AWS

SUMMIT

Migrating your Databases to AWS: Deep Dive on Amazon RDS and AWS Database Migration Service

Blair Layton, Business Development, Database Services, APAC

June 21, 2017

amazon webservices

Amazon RDS

Relational Databases

Managed Service

Simple and fast to scale

Fast, predictable performance

Low cost, pay for what you use

MySQL

Microsoft SQL Server

ORACLE

PostgreSQL

MariaDB

Amazon Aurora

Key Features

Provision a database in 6 minutes

Provision a MAZ database with a few mouse clicks

Scale a database up/down with 60-90 seconds downtime

Apply patches with 60-90 seconds downtime

Add read replicas with a few mouse clicks

Protect your backups and logs with 11 9’s of durability

Recover to any point in time from nightly backups + logs

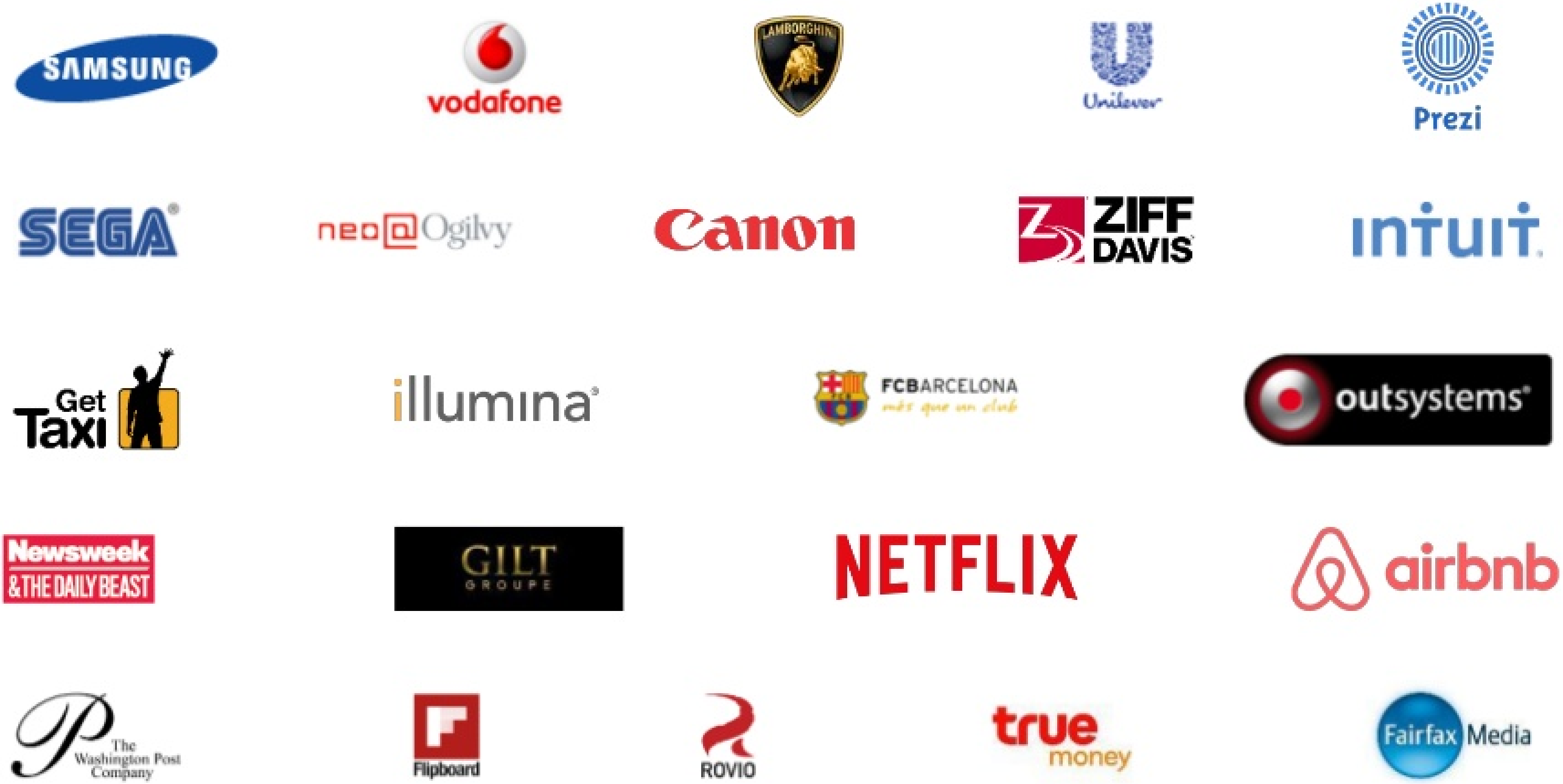
Detailed metrics, down to 1 second intervals

Secure your data with single click encryption at rest

Start/Stop your dev/test databases for up to 7 days

Amazon
RDS

Amazon RDS Customers



What is Amazon Aurora?

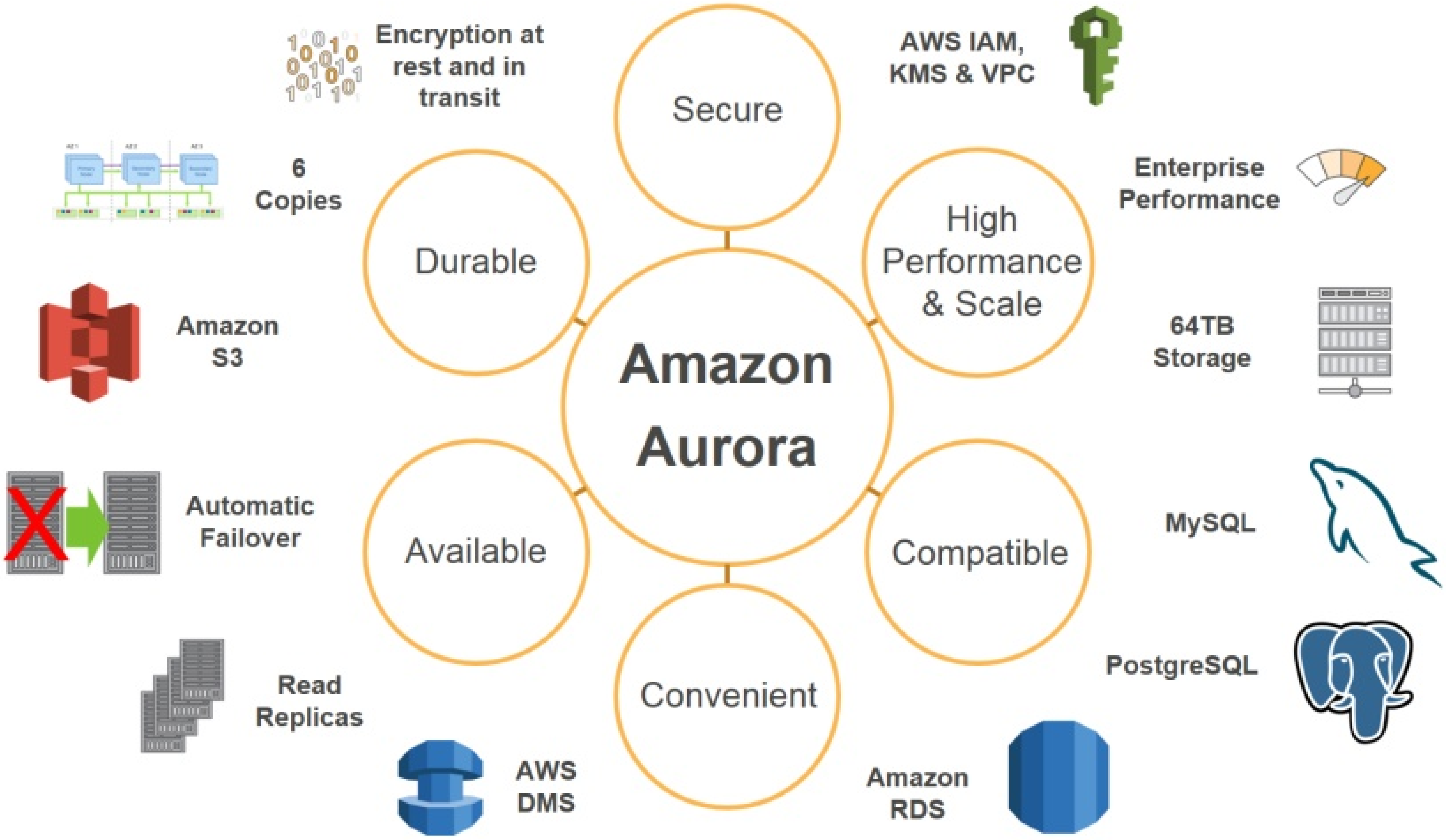
MySQL-compatible and PostgreSQL-compatible relational database platform

Performance and availability of commercial databases

Simplicity and cost-effectiveness of open source databases

Delivered as a managed service

The Amazon Aurora Database Family



Amazon Aurora

Secure

Encryption at rest and in transit

AWS IAM, KMS & VPC

High Performance & Scale

Enterprise Performance

64TB Storage

Compatible

MySQL

PostgreSQL

Convenient

Amazon RDS

AWS DMS

Available

Automatic Failover

Read Replicas

Durable

6 Copies

Amazon S3

Recent Announcements

Performance enhancements: Fast DDL, fast index build, spatial indexing, hot row contention

Availability features: Zero-downtime patching, database cloning (Q2), database backtrack (Q2)

Eco-system integration: Load from S3, IAM integration (Q2), select into S3 (Q2), log upload to CloudWatch Logs & S3 (Q2)

Cost reduction: t2.small – cuts cost of entry by half – you can run Aurora for $1 / day

Growing footprint: London, Montreal, Ohio, and San Francisco – now available in all 3AZ regions

2/3 of top 100 AWS customers

8 of top 10 gaming customers

Expedia™ GE Nasdaq

ticketmaster® TalentBin BY MONSTER ancestry.com

FANDUEL PEARSON Education INTERCOM

zynga igt DOUBLE DOWN gumi One Step Beyond.

