Amazon Web Services
Data Engineering Immersion Day

Prelab1. Instructor Environment Setup

August 2020

Table of Contents

Limit Instruction: 2
Introduction 2
Create the Instructor Environment 3
Changing RDS Security Group 6
Access Database from SQL Client (Optional) 9
Generate and Replicate the CDC Data (Optional) 10

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 DPUs 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.

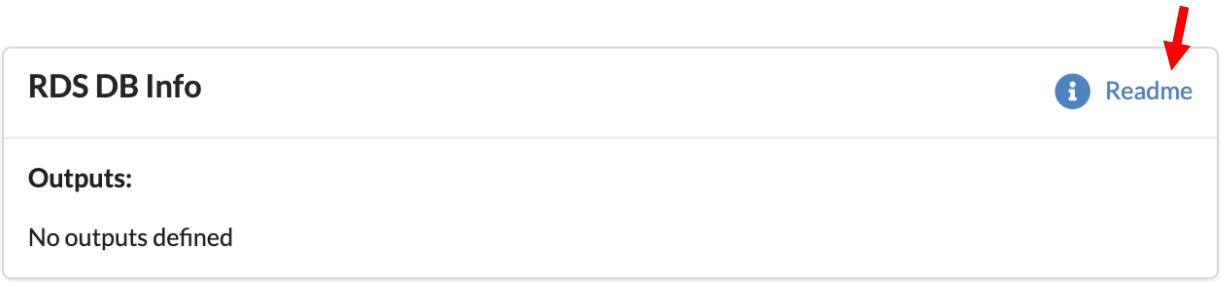
Once the full data replication is finished by DMS, go to the next step if the CDC lab is required:

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

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)

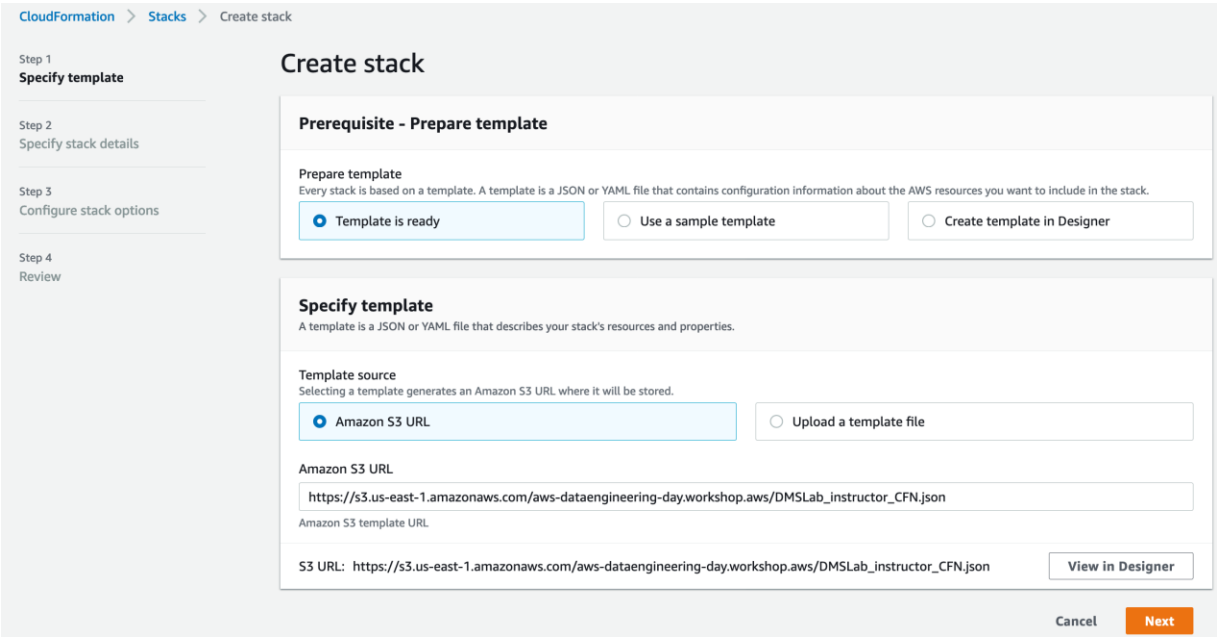


The instructor setup is also available in our online workshop: https://aws-dataengineering-day.workshop.aws/en/400/410-pre-lab-1.html

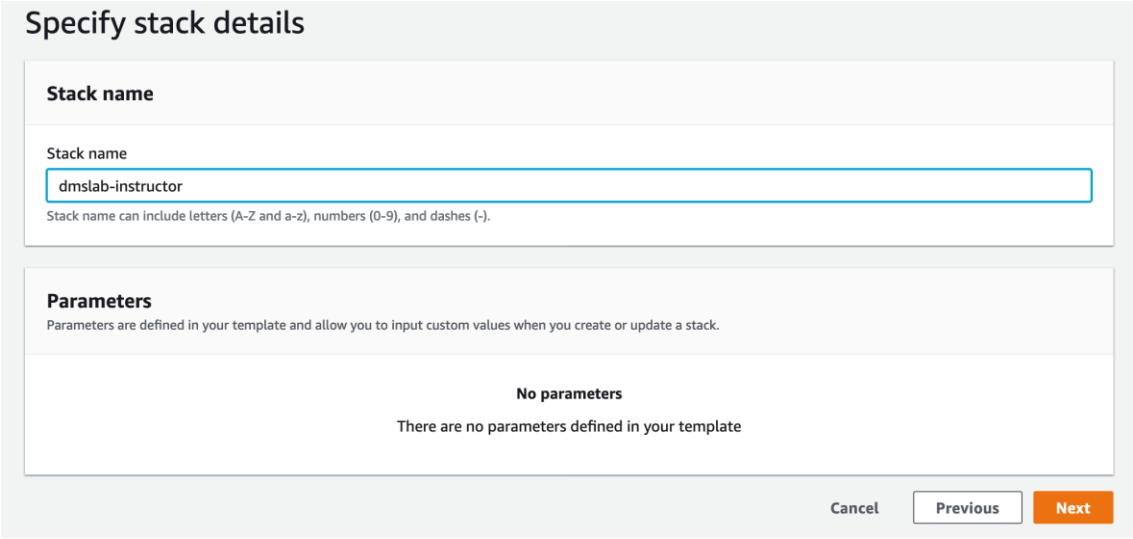
Create the Instructor Environment

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.

