**Data Migration Project**

Ample Technologies is planning to migrate data from on premise MySQL to AWS RDS.

The current in-house production database is based on a 10TB, standby in DR site. The database has 1 main schemas, comprising of 1 table, 5000 concurrent session of which 100 are active at a given time and has an IOPS requirement of 50K read and 30K write IOPS. The development database is 1/10th of the production capacity.

Objective of DB Migration Plan:

Recommend effectively process to manage the migration of Database from on-premise MySQL to AWS RDS. Core capabilities migration plan should involve.

* Scoping of the current database
* Approach to migration
* Production cutover plan
* Rollback Plan
* Execution Steps

Deliverables:

1. Discuss a Database Migration Plan
2. Implement your solution

**Scoping of the current database**

|  |  |
| --- | --- |
| **Production Database** | **Development Database** |
| 10TB | 1TB |
| 1 TABLE | 1 TABLE |
| 5000 CONCURRENT SESSION | 500 CONCURRENT SESSION |
| 100 ACTIVE | 10 ACTIVE |
| 50K READ | 5K READ |
| 30K WRITE | 3K WRITE |

**Approach to Migration**



**Data Migration**



**Homogenous Heterogenous**

**Same engine different engine**



**(on premise MySQL) RDS(MySQL) on premise MySQL RDS Postgre**



**Production Cutover plan**

**A diagram of a data input

AI-generated content may be incorrect.**

Data from on premise to AWS

Users input data from the app into the on-premises database through point 1, the developers start to migrate the data from on-premises database to AWS, this can happen in the test or development environment. The developers will cut off the data input to the on-premises at point 1 and connect the app to the test or development at point 2. Users begin to send data input to the AWS RDS. If there arise any issue with the test or development environment, that is the RDS probably due to the CPU issues of accessibility by users, they will reconnect the app back to the on-premises at point 3. After resolving the issues, they will reconnect to the AWS RDS database once everything about the AWS RDS is working fine. This can happen several times, just to get the AWS RDS database running perfectly while also not allowing the database to go down. This is called the **production cutover plan.**

**Execution Steps**

A diagram of a software system

AI-generated content may be incorrect.

1. Install SQL Client tool which is sqlelection tool on your local machine.

Click on connect to view the on-premises database and tables

A screenshot of a computer

AI-generated content may be incorrect.

The test database is available on premises.

Click on test to view tables

A screenshot of a computer

AI-generated content may be incorrect.

We have the test database with Client, Jasper, Persons, Person123, and Student tables.

A screenshot of a computer

AI-generated content may be incorrect.

Clicking on the Client table, you will see the sql (Structured Query Language)select statement and the selected table

A screenshot of a computer

AI-generated content may be incorrect.

Since our migration type is homogenous (same database engine migration), we ignore step 2 and 3 i.e, install JDBC drivers and sql conversion tool (SCT). Heterogenous migration is for different database engines, you will need SCT.

Next is to go to AWS Database Migration Service.

Turn off the New navigation button and under Migrat data, click on Replication instances

Then click on Create replication instance

A screenshot of a computer

AI-generated content may be incorrect.

Give the replication instance a name and screw down and on the high availability section click on dev or test workload (Single AZ). And on storage section enter 20 and create it. This will take up to 10 or 15 minutes to create.

A screenshot of a computer

AI-generated content may be incorrect.

The on-premise database credential are follow:

