Getting Started with AWS Database Migration Service

Blair Layton, Business Development, APAC

June, 2016

amazon web services

Customers Want to Migrate to AWS, but…

They can’t afford long periods of application downtime

Tools that enable minimal downtime are expensive

It seems too complex and expensive to migrate

They still need a copy of the data on-premise

They want to migrate to an open source database

Sending large volumes of data to AWS requires an expensive international network link

They don’t have the skills inside 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

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

Steps 4-9 could take a week or more!

Now There is a Better Way!

AWS Database Migration Service

PostgreSQL ORACLE Amazon Aurora

AMAZON REDSHIFT MySQL MariaDB Microsoft SQL Server

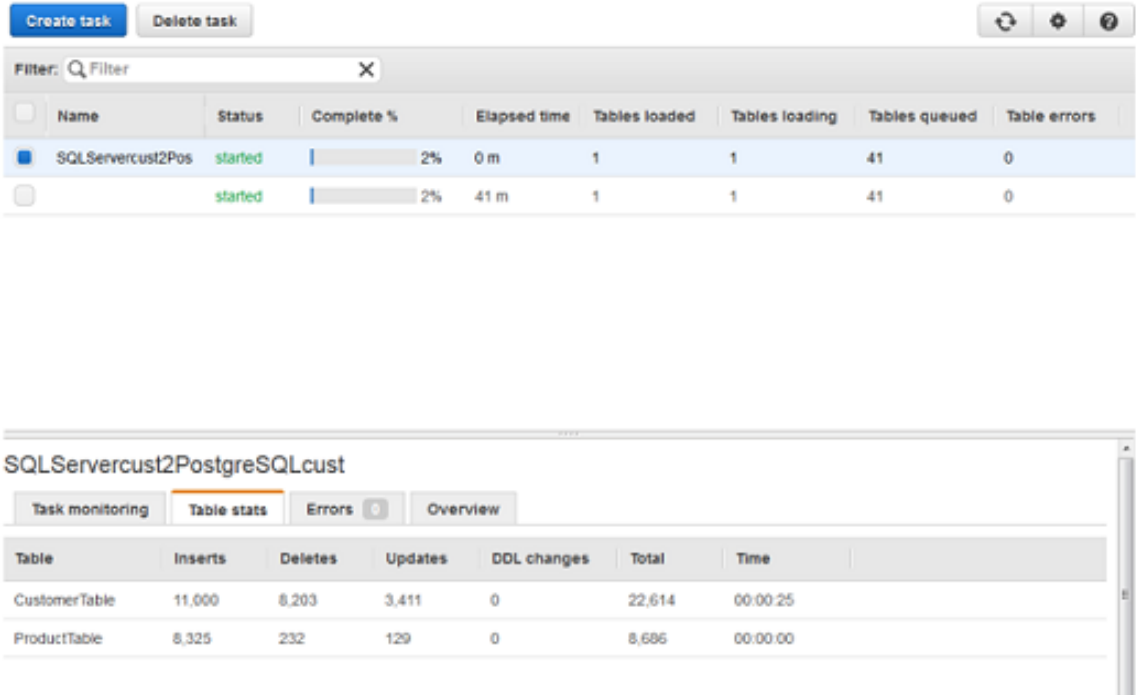
Start your first migration in 10 minutes or less

Keep your apps running during the migration

Replicate within, to or from Amazon EC2 or RDS

Move data to the same or a different database engine

DMS Console



Console (or API) controlled

Set up replication instances, tasks

Use as many or as few as you want, even against the same database instance

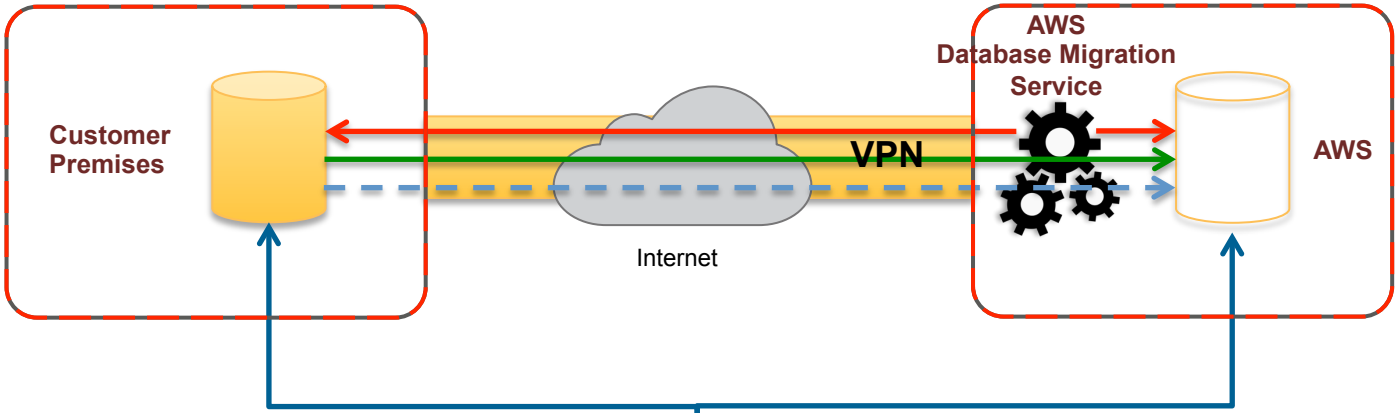
Choose the power/speed/cost of your migration

