



MIGRATE RDS MYSQL TO SKYSQL MARIADB USING AMAZON DATA MIGRATION SERVICE

HOW TO MIGRATE FROM RDS TO SKYSQL WITH LITTLE/NO DOWNTIME?



This is probably the most often asked question to date. Fortunately, AWS makes this quite easy. By following the steps detailed here, migrating data and cutting over to SkySQL can be painless. This document is intended for existing AWS RDS customers looking to migrate to MariaDB SkySQL.

ASSUMPTIONS / PREREQUISITES

Amazon Web Services Database Migration Service will hereon be called AWS DMS.

This demonstration assumes an AWS MariaDB or MySQL RDS instance already exists and has a read replica. For more information:

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html#USER_ReadRepl.Create.

AWS DMS requires an AWS IAM user. The AWS documentation states the user must have the "AdministratorAccess" role. However, this link gives an alternative without giving admin: https://github.com/awsdocs/aws-dms-user-guide/blob/master/doc_source/CHAP_Security.IAMPermissions.md. In addition to those roles outlined, it was also necessary to add the "ListAttachedRolePolicies" role for this demonstration.

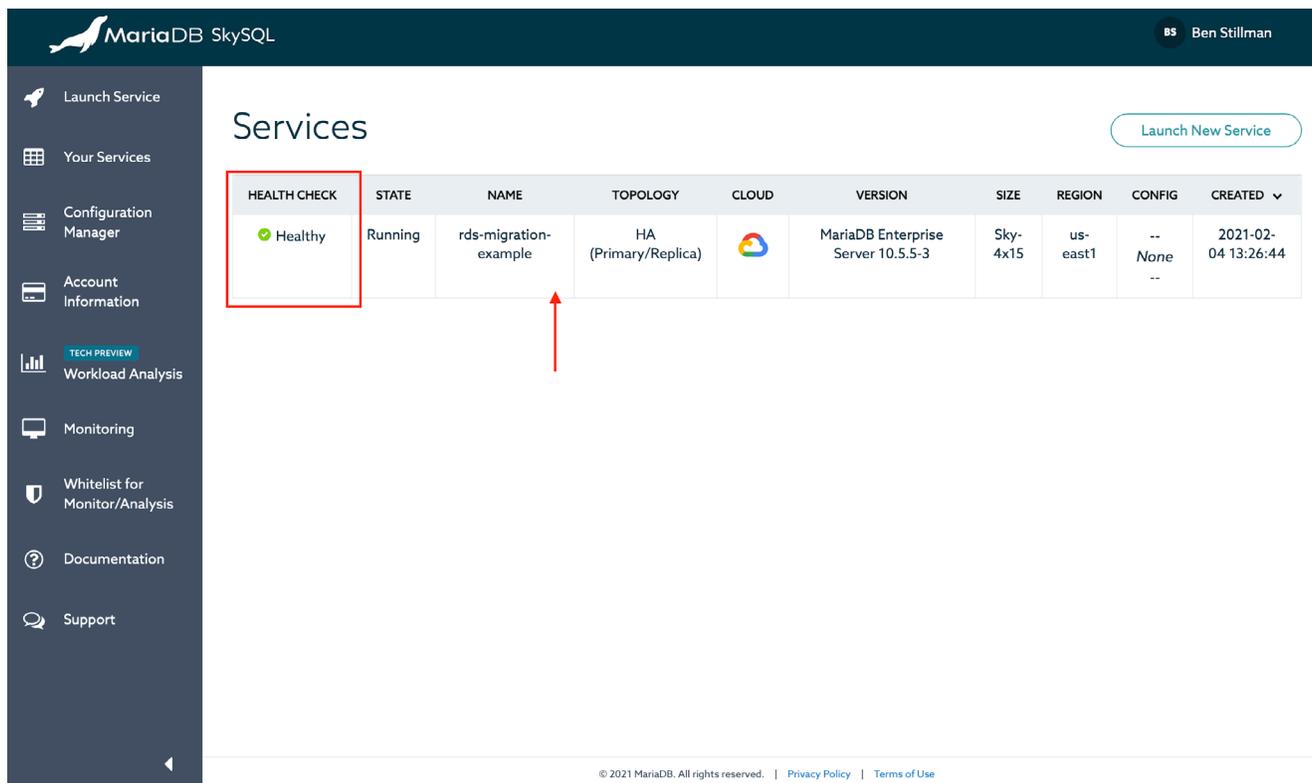
This demonstration assumes a MariaDB SkySQL service has already been started. For more information, see <https://mariadb.com/products/skysql/docs/get-started/>.

Lastly, it is assumed that the SkySQL service's default password has been changed as recommended.

Let's get started!

GATHER INFORMATION FROM DESTINATION SKYSQL SERVICE

Assuming the SkySQL service is created and the "Health Check" column reports the service as "Healthy", click anywhere on the service's row to open the service details. For this demonstration, a "Transactions" topology "HA (Primary/Replica)" service was created within the Google Cloud Platform using "Sky-4x15" instance sizes.



The screenshot shows the MariaDB SkySQL console interface. The top navigation bar includes the MariaDB logo and the user name 'Ben Stillman'. A sidebar on the left contains various navigation options. The main content area displays a table of services. The 'HEALTH CHECK' column for the 'rds-migration-example' service is highlighted with a red box, and a red arrow points to the service name.

HEALTH CHECK	STATE	NAME	TOPOLOGY	CLOUD	VERSION	SIZE	REGION	CONFIG	CREATED
Healthy	Running	rds-migration-example	HA (Primary/Replica)		MariaDB Enterprise Server 10.5.5-3	Sky-4x15	us-east1	-- None	2021-02-04 13:26:44

From the service details page, some information needs to be gathered and noted for later use. This includes the service's "Fully Qualified Domain Name" and "Read-Write Port".

For this demonstration, these are:

Fully Qualified Domain Name: rds-migration-example.mdb0001941.db.skysql.net
Read-Write Port: 5001

In addition, verify the IP which this service will be accessed from has been [whitelisted](#).

The screenshot shows the MariaDB SkySQL console interface. The top navigation bar includes the MariaDB SkySQL logo and the user name 'Ben Stillman'. A left sidebar contains navigation options: Launch Service, Your Services, Configuration Manager, Account Information, Workload Analysis (marked as TECH PREVIEW), Monitoring, Whitelist for Monitor/Analysis, Documentation, and Support. The main content area displays the service 'rds-migration-example' with a 'Healthy' status and a 'Launch New Service' button. Below the service name are three buttons: 'Show Credentials', 'Stop Service', and 'Delete Service'. A table provides service specifications:

TOPOLOGY	CLOUD	REGION	SIZE	STORAGE	READ-WRITE PORT	READ-ONLY PORT	REPLICAS
HA (Primary/Replica)	Google Cloud	us-east1	Sky-4x15	100 GB	5001	5002	2

Below the table, the 'Fully Qualified Domain Name' is shown as 'rds-migration-example.mdb0001941.db.skysql.net'. The 'Custom Configuration' section indicates 'No custom configuration currently applied to this service'. The 'Whitelisted IP Addresses' section shows a redacted IP address. A 'Request Workload Analysis' button is located at the bottom of the main content area. The footer contains copyright information: '© 2021 MariaDB. All rights reserved. | Privacy Policy | Terms of Use'.

Next, click on “Show Credentials”.

This screenshot is identical to the one above, but with a red arrow pointing to the 'Show Credentials' button. The rest of the interface, including the sidebar, service details, and footer, remains the same.

Note the "Username" and connection string ("Connect using MariaDB CLI"). This demonstration assumes the "Default Password" has been changed, so it does not need to be noted. Also download the "Certificate authority chain" by clicking "Download".

The screenshot shows the MariaDB SkySQL console interface. A modal dialog titled "Temporary Service Login Credentials" is displayed in the center. The dialog contains the following information:

- Login Credentials:**
 - Username: DB00003785
 - Default Password: VE1#TmchRadshbrim)Z7sTIHmg
 - Certificate authority chain: [Download](#) (with a red arrow pointing to the link)
- 1. Whitelist IP addresses:**

Please remember to "whitelist" your IP address from the services page. For further info please refer to [documentation](#).
- 2. Connect using MariaDB CLI:**

```
mariadb --host rds-migration-example.mdb0001941.db.skysql.net --port 5001 --user DB00003785 -p --ssl-ca ~/Downloads/skysql_chain.pem
```

Note: If you use a MariaDB client older than 10.4, please replace "mariadb" with "mysql"
- 3. Change the default password:**

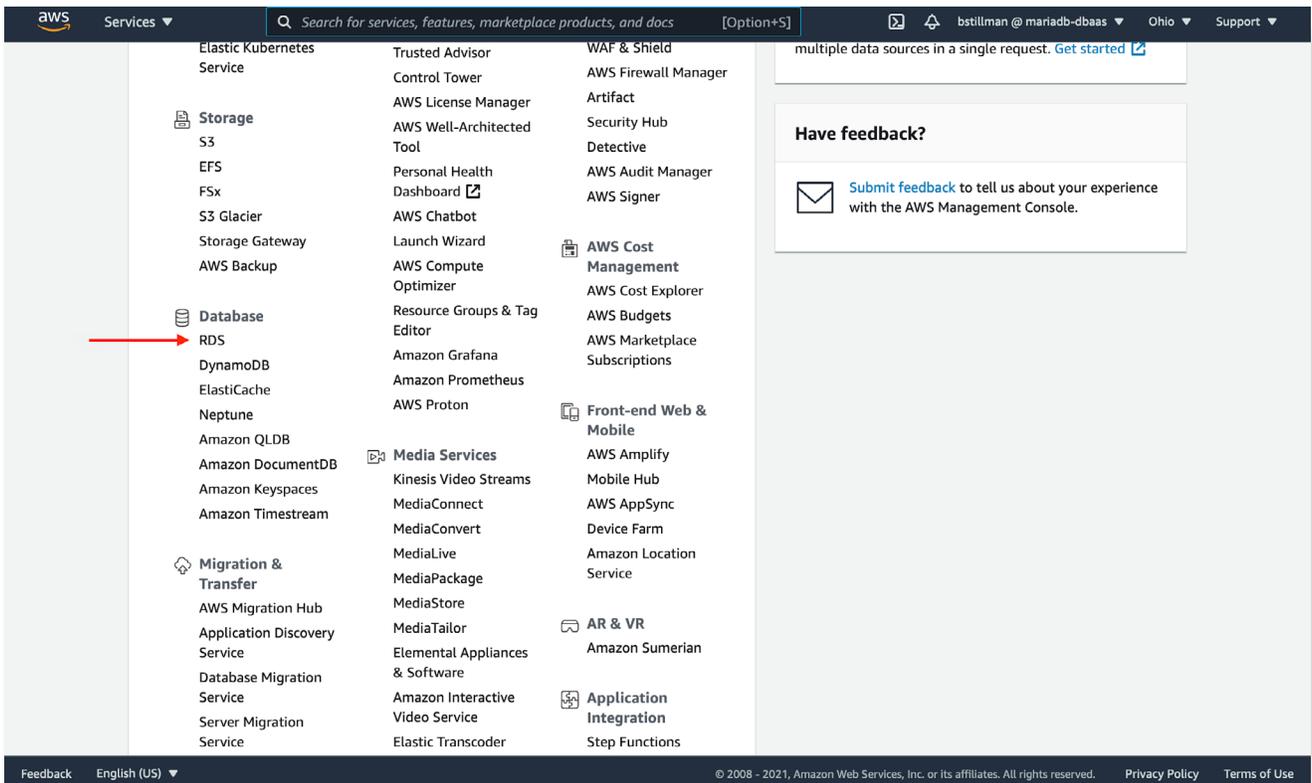
```
SET PASSWORD FOR 'DB00003785'@'%' = PASSWORD('newpass');
```

A "Close" button is located at the bottom right of the dialog. The background shows the console interface with a sidebar on the left and a main content area on the right. The user's name "Ben Stillman" is visible in the top right corner of the console.

GATHER INFORMATION FROM SOURCE RDS INSTANCE

Next, gather the necessary information from the RDS instance which is to be migrated.

From the AWS Portal, under “Databases”, click “RDS”.



Next, click “DB Instances”.

The screenshot shows the Amazon RDS console interface. On the left is a navigation sidebar with categories like Dashboard, Databases, Query Editor, Performance Insights, Snapshots, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, Recommendations (1), and Certificate update. The main content area is titled 'Amazon Aurora' and includes a 'Create database' button and a link to 'Restore Aurora DB cluster from S3'. Below this is a 'Resources' section with a 'Refresh' button and a summary of resources in the US East (Ohio) region. A red arrow points to the 'DB Instances (1/40)' link. The 'DB Instances' section lists: Allocated storage (0.02 TB/100 TB), Click here to increase DB instances limit, DB Clusters (0/40), Reserved instances (0/40), Snapshots (4) - Manual (1/100) and Automated (3), Recent events (4), Event subscriptions (0/20), Parameter groups (7) - Default (5) and Custom (2/100), Option groups (4) - Default (4) and Custom (0/20), Subnet groups (4/50), Supported platforms VPC, and Default network vpc-50f77938. To the right is a 'Recommended for you' section with cards for 'RDS Proxy - Now Available', 'Database Performance Tuning', 'Aurora Global Database', and 'Build a Serverless Database'. The footer contains 'Feedback', 'English (US)', and copyright information.

