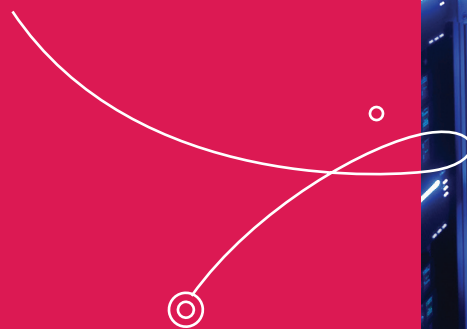




Mindtree

A Larsen & Toubro Group Company

Migration: Databases to Amazon RDS PostgreSQL from Oracle



A decorative graphic on the left side of the page consists of three vertical bars of varying heights and colors (red, orange, and yellow) and a yellow arrow pointing from the top of the bars towards the title.

Table of contents

- Introduction
- High-level architecture:
- Tools for migration
- Migration process:
 - Schema creation in the target database
 - Drop foreign key
 - Data Migration using AWS database migration service
 - Cutover
- Benefits

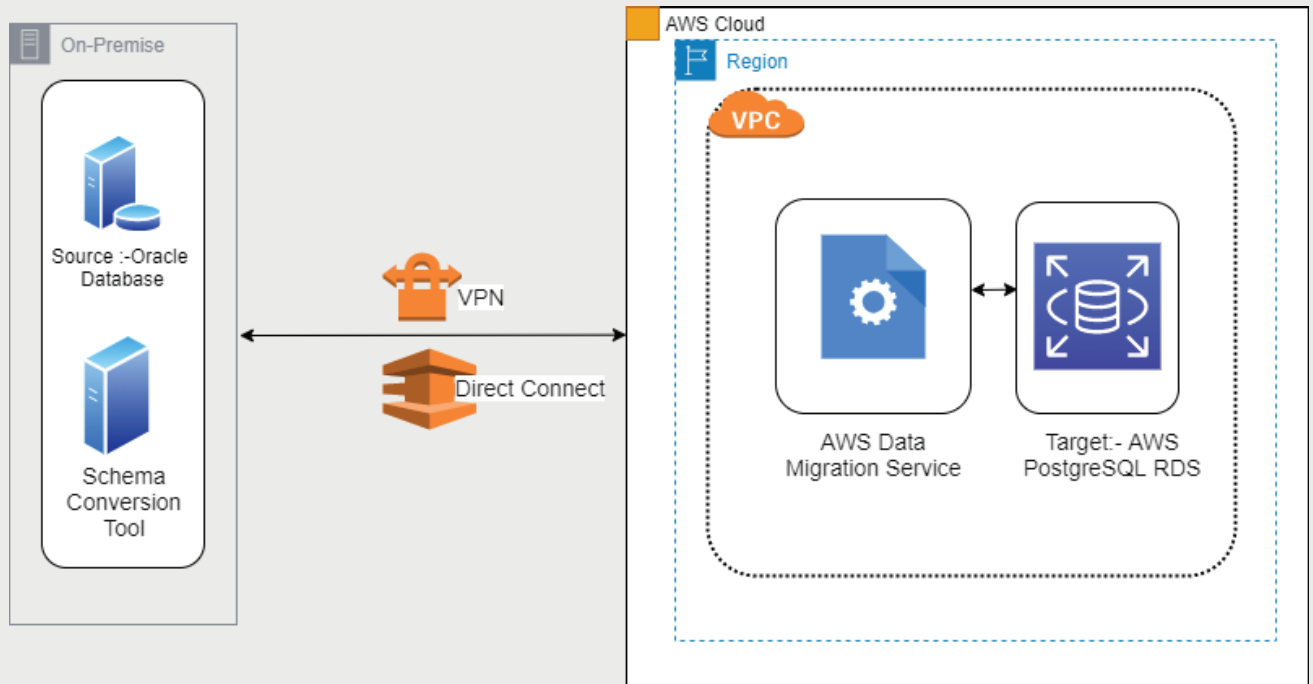


Introduction

With more and more IT infrastructures switching to the cloud and looking for massively scalable cloud adoption methods, migration from an Oracle database to AWS RDS for PostgreSQL is gaining popularity. Migration in AWS can be achieved in multifold ways — using different tools and technologies. There are various reasons to migrate a database from Oracle to AWS PostgreSQL. Some of the most significant reasons are high Oracle licensing cost, AWS PostgreSQL flexibility, and PostgreSQL customizability and scalability. The overall migration process from Oracle to AWS RDS for PostgreSQL is divided into automated and manual tasks. The automated tasks involve AWS Schema Conversion Tool and AWS Database Migration Service. Manual tasks are required for the objects that cannot be migrated automatically.

High-level architecture

One can understand the overview of the migration process by looking at the below diagram.



The high-level solution architect is flexible enough to cater to various Relational Database Management System's (RDBMS) heterogeneous needs using various tools like AWS, SCT, and AWS DMS.

Tools for migration

