Migrating from Oracle to PostgreSQL with AWS DMS and SCT

[Eli DOE – Database Engineer]
[Amit Grover – Senior Database Engineer]

FEB 2020

Agenda

- ❑ How can I get to the Cloud
- ❑ AWS Schema Conversion Tool [SCT]
- ❑ AWS SCT – Best Practices
- ❑ AWS Database Migration Service [DMS]
- ❑ AWS DMS – Components
- ❑ AWS DMS – Best Practices
- ❑ Oracle – As Source
- ❑ Postgres – As Target

aws

How can I get to the cloud

- ❑ How will my on-premise data migrate to the Cloud?
- ❑ How can I make it less intrusive and minimize downtime?
- ❑ How can I synchronize my on-premise database to the Cloud?
- ❑ Can I get help moving off of commercial databases?
- ❑ How can I move data to my data lake?

aws

Migration use to be Cost + Complexity + Time

- ❑ Commercial data migration and replication software
- ❑ Complex to set up and manage
- ❑ Application downtime
- ❑ Database-engine-specific application code

aws

Migration Today

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

Agenda

- ❑ How can I get to the Cloud
- ❑ AWS Schema Conversion Tool [SCT]
- ❑ AWS SCT – Best Practices
- ❑ AWS Database Migration Service [DMS]
- ❑ AWS DMS – Components
- ❑ AWS DMS – Best Practices
- ❑ Oracle – As Source
- ❑ Postgres – As Target

aws

What are DMS and SCT?

Our goal: Allow customers the freedom to choose the best data platform for their needs #DBFreedom

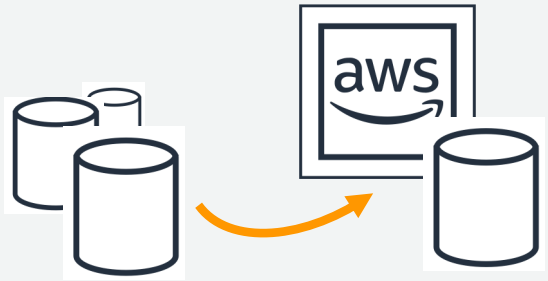
AWS Schema Conversion Tool (SCT) converts your commercial database and data warehouse schemas to open-source engines or AWS-native services, such as Amazon Aurora and Redshift

AWS Database Migration Service (DMS) easily and securely migrates and/or replicate your databases *and* data warehouses to AWS

When to use DMS and SCT?

Modernize

Migrate



Replicate

aws

AWS Schema Conversion Tool (SCT)

Modernize

- **Modernize your database**

  Convert your Oracle, SQL Server, or Db2 LUW to PostgreSQL, MySQL, or Amazon Aurora

- **Modernize your data warehouse**
  - Convert your Oracle, SQL Server, Netezza, Greenplum, Vertica or Teradata to
  - Amazon Redshift

MySQL

Amazon Aurora

PostgreSQL

Amazon Redshift

aws

AWS Database Migration Service (DMS)

Migrate

- Migrate business-critical applications
- Migrate data warehouse to Redshift
- Upgrade to a minor version
- Consolidate shards into Aurora
- Archive old data
- Migrate from NoSQL to SQL, SQL to NoSQL or NoSQL to NoSQL

Amazon Redshift

Amazon RDS

Amazon S3

Amazon DynamoDB

Amazon Aurora

aws

AWS Database Migration Service (DMS)

Replicate

- **Create** cross region Read Replicas
- **Run** your analytics in the cloud
- **Populate** your data lake

aws

Agenda

- ❑ How can I get to the Cloud
- ❑ AWS Schema Conversion Tool [SCT]
- ❑ AWS SCT – Best Practices
- ❑ AWS Database Migration Service [DMS]
- ❑ AWS DMS – Components
- ❑ AWS DMS – Best Practices
- ❑ Oracle – As Source
- ❑ Postgres – As Target

aws

When to use SCT?

Modernize

Modernize your database tier

ORACLE

IBM DB2

Microsoft SQL Server

SAP SYBASE

Amazon Aurora

PostgreSQL

MySQL

Modernize and **Migrate**

Cassandra to Amazon DynamoDB

cassandra

Amazon DynamoDB

Modernize and **Migrate** your Data Warehouse to Amazon Redshift

VERTICA

Microsoft SQL Server

TERADATA

NETEZZA

Greenplum

ORACLE

Amazon Redshift

aws

AWS SCT Product Highlights

Assessment Report

- ❑ Assessment of migration compatibility of source databases with open-source database engines – RDS for MySQL, RDS for PostgreSQL, and Amazon Aurora
- ❑ Recommends best target engine
- ❑ Provides details level of efforts to complete migration

Converts Schema and Code

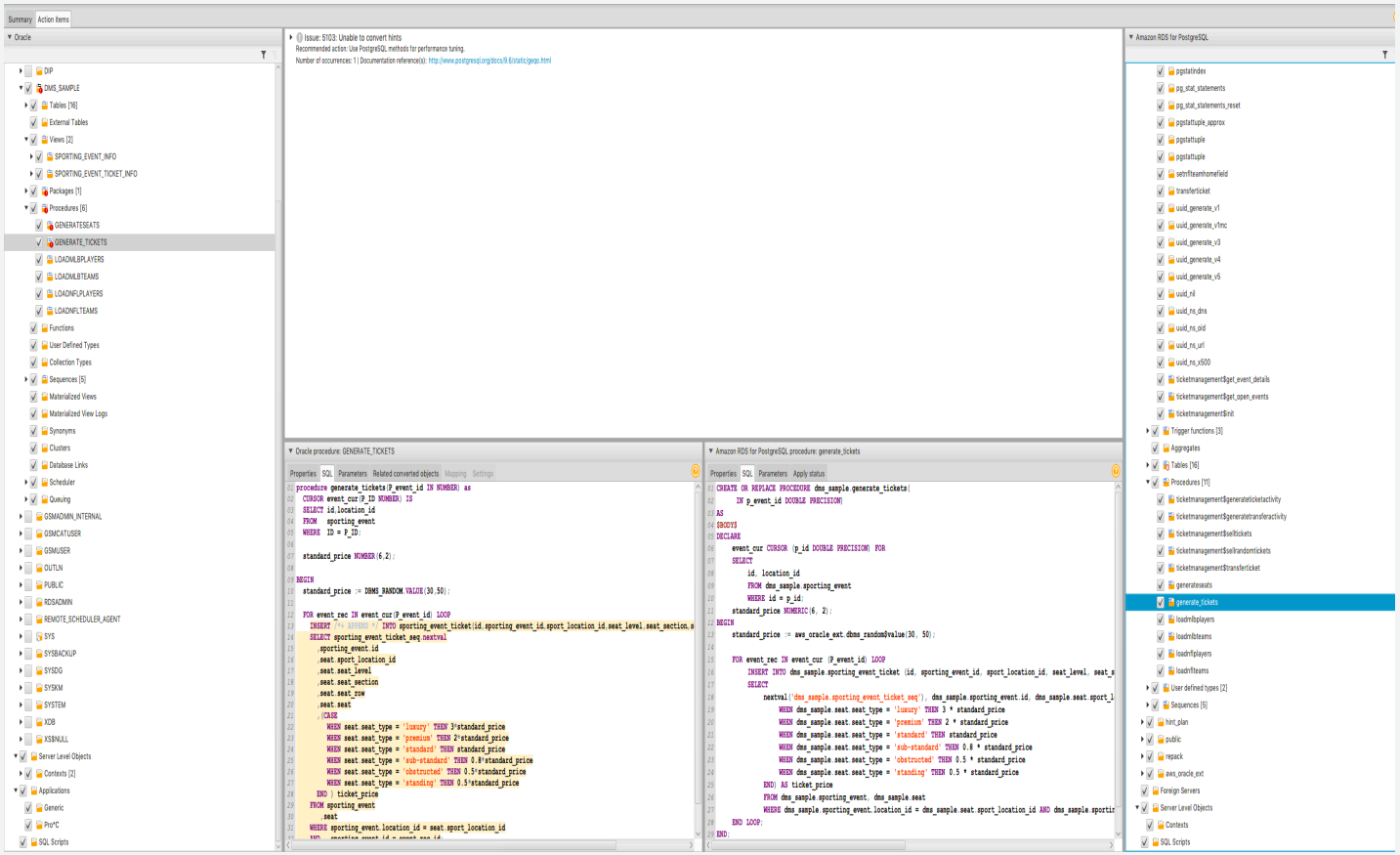
- ❑ Attempts to convert all schema and code objects to the target engine, including stored procedures and functions
- ❑ Scans and converts embedded SQL statements in app code
- ❑ Generates a report with recommendations

Extracts and Migrates DW to Amazon Redshift

- ❑ Extracts data through local migration agents
- ❑ Files are loaded to an Amazon S3 bucket and to Amazon Redshift
- ❑ Netezza
- ❑ Vertica
- ❑ Greenplum
- ❑ Teradata
- ❑ Oracle
- ❑ SQL Server

aws

SCT helps with converting tables, views, and code



- ❑ Sequences
- ❑ User-defined types
- ❑ Synonyms
- ❑ Packages
- ❑ Stored procedures
- ❑ Functions
- ❑ Triggers
- ❑ Schemas
- ❑ Tables
- ❑ Indexes
- ❑ Views
- ❑ Sort and distribution keys

SCT Migration Assessment Report

- Assessment of migration compatibility of source databases with open-source database engines – RDS MySQL, RDS PostgreSQL and Aurora
- Recommends best target engine
- Provides details level of efforts to complete migration

aws

Workload Qualification Framework (WQF)

Assesses and classifies OLTP and OLAP workloads to determine the ease of migration, staff-hour consumption, and inform on appropriate target AWS Services

- ❑ Designed for Solutions Architects, Partners and Consultants
- ❑ Assess workloads by evaluating proprietary features, complexity, technology, size, etc.
- ❑ Recommends a migration strategy and migration tools
- ❑ Gives migration engineers clear and actionable information
- ❑ Integrated with AWS SCT and DMS

WQF classifies OLTP and DW workloads into five categories:

	Category 1	ODBC/JBDC workloads
	Category 2	Light, proprietary feature workloads
	Category 3	Heavy, proprietary feature workloads
	Category 4	Engine-specific workloads
	Category 5	Non-portable, high-risk or lift-and-shift

- ❑ Migration Complexity Assessments
- ❑ Workload Migration Strategy
- ❑ Migration Tools

aws

WQF Assessment Report

	WQF Assessment Report	
Workload category	Workload Category 5: Non-portable, Unacceptable Risk, or Lift & Shift workloads. 6 significant problems were found.	
Analysis	**Migration to Amazon RDS for MySQL**	**Migration to Amazon RDS for PostgreSQL**
	Critical complexity	**Critical complexity**
Summary of Object Analysis	We assessed the modules below and here is an aggregated view of the analysis per module: 1. Databases: We analyzed and assessed 0 databases along with all objects in it. 2. Schemas: We analyzed and assessed 62 schemas in total. More detail about each schema object can be found in the detail reports for each engine. 3. Application code modules: We analyzed 0 applications in total. More details about each application can be found in detail reports for each engine	We assessed the modules below and here is an aggregated view of the analysis per module: 1. Databases: We analyzed and assessed 0 databases along with all objects in it. 2. Schemas: We analyzed and assessed 62 schemas in total. More detail about each schema object can be found in the detail reports for each engine. 3. Application code modules: We analyzed 0 applications in total. More details about each application can be found in detail reports for each engine
Database object analysis	We completed the analysis of your database servers. We found 50003 objects in 62 schemas and their average compatibility is 90%. There were 83470 Action items found. For more details please see tab 'Migration Efforts for MySQL'.	We completed the analysis of your database servers. We found 50003 objects in 62 schemas and their average compatibility is 95%. There were 29776 Action items found. For more details please see tab 'Migration Efforts for PG'.
Application code analysis	There was no application code provided in this workload.	There was no application code provided in this workload.
Physical object analysis	After analyzing of your physical objects we found that 100% can be moved to EC2.	After analyzing of your physical objects we found that 100% can be moved to EC2.
Summary of Migration Effort	Based on the above summary and detail, we estimate that this conversion and migration will take 9292 days to complete with a 16 person team. You can get more details on this and change required inputs to look at different effort scenarios in the 'Migration Efforts to MySQL' tab.	Based on the above summary and detail, we estimate that this conversion and migration will take 3233 days to complete with a 16 person team. You can get more details on this and change required inputs to look at different effort scenarios in the 'Migration Efforts to PG' tab.

