
AWS Database Migration Services

Workshop: Oracle to PostgreSQL Migration

Revised: 2017.10.22

© 2017 Amazon Web Services



This document is © 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. PostgreSQL is a registered trademark of the PostgreSQL Community Association of Canada.

Contents

1 Introduction	7
1.1 Objective	7
1.1.1 Lab Setup	7
1.1.2 Lab Steps	7
1.1.3 Lab Teardown	7
1.2 About AWS Database Migration Service (AWS DMS)	8
1.3 About the AWS Schema Conversion Tool (AWS SCT)	9
2 Lab Setup	10
2.1 Setup EC2 Key Pair	10
2.2 Setup AWS CloudFormation Stack	11
2.2.1 AWS CloudFormation Stack Selection	12
2.2.2 AWS CloudFormation Stack Settings	13
2.2.3 AWS CloudFormation Stack Options	15
2.2.4 AWS CloudFormation Stack Review	17
2.2.5 AWS CloudFormation Stack: List Stacks	17
2.2.6 AWS CloudFormation Stack: Outputs	18
2.3 Setup JDBC Drivers	18
2.4 Setup AWS Schema Conversion Tool	19
2.4.1 AWS SCT Download	19
2.4.2 Download JDBC Drivers	19
2.4.3 Configure AWS SCT with JDBC Drivers	19
2.4.4 Permit Local Access to Source/Target Databases	21
2.5 Setup SQL database management tools (Optional)	22
3 Lab Steps	23
3.1 Create an AWS SCT Project	23
3.1.1 New Project Wizard – Select Source	23
3.1.2 New Project Wizard – Connect to Source Database	24
3.1.3 New Project Wizard - Select Schema	26
3.1.4 New Project Wizard - Run Database Migration Assessment	27
3.1.5 New Project Wizard - Select Target	28
3.1.6 New Project Wizard - Test Target Connection	28
3.2 Schema Conversion in AWS SCT	29
3.2.1 Schema Conversion - Review Report	29
3.2.2 Schema Conversion - Review and Convert Schema	29
3.2.3 Schema Conversion - Convert Source Schema	29
3.2.4 Schema Conversion - Review Target Schema	31

3.2.5	Schema Conversion - Apply Target Schema	32
3.3	AWS Database Migration Services	34
3.3.1	AWS DMS Replication Instances	34
3.3.2	AWS DMS Endpoints	36
3.3.3	AWS DMS Tasks	44
3.4	AWS DMS Troubleshooting	52
3.4.1	AWS DMS Troubleshooting - List Tasks (Errors)	52
3.4.2	AWS DMS Troubleshooting - View Logs	52
3.5	AWS DMS Errors	54
3.5.1	Error: "CANNOT TRUNCATE A TABLE REFERENCED IN A FOREIGN KEY CONSTRAINT"	54
3.6	AWS DMS Errors	59
3.6.1	Error: "SUPPLEMENTAL LOGGING FOR TABLE 'DMS_SAMPLE.MLB_DATA' IS NOT ENABLED PROPERLY"	59
4	Lab Teardown	60
4.1	Teardown AWS CloudFormation Stack	60
4.2	Teardown AWS DMS Resources	60

List of Tables

List of Figures

1	AWS Schema Conversion Tool: Supported Platforms	9
2	Key Pairs: List All Keys	10
3	Key Pair: Create Dialog	11
4	Key Pair: Created	11
5	AWS CloudFormation Stack Selection: Template	12
6	AWS CloudFormation Stack Selection: Confirmation	12
7	AWS CloudFormation: Settings Overview	13
8	AWS CloudFormation: Stack Name	14
9	AWS CloudFormation: Key Pair	14
10	AWS CloudFormation: Oracle	14
11	AWS CloudFormation: Postgres	15
12	AWS CloudFormation Stack Parameters: Confirmation	15
13	AWS CloudFormation Stack Options: All	16
14	AWS CloudFormation Stack Options: Confirmation	16
15	AWS CloudFormation Stack: Review	17
16	AWS CloudFormation Stack Review: Creation	17
17	AWS CloudFormation Stack: List Stacks	18

18	AWS CloudFormation Stack: Outputs	18
19	AWS Schema Conversion Tool: Initialized	20
20	AWS Schema Conversion Tool: Global Settings	20
21	AWS Schema Conversion Tool: List of Security Groups	21
22	AWS SCT New Project Wizard	23
23	AWS SCT New Project Wizard - Step 1	24
24	AWS SCT New Project Wizard - Test Source Connection	25
25	AWS SCT New Project Wizard - Select Schema	26
26	AWS SCT New Project Wizard - Database Migration Assessment	27
27	AWS SCT New Project Wizard - Select Target	28
28	AWS SCT New Project Wizard - Test Target Connection	28
29	Schema Conversion - Review Conversion Report	29
30	Schema Conversion - Source Apply (via right-click)	30
31	Schema Conversion - Source Apply (via Actions)	31
32	Schema Conversion - Target Review	32
33	Schema Conversion - Target Apply (via right-click)	33
34	Schema Conversion - Target Apply (via Actions)	33
35	AWS DMS Replication Instances - List All	34
36	AWS DMS Replication Instances - Create New	35
37	AWS DMS Replication Instances - List New Creating	36
38	AWS DMS Replication Instances - List New Available	36
39	AWS DMS Endpoints - List All	37
40	AWS DMS Endpoints - Create Endpoint	38
41	AWS DMS Endpoints - Create Source Endpoint (Details)	39
42	AWS DMS Endpoints - Create Source Endpoint (Test)	40
43	AWS DMS Endpoints - Create Source Endpoint (Test Success)	41
44	AWS DMS Endpoints - List All (Updated)	41
45	AWS DMS Endpoints - Create Target Endpoint (Details)	42
46	AWS DMS Endpoints - Create Target Endpoint (Test)	43
47	AWS DMS Endpoints - Create Target Endpoint (Test Success)	43
48	AWS DMS Endpoints - List Endpoints (Source and Target)	44
49	AWS DMS Tasks - Create Task	44
50	AWS DMS Tasks - Create Task	45
51	AWS DMS Tasks - Create Task (Continued)	46
52	AWS DMS Tasks - Create Task (Table Mappings Selection)	47
53	AWS DMS Tasks - Create Task (Table Mappings Transform #1)	48
54	AWS DMS Tasks - Create Task (Table Mappings Transform #2)	49
55	AWS DMS Tasks - Create Task (Table Mappings Transform #3)	50
56	AWS DMS Tasks - Create Task (Final)	51

57	AWS DMS Tasks - List Tasks (Updated)	51
58	AWS DMS Troubleshooting - List Tasks (Errors)	52
59	AWS DMS Troubleshooting - View Log (Link)	52
60	AWS DMS Troubleshooting - View CloudWatch Log (All)	53
61	AWS DMS Troubleshooting - View CloudWatch Log (Errors Only)	53
62	AWS DMS Errors - Truncate	54
63	AWS DMS Errors - Delete Log Stream	55
64	AWS DMS Errors - Stop DMS Task	55
65	AWS DMS Errors - Modify DMS Task	56
66	AWS DMS Errors - Change Table Prep (Before)	57
67	AWS DMS Errors - Change Table Prep (After)	57
68	AWS DMS Errors - Restart Task	58
69	AWS DMS Errors - CloudWatch Log Filtered for PERSON table	59
70	AWS DMS Errors - Supplemental Logging	59

1 Introduction

1.1 Objective

In this lab, you will be performing a migration from Oracle to PostgreSQL using the AWS Schema Conversion Tool (AWS SCT) and the AWS Databases Migration Service (AWS DMS).

1.1.1 Lab Setup

- Create EC2 Key Pair
- Launch AWS CloudFormation Stack
- Access Amazon AppStream 2.0 Tools
- OPTIONAL: download the JDBC drivers locally
- OPTIONAL: install the AWS Schema Conversion Tool locally
- OPTIONAL: install database management tools locally

1.1.2 Lab Steps

- Create AWS Database Migration Instances
- Connect to your environment
- Setup AWS Schema Conversion Tool
- Convert the Oracle schema to PostgreSQL
- Create Source Endpoint in AWS DMS
- Create Target Endpoint in AWS DMS
- Create a Migration Task in AWS DMS
- Start the migration
- Generate transactions on Oracle and see the data being migrated to PostgreSQL - CDC

1.1.3 Lab Teardown

- Delete AWS CloudFormation Stack
- Delete EC2 Key Pair

1.2 About AWS Database Migration Service (AWS DMS)

The AWS Database Migration Service helps you migrate databases to AWS quickly and securely. The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database. The AWS Database Migration Service can migrate your data to and from most widely used commercial and open-source databases.

The service supports homogenous migrations such as Oracle to Oracle, as well as heterogeneous migrations between different database platforms, such as Oracle to Amazon Aurora or Microsoft SQL Server to MySQL. It also allows you to stream data to Amazon Redshift, Amazon DynamoDB, and Amazon S3 from any of the supported sources including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, SAP ASE, SQL Server and MongoDB, enabling consolidation and easy analysis of data in the petabyte-scale data warehouse. AWS Database Migration Service can also be used for continuous data replication with high-availability.

To find out more about AWS DMS, see the product page at <https://aws.amazon.com/dms/>

1.3 About the AWS Schema Conversion Tool (AWS SCT)

The AWS Schema Conversion Tool makes heterogeneous database migrations predictable by automatically converting the source database schema and a majority of the database code objects, including views, stored procedures, and functions, to a format compatible with the target database. Any objects that cannot be automatically converted are clearly marked so that they can be manually converted to complete the migration. SCT can also scan your application source code for embedded SQL statements and convert them as part of a database schema conversion project. During this process, SCT performs cloud native code optimization by converting legacy Oracle and SQL Server functions to their equivalent AWS service thus helping you modernize the applications at the same time of database migration. Once schema conversion is complete, SCT can help migrate data from a range of data warehouses to Amazon Redshift using built-in data migration agents.

Your source database can be on-premises, or in Amazon RDS or EC2 and the target database can be in either Amazon RDS or EC2. The AWS Schema Conversion Tool supports the following conversions:

Source Database	Target Database on Amazon RDS
Oracle Database	Amazon Aurora, MySQL, PostgreSQL, MariaDB
Oracle Data Warehouse	Amazon Redshift
Microsoft SQL Server	Amazon Aurora, Amazon Redshift, MySQL, PostgreSQL, MariaDB
Teradata	Amazon Redshift
IBM Netezza	Amazon Redshift
Greenplum	Amazon Redshift
HPE Vertica	Amazon Redshift
MySQL and MariaDB	PostgreSQL
PostgreSQL	Amazon Aurora, MySQL, MariaDB
Amazon Aurora	PostgreSQL

Figure 1: AWS Schema Conversion Tool: Supported Platforms

2 Lab Setup

2.1 Setup EC2 Key Pair

In this step, you will generate an EC2 key pair for use in the Database Migration Workshop labs.

Make sure you have selected the **Asia Pacific (Tokyo)** region by visiting the following link:

<http://amzn.to/aws-tokyo-keypairs> (=> <https://ap-northeast-1.console.aws.amazon.com/ec2/v2/home?region=ap-northeast-1#KeyPairs:sort=keyName>)

If no EC2 key pairs have been created in this region yet, you will see the following screen:

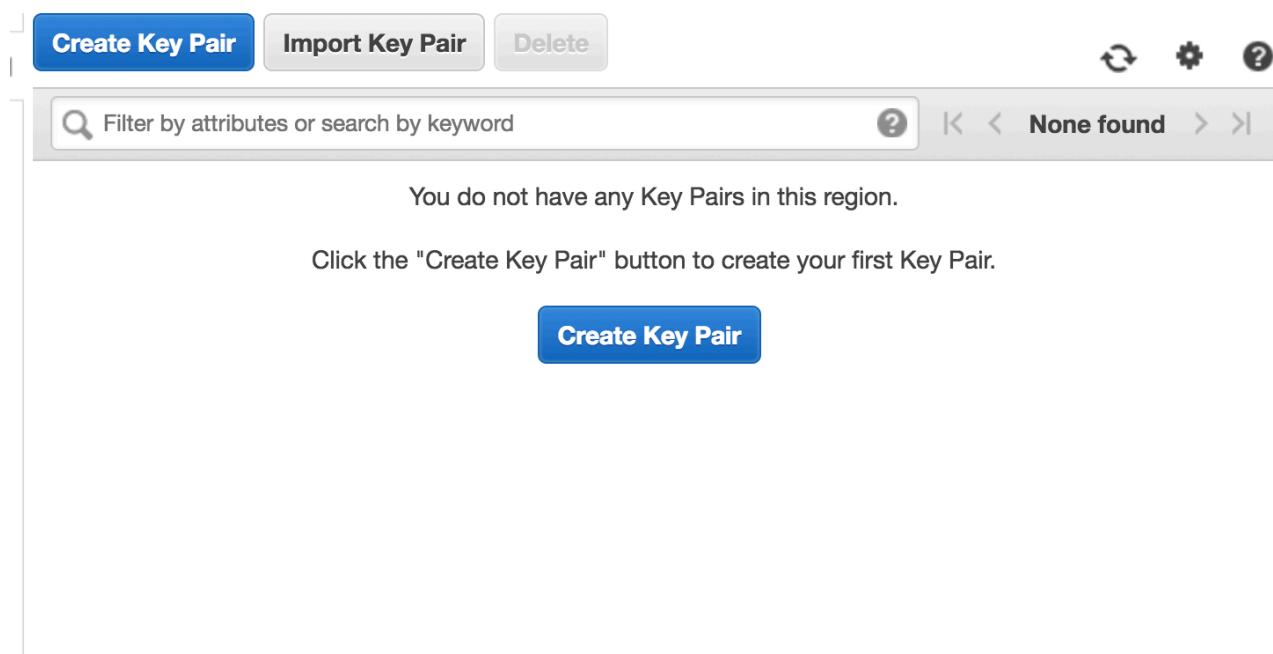


Figure 2: Key Pairs: List All Keys

For this workshop, we recommend the creation of a new EC2 key pair to be used associated with workshop resources. We will later delete this key pair along with the workshop resources.

You should click the **Create Key Pair** button and enter **workshop** as the name for the new key pair as shown below, clicking the **Create** button to complete the creation of the new key pair.