Fastest growing service in AWS history

RDS Engine Capability Matrix

Feature	Aurora	MySQL	MariaDB	PostgreSQL	Oracle	SQL Server
VPC	✓	✓	✓	✓	✓	✓
High availability	✓	✓	✓	✓	✓	✓
Instance Scaling	✓	✓	✓	✓	✓	✓
Encryption	✓	✓	✓	✓	✓	✓
Read replicas	✓	✓	✓	✓	Oracle Golden Gate / DMS	DMS
Cross region replicas	✓	✓	✓	✓		
Max Storage	64 TB	6 TB	6 TB	6 TB	6 TB	4 TB
Scale Storage	Auto scaling	✓	✓	✓	✓	
Provisioned IOPS	NA	30,000	30,000	30,000	30,000	20,000
Largest Instance	R3.8XL	R3.8XL M4.10XL	R3.8XL M4.10XL	R3.8XL M4.10XL	R3.8XL M4.10XL	R3.8XL M4.10XL

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

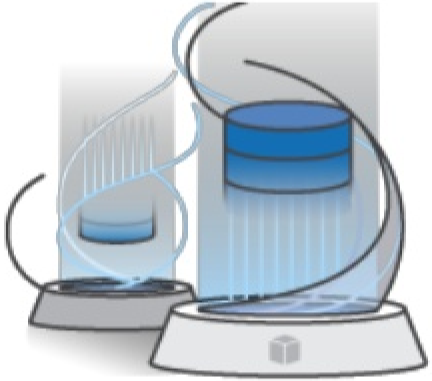
1. Create your AWS account
2. Setup your Virtual Private Cloud (VPC) in AWS
3. Connect to AWS with a VPN or Direct Connect
4. Shutdown and backup your database
5. Transmit the backup to S3
6. Configure an EC2 instance with the DB software
7. Restore the backup
8. Configure EC2 instances for the application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



