |              |             |                   |
| SCTHostUsername      | Administrator  | None Yet    |        |          |            |      |              |             |                   |
| TargetDBUser         | auradmin   | None Yet    |        |          |            |      |              |             |                   |
| TargetDBHostname     | dmslab-targetdb-cluster.cluster-cx8adsyv4hit.us-west-2.rds.amazonaws.com | None Yet    |        |          |            |      |              |             |                   |
| SCTHostName          | dmslab-SCT-Instance  | None Yet    |        |          |            |      |              |             |                   |
| TargetDBPort         | 3306   | None Yet    |        |          |            |      |              |             |                   |
| SourceDBUserPassword | srcadmin123  | None Yet    |        |          |            |      |              |             |                   |
| SourceDBPort         | 1521   | None Yet    |        |          |            |      |              |             |                   |
| StackName            | dmslab   |             |        |          |            |      |              |             |                   |
| SourceDBHostname     | dmslab-sourcedb.cx8adsyv4hit.us-west-2.rds.amazonaws.com                 | None Yet    |        |          |            |      |              |             |                   |

9. Once stack is available, Log in to the SCT host using RDP and the information provided in the stack Output section. The username to connect as is “Administrator”. You will need to open a new tab and go to the EC2 Console to decrypt the password using the KeyPair you provided during the Stack creation.
- Go to the EC2 console.
  - Find and select the ec2 instance you just created. The name should contain your stack name.
  - Click the “connect” button at the top of the console.
  - Supply the keypair when prompted to do so.
  - Launch RDP and connect as Administrator using the password just obtained.
10. Once on the host, open the document “JDBC Driver Install Instructions.rtf” located on the desktop and follow the instructions for installing the JDBC drivers for the database engines selected during stack creation.
- If you are doing the Oracle -> Aurora PostgreSQL lab you will need the Oracle and PostgreSQL drivers
  - If you are doing the SQL Server -> Aurora MySQL lab you will need the SQL Server and MySQL drivers

## JDBC Driver Install Instructions for AWS Schema Conversion Tool

### **Oracle:**

1) Download the Oracle JDBC driver from the following location:

<https://www.oracle.com/technetwork/database/features/jdbc/jdbc-ucp-122-3110062.html>

You want the file named "ojdbc8.jar", you will also need to accept the license agreement and sign in using an OTN account.

2) Once you have downloaded this file you should move it to the following location:

`C:\JDBC Drivers\Oracle`

3) When providing connection information for the database endpoint in the Schema Conversion Tool ensure you also specify the path to the location where you placed the JDBC driver.

### **PostgreSQL:**

1) Now you need to download the PostgreSQL JDBC Driver from the following location:

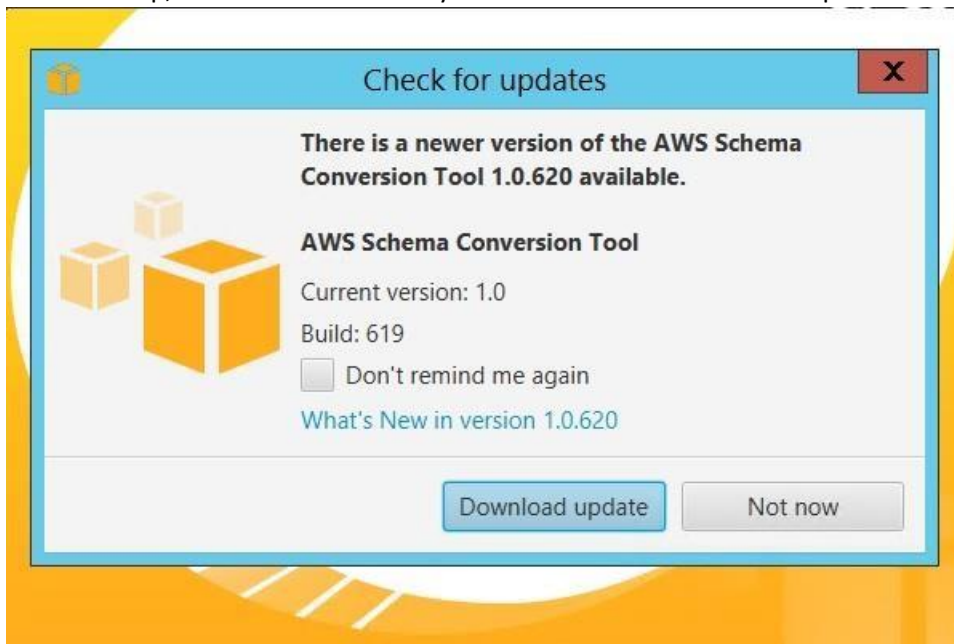
<https://jdbc.postgresql.org/download.html>

you want the file named "PostgreSQL JDBC 4.2 Driver, 42.2.5".

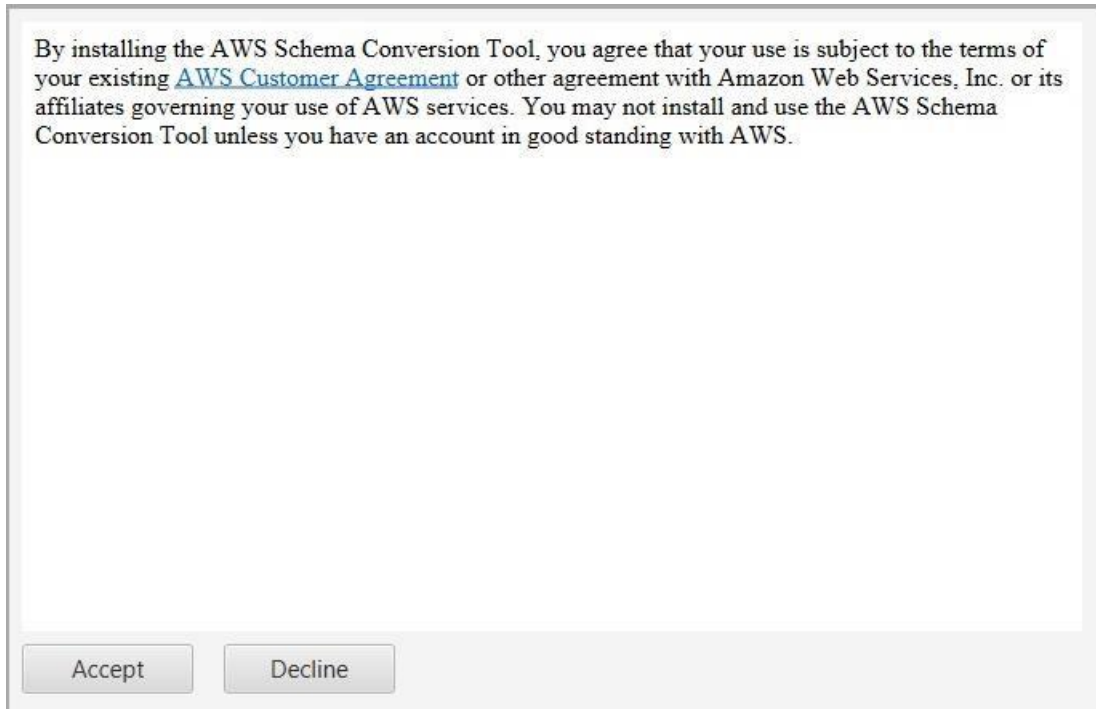
2) Next the PostgreSQL JDBC driver("postgresql-42.2.5.jar") should be moved to the following location:

### **Schema Conversion:**

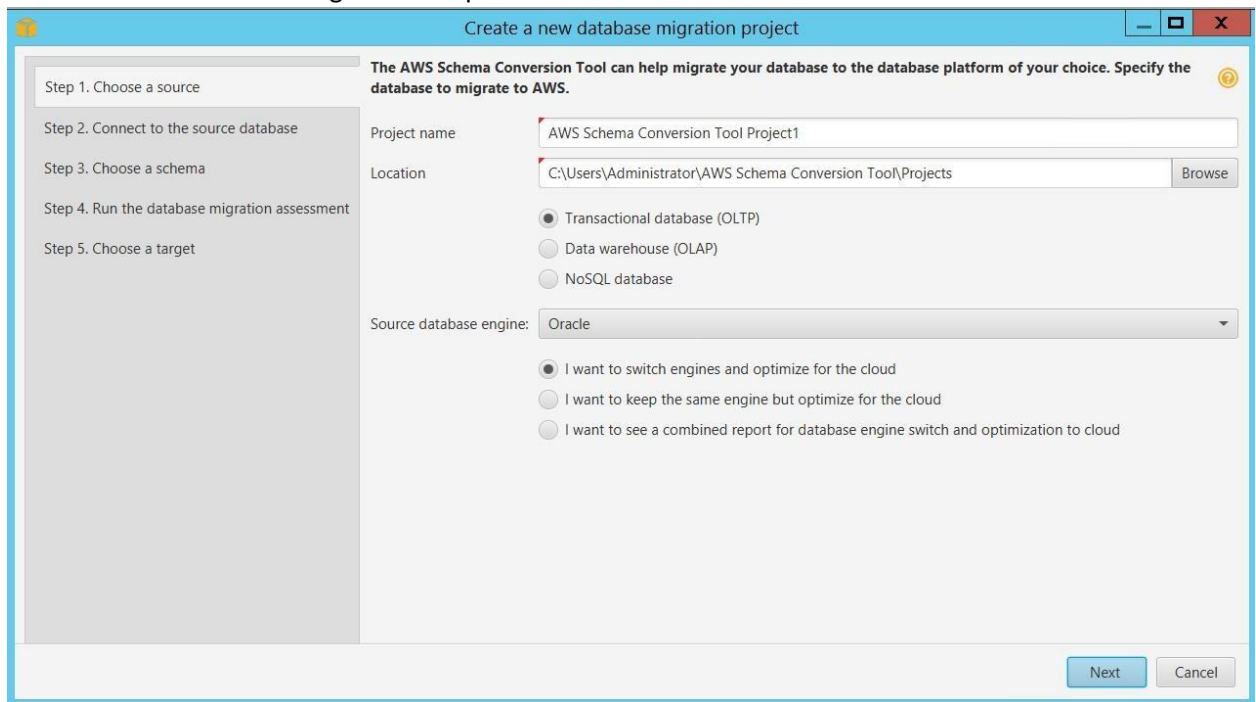
1. Launch SCT on your Windows Host – there's a shortcut for it on your desktop
2. You can choose to update the SCT installation or dismiss update dialog(this is recommended for the Workshop, in all other situations you will want to download the update and install it)



3. Accept the license agreement



4. The "Create a new database Migration Project" Wizard will launch. If it doesn't you should select this option from the File Menu. Make sure you have the following selected and then click Next.
  - a. Transactional database (OLTP)
  - b. Source database engine: Oracle
  - c. I want to switch engines and optimize for the cloud.



5. Provide the connection information for your source RDS Database.  
This information can be obtained from the Outputs section of the CloudFormation Stack.

Leave “Use SSL” unchecked

Check “Store password”

Browse for and find the jdbc driver you just installed.

For Oracle sources the SID to use is SRCDB.

For SQL Server sources – leave Instance Name blank

When you are finished press the “Test Connection” button.

Create a new database migration project

Step 1. Choose a source  
Step 2. Connect to the source database  
Step 3. Choose a schema  
Step 4. Run the database migration assessment  
Step 5. Choose a target

**Specify information about the source database to connect to.**

Note: The AWS Schema Conversion Tool doesn't store the password. If you close your AWS Schema Conversion Tool project and reopen it, you are prompted for the password to connect your source database as needed.

**Connect to Oracle**

Connection: SSL

Type: SID

Server name: dmslab-sourcedb.cx8adsyv4hit.us-west-2.rds.amazonaws.com

Server port: 1521

Oracle SID: SRCDB

User name: dbmaster

Password: .....

☐ Use SSL

☒ Store password

Oracle driver path: C:\JDBC Drivers\Oracle\ojdbc8.jar Browse

Test connection Previous Next Cancel

6. After a successful connection test. Click “OK”, then click “Next”

Create a new database migration project

Step 1. Choose a source  
Step 2. Connect to the source database  
Step 3. Choose a schema  
Step 4. Run the database migration assessment  
Step 5. Choose a target

**Specify information about the source database to connect to.**

Note: The AWS Schema Conversion Tool doesn't store the password. If you close your AWS Schema Conversion Tool project and reopen it, you are prompted for the password to connect your source database as needed.

**Connect to Oracle**

Connection: SSL

Type: SID

Server name: dmslab-sourcedb.cx8adsyv4hit.us-west-2.rds.amazonaws.com

Server port: 1521

Oracle SID: SRCDB

User name: dbmaster

Password: .....