- T2.micro – C4.4xlarge

Choose the tables you want

On-prem->RDS/EC2, EC2<>EC2, RDS<>RDS, RDS/EC2->on-prem

Keep Your Apps Running During the Migration



Start a replication instance

Connect to source and target databases

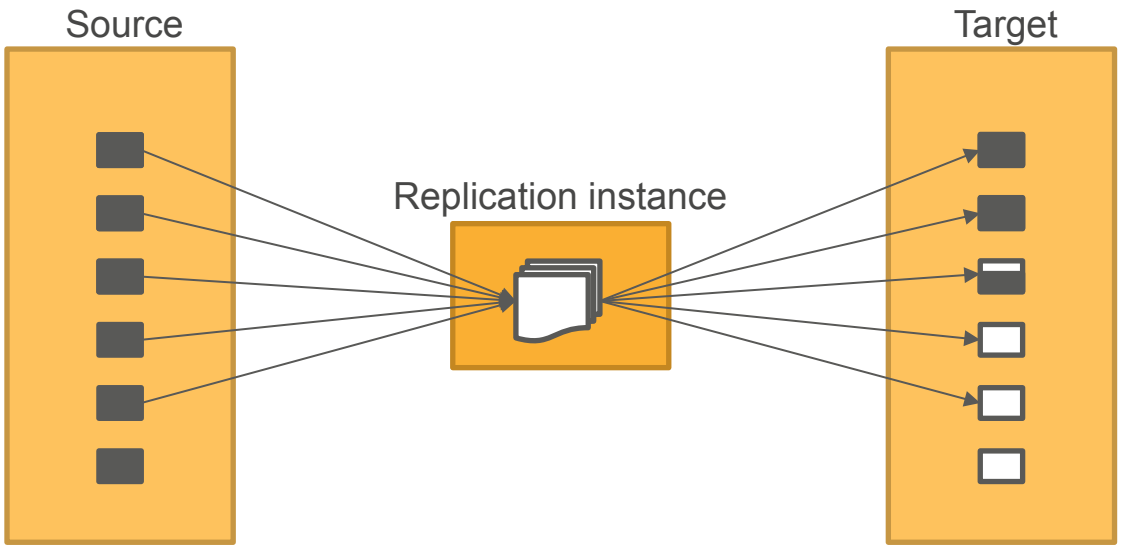
Select tables, schemas, or databases

Application Users

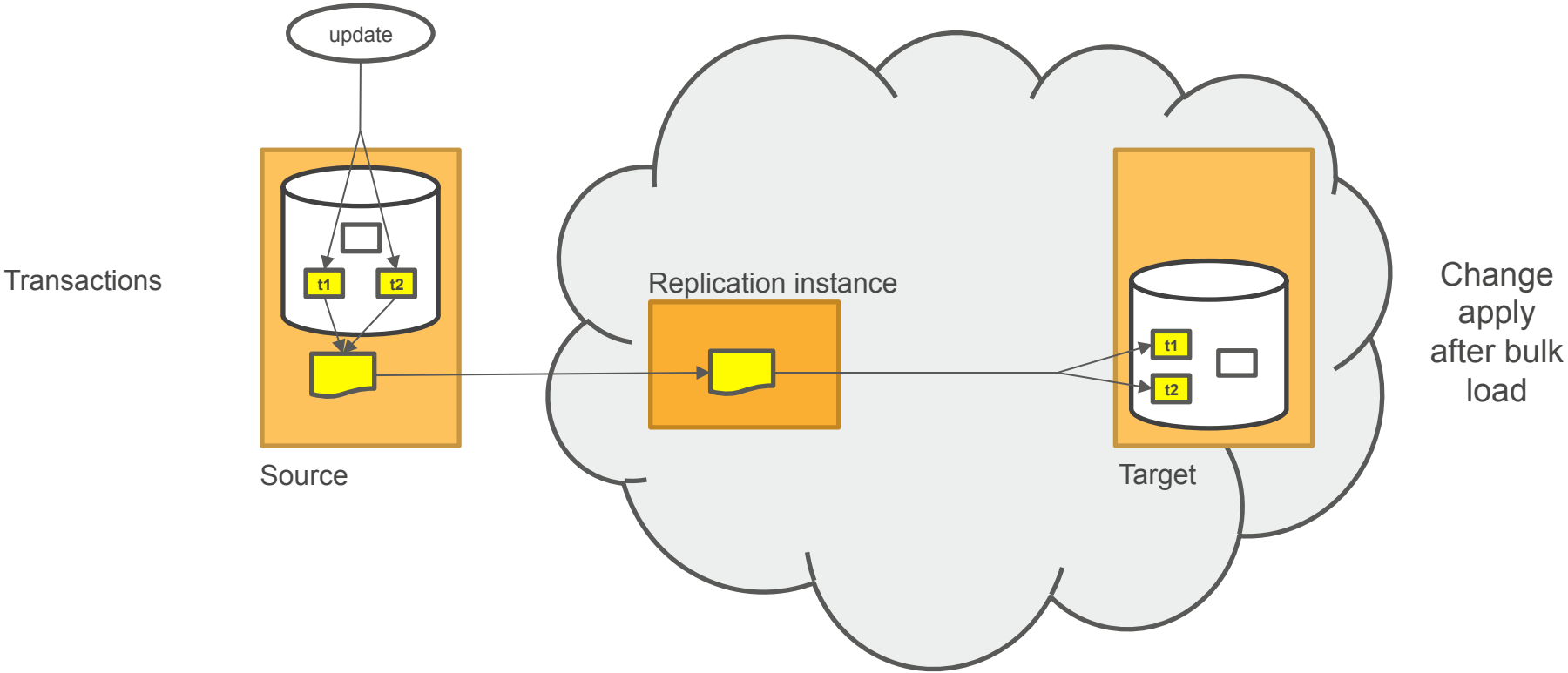
Let AWS DMS create tables, load data, and keep them in sync