aws

Best Practices SCT Oracle to PostgreSQL

- ❑ Use Lower Case for DB Objects Names
- ❑ Boolean Column Type
- ❑ Date Column Type
- ❑ Number Column Type
- ❑ Sequences Handling
- ❑ Sequence Cache
- ❑ Inserting Null Value
- ❑ Partitioning

aws

Migration Playbooks for code conversion

SQL & PL/SQL (Manual)

	Oracle Feature	Aurora PostgreSQL Feature	Compatibility
Link	Anonymous Block	Do	Yes
Link	Execute Immediate	Execute & Prepare	Yes
Link	DBMS_RANDOM	random()	Yes*
Link	DBMS_OUTPUT	RAISE	Yes
Link	Procedures & Functions	Functions	Yes*
Link	User Defined Functions (UDFs)	Functions	Yes*
Link	UTL_FILE	N/A	None
Link	JSON Document Support	JSON Document Support	Yes*
Link	OLAP Functions	Window Functions	Yes
Link	PL/SQL Cursors	Cursors	Yes
Link	Single Row & Aggregate Functions	Single Row & Aggregate Functions	Yes
Link	Merge	SQL Merge	Yes
Link	Create Table As Select (CTAS)	Create Table As Select (CTAS)	Yes
Link	Common Table Expression (CTE)	Common Table Expression (CTE)	Yes
Link	Insert From Select	Insert From Select	Yes
Link	Inline Views	Inline Views	Yes
Link	DB Hints	Query Planning	Yes*

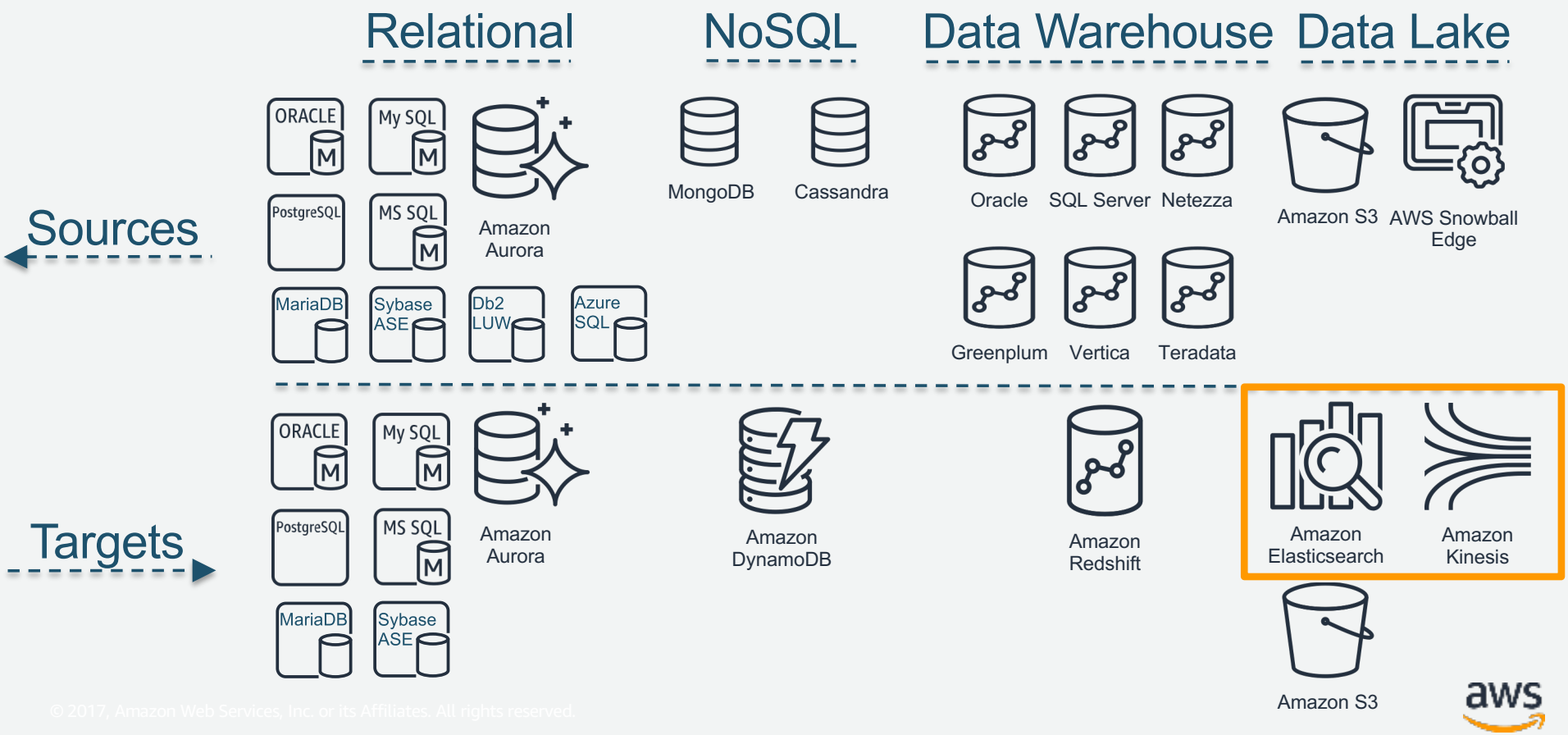
aws

Agenda

- ❑ How can I get to the Cloud
- ❑ AWS Schema Conversion Tool [SCT]
- ❑ AWS SCT – Best Practices
- ❑ AWS Database Migration Service [DMS]
- ❑ AWS DMS – Components
- ❑ AWS DMS – Best Practices
- ❑ Oracle – As Source
- ❑ Postgres – As Target

aws

When to use DMS?



AWS DMS Product Highlights

Pre-migration Assessment

- ❑ Checks migration task settings prior to launch
- ❑ Alerts for potential issues, such as unsupported data types
- ❑ Prevent unnecessary, time consuming runtime migration failures

Data Validation

- ❑ Validates that all data selected for migration migrated properly
- ❑ Includes both stages of full load and CDC
- ❑ Applies for both homogeneous and heterogeneous migrations

Snowball Integration

Use AWS Snowball & AWS DMS to migrate data to AWS

- ❑ Migrate large databases (over 5TB)
- ❑ Migrate many databases at once
- ❑ Migrate over slow network
- ❑ Push vs. Pull

aws

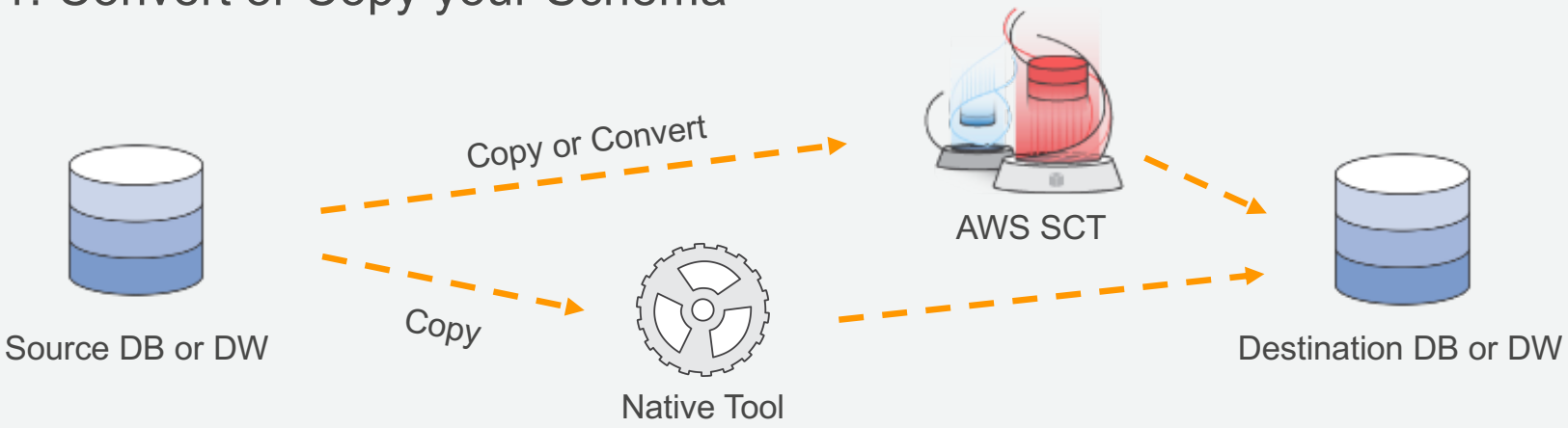
DMS + Snowball

Common use cases :

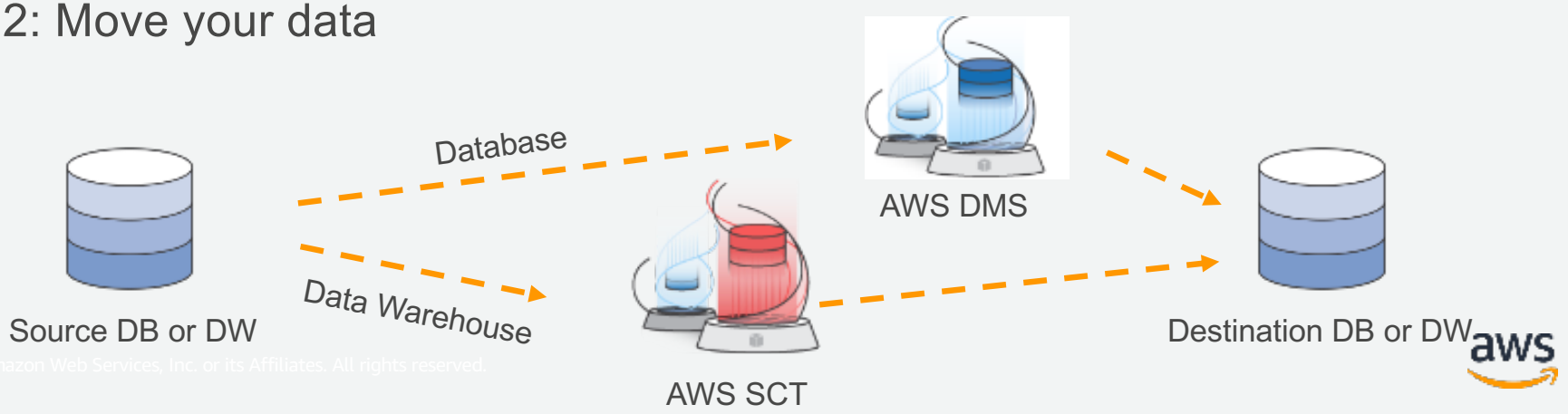
- ❑ Migrate large databases
- ❑ Migrate many databases at once
- ❑ Migrate over slow network
- ❑ Push vs. Pull

