Convert database objects with AWS Schema Conversion Tool

As outlined in Migration tools, the AWS Schema Conversion Tool (SCT) is useful to help you convert and migrate your database objects (schema, tables, views, etc.) to *either* Cloud used by BNZ (Azure, or AWS).

> ⓘ **NOTE**
>
> SCT does not migrate any data.

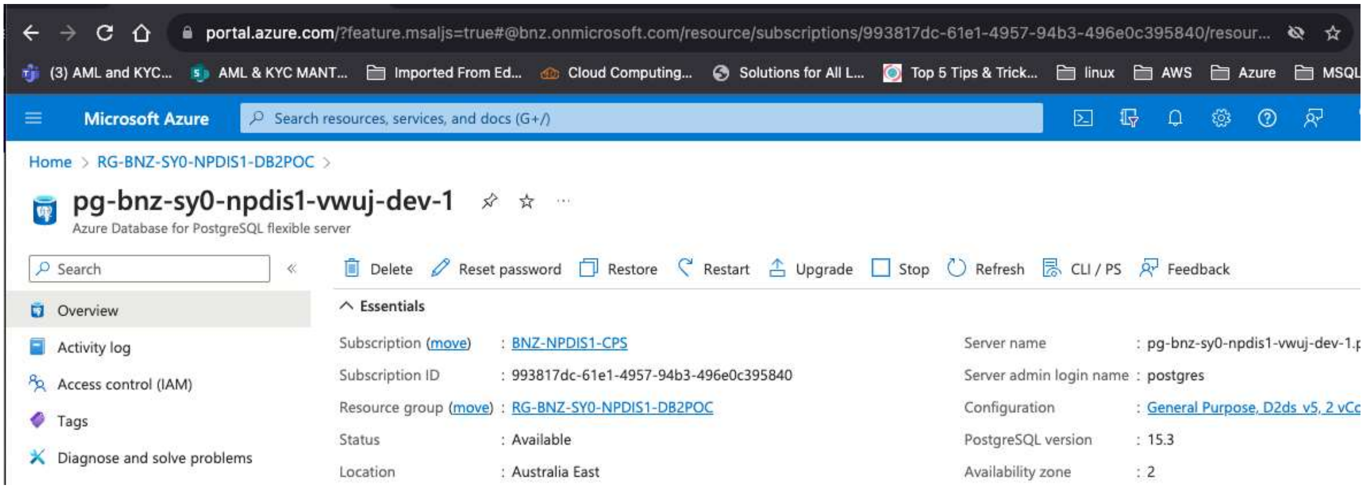
Prepare for migration includes getting an AWS account set up, opening firewalls and identifying source and target databases. In order to use SCT, you install it on an on-premise Windows server that has connectivity to your **Source** database.

In the example below we use SCT to analyse and then convert a test DB2 LUW database (the *source*), and to migrate this design to an Azure PostgreSQL database (the *target*). You can adapt this procedure for your particular database set up.

Step 1. Ensure you have already set up your environment and accesses

> **INFO**
>
> Note: you need to customise any settings shown in the examples to suit your database, Landing Zone, and credentials. Our example settings will not work for your database.

- Our example *Target* database is using our test Azure PostgreSQL Flexible Server, pg-bnz-sy0-npdis1-vwuj-dev-1


portal.azure.com/?feature.msaljs=true#@bnz.onmicrosoft.com/resource/subscriptions/993817dc-61e1-4957-94b3-496e0c395840/resour...
(3) AML and KYC... AML & KYC MANT... Imported From Ed... Cloud Computing... Solutions for All L... Top 5 Tips & Trick... linux AWS Azure MSQL
Microsoft Azure
Search resources, services, and docs (G+/)
Home > RG-BNZ-SY0-NPDIS1-DB2POC >
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Delete Reset password Restore Restart Upgrade Stop Refresh CLI / PS Feedback
Essentials

pg-bnz-sy0-npdis1-vwuj-dev-1

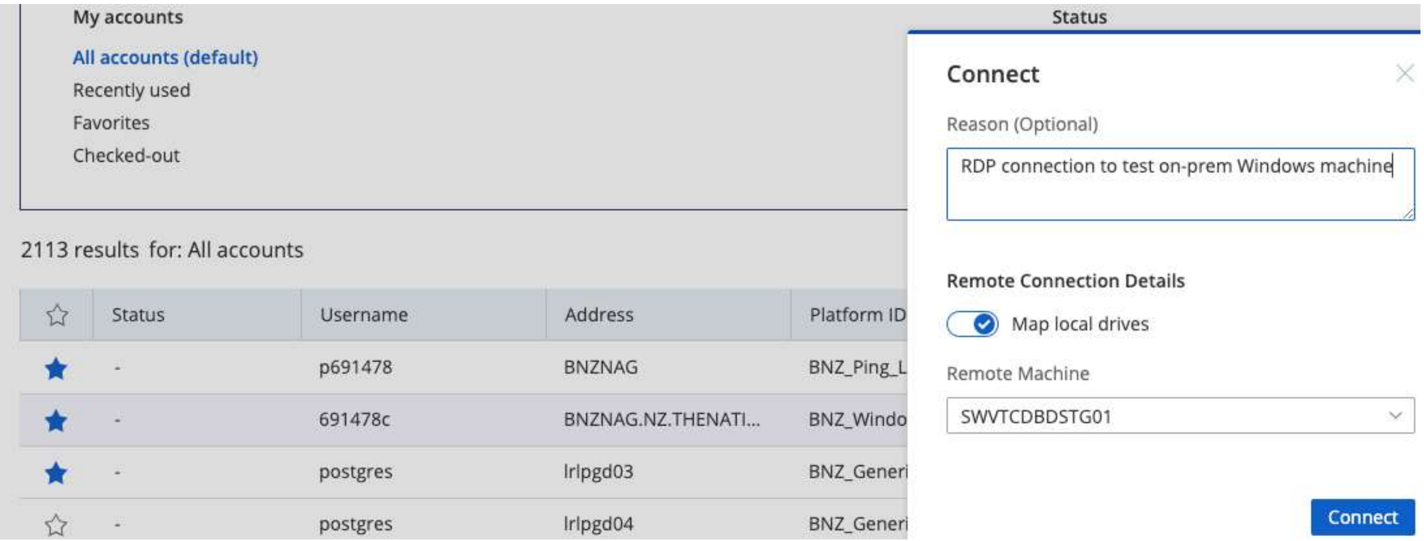
Azure Database for PostgreSQL flexible server

Subscription (move)	: BNZ-NPDIS1-CPS	Server name	: pg-bnz-sy0-npdis1-vwuj-dev-1.p
Subscription ID	: 993817dc-61e1-4957-94b3-496e0c395840	Server admin login name	: postgres
Resource group (move)	: RG-BNZ-SY0-NPDIS1-DB2POC	Configuration	: General Purpose, D2ds_v5, 2 vCo
Status	: Available	PostgreSQL version	: 15.3
Location	: Australia East	Availability zone	: 2

Step 2. Set up SCT on your on-premise Windows server

The example on-premise Windows Server (SWVTCDBSTG01) opens via **CyberArk**. When opening your Server in CyberArk:

- Use your Admin Account (your Staff ID with a "c" suffix), so you have software install access.
- After highlighting your Admin Account, right click to select an RDP connection.
- Enter or select your Server name to open it on Windows Remote Desktop.



2113 results for: All accounts

☆	Status	Username	Address	Platform ID
★	-	p691478	BNZNAG	BNZ_Ping_L
★	-	691478c	BNZNAG.NZ.THENATI...	BNZ_Windo
★	-	postgres	lrlpgd03	BNZ_Generi
☆	-	postgres	lrlpgd04	BNZ_Generi

Connect

Reason (Optional)

RDP connection to test on-prem Windows machine

Remote Connection Details

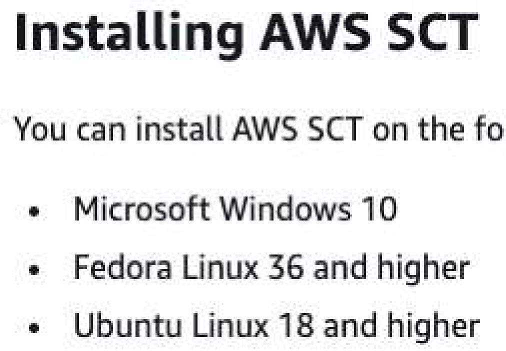
● Map local drives

Remote Machine

SWVTCDBDSTG01

Connect

- Download SCT from the AWS website onto your Windows server.



Installing AWS SCT

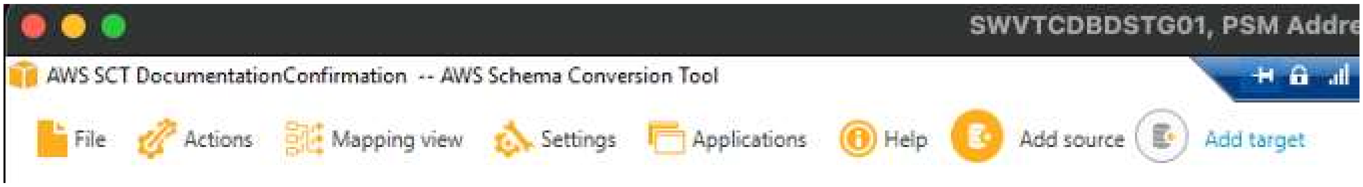
You can install AWS SCT on the following operating systems:

- Microsoft Windows 10
- Fedora Linux 36 and higher
- Ubuntu Linux 18 and higher

- Install the **database driver** for your particular database. In the example we have installed the DB2 Db Driver. If your source database is an Oracle database, you would install an Oracle db Driver.

