AWS Database Migration Service

AWS

Agenda

Overview

DMS components

Database migration options

Migration use cases

Best practices

Questions

Overview

Overview

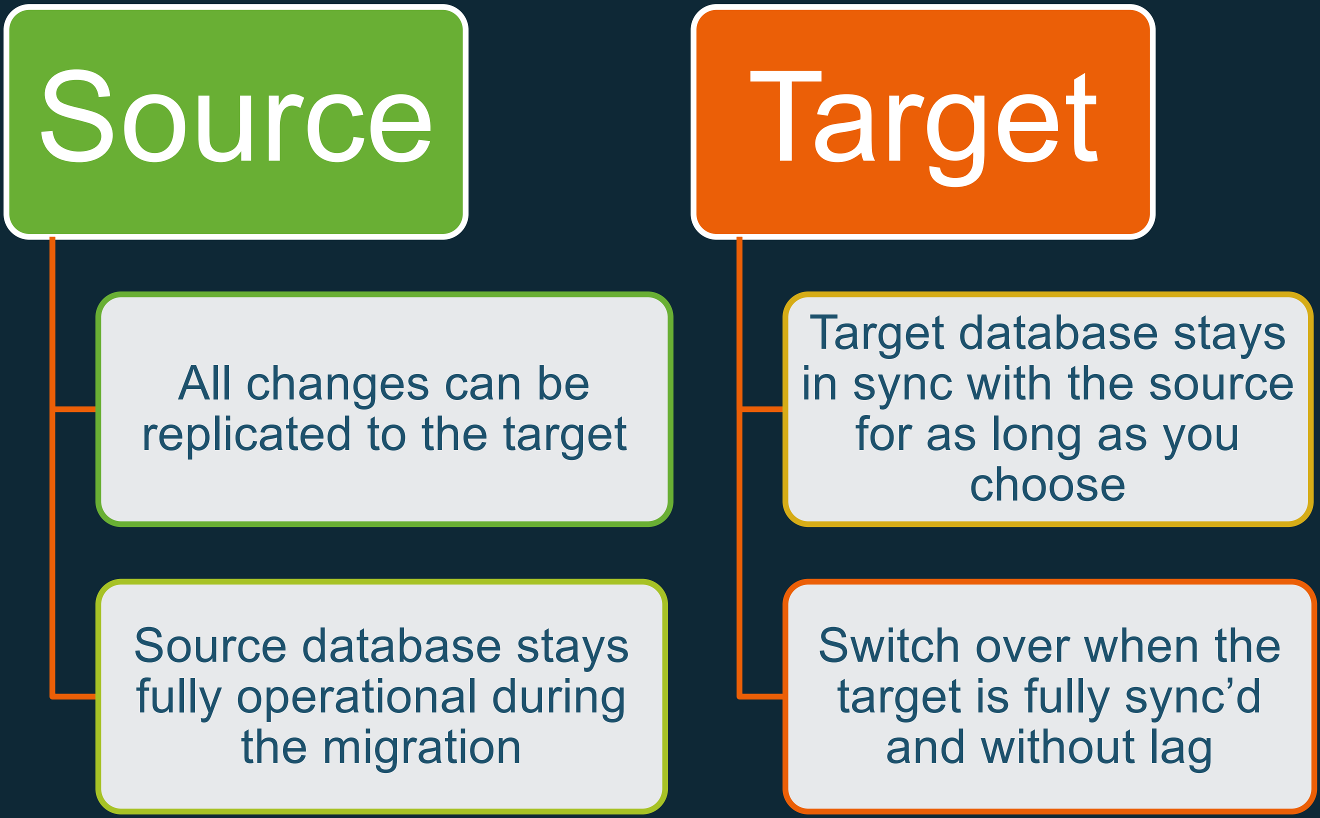
AWS Database Migration Service

- Simple to use
- Minimal downtime
- Supports widely used databases
- Low cost
- Fast and easy to setup
- Reliable

Simple to use

- No drivers or applications to install
- No changes to the source database in most cases
- Just a few clicks to start a migration from the console
- DMS manages the complexities of migration for you
- Automatically replicate changes
- Can be used for continuous replication

Minimal downtime



Source

All changes can be replicated to the target

Source database stays fully operational during the migration

Target

Target database stays in sync with the source for as long as you choose

Switch over when the target is fully sync'd and without lag

Supports widely used databases

Amazon RDS

Amazon EC2

Sources*	Targets**
Oracle	Oracle
SQL Server	SQL Server
Azure SQL	PostgreSQL
PostgreSQL	MySQL
MySQL	Amazon Redshift
SAP ASE	SAP ASE
MongoDB	Amazon S3
Amazon S3	Amazon DynamoDB
IBM DB2	Amazon Kinesis
	Amazon ElasticSearch

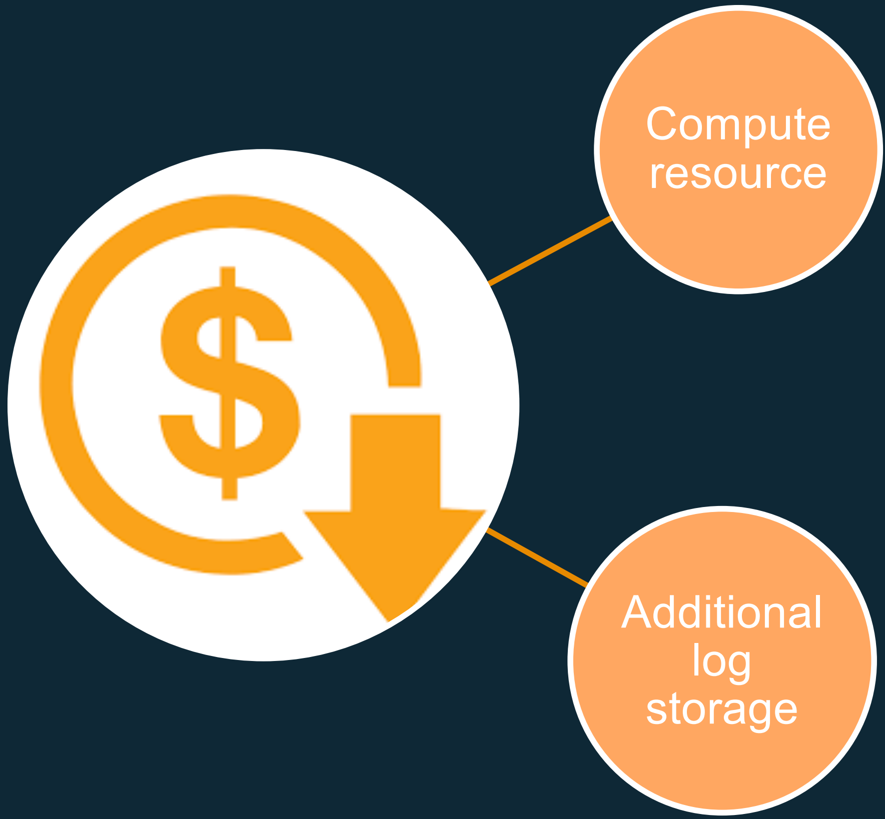
On-premises database

* https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Source.html

** https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Target.html

Low cost

https://aws.amazon.com/dms/pricing/



Pricing Example

Instance Type	Hourly Rate	Duration	Activities	Total
t2.small	$0.036	2 weeks	Testing	$12.096
c4.large	$0.154	2 weeks	Initial Load & CDC Until Cutover	$51.744

Migrate a 1 TB DB for under $65 ($63.84)

Fast and easy to setup

Set up a migration task in minutes

Connect to the source database	Connect to the target database	Create a replication instance to run the migration	Create a task	Run the task

You can use different tasks with different settings for different environments

Reliable

- Highly resilient and self–healing
- Continual monitoring
  - Source and target databases
  - Network connectivity
  - Replication instance
- In case of interruption, it automatically restarts the process and continues the migration from where it was halted
- Multi-AZ option for high-availability

CloudWatch

DMS Components

Database Migration Service components

Replication instance

Endpoint

Task

Replication instance

Replication Instance

Dedicated EC2 instance

- T2, C4, or R4 instance types

Public or private IP address

- Private: Source or Target inside VPC / VPN / DC
- Public: Source or Target outside VPC

Task execution

- Moves the data from the source to the target
- Support for multiple tasks

Endpoint

Connection information to source or target database

- Endpoint type, identifier, engine (source or target)
- Server name, port, SSL mode
- Username, password