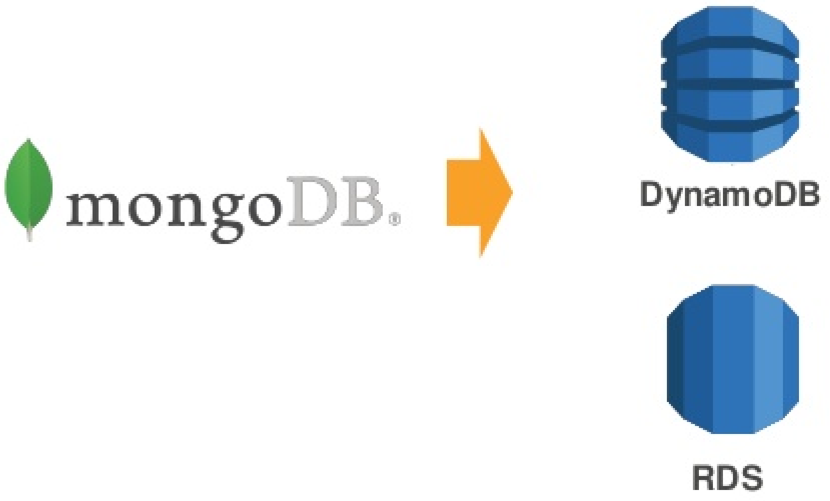
PostgreSQL ORACLE Amazon Aurora MariaDB

AMAZON REDSHIFT MySQL Microsoft SQL Server SAP ASE

New NoSQL support

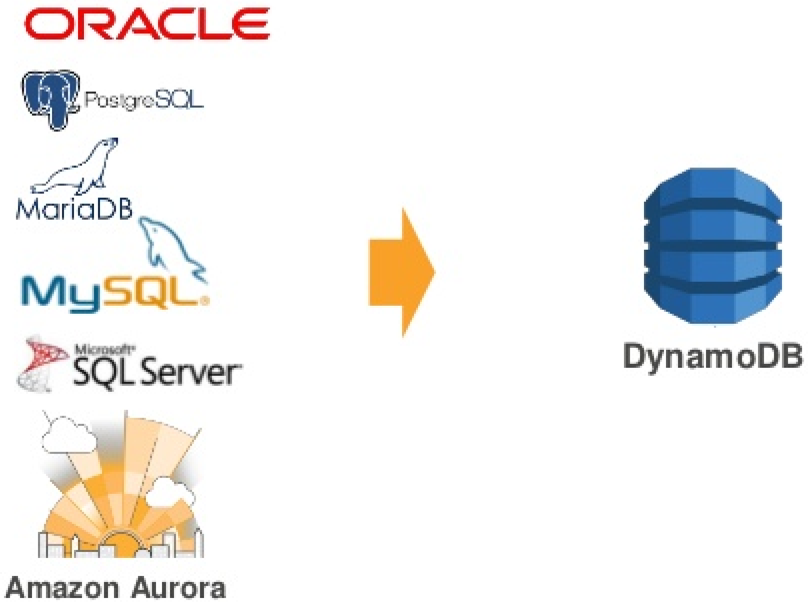
Migrate to AWS

- Move from MongoDB to Amazon DynamoDB
- Move from MongoDB to relational db's

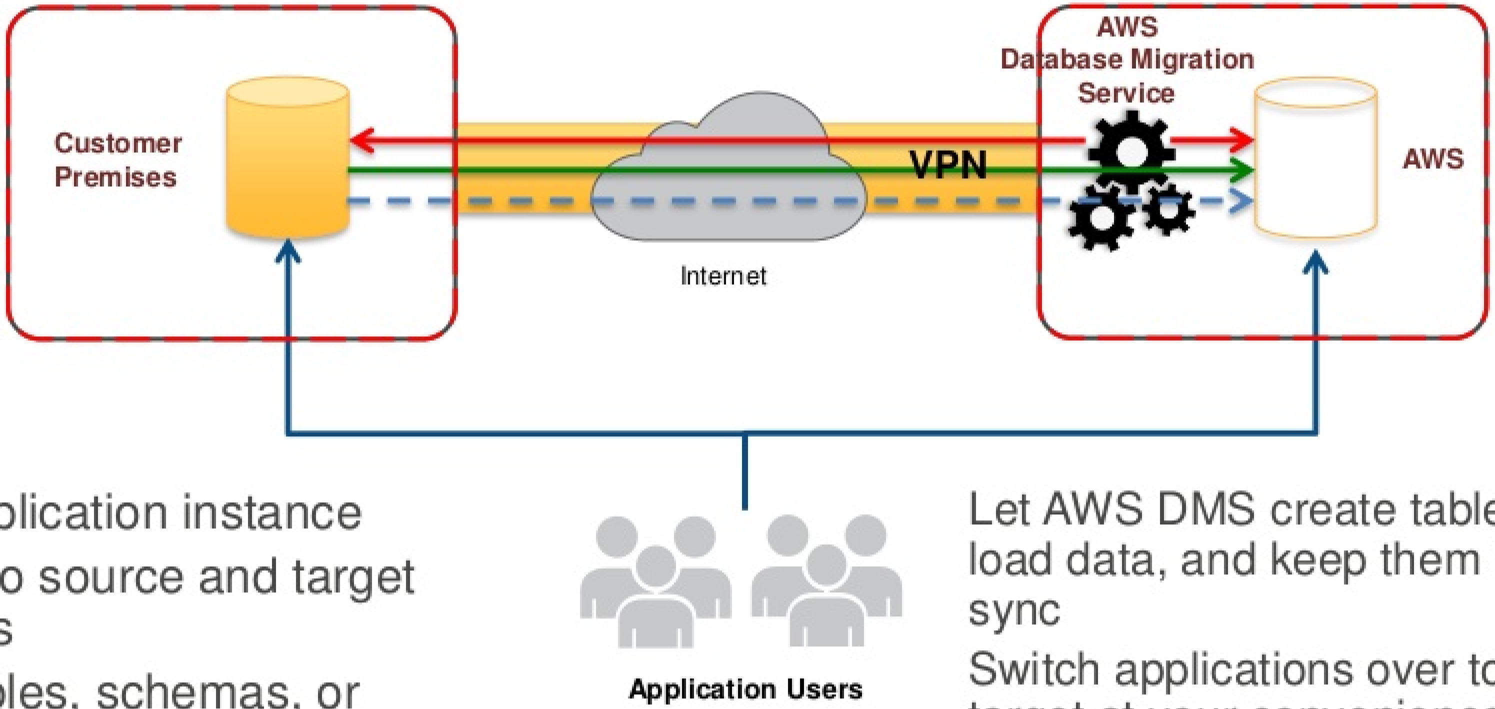


Move between NoSQL and SQL

- Change technologies



Keep Your Apps Running During the Migration



Start a replication instance

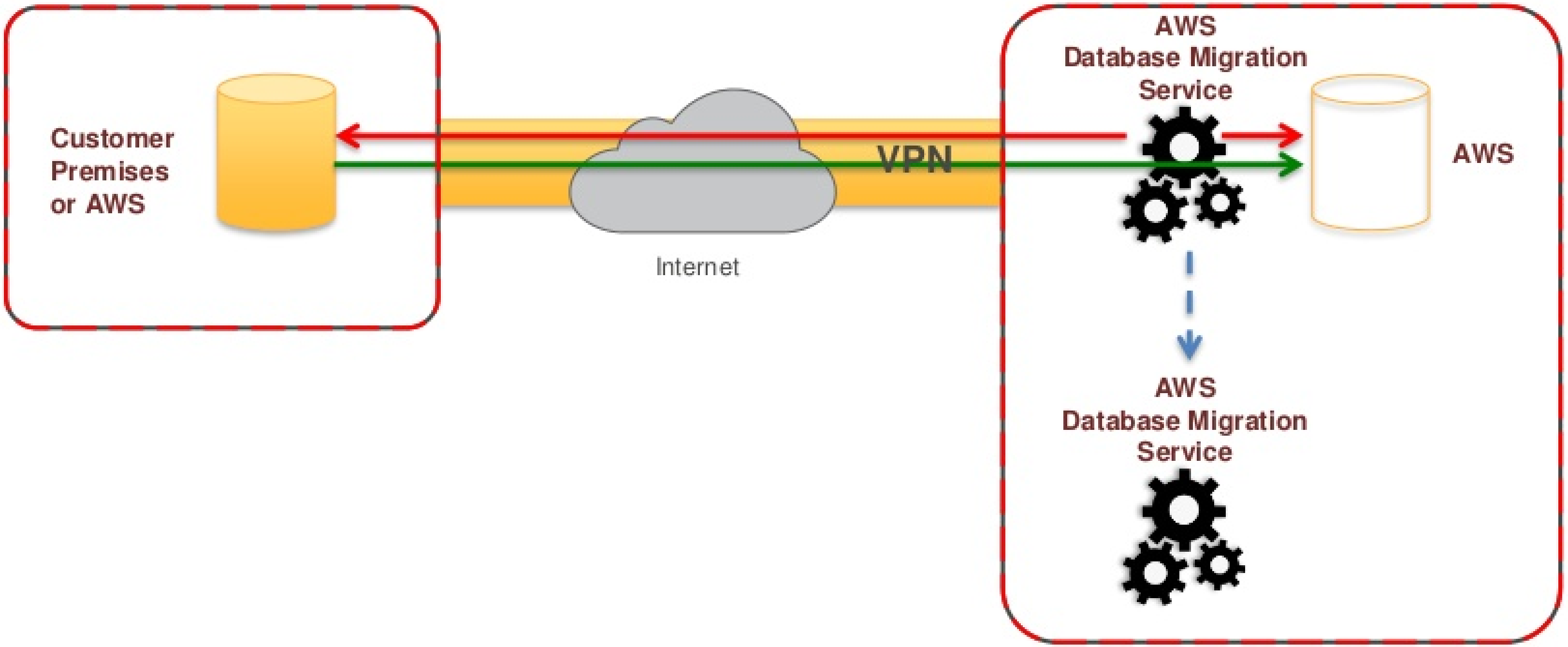
Connect to source and target databases

Select tables, schemas, or databases

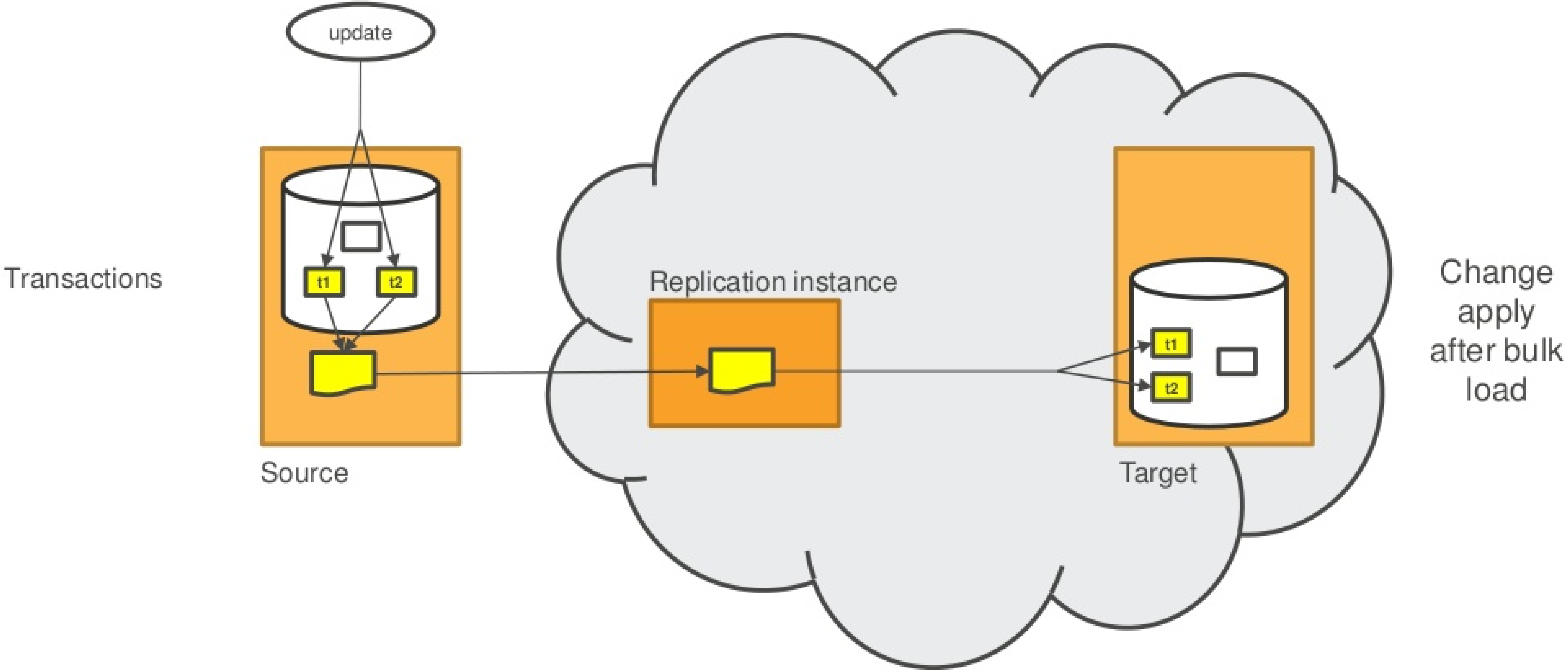
Let AWS DMS create tables, load data, and keep them in sync

