



Workshop

**Migrate Your On-Premises Data Warehouse to
Amazon Redshift with AWS SCT**

Table of Contents

<u>LAB OVERVIEW</u>	<u>3</u>
<u>LAUNCH CLOUDFORMATION TEMPLATE TO SETUP ENVIRONMENT IN IRELAND.....</u>	<u>4</u>
<u>CONNECT TO YOUR ENVIRONMENT</u>	<u>9</u>
<u>INSTALL AWS SCHEMA CONVERSION TOOL AND VERIFY CONNECTIVITY TO ORACLE RDS AND AMAZON REDSHIFT FROM AWS SCT.....</u>	<u>11</u>
<u>VALIDATE THE SCHEMA CONVERSION AND WORK THROUGH CONVERSION ISSUES.....</u>	<u>24</u>
<u>CREATE POLICY.</u>	<u>29</u>
<u>ENSURE YOU SAVE THESE CREDENTIALS FOR SCT. ONCE YOU SELECT CLOSE THE SECRET KEY WILL NO LONGER BE AVAILABLE.</u>	<u>31</u>
<u>STEPS TO CREATE A GLOBAL PROFILE:</u>	<u>32</u>
<u>STEPS FOR SSH LOGIN TO AGENT EXTRACTOR HOSTS</u>	<u>35</u>
<u>STEPS FOR INSTALLING AND CONFIGURING THE SCT EXTRACTION AGENTS ON HOST 2.....</u>	<u>37</u>
<u>USING RANGE VIRTUAL PARTITIONING TO MIGRATE THE PERSON TABLE.....</u>	<u>47</u>
<u>VERIFY THAT YOUR DATA MIGRATION COMPLETED SUCCESSFULLY</u>	<u>51</u>

Lab Overview

Migrating your Oracle data warehouse to Amazon Redshift can substantially improve query and data load performance, increase scalability, and save costs. Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse that makes it simple and cost-effective to analyze all your data using your existing business intelligence tools. AWS Schema Conversion Tool (SCT) and AWS SCT Data Extraction agents make it easier to migrate your schema and data from your Oracle data warehouse, both on-premises and on AWS, to Amazon Redshift without disruption to the applications that rely on the data source.

The AWS SCT makes heterogeneous database migrations easy by automatically converting the source database schema and a majority of the custom code to a format compatible with the target database. The custom code that the tool converts includes views, stored procedures, and functions. Any code that the tool cannot convert automatically is clearly marked so that you can convert it yourself.

The AWS SCT Data Extraction Agent helps you to extract data from your on-premises data warehouse and migrate it to Amazon Redshift easily and securely. The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database. Agents extracts your data and uploads the data to either Amazon S3 or AWS Snowball devices. You can then use AWS SCT to copy the data to Amazon Redshift. The service supports homogenous migrations such as SQL Server to SQL Server, as well as heterogeneous migrations between different database platforms, such as SQL Server to Amazon Aurora or Oracle to Redshift. AWS DMS can also be used for continuous data replication with high-availability.

This lab will walk you through migrating a sample Oracle data warehouse to Amazon Redshift. We will first setup the environment using a CloudFormation template. The CloudFormation template will create a RDS Oracle data warehouse and Redshift cluster. The CloudFormation template will also create three EC2 instance. The EC2 Windows machine will have the AWS SCT installer and drivers for Oracle and Redshift in the Workshop folder on the desktop. You will need to install the SCT on the EC2 instance per the instructions in this document. The other two EC2 Linux machine will have the AWS SCT Data Extraction agents RPM in the /home/ec2-users/agents folder. You will need to install the agents and configure the agents to extract your data and move to Amazon Redshift.

Launch CloudFormation template to setup environment in Ireland

In this lab we will first create all the AWS resources needed for the workshop using a CloudFormation template. Please download the zip file provided here

<https://tiny.amazon.com/1f34lh7i0/D1S04>

This zip file contains the following:

- CloudFormation template
- Lab guide
- SQL file for copying and pasting SQL's
- PowerPoint presentation

Unzip the file on your local drive.

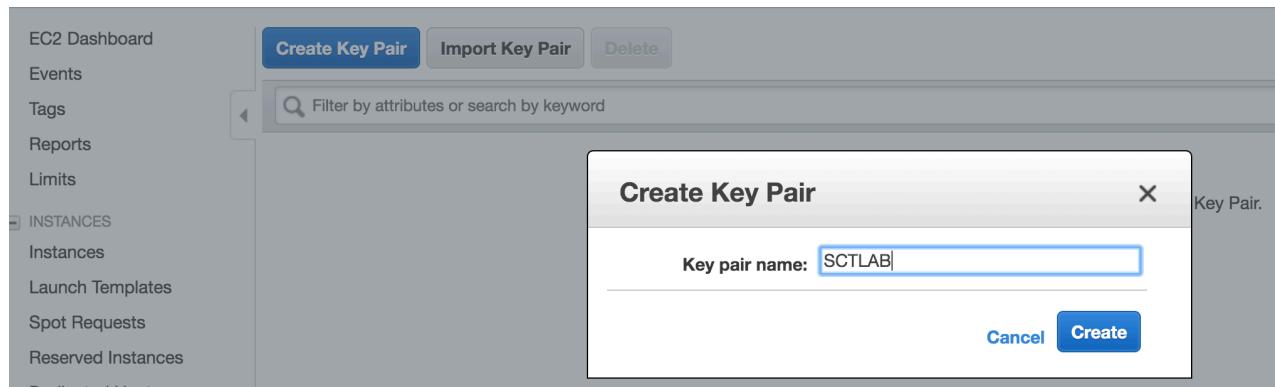
The CloudFormation template will create the following:

- 1) An Oracle RDS Data warehouse. The Oracle data warehouse contains ticketing data related to football and baseball games.
The data warehouse contains 5 dimension tables; Dim_Sport_locaton_Seats, Dim_Person, Dim_Sporting_Event, Dim_Sport_Team, Dim_Player; and one fact table; Fact_Sport_Event_Ticket_Info. This data warehouse also contains other database objects.
- 2) An EC2 Windows instance with SCT installation msi, drivers for Oracle and Redshift and SQL Workbench on the Desktop.
- 3) An Amazon Redshift 1 node cluster – this will be used as a target for the migration.
- 4) Two EC2 Linux instance with SCT Data extraction agent for extracting and moving the data to Amazon Redshift.

Note: - This lab should be run in the Ireland region. Please make sure you select Ireland region before you go to the next steps outlined below.

1. If you do not have a key in the **Ireland region**, please follow these steps else skip to step2.

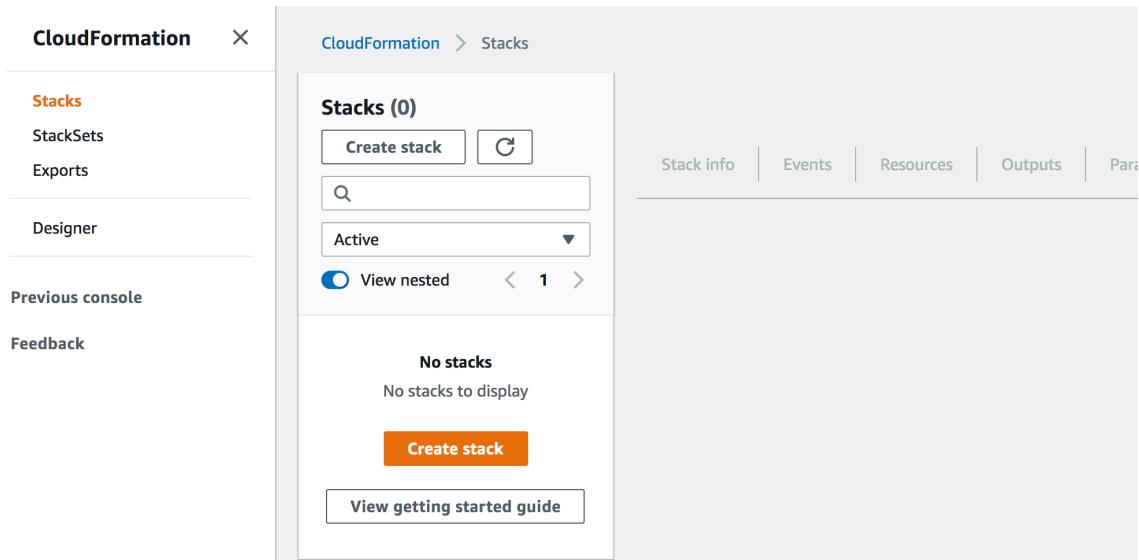
- Open the AWS Management console at
<https://console.aws.amazon.com/ec2/v2/home>
- Click on **Key Pairs** on the left-hand menu and then click on **Create Key Pair**.
- Give your key pair a name e.g. **SCTLAB** and click **Create**.
- A dialog box will open asking you to save the key file. Click on Save File to save the key. Remember where you saved the key or move it to another folder of your choice. You will need this later.



2. Open the CloudFormation console at:

<https://console.aws.amazon.com/cloudformation/home>

- Click **Create Stack** in the left-hand side.



3. Click **Choose File** and select **Upload a template to Amazon S3** and **Browse** to where you downloaded the Zip file.

4. Select the CloudFormation template called **D1S04-MigrationCloudFormation.json** and then click **Next**.

Step 1
Specify template

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Create stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready Use a sample template Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL Upload a template file

Upload a template file
Choose file D1S04-MigrationCloudFormation.json
JSON or YAML formatted file

S3 URL: <https://s3-eu-west-1.amazonaws.com/cf-templates-1xg5b2oyrp9r9-eu-west-1/2019170aTJ-D1S04-MigrationCloudFormation.json>

Cancel

5. Populate the form with the values specified below and then click **Next**.

Stack Name	Any name you would like to give the stack. Recommended name LoftD1S04 . Please append your initials if multiple people are running the lab in the same account.
KeyName	Select a key in Ireland region from the dropdown. If you created a key in step 1 then select that key or else select another key already existing in your account.

CloudFormation > Stacks > Create stack

Step 1
Specify template

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Specify stack details

Stack name

Stack name
 Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

KeyName

Enter the name of your AWS key pair

Cancel Previous Next

6. On the **Options** page, put the following values for Tags and click **Next**. All other values can remain as default.

Key = Environment Value = D1S04

CloudFormation > Stacks > Create stack

Step 1
Specify template

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Configure stack options

Tags

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

<input type="text" value="Environment"/>	<input type="text" value="D1S04"/>	Remove
Add tag		

Permissions

Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more](#).

IAM role - optional

Choose the IAM role for CloudFormation to use for all operations performed on the stack.

IAM role name	<input type="text" value="Sample-role-name"/>	Remove
-------------------------------	---	------------------------

7. On the **Review** page, at the bottom of the screen please make sure you
- Check the box '**I acknowledge that AWS CloudFormation might create IAM resources with custom names.**'
 - Click **Create**.

The screenshot shows the 'Review' step of creating a new AWS CloudFormation stack. The top navigation bar includes 'CloudFormation > Stacks > Create stack'. The main area is titled 'Review LoftD1S04'. On the left, a vertical sidebar lists steps: Step 1 'Specify template' (selected), Step 2 'Specify stack details', Step 3 'Configure stack options', and Step 4 'Review' (disabled). The 'Step 1: Specify template' section contains a 'Template' card with a 'Template URL' field showing a specific S3 URL and a 'Stack description' field containing a detailed note about the template's purpose. The 'Step 2: Specify stack details' section contains a 'Parameters (1)' table with one row: 'KeyName' set to 'SCLAB'. Below these sections is a 'Capabilities' section with a note about IAM resource creation and a checked checkbox for acknowledging it. At the bottom are buttons for 'Cancel', 'Previous', 'Create change set', and a large orange 'Create stack' button.

Step 1: Specify template

Template

Template URL
<https://s3-eu-west-1.amazonaws.com/cf-templates-1xg5b2oyrp9r9-eu-west-1/2019170aTj-D1S04-MigrationCloudFormation.json>

Stack description
This template creates resources necessary for Relnvent Workshop ANT371 - Migrating Your Oracle Warehouse to Amazon Redshift Using AWS DMS and SCT.

Estimate cost not available

Step 2: Specify stack details

Parameters (1)

Key	Value
KeyName	SCLAB

Capabilities

i The following resource(s) require capabilities: [AWS::IAM::Role]

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

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

Cancel Previous Create change set **Create stack**

8. At this point, you will be directed back to the CloudFormation console and will see a status of **CREATE_IN_PROGRESS**. Do not continue until the status changes to **CREATE_COMPLETE**.

9. Once CloudFormation has completed building the stack, the status will change to **create_complete**. Make sure to note the values for the following keys in the **OUTPUTS** tab. These will be needed as input for SCT and DMS setup.

LoftD1S04RedshiftEndpoint	End point for the Redshift cluster
LoftD1S04VpcId	VPC ID of the newly created VPC
LoftD1S04OracleEndpoint	End point of the source RDS database
LoftD1S04SCTDNS	End point for the EC2 instance with SCT and drivers installed.
LoftD1S04AGENT1PublicDNS	Public end point for first EC2 instance with SCT Extraction agent.
LoftD1S04AGENT1PrivateIP	Private IP for first EC2 instance with SCT Extraction agent.
LoftD1S04AGENT2PublicDNS	End point for second EC2 instance with SCT Extraction agent.
LoftD1S04AGENT2PrivateIP	Private IP for second EC2 instance with SCT Extraction agent.
LoftD1S04S3Bucket	S3 Bucket name

Connect to your environment

Now that the environment has been built, the next step is to connect to the EC2 instance.

10. Once CloudFormation has completed building the stack,
 - Go to the EC2 console located at: <https://console.aws.amazon.com/ec2/v2/home> and
 - click **Instances** in the left column.

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'Instances' section, 'Instances' is selected. The main content area displays 'Resources' and 'Service Health'. Under 'Resources', it lists 3 Running Instances, 1 Elastic IP, 0 Dedicated Hosts, 4 Snapshots, 7 Volumes, 0 Load Balancers, 3 Key Pairs, 30 Security Groups, and 0 Placement Groups. Under 'Service Health', 'Service Status' shows 'US East (N. Virginia)' with 'No events'. 'Availability Zone Status' shows 'us-east-1a' and 'us-east-1b' both operating normally. The right sidebar contains sections for 'Account Attributes', 'Additional Information', and 'AWS Marketplace'.

11. Select the instance with LoftD1S04-SCTWinEC2 (or whatever name you gave your stack) in its name and obtain the Public IP Address from the instance.

The screenshot shows the AWS EC2 Instances page. The left sidebar shows 'Instances' selected. The main table lists three instances: LoftD1S04-LoftD1S04AGENTINSTANCE-1, LoftD1S04-LoftD1S04AGENTINSTANCE-2, and LoftD1S04-SCTWinEC2. The third instance is selected. Below the table, the instance details for 'LoftD1S04-SCTWinEC2' are shown, including its description, status checks, monitoring, and tags. The instance has an Instance ID of i-0ab5f6cbfc6d12ce3, is running, and is in the eu-west-1a availability zone. It has a Public DNS of ec2-18-202-175-154.eu-west-1.compute.amazonaws.com and a Public IP of 18.202.175.154.

12. Click on the **Connect** button to popup the window as shown below and click on Download Remote Desktop File button to download the RDP file.

Note: You **do not** need to use the '**Get Password**' button for this lab. The password is provided in step 13.

Connect To Your Instance

X

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

[Download Remote Desktop File](#)

When prompted, connect to your instance using the following details:

Public DNS ec2-18-202-175-154.eu-west-1.compute.amazonaws.com

User name Administrator

Password [Get Password](#)

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

[Close](#)

13. Using the RDP tool in Mac/Windows, open the RDP file downloaded.

- a. Use the username '**Administrator**' if it's blank and
- b. the password as "**DWMigration@1**".