From here, click on the “DB identifier” of the instance to be migrated. For this demonstration, it’s “stillman-dms-sky”.

The screenshot shows the Amazon RDS console 'Databases' page. The breadcrumb is 'RDS > Databases'. There are buttons for 'Group resources', 'Modify', 'Actions', 'Restore from S3', and 'Create database'. A search bar is labeled 'Filter databases'. Below is a table with columns: DB identifier, Role, Engine, Region & AZ, and Size. The table contains one entry: 'stillman-dms-sky' (Instance, MySQL Community, us-east-2b, db.t2.micro). A red arrow points to the 'stillman-dms-sky' identifier. The footer contains 'Feedback', 'English (US)', and copyright information.

From this page, note the “Endpoint” and “Port”.

The screenshot shows the AWS Management Console for an Amazon RDS instance named 'stillman-dms-sky'. The 'Summary' section displays the following details:

DB identifier stillman-dms-sky	CPU 2.07%	Status Available	Class db.t2.micro
Role Instance	Current activity 0 Connections	Engine MySQL Community	Region & AZ us-east-2b

The 'Connectivity & security' section is expanded, showing the 'Endpoint & port' details:

Endpoint stillman-dms-sky.cmdxr9mqv3ld.us-east-2.rds.amazonaws.com	Port 3306
---	--------------

The 'Networking' section shows:

Availability zone us-east-2b	VPC vpc-04791123e8a890d1e	Subnet group default-vpc-04791123e8a890d1e
---------------------------------	------------------------------	---

The 'Security' section shows:

VPC security groups ben-benchmark-08282019 (sg-096d9ff718b4e8078) (active)	Public accessibility Yes	Certificate authority
---	-----------------------------	-----------------------

Click on “Configuration”. Find and note the Amazon Resource Name (ARN).

The screenshot shows the AWS Management Console for the 'stillman-dms-sky' RDS instance, with the 'Configuration' tab selected. The 'Instance' section displays the following details:

Configuration	Instance class	Storage	Performance Insights
DB instance id stillman-dms-sky	Instance class db.t2.micro	Encryption Not Enabled	Performance Insights enabled No
Engine version 5.7.31	vCPU 1	Storage type General Purpose (SSD)	
DB name -	RAM 1 GB	IOPS -	
License model General Public License	Availability	Storage 20 GiB	
Option groups default:mysql-5-7	Master username admin	Storage autoscaling Enabled	
Amazon Resource Name (ARN) arn:aws:rds:us-east-2:347119114792:db:stillman-dms-sky	IAM db authentication Not Enabled	Maximum storage threshold 1000 GiB	
	Multi-AZ No		

Now that the necessary information has been gathered, next is to configure the RDS instance and SkySQL service.

CONFIGURE THE RDS INSTANCE

In order for AWS DMS to function properly, a few things need to be configured on the RDS instance to be migrated.

First, change the binary logging format from MIXED to ROW. For this demonstration, a new "Parameter Group" was created named "dms0testing" because this demonstration instance did not have a pre-existing "Parameter Group". The only change made was setting "binlog_format" to "ROW". (Unfortunately, although "binlog_format" is a dynamic variable, it cannot be set dynamically in RDS.)

The screenshot shows the AWS Management Console interface for configuring an Amazon RDS Parameter Group. The breadcrumb navigation indicates the path: RDS > Parameter groups > dms0testing. The main content area is titled 'dms0testing' and contains a 'Parameters' section with a search bar and an 'Edit parameters' button. A table lists the parameters, with 'binlog_format' highlighted in a red box, showing its value as 'ROW'. Below the parameters table is a 'Recent events' section with a search bar and a 'Tags (0)' section with a search bar. The left sidebar shows the navigation menu with 'Parameter groups' highlighted and a red arrow pointing to it. The footer contains 'Feedback', 'English (US)', and copyright information.

<input type="checkbox"/>	Name	Values	Allowed values	Modifiable	Source	Apply type	Data type
<input type="checkbox"/>	binlog_format	ROW	ROW, STATEMENT, MIXED	true	user	dynamic	string

Modify the RDS instance to use the new "Parameter Group". Note this change must take effect before starting the AWS DMS "Database Migration Task" later.

Connect to the AWS RDS instance.

```
mysql --host stillman-dms-sky.cmdxr9mqv3ld.us-east-2.rds.amazonaws.com --Port 3306
--user admin -p
Enter password:
```

Verify the variable “binlog_format” has been correctly applied.

```
SHOW GLOBAL VARIABLES LIKE 'binlog_format';
```

Next, configure the binary logs to not expire for at least 24 hours. This is done via the command-line interface.

Use rds_set_configuration to set the binlog retention.

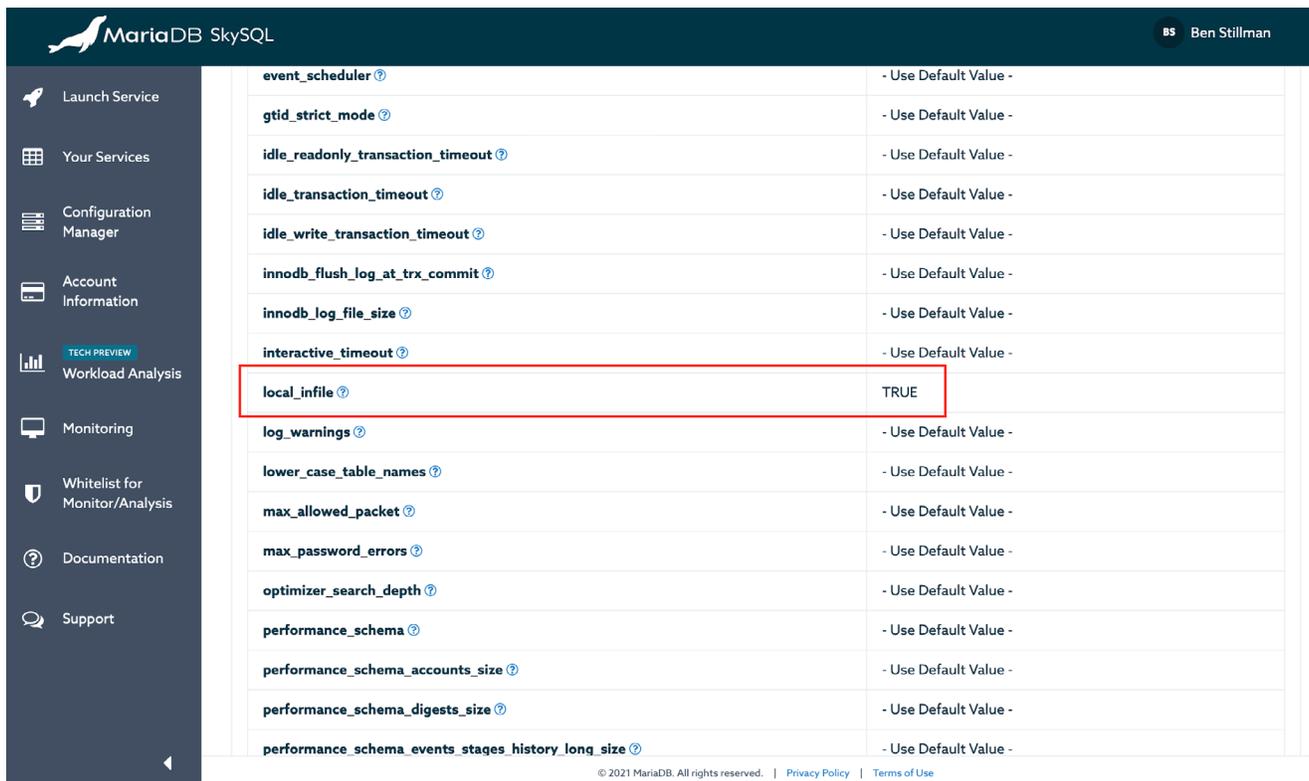
```
CALL mysql.rds_set_configuration('binlog retention hours', 24);
```

CONFIGURE THE SKYSQL SERVICE

AWS DMS uses `LOAD DATA LOCAL INFILE` for the initial data load process. By default, the variable `LOCAL_INFILE` is disabled in SkySQL. For this demonstration, a new configuration was created in SkySQL using the “Configuration Manager” named “DMS Testing”. Only `LOCAL_INFILE` was modified. If the SkySQL target instance already has a custom configuration, that configuration can be edited instead of creating a new custom configuration.

For more information about “Configuration Manager” and applying custom configurations to SkySQL services, see <https://mariadb.com/products/skysql/docs/instructions/configuration-manager/>.

This change must take effect before starting the AWS DMS “Database Migration Task” later.



The screenshot shows the MariaDB SkySQL Configuration Manager interface. The left sidebar contains navigation options: Launch Service, Your Services, Configuration Manager, Account Information, Workload Analysis (marked as TECH PREVIEW), Monitoring, Whitelist for Monitor/Analysis, Documentation, and Support. The main content area displays a list of configuration variables. The variable `local_infile` is highlighted with a red box, showing its value as `TRUE`. Other variables listed include `event_scheduler`, `gtid_strict_mode`, `idle_readonly_transaction_timeout`, `idle_transaction_timeout`, `idle_write_transaction_timeout`, `innodb_flush_log_at_trx_commit`, `innodb_log_file_size`, `interactive_timeout`, `log_warnings`, `lower_case_table_names`, `max_allowed_packet`, `max_password_errors`, `optimizer_search_depth`, `performance_schema`, `performance_schema_accounts_size`, `performance_schema_digests_size`, and `performance_schema_events_stages_history_long_size`. The top right corner of the interface shows the user name "Ben Stillman".

Connect to the SkySQL service via the command-line interface.

```
mariadb --host rds-migration-example.mdb0001941.db.skysql.net --port 5001 --user
DB00003785 -p --ssl
Enter password:
```

Verify the variable “local_infile” has been correctly applied.

```
SHOW GLOBAL VARIABLES LIKE 'local_infile';
```

CREATE USER FOR AWS DMS CONNECTION ON SKYSQL SERVICE

It is best practice to have separate users for each application or action. For this demonstration, create a user to be used exclusively for AWS DMS to connect to the SkySQL service.

Create the AWS DMS user.

```
CREATE USER `dmsuser`@'%'  
IDENTIFIED BY `6Hvmw@GQERbRygNx#j3dMwN!rrH2t2v&WH`;
```

Grant the appropriate privileges to the AWS DMS user for the database to be migrated. For this demonstration, the database which will be migrated is named "dmstestdb". This database does not yet exist within the SkySQL service.

```
GRANT ALTER, CREATE, DROP, INDEX, INSERT, UPDATE, DELETE, SELECT  
ON dmstestdb.*  
TO `dmsuser`@'%' ;
```

Grant the appropriate privileges to the AWS DMS user for the "awsdms_control" database used by AWS DMS to keep track of the replication status.

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY TABLES,  
LOCK TABLES, EXECUTE, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EVENT,  
TRIGGER,  
DELETE HISTORY  
ON awsdms_control.*  
TO `dmsuser`@'%' ;
```

CREATE DATABASE SCHEMA ON SKYSQL

While connected to the SkySQL service, create the database and table(s) that will be migrated from RDS. This is done directly on the SkySQL service rather than allowing AWS DMS to create them because AWS DMS only creates the most basic structure to migrate the data. Thus things like auto-increments are left out.

```
CREATE DATABASE testing;
CREATE TABLE TableA (
  id int(11) NOT NULL AUTO_INCREMENT,
  uuid_junk varchar(50) NOT NULL,
  create_ts timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
);
```

CREATE USER FOR AWS DMS CONNECTION ON RDS INSTANCE

Following best practice, create an AWS DMS specific user on the RDS instance to be migrated. This user will be used by AWS DMS to connect to the RDS instance.

Connect to the AWS RDS instance.

```
mysql --host stillman-dms-sky.cmdxr9mqv3ld.us-east-2.rds.amazonaws.com --Port 3306  
--user admin -p  
Enter password:
```

Create the AWS DMS user.