☐ Use SSL

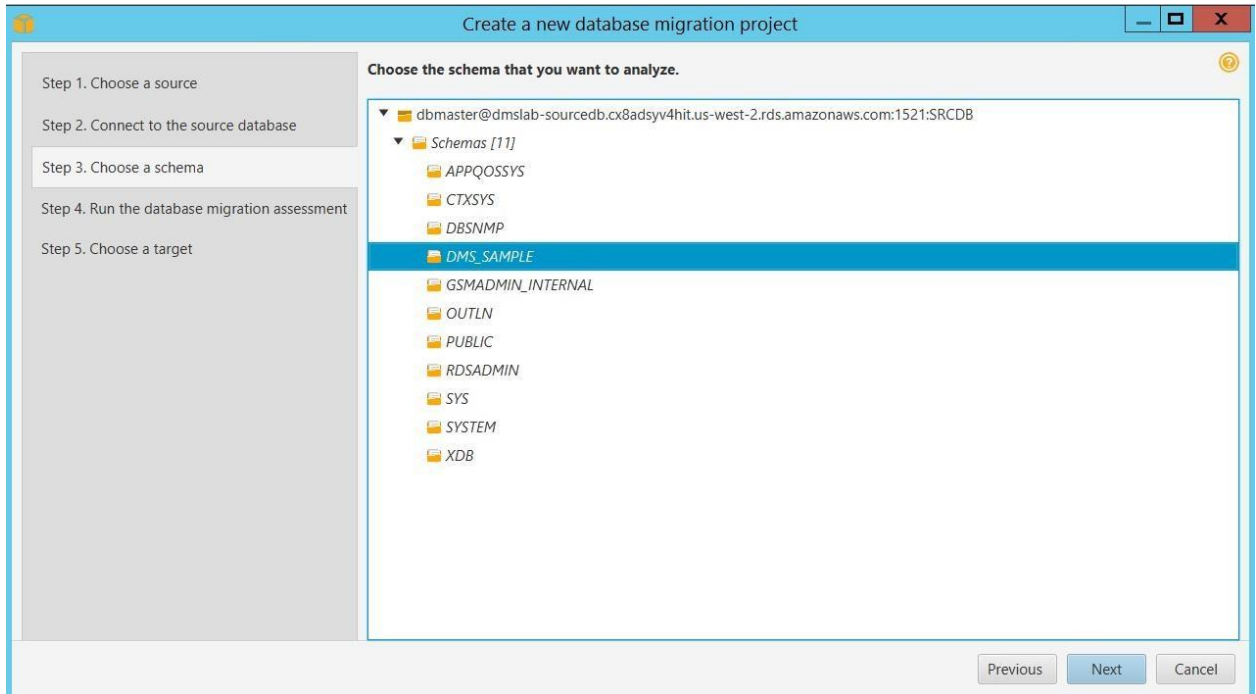
☒ Store password

Oracle driver path: C:\JDBC Drivers\Oracle\ojdbc8.jar Browse

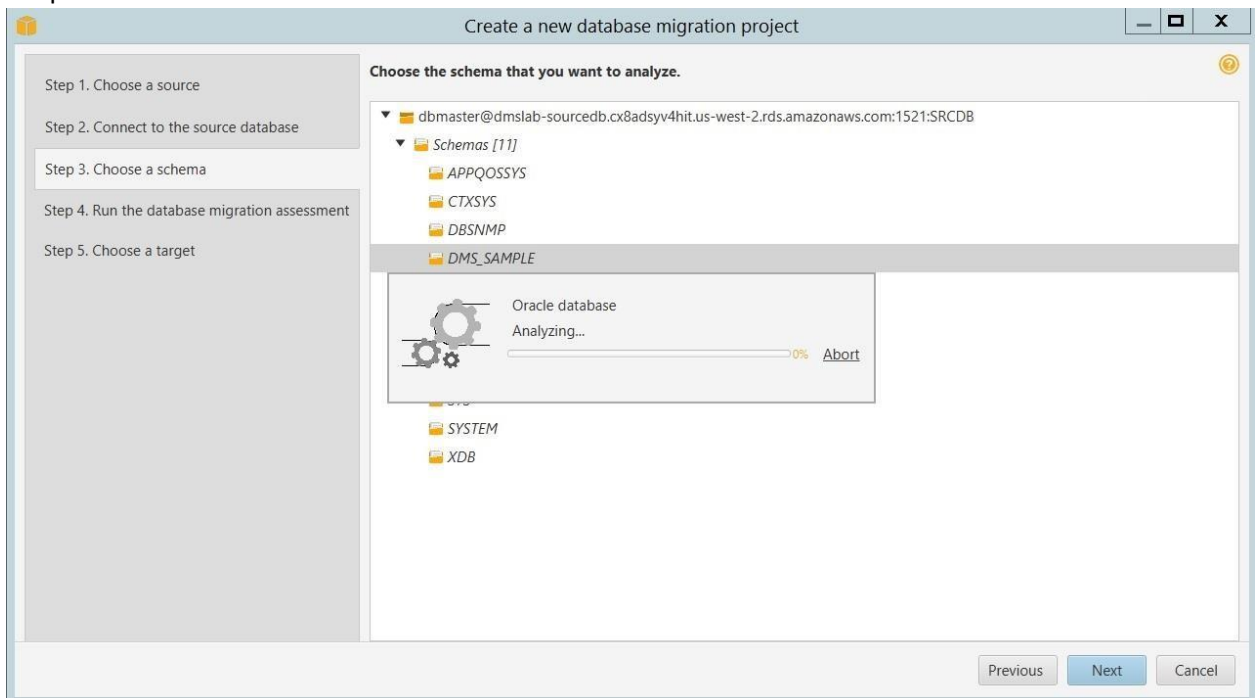
Test connection Previous Next Cancel



7. Now select the schema to convert and click next. For this workshop the schema will be “DMS\_SAMPLE” (dms\_sample.dbo for SQL Server)



8. The Schema Conversion Tool will now perform an analysis of the DMS\_SAMPLE, wait while it completes.



9. When the analysis is complete you will see the “Database migration assessment report”. Review it as it will provide information on various target databases and how much can automatically be converted. When done reviewing the report click “Next”



Create a new database migration project

Step 1. Choose a source  
Step 2. Connect to the source database  
Step 3. Choose a schema  
Step 4. Run the database migration assessment  
Step 5. Choose a target

Database Switch Assessment

### Database migration assessment report

Source database:  
DMS\_SAMPLE.dbmaster@dmslab-sourcedb.cx8adsyv4hit.us-west-2.rds.amazonaws.com:1521:SRCDDB  
Oracle Database 12c Standard Edition 12.1.0.2.0 (64bit Production), Standard edition

### Executive summary

We completed the analysis of your Oracle source database and estimate that 100% of the database storage objects and 57% of database code objects can be converted automatically or with minimal changes if you select Amazon RDS for MySQL as your migration target. Database storage objects include schemas, tables, table constraints, indexes, types, collection types, sequences, synonyms, view-constraints, clusters and database links. Database code objects include triggers, views, materialized views, materialized view logs, procedures, functions, packages, package constants, package cursors, package exceptions, package variables, package functions, package procedures, package types, package collection types, scheduler-jobs, scheduler-programs and scheduler-schedules. Based on our analysis of SQL syntax elements of your source database schema, we estimate that 90% of your entire database schema can be converted to Amazon RDS for MySQL automatically. To complete the migration, we recommend 64 conversion action(s) ranging from simple tasks to medium-complexity actions to significant conversion actions.

Previous Next Cancel

10. Now you will setup your connection to your target database. Select the correct target engine type (Amazon Aurora PostgreSQL Compatible d). For Aurora PostgreSQL compatible use postgres for the Database field, The rest of the information for filling in the option can be obtained from the outputs section of the CloudFormation stack you created. Once you have filled out all of the fields click “Test Connection”

Create a new database migration project

Step 1. Choose a source  
Step 2. Connect to the source database  
Step 3. Choose a schema  
Step 4. Run the database migration assessment  
Step 5. Choose a target

Specify the target database engine and the connection information.

Target database engine: Amazon Aurora (PostgreSQL compatible)

Connect to Amazon Aurora (PostgreSQL compatible)

Connection: SSL

Server name: dmslab-targetdb-cluster.cluster-cx8adsyv4hit.us-west-2.rds.amazonaws.com

Server port: 3306

Database: postgres

User name: auradmin

Password: .....