Database migration process

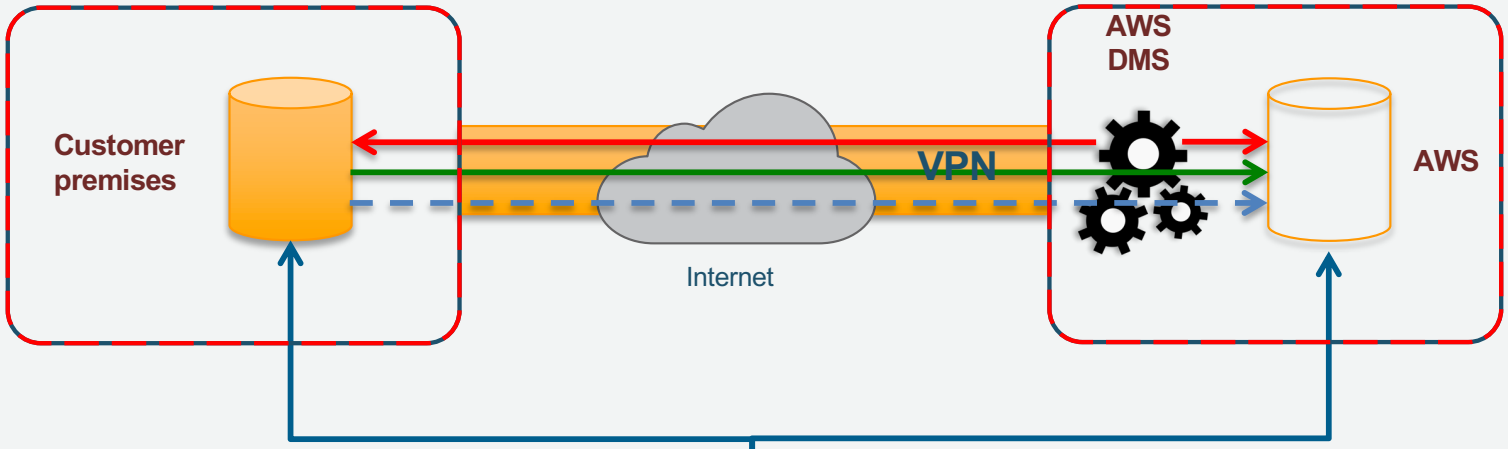
Step 1: Convert or Copy your Schema



Step 2: Move your data



Keep your apps running during the migration

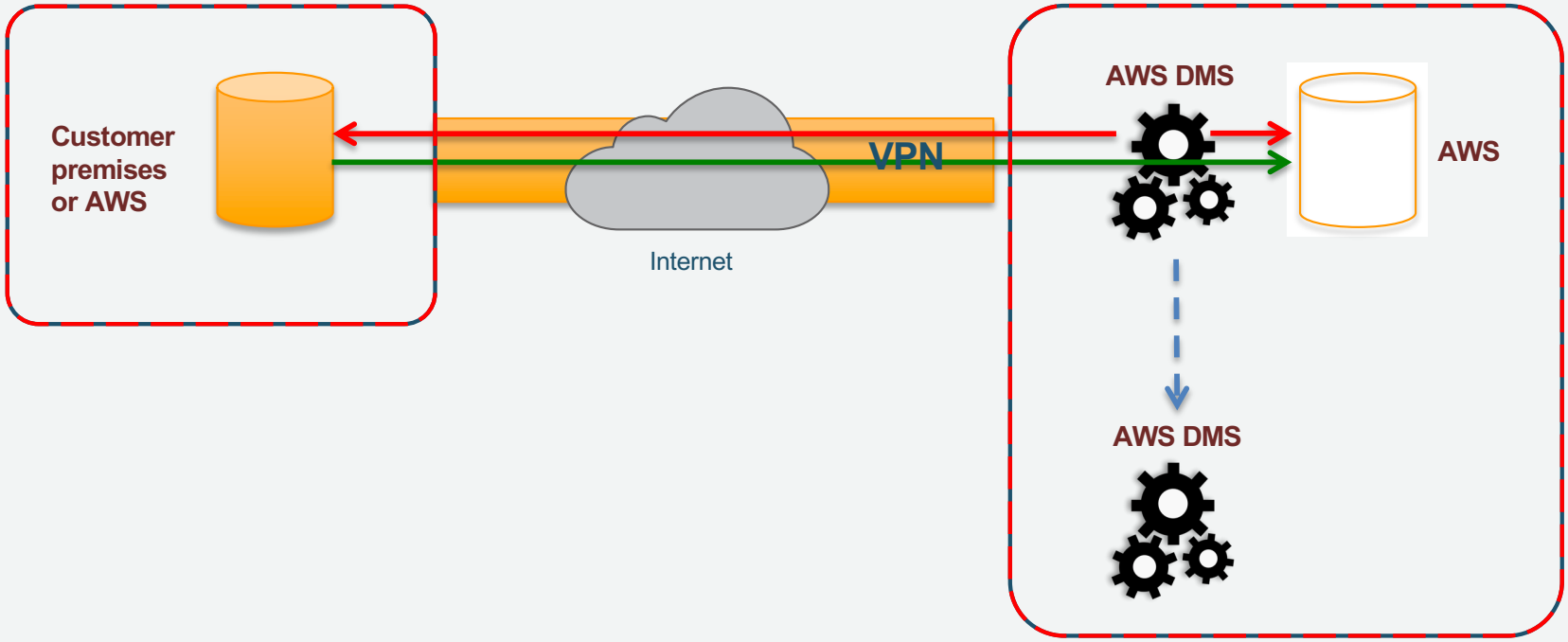


- ❑ Start a replication instance
- ❑ Connect to source and target databases
- ❑ Select tables, schemas, or databases