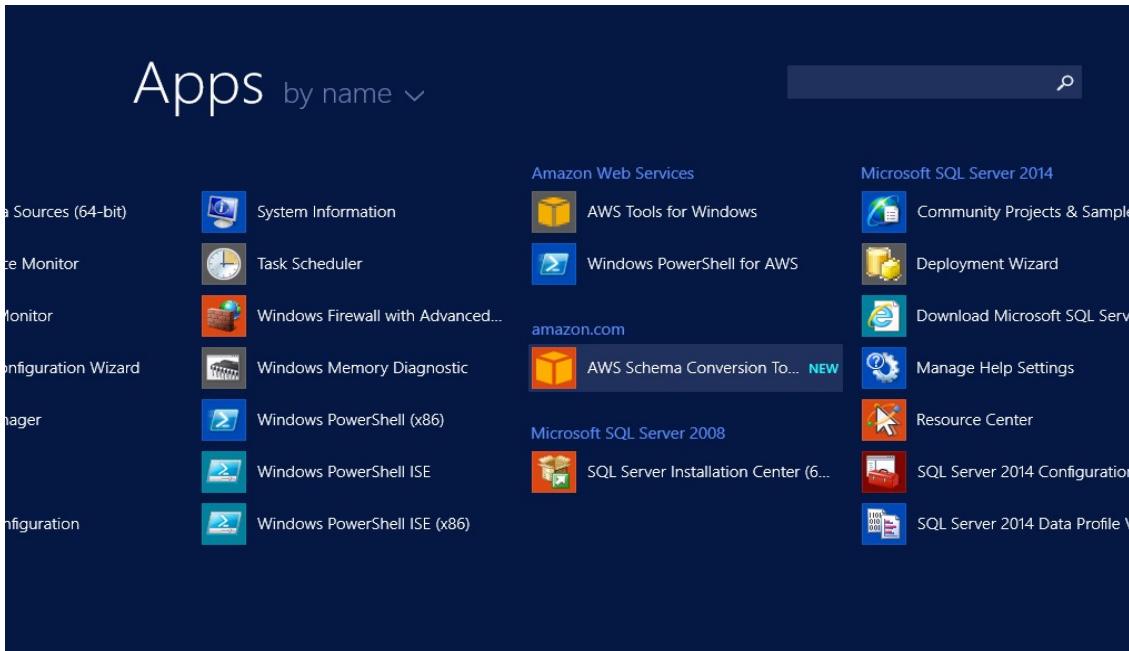
Install AWS schema Conversion Tool and verify connectivity to Oracle RDS and Amazon Redshift from AWS SCT

1. Once connected:

- Launch SCT installation msi from the **ANT371** folder on the desktop of the EC2 instance.
- You will find it under the ANT371 folder as '**AWS Schema Conversion Tool-1.0.619.msi**'.
- **Double click** it to launch Schema Conversion Tool installation.

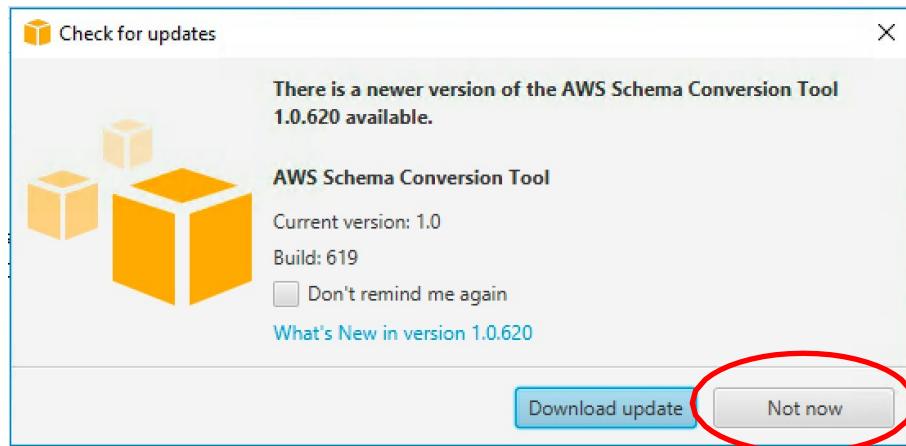
2. After clicking AWS Schema Conversion Tool,

- Wait for a minute for the tool to launch, hit Run and install AWS Schema Conversion Tool.
- Once installed, be aware there will be no confirmation message of the installation.
- Click on Windows Start button, and
- Click on Down Arrow to list Apps, drag the bar to the right until you see AWS Schema Conversion Tool as seen below:



3. Click on Schema Conversion Tool to open the application.

You may see a message that asks to download new version, click **Not now** button and proceed. If you do upgrade then your screen shots may vary from what we have in the lab document.



4. A New Database Migration Project page opens. If a new project does not open then

- click on **File and New Project**.
- Provide a Project Name of your choice (recommended **LoftD1S04**) and
- leave Location default.
- Make sure you select **Data Warehouse (OLAP)** and Source Database Engine as **Oracle DW** and the Target Database Engine as **Amazon Redshift**.
- Click **OK**.

New project

Enter the name, location and type of the new migration project.

Project name: LoftD1S04

Location: C:\Users\Administrator\AWS Schema Conversion Tool\Projects [Browse](#)

Transactional database (OLTP)

Data warehouse (OLAP)

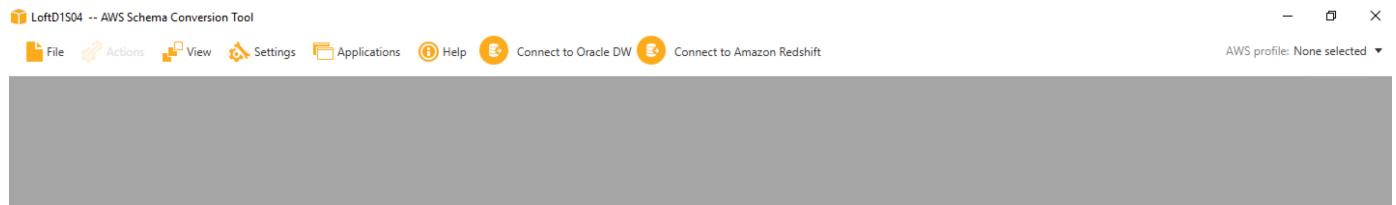
NoSQL database

Source database engine: Oracle DW

Target database engine: Amazon Redshift

[OK](#) [Cancel](#)

5. Click on **Connect to Oracle DW** on the top menu.

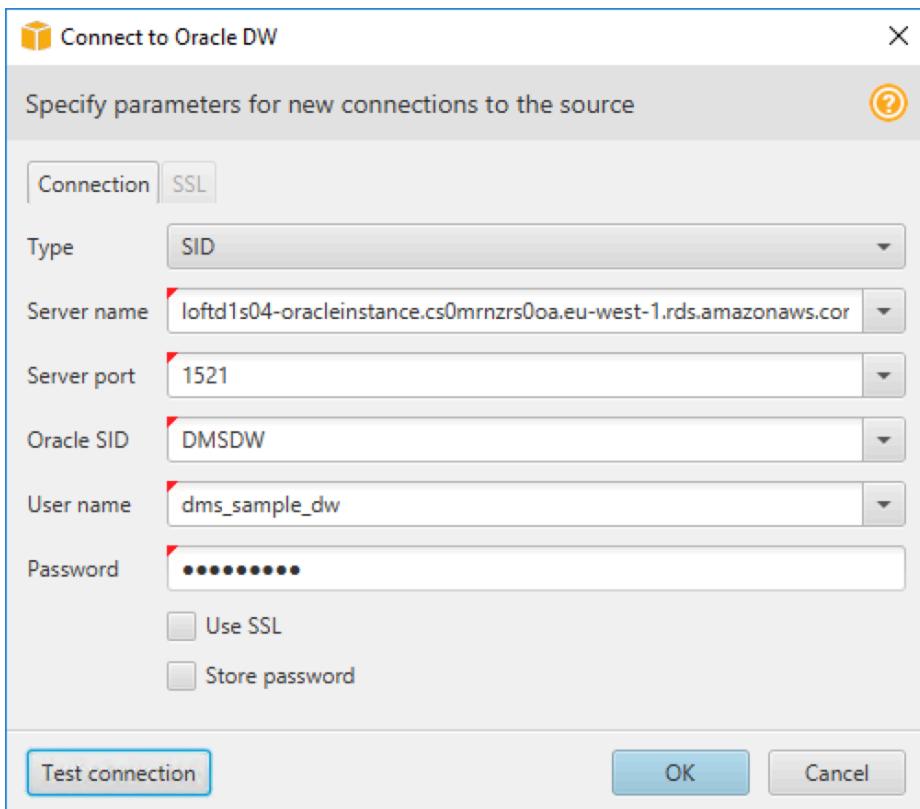


6. Enter the following details and click **Test Connection**.

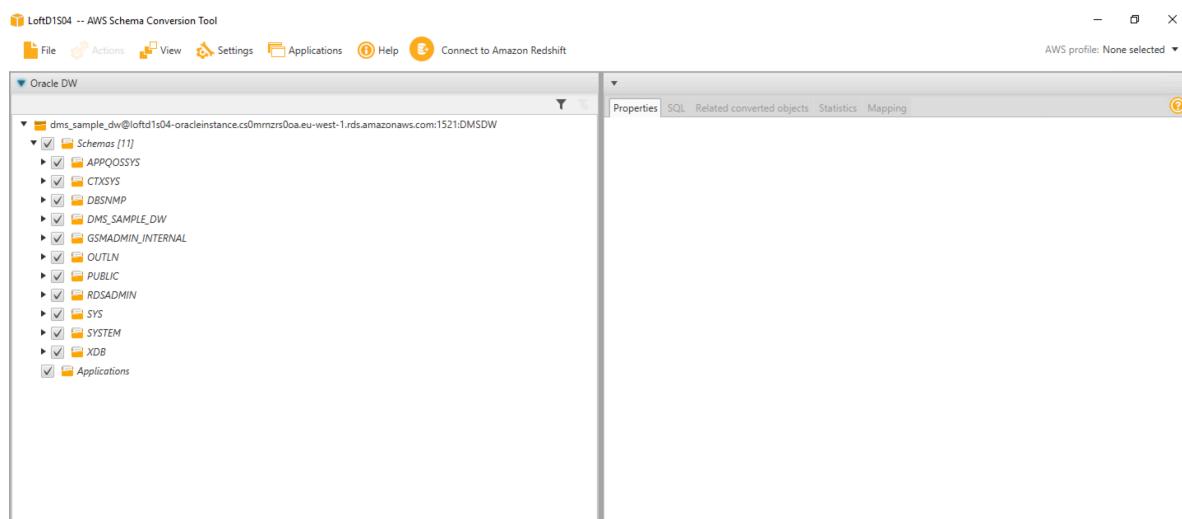
It should come back with a message '**Connection successful**'.

Then click **OK**. Update screen shot

Type	SID
Server name	Take this from the output of the CloudFormation template in Step 9. It is the LoftD1S04OracleEndpoint output field.
Server port	1521
Oracle SID	DMSDW
User name	dms_sample_dw
Password	Password1
Oracle Driver Path	Specify the Oracle driver path as C:\Users\Administrator\Desktop\ANT371\OracleDrivers\ojdbc7.jar



7. Now you should see the Oracle DW schema on your left as shown below.



8. Click on the **Connect to Amazon Redshift** on the top menu and put in the following details and click **Test Connection**. It should come back with a message 'Connection successful'. Then click OK.

Server name	Take this from the output of the CloudFormation template in Step 9. It is the LoftD1S04RedshiftEndpoint output field.
Server port	5439
Database	dmsdwttarget

User name	dms_sample_dw
Password	Password1
Amazon Redshift Driver Path	C:\Users\Administrator\Desktop\ANT371\RedshiftDrivers\Redshift JDBC42- 1.2.16.1027.jar

Connect to Amazon Redshift

Specify parameters for new connections to the target

Connection SSL

Server name: loftd1s04-redshifttargetcluster-1t0w9k68d259.cbbvwdxhrfgn.eu-west-1

Server port: 5439

Database: dmsdwtarget

User name: dms_sample_dw

Password: [REDACTED]

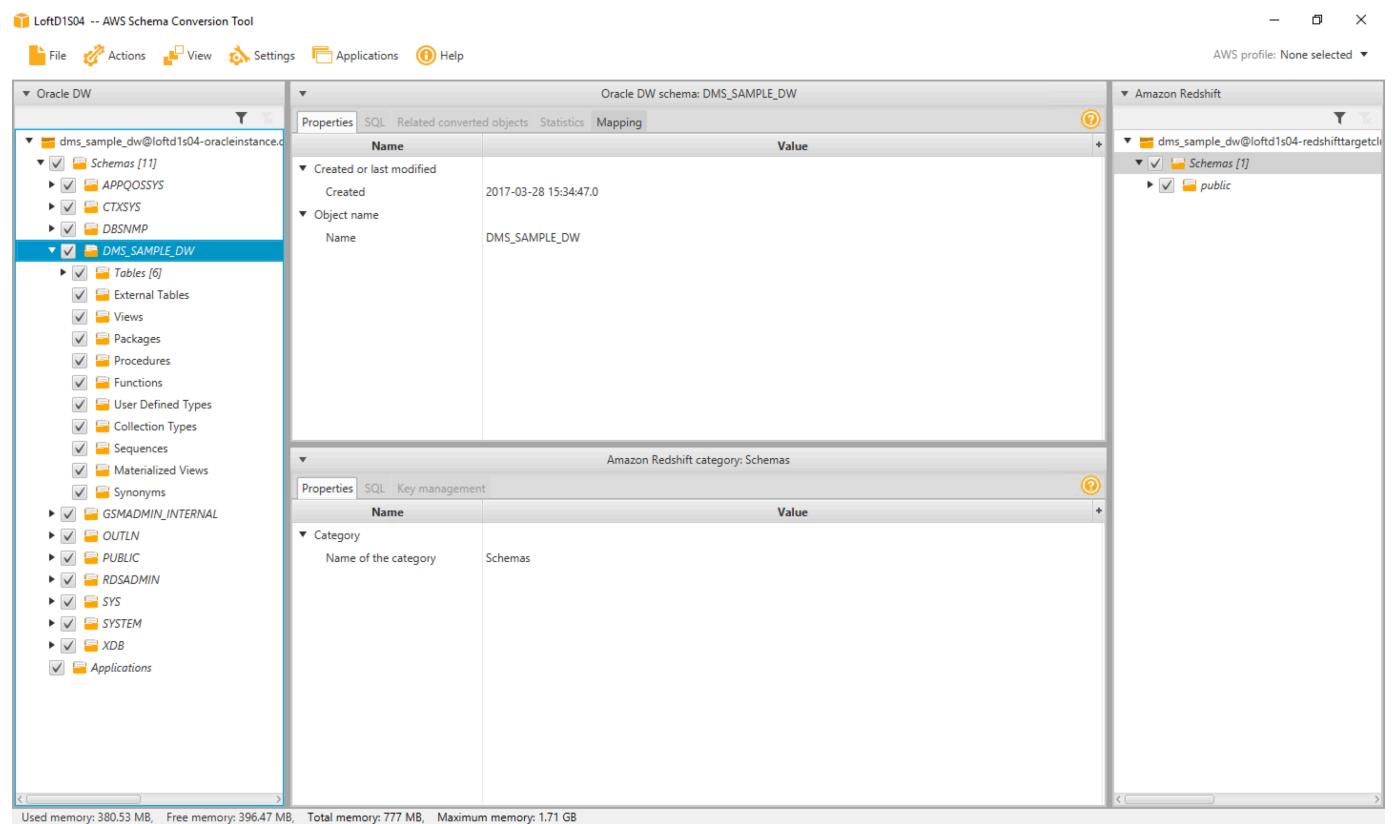
Use SSL

Store password

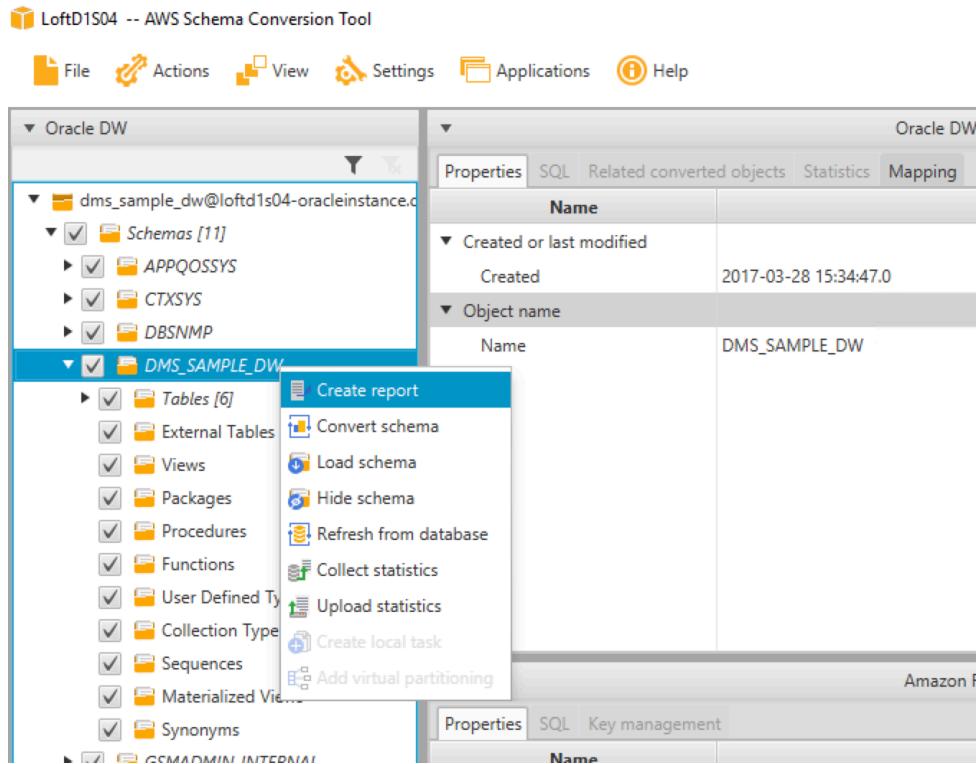
Test connection **OK** **Cancel**

Use AWS SCT to convert schema from source Oracle to target Amazon Redshift

1. Now you should see Redshift on the right hand side. Click on the dms_sample_dw schema on the left hand side in the Oracle DW. You may see a message saying loading metadata. Once that is done your screen should look like below.