Switch applications over to the target at your convenience

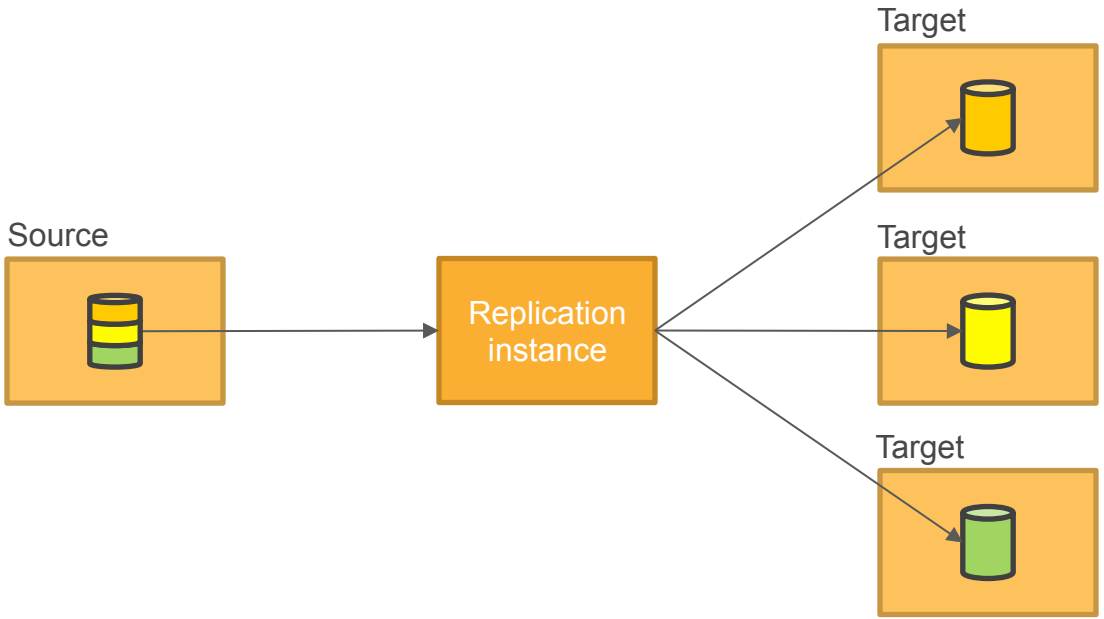
Load is Table by Table



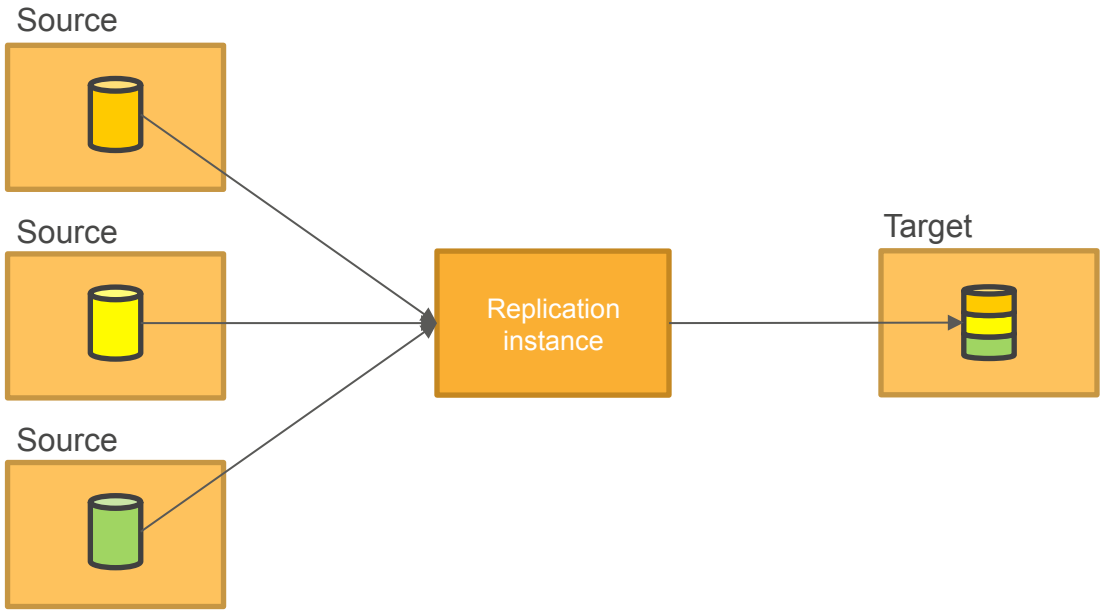
Change Data Capture (CDC) and Apply



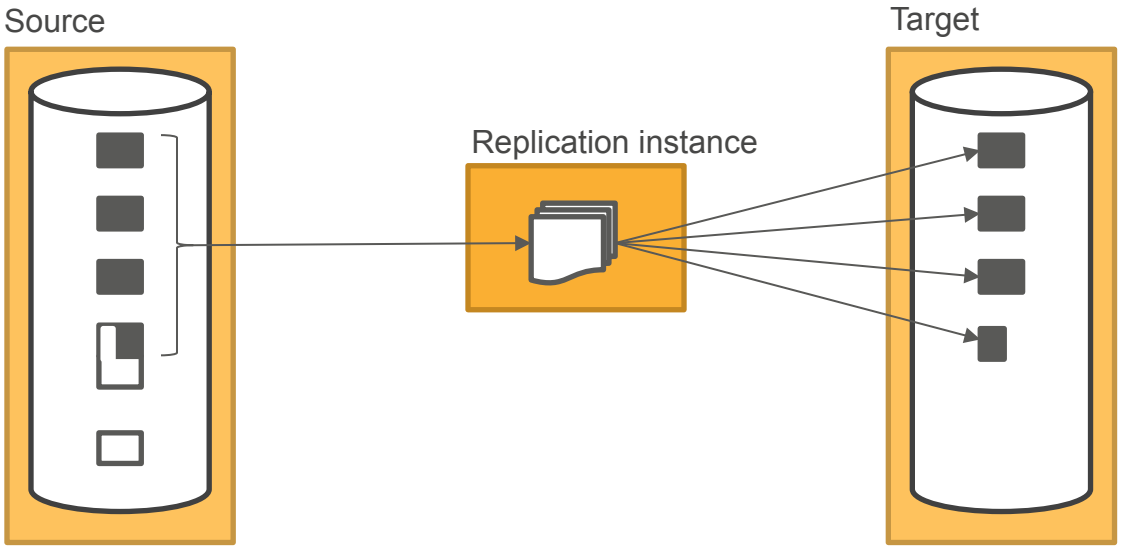
Multiple Targets



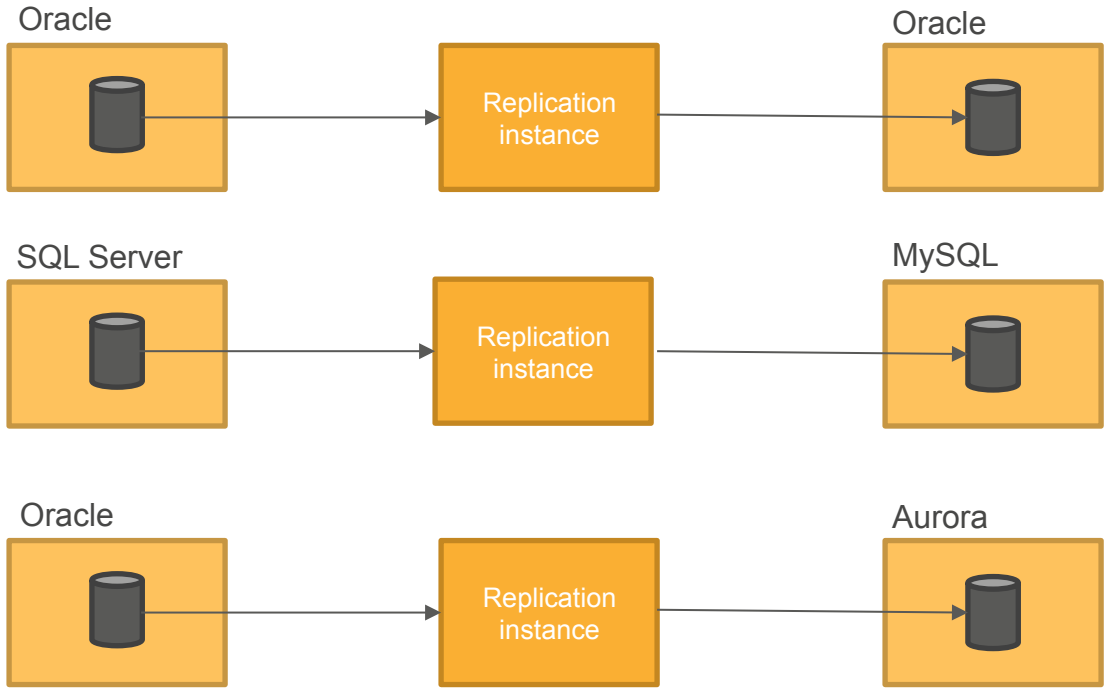
Multiple Sources



Customers Don’t Have to Take Everything



Homogenous or Heterogeneous



Sources for AWS Database Migration Service

Customers can use the following databases as a source for data migration using AWS DMS:

On-premises and Amazon EC2 instance databases:

- Oracle Database 10g – 12c
- Microsoft SQL Server 2005 – 2014