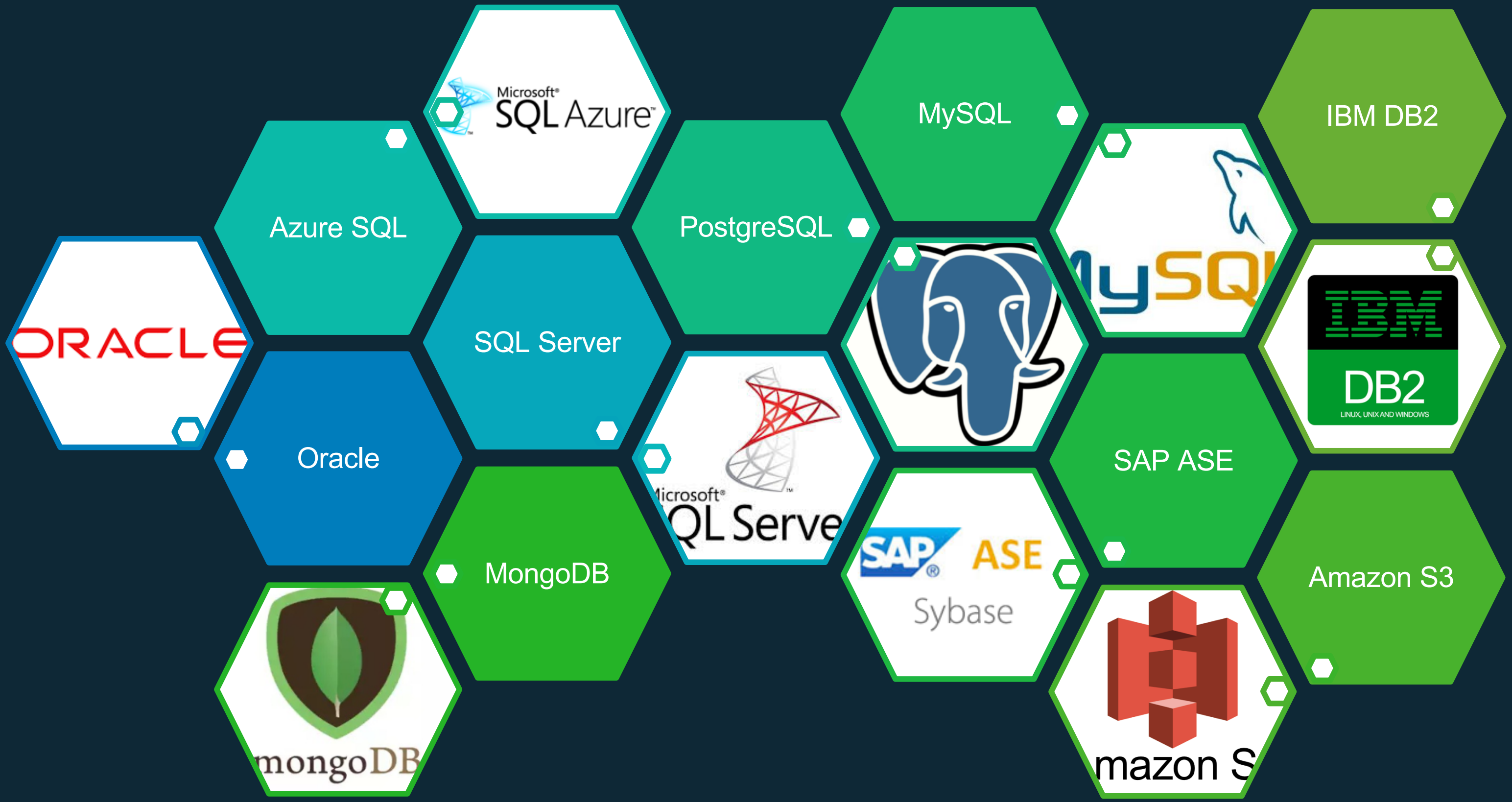
Advanced attributes

- Extra connection information
- KMS key

Test with a replication instance

Endpoint sources

https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Source.html



Azure SQL

SQL Server

Oracle

MongoDB

PostgreSQL

MySQL

SAP ASE

IBM DB2

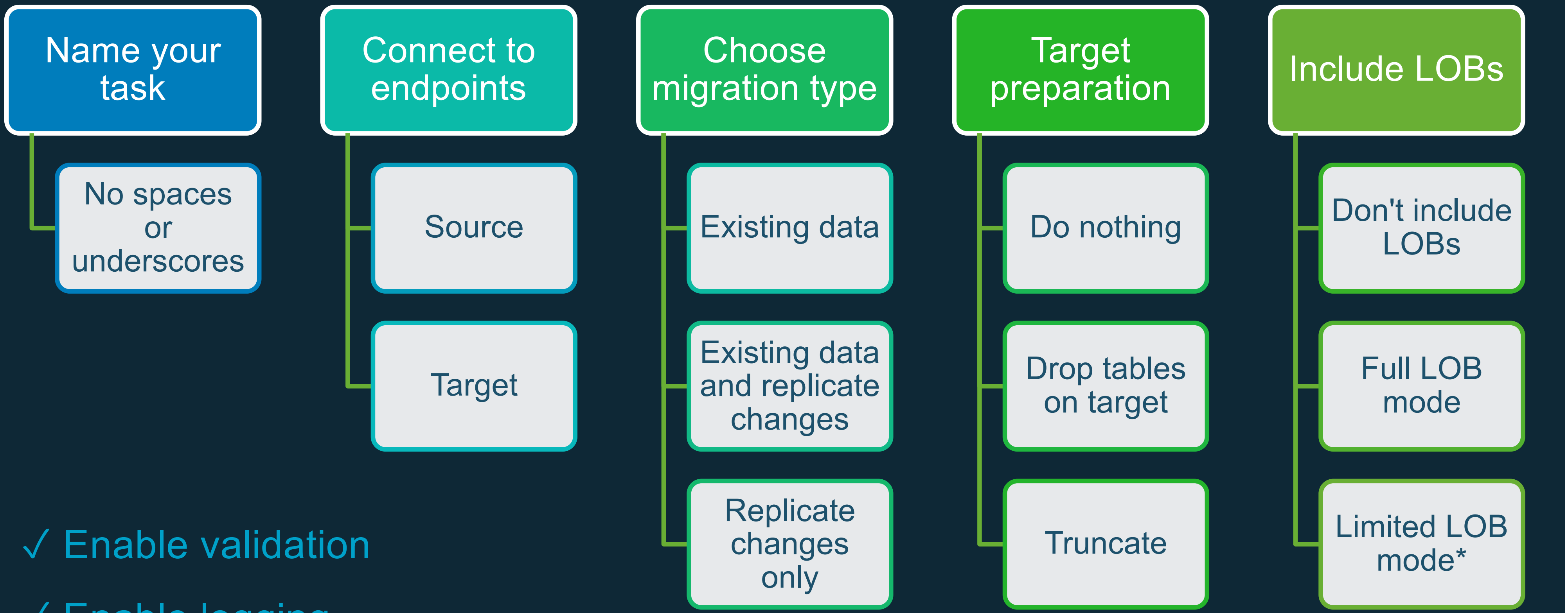
Amazon S3

Endpoint targets

https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Target.html

Targets
Oracle
SQL Server
PostgreSQL
MySQL
Amazon Redshift
SAP ASE
Amazon S3
Amazon DynamoDB
Amazon Kinesis
Amazon ElasticSearch

Task

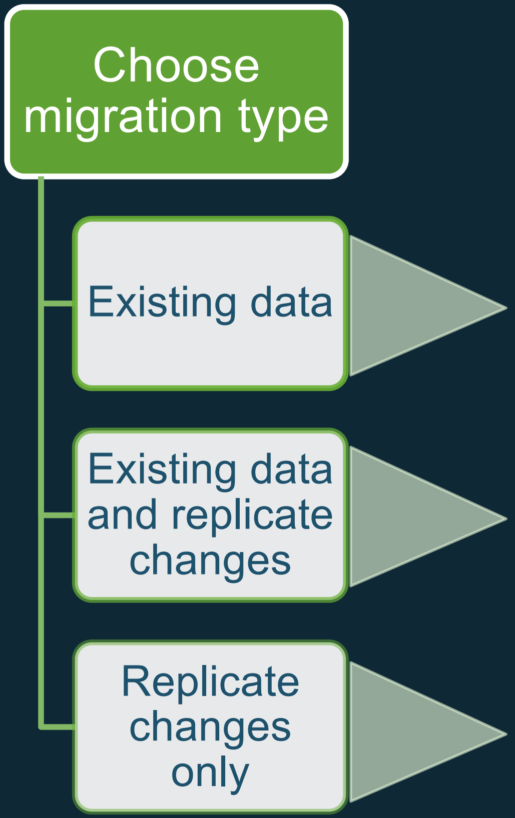


- Name your task
  - No spaces or underscores
- Connect to endpoints
  - Source
  - Target
- Choose migration type
  - Existing data
  - Existing data and replicate changes
  - Replicate changes only
- Target preparation
  - Do nothing
  - Drop tables on target
  - Truncate
- Include LOBs
  - Don't include LOBs
  - Full LOB mode
  - Limited LOB mode*

✓ Enable validation

✓ Enable logging

Task – migration type



Creates files or tables in the target database
Populates the tables with data from the source.
Migrate existing data option in the AWS console and `Full Load` in the API.

