



Workshop

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

Table of Contents

| | |
|---|-----------|
| LAB OVERVIEW..... | 3 |
| SETUP WORKSHOP ENVIRONMENT USING AWS EVENT ENGINE | 4 |
| CONNECT TO YOUR ENVIRONMENT | 6 |
| INSTALL AWS SCHEMA CONVERSION TOOL AND VERIFY CONNECTIVITY | 8 |
| VALIDATE THE SCHEMA CONVERSION AND WORK THROUGH CONVERSION ISSUES | 18 |
| IAM ACCESS KEY & ACCESS SECRET KEY FOR SCT AUTHENTICATION | 21 |
| STEPS TO CREATE A GLOBAL PROFILE: | 22 |
| STEPS FOR SSH LOGIN TO AGENT EXTRACTOR HOSTS | 23 |
| STEPS FOR INSTALLING AND CONFIGURING THE SCT EXTRACTION AGENTS ON HOST 2 | 25 |
| REGISTERING EXTRACTION AGENTS WITH AWS SCT TOOL..... | 25 |
| CREATING, RUNNING AND MONITORING THE AWS SCT DATA EXTRACTION TASKS..... | 28 |
| MIGRATING LARGE TABLES USING THE VIRTUAL PARTITIONING FEATURE..... | 32 |
| USING RANGE VIRTUAL PARTITIONING TO MIGRATE THE PERSON TABLE..... | 32 |
| VERIFY THAT YOUR DATA MIGRATION COMPLETED SUCCESSFULLY | 36 |
| CONCLUSION..... | 36 |
| DELETE ALL RESOURCES | 37 |

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 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.

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 while 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.

AWS DMS can also be used for continuous data replication with high- availability.

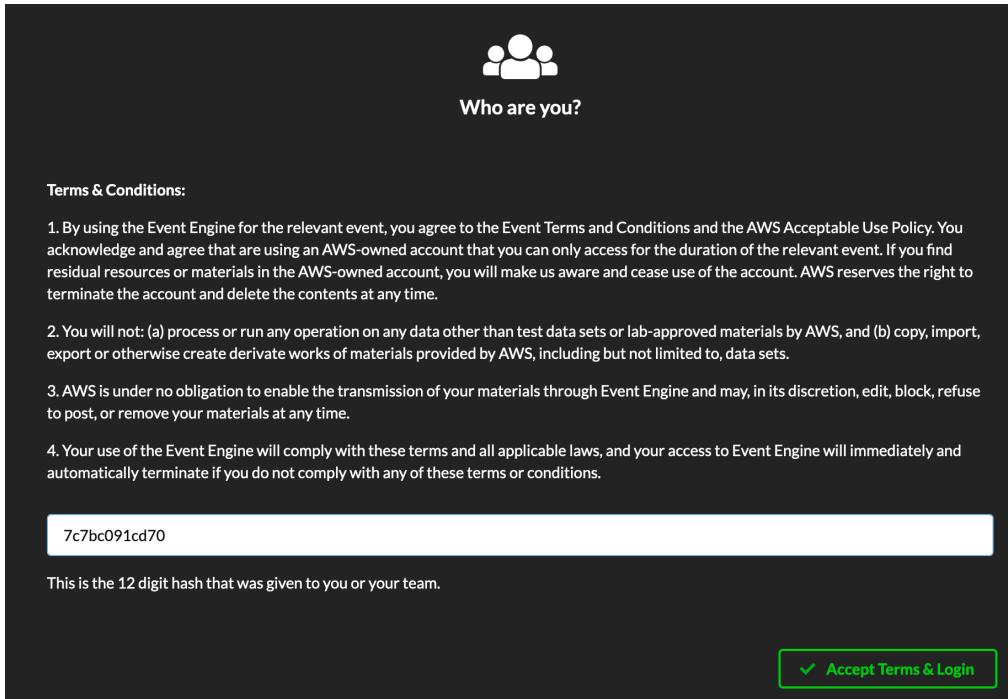
This lab will walk you through migrating a sample Oracle data warehouse to Amazon Redshift.

Setup Workshop Environment using AWS Event Engine

AWS Event Engine has provisioned an AWS account with following resources.

- An Oracle RDS Data warehouse.
 - ticketing data related to football and baseball games.
 - Five dimension tables:
 1. Dim_Sport_locaton_Seats,
 2. Dim_Person, Dim_Sporting_
 3. Event,
 4. Dim_Sport_Team,
 5. Dim_Player;
 - One fact table:
 1. Fact_Sport_Event_Ticket_Info.
 - Other database objects.
- An EC2 Windows instance with:
 - SCT installation msi, drivers for Oracle and Redshift and SQL Workbench on the Desktop. You will need to install the SCT on the EC2 instance per the instructions in this document.
- Two EC2 Linux instance with:
 - SCT Data extraction agent in the /home/ec2- users/agents folder. AWS SCT Data Extraction agents RPM You will need to install the agents and configure the agents to extract your data and move to Amazon Redshift.
- An Amazon Redshift 1 node cluster – this will be used as a target for the migration

1. Open AWS Event Engine dashboard: <https://dashboard.eventengine.run/dashboard>
Enter "Team Hash" code, provided by lab instructor, and click "Accept terms & Login".



2. Click on "SSH Key" and download EC2 SSH KEY (ee-default-keypair.pem). You will be using this later.

- ### Team Dashboard

Event

Set Team NameAWS ConsoleSSH Key
3. Click "AWS Console" to access AWS console of the AWS account created through Event engine.
 4. Open the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation/home>
 5. Review the CloudFormation **Stack** created by AWS Event engine. Do not continue until the you see status as **CREATE_COMPLETE**.

6. 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.

| | |
|---------------------|--|
| TKORedshiftEndpoint | End point for the Redshift cluster |
| TKOVpcId | VPC ID of the newly created VPC |
| TKOOracleEndpoint | End point of the source RDS database |
| TKOSCTDNS | End point for the EC2 instance with SCT and drivers installed. |
| TKOAGENT1PublicDNS | Public end point for first EC2 instance with SCT Extraction agent. |
| TKOAGENT1PrivateIP | Private IP for first EC2 instance with SCT Extraction agent. |
| TKOAGENT2PublicDNS | End point for second EC2 instance with SCT Extraction agent. |
| TKOAGENT2PrivateIP | Private IP for second EC2 instance with SCT Extraction agent. |
| TKOS3Bucket | S3 Bucket name |

The screenshot shows the AWS CloudFormation console with the 'Outputs' tab selected. On the left, the 'Stacks (1)' section displays a single stack named 'mod-279446afb01246d1' in 'Active' status, with a green 'CREATE_COMPLETE' button. The main area shows the 'Outputs (9)' table with the following data:

| Key | Value | Description |
|---------------------|--|--|
| TKOAGENT1PrivateIP | 172.31.0.236 | Private IP of the TKO SCT Agent1 EC2 linux instance |
| TKOAGENT1PublicDNS | ec2-34-254-179-239.eu-west-1.compute.amazonaws.com | Public DNS Name of the TKO SCT Agent1 EC2 linux instance |
| TKOAGENT2PrivateIP | 172.31.0.17 | Private IP of the TKO SCT Agent2 EC2 linux instance |
| TKOAGENT2PublicDNS | ec2-34-244-138-173.eu-west-1.compute.amazonaws.com | Public DNS Name of the TKO SCT Agent2 EC2 linux instance |
| TKOOracleEndpoint | mod-279446afb01246d1-oracleinstance.csjzogwk5vv9.eu-west-1.rds.amazonaws.com | Endpoint of source Oracle RDS DB instance |
| TKORedshiftEndpoint | mod-279446afb01246d1-redshifttargetcluster-1r71s1bjfjorm.cztnntf65wba.eu-west-1.redshift.amazonaws.com | Endpoint of target Redshift instance |
| TKOS3Bucket | mod-279446afb01246d1-s3bucket-13xtncdgxkpd | TKO workshop S3 bucket |
| TKOSCTDNS | ec2-34-243-188-212.eu-west-1.compute.amazonaws.com | Public DNS Name of the TKO SCT Windows instance |
| TKOVpcId | vpc-0362457e105272aec | TKO workshop VPC Identifier |