- MySQL 5.5 – 5.7
- MariaDB (MySQL-compatible data source)
- PostgreSQL 9.4 – 9.5

Amazon RDS instance databases:

- Oracle Database 11g – 12c
- Microsoft SQL Server 2008R2 and 2012. CDC operations are not supported yet.
- MySQL versions 5.5 – 5.7
- MariaDB (MySQL-compatible data source)
- PostgreSQL 9.4 – 9.5. CDC operations are not supported yet.
- Amazon Aurora (MySQL-compatible data source)

Targets for AWS Database Migration Service

Customers can use the following databases as a target for data replication using AWS DMS:

On-premises and Amazon EC2 instance databases:

- Oracle Database 10g – 12c
- Microsoft SQL Server 2005 – 2014
- MySQL 5.5 – 5.7
- MariaDB (MySQL-compatible data source)
- PostgreSQL 9.3 – 9.5

Amazon RDS instance databases:

- Oracle Database11g – 12c
- Microsoft SQL Server 2008R2 and 2012
- MySQL 5.5 – 5.7
- MariaDB (MySQL-compatible data source)
- PostgreSQL 9.3 – 9.5
- Amazon Aurora (MySQL-compatible data source)

Amazon Redshift

AWS Database Migration Service Pricing

Instance Type	vCPU	Memory (GiB)	Price/Hr
General Purpose			
dms.t2.micro	1	1	$0.018
dms.t2.small	1	2	$0.036
dms.t2.medium	2	4	$0.072
dms.t2.large	2	8	$0.144
Compute Optimized			
dms.c4.large	2	3.75	$0.154
dms.c4.xlarge	4	7.5	$0.308
dms.c4.2xlarge	8	15	$0.616
dms.c4.4xlarge	16	30	$1.232

