VERSENT

Insights

Database freedom: moving from Oracle to Aurora Serverless

JOSH LOPEZ - PARTNER, CLOUD ADOPTION
JOEL SANTIAGO - SENIOR ENGINEER

Is your enterprise looking to modernise its legacy Oracle databases? If so, you're probably investigating how to utilise cloud-native options and move toward new, more strategic data solutions.

This article is a guide to the virtues of database modernisation from Oracle to Aurora PostgreSQL Serverless. We'll walk through the process of migrating a terabyte-scale database from on-premises to AWS.

Amazon Web Services (AWS) has the capabilities to tackle every aspect of database modernisation: Athena to reduce cost on storage and compute, Aurora for utilising open source databases, and Redshift for performance, user-defined functions and indexes.

For the sake of this discussion, we'll be assuming unpredictable usage characteristics, an unknown feature set, and migration onto Aurora PostgreSQL Serverless. This configuration offers the cost benefits of serverless computing with scalable performance and the database functionalities required for a successful migration.

DATABASE MODERNISATION **STEP 1:** DISCOVERY

Versent commences every database modernisation project with a discovery phase that focuses on potential complications that may arise during the migration. Some common challenges include: out of support Oracle EE databases, signs of database corruption and large data dictionaries.

Versent works closely with your database Subject Matter Experts (SMEs) to identify during discovery any data that can be archived prior to migration. In some cases, this process can reduce database volumes by 300%.

It's important to note that the AWS Database Migration Service (DMS) only supports migrations from Oracle version 10.2 and above, so older versions will demand further steps during the Data Migration phase.

During discovery, we analyse all database objects focusing on:

- *Tables*
- *External Tables*
- *Views*
- *Materialised Views*
- *Sequences*
- *Packages*
- *Procedures*
- *Functions*

The key to a smooth database modernisation is to migrate all objects that contain data focusing on tables, external tables, and materialised views. We attempt automatic conversion of all objects using AWS Schema Conversion Tool (SCT).

SCT can be disrupted if the size of the data dictionary is too large, so it's vital to clean the dictionary up prior to modernisation. Increasing the memory parameters of SCT and selecting smaller subsets of objects is a useful tactic in this context: it enables the DB Migration Reports and converted database object code.

