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.

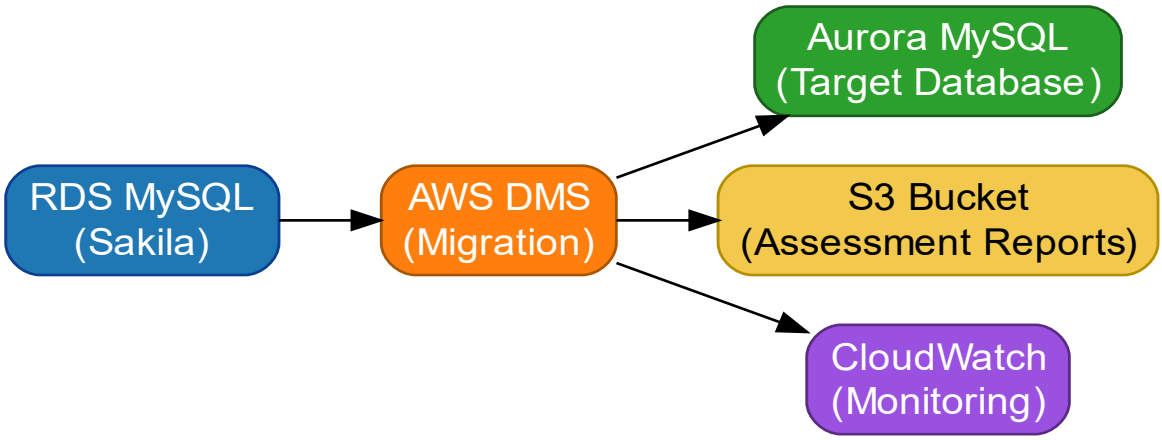
5. Summary and Final Thoughts for Beginners

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.