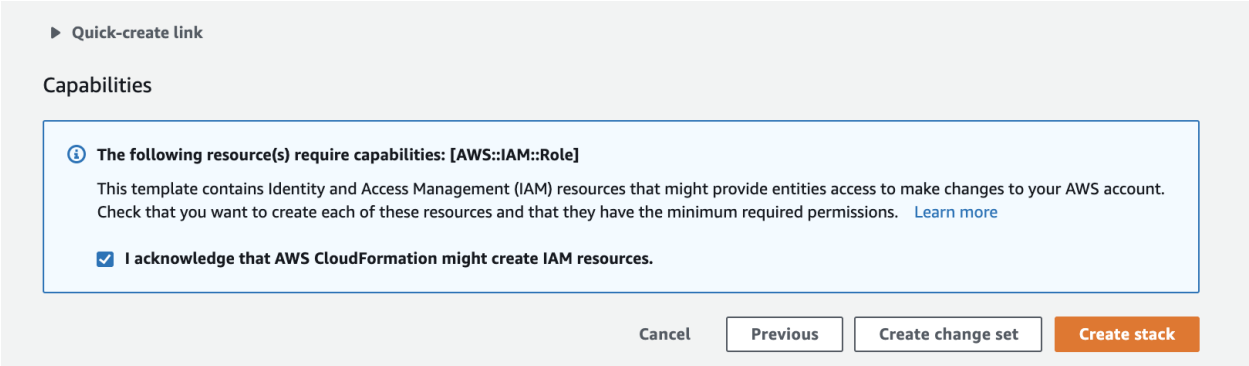
1. Launch the instructor CloudFormation stack. Make sure the region is us-east-1 (Virginia).



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



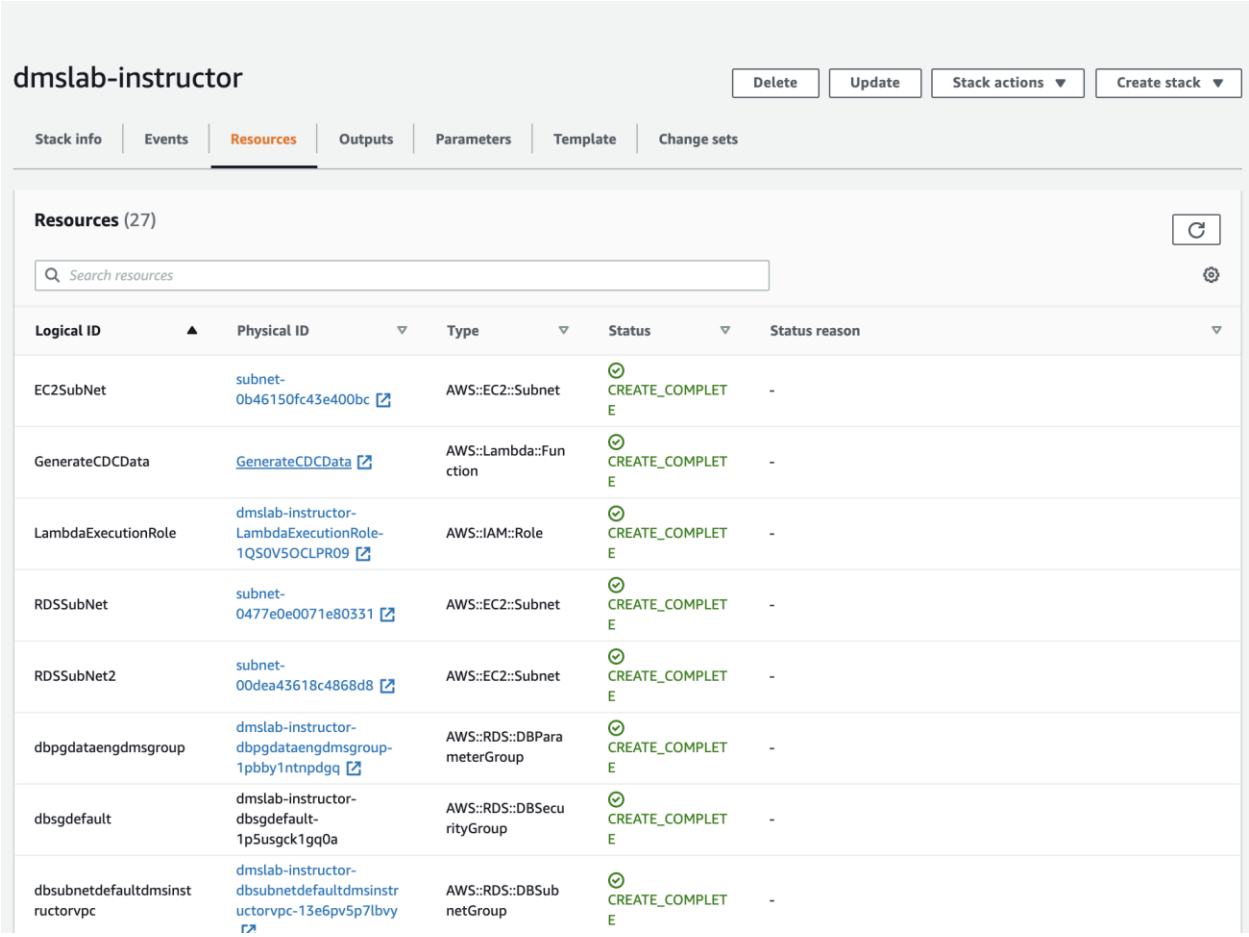
b. Click on **Next**.
c. In review page, review all the details, scroll down and check the box to acknowledge the policy and then click on **Create Stack.**



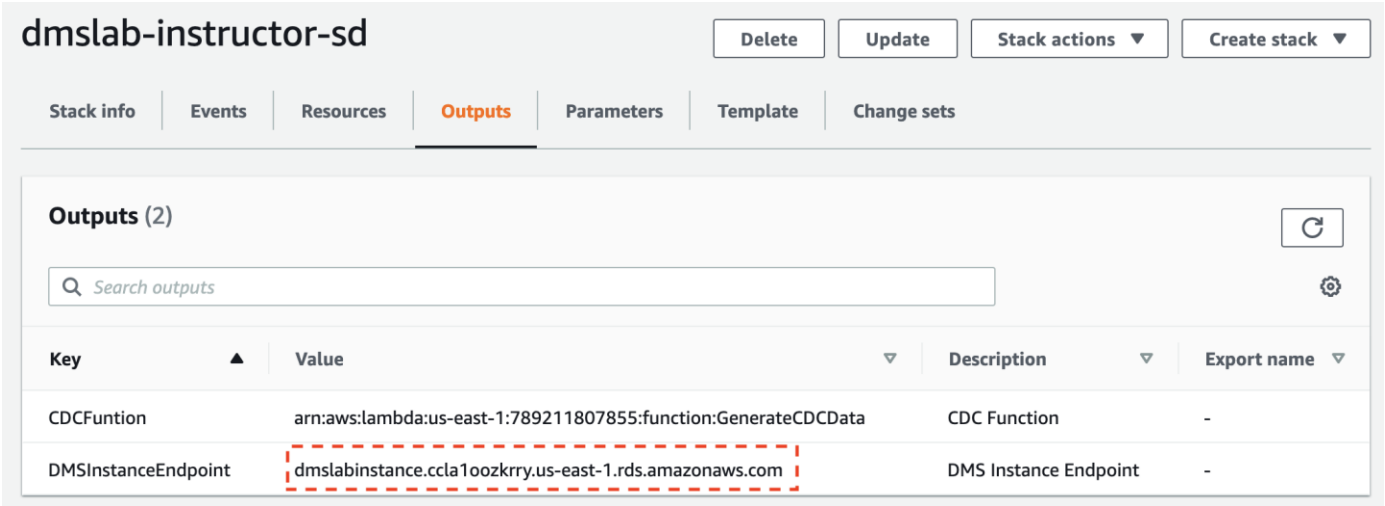
d. Launch the stack. It may take 15 minutes for the stack to launch.
This stack creates a new VPC, Subnets, Security groups, EC2 instance, Route table, Routes, and an RDS Postgres instance.

