



Amazon Web Services Data Engineering Immersion Day

Database Migration Services Instructor Lab Setup
March 2020

Table of Contents

<i>Limit Instruction:</i>	2
<i>Introduction</i>	2
<i>Create the Instructor Environment</i>	3
<i>Changing RDS Security Group</i>	6
<i>Access Database from SQL Client (Optional)</i>	8
<i>Generate and Replicate the CDC Data (Optional)</i>	9

Limit Instruction:

This immersion day required each student to have their own account. If you are sharing single account with multiple students by creating a multiple IAM users, Account can hit following default service limit:

- VPC – VPCs per Region 5
- Glue - Number of crawlers per account 50
- Glue - Number of concurrent jobs runs per account 50
- Glue - Maximum DPU's used by a role at one time 300
- S3 – Number of buckets per account 100
- Athena - Number of DDL queries you can submit at the same time 20
- Athena - Number of DML queries you can submit at the same time 20
- RDS – Make sure you have enough disk space available in your RDS instance, if want to run DMS Change Data Capture (CDC) as generating large amount of data can exhaust RDS disk space.
- DMS - Make sure you have enough disk space available in your DMS replication instance, if want to run DMS Change Data Capture (CDC) as transferring large amount of CDC data can exhaust disk space.

Introduction

*****Make sure you select the us-east-1 (Virginia) region*****

The Database Migration Services (DMS) hands-on lab provide a scenario, where participant learns to hydrate Amazon S3 data lake with a relation database. To achieve that, participants need a source endpoint and this guide helps instructors set up a PostgreSQL database with public endpoint as the source database.

In this lab, you will complete the following tasks using AWS CloudFormation template:

1. Create the source database environment.
2. Hydrate the source database environment.
3. Update the source database environment to demonstrate CDC (Change Data Capture) replication within DMS.
4. Create Lambda function to trigger CDC data which will be replicated to Amazon S3 by DMS CDC endpoint.

Relevant information about this lab:

- Expected setup time: 15 minutes
- Source database name: sportstickets
- Source schema name: dms_sample

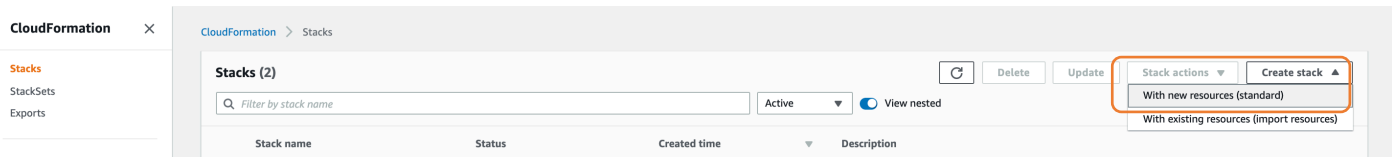
Instructor will provide source database details to participants during main lab to configure source endpoint.

Labs are also available in GitHub - <https://github.com/aws-samples/data-engineering-for-aws-immersion-day>

