AWS re:Invent

DAT362-R

Dive deep into AWS SCT and AWS DMS
A Jobvite case study

John Winford
Global Lead Database Freedom
Amazon Web Services

Chaitanya Konduri
DevOps Manager
Jobvite

Pranav Bhargava
Director of Engineering
Jobvite

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

Agenda

Introduction to AWS Database Migration Service (AWS DMS) and AWS Schema Conversion Tool (AWS SCT)

AWS DMS and AWS SCT product highlights

Jobvite use case: Migrating SQL server databases to Aurora MySQL

Migration resources

Q&A

Related breakouts

DAT214 Automation framework to migrate relational databases

DAT342 Lessons learnt from migrating Oracle databases to Amazon Aurora

DAT343 Lessons learnt from migrating SQL Server databases to Amazon Aurora

DAT345 Assessing and categorizing your database migrations

DAT360 Analytical use cases with AWS DMS

DAT362-R1 Dive deep into AWS SCT and AWS DMS (Jobvite use case)

DAT363 Migrating your data warehouses to Amazon Redshift

DAT366 Running Oracle on Amazon RDS and migrating to Aurora PostgreSQL

DAT367 Running SQL Server on Amazon RDS and migrating to Aurora PostgreSQL

DAT377 Getting started with AWS DMS and AWS SCT

Introduction to AWS DMS and AWS SCT

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

AWS migration tooling



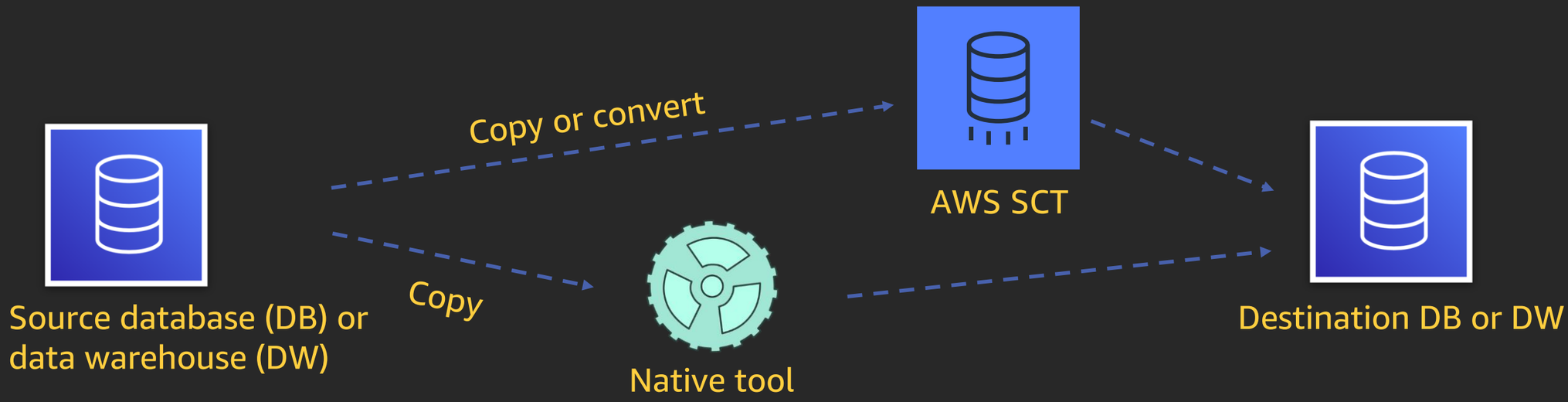
AWS Schema Conversion Tool (AWS SCT) converts your commercial database and data warehouse schemas to open-source engines or AWS-native services such as Amazon Aurora and Amazon Redshift

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

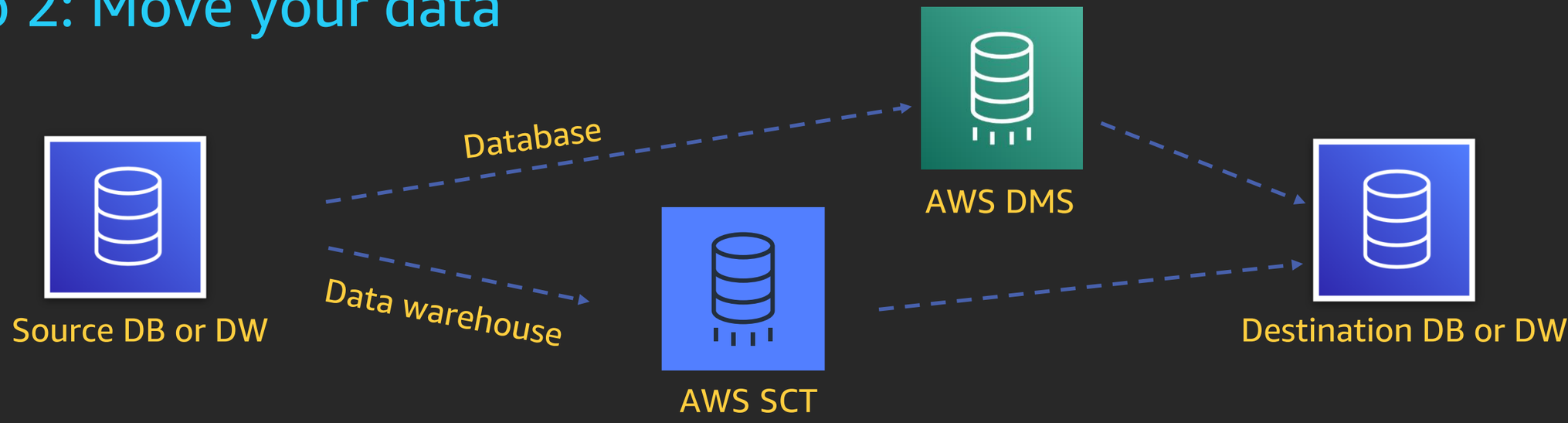


Database migration process

Step 1: Convert or copy your schema



Step 2: Move your data



>200,000 databases migrated with DMS



Plan, assess, and convert

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Workload Qualification Framework (WQF)

Assesses and classifies OLTP workloads to determine the ease of migration and staff-hour consumption, and to 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 AWS DMS

WQF classifies OLTP workloads into five categories:

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

- Migration complexity assessments
- Workload migration strategy
- Migration tools

Sample WQF report: Migration effort

	WQF Assessment Report										
Migration Steps\Roles	**Migration Architect**		**Database Developer**		**Application Developer**		**Application QA**		**Project Manager**		**Total**
	Head Count	**Days**	**Head Count**	**Days**	**Head Count**	**Days**	**Head Count**	**Days**	**Head Count**	**Days**	
1. Envisioning/Assessment	2	1	0		0		0		1	0	**2**
2. Database schema conversion	0		5	6	0		0		1	0	**31**
3. Application conversion/remediation	0		3	0	2	0	0		1	0	**0**
4. Script conversion	0		3	5	0		0		1	2	**17**
5. Integration with 3rd party applicatio	0		0		0		0		1	0	**0**
6. Data migration	0		2	3	0		0		1	1	**7**
7. Functional testing of the entire syste	0		0		0		4	4	1	0	**15**
8. Performance tuning	0		4	1	0		0		1	0	**5**
9. Integration and deployment	0		0		0		0		1	0	**0**
10. Training and knowledge transfer	2	1	0		0		0		1	0	**2**
11. Documentation and version suppor	1	1	0		0		0		1	0	**1**
12. Post production support	1	1	0		0		0		1	0	**1**
Total (days)											**80**