T2 for developing and periodic data migration tasks

C4 for large databases and minimizing time

T2 pricing starts at $0.018 per Hour for T2.micro

C4 pricing starts at $0.154 per Hour for C4.large

50GB GP2 storage included with T2 instances

100GB GP2 storage included with C4 instances

Data transfer inbound and within AZ is free

Data transfer across AZs starts at $0.01 per GB

Migrate 5 TB in 33 Hours!

*In our testing, under mostly ideal conditions, we were able to migrate **5 TB** of relatively evenly distributed data from a database on Amazon EC2 to a database on Amazon RDS in about **33 hours**. The data included 4 large (250 GB) tables, a huge (1 TB) table, 1000 small to moderately sized tables, 3 tables which contained LOBs varying between 25 GB and 75 GB, and 10,000 very small tables.*

- DMS Documentation

For less than $10 per TB!

New Use Cases with DMS

New Use Cases with DMS

Migration of business critical applications

Migration from Classic to VPC

Cheap read replicas for Oracle

Read replicas on other engines

Cross region read replicas for Oracle and SQL Server

Analytics in the Cloud

Dev/Test and Production environment sync.

What Else Could You Do With DMS?

My Database is Too Big!

Data Ingestion with AWS

AWS Import/Export Disk – Ship your hard disks to AWS

AWS Import/Export Snowball – A secure storage appliance with up to 80 TB that AWS ships to you

Amazon S3 Transfer Acceleration – Use AWS Edge Locations nearest to you to transfer data on Amazon's optimized network up to 300% faster. You only need to pay for a local network connection!

AWS Schema Conversion Tool

The AWS Schema Conversion Tool helps automate many database schema and code conversion tasks when migrating from Oracle and SQL Server to open source database engines.

