



# **Amazon Web Services Data Engineering Immersion Day**

---

Prelab1. Instructor Environment Setup

*August 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 relational 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 prelab, you will complete the following tasks:

1. Create a Postgres RDS source database environment.
2. Install the source database.

Once the full data replication is finished by DMS, go to the next step if the CDC lab is required:

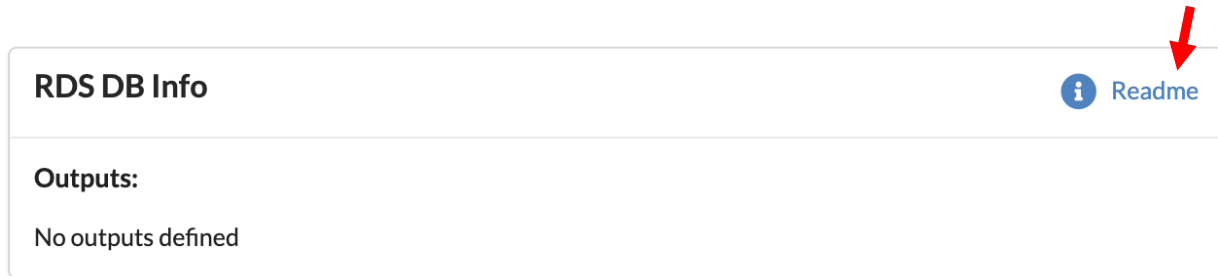
3. Execute Lambda function to generate CDC data at the source database environment, to demonstrate CDC (Change Data Capture) replication within DMS.

Relevant information about this prelab:

- CloudFormation execution time: 15 minutes
- Source DB installation time: 20 mins

## Prelab1. Database Migration Services Instructor Environment Setup

In an instructor-led AWS event, participants can get the Postgres RDS database detail from an event dashboard. (The instructor is required to update the database information before each event)

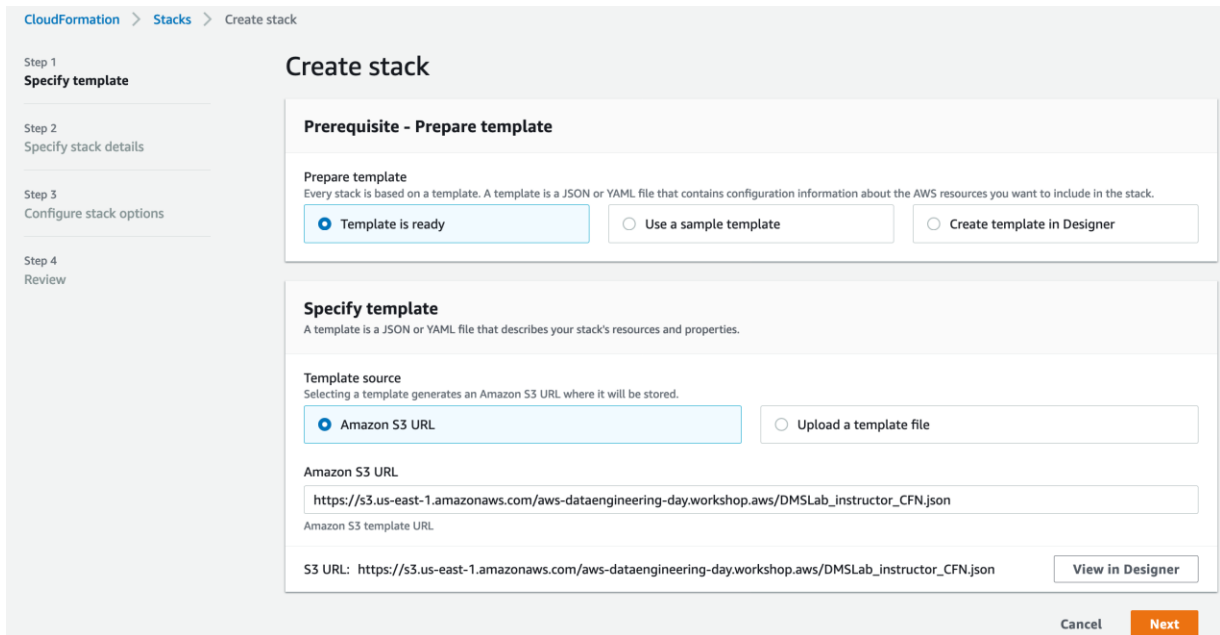


The instructor setup is also available in our online workshop: <https://aws-dataengineering-day.workshop.aws/en/400/410-pre-lab-1.html>

## 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, for data migration to Amazon S3 data lake.

1. Launch the instructor [CloudFormation](#) stack. Make sure the region is us-east-1 (Virginia).



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

## Prelab1. Database Migration Services Instructor Environment Setup

**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

- Click on **Next**.
- 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

- 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.

