PROJECT – 2

Designing an Automatic Data Collection and Storage System with AWS Lambda and Slack Integration for Server Availability Monitoring and Slack Notification

Problem Statement:

You are tasked with creating an AWS Lambda function that will periodically fetch data from an [API](http://api.open-notify.org/iss-now.json) and store it in an Amazon RDS instance. The function should be triggered by an Amazon CloudWatch Event that occurs every 15 seconds.

Imported Libraries for this project :

import psycopg2

import requests

import json

Steps:

STEP 1:

To extract the data from a API we have done using below steps :

* 1.I have imported request and json Libraries.
* 2.With the help of get request library I am able to connect to the provided API and get the content.
* 3.The code that I used to get the content is shown below

import psycopg2

import requests

import json

# Make API request to retrieve data

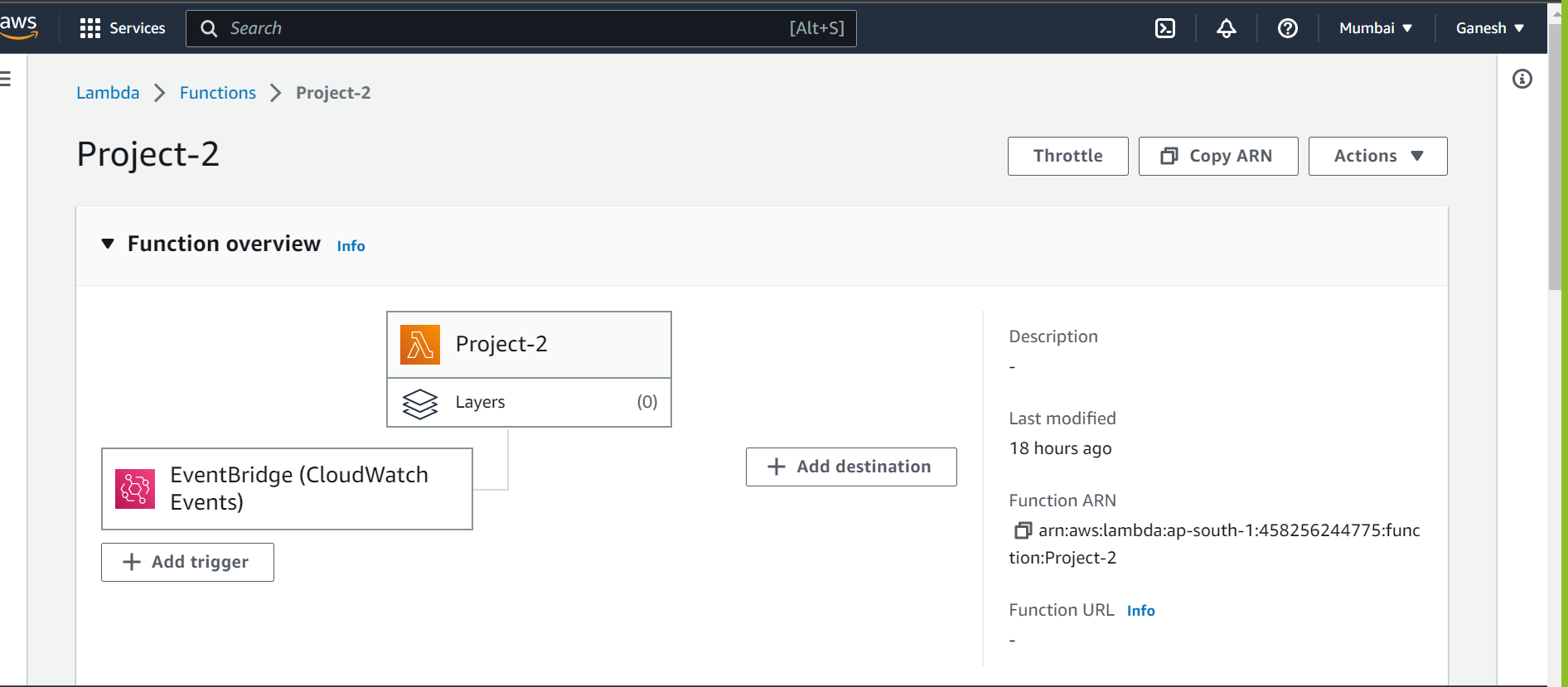
def lambda\_handler(event, context):

  try:

    response = requests.get('http://api.open-notify.org/iss-now.json')