```
CREATE USER 'dmsuser'@'%'  
IDENTIFIED BY 'kfjR67HEWFklg98FGE#hyghdfp7@!';
```

Grant the necessary privileges to the AWS DMS user.

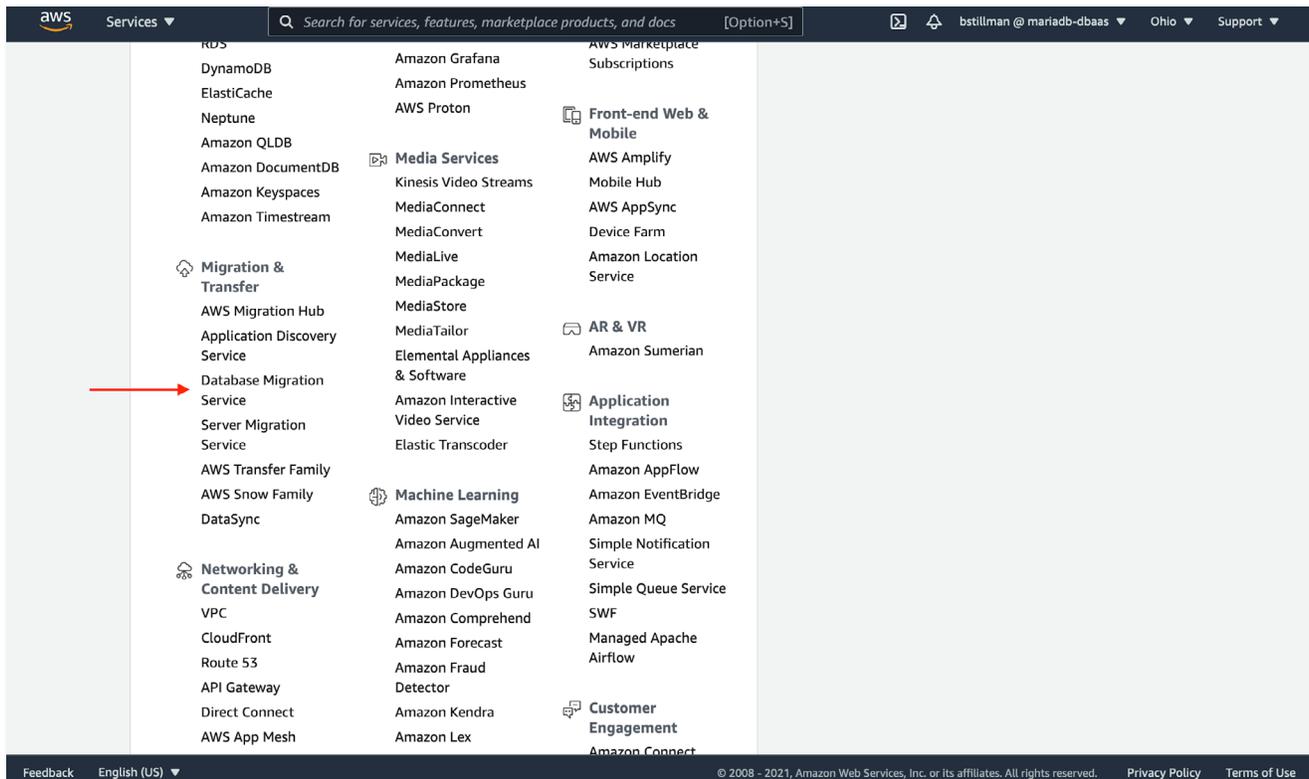
```
GRANT REPLICATION CLIENT, REPLICATION SLAVE  
ON *.* TO 'dmsuser'@'%';
```

At this point, the AWS DMS user is created on both the source (AWS RDS instance) and the target (SkySQL service).

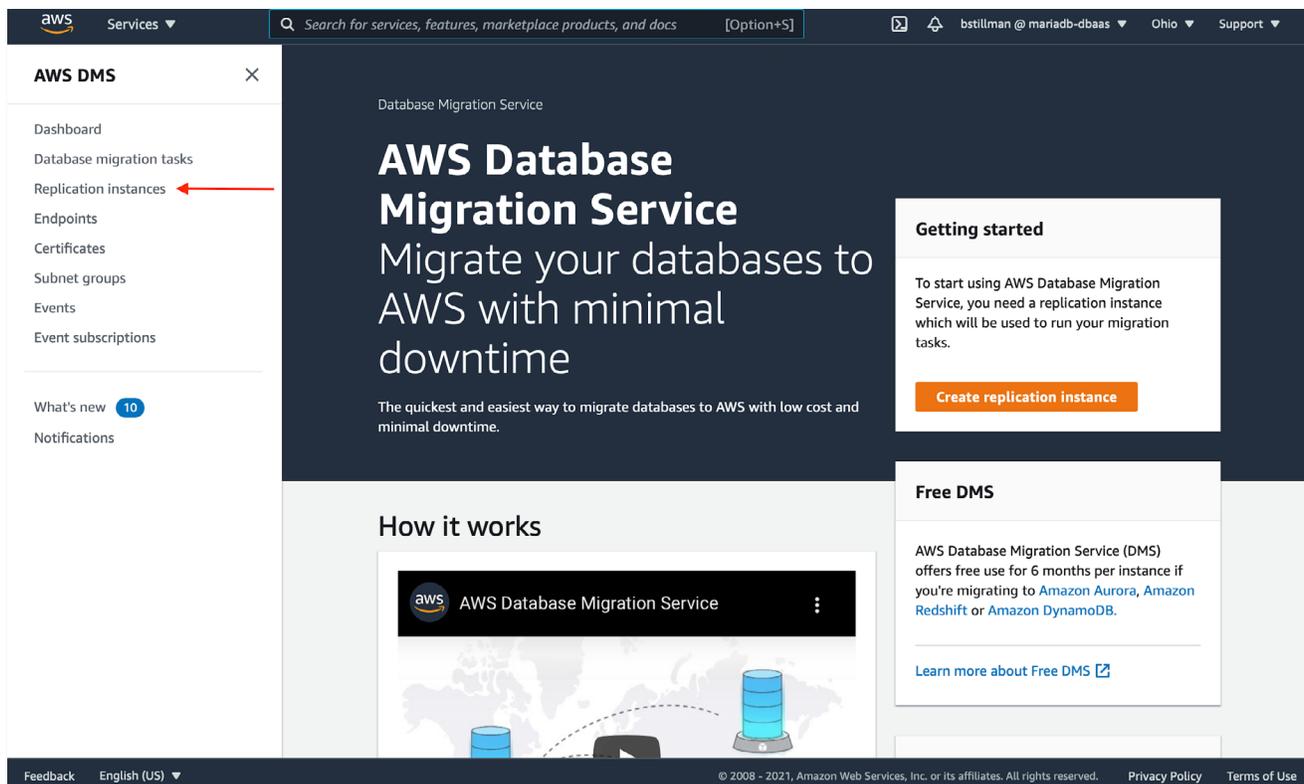
CREATE AWS DMS REPLICATION INSTANCE

In order to begin setting up AWS DMS to migrate the RDS database to SkySQL, a “Replication instance” needs to be created.

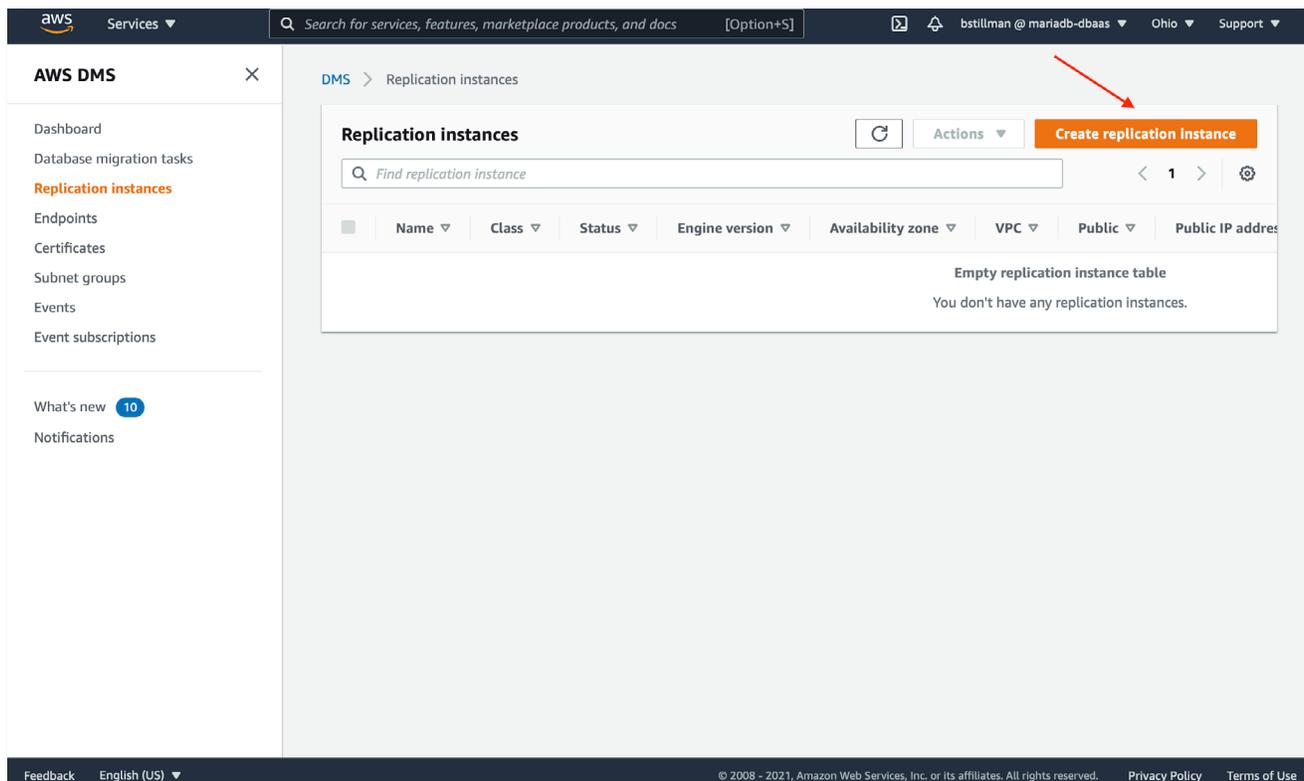
From the AWS Console, click on “Database Migration Service” under “Migration & Transfer”.



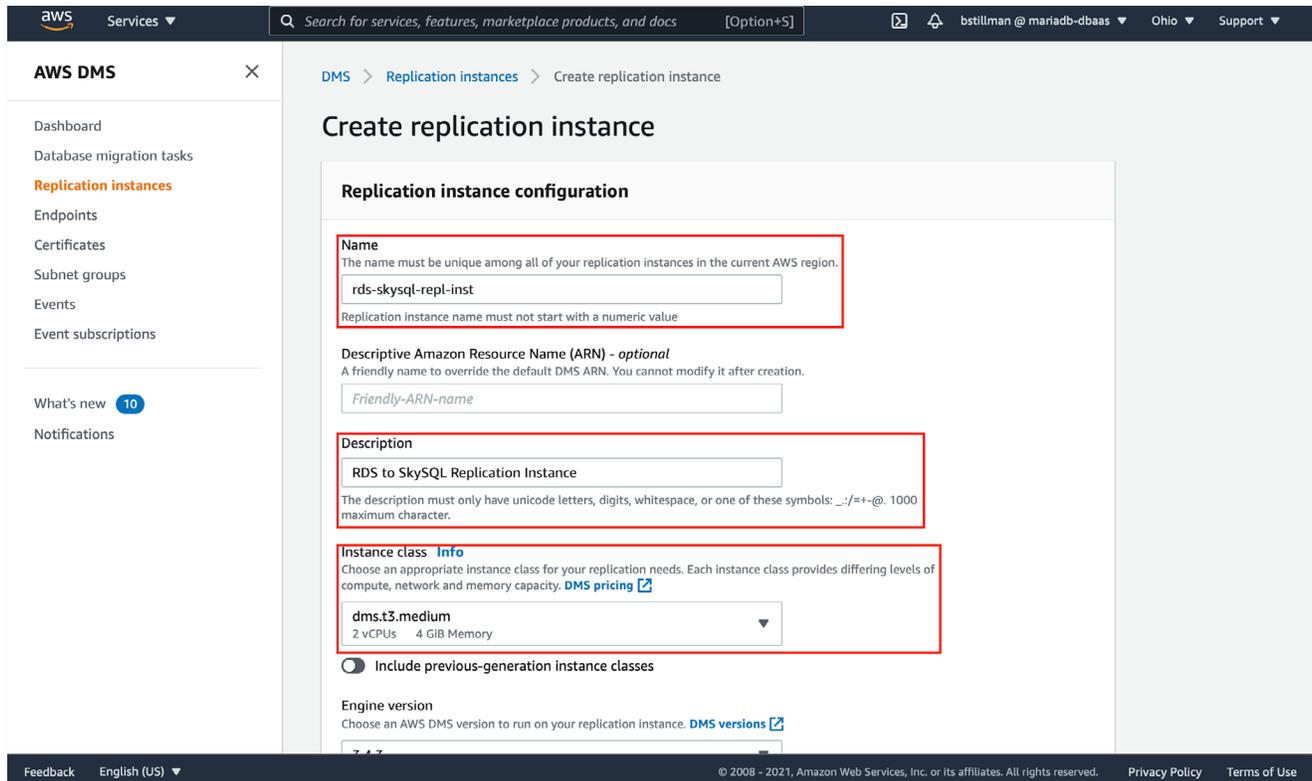
Click on “Replication instances”.



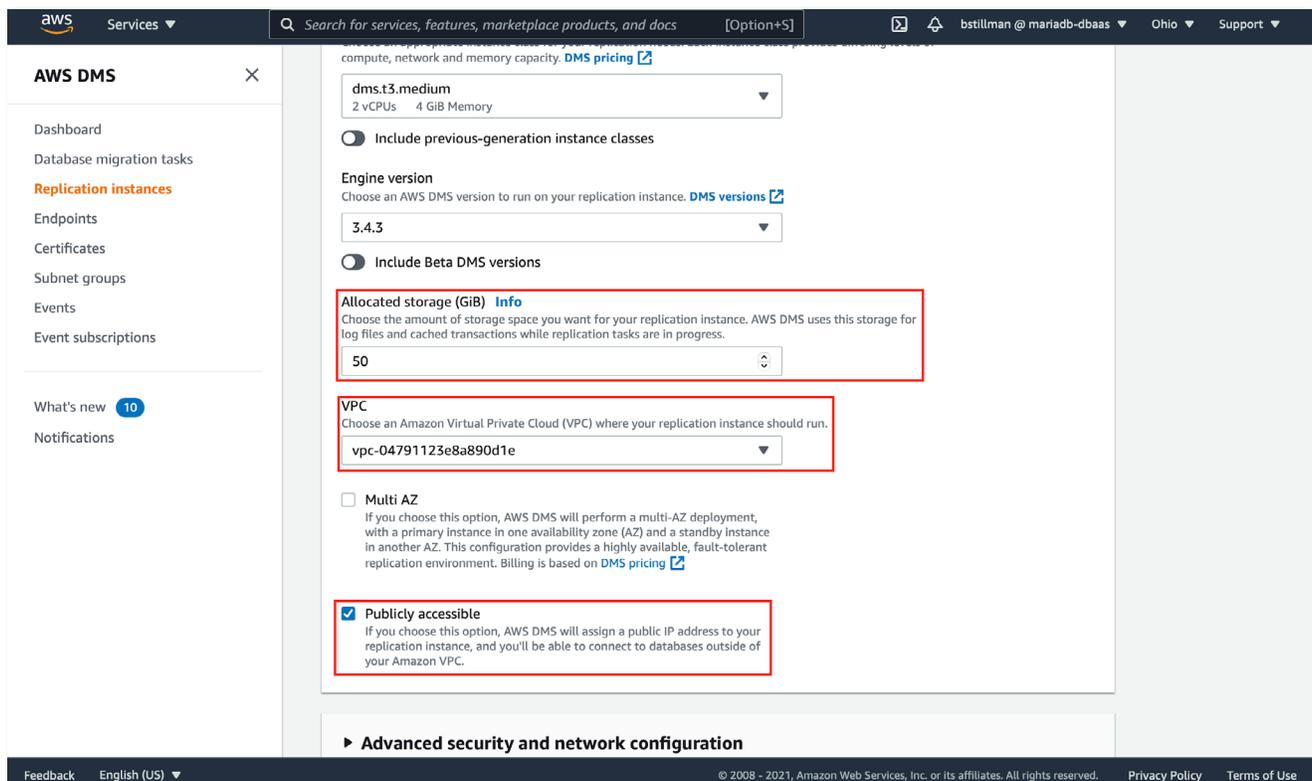
Click on the “Create replication instance” button.



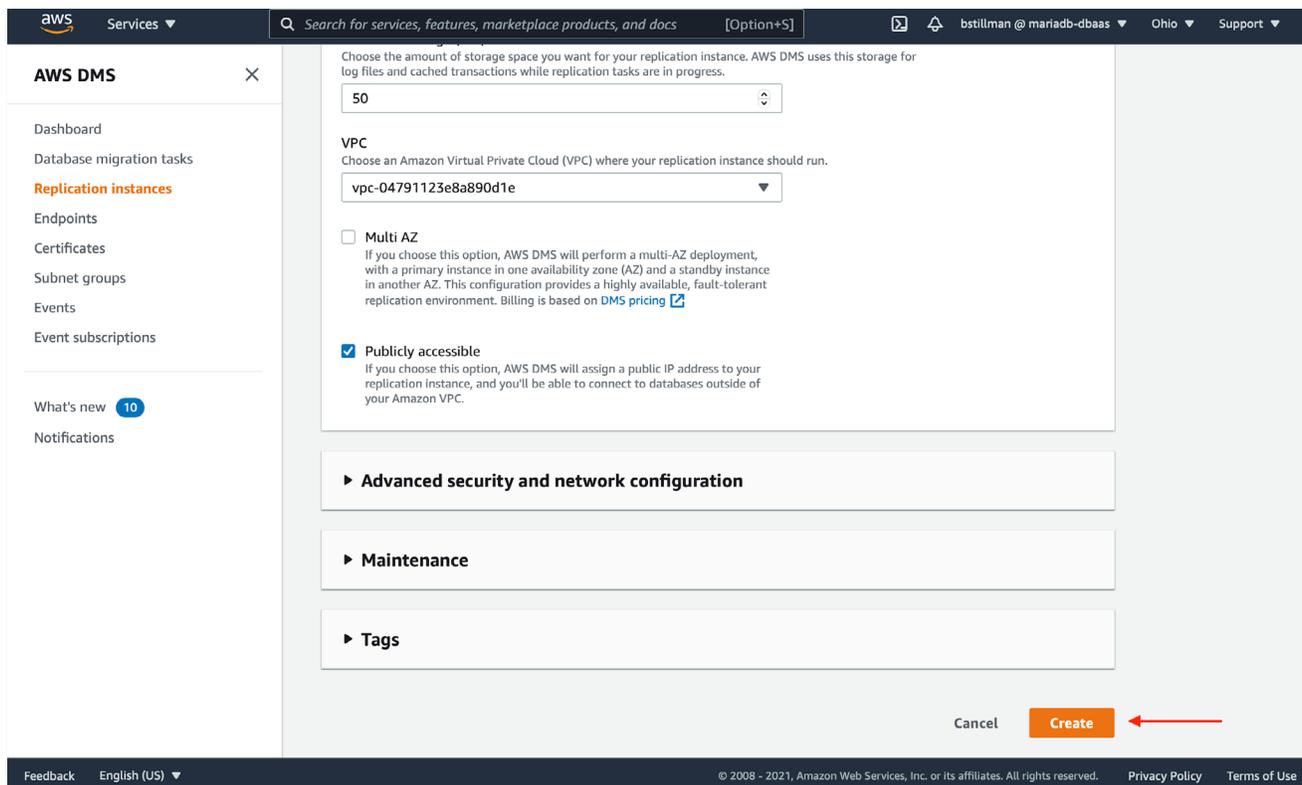
Now configure the “Replication instance”. Enter a “Name” and “Description” for this “replication instance”. Then select an “Instance class”.



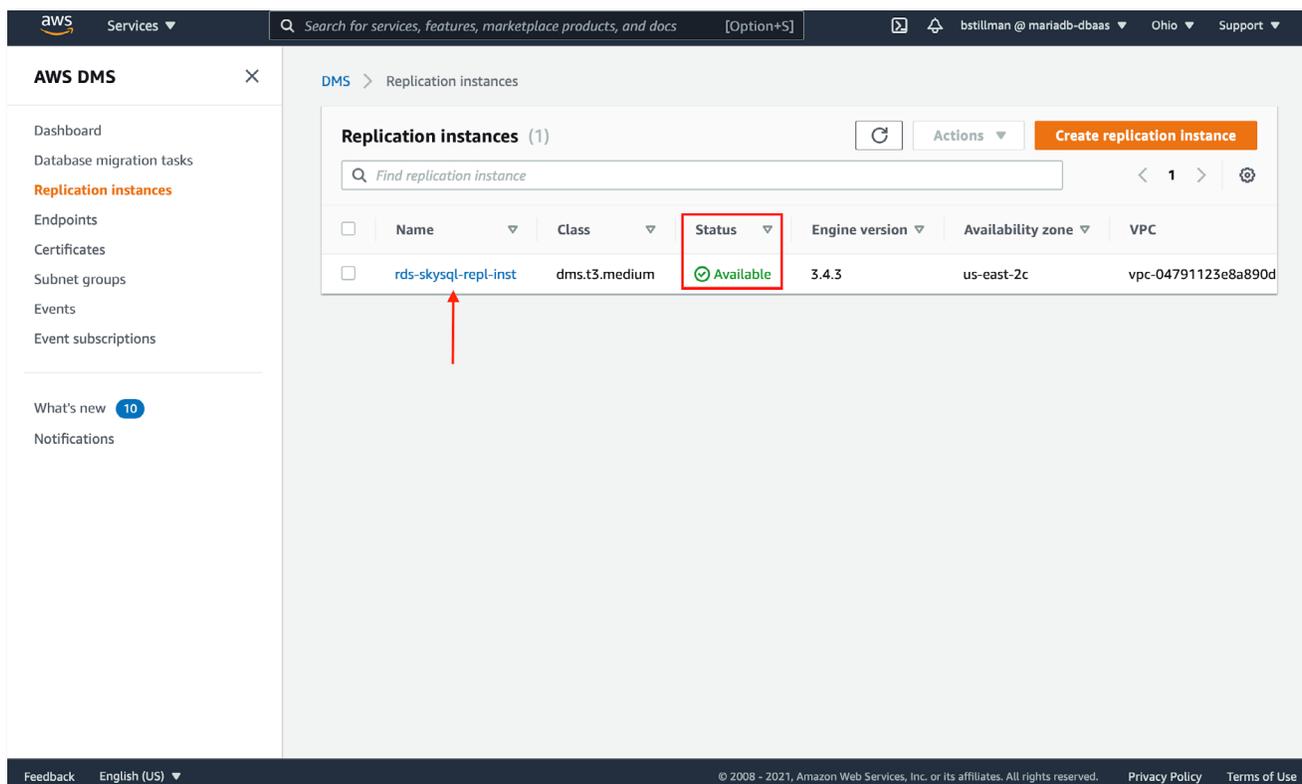
Scroll down to continue the configuration. “Allocated storage” defaults to 50 GB. For this demonstration, this is sufficient. Select the “VPC” which this “replication instance” should run. For this demonstration, the VPC where the RDS instance to be migrated runs was selected. Next, “Publicly accessible” should be checked by default. Verify that it is checked.



Scroll to the bottom of the page. For this demonstration, no other configuration is needed. Click the “Create” button.



Once the newly created “replication instances” has a “Status” of “Available”, click on the instance’s “Name” to see more details.



From here, note the “Public IP address” and “Private IP address”.

The screenshot shows the AWS DMS console interface. On the left is a navigation menu with 'Replication instances' selected. The main content area displays the details for the replication instance 'rds-skysql-repl-inst'. At the top, there is a 'Replication instance summary' table with the following data:

Class	Engine version	Status	Associated migration tasks
dms.t3.medium	3.4.3	Available	0

Below the summary are tabs for 'Overview details', 'CloudWatch metrics', 'Migration tasks', 'Log management', and 'Tags'. The 'Overview details' tab is active, showing a 'Details' section with 'Basic configuration'.

Basic configuration

ARN	arn:aws:dms:us-east-2:347119114792:rep:S7EALKD4BFCJVMLJ3FAKWOTAVA25SG2IMLG2JUI