Features

Oracle and SQL Server schema conversion to MySQL/Aurora/MariaDB and PostgreSQL

Database Migration Assessment report for choosing the best target engine

Code browser that highlights places where manual edits are required

Secure connections to your databases with SSL

SCT helps with converting tables, views, & code

ORACLE

Microsoft SQL Server

PostgreSQL

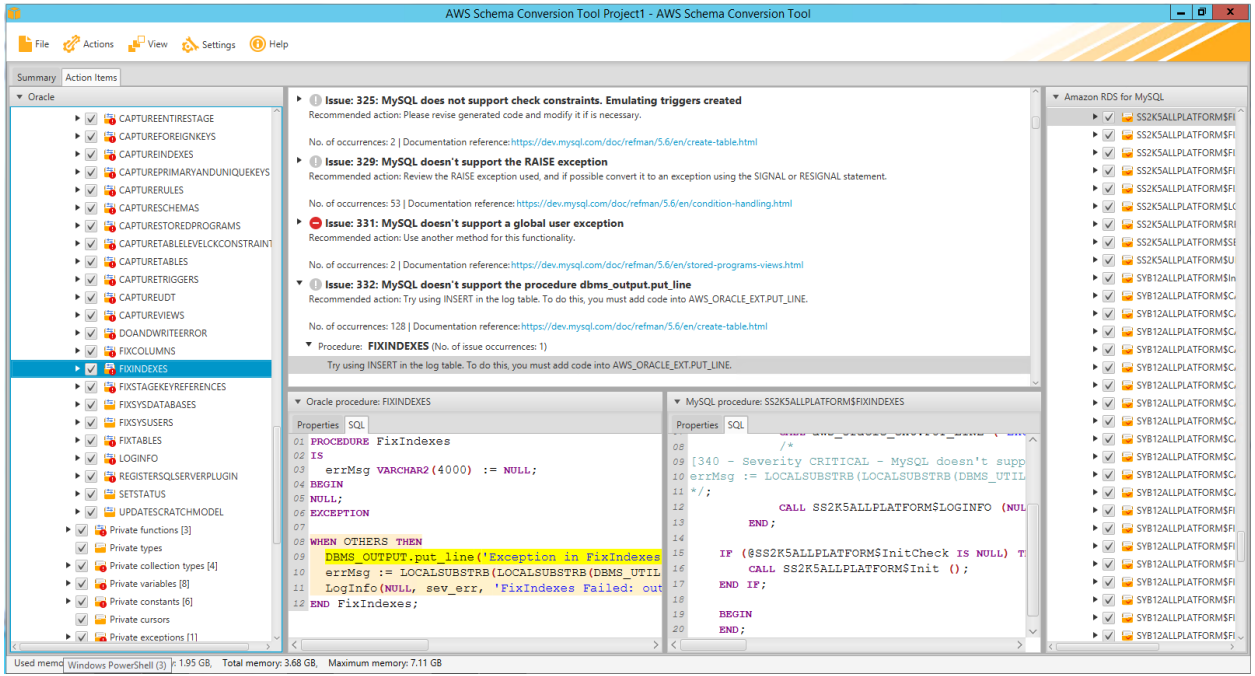
MySQL

PostgreSQL

MariaDB

Amazon Aurora

MySQL



- Sequences
- User Defined Types
- Synonyms
- Packages
- Stored Procedures
- Functions
- Triggers
- Schemas
- Tables
- Indexes
- Views

SCT can tell you how hard the migration will be

Database Migration Assessment Report

Source Database: RDS_ADMINISTRATION.rds_administration@ec2-54-172-36-60.compute-1.amazonaws.com:8192:ORCL

Oracle Database 12c Enterprise Edition 12.1.0.1.0 (64bit Production)

amazon web services

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 define types and types. Database code objects include functions, procedures, packages, 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 1,576 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 objects 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

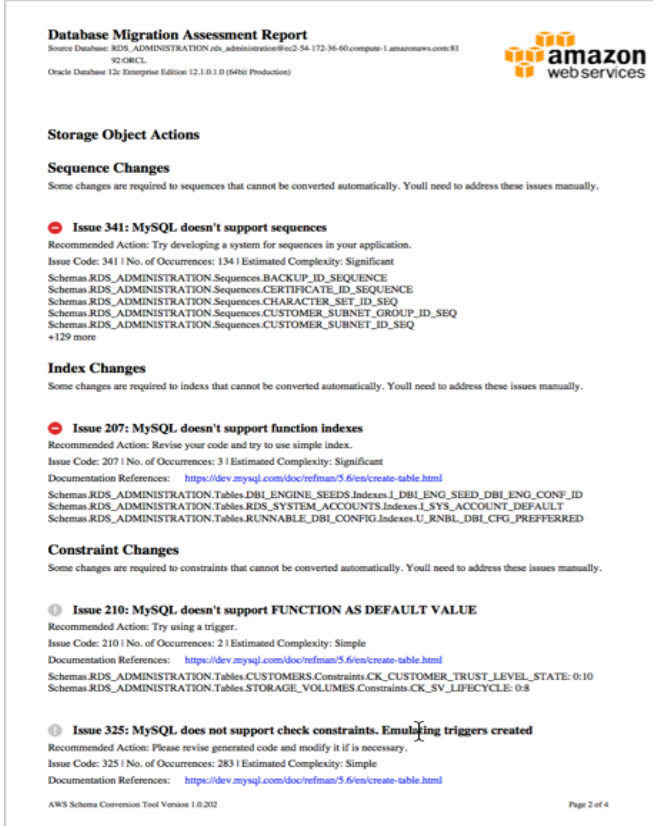
If you choose to migrate your Oracle database to Amazon Aurora, we recommend the following actions.

1. Connect SCT to Source and Target databases.

2. Run Assessment Report.

3. Read Executive Summary.

4. Follow detailed instructions.



Database Migration Assessment Report

Source Database: RDS_ADMINISTRATION.rds_administration@ec2-54-172-36-60.compute-1.amazonaws.com:8192:ORCL

Oracle Database 12c Enterprise Edition 12.1.0.1.0 (64bit Production)

amazon web services

Storage Object Actions

Sequence Changes

Some changes are required to sequences that cannot be converted automatically. Youll 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 indexs that cannot be converted automatically. Youll need to address these issues manually.

Issue 207: MySQL doesn't support function indexes

Recommended Action: Revise your code and try to use simple index.