- Right click on the **dms_sample_dw** schema in the left pane and click **Create Report**. This will create an assessment report.



- You will get a message saying that 'Your Optimization Settings require run-time statistics information from the source database. Would you like to load statistics?' Click **Continue**.

This will take about 30-40 secs and your Database Migration Assessment Report will be ready.

The AWS SCT creates a *database migration assessment report* to help you convert your schema. The database migration assessment report provides important information about the conversion of the schema from your source database to your target database instance. The report summarizes all of the schema conversion tasks and details the action items for schema that can't be converted to the DB engine of your target DB instance. The report also includes estimates of the amount of effort that it will take to write the equivalent code in your target DB instance that can't be converted automatically.

LoftD1s04 -- AWS Schema Conversion Tool

File Actions View Settings Applications Help

AWS profile: None selected

Summary Action items

Database migration assessment report

aws

Source database:
DMS_SAMPLE_DW.dms_sample_dw@loftd1s04-oracleinstance.cs0mrnzrsoa.eu-west-1.rds.amazonaws.com:1521:DMSDW
Oracle Database 12c Enterprise Edition 12.1.0.2.0 (64bit Production)

Executive summary

We completed the analysis of your Oracle DW source database and estimate that 100% of the database storage objects and none of database code objects can be converted automatically or with minimal changes if you select Amazon Redshift as your migration target. Database storage objects include schemas, tables, external tables, table constraints, indexes, partitions, types, collection types, sequences and synonyms. Database code objects include triggers, views, materialized views, procedures, functions, packages, package constants, package cursors, package exceptions, package variables, package functions and package procedures. Based on our analysis of SQL syntax elements of your source database schema, we estimate that 100% of your entire database schema can be converted to Amazon Redshift automatically.

Database objects with conversion actions for Amazon Redshift

Of the total 13 database storage object(s) in the source database, we identified 13 (100%) database storage object(s) that can be converted to Amazon Redshift automatically or with minimal changes.

LoftD1s04 -- AWS Schema Conversion Tool

File Actions View Settings Applications Help

AWS profile: None selected

Summary Action items

Database migration assessment report

aws

Source database:
DMS_SAMPLE_DW.dms_sample_dw@loftd1s04-oracleinstance.cs0mrnzrsoa.eu-west-1.rds.amazonaws.com:1521:DMSDW
Oracle Database 12c Enterprise Edition 12.1.0.2.0 (64bit Production)

Executive summary

We completed the analysis of your Oracle DW source database and estimate that 100% of the database storage objects and none of database code objects can be converted automatically or with minimal changes if you select Amazon Redshift as your migration target. Database storage objects include schemas, tables, external tables, table constraints, indexes, partitions, types, collection types, sequences and synonyms. Database code objects include triggers, views, materialized views, procedures, functions, packages, package constants, package cursors, package exceptions, package variables, package functions and package procedures. Based on our analysis of SQL syntax elements of your source database schema, we estimate that 100% of your entire database schema can be converted to Amazon Redshift automatically.

Database objects with conversion actions for Amazon Redshift

Of the total 13 database storage object(s) in the source database, we identified 13 (100%) database storage object(s) that can be converted to Amazon Redshift automatically or with minimal changes.

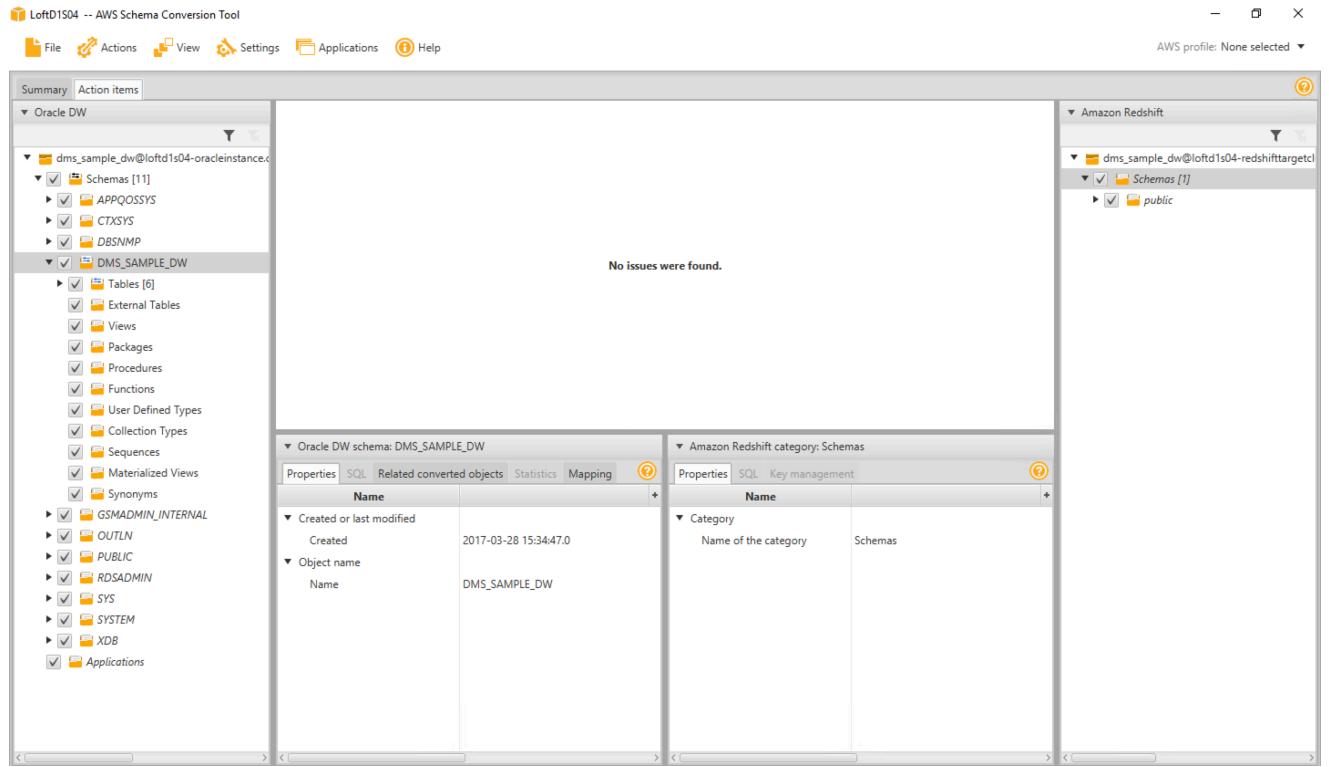
Figure: Conversion statistics for database storage objects

Object Type	Count	Percentage	Action Type
Schema (1:1/0/0/0)	1	100%	Objects Automatically Converted
Table (6: 6/0/0/0)	6	100%	Objects Automatically Converted
Constraint (6: 6/0/0/0)	6	100%	Objects Automatically Converted

Used memory: 343.94 MB, Free memory: 459.06 MB, Total memory: 803 MB, Maximum memory: 1.71 GB

You can click on Save to pdf to save this report to refer to it at a later time. The report discusses the type of objects that can be converted by using AWS SCT, along with potential migration issues and actions to resolve these issues.

4. Choose the **Action Items** tab. This tab will show all the items that need to be fixed before they can be migrated. E.g. data types that mismatch, objects not supported by Redshift like stored procedures, materialized views etc.



Some of the few issues depending on the source schema, which our customers may encountered during the schema conversion.

Unable to convert datatypes – This is for an CLOB datatype which is converted to VARCHAR. For a list of target data types see the link below. SCT also specifies using S3 as storage for this datatype.

https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Target.Redshift.html#CHAP_Target.Redshift.DataTypes

Amazon redshift doesn't support stored procedures – This will need to be manually converted to a User-Defined Function (UDF) in Redshift. Additional information can be found in the link below.

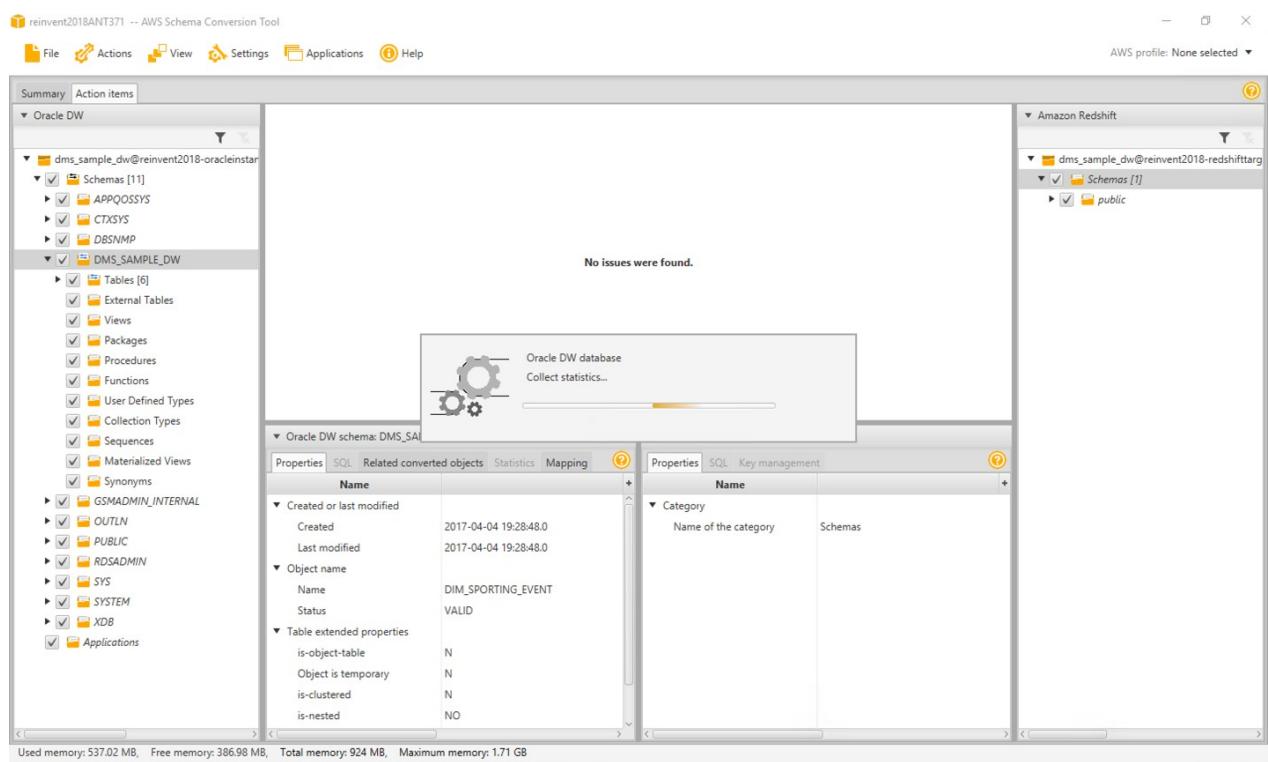
<https://docs.aws.amazon.com/redshift/latest/dg/user-defined-functions.html>

Amazon Redshift doesn't support sequences – This will need to be manually converted. All the materialized views need to be converted to tables in Redshift and data loading scripts need to be defined separately to populate them.

In a real life scenario, you will need to work through these issues and resolve them before moving on.

5. Open the context (right-click) menu for the **dms_sample_dw** item in the **Schemas** list in the Oracle DW pane on the left, and then choose **Collect Statistics**. AWS SCT analyzes the source data to recommend the best keys for the target Amazon Redshift database. For more information, see the link below.

https://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/CHAP_Converting.DW.html#CHAP_Converting.DW.Statistics



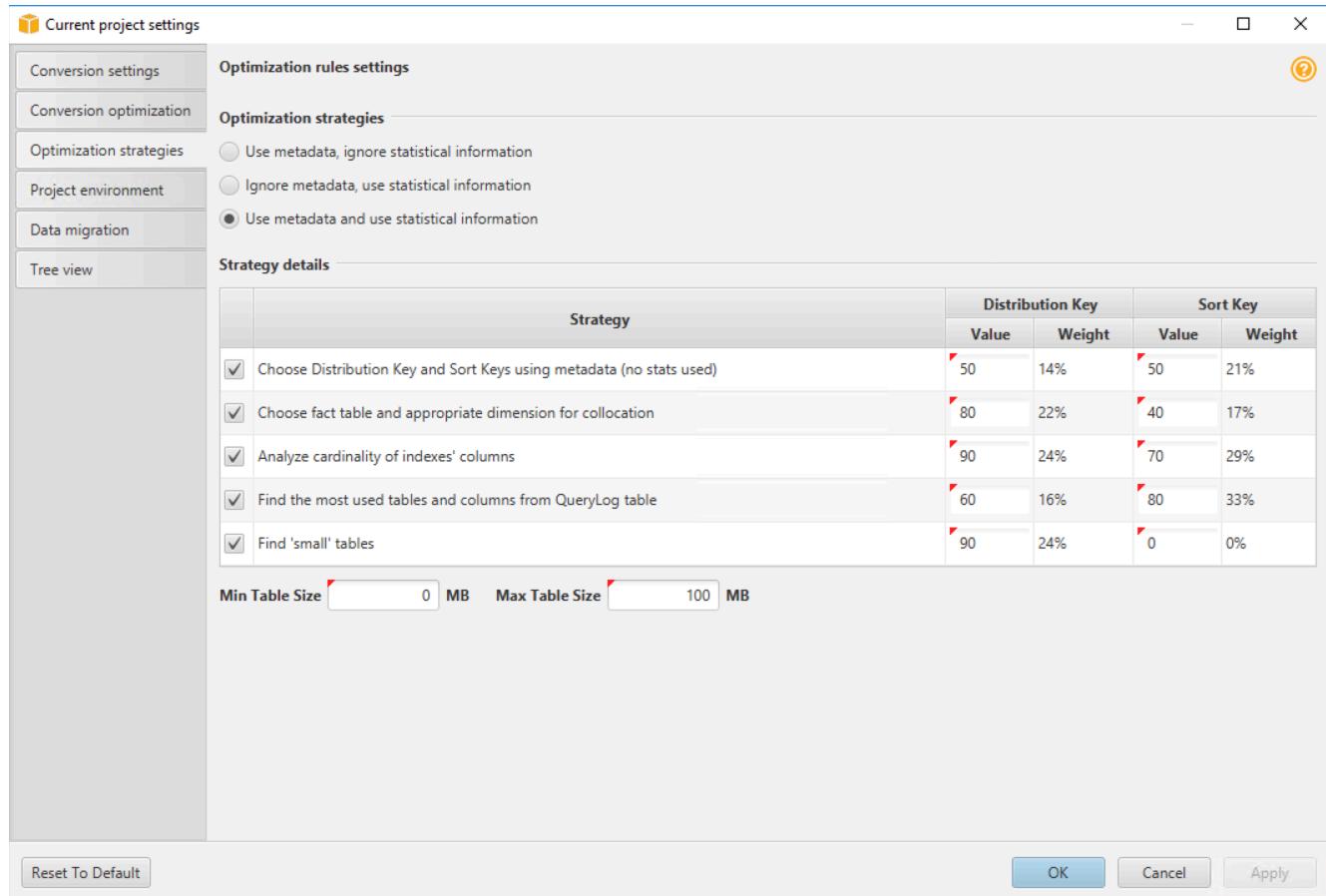
6. To optimize how the AWS Schema Conversion Tool (AWS SCT) converts your data warehouse schema, you can choose the strategies and rules you want the tool to use. After converting your schema, and reviewing the suggested keys, you can adjust your rules or change your strategy to get the results you want. Choose **Settings**, and then choose **Project Settings**. The Current project settings dialog box appears. In the left pane, choose **Optimization Strategies**. The optimization strategies appear in the right pane with the defaults selected. For **Optimization Strategies**, choose the optimization strategy you want to use. You can choose from the following:

- Use metadata, ignore statistical information
- Ignore metadata, use statistical information
- Use metadata and use statistical information

After you choose your optimization strategy, you can choose the rules you want to use and provide weights to each. You can choose from the following:

- Choose Distribution Key and Sort Keys using metadata
- Choose fact table and appropriate dimension for collation
- Analyze cardinality of indexes' columns
- Find the most used tables and columns from QueryLog table

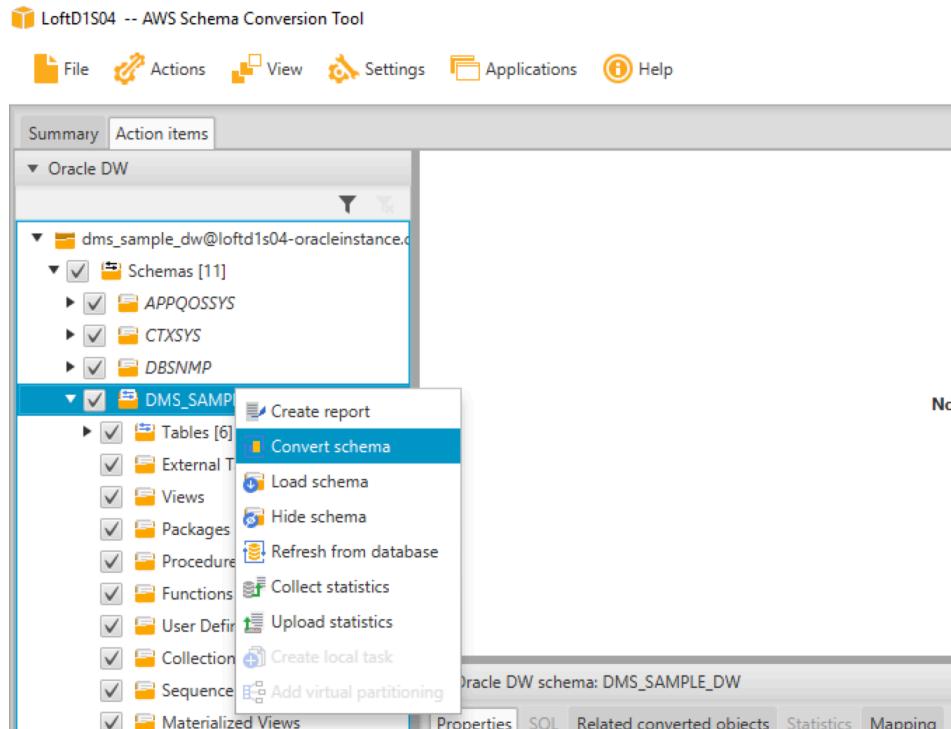
For the purposes of this lab we will not change any settings and use the default.



