

AWS DMS Workshop

Core Concepts

AWS ASEAN Team

Revised 2017.10.26



About

This workshop is composed of three parts:

- Part 1: Introduction to core concepts of AWS Database Migration Services (AWS DMS) and the AWS Schema Conversion Tool (AWS SCT)
- Part 2: Lab providing hands-on with a SQL use case, specifically migrating Oracle DB -> Postgres DB
- Part 3: Lab providing hands-on with a NoSQL use case, specifically migrating MongoDB -> Amazon DynamoDB

Agenda

- Challenges of Database Migration
- AWS Database Migration Service (AWS DMS)
- AWS Schema Conversion Tool (AWS SCT)

Challenges of Database Migration

Customers Want to Migrate to AWS, but...

- Migrations create **long periods of application downtime**
- Migration tools that minimize downtime are **expensive**
- Migrations seem **too complex and expensive, especially across two different platforms**
- Migrations still **require a copy of data on-premise**
- Migrations require **skills outside their organization**

Traditional Approach to Migrate to AWS

1. Create your AWS account
2. Setup your Virtual Private Cloud (VPC) in AWS
3. Connect to AWS with a VPN or Direct Connect
4. Shutdown and backup your database
5. Transmit the backup to S3
6. Configure an EC2 instance with the DB software
7. Restore the backup
8. Configure EC2 instances for the application
9. Switch the users to use AWS

Traditional Approach to Migrate to AWS

Steps 4-9 could take weeks!

1. Create your AWS account
2. Setup your Virtual Private Cloud (VPC) in AWS
3. Connect to AWS with a VPN or Direct Connect
4. Shutdown and backup your database
5. Transmit the backup to S3
6. Configure an EC2 instance with the DB software
7. Restore the backup
8. Configure EC2 instances for the application
9. Switch the users to use AWS

AWS Database Migration Service (AWS DMS)

AWS Database Migration Service (AWS DMS)

DMS migrates databases to AWS easily and securely with minimal downtime. It can migrate your data to and from most widely used commercial and open-source databases.