AWS SCT

AWS SCT helps automate database schema and code conversion tasks when migrating from source to target database engines

Features

Create assessment reports for homogeneous/heterogeneous migrations

Convert DB schema

Convert DW schema

Convert embedded application code

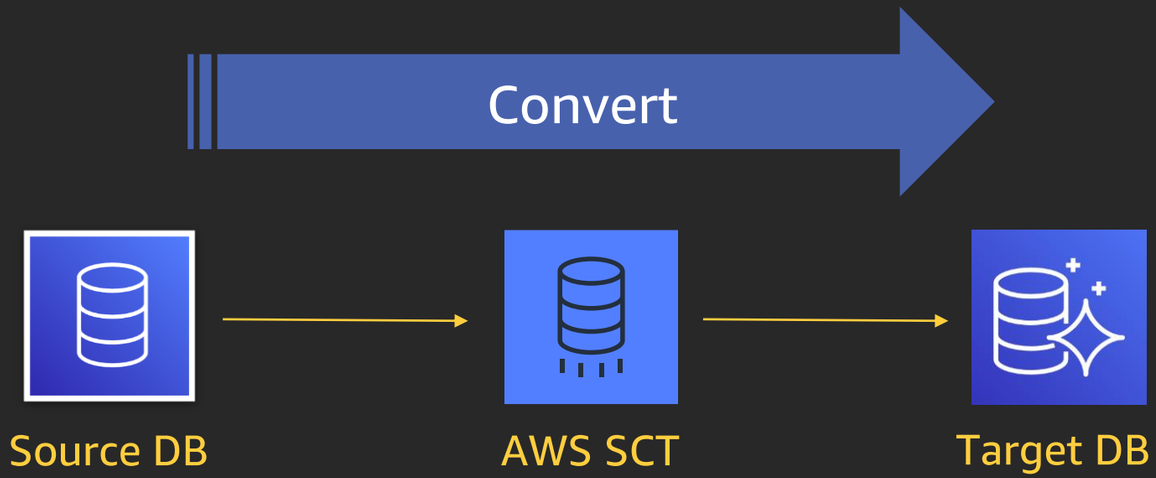
Code browser that highlights places where manual edits are required

Secure connections to your databases with SSL

Service substitutions/ETL modernization to AWS Glue

Migrate data to data warehouses using SCT data extractors

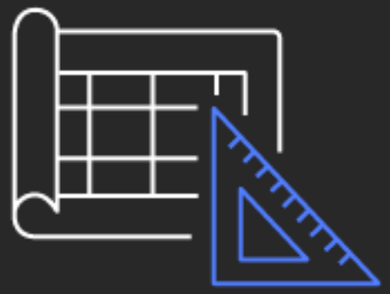
Optimize schemas in Amazon Redshift



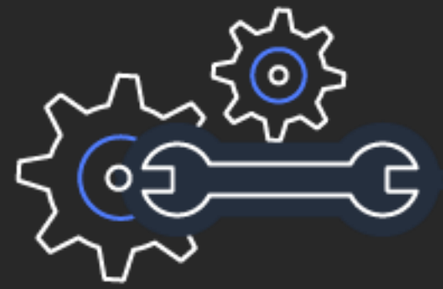
AWS SCT product highlights



Assess



Plan



Convert schema and code



Optimize



Migrate data warehouses

Database migration assessment

Database migration assessment report

Source database:
DMS_SAMPLE.tharunk@oracle.c435kv7tmkj2.us-west-2.rds.amazonaws.com:8193:TEST
Oracle Database 12c Enterprise Edition 12.1.0.2.0 (64bit Production), Enterprise edition

Executive summary

We completed the analysis of your Oracle source database and estimate that 98% of the database storage objects and 96% of database code objects can be converted automatically or with minimal changes if you select Amazon RDS for PostgreSQL as your migration target. Database storage objects include schemas, tables, table constraints, indexes, types, collection types, sequences, synonyms, view-constraints, clusters and database links. Database code objects include triggers, views, materialized views, materialized view logs, procedures, functions, packages, package constants, package cursors, package exceptions, package variables, package functions, package procedures, package types, package collection types, scheduler-jobs, scheduler-programs and scheduler-schedules. Based on the source code syntax analysis, we estimate 99.8% (based on # lines of code) of your code can be converted to Amazon RDS for PostgreSQL automatically. To complete the migration, we recommend 6 conversion action(s) ranging from simple tasks to medium-complexity actions to significant conversion actions.

Migration guidance for database objects that could not be converted automatically can be found here

Database objects with conversion actions for Amazon RDS for PostgreSQL

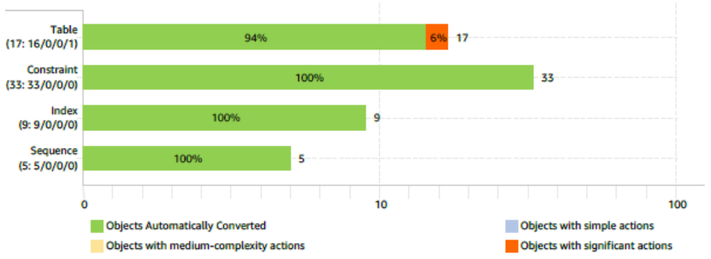
Of the total 64 database storage object(s) and 23 database code object(s) in the source database, we identifed 63 (98%) database storage object(s) and 22 (96%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

The target database version is less than PostgreSQL 11.1 (11.1). The converted code might not work properly.

1 (2%) database storage object(s) require 1 significant user action(s) to complete the conversion.

1 (4%) database code object(s) require 1 medium user action(s) to complete the conversion.

Figure: Conversion statistics for database storage objects



Connect AWS SCT to source and target databases

Run assessment report

Read executive summary

Follow detailed instructions

Database migration assessment report

Source database:
DMS_SAMPLE.tharunk@oracle.c435kv7tmkj2.us-west-2.rds.amazonaws.com:8193:TEST
Oracle Database 12c Enterprise Edition 12.1.0.2.0 (64bit Production), Enterprise edition

Package Procedure Changes

Not all package procedures can be converted automatically. Youll need to address these issues manually.

Issue 5584: Converted functions depends on the time zone settings

Recommended action: Review the transformed code, and set time zone manually if necessary.