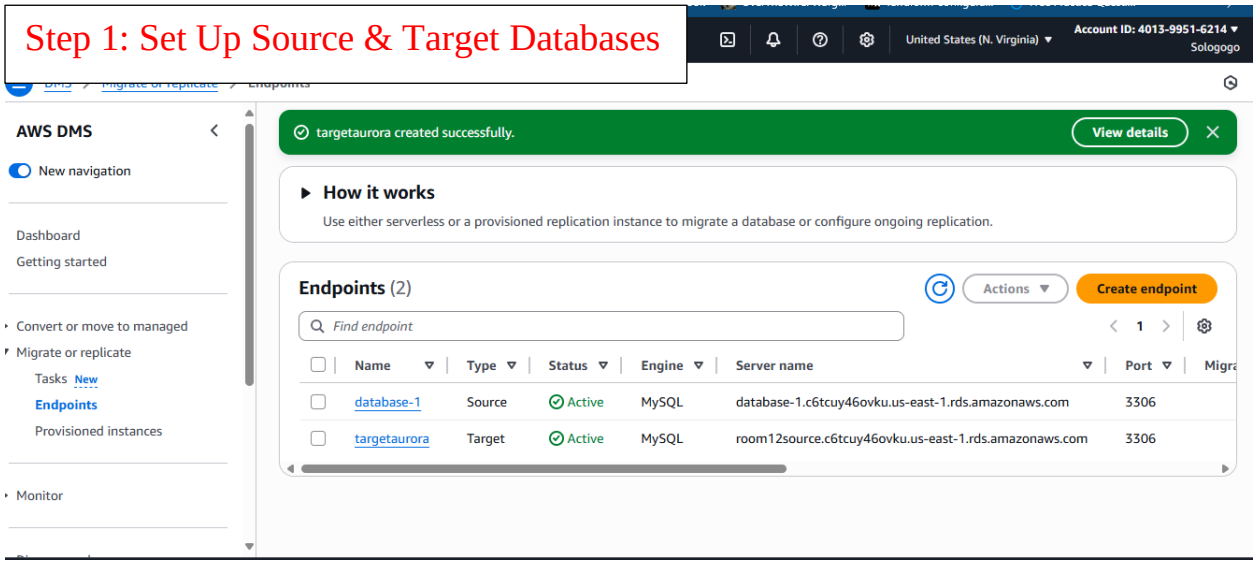
ARCHITECTURE DIAGRAM

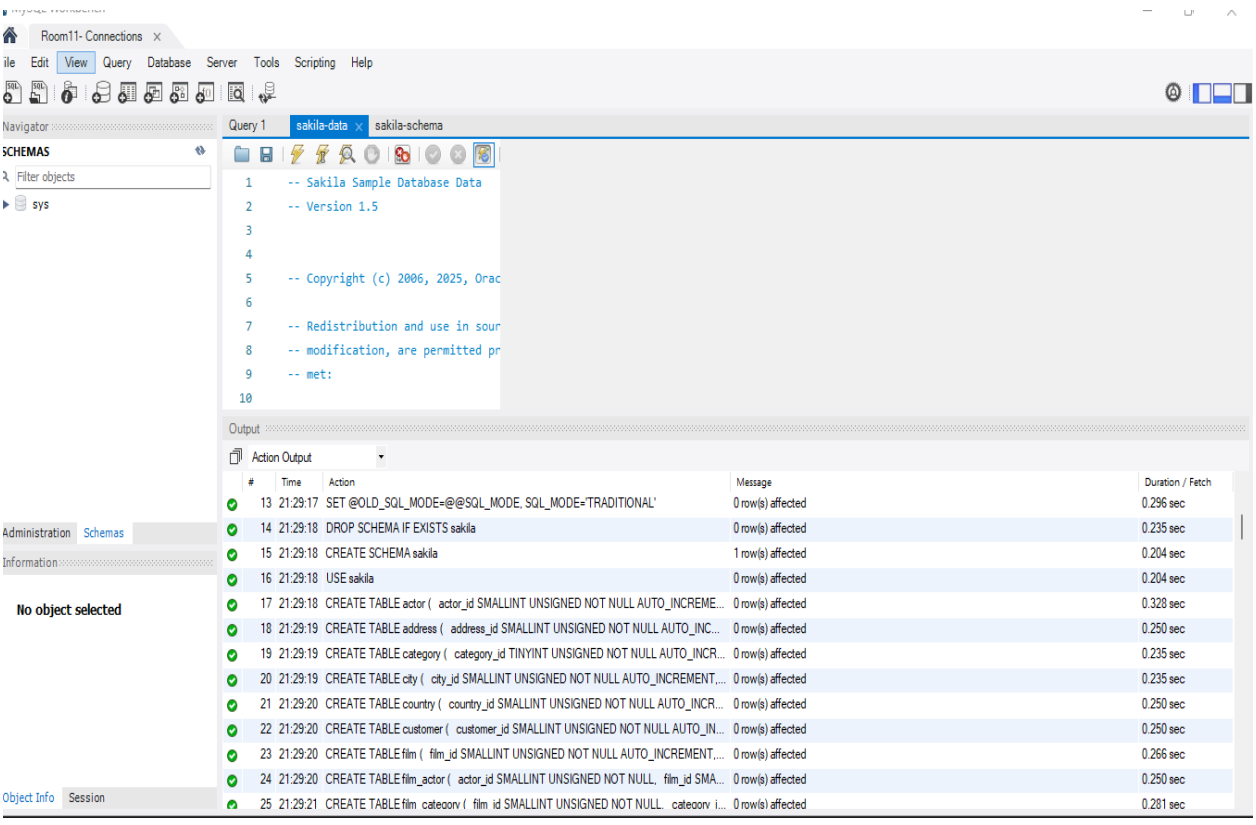


AWS DMS Migration Steps

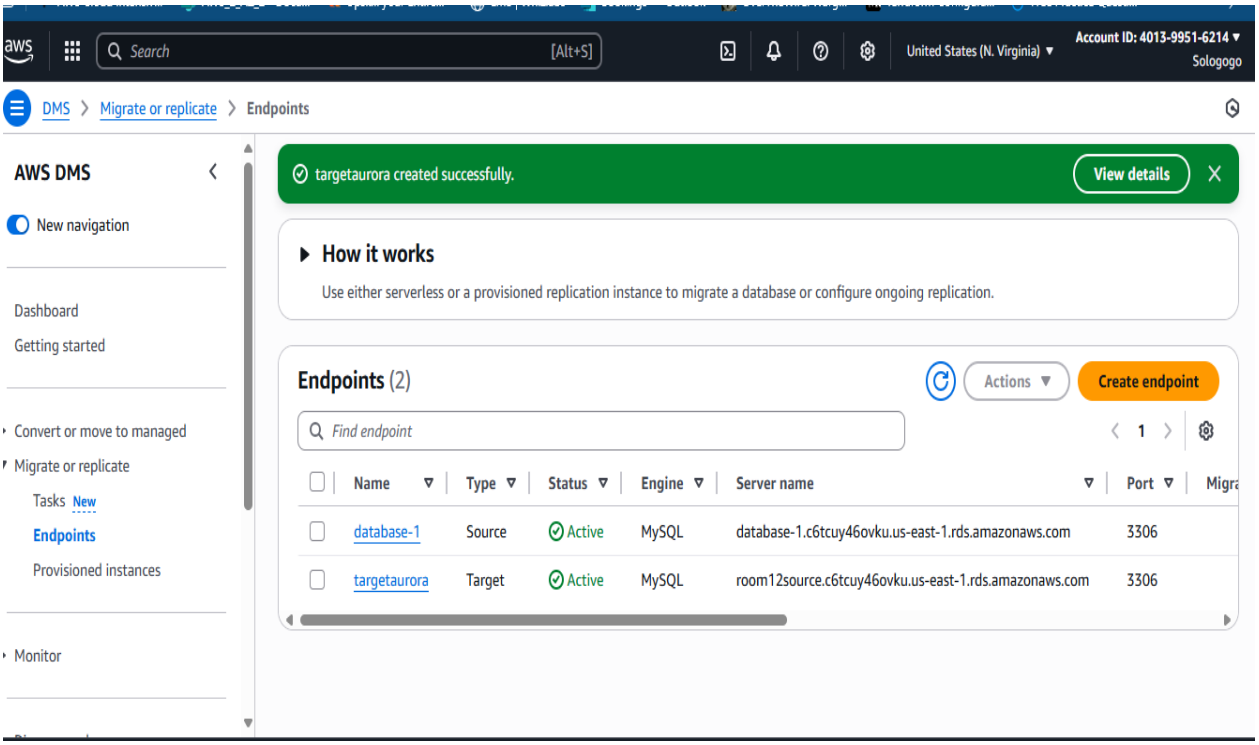
1. Set Up Source & Target Databases
   Launch RDS MySQL (source) and Aurora MySQL (target). Load Sakila data into source.