Server address: 54.163.41.196

Database type: MySQL

Port number: 3306

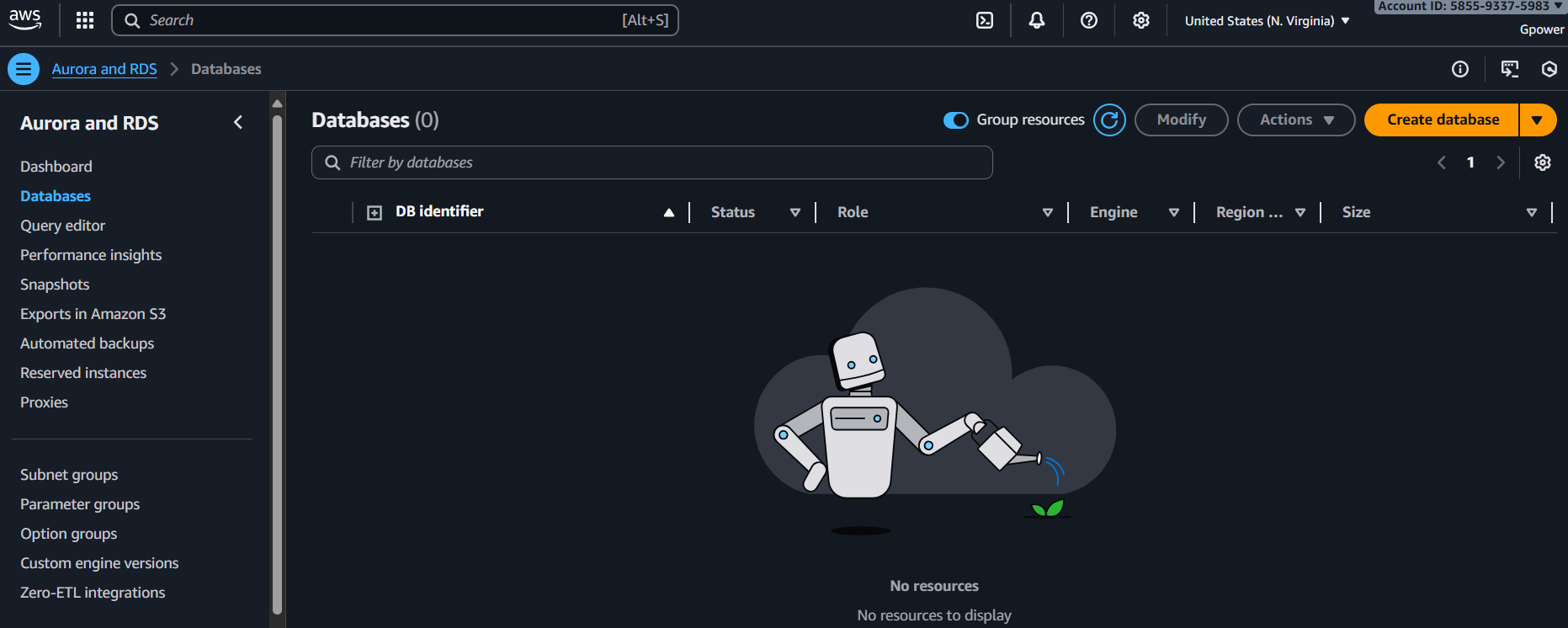
User: root