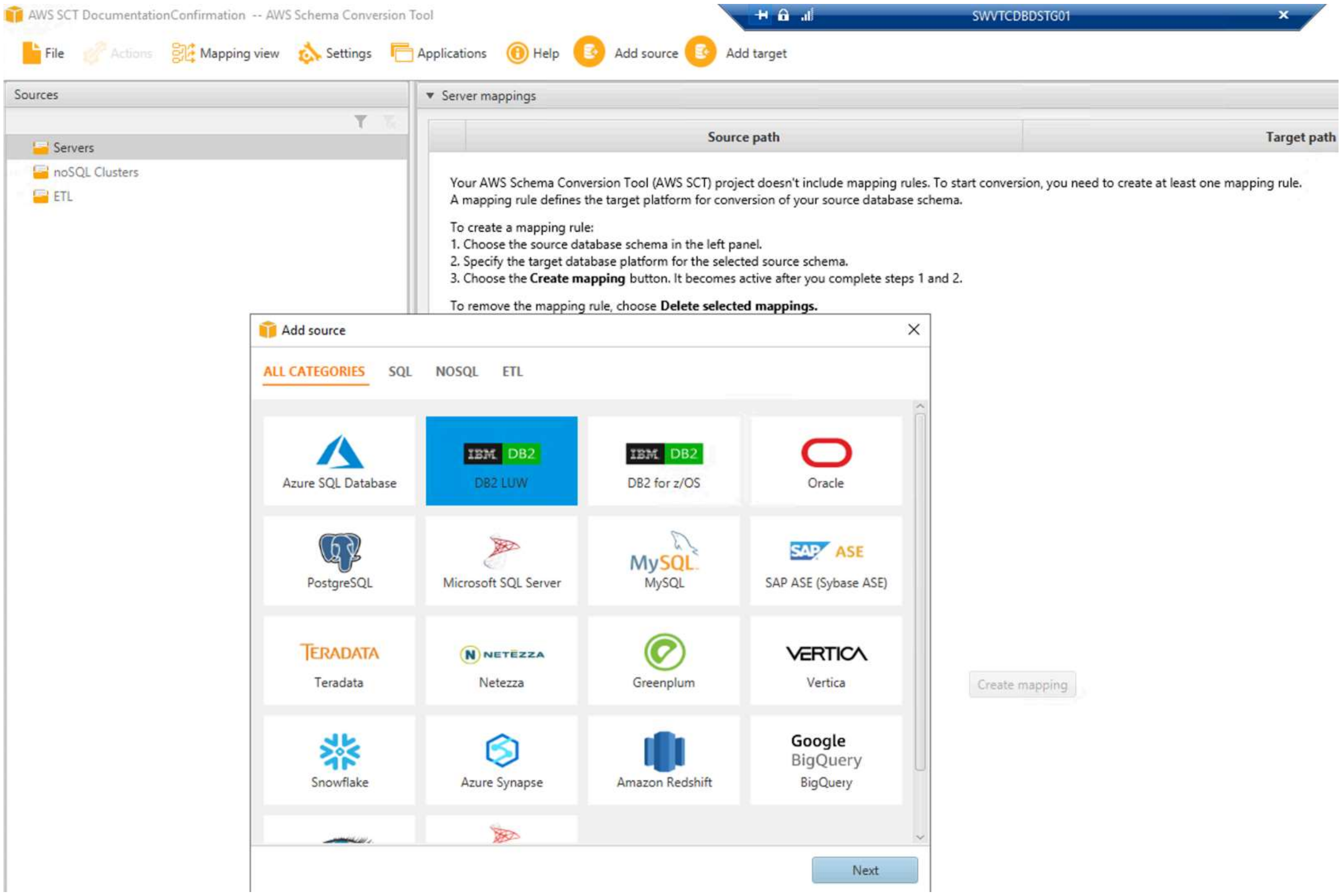
Step 3. Operate SCT

- On your on premise Windows Server, start SCT



3a. Create an endpoint to your *source* database on-premise

- Click on Add Source

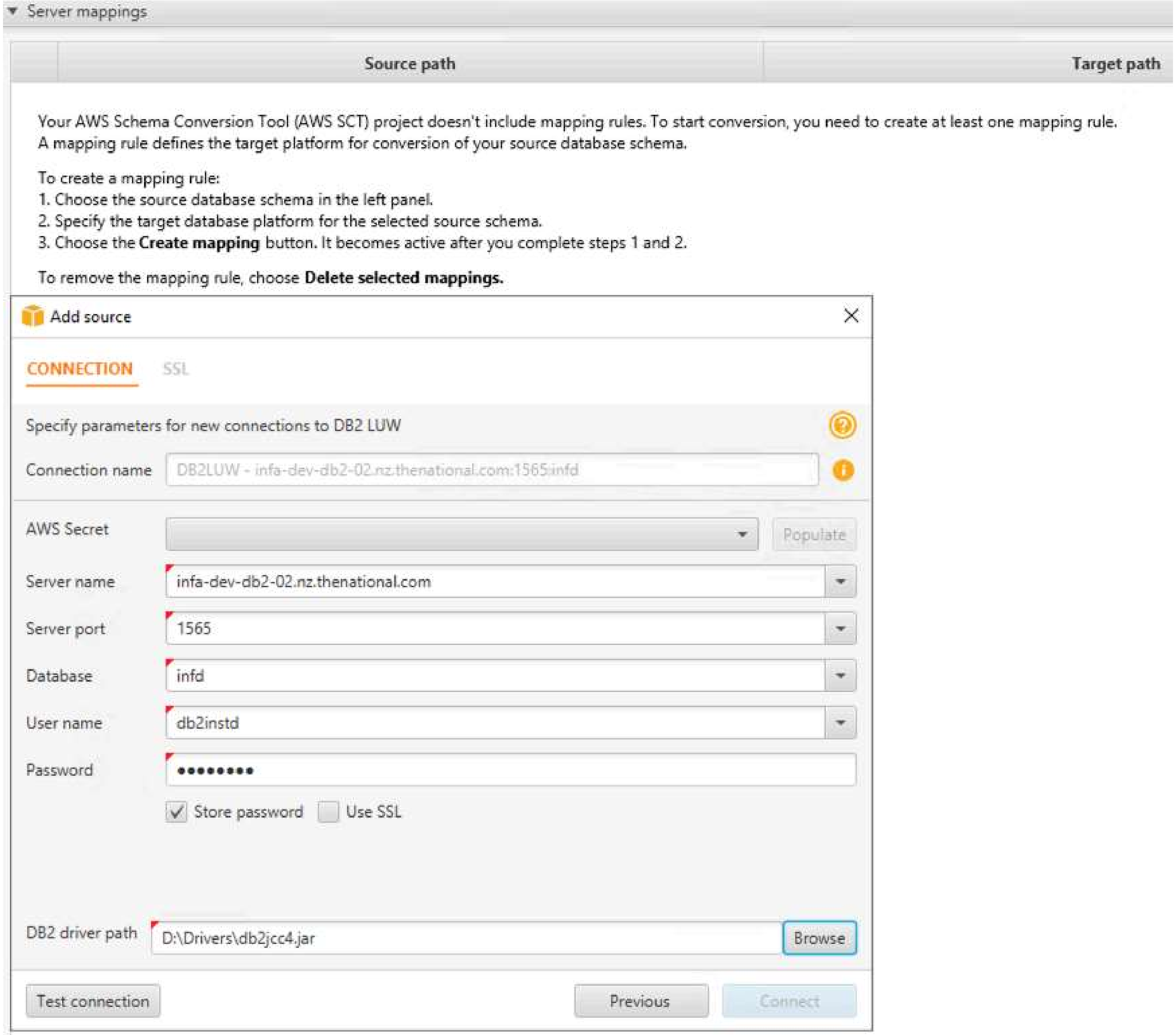


Your AWS Schema Conversion Tool (AWS SCT) project doesn't include mapping rules. To start conversion, you need to create at least one mapping rule. A mapping rule defines the target platform for conversion of your source database schema.

To create a mapping rule:
1. Choose the source database schema in the left panel.
2. Specify the target database platform for the selected source schema.
3. Choose the **Create mapping** button. It becomes active after you complete steps 1 and 2.

To remove the mapping rule, choose **Delete selected mappings.**

- Fill out the source database details



Your AWS Schema Conversion Tool (AWS SCT) project doesn't include mapping rules. To start conversion, you need to create at least one mapping rule. A mapping rule defines the target platform for conversion of your source database schema.

To create a mapping rule:
1. Choose the source database schema in the left panel.
2. Specify the target database platform for the selected source schema.
3. Choose the **Create mapping** button. It becomes active after you complete steps 1 and 2.