Connect to your environment

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

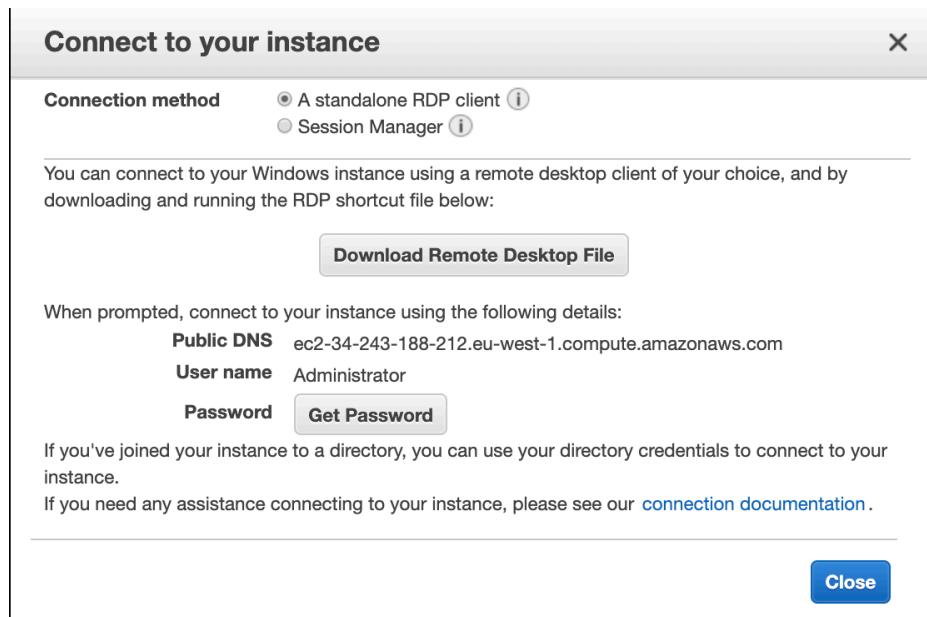
7. Once CloudFormation has completed building the stack,
- Go to the EC2 console located at: <https://console.aws.amazon.com/ec2/v2/home> and
 - click **Instances** in the left column

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

| Name | Instance ID | Instance Type | Availability Zone | Instance State | Status Checks | Alarm Status | Public DNS (IPv4) | IPv4 Public IP | IPv6 IP |
|---|---------------------|---------------|-------------------|----------------|----------------|--------------|--------------------------|----------------|---------|
| mod-279446afb01246d1-SCTWinEC2 | i-07c05d72e6feb130 | m5.large | eu-west-1a | running | 2/2 checks ... | None | ec2-34-243-188-212.eu... | 34.243.188.212 | - |
| mod-279446afb01246d1-TKOAGENTINSTANCE-2 | i-07e13c2270ac049c5 | t2.xlarge | eu-west-1a | running | 2/2 checks ... | None | ec2-34-244-138-173.eu... | 34.244.138.173 | - |
| mod-279446afb01246d1-TKOAGENTINSTANCE-1 | i-0c10eadcdebd24a45 | t2.xlarge | eu-west-1a | running | 2/2 checks ... | None | ec2-34-254-179-239.eu... | 34.254.179.239 | - |

9. 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.

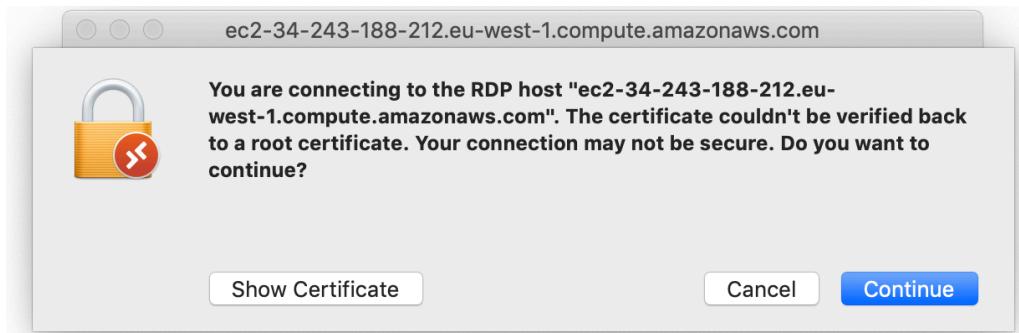
Note: You **do not** need to use the '**Get Password**' button for this lab. The password is provided in step 10.



10. 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**".

Note: Following security warning may come when you connect to the Windows EC2. If appears, you can proceed with "Continue".



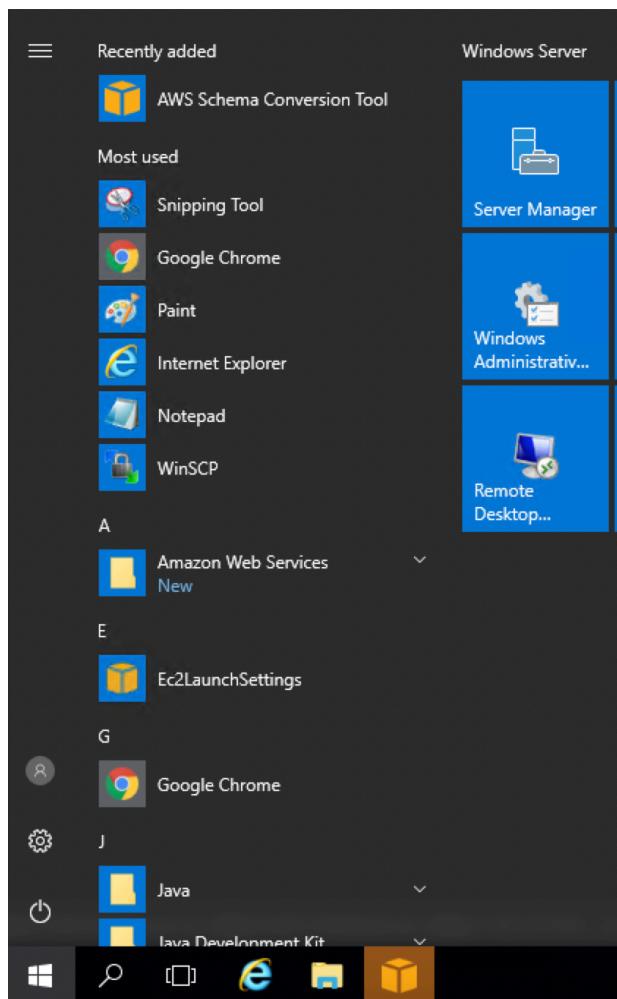
Install AWS schema Conversion Tool and verify connectivity

Once connected:

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

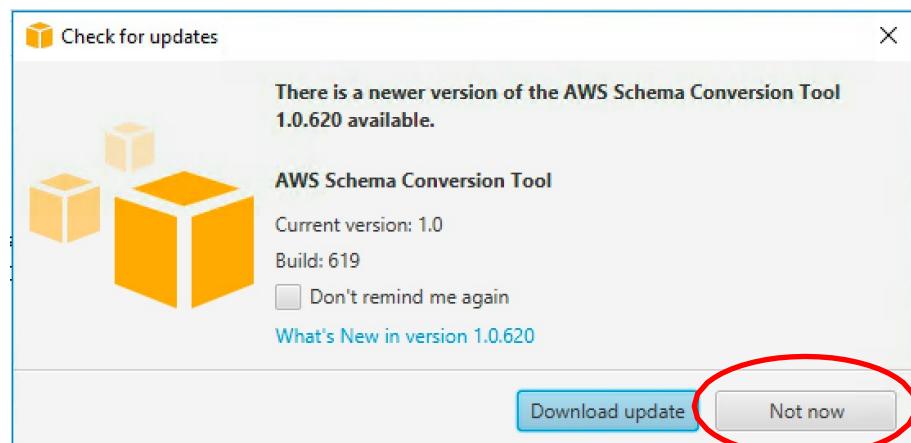
1. After clicking AWS Schema Conversion Tool,

- 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:

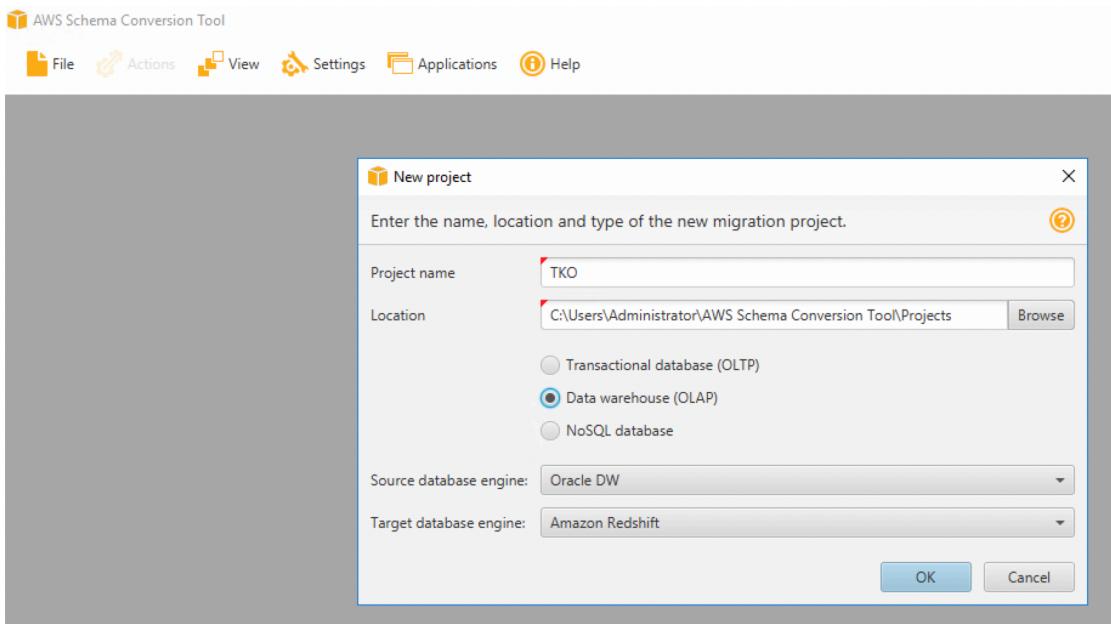


- Click on Schema Conversion Tool to open the application.

