Introduction to AWS Database Migration Service

**SPL-37 - Version 1.5.24**

© 2023 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information in the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab overview**

This lab provides you with a basic understanding of the AWS Database Migration Service. In this lab you migrate data from a MySQL database running on an Amazon EC2 instance to an Amazon Aurora RDS instance.

OBJECTIVES

By the end of this lab, you will be able to do the following:

* Connect to a pre-created Amazon EC2 instance.
* Configure MySQL Server as your source database for migration.
* Connect to a pre-created Amazon Aurora instance.
* Migrate data from your MySQL server to your Aurora instance.
* Verify that your migration was successful.

TECHNICAL KNOWLEDGE PREREQUISITES

To successfully complete this lab, you should be familiar with database principles and SQL, be comfortable using RDP to connect to a Windows server, and be familiar with Amazon Aurora.

DURATION

This lab will require **90** minutes to complete.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* **Command:** A command that you must run.
* **Note:** A hint, tip, or important guidance.
* **Additional information:** Information of special interest or importance.
* **Expected output:** A sample output that you can use to verify the output of a command or edited file.
* **Conclusion:** A conclusion or summary point in the lab.

SERVICES USED IN THIS LAB

**AWS Database Migration Service**

AWS Database Migration Service helps you migrate databases to AWS easily and securely. With AWS Database Migration Service, the source database remains fully operational during the migration, minimizing downtime to applications that rely on the database. The AWS Database Migration Service can migrate your data to and from the most widely used commercial and open-source databases. The service supports homogenous migrations such as Oracle to Oracle, as well as heterogeneous migrations between different database platforms, such as Oracle to Amazon Aurora or Microsoft SQL Server to MySQL.

**Amazon Aurora**

Amazon Aurora is a fully managed, MySQL-compatible relational database engine that combines the speed and availability of high-end commercial databases with the simplicity and cost-effectiveness of open source databases. Amazon Aurora provides up to five times better performance than MySQL with the security, availability, and reliability of a commercial database at one tenth the cost.

**Amazon Elastic Compute Cloud (Amazon EC2)**

Amazon EC2 is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

**Amazon RDS**

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, freeing you up to focus on your applications and business. Amazon RDS provides you with six familiar database engines to choose from, including Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB.

**Additional services and tools**

**MySQL Workbench:** MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. MySQL Workbench is available on Windows, Linux and Mac OS X.

 A windows Server is being prepared for you, which can take some time. While you are waiting, for the addition information on AWS Database Migration Service, see [AWS Database Migration Service](https://aws.amazon.com/dms).

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

 You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

**Task 1: Connect to your Amazon EC2 instance**

In this task, you will connect to your Amazon EC2 instance using AWS Systems Manager Fleet Manager.

CONSOLE-BASED RDP ACCESS TO WINDOWS INSTANCES

To connect to instances using RDP with Fleet Manager.

1. To the left of the instructions you are currently reading, choose  **Download PEM**.
2. Save the file to the directory of your choice.
3. Open the [AWS Systems Manager console](https://console.aws.amazon.com/systems-manager/).
4. In the navigation pane at the left of the page, under  **Node Management**, choose **Fleet Manager**.
5. Select the instance with Node name  **Lab Instance**.
6. From the **Node actions** drop-down list, choose **Connect** and then choose **Connect with Remote Desktop**.
7. For preferred **Authentication type** choose  Key pair.
8. For **Key pair content**, choose the following option  Browse your local machine to select the key pair file.
9. Select **Browse** to upload the **PEM** key from your local directory that is associated with your instance.
10. Select **Connect** .

 You will be prompted with a Networks pop-up window asking: **Do you want to allow your PC to be discoverable by other PCs and devices on this network?** Choose **No**.

 Congratulations! You have now successfully connected to your Amazon EC2 instance.

**Task 2: Configure and connect to your source MySQL database**

In this task, you configure your source MySQL database. Once your source database is configured, you connect to it using the MySQL Workbench application.