Issue code: 5584 | Number of occurrences: 2 | Estimated complexity: Simple

Documentation references: http://www.postgresql.org/docs/9.6/static/functions-datetime.html

Schemas.DMS_SAMPLE.Packages.TICKETMANAGEMENT.Private procedures.TRANSFERTICKET: 1463:1469
Schemas.DMS_SAMPLE.Packages.TICKETMANAGEMENT.Public procedures.SELLTICKETS: 1772:1778

Package Function Changes

Not all package functions can be converted automatically. Youll need to address these issues manually.

Issue 5644: Unable automatically convert assign operation of array or global nested table,

Recommended action: Perform a manual conversion.

Issue code: 5644 | Number of occurrences: 1 | Estimated complexity: Medium

Schemas.DMS_SAMPLE.Packages.TICKETMANAGEMENT.Private functions.GET_OPEN_EVENTS: 270:297

Procedure Changes

Not all procedures can be converted automatically. Youll need to address these issues manually.

Issue 5103: Unable to convert hints

Recommended action: Use PostgreSQL methods for performance tuning.

Issue code: 5103 | Number of occurrences: 2 | Estimated complexity: Simple

Documentation references: http://www.postgresql.org/docs/9.6/static/geqo.html

Schemas.DMS_SAMPLE.Procedures.GENERATESEATS: 2041:2089
Schemas.DMS_SAMPLE.Procedures.GENERATE_TICKETS: 290:1184

AWS SCT data extractors

Extract data from your DW and migrate to Amazon Redshift

- Extracts data through local migration agents
- Data are optimized for Amazon Redshift and saved in local files
- Files are loaded to an Amazon Simple Storage Service (Amazon S3) bucket (through a network or AWS Snowball Edge) and then to Amazon Redshift