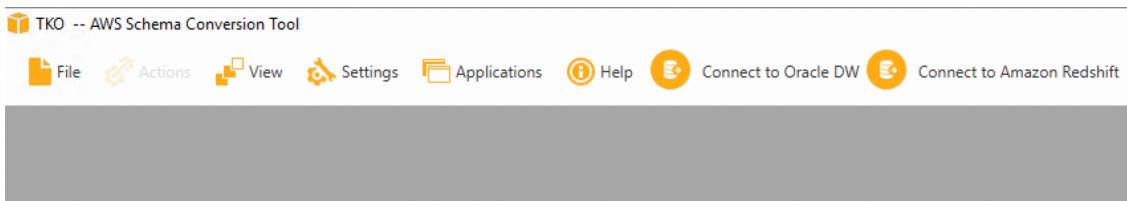
You may see a message that asks to download new version, click [Not now](#) button and proceed. If you do upgrade then your screen shots may vary from what we have in the lab document.



2. 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 **TKO**) 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**.



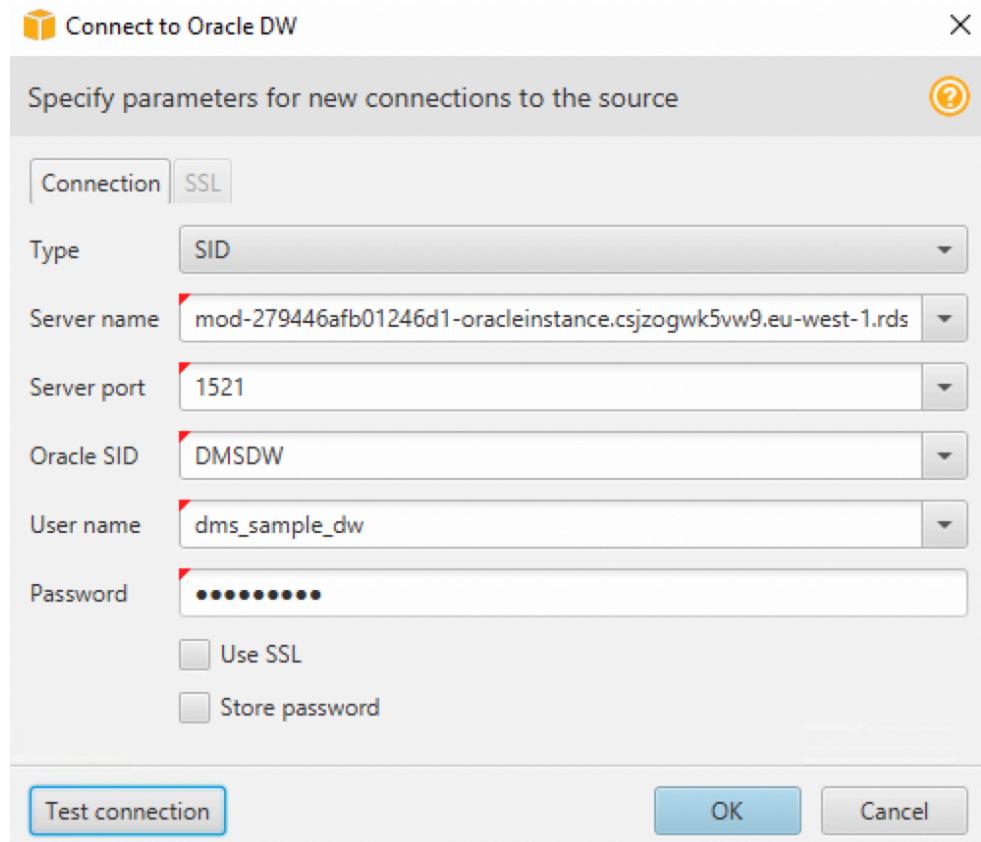
3. Click on **Connect to Oracle DW** on the top menu.



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

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

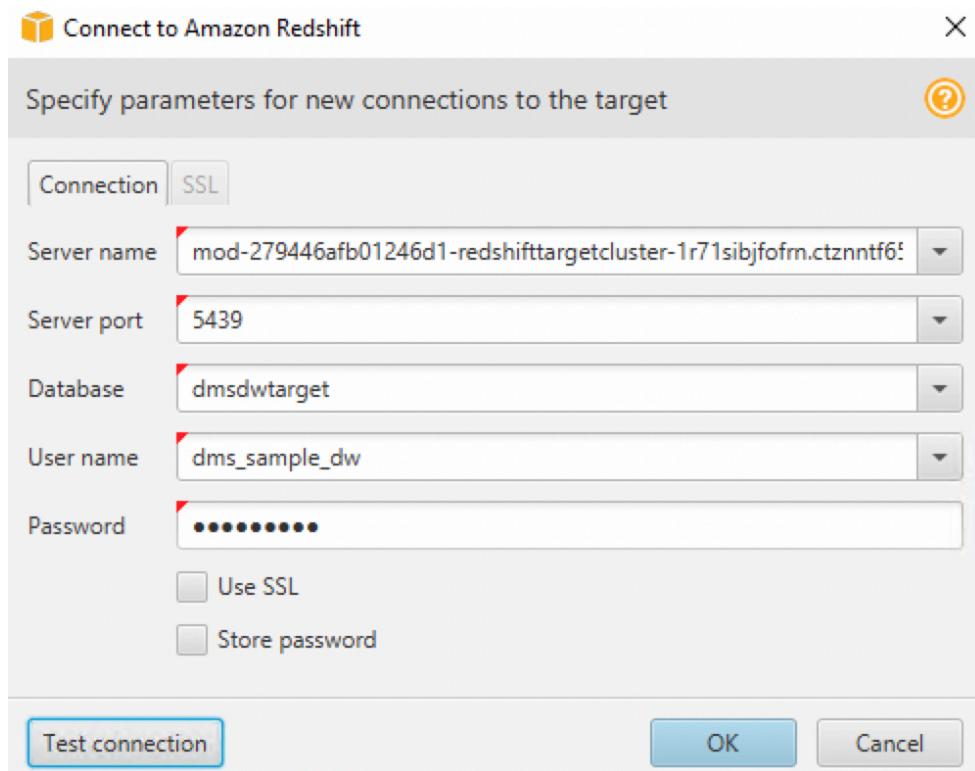
| | |
|--------------------|--|
| Type | SID |
| Server name | Take this from the output of the CloudFormation template in Step 9. It is the TKOOracleEndpoint output field. |
| Server port | 1521 |
| Oracle SID | DMSDW |
| User name | dms_sample_dw |
| Password | Password1 |
| Oracle Driver Path | Specify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



Click **OK**.

5. 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.