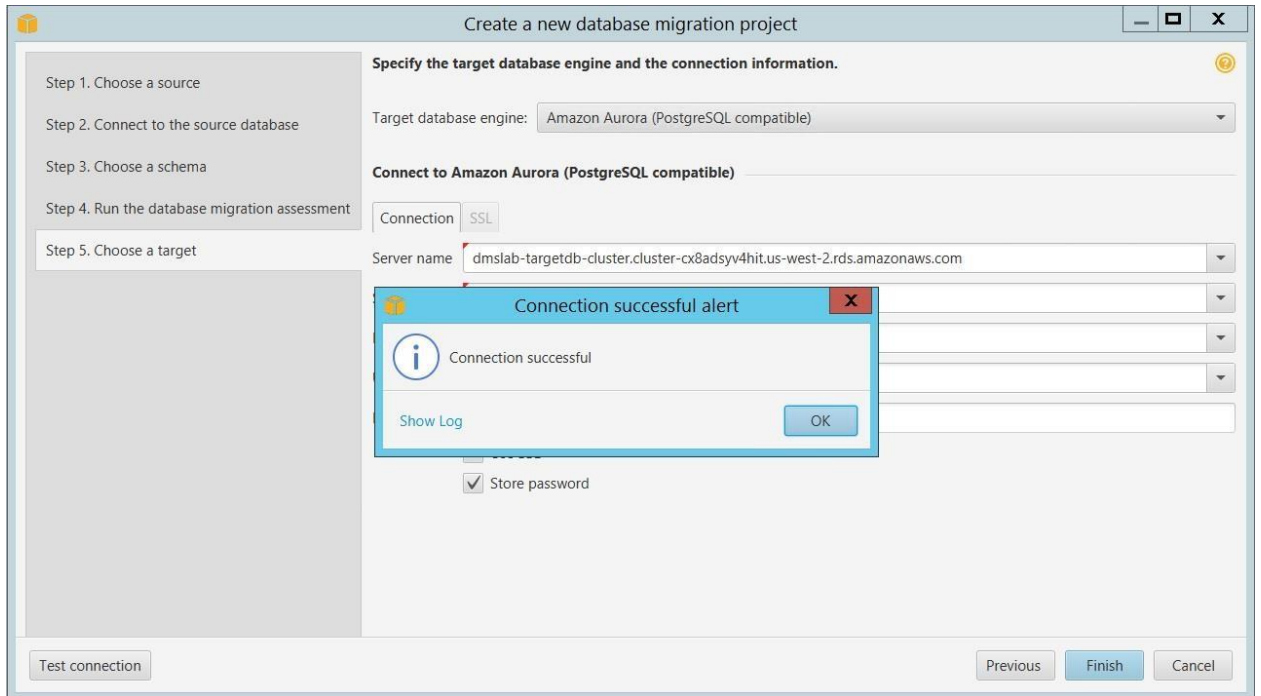
☐ Use SSL  
☒ Store password

Amazon Aurora (PostgreSQL compatible) driver path: C:\JDBC Drivers\PostgreSQL\postgresql-42.2.5.jar

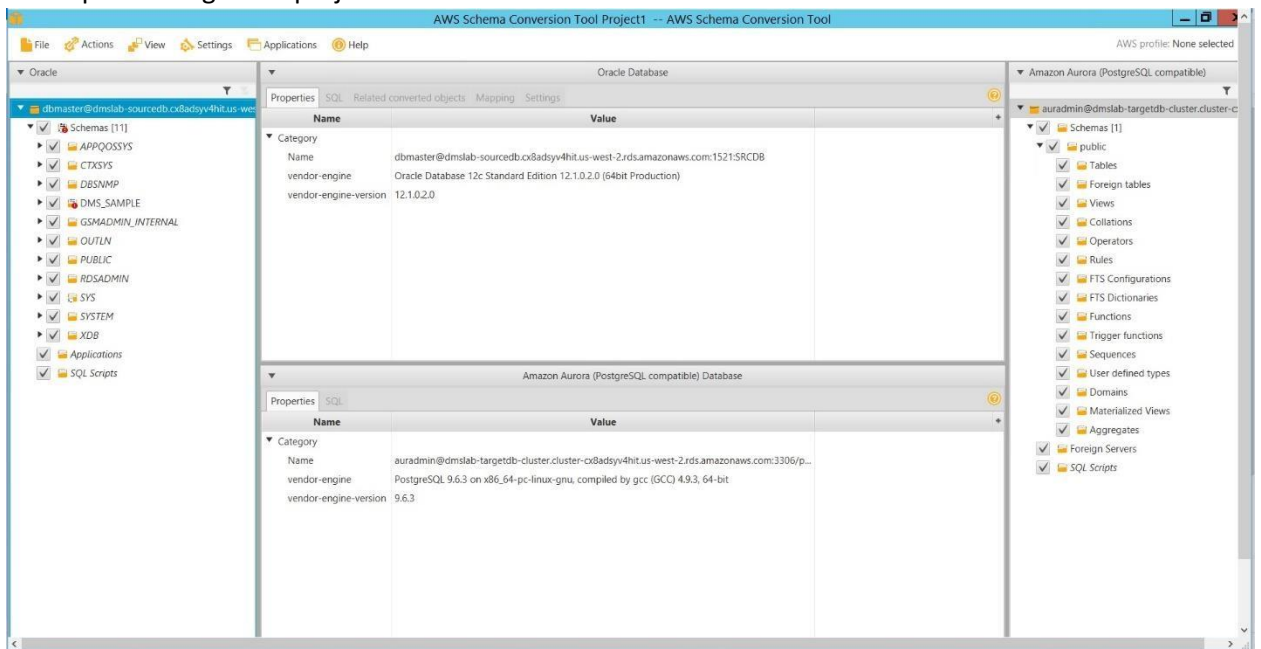
Test connection

Previous Finish Cancel

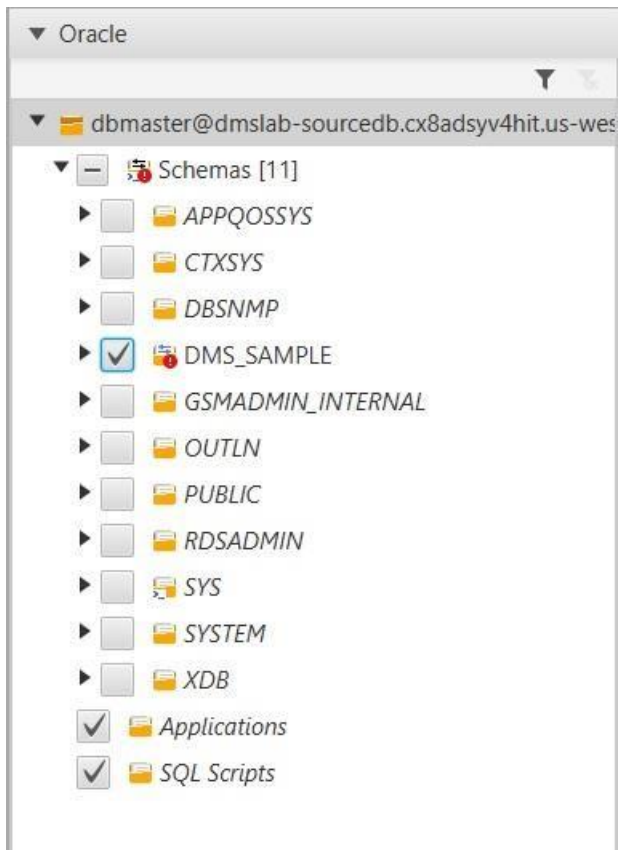
11. Once the connection test completes successfully click “OK” and then click “Finish”.