Captures changes on the source during migration
Once initial migration completes, changes are applied to the target as units of completed transactions
Migrate existing data and replicate ongoing changes option in the AWS console and `full-load-and-cdc` in the API.

Reads the recovery file on the source database
Groups together transactions and applies them to the target. Buffering as needed
Replicate data changes only option in the AWS console

Task – Target preparation

Target preparation

Do nothing

Drop tables on target

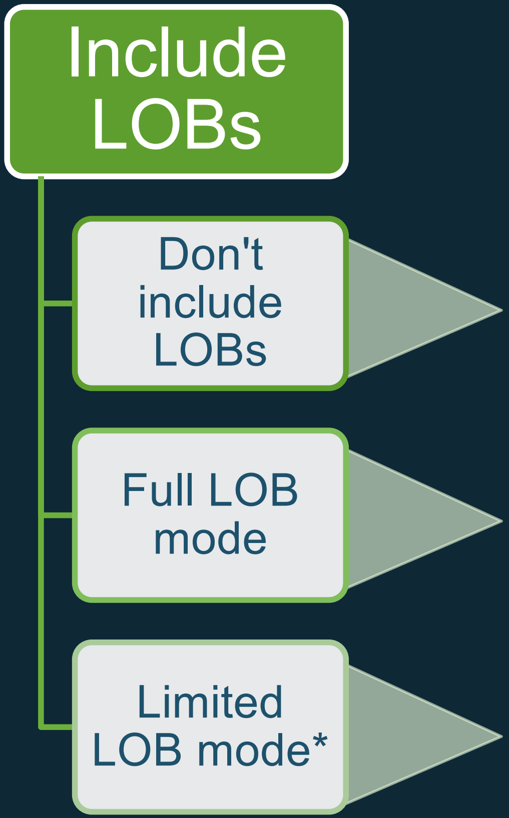
Truncate

In **Do nothing** mode, AWS DMS assumes target tables are pre-created. In full load or full load plus CDC, ensure that the target tables are empty before starting the migration.

In **Drop tables on target** mode, AWS DMS drops the target tables and recreates them before starting the migration. This ensures that the target tables are empty when the migration starts.

In **Truncate** mode, AWS DMS truncates all target tables before the migration starts.

Task – Include LOBs



Include LOBs

- Don't include LOBs: LOB columns are excluded from the migration.
- Full LOB mode: Migrate complete LOBs regardless of size. AWS DMS migrates LOBs piecewise in chunks controlled by the Max LOB size parameter. This mode is slower than using Limited LOB mode.
- Limited LOB mode*: Truncate LOBs to the value of the Max LOB size parameter. This mode is faster than using Full LOB mode.

More about replication tasks

- Tasks are the workhorse of the migration
- Tasks run on a replication instance
- Multiple tasks can be run in parallel
- Task settings JSON gives you control over performance and debugging

Migration scenarios

Possible migration preparation scenarios

Using DMS alone

- DMS Creates tables and primary keys using **Drop tables on target**
- Manually add secondary indexes and foreign key objects after initial load has completed.

Using SCT and DMS together

- Use Schema Conversion Tool to create all objects
- Disable secondary indexes and foreign key objects
- Load using DMS using **Do nothing**
- Re-enable secondary indexes and foreign key objects

Using Native Tools and DMS

- Use database scripts to create tables, primary keys, views, and sequences
- Load tables with DMS using **Truncate**
- Manually add secondary indexes and foreign key objects after initial load has completed.

