FACULTY OF COMPUTER SCIENCE

ASSIGNMENT 5

In
The Class of

CSCI5710: SERVERLESS DATA PROCESSING

by

Janvi Patel [B00863421]

Submitted to
Prof. Saurabh Dey
Department of Computer Science
Dalhousie university.

Date: 30th July 2021

Part A. Explore & Build a Use Case:

AWS Kinesis

Amazon Kinesis is a service offered by Amazon to make it easy to process and analyze the real time streaming data to get the timely insights and react quickly to new information. This provides flexibility to choose the tools that provides streaming data at any scale with cost effective. Moreover, Amazon Kinesis provides a respond to process and analyze data to respond and processing can begin.

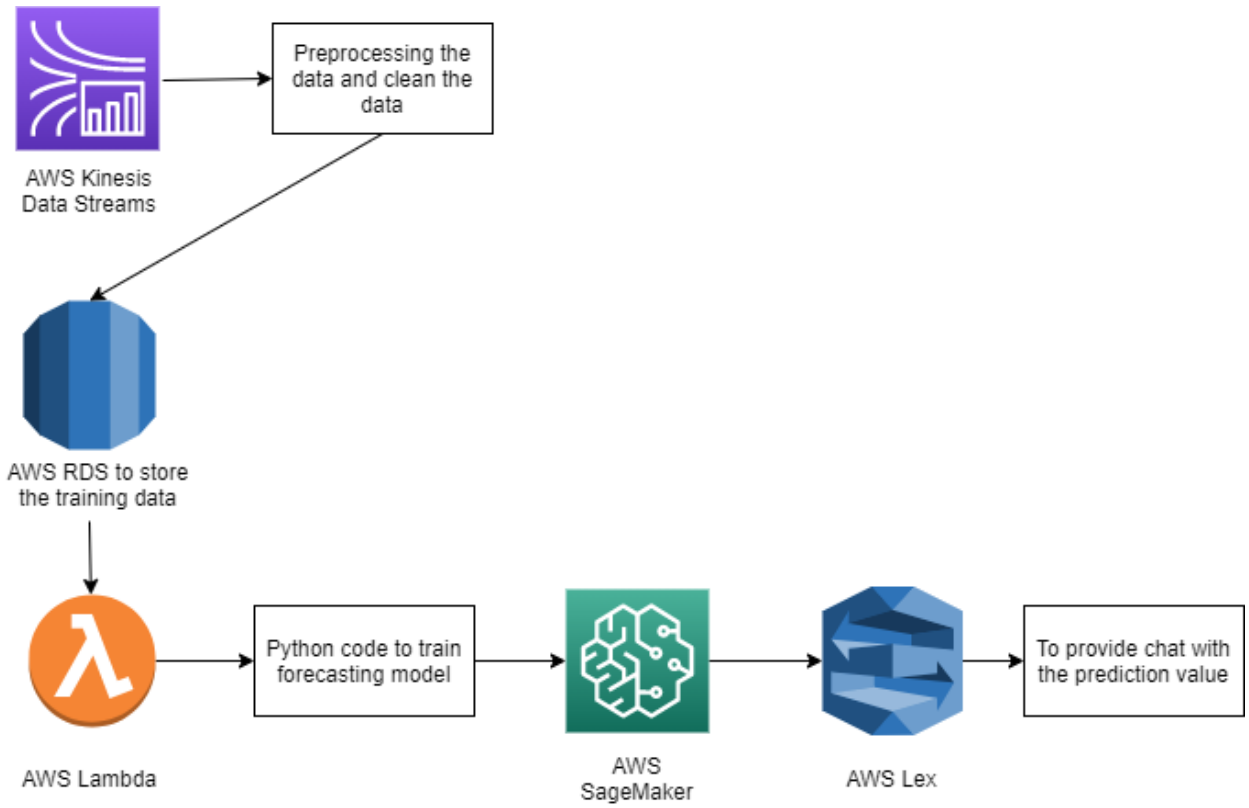
There are 3 basic components described below:

1. Kinesis data/video stream receives data from multiple sources (click streams, IoT devices, video recording devices, etc.) and divides it in shards (or partitions). It allows multiple applications to consume the same data.
2. Kinesis data firehose is service loads the streaming data (from Kinesis streams) into various data lakes, stores, AWS S3, MongoDB, etc. It is a fully managed service which scales according to the data.
3. Kinesis data analytics is service allows for advanced analytics of the data. It allows users to respond to data in real-time.

Scenario – A machine that works on heavy equipment in which depends on massively on the machines. That has some sensors that continuously provides data for example pressure, temperature, humidity, light sensor. They now want to predict that based on the sensor the machine when the machine is going to fail. Further, the company has already provided that past data with all sensor values and what was the exact sensor value when the machine has failed. This means that bots should be able to respond to chat messages.

Use-Case – The company can leverage the processing power and infrastructure provided by different AWS Services. Having huge machinery and maintaining the machine is it's self a big task. The sensor data is being fetched real time and that is streaming data fetched using AWS Kinesis streams. Further, AWS Lambda when coupled with Kinesis the streaming data will be fetched and then store them in the AWS RDS database. Training of the model will be done on AWS Sage Maker and use Machine learning forecasting. These models could be continuously updated by training and testing them on live data provided by Kinesis streams. AWS Lex could be integrated with the system to provide interaction of predicting the sensor values for machine that can be stopped in the neared future. This will make the employees life easy to find alert that this machine is going to fail.

Block Diagram –



Part B. AWS SNS and AWS SQS:

1) Create a queue in AWS SQS in figure 1.

