

Project Report: Moving a Database in the Cloud

A Beginner's Guide to Migrating from RDS to Aurora with AWS DMS

Data Migration using AWS DMS

Objective:

Migrate data from a source database (e.g., MySQL) to a target database (e.g., Amazon Aurora) using AWS DMS, including initial load and (optionally) change data capture (CDC).

1. What We Did and Why (The Big Picture)

Imagine you have a popular library of books (your data) stored in a good, reliable building (an **Amazon RDS** database). Over time, this library has become so popular that you need a bigger, faster, and more modern building that can handle more visitors and automatically make copies of every book for safety.

This new, advanced building is **Amazon Aurora**.

This project was all about moving our sample library, called "Sakila," from the old building (RDS) to the new one (Aurora) without closing the library to the public. To do this, we used a special, professional moving service provided by Amazon called **AWS Database Migration Service (DMS)**.

The Goal: Move the entire Sakila database smoothly and safely, with almost no disruption.

2. The Key Players (Our Tools and Services)

To understand the move, let's meet the key players involved.

- **The Old Home (Source Database): Amazon RDS MySQL**
 - This is where our Sakila database lived initially. It's a standard, reliable database service on AWS.
- **The New Home (Target Database): Amazon Aurora**
 - This is our destination. Aurora is a high-performance, super-reliable database designed by AWS. It's like upgrading from a great house to a self-repairing smart mansion.
- **The Professional Moving Company: AWS DMS**
 - This is the star of our show. DMS is a service that specializes in moving data between databases. It does two amazing things:

1. **The Big Move (Full Load):** It copies all the existing data from the old home to the new one.
 2. **The "While We Move" Service (Change Data Capture - CDC):** While the main move is happening, if any new data arrives at the old home, DMS immediately makes a copy and takes it to the new home. This ensures nothing gets lost and the two databases stay in sync.
- **The Security Pass (IAM Role)**
 - You can't just let a moving company into your home without permission. An **IAM Role** is a secure "ID badge" that we give to our moving company (DMS). This badge proves it has permission to access the old database, the new database, and other tools it needs for the job.
 - **The Checklist Box (S3 Bucket)**
 - Before a big move, you make an inventory list. DMS can create an "assessment report"—a checklist of potential issues. We needed a place to store this report, and we used an **S3 Bucket**, which is like a secure, infinitely large digital storage box in the cloud.
 - **The Mover's Logbook (CloudWatch Logs)**
 - This is a detailed log of every single action the moving company (DMS) takes. If something goes wrong, we can look at this logbook to see exactly what happened and when.

3. The Step-by-Step Moving Plan

Here's exactly how we completed the move, step by step.

Step 1: Getting the Old Home Ready

- **What We Did:** We set up our starting database (RDS MySQL) and put the sample "Sakila" library data inside it.
- **Why It's Important:** You can't start a move until the house you're moving from is ready and all the contents are in place.

Step 2: Giving the Mover the Addresses

- **What We Did:** We told our moving company (DMS) the exact addresses of the old database and the new one. In AWS, these "addresses" are called **Endpoints**.

- **Why It's Important:** The movers need to know precisely where to pick up the data from and where to drop it off. We also tested the connection to make sure the moving truck could get to both locations.

Step 3: Handing Out the Security Passes

- **What We Did:** We created the **IAM Role** (our security pass) and gave it to DMS.
- **Why It's Important:** This is a critical security step. It ensures that only our authorized moving service can access our data, preventing any unauthorized access.

Step 4: Creating the Migration Task (The Official Moving Job)

- **What We Did:** We gave the moving company (DMS) its official instructions. This is called creating a **Migration Task**. The instructions included:
 - "Move everything from the 'sakila' library." (We used a wildcard sakila.% which means "all tables in the Sakila database").
 - "After the initial move, keep watching the old home for any new items and bring them over too." (This is enabling CDC).
 - "Keep a detailed log of everything you do." (This is enabling CloudWatch logging).
- **Why It's Important:** Clear instructions ensure the move happens exactly as we want it to, without any confusion.

Step 5: Starting the Move and Watching Closely

- **What We Did:** We told DMS to start the job. We watched the status change from Starting → Running → Load Complete.
- **Why It's Important:** Just like a real move, you want to keep an eye on the process to make sure everything is going smoothly. We used the **CloudWatch Logbook** to see the detailed progress.

