AWS Database Migration Services

Workshop: Oracle to PostgreSQL Migration

Revised: 2017.10.22

Contents

List of Tables

List of Figures

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 are 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:

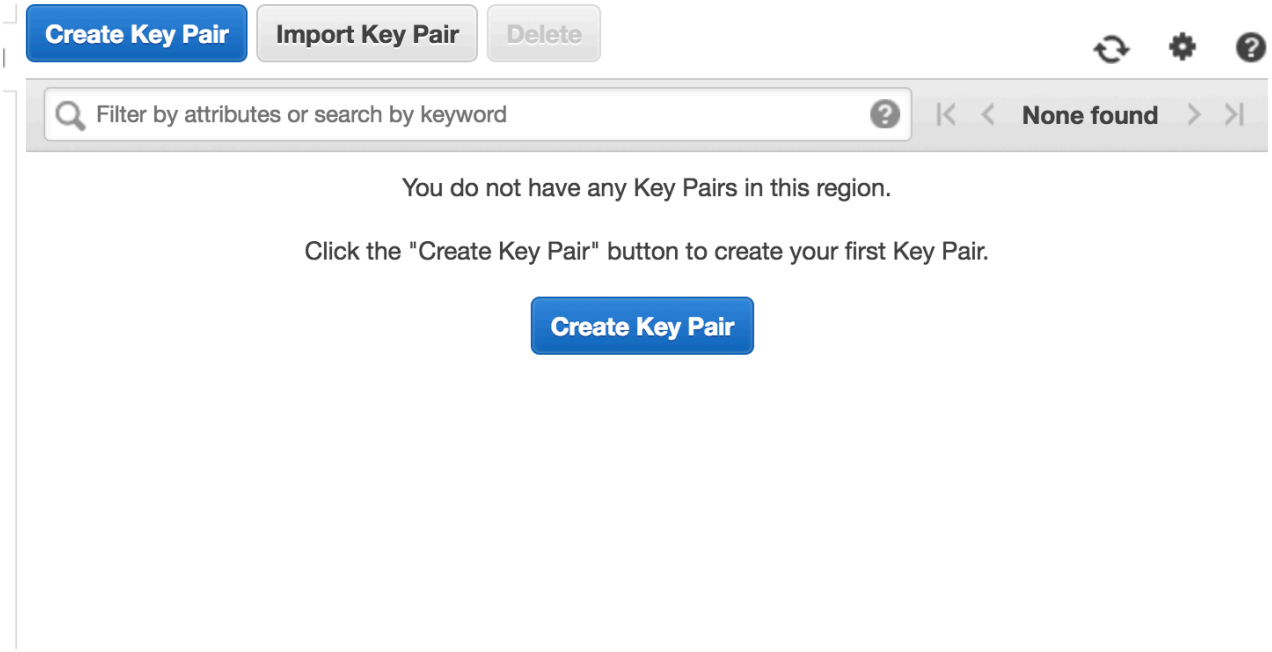

Figure 2: *Key Pairs: List All Keys*

> *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.

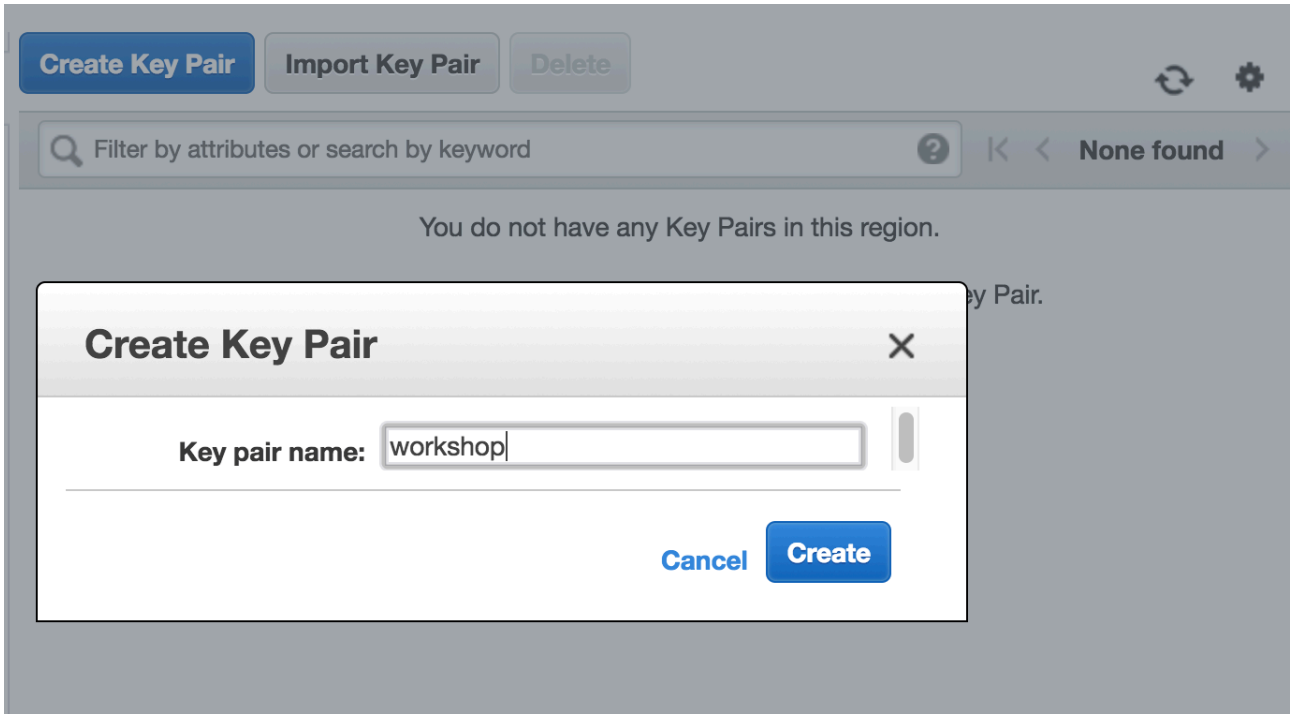

Figure 3: *Key Pair: Create Dialog*

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

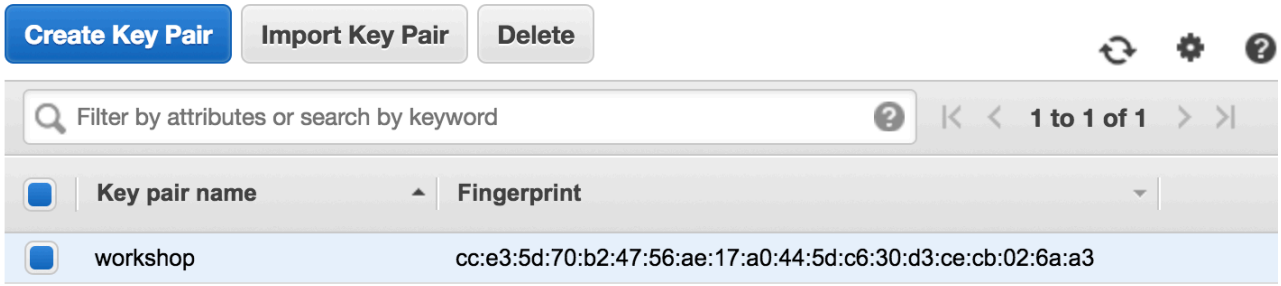

Figure 4: *Key Pair: Created*

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 that you will be spinning up 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:

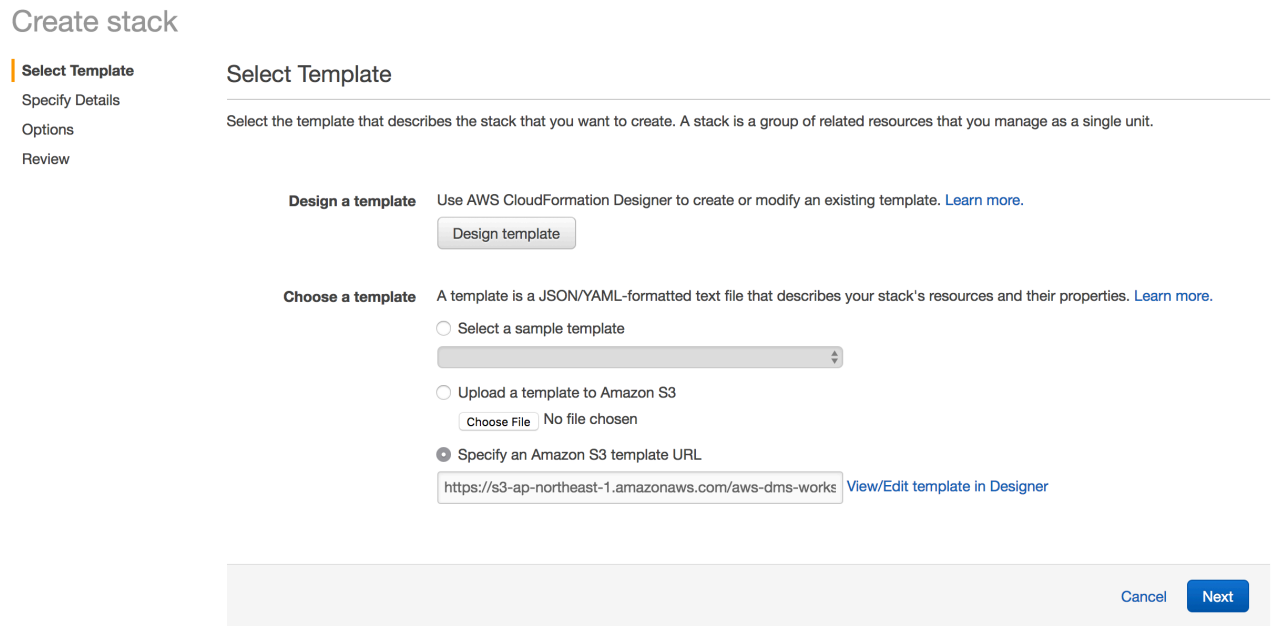

Figure 5: *AWS CloudFormation Stack Selection: Template*

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.