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.DBI_ENGINE_SEEDS.Indexes.I_DBI_ENG_SEED_DBI_ENG_CONF_ID
Schemas.RDS_ADMINISTRATION.Tables.RDS_SYSTEM_ACCOUNTS.Indexes.I_SYS_ACCOUNT_DEFAULT
Schemas.RDS_ADMINISTRATION.Tables.RUNNABLE_DBI_CONFIG.Indexes.U_RNBL_DBI_CFG_PREFFERRED

Constraint Changes

Some changes are required to constraints that cannot be converted automatically. Youll 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 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

Pricing and Terms and Conditions

$0

for software license

Pricing

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

Allowed Use

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

Database Migration Process

STEP 1:

Source: Oracle Database on-premises, in EC2 or RDS

AWS Schema Conversion Tool

Target: Amazon Aurora Database

STEP 2:

Source: Oracle Database on-premises, in EC2 or RDS

AWS Database Migration Service

Target: Amazon Aurora Database

Thomas Publishing 

Thomas Publishing has been in business for over a century, connecting buyers and suppliers across all industrial sectors, evolving from an industrial trade print publisher into industry's most respected group of digital-friendly businesses.

Previously ran homegrown applications on a single, monolithic Oracle database.

With a growing user base, performance declined as licensing costs increased.

Working with Apps Associates, Thomas Publishing championed a proof of concept project to migrate to the cloud.

They leveraged the Schema Conversion Tool to convert their Oracle schema to Amazon Aurora and used DMS to migrate the data.

The proof of concept was successful and they are moving the remainder of their applications and data to AWS.

Pegasystem

Pegasystems, whose customers include many of the world's most sophisticated and successful enterprises, develops strategic applications for sales, marketing, service, and operations.

Pegasystems used DMS to migrate customers from their legacy cloud environment (Oracle) to their new cloud 2.1 environment using RDS PostgreSQL.

They experienced better availability and performance. The new 2.1 environment is built using AWS best practices and services such as ELB, auto scaling, and of course RDS Multi-AZ to remove single points of failure in the architecture.

They also experienced cost savings by moving from Oracle to RDS PostgreSQL.

AWS Database Migration Partners

Summary

Customers Want to Migrate to AWS, but…

They can’t afford any application downtime

Tools that enable minimal downtime are expensive

It seems too complex to migrate successfully

They still need a copy of the data on-premise

They want to migrate to an open source database

Sending large volumes of data to AWS requires an expensive international network link

They don’t have the skills inside their organization

Customers Want to Migrate to AWS, but…

~~They can't afford any application downtime~~

Tools that enable minimal downtime are expensive

It seems too complex to migrate successfully

They still need a copy of the data on-premise

They want to migrate to an open source database

Sending large volumes of data to AWS requires an expensive international network link

They don't have the skills inside their organization

Customers Want to Migrate to AWS, but…

~~They can't afford any application downtime~~

~~Tools that enable minimal downtime are expensive~~

It seems too complex to migrate successfully

They still need a copy of the data on-premise

They want to migrate to an open source database

Sending large volumes of data to AWS requires an expensive international network link

They don't have the skills inside their organization

Customers Want to Migrate to AWS, but…

~~They can't afford any application downtime~~

~~Tools that enable minimal downtime are expensive~~

~~It seems too complex to migrate successfully~~

They still need a copy of the data on-premise

They want to migrate to an open source database

Sending large volumes of data to AWS requires an expensive international network link

They don't have the skills inside their organization

Customers Want to Migrate to AWS, but…

~~They can't afford any application downtime~~

~~Tools that enable minimal downtime are expensive~~

~~It seems too complex to migrate successfully~~

~~They still need a copy of the data on-premise~~

They want to migrate to an open source database

Sending large volumes of data to AWS requires an expensive international network link

They don't have the skills inside their organization

Customers Want to Migrate to AWS, but…

~~They can't afford any application downtime~~

~~Tools that enable minimal downtime are expensive~~

~~It seems too complex to migrate successfully~~

~~They still need a copy of the data on-premise~~

~~They want to migrate to an open source database~~

Sending large volumes of data to AWS requires an expensive international network link

They don't have the skills inside their organization

Customers Want to Migrate to AWS, but…

~~They can't afford any application downtime~~

~~Tools that enable minimal downtime are expensive~~

~~It seems too complex to migrate successfully~~

~~They still need a copy of the data on-premise~~

~~They want to migrate to an open source database~~

~~Sending large volumes of data to AWS requires an expensive international network link~~

They don't have the skills inside their organization

Customers Want to Migrate to AWS, but…

~~They can't afford any application downtime~~

~~Tools that enable minimal downtime are expensive~~

~~It seems too complex to migrate successfully~~

~~They still need a copy of the data on-premise~~

~~They want to migrate to an open source database~~

~~Sending large volumes of data to AWS requires an expensive international network link~~

~~They don't have the skills inside their organization~~

Now you can Migrate Your Business Critical Applications to AWS!

Remember to complete your evaluations!

Thank You!

amazon web services