7. In the **Oracle DW** view open the context (right-click) menu for the **dms_sample_dw** schema, and then choose **Convert schema**.

Choose **Yes** for the confirmation message.

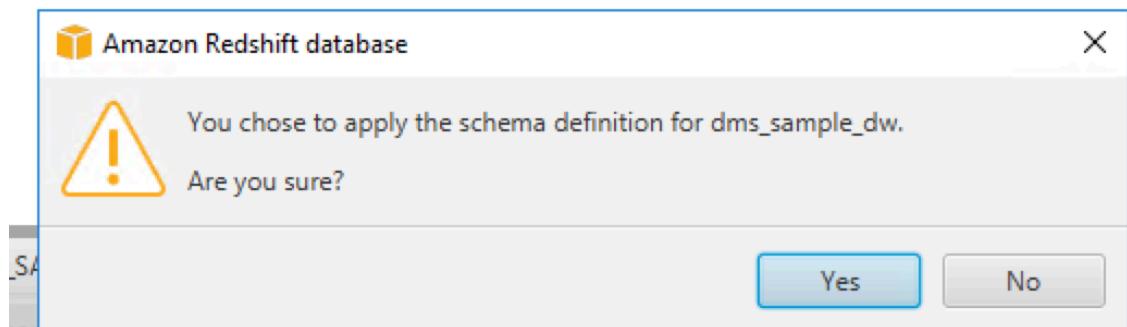
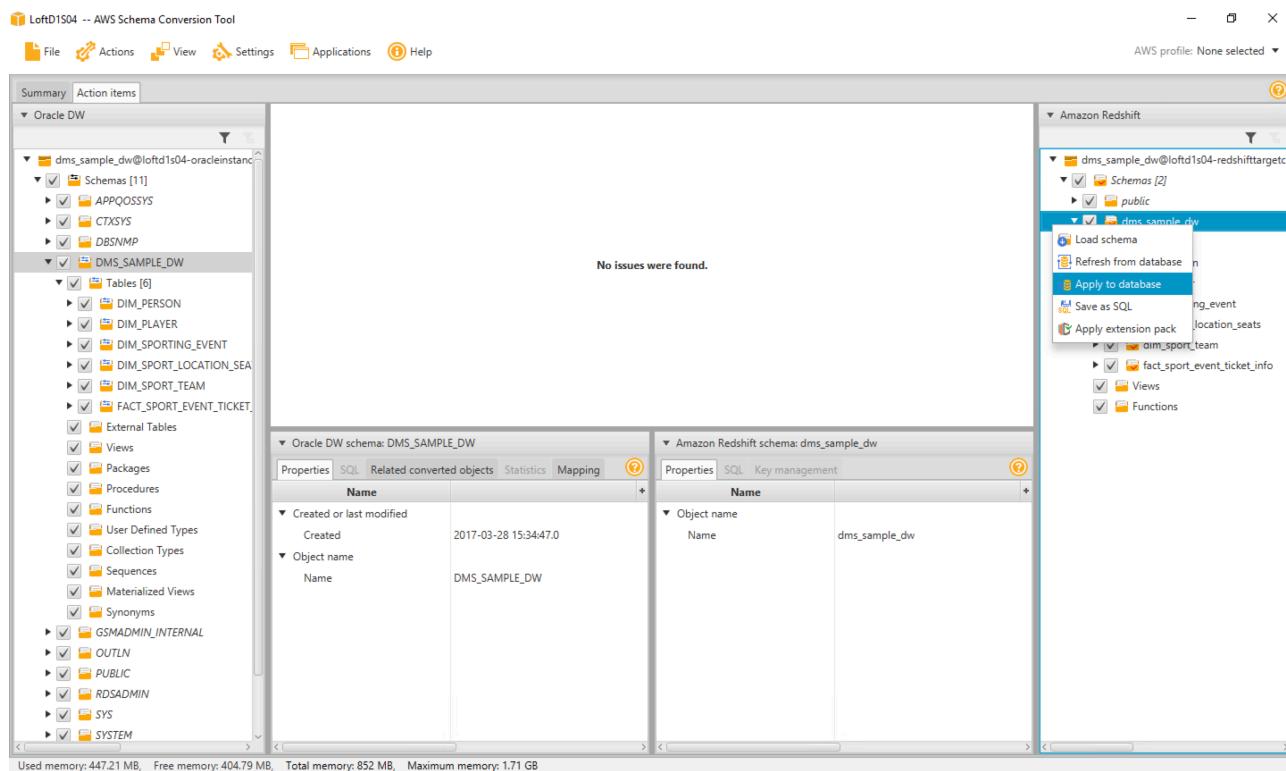
AWS SCT then converts your schema to the target database format.



The screenshot shows the AWS Schema Conversion Tool (SCT) interface with two panes: "Oracle DW" on the left and "Amazon Redshift" on the right. The Oracle DW pane displays the converted schema "dms_sample_dw" with its tables (e.g., DIM_PERSON, DIM_PLAYER, DIM_SPORTING_EVENT, DIM_SPORT_LOCATION_SEA, DIM_SPORT_TEAM, FACT_SPORT_EVENT_TICKET) and other objects. The Amazon Redshift pane displays the target schema "dms_sample_dw" with its tables (e.g., dim_person, dim_player, dim_sporting_event, dim_sport_location_seats, dim_sport_team, fact_sport_event_ticket_info) and other objects. Both panes have tabs for Properties, SQL, Related converted objects, Statistics, and Mapping. The status bar at the bottom of both panes indicates "No issues were found.".

- In the **Amazon Redshift view**, open the context (right-click) menu for the **dms_sample_dw** schema, and then choose **Apply to database** to apply the schema scripts to the target Amazon Redshift instance.

When complete, open the context (right-click) menu for the **dms_sample_dw** schema, and then choose **Refresh from Database** to refresh from the target database. Click **Yes** to confirm. The database schema has now been converted and imported from source to target.

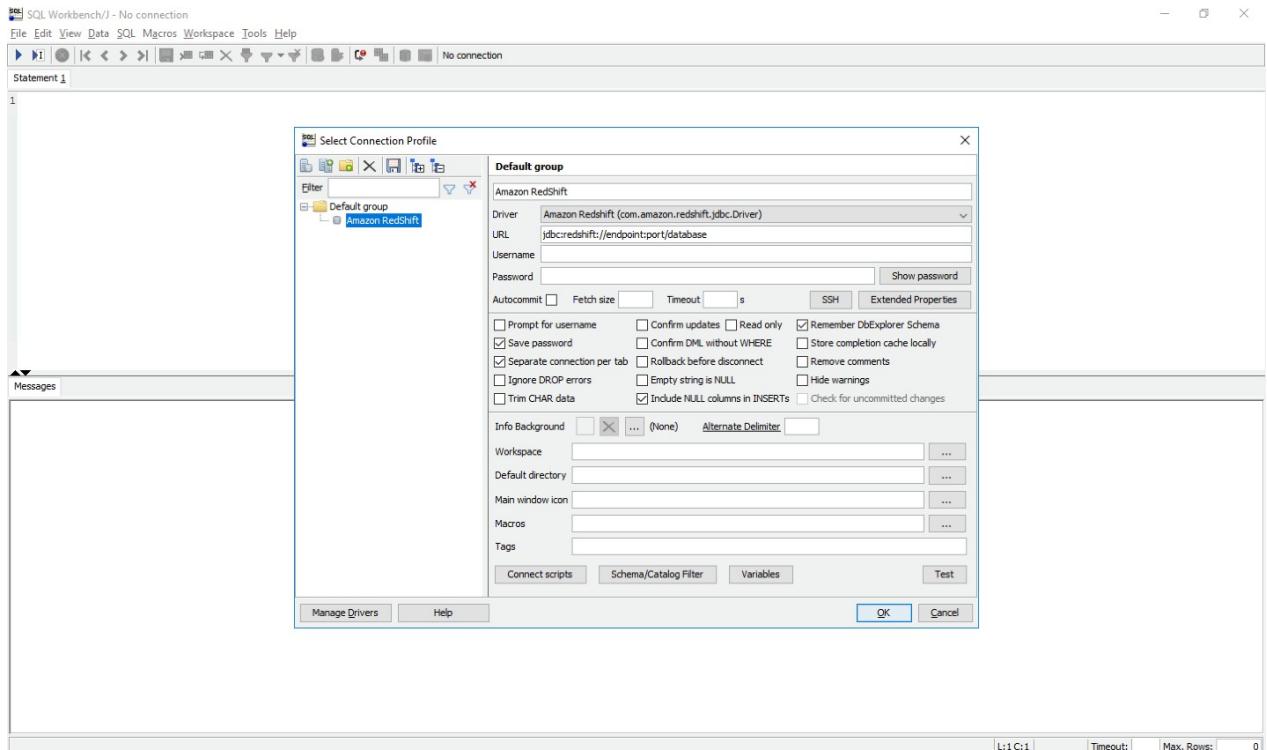


Validate the schema conversion and work through conversion issues

1. To validate the schema conversion, you compare the objects found in the Oracle and Amazon Redshift databases using SQL Workbench/J.

Double click the SQL Workbench shortcut  in the desktop folder on your EC2 instance.

2. Click on **File, Connect Window** and a Select Connection Profile screen will pop up.



3. Create two connections, **OracleDW** and **RedshiftDW** with the following information.

Name	OracleDW
Driver	Choose the Oracle driver which is installed on your EC2 instance. You can specify the driver's path by clicking on Manage Drivers and pointing the path to C:\Users\Administrator\Desktop\ANT371\OracleDrivers\ojdbc7.jar
URL	Jdbc url as follows- jdbc:oracle:thin:@ OracleEndpoint :1521:DMSDW Get the LoftD1S04OracleEndpoint from step 8 Example: jdbc:oracle:thin:@loftd1s04-oracleinstance.cs0mrnzrs0oa.eu-west-1.rds.amazonaws.com: 1521:DMSDW
Username	dms_sample_dw
Password	Password1

Check **AutoCommit** and click **OK**. **OracleDW** will connect to the Source Oracle database.

4. The **RedshiftDW** will connect to the target Redshift cluster.

Name	RedshiftDW
Driver	Choose the Redshift driver which is installed on your EC2 instance. You can specify the driver's path by clicking on Manage Drivers Drivers and pointing the path to C:\Users\Administrator\Desktop\ANT371\RedshiftDrivers\RedshiftJDBC42-1.2.16.1027.jar
URL	Jdbc url can be found on Redshift cluster page on the AWS Management Console. Or Get the LoftD1S04RedshiftEndpoint from step 9. Its format is as follows: jdbc:redshift://Redshift Endpoint:5439/dmsdwttarget
Username	dms_sample_dw
Password	Password1

5. In SQL Workbench/J, choose **File**, then choose **Connect window**. Choose the **RedshiftConnection** you created in an earlier step. Choose **OK**.
6. Run the following script to verify the number of object types and count in `dms_sample_dw` schema in the target Amazon Redshift database. These values should match the number of objects in the source Oracle database.

```
SELECT 'TABLE' AS OBJECT_TYPE, TABLE_NAME AS OBJECT_NAME,
TABLE_SCHEMA AS OBJECT_SCHEMA
FROM information_schema.TABLES WHERE TABLE_TYPE = 'BASE TABLE'
AND OBJECT_SCHEMA = 'dms_sample_dw';
```

The output from this query should be similar to the following:

Result 1	Messages
object_type	object_name
TABLE	dim_person
TABLE	dim_player
TABLE	dim_sport_location_seats
TABLE	dim_sport_team
TABLE	dim_sporting_event
TABLE	fact_sport_event_ticket_info

7. Verify the sort and distributions keys that are created in the Amazon Redshift cluster by using the following query.

```
SET search_path TO '$user', 'public', 'dms_sample_dw';
SELECT tablename,
       "column",
       TYPE,
       encoding, distkey, sortkey, "notnull"
  FROM pg_table_def
 WHERE (distkey = TRUE OR sortkey <> 0);
```

The results of the query reflect the distribution key (distkey) and sort key (sortkey) choices made by using AWS SCT key management.

tablename	column	type	encoding	distkey	sortkey	notnull
dim_person	person_id	numeric(38,10)	none	true		1 true
dim_person_dim_person_pk	person_id	numeric(38,10)	none	true		1 false
dim_player	player_id	numeric(38,10)	none	true		1 true
dim_player_dim_player_pk	player_id	numeric(38,10)	none	true		1 false
dim_sport_loca00054c5c49205285	sport_location_seat_id	numeric(38,10)	none	true		1 false
dim_sport_location_seats	sport_location_seat_id	numeric(38,10)	none	true		1 true
dim_sport_team	sport_team_id	numeric(38,10)	none	true		1 true
dim_sport_team00054c5c494c5a88	sport_team_id	numeric(38,10)	none	true		1 false
dim_sporting_e00054c5c48fa46a2	sporting_event_id	numeric(38,10)	none	true		1 false
dim_sporting_event	sporting_event_id	numeric(38,10)	none	true		1 true
fact_sport_eve00054c5c59926517	sporting_event_ticket_id	numeric(38,10)	none	true		1 false
fact_sport_event_ticket_info	sporting_event_ticket_id	numeric(38,10)	none	true		1 true

Create IAM Access Key and Access Secret Key for SCT Authentication

SCT requires IAM Access Key and Access Secret Key for authentication. Best practice is to create a minimum privilege IAM user for this purpose, which you can later delete.

You will start with creating two IAM policies. One which will allow SCT to list your bucket for the lab, and the other to Delete, Put, and Get objects in the bucket. Next you will create a new IAM user, attach these two policies to the user, then save the Access Key and Access Secret Key to be used in SCT.

1. Select **IAM** in the AWS Console. Select **Policies** and **Create Policy**.
2. In the Create policy page, select JSON, and copy/paste the following policy. Ensure to replace **YOURBUCKETNAME** with the S3 Bucket name created in CloudFormation.
Select **Review Policy**.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Stmt1497651415000",  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::YOURBUCKETNAME"  
            ]  
        }  
    ]  
}
```

Create policy

1 2

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

Visual editor JSON Import managed policy

```
1 {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Sid": "Stmt1497651415000",  
6             "Effect": "Allow",  
7             "Action": [  
8                 "s3>ListBucket"  
9             ],  
10            "Resource": [  
11                "arn:aws:s3:::YOURBUCKETNAME"  
12            ]  
13        }  
14    ]  
15}
```

3. In the Review Policy page, provide a policy name, such as **SCTListBucket** and select **Create Policy**.
4. You will return to the IAM Policies page. You need to create one more policy. Select **Create Policy**, select the JSON tab, and use the following policy. Ensure to replace **YOURBUCKETNAME** with the S3 Bucket name created in CloudFormation. Select **Review Policy**.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Stmt1497631447000",  
            "Effect": "Allow",  
            "Action": [  
                "s3:DeleteObject",  
                "s3:GetObject",  
                "s3:PutObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::YOURBUCKETNAME/*"  
            ]  
        }  
    ]  
}
```

5. In the Review Policy page, provide a policy name, such as **SCTReadWriteBucket** and select **Create Policy**.
6. You now need to create an IAM user. In IAM, go to **Users** and select **Add User**.
7. In the Add user page, provide a name, such as **SCTUser**, and select the **Programmatic Access** checkbox. Select **Next**.

Add user

1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

SCTUser

[+ Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type* Programmatic access

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

AWS Management Console access

Enables a **password** that allows users to sign-in to the AWS Management Console.

* Required

[Cancel](#)

[Next: Permissions](#)

8. In the Set permissions page, select **Attach existing policies directly**. In the **Filter policies** field, enter **SCT**. You should see both policies created. Select both and select **Next**.