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:



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



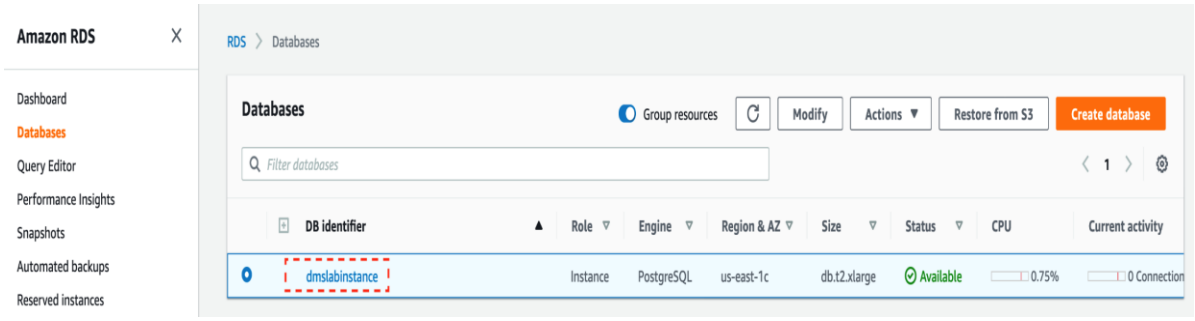
Changing RDS Security Group

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).

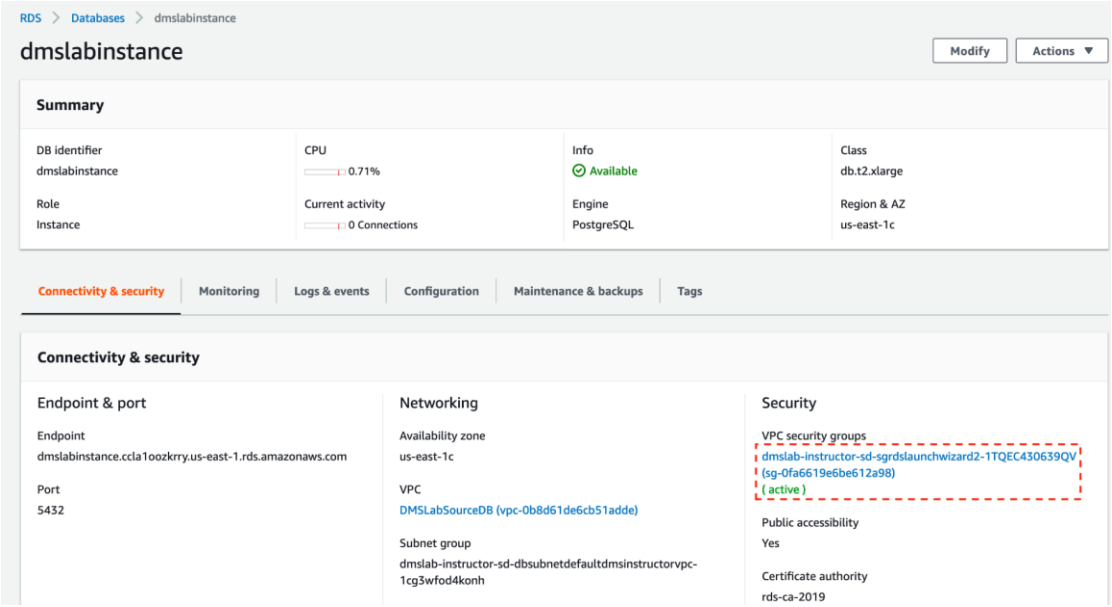
Warning: It is not best practice to allow ALL CIDR range in your database security group. You should never apply open to all IP CIDR range while working on actual workload. If you are in a self-paced workshop, the better secure way is to whitelist an IP address from DMS lab, ie. add an Elastic IP address of a NAT Gateway to the RDS security group.

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

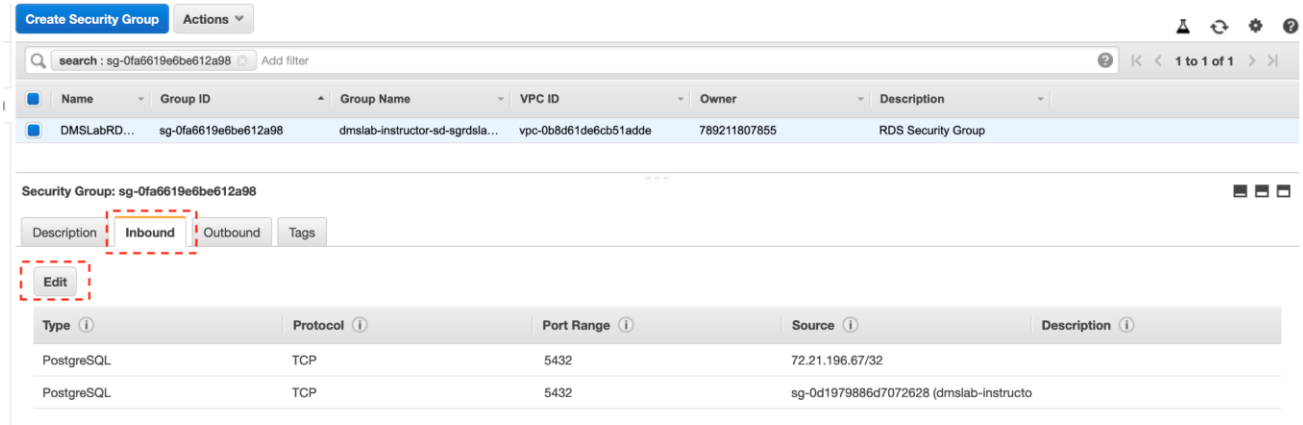
1. Go to the RDS Console and double click on "dmslabinstance" DB identifier as shown below:



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



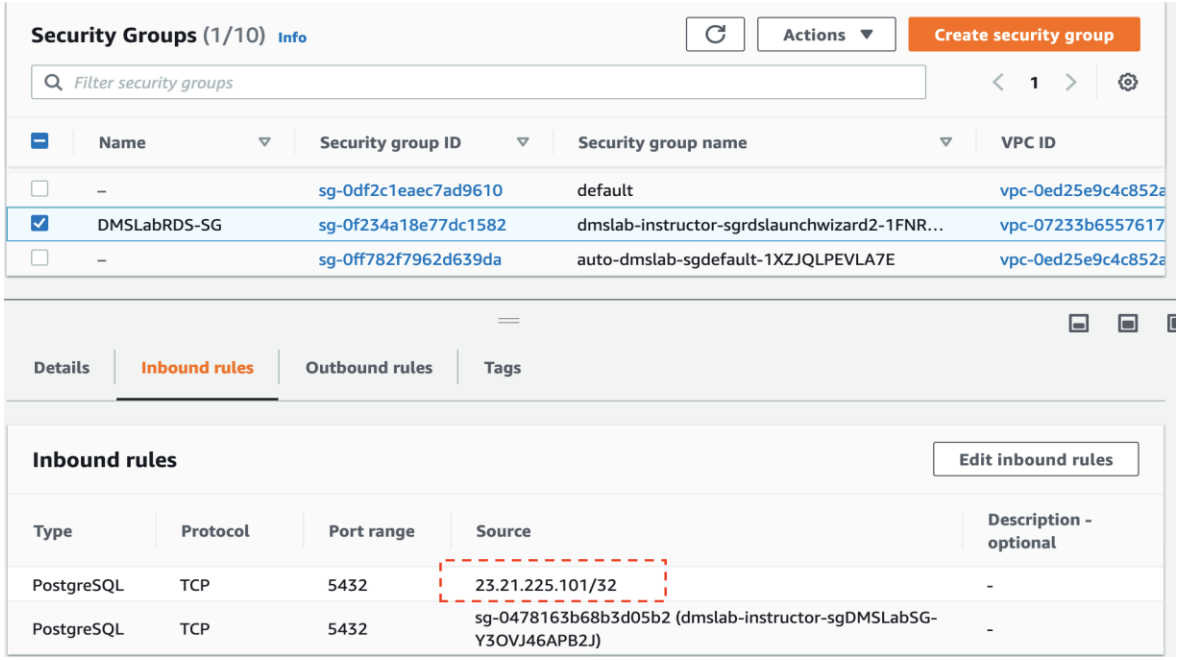
3. In Security group screen, Go to **Inbound** tab and click on **Edit** as shown below



4. Update Inbound rule to "Anywhere" from hard coded value "72.21.196.67/32", as shown in below screen. Make sure to remove the "Anywhere" inbound rule from security group, as soon as you are done with DMS lab.

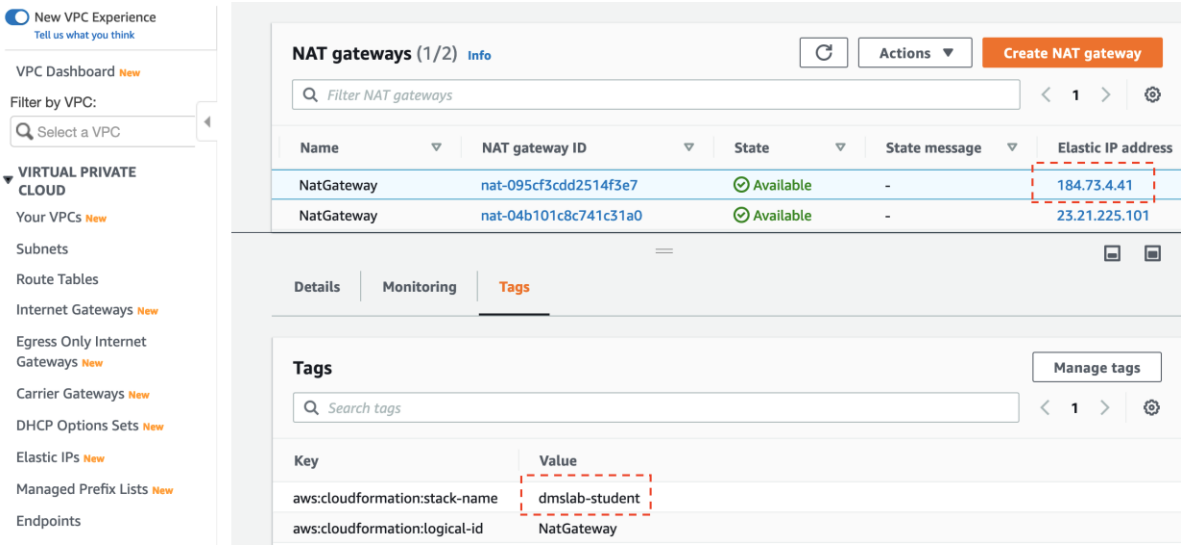
5. If you are in an AWS hosted event, proceed to **step 8**.

6. If you are running both instructor and student labs in a single AWS account, for example in a self-paced environment, replace the "Anywhere" rule by an IP address from student lab instead. In this example, we will allow AutoComplete DMS lab to access the RDS.