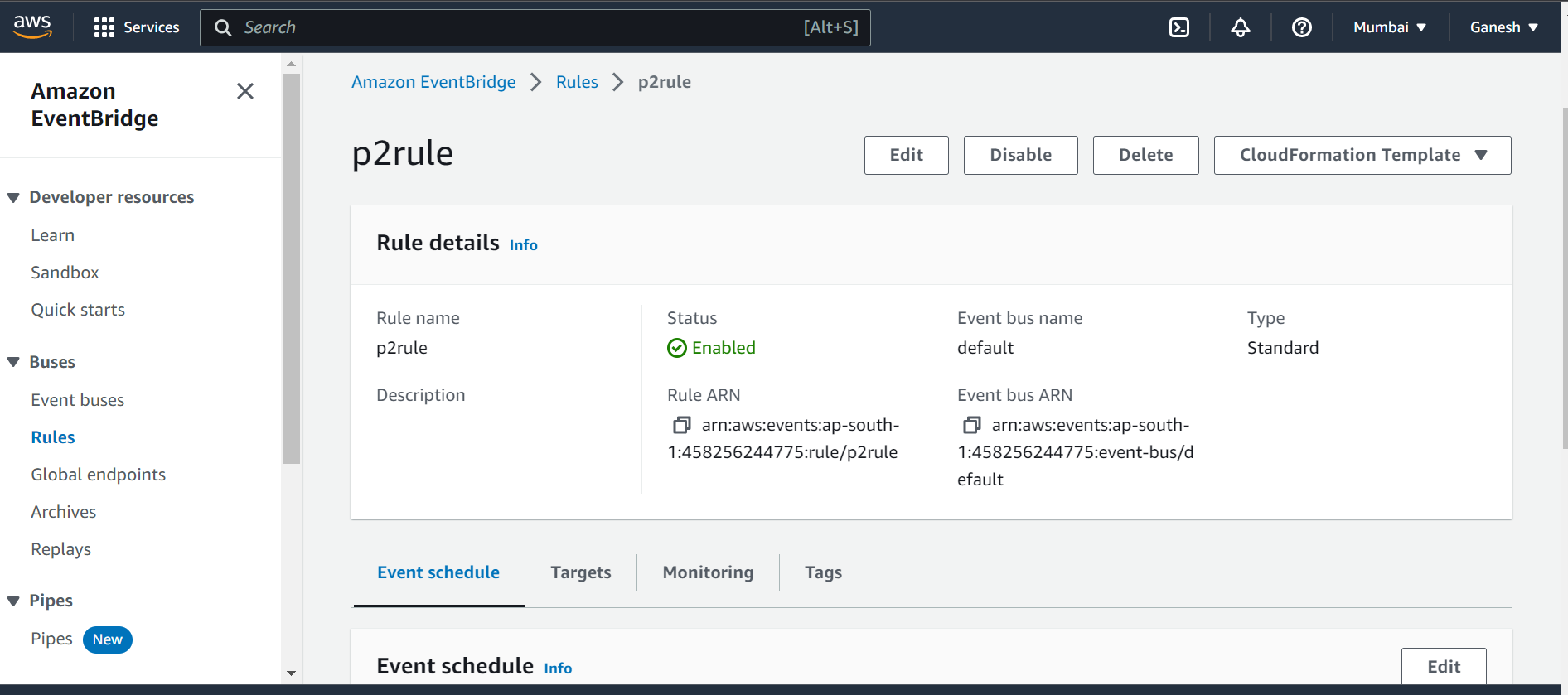
Step 2:

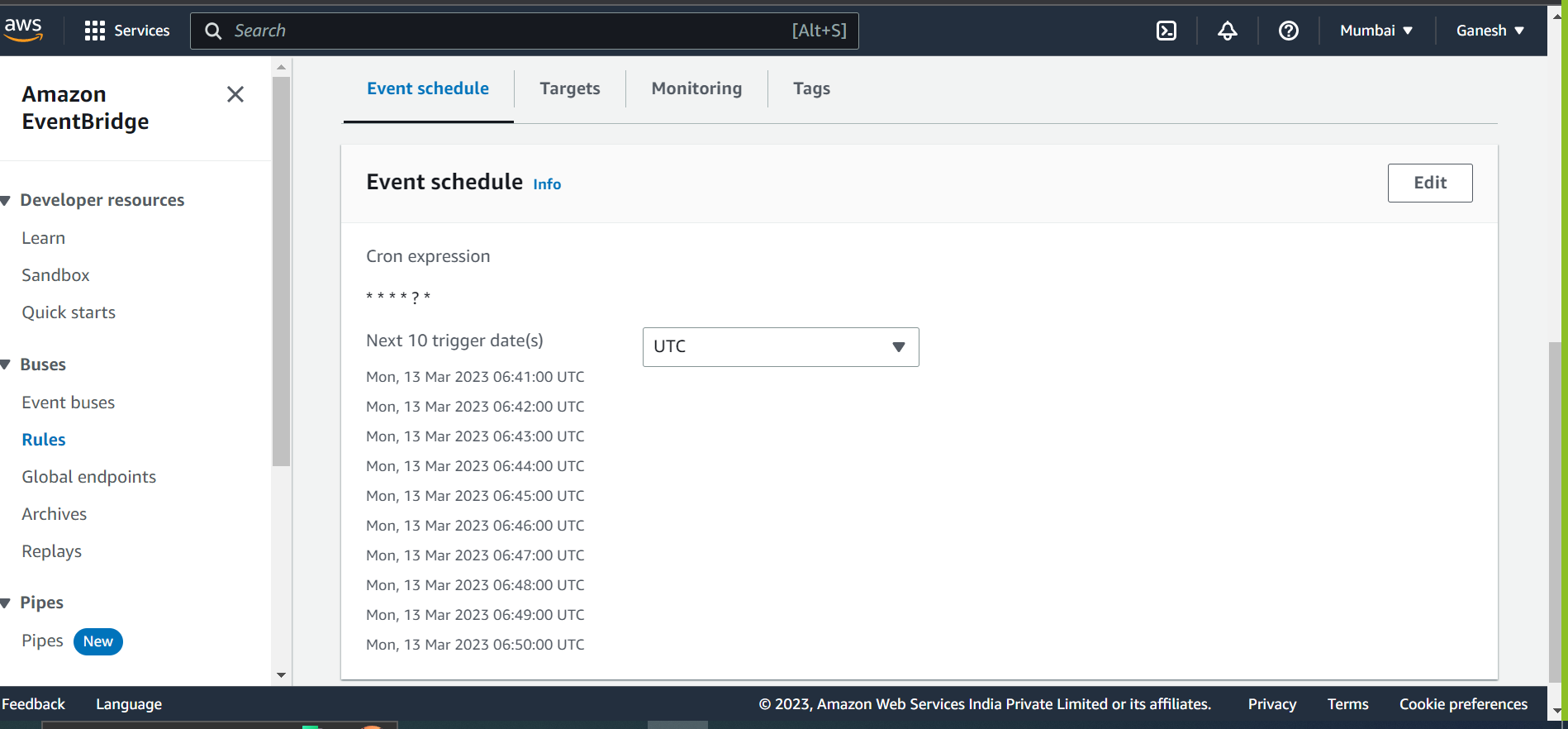
* I have created a Lambda function and also created a Event Bridge(Cloud Watch Events) trigger which will invoke my lambda function as you can see in the screenshot below.



Step 3:

* A rule was created in Amazon Event Bridge and a cron job of one minute was configured in the rule.
* Also configured my lambda function as the target in the rule for my trigger. so that the Event Bridge(Cloud watch Events) will trigger the lambda function for every one minute.
* Please refer the screenshots below :





Step 4:

* I have Created an RDS INSTANCE of POSTGRE SQL Engine in Amazon RDS with proper security and networking Settings.