**Warning: make sure the Postgres database is fully populated before proceed to the DMS lab. It takes additional 20 minutes to finish, after the CloudFormation setup is completed.**

You can see all resources listed below:

## Prelab1. Database Migration Services Instructor Environment Setup

**dmslab-instructor** Delete Update Stack actions ▼ Create stack ▼

Stack info | Events | **Resources** | Outputs | Parameters | Template | Change sets

**Resources (27)** ↻

🔍 Search resources ⚙️

Logical ID	Physical ID	Type	Status	Status reason
EC2SubNet	subnet-0b46150fc43e400bc <a href="#">↗</a>	AWS::EC2::Subnet	✔️ CREATE_COMPLETE	-
GenerateCDCData	<a href="#">GenerateCDCData</a> <a href="#">↗</a>	AWS::Lambda::Function	✔️ CREATE_COMPLETE	-
LambdaExecutionRole	dmslab-instructor-LambdaExecutionRole-1QS0V5OCLPR09 <a href="#">↗</a>	AWS::IAM::Role	✔️ CREATE_COMPLETE	-
RDSSubNet	subnet-0477e0e0071e80331 <a href="#">↗</a>	AWS::EC2::Subnet	✔️ CREATE_COMPLETE	-
RDSSubNet2	subnet-00dea43618c4868d8 <a href="#">↗</a>	AWS::EC2::Subnet	✔️ CREATE_COMPLETE	-
dbpgdataengdmsgroup	dmslab-instructor-dbpgdataengdmsgroup-1pbby1ntnpgq <a href="#">↗</a>	AWS::RDS::DBParameterGroup	✔️ CREATE_COMPLETE	-
dbsgdefault	dmslab-instructor-dbsgdefault-1p5usgck1gq0a	AWS::RDS::DBSecurityGroup	✔️ CREATE_COMPLETE	-
dbsubnetdefaultdmsinstructorvpc	dmslab-instructor-dbsubnetdefaultdmsinstructorvpc-13e6pv5p7lbvy <a href="#">↗</a>	AWS::RDS::DBSubnetGroup	✔️ CREATE_COMPLETE	-

- e. Go to the **Outputs** tab of AWS CloudFormation stack and you can find the Postgres RDS database 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)** ↻

🔍 Search outputs ⚙️

Key	Value	Description	Export name
CDCFunction	arn:aws:lambda:us-east-1:789211807855:function:GenerateCDCData	CDC Function	-
DMSInstanceEndpoint	<span style="border: 2px dashed red;">dmslabinstance.ccla1oozkrry.us-east-1.rds.amazonaws.com</span>	DMS Instance Endpoint	-

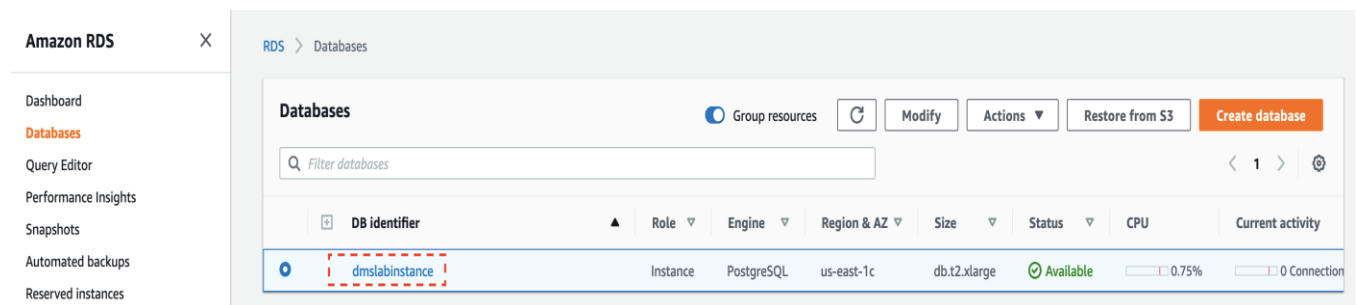
## Changing RDS Security Group

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).

