# Exercise 3: Create Replication Instance and use Database Migration Service to copy data from the source SQL Server database to S3 and Aurora MySQL database

AWS [Database Migration Service](https://aws.amazon.com/dms/getting-started/) helps you migrate databases to AWS quickly and securely. 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 most widely used commercial and open-source databases.

The AWS [Schema Conversion Tool](http://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/Welcome.html) makes heterogeneous database migrations predictable by automatically converting the source database schema and a majority of the database code objects, including views, stored procedures, and functions, to a format compatible with the target database. Any objects that cannot be automatically converted are clearly marked so that they can be manually converted to complete the migration.

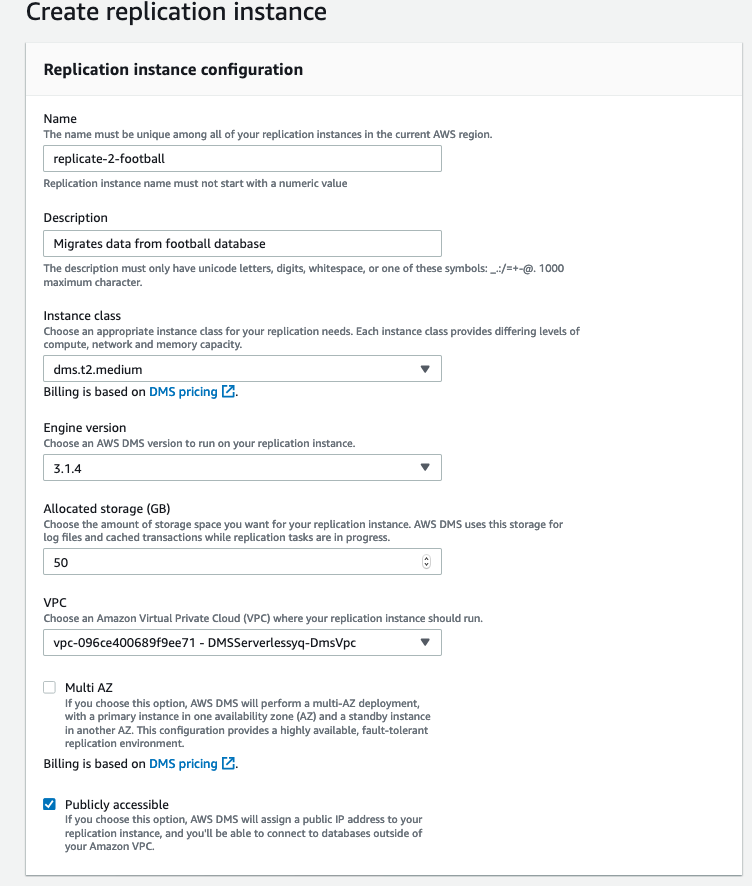
**Setting up the replication instance in Database Migration Service (DMS), source and target database endpoints and the DMS migration tasks**

In this exercise, you will do the following –

1. Create 2 DMS replication instances (one to migrate the football database and one for the baseball database)
2. Create 2 SQL Server source endpoints (one for the replication into S3 and one for replication into Aurora Serverless MySQL)
3. Create 2 target endpoints – one Aurora Serverless MySQL target endpoint and one S3 target endpoint
4. Create 2 replication tasks – one from SQL Server to S3 (for the baseball database) and one from SQL Server to Aurora Serverless MySQL (for the football database)

**Step 1: Create 2 DMS replication instances**

1. Navigate to the **Database Migration Service** in your AWS console
2. Choose the **Replication Instances** tab (on the left)
3. Click on the **Create replication instance** button and fill out required details. The page should look like this after filling out the details –



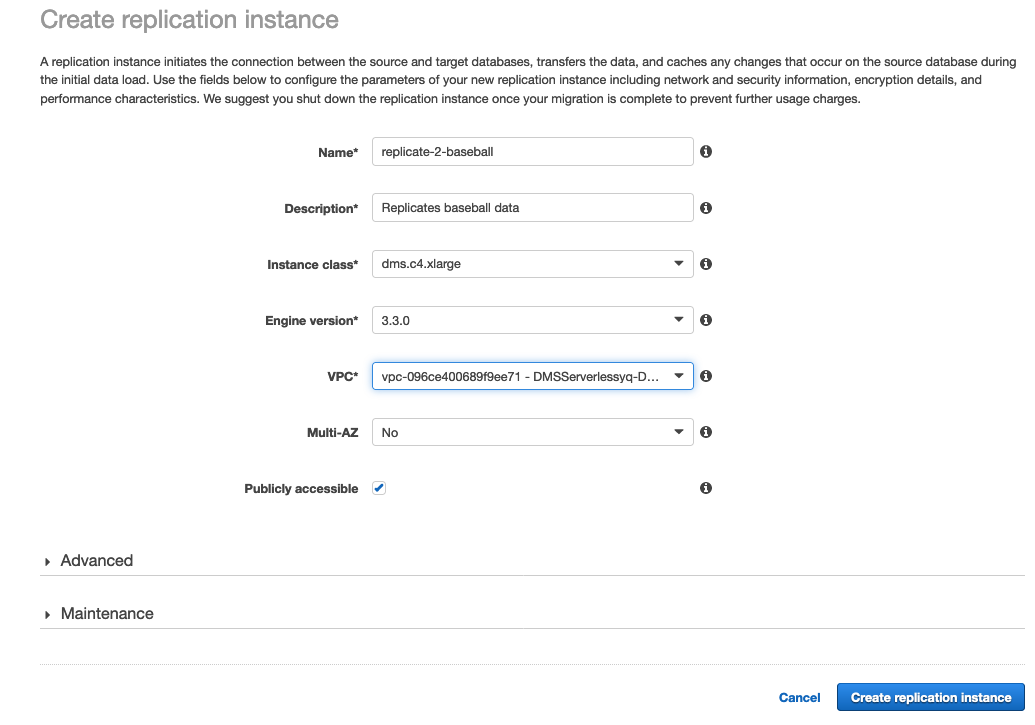
**Name** – *replicate-2-football*

**Description** – *Migrates data from football database*

**VPC** – *Choose the VPC that the CloudFormation template created from the previous exercise (you will see it from the CloudFormation output - ensure it has “DMSServerless” in its name)*

**Note** – Leave everything else default and click “**Create replication instance**”. Wait for the replication instance to become available before proceeding to the next step.