Switch applications over to the target at your convenience

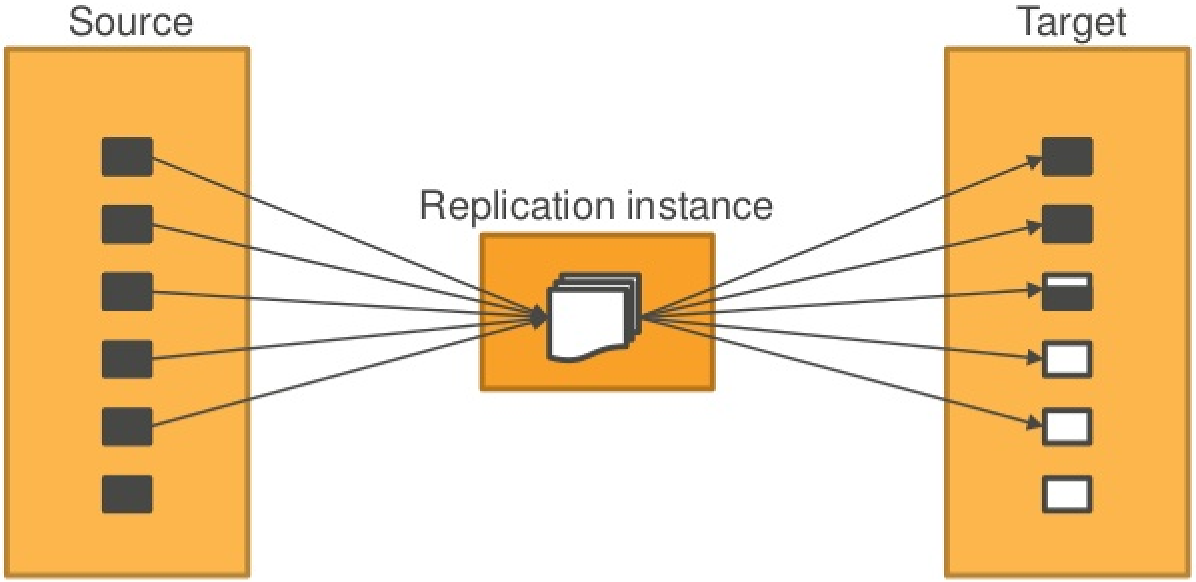
Multi-AZ Option for High Availability



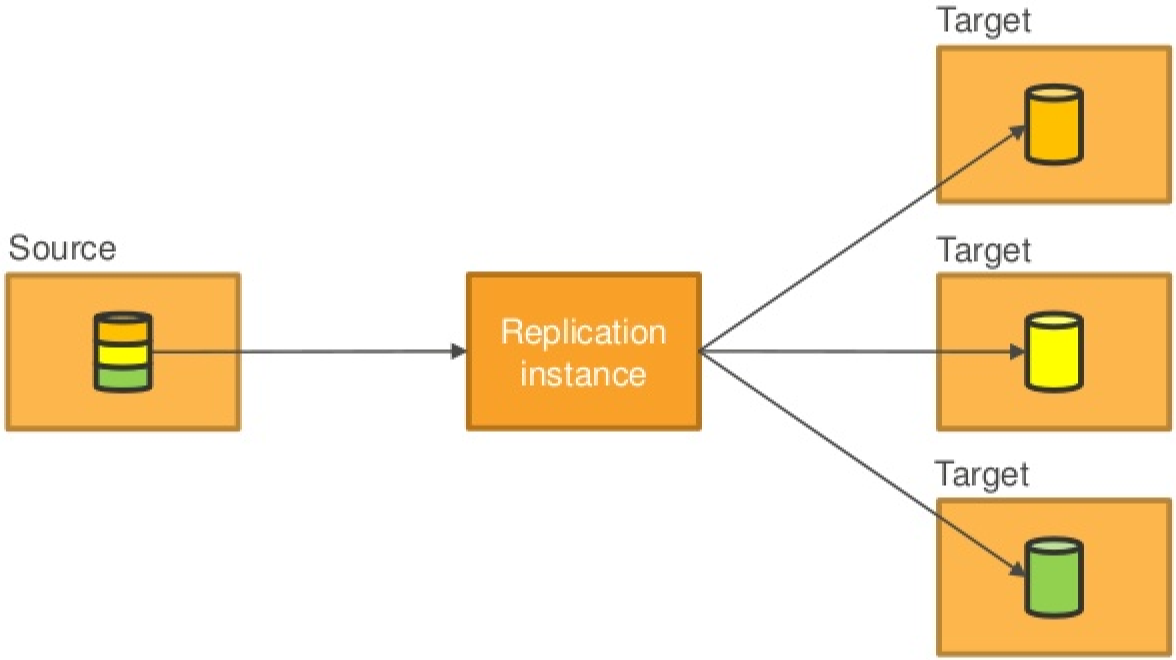
Change Data Capture (CDC) and Apply



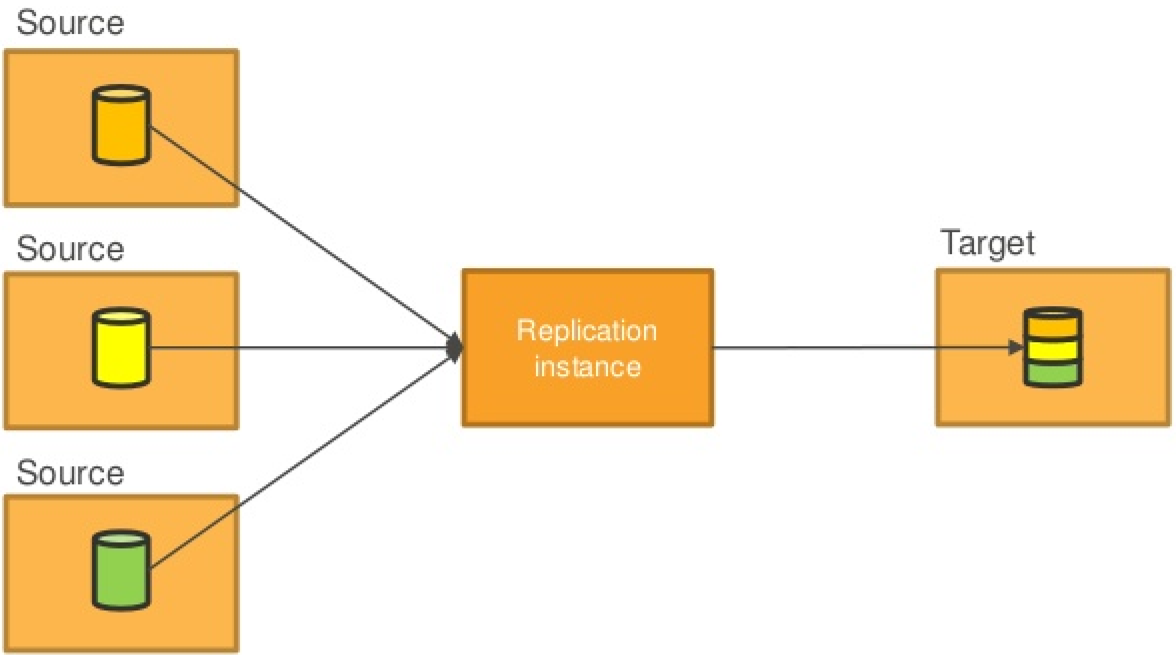
Load is Table by Table



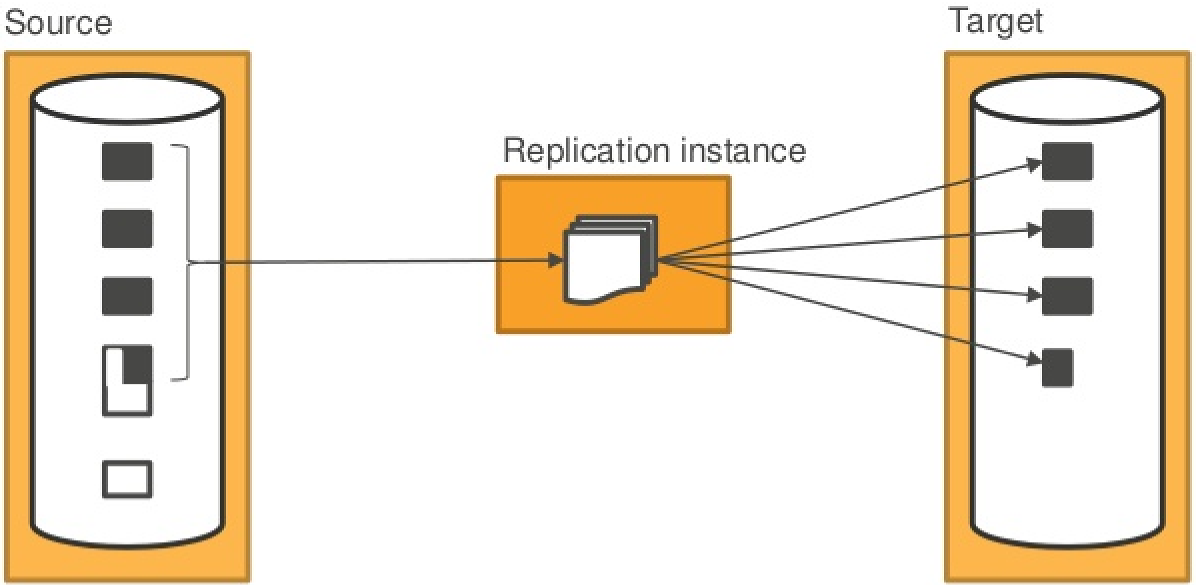
Multiple Targets



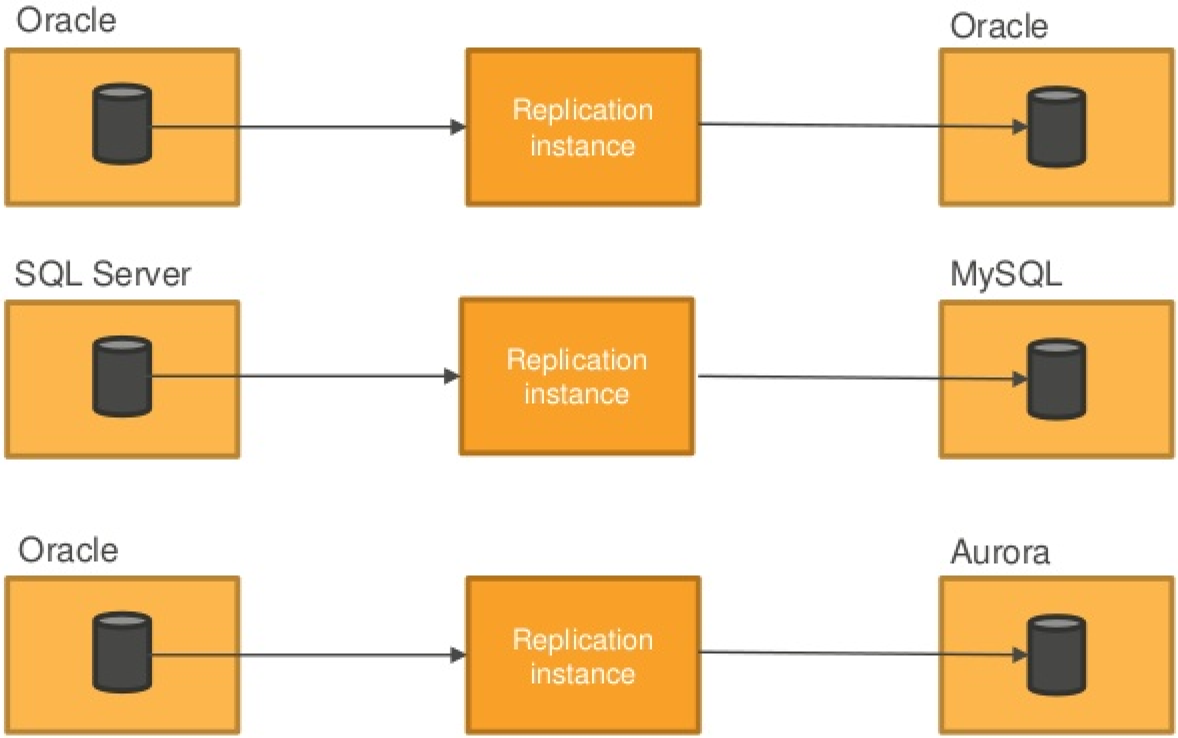
Multiple Sources



Customers Don’t Have to Take Everything



Homogenous or Heterogeneous



For less than $10 per TB!