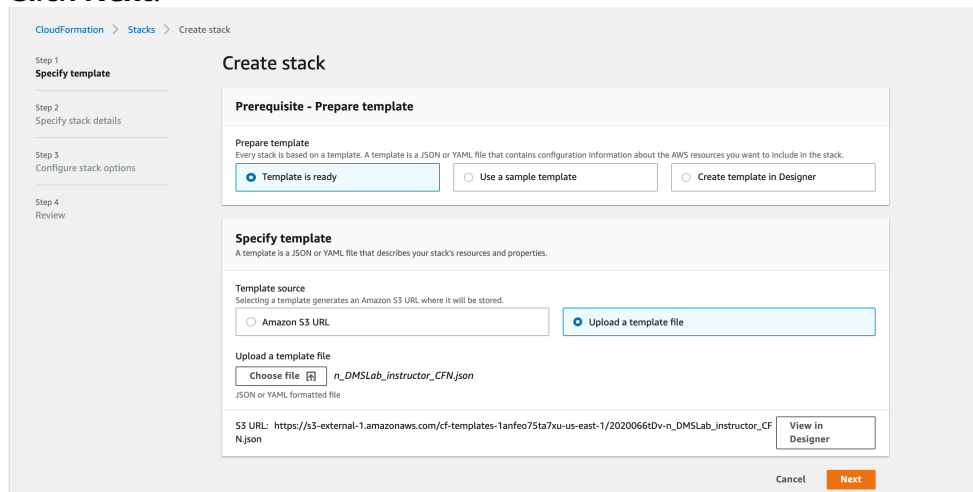
Create the Instructor Environment

In this section, you are going to create a PostgreSQL RDS instance as data source for AWS Data Migration Service to consume by lab attendees for data migration to Amazon S3 data lake.

1. Sign in to the Console where you will host the source database environment.
2. Navigate to the **AWS CloudFormation** page.
3. Launch a new stack with the AWS CloudFormation template **DMSLab_instructor_CFN.json** provided with your lab package. Make sure to select us-east-1 (Virginia) region.
 - a. On top right corner, Click on **"Create Stack"** and select **"With new resources"**.



- b. In **"Create Stack"** Page, select **"Template is ready"** and for template source, select **"Upload a template file"**.
- c. Locate the **DMSLab_Instructor_CFN.json** template from your local machine.
- d. Click **Next**.



- e. In Specify stack details, provide a name for **Stack Name** as “**dmslab-instructor**”.

Specify stack details

Stack name

Stack name

dmslab-instructor

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

No parameters

There are no parameters defined in your template

Cancel Previous **Next**

- f. Click on **Next**.
- g. In review page, review all the details, scroll down and check the box to acknowledge the policy and then click on **Create Stack**.

► Quick-create link

Capabilities

ⓘ The following resource(s) require capabilities: [AWS::IAM::Role]

This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions. [Learn more](#)

☒ I acknowledge that AWS CloudFormation might create IAM resources.

Cancel Previous Create change set **Create stack**

- h. Launch the stack. It may take 15 minutes for the stack to launch. This stack creates a new VPC, Subnets, Security groups, EC2 instance, Route table, Routes, and an RDS Postgres instance. You can see all resources listed below:

Database Migration Services Instructor Environment for the Lab

dmslab-instructor

Delete

Update

Stack actions ▼

Create stack ▼

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Resources (27)

Q Search resources

Logical ID	Physical ID	Type	Status	Status reason
EC2SubNet	subnet-0b46150fc43e400bc	AWS::EC2::Subnet	CREATE_COMPLETE	-
GenerateCDCData	GenerateCDCData	AWS::Lambda::Function	CREATE_COMPLETE	-
LambdaExecutionRole	dmslab-instructor-LambdaExecutionRole-1QS0V5OCLPR09	AWS::IAM::Role	CREATE_COMPLETE	-
RDSSubNet	subnet-0477e0e0071e80331	AWS::EC2::Subnet	CREATE_COMPLETE	-
RDSSubNet2	subnet-00dea43618c4868d8	AWS::EC2::Subnet	CREATE_COMPLETE	-
dbpgdataengdmsgroup	dmslab-instructor-dbpgdataengdmsgroup-1pbby1ntnpggq	AWS::RDS::DBParameterGroup	CREATE_COMPLETE	-
dbsgdefault	dmslab-instructor-dbsgdefault-1p5usgck1gq0a	AWS::RDS::DBSecurityGroup	CREATE_COMPLETE	-
dbsubnetdefaultdmsinst ructorvpc	dmslab-instructor-dbsubnetdefaultdmsinst ructorvpc-13e6pv5p7lbvy	AWS::RDS::DBSubnetGroup	CREATE_COMPLETE	-

- i. Go to the **Outputs** tabs of AWS CloudFormation stack and note down the instance Endpoint information for your RDS endpoint, which will be similar to information shown in below screenshot

dmslab-instructor-sd

Delete

Update

Stack actions ▼

Create stack ▼

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Outputs (2)