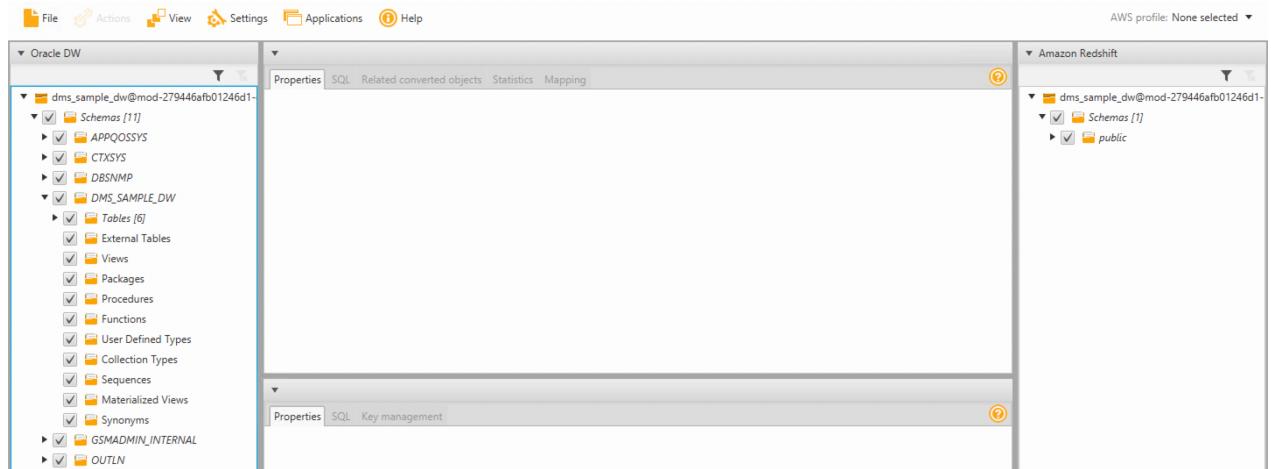
| | |
|-----------------------------|--|
| Server name | Take this from the output of the CloudFormation template in Step 9. It is the TKORedshiftEndpoint output field. |
| Server port | 5439 |
| Database | dmsdwtarget |
| User name | dms_sample_dw |
| Password | Password1 |
| Amazon Redshift Driver Path | C:\Users\Administrator\Desktop\ANT371\RedshiftDrivers\RedshiftJDBC42- 1.2.16.1027.jar |



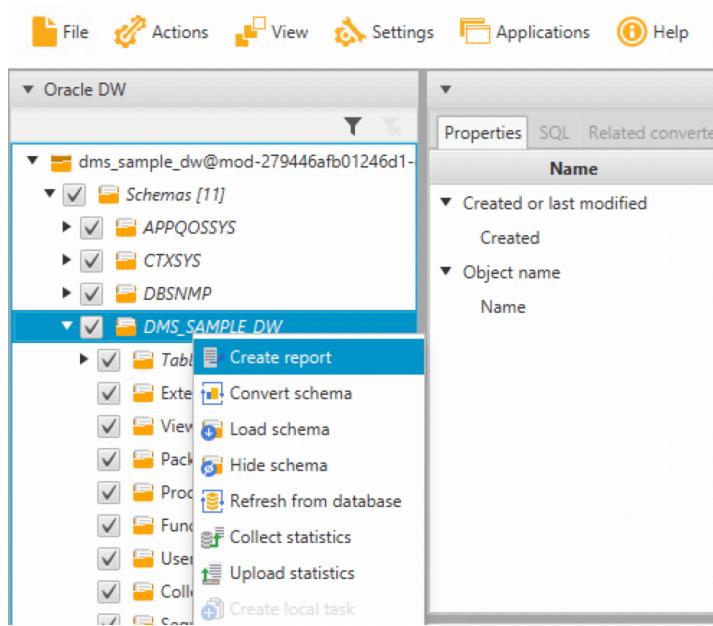
Click **OK**.

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.



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



3. 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.

Database migration assessment report

Source database:
DMS_SAMPLE_DW.dms_sample_dw@mod-279446afb01246d1-oracleinstance.csjzogwk5vw9.eu-west-1.rds.amazonaws.co
m: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 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 | Percentage | Count |
|-------------|------------|-------|
| Schema | 100% | 1 |
| Table | 100% | 6 |
| Constraint | 100% | 6 |

You can click on **Save to pdf** to save this report to refer to it at a later time.

4. Choose the Action Items tab.

No issues were found.

| Name | Created or last modified | Name | Category |
|---------------|--------------------------|------|----------|
| DMS_SAMPLE_DW | 2017-03-28 15:34:47.0 | | |
| Object name | DMS_SAMPLE_DW | Name | Schemas |

In this example, you don't see any action items; but In a real-life scenario, you will need to work through these issues and resolve them before moving on.

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

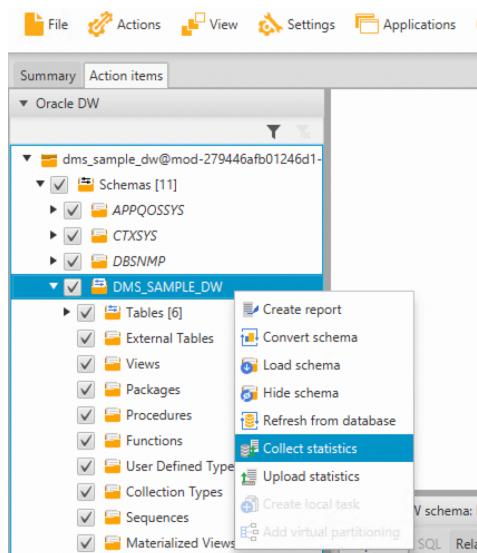
Amazon redshift doesn't support stored procedures – This will need to be manually converted to a User-Defined Function (UDF) in Redshift. Additional information can be found in the link below.

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

Redshift stored procedures Supports - The latest version of SCT uses AWS Glue to convert stored procedures.

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.

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.

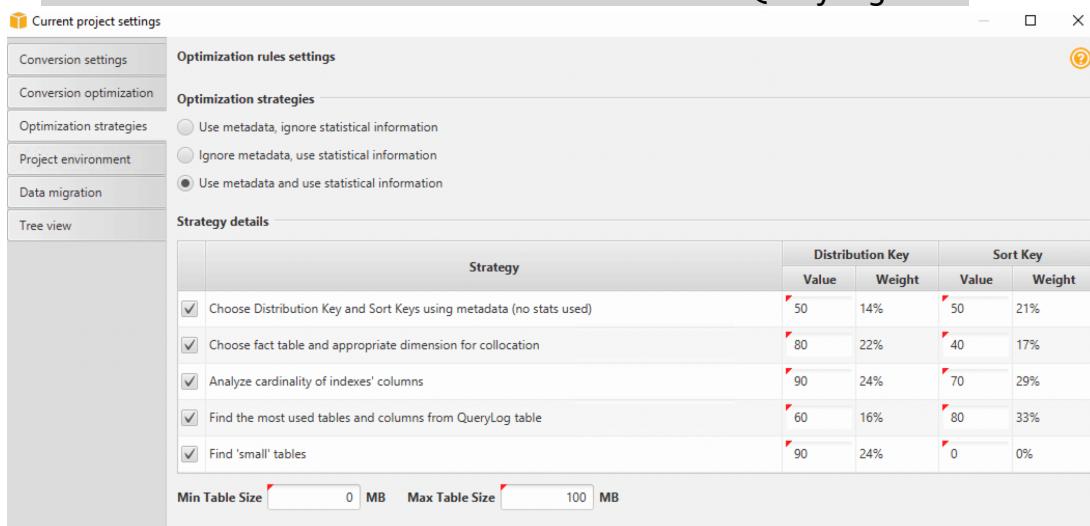
Note: This step takes **few minutes** so please wait until it completes.

6. Optimize how the AWS Schema Conversion Tool (AWS SCT) converts your data warehouse schema.

- Choose **Settings**,
- 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 can choose from the following:
 - Use metadata, ignore statistical information
 - Ignore metadata, use statistical information
 - Use metadata and use statistical information

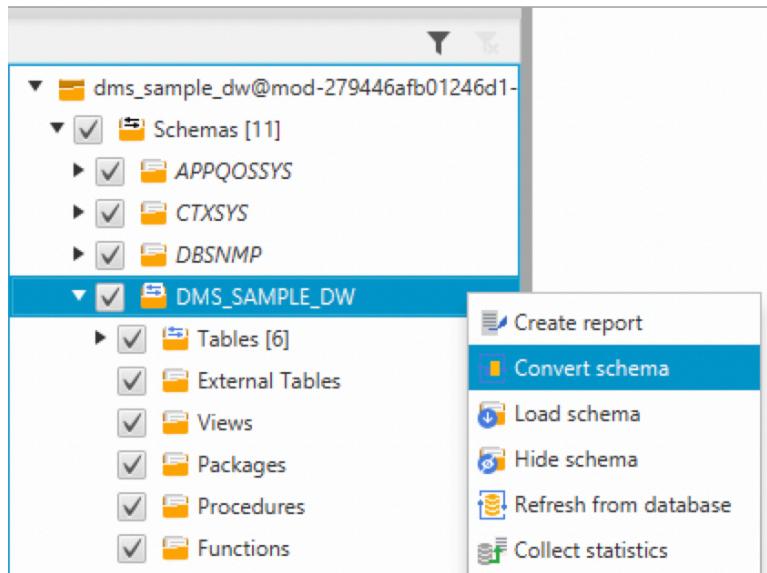
After you choose your optimization strategy, you can choose the rules you want to use and provide weights to each. You can choose from the following:

- 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**.