1. Perform the same steps to create another replication instance –



**Name** – *replicate-2-baseball*

**Description** – *Replicates data from baseball database*

**VPC** – Choose the VPC that the *CloudFormation template created from the previous exercise* (you will see it from the CloudFormation output)

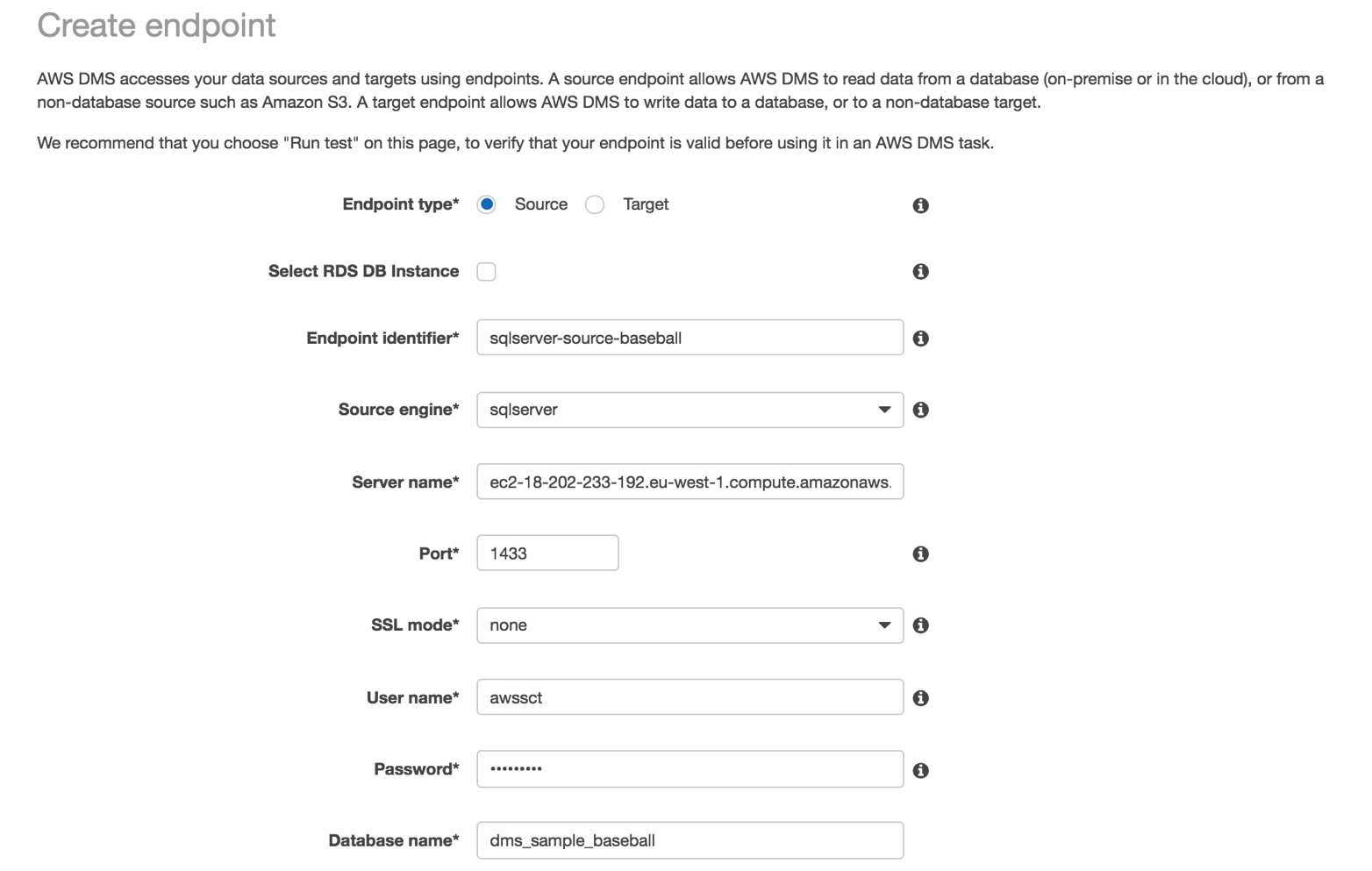
**Instance class** - *select dms.c4.xlarge*

**Multi-AZ** - *Select No*

Note – Leave everything else default and click “**Create replication instance**”.

**Step 2: Create 2 SQL Server endpoints (one for the baseball database and one for the football database)**

1. **Creating the SQL Server source endpoints**
2. Once the replication instances are available, click on the **Endpoints** tab on the left
3. Click **Create endpoint**
4. Fill required info –



**Endpoint type** – choose **Source** (as this is going to be the source endpoint)

**Endpoint identifier** – sqlserver-source-baseball

**Source engine** – choose **SQL Server**

**Server name** – put in the DNS name of the SQL Server EC2 instance

**Port** – type 1433

**SSL mode** – choose **none**

**User name** – awssct

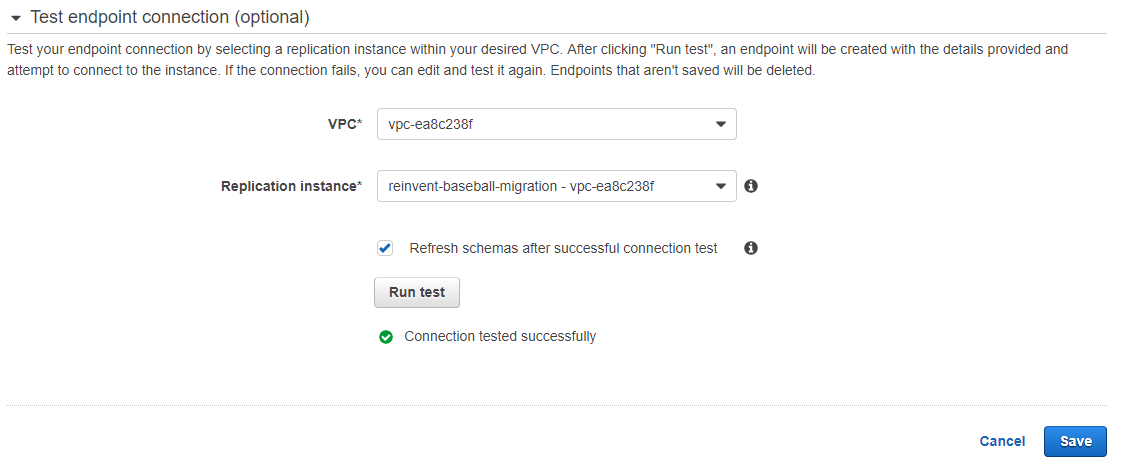
**Password** – Password1

**Database name** – dms\_sample\_baseball

Scroll down to get the optional endpoint test screen. Endpoint tests help you understand if the endpoint has been set up following all DMS pre-requisites and also checks if DMS can make connections to the endpoint.