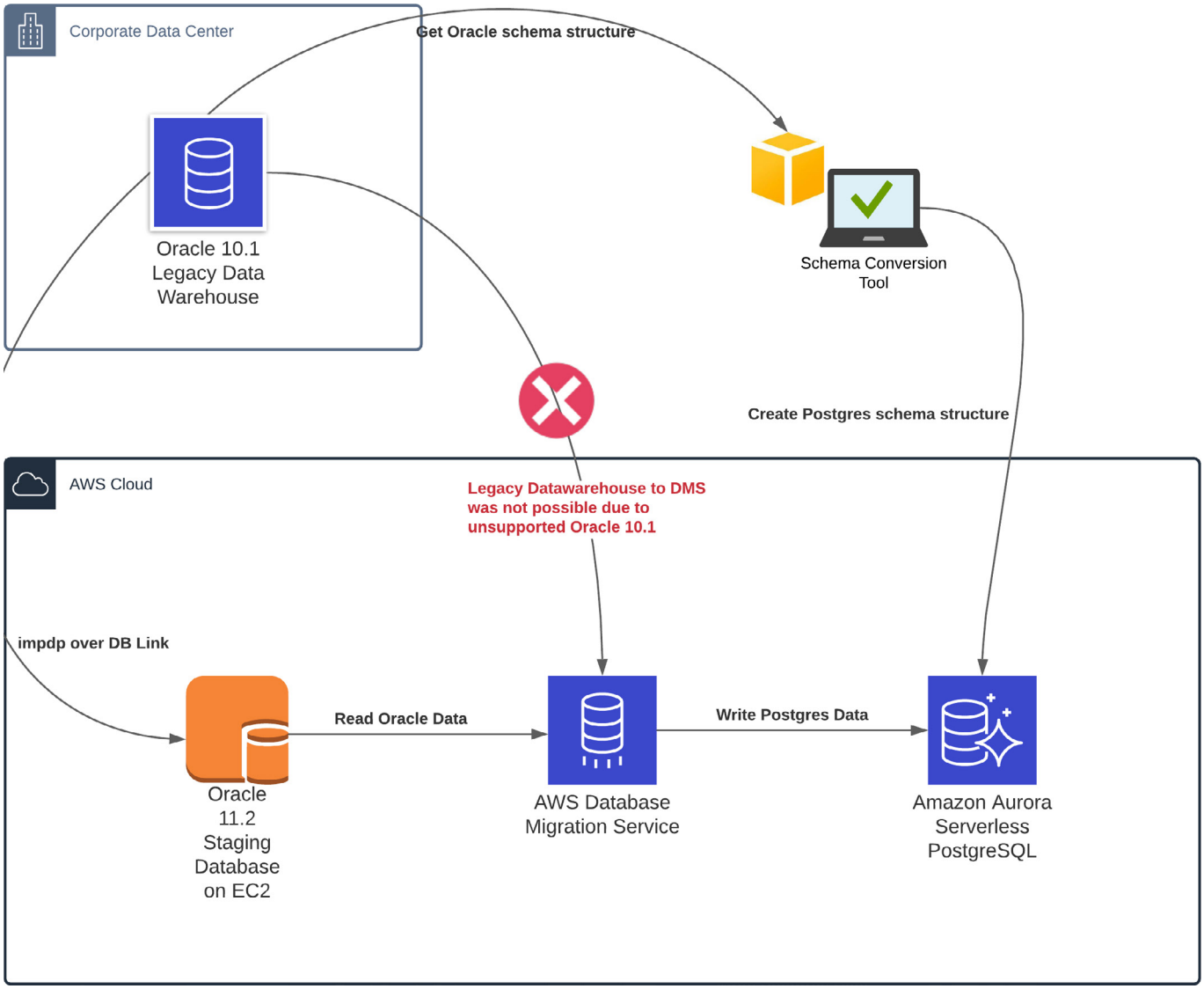
STEP 2: DATA MIGRATION

The data migration has two main parts: converting the database objects and migrating the data to the modernised environment.

We use SCT to convert the objects from Oracle to PostgreSQL automatically. The DMS then comes into play to migrate the data from the source to the target.

Oracle 10.2 and above is supported by DMS, so we need to stage the data in a supported Oracle database before executing DMS. Data is hosted in AWS on an EC2 or RDS using Oracle 11.2. Then we use Oracle's Data Pump import functionality to migrate the data from the source to the staging environment. It's worth noting that Data Pump can use database links instead of dump files.

The diagram below shows the migration pattern from Oracle to Aurora PostgreSQL Serverless:



Legacy data warehouse to: staging; to Aurora Serverless PostgreSQL.

We always recommend creating multiple DMS tasks to divide the database tables across manageable sets. The majority of tables can be migrated without issues, but some notable issues that may be encountered are corrupted date data in tables and tables with Large Objects (LOBs) but no primary keys. It's best to fix these types of issues on the staging database to avoid changes to the source.

STEP 3: VERIFICATION

After performing a migration, it's important to verify the row and column counts and check a sample set of tables to confirm the migration was successful.

DMS validation functionality is not yet supported for Aurora PostgreSQL Serverless, but once available it will be a useful feature to include in the verification process.

TAKEAWAYS

There are a few key learnings we've discovered through the process of conducting database modernisations over the years. Let's review some key takeaways that can give any team a head start.

- *Legacy databases have many backup tables, old unused code and deprecated objects that add complexity to modernisation work. It's vital to work hand in hand with database SME's to determine what really needs to be migrated and archive redundant data.*
- *Ensure that Oracle versions are confirmed during discovery because AWS DMS only supports versions 10.2 and above.*
- *Knowledge of the legacy system integrations is invaluable in any database migration validation.*
- *It's helpful to use tooling such as SCT to avoid manually converting complex code and objects.*
- *A staging database is essential for heterogeneous Oracle to PostgresSQL migrations.*
- *Migration planning should focus first on a staging database. This will minimise source database dependencies, such as data type fixes or tweaks to the schema structure to support the data migration. It also gives us a way to repeatedly test and tune the DMS tasks to further reduce migration duration.*
- *Using Oracle Data Pump with network links provides better options to utilise RDS.*
- *Using network links during the migration to RDS Oracle databases eliminates the need to put dump files in S3.*

DO YOUR DATA MIGRATION RIGHT

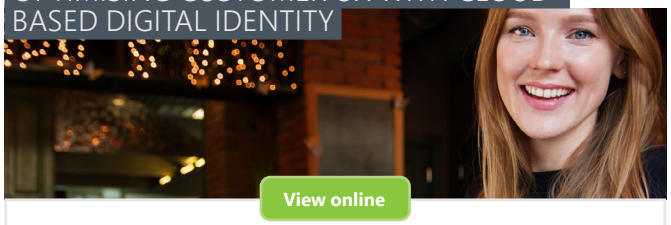
It's clear by now that modernised databases have superior performance metrics compared to legacy Oracle databases. AWS serverless databases offer all the functionality required for modern business applications and unlimited potential for growth and adaptation.

If you've got questions about database migration options, please get in touch. We'll be happy to help you make your database modernisation a success.

LEARN MORE

Want to get your enterprise's data modernisation started?
Get in touch with a Versent Expert.

OPTIMISING CUSTOMER UX WITH CLOUD-BASED DIGITAL IDENTITY



View online

When I talk to executives about customer identity & access management – CIAM – most people immediately associate it with login pages and passwords, adding friction to the customer's experience.

MODERNISE DATA TRANSFER WITH AWS TRANSFER FAMILY

View online

Is your organisation looking to modernise its legacy data transfer services? If so, you're probably investigating ways to deploy a highly available and scalable solution that can manage in-flight data transmission even during an outage or maintenance.