Add user

1 2 3 4 5

▼ Set permissions

 Add user to group

 Copy permissions from existing user

 Attach existing policies directly

[Create policy](#)



[Filter policies](#) ▾

SCT

Showing 3 results

	Policy name ▾	Type	Used as	Description
<input checked="" type="checkbox"/>	SCTReadWriteBucket	Customer managed	<i>None</i>	
<input checked="" type="checkbox"/>	SCTLListBucket	Customer managed	<i>None</i>	

9. In the Review screen select **Create User**.

Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	SCTUser
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	SCTListBucket
Managed policy	SCTReadWriteBucket

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

10. The final screen shows the IAM Access key and Secret access key. Select **Show** to view the secret key.

Ensure you save these credentials for SCT. Once you select close the Secret key will no longer be available.

Add user

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://awpsa-redshift.signin.aws.amazon.com/console>

[Download .csv](#)

	User	Access key ID	Secret access key
▶	SCTUser	AKIAWALS64BG45SQYZW6	***** Show

[Close](#)

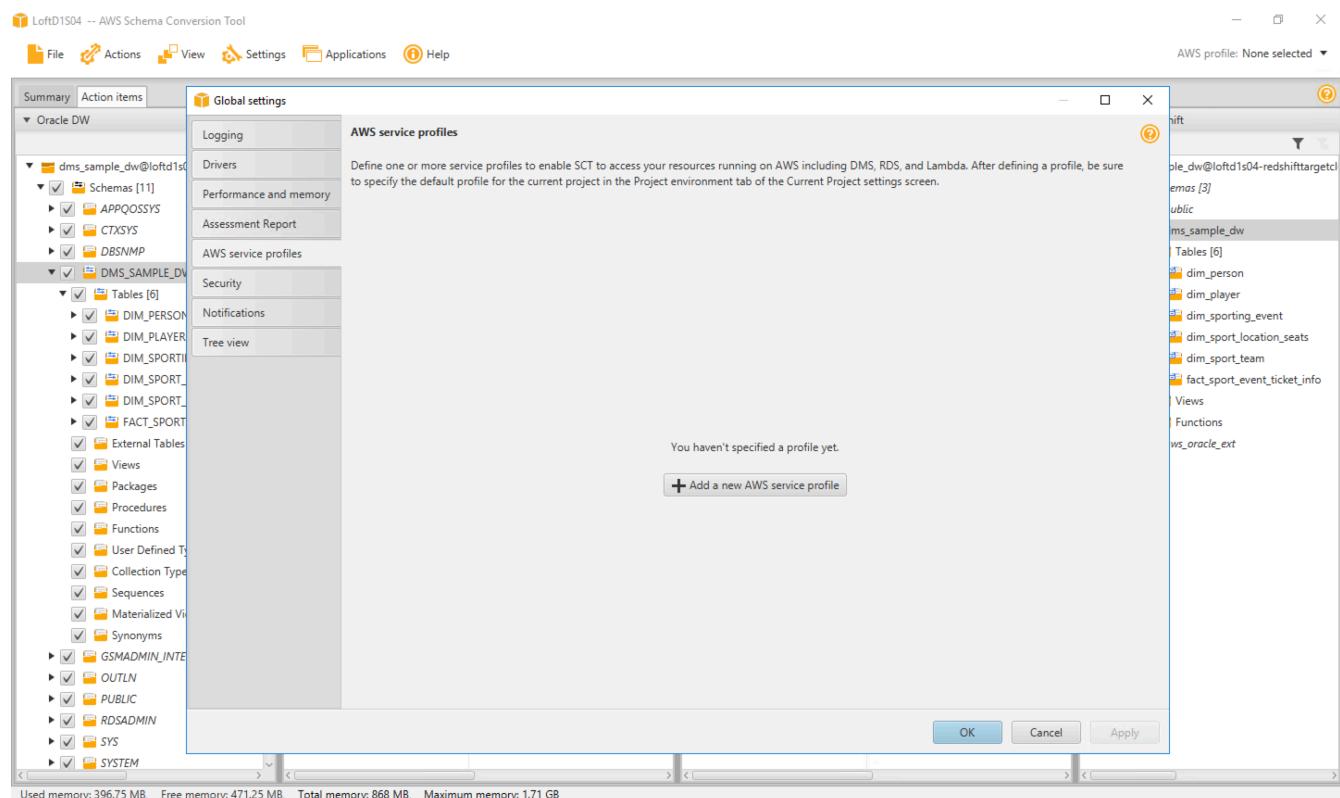
AWS SCT Global Settings and Amazon S3 Bucket Configuration

In this section, we will install and configure the SCT Extraction Agents. They will extract your data, they upload it to your Amazon S3 bucket, and depending on the option you select, they can load the data into the Amazon Redshift cluster.

Before you continue, you must provide the credentials to connect to your AWS account and your Amazon S3 bucket. You store your credentials and bucket information in a profile in the global application settings, and then associate the profile with your AWS SCT project.

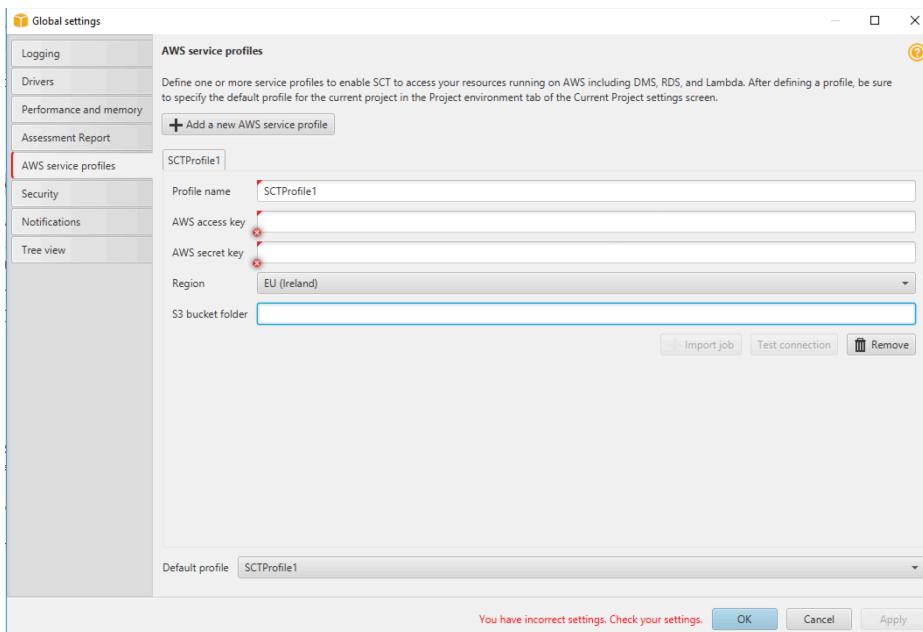
Steps to Create a Global Profile:

1. Click on ‘Settings’ -> ‘Global Settings’ and then select ‘AWS Service Profiles’.



2. Click on “Add a new AWS Service Profile”

3. Enter details such as **Profile Name**, **AWS Access Key** and **AWS Secret Key**



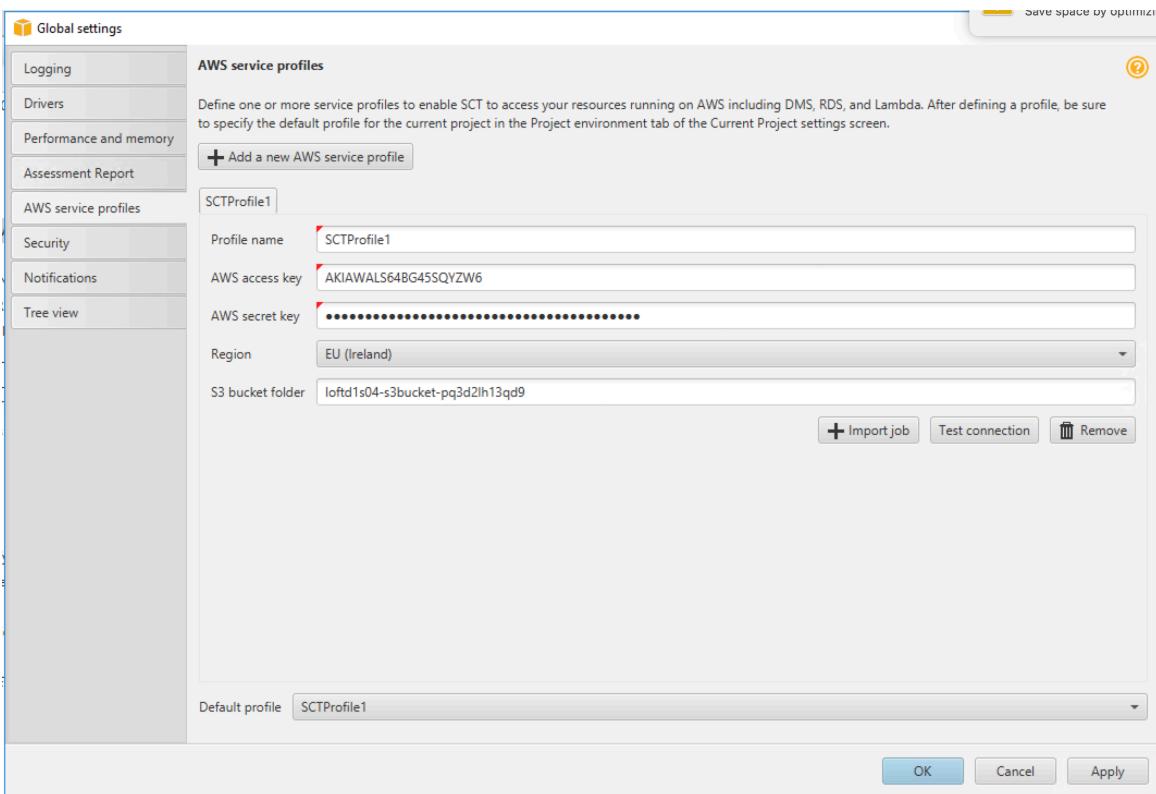
NOTE: You will need an AWS Access Key and AWS Secret Access key in AWS IAM for connecting to the AWS Services. For instructions to create the appropriate policies and user, go to **Appendix A** at the end of the lab

4. Select the region that's applicable or selected during the execution of CloudFormation script, for example: we suggested to use **EU-IRELAND** as your region.
5. For the “S3 bucket folder”, use the “**LoftD1S04S3Bucket**” name from the **output tab of the CloudFormation console**.

CloudFormation > Stacks: LoftD1S04

Stacks (1)		Outputs (9)																																									
Create stack		Search outputs																																									
<input type="text" value="Q Loft"/>	<input type="button" value="X"/>	<input type="button" value="C"/>	<input type="button" value="G"/>																																								
<input type="button" value="Active"/>	<input type="button" value="View nested"/>	<input type="button" value="1 < >"/>	<input type="button" value="Export name"/>																																								
LoftD1S04 2019-06-18 23:28:49 UTC-0700 <input checked="" type="button" value="CREATE_COMPLETE"/>		<table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> <th>Description</th> <th>Export name</th> </tr> </thead> <tbody> <tr> <td>LoftD1S04AGENT1PrivateIP</td> <td>172.31.0.155</td> <td>Private IP of the LoftD1S04 SCT Agent1 EC2 linux instance</td> <td>-</td> </tr> <tr> <td>LoftD1S04AGENT1PublicDNS</td> <td>ec2-34-244-85-154.eu-west-1.compute.amazonaws.com</td> <td>Public DNS Name of the LoftD1S04 SCT Agent1 EC2 linux instance</td> <td>-</td> </tr> <tr> <td>LoftD1S04AGENT2PrivateIP</td> <td>172.31.0.124</td> <td>Private IP of the LoftD1S04 SCT Agent2 EC2 linux instance</td> <td>-</td> </tr> <tr> <td>LoftD1S04AGENT2PublicDNS</td> <td>ec2-34-243-221-212.eu-west-1.compute.amazonaws.com</td> <td>Public DNS Name of the LoftD1S04 SCT Agent2 EC2 linux instance</td> <td>-</td> </tr> <tr> <td>LoftD1S04OracleEndpoint</td> <td>loftd1s04-oracleinstance.cs0mrnzsrs0aa.eu-west-1.rds.amazonaws.com</td> <td>Endpoint of source Oracle RDS DB instance</td> <td>-</td> </tr> <tr> <td>LoftD1S04RedshiftEndpoint</td> <td>loftd1s04-redshifttargetcluster-1t0w9k68d259.cbbvjdhrfqn.eu-west-1.redshift.amazonaws.com</td> <td>Endpoint of target Redshift instance</td> <td>-</td> </tr> <tr> <td>LoftD1S04S3Bucket</td> <td>loftd1s04-s3bucket-pq3d2lh13qd9</td> <td>ANT371 workshop S3 bucket</td> <td>-</td> </tr> <tr> <td>LoftD1S04SCTDNS</td> <td>ec2-18-202-175-154.eu-west-1.compute.amazonaws.com</td> <td>Public DNS Name of the LoftD1S04 SCT Windows instance</td> <td>-</td> </tr> <tr> <td>LoftD1S04VpcId</td> <td>vpc-08b7bec203db5c8da</td> <td>ANT371 workshop VPC Identifier</td> <td>-</td> </tr> </tbody> </table>		Key	Value	Description	Export name	LoftD1S04AGENT1PrivateIP	172.31.0.155	Private IP of the LoftD1S04 SCT Agent1 EC2 linux instance	-	LoftD1S04AGENT1PublicDNS	ec2-34-244-85-154.eu-west-1.compute.amazonaws.com	Public DNS Name of the LoftD1S04 SCT Agent1 EC2 linux instance	-	LoftD1S04AGENT2PrivateIP	172.31.0.124	Private IP of the LoftD1S04 SCT Agent2 EC2 linux instance	-	LoftD1S04AGENT2PublicDNS	ec2-34-243-221-212.eu-west-1.compute.amazonaws.com	Public DNS Name of the LoftD1S04 SCT Agent2 EC2 linux instance	-	LoftD1S04OracleEndpoint	loftd1s04-oracleinstance.cs0mrnzsrs0aa.eu-west-1.rds.amazonaws.com	Endpoint of source Oracle RDS DB instance	-	LoftD1S04RedshiftEndpoint	loftd1s04-redshifttargetcluster-1t0w9k68d259.cbbvjdhrfqn.eu-west-1.redshift.amazonaws.com	Endpoint of target Redshift instance	-	LoftD1S04S3Bucket	loftd1s04-s3bucket-pq3d2lh13qd9	ANT371 workshop S3 bucket	-	LoftD1S04SCTDNS	ec2-18-202-175-154.eu-west-1.compute.amazonaws.com	Public DNS Name of the LoftD1S04 SCT Windows instance	-	LoftD1S04VpcId	vpc-08b7bec203db5c8da	ANT371 workshop VPC Identifier	-
Key	Value	Description	Export name																																								
LoftD1S04AGENT1PrivateIP	172.31.0.155	Private IP of the LoftD1S04 SCT Agent1 EC2 linux instance	-																																								
LoftD1S04AGENT1PublicDNS	ec2-34-244-85-154.eu-west-1.compute.amazonaws.com	Public DNS Name of the LoftD1S04 SCT Agent1 EC2 linux instance	-																																								
LoftD1S04AGENT2PrivateIP	172.31.0.124	Private IP of the LoftD1S04 SCT Agent2 EC2 linux instance	-																																								
LoftD1S04AGENT2PublicDNS	ec2-34-243-221-212.eu-west-1.compute.amazonaws.com	Public DNS Name of the LoftD1S04 SCT Agent2 EC2 linux instance	-																																								
LoftD1S04OracleEndpoint	loftd1s04-oracleinstance.cs0mrnzsrs0aa.eu-west-1.rds.amazonaws.com	Endpoint of source Oracle RDS DB instance	-																																								
LoftD1S04RedshiftEndpoint	loftd1s04-redshifttargetcluster-1t0w9k68d259.cbbvjdhrfqn.eu-west-1.redshift.amazonaws.com	Endpoint of target Redshift instance	-																																								
LoftD1S04S3Bucket	loftd1s04-s3bucket-pq3d2lh13qd9	ANT371 workshop S3 bucket	-																																								
LoftD1S04SCTDNS	ec2-18-202-175-154.eu-west-1.compute.amazonaws.com	Public DNS Name of the LoftD1S04 SCT Windows instance	-																																								
LoftD1S04VpcId	vpc-08b7bec203db5c8da	ANT371 workshop VPC Identifier	-																																								

6. The AWS profile screen with your inputs provided should look similar to the screenshot given below:



7. Click **Apply** button to save the profile and click **OK**.

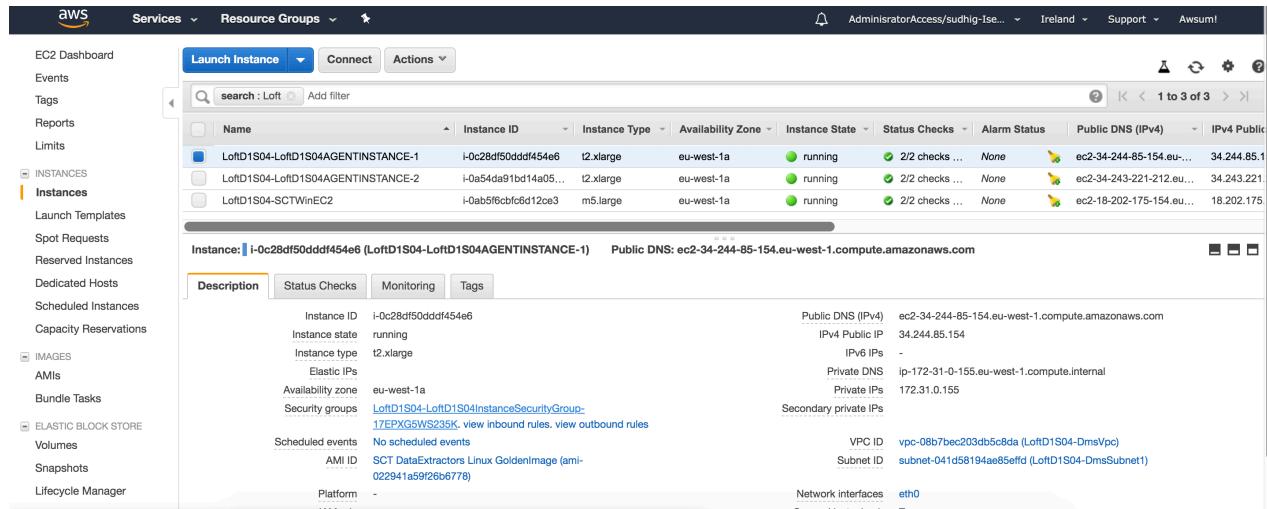
Installing and Configuring the AWS SCT Extraction Agents

In this step, we will install and configure the AWS SCT Extraction Agents on two RedhatLinux hosts. SSH to the Redhat Linux host using the IP address and private key as mentioned in the below steps.