Strengths and Focus Areas

DMS can act as a replication/migration Swiss Army knife, but is not a magic wand.

Use It

- Heterogeneous migrations
- Minimal downtime required
- No native solution

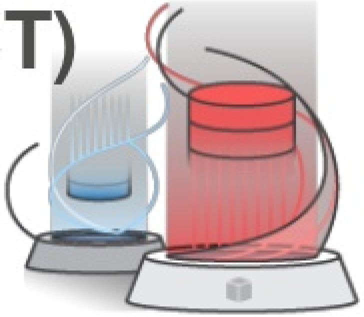
Caution

- Some tables with large LOBs
- Complex data types
- High load database

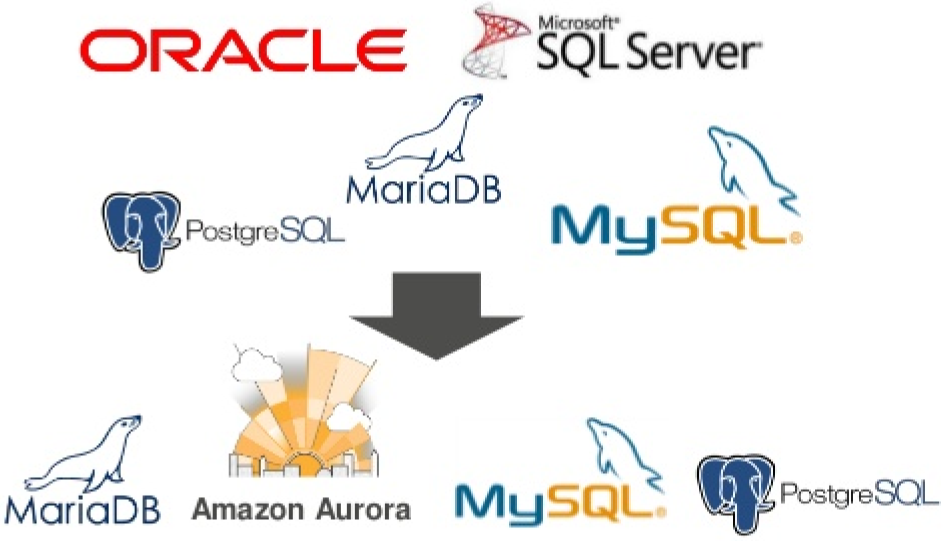
Don't Use It

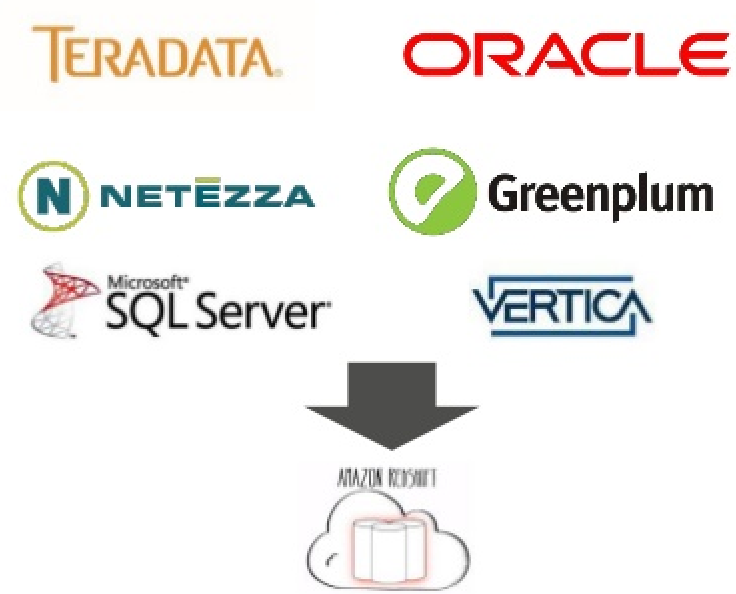
- Native no downtime solution exists
- Can take downtime + native
- > 5 TB + slow Internet****

AWS Schema Conversion Tool (AWS SCT)



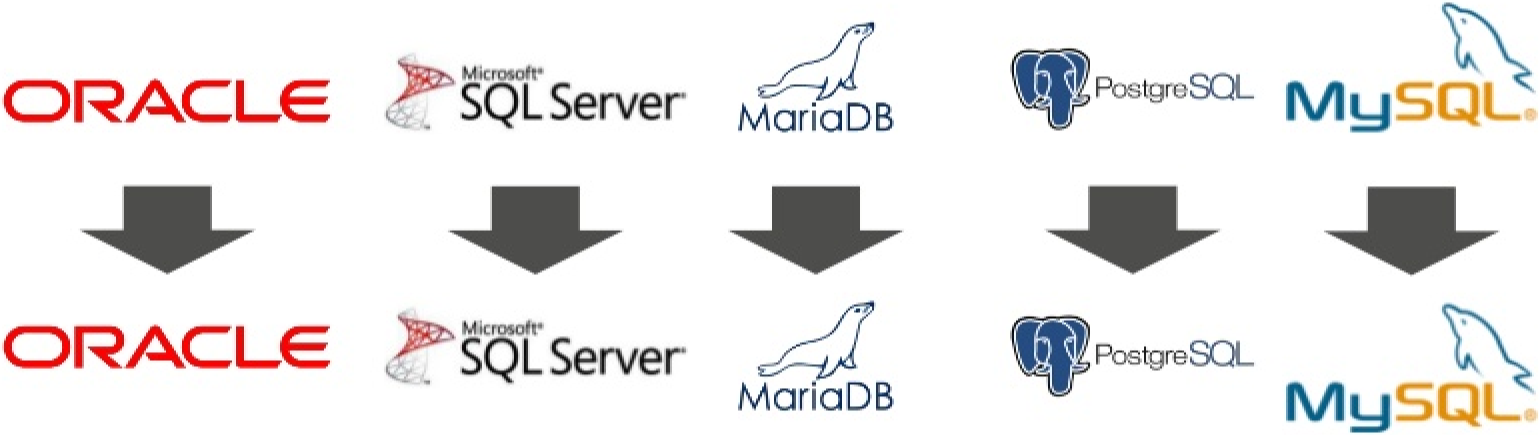
SCT helps automate many database schema and code conversion tasks when migrating between database engines or data warehouse engines





AWS Schema Conversion Tool

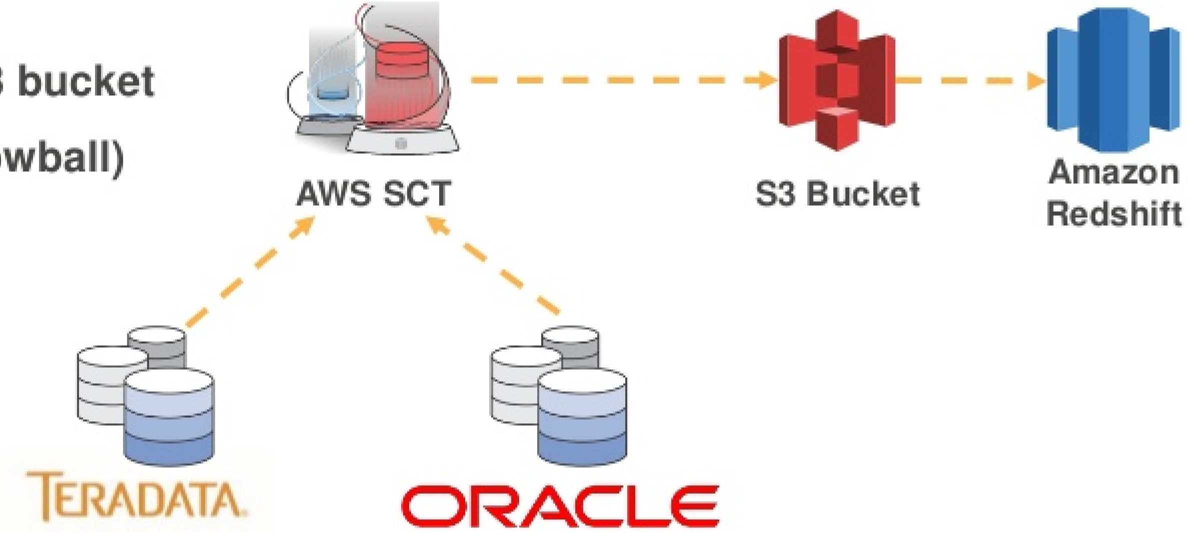
Supports Schema Copy and RDS Recommendations