- **AWS DMS** - AWS Database Migration Service (DMS) enables quick and secure database migration from customer datacenter to AWS. DMS helps in minimizing the application downtime during the migration process. It supports database migration from various source DB platforms to target databases.
- **AWS SCT** - There are various data migration object challenges during the migration process. However, AWS Schema Conversion Tool (SCT) helps to migrate at ease. It takes care of heterogeneous databases source schemas and converts them automatically to target databases compatible format.
- **Amazon RDS** - AWS RDS service is a quick, scalable, and secure database service. It is easy to set up and operate — with high availability and scalability.
To start the migration process, the first step is to convert the source database (Oracle schema to PostgreSQL database schema). Schema migration tools that will help automate schema conversion are mentioned below:
- **Ora2pg** - Ora2pg tool helps ease the process of migration. After connecting to an Oracle database, it extracts schemas, objects, and tables and generates the required SQL scripts. This SQL script can then be loaded into the target AWS PostgreSQL RDS.
- **Ora_migrator** - Oracle fdw (an extension) used to migrate Oracle DB to PostgreSQL.
- **Orafce** - This extension contains some useful functions that can be used to port Oracle applications to PostgreSQL.



For demonstration, the AWS Schema Conversion Tool will help analyze the source database servers on Oracle.

Migration process

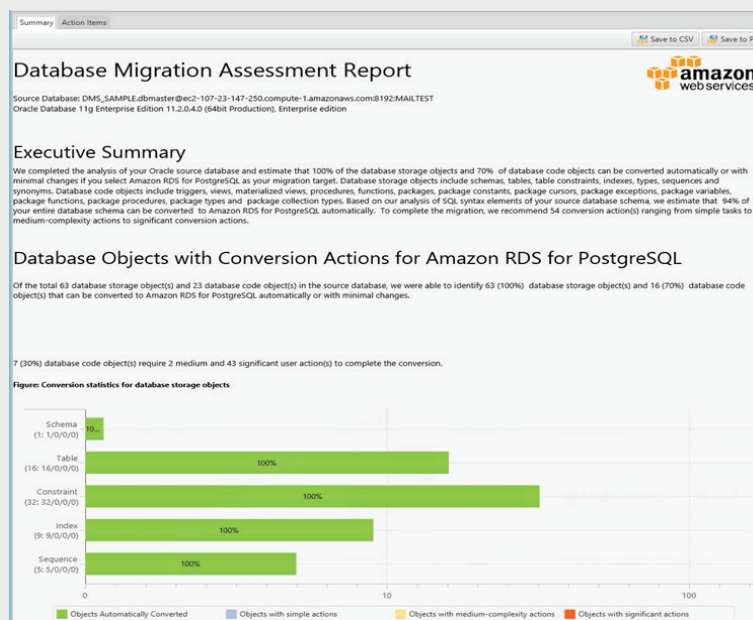
PostgreSQL, an open-source RDBMS, has become the most popular choice among several enterprise developers. Using AWS, one can deploy scalable PostgreSQL deployments in just a few minutes with a cost-efficient and scalable solution. The below process applies for migrating Oracle DB to AWS PostgreSQL RDS.