Q Search outputs

Key	Value	Description	Export name
CDCFunction	arn:aws:lambda:us-east-1:789211807855:function:GenerateCDCData	CDC Function	-
DMSInstanceEndpoint	dmslabinstance.ccla1oozkrry.us-east-1.rds.amazonaws.com	DMS Instance Endpoint	-

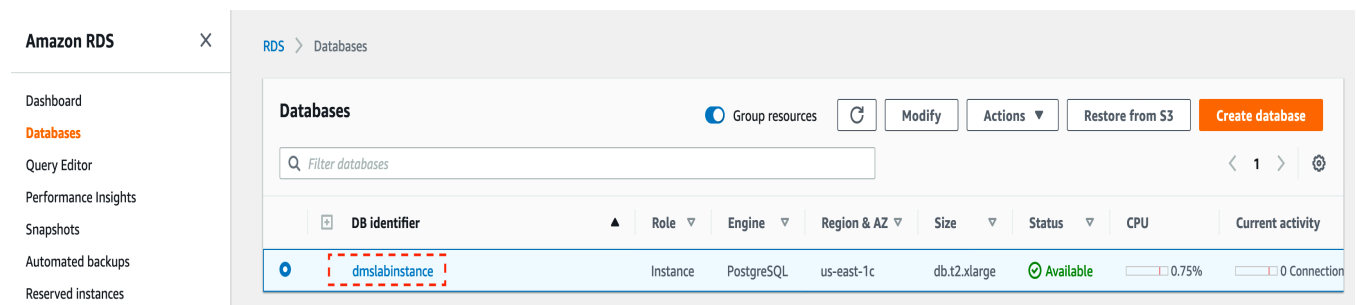
Changing RDS Security Group

Currently your RDS source end point is not open to connect to outside world for security reason. You need to open RDS security group to accept traffic from intended range of IP address. As it is difficult to determine range of IP address of workshop environment, so to have smooth experience of running lab you can temporally allow inbound traffic from all IP address (0.0.0.0/0 CIDR range).

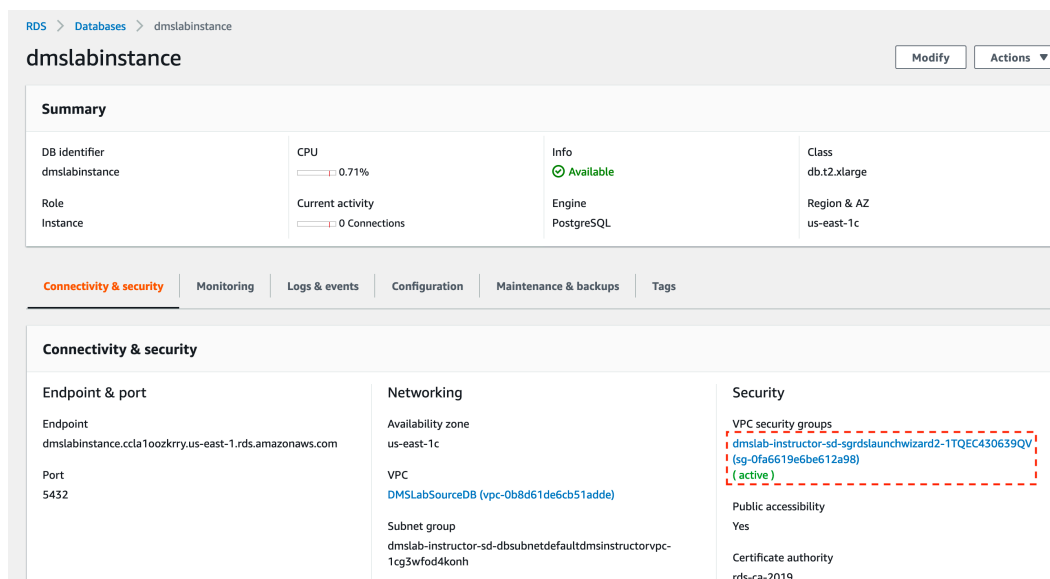
Warning: It is not best practice to allow ALL CIDR range in your database security group. You should never apply open to all IP CIDR range while working on actual workload.