Vertica

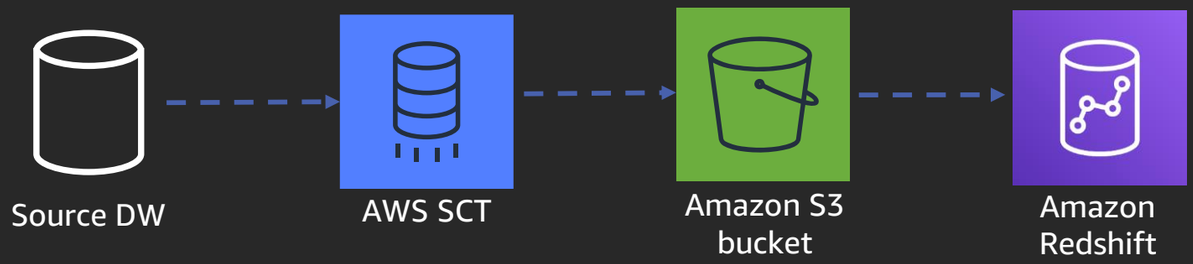
Oracle

Microsoft SQL Server

Greenplum

Teradata

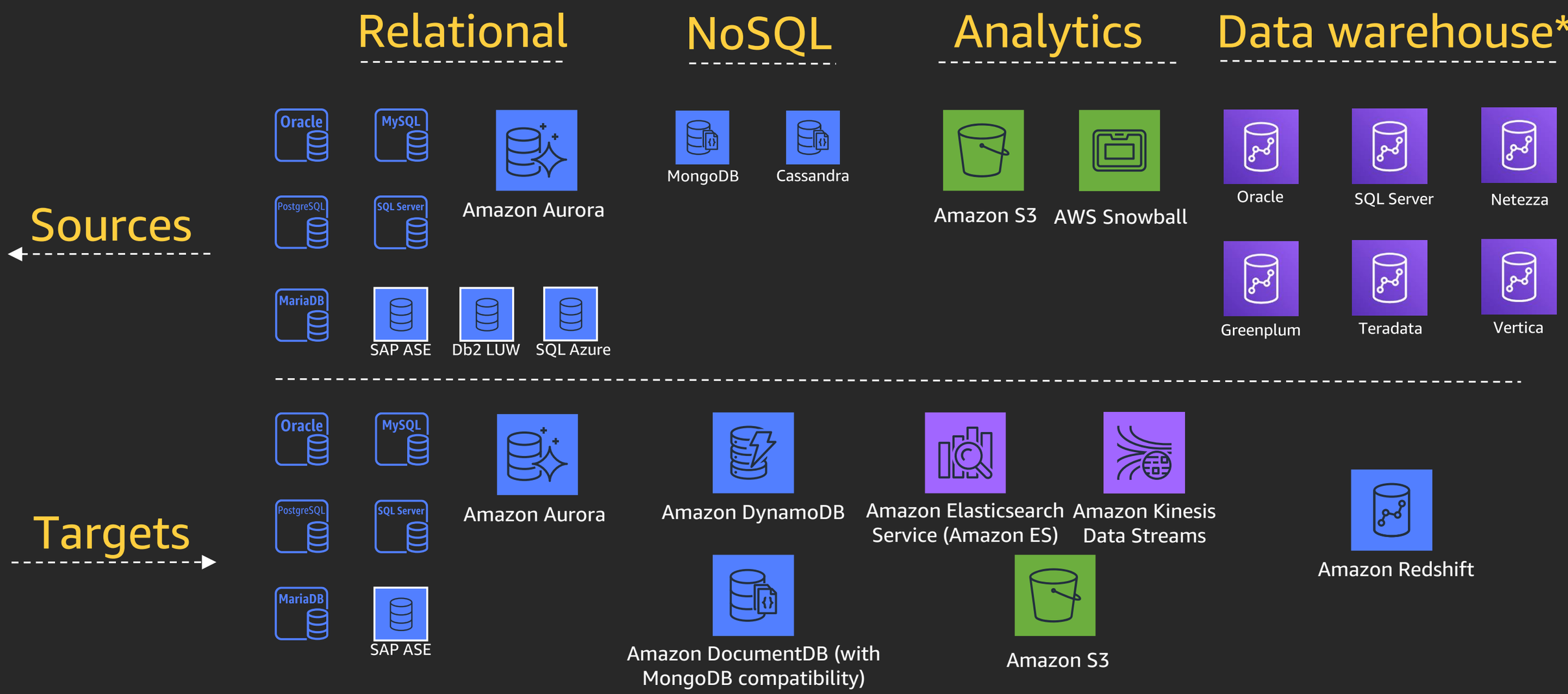
Netezza



Migrate

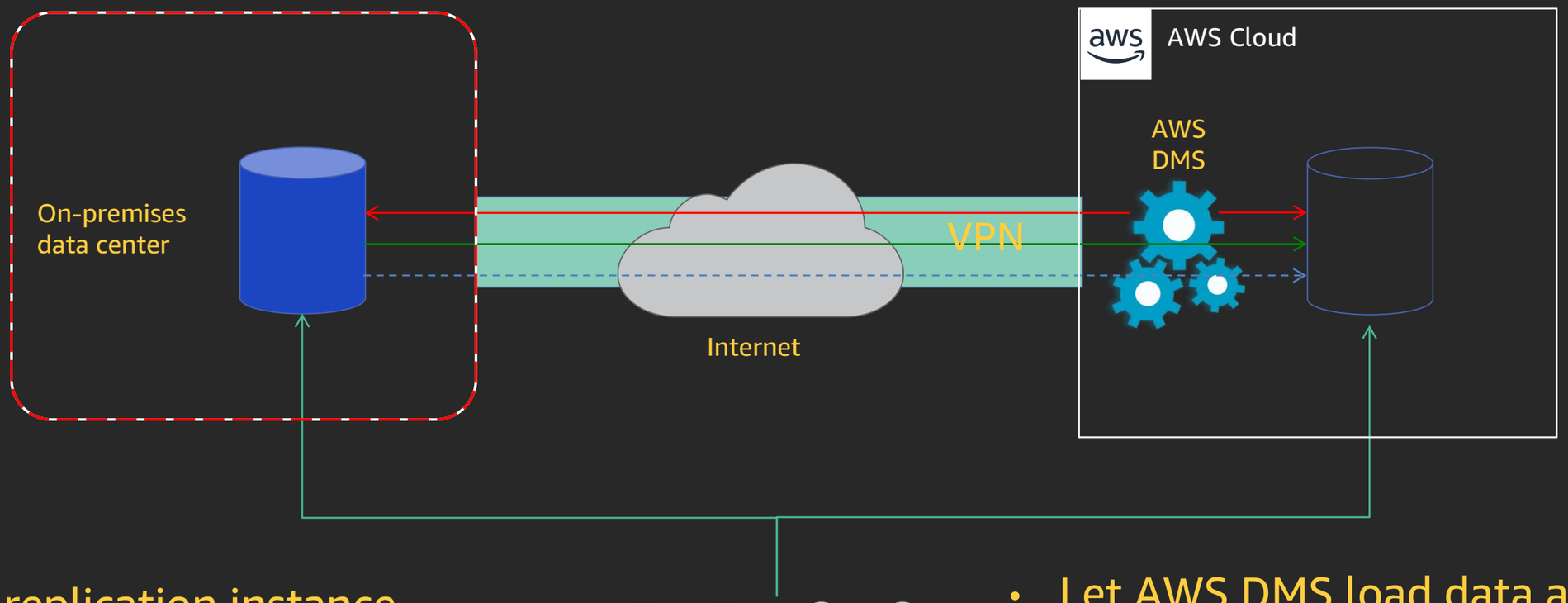
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Supported source and targets



* Supported via SCT data extractors

The data migration process



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

- Let AWS DMS load data and keep them in sync
- Once in sync, switch applications to the target at your convenience