SCT Data Extractors

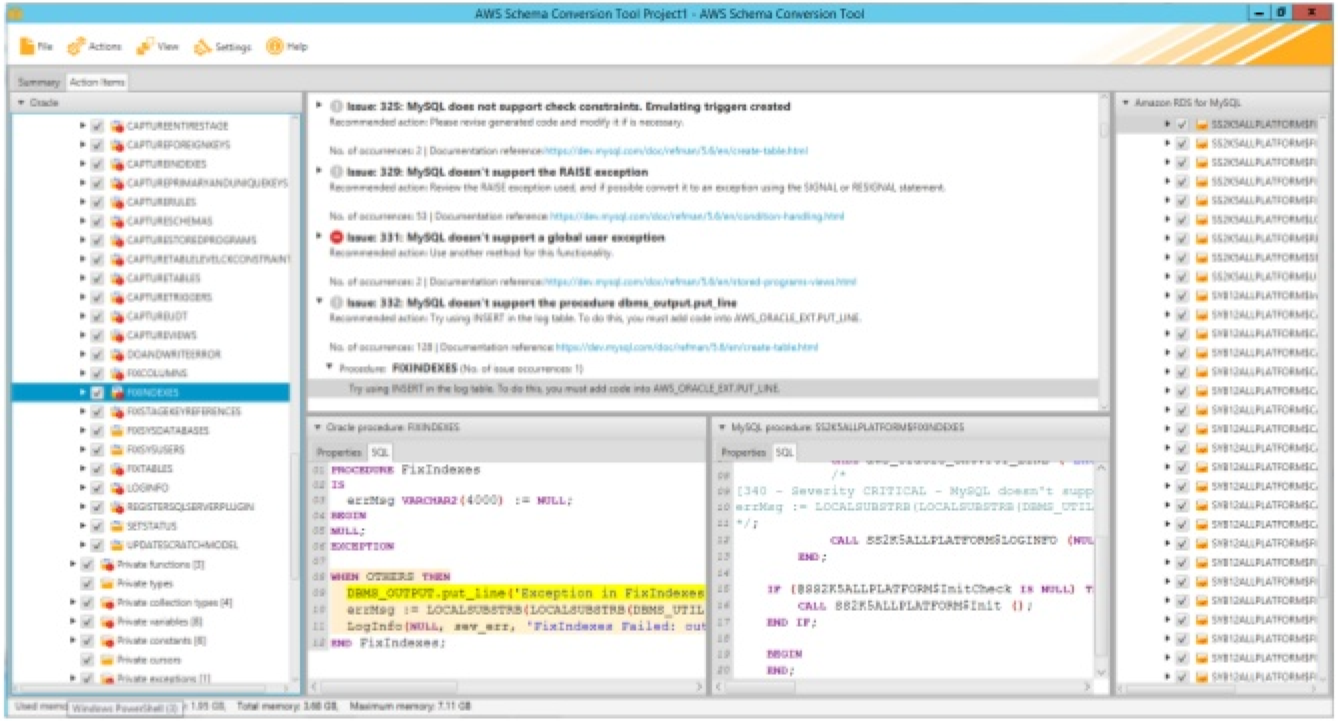
Extract Data from your data warehouse* and migrate to Amazon Redshift

- Extracts through local migration agents
- Data is optimized for Redshift and Saved in local files
- Files are loaded to an Amazon S3 bucket (through network or Amazon Snowball) and then to Amazon Redshift



* 1st release supports Oracle v11 and up and Teradata v14 and up

SCT Helps with Converting Tables, Views & Code



Issue: 325: MySQL does not support check constraints. Emulating triggers created
Recommended action: Please revise generated code and modify it if is necessary.

No. of occurrences: 2 | Documentation reference:https://dev.mysql.com/doc/refman/5.6/en/create-table.html

Issue: 329: MySQL doesn't support the RAISE exception
Recommended action: Review the RAISE exception used, and if possible convert it to an exception using the SIGNAL or RESIGNAL statement.

No. of occurrences: 53 | Documentation reference: https://dev.mysql.com/doc/refman/5.6/en/condition-handling.html

Issue: 331: MySQL doesn't support a global user exception
Recommended action: Use another method for this functionality.

No. of occurrences: 2 | Documentation reference:https://dev.mysql.com/doc/refman/5.6/en/stored-programs-views.html

Issue: 332: MySQL doesn't support the procedure dbms_output.put_line
Recommended action: Try using INSERT in the log table. To do this, you must add code into AWS_ORACLE_EXT.PUT_LINE.

No. of occurrences: 128 | Documentation reference: https://dev.mysql.com/doc/refman/5.6/en/create-table.html

Procedure: **FIXINDEXES** (No. of issue occurrences: 1)

Try using INSERT in the log table. To do this, you must add code into AWS_ORACLE_EXT.PUT_LINE.

Oracle procedure: FIXINDEXES

```
PROCEDURE FixIndexes
IS
  errMsg VARCHAR2(4000) := NULL;
BEGIN
NULL;
EXCEPTION

WHEN OTHERS THEN
  DBMS_OUTPUT.put_line('Exception in FixIndexes
  errMsg := LOCALSUBSTRB(LOCALSUBSTRB(DBMS_UTIL
  LogInfo(NULL, sev_err, 'FixIndexes Failed: out
END FixIndexes;
```

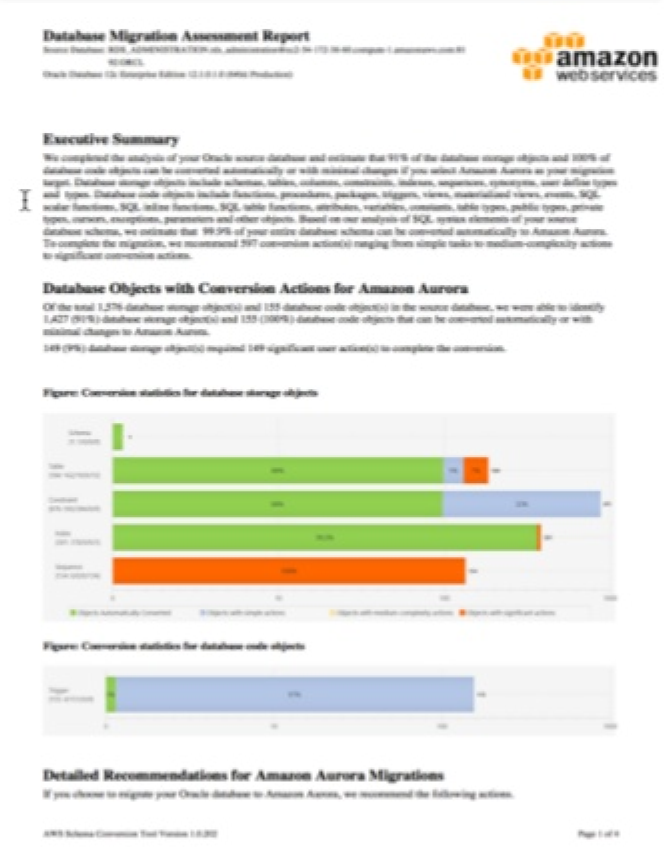
MySQL procedure: SS2K5ALLPLATFORM$FIXINDEXES

```
                /*
[340 - Severity CRITICAL - MySQL doesn't supp
errMsg := LOCALSUBSTRB(LOCALSUBSTRB(DBMS_UTIL
*/;
                CALL SS2K5ALLPLATFORM$LOGINFO (NUL
            END;

        IF (@SS2K5ALLPLATFORM$InitCheck IS NULL) T
            CALL SS2K5ALLPLATFORM$Init ();
        END IF;

        BEGIN
        END;
```

Used memo Windows PowerShell (3) : 1.95 GB, Total memory: 3.88 GB, Maximum memory: 7.11 GB

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

SCT can tell you how hard the migration will be



Database Migration Assessment Report

Source Database: RDS_ADMINISTRATION.rds_administration@ec2-54-172-56-60.compute-1.amazonaws.com:81 92:ORCL

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 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