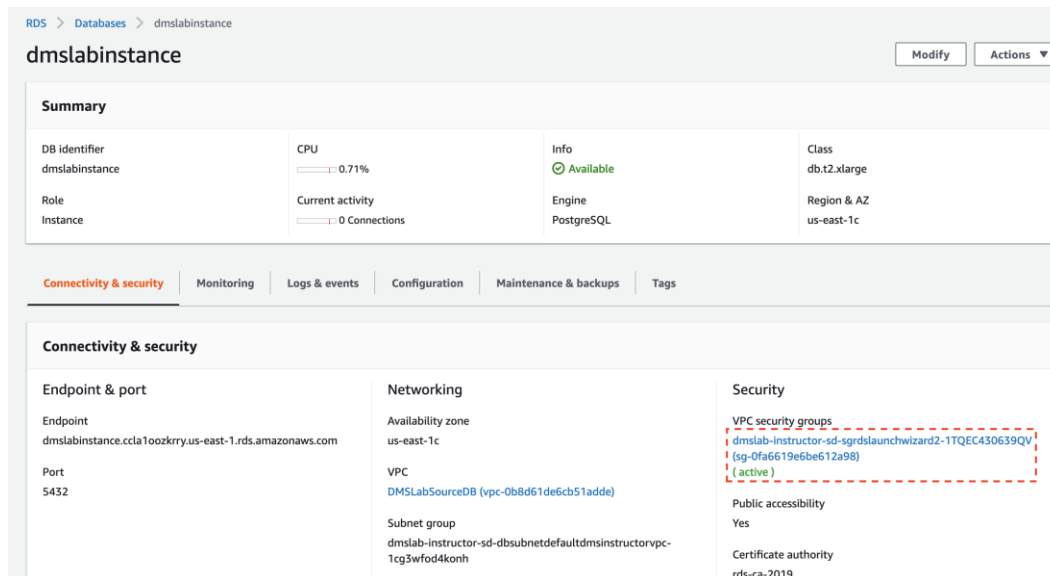
**Warning: It is not best practice to allow ALL CIDR range in your database security group. You should never apply open to all IP CIDR range while working on actual workload.**

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:

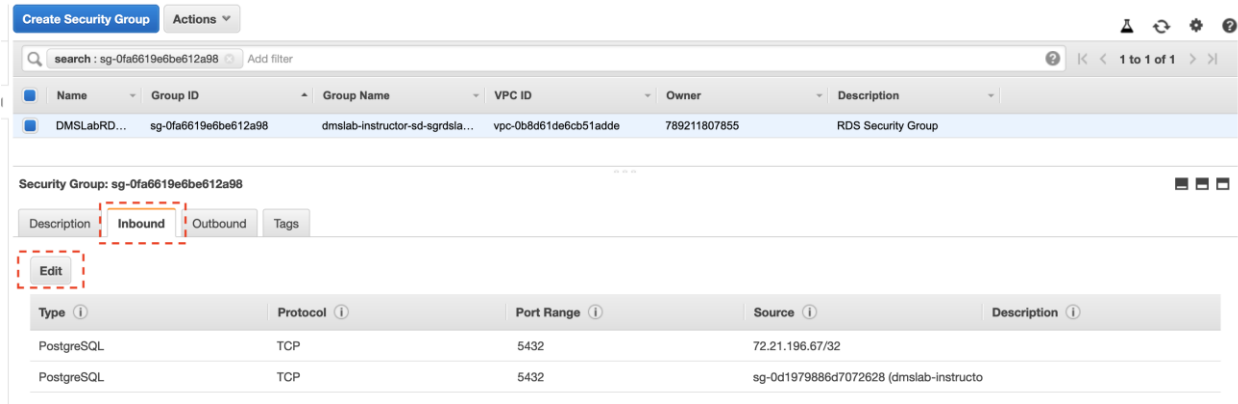


2. Click **VPC security groups** under **Connectivity & security** tab as shown below:

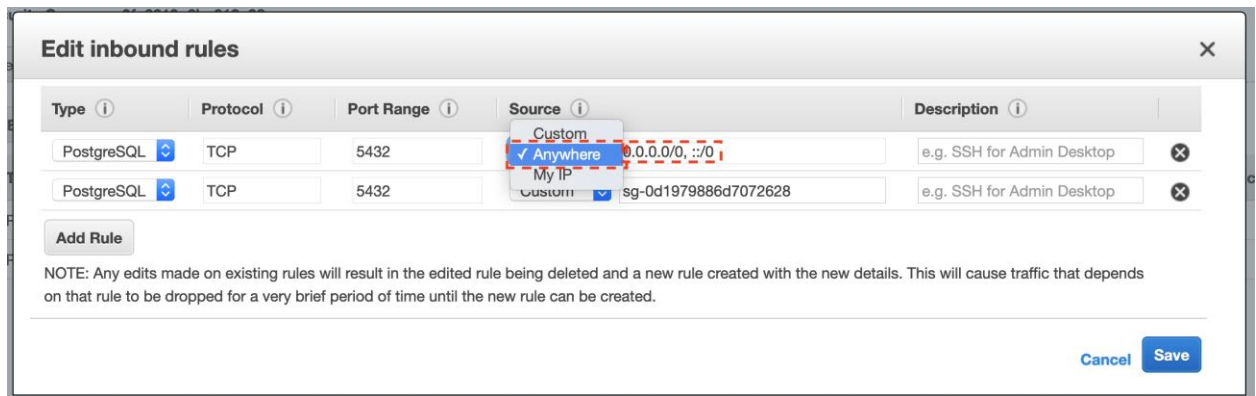


3. In Security group screen, Go to **Inbound** tab and click on **Edit** as shown below

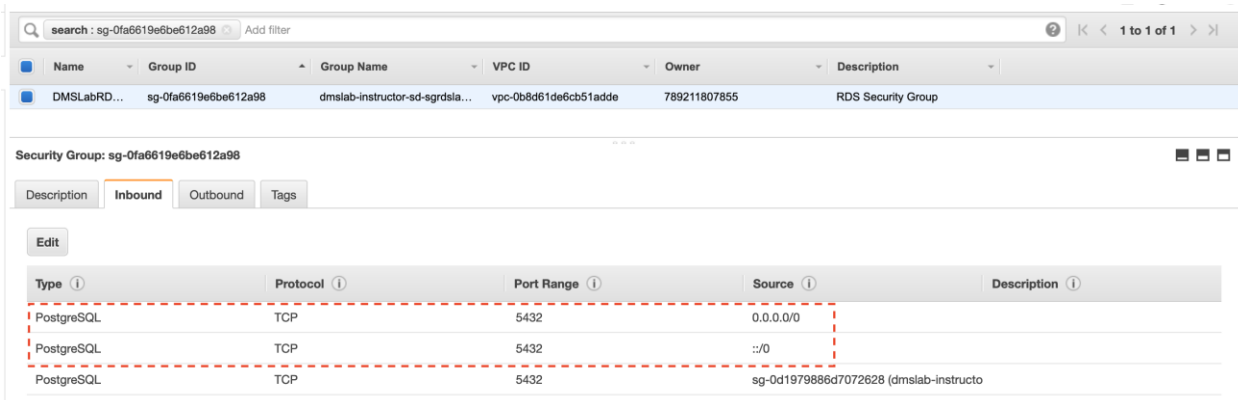
## Prelab1. Database Migration Services Instructor Environment Setup



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



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.



**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 though the documentation to better understand the source database environment. The GitHub repository for aws-database-migration-samples is located here:



## Prelab1. Database Migration Services Instructor Environment Setup

<https://github.com/aws-samples/aws-database-migration-samples/tree/master/PostgreSQL/sampledbs/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;
```

Ensure the following 2 functions exists. If anything is missing, check the solution at **Troubleshooting** section.

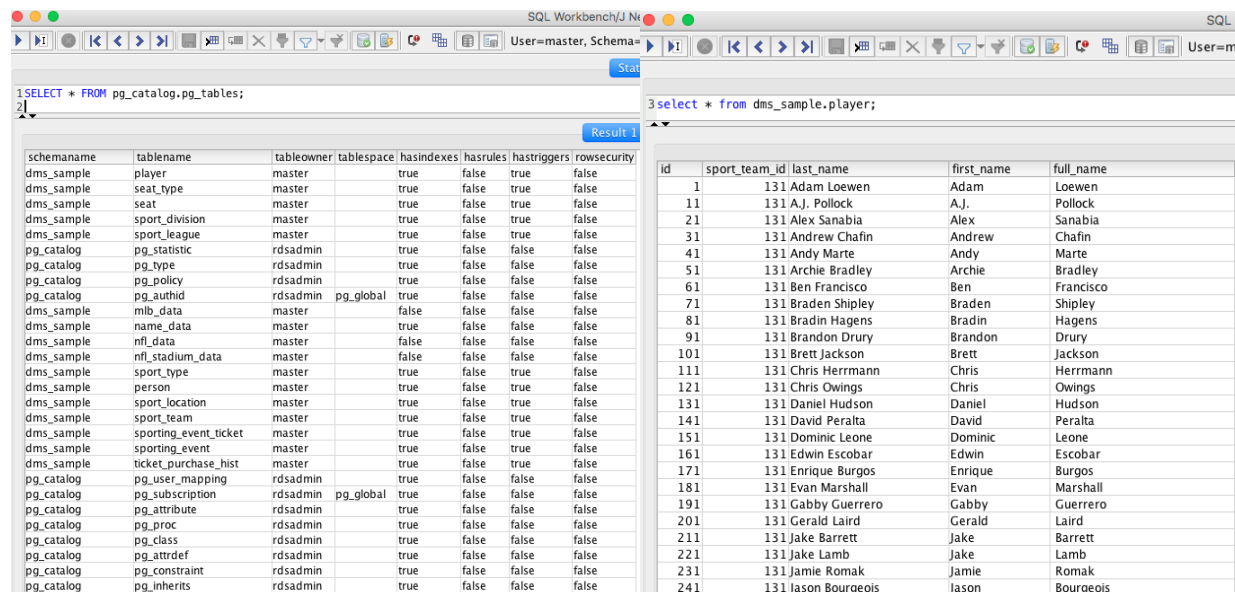
```
SELECT * FROM pg_stat_user_functions  
WHERE funcname in ('generateticketactivity','generatetransferactivity')
```

Use following query to analyze a table

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

For example:

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



The screenshot displays the SQL Workbench interface with two queries executed. The left pane shows the result of the query `SELECT * FROM pg_catalog.pg_tables;`, which lists all tables in the database. The right pane shows the result of the query `select * from dms_sample.player;`, which displays a list of players.

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	rsadmin		true	false	false	false
pg_catalog	pg_type	rsadmin		true	false	false	false
pg_catalog	pg_policy	rsadmin		true	false	false	false
pg_catalog	pg_authid	rsadmin	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	rsadmin		true	false	false	false
pg_catalog	pg_subscription	rsadmin	pg_global	true	false	false	false