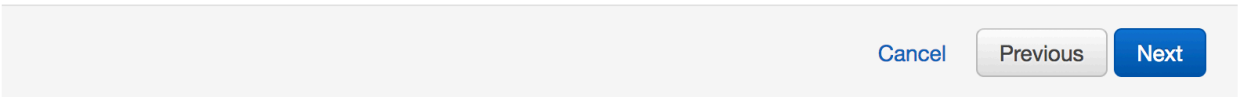

Figure 6: *AWS CloudFormation Stack Selection: Confirmation*

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.**

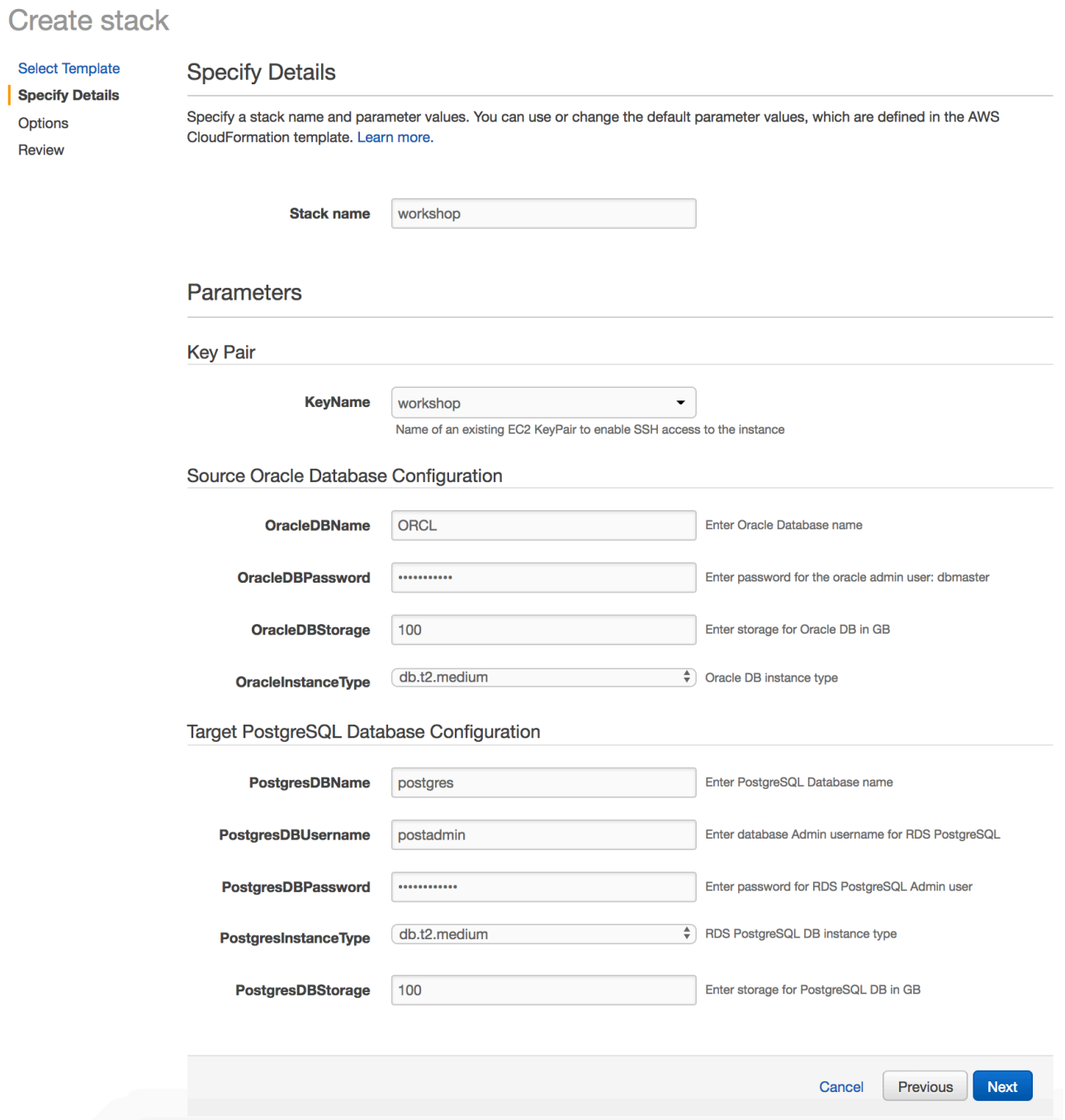

Figure 7: *AWS CloudFormation: Settings Overview*

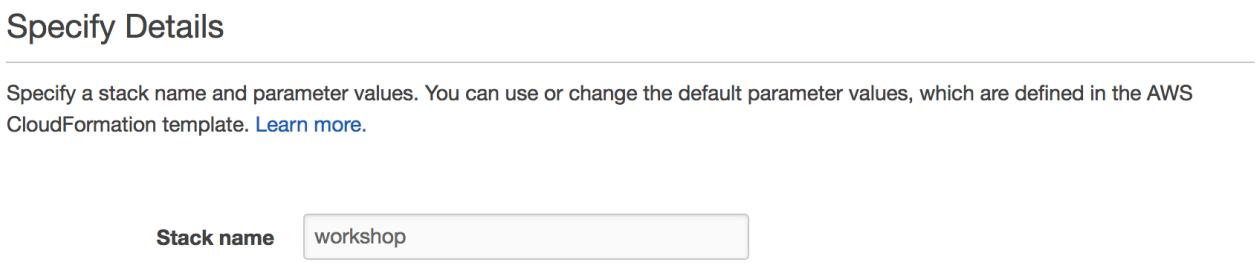

Figure 8: *AWS CloudFormation: Stack Name*

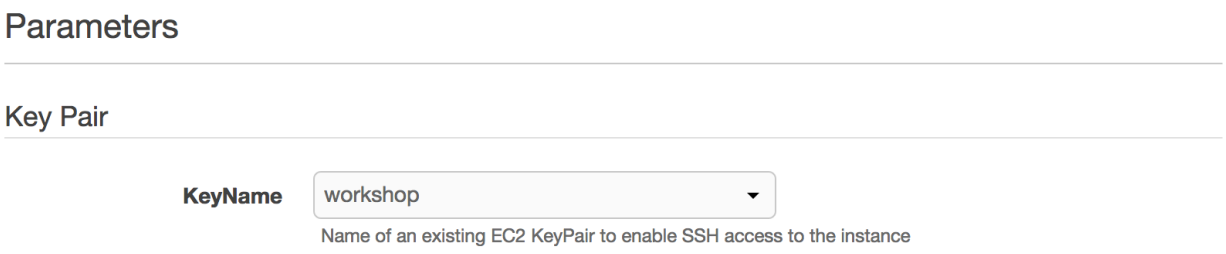

Figure 9: *AWS CloudFormation: Key Pair*

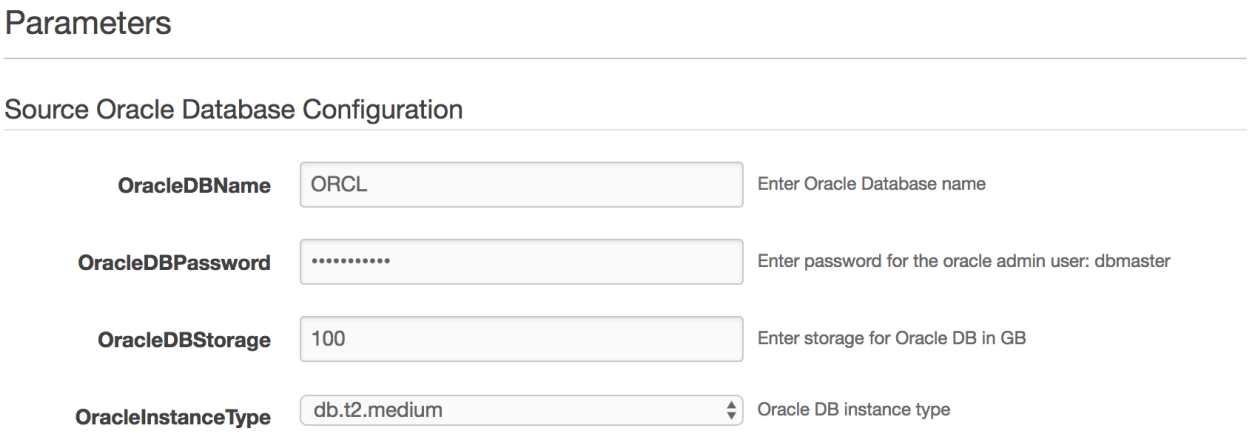

Figure 10: *AWS CloudFormation: Oracle*

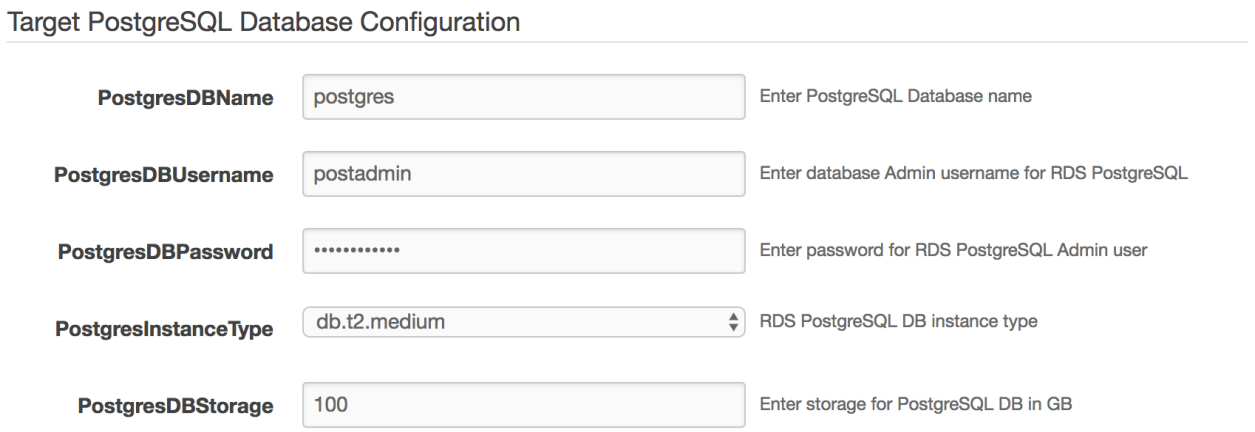

Figure 11: *AWS CloudFormation: Postgres*

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