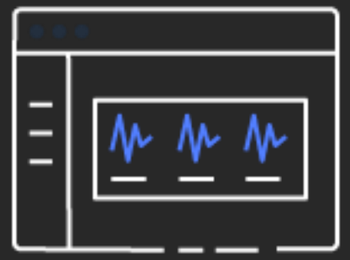
AWS DMS product highlights



Secure



Assess



Validate



Snowball integration



Monitor



Stream data

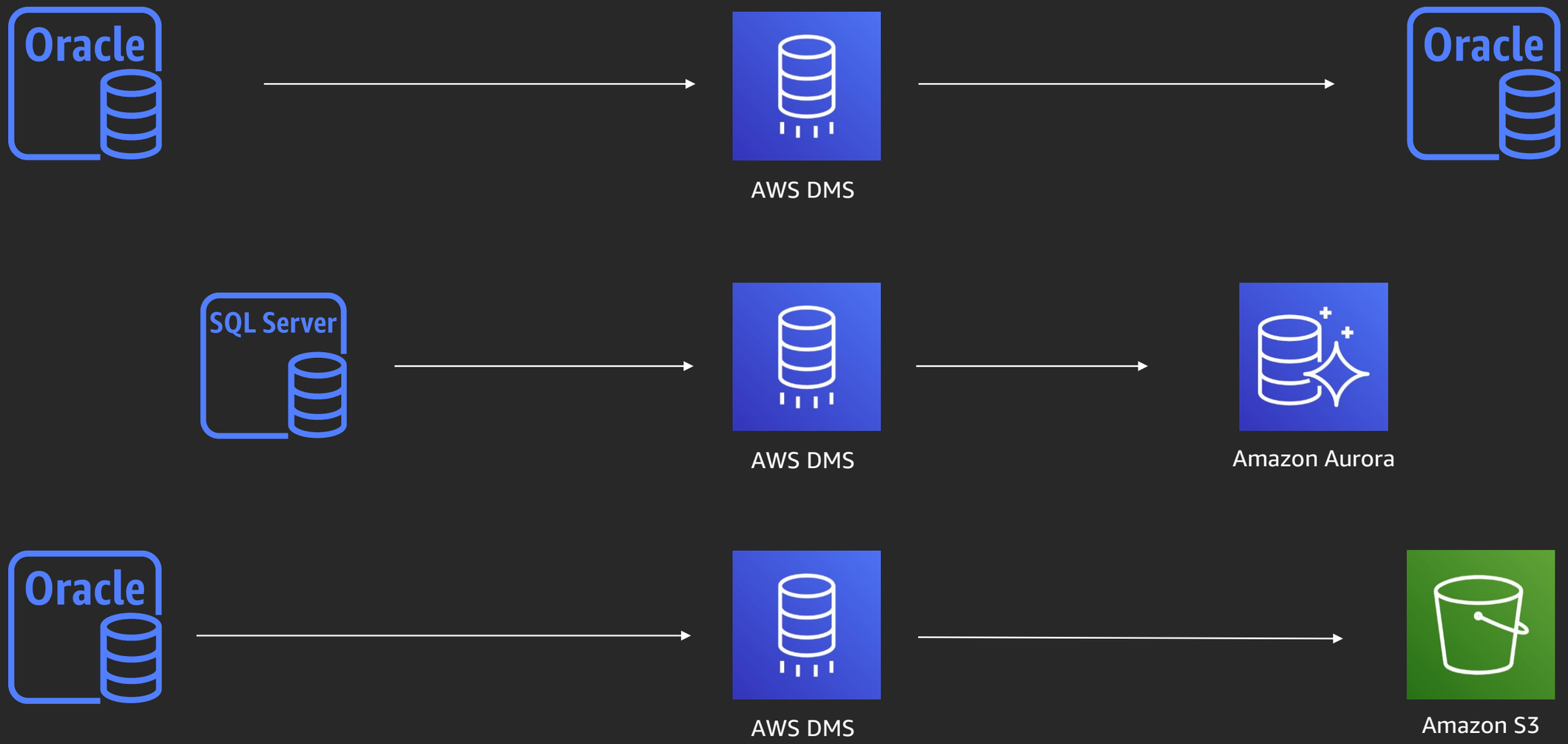


Low cost

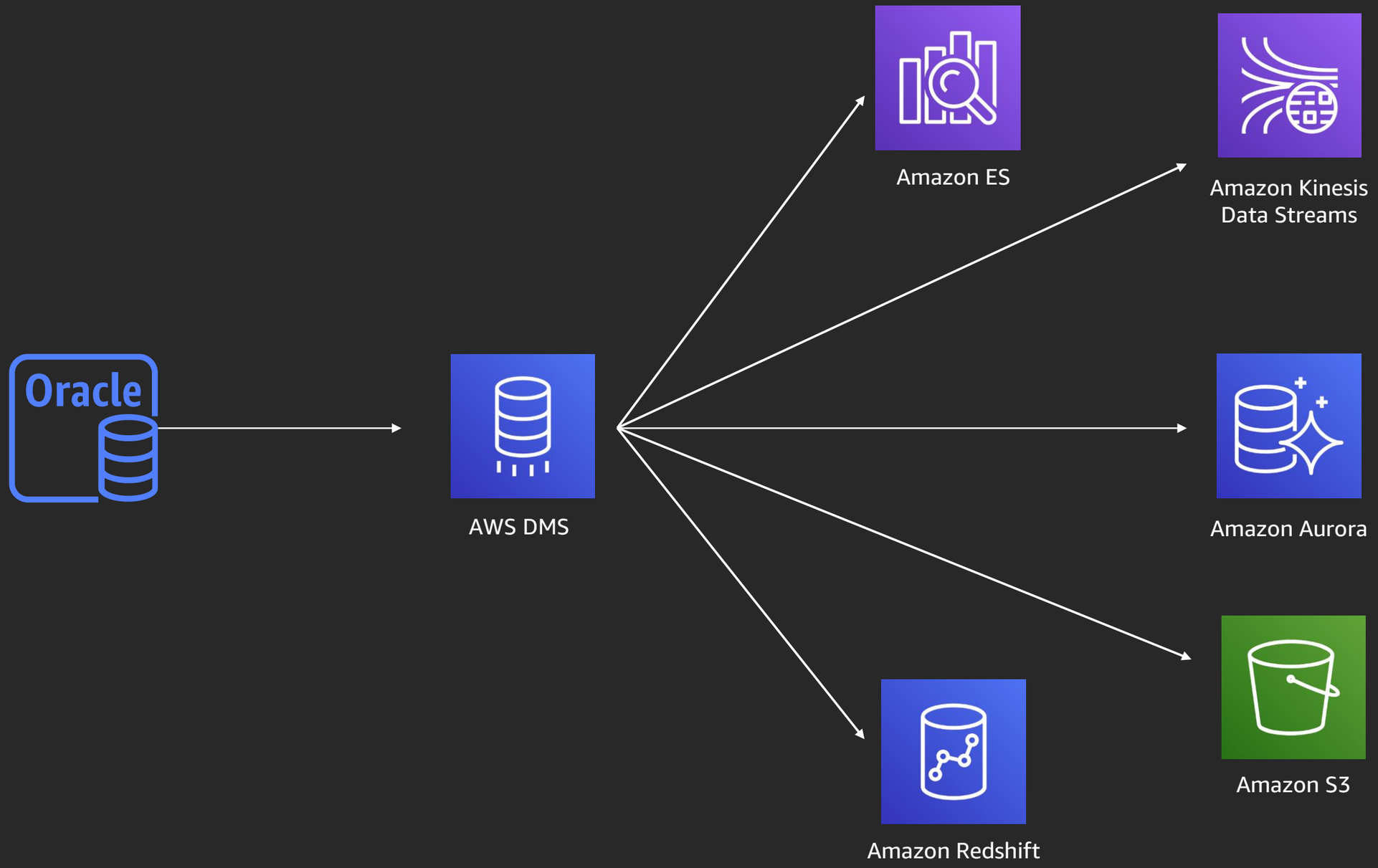


Multiple options

Homogenous or heterogeneous



Fan out: Microservices architecture



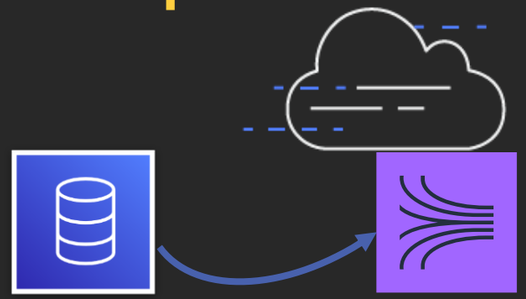
Use cases

Migrate



- Migrate business-critical applications
- Migrate data warehouses to Amazon Redshift
- Upgrade to a minor/major version
- Consolidate shards into Amazon Aurora
- Archive old data to Amazon S3
- Migrate from NoSQL to SQL, SQL to NoSQL, or NoSQL to NoSQL

Replicate



- Run your analytics in the cloud
- Hydrate your data lakes
- Create read replicas
- Replicate to streaming platforms

Demo

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Jobvite use case

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

About Jobvite

- Laser-focused on recruiting **past 20 years**
- **Industry pioneer** in social referrals, native recruitment marketing, Evolve Maturity Assessment, smart self-scheduler, benchmark reporting
- **Recognized market leader** by IDC and Forrester Research
- **2000 customers**
- **$200M+ recent investment**