To remove the mapping rule, choose **Delete selected mappings.**

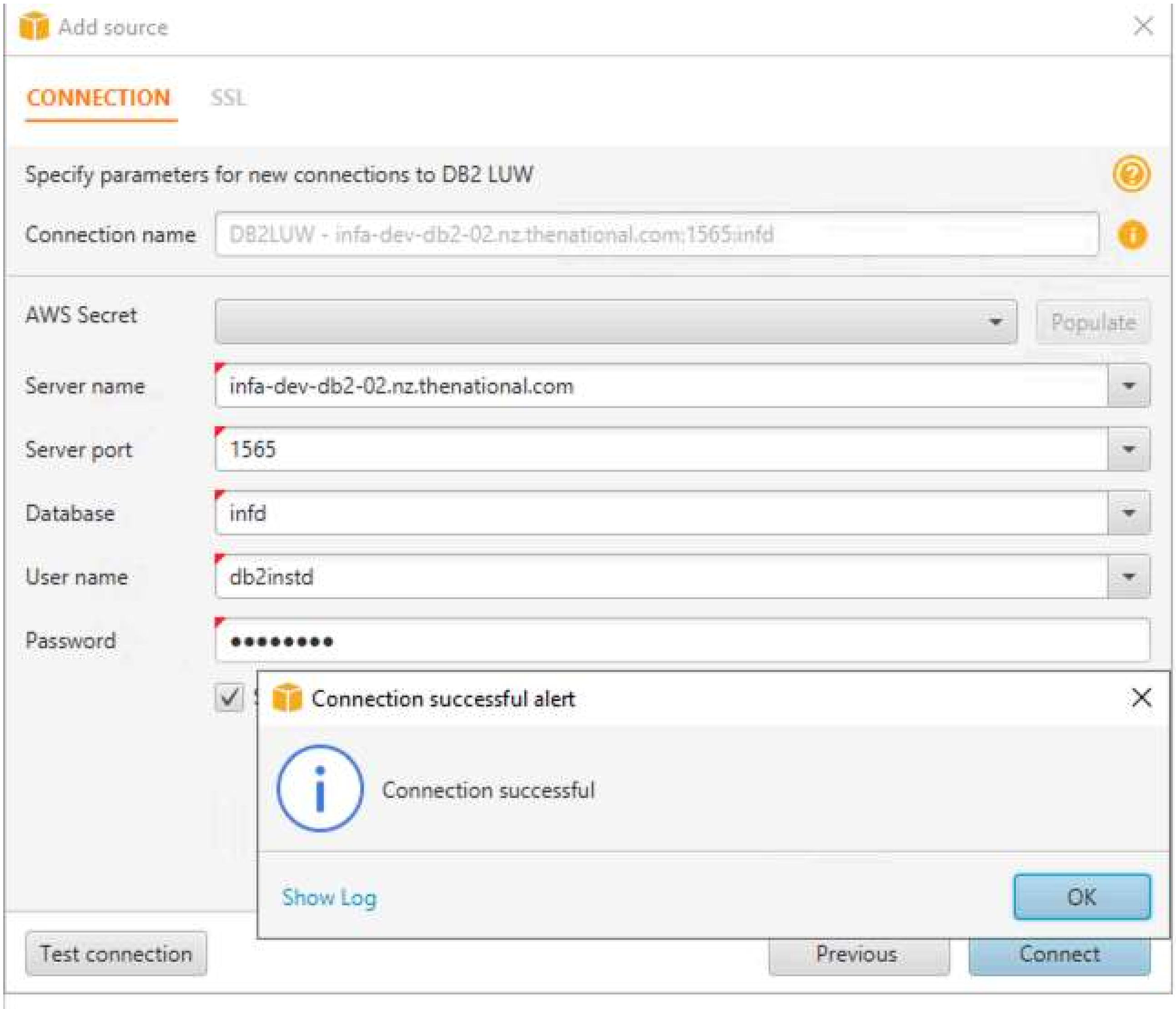
Specify parameters for new connections to DB2 LUW

Connection name	DB2LUW - infa-dev-db2-02.nz.thenational.com:1565:infd
AWS Secret	
Server name	infa-dev-db2-02.nz.thenational.com
Server port	1565
Database	infd
User name	db2instd
Password	●●●●●●●●●

☑ Store password □ Use SSL

DB2 driver path: D:\Drivers\db2jcc4.jar

- Test the connection via the Test Connection button:



Specify parameters for new connections to DB2 LUW

Connection name: DB2LUW - infa-dev-db2-02.nz.thenational.com:1565:infd

AWS Secret:

Server name: infa-dev-db2-02.nz.thenational.com

Server port: 1565

Database: infd

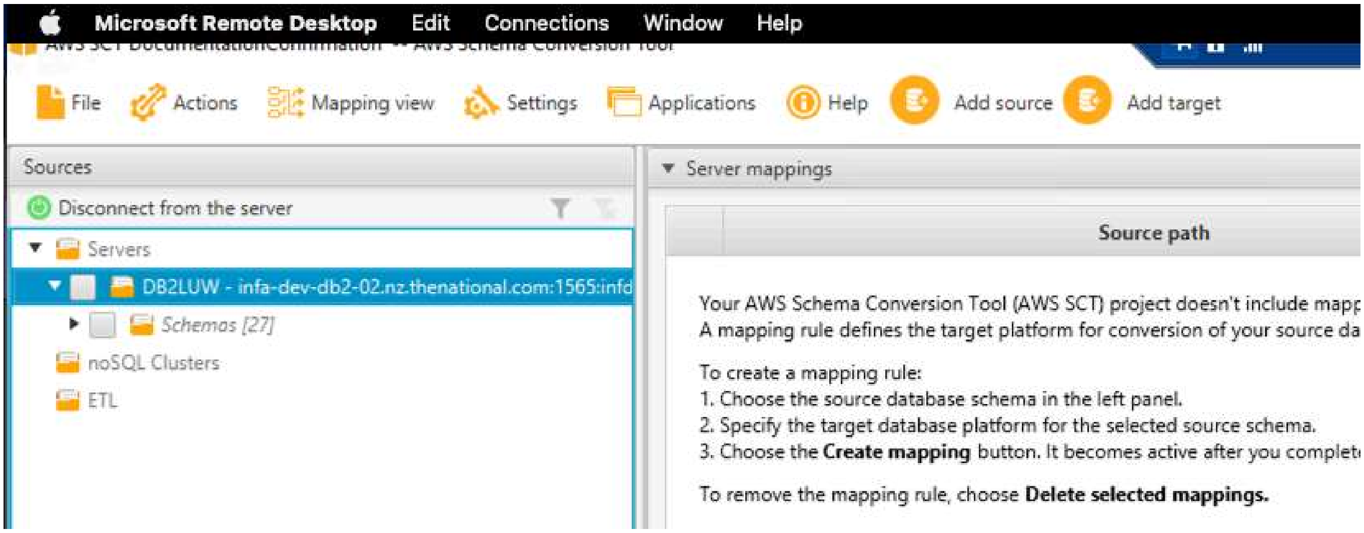
User name: db2instd

Password: ●●●●●●●●●

Connection successful alert

Connection successful

- Click on Connect, and your source database appears on the left of the screen



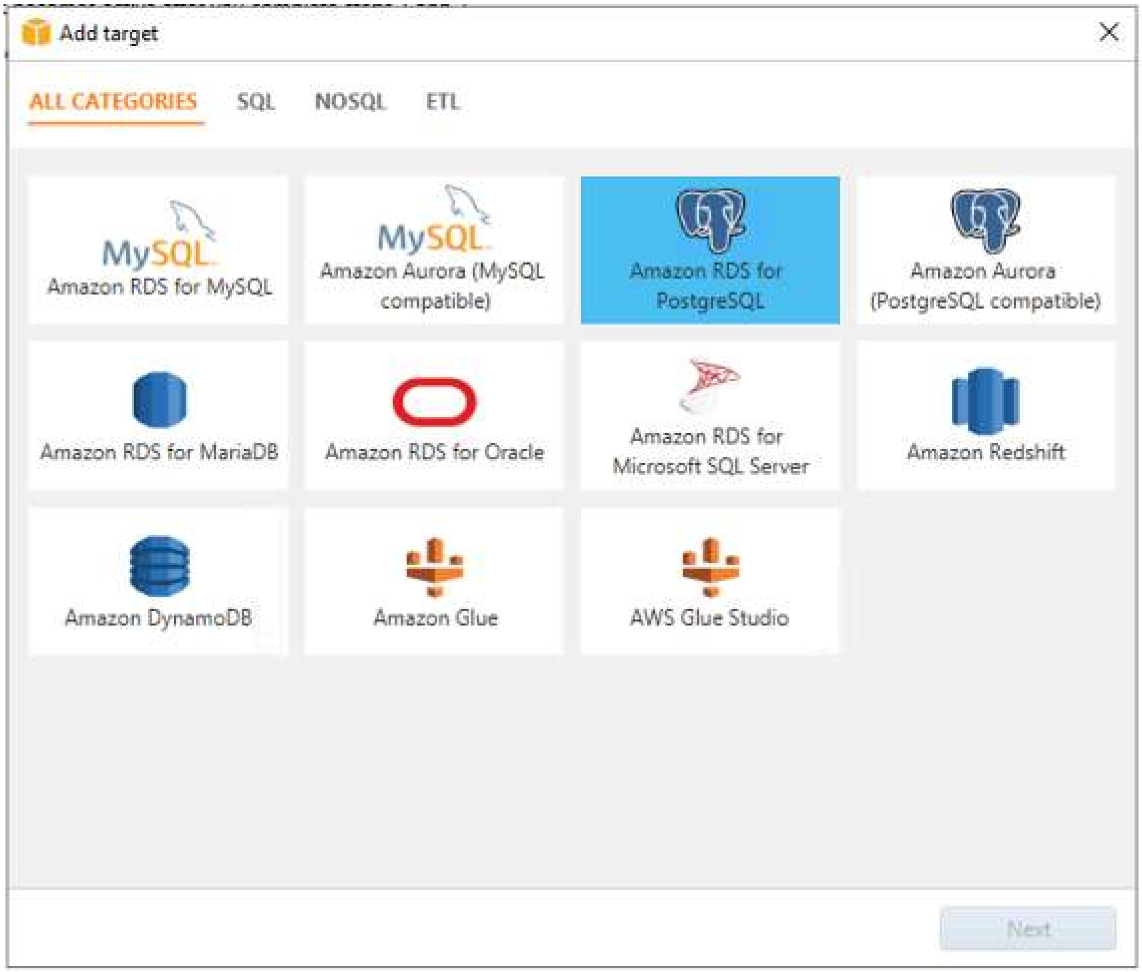
Your AWS Schema Conversion Tool (AWS SCT) project doesn't include mapp
A mapping rule defines the target platform for conversion of your source da

To create a mapping rule:
1. Choose the source database schema in the left panel.
2. Specify the target database platform for the selected source schema.
3. Choose the **Create mapping** button. It becomes active after you complet