1. Choose the VPC and the replication instance created in step 1.
2. Click **Run test**

If everything has been set up well, you should see '**Connection tested successfully**' check mark after the endpoint connection test completes –

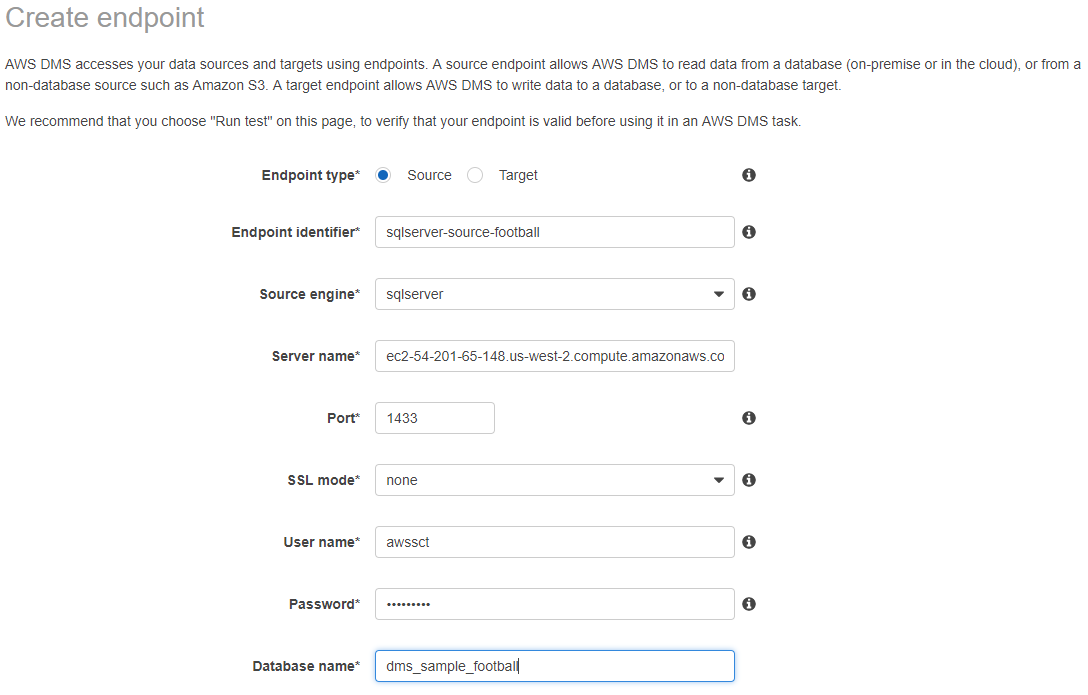


c

Click ‘**Create endpoint**’ and create other endpoints (1 more source endpoint for the football database)

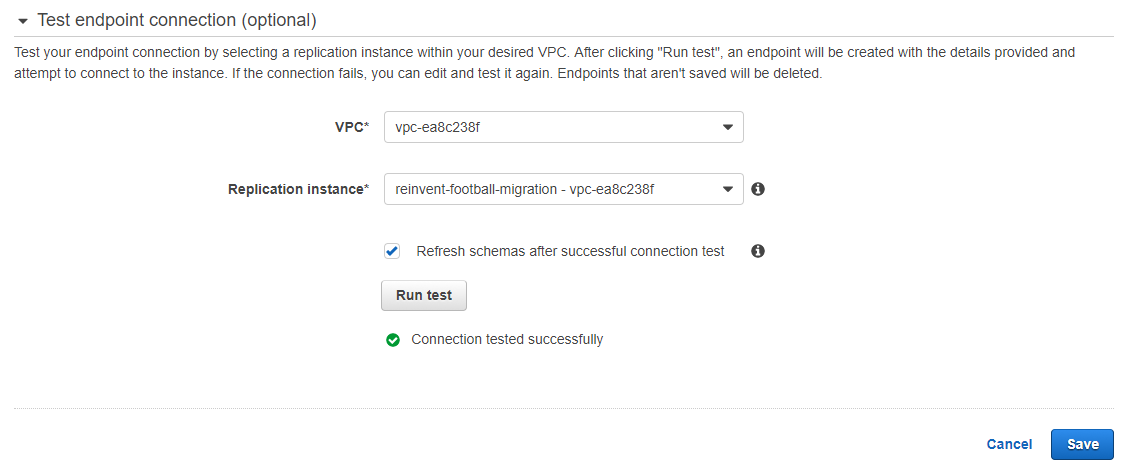
Given the other SQL Server source endpoint is similar, use the above instructions to create one but with the following differences –

1. Choose another endpoint identifier (Eg: sqlserver-source-football)
2. Change database name to dms\_sample\_football
3. Test connection with the second replication instance we created



c

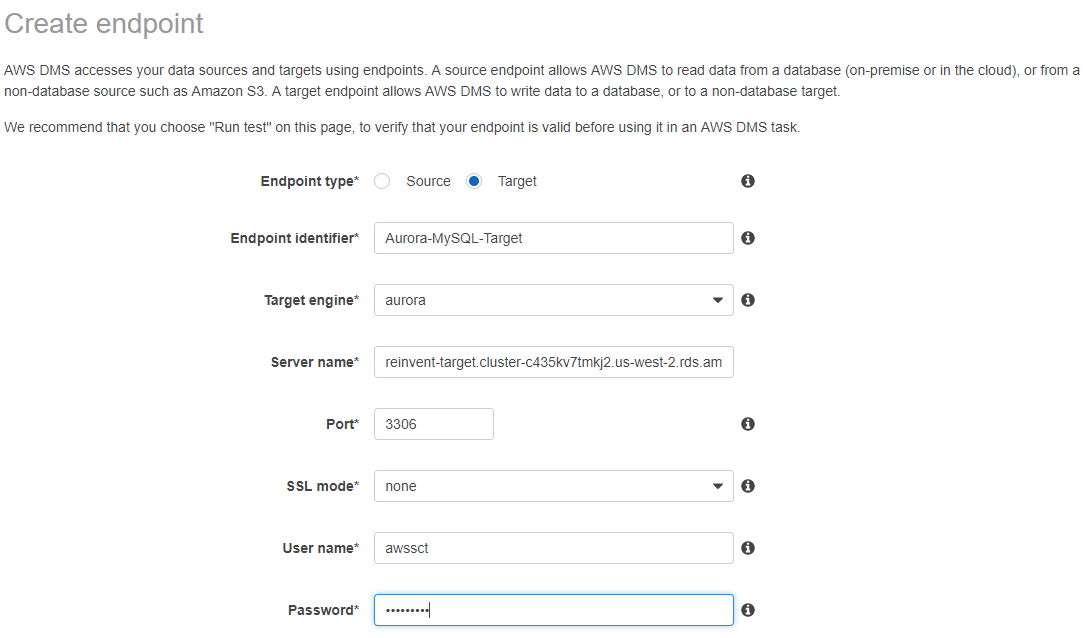
Test endpoint connection using **Run test** with the other replication instance –