Step 1: Set Up Source & Target Databases

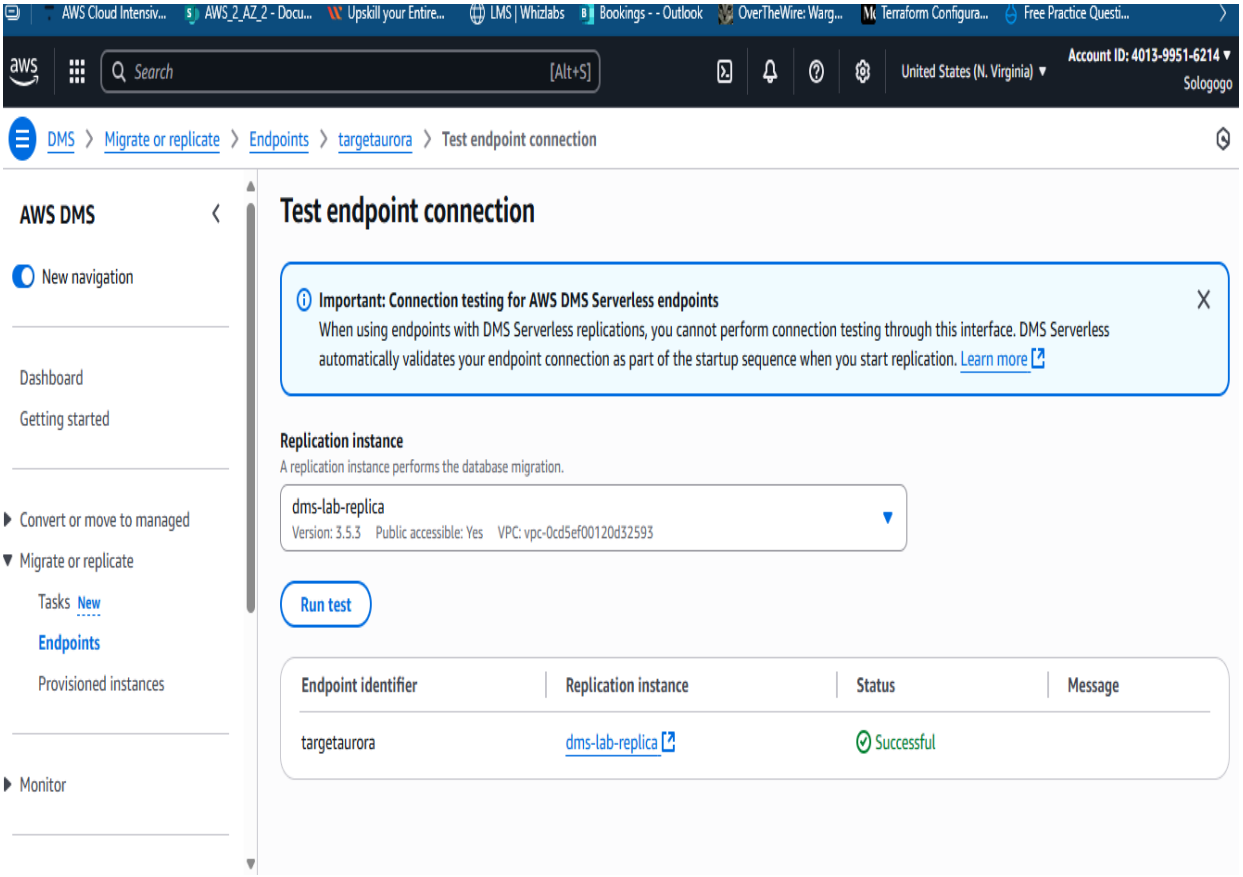


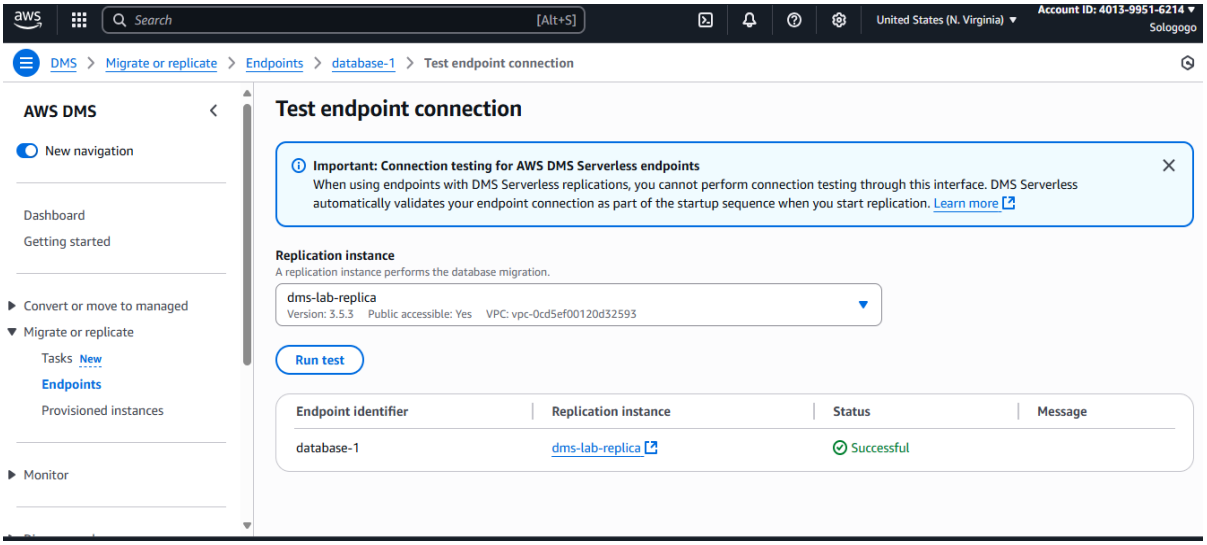


2. Create DMS Replication Instance
   Create a replication instance in AWS DMS console

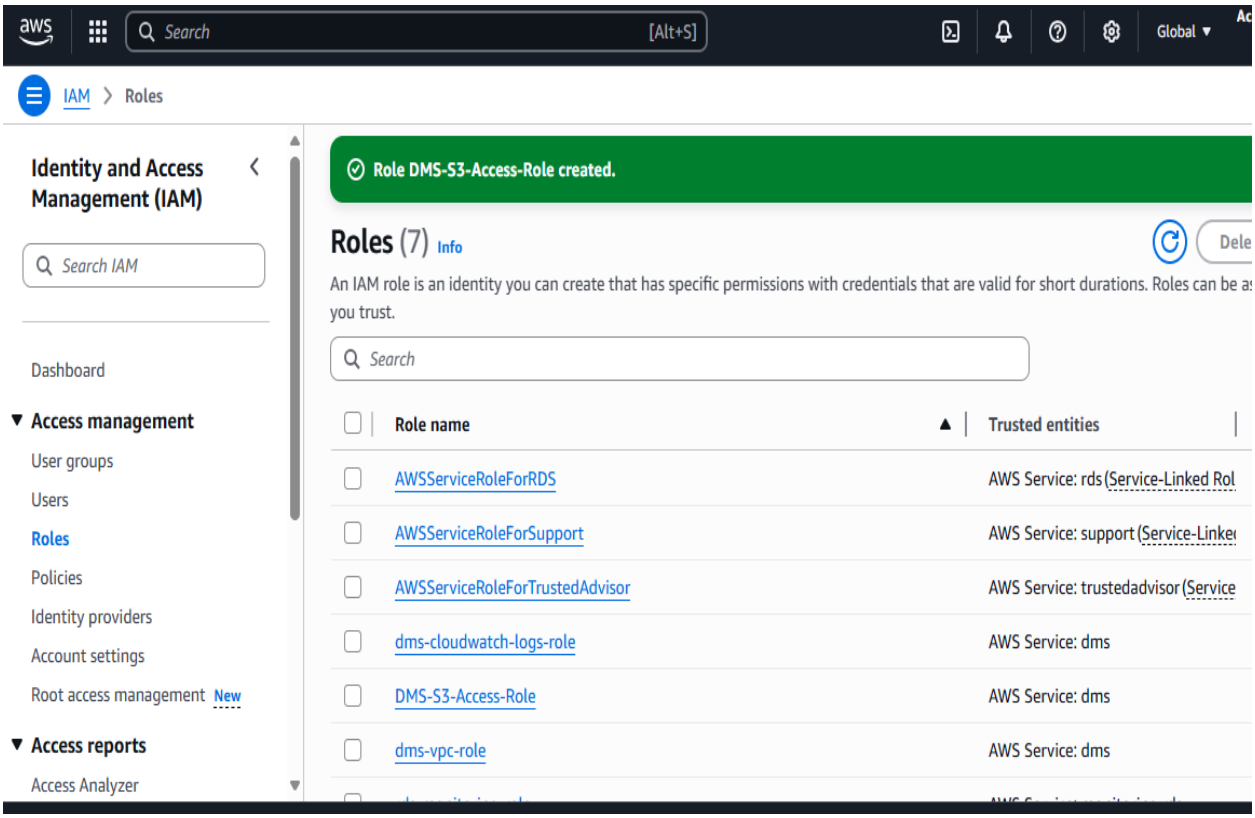


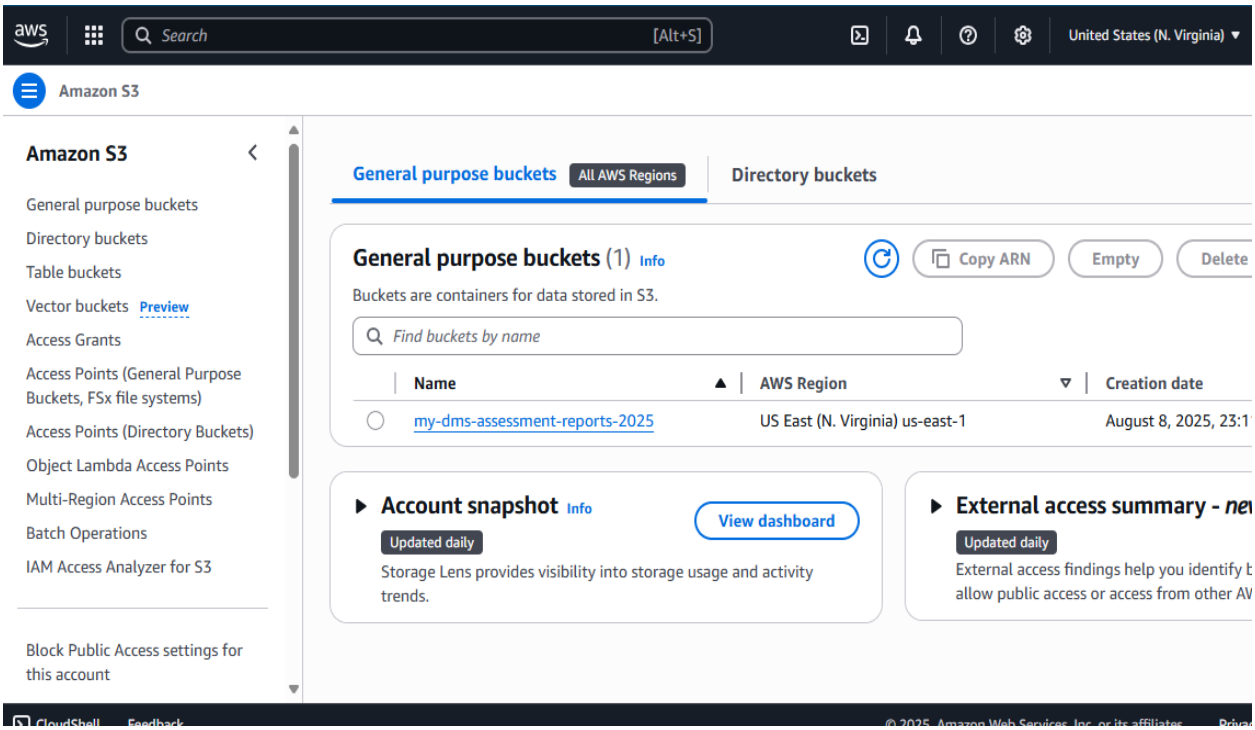
3. Create Source and Target Endpoints
   Configure endpoints for RDS MySQL and Aurora MySQL, test connections

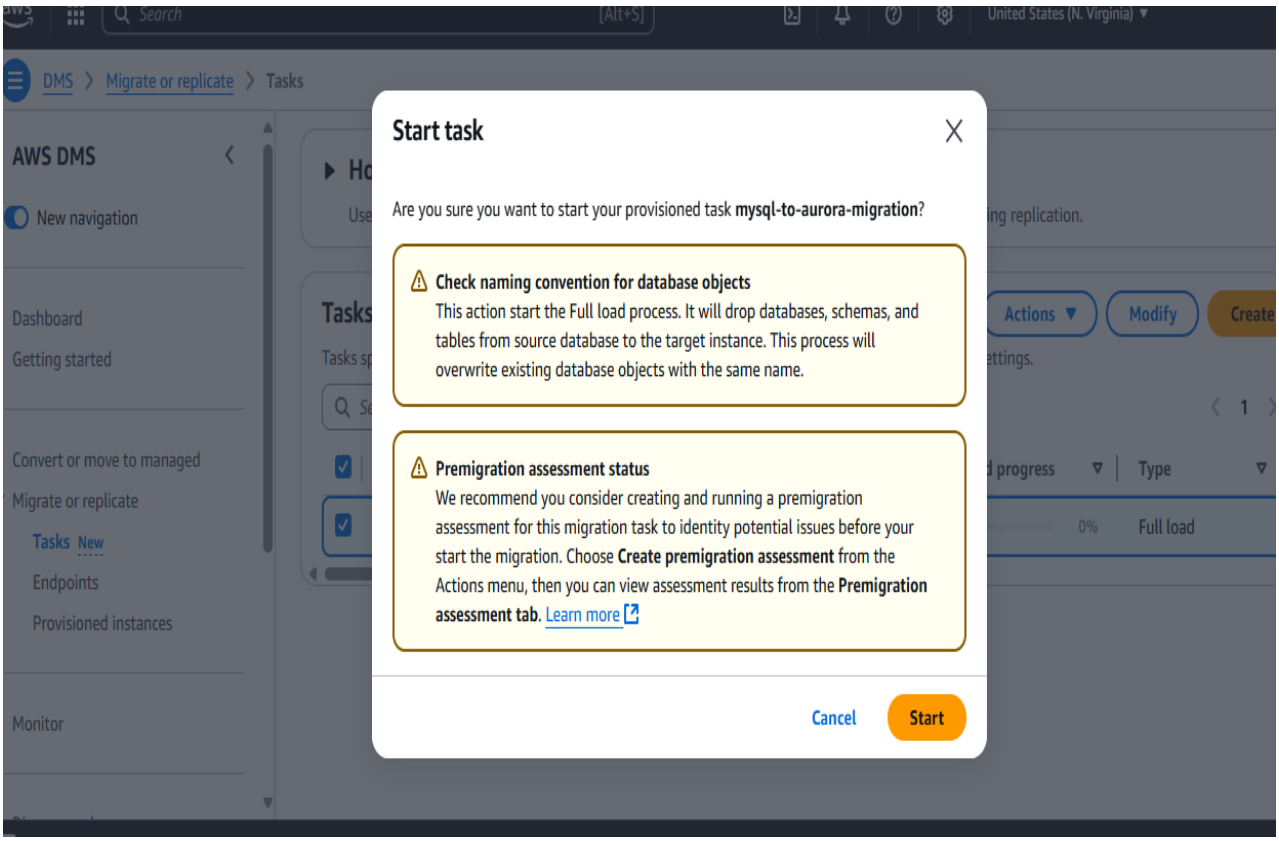

AWS Cloud Intensiv...
AWS_2_AZ_2 - Docu...
Upskill your Entire...
LMS | Whizlabs
Bookings - - Outlook
OverTheWire: Warg...
Terraform Configura...
Free Practice Questi...
aws
Search
[Alt+S]
United States (N. Virginia)
Account ID: 4013-9951-6214
Sologogo
DMS > Migrate or replicate > Endpoints > targetaurora > Test endpoint connection
AWS DMS
New navigation
Dashboard
Getting started
Convert or move to managed
Migrate or replicate
Tasks New
Endpoints
Provisioned instances
Monitor
Test endpoint connection
Important: Connection testing for AWS DMS Serverless endpoints
When using endpoints with DMS Serverless replications, you cannot perform connection testing through this interface. DMS Serverless automatically validates your endpoint connection as part of the startup sequence when you start replication. Learn more
Replication instance
A replication instance performs the database migration.
dms-lab-replica
Version: 3.5.3 Public accessible: Yes VPC: vpc-0cd5ef00120d32593
Run test
Endpoint identifier
Replication instance
Status
Message
targetaurora
dms-lab-replica
Successful


aws
Search
[Alt+S]
United States (N. Virginia)
Account ID: 4013-9951-6214
Sologogo
DMS > Migrate or replicate > Endpoints > database-1 > Test endpoint connection
AWS DMS
New navigation
Dashboard
Getting started
Convert or move to managed
Migrate or replicate
Tasks New
Endpoints
Provisioned instances
Monitor
Test endpoint connection
Important: Connection testing for AWS DMS Serverless endpoints
When using endpoints with DMS Serverless replications, you cannot perform connection testing through this interface. DMS Serverless automatically validates your endpoint connection as part of the startup sequence when you start replication. Learn more
Replication instance
A replication instance performs the database migration.
dms-lab-replica
Version: 3.5.3 Public accessible: Yes VPC: vpc-0cd5ef00120d32593
Run test
Endpoint identifier
Replication instance
Status
Message
database-1
dms-lab-replica
Successful