Schema creation in the target database

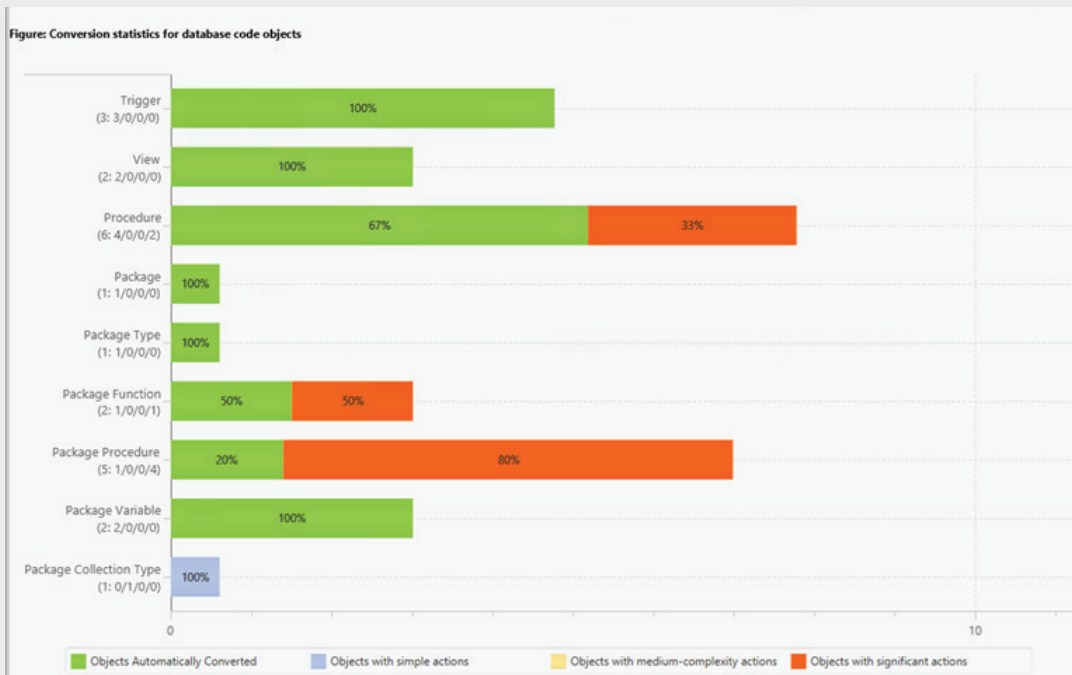
Amazon Schema Conversion Tool helps convert existing database schema from one database to another database engine. It provides the interface to convert the source Oracle schema into a format compatible with the target AWS PostgreSQL RDS. It also provides guidelines to follow while creating the corresponding schema in the target RDS database.

Upon successful installation, continue to create a new project in AWS Schema Conversion Tool by following the below steps:

- Select source as Oracle DB and target PostgreSQL
- Choose which schema needs migration by selecting the name
- Right-click the schema name and select "Convert Schema"
- Then select "Assessment Report View"
- The report shows a summary of the efforts required to convert the Oracle database to PostgreSQL.



SCT also tells why to fix these objects and how to do it.



Once the modifications are done, we can regenerate the schema report and see if there are any other issues. Once schemas look good we can apply them to the target AWS PostgreSQL RDS.

The next step is to focus on the target database, column data types, object names, etc. Validate the respective schema. One challenge here is capitalization. Oracle uses uppercase for object names. Whereas, PostgreSQL uses lowercase.

Summary Action Items

- Issue: 5028: Unable to convert unsupported datatypes**
Recommended action: The stub-code created. Please perform a manual conversion.
Number of occurrences: 3
- Issue: 5029: Unable to convert simple usage of unsupported datatype objects**
Recommended action: The stub-code created. Please perform a manual conversion.
Number of occurrences: 5
- Issue: 5030: Unable to convert complex usage of unsupported datatype objects**
Recommended action: The stub-code created. Please perform a manual conversion.
Number of occurrences: 13
- Issue: 5103: Unable to convert hints**
Recommended action: Use PostgreSQL methods for performance tuning.
Number of occurrences: 2 | Documentation reference: <http://www.postgresql.org/docs/9.4/static/geqo.html>
- Issue: 5118: PostgreSQL doesn't support associative arrays**
Recommended action: Use array or hstore types.