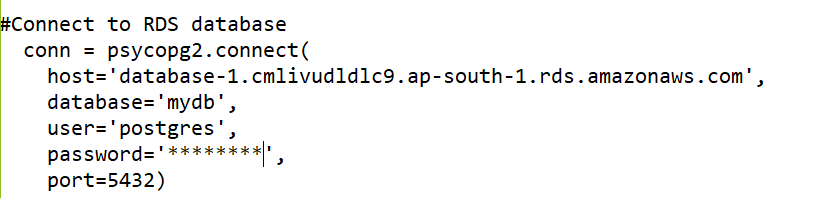


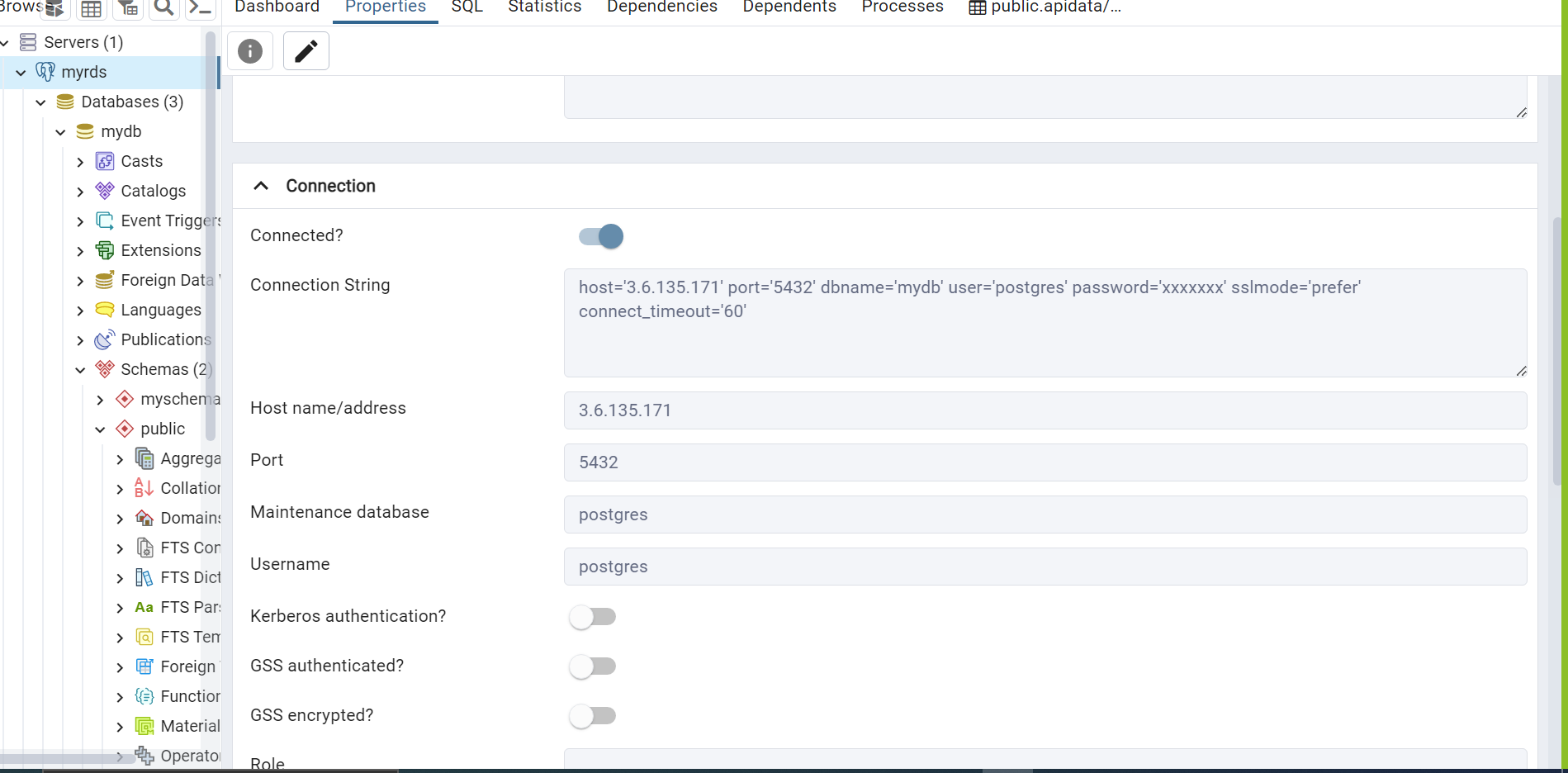
STEP 5:

I have downloaded the pgAdmin tool and connected my RDS instance with the pgAdmin Database.