Steps to migrate - overview

Start the full load

While loading data, also capture changes

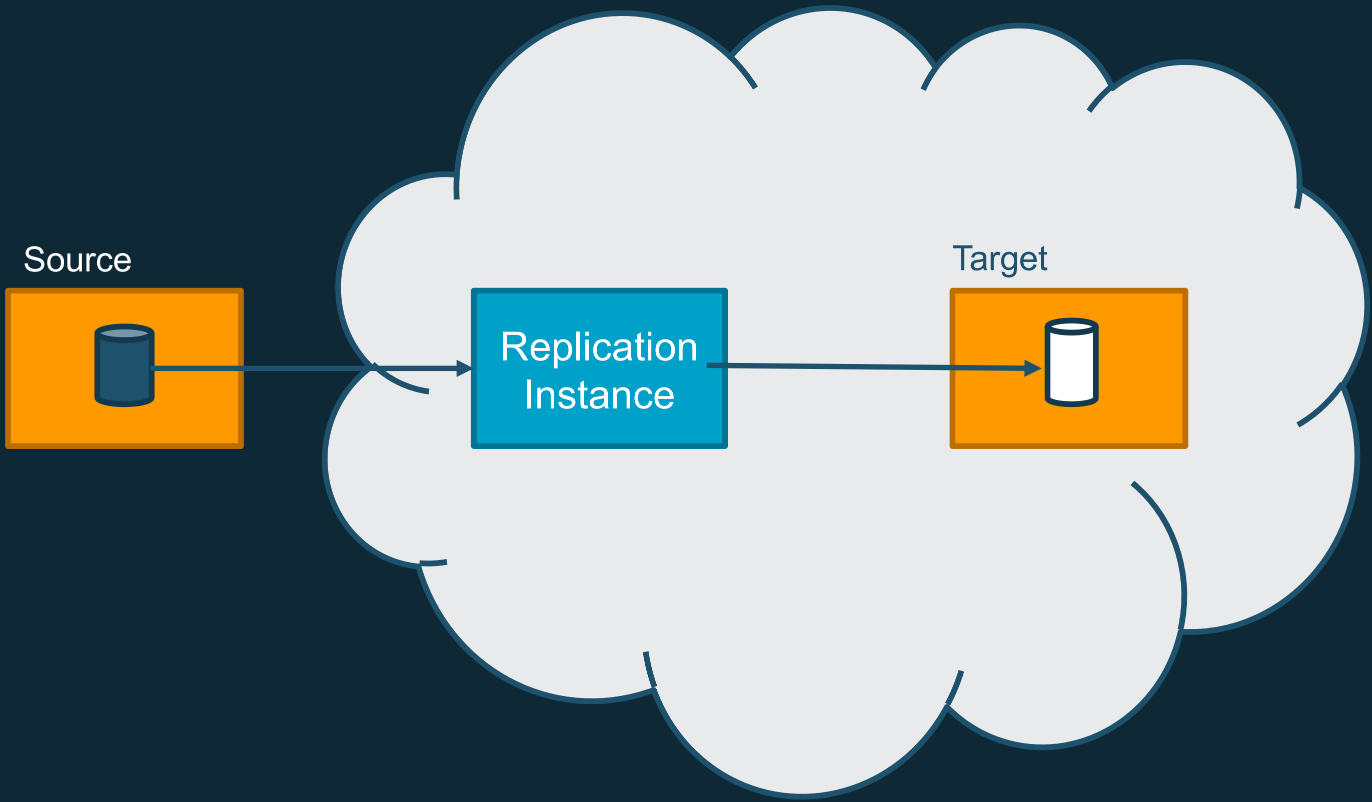
Load complete - apply captured changes

Changes reach steady state

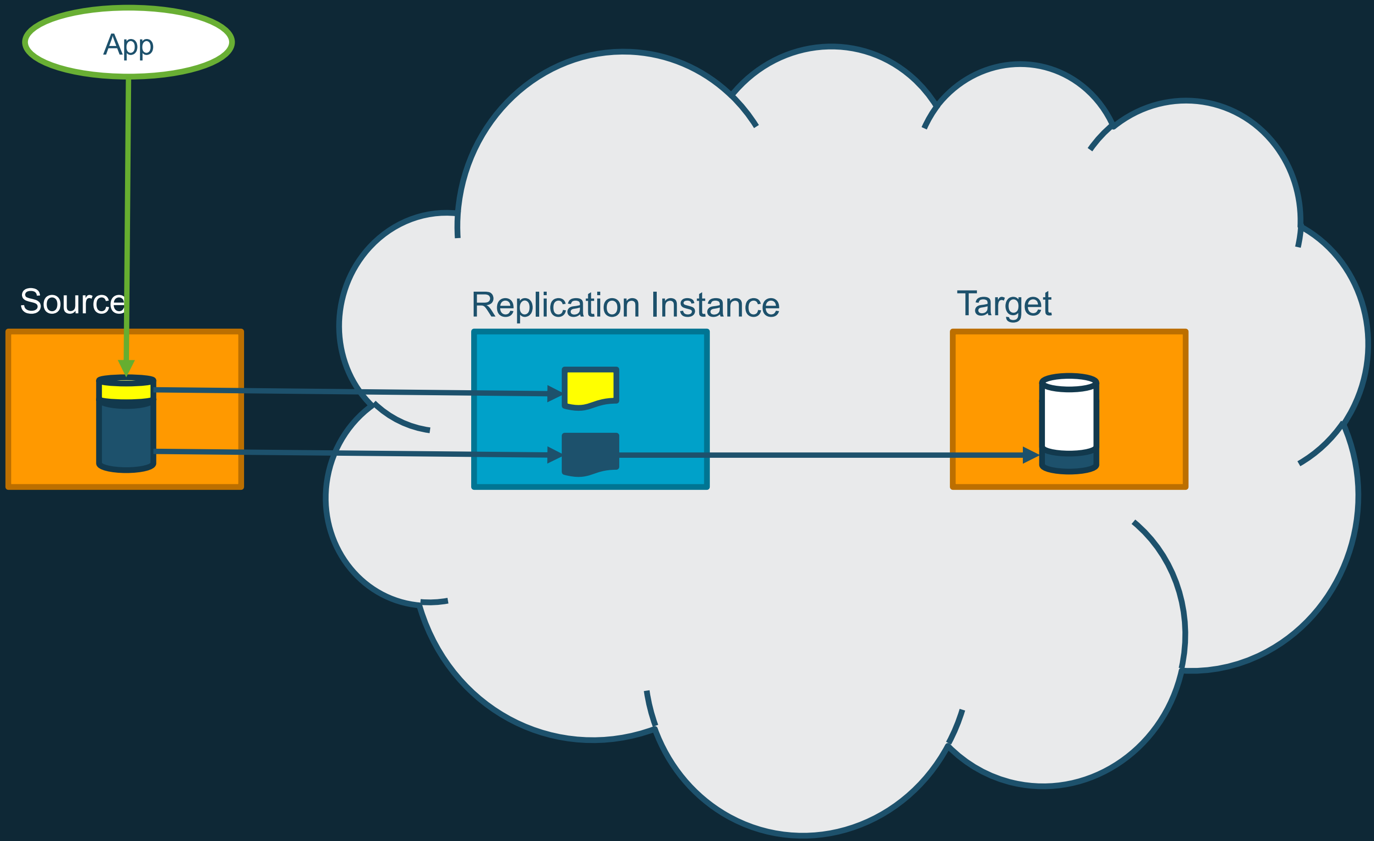
Shutdown apps and apply remaining changes