Steps for SSH login to Agent Extractor hosts

If you are using windows, please follow the instructions here:
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>

1. Login to AWS Management Console and note the public IP address of Agent1instance.



The screenshot shows the AWS Management Console with the EC2 Instances page open. The left sidebar shows navigation links like Services, Resource Groups, Instances, and AMIs. The main area displays a table of instances with columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), and IPv4 Public. Three instances are listed:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public
LoftD1S04-LoftD1S04AGENTINSTANCE-1	i-0c28df50ddd454e6	t2.xlarge	eu-west-1a	running	2/2 checks ...	None	ec2-34-244-85-154.eu...	34.244.85.1
LoftD1S04-LoftD1S04AGENTINSTANCE-2	i-0a54da91bd14a05...	t2.xlarge	eu-west-1a	running	2/2 checks ...	None	ec2-34-243-221-212.eu...	34.243.221
LoftD1S04-SCTWinEC2	i-0ab5f6cbfd12ce3	m5.large	eu-west-1a	running	2/2 checks ...	None	ec2-18-202-175-154.eu...	18.202.175

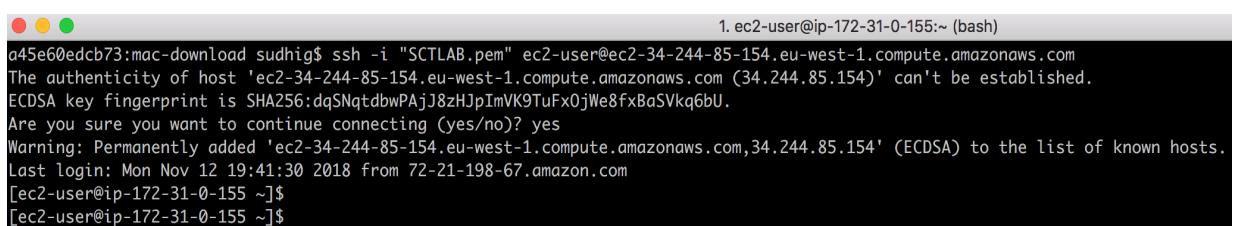
Details for the selected instance (i-0c28df50ddd454e6) are shown in the bottom panel:

Description	Value	Description	Value
Instance ID	i-0c28df50ddd454e6	Public DNS (IPv4)	ec2-34-244-85-154.eu-west-1.compute.amazonaws.com
Instance state	running	IPv4 Public IP	34.244.85.154
Instance type	t2.xlarge	IPv6 IPs	-
Elastic IPs	-	Private DNS	ip-172-31-0-155.eu-west-1.compute.internal
Availability zone	eu-west-1a	Private IPs	172.31.0.155
Security groups	LoftD1S04-LoftD1S04InstanceSecurityGroup-17EPXGWS235K, view inbound rules, view outbound rules	Secondary private IPs	-
Scheduled events	No scheduled events	VPC ID	vpc-08b7bec203db5c8da (LoftD1S04-DmsVpc)
AMI ID	SCT DataExtractors Linux GoldenImage (ami-022941a59f26b6778)	Subnet ID	subnet-041d5819ae85effd (LoftD1S04-DmsSubnet1)
Platform	-	Network interfaces	eth0
		Source/dest. check	True

2. Open the “Terminal” application in Mac and go to the location where you saved the EC2 Private Key (.pem) file

Execute below SSH command to connect to AWS SCT Extraction Agent EC2 host. Please remember to replace with your Keypair file Name and the Public IP address of your EC2 instances.

```
ssh -i <keypair_file_name> ec2-user@<public-ip-address-of-agent-host>
```



```
1. ec2-user@ip-172-31-0-155:~ (bash)
a45e60edcb73:mac-download sudhig$ ssh -i "SCLTALB.pem" ec2-user@ec2-34-244-85-154.eu-west-1.compute.amazonaws.com
The authenticity of host 'ec2-34-244-85-154.eu-west-1.compute.amazonaws.com (34.244.85.154)' can't be established.
ECDSA key fingerprint is SHA256:dqSNqtdbwPAjj8zHJpImVK9TuFx0jWe8fxBaSVkq6bU.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-34-244-85-154.eu-west-1.compute.amazonaws.com,34.244.85.154' (ECDSA) to the list of known hosts.
Last login: Mon Nov 12 19:41:30 2018 from 72-21-198-67.amazonaws.com
[ec2-user@ip-172-31-0-155 ~]$ [ec2-user@ip-172-31-0-155 ~]$
```

- Once you are connected to Agent Host using SSH, perform the below steps from the terminal to install and configure the AWS SCT Extraction Agent.

NOTE: The agent rpm executables and JDBC driver files have been copied to appropriate locations.

- Run below commands from the terminal to install the agent

```
cd /home/ec2-user/agents  
sudo rpm -ivh aws-schema-conversion-tool-extractor-1.0.620-1.x86_64.rpm
```

After installing the SCT Agent, we configure the agent to work with the AWS SCT, source Data Warehouse and target Data Warehouse end-to-end.

- Execute the following command to configure the SCT Extraction Agent

```
/usr/share/aws/sct-extractor/bin/sct-extractor-setup.sh -configuration
```

Specify values for the properties as highlighted below –

```
[ec2-user@ip-172-31-0-155 agents]$ /usr/share/aws/sct-extractor/bin/sct-extractor-setup.sh -  
configuration  
Empty settings.properties file. Load default properties  
*****  
* *  
* AWS SCT Data Extractor Configuration *  
* *  
*****  
User name: ec2-user  
User home: /home/ec2-user  
*****  
Warehouse vendor [ORACLE]: ORACLE  
Listening port [8192]: 8192  
ORACLE JDBC driver file or files: /home/ec2-user/drivers/ojdbc7.jar  
REDSHIFT JDBC driver file or files: /home/ec2-user/drivers/RedshiftJDBC42-1.2.15.1025.jar  
Working folder [/home/ec2-user]: /home/ec2-user/extractor-work  
Enter agent name if you are recovering a failed agent, or press Enter otherwise:  
Enable SSL communication [YES/no]: no  
You're selecting a communication method that isn't secure. If you do this, sensitive information  
will be transferred through an unencrypted channel and might be intercepted  
by attackers.  
Are you sure [yes/NO]: yes
```

The AWS SCT data extractor was successfully configured.

NOTE: We selected “no” for “Enable SSL communication” option while configuring the AWS SCT Extraction Agent. This is ONLY for the lab purposes. When you are migrating the production data warehouses on AWS, it is highly recommended to enable SSL

communication between the Extraction Agents, AWS SCT, and the source and target database.

6. Run the following command to start the Extractor Agent on Host 1

```
sudo service sct-extractor start
```

```
[ec2-user@ip-172-31-0-155 agents]$ sudo service sct-extractor start
Starting AWS SCT Data Extractor ...
AWS SCT Data Extractor started ...
[ec2-user@ip-172-31-0-155 agents]$
```

Steps for Installing and Configuring the SCT Extraction Agents on Host 2

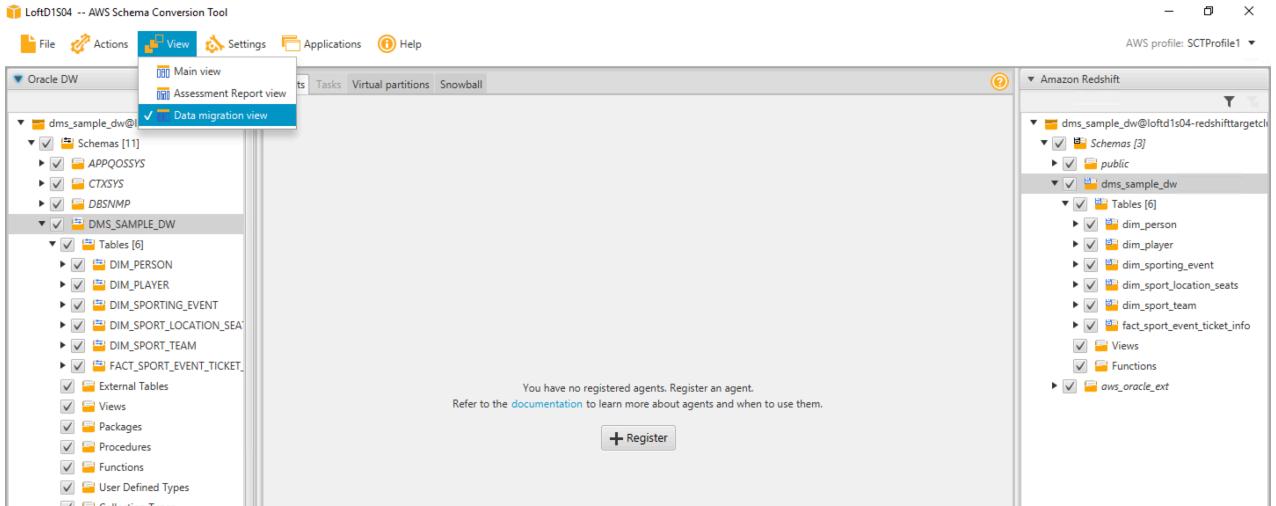
Please repeat the previous steps (1 to 6) to configure the SCT extractor agent on host 2.

Registering Extraction Agents with AWS SCT tool

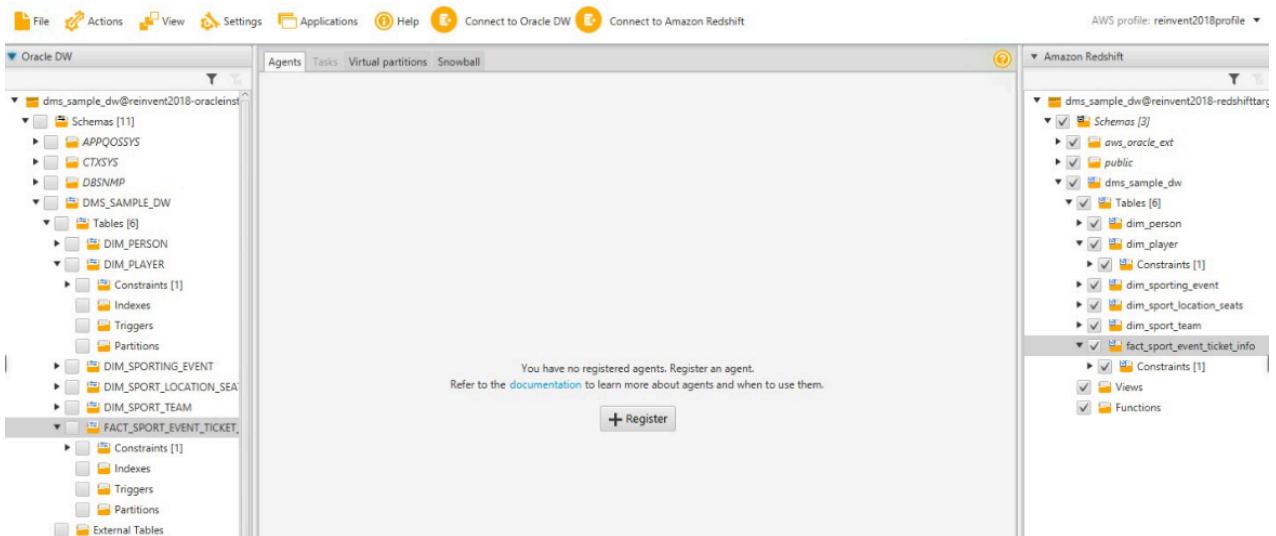
Now that we have SCT Extraction Agent installed, configured and started on both the hosts successfully, we can register them with the AWS SCT tool.

Please follow below steps to register Agents with AWS SCT:

1. Connect to Windows server using RDP client and open AWS Schema Conversion Tool.
2. Select View -> Data Migration View



- From the central panel, click “Register” button under “Agents” tab.



- Fill-in the required details for Agent registration as shown below:

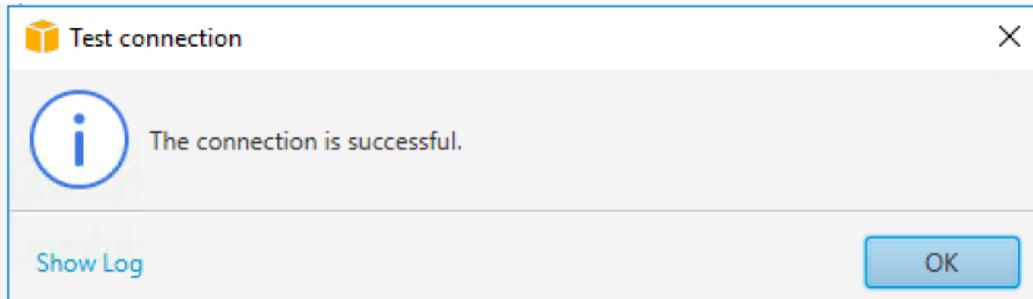
 New agent registration

X

Connection	SSL	
Description	dw-sct-agent-1	
Host name	172.31.0.124	
Port	8192	
<input type="checkbox"/> Use SSL		
Test Connection	Register	Cancel

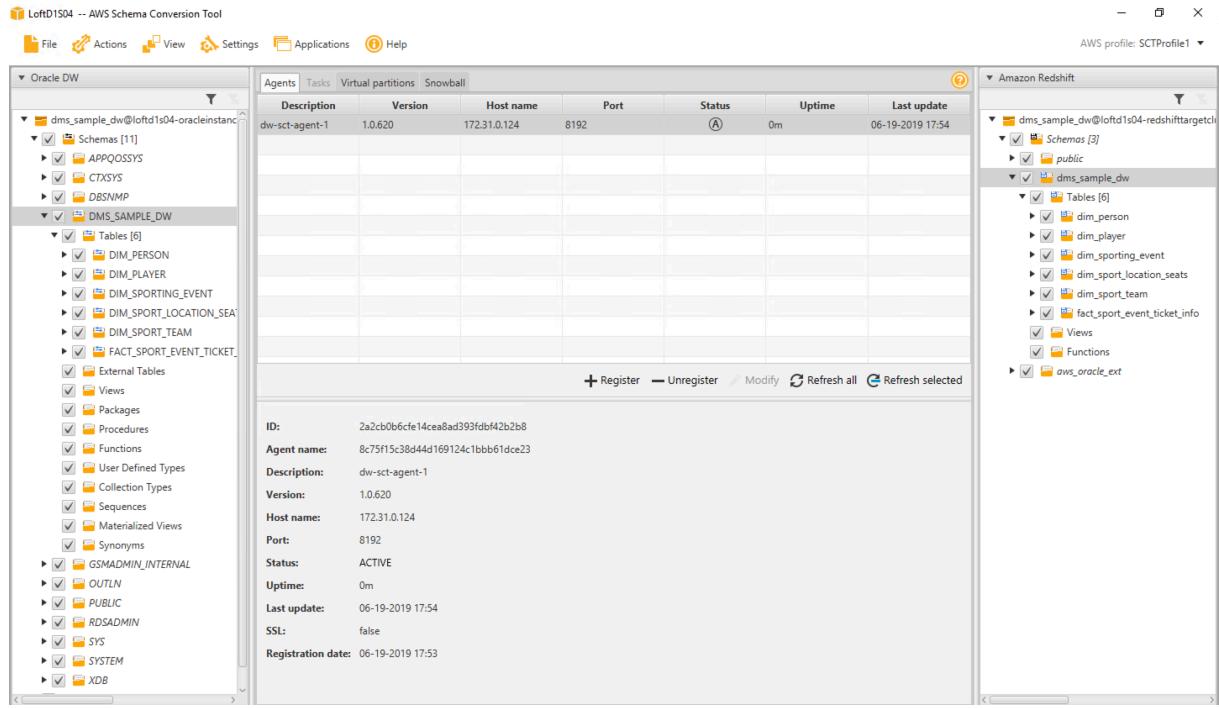
Note that Hostname is the Private IP Address of your Hosts on which SCT Extractor Agent is running. This address can be obtained from Amazon EC2 console UI or from “**LoftD1S04AGENT1PrivateIP**” and “**LoftD1S04AGENT2PrivateIP**” outputs in the CloudFormation console.