Public IP address	3.15.80.73
Private IP address	172.30.2.157
Instance class	dms.t3.medium
Status	Available
Free instance expiration	N/A

The footer of the console shows 'Feedback', 'English (US)', '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

CREATE AWS DMS SOURCE ENDPOINT

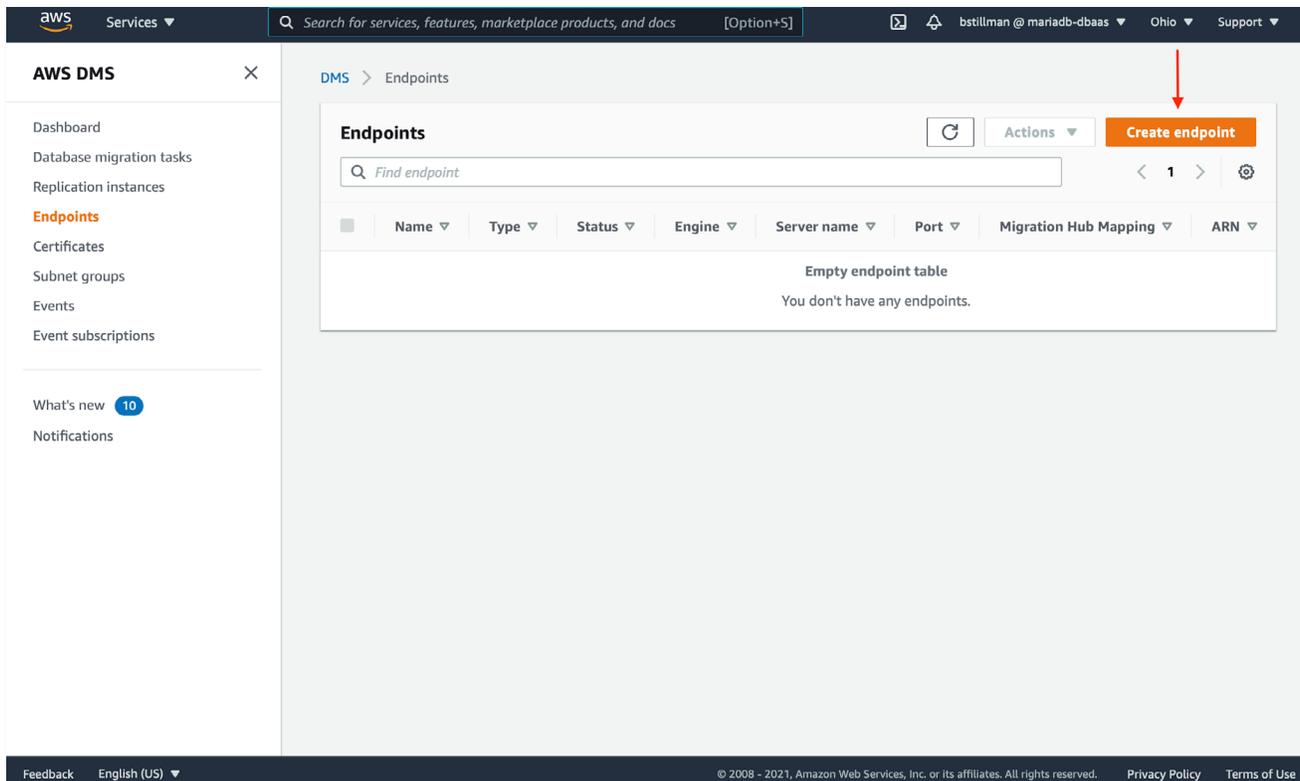
Next, the Source Endpoint needs to be created.

Click on “Endpoints” in the navigation menu on the left side.

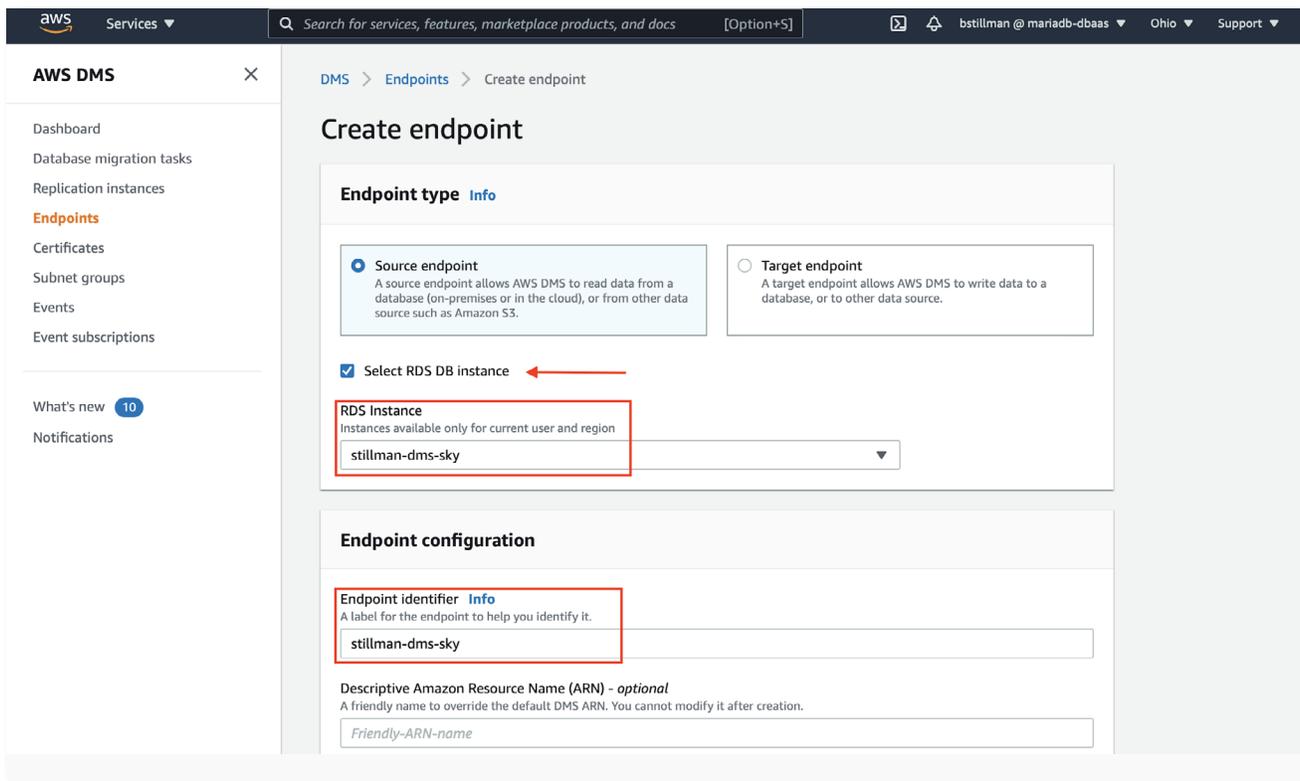
The screenshot displays the AWS Management Console interface for the AWS DMS service. The left-hand navigation menu is visible, with 'Endpoints' highlighted by a red arrow. The main content area shows the 'Replication instances' page, which includes a search bar, a 'Create replication instance' button, and a table listing the instances.

<input type="checkbox"/>	Name	Class	Status	Engine version	Availability zone	VPC
<input type="checkbox"/>	rds-skysql-repl-inst	dms.t3.medium	Creating	3.4.3	us-east-2c	vpc-04791123e8a890d1

Click on the “Create endpoint” button.



“Source endpoint” should already be selected by default. Click the checkbox next to “Select RDS DB instance”. Then under “RDS Instance”, select the RDS instance to be migrated. In this example, it’s the instance named “stillman-dms-sky” mentioned earlier. Notice the “Endpoint identifier” field under “Endpoint configuration” is auto-populated.



Scroll down and complete the "Endpoint configuration". Under "Source engine", select "MySQL" (or whatever engine the source endpoint is) if it isn't already. Under "Access to endpoint database", select the correct option. For this demonstration, "Provide access information manually" was selected. Since an RDS instance was chosen as the source, "Server name", "Port", and "User name" were auto-populated. Enter the password at "Password".

The screenshot shows the AWS DMS console interface for configuring an endpoint. The left sidebar contains navigation options like Dashboard, Database migration tasks, Replication instances, Endpoints, Certificates, Subnet groups, Events, and Event subscriptions. The main content area is titled "Endpoint configuration" and includes the following fields:

- Endpoint identifier:** stillman-dms-sky
- Descriptive Amazon Resource Name (ARN) - optional:** Friendly-ARN-name
- Source engine:** MySQL (highlighted with a red box)
- Access to endpoint database:** Provide access information manually (selected, highlighted with a red box)