TASK 2.1: INSTALL MYSQL ON YOUR INSTANCE

1. On your remote desktop, open (double-click) **mysql-installer-community-5.7.22.1**.
2. On the **License Agreement** page:

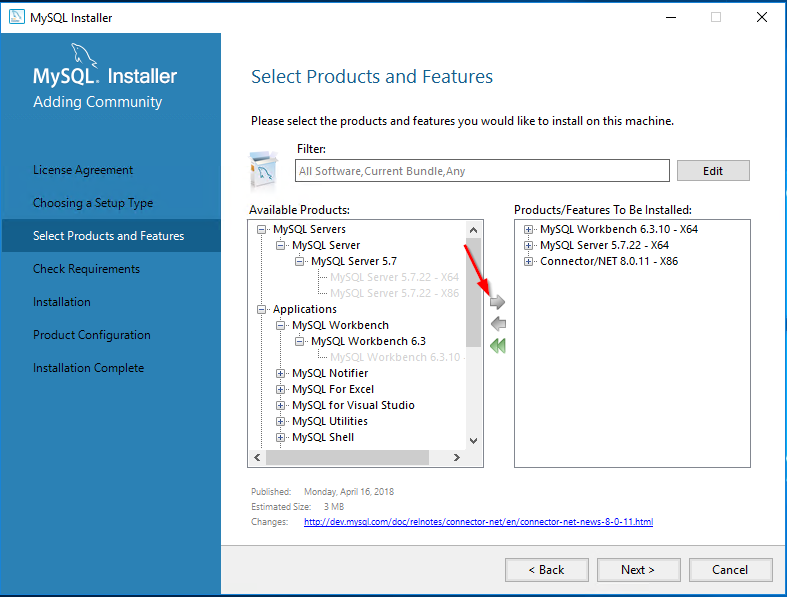
* Select  **I accept the license terms**
* Choose **Next**

1. On the **Choosing a Setup Type** page:

* Select  **Custom**
* Choose **Next**

1. On the **Select Products and Features** page, expand  and select each of the products below one at a time then choose the right arrow to move them to the right window.

* **MySQL Servers:** *MySQL Server 5.7.22 - X64*
* **Applications:** *MySQL Workbench - 6.3.10 - X64*
* **MySQL Connectors:** *Connector/NET 8.0.11 - X86*



1. Choose **Next** .
2. On the Installation page, choose **Execute** .

This will install the server, workbench, and the Connector/NET.

Next, you configure your MySQL server.

1. Choose **Next** until you get to the Accounts and Roles page.
2. On the Accounts and Roles page, configure:

* **MySQL Root Password:**

master123

* **Repeat Password:**

master123

 Copy and paste will sometimes result in command errors. If a command does not work, try manually typing it in the prompt.

1. Choose **Add User** , then configure:

* **Username:**

master

* **Password:**

master123

* **Confirm Password:**

master123

* Choose **OK**

You will use this user to log in and connect to your source database.

1. Choose **Next** until you reach the Apply Configuration page.
2. On the Apply Configuration page, choose **Execute** .

This will configure and start your source SQL server.

1. Choose **Finish** , then choose **Next** .
2. Once the Installation is Complete, choose **Finish** .

This will start your MySQL Workbench so that you can log into your MySQL server.

1. If prompted with an **Unsupported Operating System** popup window:

* Select  **Don’t show this message again**
* Choose **OK**

You will still be able to manage your SQL servers using this OS type.

TASK 2.2: CONNECT AND CONFIGURE YOUR SOURCE MYSQL SERVER

1. Choose the **Local Instance MySQL57** box to connect.
2. On the **Please enter password for the following service** popup, configure:

* **Password:**

master123

* Select  **Save password in vault**
* Choose **OK**

1. In the **Query 1** window, enter:

CREATE DATABASE mydb;

 Copy and paste will sometimes result in command errors. If a command does not work, try manually typing it in the prompt.

1. At the top of the screen, choose **Execute**  .

This will run your query and create the source database.

1. Import data into your database by choosing **Server** > **Data Import**.
2. In the **Import from Disk** window:

* Choose the ellipsis
* Browse to and choose the **dumps** folder if it is not already selected.
* Choose **OK**

1. Choose **Start Import** .

 If you do not see **Start Import**, resize the window to fit in the remote desktop.

The data will be imported into your database.