Oracle schema: DMS_SAMPLE	Amazon RDS for PostgreSQL schema: dms_sample
Properties	SQL
1 DMS_SAMPLE IDENTIFIED BY DMS_SAMPLE;	1 :ATE SCHEMA IF NOT EXISTS dms_sample;



Drop foreign key

The target should be free from any foreign key constraints and triggers. Identifying and disabling them is the next step. Disabling can be done by generating the table DDL. This is a manual task. Once the DDL table is generated, one can disable the foreign key components. Alternatively, you can delete them for a temporary period.

AWS DMS migration

Once the Schema migration is done from the source Oracle DB to target PostgreSQL RDS, we can start replicating the data. Apart from replicating the data from the source to the target, the AWS database migration service can also keep the data up-to-date with the CDC mode.

Below are some prerequisites:-

- Oracle login with necessary permissions on the source database.
Enabling supplemental logging to capture changes from the source Oracle database
- Archive logs must be enabled – Make sure that the source database is in ARCHIVELOG mode.

A replication instance must be created for DMS. AWS uses this replication instance to connect to the source Oracle DB server and target PostgreSQL.

Create replication instance

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

Name*	<input type="text" value="e.g. production-replication-server"/>	
Description*	<input type="text" value="e.g. migrates prod data"/>	
Instance class*	<input type="text" value="dms.t2.medium"/>	
VPC*	<input type="text" value="- Select One -"/>	
Multi-AZ	<input type="text" value="No"/>	
Publicly accessible	<input checked="" type="checkbox"/>	

▶ Advanced

Create two endpoints for the source and target database, respectively. This endpoint identifies the datastore type and location details about the source and target DB. It uses this information for creating the connection between the DBs and replication instances. The following endpoints need to be configured to proceed further:

Source endpoint – Oracle:

Go to DMS console and click on “Endpoints”

- Select endpoint type as – “Source Oracle”
- Endpoint identifier – Name of the endpoint identifier
- Source engine – Oracle
- Server name – On-premise Oracle Database Server (IP)
- Port – Enter the port number
- Enter the username, password, and database name

Click on “Run Test” to check if the DMS is communicating to the source Oracle DB server.

Target Endpoint – PostgreSQL

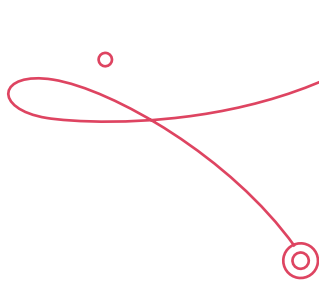
- Select endpoint type as – “Target_PostgreSQL”
- Endpoint identifier –Endpoint identifier name
- Source engine – Checkbox RDS and select the RDS instance
- Server name – Target PostgreSQL
- Port – Enter the port number
- Enter the username, password, and database name

Click on “Run Test” to check if DMS is communicating to the target RDS instance.

Once endpoint creation is completed, we need to create the tasks as per the below instructions:

Follow the below-mentioned points as per the options available:

- Choose migration type: – (Option 2 of below diagram) “Migrate the existing data and replicate ongoing changes”.
- Choose “Do Nothing” for target table preparation mode
- Include LOB columns must be selected in replication:- Limited LOB mode must be used (with default settings) for tables. This is applicable only for LOB columns that are 32 KB or less. Retain customized settings for Limited LOB mode. This is applicable only if the LOB columns exceed 32 KB.
- Stop task after the full load completes – Stop after applying cached changes.
- Enable logging.



Create task

A task can contain one or more table mappings which define what data is moved from the source to the target. If a table does not exist on the target, it can be created automatically.