Steps performed to connect the rds to pgAdmin :

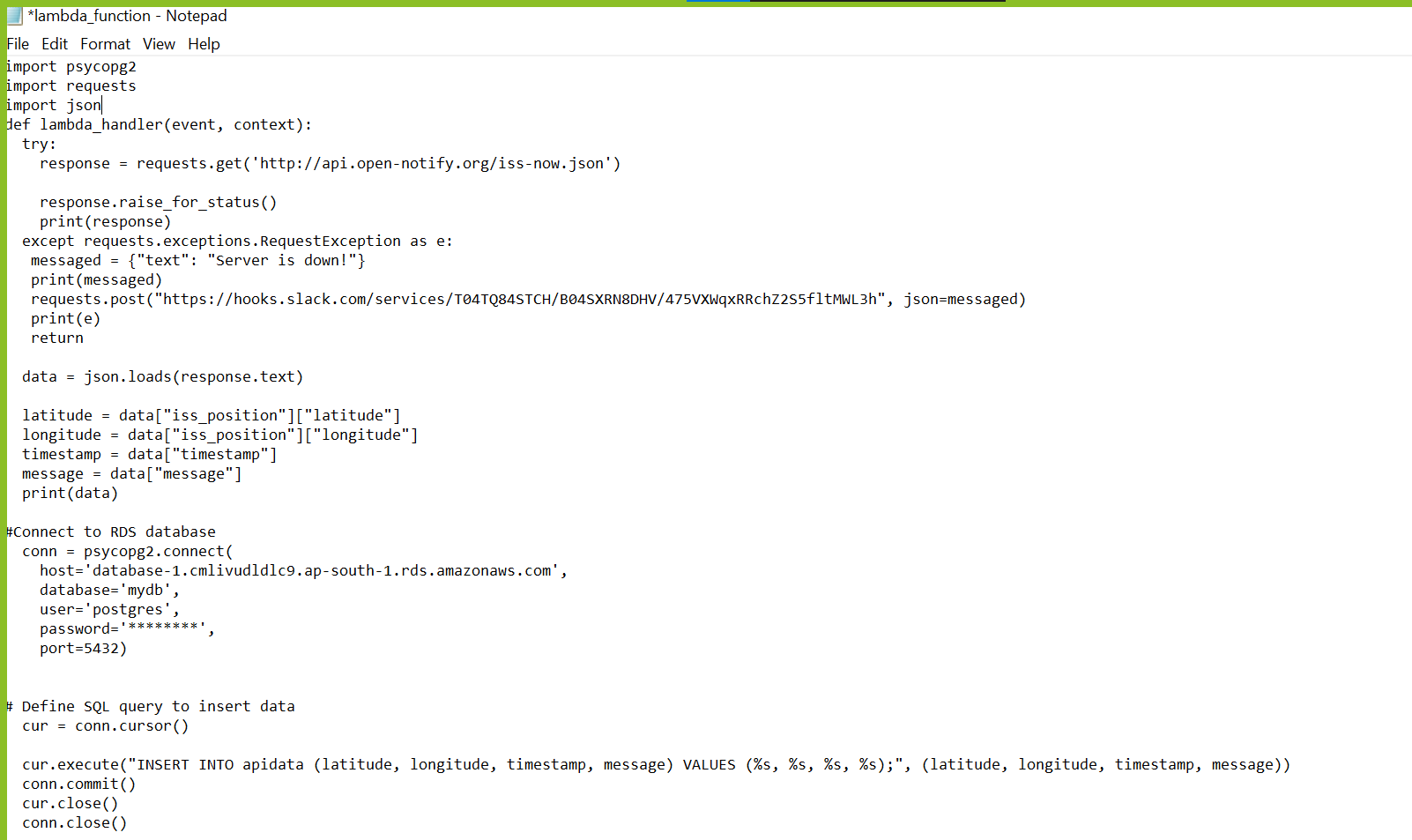
* 1.I have imported a library called PSYCOPG2 and used it to connect my RDS instance with pgAdmin.
* 2.Below is the code that is used to perform the above operation and screenshot for the established connection.