5. Click on “Test Connection”



6. Click on “Register” to register the agent with the AWS SCT tool

7. Register another agent running on host2 with the AWS SCT tool using the same steps as above. You should see two SCT extraction agents registered in AWS SCT console as given below:

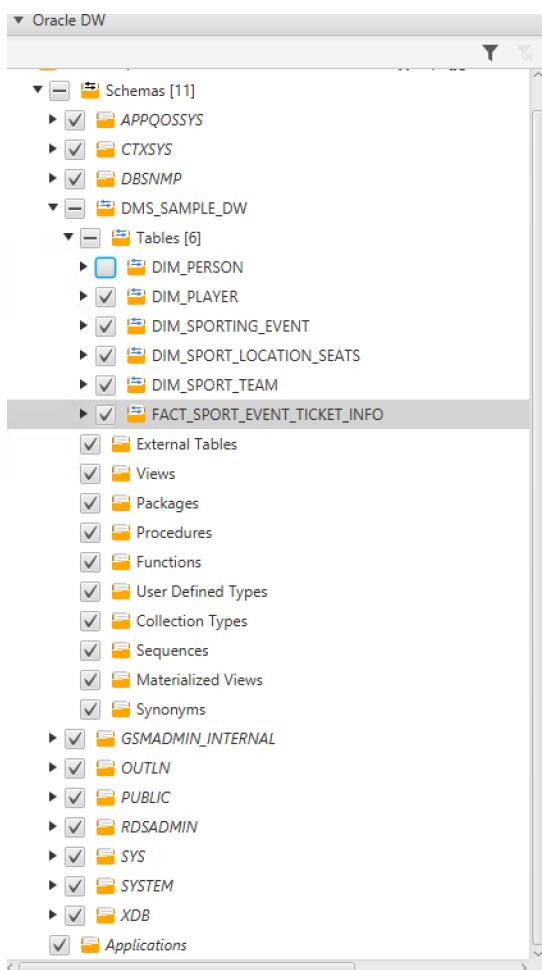


Note that, in order to connect to AWS SCT Extractors Agents from AWS SCT GUI, entries are already made in the Security Groups associated with Agent hosts in CloudFormation template.

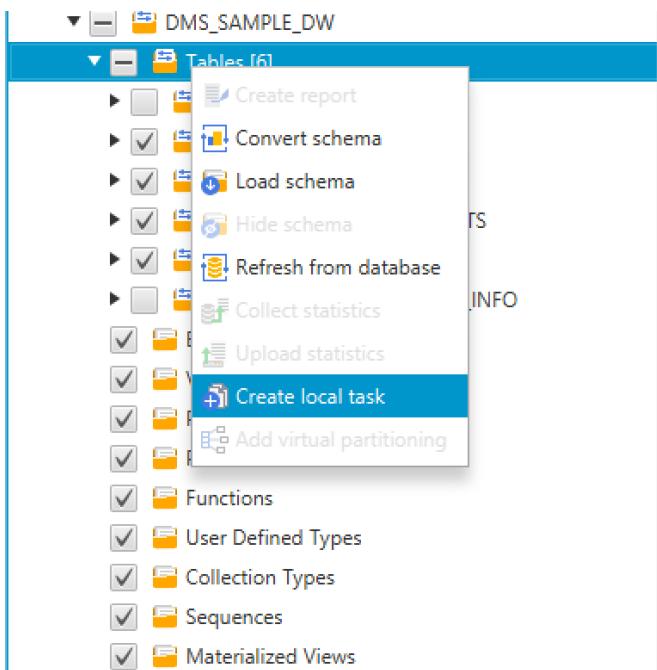
Creating, Running and Monitoring the AWS SCT Data Extraction Tasks

You can select one or more tables to be migrated at a time when creating Data Extraction Tasks. You can choose all tables, but we recommend against that for performance reasons. We recommend that you create multiple tasks for multiple tables based on the size of the tables in your data warehouse.

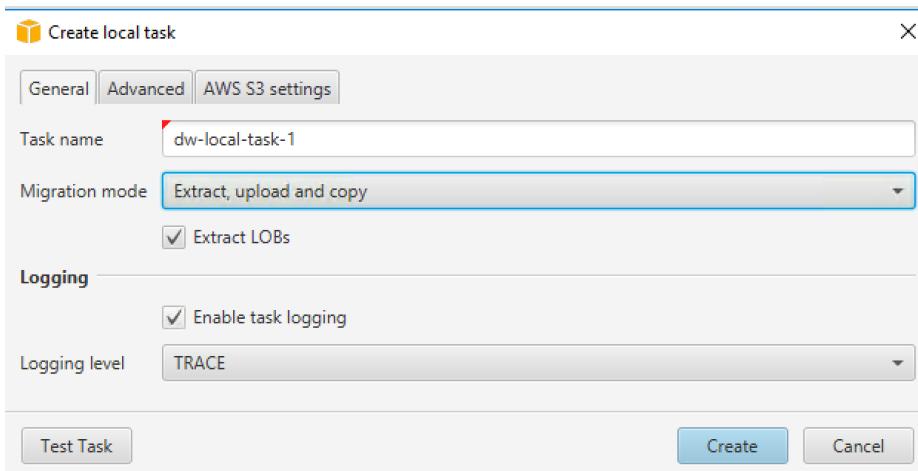
1. In the AWS SCT application UI, for the current reinventANT371 project in the source oracle database tab, select all the tables under the “**DMS_SAMPLE_DW**”. Now uncheck only the “**DIM_PERSON**” tables. This table is very large and we will be using Virtual Partitioning feature to migrate these tables to Amazon Redshift. Overall the final table selection for migration would look like the screenshot below:



2. Right Click the “Tables” node and select “Create Local Task”. If the “Create Local Task” is not enabled, then click on “**Connect to Oracle**” button on top of the AWS SCTWindow.



3. In the “Create local task” window, fill-in the required details as shown below:



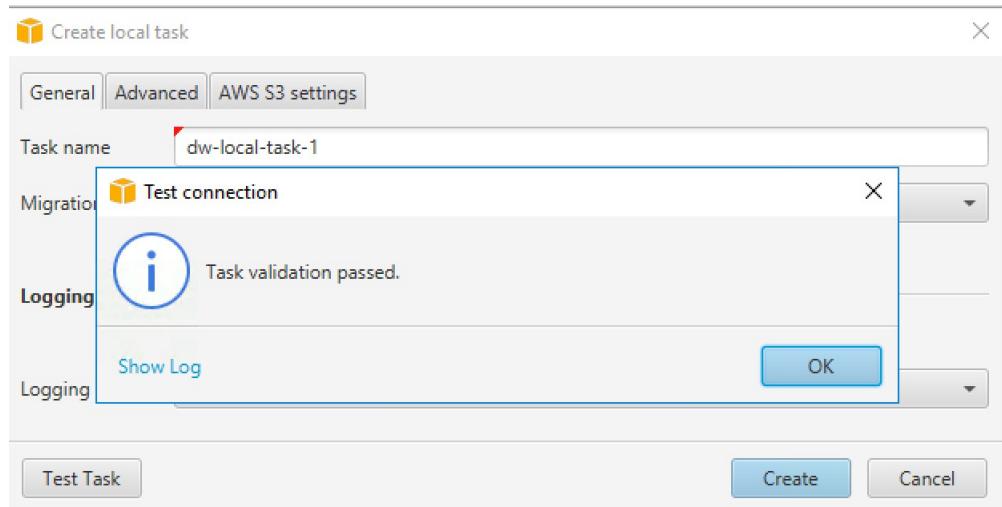
For Migration Mode, you can choose one of the following:

- Extract Only – Extract your data and save the data to your local working folders.
- Extract and Upload – Extract your data and upload your data to Amazon S3.
- Extract, Upload and Copy – Extract your data, upload your data to Amazon S3, and copy it into your Amazon Redshift data warehouse.

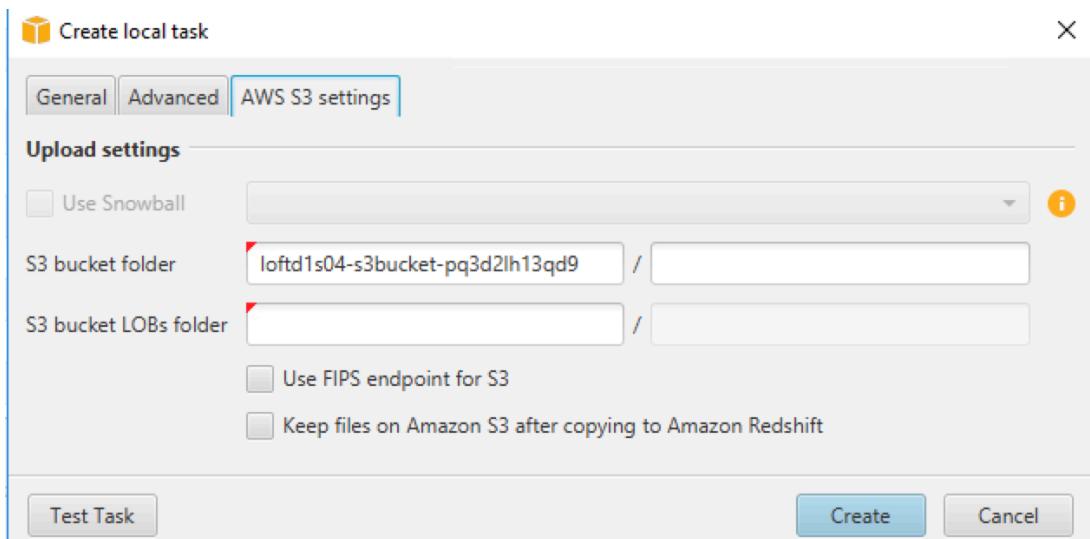
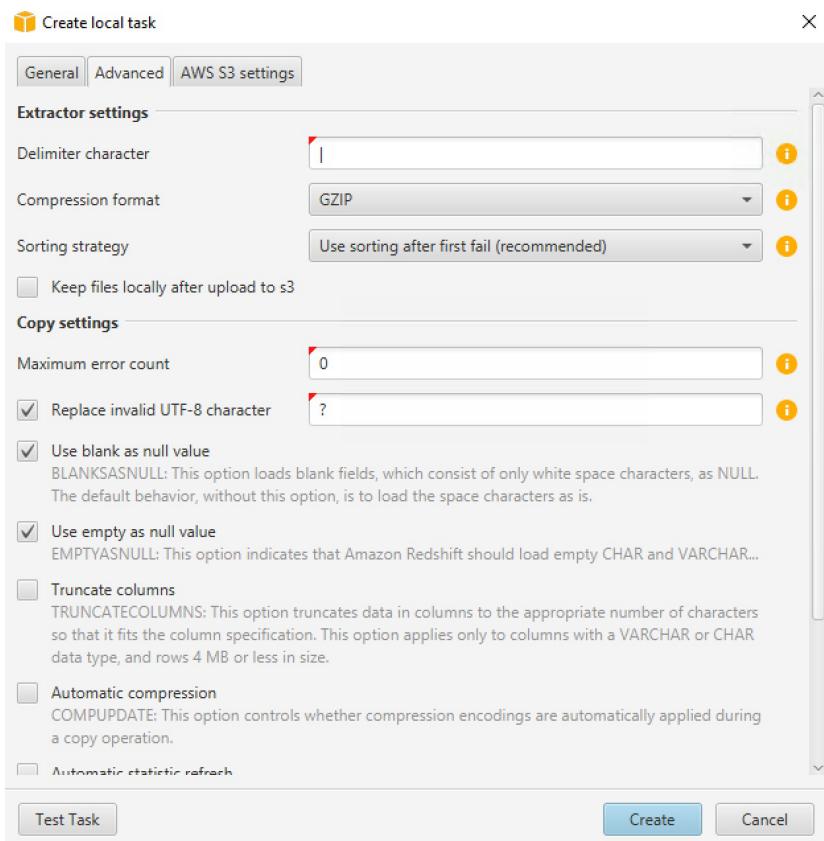
In this lab, we will select the third option, i.e. “**Extract, upload and copy**”.

This will extract the data from Oracle data warehouse, move it to Amazon S3 and load the data into Amazon Redshift using COPY command.

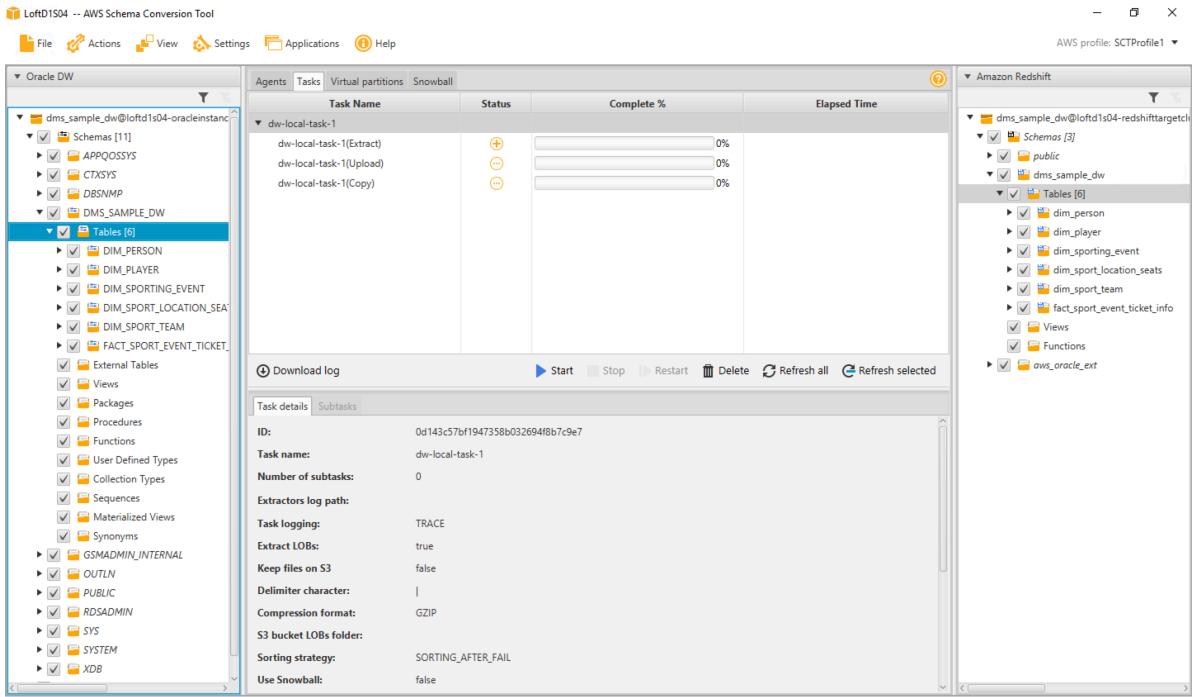
4. Click ‘Test Task’. ‘Test Validation Passed’ message is displayed.



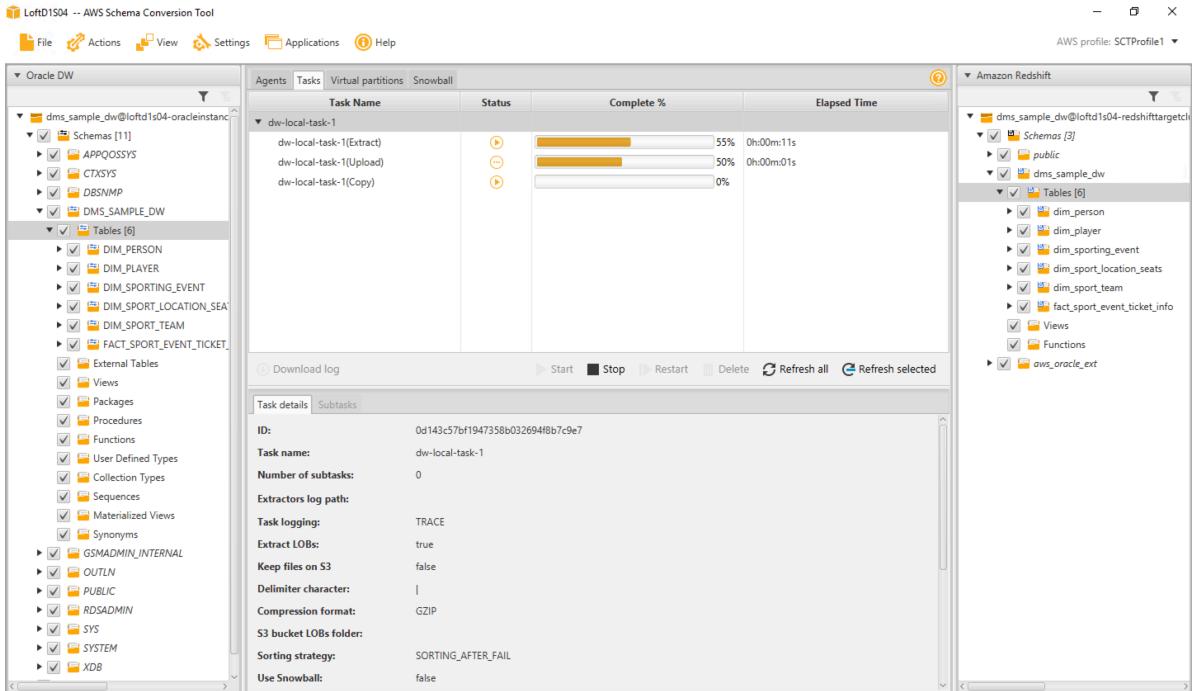
5. Go through various options available under “Advanced” and “AWS S3 settings” tabs.