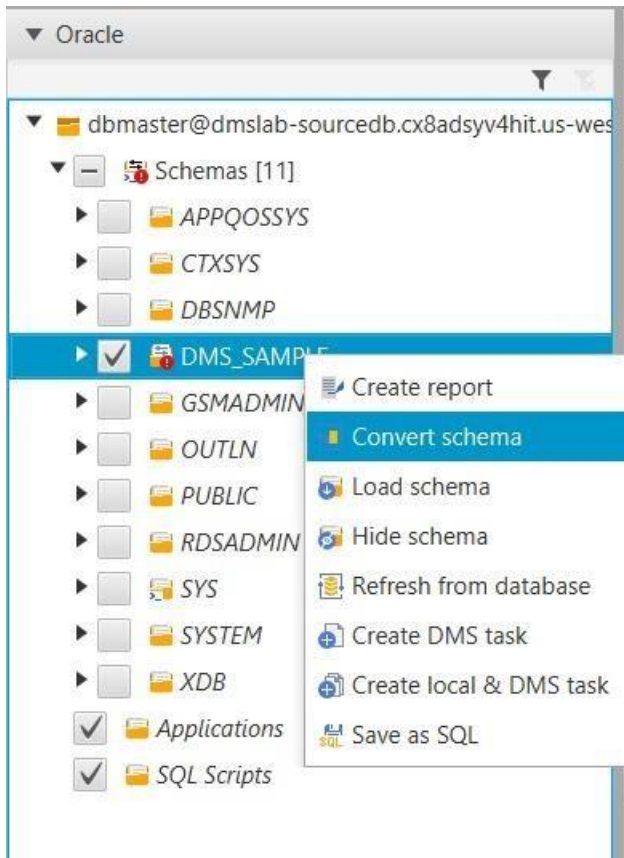
12. At this point a migration project will launch.



13. In the left hand panel deselect all of the source schemas except for DMS\_SAMPLE.



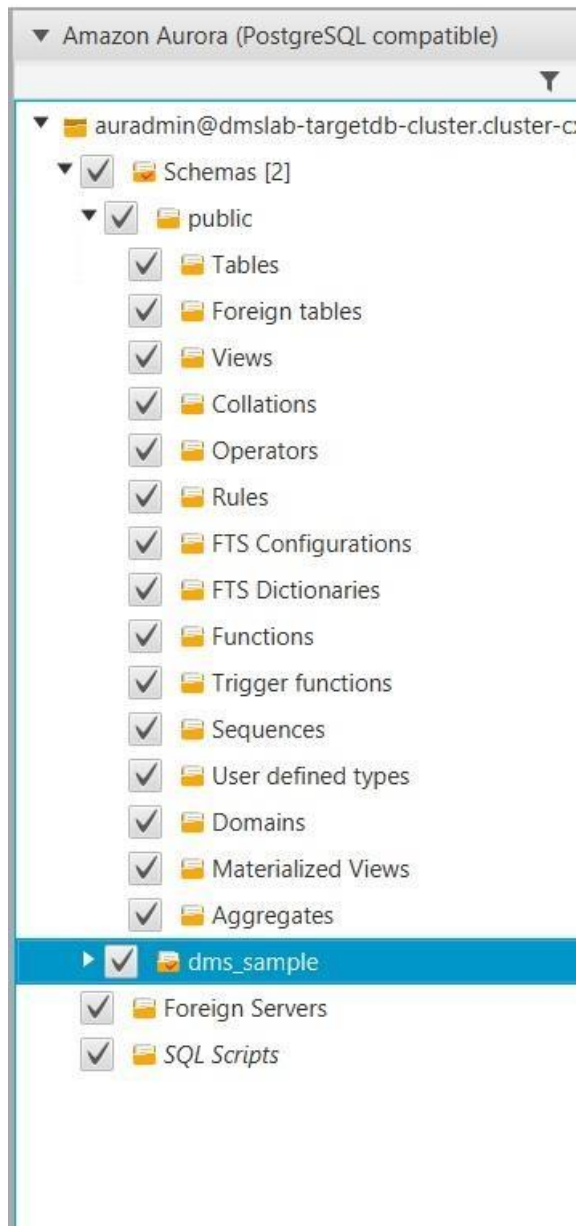
14. Now on the source schema panel. Select the DMS\_SAMPLE (or dms\_sample -> dbo) and right click to bring up the menu to convert the schema. You can also select convert schema from the Actions menu.



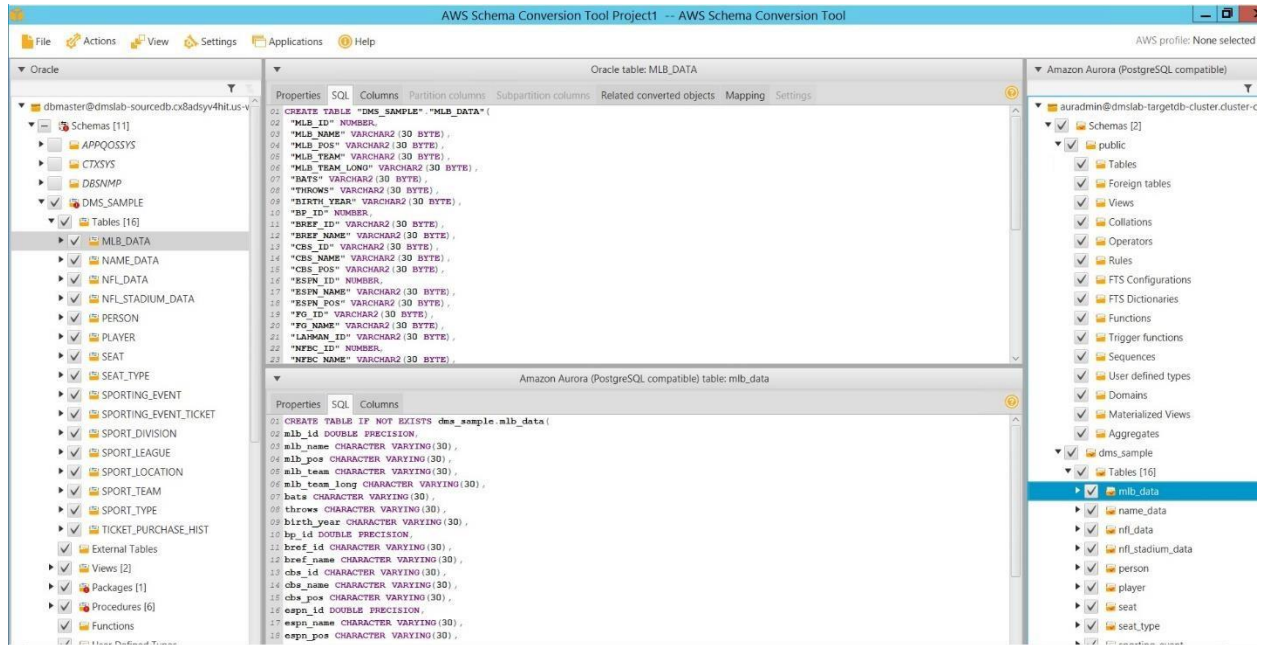
15. A confirmation dialog box will come popup. Click “Yes” to continue as there are no objects in the target database at this point.