To remove the mapping rule, choose **Delete selected mappings.**

3a. Create an endpoint to your target database on AWS

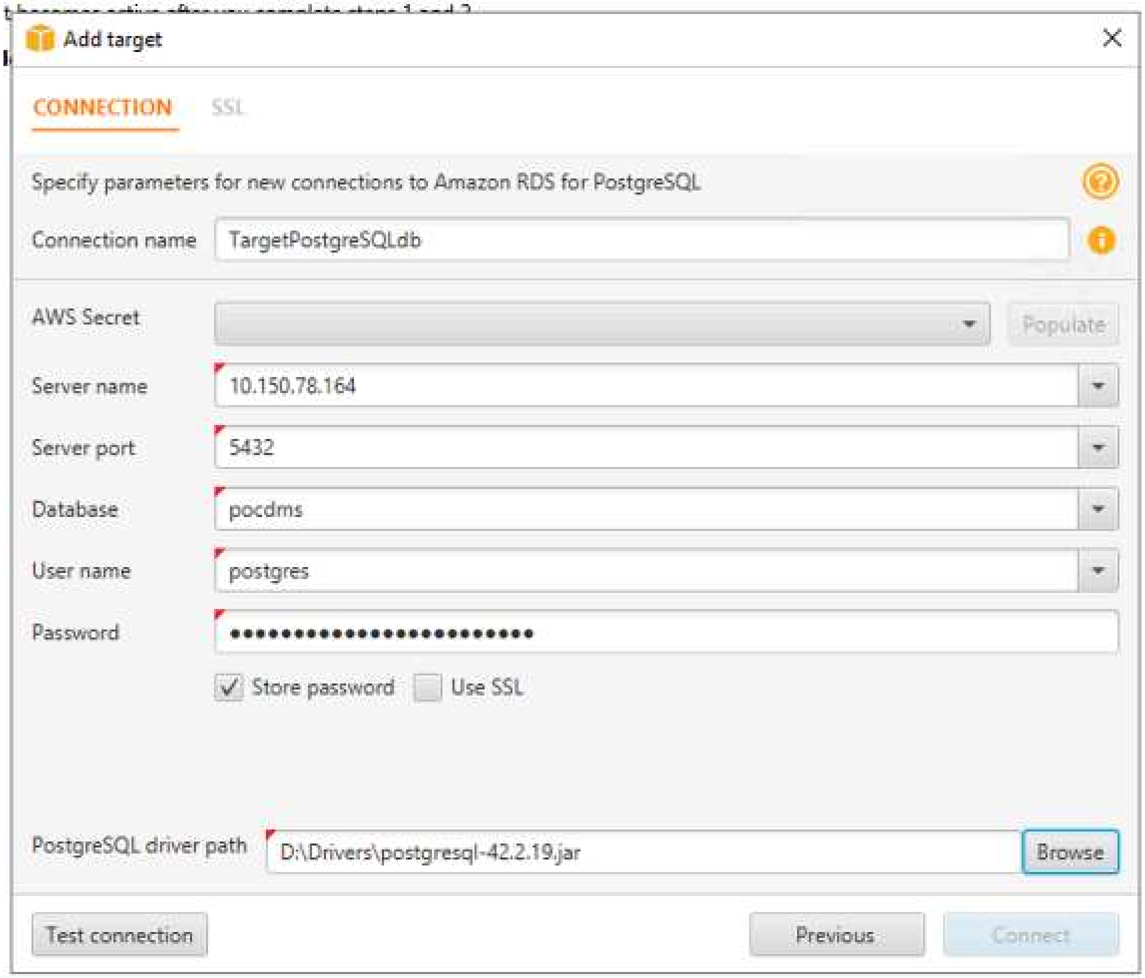
- Click on Add Target



CAUTION

AWS will only support your activities if either a target or source is in the AWS cloud. If you think you may require support and your target is Azure, you may want to do a 2-step migration (on-premise -> AWS and then AWS -> Azure).

- Fill out the Target Db details (the example shown is on Azure, but it could be on AWS)



Specify parameters for new connections to Amazon RDS for PostgreSQL

Connection name: TargetPostgreSQLdb

AWS Secret:

Server name: 10.150.78.164

Server port: 5432

Database: pocdms

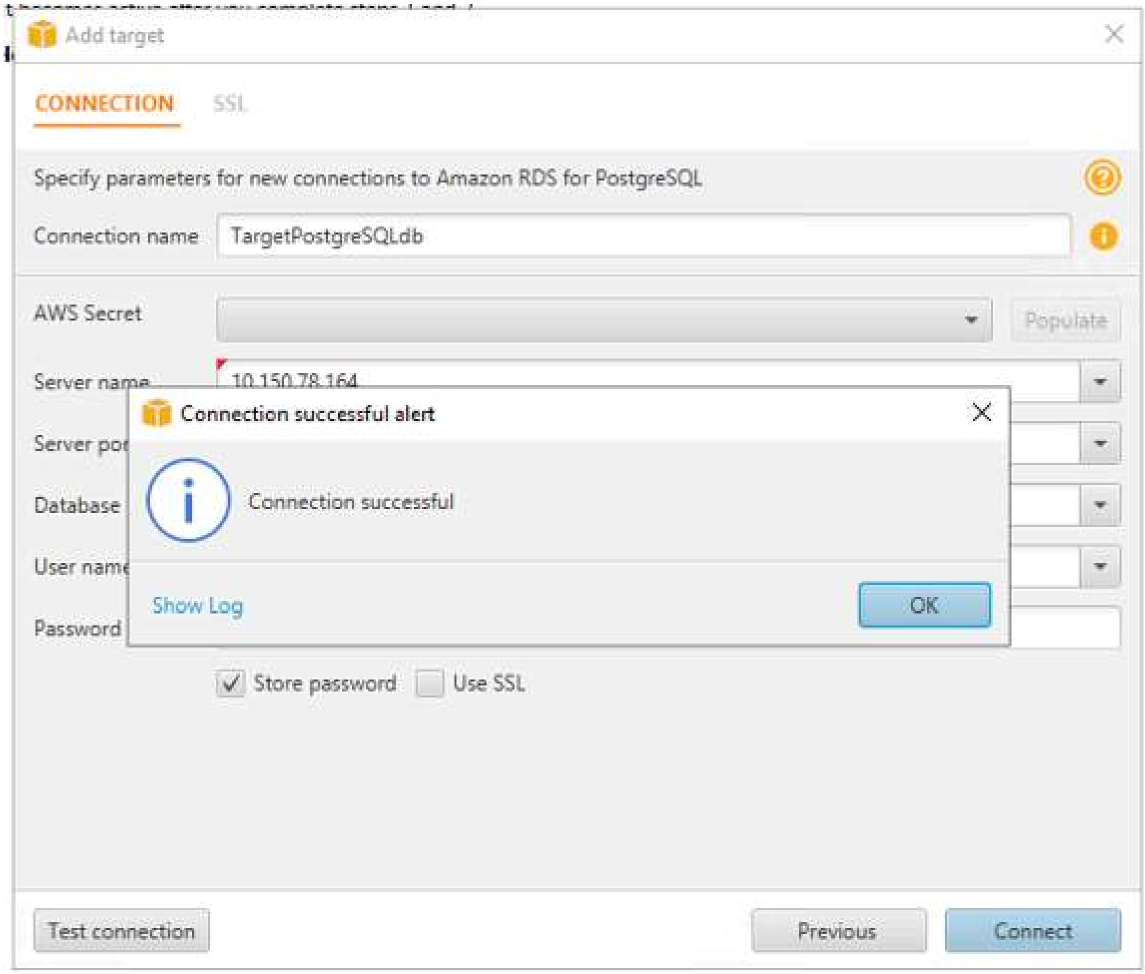
User name: postgres

Password: ••••••••••••••••••••••••

☑ Store password □ Use SSL

PostgreSQL driver path: D:\Drivers\postgresql-42.2.19.jar

- Test the connection via the Test Connection button:



Add target

CONNECTION SSL

Specify parameters for new connections to Amazon RDS for PostgreSQL

Connection name: TargetPostgreSQLdb

AWS Secret:

Server name: 10.150.78.164

Server port

Database

User name

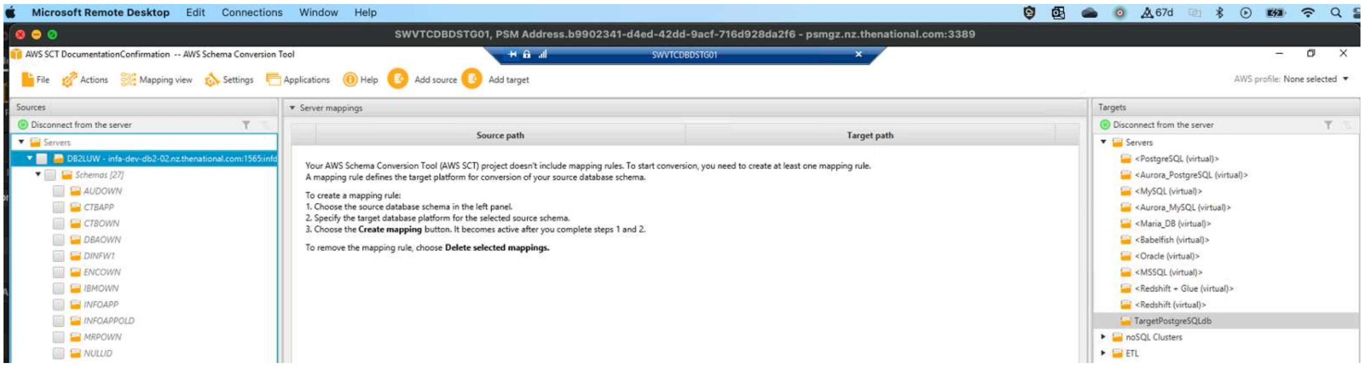
Password

☑ Store password □ Use SSL

Connection successful alert

Connection successful

- Click on Connect, and your Target DB appears on the right of the screen



Microsoft Remote Desktop Edit Connections Window Help

SWVTCDBDSTG01, PSM Address.b9902341-d4ed-42dd-9acf-716d928da2f6 - psmgz.nz.thenational.com:3389

AWS SCT DocumentationConfirmation -- AWS Schema Conversion Tool

AWS profile: None selected

Sources

Disconnect from the server

- Servers
  - DB2LUW - infa-dev-db2-02.nz.thenational.com:1565:infd
    - Schemas [27]
      - AUDOWN
      - CTBAPP
      - CTBOWN
      - DBAOWN
      - DINFW1
      - ENCOWN
      - IBMOWN
      - INFOAPP
      - INFOAPPOLD
      - MRPOWN
      - NULLID

Server mappings

Source path	Target path

Your AWS Schema Conversion Tool (AWS SCT) project doesn't include mapping rules. To start conversion, you need to create at least one mapping rule.
A mapping rule defines the target platform for conversion of your source database schema.