- Server name:** stillman-dms-sky.cmdxr9mqv3ld.us-east-2.rds.amazonaws.com
- Port:** 3306
- Secure Socket Layer (SSL) mode:** none
- User name:** admin
- Password:** [Redacted, highlighted with a red box]

At the bottom of the configuration area, there is a section for "Endpoint-specific settings".

Configuration is complete and the endpoint can be tested. Continue scrolling down. At the bottom of the endpoints page, click on "Test endpoint connection (optional)" to open the endpoint testing configuration. Select the "VPC" and "Replication instance" to test. Next, click "Run test". This will create the endpoint and test. If the test is successful, continue. If the test is unsuccessful, it is most likely that the replication instance's "Private IP address" noted earlier needs to be granted access in the RDS instance's security group. More information can be found here: <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.RDSSecurityGroups.html>.

AWS DMS ×

- Dashboard
- Database migration tasks
- Replication instances
- Endpoints**
- Certificates
- Subnet groups
- Events
- Event subscriptions

What's new 10

Notifications

▸ Tags

▼ **Test endpoint connection (optional)** ←

VPC
vpc-04791123e8a890d1e

Replication instance
A replication instance performs the database migration
rds-skysql-repl-inst

⚠ Your endpoint will always be created even if the connection fails ×
After clicking 'Run test', DMS creates the endpoint with the details you provided and attempts to connect to it. If the connection fails, you can edit the endpoint definition and test the connection again. You can also delete the endpoint manually.

Run test ←

Endpoint identifier	Replication instance	Status	Message
No records found			

Cancel **Create endpoint**

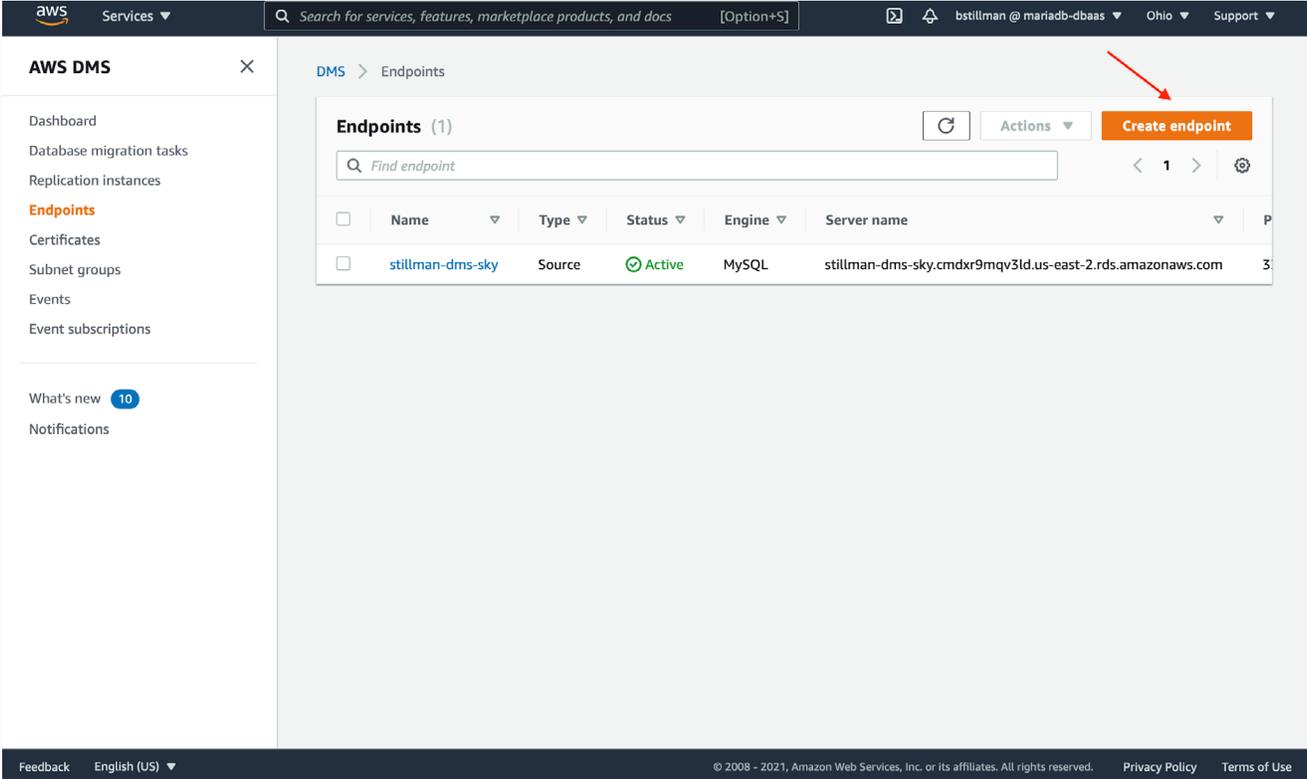
Feedback English (US) ▼

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

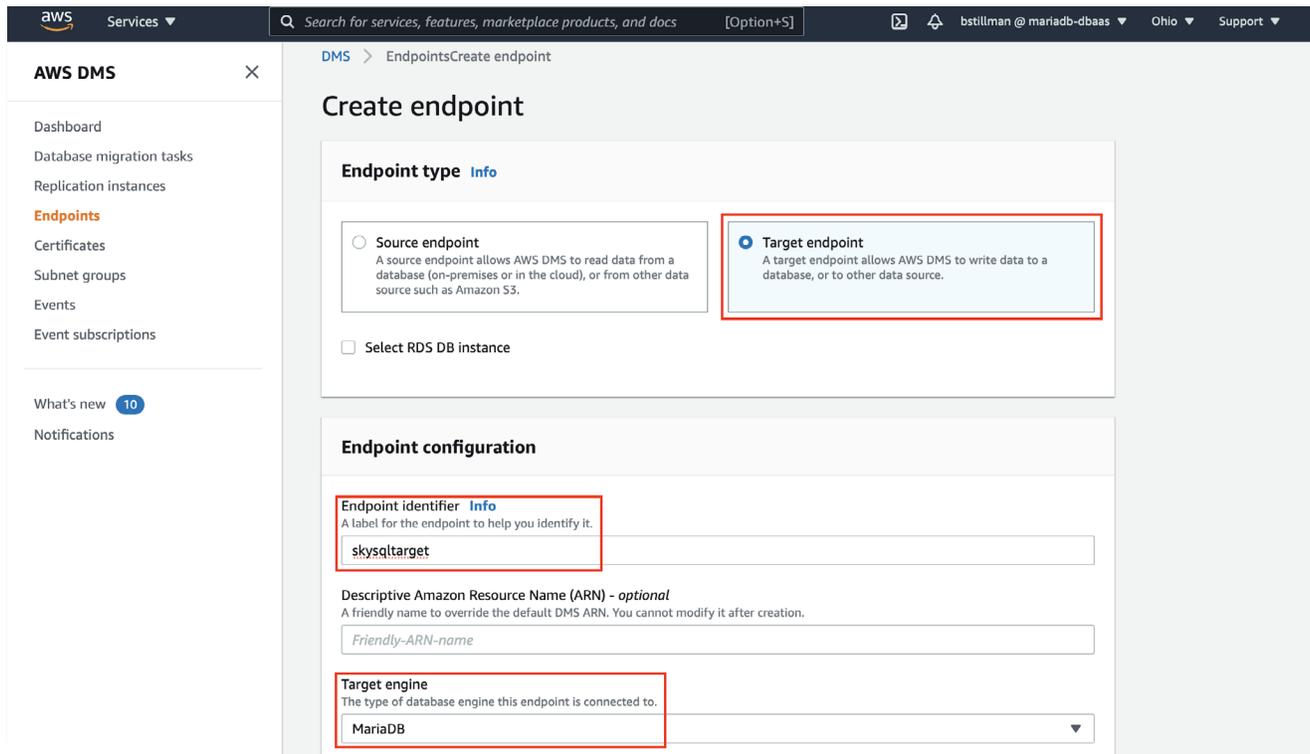
CREATE SKYSQL TARGET ENDPOINT

Next, the Target Endpoint needs to be created.

Click the “Create endpoint” button.



Select “Target endpoint”. Enter a descriptive “Endpoint identifier”. Choose “MariaDB” as the “Target engine”.



Scroll down to complete the configuration.

- Under “Access to endpoint database”, select “Provide access information manually”.
- For “Server name”, enter the “Fully Qualified Domain Name” from the SkySQL service.
- “Port” is the “Read-Write Port” from the SkySQL service.