Password: Apple@12345

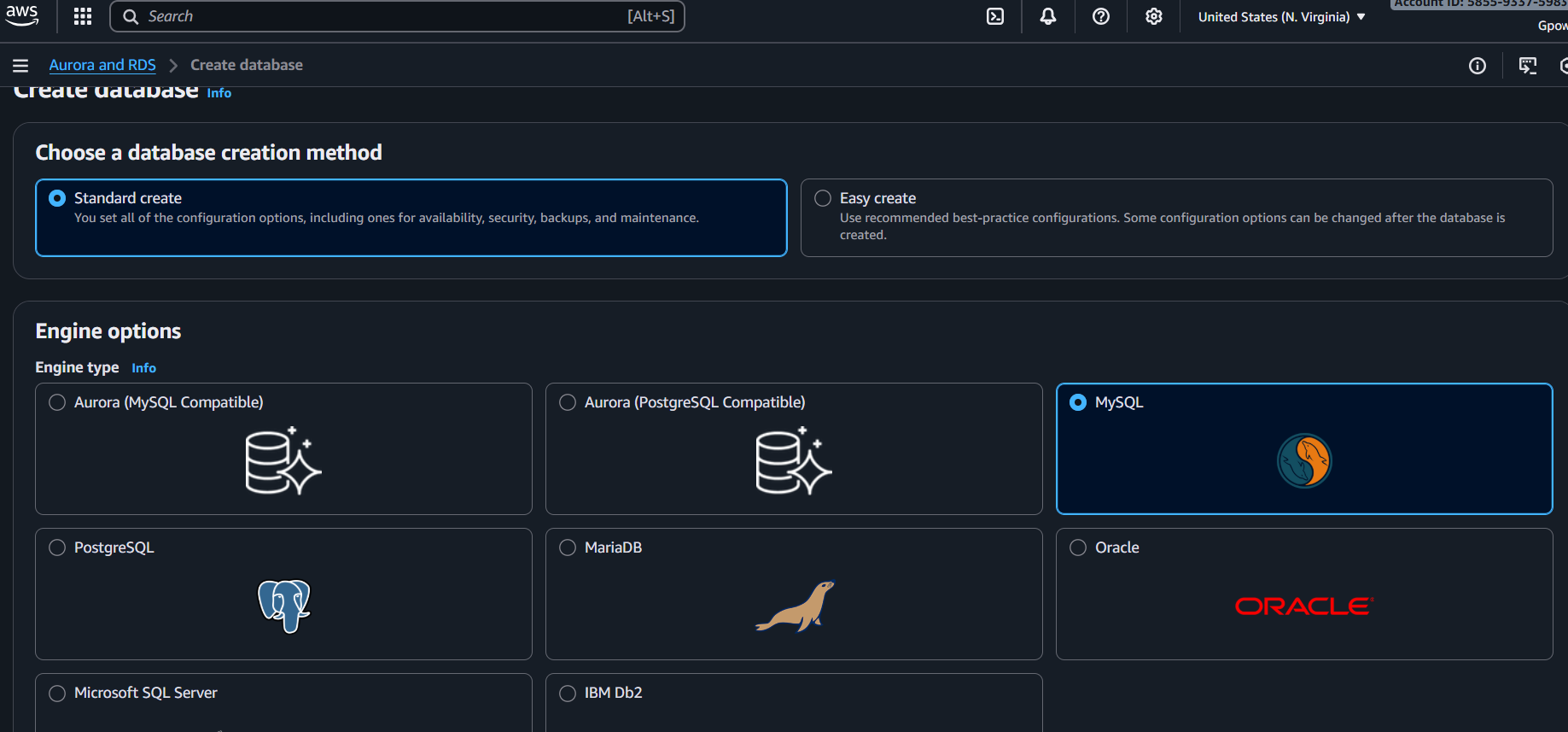
**Step 3:**

Next is to go to AWS RDS service to create the target database.