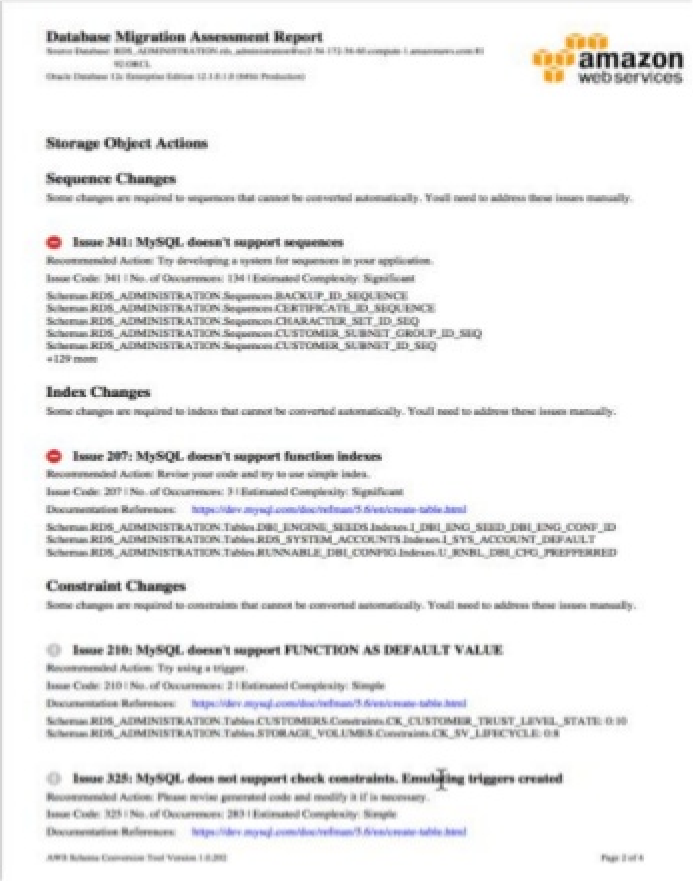
Detailed Recommendations for Amazon Aurora Migrations

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

AWS Schema Conversion Tool Version 1.0.202

Page 1 of 4

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@ec2-54-172-56-60.compute-1.amazonaws.com:81 92:ORCL

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 indexs that cannot be converted automatically. You'll 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. 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 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

Pricing and Terms and Conditions

$0

for software license

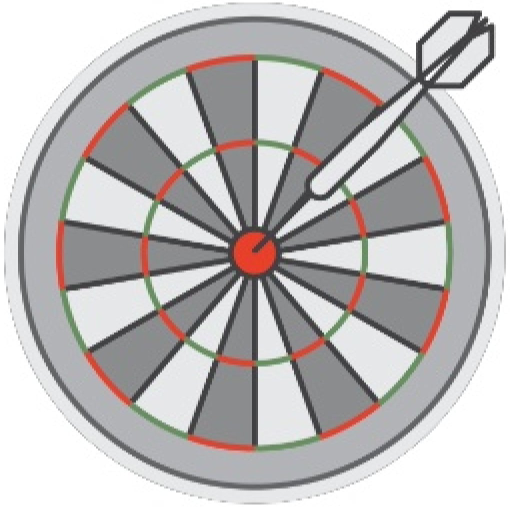
Pricing

- 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

Customer Successes



Split migration

shaadi.com®

The World's Largest Matrimonial Service

35 million members on it’s site

Many interdependent applications built over the last 15 years with unique High IOPS requirements

Migration from legacy cloud service to AWS

Combination of migration to MySQL on EC2 and Amazon Aurora

Benefits:

- DMS reduced the time required to migrate our databases by 40 percent
- Realized 55 percent cost savings by moving some db’s to Amazon Aurora

Case Study – Migration from Microsoft SQL Server to RDS PostgreSQL

Uberfusion Case Study – Data Migration for SSO Media

Company Profile

- ASEAN Media Company
- **4.7 million** residential customers
- Approx. **66% penetration** of TV households
- **183 TV** channels including **53 HD** channels

~37

Integrated Applications
(June 2016)

«

On Premises Database

MSSQL Server 2008 R2 Enterprise Edition

Database Stats (April 2016):

- Number of Tables: 102
- Database Size: ~ 55GB
- Server Specs: 128GB RAM, Intel Xeon CPU with 24 Cores

»

>2 million

Approximate Registered SSO Users
(June 2016)

Ideal State

- database in AWS **caters for SSO's migration** to the cloud
- **able to manage** current on-premises database load
- **able to scale** with the application
- must be **highly available**
- **maintain cost efficiency** while performing all of the above

Problem Statement, Mandate and Challenge

State of On-premise MSSQL Sever Database

Limited Scalability

As the on-premise MSSQL Server Database is a physical machine, scaling requires purchasing and installing additional hardware.

Higher Costs

Hardware upgrades are costly with a high hardware delivery turn-around time. Additionally, MSSQL Server Enterprise Edition licenses are high in costs.

Shared Infrastructure

The on-premises database infrastructure is shared across multiple applications. A problem caused by another application could potentially affect SSO.

New Mandate for SSO

To migrate the on-premise SSO system into Amazon Web Services (AWS) cloud to reduce cost and improve scalability.

Urgency

To be ready to handle high load during Euro and Olympics 2016

Migration into the Cloud

No data loss and completed in a short time frame at minimal costs

The decision was made to go with RDS PostgreSQL, as:

- MSSQL on RDS does not support Multi Availability Zone in AWS Singapore and hence cannot be highly available
- No need to maintain a VPN tunnel back to premise
- Lower OPEX costs compared to running MSSQL on RDS or managing our own DB servers

~11.5x cheaper

Compared to RDS MSSQL Server Enterprise

The Challenge

Zero Data Loss

Zero Data Loss
All user-related data needed to be migrated with zero data loss up to the point of migration.

Transparent

Transparent to Users
There should be no visible impact to SSO users. Encrypted data such as passwords needed to be migrated seamlessly.

Time

Short Migration Window
Data migration had to be completed within a 90 minute window.

Legacy

Legacy Data
There is a lot of legacy data in the on-premises database which is not clean. Dummy data and special characters exist where they should not.

Managed Service

Leverage Managed Services
To leverage Amazon's relational database service (RDS), instead of managing our own database servers.

Testing

System Needed to Be Testable
System needed to be testable on migration night by a small team of testers while ensuring the general public cannot change the data.

The Solution

Uberfusion's Delivery for Data Migration for SSO System

Challenge: Zero Data Loss

- All user-related data tables were identified
- System was put into read-only mode before starting migration
- Views were created in the schema of the new PostgreSQL database, and exported to flat files
- Number of records were verified for pre- and post-migration
- Data was verified for accuracy from a sample of users post-migration

Amazon RDS PostgreSQL Instance TXT

Total Migration Time

<1hr

Challenge: Transparent to Users

- PostgreSQL Membership library was modified to mimic ASP.NET Membership in regards to password encryption
- Migration was tested beforehand to ensure users would still be able to login with their existing passwords

PostgreSQL Instance Microsoft .net

Challenge: Short Migration Window & Legacy Data

- Data cleansing was performed
- Instance created with access to both source and target database for ease of migration
- Export scripts automated for speed and accuracy
- Multiple scripts to cater for special characters in data
- Logs offloaded from database to ElasticSearch and Kibana

Amazon Kinesis Firehose Amazon Elasticsearch Service Kibana

Data size reduced to

~1.5GB

Challenge: Testing

- Services were disabled pre-DNS migration for the SSO System
- Whitelist for specific users were created for testing and verification of data

Challenge: Leverage Managed Service

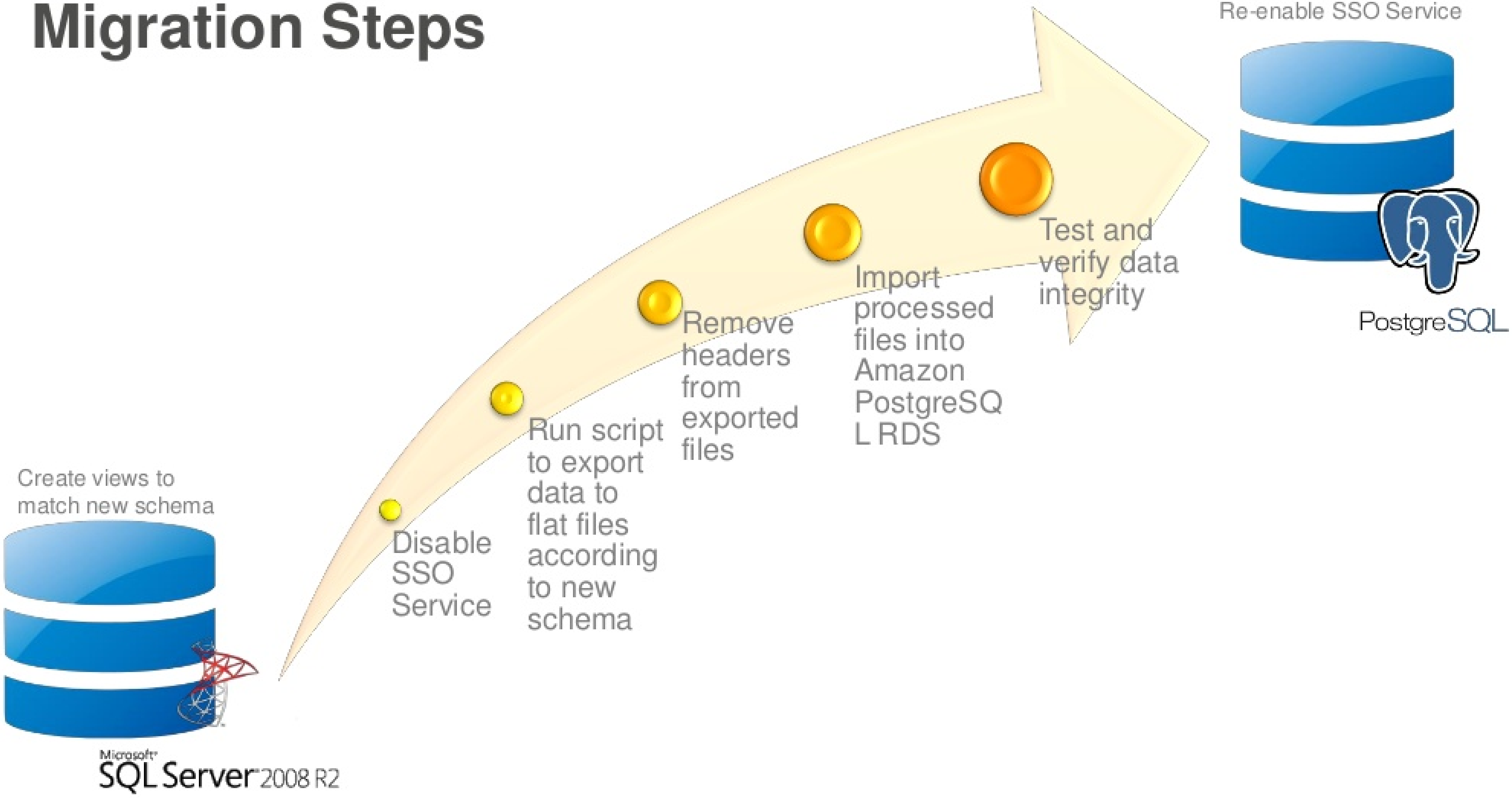
- Amazon's RDS service for PostgreSQL was used
- System code base was updated to integrate with the new database