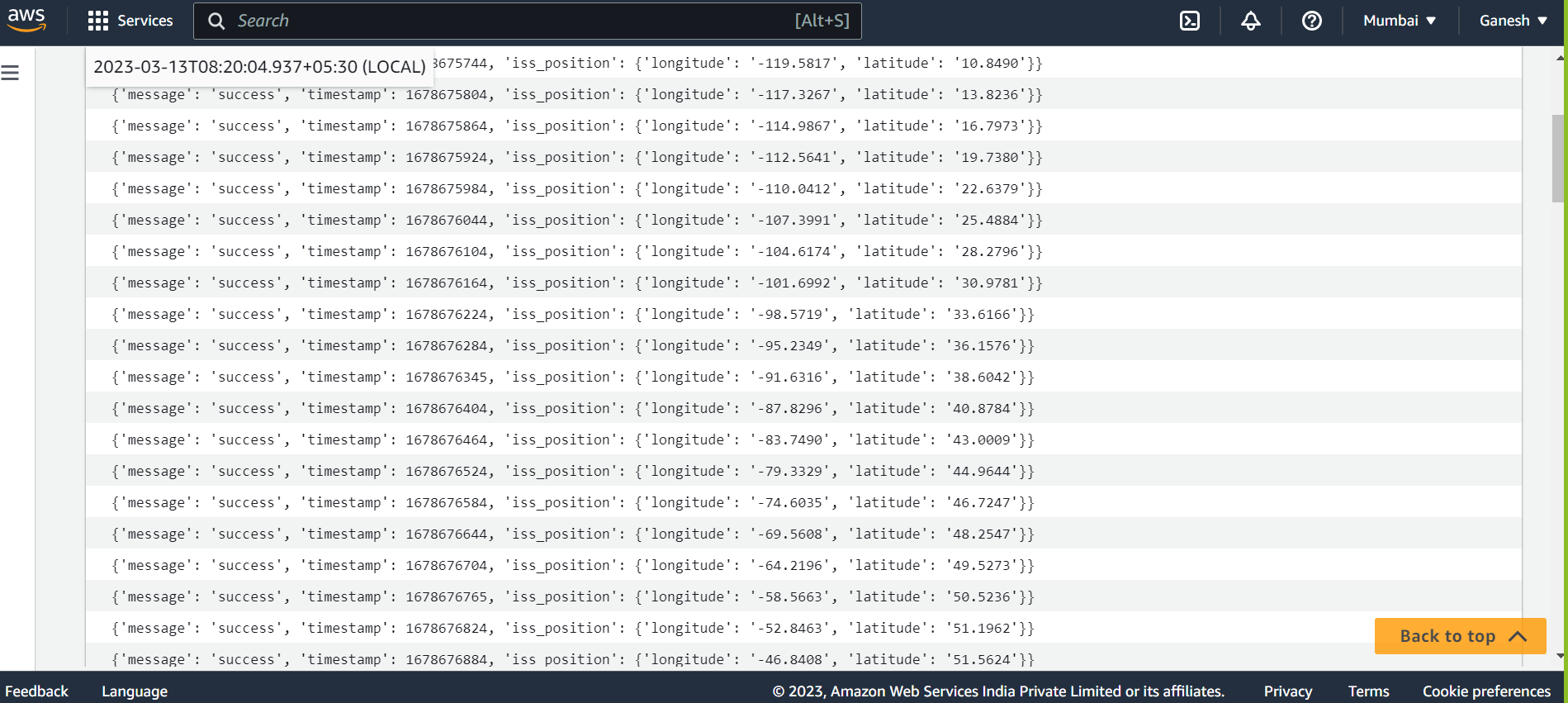


Step 6:

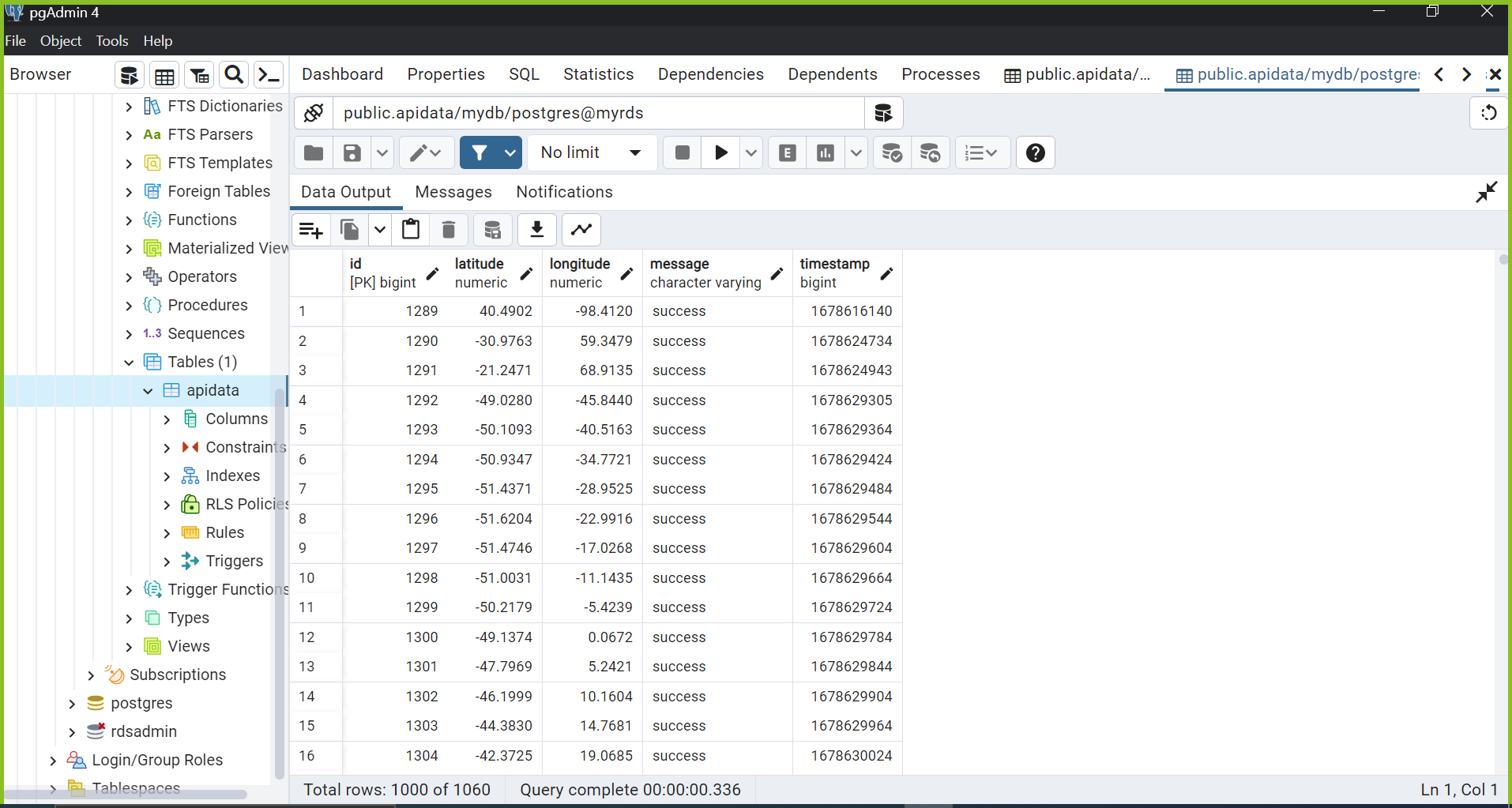
* Then uploaded the below code to the lambda function and tested it to get the desired results.
* Deployed the function to run indefinitely, continuing to fetch and store the data on a regular basis.