ORACLE



AWS DMS Support for NoSQL

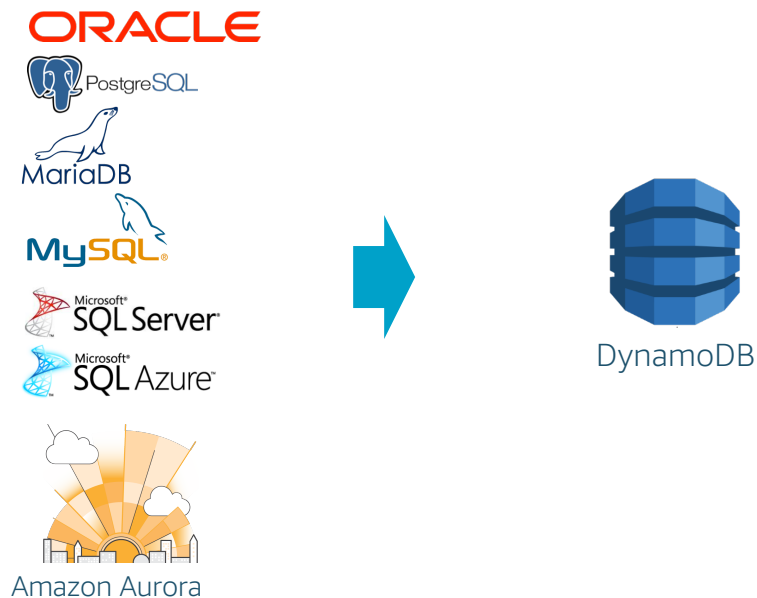
Migrate to AWS

- Move from MongoDB to Amazon DynamoDB
- Move from MongoDB to relational DBs



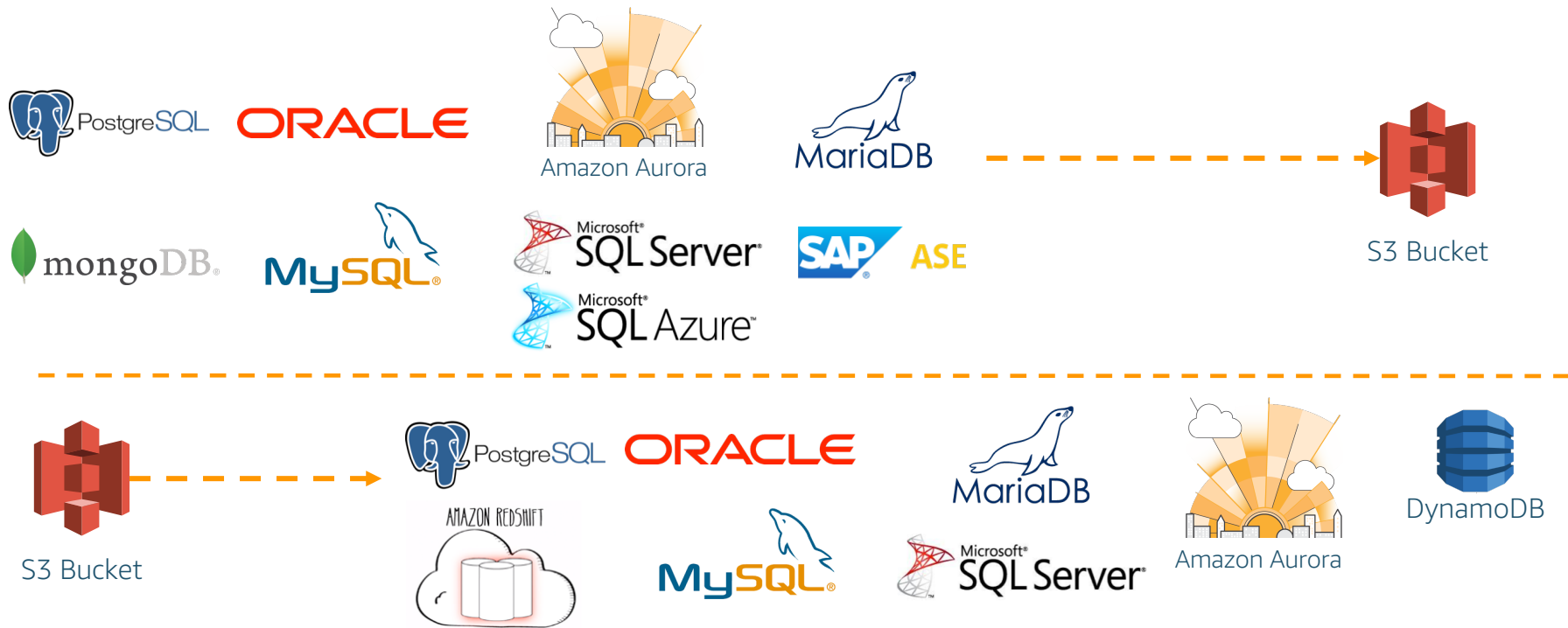
Move between NoSQL and SQL

- Re-platform database technology

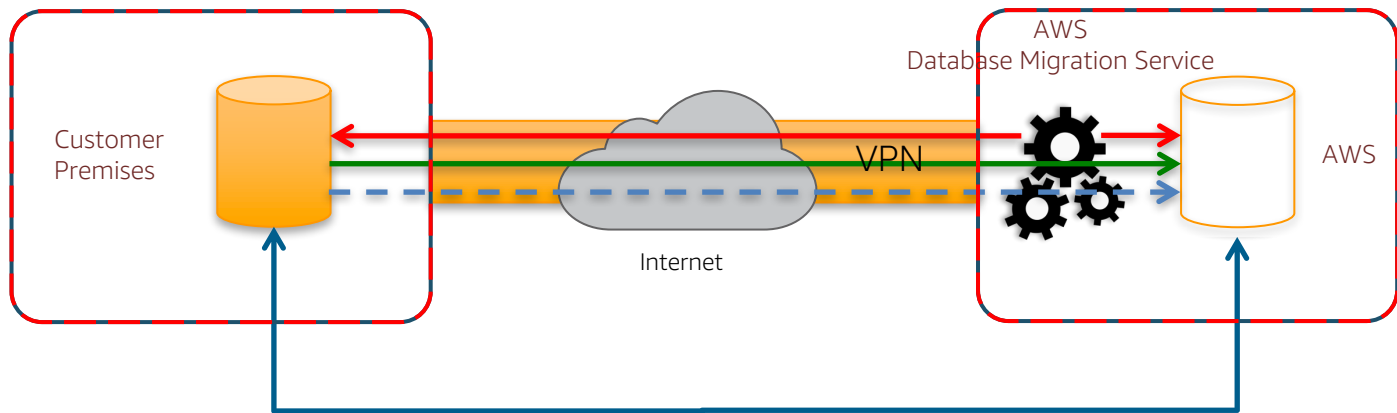


AWS DMS Support for S3 (as Source or Target)

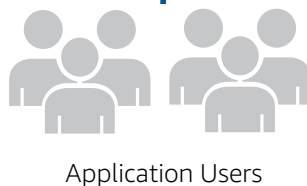
Extract Data from any supported DMS source to S3 and to any DMS target



Keep Your Apps Running During the Migration



1. Start a replication instance
2. Connect to source and target databases
3. Select tables, schemas, or databases



4. Let AWS DMS create tables, load data, and keep them in sync
5. Switch applications over to the target at your convenience

AWS DMS Key Concepts

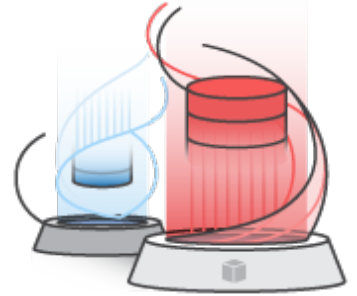
- **Replication instances** are EC2 instances that provide the processing engine for data migrations
- **Endpoints** are wrappers around the source and target databases, used by the replication instances during data migration
- **Tasks** oversee the data migration process, provide source filters and/or data transforms, and determine if data will continue to be replicated after the initial transfer is performed

AWS Schema Conversion Tool (AWS SCT)