TASK 2.3: VERIFY YOUR DATA WAS IMPORTED

1. In the left navigation pane, choose the refresh  icon next to **SCHEMAS**.

You should see your  **mydb** database.

1. At the top of the screen, choose the **Query 1** tab, then:

* Delete the existing Query
* Run the following query:

Select \* from mydb.employee;

 Copy and paste will sometimes result in command errors. If a command does not work, try manually typing it in the prompt.

1. At the top of the screen, choose **Execute**  .

 You will see that there is data in your database.

 Congratulations! You have successfully configured your source MySQL server to be used for database migration.

**Task 3: Use MySQL Workbench to connect to your RDS instance**

In this task, you connect to your Aurora database instance. This instance was created for you using CloudFormation when you started the lab. Aurora will be used as your target for migration.

1. Copy the **Ec2InstanceSessionCLI** value from the list to the left of these instructions, and then paste it into a new web browser tab.
2. Copy the **ClusterEndpoint** value from the list to the left of these instructions and paste the value in a text editor.
3. In the instance terminal session, run the following command to copy the **ClusterEndpoint** value to a text file on the Amazon Windows instance’s Desktop. This will be used under this task in later step to configure and connect to the Aurora database instance:

cd C:\Users\Administrator\Desktop\

echo "Replace with the Cluster Endpoint that you copied in the previous step" > ClusterEndpoint.txt

 The copied value should look similar to this:

cd C:\Users\Administrator\Desktop\

echo "labstack-abcdefgh-b8ynt5xgnln9sbbgss-auroracluster-k6j7ggkcmdz4.cluster-cbpgzpfkl3ki.us-west-2.rds.amazonaws.com" > ClusterEndpoint.txt

1. Return to the **Fleet Manager - Remote Desktop** browser tab and on your remote dekstop, open the **ClusterEndpoint.txt** file that you created in the previous step.
2. Now go to the MySQL Workbench and at the top of the screen, choose the **Home**  button.
3. Create a new MySQL connection by choosing the plus  button.
4. In the **Setup New Connection** window, configure the following:

* **Connection Name:**

Aurora

* **Hostname:** Delete **127.0.0.1**. Copy the **ClusterEndpoint** from the text file named **ClusterEndpoint.txt** that you opened in the earlier step and paste in the window.
* **Username:**

master

* Choose **Store in Vault …**
* **Password:**

master123

* Choose **OK**

1. Choose **Test Connection** .

 This will bring up a pop up dialog box to indicate that the connection was successfully made.

1. Choose **OK** to dismiss the connection confirmation dialog box.
2. Create the connection to your Aurora database by choosing **OK** .

 This closes the window and returns you to MySQL Workbench. You should now see two connections.

1. In the **MySQL Workbench** menu, choose **Aurora** to connect to the database.

The **SQL Editor** window opens, showing a successful connection to the database.

1. **Execute**  the following query to verify that no data exists in your Aurora instance.

Select \* from mydb.employee;

 Copy and paste will sometimes result in command errors. If a command does not work, try manually typing it in the prompt.

 You should receive a message showing that **mydb.employee** does not exist.

 Congratulations! You have now successfully connected to your Aurora database using MySQL Workbench.

**Task 4: Migrate your source MySQL database to your Aurora instance using AWS Database Migration Service**

In this task, you use the AWS Database Migration Service to migrate your mydb database running on your MySQL instance to your Aurora instance.

TASK 4.1: CREATE YOUR REPLICATION INSTANCE

The first step in migrating data using AWS Database Migration Service is to create a replication instance. An AWS DMS replication instance runs on an Amazon Elastic Compute Cloud (Amazon EC2) instance. A replication instance provides high availability and failover support using a Multi-AZ deployment.

AWS DMS uses a replication instance that connects to the source data store, reads the source data, and formats the data for consumption by the target data store. A replication instance also loads the data into the target data store. Most of this processing happens in memory. However, large transactions might require some buffering on disk. Cached transactions and log files are also written to disk.

1. At the top of the AWS Management Console, in the search bar, search for and choose

Database Migration Service

.