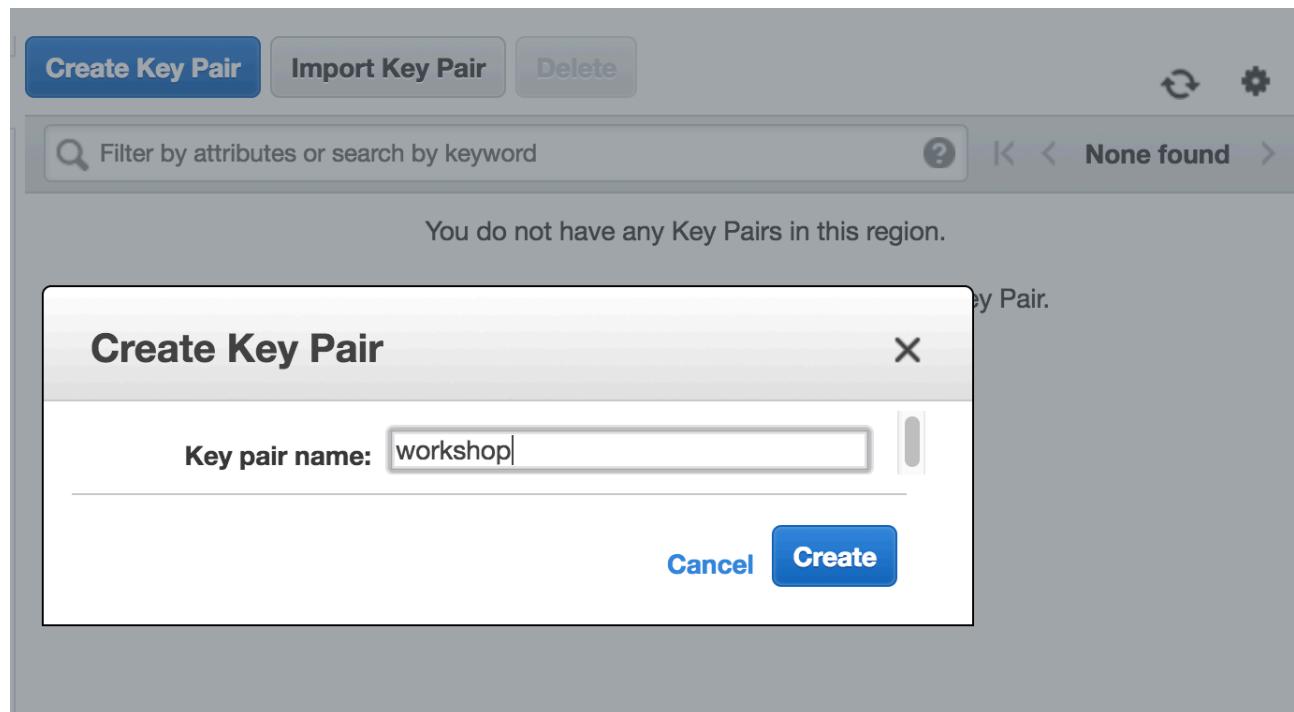


Figure 3: Key Pair: Create Dialog

You should now see a new EC2 key pair labeled **workshop**:

		Create Key Pair	Import Key Pair	Delete	Refresh	Settings	Help
		Filter by attributes or search by keyword		?	<	1 to 1 of 1	> >
		Key pair name	Fingerprint				
<input checked="" type="checkbox"/>	workshop		cc:e3:5d:70:b2:47:56:ae:17:a0:44:5d:c6:30:d3:ce:cb:02:6a:a3				

Figure 4: Key Pair: Created

The private key for this EC2 key pair will have automatically downloaded to your browser's default directory, and this file will be called **workshop.pem**.

2.2 Setup AWS CloudFormation Stack

In this step, you will launch a AWS CloudFormation template that will setup the following resources needed for this lab:

- Source Database: Amazon RDS Oracle (this database will be pre-populated with sample database installed from <https://github.com/awslabs/aws-database-migration-samples>)
- Target Database: Amazon RDS PostgreSQL

CAUTION: The resources created automatically with this CloudFormation template will continue to run until the CloudFormation stack is deleted or the individual resources are shutdown -- the steps for teardown are located in this document at [Teardown: AWS Cloudformation Stack](#)

To launch this template, use the following link:

<http://amzn.to/aws-dms-workshop-lab-1> (=> <https://ap-northeast-1.console.aws.amazon.com/cloudformation/home?region=ap-northeast-1#/stacks/new?stackName=workshop&templateURL=https://s3-ap-northeast-1.amazonaws.com/aws-dms-workshop/workshop-lab-1.yaml>)

2.2.1 AWS CloudFormation Stack Selection

You should now see the following:

Create stack

Select Template [Select Template](#)

Specify Details [Options](#) [Review](#)

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
 Specify an Amazon S3 template URL
<https://s3-ap-northeast-1.amazonaws.com/aws-dms-works/> [View/Edit template in Designer](#)

[Cancel](#) [Next](#)

Figure 5: AWS CloudFormation Stack Selection: Template

The default settings are recommended, and no changes are required to this page. Click the Next button on the bottom of the page to continue.

[Cancel](#) [Previous](#) [Next](#)

Figure 6: AWS CloudFormation Stack Selection: Confirmation

2.2.2 AWS CloudFormation Stack Settings

This page displays the settings and parameters for the CloudFormation stack. **The default settings are recommended, and no changes are required to this page.**

Create stack

Select Template **Specify Details**

Options Review

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

Key Pair

KeyName Name of an existing EC2 KeyPair to enable SSH access to the instance

Source Oracle Database Configuration

OracleDBName Enter Oracle Database name

OracleDBPassword Enter password for the oracle admin user: dbmaster

OracleDBStorage Enter storage for Oracle DB in GB

OracleInstanceType Oracle DB instance type

Target PostgreSQL Database Configuration

PostgresDBName Enter PostgreSQL Database name

PostgresDBUsername Enter database Admin username for RDS PostgreSQL

PostgresDBPassword Enter password for RDS PostgreSQL Admin user

PostgresInstanceType RDS PostgreSQL DB instance type

PostgresDBStorage Enter storage for PostgreSQL DB in GB

Figure 7: AWS CloudFormation: Settings Overview

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

Figure 8: AWS CloudFormation: Stack Name

Parameters

Key Pair

KeyName ▼
Name of an existing EC2 KeyPair to enable SSH access to the instance

Figure 9: AWS CloudFormation: Key Pair

Parameters

Source Oracle Database Configuration

OracleDBName	<input type="text" value="ORCL"/>	Enter Oracle Database name
OracleDBPassword	<input type="password" value="....."/>	Enter password for the oracle admin user: dbmaster
OracleDBStorage	<input type="text" value="100"/>	Enter storage for Oracle DB in GB
OracleInstanceType	<input type="text" value="db.t2.medium"/> ▼	Oracle DB instance type

Figure 10: AWS CloudFormation: Oracle

Target PostgreSQL Database Configuration

PostgresDBName	<input type="text" value="postgres"/>	Enter PostgreSQL Database name
PostgresDBUsername	<input type="text" value="postadmin"/>	Enter database Admin username for RDS PostgreSQL
PostgresDBPassword	<input type="password" value="....."/>	Enter password for RDS PostgreSQL Admin user
PostgresInstanceType	<input type="text" value="db.t2.medium"/>	RDS PostgreSQL DB instance type
PostgresDBStorage	<input type="text" value="100"/>	Enter storage for PostgreSQL DB in GB

Figure 11: AWS CloudFormation: Postgres

Once you have confirmed the settings, click the **Next** button on the bottom of the page to continue.

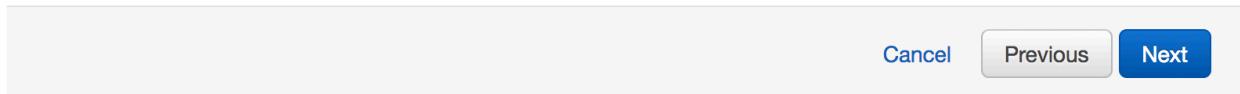


Figure 12: AWS CloudFormation Stack Parameters: Confirmation

2.2.3 AWS CloudFormation Stack Options

You will now see the Options for this CloudFormation template. **The default settings are recommended, and no changes are required to this page.**

Create stack

Select Template
Specify Details
Options
Review

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	<input type="text"/>	<input type="text"/>	+

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 [Choose a role \(optional\)](#)
Enter role arn

► Advanced

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

[Cancel](#) [Previous](#) **Next**

Figure 13: AWS CloudFormation Stack Options: All

Click the **Next** button on the bottom of the page to continue.

[Cancel](#) [Previous](#) **Next**

Figure 14: AWS CloudFormation Stack Options: Confirmation

2.2.4 AWS CloudFormation Stack Review

Create stack

Select Template
Specify Details
Options
Review

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	<input type="text"/>	<input type="text"/>	+

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

► Advanced

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

[Cancel](#) [Previous](#) **Next**

Figure 15: AWS CloudFormation Stack: Review

Click the **Create** button on the bottom of the page to continue.

[Cancel](#) [Previous](#) **Create**

Figure 16: AWS CloudFormation Stack Review: Creation

AWS resources associated with this workshop lab will now be automatically created immediately, and billing will proceed until this CloudFormation stack is deleted or those resources are individually stopped/terminated -- the steps for teardown are located in this document at [Teardown: AWS Cloudformation Stack](#)

2.2.5 AWS CloudFormation Stack: List Stacks

You should now see a list of any existing CloudFormation templates for this region in your account, and the new CloudFormation stack called **workshop** should now appear in your console.

Stack Name				Created Time	Status	Description
<input checked="" type="checkbox"/>	workshop			2017-10-22 12:12:32 UTC+0800	CREATE_COMPLETE	This CloudFormation te...

Figure 17: AWS CloudFormation Stack: List Stacks

If the stack does not immediately appear, you may need to click the **Refresh** button above the list panel.

2.2.6 AWS CloudFormation Stack: Outputs

When the stack and its underlying resource creations are completed, the status will be displayed as **CREATE_COMPLETE**. Once this has occurred, we can gather Outputs from the CloudFormation stack.

Key	Value	Description	Export Name
OracleJDBCConnectionString	jdbc:oracle:thin:@wo1mo68832b8er8.cnlavnyl m1rz.ap-northeast-1.rds.amazonaws.com:152 1:ORCL	JDBC connection string for Oracle database	
Regionname	ap-northeast-1		
PostgresJDBCConnectionString	jdbc:postgresql://wpofi3ugd73u6p.cnlavnylm1 rz.ap-northeast-1.rds.amazonaws.com:5432/p ostgres	JDBC connection string for PostgreSQL dat...	
StackName	workshop		

Figure 18: AWS CloudFormation Stack: Outputs

2.3 Setup JDBC Drivers

If you have chosen to locally install the AWS Schema Conversion Tool and your preferred database management tool, you will also need to download the database-specific JDBC drivers to connect to the database resources in this workshop. These JDBC drivers can be found at the following links:

- <http://bit.ly/postgres-jdbc>
- <http://bit.ly/oracle-jdbc>

*The Oracle JDBC requires that you accept the associated license agreement before downloading -- the actual file required from the linked page is called **ojdbc7.jar***

You will need to download both drivers to your local hard drive for later use by both the AWS Schema Conversion Tool and your preferred database management tool.

2.4 Setup AWS Schema Conversion Tool

In this step, you will install the AWS Schema Conversion Tool locally.

For those participants not wishing to install the AWS Schema Conversion Tool locally, you can use the Schema Conversion Tool via Amazon AppStream 2.0 resources that will have been temporarily provisioned for your use during this lab. See [Setup: Accessing Amazon AppStream 2.0 Tools](#)

2.4.1 AWS SCT Download

Download the latest version of AWS Schema Conversion Tool (SCT) from the following link:

<http://amzn.to/aws-sct> (=> http://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/CHAP_SchemaConversionTool.Installing.html)

If you already have SCT installed, we recommend that you download and install the latest version

2.4.2 Download JDBC Drivers

For connecting to your source database (Oracle) and target database (PostgreSQL), you will need the appropriate JDBC drivers for both databases. If you have not already done so, download the [JDBC Drivers](#) before moving forward.

2.4.3 Configure AWS SCT with JDBC Drivers

Once downloaded, launch AWS Schema Conversion Tool. On first launch, you will be presented with a terms and conditions statement, click **Agree** if you agree to the terms and conditions specified.

Next, you should see the following page:

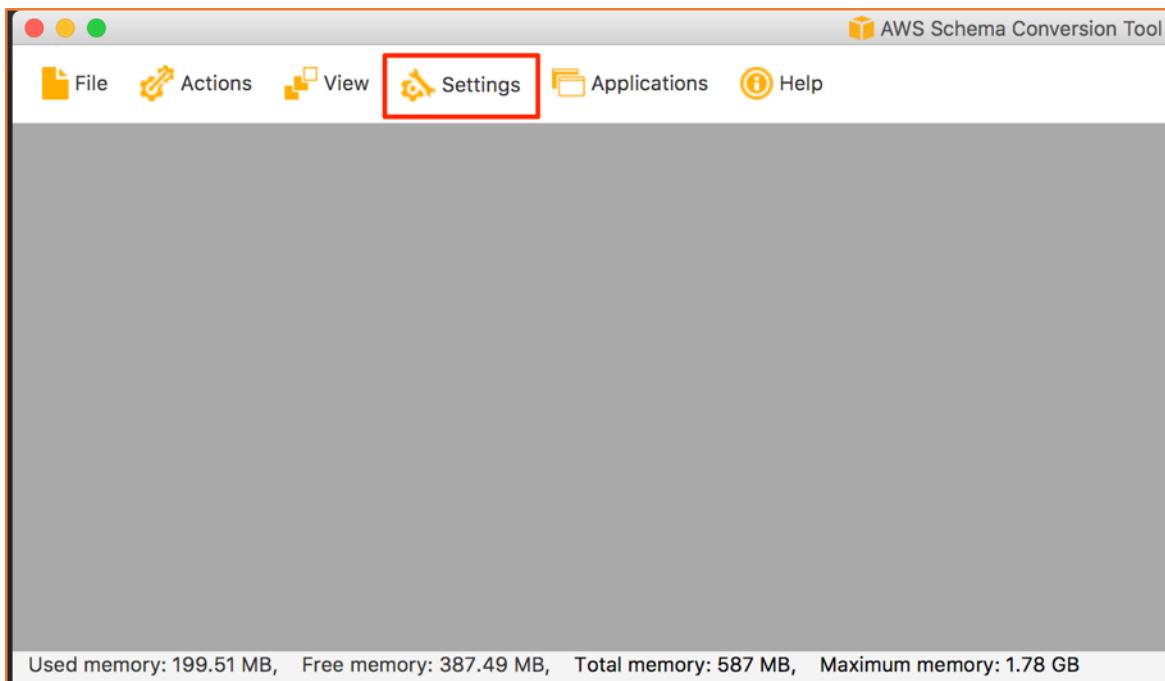


Figure 19: AWS Schema Conversion Tool: Initialized

Click on **Settings > Global Settings**

Now you should see the following Global Settings dialog:

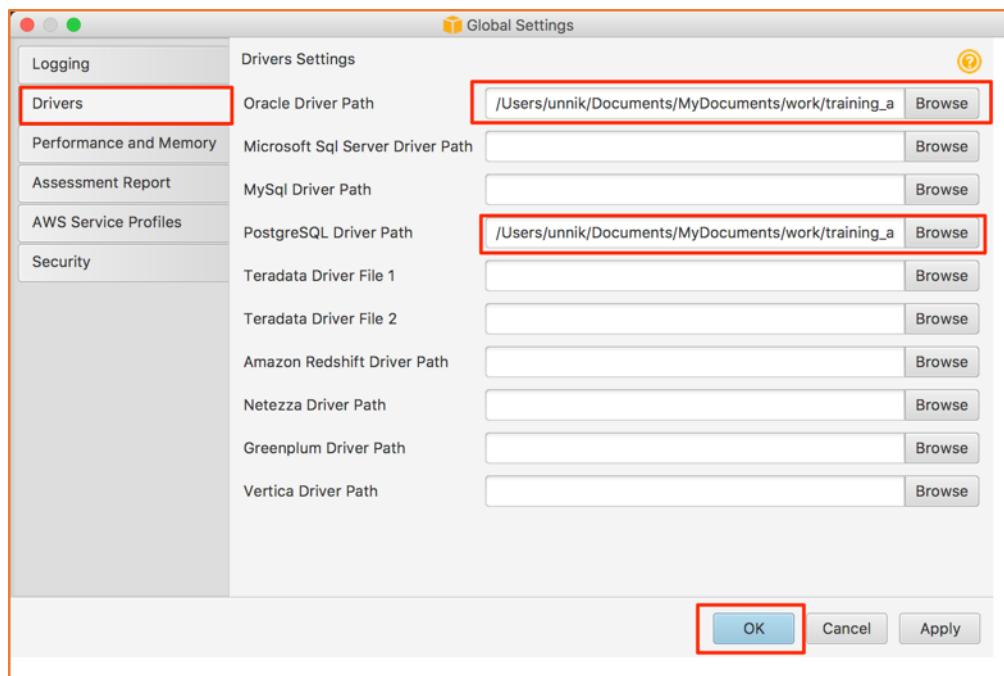


Figure 20: AWS Schema Conversion Tool: Global Settings

Make the following changes to the Global Settings:

- Select **Drivers** on the left-side panel
- For the **Oracle Driver Path**, select the location of your local Oracle jar file
- For the **PostgreSQL Driver Path**, select the location of your local PostgreSQL jar file
- Click **OK** to Proceed

TODO: remove the following as permissions are completely open now

2.4.4 Permit Local Access to Source/Target Databases

You will now modify both Source and Target database permissions so that you can connect to those databases with the AWS SCT locally.