Follow below steps to open security group for students to connect with source RDS data base for DMS full data and CDC data dump:

1. Go to the RDS and double click on “dmslabinstance” **DB identifier** as shown below:



2. Click **VPC security groups** under **Connectivity & security** tab as shown below:



3. In Security group screen, Go to **Inbound** tab and click on **Edit** as shown below

Database Migration Services Instructor Environment for the Lab

Search: sg-0fa6619e6be612a98 Add filter

Name	Group ID	Group Name	VPC ID	Owner	Description
DMSLabRD...	sg-0fa6619e6be612a98	dmslab-instructor-sd-sgrdsla...	vpc-0b8d61de6cb51adde	789211807855	RDS Security Group

Security Group: sg-0fa6619e6be612a98

Description Inbound Outbound Tags

Edit

Type	Protocol	Port Range	Source	Description
PostgreSQL	TCP	5432	72.21.196.67/32	
PostgreSQL	TCP	5432	sg-0d1979886d7072628 (dmslab-instructo	

4. Update Inbound rule to "Anywhere" from hardcoded value "72.21.196.67/32", as shown in below screen. You can also update to your own IP address if want running both lab in same

Edit inbound rules

Type	Protocol	Port Range	Source	Description
PostgreSQL	TCP	5432	Anywhere	e.g. SSH for Admin Desktop
PostgreSQL	TCP	5432	My IP	e.g. SSH for Admin Desktop

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel Save

5. Click on **Save** and now everyone will be able to connect to source RDS instance for lab purpose to ingest data using DMS endpoint.

Search: sg-0fa6619e6be612a98 Add filter

Name	Group ID	Group Name	VPC ID	Owner	Description
DMSLabRD...	sg-0fa6619e6be612a98	dmslab-instructor-sd-sgrdsla...	vpc-0b8d61de6cb51adde	789211807855	RDS Security Group

Security Group: sg-0fa6619e6be612a98

Description Inbound Outbound Tags

Edit

Type	Protocol	Port Range	Source	Description
PostgreSQL	TCP	5432	0.0.0.0/0	
PostgreSQL	TCP	5432	:::0	
PostgreSQL	TCP	5432	sg-0d1979886d7072628 (dmslab-instructo	

Note: Make sure to remove "Anywhere" inbound rule from security group as soon as you are done with DMS main lab.

Optionally, You can read through the documentation to better understand the source database environment. The GitHub repository for aws-database-migration-samples is located here:

<https://github.com/aws-samples/aws-database-migration-samples/tree/master/PostgreSQL/sampledb/v1>

Access Database from SQL Client (Optional)

You can follow below instruction to setup SQL Workbench to access your Postgres Database from SQL client:

<https://aws.amazon.com/getting-started/tutorials/create-connect-postgresql-db/>

In SQL Workbench:

Run following query to find out all Schema and table created.

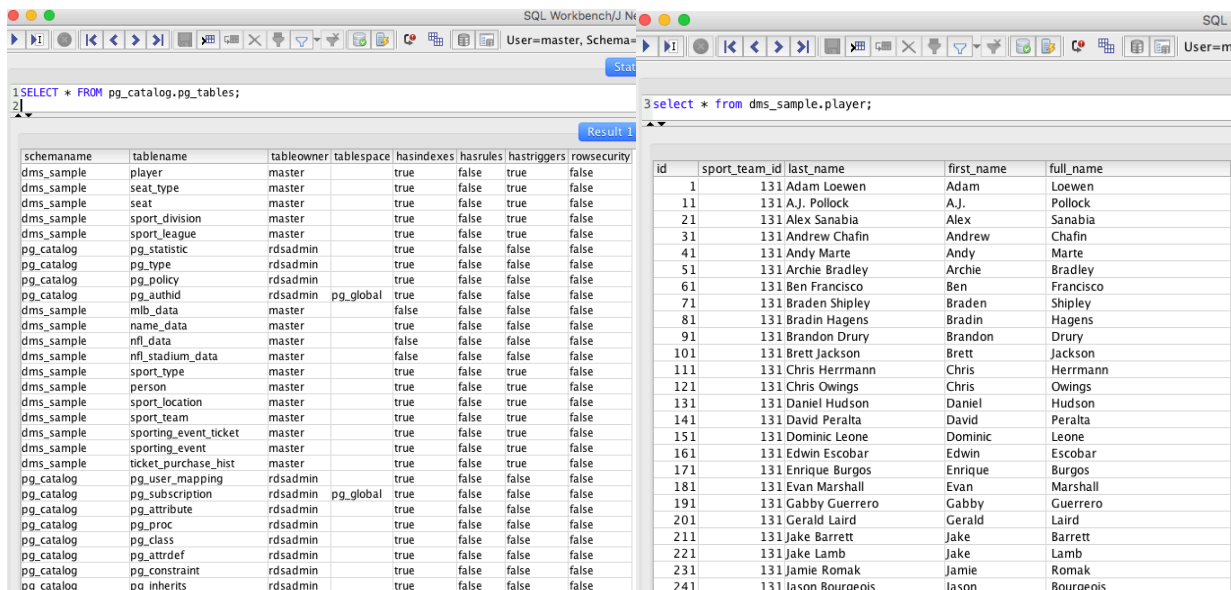
```
SELECT * FROM pg_catalog.pg_tables;
```