pg_catalog	pg_attribute	rsadmin		true	false	false	false
pg_catalog	pg_proc	rsadmin		true	false	false	false
pg_catalog	pg_class	rsadmin		true	false	false	false
pg_catalog	pg_attrdef	rsadmin		true	false	false	false
pg_catalog	pg_constraint	rsadmin		true	false	false	false
pg_catalog	pg_inherits	rsadmin		true	false	false	false

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

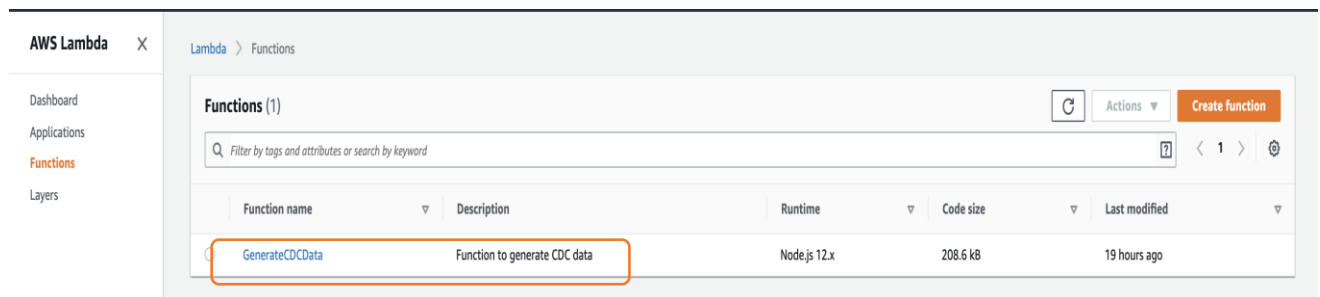
**Following sections are optional you only need to execute, if you want to show change data capture replication with DMS.**

## Generate and Replicate the CDC Data (Optional)

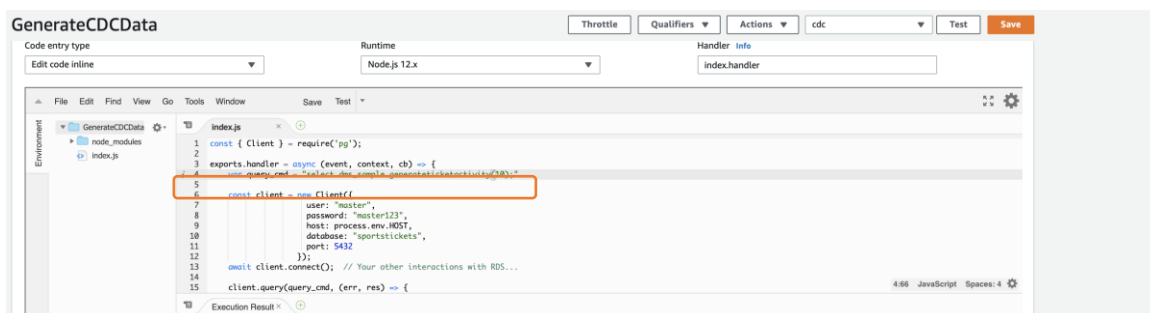
**Warning: This step is not required at your initial lab environment setup.**

Once the full data replication in DMS lab is completed, you can start to generate extra transactions in source database to demonstrate DMS CDC (Change Data Capture) functionality.

Navigate to Lambda console and you will see a pre-built Lambda function named **"GenerateCDCData"**.



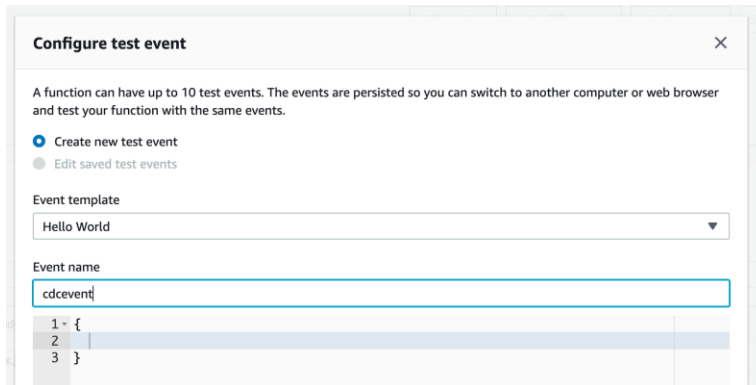
1. Click on the function and scroll down. You will see the code for this function. Copy the below query and paste it in the placeholder (value) of this code line:  
" var query\_cmd= "<insert-SQL-query-here>" "
2. Run this query first: **select dms\_sample.generateticketactivity(10);**