To do so, you will modify the Security Groups attached to the databases. The following link will provide you access to the list of Security Groups in the region, filtered by the default CloudFormation stack name of **workshop**:

<http://amzn.to/aws-tokyo-sg-workshop> (=> <https://ap-northeast-1.console.aws.amazon.com/ec2/v2/home?region=ap-northeast-1#SecurityGroups:search=workshop;sort=groupId>)

The list should look similar to the following, with two different SGs created by the earlier CloudFormation template, one for Oracle and the other for PostgreSQL:

The screenshot shows a table listing two security groups. The columns are Name, Group ID, Group Name, VPC ID, and Description. The first row has a Group ID of sg-ad303ccb, a Group Name of workshop-OraVPCSecurityG..., a VPC ID of vpc-76d2cf12, and a Description of 'Security group for Oracle Instance.'. The second row has a Group ID of sg-d5323eb3, a Group Name of workshop-PostgresVPCSec..., a VPC ID of vpc-76d2cf12, and a Description of 'Security group for RDS PostgreSQL Instances.'

Name	Group ID	Group Name	VPC ID	Description
	sg-ad303ccb	workshop-OraVPCSecurityG...	vpc-76d2cf12	Security group for Oracle Instance.
	sg-d5323eb3	workshop-PostgresVPCSec...	vpc-76d2cf12	Security group for RDS PostgreSQL Instances.

Figure 21: AWS Schema Conversion Tool: List of Security Groups

Open the following ports to provide access from your current IP address.

- Modify **workshop-oracle-sg** Security Group as follows:
 - Add rule Oracle Port – 1521 > Open to 'My IP'
- Modify **workshop-postgres-sg** Security Group as follows:
 - Add rule Postgres Port – 5432 > Open to 'My IP'

2.5 Setup SQL database management tools (Optional)

In this step, we will install a database management tool of your choice locally.

For those participants not wishing to install a database management tool locally, you can use the DBeaver tool via Amazon AppStream 2.0 resources that will have been provisioned for you temporarily. See [Setup: Accessing Amazon AppStream 2.0 Tools](#)

Some popular and free tools include:

- SQL WorkbenchJ: <http://www.sql-workbench.net/downloads.html>
- DBeaver: <http://dbeaver.jkiss.org/>
- SQuirrel: <http://squirrel-sql.sourceforge.net/>

In this lab, we will be using SQL WorkbenchJ screenshots to demonstrate the lab activities.

3 Lab Steps

3.1 Create an AWS SCT Project

In AWS SCT, select **File > New Project Wizard**

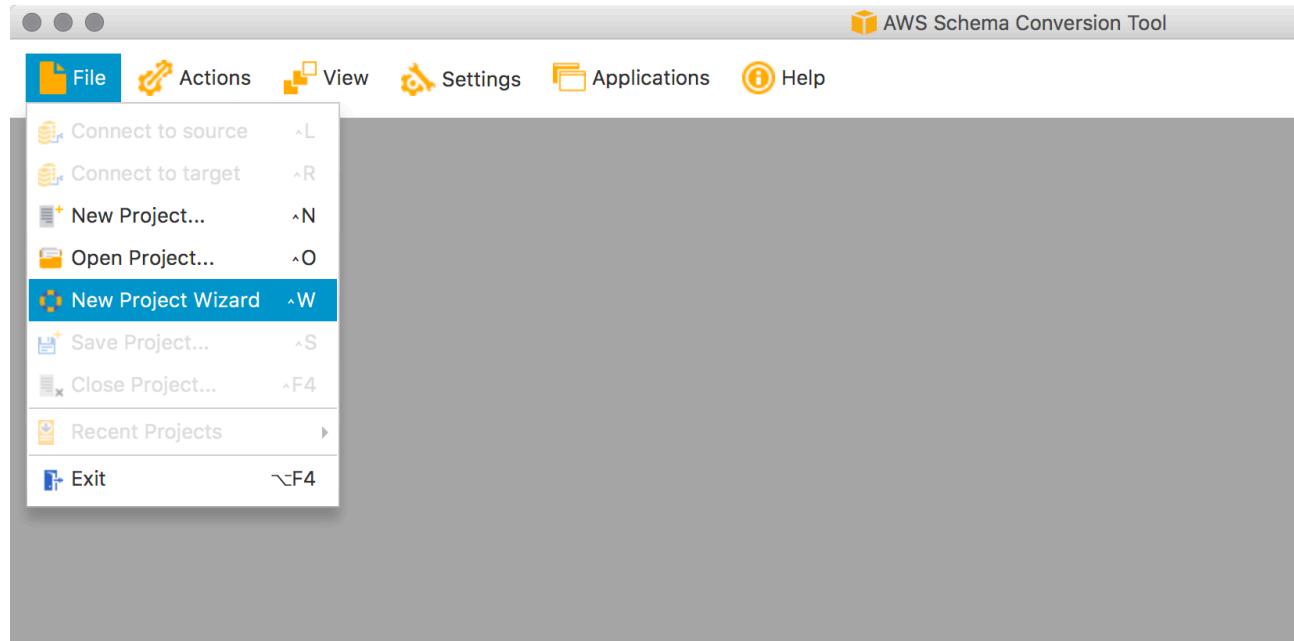


Figure 22: AWS SCT New Project Wizard

3.1.1 New Project Wizard – Select Source

Enter the following in the dialog displayed:

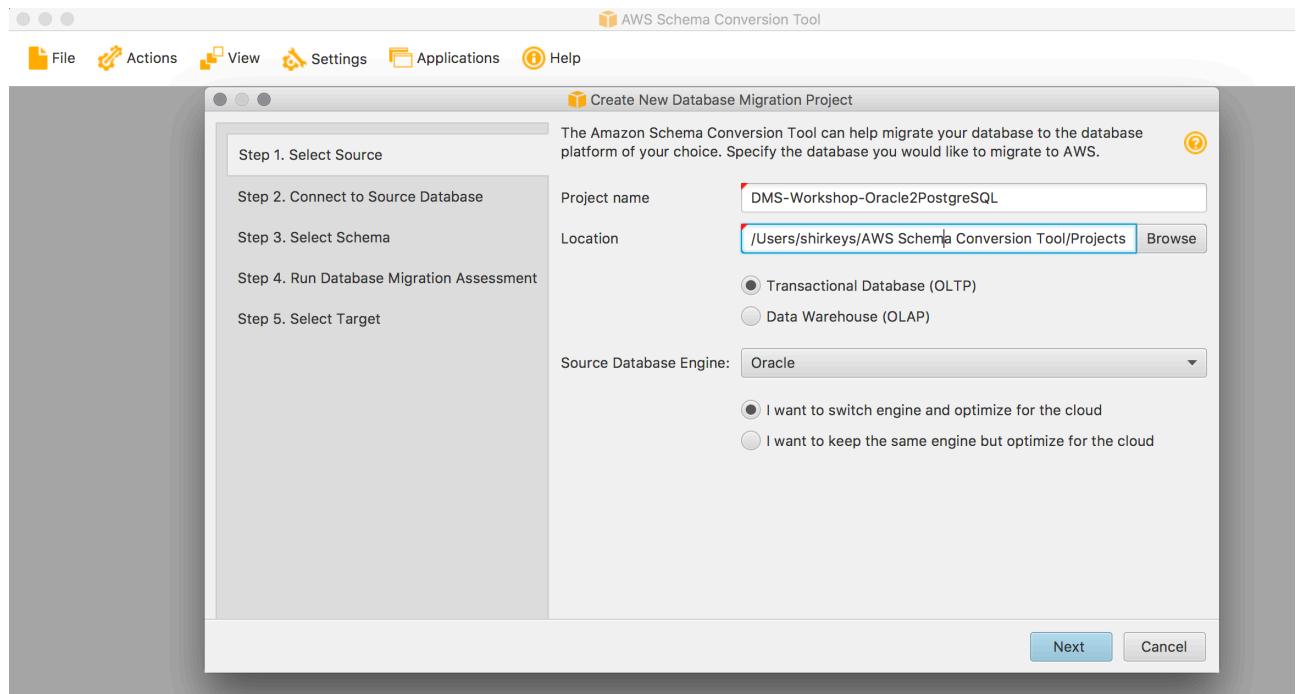


Figure 23: AWS SCT New Project Wizard - Step 1

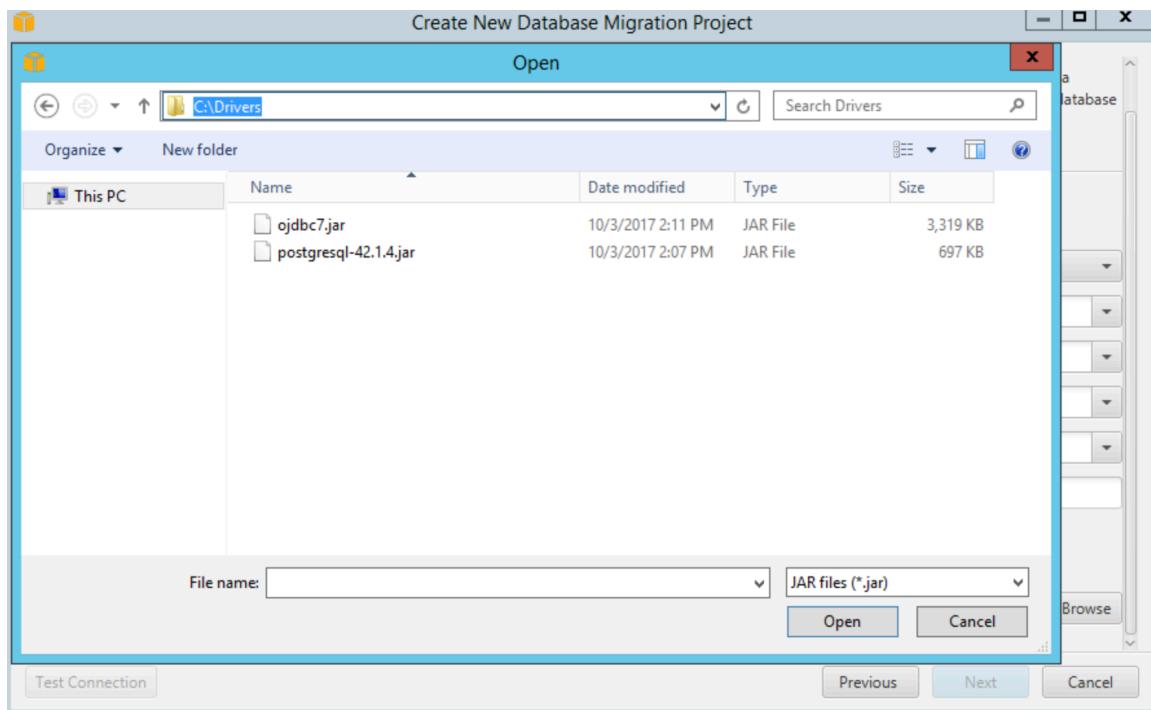
- **Project Name** | DMS-Workshop-Oracle2PostgreSQL|
- **Location:** (leave default)
- For **Source Database Engine**, specify:
 - Transactional Database (OLTP)
 - Oracle
 - I want to switch engine and optimize for the cloud

Click **Next** to continue

3.1.2 New Project Wizard – Connect to Source Database

Enter the following in the next dialog displayed:

- **Type:** SID
- **Server Name:** (enter DNS name of your Oracle instance)
- **Server Port:** 1521
- **Oracle SID:** (enter database name, default is Oracle)
- **User name:** dbmaster
- **Password:** (enter the password for your Oracle instance)
- **Oracle Driver Path:** (see notes below)
 - Local: when running AWS SCT locally, then the JDBC jar file for Oracle must be selected from where it was downloaded to earlier
 - DMS Workshop AppStream 2.0 Clients: the path is **C:\Drivers** as shown below:



When finished, click **Test Connection**. If you receive a **Connection Successful** message, then proceed, otherwise reconfirm the values you have entered earlier and try again.