Click on Aurora and rds. Click on databases and Create database

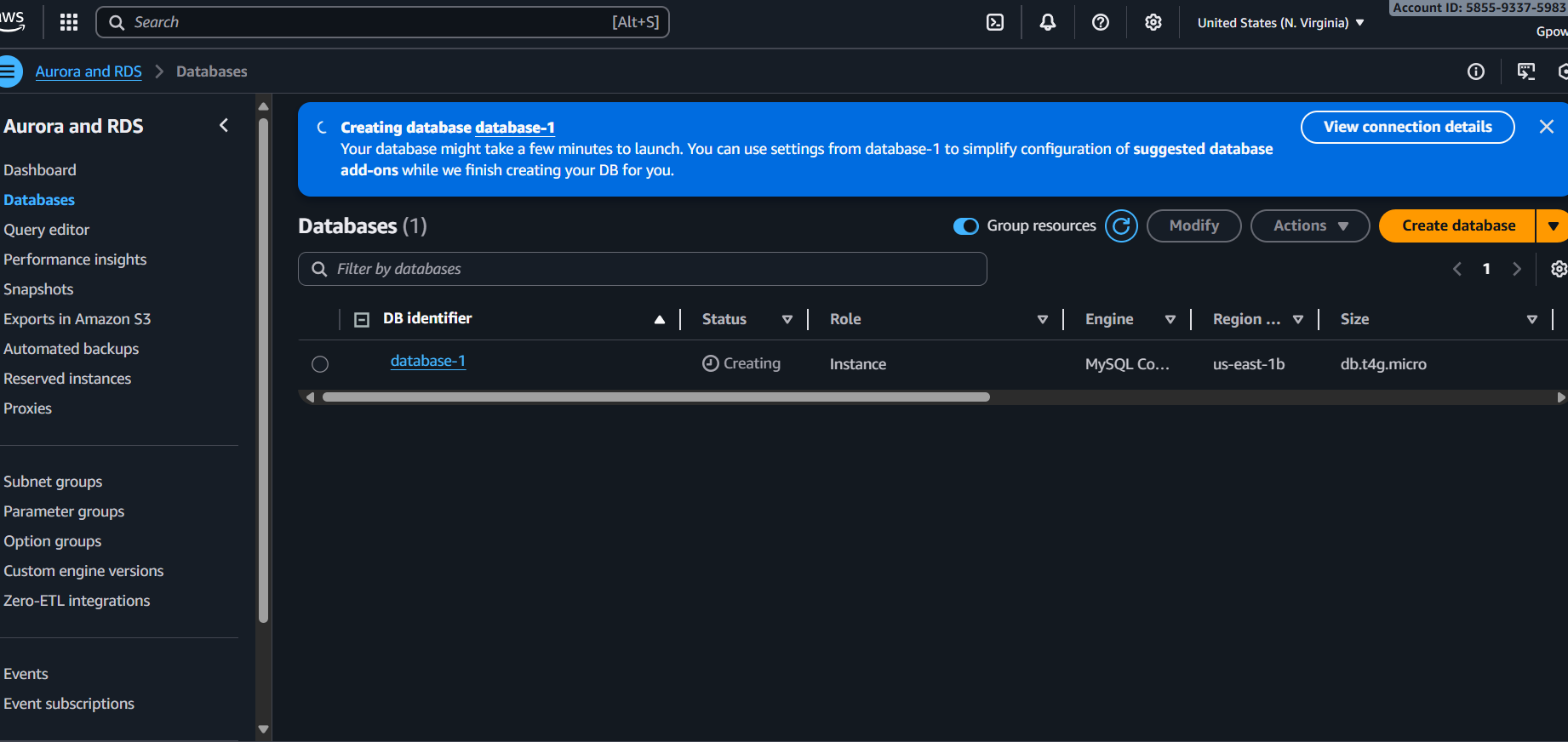


Click on MySQL engine



Click MySQL and then screw down and select **MySQL latest engine version**. On Template, select free tier. Master username is admin and enter a password you can remember. Enter 20 for the allocated storage. Public access should be yes and every other as default and create database. This will take some time to create.

Click to open the database



Whenever you create a database, make sure you go to the default security group and open it.

A screenshot of a computer

AI-generated content may be incorrect.

Security Group and go to inbound rule tab, then click on edit inbound rule

A screenshot of a computer

AI-generated content may be incorrect.

Click on add rule and select MySQL/Aurora and source anywhere, save it.

Next is to copy the endpoint of the database we just created.

A screenshot of a computer

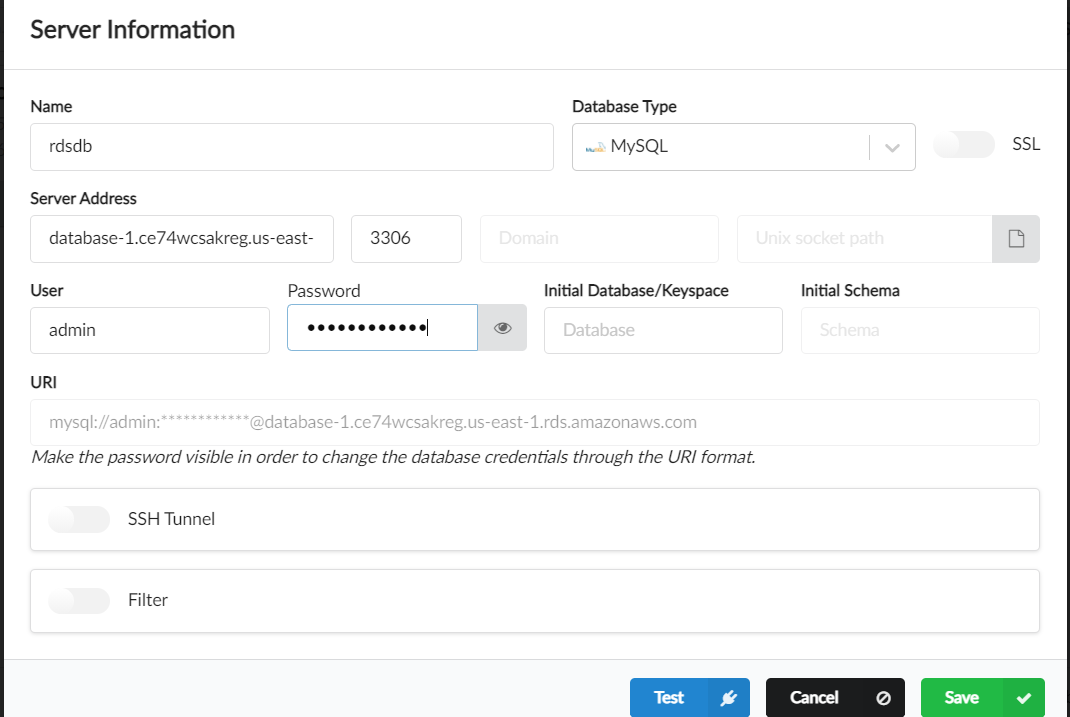
AI-generated content may be incorrect.