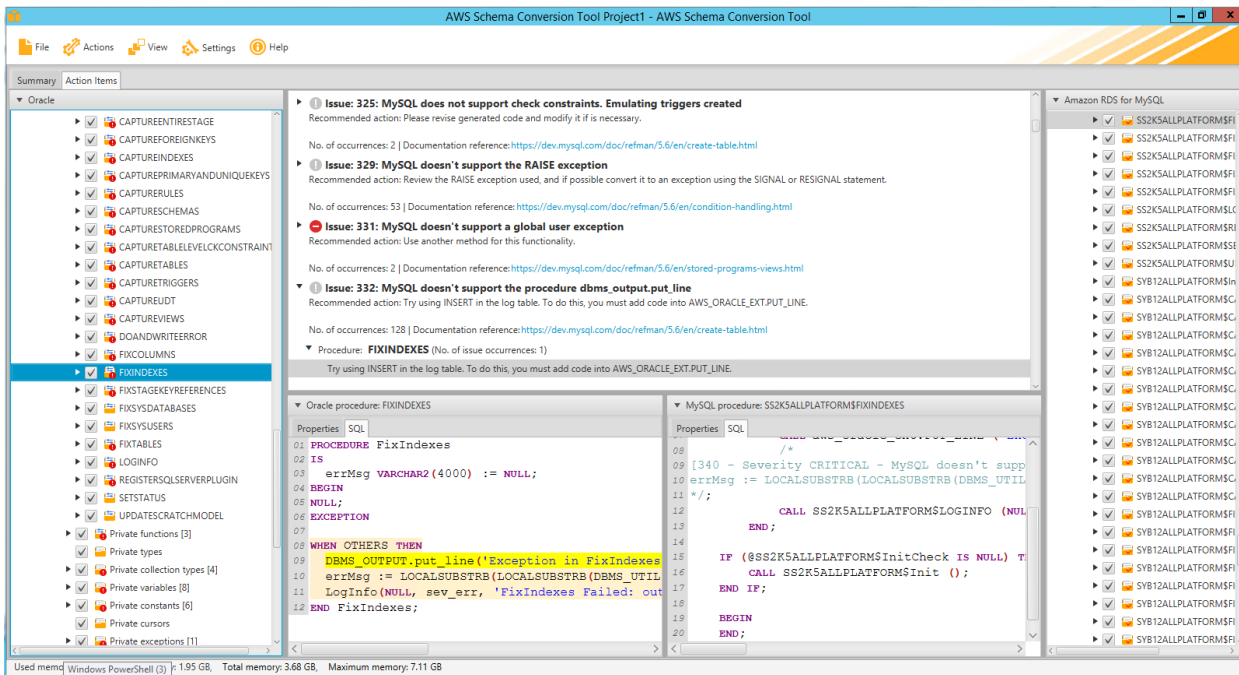


AWS Schema Conversion Tool (AWS SCT)

AWS SCT is a *desktop application* that helps *automate many database schema and code conversion tasks* when migrating between database engines or data warehouse engines



AWS SCT Helps Convert Tables, Views & Code



- Sequences
- User-Defined Types
- Synonyms
- Packages
- Stored Procedures
- Functions
- Triggers
- Schemas
- Tables
- Indexes
- Views
- Sort and distribution keys

AWS SCT Assessment Report

AWS SCT provides an Assessment Report to help you plan for any potential conflicts in your planned migration

1. Connect AWS SCT to Source and Target DBs
2. Run **Assessment Report**
3. Read **Executive Summary**
4. Follow **detailed guidance**

Database Migration Assessment Report

Source Database: RDS_ADMINISTRATION_01s_administrator@2-34-172-36-60.compute-1.amazonaws.com:81
92.0KCL
Oracle Database 12c Enterprise Edition (12.1.0.1.0 (64bit Production))



Executive Summary

We completed the analysis of your Oracle source database and estimate that 91% of the database storage objects and 100% of database code objects can be converted automatically or with minimal changes if you select Amazon Aurora as your migration target. Database storage objects include schemas, tables, columns, constraints, indexes, sequences, synonyms, user-defined types and types. Database code objects include functions, procedures, triggers, views, materialized views, events, SQL scalar functions, SQL inline functions, SQL table functions, attributes, variables, constants, table types, public types, private types, cursors, exceptions, parameters and other objects. Based on our analysis of SQL syntax elements of your source database schema, we estimate that 99.9% of your entire database schema can be converted automatically to Amazon Aurora. To complete the migration, we recommend 597 conversion action(s) ranging from simple tasks to medium-complexity actions to significant conversion actions.

Database Objects with Conversion Actions for Amazon Aurora

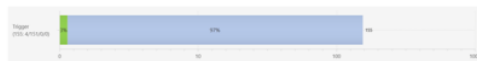
Of the total 1,276 database storage object(s) and 155 database code object(s) in the source database, we were able to identify 1,427 (91%) database storage object(s) and 155 (100%) database code object(s) that can be converted automatically or with minimal changes to Amazon Aurora.

149 (9%) database storage object(s) required 149 significant user action(s) to complete the conversion.