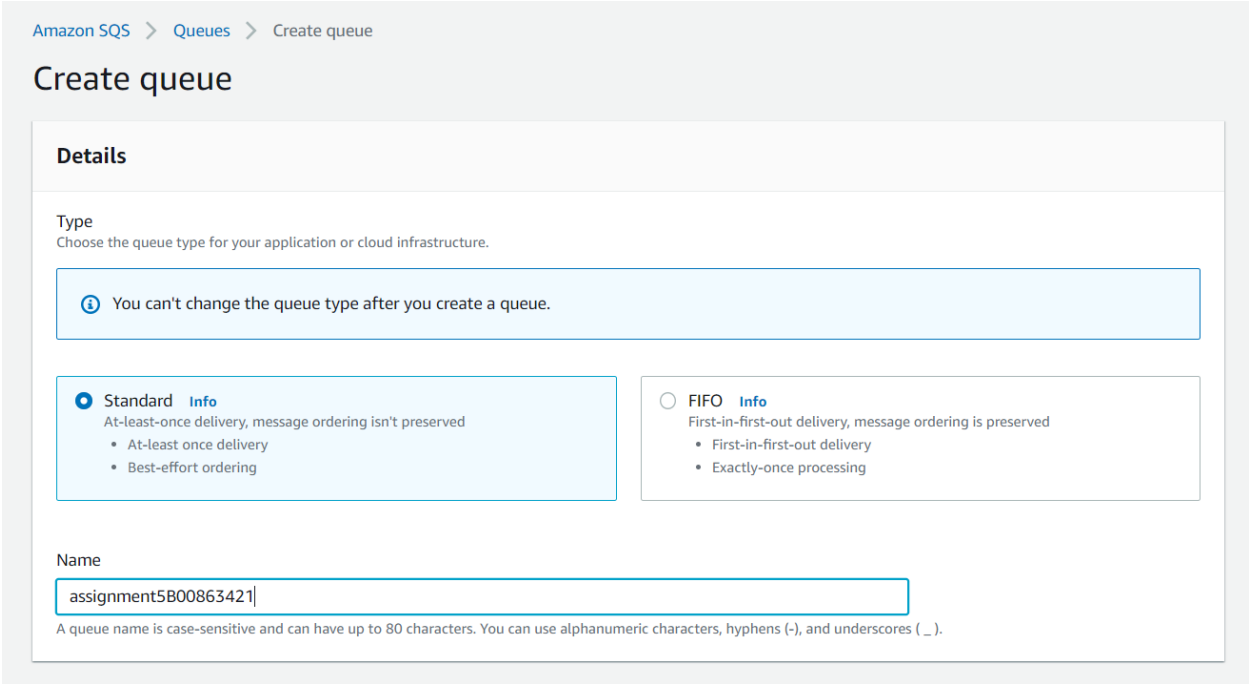


Figure 1 create queue

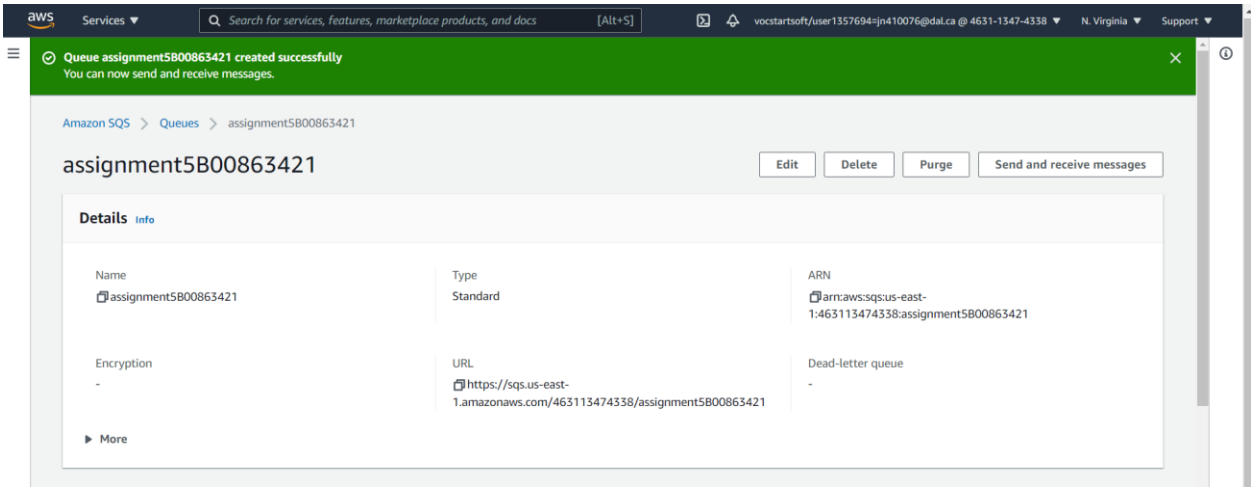


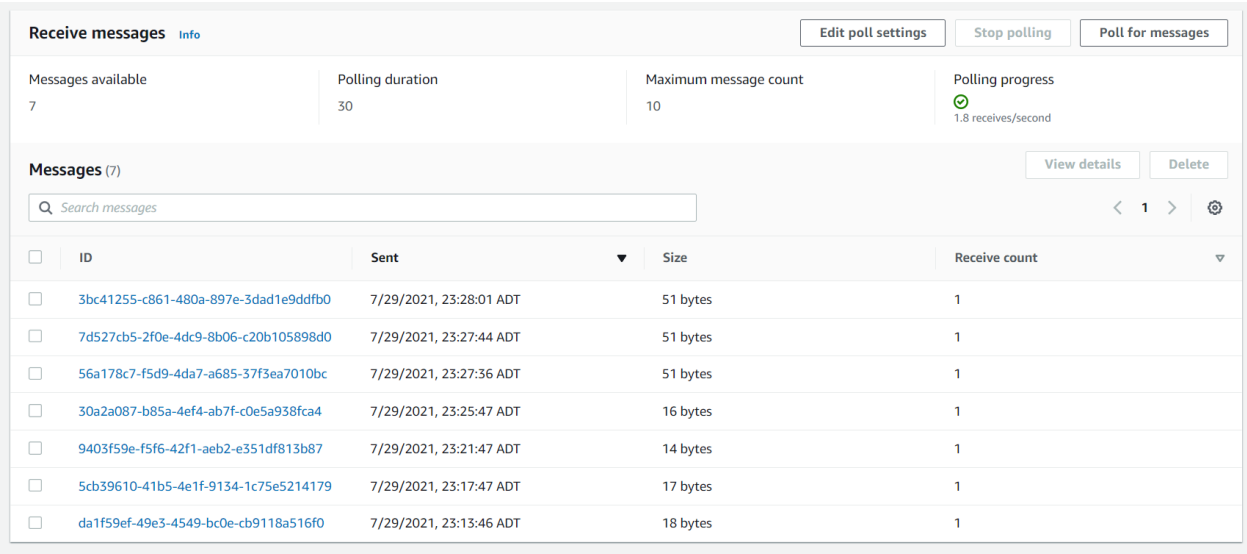
Figure 2 Successfully created SQS

The code below provides details of AWS SQS code:

```
# -*- coding: utf-8 -*-


import boto3
import time
import random

client = boto3.resource('sqs', region_name='us-east-1')

queue = client.get_queue_by_name(QueueName='assignment5B00863421')

OrderMessage = ['Please confirm order for a fabulous collection of white and
cream flowers make this the perfect gift',
                '5 Mamma Mia order is placed',
                'Can you provide Orchid bouquet at 1991 brunswick street?',
                'I have received wrong order for Tulip bouquet',
                'Do you have white peaceful lily bouquet today?']

Size = ['LARGE', 'MEDIUM', 'SMALL', 'JUNIOR', 'MINI', 'TOSS', 'WAND']


def sendToSQS():
    while True:
        bouquetOrderMessage = (random.choice(OrderMessage))
        bouquetSize = random.choice(Size)
        sendData = bouquetOrderMessage + " " + bouquetSize
        response = queue.send_message(MessageBody= sendData)
        print(response)
        time.sleep(300)


sendToSQS()
```

Python Code for AWS SQS

Message received in SQS mentioned in figure 3.