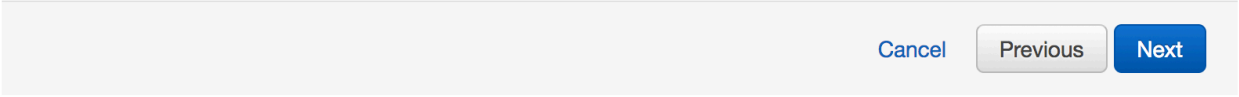

Figure 12: *AWS CloudFormation Stack Parameters: Confirmation*

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.**

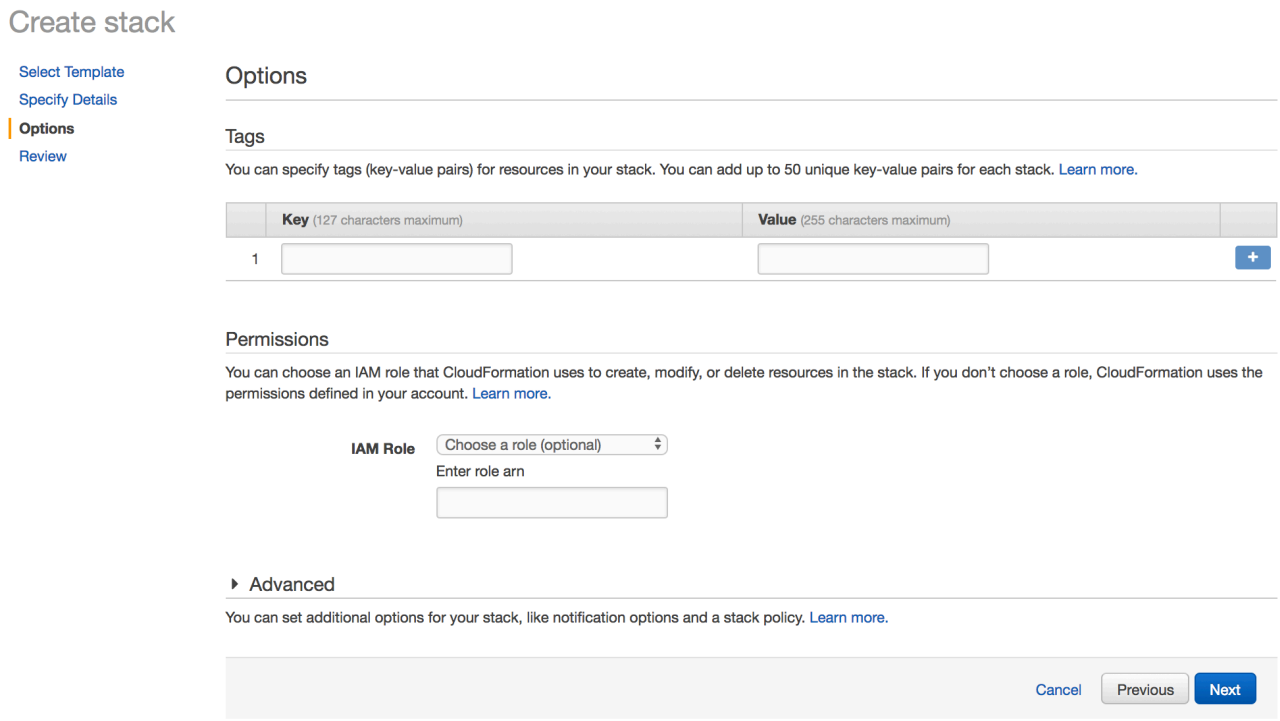

Figure 13: *AWS CloudFormation Stack Options: All*

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

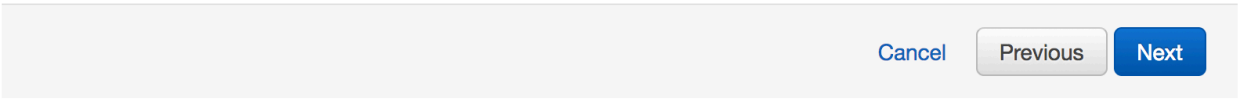

Figure 14: *AWS CloudFormation Stack Options: Confirmation*

2.2.4 AWS CloudFormation Stack Review

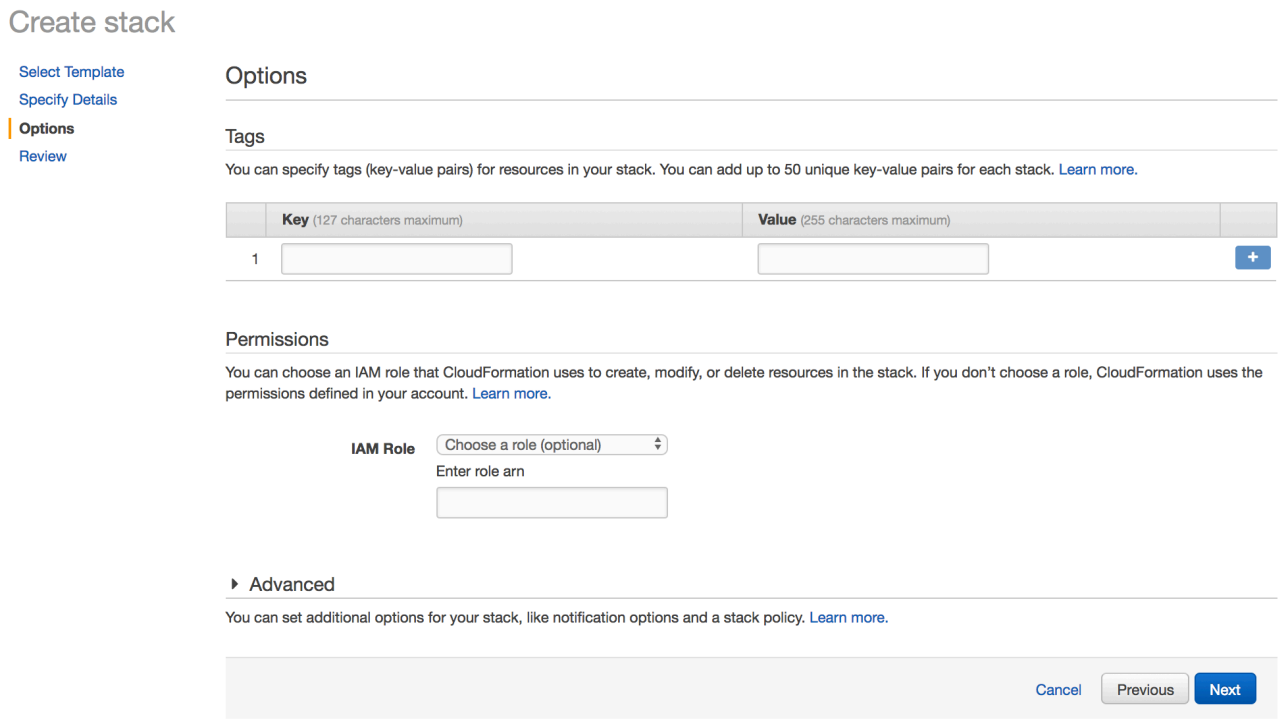

Figure 15: *AWS CloudFormation Stack: Review*

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

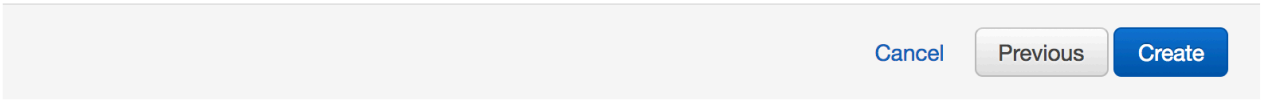