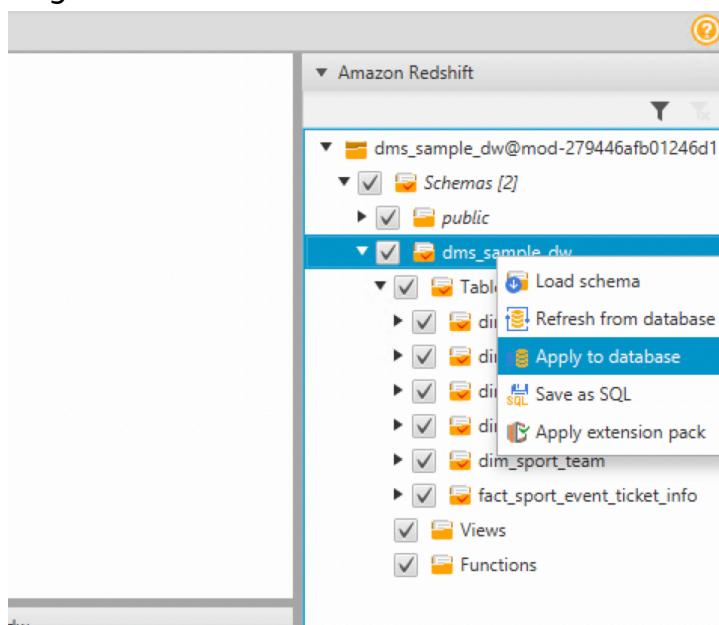


Choose **Yes** for the confirmation message.

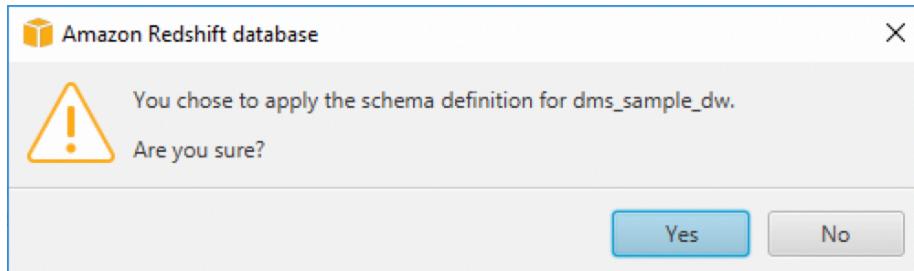
AWS SCT then converts your schema to the target database format.

8. In the **Amazon Redshift view** (right-hand side)

- Open the context (right-click) menu for the **dms_sample_dw** schema,
- 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,
- Then choose **Refresh from Database** to refresh from the target database.



- Click **Yes** to confirm.

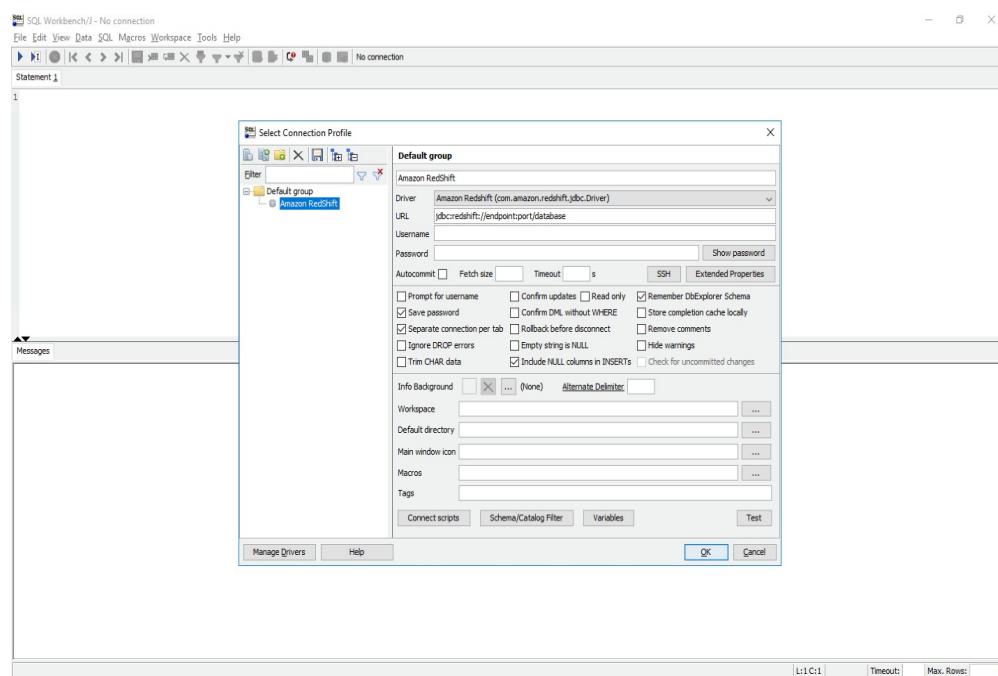
The database schema has now been converted and imported from source to target.

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 TKOOracleEndpoint from step 8 Example: jdbc:oracle:thin:@TKO-oracleinstance.cs0mrnzrs0oa.eu-west-1.rds.amazonaws.com: 1521:DMSDW |
| Username | dms_sample_dw |
| Password | Password1 |

4. Check **AutoCommit** and
5. Click **OK**. **OracleDW** will connect to the Source Oracle database.
6. 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 TKORedshiftEndpoint from step 9. Its format is as follows: jdbc:redshift:// Redshift Endpoint :5439/dmsdwtarget |
| Username | dms_sample_dw |
| Password | Password1 |

7. In SQL Workbench/J, choose **File**, then choose **Connect window**. Choose the **RedshiftConnection** you created in an earlier step. Choose **OK**.
8. 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:

| object_type | object_name | object_schema |
|-------------|------------------------------|---------------|
| TABLE | dim_person | dms_sample_dw |
| TABLE | dim_player | dms_sample_dw |
| TABLE | dim_sport_location_seats | dms_sample_dw |
| TABLE | dim_sport_team | dms_sample_dw |
| TABLE | dim_sporting_event | dms_sample_dw |
| TABLE | fact_sport_event_ticket_info | dms_sample_dw |

- 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 |

IAM Access Key & 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.

CloudFormation stack has already created IAM user “SCTUser” with the programmatic access.

Note down IAM Access key and Secret access key from CloudFormation output Stack.

The screenshot shows the AWS CloudFormation Outputs page for a stack named "mod-279446afb01246d1". The "Outputs" tab is selected. There are two output entries:

| Key | Value | Description | Export name |
|-----------|---------------|---------------------------------------|-------------|
| AccessKey | AKI[REDACTED] | AWS AccessKey of IAM user SCTUser: | - |
| SecretKey | sek[REDACTED] | Secret AccessKey of IAM user SCTUser: | - |

Also note down S3 bucket name.

The screenshot shows the AWS CloudFormation Outputs page for the same stack "mod-279446afb01246d1". The "Outputs" tab is selected. There is one output entry:

| Key | Value | Description | Export name |
|-------------|--|------------------------|-------------|
| TKOS3Bucket | mod-279446afb01246d1-s3bucket-4uk81ogo1jd1 | TKO workshop S3 bucket | - |

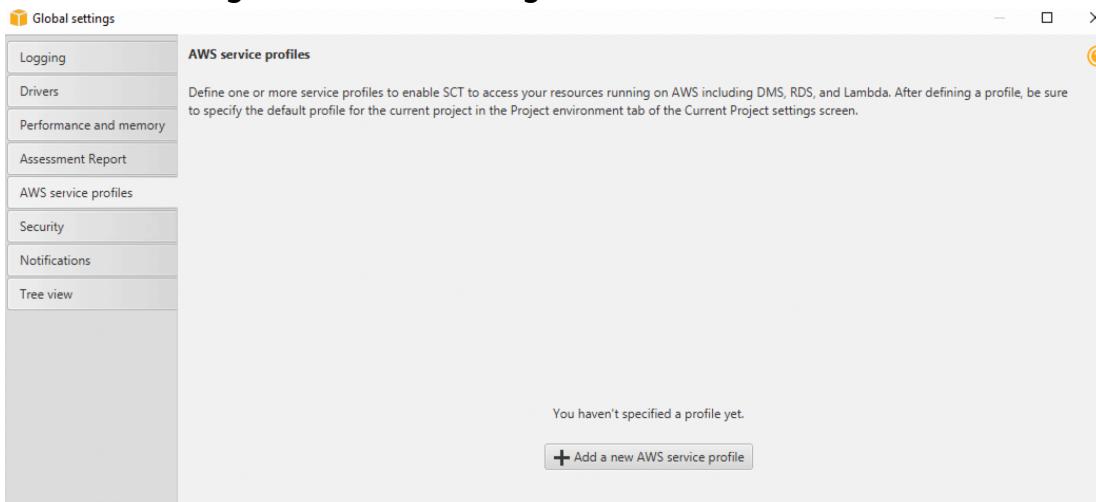
AWS SCT Global Settings and Amazon S3 Bucket Configuration

In this section, we will install and configure the SCT Extraction Agents. Agents will extract your data, 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.