To create a mapping rule:
1. Choose the source database schema in the left panel.
2. Specify the target database platform for the selected source schema.
3. Choose the **Create mapping** button. It becomes active after you complete steps 1 and 2.

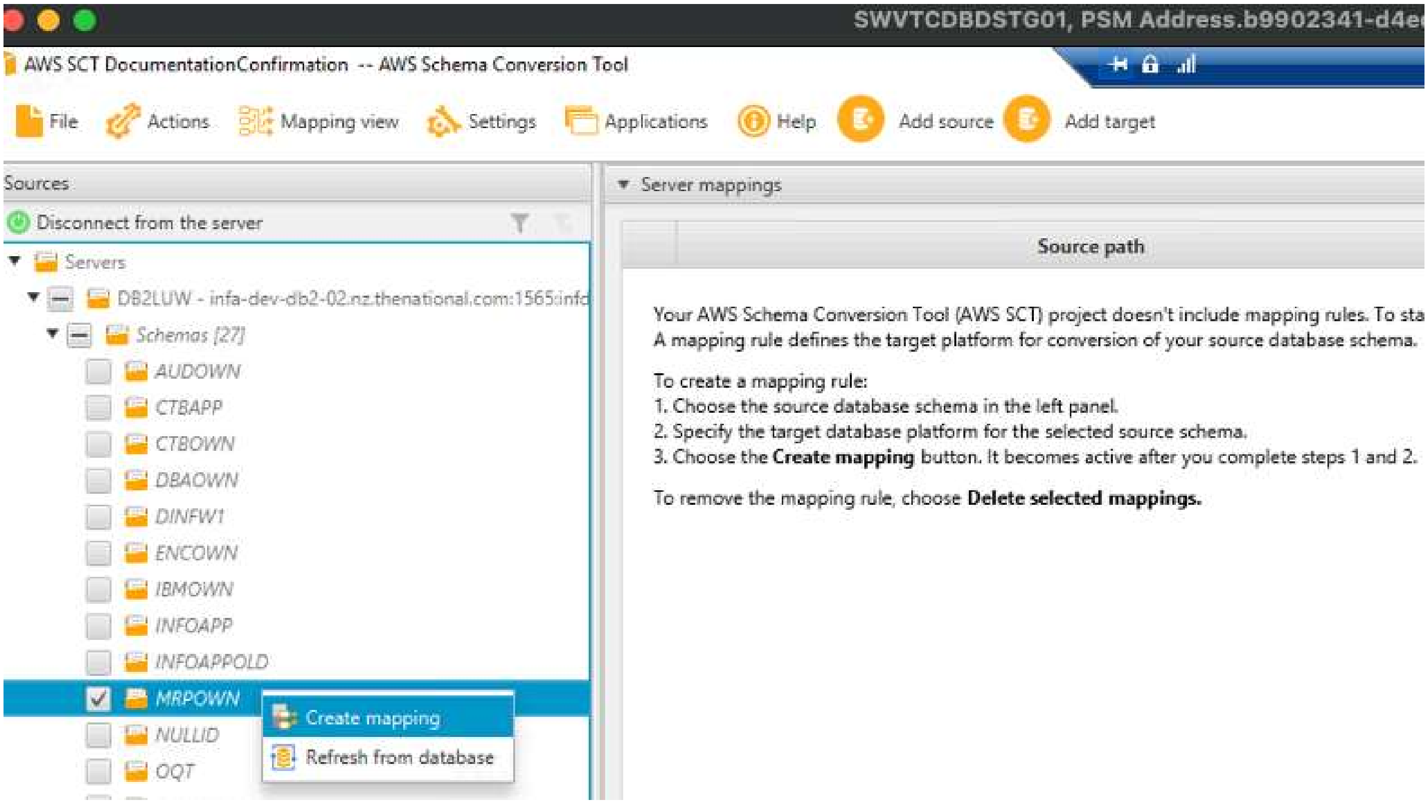
To remove the mapping rule, choose **Delete selected mappings.**

Targets

Disconnect from the server

- Servers
  - <PostgreSQL (virtual)>
  - <Aurora_PostgreSQL (virtual)>
  - <MySQL (virtual)>
  - <Aurora_MySQL (virtual)>
  - <Maria_DB (virtual)>
  - <Babelfish (virtual)>
  - <Oracle (virtual)>
  - <MSSQL (virtual)>
  - <Redshift + Glue (virtual)>
  - <Redshift (virtual)>
  - TargetPostgreSQLdb
- noSQL Clusters
- ETL

3b. Create a Mapping rule



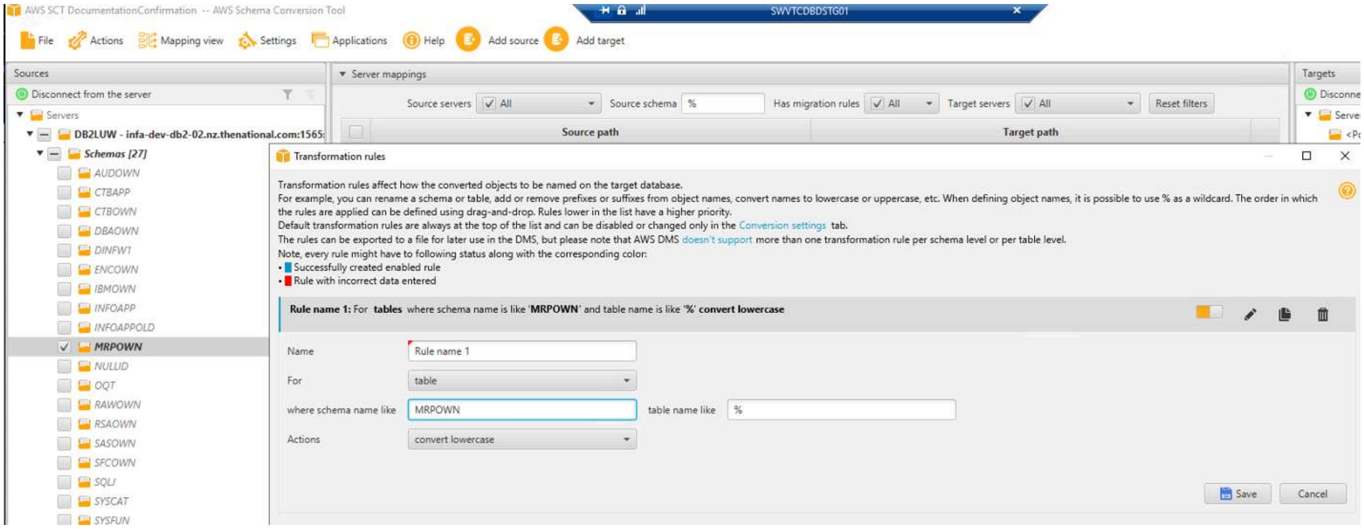
Your AWS Schema Conversion Tool (AWS SCT) project doesn't include mapping rules. To sta
A mapping rule defines the target platform for conversion of your source database schema.

To create a mapping rule:
1. Choose the source database schema in the left panel.
2. Specify the target database platform for the selected source schema.
3. Choose the **Create mapping** button. It becomes active after you complete steps 1 and 2.

To remove the mapping rule, choose **Delete selected mappings.**

Create a Transformation Rule.

- Here we are converting the uppercase DB2 Standard, to the lowercase PostgreSQL standard. Remember to click on *Save*.