- Select “verify-ca” under “Secure Socket Layer (SSL) mode”.
- Click “Add new CA certificate”.
- Upload the “Certificate authority chain” from SkySQL.
- Now “skysql-chain” should be available to select in the “CA certificate” drop-down.
- Lastly, enter the “User name” and “Password” for the SkySQL service.

Access to endpoint database

Choose AWS Secrets Manager

Provide access information manually

Server name

rds-migration-example.mdb0001941.db.skysql.net

Port

The port the database runs on for this endpoint.

5001

Secure Socket Layer (SSL) mode

The type of Secure Socket Layer enforcement

verify-ca

CA certificate

skysql-chain

[Add new CA certificate](#)

User name [Info](#)

DB00003785

Password [Info](#)

.....

► Endpoint-specific settings

► KMS master key

Configuration is complete and the endpoint can be tested. Continue scrolling down. At the bottom of the “Endpoints” page, click on “Test endpoint connection (optional)” to expand the testing configuration. Select the correct “VPC” and “Replication instance”. Then click the “Run test” button. This will create the endpoint and test. If the test is successful, continue. If the test is unsuccessful, it is most likely that the replication instance’s “Public IP address” noted earlier needs to be whitelisted in the SkySQL service. More information can be found here: <https://mariadb.com/products/skysql/docs/instructions/ip-whitelist-services/>.

► Tags

▼ **Test endpoint connection (optional)**

VPC

vpc-04791123e8a890d1e

Replication instance

A replication instance performs the database migration

rds-skysql-repl-inst

Warning

Your endpoint will always be created even if the connection fails

After clicking 'Run test', DMS creates the endpoint with the details you provided and attempts to connect to it. If the connection fails, you can edit the endpoint definition and test the connection again. You can also delete the endpoint manually.

Run test

Endpoint identifier	Replication instance	Status	Message
No records found			

Cancel **Create endpoint**

CREATE AWS DMS DATABASE REPLICATION TASK

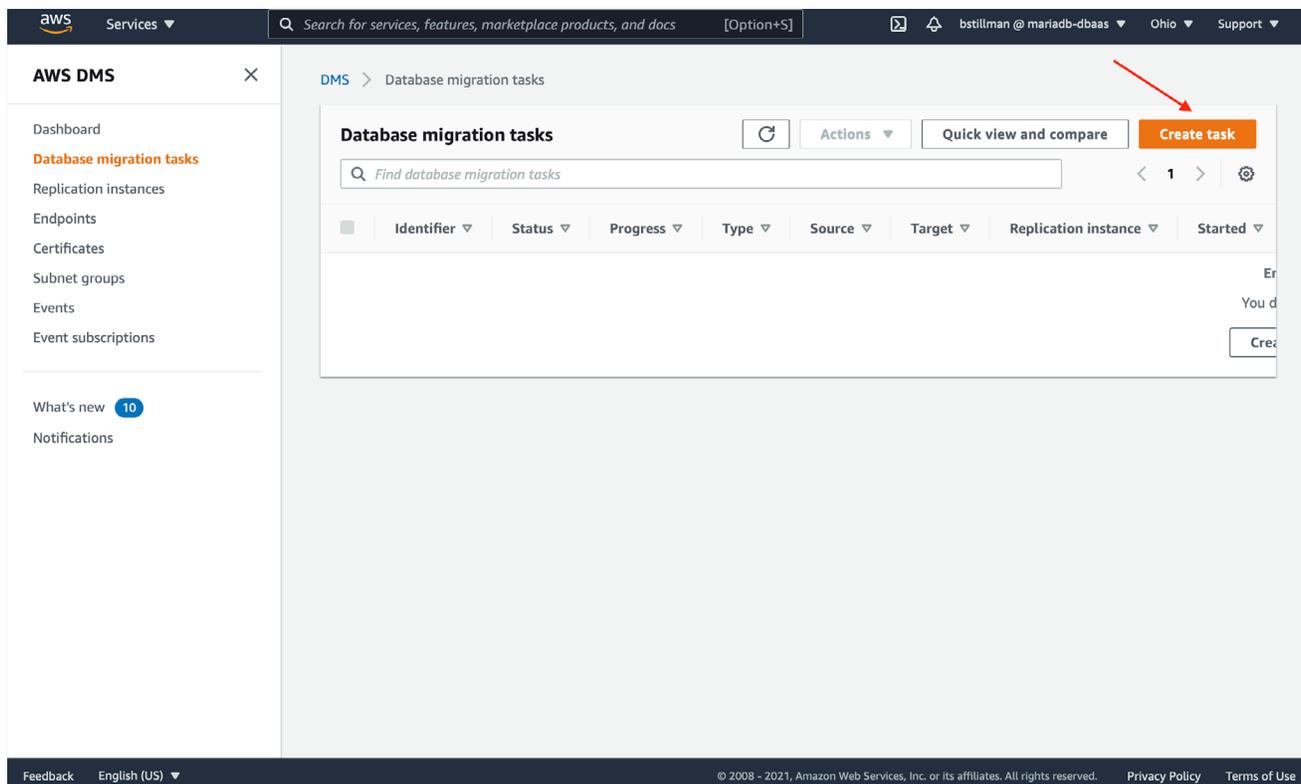
Next, create the “Database replication task”.

Click on “Database replication tasks” in the navigation menu on the left side.

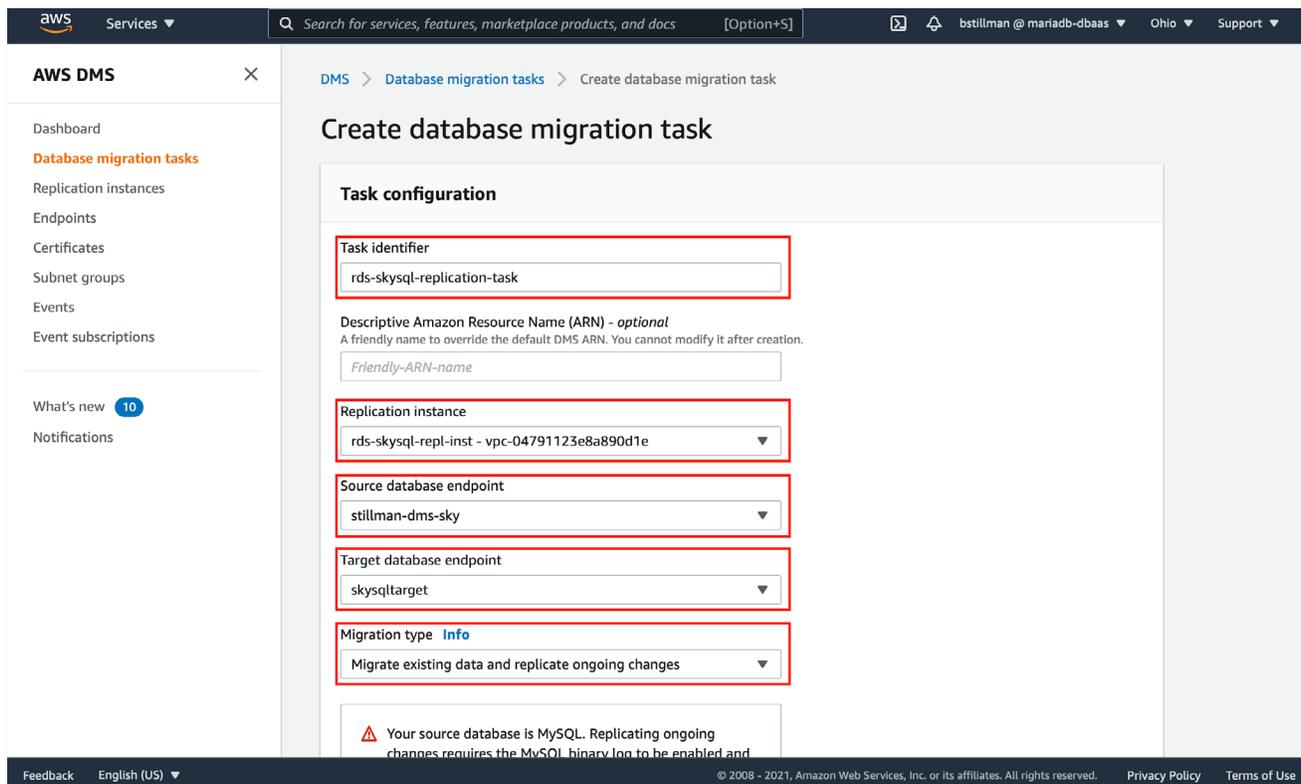