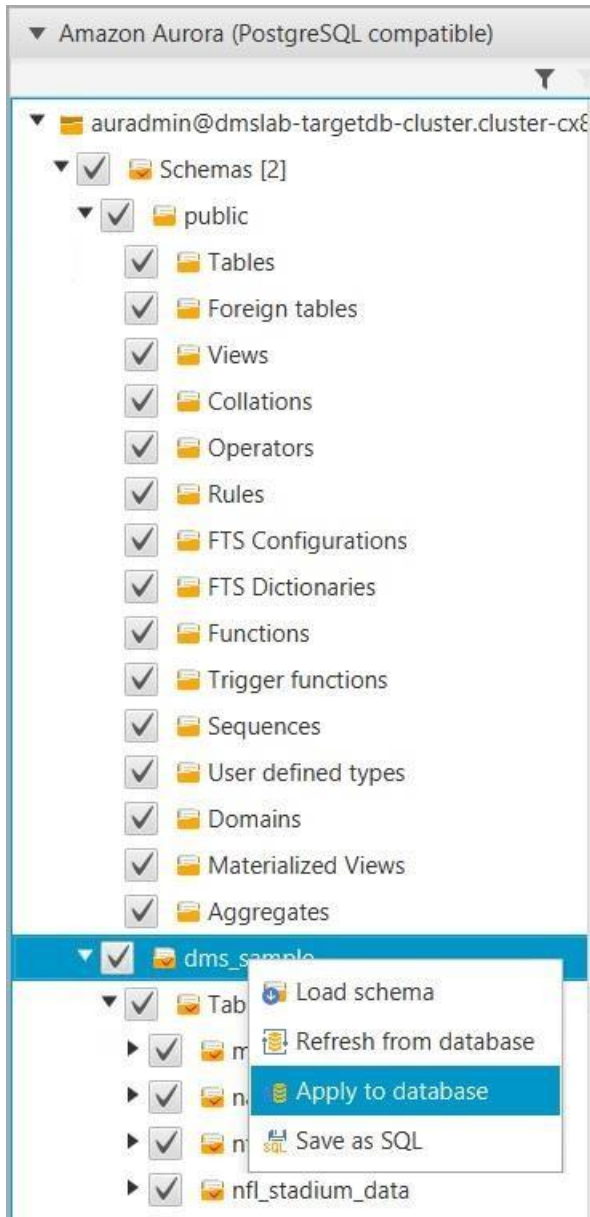
16. SCT will convert your source schema to a target schema that will be displayed in the right hand panel as either dms\_sample or dms\_sample\_dbo.



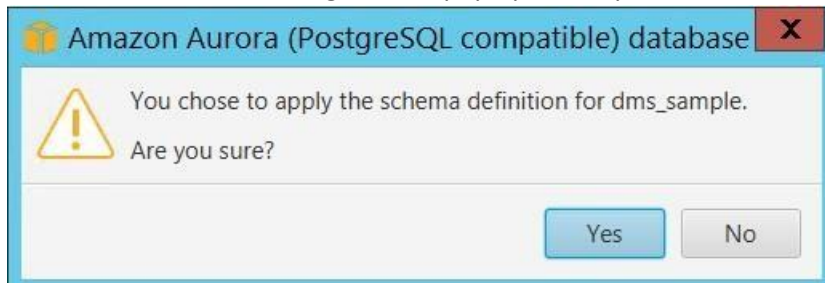
17. At this point you can select objects in either the source or target panels and see a comparison of original and converted code and objects. You can also make changes to the converted objects as desired. Look through which objects could be automatically converted and which could not be.



18. Since the majority of the objects which could not be converted are secondary objects like functions or procedures, right click on the schema that was created (in the right panel) and select "Apply to database". This will write the converted objects to the target database.



19. Another confirmation dialog box will pop up at this point. Click “Yes” to continue.



20. The schema conversion is now complete. You are ready to move on to the next step and use the Database Migration Service to move the data from your Source to your Target.

### **Data Migration:**



Now we will use the Database Migration Service (DMS) to move the data from the Source to the Target. Go to the AWS Console and bring up the Database Migration Service page. To migrate data from source to target we will:

1. Create a source endpoint
2. Create a target endpoint
3. Create a task to move data from the source endpoint to the target endpoint

Click Endpoints on the left hand side of the screen. Then click "Create Endpoint" and the create endpoint page will appear. You will do this twice, once to create a source endpoint and once to create a target endpoint.

## Create endpoint

AWS DMS accesses your data sources and targets using endpoints. A source endpoint allows AWS DMS to read data from a database (on-premise or in the cloud), or from a non-database source such as Amazon S3. A target endpoint allows AWS DMS to write data to a database, or to a non-database target.

We recommend that you choose "Run test" on this page, to verify that your endpoint is valid before using it in an AWS DMS task.

Endpoint type\* ☒ Source ☐ Target ⓘ

Select RDS DB Instance ☐ ⓘ

Endpoint identifier\*  ⓘ

Source engine\*  ⓘ

Server name\*

Port\*  ⓘ

SSL mode\*  ⓘ

User name\*  ⓘ

Password\*  ⓘ

▶ Advanced

▼ 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\*

Replication instance\*  ⓘ

☒ Refresh schemas after successful connection test ⓘ

### Create a source endpoint:

- Select Endpoints from the menu on the left and "click the "create endpoint" button at the top.
- Endpoint Type: source

Note: if you check "select RDS DB Instance" you can select your particular instance from drop down menus and several entries will be populated for you.

- Endpoint identifier: dbsource

- Engine: Oracle
- Server Name: RDS Source endpoint from your account (found in your cloud formation stack outputs)
- Port: 1521
- SSL Mode: none
- User name: dbmaster
- Password: srcadmin123 ( Please note that if you changed the default settings when creating the Cloudformation stack, then you need to enter the password that you have provided)
- Database name: dms\_sample
- *For Oracle* - SID: SRCDB
- VPC: VPC launched by the CloudFormation template
- Replication Instance: choose the instance that was created as part of the stack creation ☐
- Refresh schemas: Yes - check this box

**Test the connection and save it if the test is successful.**

#### **Create a target endpoint:**

- Click the "create endpoint" button at the top
- Endpoint Type: Target

Note: if you check “select RDS DB Instance” you can select your particular instance from drop down menus and several entries will be populated for you.

- Endpoint identifier: targetdb
- Engine: *aurora-postgresql*
- Server name: endpoint of Aurora target instance
- Port: 5432
- SSL Mode: none
- Username: auradmin
- Password: auroradmin123 ( Please note that if you changed the default settings when creating the Cloudformation stack, then you need to enter the password that you have provided)
- For Aurora PostgreSQL - Database name: postgres
- VPC: VPC launched by the CloudFormation template
- Replication Instance: choose the instance that was created as part of the workshop stack

**Test the connection and save it if the test is successful**

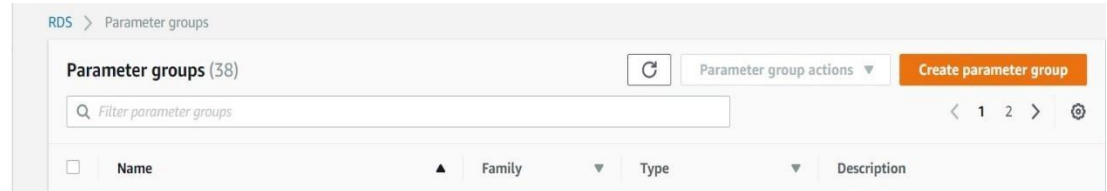
#### **Dealing with Foreign Keys on the target.**

The initial load in DMS is done table by table which means that the target tables cannot have active foreign key constraints. As we are using SCT to convert source objects into target objects, all secondary objects were created as part of the process. This means that we would need to disable all foreign key constraints on the target for the initial full load to be successful..

## Aurora PostgreSQL

One of the ways to do this is to use the `session_replication_role` parameter in PostgreSQL. Triggers also have a state in PostgreSQL (Origin, replica, always or disabled). When the `session_replication_role` parameter is set to `replica`, only triggers of the state `replica` will be active and are fired when called. If not, the triggers remain inactive. So you need to change this parameter on your PostgreSQL instance. Here are the steps to do that.