4. How We Checked Everything Arrived Safely (Verification)

Once the movers said they were done, we didn't just take their word for it. We checked everything ourselves.

1. **Counting the Boxes (Row Count):** We counted the number of data entries (rows) in the old database and compared it to the count in the new one. They matched perfectly!
2. **Spot-Checking the Items (Sample Checks):** We randomly picked a few pieces of data from the old database and made sure they looked exactly the same in the new one.

3. **Testing the "While We Move" Service (CDC Check):** We added a new piece of data to the old database and, within seconds, it appeared in the new database. This confirmed our ongoing sync was working.

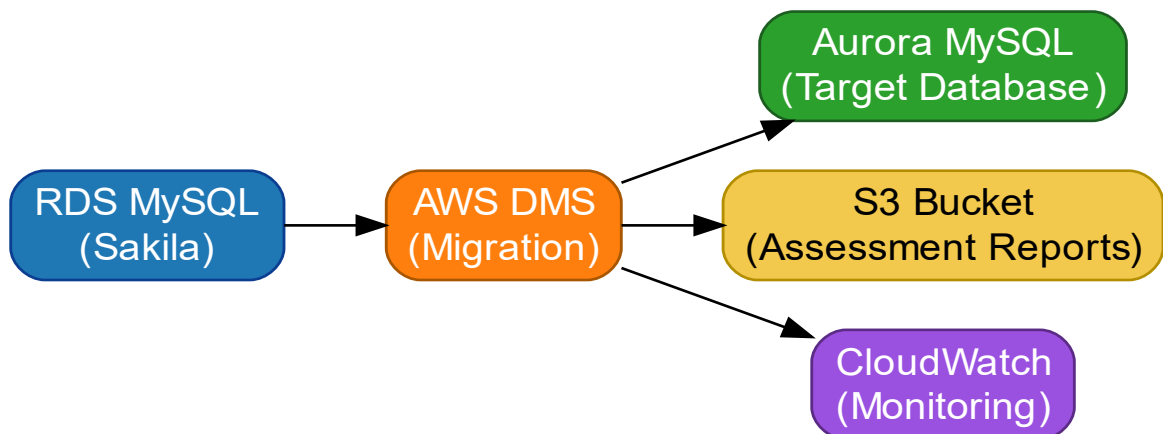
5. Summary and Final Thoughts for Beginners

This project showed that moving a database, which sounds complicated, is made much easier and safer with a service like **AWS DMS**.

Key Takeaways for Anyone New to AWS:

- **Think in Analogies:** Relating services like DMS to a "moving company" or IAM to a "security pass" makes it much easier to understand their purpose.
- **Automation is Your Friend:** AWS DMS does most of the heavy lifting. Our job was to give it the right instructions and permissions.
- **Security First:** Always think about permissions (like IAM Roles). It's the foundation of working securely in the cloud.
- **Always Verify:** Don't assume a process worked just because you see a "Complete" status. Always check your data to be sure.

ARCHITECTURE DIAGRAM



AWS DMS Migration Steps

1. **Set Up Source & Target Databases**
Launch RDS MySQL (source) and Aurora MySQL (target). Load Sakila data into source.

Step 1: Set Up Source & Target Databases

The image shows two screenshots related to database setup. The top screenshot is from the AWS DMS console, and the bottom screenshot is from the DBeaver database client.

AWS DMS Console:

- Header:** United States (N. Virginia), Account ID: 4013-9951-6214, Sologogo.
- Left Navigation:** AWS DMS, New navigation, Dashboard, Getting started, Convert or move to managed, Migrate or replicate, Tasks, Endpoints, Provisioned instances, Monitor.
- Message:** targetaurora created successfully. (View details)
- How it works:** Use either serverless or a provisioned replication instance to migrate a database or configure ongoing replication.
- Endpoints (2):**
 - Find endpoint
 - Actions
 - Create endpoint
- Endpoints Table:**

	Name	Type	Status	Engine	Server name	Port	Migrate
<input type="checkbox"/>	database-1	Source	Active	MySQL	database-1.c6tcuy46ovku.us-east-1.rds.amazonaws.com	3306	
<input type="checkbox"/>	targetaurora	Target	Active	MySQL	room12source.c6tcuy46ovku.us-east-1.rds.amazonaws.com	3306	

DBeaver Database Client:

- Room11-Connections**
- File Edit View Query Database Server Tools Scripting Help**
- Navigator:** Filter objects, sys
- SCHEMAS:** sys
- Query 1:** sakila-data sakila-schema
 - Sakila Sample Database Data
 - Version 1.5
 - Copyright (c) 2006, 2025, Oracle
 - Redistribution and use in source and binary forms, without modification, are permitted provided that the following conditions are met:
- Output:**

#	Time	Action	Message	Duration / Fetch
13	21:29:17	SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='TRADITIONAL'	0 row(s) affected	0.296 sec
14	21:29:18	DROP SCHEMA IF EXISTS sakila	0 row(s) affected	0.235 sec
15	21:29:18	CREATE SCHEMA sakila	1 row(s) affected	0.204 sec
16	21:29:18	USE sakila	0 row(s) affected	0.204 sec
17	21:29:18	CREATE TABLE actor (actor_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT...	0 row(s) affected	0.328 sec
18	21:29:19	CREATE TABLE address (address_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT...	0 row(s) affected	0.250 sec
19	21:29:19	CREATE TABLE category (category_id TINYINT UNSIGNED NOT NULL AUTO_INCREMENT...	0 row(s) affected	0.235 sec
20	21:29:19	CREATE TABLE city (city_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT...	0 row(s) affected	0.235 sec
21	21:29:20	CREATE TABLE country (country_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT...	0 row(s) affected	0.250 sec
22	21:29:20	CREATE TABLE customer (customer_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT...	0 row(s) affected	0.250 sec
23	21:29:20	CREATE TABLE film (film_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT...	0 row(s) affected	0.266 sec
24	21:29:20	CREATE TABLE film_actor (actor_id SMALLINT UNSIGNED NOT NULL, film_id SMA...	0 row(s) affected	0.250 sec
25	21:29:21	CREATE TABLE film_category (film_id SMALLINT UNSIGNED NOT NULL, category_id...	0 row(s) affected	0.281 sec

2. Create DMS Replication Instance

Create a replication instance in AWS DMS console

aws Search [Alt+S] United States (N. Virginia) Account ID: 4013-9951-6214 Sologogo

DMS > Migrate or replicate > Endpoints

AWS DMS

- New navigation
- Dashboard
- Getting started
- Convert or move to managed
- Migrate or replicate
 - Tasks **New**
 - Endpoints**
 - Provisioned instances
- Monitor

targetaurora created successfully. [View details](#)

How it works
Use either serverless or a provisioned replication instance to migrate a database or configure ongoing replication.

Endpoints (2) [Refresh](#) [Actions](#) [Create endpoint](#)

Find endpoint

<input type="checkbox"/>	Name	Type	Status	Engine	Server name	Port	Migrate
<input type="checkbox"/>	database-1	Source	Active	MySQL	database-1.c6tcuy46ovku.us-east-1.rds.amazonaws.com	3306	
<input type="checkbox"/>	targetaurora	Target	Active	MySQL	room12source.c6tcuy46ovku.us-east-1.rds.amazonaws.com	3306	

3. Create Source and Target Endpoints

Configure endpoints for RDS MySQL and Aurora MySQL, test connections

4. Create Migration Task

Define migration task: full load (optionally CDC), map tables, enable logging, start task.

The screenshot shows the AWS IAM console interface. At the top, there's a dark navigation bar with the AWS logo, a search bar, and various utility icons. Below this, the breadcrumb navigation shows 'IAM > Roles'. The left sidebar, titled 'Identity and Access Management (IAM)', contains a search bar and a list of navigation items: 'Dashboard', 'Access management' (expanded), 'User groups', 'Users', 'Roles' (selected), 'Policies', 'Identity providers', 'Account settings', 'Root access management', 'Access reports', and 'Access Analyzer'. The main content area features a green notification banner stating 'Role DMS-S3-Access-Role created.' Below this, the 'Roles (7)' section includes an 'Info' link and a description: 'An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assigned to users or groups you trust.' A search bar is provided for filtering roles. The roles are listed in a table with columns for 'Role name' and 'Trusted entities'. The roles listed are: 'AWSServiceRoleForRDS', 'AWSServiceRoleForSupport', 'AWSServiceRoleForTrustedAdvisor', 'dms-cloudwatch-logs-role', 'DMS-S3-Access-Role', and 'dms-vpc-role'. The 'DMS-S3-Access-Role' is highlighted in blue.