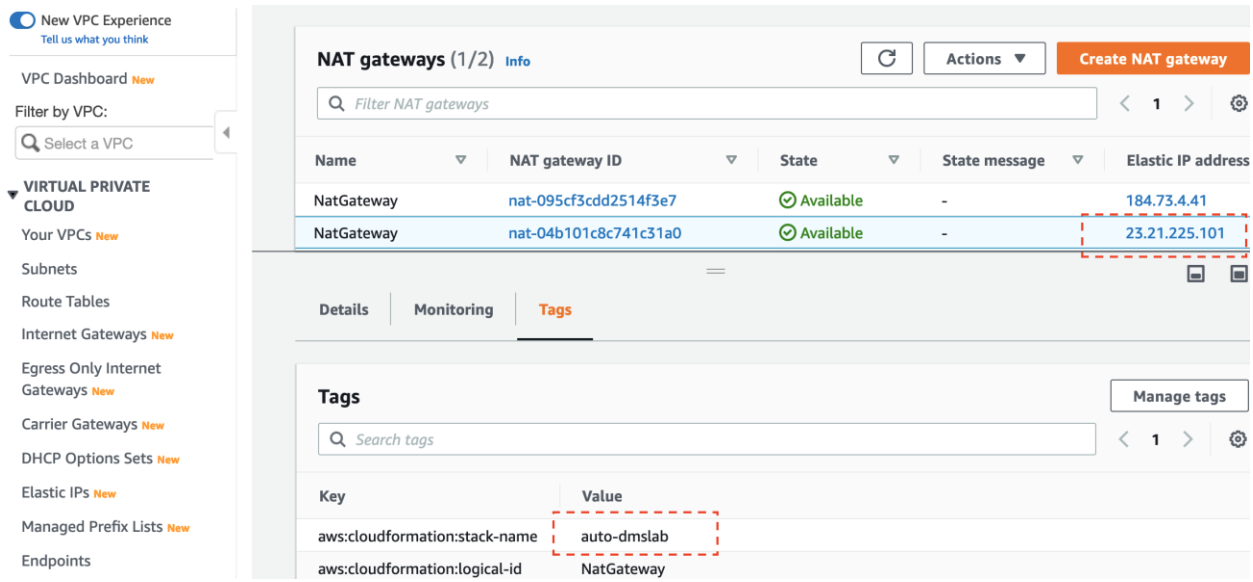


7. Go to [VPC NAT gateways Console](), and look for the IP address you need to add to the RDS security group.

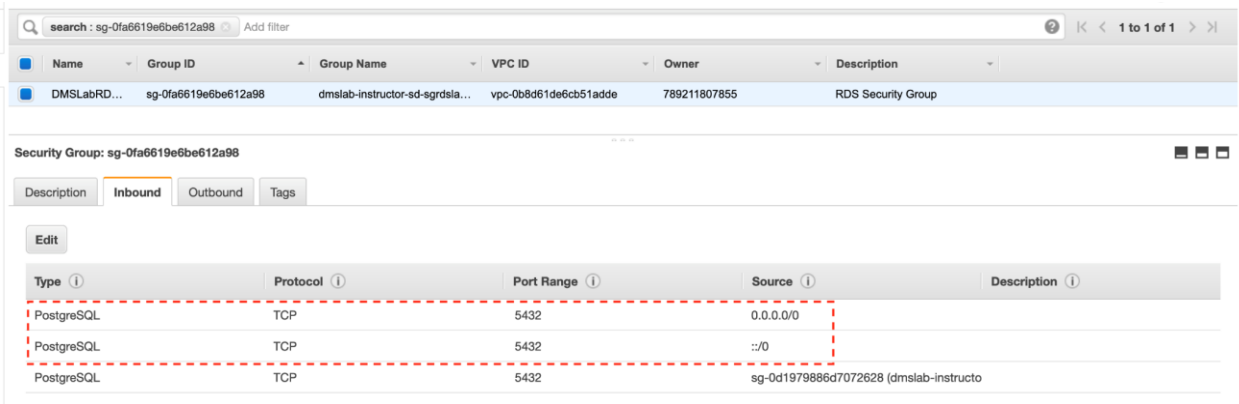
   - If you are running DMS hands-on lab, note down the IP address tagged with "dmslab-student".



   - Or if you are running the AutoComplete DMS Lab, copy the IP address tagged with "auto-dmslab".



8. Click on **Save**. 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 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:

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;
```

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

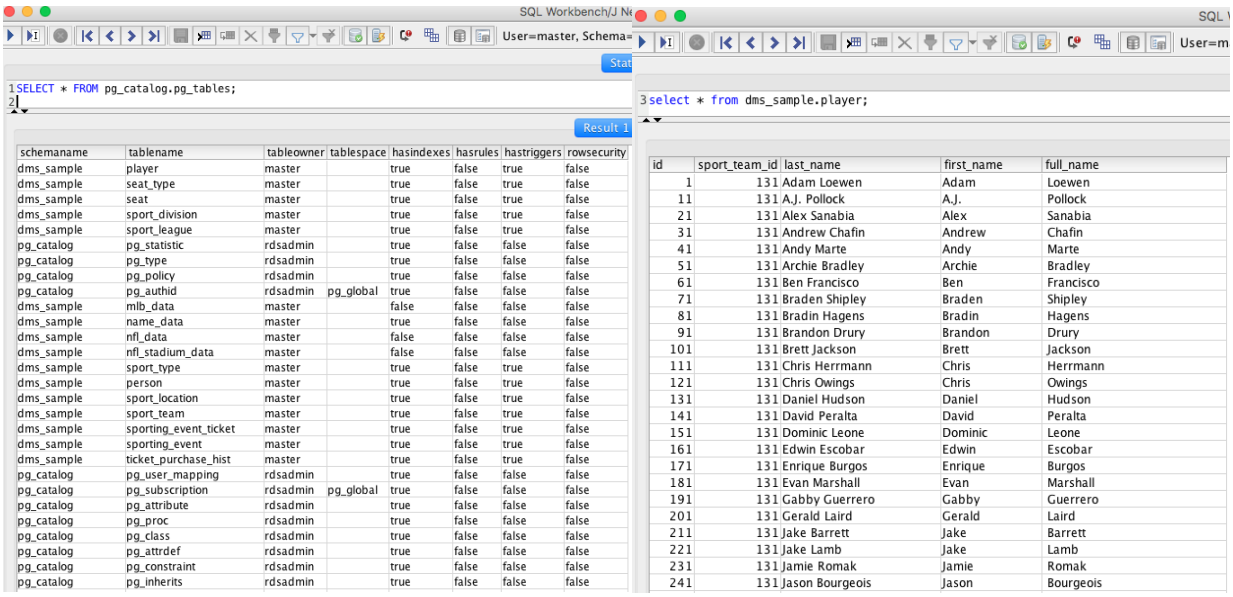
```
SELECT * FROM pg_stat_user_functions
WHERE funcname in ('generateticketactivity','generatetransferactivity')
```

Use following query to analyze a table

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

For example:

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



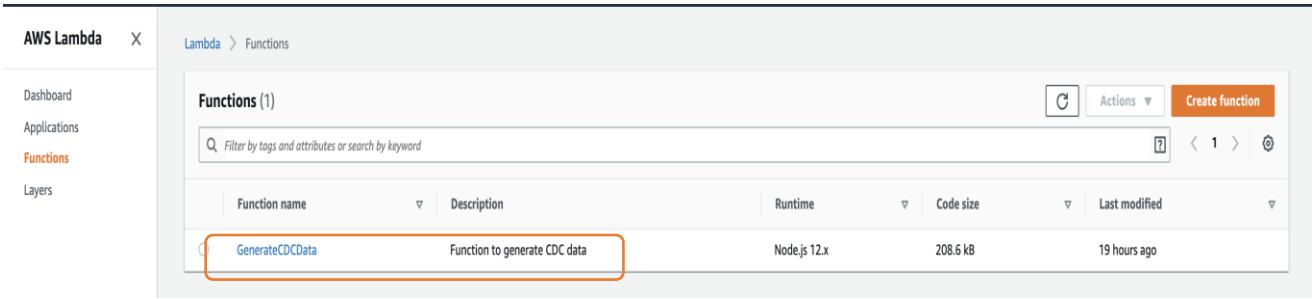
Following sections are optional you only need to execute, if you want to show change data capture replication with DMS.

Generate and Replicate the CDC Data (Optional)

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

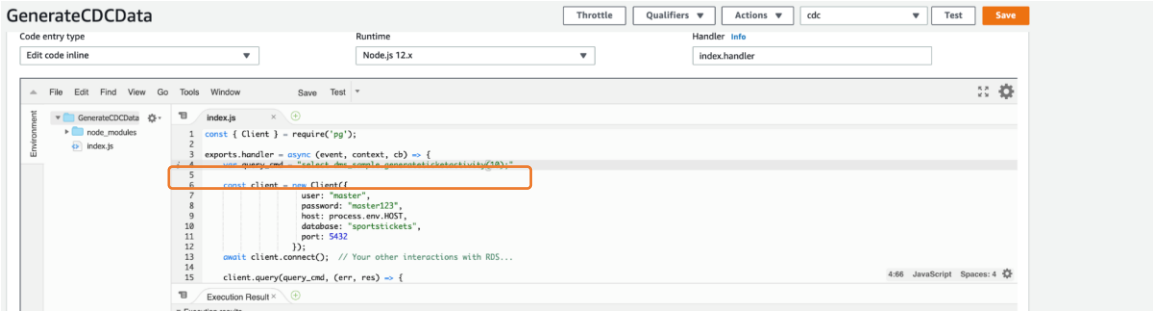
Once the full data replication in DMS lab is completed, you can start to generate extra transactions in source database to demonstrate DMS CDC (Change Data Capture) functionality.

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



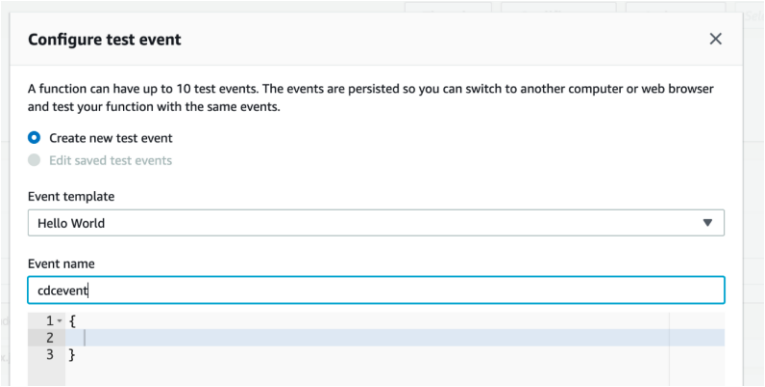