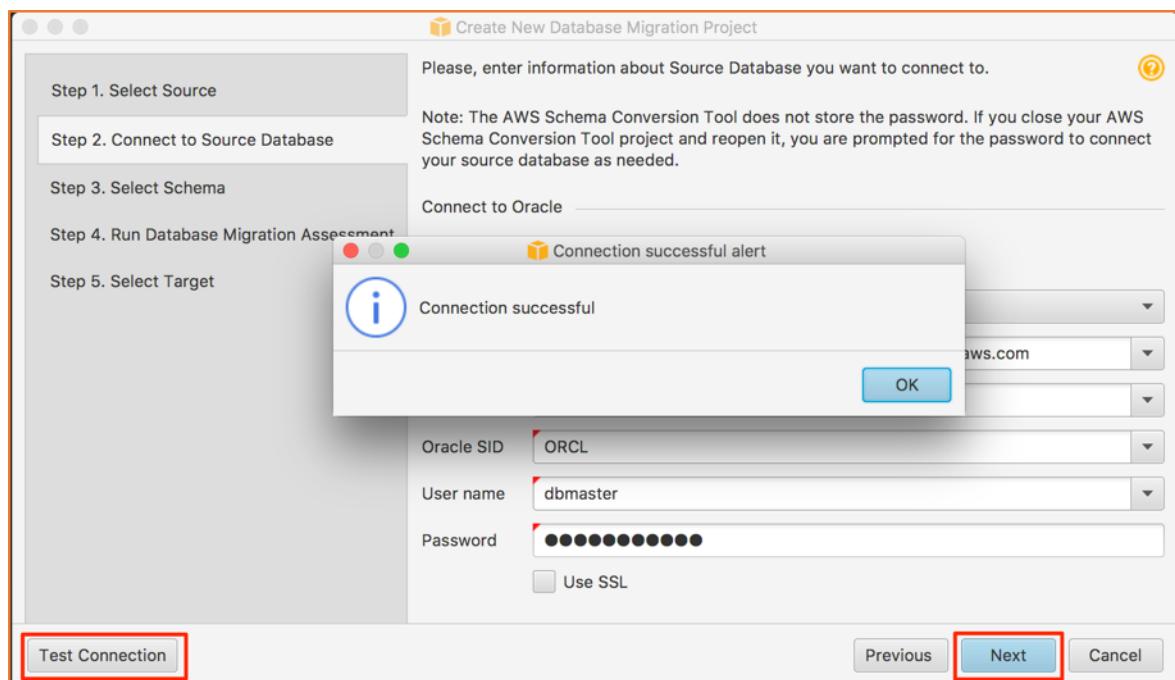


Figure 24: AWS SCT New Project Wizard - Test Source Connection

3.1.3 New Project Wizard - Select Schema

From the Select Schema step, select **DBMASTER** as the Source Schema

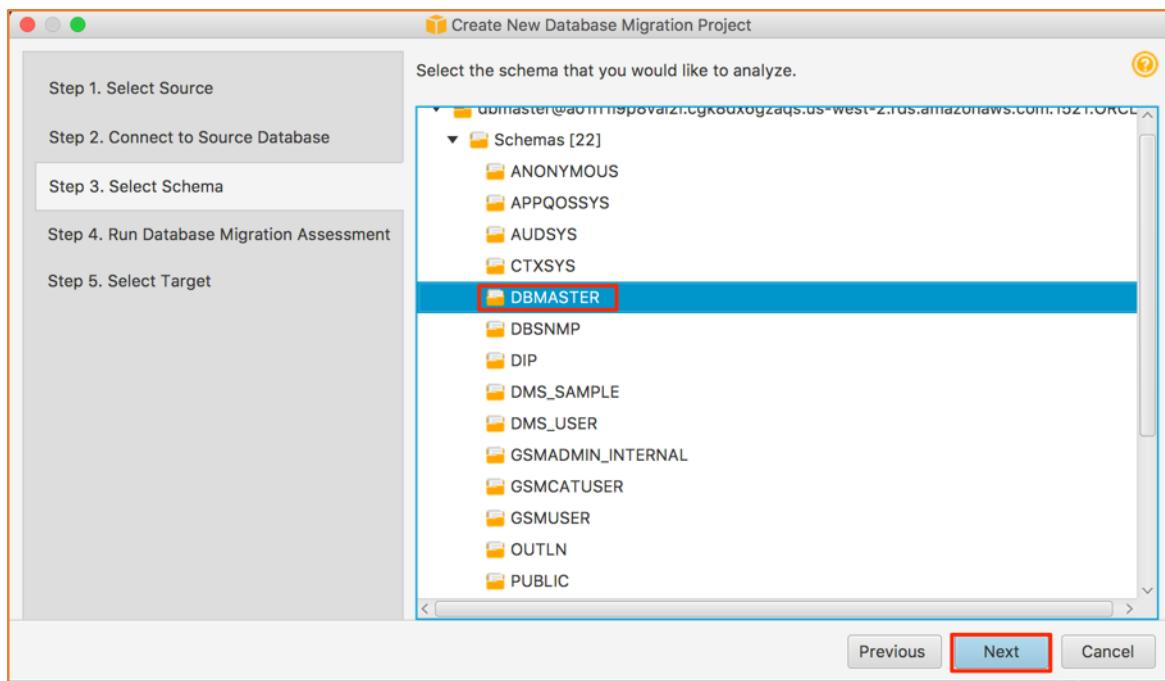


Figure 25: AWS SCT New Project Wizard - Select Schema

3.1.4 New Project Wizard - Run Database Migration Assessment

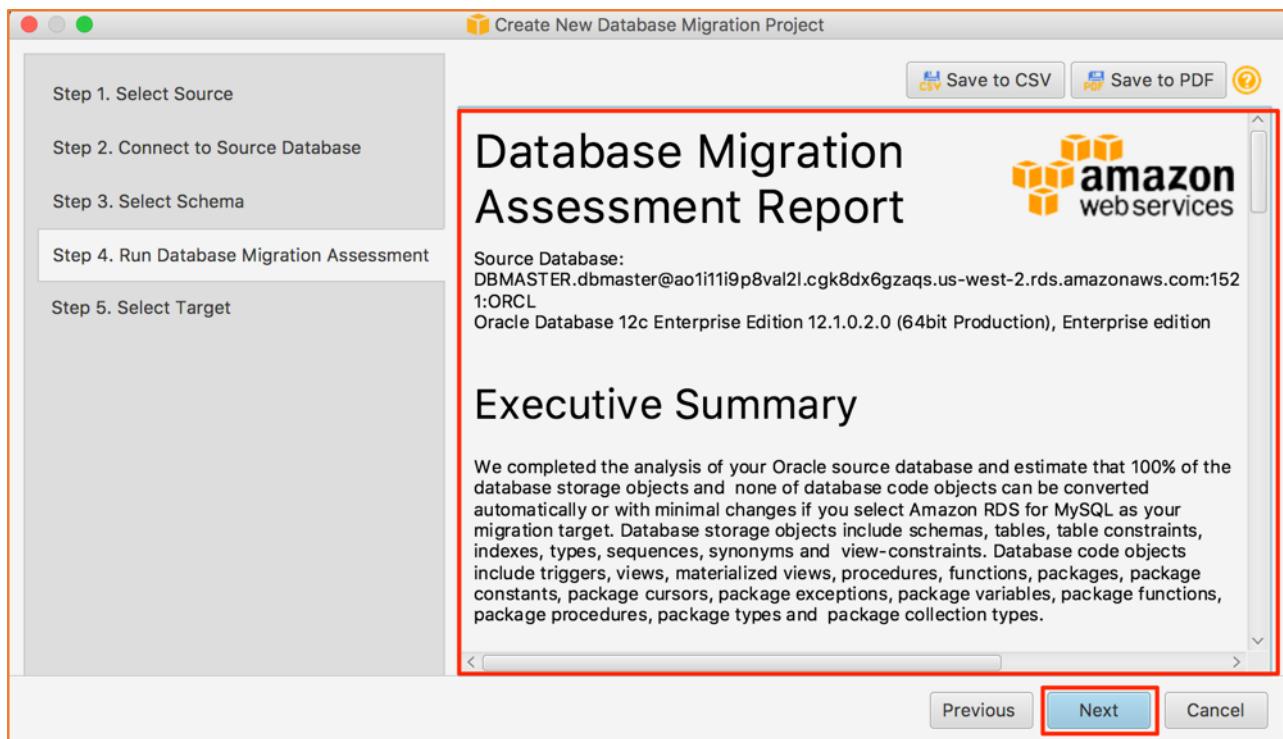


Figure 26: AWS SCT New Project Wizard - Database Migration Assessment

3.1.5 New Project Wizard - Select Target

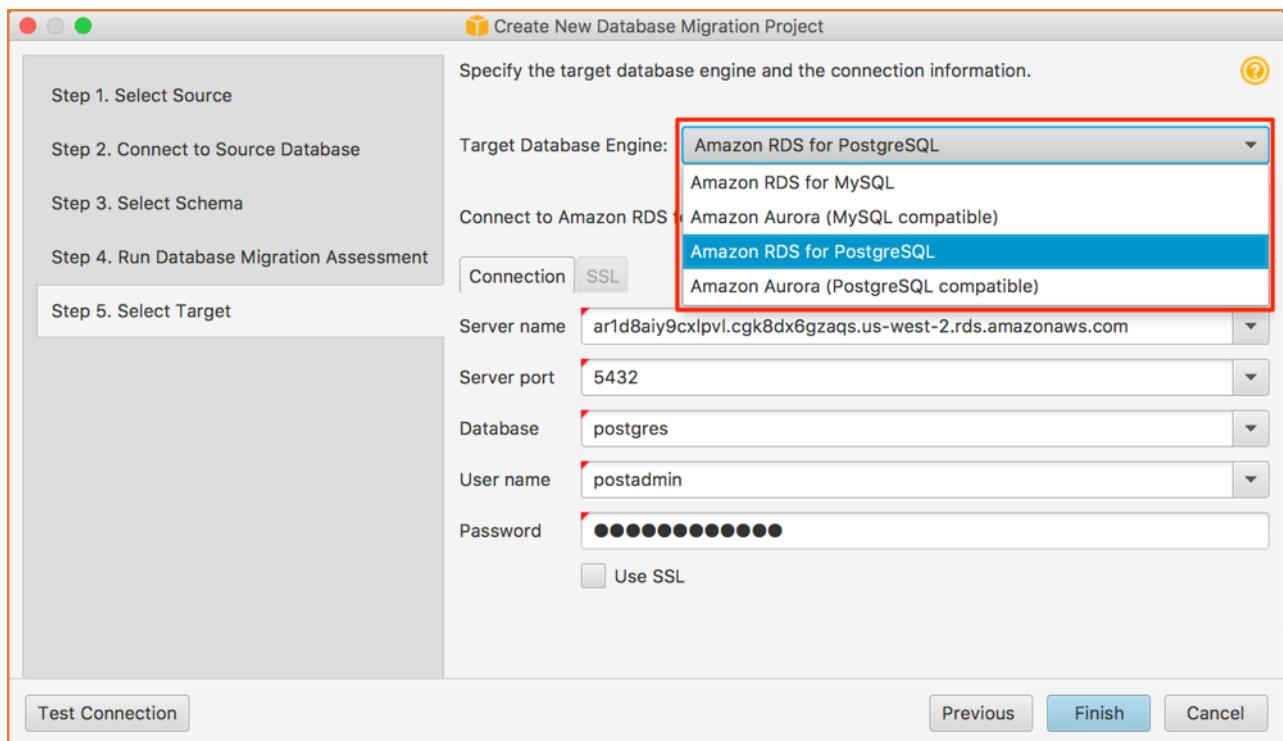


Figure 27: AWS SCT New Project Wizard - Select Target

3.1.6 New Project Wizard - Test Target Connection

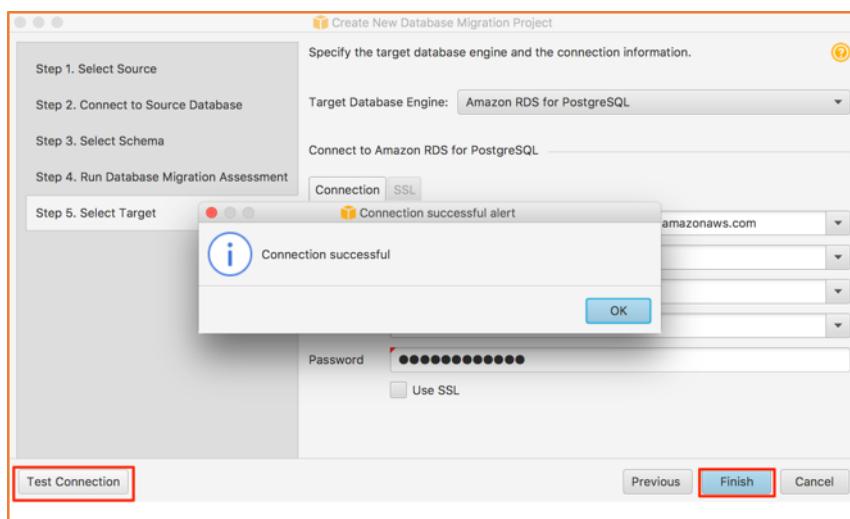


Figure 28: AWS SCT New Project Wizard - Test Target Connection

3.2 Schema Conversion in AWS SCT

3.2.1 Schema Conversion - Review Report

Review the project screen:

- Uncheck all schemas on the left-pane except for the DMS_SAMPLE schema.
- Click **Actions > Create Report**
- Go to the **Summary** tab on the top and review the generated report

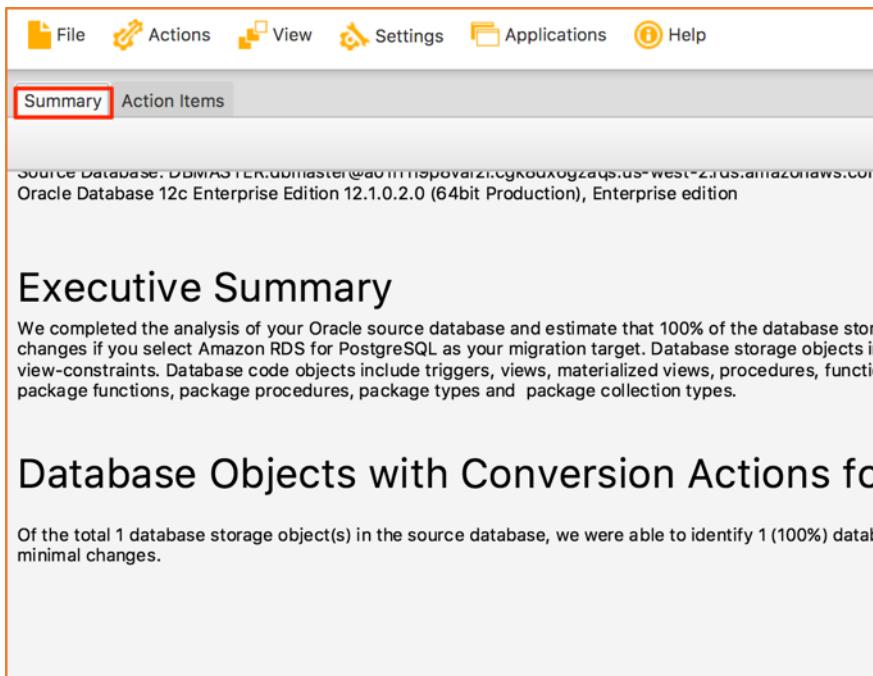


Figure 29: Schema Conversion - Review Conversion Report

3.2.2 Schema Conversion - Review and Convert Schema

On the left-hand pane, look through the schema tree of Oracle objects and note what could be automatically converted and what could not be converted.

3.2.3 Schema Conversion - Convert Source Schema

Now, right-click and click **Convert Schema** as shown below:

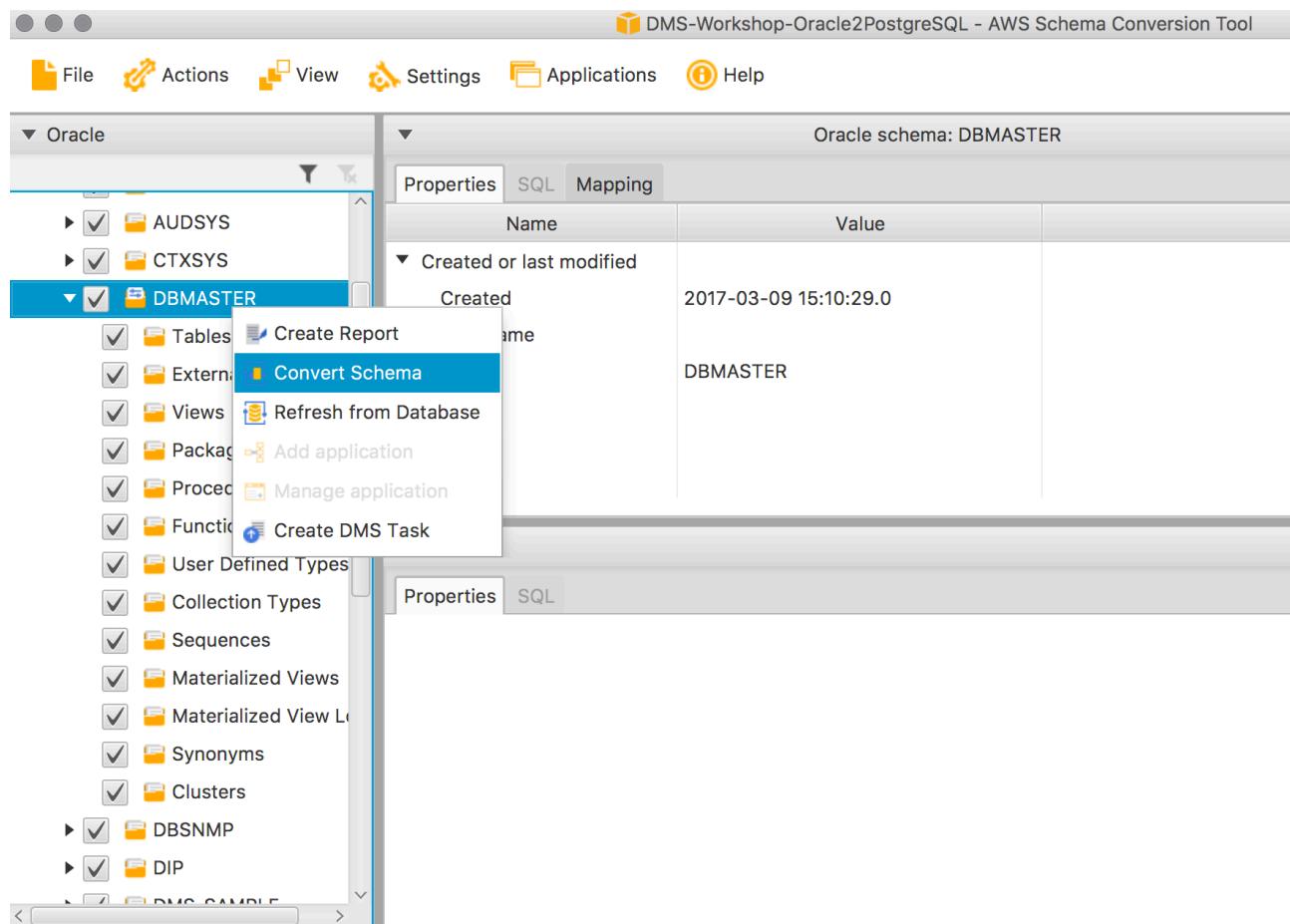


Figure 30: Schema Conversion - Source Apply (via right-click)