Change your application endpoint

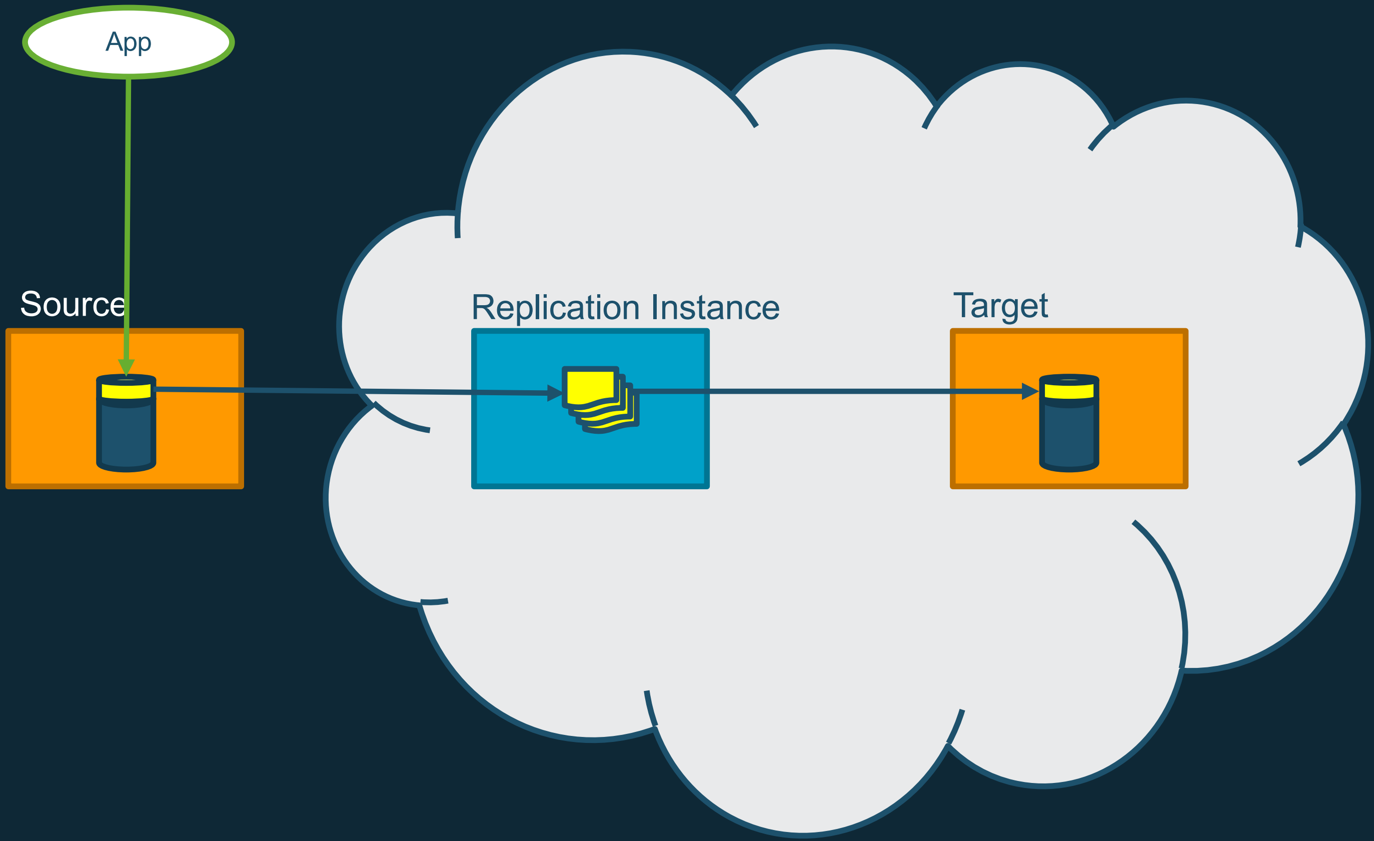
Start full load



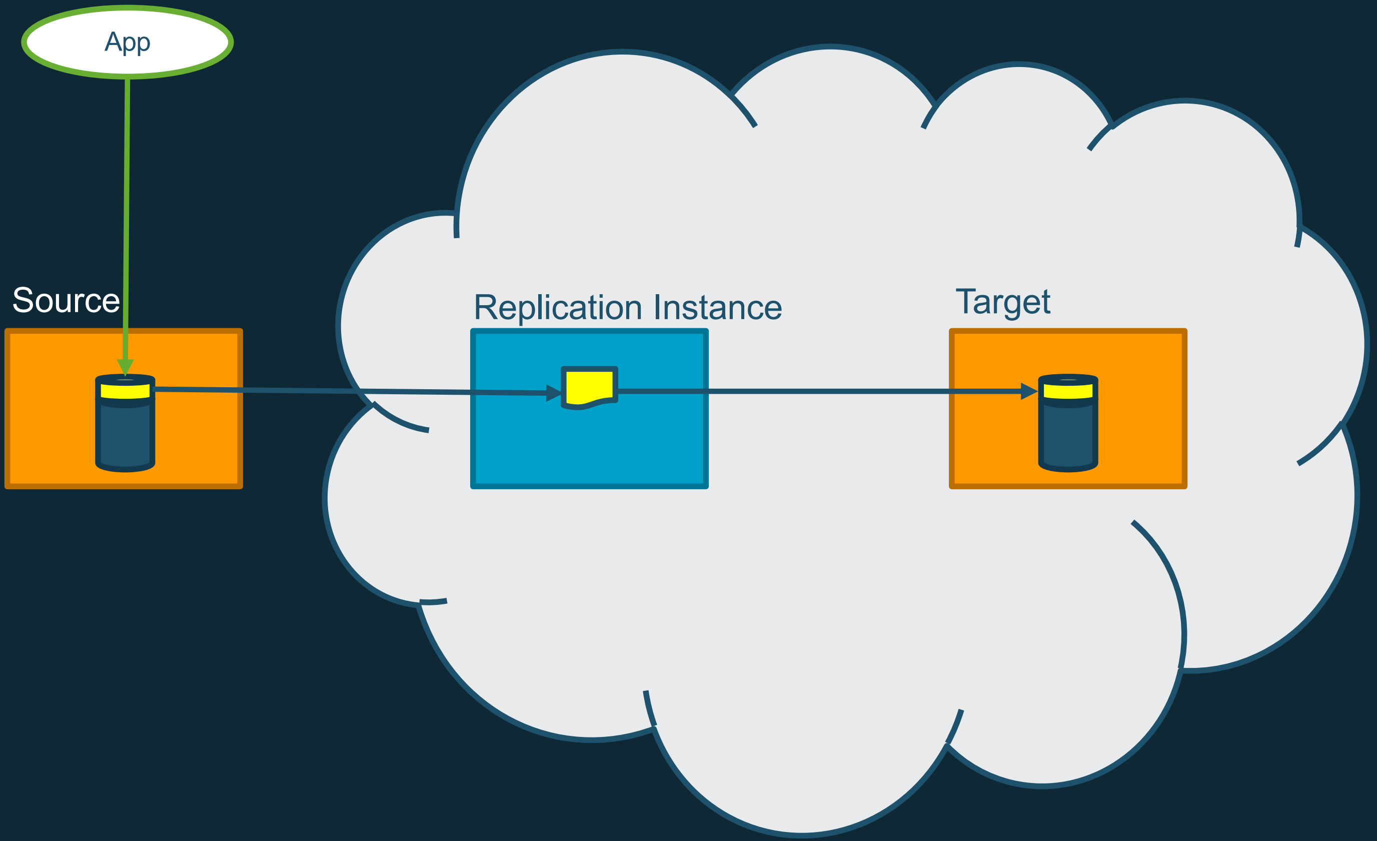
While loading data also capture changes



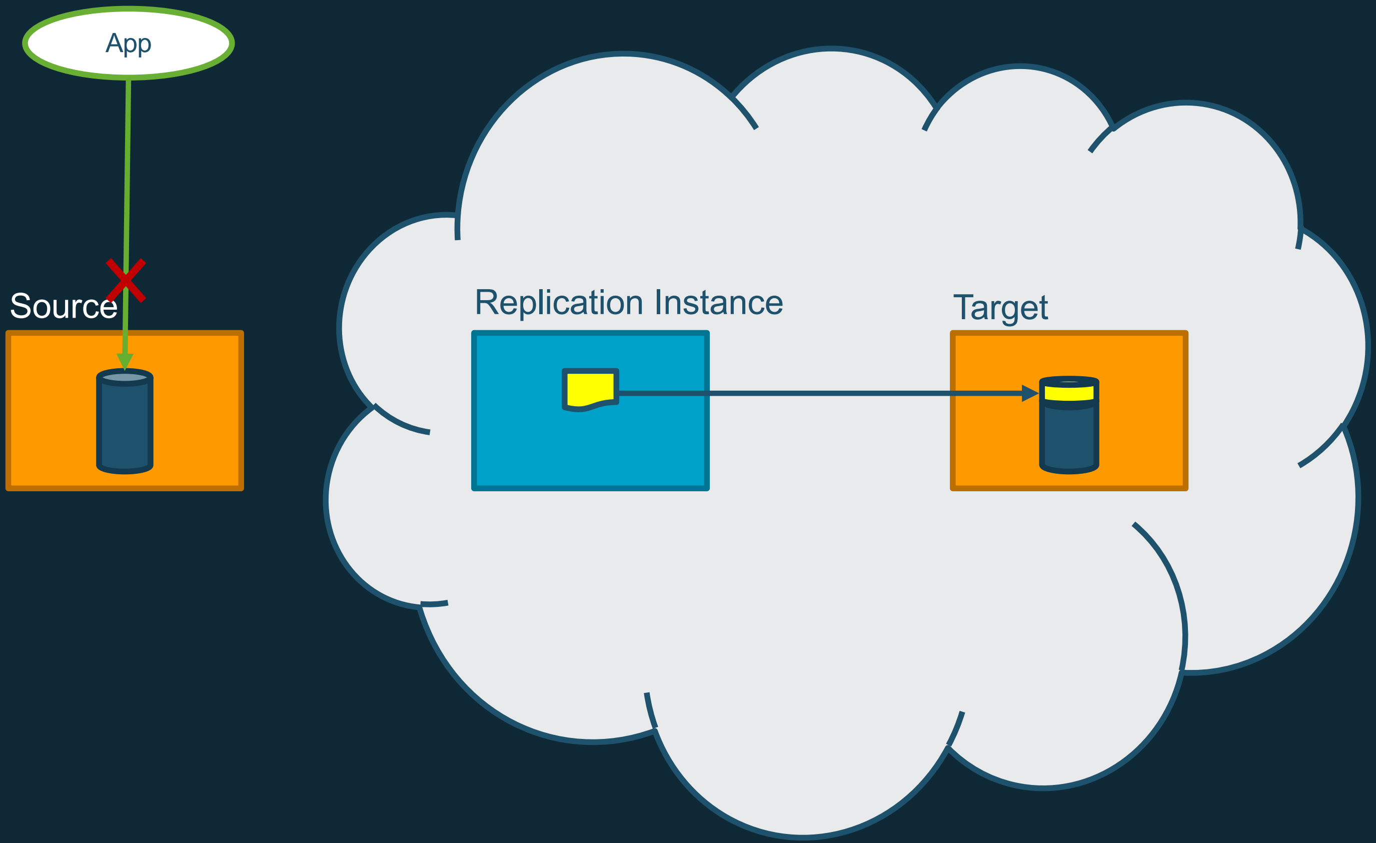
Load complete - apply captured changes



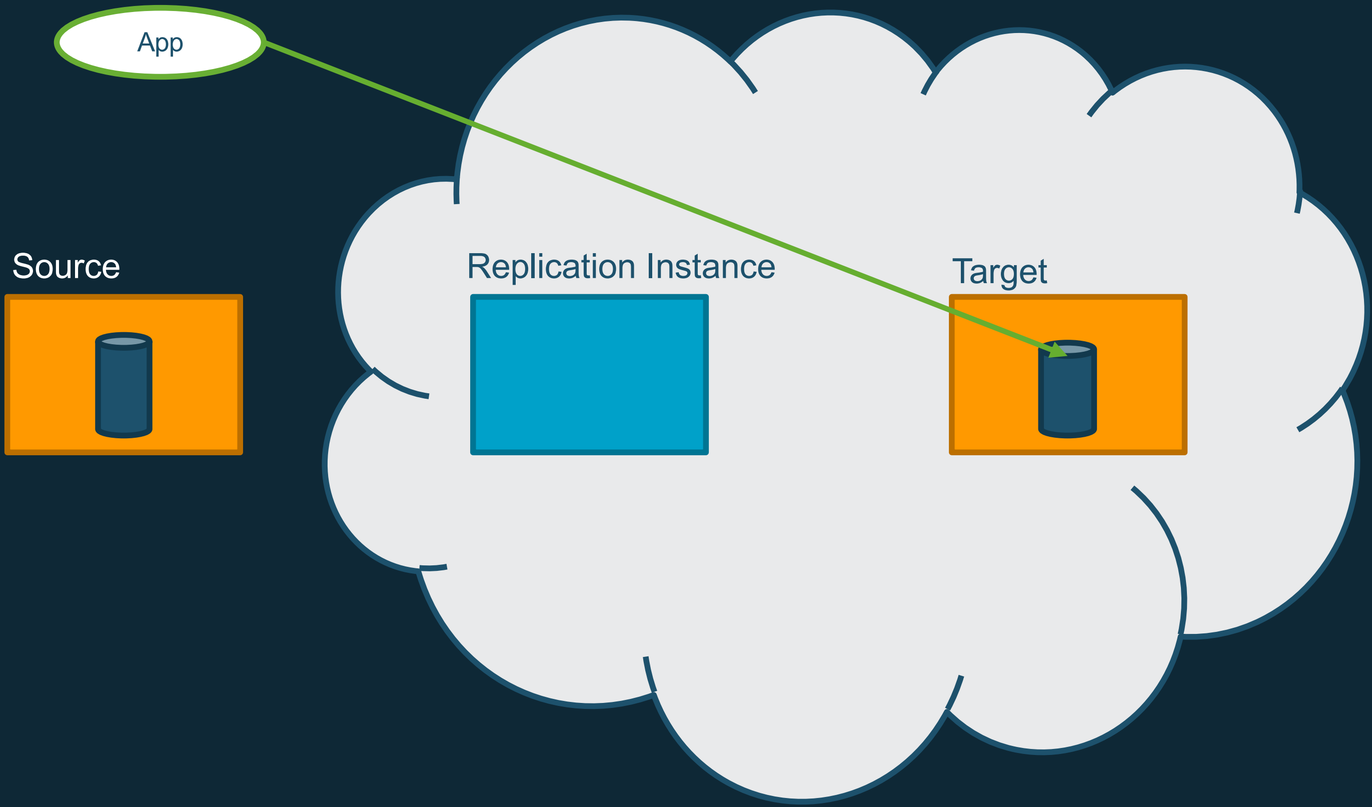