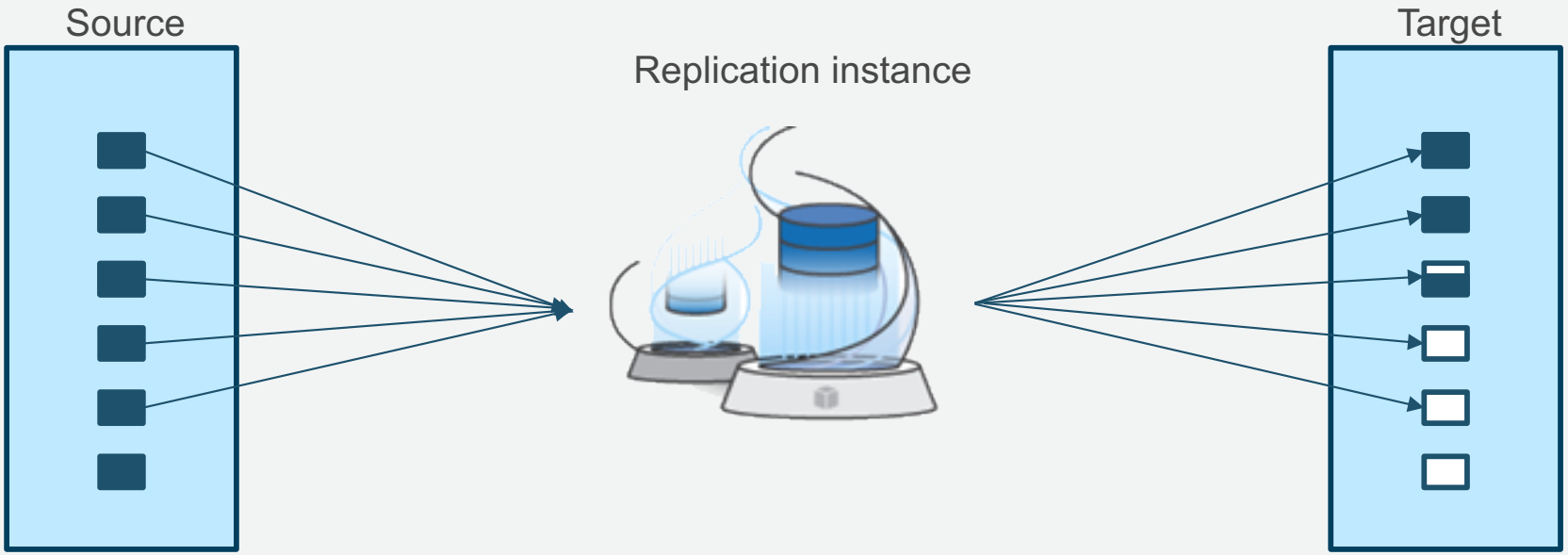
Application users

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

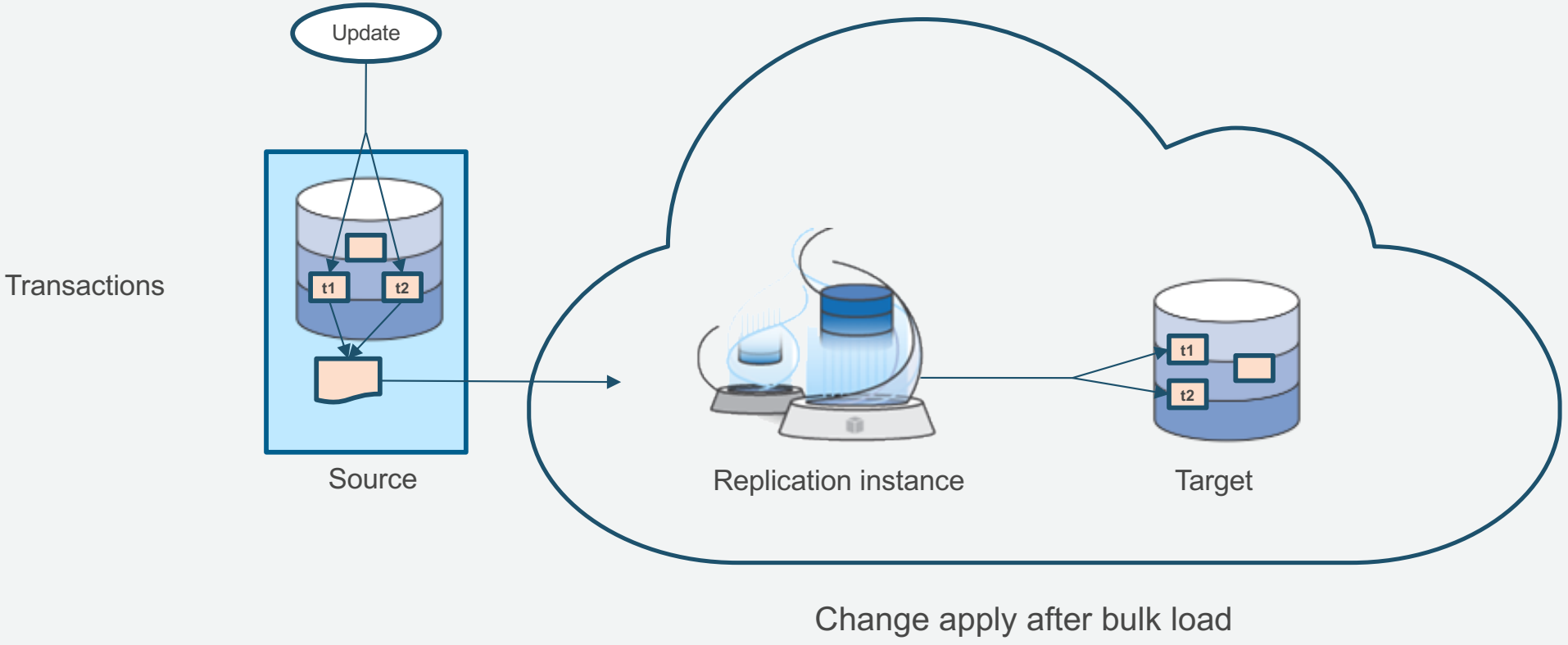
Multi-AZ option for high availability



Load is table by table

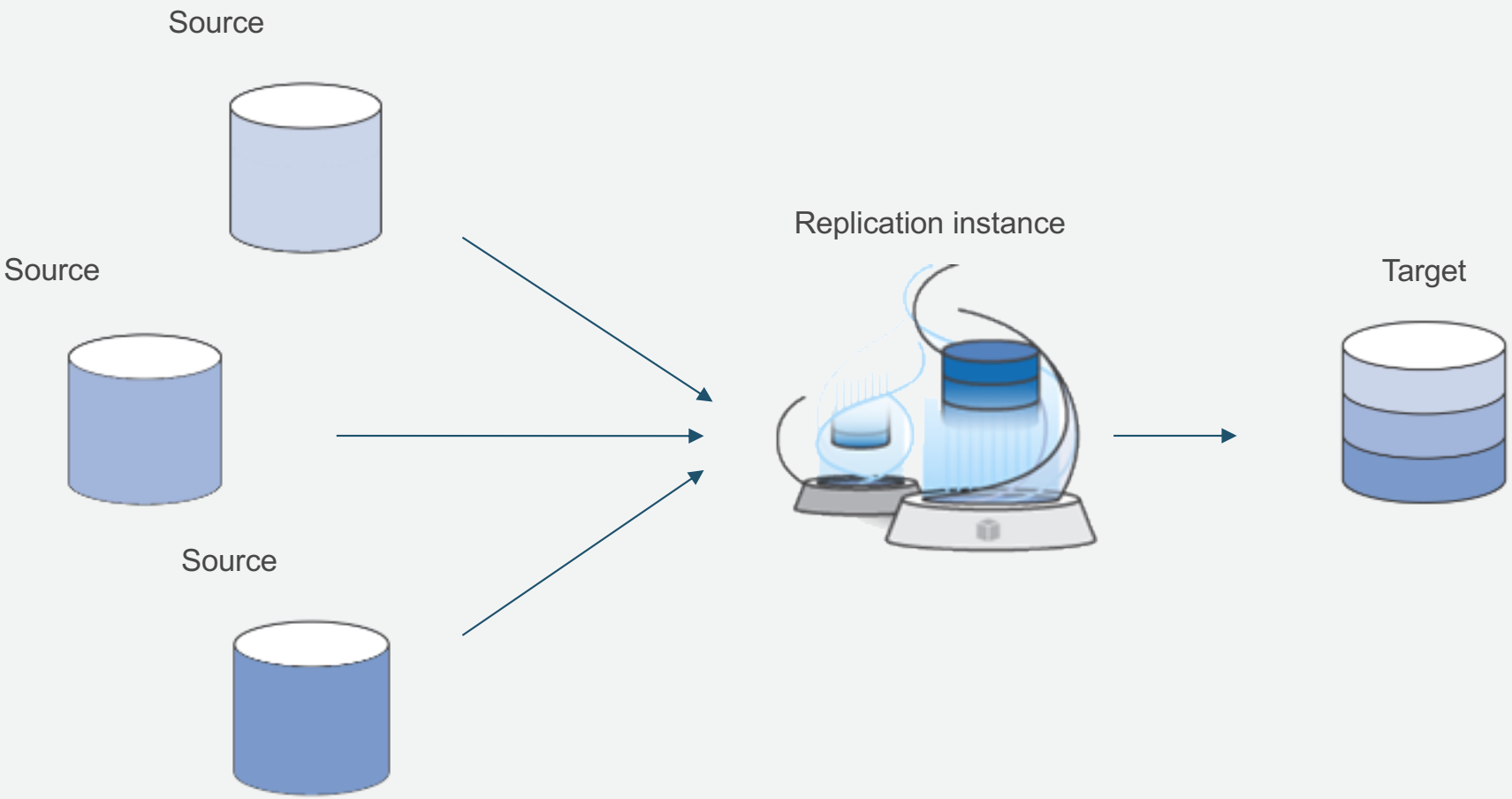


Change data capture (CDC) and apply

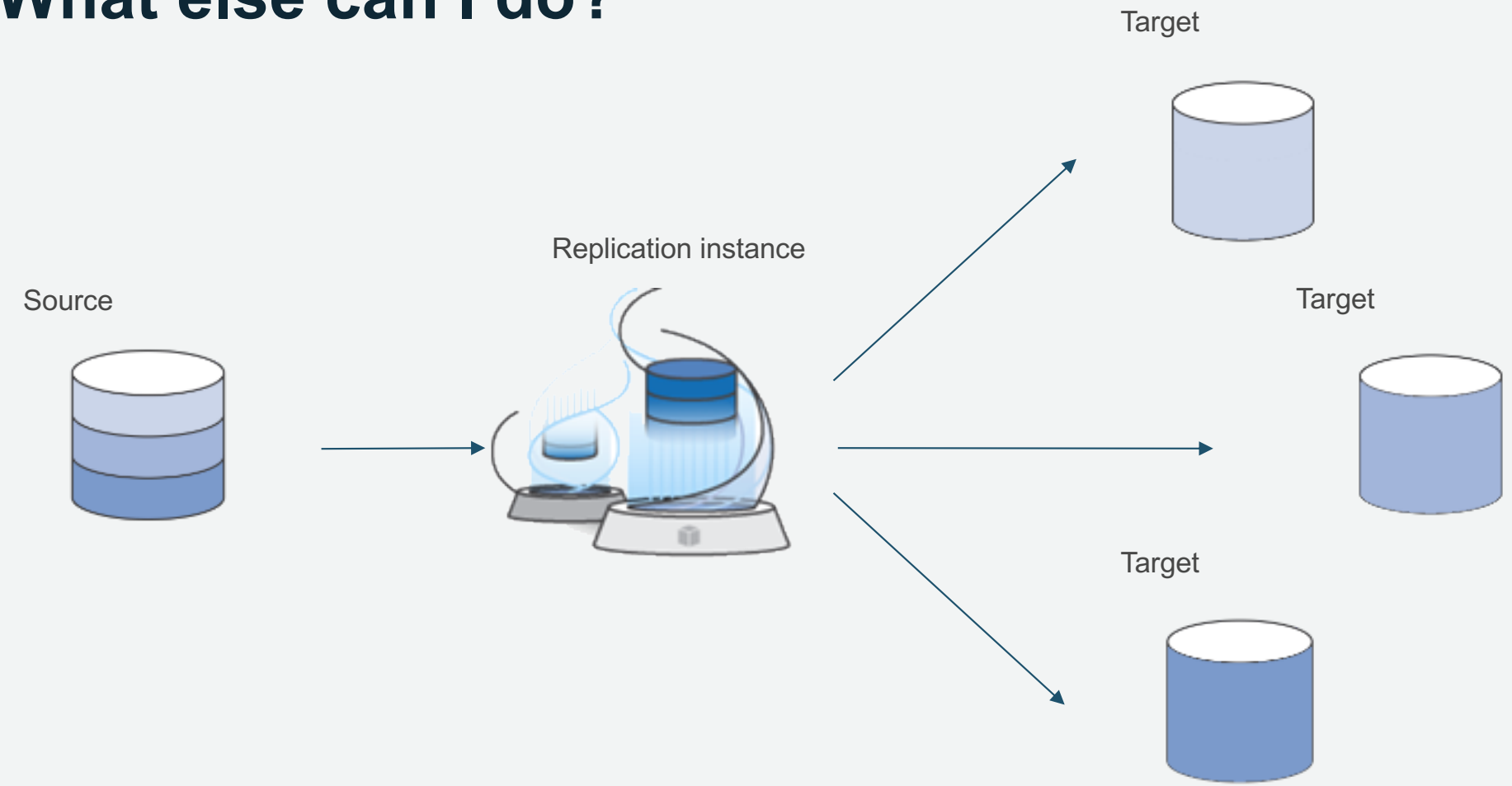


Change apply after bulk load

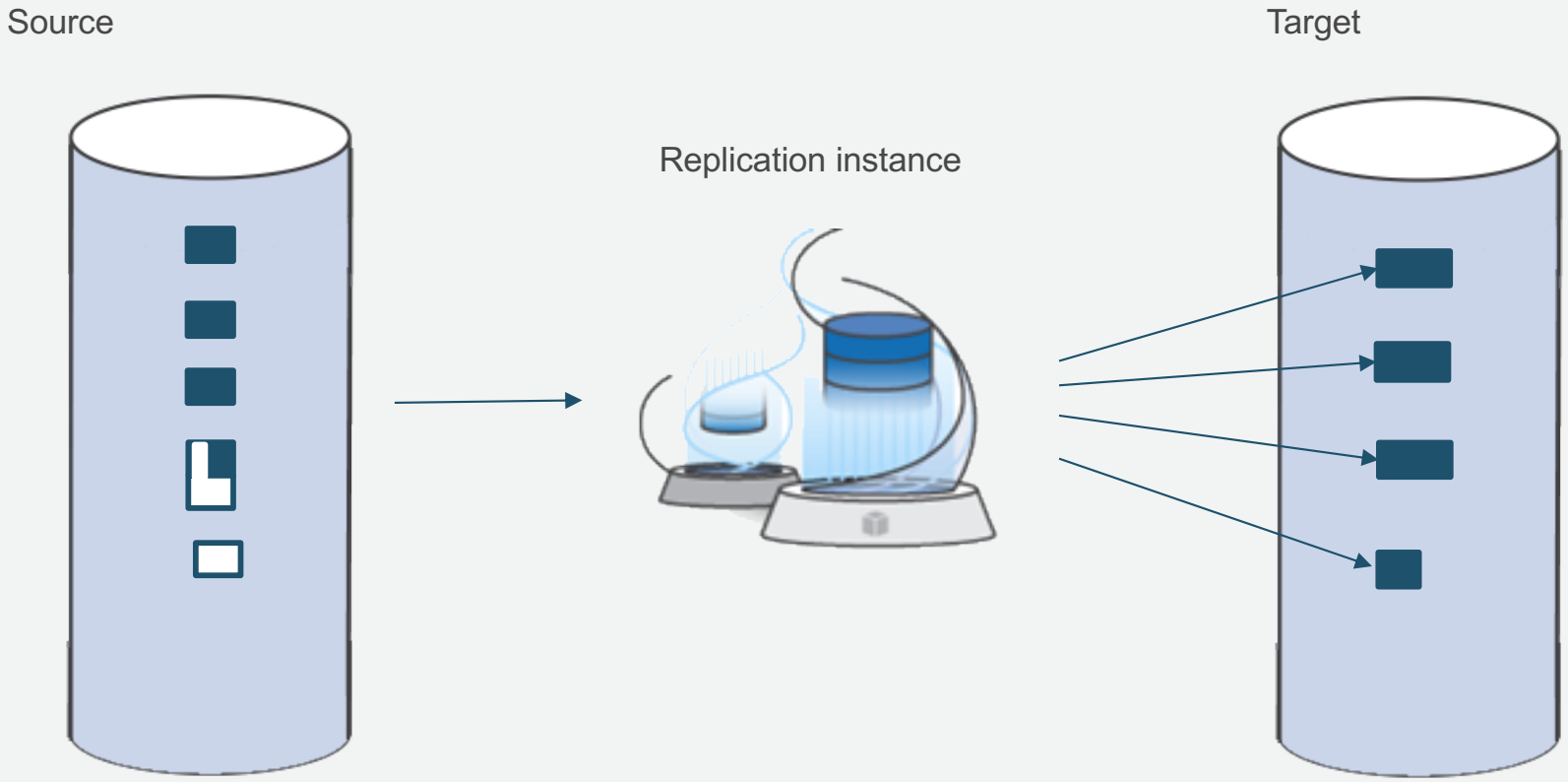
What else can I do?