Receive messages Info

Edit poll settings | Stop polling | Poll for messages

Messages available	Polling duration	Maximum message count	Polling progress
7	30	10	1.8 receives/second

Messages (7)

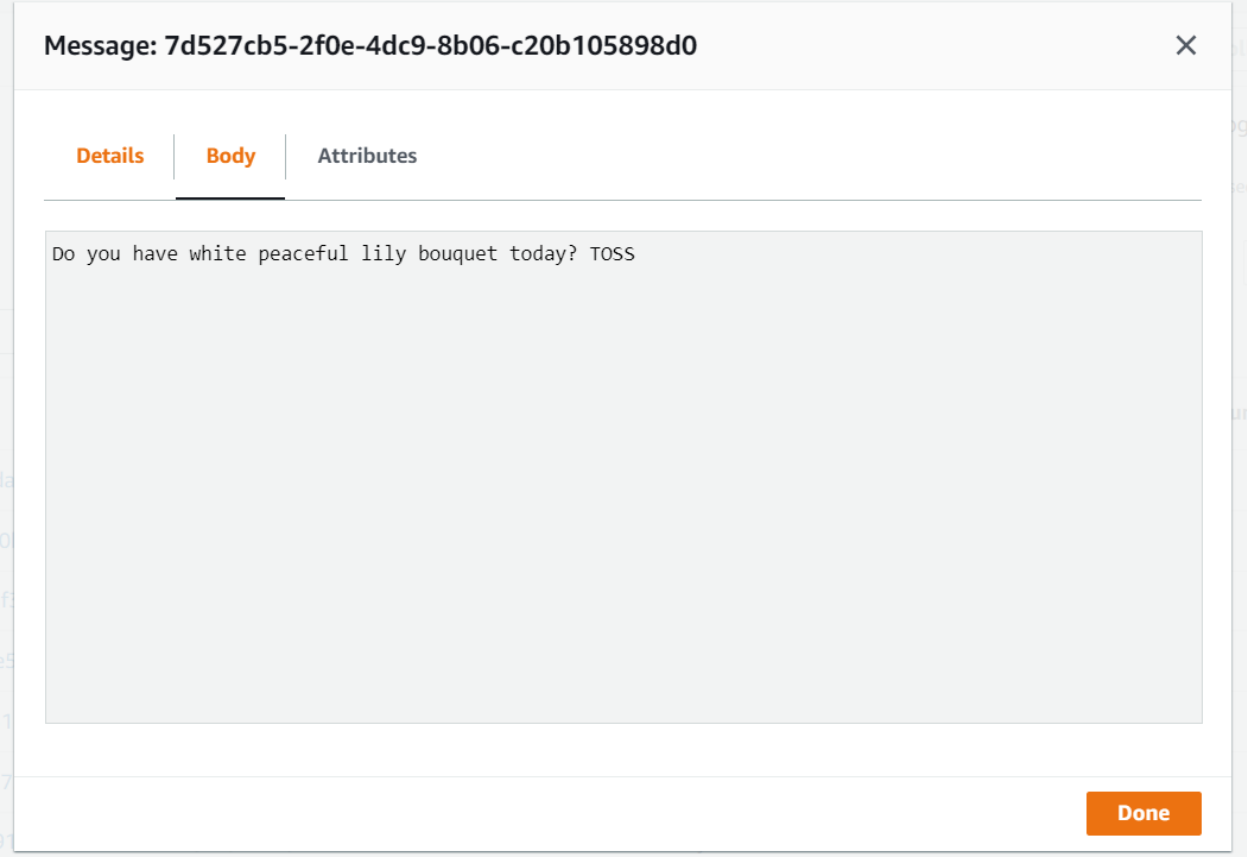
View details | Delete

Search messages

1

ID	Sent	Size	Receive count
3bc41255-c861-480a-897e-3dad1e9ddfb0	7/29/2021, 23:28:01 ADT	51 bytes	1
7d527cb5-2f0e-4dc9-8b06-c20b105898d0	7/29/2021, 23:27:44 ADT	51 bytes	1
56a178c7-f5d9-4da7-a685-37f3ea7010bc	7/29/2021, 23:27:36 ADT	51 bytes	1
30a2a087-b85a-4ef4-ab7f-c0e5a938fca4	7/29/2021, 23:25:47 ADT	16 bytes	1
9403f59e-f5f6-42f1-aeb2-e351df813b87	7/29/2021, 23:21:47 ADT	14 bytes	1
5cb39610-41b5-4e1f-9134-1c75e5214179	7/29/2021, 23:17:47 ADT	17 bytes	1
da1f59ef-49e3-4549-bc0e-cb9118a516f0	7/29/2021, 23:13:46 ADT	18 bytes	1

Figure 3: Message received on polling



Message: 7d527cb5-2f0e-4dc9-8b06-c20b105898d0

Details | Body | Attributes

```
Do you have white peaceful lily bouquet today? TOSS
```

Done

Figure 4: Body message of one of the page

AWS SNS with the Standard Type of the topic mentioned in figure 5:

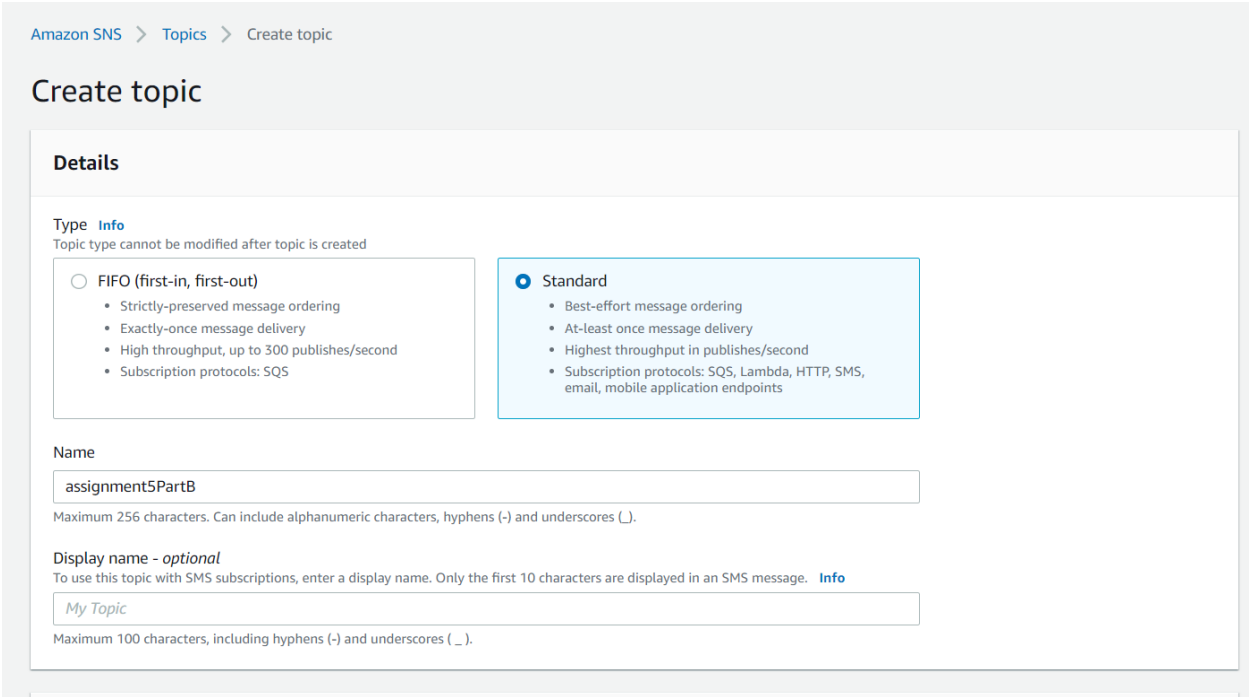


Figure 5: create topic

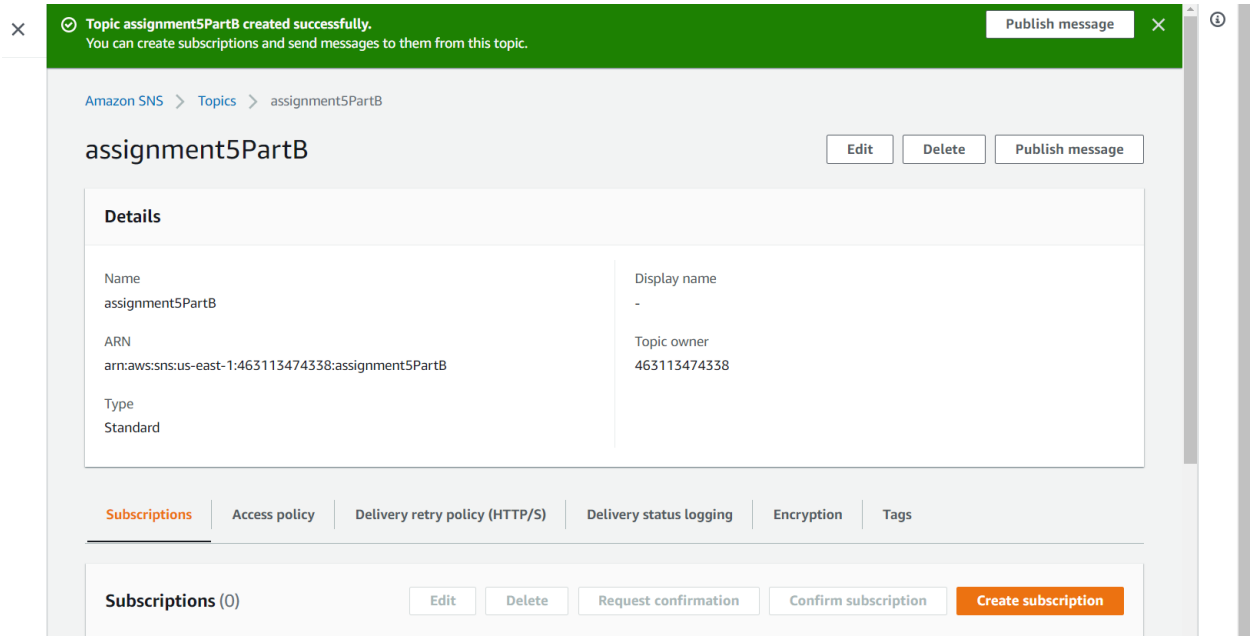


Figure 6: Successfully generated SNS

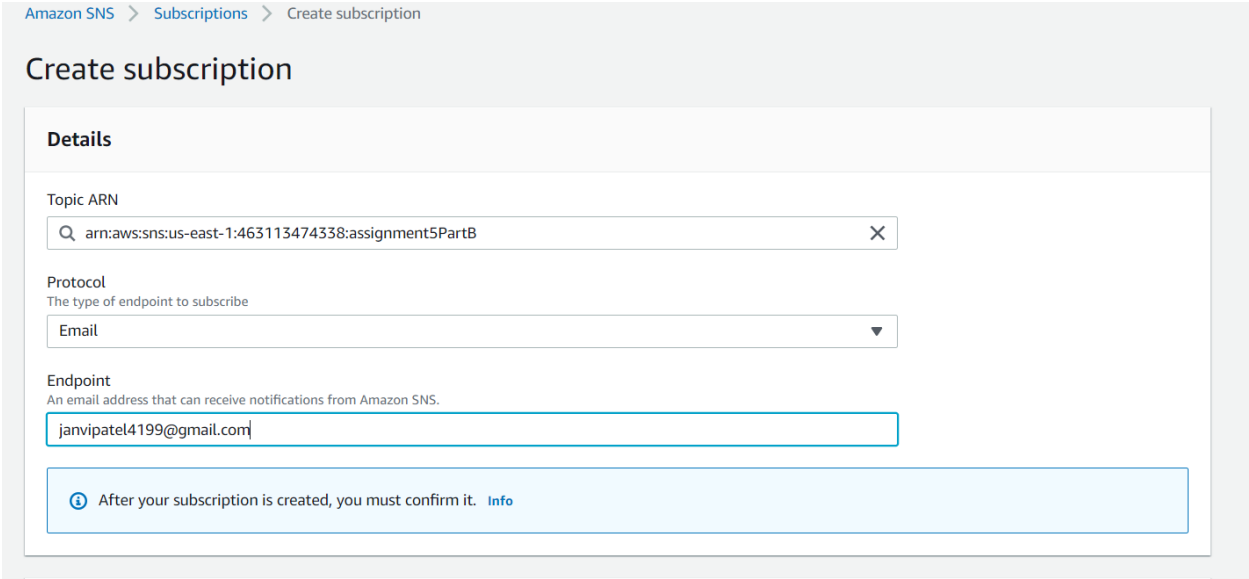