If right-click is not available to you, you can also select the source database and click **Actions > Convert Schema** as shown below:

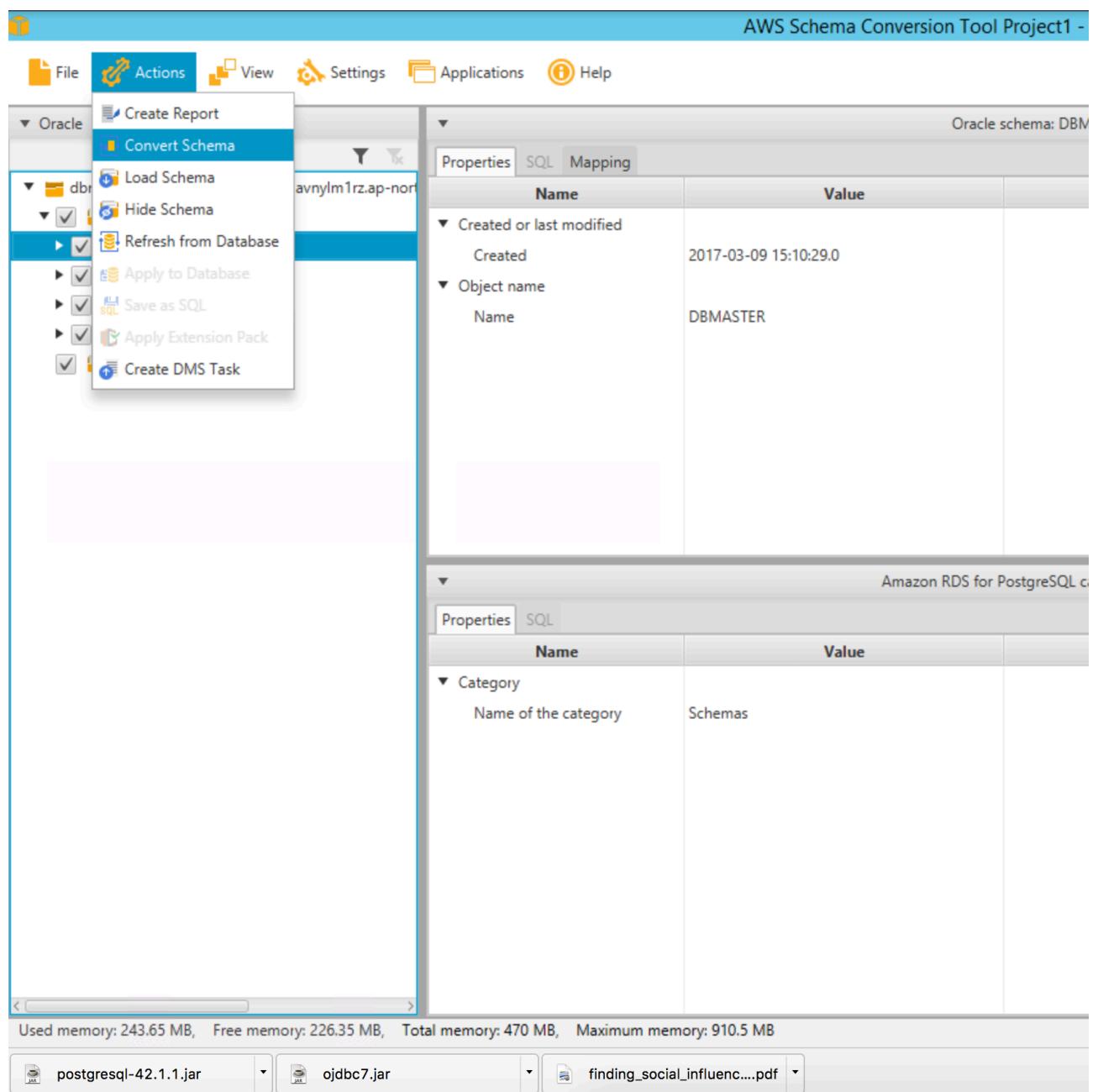


Figure 31: Schema Conversion - Source Apply (via Actions)

3.2.4 Schema Conversion - Review Target Schema

The schema will be converted and shown on the PostgreSQL instance (it has not been applied yet) as shown below:

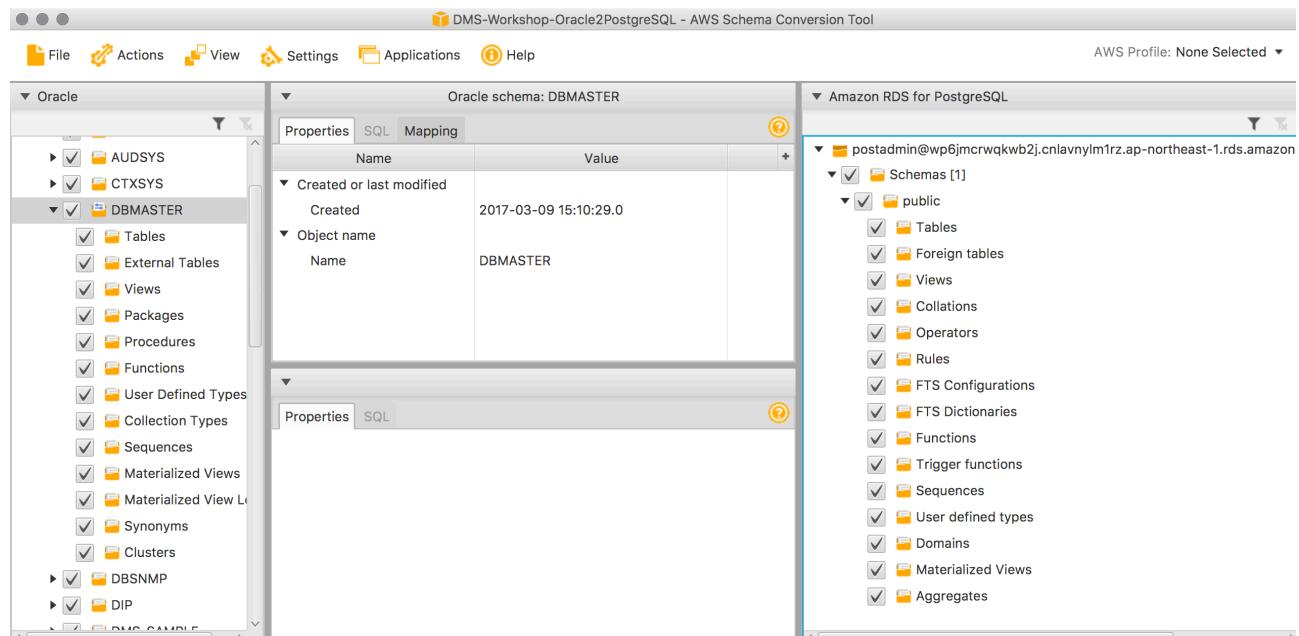


Figure 32: Schema Conversion - Target Review

Take a few minutes to review the objects being converted.

Since the majority of the objects which could not be converted are secondary objects like functions or procedures, we will proceed with the migration.

3.2.5 Schema Conversion - Apply Target Schema

Right click on the created schema on the right-hand panel, representing the target Postgres database, and click **Apply to database**, as shown below:

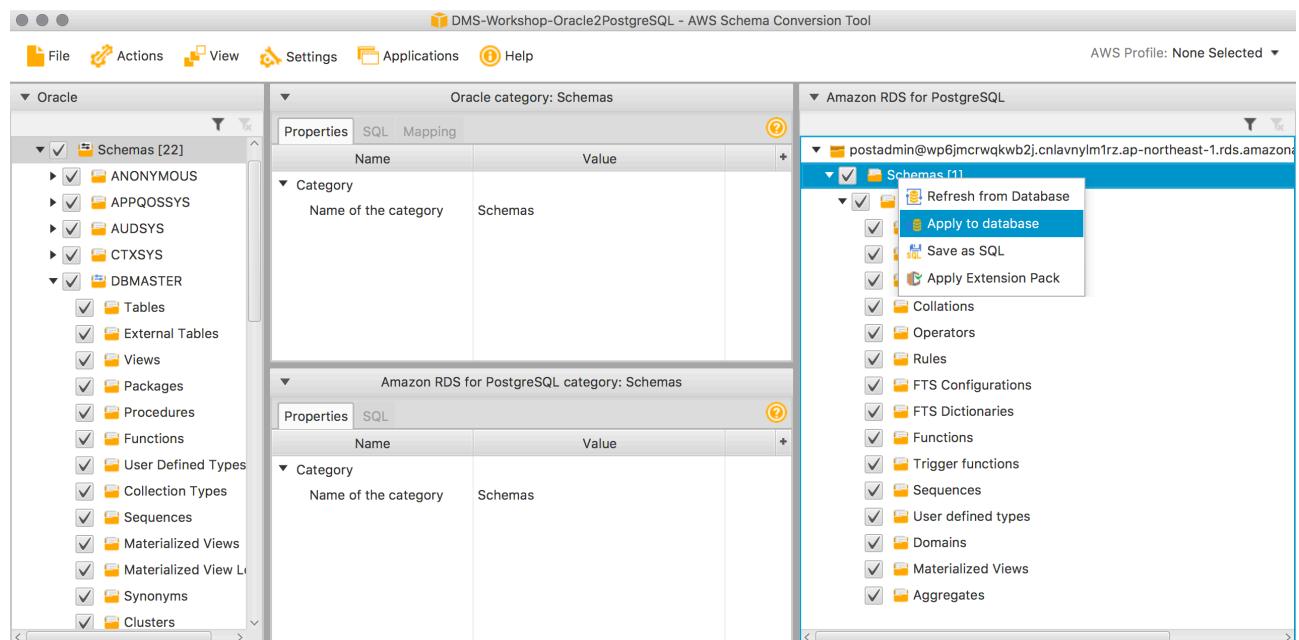


Figure 33: Schema Conversion - Target Apply (via right-click)

If right-click is not available, then select the target database name and select **Actions > Apply to Database** as shown below:

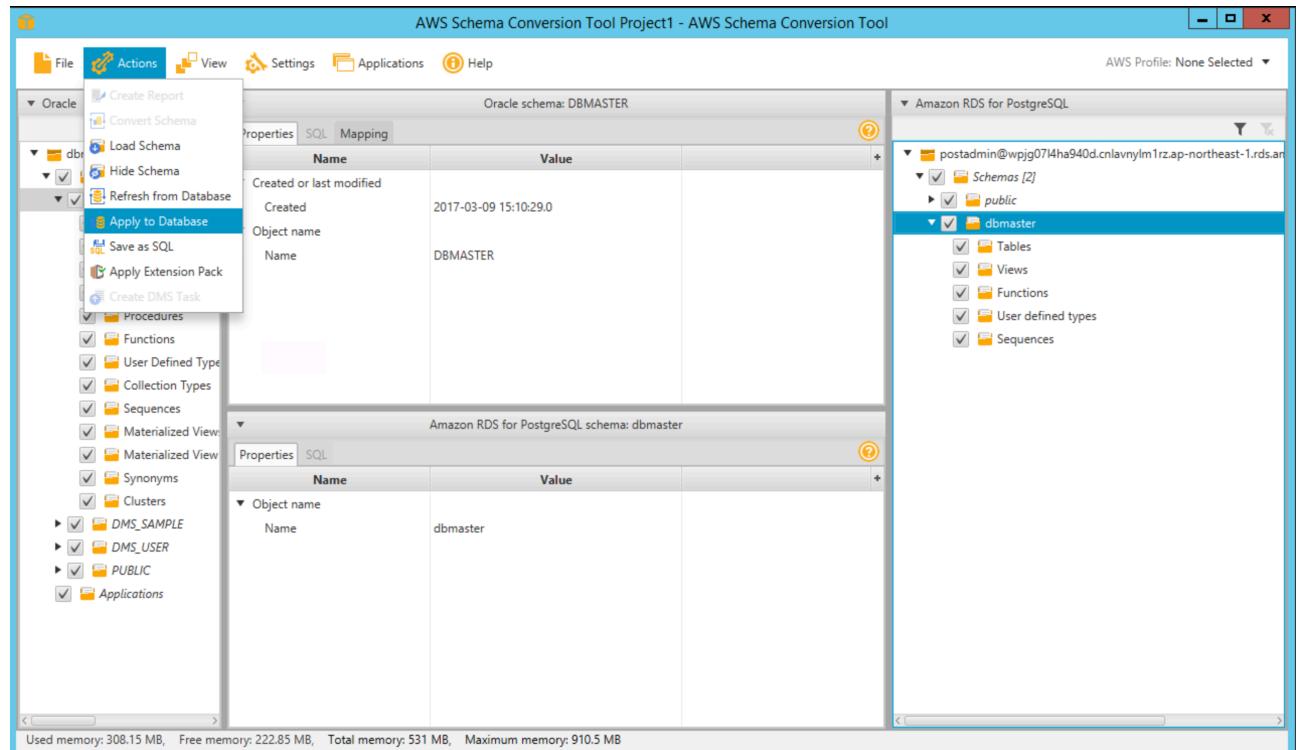


Figure 34: Schema Conversion - Target Apply (via Actions)

This will apply all those converted objects in the PostgreSQL target.

The above steps will convert all your Oracle objects into PostgreSQL objects. Objects which could not be converted automatically must be taken care of manually after migration at a later time. Now you are ready to create the AWS DMS components.

3.3 AWS Database Migration Services

The AWS Database Migration Service (AWS DMS) consist of three main components:

- **Replication Instances** for performing the movement of data
- **Endpoints** for specifying the source and target databases
- **Tasks** for controlling the execution of migrations on the replication instances

We will provide instructions on how to set up each of these three components.

3.3.1 AWS DMS Replication Instances

AWS DMS Replication Instances - List All To create our new AWS DMS replication instances, we will first view the console for replication instances by visiting the following link:

<http://amzn.to/aws-tokyo-dms-instances> (=> <https://ap-northeast-1.console.aws.amazon.com/dms/home?region=ap-northeast-1#replication-instances:>)

You should now see the following screen:

Figure 35: AWS DMS Replication Instances - List All

AWS DMS Replication Instances - Create New Next, click on **Create Replication Instance** button, and populate the values on this page:

Create replication instance

A replication instance initiates the connection between the source and target databases, transfers the data, and caches any changes that occur on the source database during the initial data load. Use the fields below to configure the parameters of your new replication instance including network and security information, encryption details, and performance characteristics. We suggest you shut down the replication instance once your migration is complete to prevent further usage charges.