4. **Create Migration Task**
   Define migration task: full load (optionally CDC), map tables, enable logging, start task.

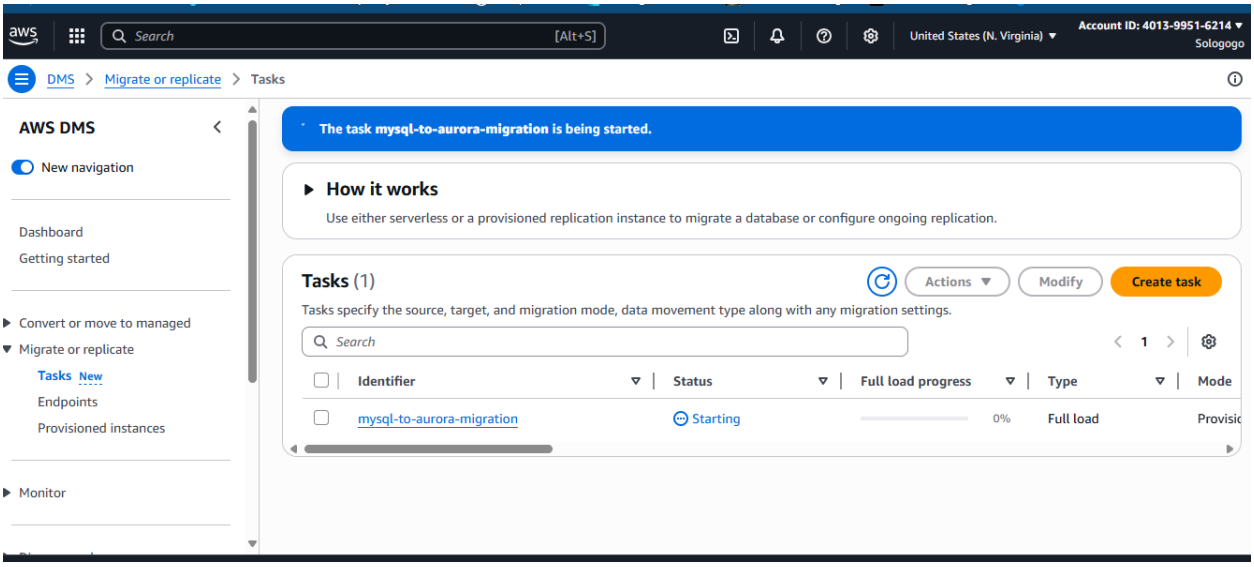


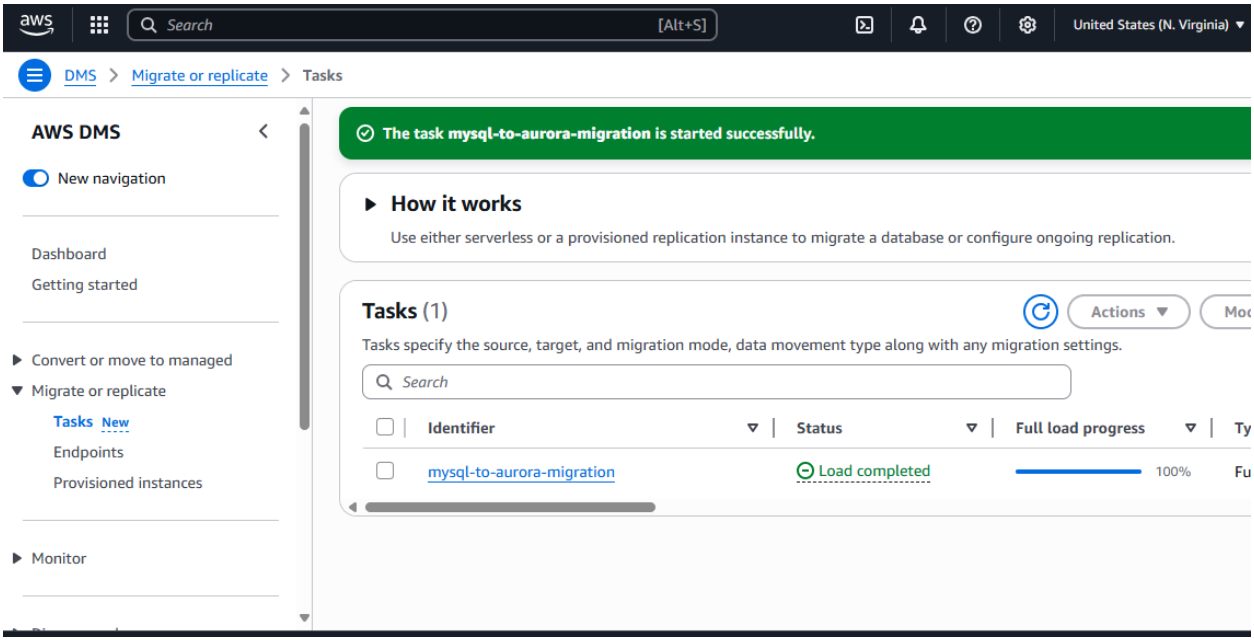

Search
[Alt+S]
United States (N. Virginia)
Amazon S3
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Vector buckets Preview
Access Grants
Access Points (General Purpose Buckets, FSx file systems)
Access Points (Directory Buckets)
Object Lambda Access Points
Multi-Region Access Points
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
Copy ARN
Empty
Delete
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
View dashboard
Storage Lens provides visibility into storage usage and activity trends.
External access summary - ne
Updated daily
External access findings help you identify b
allow public access or access from other AV
CloudShell
Feedback


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
Actions
Modify
Type
Full load