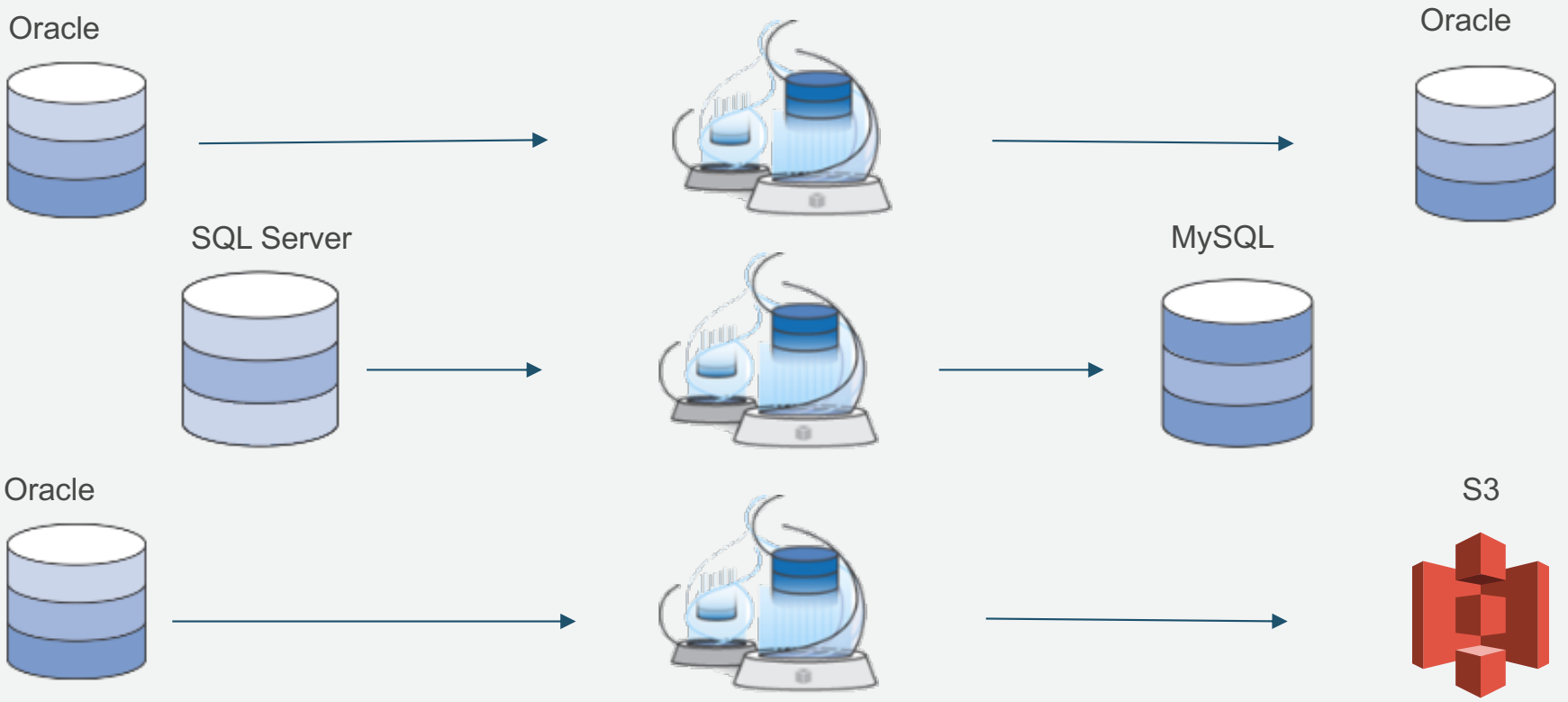
What else can I do?



Take it all—or not



Homogenous or heterogeneous



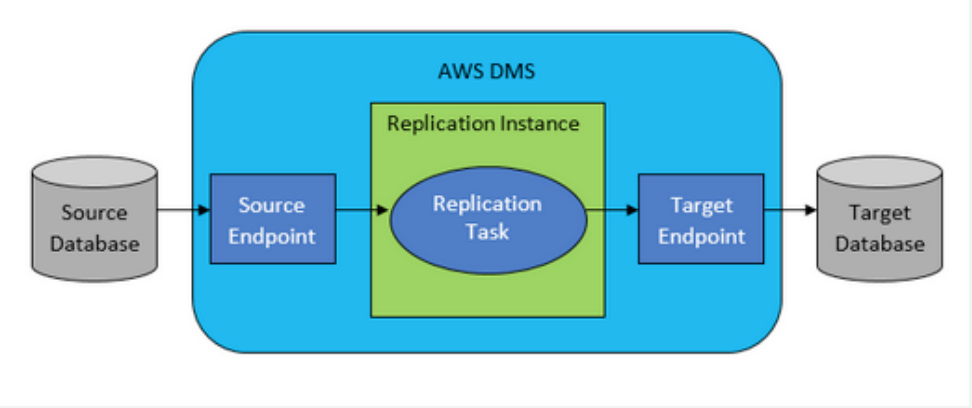
Agenda

- ❑ How can I get to the Cloud
- ❑ AWS Schema Conversion Tool [SCT]
- ❑ AWS SCT – Best Practices
- ❑ AWS Database Migration Service [DMS]
- ❑ AWS DMS – Components
- ❑ AWS DMS – Best Practices
- ❑ Oracle – As Source
- ❑ Postgres – As Target

aws

Three Main Components.

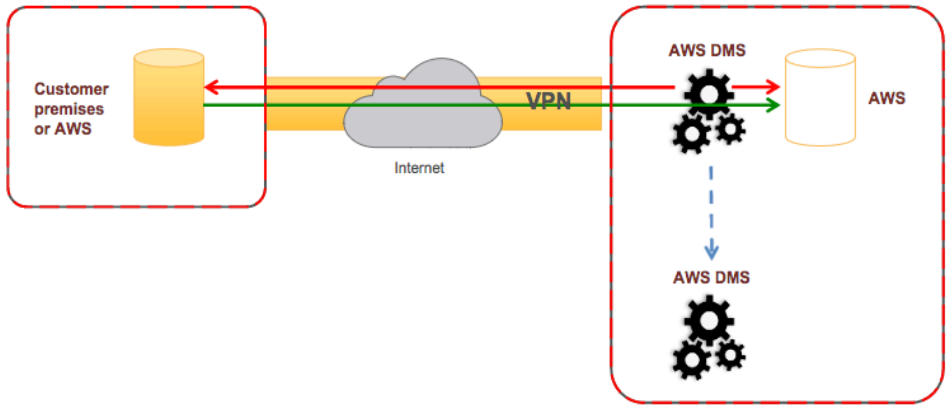
1. Replication Instances
2. Endpoints
3. Replication Tasks



You may need proper IAM permissions and roles assigned to access DMS service.

Replication Instance

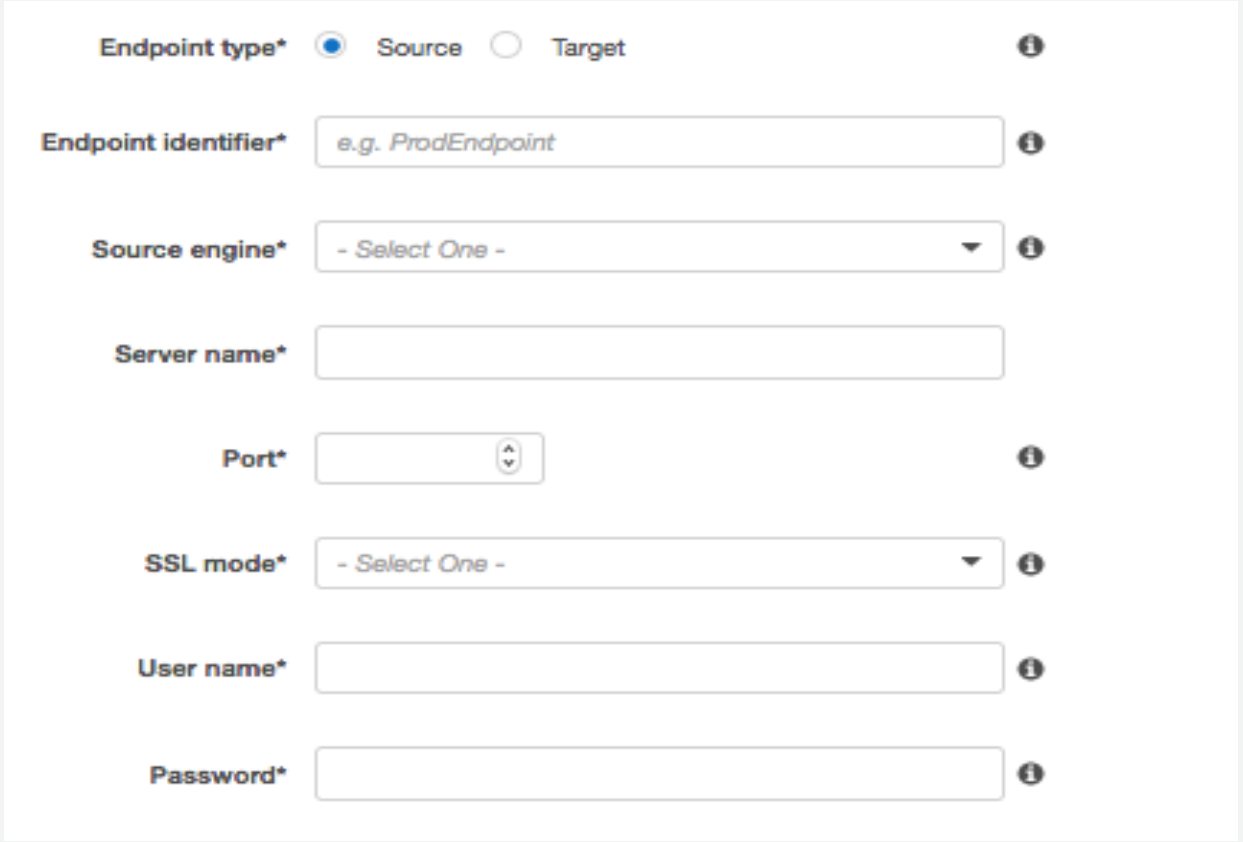
- Replication Instance Class
  - C4/R4 - Instance Performance
  - Number of Task/Tables
  - CPU Usage
- Storage
  - Default 50 or 100 GB data
- Multi-AZ
  - Useful during long term replication.