1. In the left navigation pane, choose **Replication instances**.
2. Choose **Create replication instance** then configure:

* **Name**:

replicationInstance

* **Description**:

replicationInstance

* **Instance class**: *dms.t3.micro*
* **High Availability**: *Dev or test workload(Single-AZ)*
* **Virtual private cloud (VPC) for IPv4**: Lab-VPC
* **Public accessible**: Deselect

1. Choose **Create replication instance** .
2. Wait for the status of your replication instance to display Available.

This will take 5-10 minutes.

TASK 4.2: CREATE YOUR SOURCE ENDPOINT

An endpoint provides connection, data store type, and location information about your data store. AWS Database Migration Service uses this information to connect to a data store and migrate data from a source endpoint to a target endpoint. You can specify additional connection attributes for an endpoint by using extra connection attributes. These attributes can control logging, file size, and other parameters; for more information about extra connection attributes, see the documentation section for your data store.

1. In the left navigation pane, choose **Endpoints**.
2. Choose **Create endpoint** then configure:

* **Source endpoint**
* **Endpoint identifier:**

MySQL

* **Source engine:** MySQL
* **Access to endpoint database:**  *Provide access information manually*
* **Server name**: Paste the value of **WindowsPrivateIP** located to the left of these instructions
* **Port:**

3306

* **User name**:

master

* **Password**:

master123

1. Expand  **Test endpoint connection (optional)**, then configure:

* **VPC:** *Lab-VPC*
* Choose **Run test**

 You may have to wait a few minutes for your replication instance to become active.

1. Once your test is successful, choose **Create endpoint** .

TASK 4.3: CREATE YOUR TARGET ENDPOINT TO YOUR AURORA INSTANCE

1. At the top of the screen, choose **Create endpoint** , then configure:

* Select  **Target endpoint**
* Check  **Select RDS DB instance**
* For **RDS Instance**, select the RDS instance that appears.

1. In the **Endpoint configuration** section, configure:

* **Endpoint identifier:**

aurora

* **Access to endpoint database:**  *Provide access information manually*
* **User name:**

master

* **Password:**

master123

1. Expand  **Test endpoint connection**, then configure:

* **VPC:** *Lab-VPC*
* Choose **Run test**

1. Once your test is successful, choose **Create endpoint** .

TASK 4.4: CREATE A DATABASE MIGRATION TASK

An AWS Database Migration Service (AWS DMS) task is where all the work happens. You use tasks to migrate data from the source endpoint to the target endpoint, and the task processing is done on the replication instance. You specify what tables and schemas to use for your migration and any special processing, such as logging requirements, control table data, and error handling.

1. In the left navigation pane, choose **Database migration tasks**.
2. Choose **Create task** then configure:

* **Task identifier:**

MySQL-Aurora

* **Replication instance:** Select your replication instance
* **Source database endpoint:** *mysql*
* **Target database endpoint:** *aurora*
* **Migration type:** *Migrate existing data*

1. Under **Table mappings**, choose **Add new selection rule** and then configure:

* **Schema:** *Enter a schema*
* **Source name:**

mydb

 Table mapping uses several types of rules to specify the data source, source schema, data, and any transformations that should occur during the task. You can use table mapping to specify individual tables in a database to migrate and the schema to use for the migration. In addition, you can use filters to specify what data from a given table column you want replicated and you can use transformations to modify the data written to the target database.

1. Choose **Create task** .
2. Wait for the **Status** of your task to change from *Creating* to  *Load complete*.
3. Choose your **mysql-aurora** task.
4. Choose the **Table statistics** tab.