©2019 Jobvite. All rights reserved

Powering people and company growth

100 million candidates

8 million career site users

300 thousand live jobs at any given time

2.5 million applications completed a month

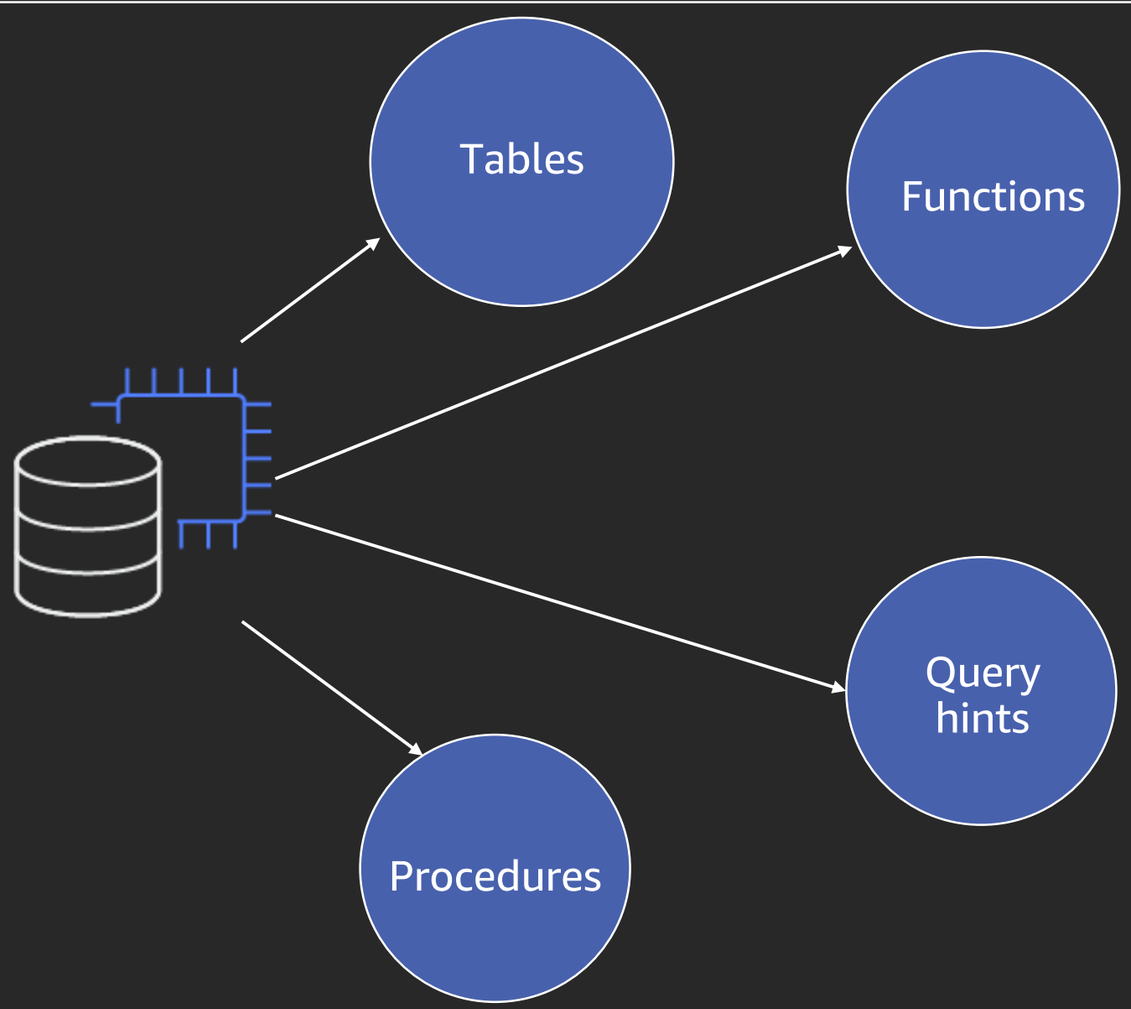
70 thousand new hires every month

©2019 Jobvite. All rights reserved

Agenda

- Datastore
- Drivers for migration
- Migration strategy
- Challenges with migration
- Lessons learned

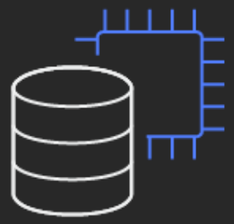
Datastore



Jobvite data

- Database size ~ 4 TB
- Data spread across ~ 250 tables
- 100 rows to 1.2 billion rows
- Partition views
- 1000’s of procedures, functions, queries etc.

Drivers for migration

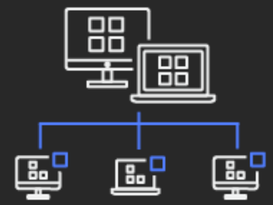


Infrastructure

- EC2 instances
- One way transactional replication
- Failure at different layers

Scalability

- Primary node as bottleneck
- Replication latency with DDL and bulk DML changes
- Load balance read traffic seamlessly





Cost optimization

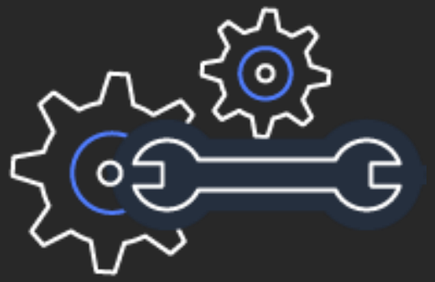
- Cost saving of thousands of dollars per month
- Overall cost reduction by 60-70%