Source Endpoint

1. In simple word, Its way to connect to your source data base.

2. Make sure you are able to connect to your Source Database.

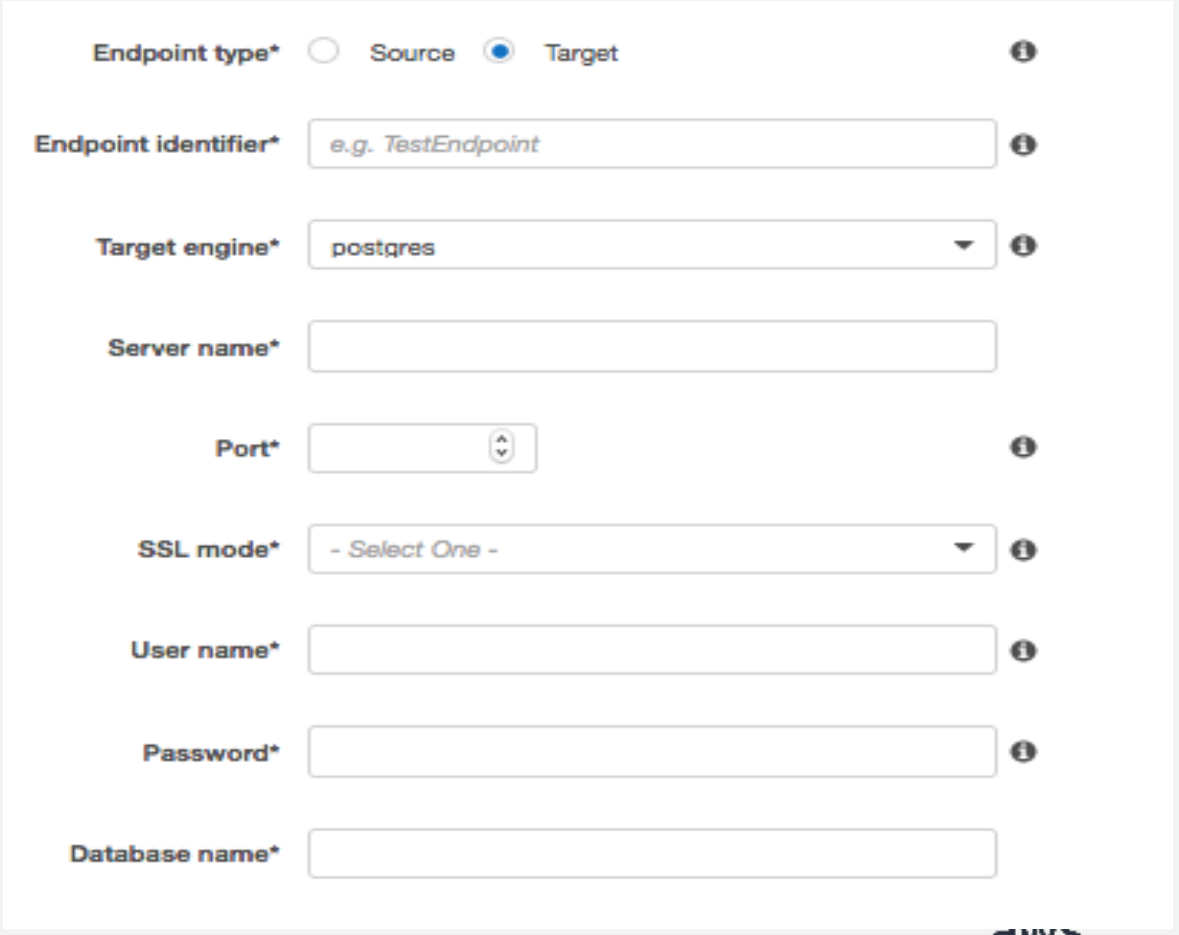


aws

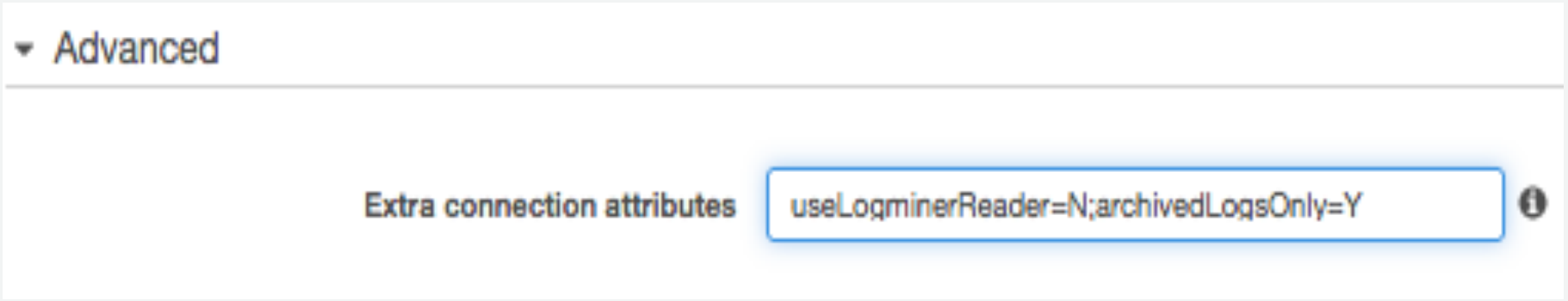
Target Endpoint

1. In simple word, Its way to connect to your target data base.

2. Make sure you are able to connect to your Target Database.



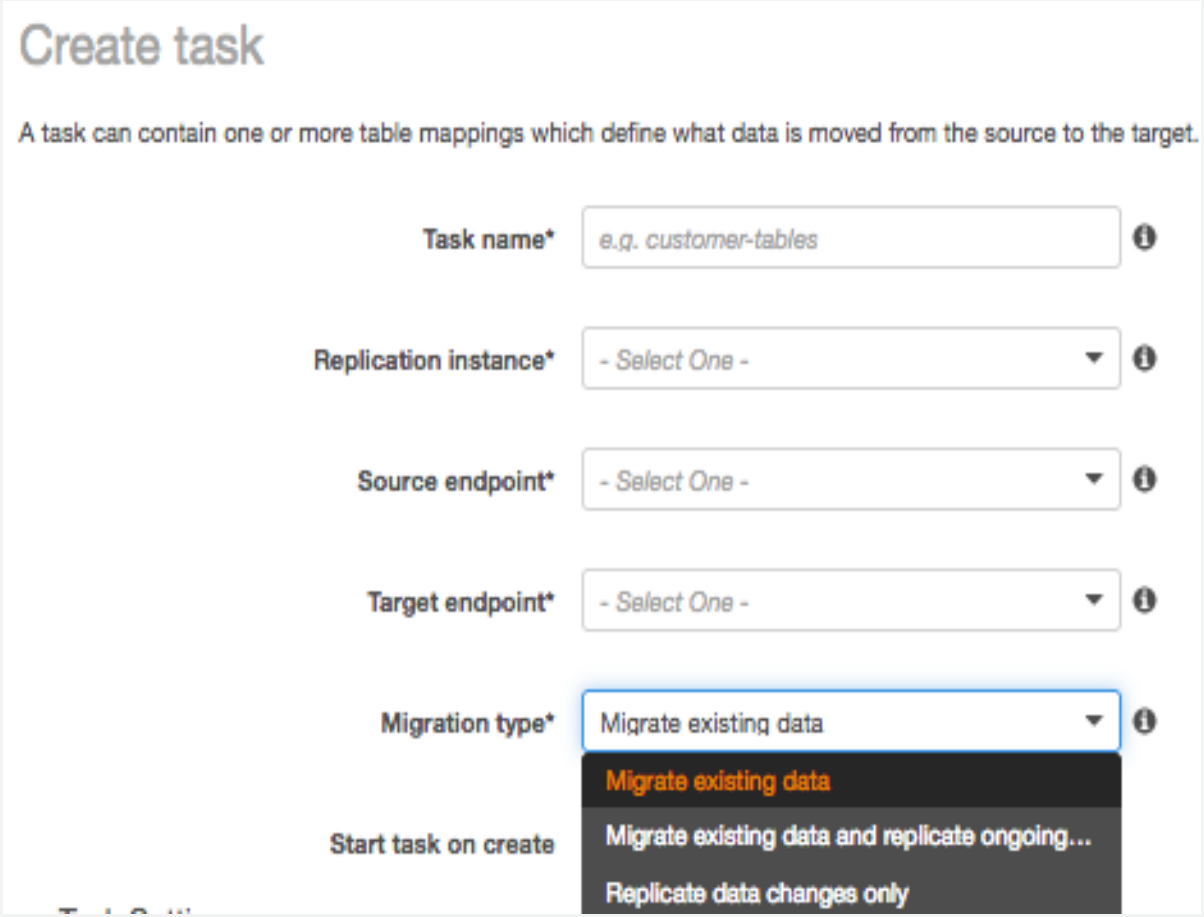
Extra Connection Attributes (ECA)



DMS Task

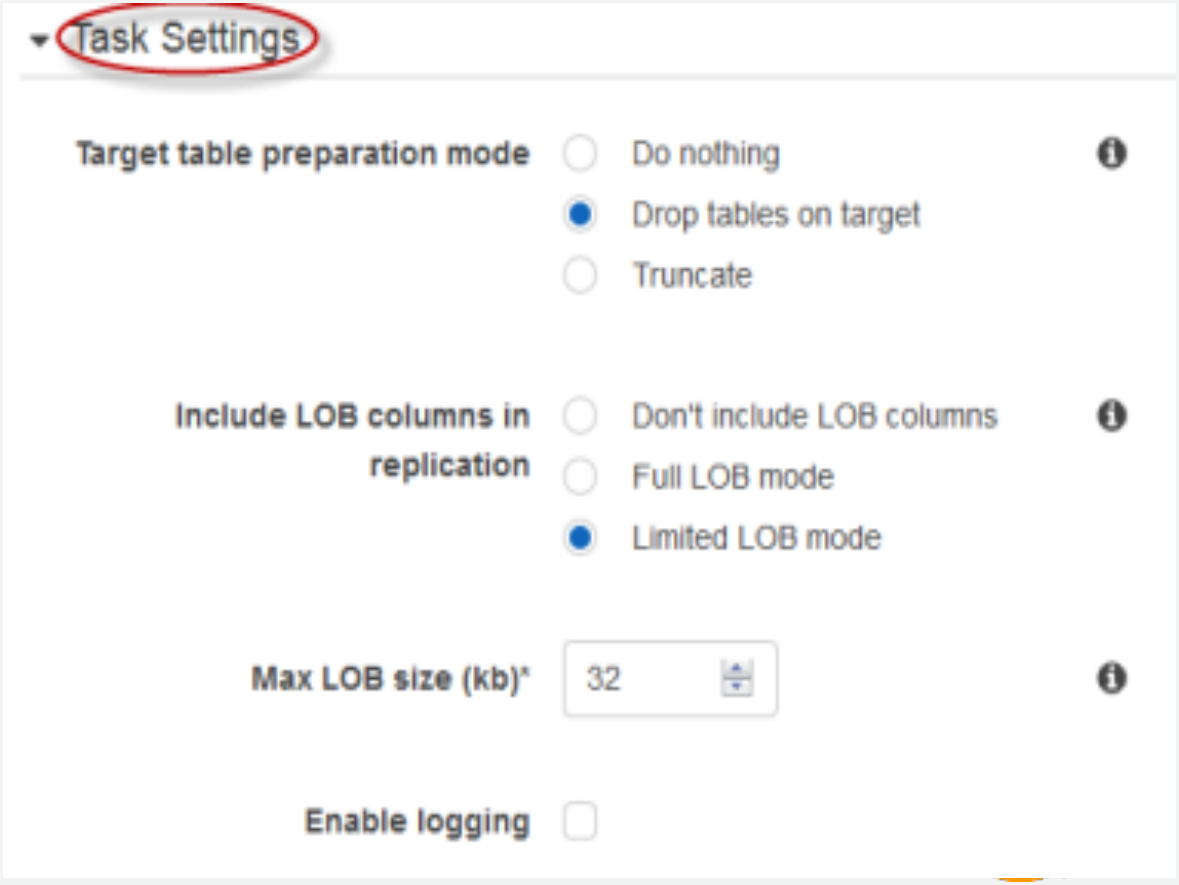
Migration Type:

1. Migrate existing data
2. Migrate existing data and changes
3. Replicate data changes only



DMS Task Settings

1. Are you manually creating Target Tables ?
2. Do you have LOBs in your DB ?
3. Do you know max size of your LOBs ?
4. Don’t forget to enable Logging.



Agenda

- ❑ How can I get to the Cloud
- ❑ AWS Schema Conversion Tool [SCT]
- ❑ AWS SCT – Best Practices
- ❑ AWS Database Migration Service [DMS]
- ❑ AWS DMS – Components
- ❑ AWS DMS – Best Practices
- ❑ Oracle – As Source
- ❑ Postgres – As Target

aws

Few Thumb Rules.. Optimizing Full Load/CDC

Full Load

- ❑ Load multiple Tables in Parallel
- ❑ Remove bottlenecks on the target
- ❑ Use Multiple Tasks
- ❑ Use Proper Replication Instance or Multiple Replication Servers
- ❑ Improve LOB performance
- ❑ Use Row filters.

On going Replication

By default, AWS DMS processes changes in a transactional mode

- ❑ If you can afford temporary lapses in transactional integrity, you can use the batch optimized apply
- ❑ Batch Apply Mode groups transactions and applies them in batches
- ❑ Use proper Indexes on target during CDC

aws

Migrate LOBs

Limited LOB Mode:

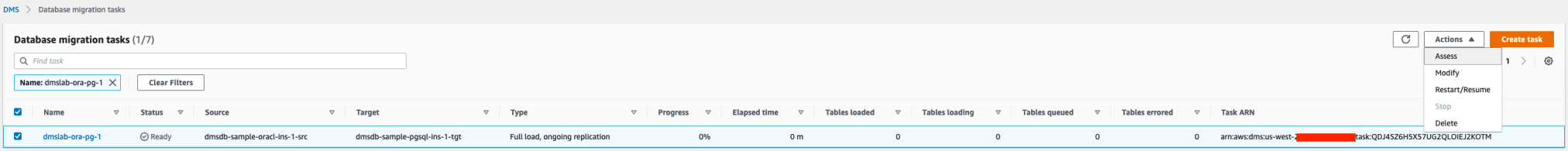
- ❑ DMS will pre-allocate memory and migrate LOB in bulk.
- ❑ Fast and efficient.
- ❑ Query your source data dictionary and find largest LOB size.
- ❑ LOB Column size should be less than **Max_Lob_Size.**

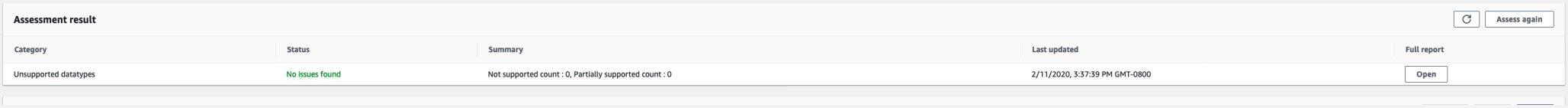
Full LOB Mode:

- ❑ DMS have no information about Source LOB Size.
- ❑ LOB will be migrated in pieces by **LOB Chunk Size**. (64K)
- ❑ LOB chunk size can be modified if need be

aws

DMS Preflight check





DMS Preflight check, Sample Report

Summary
Not supported datatypes : json
Partially supported datatypes : enum, geometry, geometrycollection, linestring, longtext, mediumtext, multilinestring, multipoint, multipolygon, point, polygon, set, tinytext

Tables with partially supported datatype name : enum

Schema : ms
Table (Columns) : T_ENUM (a)

Tables with partially supported datatype name : geometry

Schema : ms
Table (Columns) : T_GEOMETRY (a)

Tables with partially supported datatype name : geometrycollection

Schema : ms
Table (Columns) : T_GEOMETRYCOLLECTION (a)

Tables with not supported datatype name : json

aws

Validations Before you start Data Migration

- ❑ Use Schema Compare Tool for comparing your Source/Target Schema.
- ❑ Check for any long running transactions on the source database.
- ❑ FK are disabled on target tables during Full Load task.
- ❑ Triggers are Disabled during DMS data replication.
- ❑ Check the owner of all the objects in application schema. As schema and object owner are two different entities in Postgres, we need to validate this in order to have right permission for DMS user to insert the data.
- ❑ All the tables in Postgres should have Primary Keys in order to replicate data to Oracle.