Role name	Trusted entities
AWSServiceRoleForRDS	AWS Service: rds (Service-Linked Role)
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)
dms-cloudwatch-logs-role	AWS Service: dms
DMS-S3-Access-Role	AWS Service: dms
dms-vpc-role	AWS Service: dms

aws

Search

[Alt+S]

United States (N. Virginia)

Amazon S3

Amazon S3

General purpose buckets

Directory buckets

Table buckets

Vector buckets [Preview](#)

Access Grants

Access Points (General Purpose Buckets, FSx file systems)

Access Points (Directory Buckets)

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

General purpose buckets

All AWS Regions

Directory buckets

General purpose buckets (1) [Info](#)

Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	Creation date
my-dms-assessment-reports-2025	US East (N. Virginia) us-east-1	August 8, 2025, 23:1

Account snapshot [Info](#)

Updated daily

Storage Lens provides visibility into storage usage and activity trends.

[View dashboard](#)

External access summary - new

Updated daily

External access findings help you identify t allow public access or access from other AV

CloudShell

Feedback

© 2025 Amazon Web Services, Inc. or its affiliates. Privacy

DMS

Migrate or replicate

Tasks

AWS DMS

New navigation

Dashboard

Getting started

Convert or move to managed

Migrate or replicate

Tasks [New](#)

Endpoints

Provisioned instances

Monitor

Start task

Are you sure you want to start your provisioned task **mysql-to-aurora-migration**?

⚠ Check naming convention for database objects

This action start the Full load process. It will drop databases, schemas, and tables from source database to the target instance. This process will overwrite existing database objects with the same name.

⚠ Premigration assessment status

We recommend you consider creating and running a premigration assessment for this migration task to identity potential issues before your start the migration. Choose **Create premigration assessment** from the Actions menu, then you can view assessment results from the **Premigration assessment tab**. [Learn more](#)

Cancel

Start

Step 5: Monitor the Migration•Watch task status:

Starting →Running →Load complete

The screenshot shows the AWS DMS console interface. The top navigation bar includes the AWS logo, a search bar, and account information (United States (N. Virginia), Account ID: 4013-9951-6214). The left sidebar shows the navigation menu with 'DMS' selected, and 'Migrate or replicate' > 'Tasks' is the current view. A blue banner at the top of the main content area states: 'The task mysql-to-aurora-migration is being started.' Below this, a 'How it works' section explains that users can use either serverless or a provisioned replication instance. The 'Tasks (1)' section shows a table with one task:

Identifier	Status	Full load progress	Type	Mode
mysql-to-aurora-migration	Starting	0%	Full load	Provisioned

The screenshot shows the AWS DMS console interface with the task 'mysql-to-aurora-migration' now in a 'Load completed' status. A green banner at the top of the main content area states: 'The task mysql-to-aurora-migration is started successfully.' The 'How it works' section remains the same. The 'Tasks (1)' section shows the task table with the following data:

Identifier	Status	Full load progress	Type	Mode
mysql-to-aurora-migration	Load completed	100%	Full load	Provisioned

Check CloudWatch Logsfor errors or warnings

CloudWatch

Log groups

dms-tasks-dms-lab-replica

dms-task-YBSFE42ZDFCWL13FKZLULK6FQ

CloudWatch

Log groups

Log groups

Log Anomalies

Live Tail

Logs Insights

Contributor Insights

Metrics

All metrics

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Filter events - press enter to search

1m 1h

UTC timezone

Display

Timestamp

Message

▶ 2025-08-08T23:34:08.000Z

2025-08-08T23:34:08 [SOURCE_UNLOAD]I: Unload finished for table 'sakila'. 'film' (Id = 7). 0 rows sent. ...

▶ 2025-08-08T23:34:08.000Z

2025-08-08T23:34:08 [SOURCE_CAPTURE]I: Mysql endpoint initialization finished for the stream component ...

▶ 2025-08-08T23:34:08.000Z

2025-08-08T23:34:08 [TARGET_LOAD]I: Load finished for table 'sakila'. 'customer' (Id = 6). 0 rows receiv...

▶ 2025-08-08T23:34:08.000Z

2025-08-08T23:34:08 [PERFORMANCE]I: End load handler time for sakila.customer = 3 microseconds (endpoin...

▶ 2025-08-08T23:34:08.000Z

2025-08-08T23:34:08 [PERFORMANCE]I: Total load time for sakila.customer = 3 microseconds (endpointshell...