The screenshot shows the 'Create new replication instance' page. It includes fields for Name (dms-workshop-oracle2postgres-repl-inst), Description (AWS DMS Workshop Lab Oracle2Postgres), Instance class (dms.t2.medium), Replication engine version (2.3.0), VPC (vpc-dbeff2bf - workshop), Multi-AZ (No), and Publicly accessible (checked). Below the form are sections for Advanced and Maintenance, and at the bottom are 'Cancel' and 'Create replication instance' buttons.

Name* dms-workshop-oracle2postgres-repl-inst

Description* AWS DMS Workshop Lab Oracle2Postgres

Instance class* dms.t2.medium

Replication engine version* 2.3.0

VPC* vpc-dbeff2bf - workshop

Multi-AZ No

Publicly accessible

Advanced

Maintenance

Cancel Create replication instance

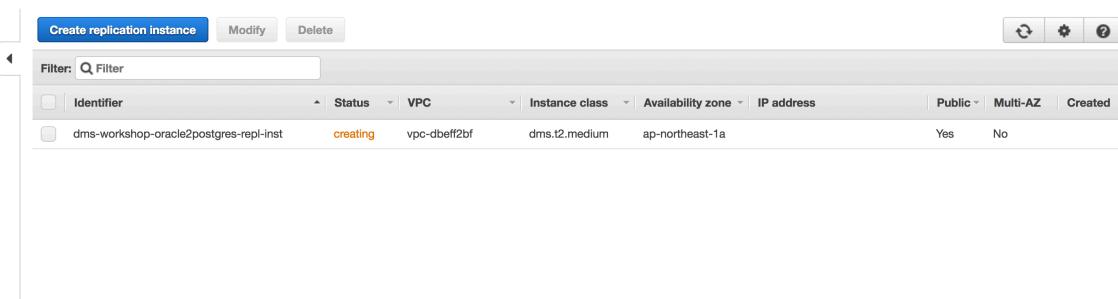
Figure 36: AWS DMS Replication Instances - Create New

using the following values:

- **Name:** dms-workshop-oracle2postgres-repl-inst
- **Description:** AWS DMS Workshop Lab Oracle2Postgres
- **Instance class:** dms.t2.medium
- **Replication engine version:** (leave as default)
- **VPC:** (select VPC created by CloudFormation stack, which should be named **workshop**)

CAUTION: The replication instance created here will continue to run until specifically deleted -- the steps for teardown are located in this document at [Teardown: AWS DMS Resources](#)

AWS DMS Replication Instances - Confirm After creation, the replication instance(s) will be provisioned and display status of **Creating**:

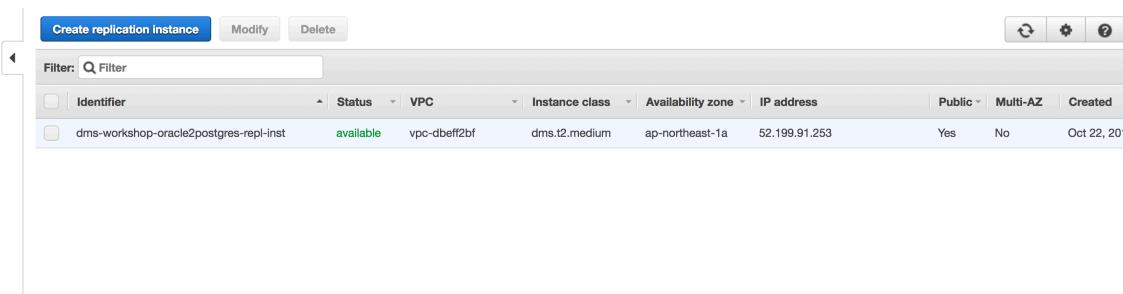


The screenshot shows the AWS DMS Replication Instances list. On the left, there's a sidebar with options like Dashboard, Get started, Tasks, Endpoints, Certificates, and Replication instances (which is selected). At the top right, there are buttons for 'Create replication instance', 'Modify', and 'Delete'. Below these are filter and search fields. The main table lists one replication instance:

Identifier	Status	VPC	Instance class	Availability zone	IP address	Public	Multi-AZ	Created
dms-workshop-oracle2postgres-repl-inst	creating	vpc-dbeff2bf	dms.t2.medium	ap-northeast-1a		Yes	No	Oct 22, 2017

Figure 37: AWS DMS Replication Instances - List New Creating

...before changing to Available:



This screenshot shows the same AWS DMS Replication Instances list as Figure 37, but the replication instance is now listed as 'available' in the status column. The rest of the information remains the same.

Identifier	Status	VPC	Instance class	Availability zone	IP address	Public	Multi-AZ	Created
dms-workshop-oracle2postgres-repl-inst	available	vpc-dbeff2bf	dms.t2.medium	ap-northeast-1a	52.199.91.253	Yes	No	Oct 22, 2017

Figure 38: AWS DMS Replication Instances - List New Available

Now that the **DMS Replication Instance** server is ready and available to perform database migrations, we will next create **DMS Endpoints** to connect the Replication Instance to the Source and Target databases.

3.3.2 AWS DMS Endpoints

AWS DMS Endpoints - List All To create our new AWS DMS endpoints, we will first view the console for endpoints by visiting the following link:

<http://amzn.to/aws-tokyo-dms-endpoints> (=> <https://ap-northeast-1.console.aws.amazon.com/dms/home?region=ap-northeast-1#endpoints:>)

You should now see the following screen:

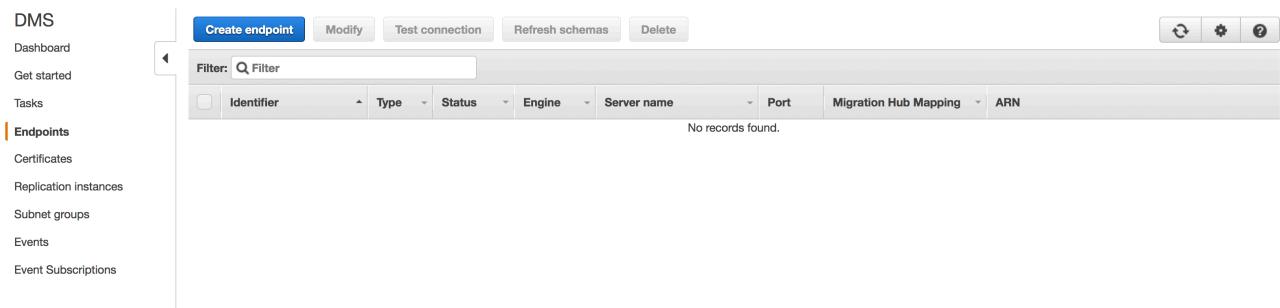


Figure 39: AWS DMS Endpoints - List All

Click the **Create Endpoint** button to proceed to creating the first endpoint.

AWS DMS Endpoints - Create Source You should now see the following page:

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 ?

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 ?

Cancel Create endpoint

Figure 40: AWS DMS Endpoints - Create 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 ?

Endpoint identifier* ?

Source engine* ?

Server name* ?

Port* ?

SSL mode* ?

User name* ?

Password* ?

SID* ?

Figure 41: AWS DMS Endpoints - Create Source Endpoint (Details)

DMS Endpoint - Enter Data for Source Endpoint

Enter the fields as follows:

- **Endpoint type:** (select default of Source)
- **Endpoint identifier:** dms-workshop-oracle
- **Source engine:** oracle
- **Server name:** (the Oracle Server Name/URL generated by the CloudFormation stack)
- **Port:** (the Oracle Server Port you provided in the CloudFormation stack parameters, default is 1521)
- **SSL mode:** none
- **User name:** dbmaster
- **Password:** (the Oracle password you provided in the CloudFormation stack parameters)
- **SID:** (the Oracle Database name you provided in the CloudFormation stack parameters, default is ORCL)

▼ 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.

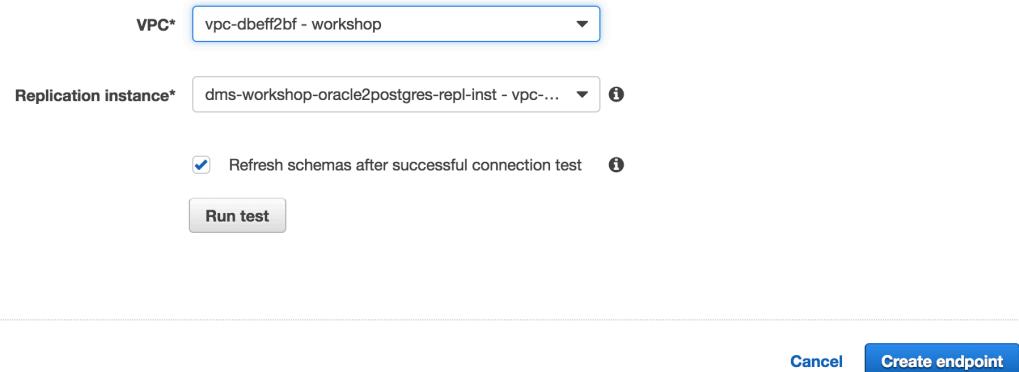


Figure 42: AWS DMS Endpoints - Create Source Endpoint (Test)

For the test portion, fill the following details:

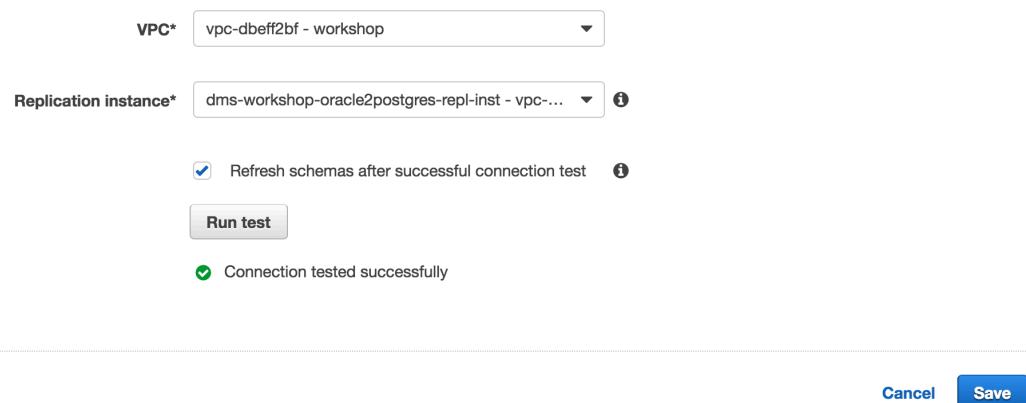
- **VPC:** (select the VPC marked with **workshop**)
- **Replication Instance:** (select the DMS Replication Instance created earlier)
- **Refresh schemas...:** (leave default of checked)

DMS Endpoint - Test Source Endpoint Selecting the **workshop** VPC and the **DMS Replication Instance** that lives in that VPC that you created earlier, you can now test the connection from that Replication Instance instance to the Source database via the **DMS Endpoint** settings above. Click the **Run Test** button to proceed. A successful test will display as Connection Successfully Tested as shown below:

▶ 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.



The screenshot shows the 'Create Source Endpoint' configuration screen. It includes fields for 'VPC*' (set to 'vpc-dbeff2bf - workshop'), 'Replication instance*' (set to 'dms-workshop-oracle2postgres-repl-inst - vpc-...'), and a checked checkbox for 'Refresh schemas after successful connection test'. Below these are 'Run test' and 'Save' buttons. A success message 'Connection tested successfully' is displayed below the test button. At the bottom right are 'Cancel' and 'Save' buttons.

Figure 43: AWS DMS Endpoints - Create Source Endpoint (Test Success)

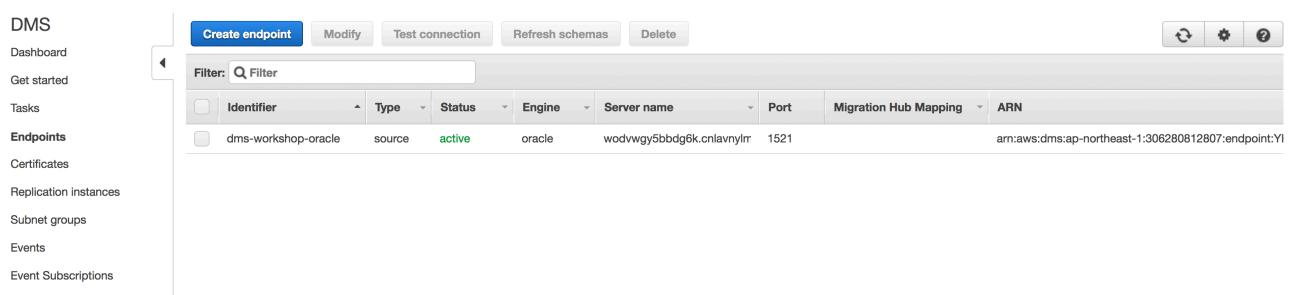
You can now click the **Save** button to save this endpoint.

Congratulations! You have successfully set up and tested the Source endpoint. We will now repeat the process for the Target endpoint.

AWS DMS Endpoints - List All (Updated) We will return to the DMS Endpoints view in the console by visiting the following link:

<http://amzn.to/aws-tokyo-dms-endpoints> (=> <https://ap-northeast-1.console.aws.amazon.com/dms/home?region=ap-northeast-1#endpoints:>)

You should now see the following screen:



The screenshot shows the 'List All' endpoint table. The table has columns: Identifier, Type, Status, Engine, Server name, Port, Migration Hub Mapping, and ARN. There is one entry: 'dms-workshop-oracle' (source, active, oracle, wodwgy5bbdg6k.cnlavnylrm, 1521, arn:aws:dms:ap-northeast-1:306280812807:endpoint:YI).

Figure 44: AWS DMS Endpoints - List All (Updated)

Again, we will click to **Create Endpoint**, seeing the Create Endpoint page as before.

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.

The screenshot shows the 'Create endpoint' form for creating a target endpoint. The 'Endpoint type*' dropdown is set to 'Target'. The 'Endpoint identifier*' field contains 'dms-workshop-postgres'. The 'Target engine*' dropdown is set to 'postgres'. The 'Server name*' field contains 'wpjg07l4ha940d.cnlavnylm1rz.ap-northeast-1.rds.amazonaws.com'. The 'Port*' field is set to '5432'. The 'SSL mode*' dropdown is set to 'none'. The 'User name*' field contains 'postadmin'. The 'Password*' field is masked. The 'Database name*' field contains 'postgres'. Below the form, there is a link labeled 'Advanced'.

Endpoint type* Source Target

Endpoint identifier* dms-workshop-postgres

Target engine* postgres

Server name* wpjg07l4ha940d.cnlavnylm1rz.ap-northeast-1.rds.amazonaws.com

Port* 5432

SSL mode* none

User name* postadmin

Password*

Database name* postgres

▶ Advanced

Figure 45: AWS DMS Endpoints - Create Target Endpoint (Details)

Enter the fields as follows:

- **Endpoint type:** Target
- **Endpoint identifier:** dms-workshop-postgres
- **Source engine:** postgres
- **Server name:** (the Postgres Server Name/URL generated by the CloudFormation stack)
- **Port:** (the Postgres Server Port you provided in the CloudFormation stack parameters, default is 5432)
- **SSL mode:** none
- **User name:** (the Postgres password you provided in the CloudFormation stack parameters, the default is postadmin)
- **Password:** (the Postgres password you provided in the CloudFormation stack parameters)
- **Database name:** (the Postgres database name you provided in the CloudFormation stack parameters, default is postgres)

▼ 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.