Figure 16: *AWS CloudFormation Stack Review: Creation*

All AWS resources associated with this workshop lab will now be 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.

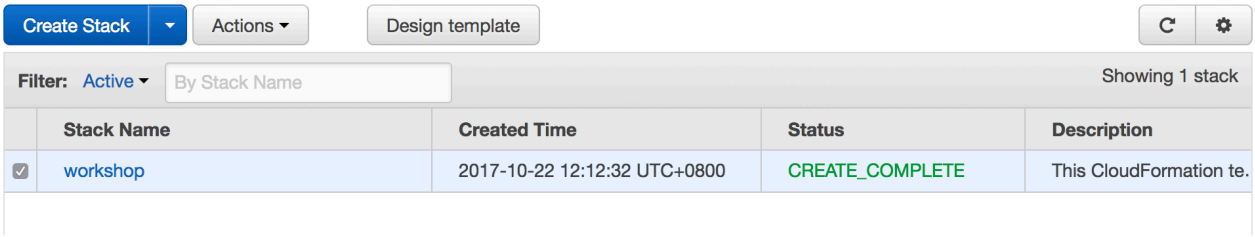

Figure 17: *AWS CloudFormation Stack: List Stacks*

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.

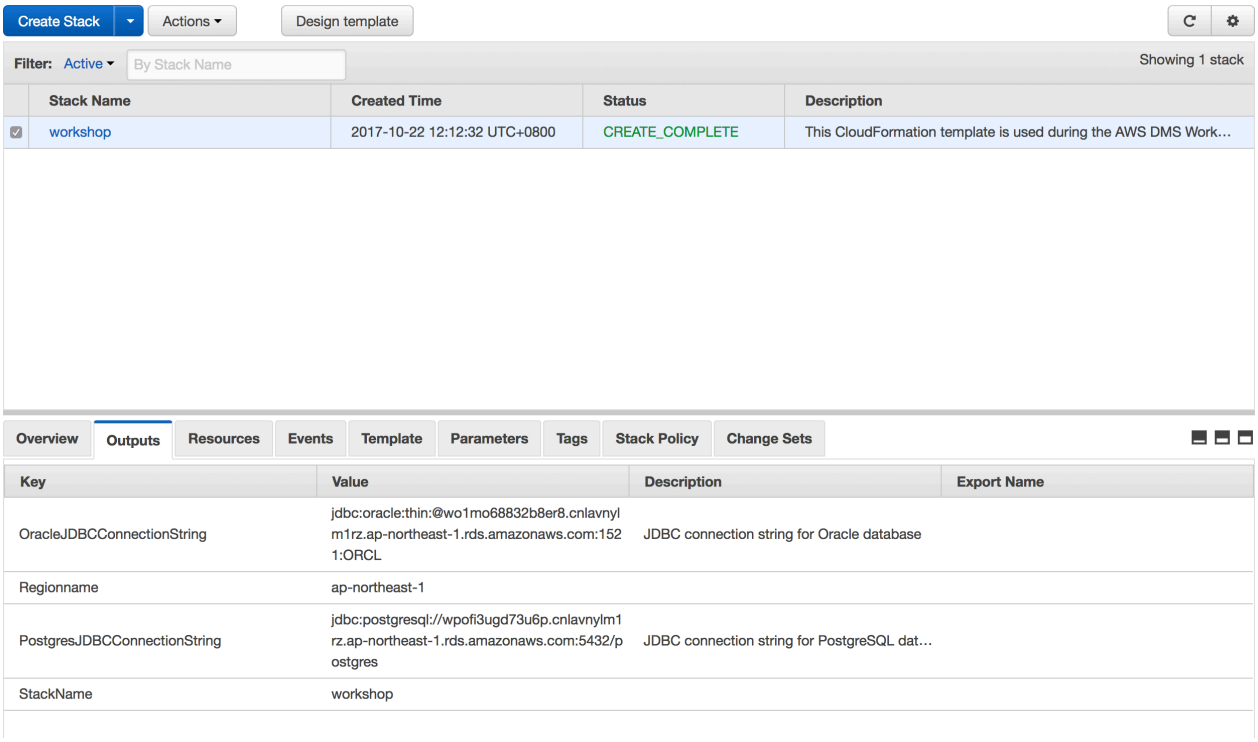

Figure 18: *AWS CloudFormation Stack: Outputs*

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:

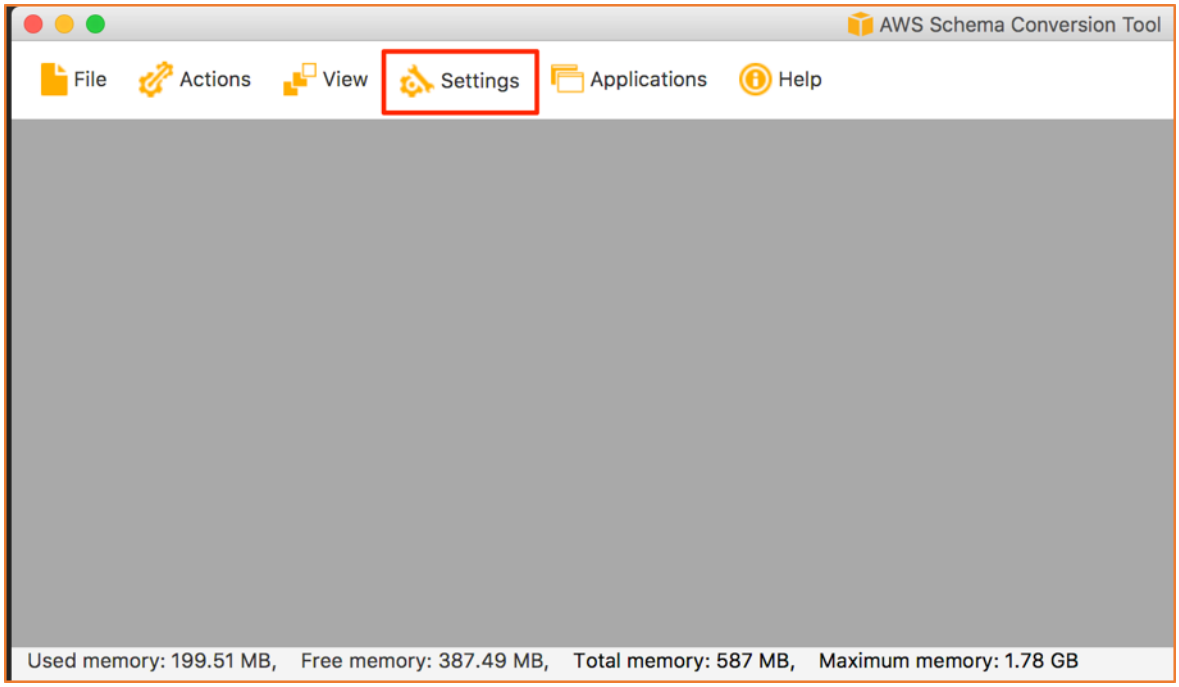

Figure 19: *AWS Schema Conversion Tool: Initialized*

Click on **Settings** > **Global Settings**

Now you should see the following Global Settings dialog:

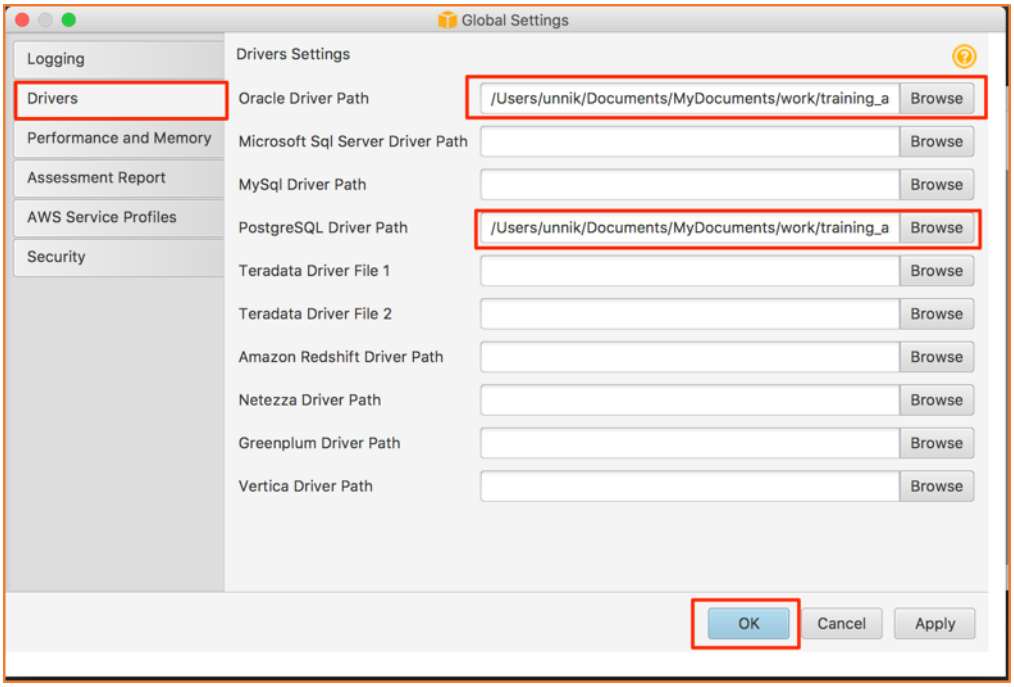

Figure 20: *AWS Schema Conversion Tool: Global Settings*

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 PostgresSQL jar file
- Click **OK** to Proceed

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 acces 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:

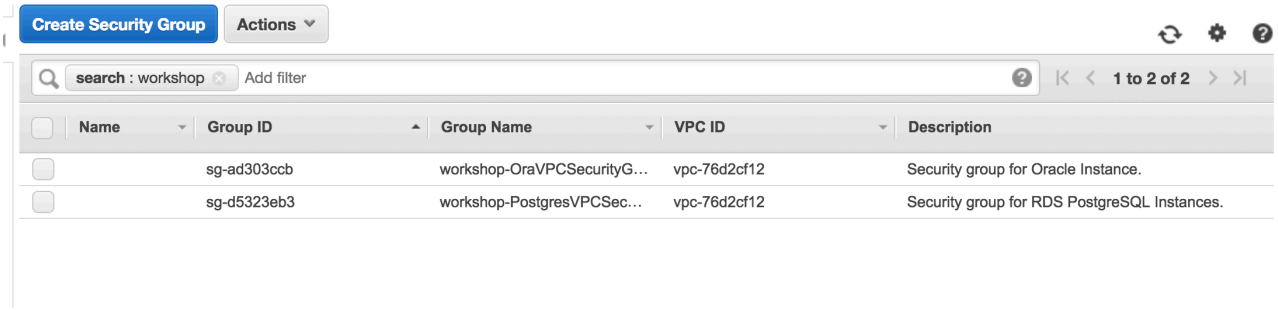

Figure 21: *AWS Schema Conversion Tool: List of Security Groups*

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**

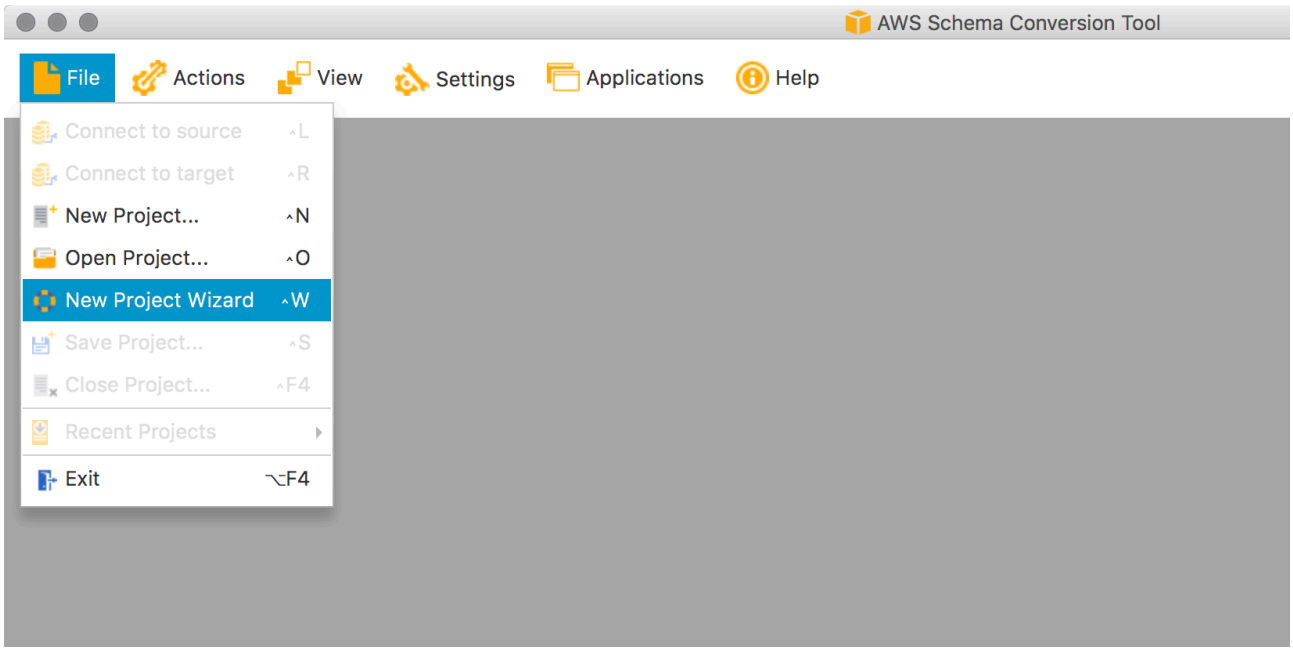

Figure 22: *AWS SCT New Project Wizard*

3.1.1 New Project Wizard – Select Source

Enter the following in the dialog displayed:

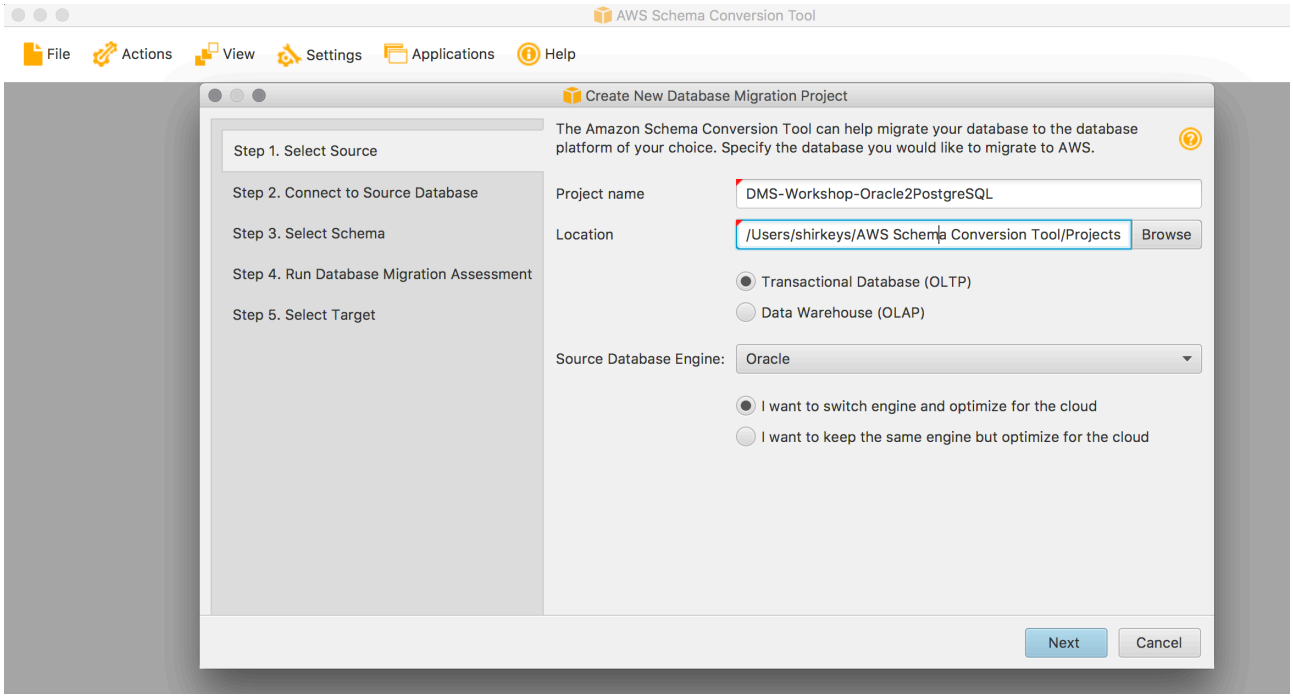

Figure 23: *AWS SCT New Project Wizard - Step 1*

- **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)

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.