Back to sqlelectron and click the add

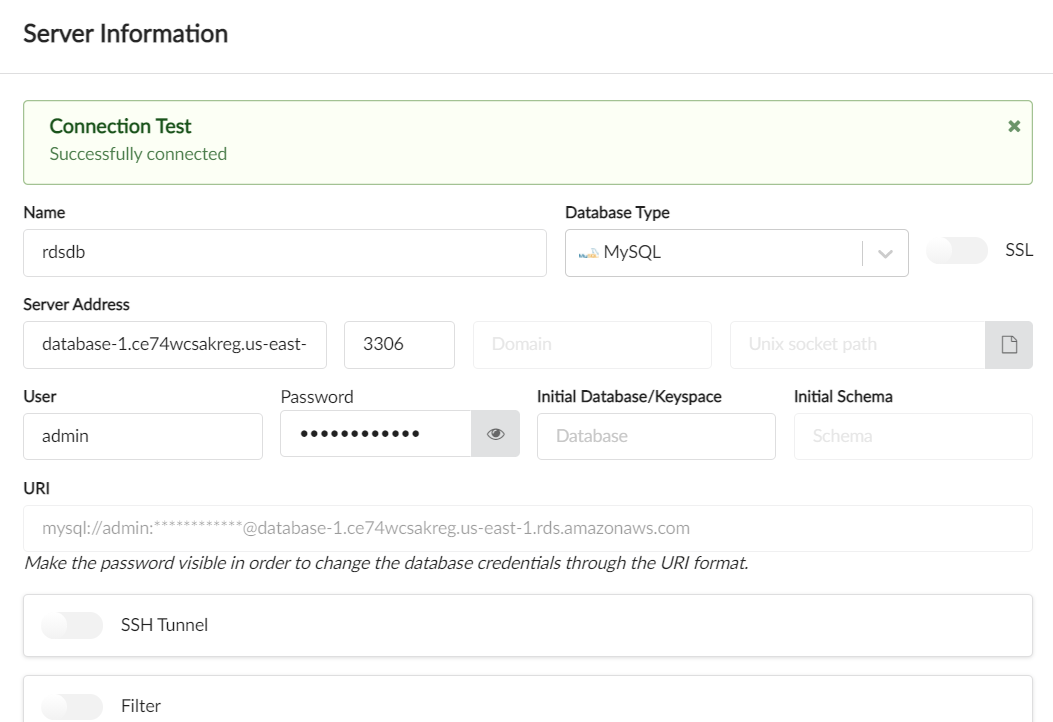
A screenshot of a computer

AI-generated content may be incorrect.

Give the database a name and select MyQSL. Paste the endpoint you copied the server address in the url of the sqlelectron. Enter the User, which is admin and password field will be what you entered when creating the RDS database.



Click on test and it has to show successfully connected. Screw down and click on save to save it. Then click on test and it will show Connection Test Successfully connected.



Click on connect on the RDS database and see the there is no database in it yet.

A screenshot of a computer

AI-generated content may be incorrect.

Remember we are moving the on-premise database to AWS RDS.

A screenshot of a computer

AI-generated content may be incorrect.

Next is to migrate the database from on-premise to AWS RDS.

Now back to DMS service, under migrate data, click on endpoints to create source and target endpoints

A screenshot of a computer

AI-generated content may be incorrect.

.

Create on premises endpoint

A screenshot of a computer

AI-generated content may be incorrect.

Give it a name and the source engine is MySQL. For the access to endpoint database, click on provide access information manually. Enter the server address for the sever name, and other credential for on-premise db. Screw down to test enpoint connection and click on run test.

You can see that its successful. Click on Create endpoint.

A screenshot of a computer

AI-generated content may be incorrect.