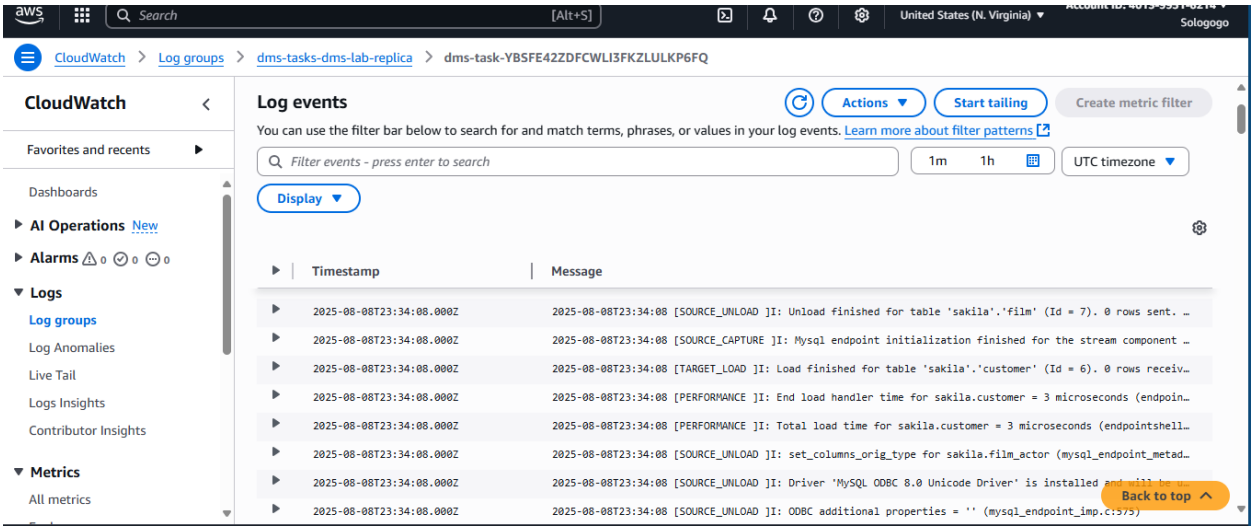
Step 5: Monitor the Migration•Watch task status:

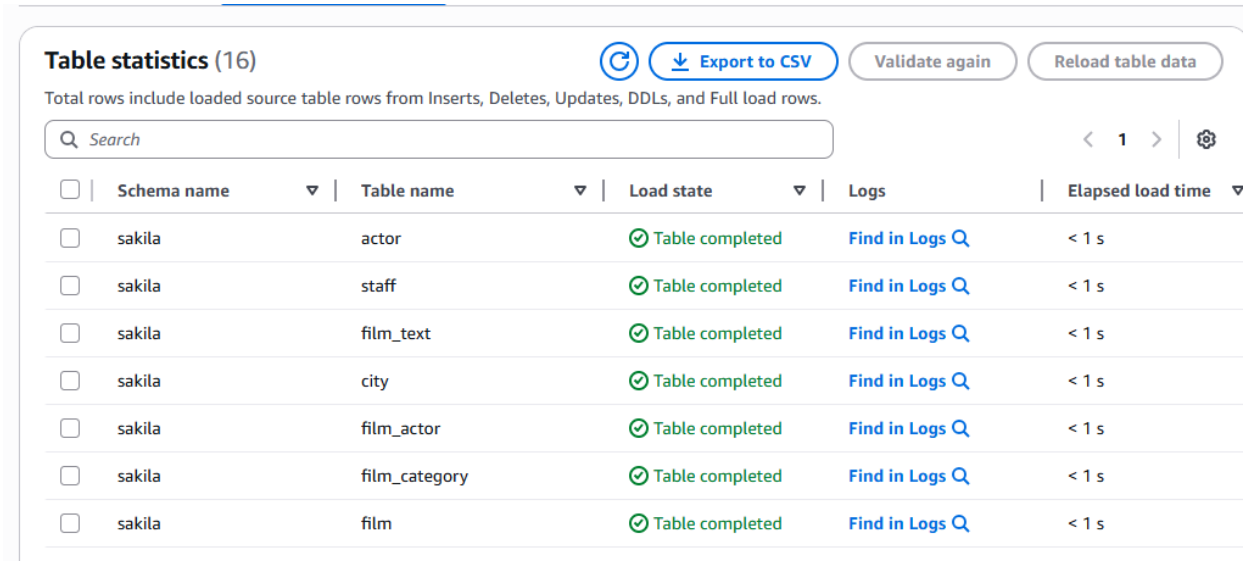
Starting →Running →Load complete





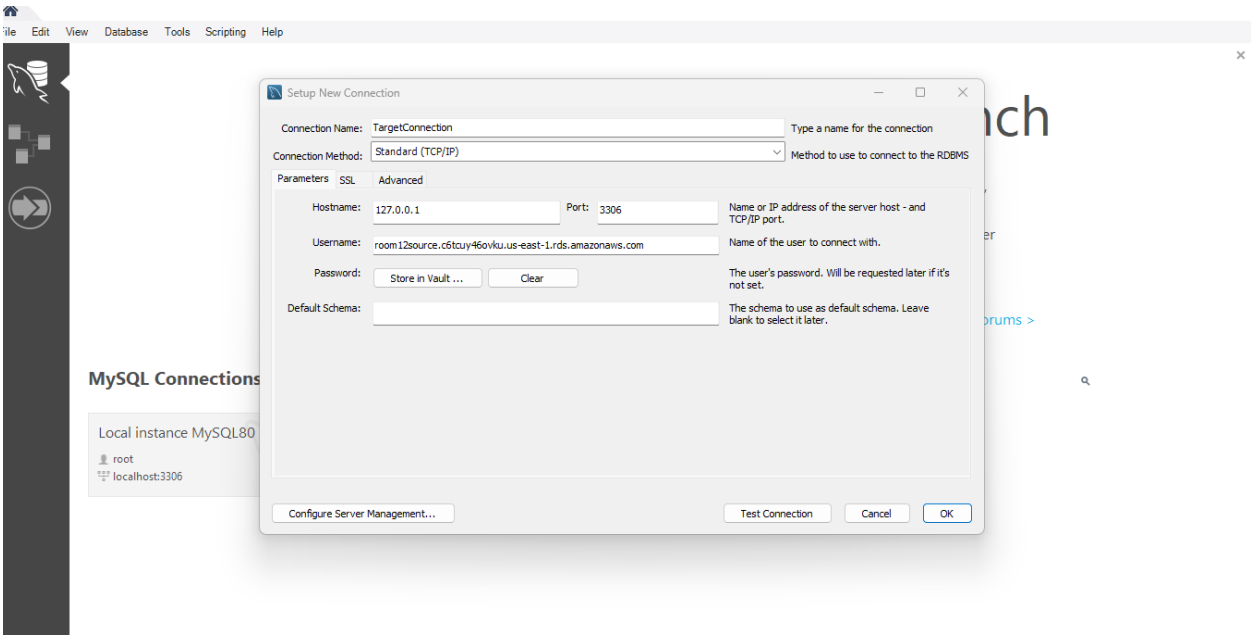
Check CloudWatch Logsfor errors or warnings