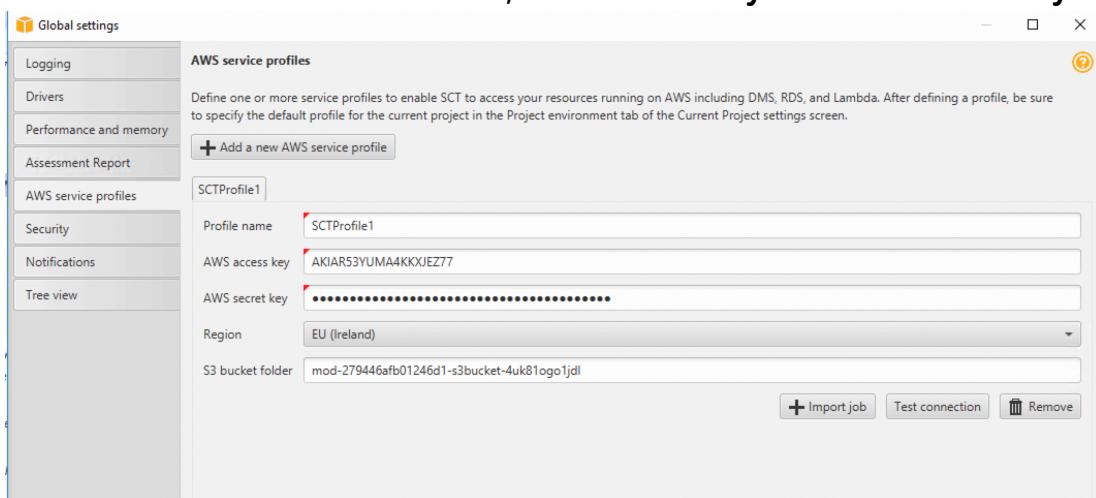
Steps to Create a Global Profile:

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**



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 "**TKOS3Bucket**" name from the **output tab of the CloudFormation** console.
6. 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 Redhat Linux 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

1. If you are using windows, please follow the instructions here:
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>
2. Login to AWS Management Console and note the public IP address of Agent1 instance.

| Name | Instance ID | Instance Type | Availability Zone | Instance State | Status Checks |
|---|---------------------|---------------|-------------------|----------------|----------------|
| mod-279446afb01246d1-TKOAGENTINSTANCE-2 | i-0c9795e4826f9b9f8 | t2.xlarge | eu-west-1a | running | 2/2 checks ... |
| mod-279446afb01246d1-TKOAGENTINSTANCE-1 | i-0c0048355sec3ae68 | t2.xlarge | eu-west-1a | running | 2/2 checks ... |
| mod-279446afb01246d1-SCTWinEC2 | i-086dcc6cc46a83ecc | m5.large | eu-west-1a | running | 2/2 checks ... |