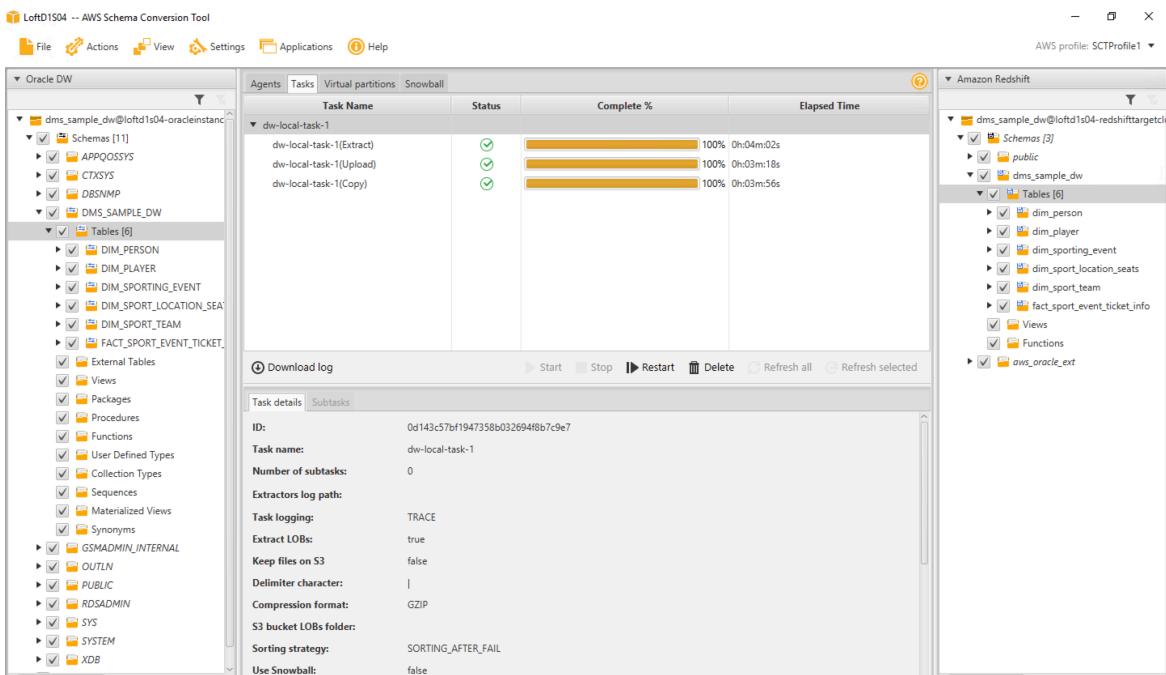
6. Click on 'Create' button to create the extraction task.



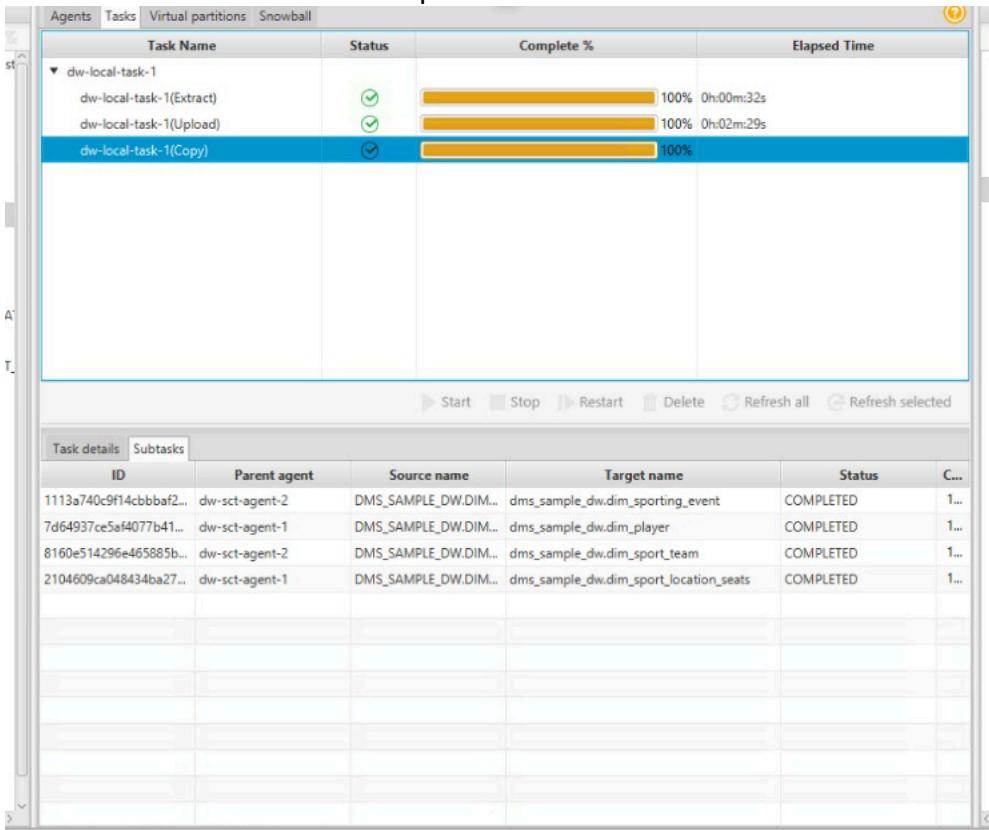
- Select the task in AWS SCT and press “Start” to start the migration from Oracle data warehouse to Amazon Redshift.



8. Observe that the “Extract” and “Upload” steps are executed in parallel.



9. Click on each step (Extract, Upload, Copy) in the AWS SCT UI and select “Subtasks” tab. Observe that for each table a separate subtask is created.



The subtasks have been distributed between the two extractor agents we had registered with the AWS SCT.

Migrating Large Tables using the Virtual Partitioning Feature

In AWS SCT, you can create virtual partitions for migrating your data. There are three partition types, which work with specific data types:

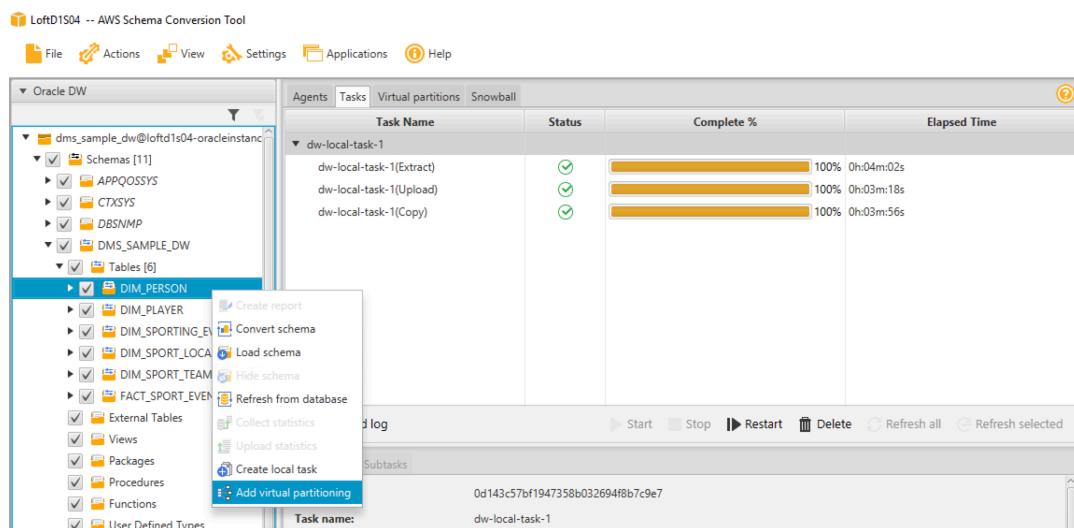
- The **RANGE** partition type works with numeric, date, and time data types.
- The **LIST** partition type works with numeric, character, and date and time data types.
- The **DATE AUTO SPLIT** partition type works with date and time data types.

For more information on Virtual Partitioning, please refer to AWS Documentation at -
https://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/CHAP_Agents.DW.html

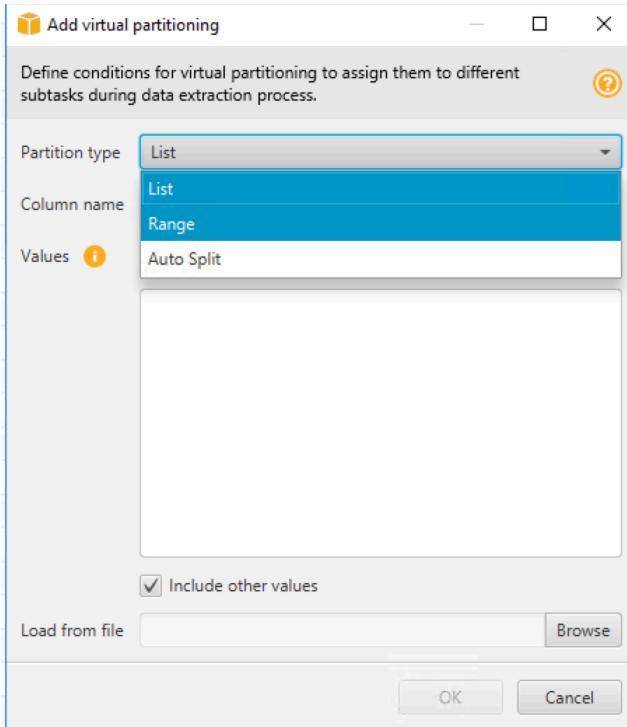
Using RANGE Virtual Partitioning to Migrate the PERSON Table

Steps:

1. Uncheck all objects from DMS_SCHEMA_DW schema from source side in the AWS SCT UI
2. Select 'DIM_PERSON' table, right click and select 'Add Virtual Partitioning'.

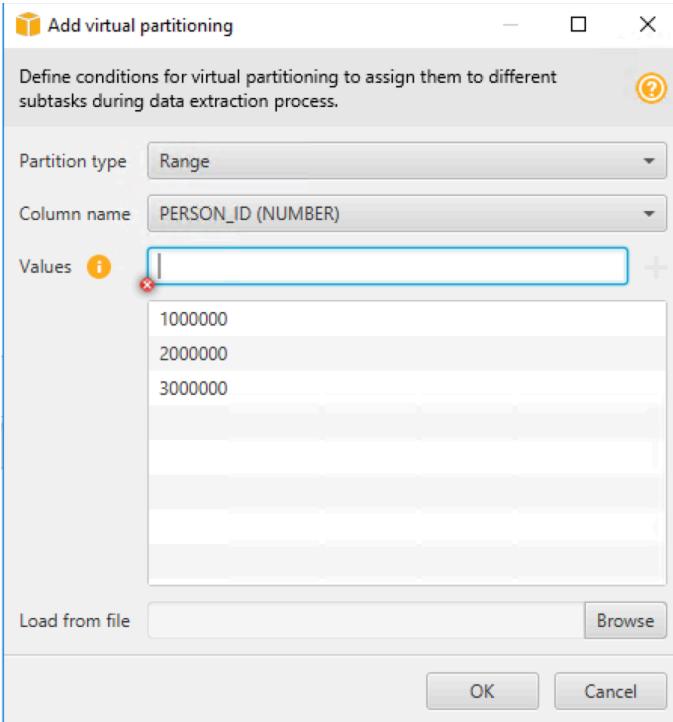


3. Select 'Range' for the 'Partition Type' option.



4. Select the ID Column, Enter Value as '1000000' and click on '+' icon or press enter.

5. Similarly add the 2000000, 3000000 values and press enter.



Note the details displayed under ‘Virtual Partitions’ tab in SCT.

The screenshot shows the AWS Schema Conversion Tool interface with two main panes: Oracle DW on the left and Amazon Redshift on the right. The Oracle DW pane displays a tree view of schemas, tables, and partitions. The Amazon Redshift pane shows the corresponding schema structure. The 'Virtual partitions' tab is selected in the Oracle DW pane, which displays the partitioning details for the 'DIM_PERSON' table. The table is partitioned by range on the 'PERSON_ID' column, with values from [1000000, 2000000, 3000000]. The partitions are defined as follows:

- PERSON_ID <= 1000000
- PERSON_ID > 1000000 AND PERSON_ID <= 2000000
- PERSON_ID > 2000000 AND PERSON_ID <= 3000000
- PERSON_ID > 3000000

We will now migrate the data in ‘RANGE – Virtual Partitioned’ table ‘DIM_PERSON’ to Amazon Redshift.

- Right Click the “DIM_PERSON” table in AWS SCT and select “Create Local Task”. The required steps would be similar to the ones for migrating the other tables in the previous section. Please select ONLY the ‘DIM_PERSON’ table.

The 'Create local task' dialog box is shown with the following settings:

- General Tab:**
 - Task name:** LocalTask-DIM-PERSON
 - Migration mode:** Extract, upload and copy
 - Extract LOBs
 - Logging:**
 - Enable task logging
 - Logging level:** TRACE
- Buttons:** Test Task, Create, Cancel

The screenshot shows the AWS SCT UI interface. At the top, there is a table for the task "LocalTask-DIM-PERSON" with three subtasks: Extract, Upload, and Copy, all at 0% completion. Below this is a toolbar with buttons for Download log, Start, Stop, Restart, Delete, Refresh all, and Refresh selected. A detailed view of the "LocalTask-DIM-PERSON" task is shown in a modal window. The "Task details" tab is selected, displaying the following configuration:

ID:	e40013becad84167bea84312297f7684
Task name:	LocalTask-DIM-PERSON
Number of subtasks:	0
Extractors log path:	
Task logging:	TRACE
Extract LOBs:	true
Keep files on S3:	false
Delimiter character:	
Compression format:	GZIP
S3 bucket LOBs folder:	
Sorting strategy:	SORTING_AFTER_FAIL
Use Snowball:	false

7. Click on the “Start” button above to start the AWS SCT Extractor Agent Task from the tool.
8. Click on a specific Task (such as Extract or Upload) in SCT UI and navigate the “SubTasks” tab.

The screenshot shows the AWS SCT UI interface. The "SubTasks" tab is selected for the "LocalTask-DIM-PERSON" task. The table displays four rows of subtask data:

ID	Parent agent	Source name	Target name	Status	C...
4eb66473f1f740cd84b5...	dw-sct-agent-2	DMS_SAMPLE_DW.DIM...	dms_sample_dw.dim_person	COMPLETED	1...
72c2174f2f9b4144b98d...	dw-sct-agent-1	DMS_SAMPLE_DW.DIM...	dms_sample_dw.dim_person	COMPLETED	1...
f29e39f55a0046a58745...	dw-sct-agent-2	DMS_SAMPLE_DW.DIM...	dms_sample_dw.dim_person	RUNNING	75%
53daebddb5184a038a7...	dw-sct-agent-1	DMS_SAMPLE_DW.DIM...	dms_sample_dw.dim_person	COMPLETED	1...

You will see that there are four threads running in parallel to extract and move the data to the Amazon S3 bucket. This is because we have created four virtual partitions for the “DIM_PERSON” table. These 4 concurrent threads are divided equally between 2 agents.

NOTE: To move data into Amazon Redshift directly, you would select “Extract, Upload and Copy’ option while creating the task.

Verify that your data migration completed successfully

After the migration is complete, connect to SQL Workbench and your Redshift cluster and verify counts in the following tables

Query	Count
select count(*) from dim_person;	7055276
select count(*) from dim_player;	5157

select count(*) from dim_sport_location_seats;	3565082
select count(*) from dim_sport_team;	62
select count(*) from dim_sporting_event;	1142
select count(*) from fact_sport_event_ticket_info;	56530154

Conclusion

You have completed the workshop. In this workshop you have created a simulated on-premise Oracle source database and used AWS SCT to convert your schema and migrate your data.

Delete all resources

Make sure you delete the Replication instance. Once the replication instance is deleted you should be able to delete the CloudFormation stack, which will in turn delete all resources.

Note: If you do not delete resources they will keep incurring charges.

1. Open the **CloudFormation** console at:

<https://console.aws.amazon.com/cloudformation/home>

2. Select the checkbox for the stack you created for this lab.
3. Select **Actions** and **Delete Stack**. Select **Yes, Delete**. CloudFormation will delete all resources created.