Use following query to analyze a table

```
select * from schemaname.tablename;
```

For example:

```
select * from dms_sample.player;
```



The screenshot shows the SQL Workbench interface with two queries and their results.

Query 1: `SELECT * FROM pg_catalog.pg_tables;`

Result 1:

schemaname	tablename	tableowner	tablespace	hasindexes	hasrules	hastriggers	rowsecurity
dms_sample	player	master		true	false	true	false
dms_sample	seat_type	master		true	false	true	false
dms_sample	seat	master		true	false	true	false
dms_sample	sport_division	master		true	false	true	false
dms_sample	sport_league	master		true	false	true	false
pg_catalog	pg_statistic	rdadmin		true	false	false	false
pg_catalog	pg_type	rdadmin		true	false	false	false
pg_catalog	pg_policy	rdadmin		true	false	false	false
pg_catalog	pg_authid	rdadmin	pg_global	true	false	false	false
dms_sample	mlb_data	master		false	false	false	false
dms_sample	name_data	master		true	false	false	false
dms_sample	nfl_data	master		false	false	false	false
dms_sample	nfl_stadium_data	master		false	false	false	false
dms_sample	sport_type	master		true	false	true	false
dms_sample	person	master		true	false	true	false
dms_sample	sport_location	master		true	false	true	false
dms_sample	sport_team	master		true	false	true	false
dms_sample	sporting_event_ticket	master		true	false	true	false
dms_sample	sporting_event	master		true	false	true	false
dms_sample	ticket_purchase_hist	master		true	false	true	false
pg_catalog	pg_user_mapping	rdadmin		true	false	false	false
pg_catalog	pg_subscription	rdadmin	pg_global	true	false	false	false
pg_catalog	pg_attribute	rdadmin		true	false	false	false
pg_catalog	pg_proc	rdadmin		true	false	false	false
pg_catalog	pg_class	rdadmin		true	false	false	false
pg_catalog	pg_attrdef	rdadmin		true	false	false	false
pg_catalog	pg_constraint	rdadmin		true	false	false	false
pg_catalog	pg_inherits	rdadmin		true	false	false	false