Amazon RDS PostgreSQL Instance Microsoft .net

~1.8 million

SSO users migrated

Migration Steps



Create views to match new schema

Disable SSO Service

Run script to export data to flat files according to new schema

Remove headers from exported files

Import processed files into Amazon PostgreSQ L RDS

Test and verify data integrity

Re-enable SSO Service

The Solution and Benefits

Uberfusion's Delivery for SSO Migration to AWS

Challenge: Improve System Performance

- Use of Redis cache instead of DB for sessions and time-based tickets
- Frequently accessed data is cached to reduce repeated DB reads
- Compression and Browser Caching enabled on Amazon CloudFront, improving page load times by 50%
- TPS improved to 179 TPS for login on 4 auto-scaled servers vs 40 TPS during on-premise

Amazon CloudFront · Amazon ElastiCache · Amazon S3 · Amazon SES · Amazon SQS

Page load times

⬇50%

Challenge: Maintaining Connectivity

- Citrix Cloudbridge is leveraged to ensure connectivity between AWS Cloud and on-premise DC

CITRIX CloudBridge · HA PROXY

Challenge: DB Migration from MSSQL to Postgres

- Data exported into flat files with same schema as new Postgres DB
- Membership libraries modified to replicate same encryption in PostgreSQL

Amazon RDS · PostgreSQL instance · TXT

TPS improvement for login

⬆> 4x

> 500,000 sessions in June 2016)

Challenge: End-to-End Monitoring

- Silicone and service metrics monitored with Amazon Cloudwatch
- Runscope used for API uptime and response time monitoring
- Extensive logs input into Amazon Elastisearch via Amazon Firehose, visualised via Kibana

Amazon CloudWatch · Amazon Kinesis Firehose · Amazon S3 · Amazon Elasticsearch Service · Runscope · Kibana

Challenge: Scaling and Simplified Provisioning

- Separate applications on Amazon EC2
- Individual scaling groups according to load

Auto Scaling · Amazon EC2 · Elastic Load Balancing · AWS CodeDeploy · CloudFormation

The Benefits

Fast Provisioning of Infrastructure

- Scales independently and only pay for what you use.
- <1 day to duplicate environments vs >5 days for on-premise.

Improved Development Efficiency

By leveraging AWS PaaS components, man-days can be focused on other parts of the system.

Improved Availability and System Performance

- 50% improvement in page load times.

Increased Visibility Allows for Proactive Teams

- Detected McAfee FIM tool causing CPU spikes.
- Detected iOS production application hitting SSO Staging.

Sustain High Loads

- Anticipating Euro 2016 and Olympics 2016 load.
- 179 TPS in AWS vs 40 TPS on-premise.

Understanding Database Migration Projects

It’s Not Just the Database!

Application code

Scripts and integration points

Backup and recovery

General management procedures

- Monitoring
- Notifications
- Auditing
- Tuning and Diagnostics

People!

Project Planning

Project Assessment

Database Assessment

Application Assessment

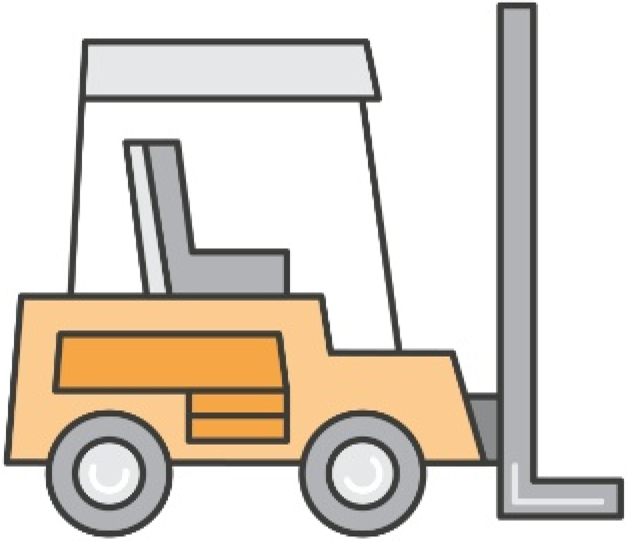
Skill Sets Required

Hiring and Developing Talent

Choosing the right POCs

Project Execution

Migration Effort



Database migration

Lift & shift

- Like to like

Write from scratch

- Switch engine

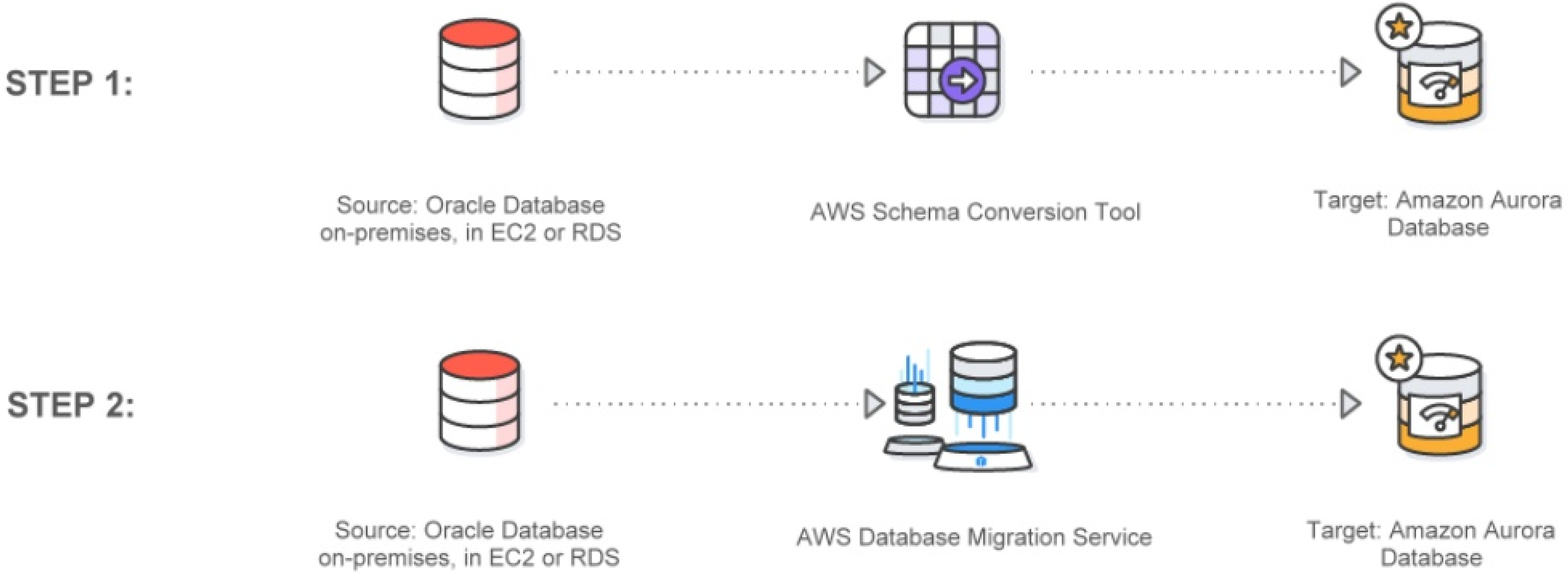
Modernize

- Fan out or consolidate
- Re-architect

Database migration – multi phase process

Phase	Description	Automation	Effort (%)
1	Assessment	SCT	2
2	Database Schema Conversion	SCT/DMS	14
3	Application Conversion/Remediation	SCT	25
4	Scripts Conversion	SCT	7
5	Integration with 3rd party applications		3
6	Data Migration	DMS	4
7	Functional testing of the entire system		29
8	Performance tuning	SCT	2
9	Integration and deployment		7
10	Training and knowledge		2
11	Documentation and version control		2
12	Post production support		3

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

AWS

SUMMIT

amazon
web services