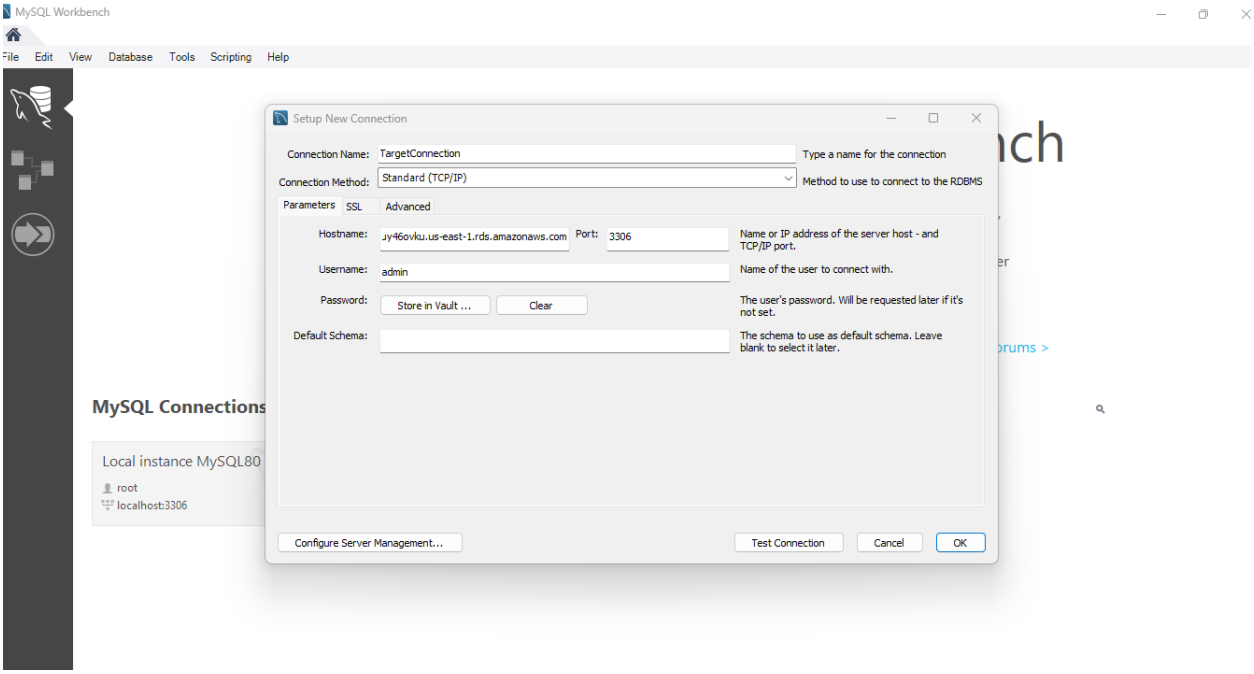


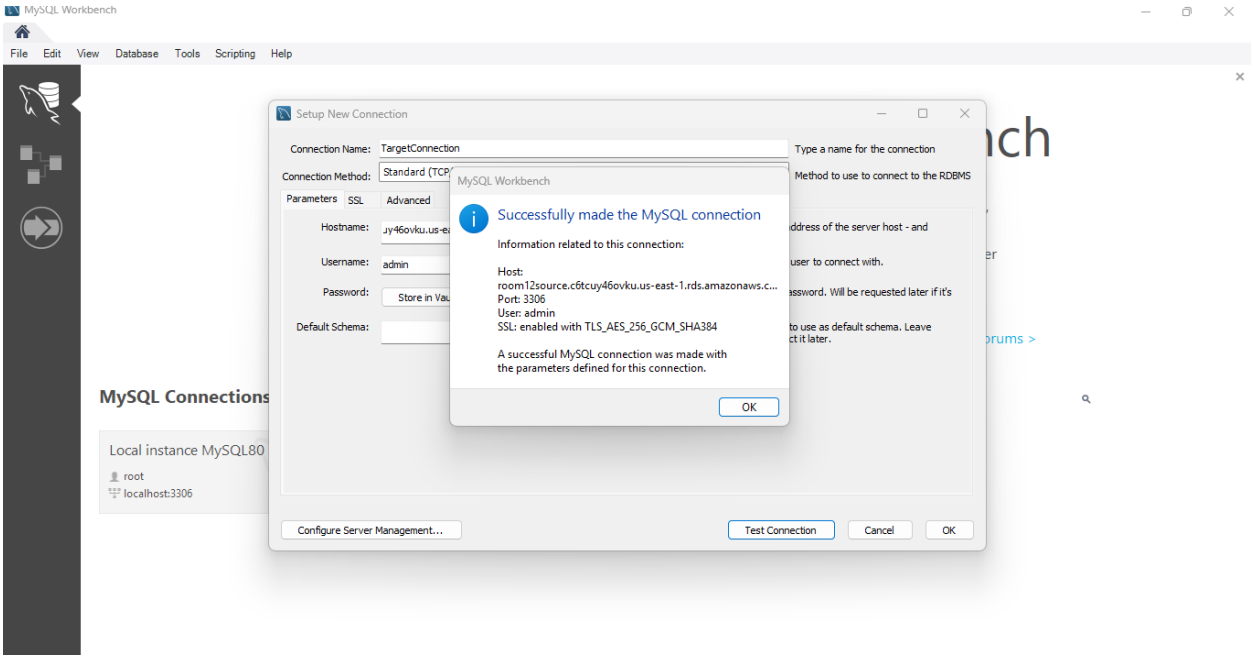


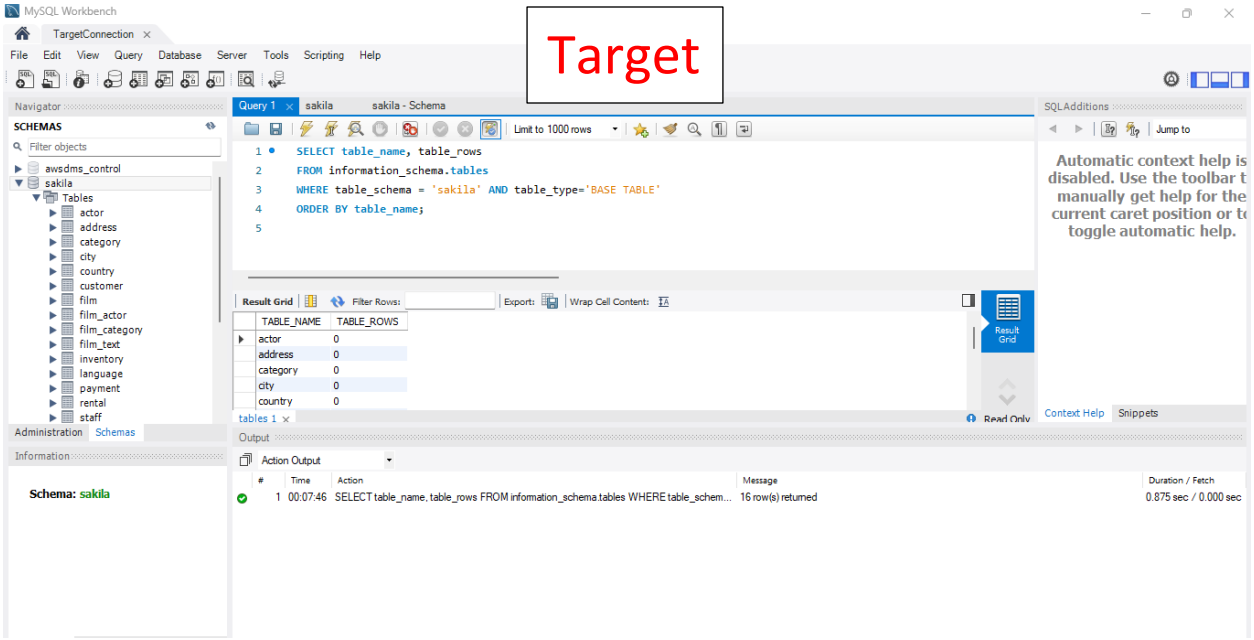
Optionally simulate inserts/updatesto test CDC