Task name* ⓘ

Replication instance*

Source endpoint*

Target endpoint*

Migration type* ⓘ

Start task on create

- Migrate existing data
- Migrate existing data and replicate ongoing ch...
- Replicate data changes only

Task Settings

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

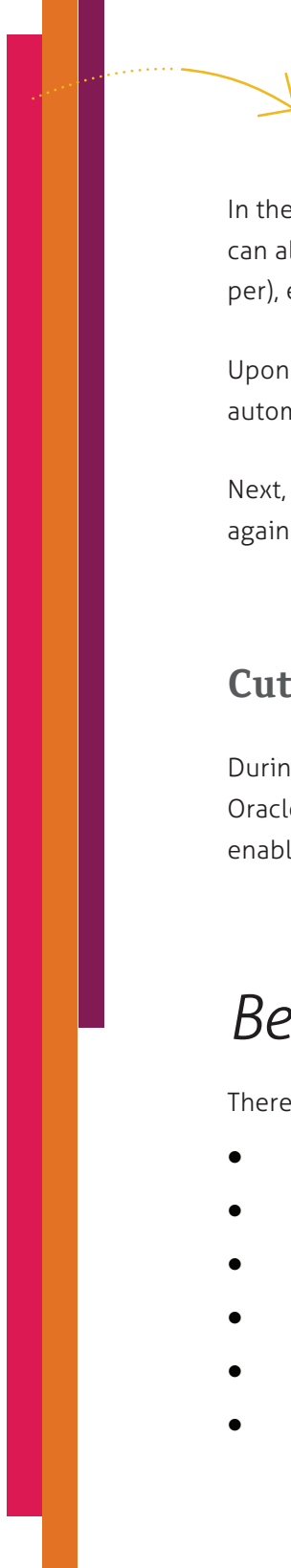
Stop task after full load completes* Don't stop ⓘ Stop Before Applying Cached Changes Stop After Applying Cached Changes

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

Max LOB size (kb)* ⓘ

Enable logging

[Advanced Settings](#)



In the next step, we can select and configure schemas, tables, and columns to include and ignore. We can also define a method like converting tables or schema names, converting capitalization (lower/up-per), etc.

Upon successful creation of the task, no manual interference is required, as the task will commence automatically. DMS console will help monitor the progress of the specific task.

Next, stop the task, add all secondary objects, enable the foreign key constraint, and start the task again to capture the on-going changes.

Cutover

During the “Cutover” phase when the application connections have stopped accessing the source Oracle DB and DMS has replicated the last data changes, we can stop the DMS task in the console and enable the triggers on the target DB.

Benefits

There are many benefits from both business and technical perspectives. Some of them are:

- PostgreSQL is an open-source database and has no licensing cost (community edition).
- There is no vendor lock-in risk in PostgreSQL DB.
- Fully Managed – Monitoring, security, alerting, backup, and disaster recovery.
- Easy availability from public cloud providers.
- Horizontal scaling and “Read Replicas” for performance.
- Scalable to meet the ever-growing demand.

Contact details:

Clouddatappractice@mindtree.com

This document is the exclusive property of Mindtree Limited (Mindtree). The recipient agrees that they will not copy, transmit, use or disclose the confidential and proprietary information in this document by any means without the expressed and written consent of Mindtree. By accepting a copy, the recipient agrees to adhere to these conditions to the confidentiality of Mindtree's practices and procedures; and to use these documents solely for responding to Mindtree's operations methodology.

About Mindtree

Mindtree [NSE: MINDTREE] is a global technology consulting and services company, helping enterprises marry scale with agility to achieve competitive advantage. "Born digital," in 1999 and now a Larsen & Toubro Group Company, Mindtree applies its deep domain knowledge to 280+ enterprise client engagements to break down silos, make sense of digital complexity and bring new initiatives to market faster. We enable IT to move at the speed of business, leveraging emerging technologies and the efficiencies of Continuous Delivery to spur business innovation. Operating in more than 15 countries across the world, we're consistently regarded as one of the best places to work, embodied every day by our winning culture made up of over 21,800 entrepreneurial, collaborative and dedicated "Mindtree Minds".