The screenshot shows the AWS DMS console interface. On the left, the navigation menu includes 'Database migration tasks' which is highlighted with a red arrow. The main content area displays the 'Endpoints (2)' page. At the top right of this page, there is a 'Create endpoint' button. Below this is a search bar labeled 'Find endpoint'. A table lists the endpoints with the following data:

<input type="checkbox"/>	Name	Type	Status	Engine	Server name	P
<input type="checkbox"/>	skysqltarget	Target	Active	MariaDB	rds-migration-example.mdb0001941.db.skysql.net	5
<input type="checkbox"/>	stillman-dms-sky	Source	Active	MySQL	stillman-dms-sky.cmdxr9mqv3ld.us-east-2.rds.amazonaws.com	3

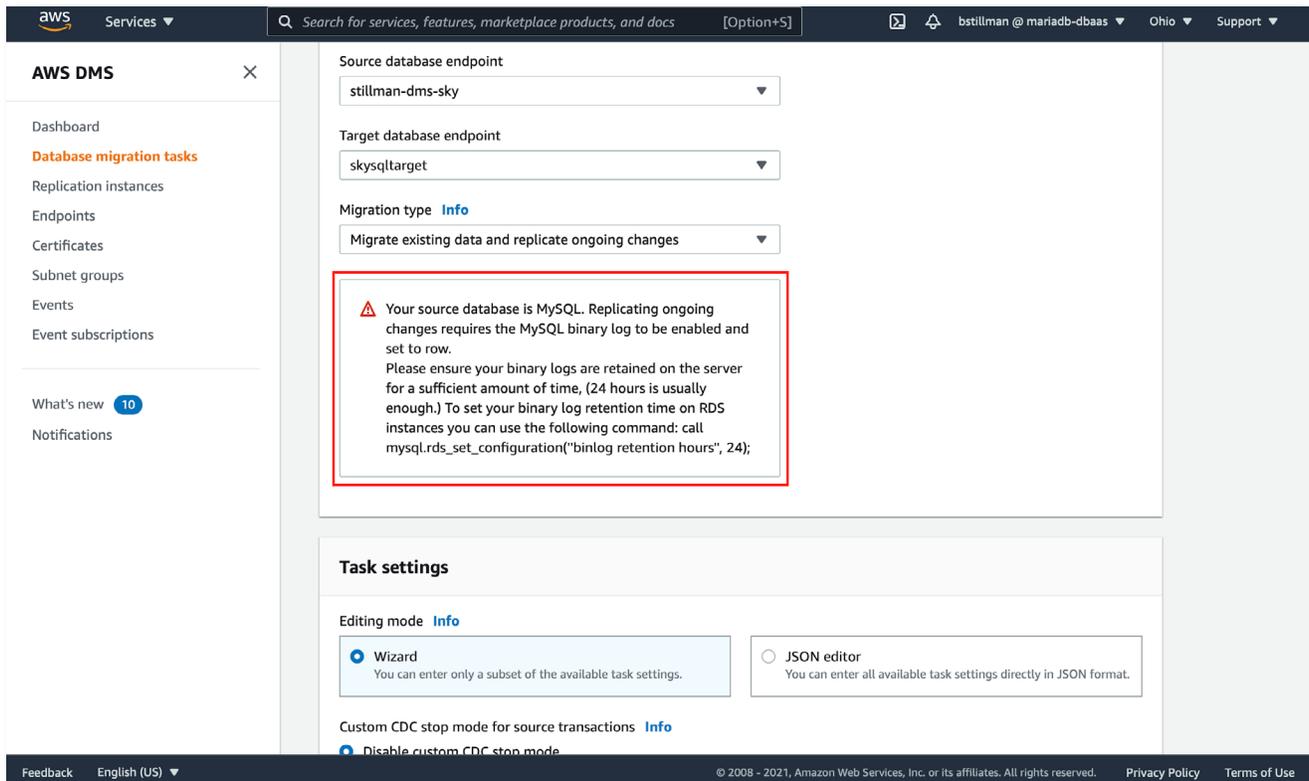
Click the “Create task” button.



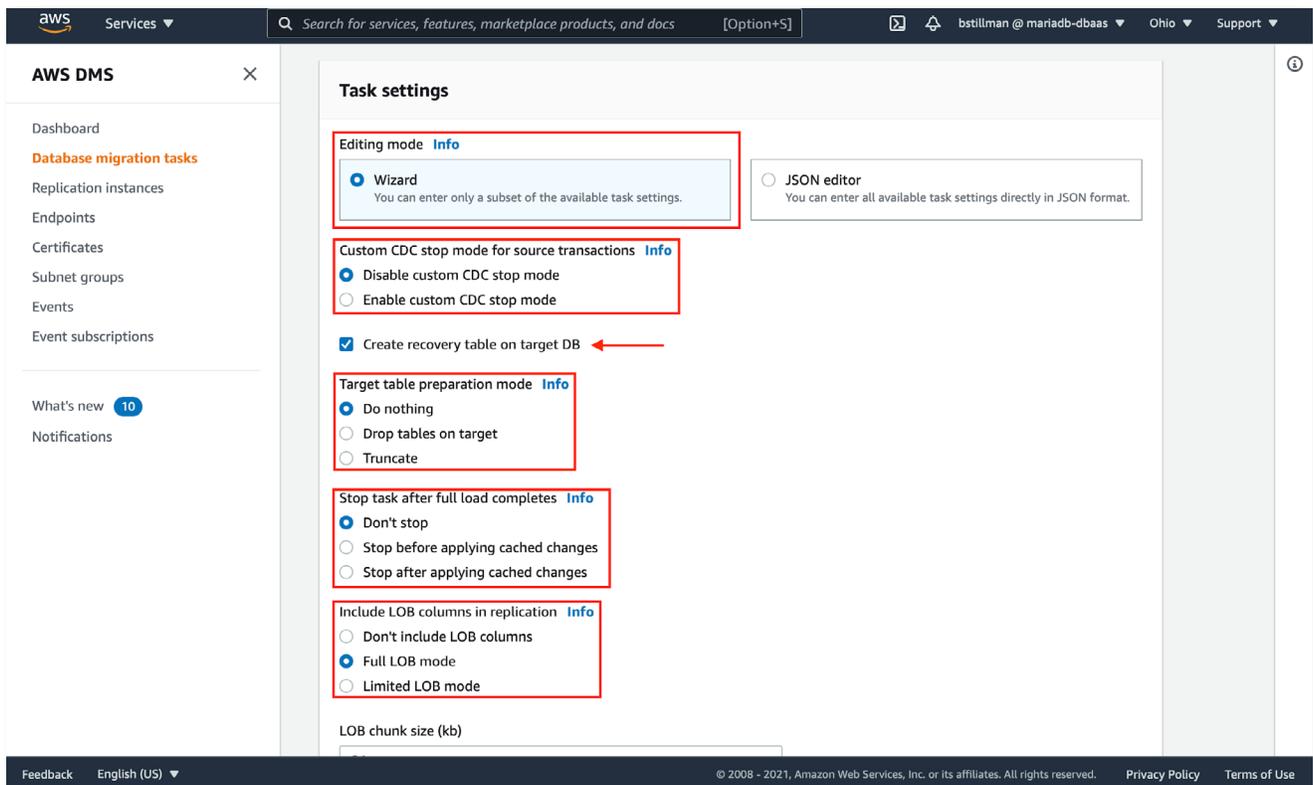
Enter a descriptive “Task identifier”. Select the correct “Replication instance”, “Source database endpoint”, “Target database endpoint”, and “Migration type”. For this demonstration, this migration will be “Migrate existing data and replicate ongoing changes”. This does a bulk load of existing data and replicates any further changes to the SkySQL service.



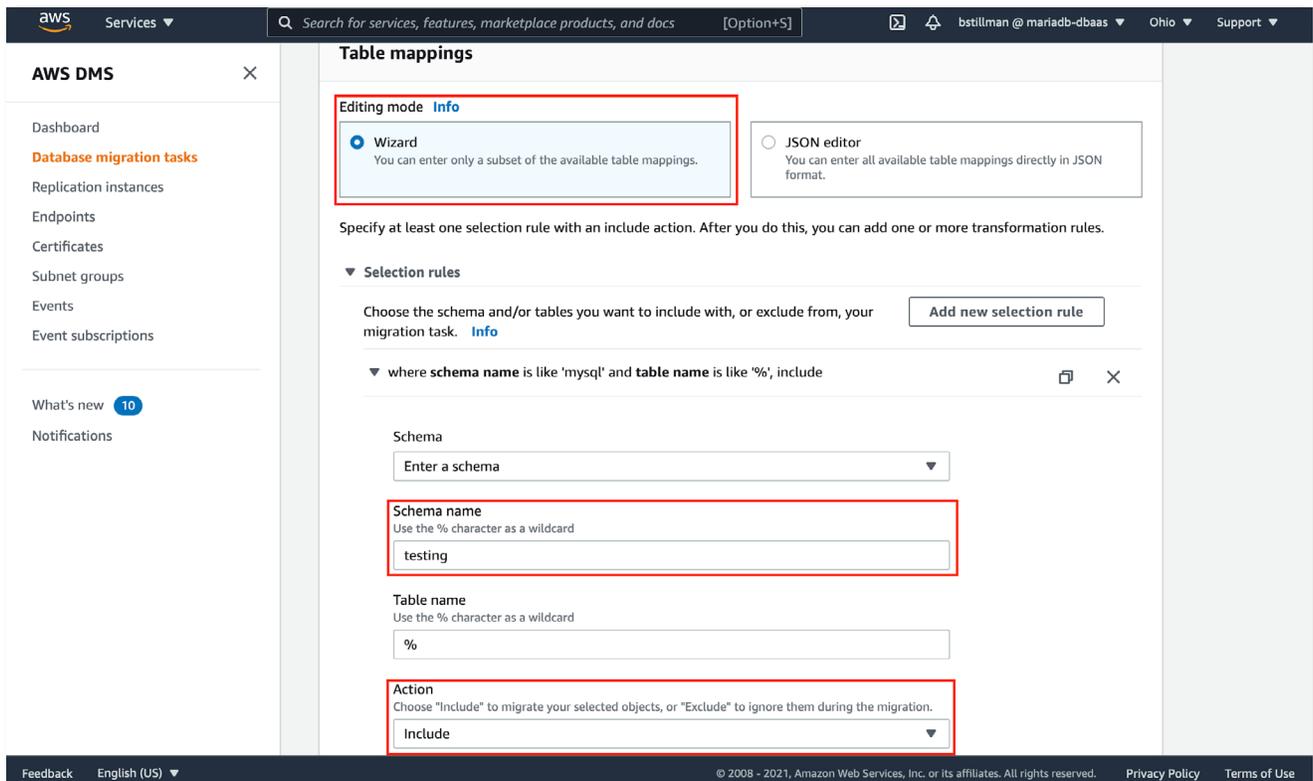
Note the warning presented. This should have already been completed during the “Configure the RDS Instance” step, thus safe to ignore.



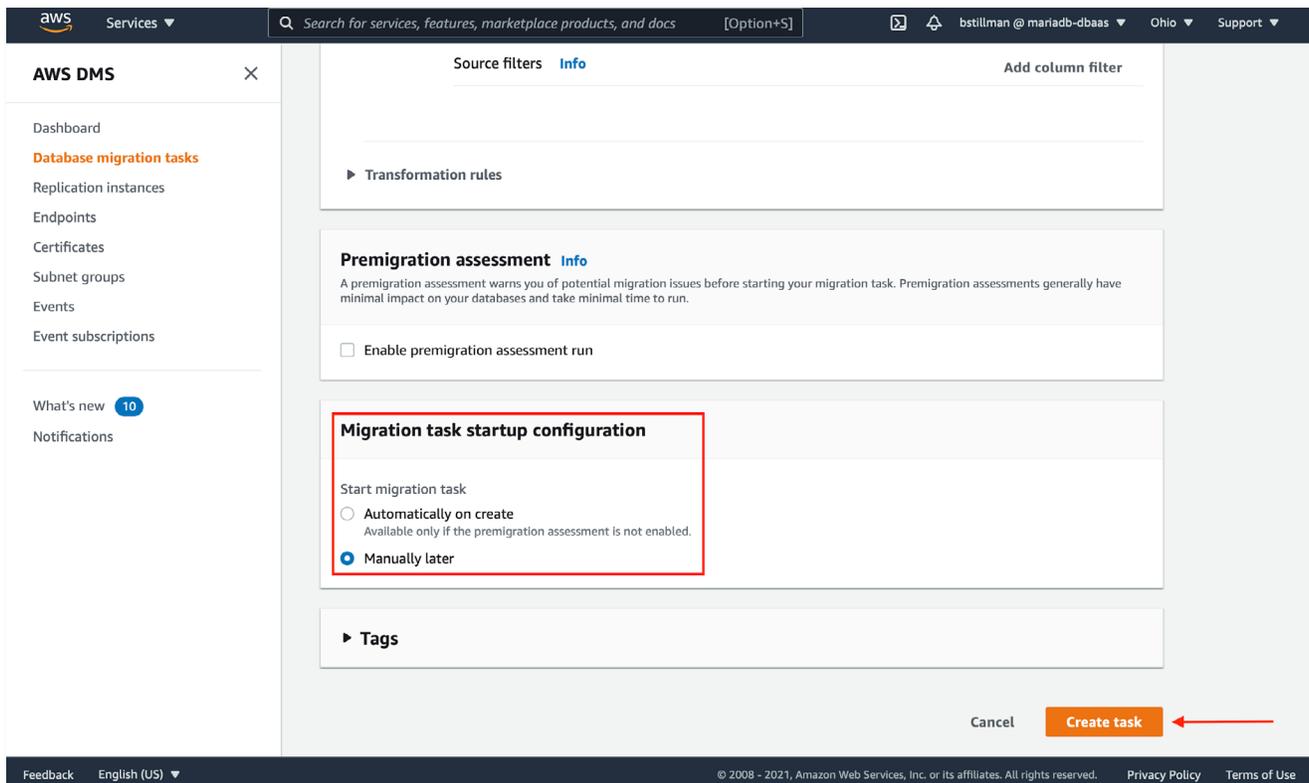
Continue scrolling down to continue configuring. Select “Wizard” for “Editing mode”. Select “Disable custom CDC stop mode”. Click the checkbox for “Create recovery table on target DB”. For “Target table preparation mode”, select the best option. For this demonstration, “Do nothing” was selected as the database to be migrated does not contain any data in the SkySQL service. Under “Stop task after full load completes”, select “Don’t stop”. Under “Include LOB columns in replication”, select the best option. For this demonstration, “Full LOB mode” was chosen although no LOBs are in the database to be migrated.



Scroll down to the “Table mappings” section. Select “Wizard” for “Editing mode”. Now create a “Selection rule”. For this demonstration, a database or schema named “testing” and all tables within will be included for migration.



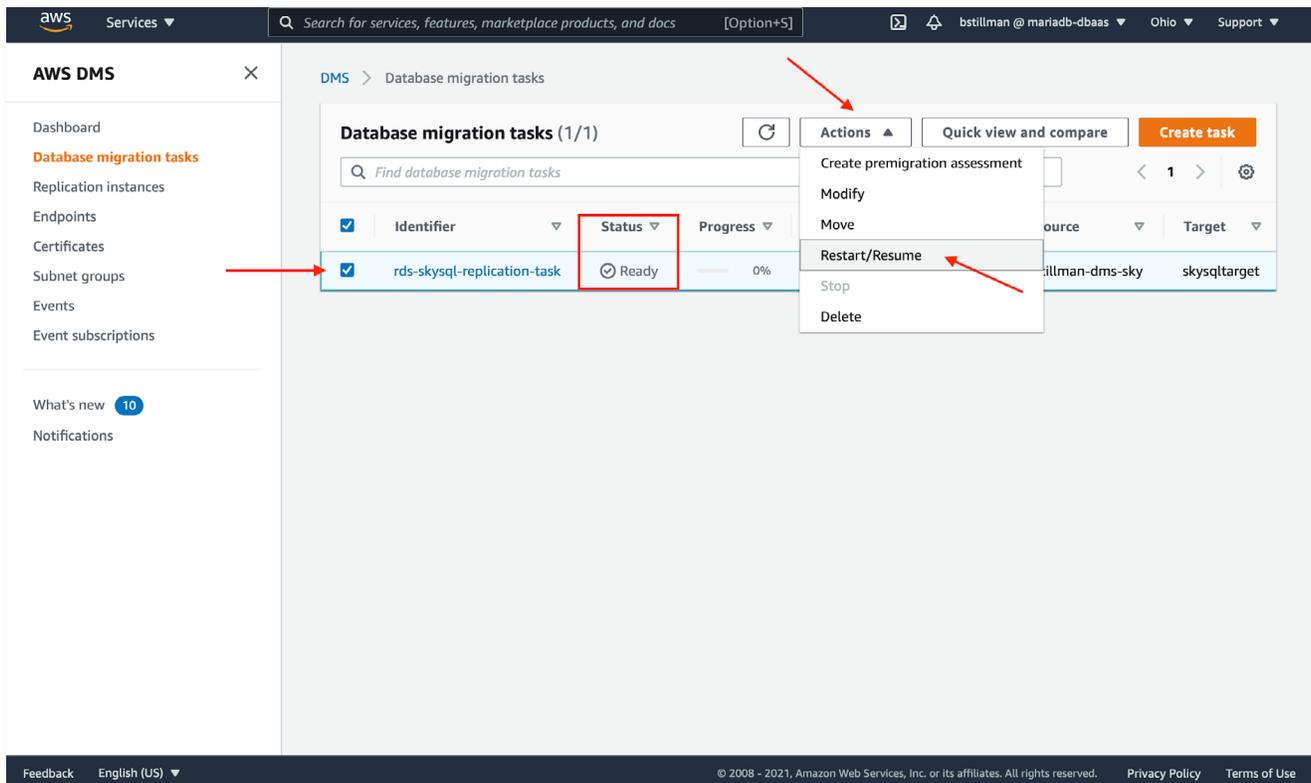
Scroll down to the bottom of the page. Under "Migration task startup configuration", select "Manually later". Then click the "Create task" button.



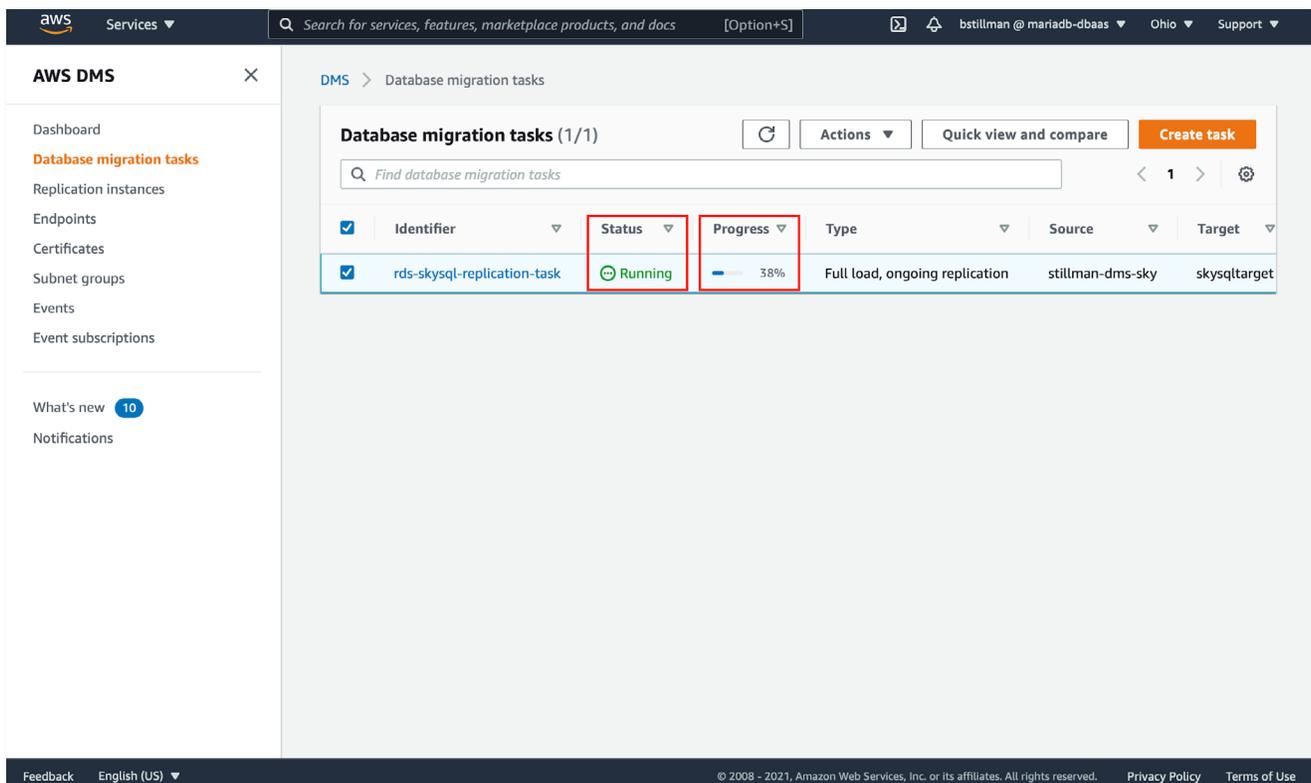
While the "replication task" is being created, verify the data within the RDS database to be migrated. For this demonstration, there is one table named "TableA" in the database named "testing".