▶ 2025-08-08T23:34:08.000Z

2025-08-08T23:34:08 [SOURCE_UNLOAD]I: set_columns_orig_type for sakila.film_actor (mysql_endpoint_metad...

▶ 2025-08-08T23:34:08.000Z

2025-08-08T23:34:08 [SOURCE_UNLOAD]I: Driver 'MySQL ODBC 8.0 Unicode Driver' is installed and ...

▶ 2025-08-08T23:34:08.000Z

2025-08-08T23:34:08 [SOURCE_UNLOAD]I: ODBC additional properties = '' (mysql_endpoint_imp.c1575)

Back to top

Table statistics (16)

Export to CSV

Validate again

Reload table data

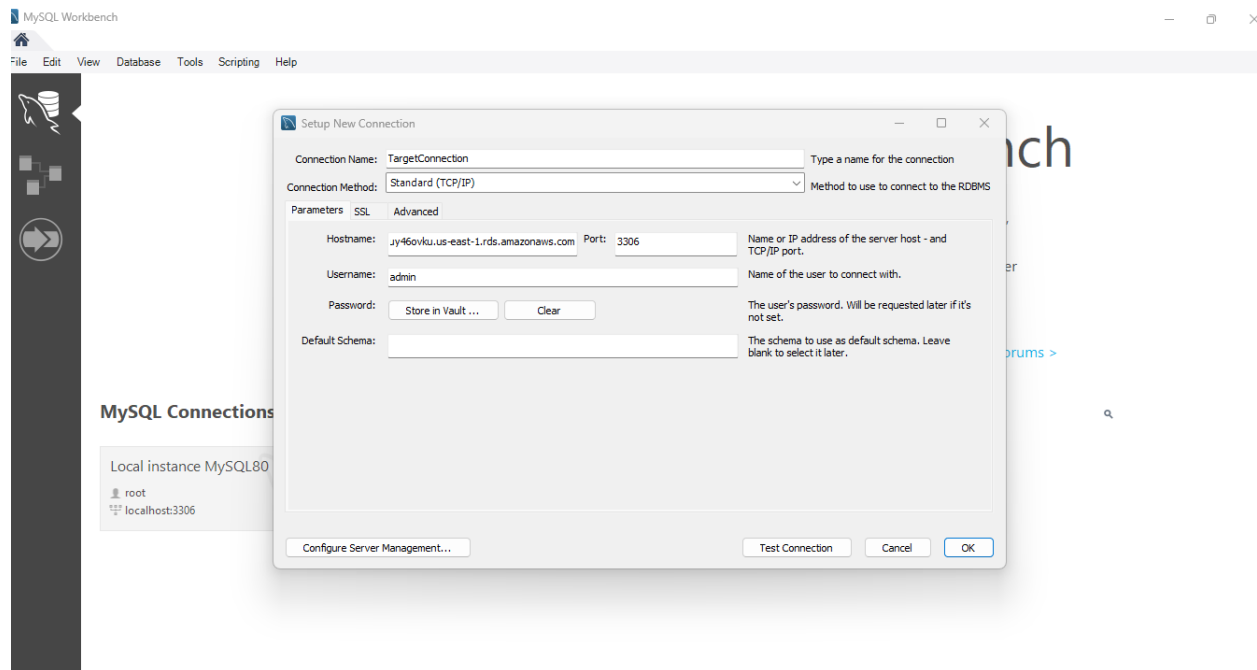
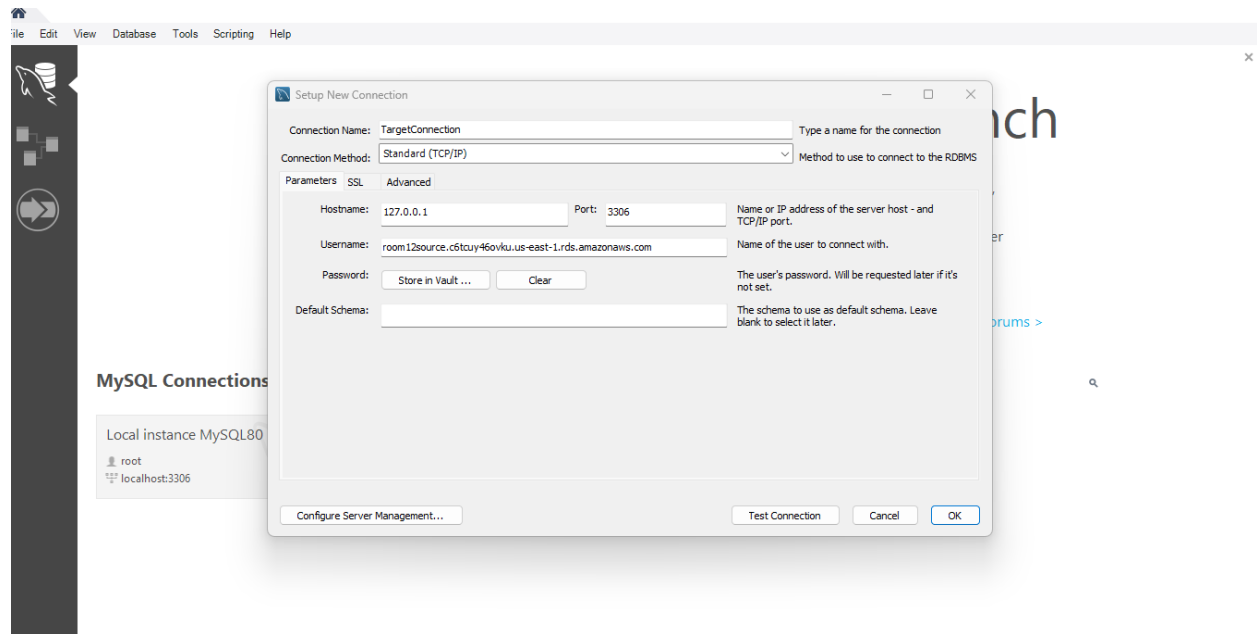
Total rows include loaded source table rows from Inserts, Deletes, Updates, DDLs, and Full load rows.