Figure 7: Provided Email as Endpoint

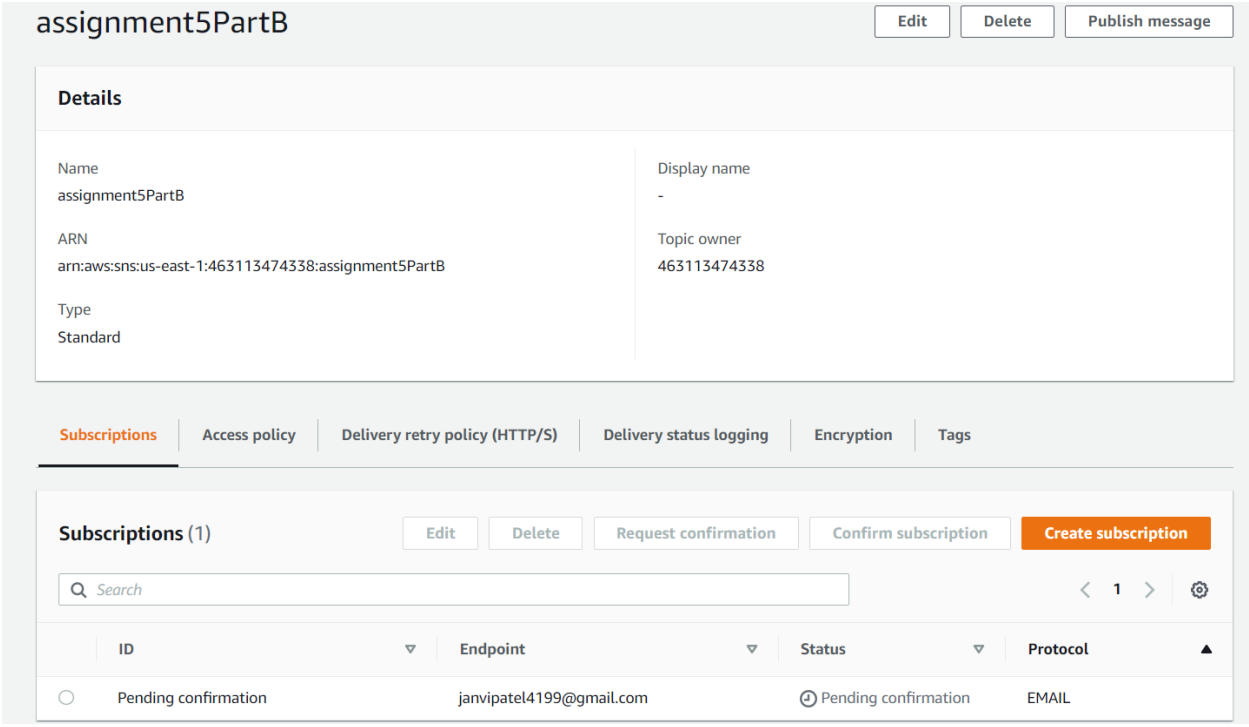


Figure 8: Details of ARN

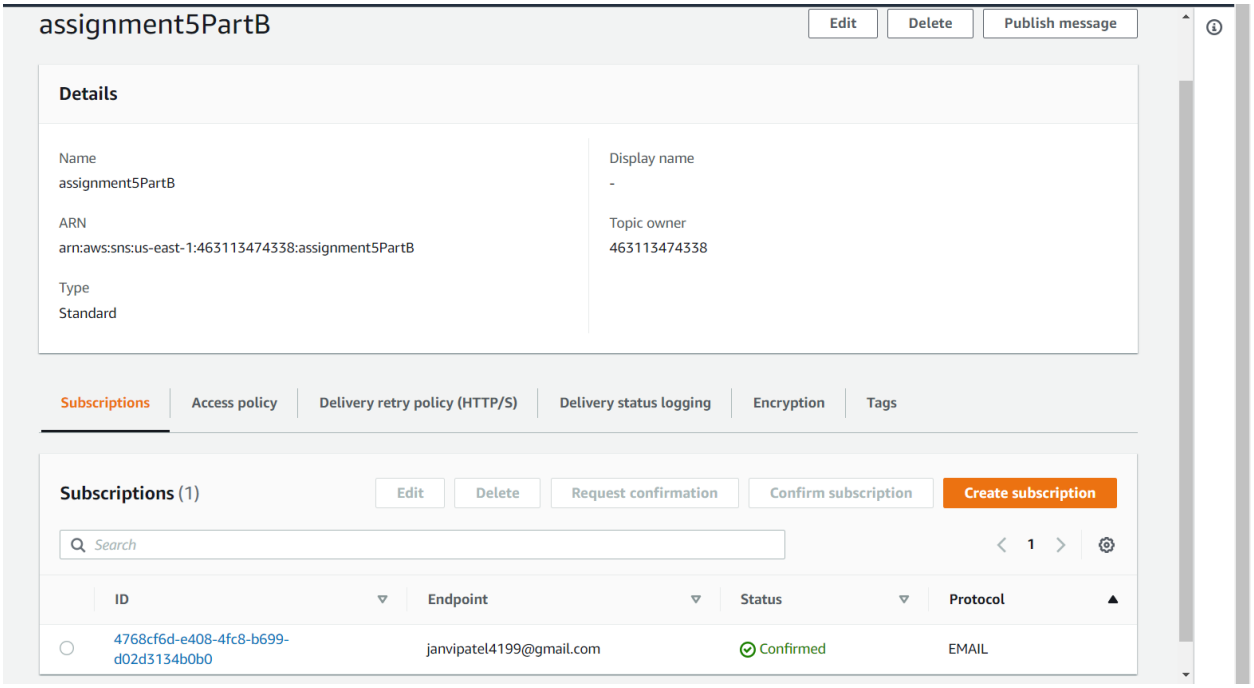


Figure 9: ARN details of SNS

IAM role:

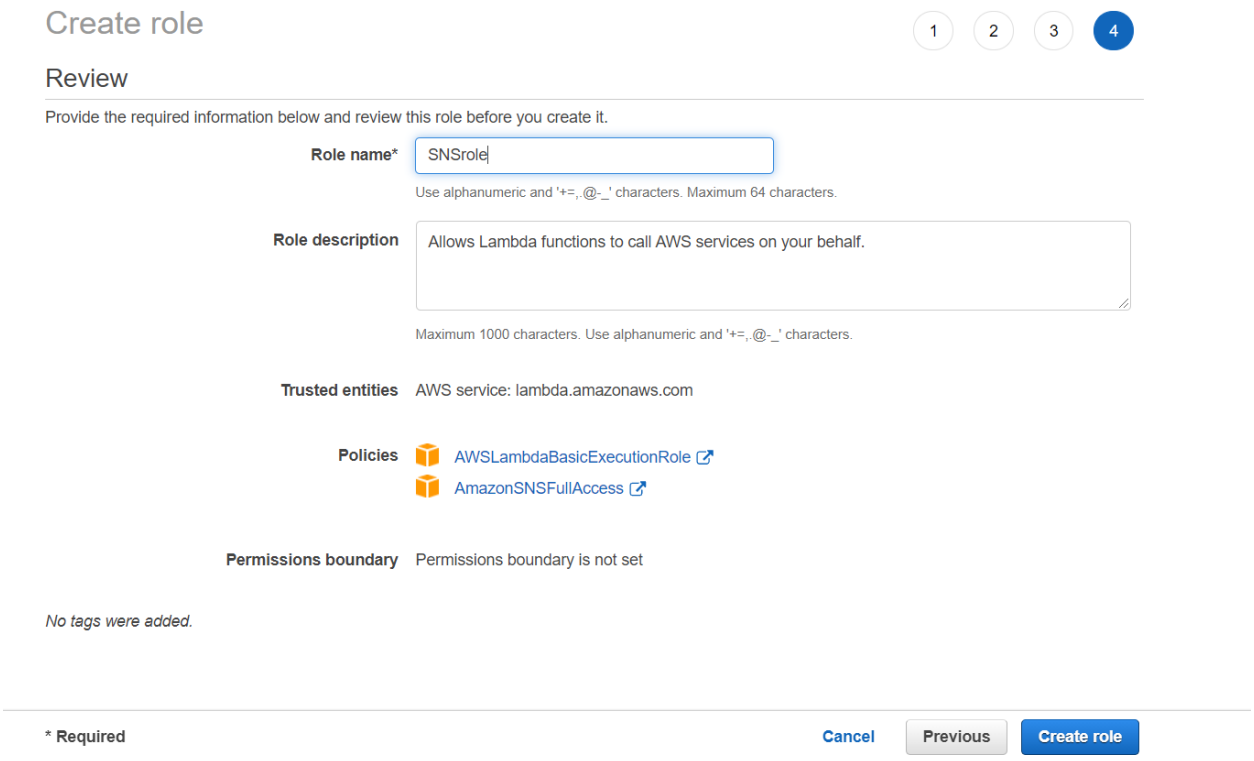


Figure 10: Provide Lambda Access

Lambda Creation:

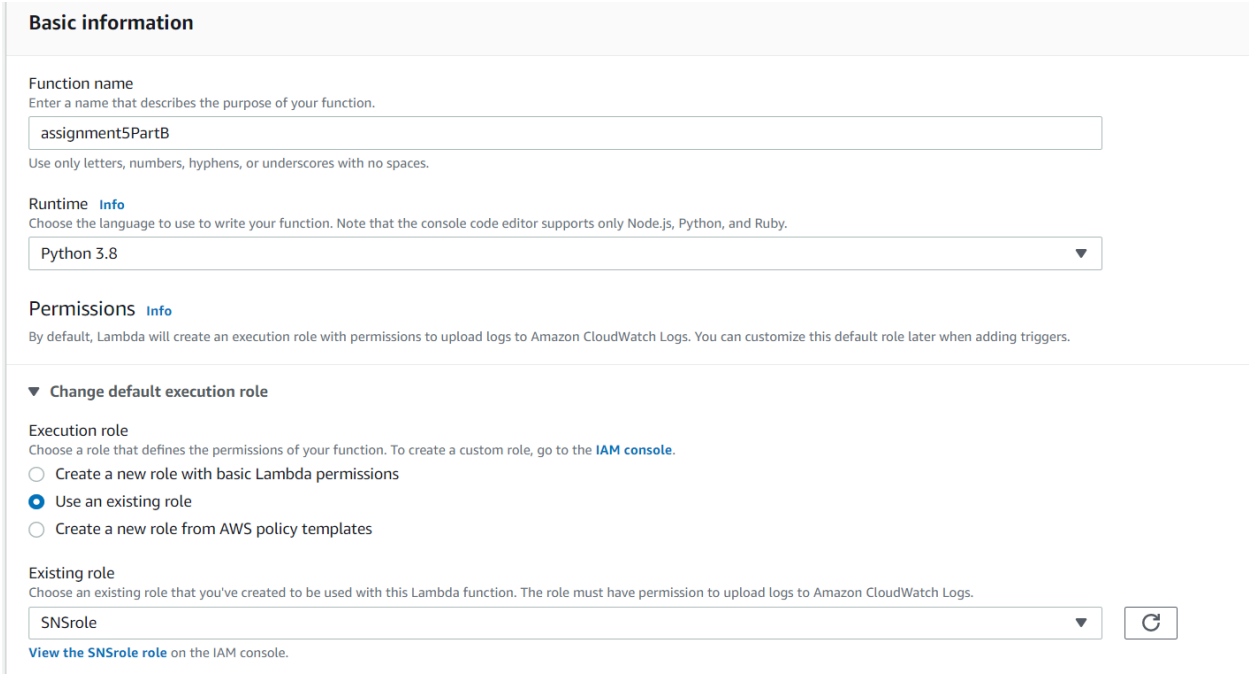


Figure 11: Lambda creation

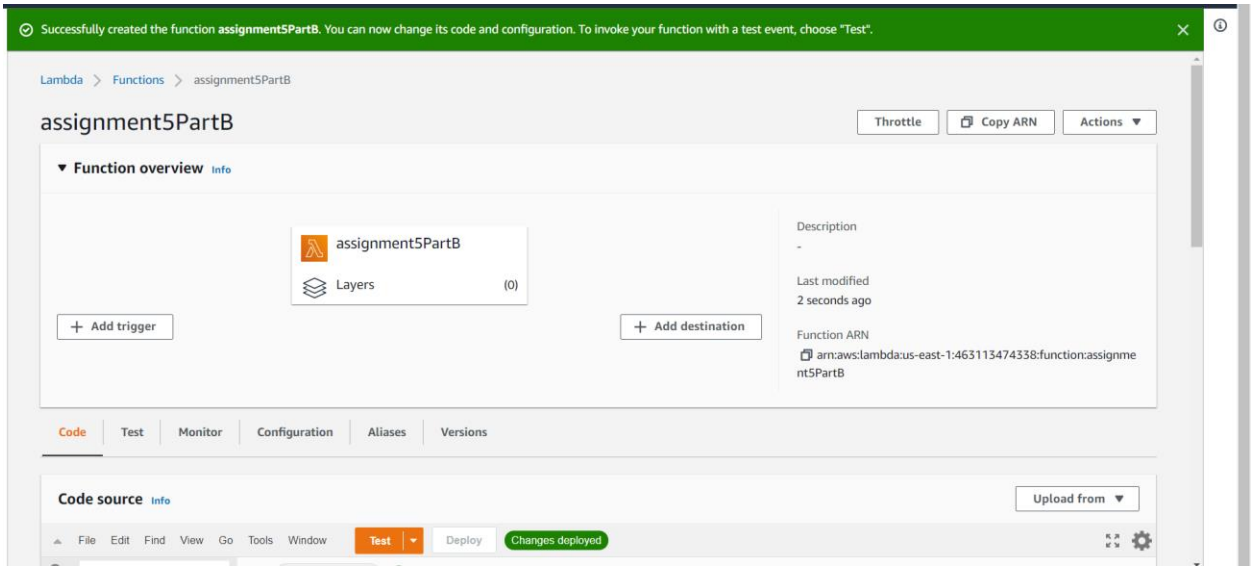


Figure 12: Lambda successfully generation

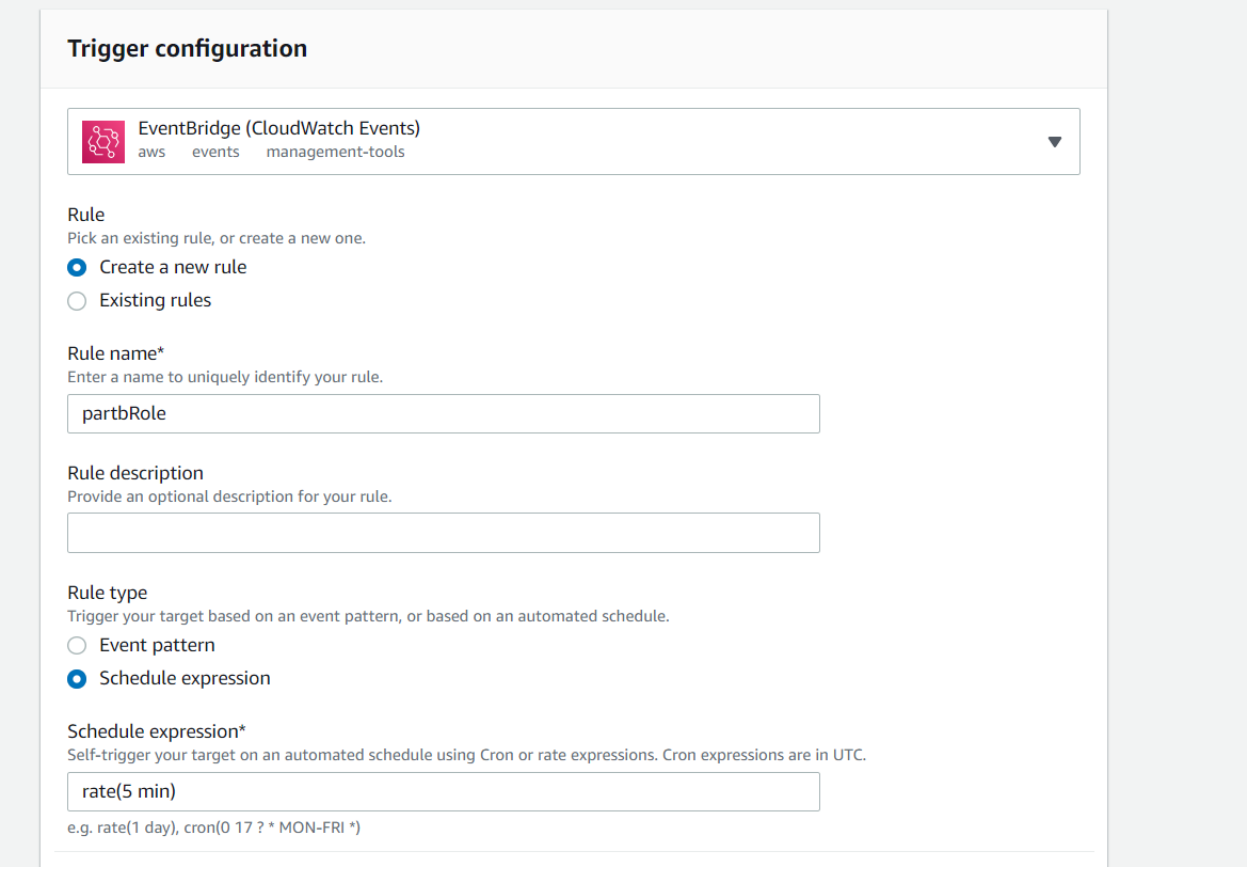


Figure 13: CloudWatch Events

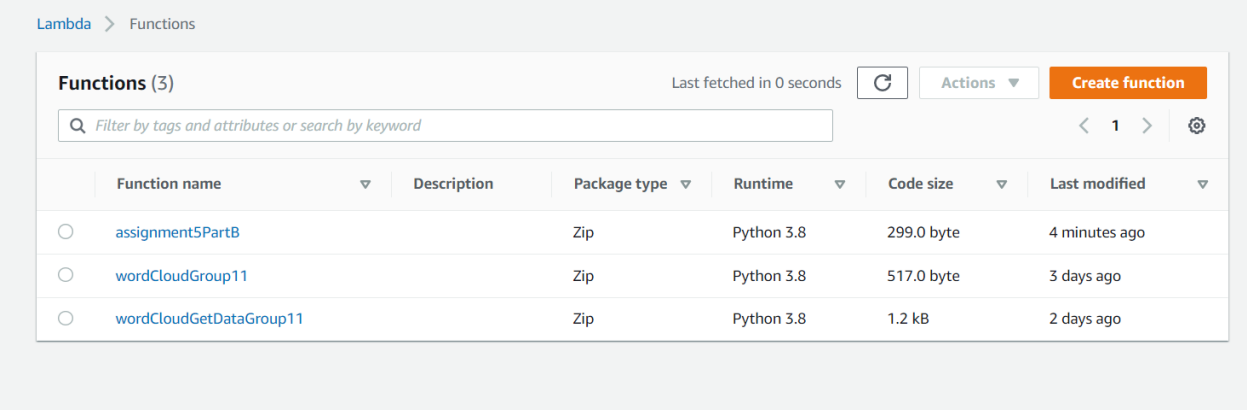
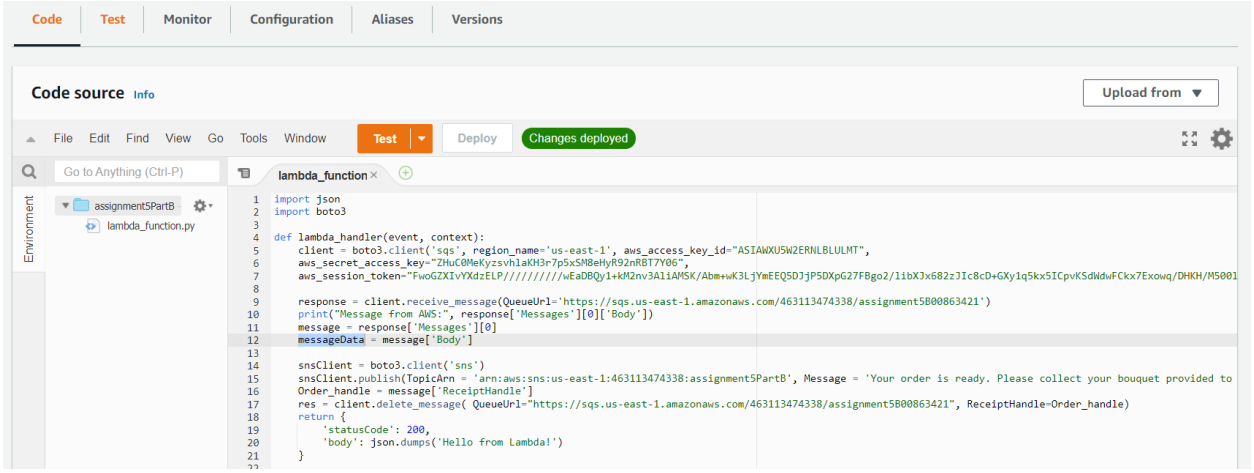