The screenshot shows the 'Test endpoint connection' section of the AWS DMS Endpoints creation interface. It includes fields for 'VPC*' (set to 'vpc-dbeff2bf - workshop'), 'Replication instance*' (set to 'dms-workshop-oracle2postgres-repl-inst - vpc-...'), and a 'Run test' button. Below the button, a status message says 'Testing endpoint connection...' with a progress bar. At the bottom right are 'Cancel' and 'Create endpoint' buttons.

Figure 46: AWS DMS Endpoints - Create Target Endpoint (Test)

For the test portion, fill the following details:

- **VPC:** (select the VPC marked with **workshop**)
- **Replication Instance:** (select the DMS Replication Instance created earlier)
- **Refresh schemas...:** (leave default of checked)

DMS Endpoint - Test Target Endpoint Selecting the **workshop** VPC and the **DMS Replication Instance** that lives in that VPC that you created earlier, you can now test the connection from that Replication Instance instance to the Target database via the **DMS Endpoint** settings above. Click the **Run Test** button to proceed. A successful test will display as Connection Successfully Tested as shown below:

▼ 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.

The screenshot shows the 'Test endpoint connection' section of the AWS DMS Endpoints creation interface after a successful test. It includes fields for 'VPC*' (set to 'vpc-dbeff2bf - workshop'), 'Replication instance*' (set to 'dms-workshop-oracle2postgres-repl-inst - vpc-...'), and a 'Run test' button. Below the button, a status message says 'Connection tested successfully' with a green checkmark. At the bottom right are 'Cancel' and 'Save' buttons.

Figure 47: AWS DMS Endpoints - Create Target Endpoint (Test Success)

You can now click the **Save** button to save this endpoint. You should now see the list of DMS Endpoints with two endpoints in an Active status as shown below:

Identifier	Type	Status	Engine	Server name	Port	Migration Hub Mapping	ARN
dms-workshop-oracle	source	active	oracle	wodwgy5bbdg6k.cnlavnylrr	1521		arn:aws:dms:ap-northeast-1:306280812807:endpoint:YI
dms-workshop-postgres	target	active	postgres	wpjg0714ha940d.cnlavnylm1	5432		arn:aws:dms:ap-northeast-1:306280812807:endpoint:5f

Figure 48: AWS DMS Endpoints - List Endpoints (Source and Target)

You have now successfully created both DMS Endpoints, Source and Target, and you can now move to the creation of a DMS Task.

3.3.3 AWS DMS Tasks

The following link will display all **DMS Tasks** in this region

<http://amzn.to/aws-tokyo-dms-tasks> (=> <https://ap-northeast-1.console.aws.amazon.com/dms/home?region=ap-northeast-1#tasks:;>)

AWS DMS Tasks -- List All You should now see the following:

ID	Status	Source	Target	Type	Complete %	Elapsed time	Tables loaded	Tables loading	Tables error
No records found.									

Figure 49: AWS DMS Tasks - Create Task

Create task

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 can be created automatically.

The screenshot shows the 'Create task' step of the AWS DMS wizard. It includes fields for Task name (dms-workshop-task-oracle2postgres), Replication instance (dms-workshop-oracle2postgres-repl-inst ...), Source endpoint (dms-workshop-oracle), Target endpoint (dms-workshop-postgres), and Migration type (Migrate existing data and replicate ongoing changes). A 'Did you know' box provides information about replicating ongoing changes and archive log retention. A note at the bottom explains that the source is Oracle and requires supplemental logging. Another note suggests setting archive log retention for RDS databases.

Task name* dms-workshop-task-oracle2postgres

Replication instance* dms-workshop-oracle2postgres-repl-inst ...

Source endpoint* dms-workshop-oracle

Target endpoint* dms-workshop-postgres

Migration type* Migrate existing data and replicate ongoi...

Did you know
When switching Schema Converter convert your dat...
[Learn more](#)

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);

Start task on create

Figure 50: AWS DMS Tasks - Create Task

AWS DMS Tasks - Create New

Enter the following data:

- **Task name:** dms-workshop-task-oracle2postgres
- **Replication instance:** (defaults to **DMS Replication Instance** you created earlier)
- **Source endpoint:** (defaults to **DMS Endpoint** for the Oracle source database that you created earlier)
- **Target endpoint:** (defaults to **DMS Endpoint** for the Postgres source database that you created earlier)
- **Migration type:** Migrate existing data and replicate ongoing changes
- **Start task on create:** (leave this checked)

▼ Task Settings

Target table preparation mode* Do nothing ?
 Drop tables on target ?
 Truncate ?

Stop task after full load completes* Don't stop ?
 Stop Before Applying Cached Changes ?
 Stop After Applying Cached Changes ?

Include LOB columns in replication* Don't include LOB columns ?
 Full LOB mode ?
 Limited LOB mode ?

Max LOB size (kb)* ?

Enable logging

CloudWatch Logs usage will be charged at standard rates. See [here](#) for more details.

Advanced Settings**▼ Table mappings****Guided** **JSON**

Figure 51: AWS DMS Tasks - Create Task (Continued)

Enter the following data:

- **Target table preparation mode:** Truncate
- **Stop task after full load completes:** Don't stop
- **Include LOB columns in replication:** Limited LOB mode
- **Max LOB size (kb):** 32
- **Enable logging:** selected/checked

▼ Table mappings

Guided JSON

Selection rules i

At least one selection rule with an include action is required. Once you have one or more selection rules, you can add transformation rules.

Where i

Schema name is DMS_SAMPLE

Table name is like %
Use % as a wildcard.

Action Include i

Filter i

Add column filter

Add selection rule

The screenshot shows the 'Table mappings' section of the AWS DMS Tasks interface. It includes tabs for 'Guided' (selected) and 'JSON'. A 'Selection rules' section is highlighted, containing a note about requiring at least one selection rule with an 'include' action. It features a 'Where' filter with 'Schema name is' set to 'DMS_SAMPLE' and 'Table name is like' set to '%'. An 'Action' dropdown is set to 'Include'. Below this is a 'Filter' section with a 'Add column filter' link. A prominent blue 'Add selection rule' button is located at the bottom right.

Figure 52: AWS DMS Tasks - Create Task (Table Mappings Selection)

AWS DMS Tasks - Add Selections

▼ Table mappings

Guided

JSON

Selection rules ⓘ

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



+ add selection rule

Transformation rules ⓘ

Target Schema ⓘ

Where ⓘ

Schema name is DMS_SAMPLE

Action ⓘ

Action Make lowercase

cancel

Add transformation rule

Figure 53: AWS DMS Tasks - Create Task (Table Mappings Transform #1)

▼ 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

Target: Table

Where ⓘ

Schema name is: DMS_SAMPLE

Table name is like: %

Use % as a wildcard.

Action ⓘ

Action: Make lowercase

[cancel](#) [Add transformation rule](#)

The screenshot shows the 'Table mappings' section of the AWS DMS Tasks configuration. It includes a 'Selection rules' section with a filter for 'schema name is like 'DMS_SAMPLE'' and 'table name is like '%''. Below it is a 'Transformation rules' section for the 'DMS_SAMPLE' schema, which applies a 'Make lowercase' action to table names that start with a wildcard (%). The interface includes standard AWS navigation and save buttons.

Figure 54: AWS DMS Tasks - Create Task (Table Mappings Transform #2)

▼ 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



Target

Column



Where ⓘ

Schema name is

DMS_SAMPLE



Table name is like

%

Column name is like

%

Use % as a wildcard.

Action ⓘ

Action

Make lowercase



cancel

Add transformation rule

Figure 55: AWS DMS Tasks - Create Task (Table Mappings Transform #3)

AWS DMS Tasks - Add Transformations

AWS DMS Tasks - Create Task (Final) Review the information and click the **Create Task** button to continue.

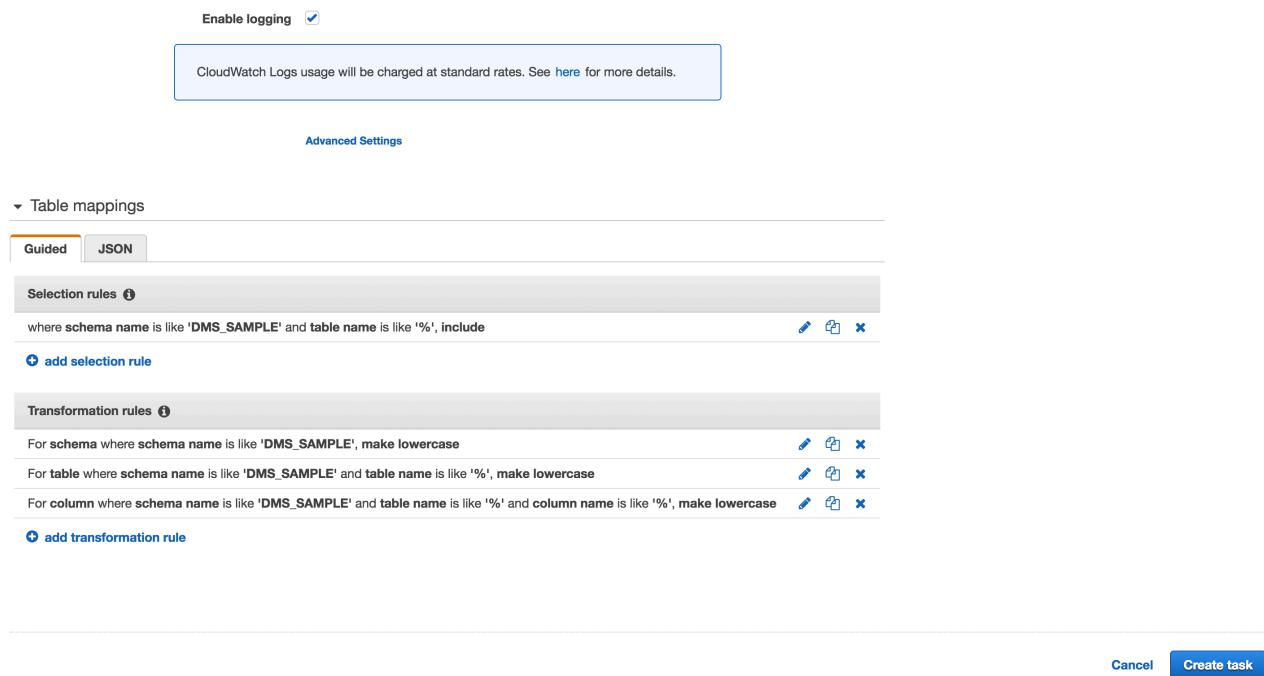


Figure 56: AWS DMS Tasks - Create Task (Final)

You will now see the new DMS Task listed with a status of Creating, then Starting, then Running, as shown below

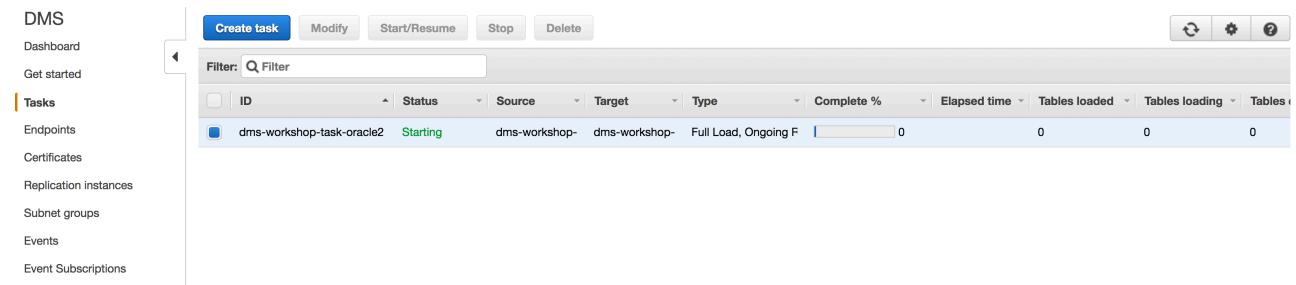


Figure 57: AWS DMS Tasks - List Tasks (Updated)

You have now successfully set up all the major components for a database migration with AWS DMS. Next, we will troubleshoot some errors that commonly occur during real-world migrations.

3.4 AWS DMS Troubleshooting

3.4.1 AWS DMS Troubleshooting - List Tasks (Errors)

After you have created your **DMS Task** and some time has passed, you can revisit the list of DMS Tasks using the following link:

<http://amzn.to/aws-tokyo-dms-tasks> (=> <https://ap-northeast-1.console.aws.amazon.com/dms/home?region=ap-northeast-1#tasks:>)

You should see the following:

ID	Status	Source	Target	Type	Complete %	Elapsed time	Tables loaded	Tables loading	Tables error
dms-workshop-task-oracle2	Error	dms-workshop-	dms-workshop-	Full Load, Ongoing F	100	0m	7	0	0

Figure 58: AWS DMS Troubleshooting - List Tasks (Errors)

3.4.2 AWS DMS Troubleshooting - View Logs

AWS DMS Logs are kept as CloudWatch Logs -- clicking the Log tab for the DMS Task will display a link to the CloudWatch Log as shown below:

Figure 59: AWS DMS Troubleshooting - View Log (Link)

You will then see the following:

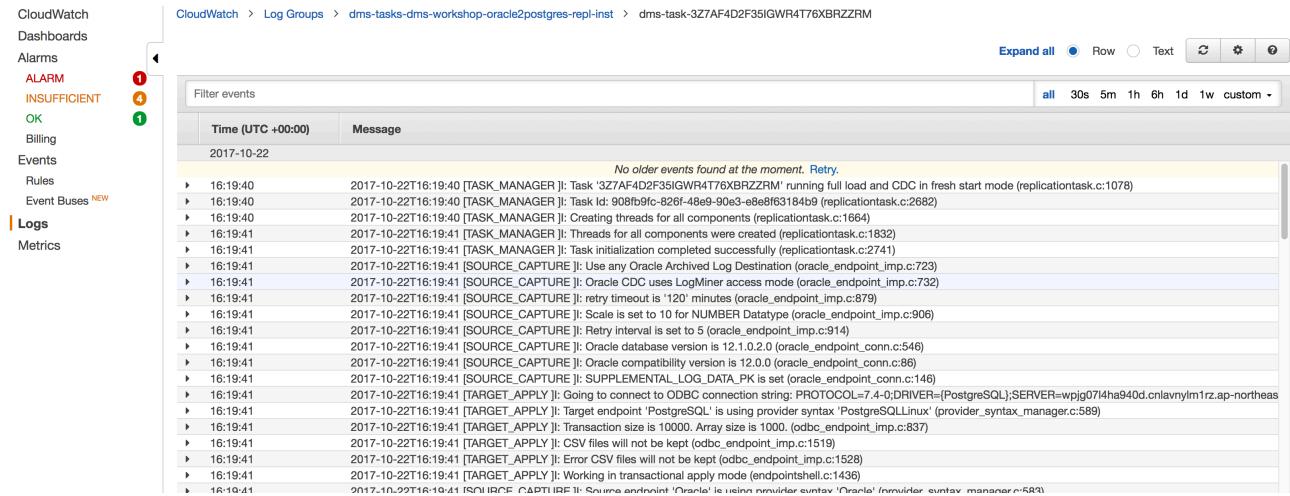


Figure 60: AWS DMS Troubleshooting - View CloudWatch Log (All)

The complete logging data is significant, so we will filter the CloudWatch logs by:

- Filter all logs for the previous seven days using the '1w' option in the upper-right of the screen
- Change display option (also in upper-right) to display all log entries as **Text** (this can help with readability as an option to the default of **Rows**)
- Filter using the text string of "**E:**" (including the double quotes), which will limit the display to messages with the log level of Error

Once changed, the page should look as follows:

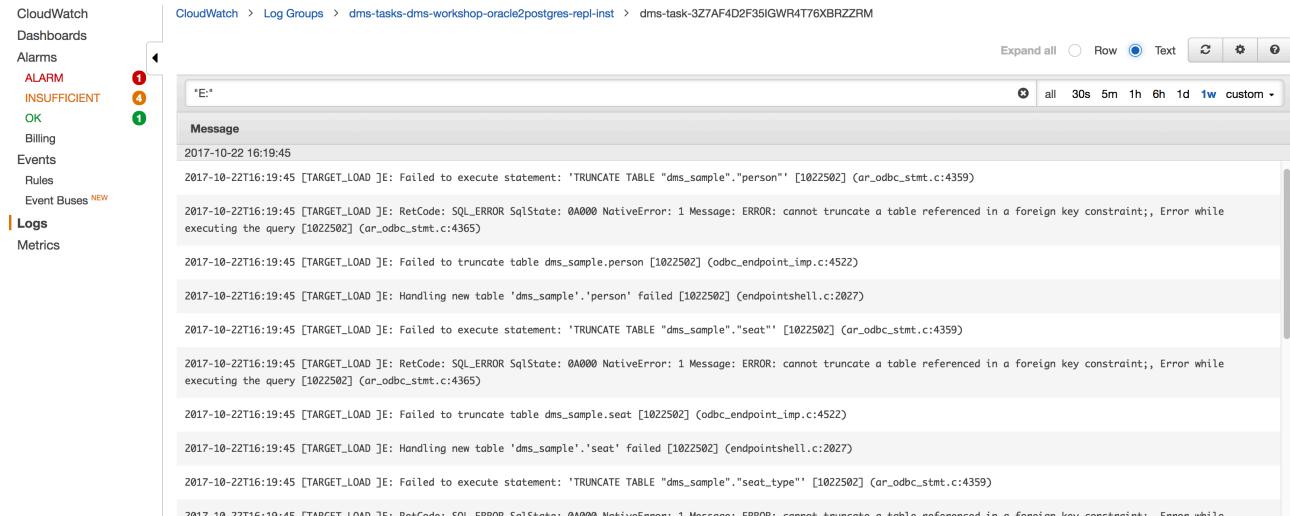


Figure 61: AWS DMS Troubleshooting - View CloudWatch Log (Errors Only)

We will address these errors one by one

3.5 AWS DMS Errors

The following are common errors and fixes

3.5.1 Error: “CANNOT TRUNCATE A TABLE REFERENCED IN A FOREIGN KEY CONSTRAINT”

```
2017-10-22T16:19:45 [TARGET_LOAD ]E: Failed to execute statement: 'TRUNCATE TABLE "dms_sample"."person"' [1022502] (ar_odbc_stmt.c:4359)
2017-10-22T16:19:45 [TARGET_LOAD ]E: RetCode: SQL_ERROR SqlState: 0A000 NativeError: 1 Message: ERROR: cannot truncate a table referenced in a foreign key constraint;, Error while
executing the query [1022502] (ar_odbc_stmt.c:4365)
```

Figure 62: AWS DMS Errors - Truncate

Why am I seeing this error? 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 Oracle source objects into PostgreSQL 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. Foreign keys or referential integrity constraints in PostgreSQL are implemented using triggers. One way to disable foreign keys is to disable all triggers temporarily from the instance and do the loads.

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 need to use do_nothing for the target table prep mode.

More details available in this blog post: <http://blog.endpoint.com/2015/01/postgresessionreplication-role.html>

How do I solve this? Delete the CloudWatch Log stream for **dms-workshop-task** – this will give you a clean environment, you can get there through the following link:

<http://amzn.to/aws-dms-workshop-log-streams> (=> <https://ap-northeast-1.console.aws.amazon.com/cloudwatch/home?region=ap-northeast-1#logStream:group=dms-tasks-dms-workshop-instance;streamFilter=typeLogStreamPrefix>)

Select the CloudWatch Log Stream, then click the **Delete Log Stream** button:

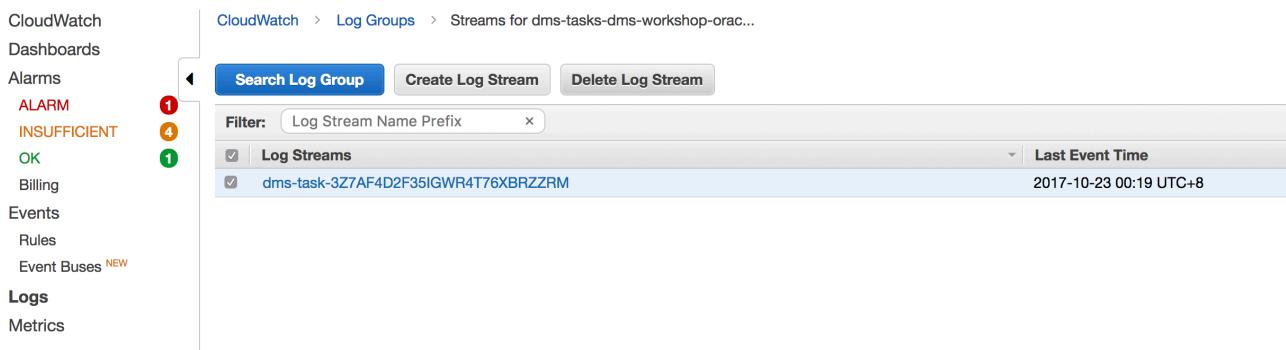


Figure 63: AWS DMS Errors - Delete Log Stream

Now, return to the **DMS Tasks** page at the following link:

<http://amzn.to/aws-tokyo-dms-tasks> (=> <https://ap-northeast-1.console.aws.amazon.com/dms/home?region=ap-northeast-1#tasks:>)

Select the **DMS Task** experiencing errors, and click **Stop** button:

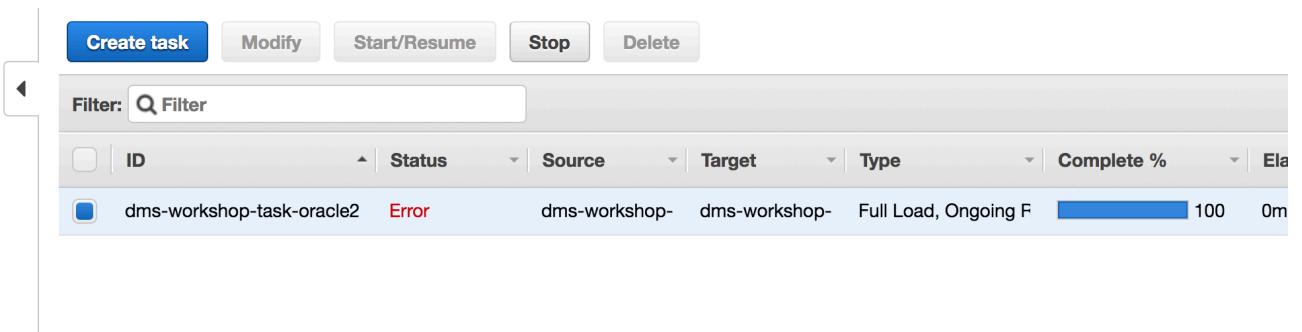


Figure 64: AWS DMS Errors - Stop DMS Task

... then, click the **Modify** button, which presents the following page:

Modify Task

Modify your replication task's properties

The screenshot shows the 'Modify Task' page in the AWS DMS console. At the top, there are fields for 'Task name' (dms-workshop-task-oracle2postgres) and 'Migration type' (Migrate existing data and replicate ongoing changes). A checkbox 'Start task on modify' is checked. Below these are two sections: 'Task Settings' (expanded) and 'Table mappings' (expanded). Under 'Table mappings', there are tabs for 'Guided' (selected) and 'JSON'. The 'Selection rules' section contains a rule: 'where schema name is like 'DMS_SAMPLE' and table name is like '%', include'. The 'Transformation rules' section contains three rules: 'For schema where schema name is like 'DMS_SAMPLE'', 'For table where schema name is like 'DMS_SAMPLE'' (both make lowercase), and 'For column where schema name is like 'DMS_SAMPLE' and table name is like '%' and column name is like '%', make lowercase'.

Figure 65: AWS DMS Errors - Modify DMS Task

Expand the Task Setting, as shown below:

Task name dms-workshop-task-oracle2postgres

Migration type Migrate existing data and replicate ongoing cha... ▼

Start task on modify

▼ Task Settings

Target table preparation mode Do nothing Drop tables on target Truncate

Stop task after full load completes Don't stop Stop Before Applying Cached Changes Stop After Applying Cached Changes

Include LOB columns in replication Don't include LOB columns -- -- -- --

Figure 66: AWS DMS Errors - Change Table Prep (Before)

and change **Target table preparation mode** to **Do Nothing** as it is shown below:

Modify your replication task's properties

Task name dms-workshop-task-oracle2postgres

Migration type Migrate existing data and replicate ongoing cha... ▼

Start task on modify

▼ Task Settings

Target table preparation mode Do nothing Drop tables on target Truncate

Stop task after full load completes Don't stop Stop Before Applying Cached Changes Stop After Applying Cached Changes

Figure 67: AWS DMS Errors - Change Table Prep (After)

Save your changes to the Task by clicking **Modify**, then select **Restart** in the following page as shown below.

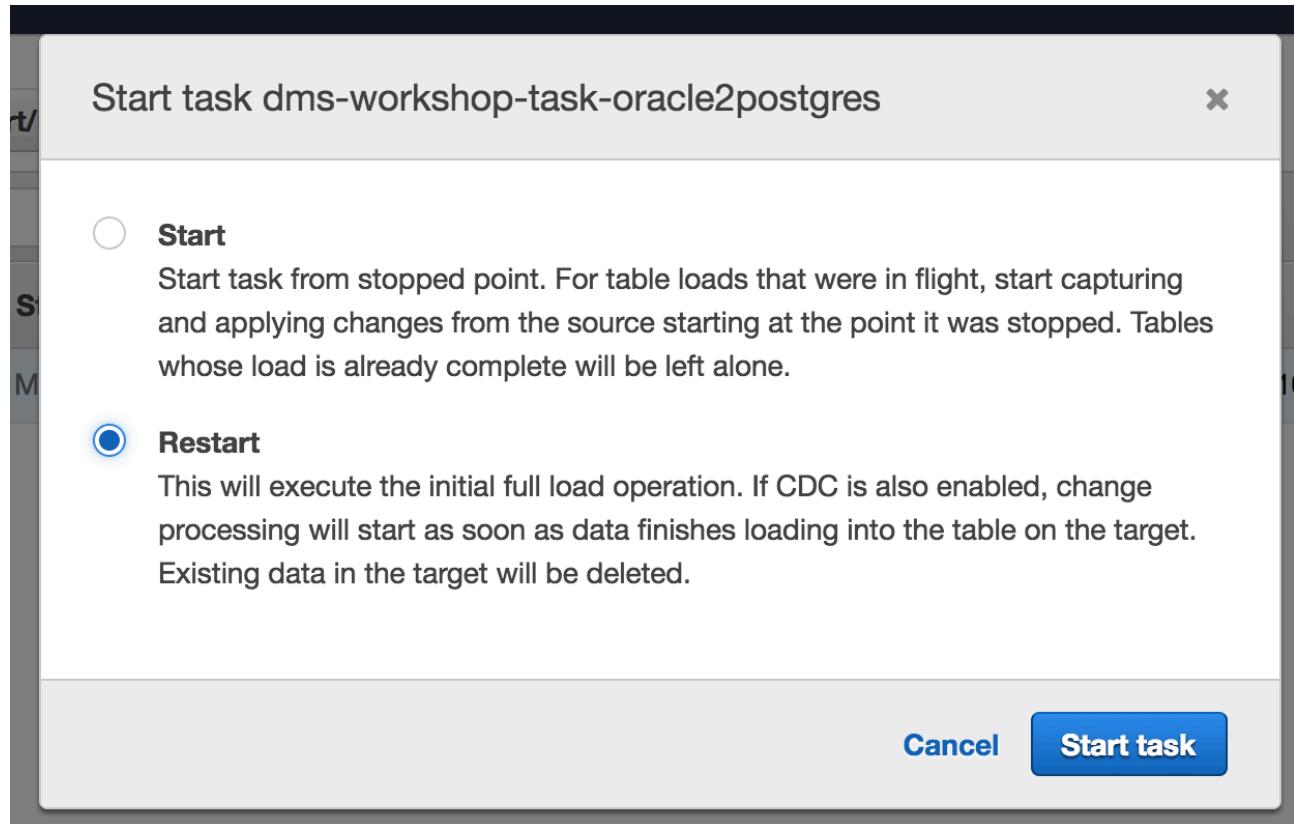


Figure 68: AWS DMS Errors - Restart Task

The Task will try to perform an initial load from the beginning, so you should wait a few minutes for the load to complete.

You can check the migration progress for each table. For example, let's check status for 'PERSON' table:

- Go to CloudWatch Logs for the **dms-workshop-task** and search for "PERSON" (including double quotes)
- You can see 3 set of messages for each table:
 - Scan and unloading of records by DMS from Oracle
 - Records being loaded to Postgres
 - Confirmation of records loaded into Postgres

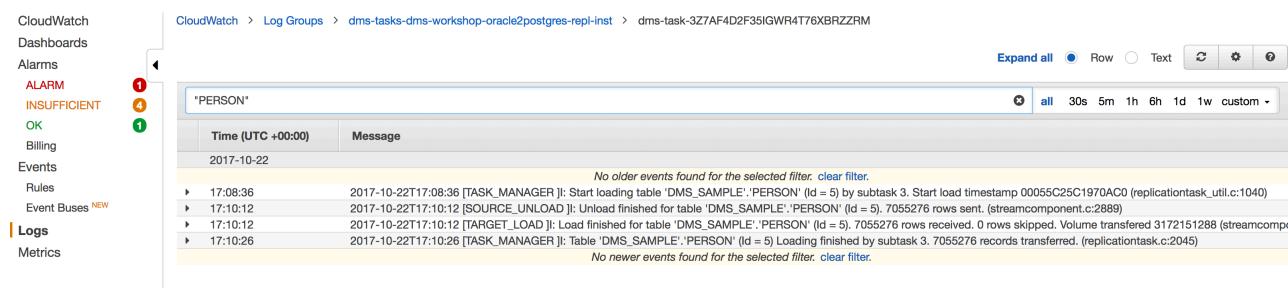


Figure 69: AWS DMS Errors - CloudWatch Log Filtered for PERSON table

You have successfully fixed the truncation issue and now data is being loaded from Oracle to Postgres, but you still see errors in the logs. You will look at another common error next related to supplemental logging.

3.6 AWS DMS Errors

The following are common errors and fixes

3.6.1 Error: “SUPPLEMENTAL LOGGING FOR TABLE ‘DMS_SAMPLE.MLB_DATA’ IS NOT ENABLED PROPERLY”

```
2017-10-22T16:19:45 [TARGET_LOAD]E: Failed to execute statement: 'TRUNCATE TABLE "dms_sample"."person"' [1022502] (ar_odbcs_stmt.c:4359)
2017-10-22T16:19:45 [TARGET_LOAD]E: RetCode: SQL_ERROR SqlState: 0A000 NativeError: 1 Message: ERROR: cannot truncate a table referenced in a foreign key constraint;, Error while
executing the query [1022502] (ar_odbcs_stmt.c:4365)
```

Figure 70: AWS DMS Errors - Supplemental Logging

4 Lab Teardown

4.1 Teardown AWS CloudFormation Stack

TODO: complete this

4.2 Teardown AWS DMS Resources

TODO: complete this