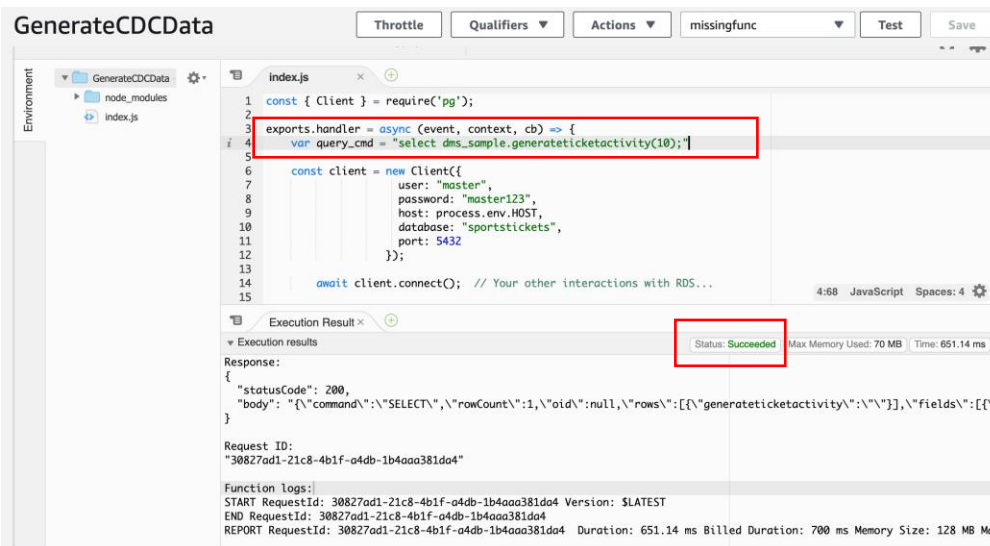
This query will generate 10 ticket sales in batches of 1-6 tickets to randomly selected people for a random price (within a range.) A record of each transaction is recorded in the **ticket\_purchase\_hist** table.

## Prelab1. Database Migration Services Instructor Environment Setup

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



You will see no error in lambda log



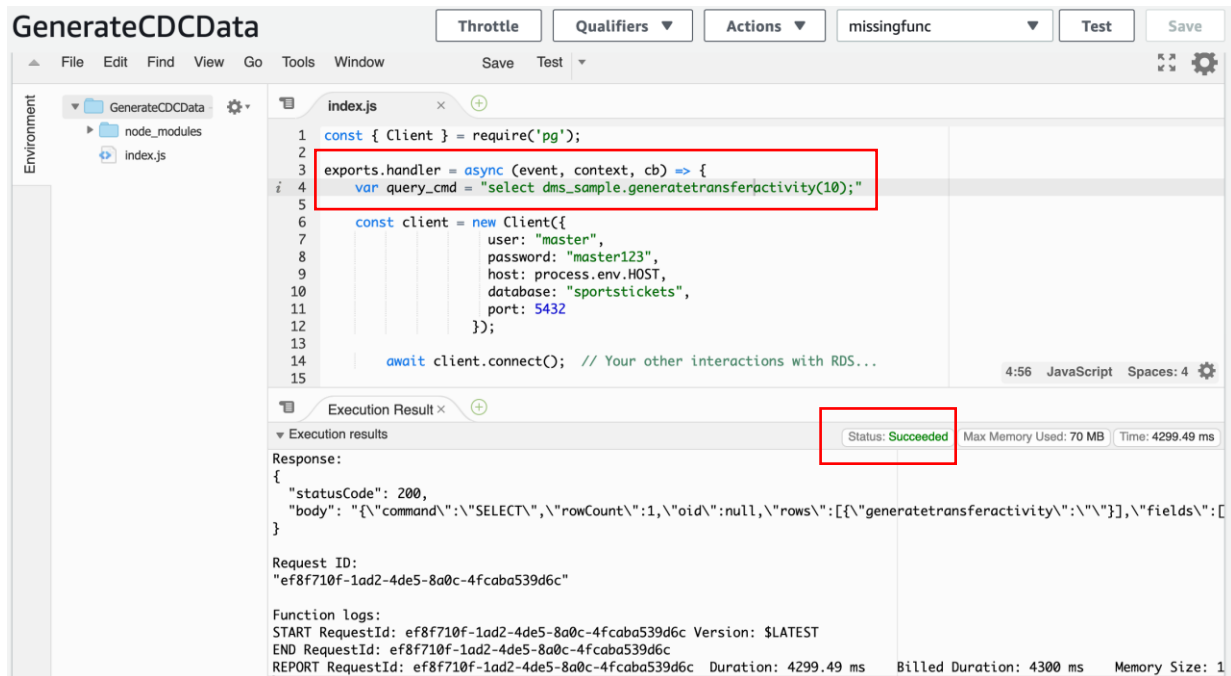
- Once you've sold some tickets you can run the generateTransferActivity procedure. The following will transfer tickets from the owner to another person. The whole "batch" of tickets purchased is transferred 80% of the time and 20% of the time an individual ticket is transferred.

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.

## Prelab1. Database Migration Services Instructor Environment Setup



### 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.

## Troubleshooting

1. Failed to run Lambda function 'GenerateCDCData'.

## Prelab1. Database Migration Services Instructor Environment Setup

### GenerateCDCData

ThrottleQualifiers ▼Actions ▼test

This function belongs to an application. [Click here](#) to manage it.

Execution result: failed ([logs](#))

▼ Details

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