1. Go to the RDS parameter group page, and select the button to create a parameter group



2. Now select `aurora-postgresql9.6` and give your parameter group a name and description.

### Create parameter group

#### Parameter group details

To create a parameter group, choose a parameter group family, then name and describe your parameter group

Parameter group family  
DB family that this DB parameter group will apply to

aurora-postgresql9.6

Type

DB Parameter Group

Group name  
Identifier for the DB parameter group

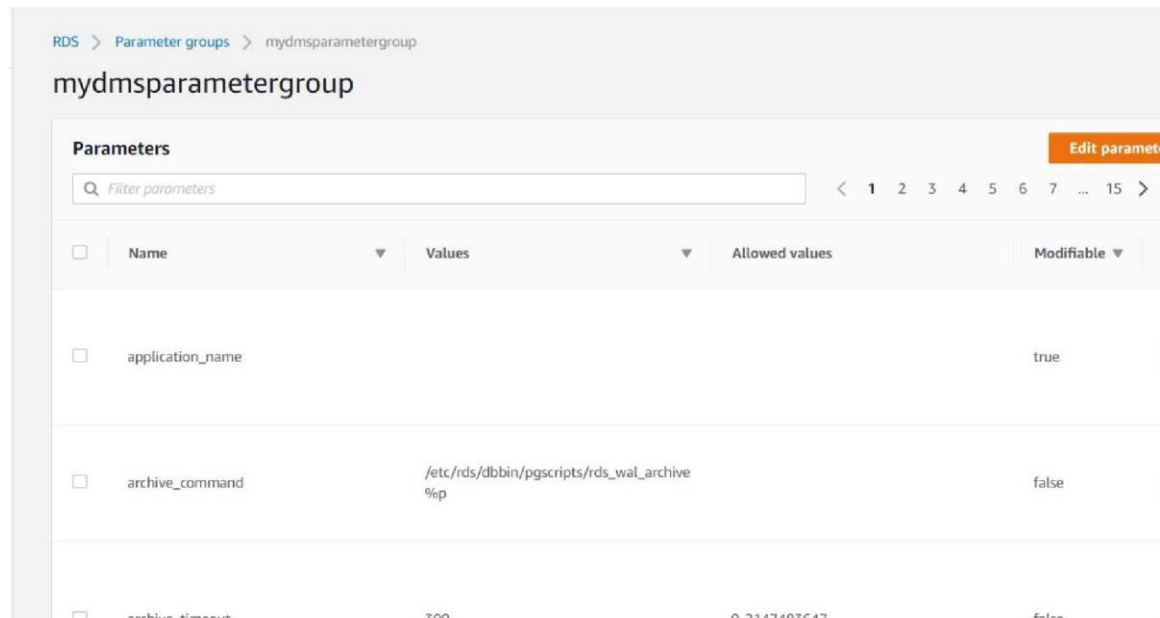
mydmsparametergroup

Description  
Description for the DB parameter group

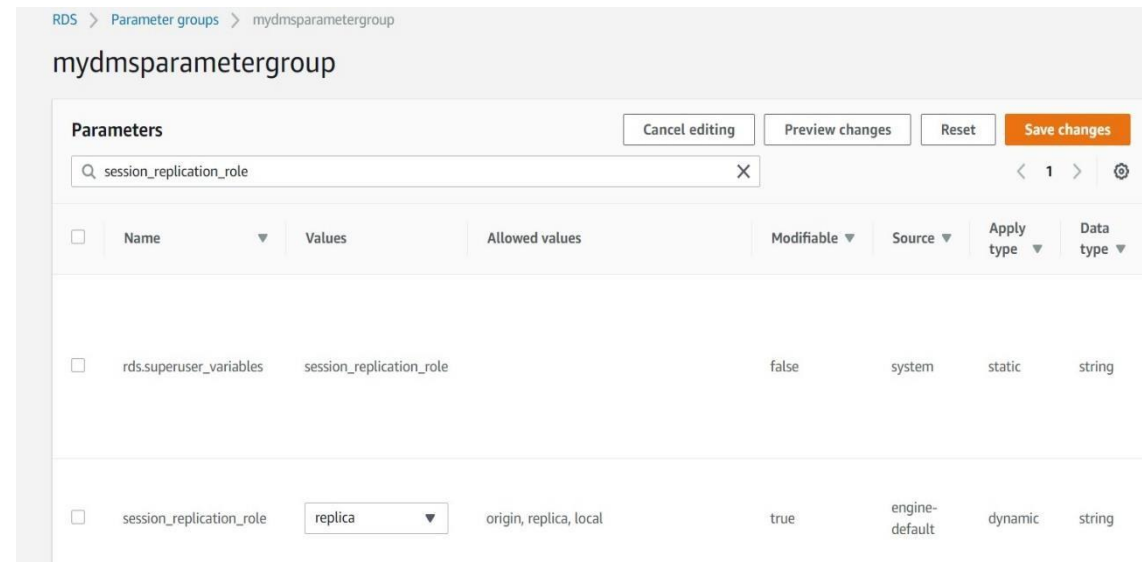
My Group

Cancel Create

3. Now in the parameter group list select your parameter group
4. In the filter parameters box enter `"session_replication_role"`



Now click the “Edit Parameters” button, select the “session\_replication\_role” parameter and change its value to “replica”, then click the “save changes” button.



5. Find your target DB instance under the RDS instance page.
6. Select your target DB instance.
7. Scroll down until you see Details for the instance.
8. Now click on the Modify button

| Details <span>Modify</span>  |   |   |   |
|--|---|---|---|
| <b>Configurations</b><br><br>ARN<br>arn:aws:rds:us-west-2:343299325021:db:dmslab-targetdb<br><br>Engine<br>Aurora PostgreSQL 9.6.3<br><br>Created Time<br>Fri Nov 16 16:36:20 GMT-800 2018<br><br>Username<br>auradmin<br><br>Option Group<br>default:aurora-postgresql-9-6<br><br>Parameter group<br>default:aurora-postgresql9.6 (in-sync)<br><br>DB cluster parameter group<br>default:aurora-postgresql9.6 (in-sync) | <b>Security and network</b><br><br>Availability zone<br>us-west-2a<br><br>VPC<br>dmslab (vpc-0d4a55ec3c2e09510)<br><br>Subnet group<br>dmslab-dbsubnetgroup-ir5sqys880a5<br><br>Subnets<br>subnet-0a13971a3199a7342<br>subnet-0ad2ff921967c5843<br><br>Security groups<br>dmslab-DBSecurityGroup-18SRYA2XIZRBW (sg-0352fc0267dd41f54) ( active )<br><br>Publicly accessible<br>No | <b>Instance and IOPS</b><br><br>Instance Class<br>db.r4.large<br><br>Storage Type<br>DB Cluster<br><br><b>Availability and durability</b><br><br>DB instance status<br>available<br><br>Failover priority<br>1<br><br>Multi AZ<br>No<br><br><b>Backup and Restore</b><br><br>Copy tags to snapshots<br>No | <b>Maintenance details</b><br><br>Auto minor version upgrade<br>Yes<br><br>Maintenance window<br>thu:11:37-thu:12:07 UTC (GMT)<br><br>Pending Modifications<br>None<br><br>Pending maintenance<br>none<br><br><b>Encryption details</b><br><br>Encryption enabled<br>No |

- Look for the Database Options section and change the database parameter group from the default to the one you just created. Then click Continue.