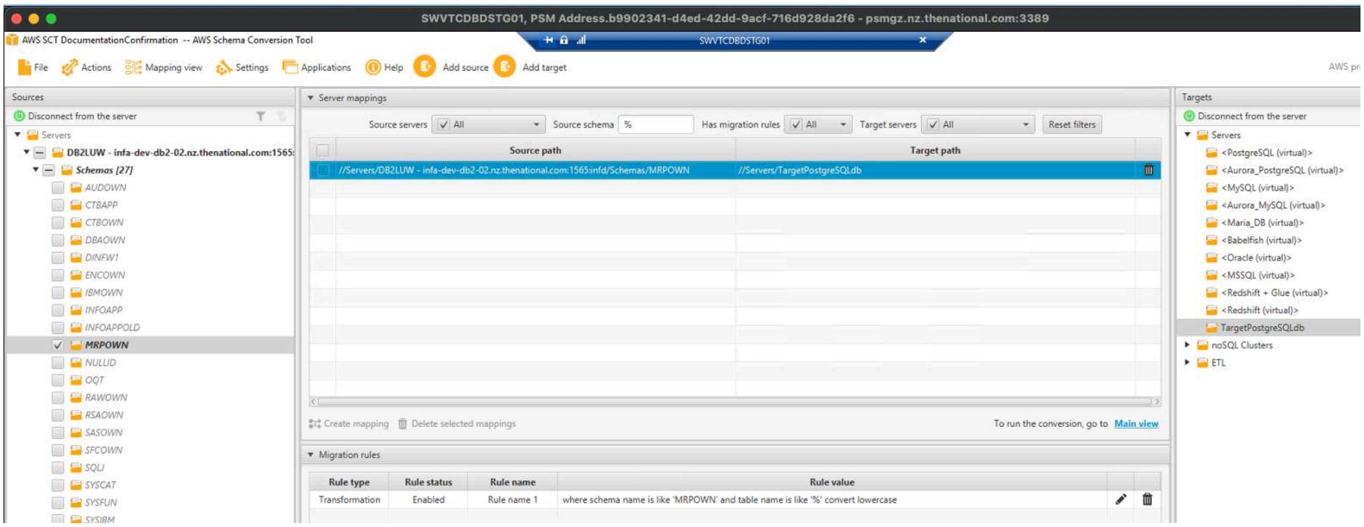


Transformation rules affect how the converted objects to be named on the target database.
For example, you can rename a schema or table, add or remove prefixes or suffixes from object names, convert names to lowercase or uppercase, etc. When defining object names, it is possible to use % as a wildcard. The order in which the rules are applied can be defined using drag-and-drop. Rules lower in the list have a higher priority.
Default transformation rules are always at the top of the list and can be disabled or changed only in the Conversion settings tab.
The rules can be exported to a file for later use in the DMS, but please note that AWS DMS doesn't support more than one transformation rule per schema level or per table level.
Note, every rule might have to following status along with the corresponding color:
- 🟦 Successfully created enabled rule
- 🟥 Rule with incorrect data entered

Rule name 1: For **tables** where schema name is like '**MRPOWN**' and table name is like '%' **convert lowercase**

Name	Rule name 1
For	table
where schema name like	MRPOWN
table name like	%
Actions	convert lowercase

- Click on Main View. From here we can run the report, and the conversion:



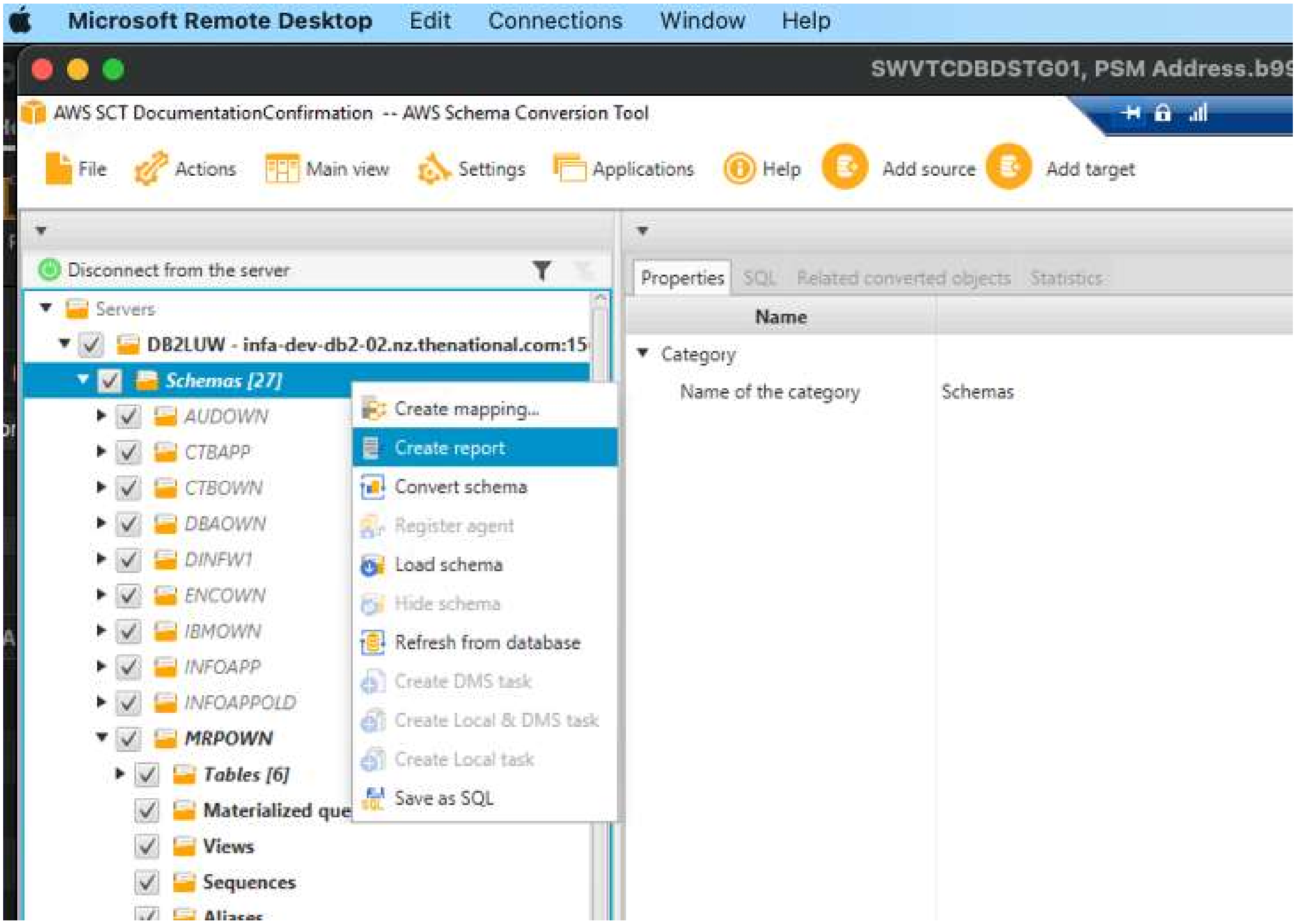
Source servers ☑ All | Source schema % | Has migration rules ☑ All | Target servers ☑ All | Reset filters

Source path	Target path
//Servers/DB2LUW - infa-dev-db2-02.nz.thenational.com:1565:infd/Schemas/MRPOWN	//Servers/TargetPostgreSQLdb

To run the conversion, go to Main view

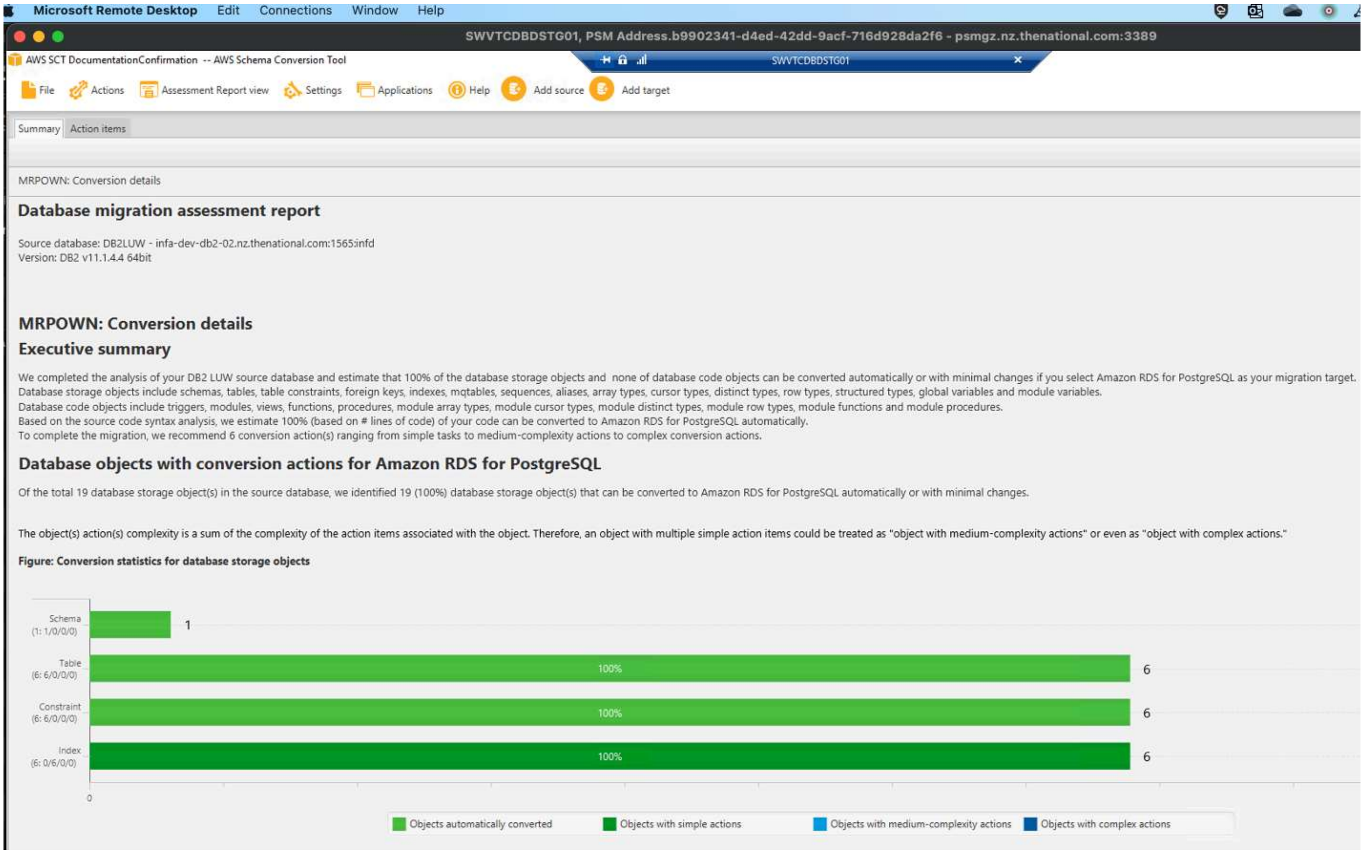
Rule type	Rule status	Rule name	Rule value
Transformation	Enabled	Rule name 1	where schema name is like 'MRPOWN' and table name is like '%' convert lowercase