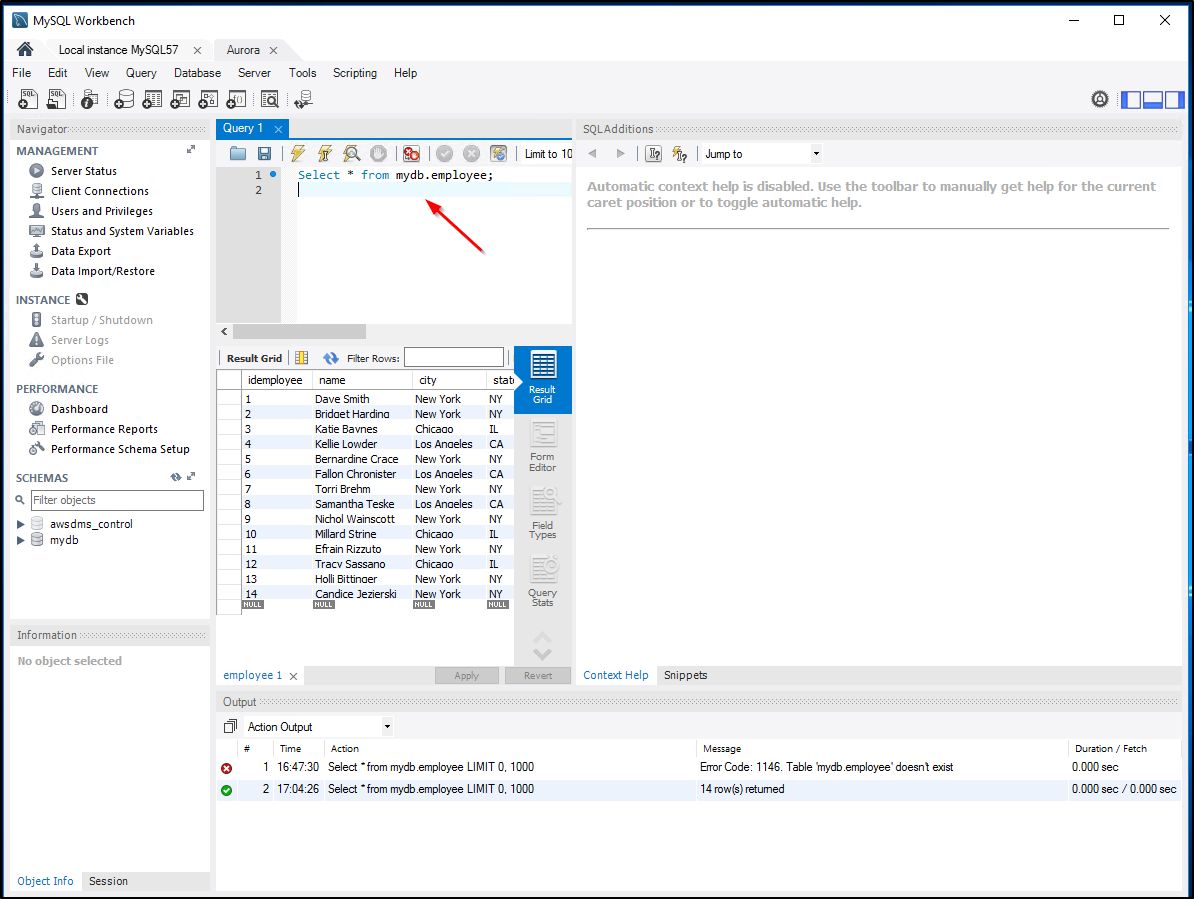
 This shows the table statistics for your database migration task. You should see that 3 tables were loaded.

1. Return to the **Fleet Manager - Remote Desktop** browser tab. In **MySQL Workbench**, on the Aurora connection tab, **Execute** :

Select \* from mydb.employee;

 Copy and paste will sometimes result in command errors. If a command does not work, try manually typing it in the prompt.

 You will see that the your mydb data has been migrated to your Aurora instance.



 Congratulations! You have now successfully used the AWS Database Migration Service to migrate data from your source MySQL server to your target Aurora instance.

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Conclusion**

 Congratulations! You have now successfully:

* Connected to a pre-created Amazon EC2 instance
* Configured MySQL Server as your source database for migration
* Connected to a pre-created Amazon Aurora instance
* Migrated data from your MySQL server to your Aurora instance
* Verified that your migration was successful

**Additional Resources**

* [AWS Database Migration Service Documentation](https://aws.amazon.com/documentation/dms/)
* [Working with AWS DMS Tasks](https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Tasks.html)
* [Working with an AWS DMS replication instance](https://docs.aws.amazon.com/dms/latest/userguide/CHAP_ReplicationInstance.html)
* [Using table mapping to specify task settings](https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Tasks.CustomizingTasks.TableMapping.html)
* [Connect using Remote Desktop](https://docs.aws.amazon.com/systems-manager/latest/userguide/fleet-rdp.html)

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).