```
{
  "errorType": "error",
  "errorMessage": "function dms_sample.generateticketactivity(integer) does not exist",
  "trace": [
    "error: function dms_sample.generateticketactivity(integer) does not exist",
    "    at Parser.parseErrorMessage (/var/task/node_modules/pg-protocol/dist/parser.js:278:15)",
    "    at Parser.handlePacket (/var/task/node_modules/pg-protocol/dist/parser.js:126:29)",
    "    at Parser.parse (/var/task/node_modules/pg-protocol/dist/parser.js:39:38)",
    "    at Socket.<anonymous> (/var/task/node_modules/pg-protocol/dist/index.js:8:42)",
    "    at Socket.emit (events.js:310:20)",
```

### Cause

The source database setup is interrupted. Some database objects, such as the function **generateticketactivity()** is missing.

### Resolution

Go to [EC2 console](#), reboot the instance **DMSLabEC2**. It will reload the DB and create any objects that were missing. Due to the [re-run issue](#), the table `sporting_event_ticket` will be doubled in size at each reboot. You can manually drop the table by the following script before each reboot. Then wait for 20 minutes before checking the missing DB object again.

```
DROP TABLE dms_sample. sporting_event_ticket CASCADE
```

2. RDS source database is **out of storage** space.

## Prelab1. Database Migration Services Instructor Environment Setup

The screenshot shows the Amazon RDS console for the instance 'dmslabinstance'. The 'Summary' section displays the following details:

DB identifier	CPU	Info	Class
dmslabinstance	0.96%	Storage-full	db.t2.xlarge
Role	Current activity	Engine	Region & AZ
Instance	3 Connections	PostgreSQL	us-east-1d

Or you may see 'No Space left on device' error from DMSLabEC2 system log

The screenshot shows the AWS Management Console for the instance 'DMSLabEC2' (ID: i-0b81b43a2993a6fdf). The 'Instance Settings' menu is open, showing options like 'Add/Edit Tags', 'Attach to Auto Scaling Group', and 'Get System Log'. The 'System Log' for the instance is displayed, showing a 'No space left on device' error during a PostgreSQL installation.

```
init 3188: (1 row)
init 3188: CREATE FUNCTION
init 3188: postgresql:install-postgresql.sql:76: ERROR: could not extend file "base/16401
init 3188: HINT: Check the disk space.
init 3188: CONTEXT: SQL statement "WITH ticket_list AS (
init 3188: SELECT
init 3188:   id AS var_v_ticket_id,
init 3188:   seat_level AS var_v_seat_level,
init 3188:   seat_section AS var_v_seat_section,
init 3188:   seat_row AS var_v_seat_row,
init 3188: FROM dms_sample.sporting_event_ticket
init 3188: WHERE sporting_event_id = rand_event_id
init 3188: ORDER BY seat_level NULLS FIRST,
init 3188: LOWER(seat_section) NULLS FIRST,
init 3188: LOWER(seat_row) NULLS FIRST
init 3188: LIMIT tick_quantity
init 3188: )
init 3188: ticket_holder_list AS (
init 3188: UPDATE dms_sample.sporting_event_ticket
init 3188: SET ticketholder_id = rand_person_id
init 3188: FROM ticket_list
init 3188: WHERE id = var_v_ticket_id
init 3188: RETURNING id,
init 3188: ticketholder_id,
init 3188: ticket_price
init 3188: )
init 3188: INSERT INTO dms_sample.ticket_purchase_hist (
init 3188:   sporting_event_ticket_id,
init 3188:   purchased_by_id,
init 3188:   transaction_date_time,
init 3188:   purchase_price)
init 3188: SELECT
init 3188:   id,
init 3188:   ticketholder_id
```

### Cause

Check the knowledge center [here](#)

### Resolution

Increate the RDS instance disk size, as a quick fix.

## Prelab1. Database Migration Services Instructor Environment Setup

RDS > Databases > dmslabinstance

dmslabinstance

Modify Actions ▼

Summary

DB identifier dmslabinstance	CPU <div>0.75%</div>	Info ⊗ Storage-full	Class db.t2.xlarge
Role Instance	Current activity <div>3 Connections</div>	Engine PostgreSQL	Region & AZ us-east-1d

Modify DB Instance: dmslabinstance

Instance specifications

DB engine version

Version number of the database engine to be used for this instance.

PostgreSQL 11.5-R1 ▼

DB instance class

Contains the compute and memory capacity of the DB instance.

db.t2.xlarge — 4 vCPU, 16 GiB RAM ▼

Multi-AZ deployment

Specifies if the DB instance should have a standby deployed in another availability zone.

☐ Yes

☒ No

Storage type

General Purpose (SSD) ▼

Allocated storage

40

 GiB

This instance supports multiple storage ranges between 20 and 65536 GiB. [See all](#)

Scheduling of modifications

When to apply modifications

☐ Apply during the next scheduled maintenance window  

Current maintenance window: sat:05:20 - sun:05:50

☒ Apply immediately  
The modifications in this request and any pending modifications will be asynchronously applied as soon as possible, regardless of the maintenance window setting for this database instance.

⚠

**Potential unexpected downtime**  
If you choose to apply changes immediately, please note that any changes in the pending modifications queue are also applied. If any of the pending modifications require downtime, choosing this option can cause unexpected downtime.

Cancel

Back

Modify DB Instance

14