Changes reach steady state



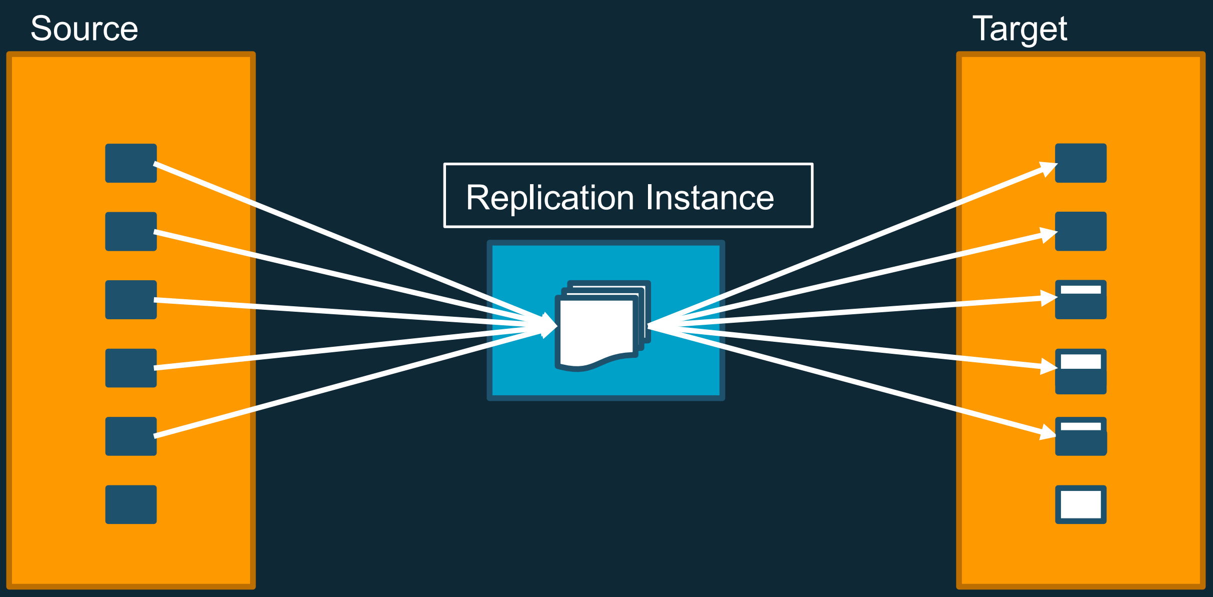
Shutdown apps and apply remaining changes



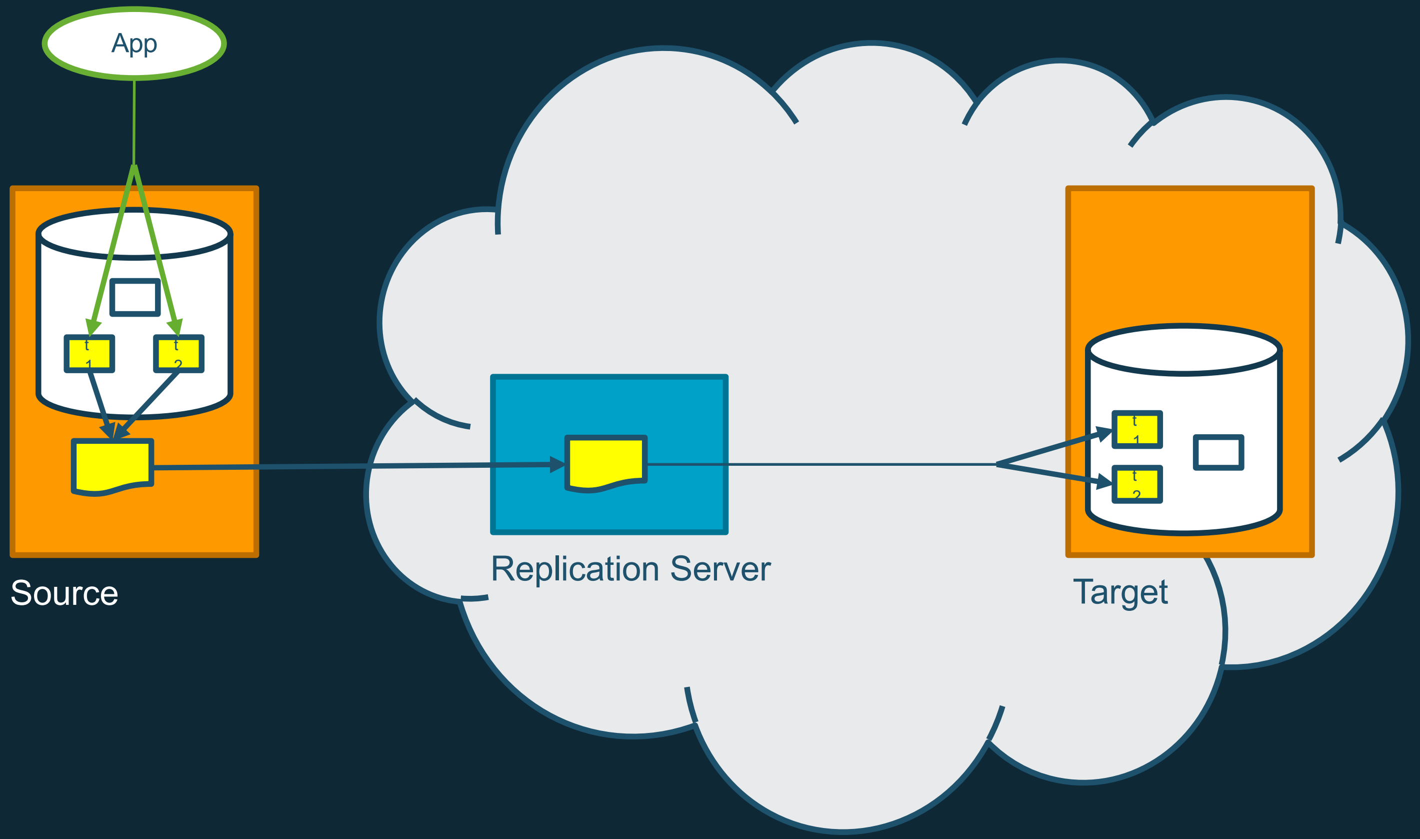
Change your application endpoint



Load is table by table



Changes are transactional - from db logs



DMS – change data capture (CDC)