1. Click on the function and scroll down. You will see the code for this function. Copy the below query and paste it in the placeholder (value) of this code line:
   " var query_cmd= "<insert-SQL-query-here>" "

2. Run this query first: **select dms_sample.generateticketactivity(10);**

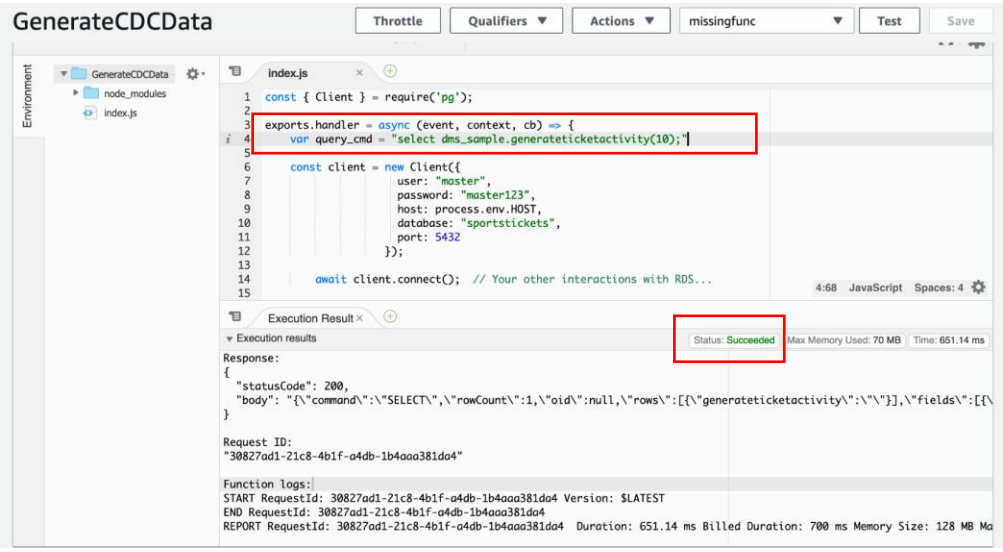


This query will generate 10 ticket sales in batches of 1-6 tickets to randomly selected people for a random price (within a range.) A record of each transaction is recorded in the **ticket_purchase_hist** table.

3. 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

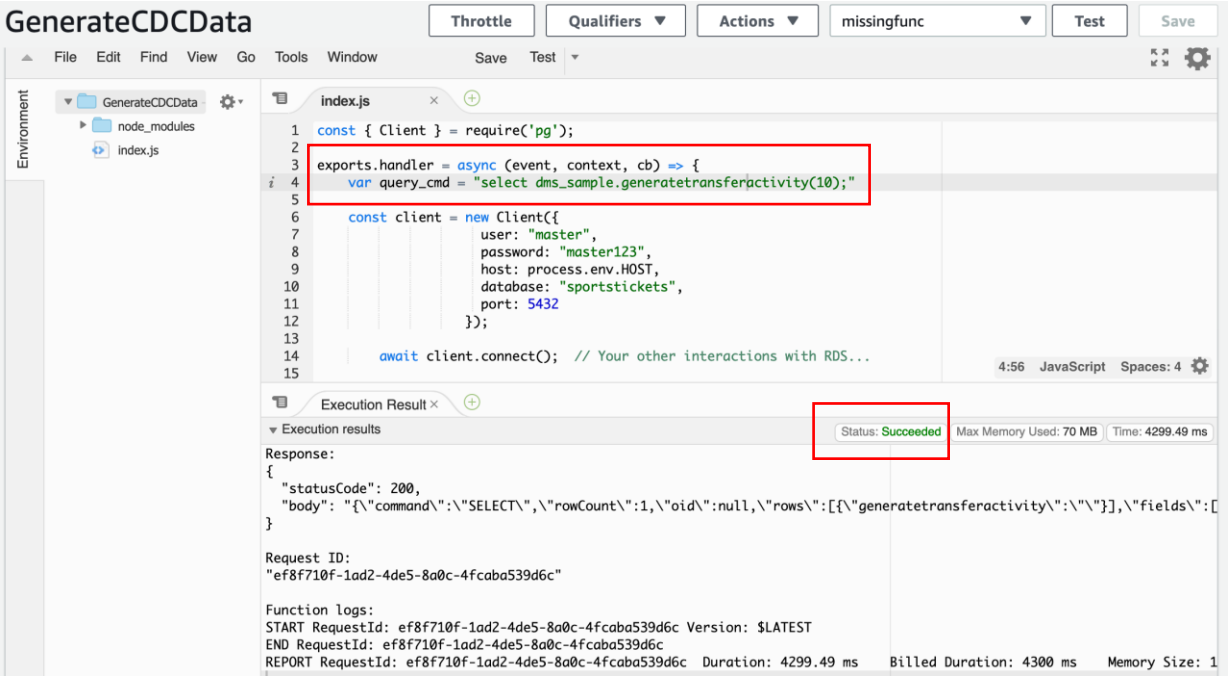


4. Once you've sold some tickets you can run the generateTransferActivity procedure. The following will transfer tickets from the owner to another person. The whole "batch" of tickets purchased is transferred 80% of the time and 20% of the time an individual ticket is transferred.

   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.



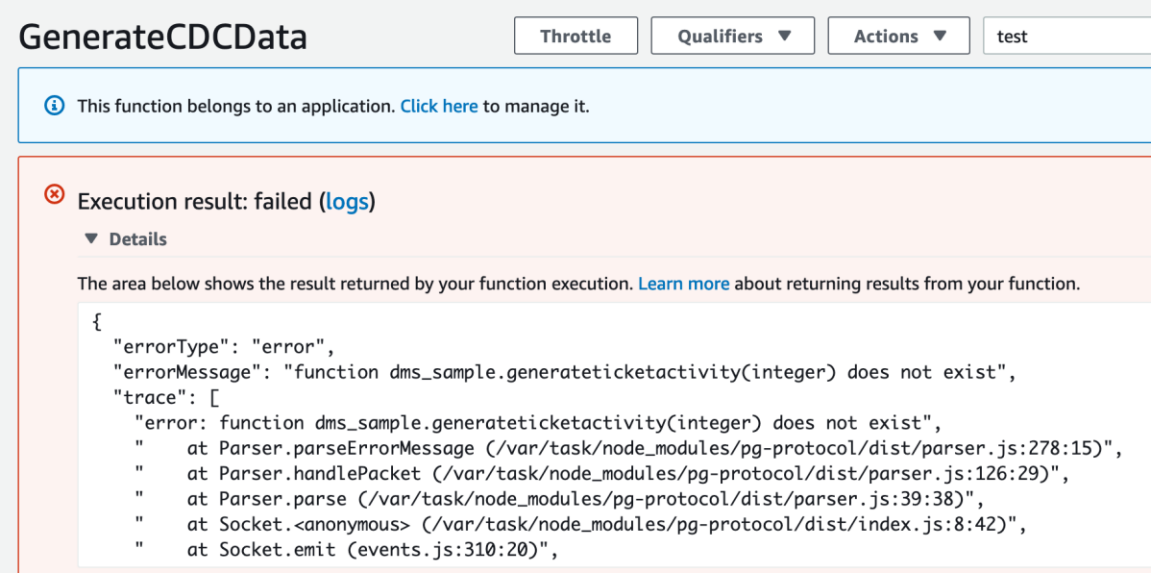
Note:

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**'.



Cause