DMS Task Validation

- Data validation is optional.
- To enable data validation, set the **EnableValidation** setting to true.
- To adjust the number of execution threads that AWS DMS uses during validation, set the **ThreadCount** value.
- The default value for **ThreadCount** is **5**.
- DMS compares each row in the source with its corresponding row at the target

Data Validation Troubleshooting

AWS DMS creates a new table at the target endpoint: awsdms_validation_failures_v1.

Following is a description of the awsdms_validation_failures_v1 table:

Colum Name	Data Type	Description
TASK_NAME	VARCHAR(128) NOT NULL	AWS DMS task identifier.
TABLE_OWNER	VARCHAR(128) NOT NULL	Schema (owner) of the table.
TABLE_NAME	VARCHAR(128) NOT NULL	Table name.
FAILURE_TIME	DATETIME(3) NOT NULL	Time when the failure occurred.
KEY	TEXT NOT NULL	This is the primary key for row record type.
FAILURE_TYPE	VARCHAR(128) NOT NULL	Severity of validation error. Can be either `Failed` or `Suspended`.

```
select * from awsdms_validation_failures where TASK_NAME = 'VFPFKH4FJR3FTYKK2RYSI'
```

Data Validation Limitations

- Requires table to have a primary key or unique index.
- Primary key columns can't be of type CLOB, BLOB, or BYTE.
- Not supported if migration uses customized filtering or when consolidating several databases into one.
- Not supported if the target database is modified outside of DMS
- If there are more than 10,000 failed or suspended records, validation will stop
- Data validation generates additional queries against the source and target databases.
  - **Note: If the sort order is different between PostgreSQL and Oracle, data validation fails to validate the records.**

aws

Important DMS Parameters

- **StopTaskCachedChangesNotApplied**: Set this option to true. Please add secondary indexes after full load completes.
- **StopTaskCachedChangesApplied**: Set this option to true to stop a task after a full load completes and cached changes are applied.
- **MaxFullLoadSubTasks** : Indicates maximum number of tables to load in parallel. If needed you should reduce the value of this parameter not to put too much of strain on the PRIMARY.
- **TransactionConsistencyTimeout** : To set the number of seconds that AWS DMS waits for transactions to close before beginning a full-load operation, if transactions are open when the task starts.

Extra Connection Attributes

- **maxFileSize**: DMS creates csv files on the DMS instance which it uses to transfer data with copy command to target DB.
- **useDirectPathFullLoad**: For reverse replication of data from Postgres to Oracle use useDirectPathFullLoad=N in case of IOT tables.
- **captureDDLs**: For reverse replication you can use captureDDLs=N while creating source endpoint to Postgres. It will make sure that no DDLs are performed on the table once CDC is started till the end of migration.

aws

Agenda

- ❑ How can I get to the Cloud
- ❑ AWS Schema Conversion Tool [SCT]
- ❑ AWS SCT – Best Practices
- ❑ AWS Database Migration Service [DMS]
- ❑ AWS DMS – Components
- ❑ AWS DMS – Best Practices
- ❑ Oracle – As Source
- ❑ Postgres – As Target

aws

Requirements: Oracle Source

1. Oracle Client installed
2. ARCHIVELOG ON
3. Minimum Supplemental logging on database level
4. Security permissions for the user that DMS uses to connect (sys.obj$)

Table Level

1. Permissions (e.g. SELECT On TABLE)
2. supplemental logging

aws

Prerequisites - ARCHIVELOG

Database to be setup with ARCHIVELOG mode. In this mode Oracle will archive the redo logs.

When working without the ARCHIVELOG mode, the redo logs will be overwritten and history of changes are lost.

To check for ARCHIVELOG mode:

```
SQL> SELECT log_mode FROM v$database;

LOG_MODE
------------
NOARCHIVELOG
```

→ **CDC will not function properly**

The expected result should be:

```
LOG_MODE
------------
ARCHIVELOG
```


Methods for reading the redo logs

❑ LogMiner

- ➢ Standard Oracle component, it provides the means to query the logged changes of the Oracle Database
- ➢ Default option
- ➢ Easy to configure

❑ Binary Reader

- ➢ AWS DMS feature that reads and parses the raw redo log files directly.
- ➢ DMS setup must create a temporary directory to configure Binary Reader for change processing. **(USE_DB_RECOVERY_FILE_DEST.)**
- ➢ Improved performance and reduced load on the Oracle server when compared with LogMiner.
- ➢ Binary Reader supports CDC for LOBs in Oracle version 12c.

aws

Migrating from Oracle ASM

- **Using Log miner**
  - transparent to DMS, no special configuration
- **Using binary reader**
  - Create a temporary directory to configure Binary Reader for change processing.
  - DMS needs to connect to the ASM instance.
  - The ASM instance should also accept traffic from the DMS replication instance
  - Current API call (DBMS_DISKGROUP) is very slow (packets up to 32 KB)
  - Recommend using Binary Reader with the DMS copy functionality.
    - **Copy to temp folder**– Copy the REDO to a OS folder, and then read with Bfile/Directly
    - If you have multiple tasks – you need to create separate subfolders

[How to Migrate from Oracle ASM to AWS using AWS DMS]

aws

ECA Example – BFILE with +ASM

- Extra Connection attributes to read from ASM in parallel
  - parallelASMReadThreads
  - asmUsePLSQLArray (3.3.x)

- Example with TDE:
  - useLogminerReader=N;useBfile=Y;asm_user=asmuser;asm_server=<IP>:<Port>/+ASM;
  - parallelASMReadThreads=6;readAheadBlocks=150000;
  - archivedLogDestId=1; archivedLogsOnly=Y;
    securityDbEncryptionName=ORACLE.SECURITY.DB.ENCRYPTION.AZEO3RS9L0+XXXXX+XXX+;
  - securityDbEncryption=XXXXXXXXXXXXXXXXXXXXXXX+XXXXXXXXXXXXXXX

aws

Using the standby

- DMS only supports Oracle Active Dataguard Standby as source.
- Find the delay between Primary and Standby
- Set this ECA on the source Oracle endpoint
  - **standbyDelayTime=600**
- As of now DMS only supports two destination Ids
- Once you have identified the dest_id, set the following ECA on the source Oracle endpoint
  - **archivedLogDestId=1;additionalArchivedLogDestId=2;**

aws

Partitions

- DMS supports CDC on tables with partitions and sub-partitions ,but there are some limitations.
- **Limitations:** support for partition operations (ADD, DROP, TRUNCATE, MOVE and EXCHANGE) is limited
  - To replicate such changes, reload the table being replicated.
    - **ADD** - Supported.
    - **DROP** - DMS will not delete the data of the dropped partition from target
    - **TRUNCATE** – Not supported for partitions or sub-partitions .
    - **MOVE** - Supported only If source table is not in task scope, and target table is in task scope. In this case, DMS will not move the existing data from the moved partition, but later operations on the moved partition will be captured.
    - **EXCHANGE** - not supported.

aws

Agenda

- ❑ How can I get to the Cloud
- ❑ AWS Schema Conversion Tool [SCT]
- ❑ AWS SCT – Best Practices
- ❑ AWS Database Migration Service [DMS]
- ❑ AWS DMS – Components
- ❑ AWS DMS – Best Practices
- ❑ Oracle – As Source
- ❑ Postgres – As Target

aws

Aurora Postgres Learnings

- ❑ Control on Query Plan Stability
- ❑ Non-declarative Partitioning(PG 10+)
- ❑ Benefit/overhead of Vacuum
- ❑ Avoid Temporary table usage
- ❑ Beware of save-point transactions

aws

Resources available to customers

Resources available to customers - DMS

Getting Started Guide: Review technical documentation.

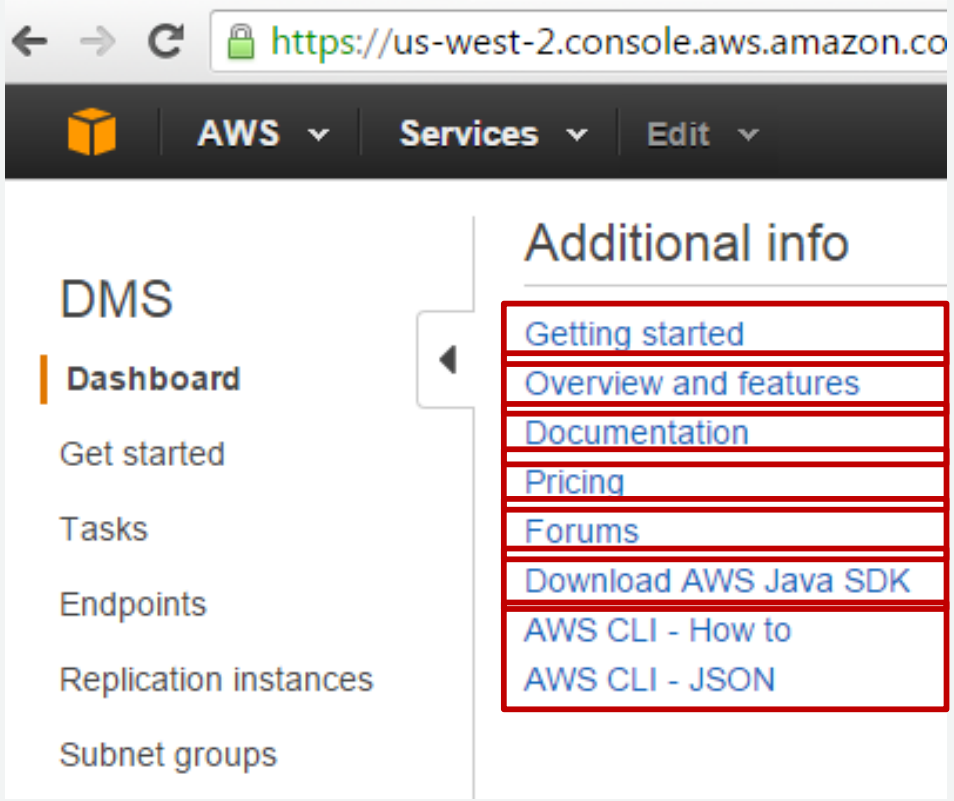
Features and benefits: Highlights DMS features.

Pricing: Prices for replication instances, storage, and data transfer.

Support: Post your questions to our Support forum.

AWS SDK: Java-based API for creating and managing data migration tasks.

AWS Command Line Interface: Start and stop replication tasks with simple commands.