“No Touch” design

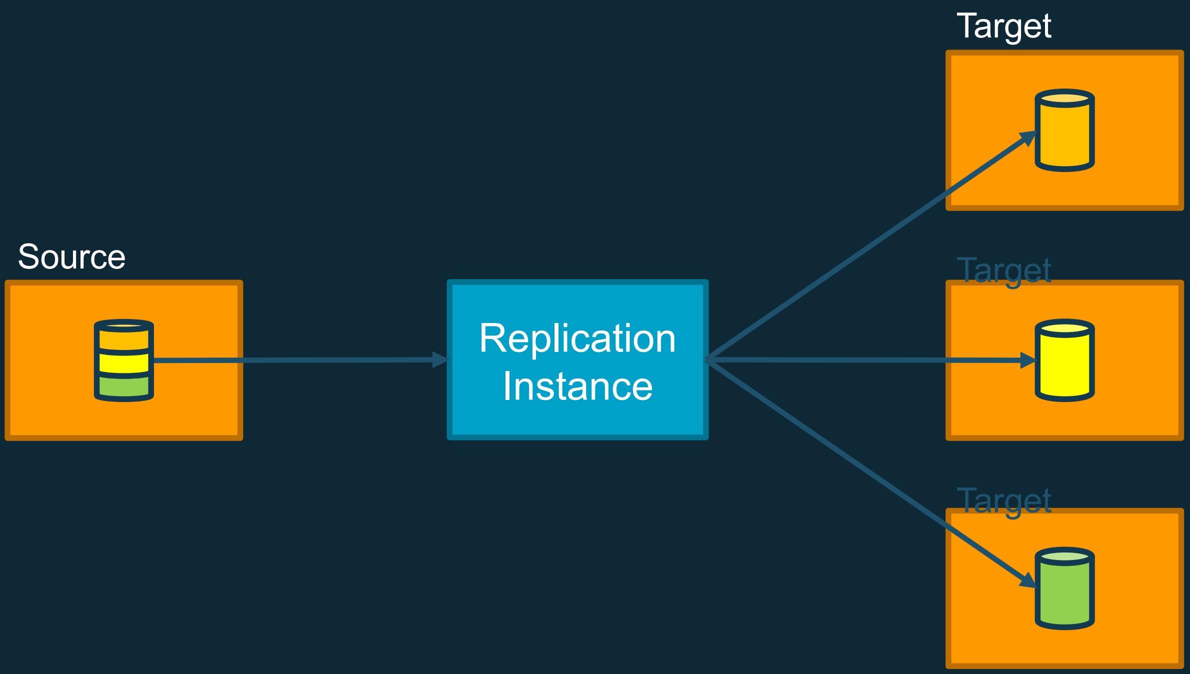
- Reads recovery log of source database
- Using the engine’s native change data capture API
- No agent required on the source
- Changes captured as transactions and applied in order
- Activated when load starts
- Changes are applied after initial load is complete

DMS – change data capture (CDC)

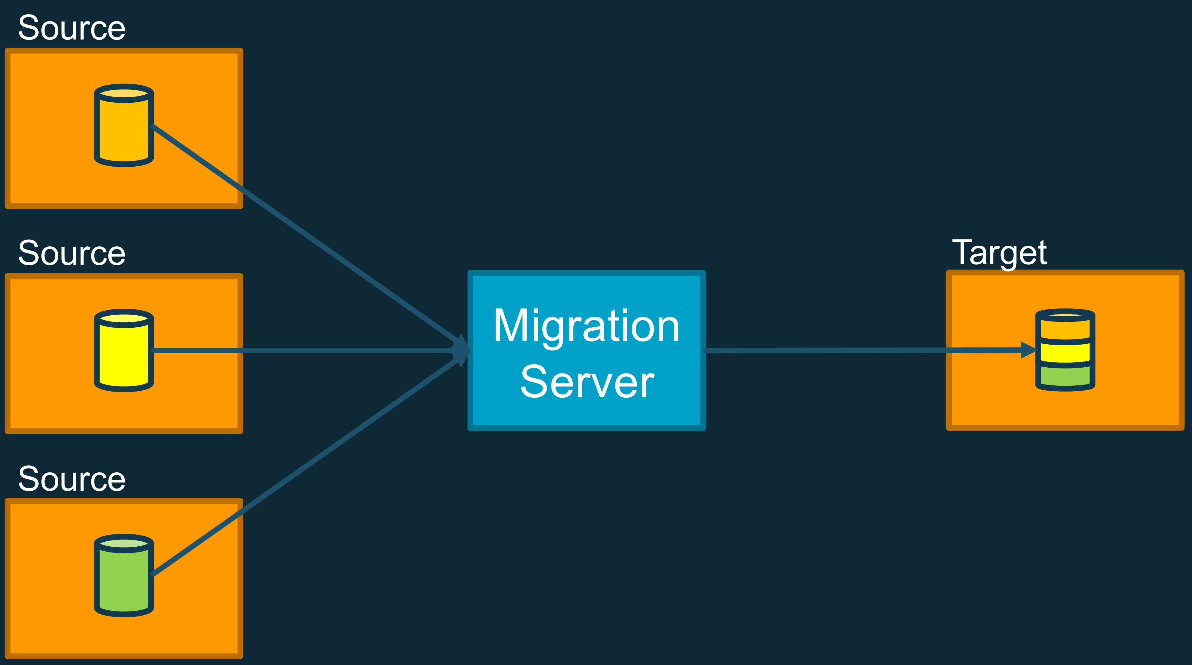
Some requirements

- Oracle: Supplemental logging required
- MySQL: Full image row level bin logging required
- SQL Server: Recovery model bulk logged or full
- Postgres: wal_level = logical; max_replication_slots >= 1; max_wal_Senders >=1; wal_sender_timeout = 0