c

**Step 3: Creating the Aurora Serverless MySQL target endpoint for the football database –**

1. Fill required info –



Endpoint type – choose target

Endpoint identifier – Aurora-MySQL-Target

Target engine – choose Aurora

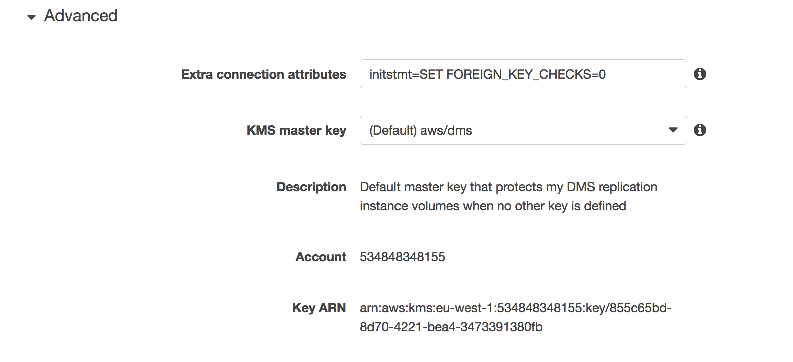
Server name – Cluster endpoint of the Aurora MySQL instance

Port – Port for Aurora MySQL instance (default is 3306)

SSL mode – none

Username – awssct

Password – Password1



c

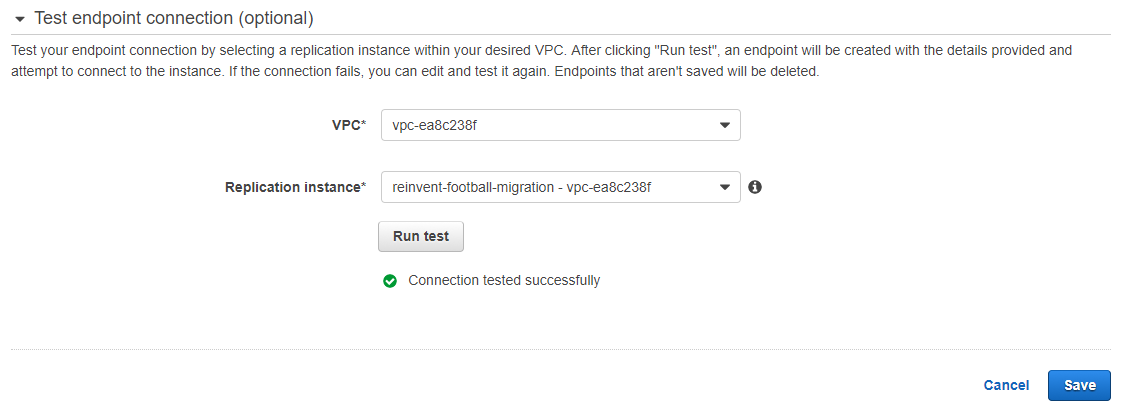
c

**IMPORTANT:** Expand advanced tab and add:

**initstmt=SET FOREIGN\_KEY\_CHECKS=0**

like in the screenshot (This will ensure all foreign key constraints are inactive for connections from DMS)

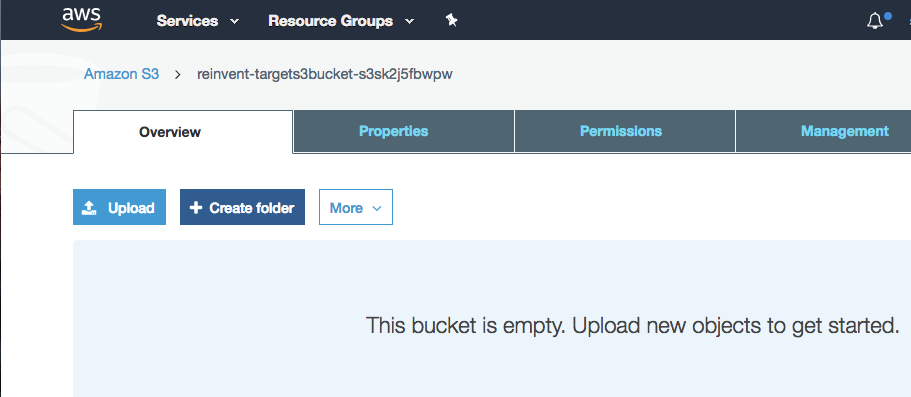
Next, do the same connection test with the required replication instance like when we did for the source endpoint. It should look like the below screenshot after a successful connection test –



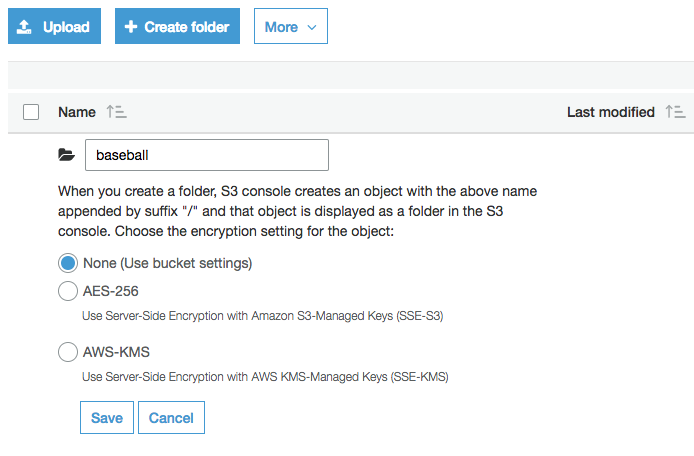
**Step 4: Creating the S3 Target Endpoint for the baseball database:**

**Create a Folder in the S3 bucket**

1. Open the **S3** page from the AWS Management Console by clicking on or searching for “**S3”**.
2. Locate the S3 Bucket created in this workshop. The S3 Bucket name can be found in the **CloudFormation outputs.**
3. Click **Create Folder**.