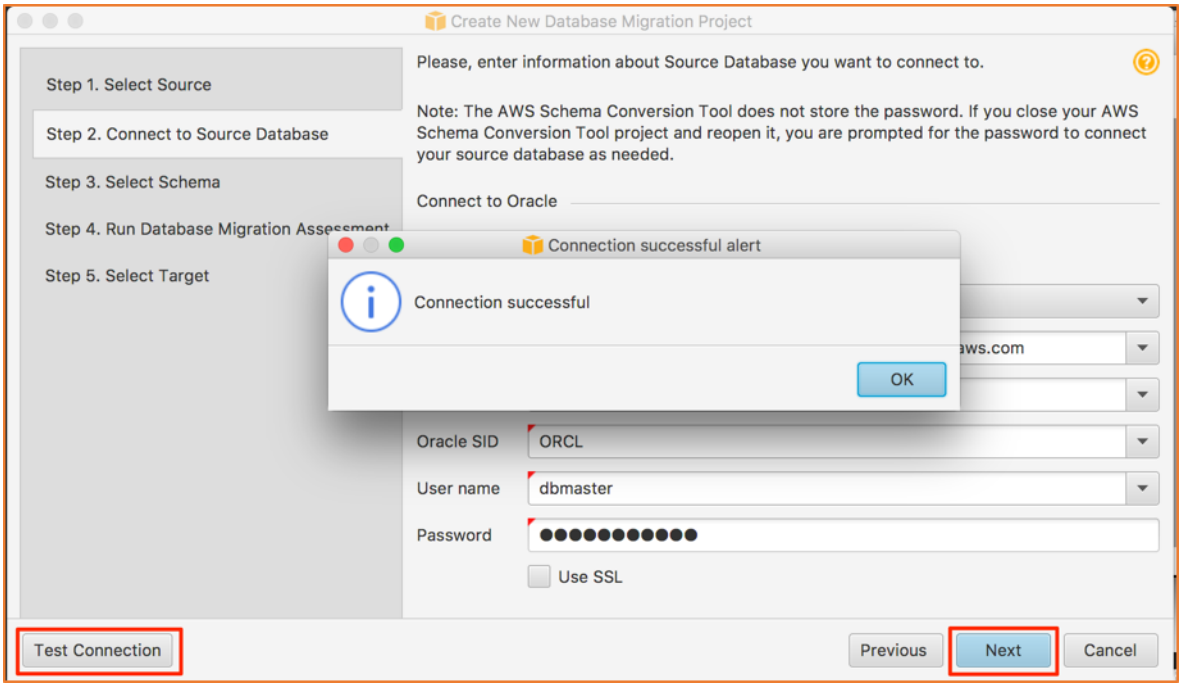

Figure 24: *AWS SCT New Project Wizard - Test Source Connection*

3.1.3 New Project Wizard - Select Schema

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

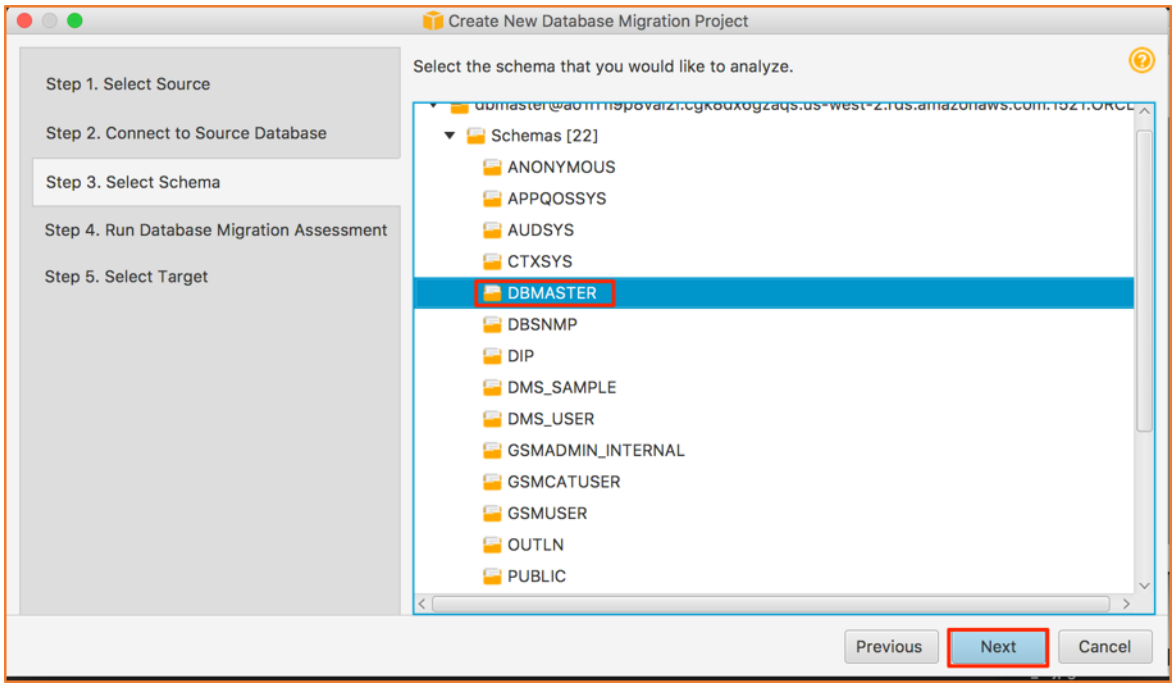

Figure 25: *AWS SCT New Project Wizard - Select Schema*

3.1.4 New Project Wizard - Run Database Migration Assessment

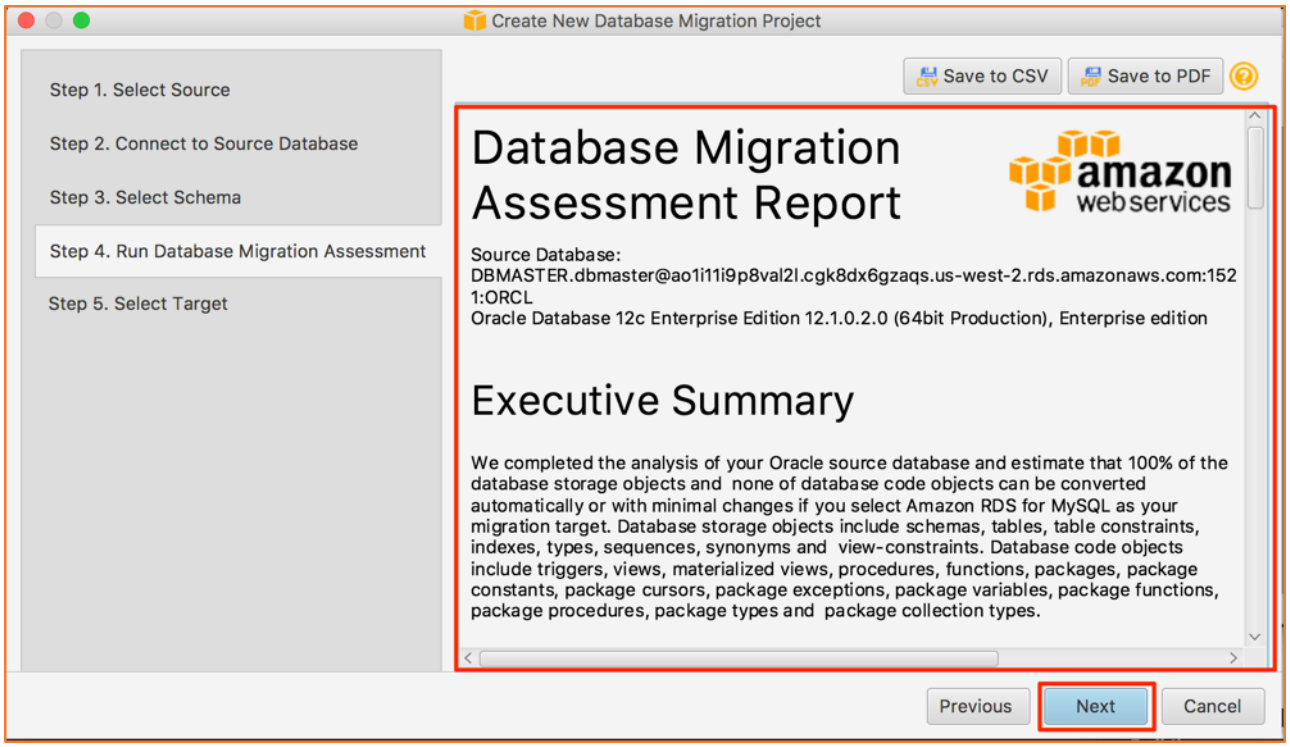

Figure 26: *AWS SCT New Project Wizard - Database Migration Assessment*

3.1.5 New Project Wizard - Select Target