Resources available to customers - SCT

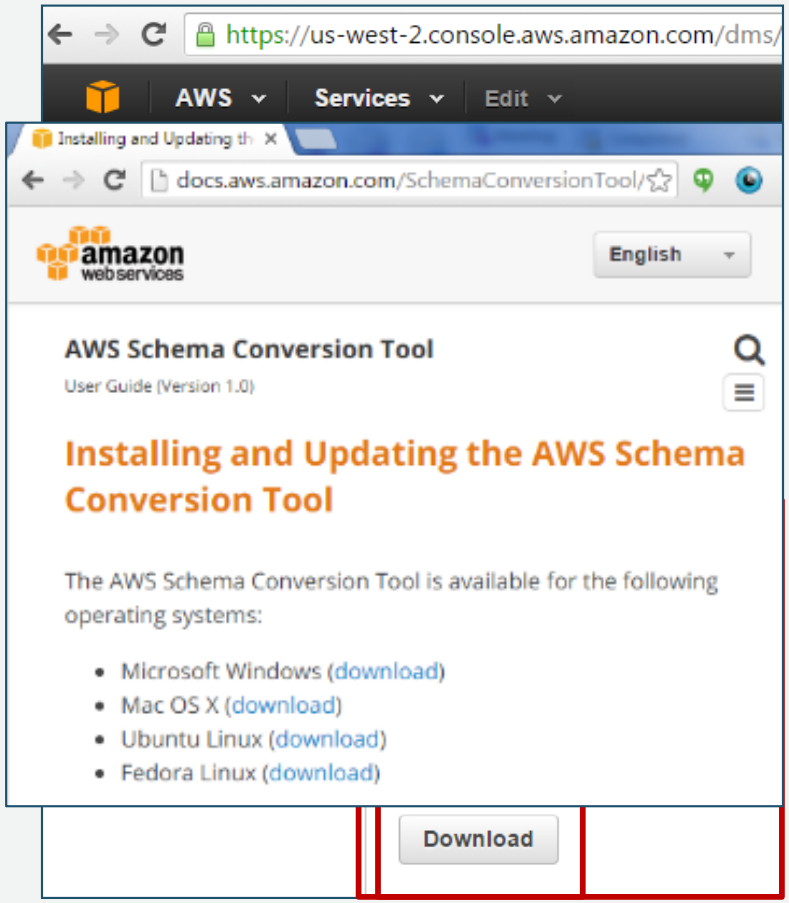
User Guide: Review technical docs at aws.amazon.com/documentation/SchemaConversionTool/

or choose the **Download** button.

Download area: Get installation files for the Schema Conversion Tool.

Support forums: Ask questions and review how-to guides.

https://forums.aws.amazon.com/forum.jspa?forumID=208.



New Content

Blogs

- How to Migrate Your Oracle Database to PostgreSQL
- Migrating Oracle Database from On-Premises or Amazon EC2 Instances to Amazon Redshift
- Using the AWS Database Migration Service, Amazon S3, and AWS Lambda for Database Analytics
- How to Migrate Your Oracle Data Warehouse to Amazon Redshift Using AWS SCT and AWS DMS
- Reduce Resource Consumption by Consolidating Your Sharded System into Aurora
- Set Up AWS DMS for Cross-Account Migration

Webinars

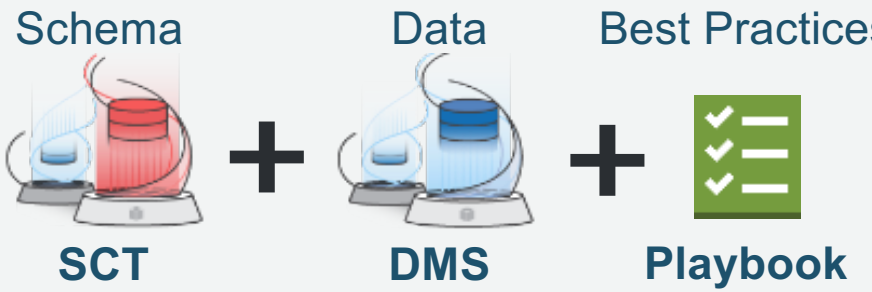
- "Consolidate MySQL Shards Into Amazon Aurora Using AWS Database Migration Service"
- "Migrate from SQL Server or Oracle into Amazon Aurora using AWS Database Migration Service"

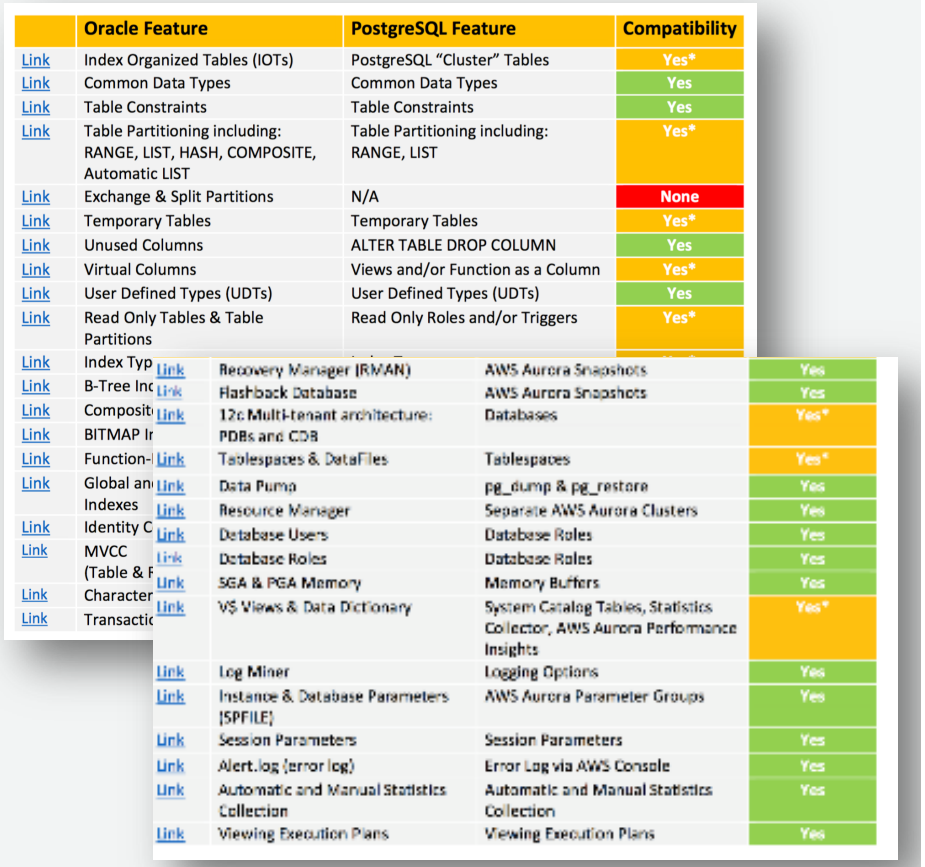
Github repository: https://github.com/awslabs/aws-database-migration-tools
https://github.com/awslabs/aws-database-migration-samples

aws

Old World to AWS Migration Playbooks

- Topic-by-topic overview of how to migrate databases and data warehouses to AWS services
- Covers all proprietary features and the different database objects
- Migration best practices
- Oracle to Aurora PostgreSQL – **available**
- SQL Server to Aurora MySQL – **available**
- SQL Server to Aurora PostgreSQL – **available**
- Oracle to Aurora MySQL, Cassandra to DynamoDB, Data warehouses to Redshift, and Db2 LUW to Aurora PostgreSQL – Q1





	Oracle Feature	PostgreSQL Feature	Compatibility
Link	Index Organized Tables (IOTs)	PostgreSQL "Cluster" Tables	Yes*
Link	Common Data Types	Common Data Types	Yes
Link	Table Constraints	Table Constraints	Yes
Link	Table Partitioning including: RANGE, LIST, HASH, COMPOSITE, Automatic LIST	Table Partitioning including: RANGE, LIST	Yes*
Link	Exchange & Split Partitions	N/A	None
Link	Temporary Tables	Temporary Tables	Yes*
Link	Unused Columns	ALTER TABLE DROP COLUMN	Yes
Link	Virtual Columns	Views and/or Function as a Column	Yes*
Link	User Defined Types (UDTs)	User Defined Types (UDTs)	Yes
Link	Read Only Tables & Table Partitions	Read Only Roles and/or Triggers	Yes*
Link	Index Typ		
Link	B-Tree Inc		
Link	Composit		
Link	BITMAP Ir		
Link	Function-		
Link	Global an Indexes		
Link	Identity C		
Link	MVCC (Table & F		
Link	Character		
Link	Transactic		
Link	Recovery Manager (RMAN)	AWS Aurora Snapshots	Yes
Link	Flashback Database	AWS Aurora Snapshots	Yes
Link	12c Multi-tenant architecture: PDBs and CDB	Databases	Yes*
Link	Tablespaces & DataFiles	Tablespaces	Yes*
Link	Data Pump	pg_dump & pg_restore	Yes
Link	Resource Manager	Separate AWS Aurora Clusters	Yes
Link	Database Users	Database Roles	Yes
Link	Database Roles	Database Roles	Yes
Link	SGA & PGA Memory	Memory Buffers	Yes
Link	V$ Views & Data Dictionary	System Catalog Tables, Statistics Collector, AWS Aurora Performance Insights	Yes*
Link	Log Miner	Logging Options	Yes
Link	Instance & Database Parameters (SPFILE)	AWS Aurora Parameter Groups	Yes
Link	Session Parameters	Session Parameters	Yes
Link	Alert.log (error log)	Error Log via AWS Console	Yes
Link	Automatic and Manual Statistics Collection	Automatic and Manual Statistics Collection	Yes
Link	Viewing Execution Plans	Viewing Execution Plans	Yes

The fine print

Sources for AWS Database Migration Service

Customers 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 9.4–9.6
- ❑ SAP ASE 15.7+

RDS instance databases:

- ❑ Oracle Database 11g–12c
- ❑ Microsoft SQL Server 2008R2–2014
- ❑ MySQL versions 5.5–5.7
- ❑ MariaDB (MySQL-compatible data source)
- ❑ PostgreSQL 9.4–9.6
- ❑ Amazon Aurora (MySQL-compatible data source)
- ❑ Amazon Aurora (PostgreSQL-compatible data source) *No CDC

Targets for AWS Database Migration Service

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

On-premises and EC2 instance databases:

- ❑ Oracle Database 10g–12c
- ❑ Microsoft SQL Server 2005–2014
- ❑ MySQL 5.5–5.7
- ❑ MariaDB (MySQL-compatible data source)
- ❑ PostgreSQL 9.3–9.6
- ❑ SAP ASE 15.7+

RDS instance databases:

- ❑ Oracle Database 11g–12c
- ❑ Microsoft SQL Server 2008 R2 - 2014
- ❑ MySQL 5.5–5.7
- ❑ MariaDB (MySQL-compatible data source)
- ❑ PostgreSQL 9.3–9.6
- ❑ Amazon Aurora (MySQL and PostgreSQL -compatible data sources)
- ❑ Amazon Redshift

aws

AWS Database Migration service pricing

US East (N. Virginia)

Instance Type	Price per hour (Single-AZ)	Price per hour (Multi-AZ)
t2.micro	$0.018	$0.036
t2.small	$0.036	$0.072
t2.medium	$0.073	$0.146
t2.large	$0.146	$0.292
c4.large	$0.154	$0.308
c4.xlarge	$0.308	$0.616
c4.2xlarge	$0.617	$1.234
c4.4xlarge	$1.235	$2.470

T2/T3* for developing and periodic data migration tasks
C4 for large databases and minimizing time
R4*/R5* for tasks requiring more memory

T2 pricing starts at $0.018 per hour for T2.micro
C4 pricing starts at $0.154 per hour for C4.large
R4 pricing starts at $0.021 per hour for R4.large*

50 GB GP2 storage included with T2/T3* instances
100 GB GP2 storage included with C4, R4* & R5* instances

Data transfer inbound and within AZ is free
Data transfer across AZs starts at $0.01 per GB

*R4, T3 and R5 in limited regions

aws

Thank You!