**Database options**

Database port

Specify the TCP/IP port that the DB instance will use for application connections. The connection string of any application connecting to the DB instance must specify the port number of the DB instance. Both the security group applied to the DB instance and your company's firewalls must allow connections to the port. [Learn More](#)

DB parameter group

Database parameter group to associate with this DB instance

DB cluster parameter group

Parameter group associated with this instance's DB Cluster

- On the next page select "Apply immediately" and then click the Modify DB instance button.

### Summary of modifications

You are about to submit the following modifications. Only values that will change are displayed. Carefully verify your changes and click Modify DB Instance.


| Attribute          | Current value                | New value           |
|--------------------|------------------------------|---------------------|
| DB parameter group | default.aurora-postgresql9.6 | mydmsparametergroup |

### Scheduling of modifications

When to apply modifications

☐ Apply during the next scheduled maintenance window  
 Current maintenance window: thu:11:37-thu:12:07

☒ **Apply immediately**  
 The modifications in this request and any pending modifications will be asynchronously applied as soon as possible, regardless of the maintenance window setting for this database instance.

 **Potential unexpected downtime**

If you choose to apply changes immediately, please note that any changes in the pending modifications queue are also applied. If any of the pending modifications require downtime, choosing this option can cause unexpected downtime.

Cancel
Back
Modify DB Instance

- Now you need to select Reboot from the Instance action menu. Once the database reboot completes you can continue with the workshop

### CREATE A MIGRATION TASK:

Next, we need to create a migration task. Select “Tasks” from the menu on the left in your DMS console. Click the “Create task” button in the upper left and fill out the form as follows:

- Name: Give your task a relevant name (dms-workshop or something like that)
- Replication Instance: choose the one you created (see your cloud formation stack outputs)
- Source Endpoint: choose the source endpoint you just created
- Target Endpoint: choose the target endpoint you just created
- Migration Type: Migrate existing data
- Start task on create: check this (or don't, up to you). If you don't, you'll need to start your task separately.

**\*\* Target table preparation mode: Do Nothing** (Important for PostgreSQL targets)



A task can contain one or more table mappings which define what data is moved from the source to the target. If a table does not exist on the target, it will be created.

Task name\*

dms-test-task

Replication instance\*

test-dms-lab-instance - vpc-7cc65715

Source endpoint\*

cf-source

Target endpoint\*

cf-target

Migration type\*

Migrate existing data and replicate on...

Your source database is Oracle. Replicating ongoing changes requires supplemental logging to be turned on.

Please ensure your archive logs are retained on the server for a sufficient amount of time, (24 hours is usually enough.) To set your archivelog retention on RDS databases you can use the following command: `exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours', 24);`

### **Why a do-nothing task?**

Since the initial load in DMS is done table by table which means that the target tables cannot have active foreign key constraints. As we are using SCT to convert source objects into target objects, all secondary objects were created as part of the process. This means that we would need to disable all foreign key constraints on the target for the initial full load to be successful.





One of the ways to do this is to use the `session_replication_role` parameter in PostgreSQL. Triggers also have a state in PostgreSQL (Origin, replica, always or disabled). When the `session_replication_role` parameter is set to replica, only triggers of the state replica will be active and are fired when called. If not, the triggers remain inactive. We have already setup the parameter group on the target to set this role to replica which means all foreign key constraints (innately triggers in the origin state) will not be active. However, PostgreSQL has a failsafe mechanism of not letting a table truncate even with this role set. As we are using prepopulated tables on the target and cannot truncate the table, we are using `do_nothing` for the target table prep mode.

More details in this awesome blog post -

<http://blog.endpoint.com/2015/01/postgressessionreplicationrole.html>

- Stop task after full load completes: don't stop
- Include LOB columns in replication: Accept the default
- Enable validation: up to you
- Enable logging: YES! (Check this box)

▼ Task Settings

|                                      |  |   |
|--------------------------------------|--|---|
| Target table preparation mode*       | <input checked="" type="radio"/> Do nothing<br><input type="radio"/> Drop tables on target<br><input type="radio"/> Truncate   |  |
| Stop task after full load completes* | <input checked="" type="radio"/> Don't stop<br><input type="radio"/> Stop Before Applying Cached Changes<br><input type="radio"/> Stop After Applying Cached Changes |  |
| Include LOB columns in replication*  | <input type="radio"/> Don't include LOB columns<br><input type="radio"/> Full LOB mode<br><input checked="" type="radio"/> Limited LOB mode                          |  |
| Max LOB size (kb)*                   | <input type="text" value="32"/>  |  |
| Enable logging                       | <input checked="" type="checkbox"/>  |   |

**Table Mappings – Selection rules(For Oracle to Aurora PostgreSQL Migrations):**

1. Select DMS\_SAMPLE from the drop-down box Click the "Add selection rule" button.
2. Click on "add transformation rule" -> Choose schema for target dropdown -> Schema name is DMS\_SAMPLE -> Choose "make lower case" for action & click add transformation rule.
3. Click on "add transformation rule" -> Choose table for target dropdown -> Schema name is DMS\_SAMPLE and table name is % -> Choose "make lower case" for action & click add transformation rule.
4. Click on "add transformation rule" -> Choose column for target dropdown -> schema name is DMS\_SAMPLE, table name is % -> column name is % -> Choose "make lower case" for action & click add transformation rule.

▼ Table mappings

**Guided** **JSON**

**Selection rules** ⓘ

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

+ add selection rule

**Transformation rules** ⓘ

For **schema** where **schema name** is like 'DMS\_SAMPLE', **make lowercase**

For **table** where **schema name** is like 'DMS\_SAMPLE' and **table name** is like '%', **make lowercase**

For **column** where **schema name** is like 'DMS\_SAMPLE' and **table name** is like '%' and **column name** is like '%', **make lowercase**

+ add transformation rule

Note – We are doing 2, 3 and 4 above as the schema has been pre-created on the target with lowercase (PostgreSQL convention). Your task json should look like this –

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "DMS_SAMPLE",
        "table-name": "%"
      },
      "rule-action": "include"
    },
    {
      "rule-type": "transformation",
      "rule-id": "2",
      "rule-name": "2",
      "rule-target": "schema",
      "object-locator": {
        "schema-name": "DMS_SAMPLE"
      },
      "rule-action": "convert-lowercase"
    },
    {
      "rule-type": "transformation",
      "rule-id": "3",
      "rule-name": "3",
      "rule-target": "table",
      "object-locator": {
        "schema-name": "DMS_SAMPLE",
        "table-name": "%"
      },
      "rule-action": "convert-lowercase"
    }
  ]
}
```

```

        "rule-type": "transformation",
        "rule-id": "4",
        "rule-name": "4",
        "rule-target": "column",
        "object-locator": {
            "schema-name": "DMS_SAMPLE",
            "table-name": "%",
            "column-name": "%"
        },
        "rule-action": "convert-lowercase"
    }
]
}

```

### **Start Moving Data**

➔ Click create task now and let the task run.

**Note:** Task may fail for 3 tables while it should go well for other tables in the DMS\_SAMPLE (or dbo) schema. As a bonus step you can investigate why this is happening and fix the problem before reloading failed tables.

### **Checking your results:**

Links to downloads for PGAdmin have been added to the windows desktop you used to run the SCT.

### **Clean up CloudFormation stack:**

1. After testing, stop the task and delete the task.
2. Delete the source/target endpoints.
3. Delete CloudFormation stack from the CloudFormation console.