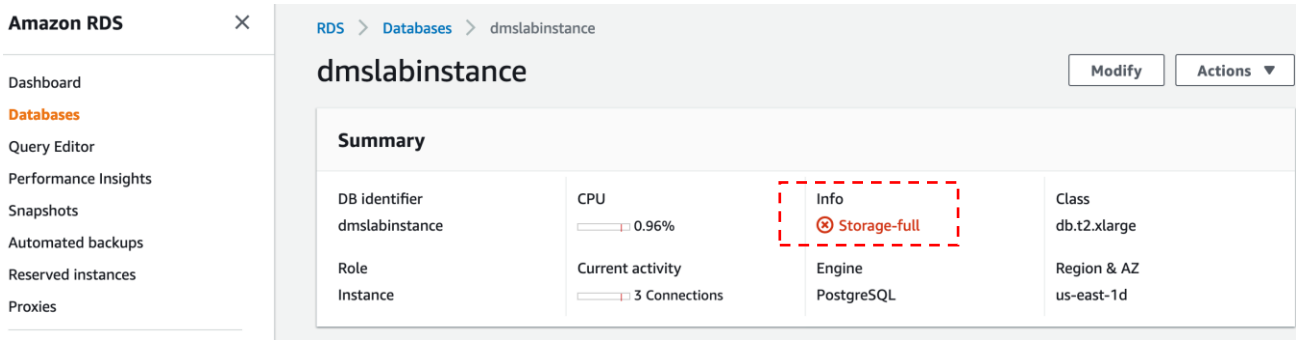
The source database setup is interrupted. Some database objects, such as the function **generateticketactivity()** is missing.

Resolution

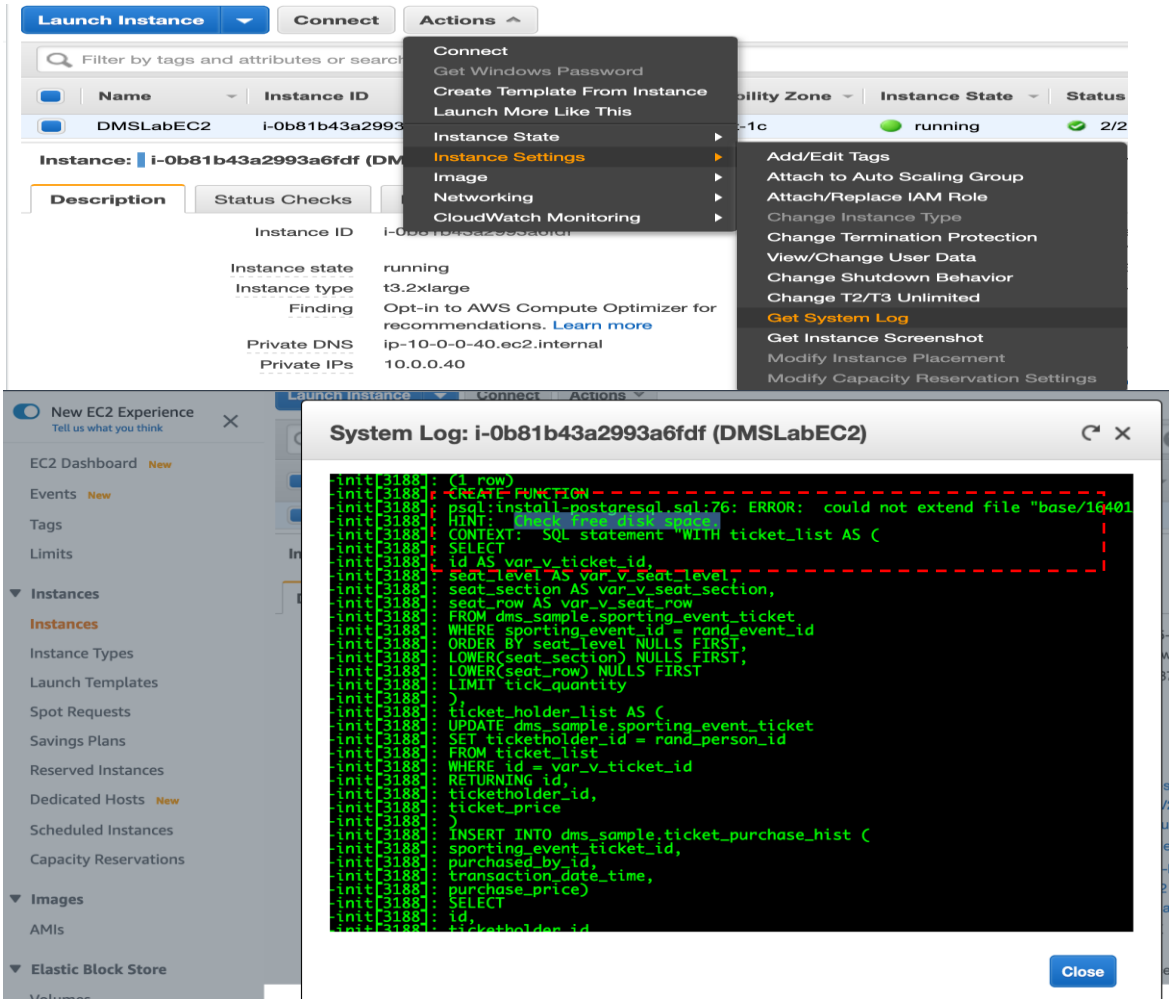
Go to EC2 console, reboot the instance **DMSLabEC2**. It will reload the DB and create any objects that were missing. Due to the re-run issue, the table sporting_event_ticket will be doubled in size at each reboot. You can manually drop the table by the following script before each reboot. Then wait 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.



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

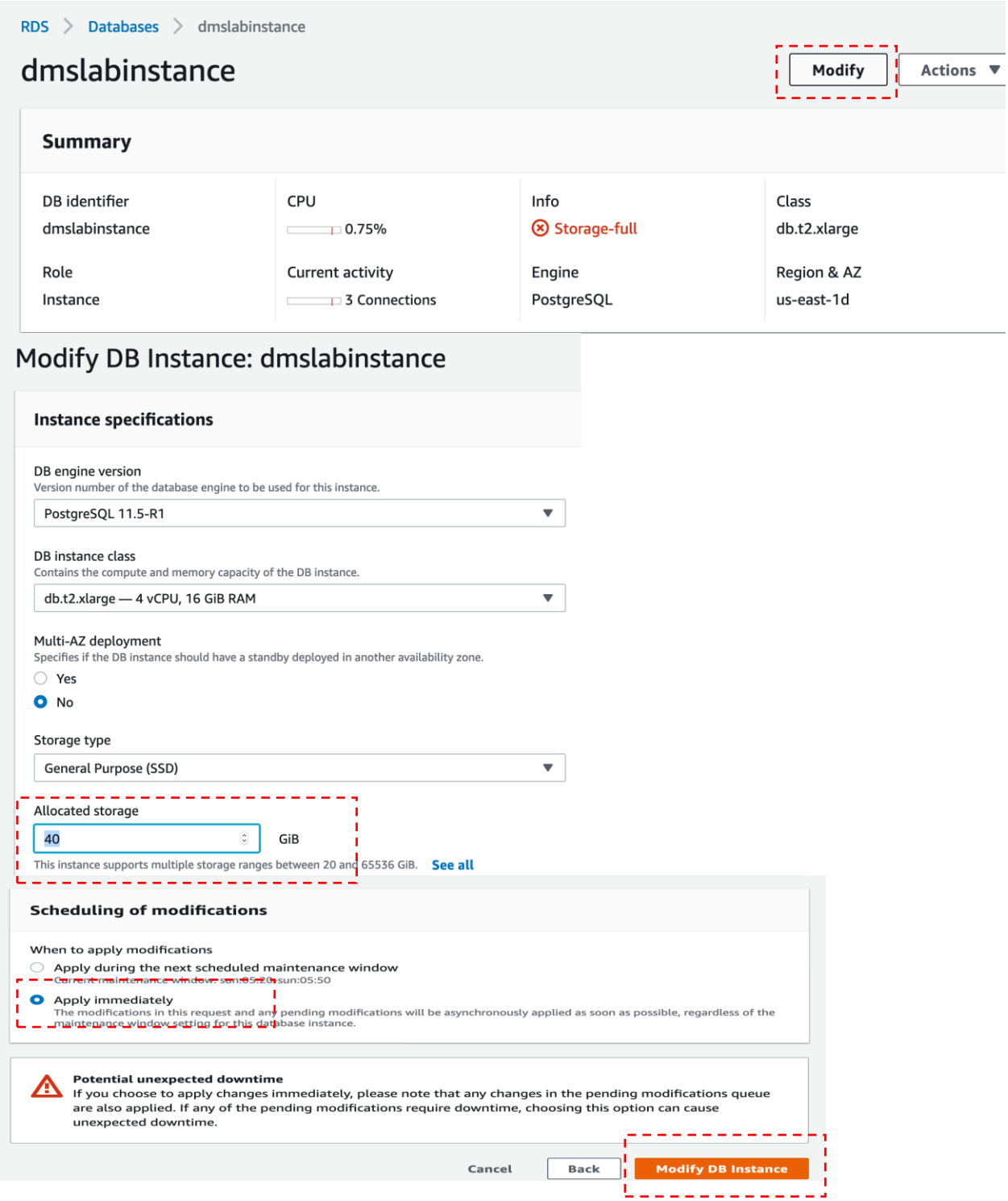


Cause

Check the knowledge center here

Resolution

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


RDS > Databases > dmslabinstance
dmslabinstance
Modify
Actions
Summary
DB identifier
dmslabinstance
CPU
0.75%
Info
Storage-full
Class
db.t2.xlarge
Role
Instance
Current activity
3 Connections
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
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