3. 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>
```

```
38f9d34fa093:~ sudhig$ ssh -i "ee-default-keypair.pem" ec2-user@ec2-54-154-169-43.eu-west-1.compute.amazonaws.com
Last login: Mon Nov 12 19:41:30 2018 from 72-21-198-67.amazonaws.com
[ec2-user@ip-172-31-0-195 ~]$
```

- 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.

7. 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

| Name | Value |
|--------------------------|-----------------------|
| Created or last modified | 2017-03-28 15:34:47.0 |
| Object name | DMS_SAMPLE_DW |
| Name | DMS_SAMPLE_DW |

3. From the central panel, click "Register" button under "Agents" tab.

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

New agent registration

Connection
SSL

Description: dw-sct-agent-1

Host name: 172.31.0.195

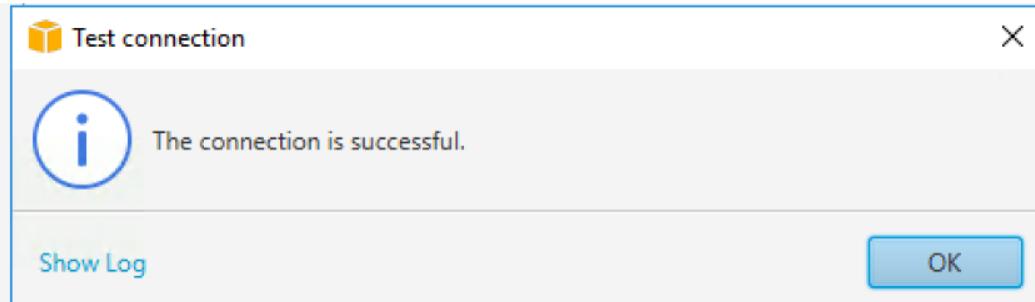
Port: 8192

Use SSL

Test Connection
Register
Cancel

Note: 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 “**TKOAGENT1PrivateIP**” and “**TKOAGENT2PrivateIP**” outputs in the CloudFormation console.

5. Click on “Test Connection”



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

| ID: | f64c0f29ea3d475d85eb31e;9053bd32 |
|--------------|----------------------------------|
| Agent name: | ef2f5e02bd77460ca9355b0fe01f9eb |
| Description: | dw-sct-agent-1 |
| Version: | 1.0.620 |
| Host name: | 172.31.0.195 |
| Port: | 8192 |
| Status: | ACTIVE |
| Uptime: | 0m |
| Last update: | 01-02-2020 05:44 |
| SSL: | false |

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:

The screenshot displays the AWS SCT GUI interface. On the left, the Oracle DW database is shown with a tree view of schemas like APPQOSSYS, CTXSYS, DBSNMP, and DMS_SAMPLE_DW. Under DMS_SAMPLE_DW, there are tables, external tables, views, packages, procedures, functions, user-defined types, collection types, sequences, materialized views, synonyms, and applications. The Agents tab is selected, listing two agents: dw-sct-agent-1 and dw-sct-agent-2. dw-sct-agent-2 is active, with details: ID: 6d9414bf0f0546a7a6681207ade00311, Agent name: 1b08cec6234d4d3f8e824eb3d9b5c911, Description: dw-sct-agent-2, Version: 1.0.620, Host name: 172.31.0.32, Port: 8192, Status: ACTIVE, and Uptime: 0m. The Tasks, Virtual partitions, and Snowball tabs are also visible. On the right, the Amazon Redshift database is shown with a tree view of schemas like public and dms_sample_dw, and tables like aws_oracle_ext.

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.

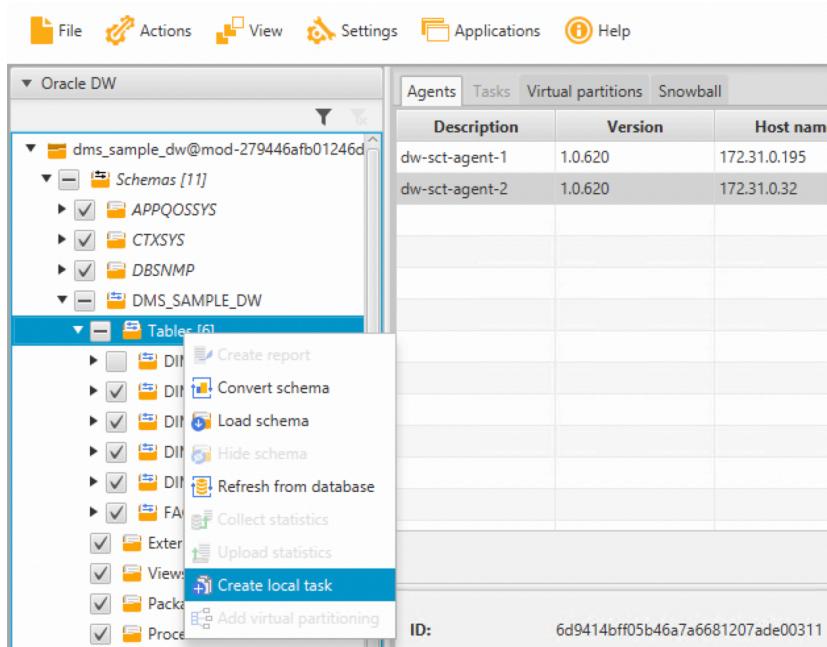
Creating, Running and Monitoring the AWS SCT Data Extraction Tasks

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 that you create multiple tasks for multiple tables based on the size of the tables in your data warehouse.

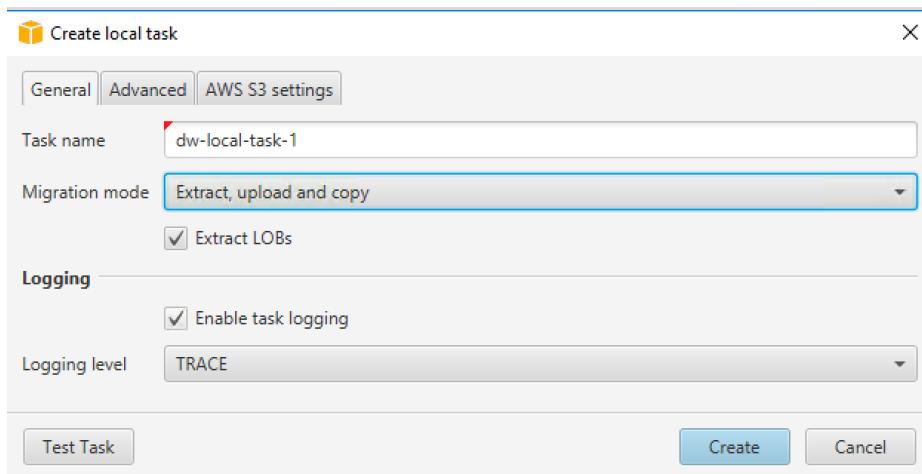
1. In the AWS SCT UI, for the current TKO project in the source oracle database tab, select all the tables under the “DMS_SAMPLE_DW”. Now uncheck only the “DIM_PERSON” table. This table is very large and we will be using Virtual Partitioning feature to migrate this table. Overall the final table selection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

The screenshot shows the AWS SCT UI with the Oracle DW database selected. The DMS_SAMPLE_DW schema is expanded, showing its tables: DIM_PERSON, DIM_PLAYER, DIM_SPORTING_EVENT, DIM_SPORT_LOCATION_SEA, DIM_SPORT_TEAM, and FACT_SPORT_EVENT_TICKET. All tables except for DIM_PERSON have their checkboxes checked. The agent list on the right shows dw-sct-agent-1 and dw-sct-agent-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 SCT Window.



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



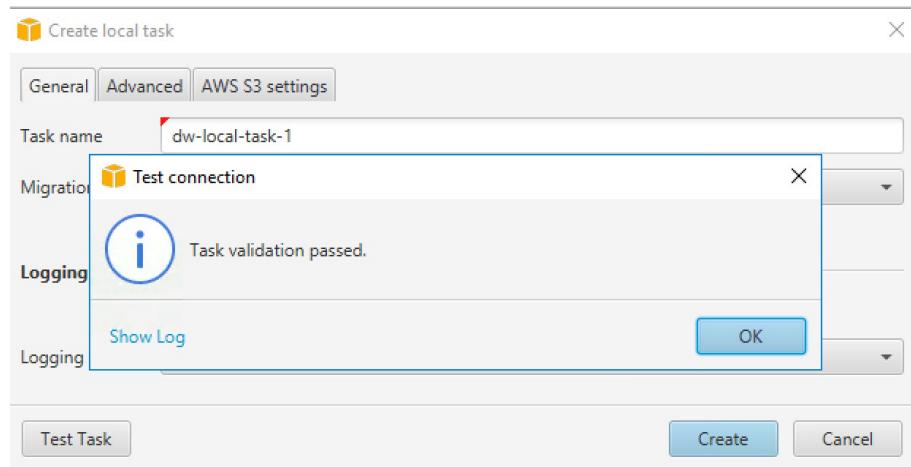
For Migration Mode, you can choose one of the following:

- 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.

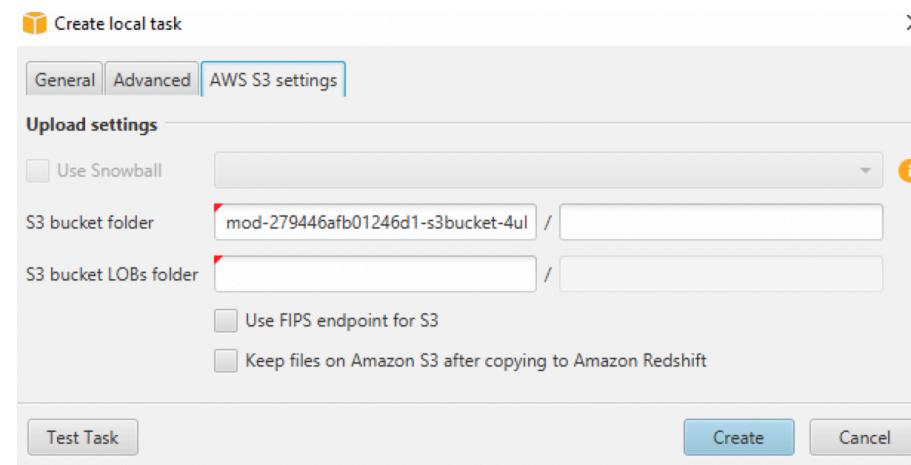
In this lab, we will select the third option, i.e. “**Extract, upload and copy**”.

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

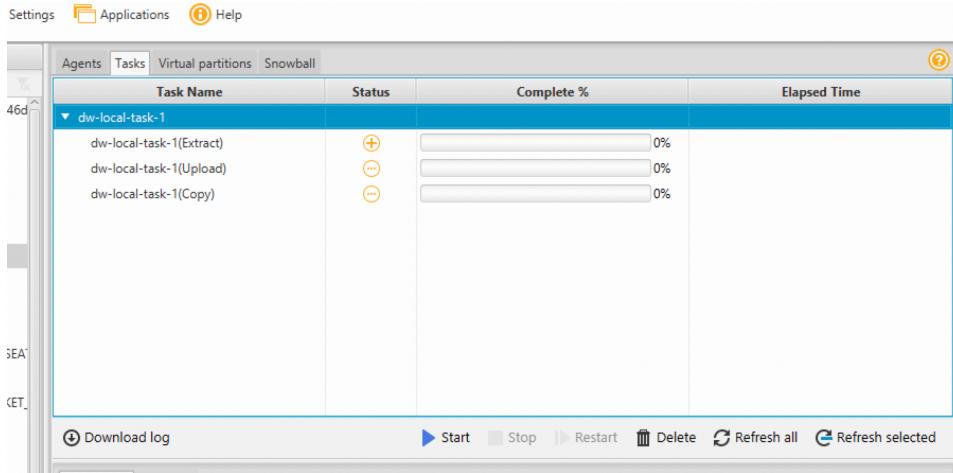
- Click ‘Test Task’. ‘Test Validation Passed’ message is displayed.