Query 2: `select * from dms_sample.player;`

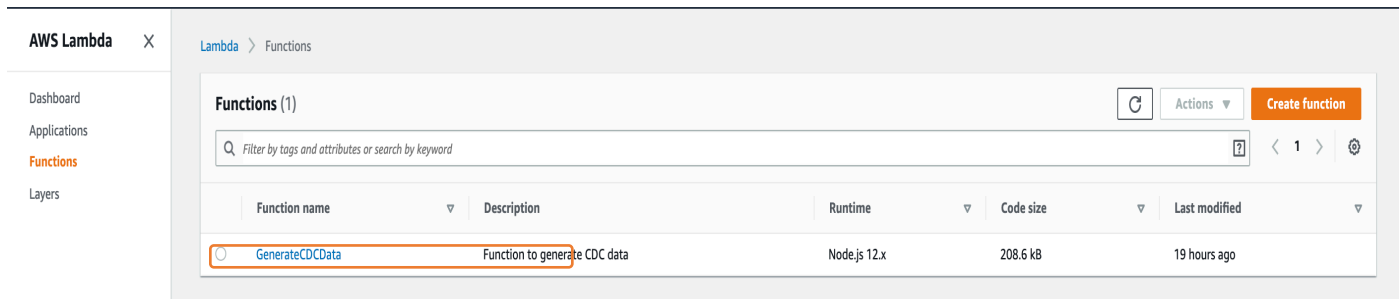
Result 2:

id	sport_team_id	last_name	first_name	full_name
1	131	Adam Loewen	Adam	Loewen
11	131	A.J. Pollock	A.J.	Pollock
21	131	Alex Sanabia	Alex	Sanabia
31	131	Andrew Chafin	Andrew	Chafin
41	131	Andy Marte	Andy	Marte
51	131	Archie Bradley	Archie	Bradley
61	131	Ben Francisco	Ben	Francisco
71	131	Braden Shipley	Braden	Shipley
81	131	Bradin Hagens	Bradin	Hagens
91	131	Brandon Drury	Brandon	Drury
101	131	Brett Jackson	Brett	Jackson
111	131	Chris Herrmann	Chris	Herrmann
121	131	Chris Owings	Chris	Owings
131	131	Daniel Hudson	Daniel	Hudson
141	131	David Peralta	David	Peralta
151	131	Dominic Leone	Dominic	Leone
161	131	Edwin Escobar	Edwin	Escobar
171	131	Enrique Burgos	Enrique	Burgos
181	131	Evan Marshall	Evan	Marshall
191	131	Gabby Guerrero	Gabby	Guerrero
201	131	Gerald Laird	Gerald	Laird
211	131	Jake Barrett	Jake	Barrett
221	131	Jake Lamb	Jake	Lamb
231	131	Jamie Romak	Jamie	Romak
241	131	Jason Bourgeois	Jason	Bourgeois

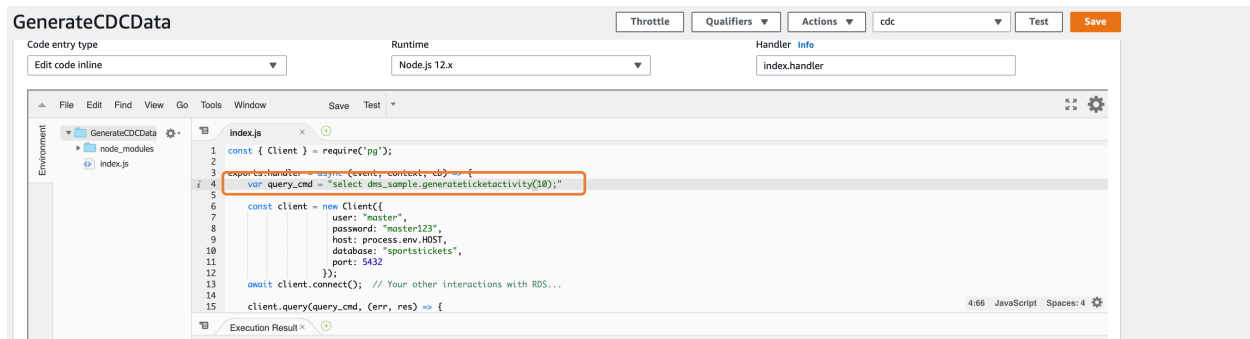
Following sections are optional you only need to execute, if you want to show change data capture replication with DMS.

Generate and Replicate the CDC Data (Optional)

When you want to generate transactions to demonstrate DMS CDC (Change Data Capture) functionality, navigate to Lambda console and you will see a pre-built Lambda function named **"GenerateCDCData"**.