3c. Select the entire database, or just the Schema, right click and *Create report*



Name	
Name of the category	Schemas

- Resultant report displays:


Microsoft Remote Desktop
Edit
Connections
Window
Help
SWVTCDBDSTG01, PSM Address.b9902341-d4ed-42dd-9acf-716d928da2f6 - psmgz.nz.thenational.com:3389
AWS SCT DocumentationConfirmation -- AWS Schema Conversion Tool
SWVTCDBDSTG01
File
Actions
Assessment Report view
Settings
Applications
Help
Add source
Add target
Summary
Action items
MRPOWN: Conversion details

Database migration assessment report

Source database: DB2LUW - infa-dev-db2-02.nz.thenational.com:1565:infd
Version: DB2 v11.1.4.4 64bit

MRPOWN: Conversion details

Executive summary

We completed the analysis of your DB2 LUW 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 RDS for PostgreSQL as your migration target.
Database storage objects include schemas, tables, table constraints, foreign keys, indexes, mqtables, sequences, aliases, array types, cursor types, distinct types, row types, structured types, global variables and module variables.
Database code objects include triggers, modules, views, functions, procedures, module array types, module cursor types, module distinct types, module row types, module functions and module procedures.
Based on the source code syntax analysis, we estimate 100% (based on # lines of code) of your code can be converted to Amazon RDS for PostgreSQL automatically.
To complete the migration, we recommend 6 conversion action(s) ranging from simple tasks to medium-complexity actions to complex conversion actions.

Database objects with conversion actions for Amazon RDS for PostgreSQL

Of the total 19 database storage object(s) in the source database, we identified 19 (100%) database storage object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

The object(s) action(s) complexity is a sum of the complexity of the action items associated with the object. Therefore, an object with multiple simple action items could be treated as "object with medium-complexity actions" or even as "object with complex actions."

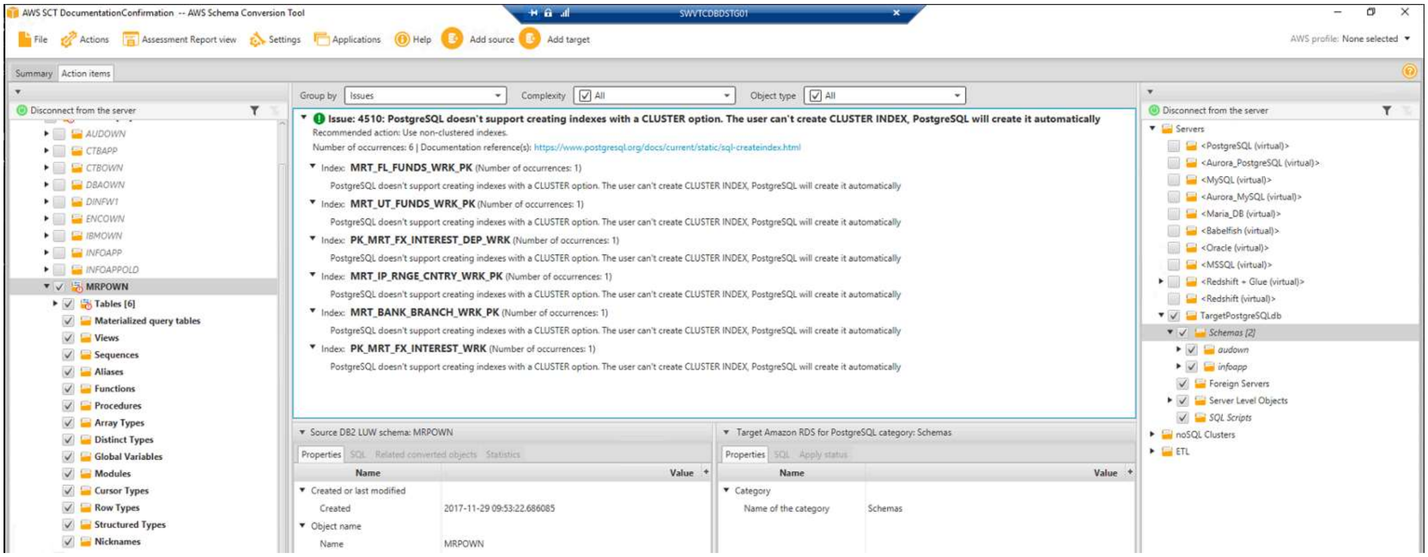
Figure: Conversion statistics for database storage objects

Schema
(1: 1/0/0/0)
1
Table
(6: 6/0/0/0)
100%
6
Constraint
(6: 6/0/0/0)
100%
6
Index
(6: 0/6/0/0)
100%
6
0
🟩 Objects automatically converted
🟩 Objects with simple actions
🟦 Objects with medium-complexity actions
🟦 Objects with complex actions

- Go to the Action Items tab to see the work required to successfully convert your Schema.

Resolve any conversion or transformation issues

Resolve any issues that SCT cannot convert automatically. You can also run the conversion with these errors unresolved. Then you can use the information as a checklist when you manually fix the issues on your *Target* database.



Issue: 4510: PostgreSQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, PostgreSQL will create it automatically
Recommended action: Use non-clustered indexes.
Number of occurrences: 6 | Documentation reference(s): https://www.postgresql.org/docs/current/static/sql-createindex.html

Index: **MRT_FL_FUNDS_WRK_PK** (Number of occurrences: 1)
PostgreSQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, PostgreSQL will create it automatically

Index: **MRT_UT_FUNDS_WRK_PK** (Number of occurrences: 1)
PostgreSQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, PostgreSQL will create it automatically

Index: **PK_MRT_FX_INTEREST_DEP_WRK** (Number of occurrences: 1)
PostgreSQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, PostgreSQL will create it automatically

Index: **MRT_IP_RNGE_CNTRY_WRK_PK** (Number of occurrences: 1)
PostgreSQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, PostgreSQL will create it automatically

Index: **MRT_BANK_BRANCH_WRK_PK** (Number of occurrences: 1)
PostgreSQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, PostgreSQL will create it automatically

Index: **PK_MRT_FX_INTEREST_WRK** (Number of occurrences: 1)
PostgreSQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, PostgreSQL will create it automatically

Source DB2 LUW schema: MRPOWN

Properties | SQL | Related converted objects | Statistics

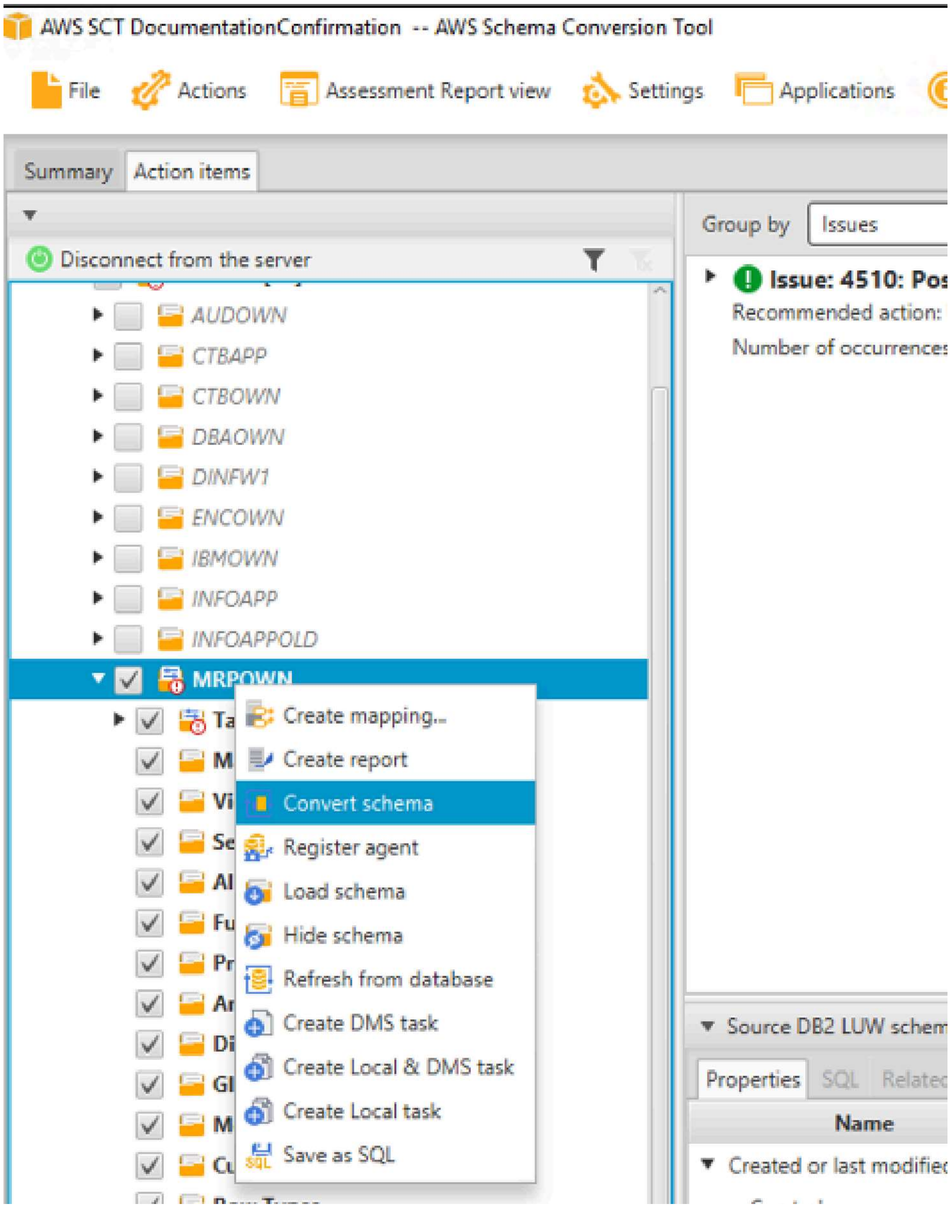
Name	Value
Created or last modified	
Created	2017-11-29 09:53:22.686085
Object name	
Name	MRPOWN