Future ready

- Modernize your database tier
- Migrate your application code to ORM



Migration strategy



Schema migration



Performance



Rollback plan

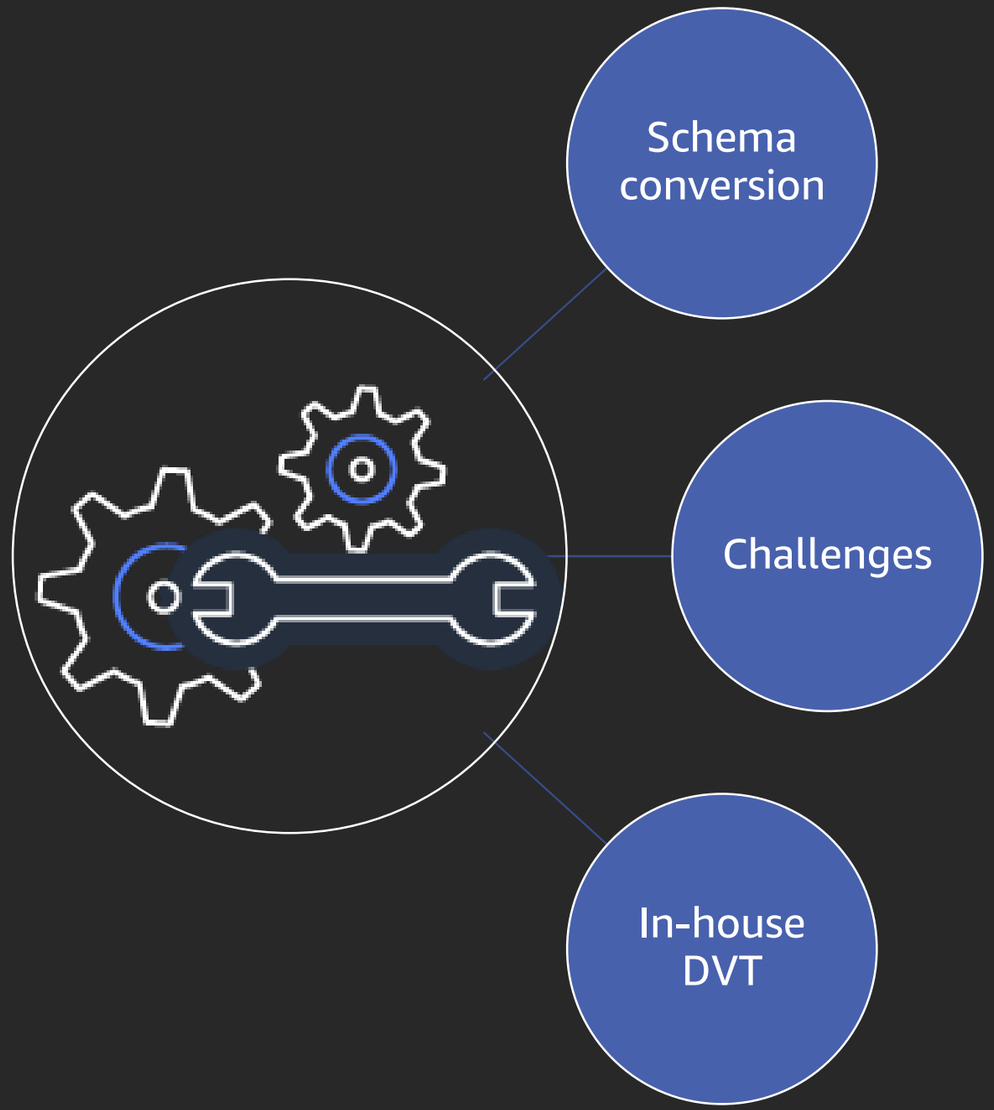


Data migration



Application migration

Schema migration



Schema conversion

- On premise schema conversion
- Knowledge base for team

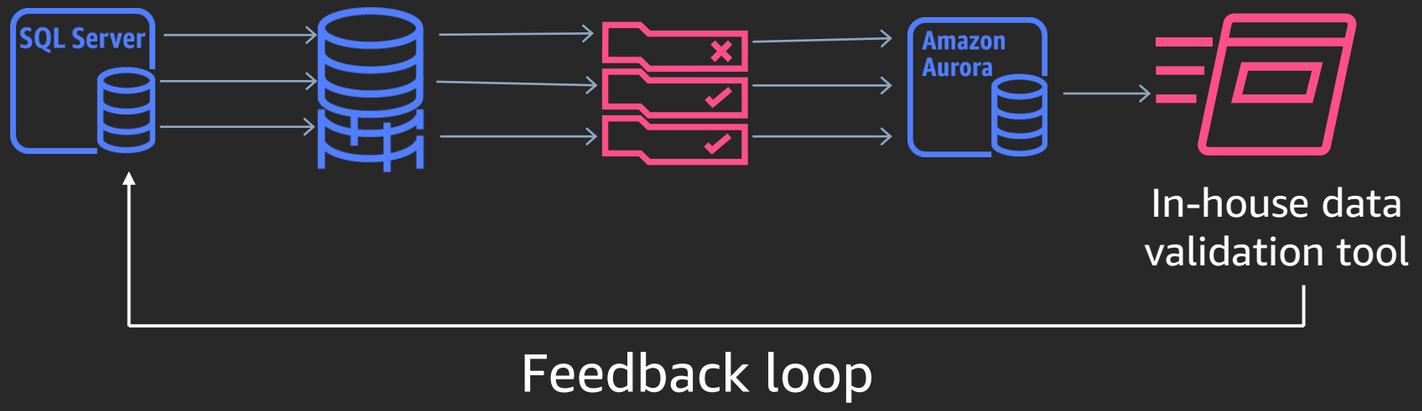
Challenges

- Charset and collation
- Default datetime precision
- Stored procedure, functions, and events
- Schema validation

In-house data validation tool

- Table validation
- Column validation
- Index, FKey, Constraints validation
- Row count, data comparison with defined sampling

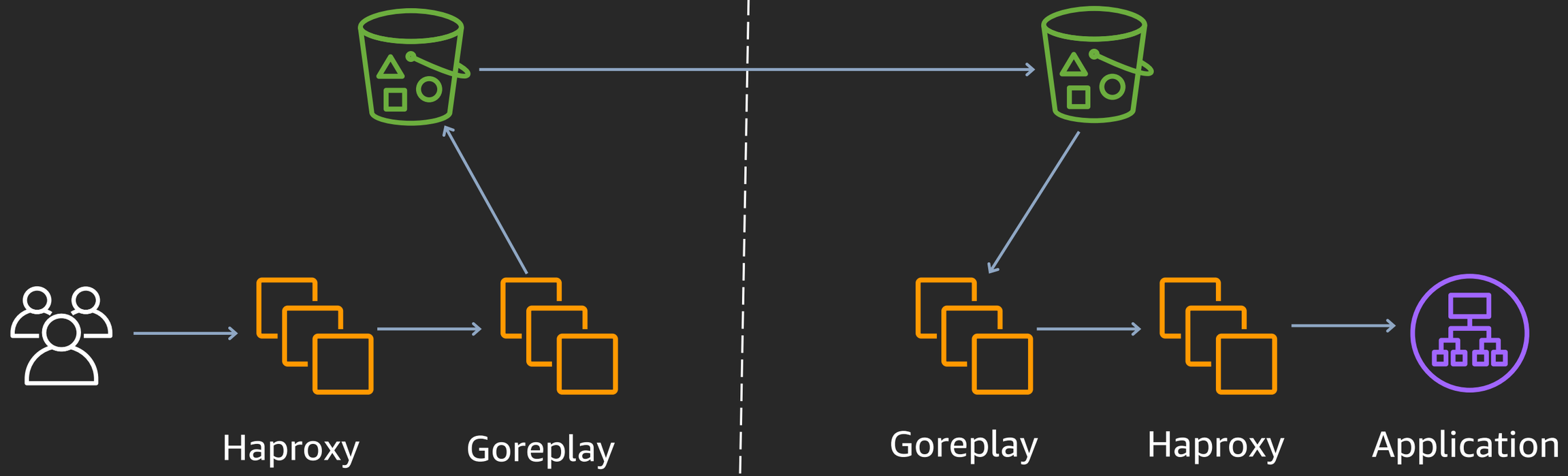
Data migration



Best practices

- Create separate task for table contains blob columns or large tables
- Use limited LOB with defined size instead of FULL LOB
- Enable task validation to ensure no errors with tasks
- Ensure no errors/warnings logged in awsdms_validation_failures_v1

Performance



Application migration







Challenges with Migration

- Missing default constraints
- Missing triggers
- Unsupported data types like XML
- Data type conversions issues (bit, datetime)
- Unsupported character/collation settings with MySQL 5.6
- Unsupported data validation for emoji characters
- Developed in-house tool to validate data
- Performance benchmarking
- Simulate production load

Lessons learned

- Difference in query optimization
- Understand your application data
- Trust but verify
- Simulate & amplify production traffic
- Things will not go per plan :)

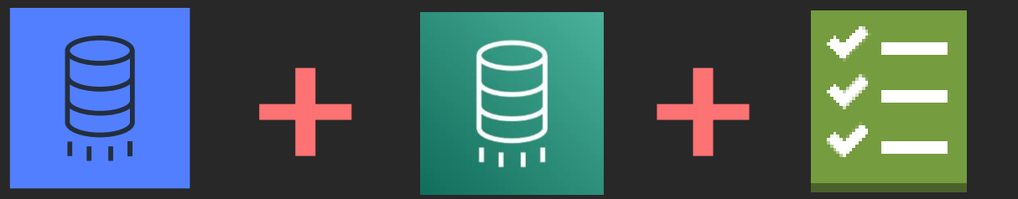
Migration resources

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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*

Schema + Data + Best practices



SCT + AWS DMS + Playbook

	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 Types	Index Types	Yes*
Link	B-Tree Indexes	B-Tree Indexes	Yes
Link	Composite Indexes	Multi-Column Indexes	Yes
Link	BITMAP Indexes	BRIN Indexes	Minimal
Link	Function-Based Indexes	Expression Indexes	Yes
Link	Global and Local Partitioned Indexes	Partitioned Indexes	Yes*
Link	Identity Columns	Serial Data Type	Yes*
Link	MVCC (Table & Row Locks)	MVCC (Table & Row Locks)	Yes*
Link	Character Sets	Encoding	Yes*
Link	Transactional Model	Transactional Model	Yes*

Database Freedom



Database Freedom is an AWS database and analytics modernization initiative focused on accelerating enterprise migrations from Oracle, Microsoft SQL Server, Teradata, Cassandra, or MongoDB platforms to AWS cloud-native database services

Oracle

Microsoft SQL Server

teradata.

cassandra

mongoDB.