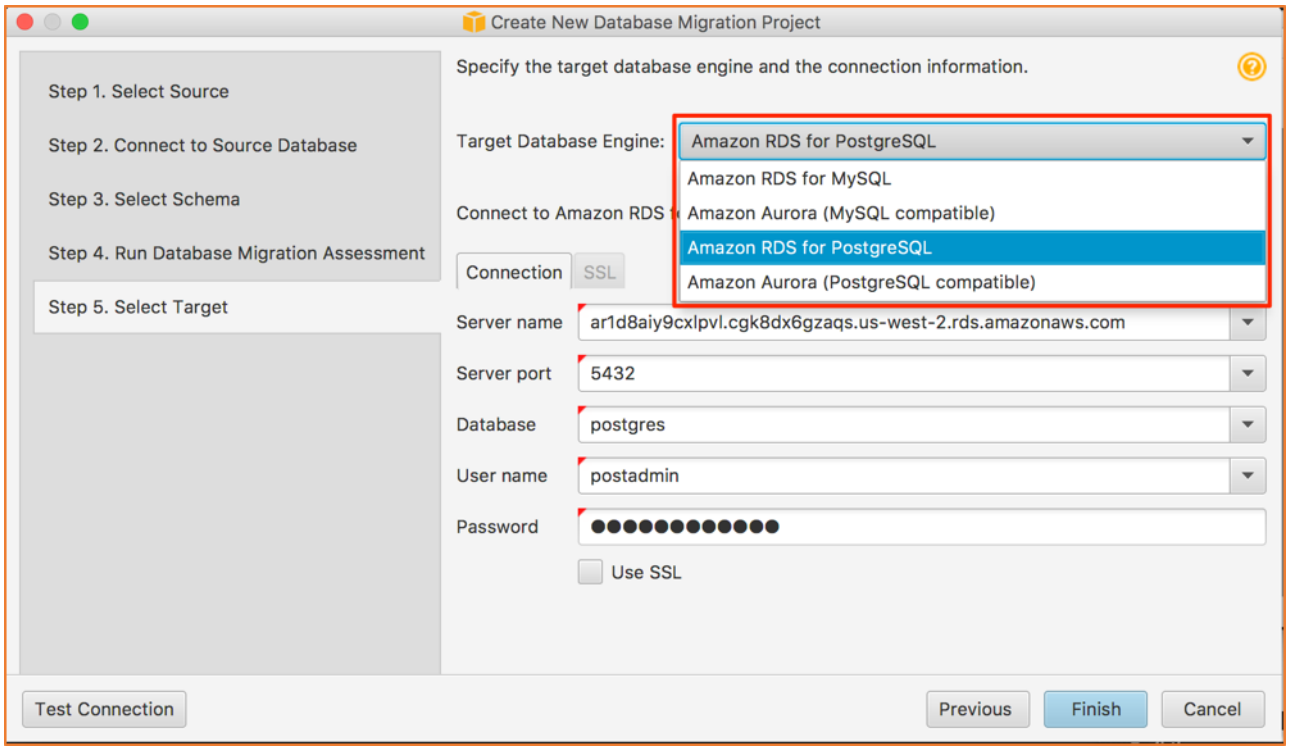

Figure 27: *AWS SCT New Project Wizard - Select Target*

3.1.6 New Project Wizard - Test Target Connection

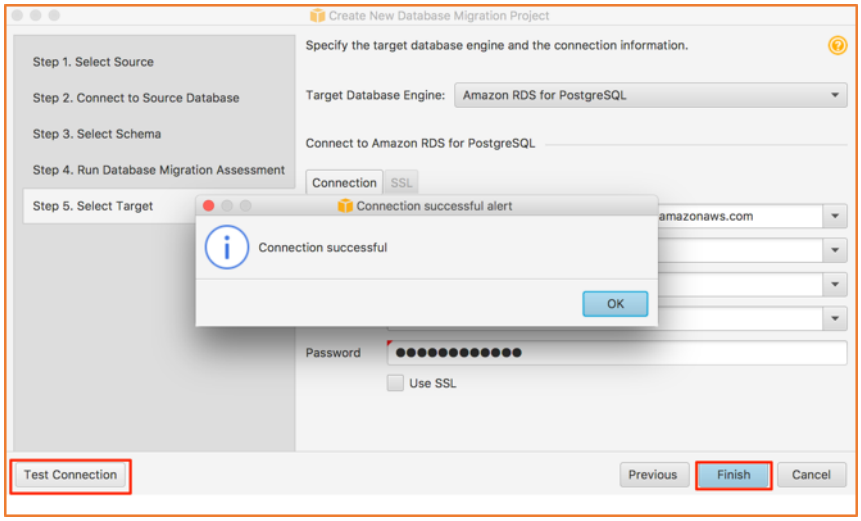

Figure 28: *AWS SCT New Project Wizard - Test Target Connection*

3.2 Convert Schema in AWS SCT

Review the project screen:

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

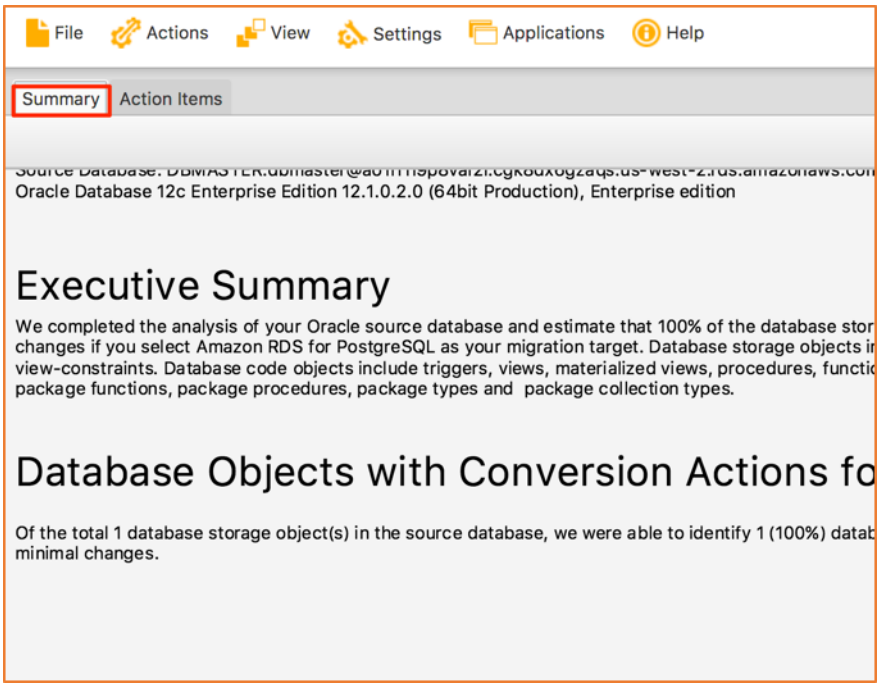

Figure 29: *Convert Schema - Summary*

3.2.1 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.