Click on create endpoint again.

A screenshot of a computer

AI-generated content may be incorrect.

This time for the target database. Tick and Select RDS DB instance. Then click provide access information manually.

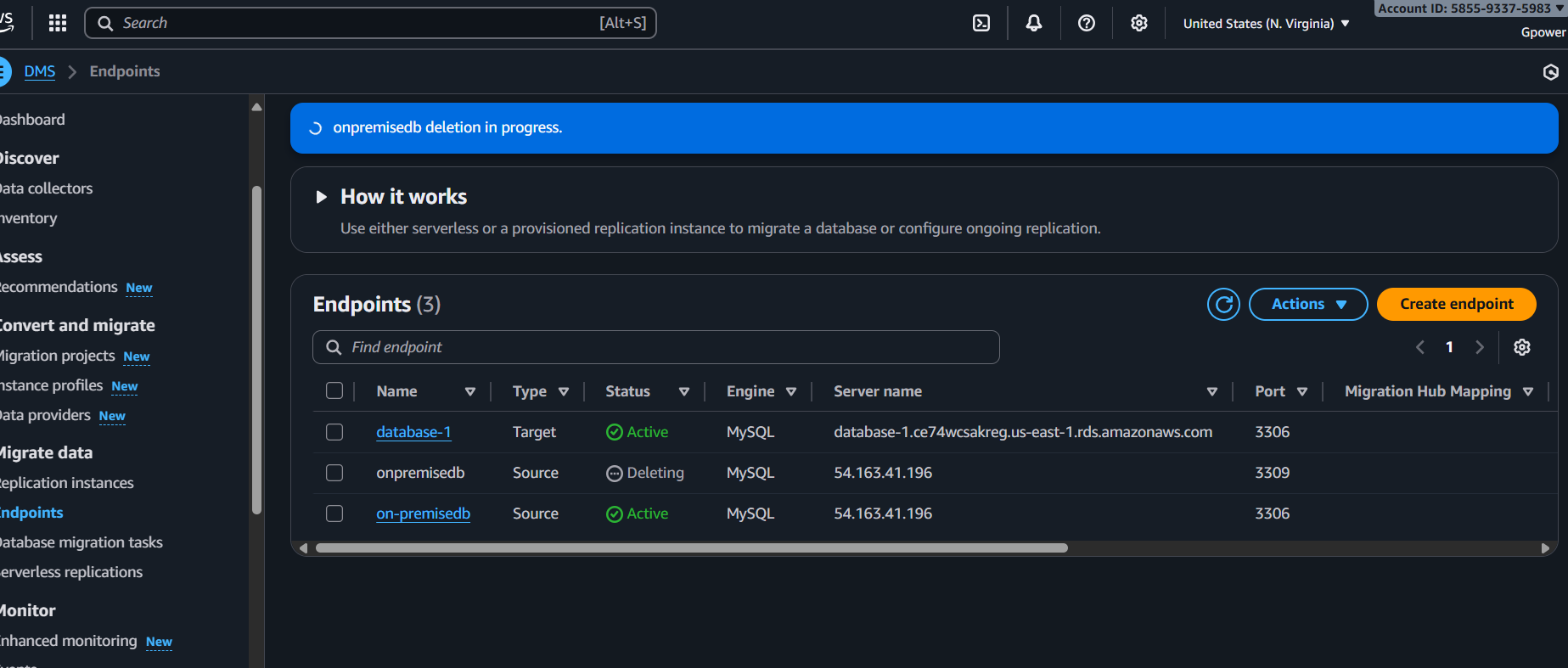
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