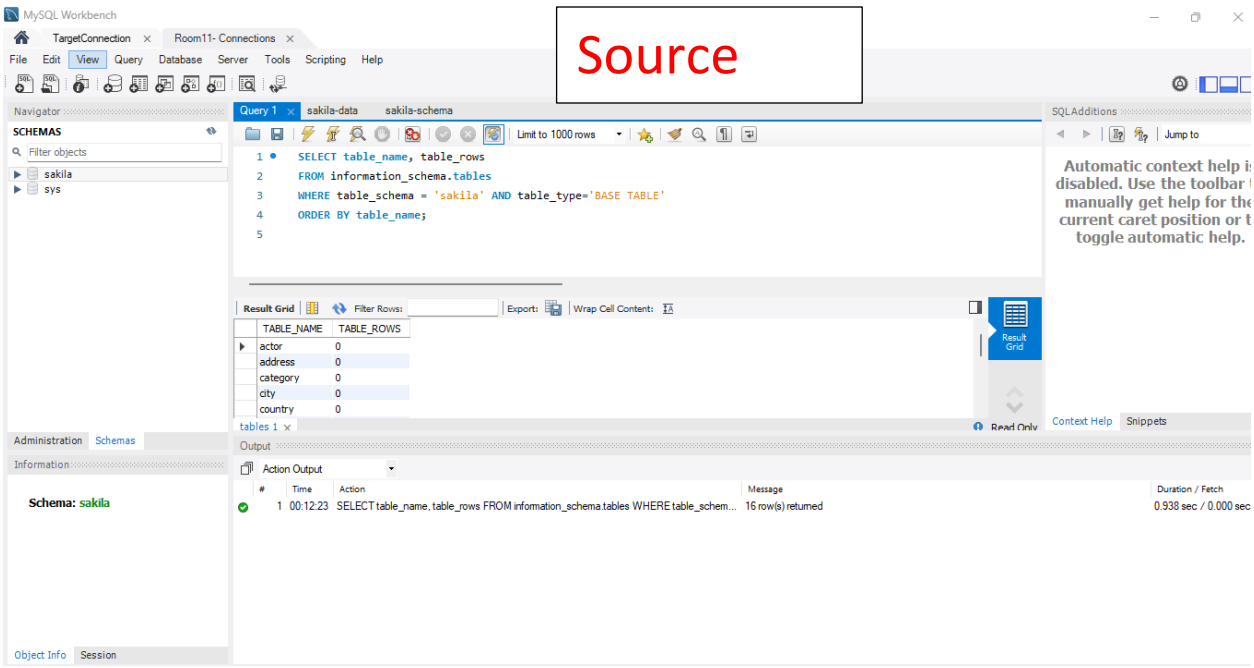
•


File Edit View Database Tools Scripting Help
Setup New Connection
Connection Name: TargetConnection
Type a name for the connection
Connection Method: Standard (TCP/IP)
Method to use to connect to the RDBMS
Parameters SSL Advanced
Hostname: 127.0.0.1
Port: 3306
Name or IP address of the server host - and TCP/IP port.
Username: room12source.c6tcuy46ovku.us-east-1.rds.amazonaws.com
Name of the user to connect with.
Password: Store in Vault ... Clear
The user's password. Will be requested later if it's not set.
Default Schema:
The schema to use as default schema. Leave blank to select it later.
Configure Server Management...
Test Connection
Cancel
OK
MySQL Connections
Local instance MySQL80
root
localhost:3306


MySQL Workbench
File Edit View Database Tools Scripting Help
Setup New Connection
Connection Name: TargetConnection
Type a name for the connection
Connection Method: Standard (TCP/IP)
Method to use to connect to the RDBMS
Parameters SSL Advanced
Hostname: .uy46ovku.us-east-1.rds.amazonaws.com
Port: 3306
Name or IP address of the server host - and TCP/IP port.
Username: admin
Name of the user to connect with.
Password: Store in Vault ... Clear
The user's password. Will be requested later if it's not set.
Default Schema:
The schema to use as default schema. Leave blank to select it later.
Configure Server Management...
Test Connection
Cancel
OK
MySQL Connections
Local instance MySQL80
root
localhost:3306


MySQL Workbench
File Edit View Database Tools Scripting Help
Setup New Connection
Connection Name: TargetConnection
Type a name for the connection
Connection Method:
Method to use to connect to the RDBMS
Parameters SSL Advanced
Hostname:
Username: admin
Password:
Store in Vau
Default Schema:
MySQL Workbench
Successfully made the MySQL connection
Information related to this connection:
Host: room12source.c6tcuy46ovku.us-east-1.rds.amazonaws.c...
Port: 3306
User: admin
SSL: enabled with TLS_AES_256_GCM_SHA384
A successful MySQL connection was made with the parameters defined for this connection.
OK
MySQL Connections
Local instance MySQL80
root
localhost:3306
Configure Server Management...
Test Connection
Cancel
OK


MySQL Workbench
TargetConnection
File Edit View Query Database Server Tools Scripting Help
Target
Navigator
SCHEMAS
Filter objects
awsdms_control
sakila
Tables
actor
address
category
city
country
customer
film
film_actor
film_category
film_text
inventory
language
payment
rental
staff
Administration Schemas
Information
Schema: sakila
Query 1 sakila 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 Filter Rows: Export: Wrap Cell Content:
TABLE_NAME TABLE_ROWS
actor 0
address 0
category 0
city 0
country 0
tables 1
Read Only
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
1 00:07:46 SELECT table_name, table_rows FROM information_schema.tables WHERE table_schem... 16 row(s) returned 0.875 sec / 0.000 sec


Source
MySQL Workbench
TargetConnection
Room11- Connections
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sakila
sys
Query 1
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
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
TABLE_NAME TABLE_ROWS
actor 0
address 0
category 0
city 0
country 0
Result Grid
Read Only
tables 1
Context Help
Snippets
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