Target Amazon RDS for PostgreSQL category: Schemas

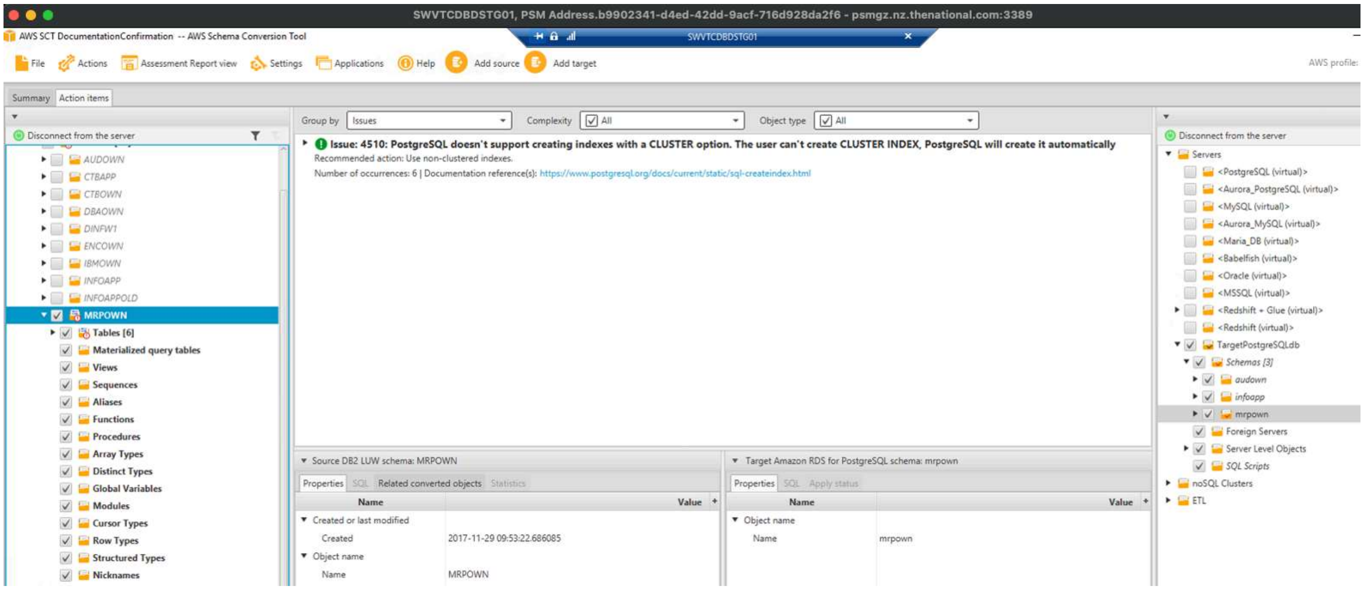
Properties | SQL | Apply status

Name	Value
Category	
Name of the category	Schemas

3d. Right click again on your Database or Schema name. This time select *Convert schema*



- Conversion complete. Note that the Target schema has been created in lowercase:



Group by: Issues | Complexity: ☑ All | Object type: ☑ All

Issue: 4510: PostgreSQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, PostgreSQL will create it automatically

Recommended action: Use non-clustered indexes.

Number of occurrences: 6 | Documentation reference(s): https://www.postgresql.org/docs/current/static/sql-createindex.html

Source DB2 LUW schema: MRPOWN

Properties | SQL | Related converted objects | Statistics

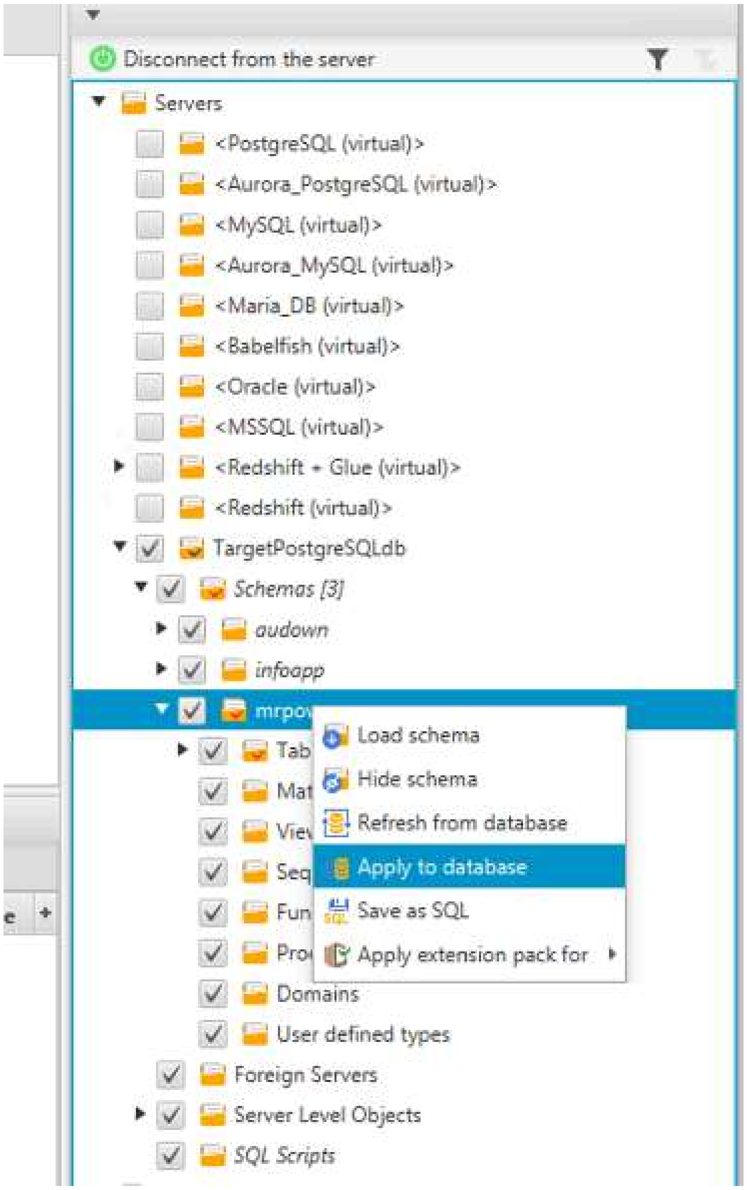
Name	Value
Created or last modified	
Created	2017-11-29 09:53:22.686085
Object name	
Name	MRPOWN

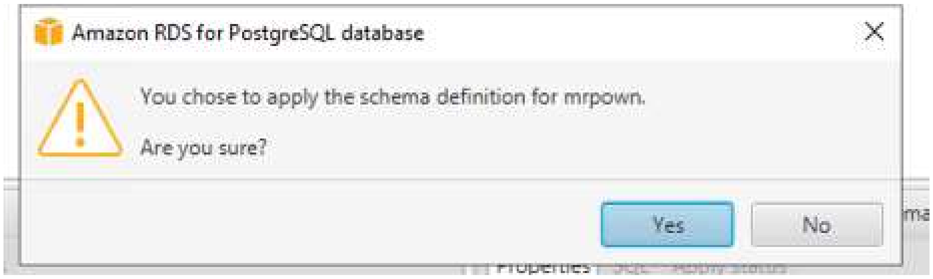
Target Amazon RDS for PostgreSQL schema: mrpown

Properties | SQL | Apply status

Name	Value
Object name	
Name	mrpown

- Now right click on the Target Schema name and click on *Apply to database*. This will save your changes in the Target Db:

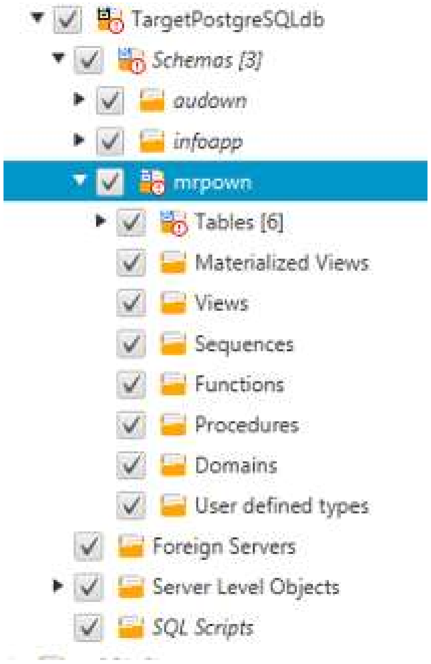



Amazon RDS for PostgreSQL database

You chose to apply the schema definition for mrpown.

Are you sure?

- The little red circle and exclamation mark shows the conversion has been applied:



- ☑ TargetPostgreSQLdb
  - ☑ *Schemas [3]*
    - ☑ *audown*
    - ☑ *infoapp*
    - ☑ mrpown
      - ☑ Tables [6]
      - ☑ Materialized Views
      - ☑ Views
      - ☑ Sequences
      - ☑ Functions
      - ☑ Procedures
      - ☑ Domains
      - ☑ User defined types
  - ☑ Foreign Servers
  - ☑ Server Level Objects
  - ☑ *SQL Scripts*

Remember, SCT has only converted the Database components (Schema, Tables, Views, etc.) to your Cloud *Target* database. SCT doesn't transfer your data. **Part 2 of our example** shows you how to use the AWS Database Migration Service (DMS) to do this.

Edit this page