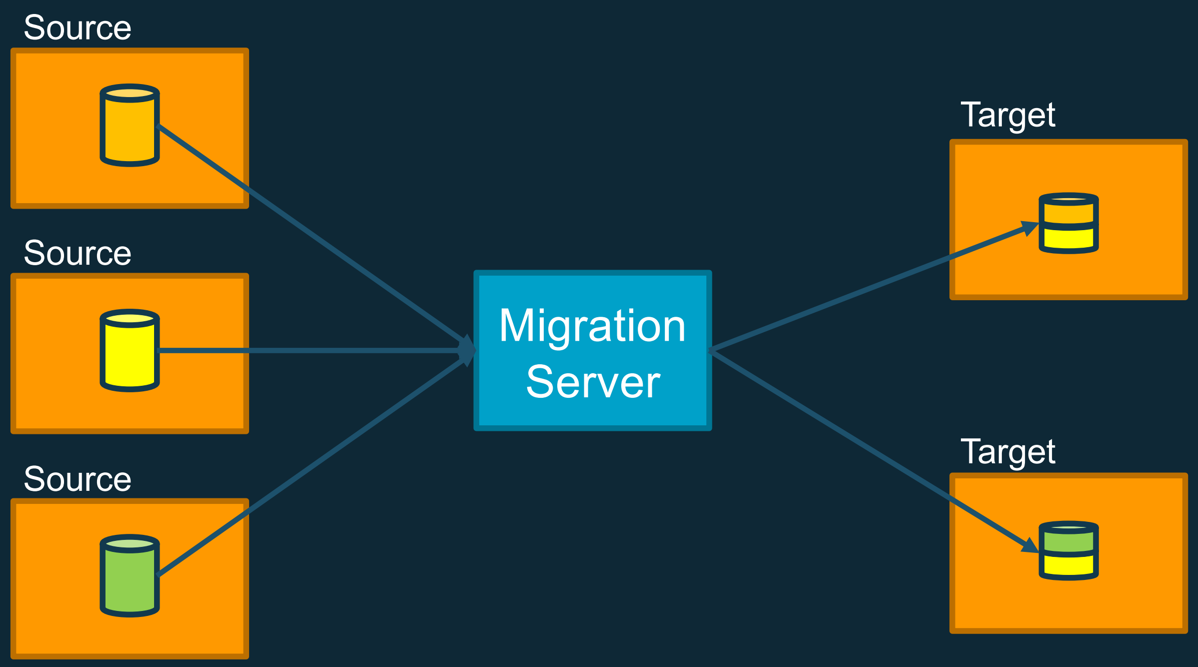
Multiple targets



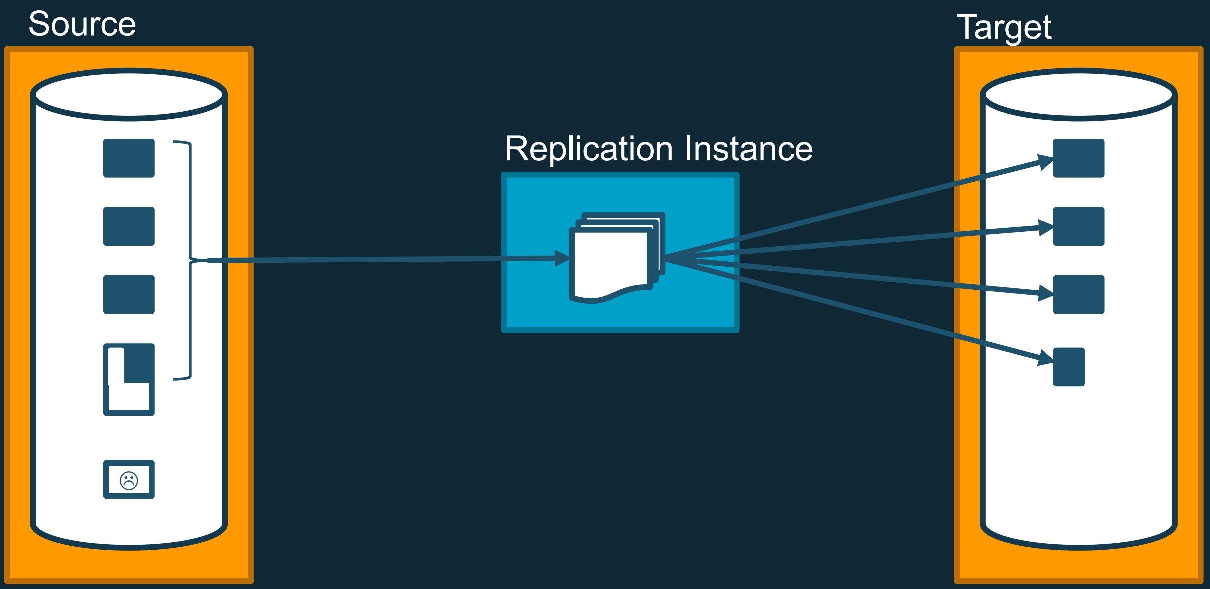
Multiple Sources



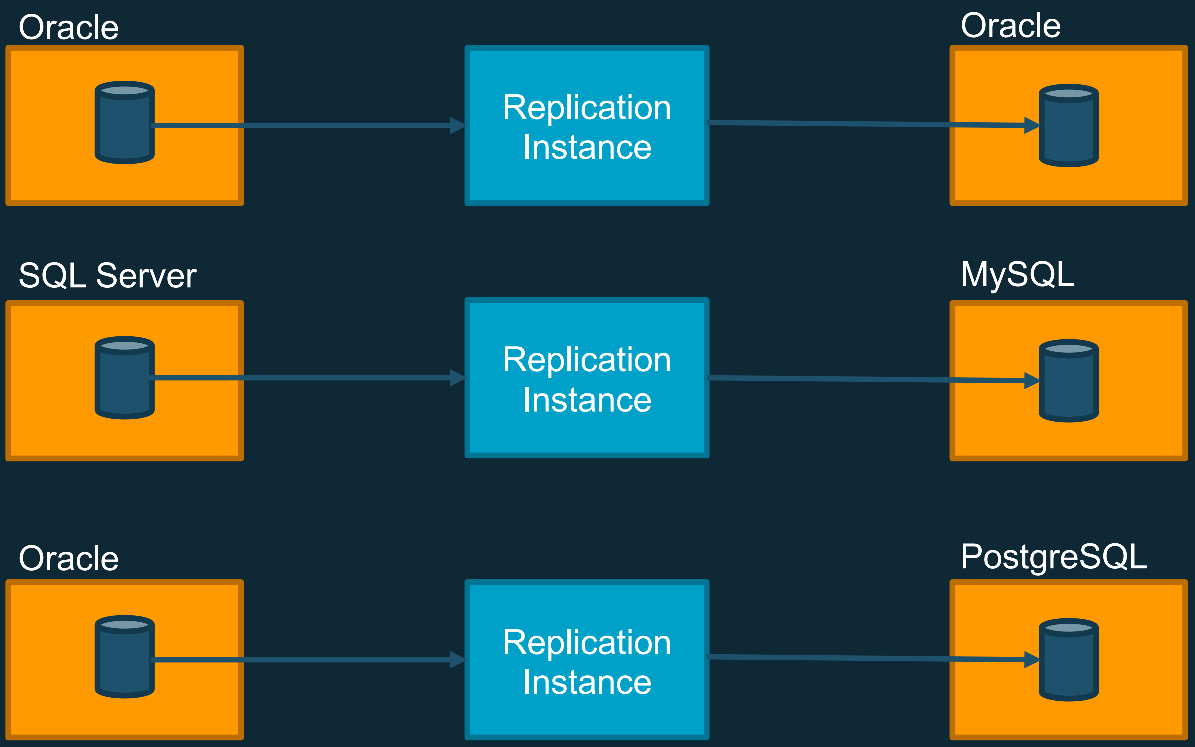
Multiple sources and targets



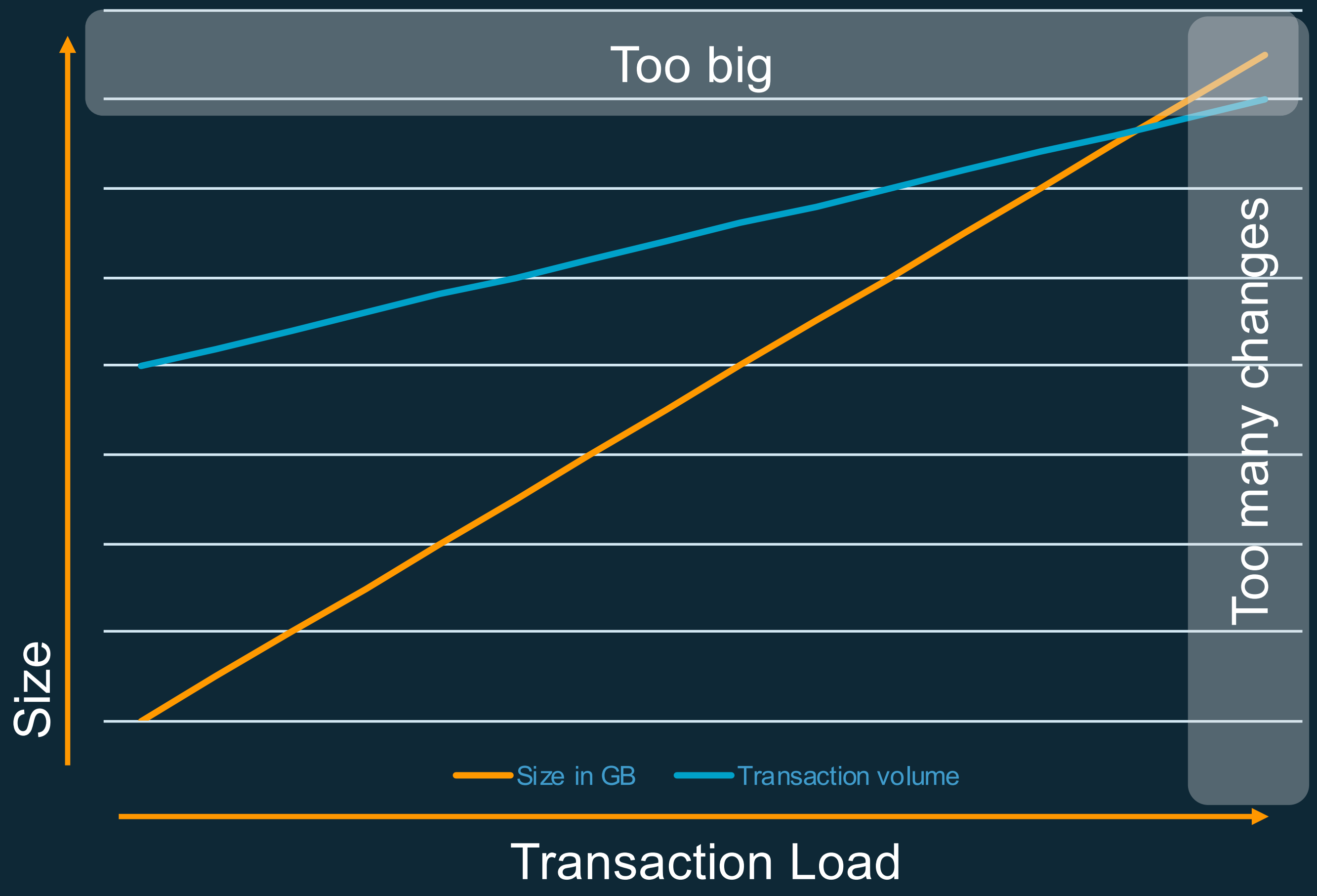
You don’t have to take everything



Homogenous or heterogeneous



DMS limits


Too big
Too many changes
Size
Size in GB
Transaction volume
Transaction Load

Setup and configuration tips

- Enable CloudWatch logs (not enabled by default)
- Choose LOB mode carefully
- Replication instance security group is default for VPC – change after creation
- Extra connection attributes can alter how the migration task operates
- Provide transformation rules for changing case

Source configuration tips

- Oracle specific settings for CDC
  - Enable supplemental logging on tables with no primary key
  - If you add a table mid-migration add supplemental logging
    - Even if the table has a primary key
  - To have DMS automatically configure supplemental logging in Oracle add an extra connection attribute
    - addSupplementalLogging=Y
- Check for unsupported datatypes
- Can use a read replica or standby as a migration source

Performance tips

- Use larger DMS instance for maximum throughput
  - CPU for type conversions
- Check network throughput
- Split load across multiple tasks and/or DMS instances
  - Remember transaction boundaries when capturing changes
- Reduce contention on your target
  - Turn off logging
  - Run in single AZ

Further reading on the AWS Database Blog

Questions

Thank You!