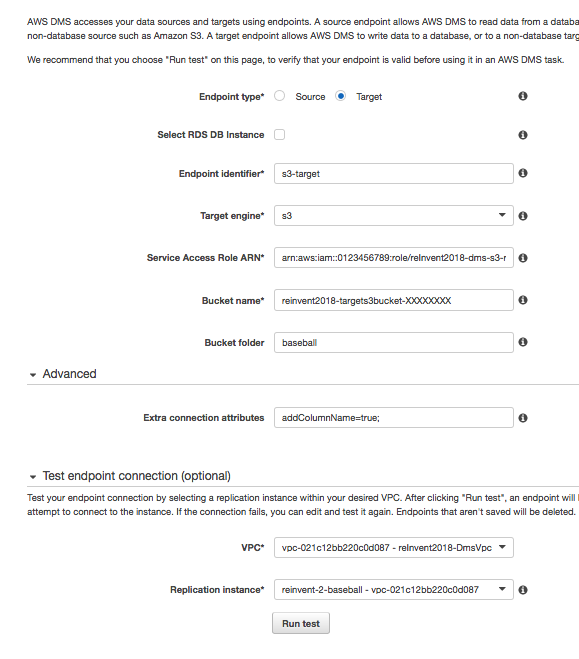


1. Create a folder called “baseball”. Use **bucket settings** for encryption and click **Save**.



**Create S3 target endpoint using the IAM role CloudFormation created–**

1. Go to DMS console and click on **Endpoints** on the left.
2. Fill out all required info –



* 1. Endpoint type – set this to **target**
  2. Endpoint identifier – **S3-target**
  3. Target engine – choose **s3**
  4. Service Access Role ARN – Paste the ARN of the IAM role from the output of the CloudFormation stack. Alternatively, you can also get this ARN from the IAM console under roles. Check to make sure there are no spaces in or after the ARN when you copy and paste.
     1. The Service Access Role ARN looks like this: *arn:aws:iam::12344566432:role/DMSServerlessxx-dms-s3-role*
  5. Bucket name – Paste the bucket name from the output of the CloudFormation stack
  6. Bucket folder- Give name of folder (baseball) within that bucket
  7. Under Advanced Tab, add **addColumnName=true;** to **extra connection attributes**.
  8. Test connection

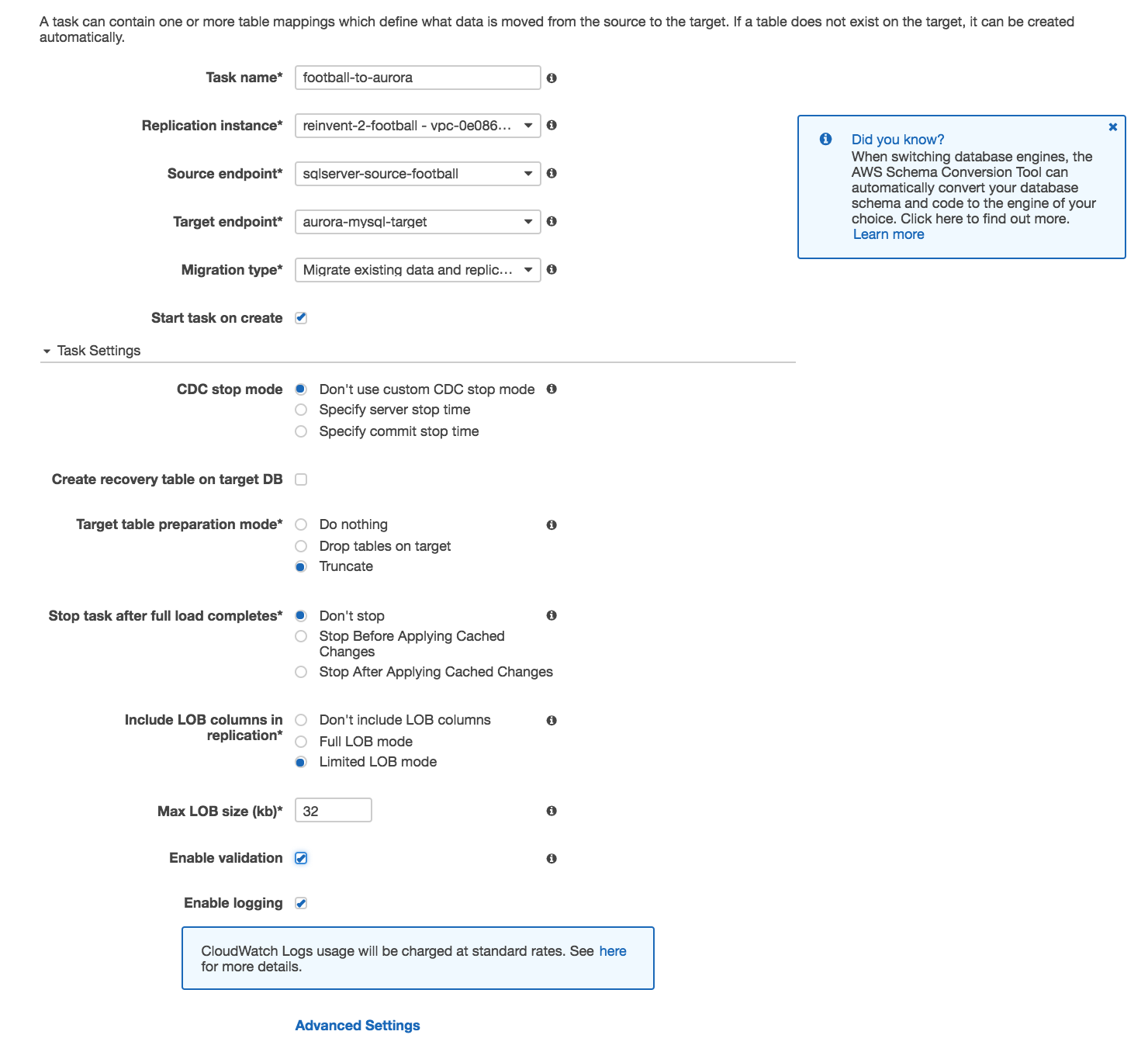
**Step 5: Creating the required DMS tasks to move the data:**