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



- Click on ‘Create’ button to create the extraction task.



7. 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.

This screenshot shows the same task 'dw-local-task-1' after it has been started. The subtasks are now in progress, indicated by orange progress bars. The 'Extract' step is at 87%, 'Upload' is at 86%, and 'Copy' is at 61%. The 'Status' column shows small orange play icons next to each subtask.

9. Click on each step (Extract, Upload, Copy) in the AWS SCT UI
10. select “Subtasks” tab.

Observe that for each table a separate subtask is created.

This screenshot shows the 'Subtasks' tab for the task. It lists individual subtasks for each table in the Oracle DW. The table on the left shows the Oracle DW schema with tables like DIM_PERSON, DIM_PLAYER, etc. The main pane lists subtasks for each table, such as 'dw-local-task-1(DIM_PERSON)', 'dw-local-task-1(DIM_PLAYER)', etc. Each subtask has a unique ID, parent agent, source name, target name, status, and a progress bar indicating completion. For example, 'dw-local-task-1(DIM_PERSON)' is completed at 100%.

Note: 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.

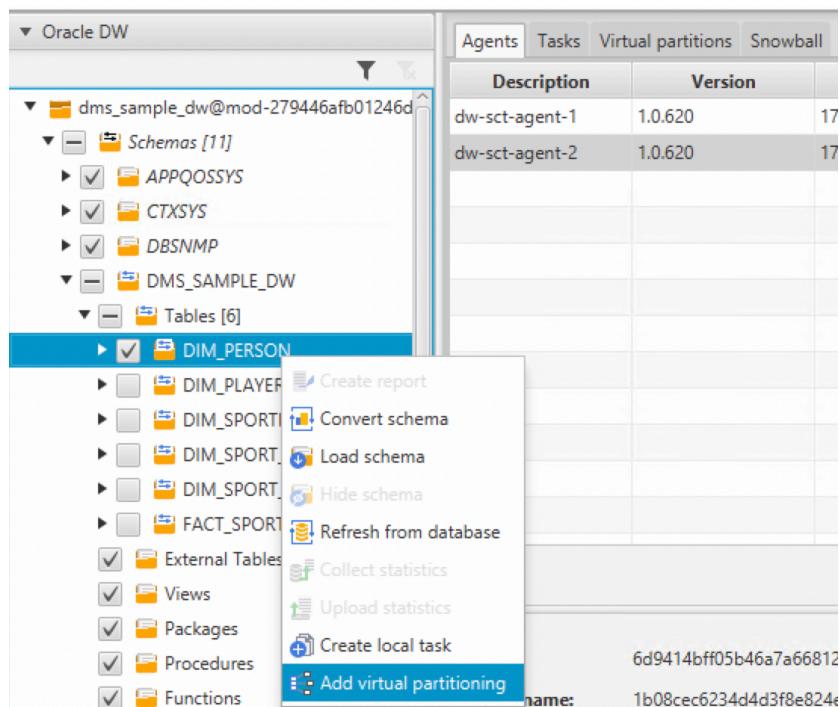
For more information on Virtual Partitioning, please refer to AWS Documentation at -

https://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/CHAP_Agents.DW.html

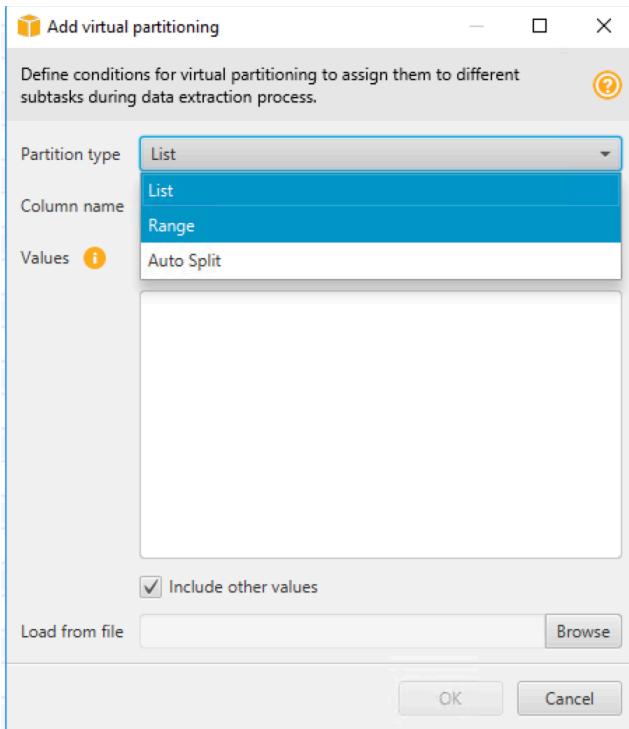
Using RANGE Virtual Partitioning to Migrate the PERSON Table

Steps:

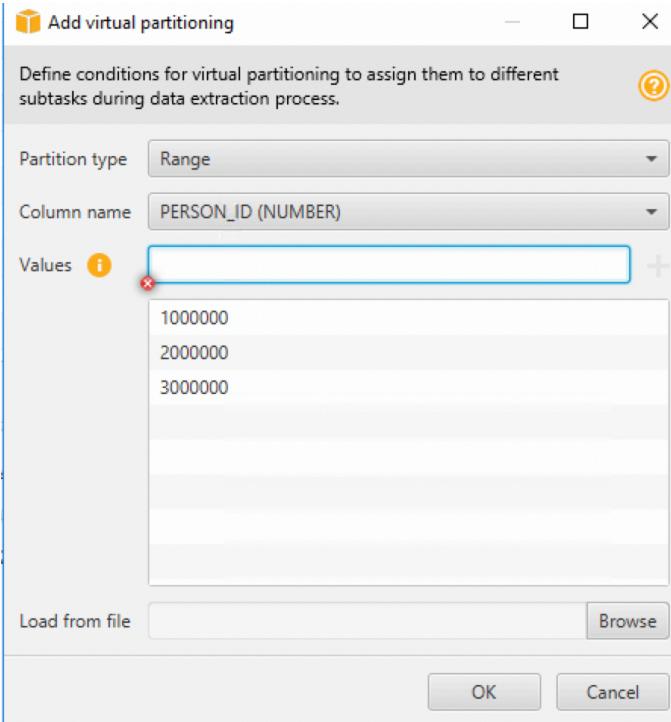
1. Uncheck all objects from DMS_SCHEMA_DW schema from source side in the AWS SCT UI
2. Select 'DIM_PERSON' table, right click and select 'Add Virtual Partitioning'.



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.

DMS_SAMPLE_DW.DIM_PERSON

Partition type: Range

Partition type: Range
Column name: PERSON_ID
Values: [1000000, 2000000, 3000000]
Partitions:
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.

Create local task

General Advanced AWS S3 settings

Task name: LocalTask-DIM-PERSON

Migration mode: Extract, upload and copy

Extract LOBs

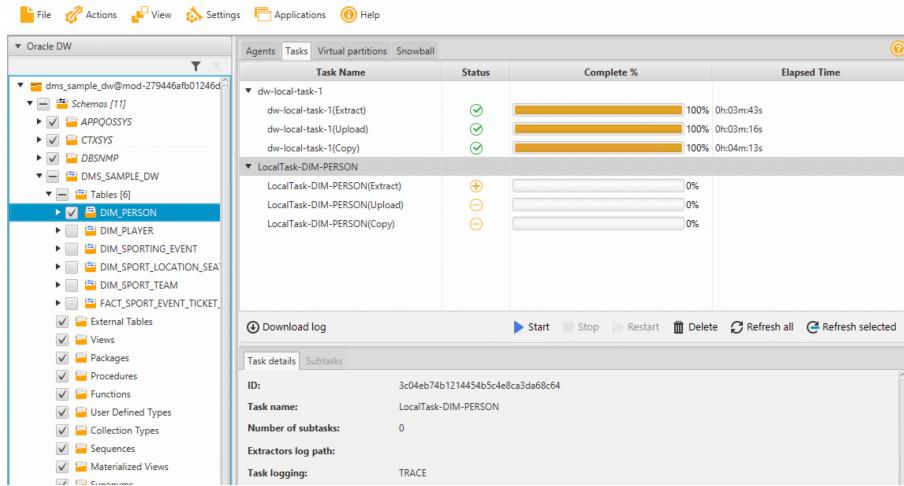
Logging:

Enable task logging

Logging level: TRACE

Test Task Create Cancel

- Click on the "Start" button above to start the AWS SCT Extractor Agent Task from the tool.



- Click on a specific Task (such as Extract or Upload) in SCT UI and navigate the "SubTasks" tab.

| Task Name | Status | Complete % | Elapsed Time |
|-------------------------------|--------|------------|--------------|
| dw-local-task-1 | | | |
| dw-local-task-1(Extract) | ✓ | 100% | 0h:03m:43s |
| dw-local-task-1(Upload) | ✓ | 100% | 0h:03m:16s |
| dw-local-task-1(Copy) | ✓ | 100% | 0h:04m:13s |
| LocalTask-DIM-PERSON | | | |
| LocalTask-DIM-PERSON(Extract) | + | 0% | |
| LocalTask-DIM-PERSON(Upload) | ○ | 0% | |
| LocalTask-DIM-PERSON(Copy) | ○ | 0% | |

Start Stop Restart Delete Refresh all Refresh selected

| Task details | Subtasks |
|--------------------------------------|----------|
| ID: 3c04eb74b1214454b5c4e8ca3da68c64 | |
| Task name: LocalTask-DIM-PERSON | |
| Number of subtasks: 0 | |
| Extractors log path: | |
| Task logging: TRACE | |

| ID | Parent agent | Source name | Target name | Status | Complete simple data % |
|-------------------------|----------------|----------------------|------------------------|-----------|--------------------------------------|
| 72cac665579e41c4a550... | dw-sct-agent-1 | DMS_SAMPLE_DW.DIM... | dms_sample_dw.dim_p... | RUNNING | <div style="width: 42%;">42%</div> |
| 7dc49362775840a8a16... | dw-sct-agent-2 | DMS_SAMPLE_DW.DIM... | dms_sample_dw.dim_p... | COMPLETED | <div style="width: 100%;">100%</div> |
| a616de5b378e4a8e8ab... | dw-sct-agent-1 | DMS_SAMPLE_DW.DIM... | dms_sample_dw.dim_p... | RUNNING | <div style="width: 50%;">50%</div> |
| f5119c1ea1fc4cd4a54e... | dw-sct-agent-2 | DMS_SAMPLE_DW.DIM... | dms_sample_dw.dim_p... | RUNNING | <div style="width: 50%;">50%</div> |

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](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.