Figure: Conversion statistics for database storage objects



Figure: Conversion statistics for database code objects



Detailed Recommendations for Amazon Aurora Migrations

If you choose to migrate your Oracle database to Amazon Aurora, we recommend the following actions.

AWS Schema Conversion Tool Version 1.0.202

Page 1 of 4

Database Migration Assessment Report

Source Database: RDS_ADMINISTRATION_01s_administrator@2-34-172-36-60.compute-1.amazonaws.com:81
92.0KCL
Oracle Database 12c Enterprise Edition (12.1.0.1.0 (64bit Production))



Storage Object Actions

Sequence Changes

Some changes are required to sequences that cannot be converted automatically. You'll need to address these issues manually.

Issue 341: MySQL doesn't support sequences

Recommended Action: Try developing a system for sequences in your application.

Issue Code: 341 / No. of Occurrences: 134 / Estimated Complexity: Significant
Schemas: RDS_ADMINISTRATION Sequences: BACKUP_ID_SEQUENCE
Schemas: RDS_ADMINISTRATION Sequences: CERTIFICATE_ID_SEQUENCE
Schemas: RDS_ADMINISTRATION Sequences: CHARACTER_SET_ID_SEQ
Schemas: RDS_ADMINISTRATION Sequences: CUSTOMER_SUBNET_GROUP_ID_SEQ
Schemas: RDS_ADMINISTRATION Sequences: CUSTOMER_SUBNET_ID_SEQ
+129 more

Index Changes

Some changes are required to indexes that cannot be converted automatically. You'll need to address these issues manually.

Issue 207: MySQL doesn't support function indexes

Recommended Action: Review your code and try to use simple indexes.

Issue Code: 207 / No. of Occurrences: 3 / Estimated Complexity: Significant
Documentation References: <https://dev.mysql.com/doc/refman/5.6/en/create-table.html>
Schemas: RDS_ADMINISTRATION Tables: DBL_ENGINE_SEEDS Indexes: I_DBL_ENG_SEED_DBL_ENG_CONF_ID
Schemas: RDS_ADMINISTRATION Tables: RDS_SYSTEM_ACCOUNTS Indexes: I_SYS_ACCOUNT_DEFAULT
Schemas: RDS_ADMINISTRATION Tables: RUNNABLE_DBL_CONFIG Indexes: I_RUNNABLE_DBL_CFG_PREFERRED

Constraint Changes

Some changes are required to constraints that cannot be converted automatically. You'll need to address these issues manually.

Issue 210: MySQL doesn't support FUNCTION AS DEFAULT VALUE

Recommended Action: Try using a trigger.

Issue Code: 210 / No. of Occurrences: 2 / Estimated Complexity: Simple
Documentation References: <https://dev.mysql.com/doc/refman/5.6/en/create-table.html>
Schemas: RDS_ADMINISTRATION Tables: CUSTOMERS Constraints: CK_CUSTOMER_TRUST_LEVEL_STATE: 0:10
Schemas: RDS_ADMINISTRATION Tables: STORAGE_VOLUMES Constraints: CK_SV_LIFECYCLE: 0:8

Issue 325: MySQL does not support check constraints. Emulating triggers created

Recommended Action: Please revise generated code and modify it if it is necessary.

Issue Code: 325 / No. of Occurrences: 283 / Estimated Complexity: Simple
Documentation References: <https://dev.mysql.com/doc/refman/5.6/en/create-table.html>

AWS Schema Conversion Tool Version 1.0.202

Page 2 of 4

Database Migration Assessment Report (Sample)

AWS SCT Pricing and Permitted Use

\$0

for software license

Pricing

- Free software license
- For active AWS customers with accounts in good standing

Permitted Use

- Use AWS SCT to migrate database schemas to Amazon RDS, Amazon Redshift, or Amazon EC2-based databases
- To use AWS SCT to migrate schemas to other destinations, contact for special pricing

Next:

SQL Lab: Oracle to PostgreSQL