We are going to create 2 tasks –

1. SQL Server football database to Aurora Serverless MySQL
2. SQL Server baseball database to S3

**Replicate SQL Server football database to Aurora Serverless MySQL :**

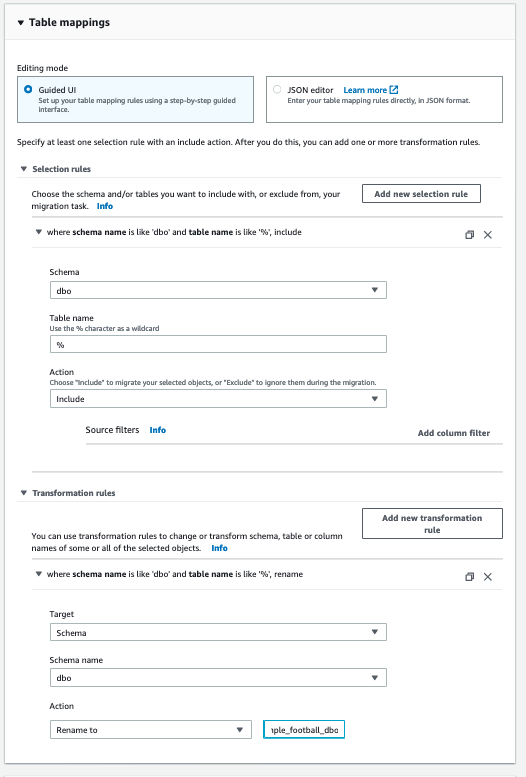
In the DMS console, click on the **Database Migration** **Tasks** section (on the left menu) and create a **new task** –



* Task name - **football-to-aurora**
* Replication instance - **replicate-football-migration**
* Source endpoint – **sqlserver-source-football**
* Target endpoint – **aurora-mysql-target**
* Migration type – choose **Migrate existing data and replicate ongoing changes**
* Target table preparation mode – choose **truncate** as we have already created data on the target
* Check **enable logging** and leave everything else default before scrolling down

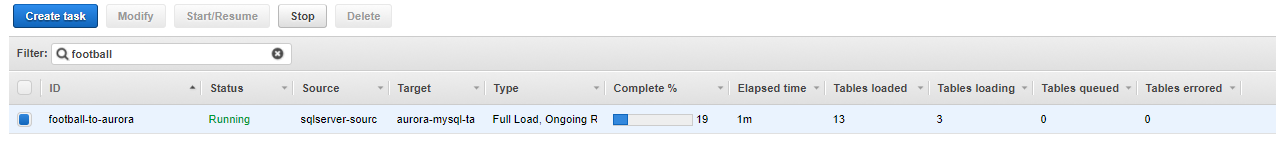
In the **Table Mappings** section:

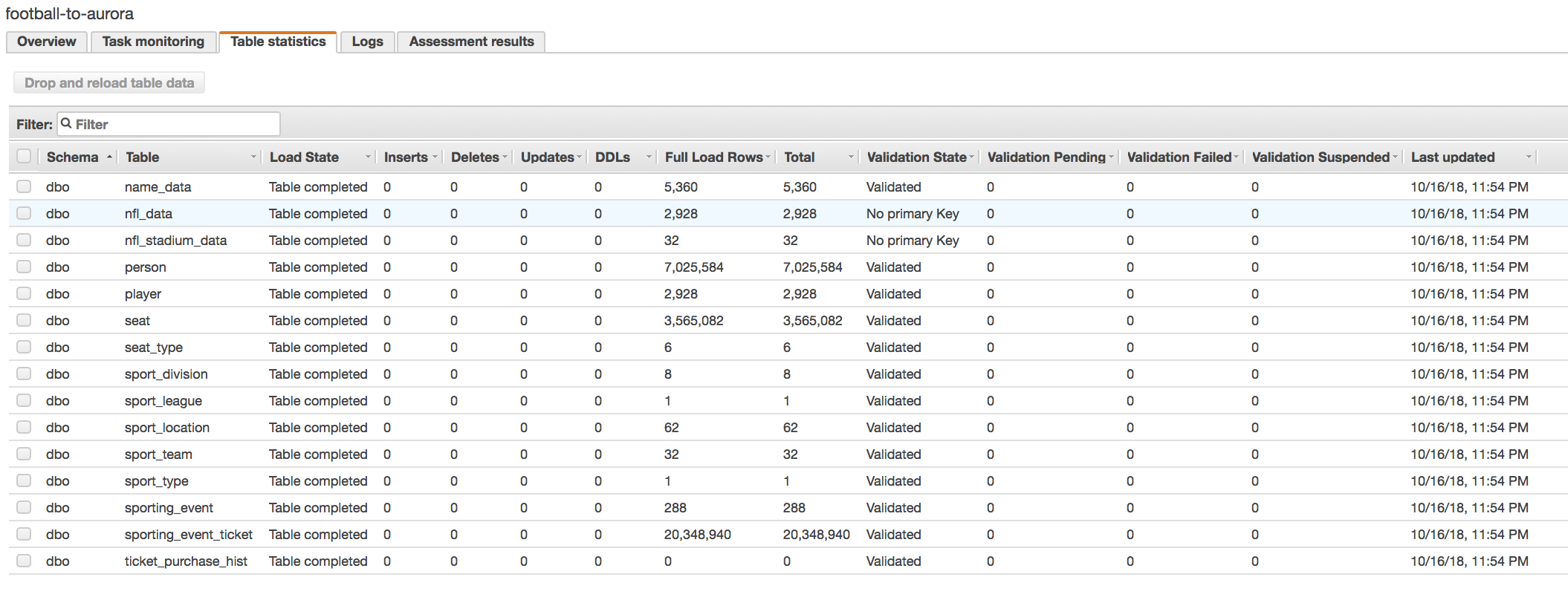
1. Select **Guided UI**
2. In the “Schema” drop down, select the **dbo schema** and click **add selection rule**
3. Click **add transformation rule**, choose **Schema** in the target drop down, choose **dbo** in the ‘schema name is’ drop down, choose **Rename to** in action and type **dms\_sample\_football\_dbo** in the text box as SCT created this schema in the target Aurora Serverless MYSQL instance.

****

Click on ‘**Create task**’ and it will start the migration task from SQL Server to Aurora MySQL. Notice the "Validation State" tab which tells you whether the data on the source and target match.