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

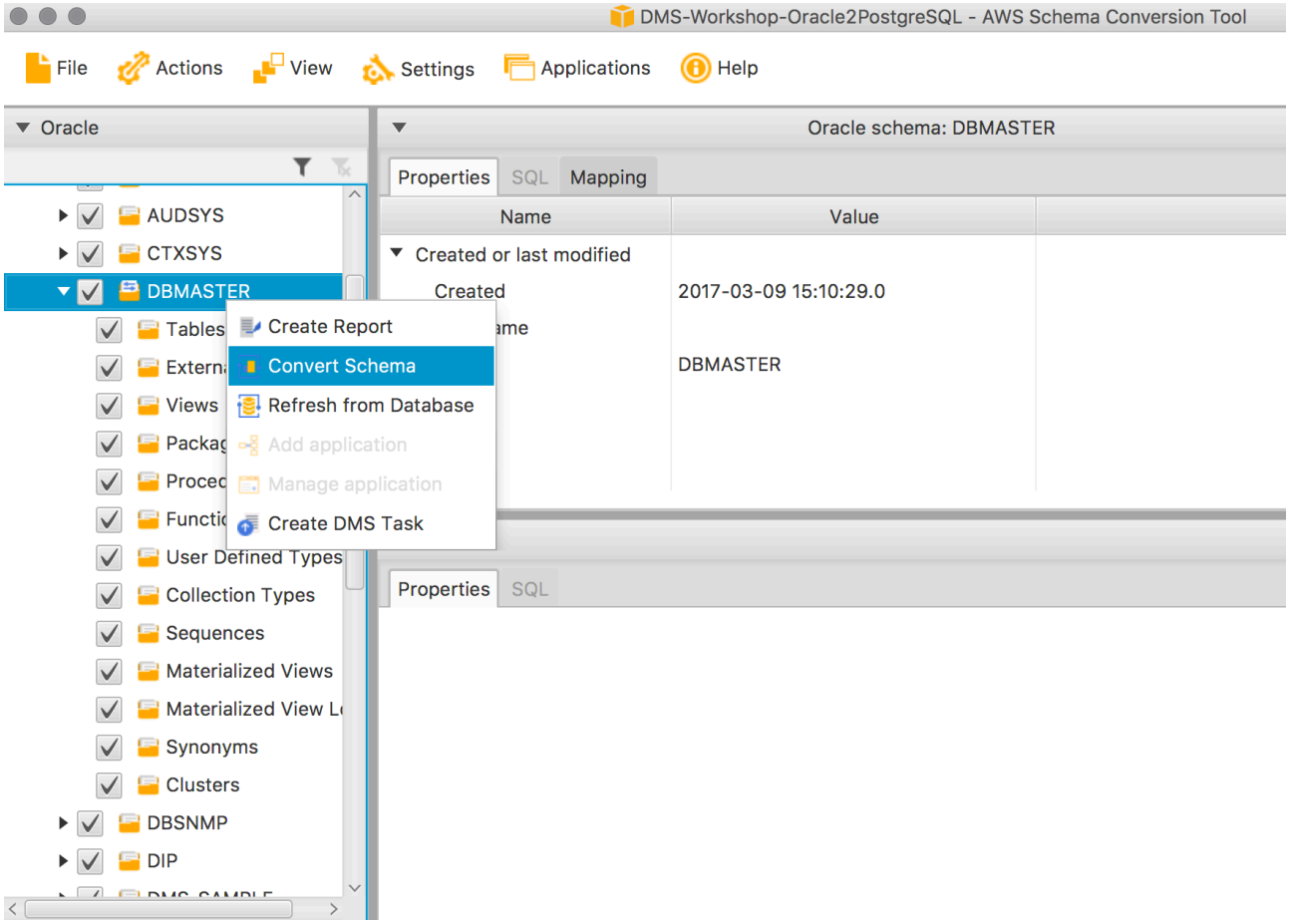

Figure 30: *Convert Schema - Execute*

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

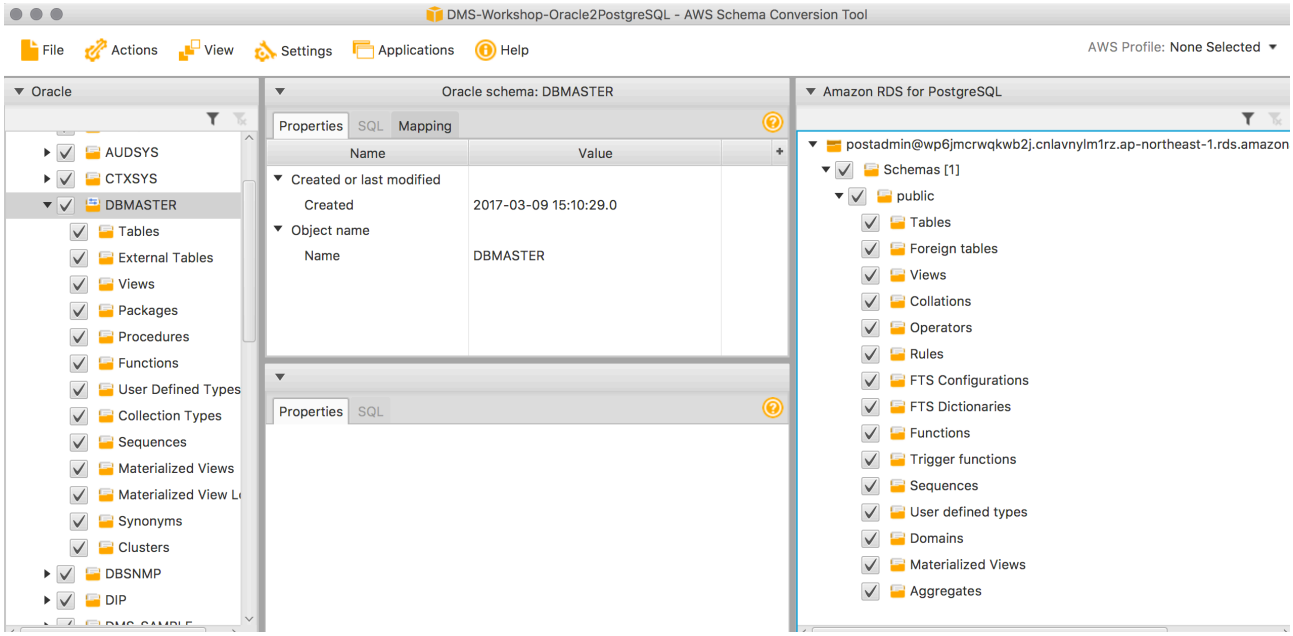

Figure 31: *Convert Schema - Target Review*

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

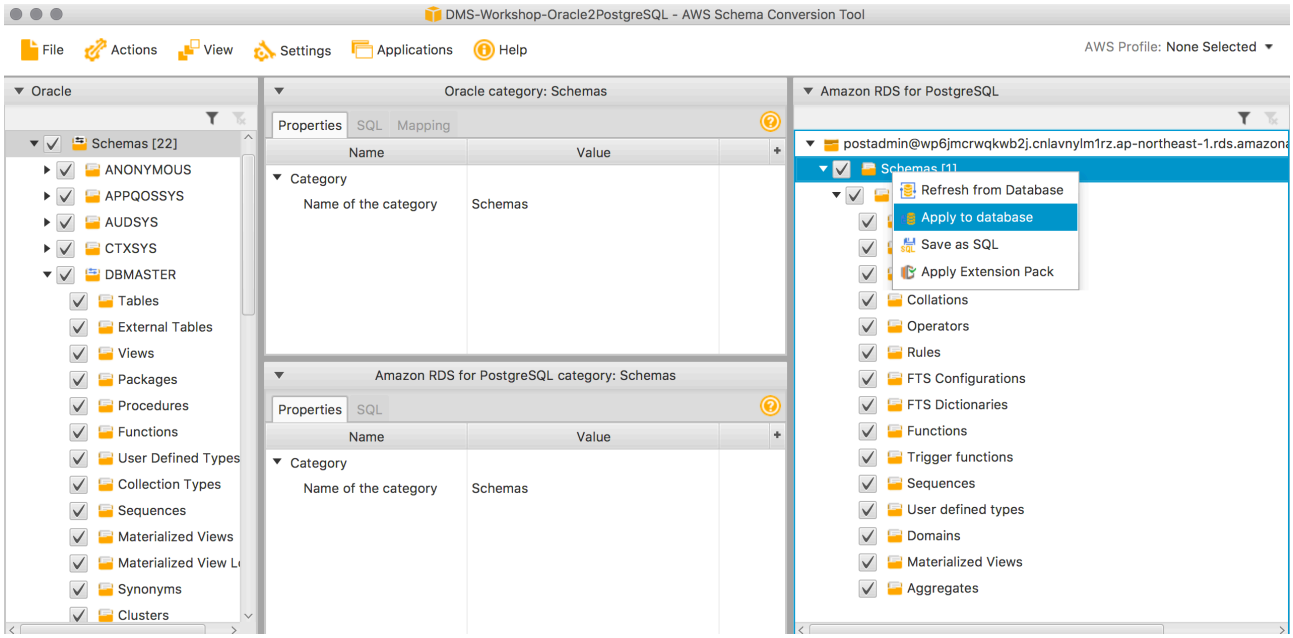

Figure 32: *Convert Schema - Target Apply*

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.

4 Lab Teardown

4.1 Teardown AWS CloudFormation Stack