Figure 14: Lambda successfully generated



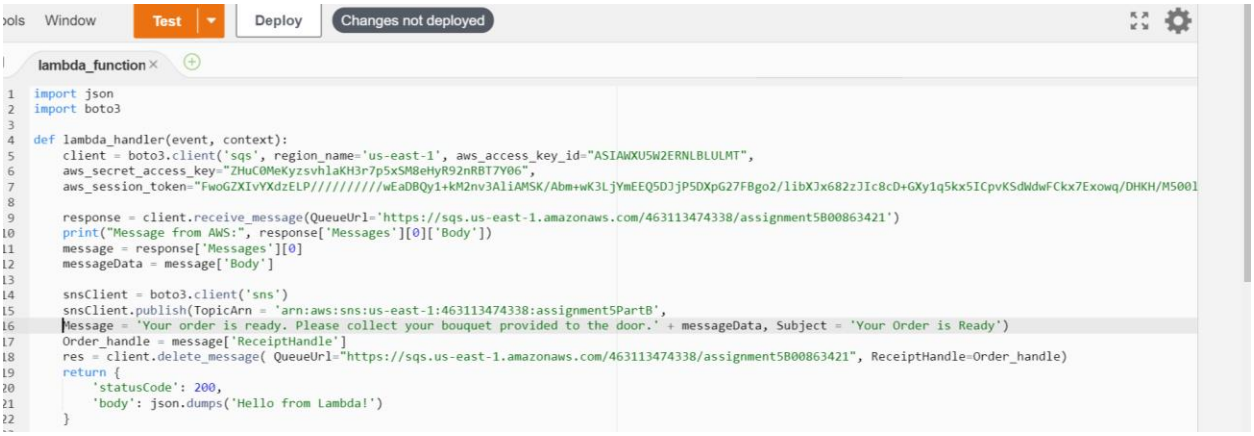


Figure 15: Lambda code

Logs that Lambda code is successfully running:

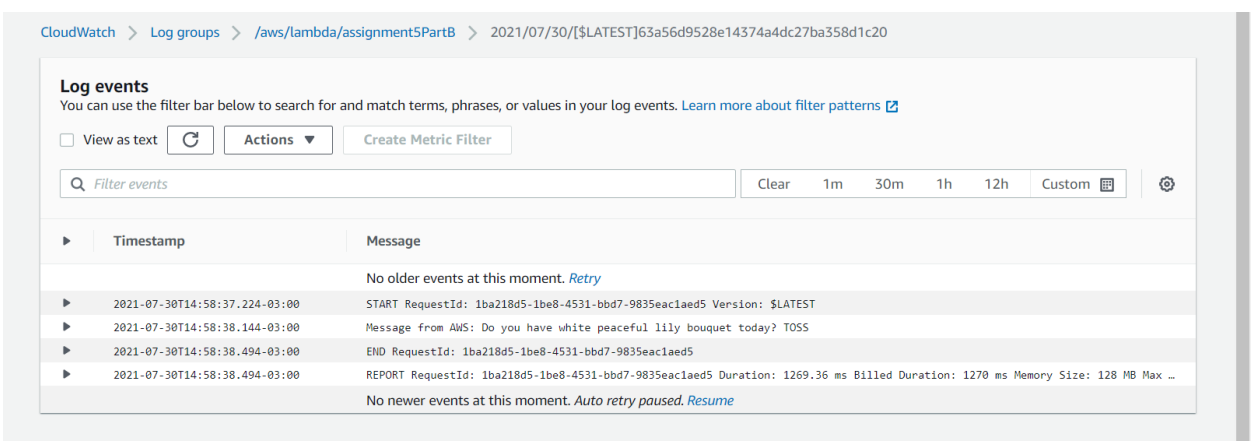
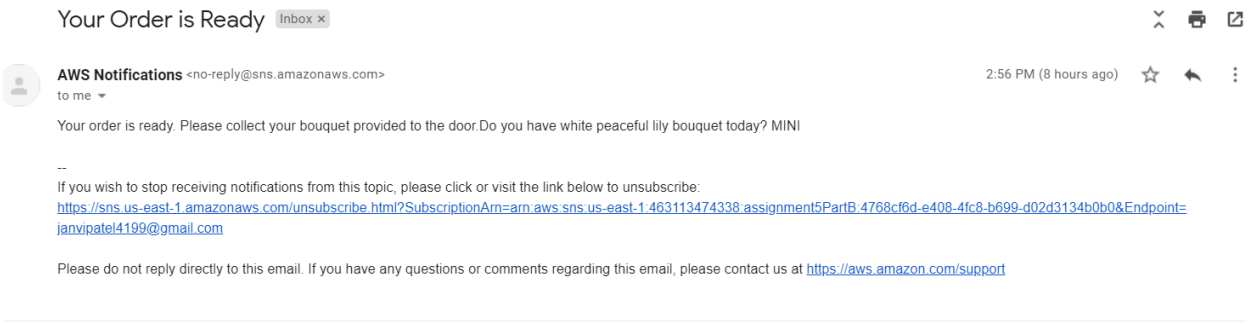


Figure 16: Cloud Watch Logs

SNS Mail received:



Your Order is Ready Inbox ×

AWS Notifications <no-reply@sns.amazonaws.com> 2:56 PM (8 hours ago)
to me

Your order is ready. Please collect your bouquet provided to the door.Do you have white peaceful lily bouquet today? MINI

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:463113474338:assignment5PartB:4768cf6d-e408-4fc8-b699-d02d3134b0b0&Endpoint=janvipatel4199@gmail.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

Figure 17: Mail received from SNS