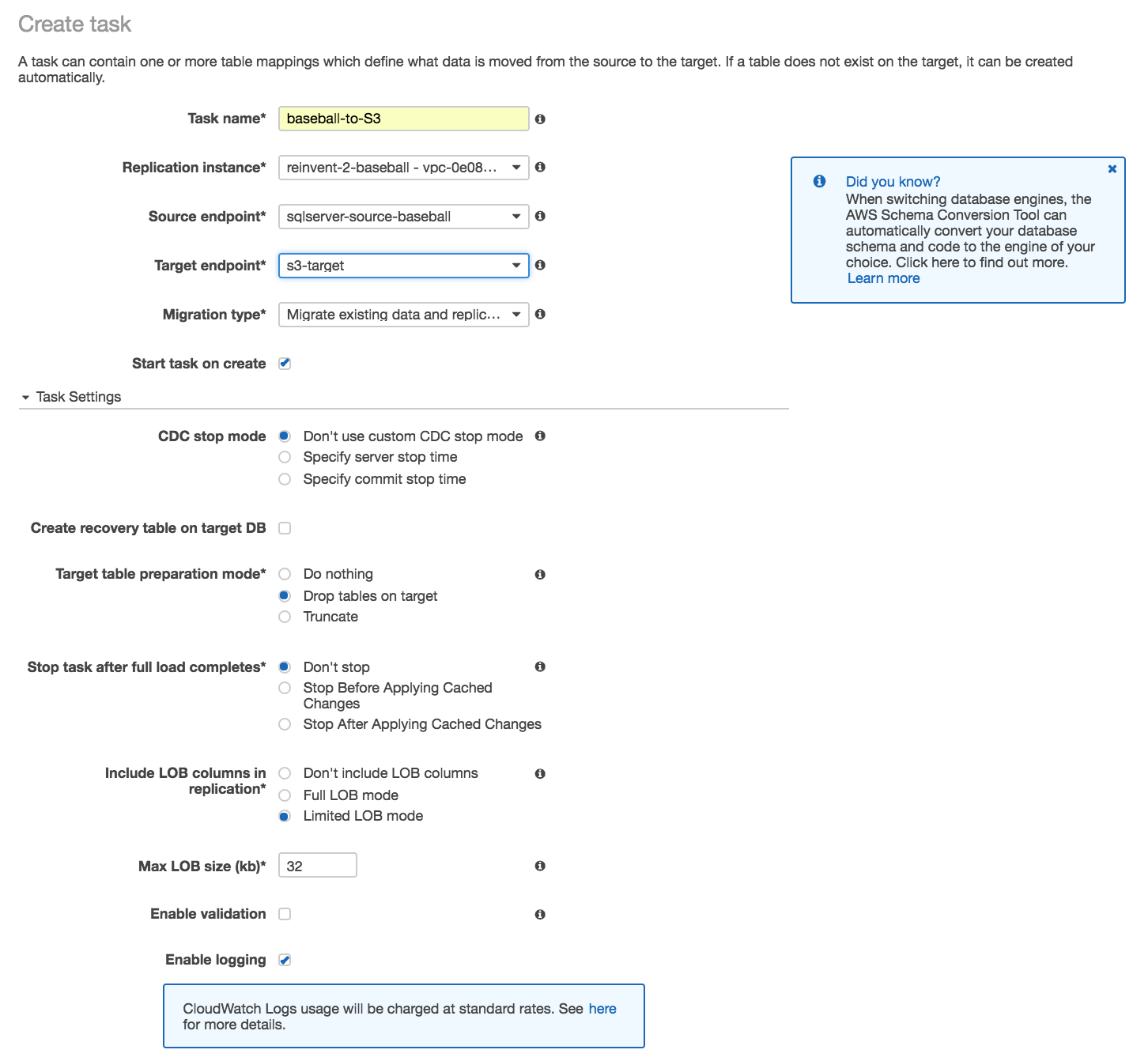
Here is a screen shot of the table statistics tab once the migration is in progress –



****

We don't need to wait for aurora replication task to complete but rather continue with rest of exercises and come back and validate later.

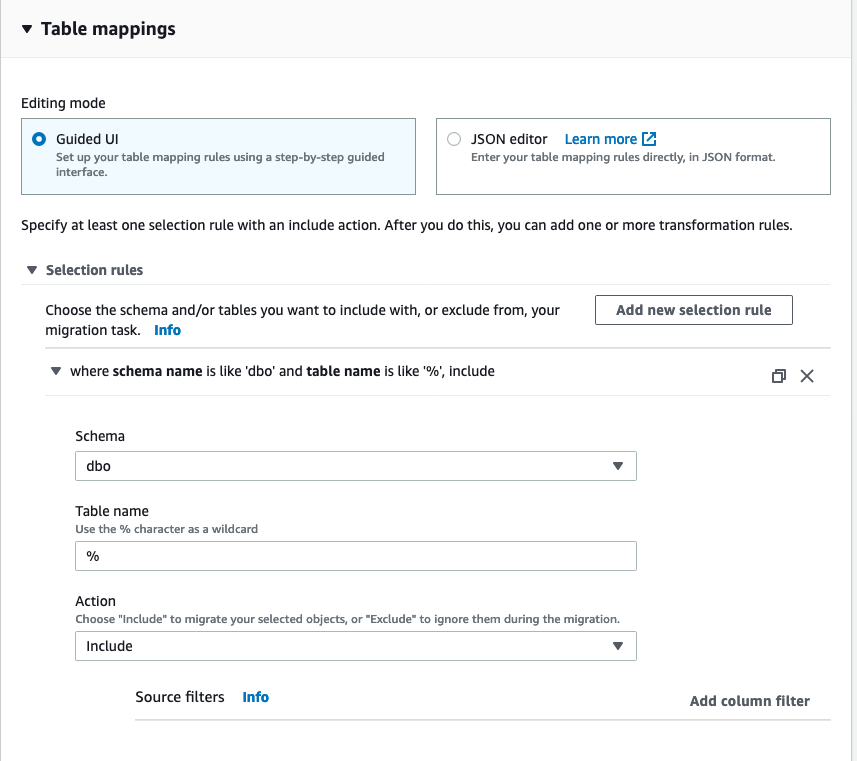
**Next:** Replicate SQL Server Baseball database to S3