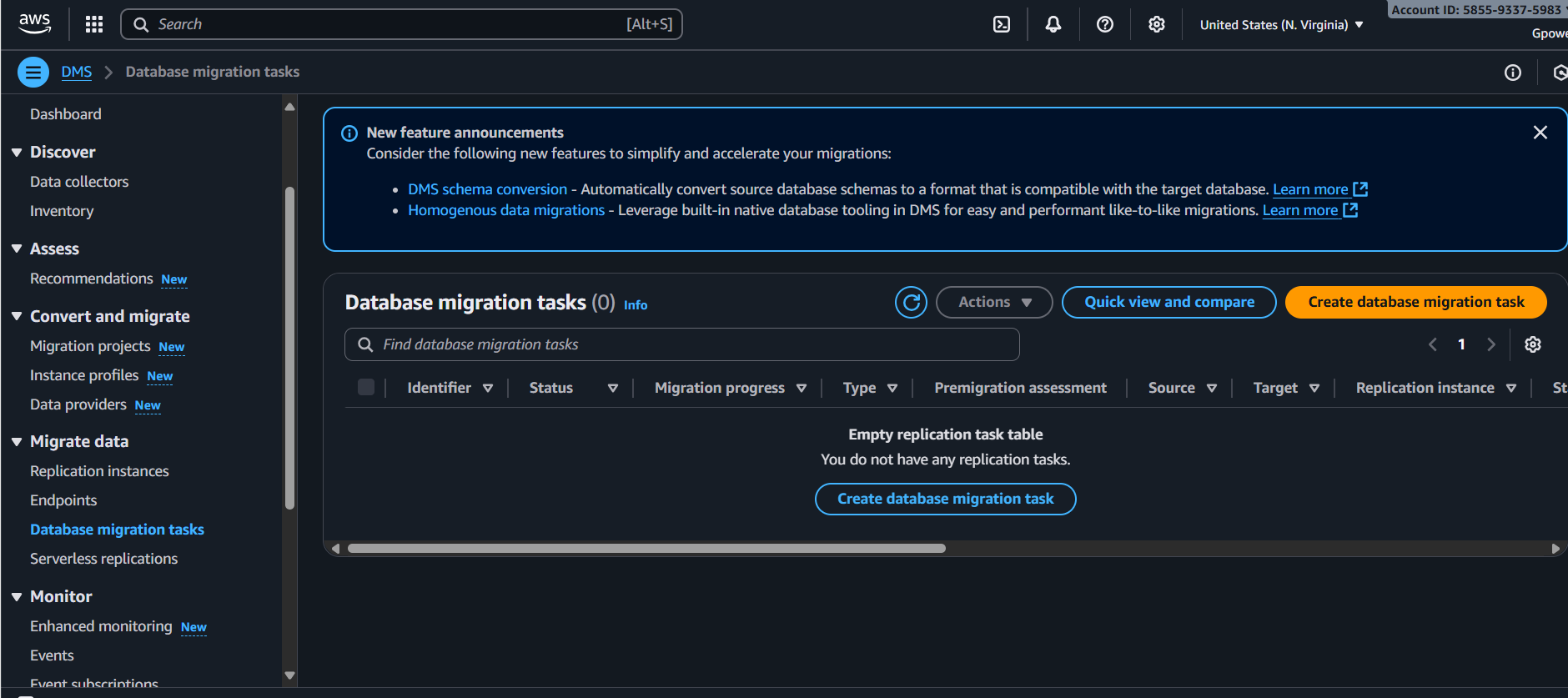
So we now have the two databases endpoints (the on premise MySQL and RDS MySQL) which have connections.



Again, test to connect for the target database (RDS) and successful. You can see the source and target.

The last step is to migrate data from on premise Database to AWS RDS.

Still in the DMS service, click on database migration task.



Click on Create database migration task. Make sure you are in us-east-1 region (N. Virginia).

For the replication instance, select the popup instance. For the source database select onpremisedb and the target database select your rds db.

Now the migration type. If you are migrating from dev environment to rds, select Migrate (first option). If its from production to RDS, select Migrate and replicate (second option) because of continuous data input from end users. If you are adding roll back plan and troubleshooting, choose replication.

But now, we are adding implementing dev database, select Migrate option.

A screenshot of a computer

AI-generated content may be incorrect.

On the LOB (Large Object) column settings, select full LOB mode. Screw down and click add new selection rule. On schema, select enter schema

A screenshot of a computer

AI-generated content may be incorrect.

Screw down again and untick “Turn on premigration assessment”. Then click Create database migration task. As its create it will start migrating all the data from on-premise to AWS RDS.

Load complete (100%)

A screenshot of a computer

AI-generated content may be incorrect.

Now, lets go back to sqlelectron to check the RDS database if the test database is there.

Confirmed

A screenshot of a computer

AI-generated content may be incorrect.

You can even issue an sql statement to select from the client table and all tables data are there.

A screenshot of a computer

AI-generated content may be incorrect.

**Project is complete**