* After the code is uploaded the trigger that is created for invoking the lambda function will trigger the lambda function every 1 minute.
* We can monitor the logs using “CLOUDWATCH LOGS” at any given time.



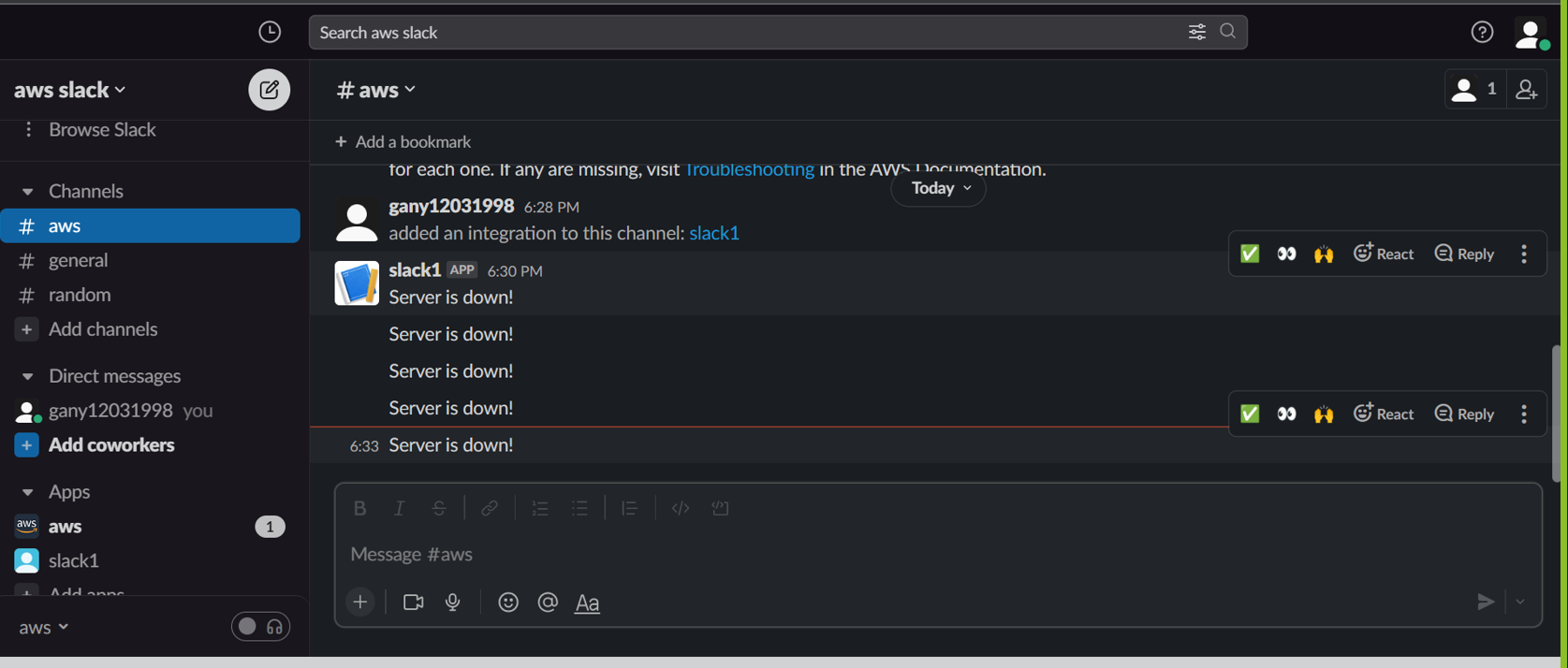
* The Data that is fetched on regular basis is stored in the pgAdmin Databases as you can see below.



Exception Case:

If the server is down or unavailable due to any reason/circumstances an alarm is triggered and a slack notification is sent to slack community/channel saying that “{Server is Down}”.

Please refer the below screenshot.



Conclusion:

* In this project I have fetched the data from API at regular intervals and uploaded it to the database by using appropriate libraries.
* Sent a Slack Notification when the server is unavailable/down due to certain reasons.