1. Click on the function and scroll down. You will see the code for this function. Copy the below query and paste it in the placeholder (value) of this code line:
" var query_cmd= "<insert-SQL-query-here>" "
2. Run this query first: **select dms_sample.generateticketactivity(10);**



This query will generate 10 ticket sales in batches of 1-6 tickets to randomly selected people for a random price (within a range.) A record of each transaction is recorded in the **ticket_purchase_hist** table.

3. Click on **Save** and then click on **Test** to run the function. You can create an empty event as shown here:

Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

☒ Create new test event
 ☐ Edit saved test events

Event template

Hello World

Event name

cdcevent

```

1 {
2   |
3 }

```

You will see no error in lambda log

```

1 const { Client } = require('pg');
2
3 exports.handler = async (event, context, cb) => {
4   var query_cmd = "select dms_sample.generateticketactivity(10);"
5
6   const client = new Client({
7     user: "master",
8     password: "master123",
9     host: process.env.HOST,
10    database: "sportstickets",
11    port: 5432
12  });
13  await client.connect(); // Your other interactions with RDS...
14  client.query(query_cmd, (err, res) => {
15    console.log(err ? err.stack : res.rows[0])
16  })
17 }

```

Execution results

Status: **Succeeded** Max Memory Used: 77 MB Time: 236.77 ms

Response: null

Request ID: "9affcfd-99ee-4895-a7ab-837851b132bf"

Function Logs:

START RequestId: 9affcfd-99ee-4895-a7ab-837851b132bf Version: \$LATEST

END RequestId: 9affcfd-99ee-4895-a7ab-837851b132bf

REPORT RequestId: 9affcfd-99ee-4895-a7ab-837851b132bf Duration: 236.77 ms Billed Duration: 300 ms Memory Size: 128 MB Max Memory Used: 77 MB Init Duration: 174.53 ms

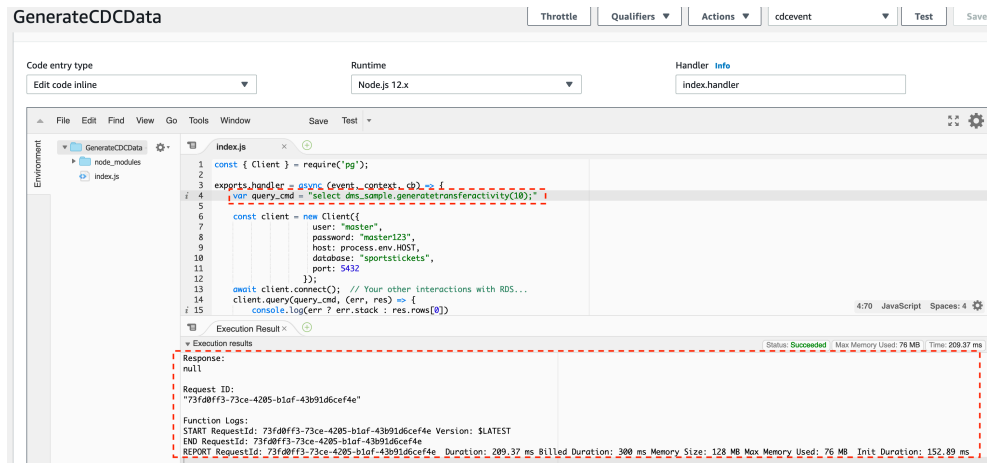
- Once you've sold some tickets you can run the generateTransferActivity procedure. The following will transfer tickets from the owner to another person. The whole "batch" of tickets purchased is transferred 80% of the time and 20% of the time an individual ticket is transferred.

Run this query next in the lambda function:

select dms_sample.generatetransferactivity(10);

Click on **Save** and then click on **Test** to run the function.

Database Migration Services Instructor Environment for the Lab



Note:

When enabling CDC functionality in DMS, only one DMS instance/task should activate "Ongoing replication" to avoid conflicts.

When replicating to multiple targets, the processing to fan out the updates should begin with the Amazon S3 bucket, that is the target of the DMS task responsible for Ongoing replication. The process should not begin with the source database, as only one CDC process should be tracking and setting the last committed transaction that was replicated.