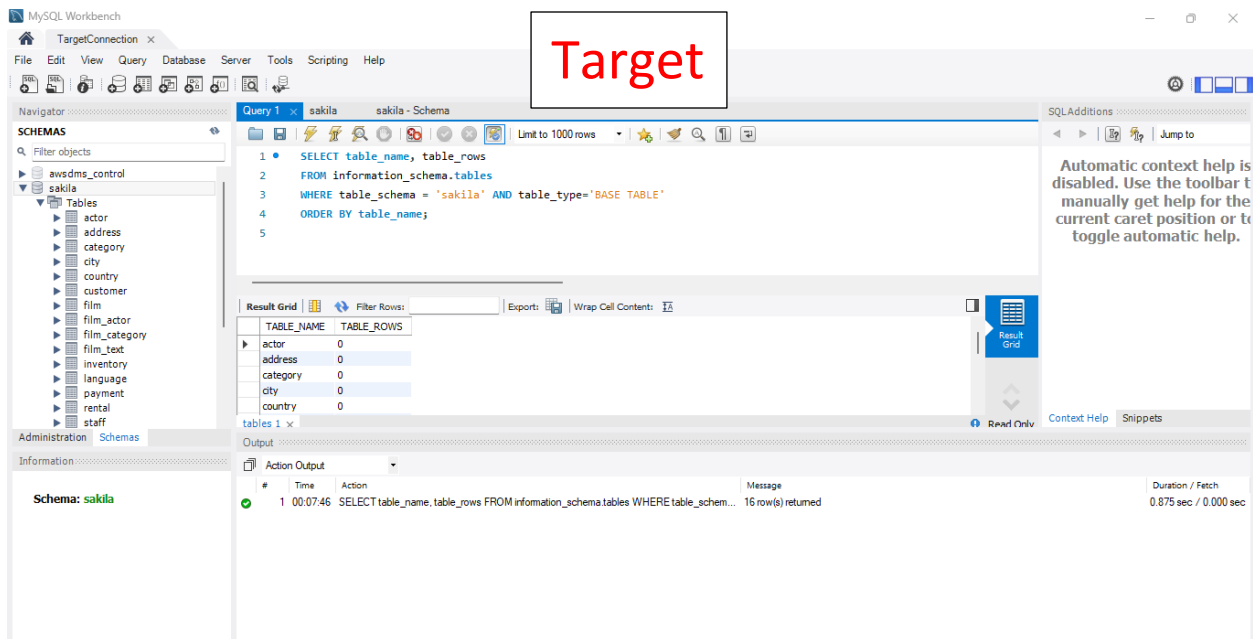
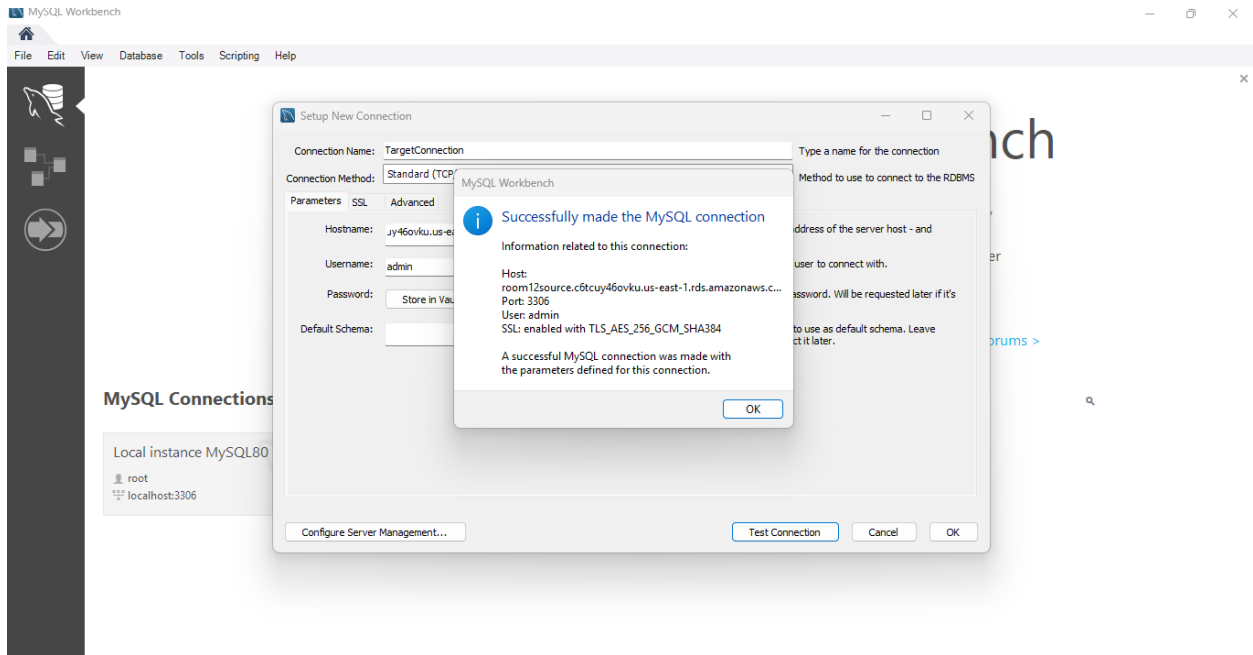
Search