```
SELECT COUNT(*) FROM testing.TableA;
+-----+
| COUNT(*) |
+-----+
| 14017096 |
+-----+
```

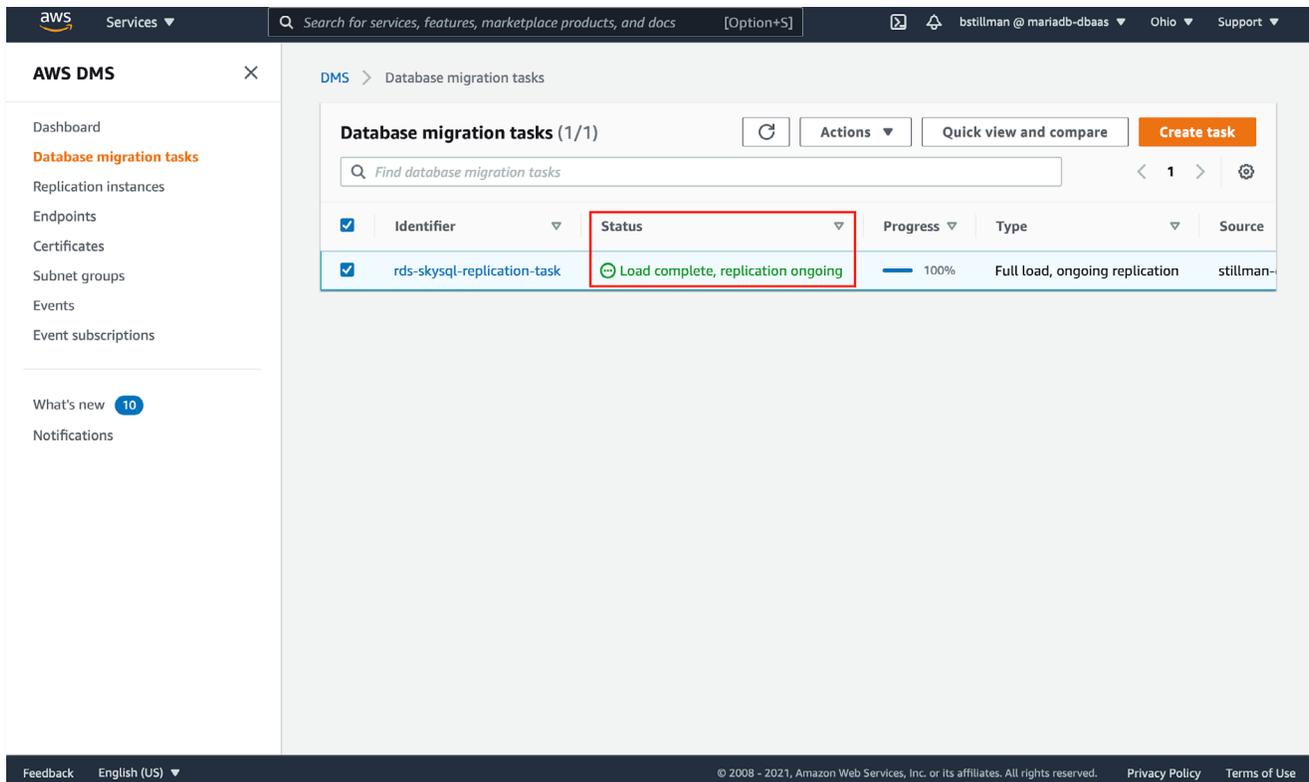
Once the replication task's "Status" reports "Ready", check the checkbox next to the service. Then click "Actions", and then "Restart/Resume" to start the task.



The replication task's "Status" should now report "Running", and the "Progress" should be reporting. This can take a considerable amount of time depending on the size of the data to be migrated.



Once the initial bulk data load is complete, the “Status” should change to “Load complete, replication ongoing”.



Verify on the SkySQL service that the database and table has been migrated successfully.

```
SELECT COUNT(*) FROM testing.TableA;
+-----+
| COUNT(*) |
+-----+
| 14017096 |
+-----+
```

This matches the results from the RDS instance.

Now verify additional changes on the RDS instance are migrated to the SkySQL service by inserting a record into the table.

```
INSERT INTO testing.TableA (uuid_junk) VALUES ('ccae845-6c9e-11eb-bf8d-06676091bc04');
```

Verify that record now exists in the SkySQL service.

```
SELECT uuid_junk FROM testing.TableA
WHERE uuid_junk = 'ccae845-6c9e-11eb-bf8d-06676091bc04';
```

```
+-----+
| uuid_junk
| +-----+
| ccaae845-6c9e-11eb-bf8d-06676091bc04 |
+-----+
```

AWS DMS has been successfully set up and is migrating any changes to the RDS instance to the SkySQL service. At this point, the application using the RDS instance can now be tested against real, live data on the SkySQL instance and cutover can be scheduled.

TESTING AND VALIDATION

Through the successful completion of this migration process, a second copy of the data in AWS RDS has been established on MariaDB SkySQL. This process is repeatable, which enables teams to perform a range of data validation and application testing before performing application migration.