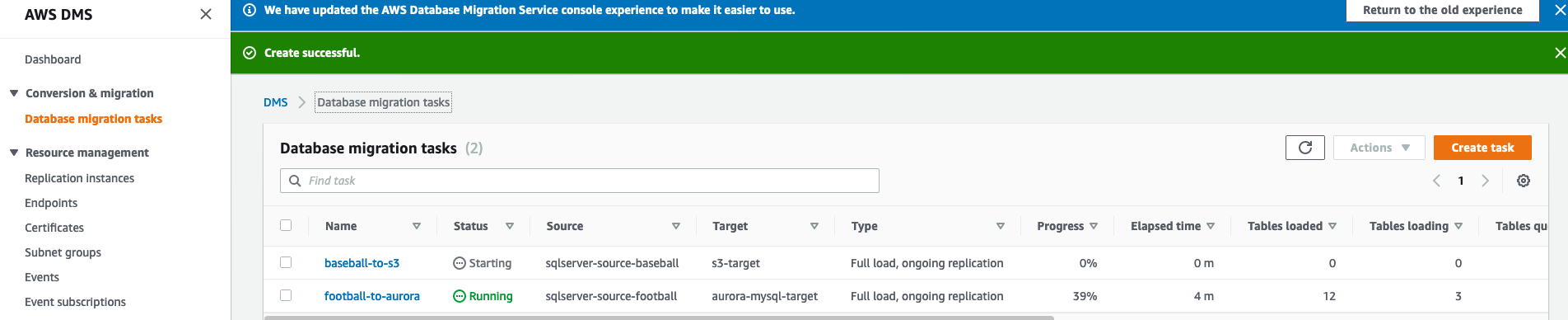
* Task name: **baseball-to-S3**
* Replication instance – **replicate-baseball-migration**
* Source endpoint – **sqlserver-source-baseball**
* Target endpoint – **s3-target**
* Migration type – choose **Migrate existing data and replicate ongoing changes**
* Target table preparation mode – choose **drop table on target**
* **Leave the enable validation box unchecked** as DMS currently doesn't support validation on S3 targets. Cheking the box will throw web service error.
* Check **enable logging** and leave everything else default before scrolling down

In the **Table Mappings** section:

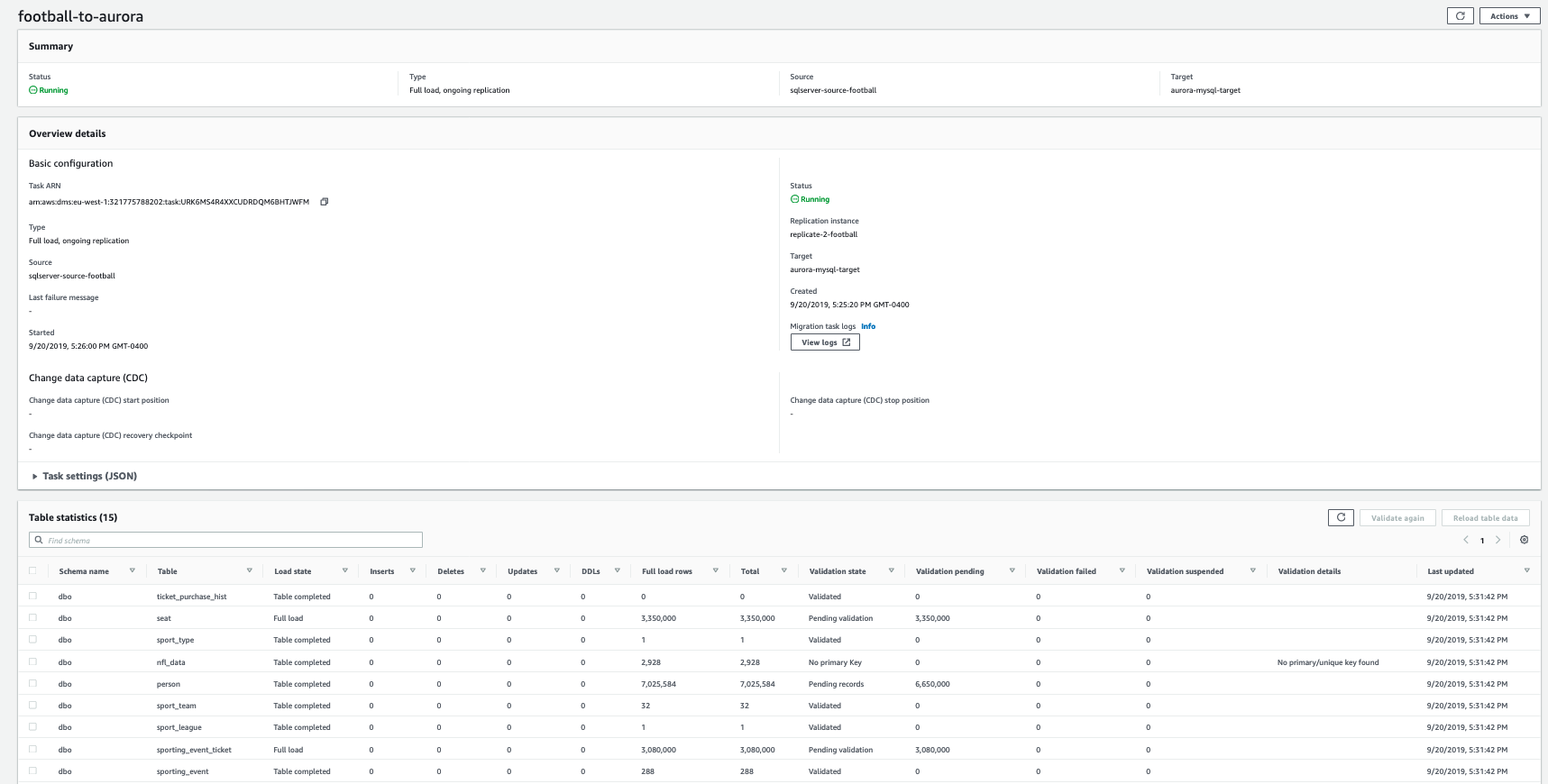
1. Select **Guided UI**
2. select the **dbo schema** and click **add selection rule**
3. We will **not** add any transformation rules here



Here is a screen shot of the table statistics tab once the migration is in progress



Click on each task. You will see the following view:



Scroll down to the **Table Statistics** section to see the tables that are being replicated from the source SQL Server to the target repositories for each task. Analyze the data.

**Exercise recap**

In this exercise, we learnt to do the following:

1. Launch DMS replication instance
2. Create source and target endpoints in DMS
3. Run migration tasks using those endpoints to replicate data