< 1 >

<input type="checkbox"/>	Schema name	Table name	Load state	Logs	Elapsed load time
<input type="checkbox"/>	sakila	actor	✔ Table completed	Find in Logs	< 1 s
<input type="checkbox"/>	sakila	staff	✔ Table completed	Find in Logs	< 1 s
<input type="checkbox"/>	sakila	film_text	✔ Table completed	Find in Logs	< 1 s
<input type="checkbox"/>	sakila	city	✔ Table completed	Find in Logs	< 1 s
<input type="checkbox"/>	sakila	film_actor	✔ Table completed	Find in Logs	< 1 s
<input type="checkbox"/>	sakila	film_category	✔ Table completed	Find in Logs	< 1 s
<input type="checkbox"/>	sakila	film	✔ Table completed	Find in Logs	< 1 s

Optionally simulate inserts/updates to test CDC





MySQL Workbench

TargetConnection x Room11- Connections x

File Edit View Query Database Server Tools Scripting Help

sakila-data sakila-schema

Limit to 1000 rows

1 SELECT table_name, table_rows
2 FROM information_schema.tables
3 WHERE table_schema = 'sakila' AND table_type='BASE TABLE'
4 ORDER BY table_name;
5

Result Grid

TABLE_NAME TABLE_ROWS
actor 0
address 0
category 0
city 0
country 0

tables 1 x

Read Only

Context Help Snippets

Automatic context help is disabled. Use the toolbar manually get help for the current caret position or toggle automatic help.

Administration Schemas

Information

Schema: sakila

Object Info Session

Output

Action Output

Time Action Message Duration / Fetch
1 00:12:23 SELECT table_name, table_rows FROM information_schema.tables WHERE table_schem... 16 row(s) returned 0.938 sec / 0.000 sec

Source