AWS database migration partners

slalom

DATAPIPE

TriNimbus
an Onica company

logicworks

iTMethods.

USTGlobal®
Innovation • Information • Technology

Pythian
love your data®

CLOUDNEXA

KNOWARTH

BRLink

REĀN
CLOUD

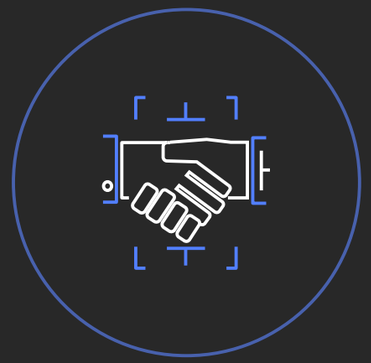
Minjar

TRIANZ Driving
Business Execution

BigData©
Systems

2ND WATCH

AWS IQ lets you find, engage, and pay AWS Certified third-party experts for on-demand work



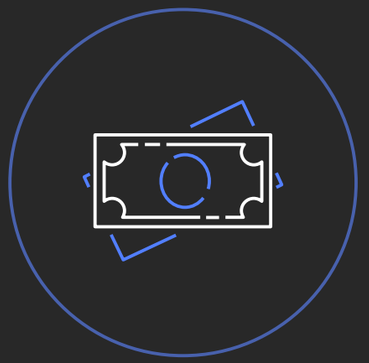
On-demand AWS Certified talent



Verified experts



Easy-access management



Integrated payments

aws.amazon.com/iq

Learn more at Suite 29-235 in the Venetian on Monday (3 – 4 PM) and Wednesday (1 – 2 PM)

Resources

Getting started guide: Review technical documentation

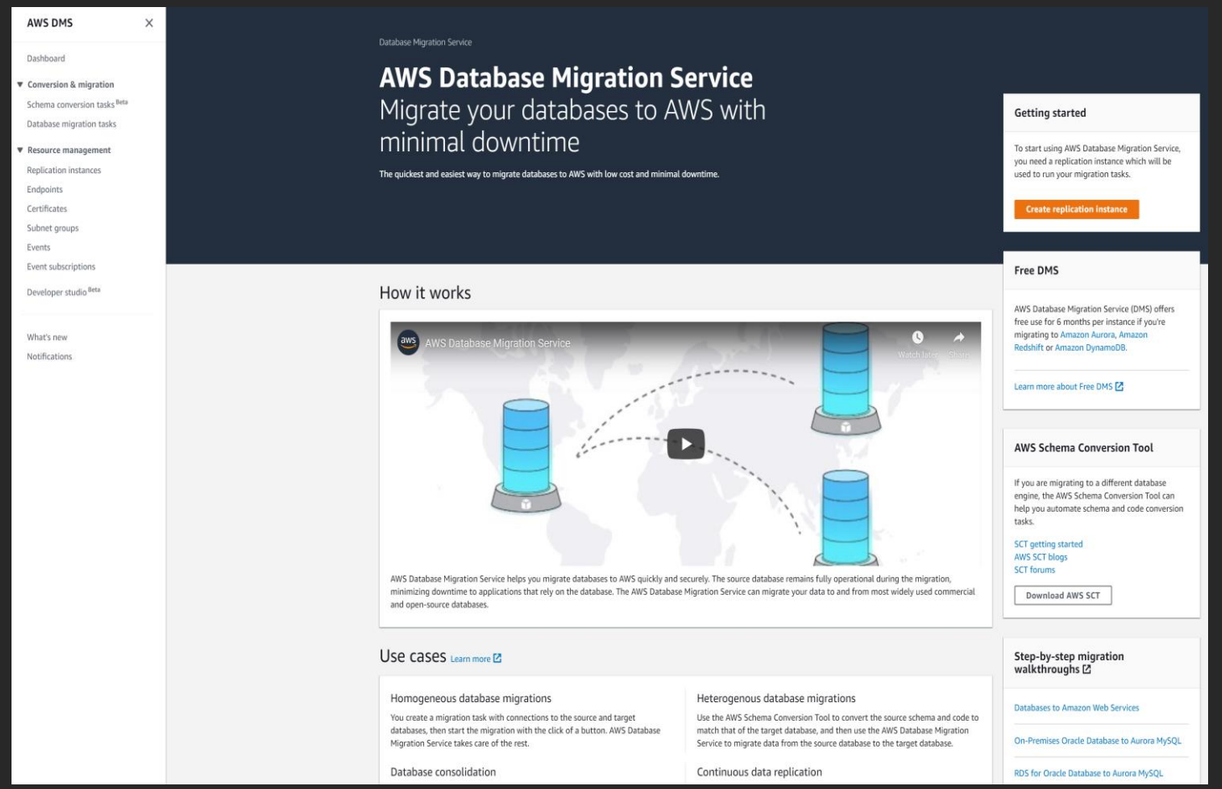
Features and benefits: Highlights DMS features

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 (AWS CLI): Start and stop replication tasks with simple commands

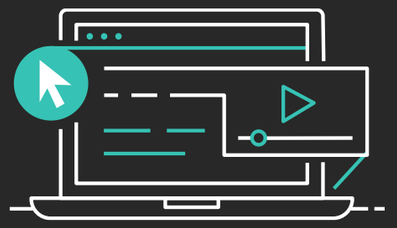


Q&A

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

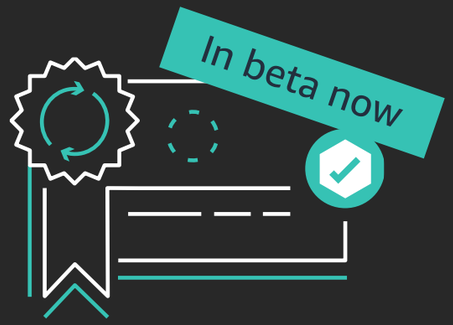
Learn databases with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate database skills



25+ free digital training courses cover topics and services related to databases, including:

- Amazon Aurora
- Amazon Neptune
- Amazon DocumentDB
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon Redshift
- Amazon RDS



Validate expertise with the new **AWS Certified Database - Specialty** beta exam

Visit aws.training

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws training and certification

Thank you!

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws



Please complete the session survey in the mobile app.

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.