CSCI 5410 - Serverless Data Processing

Assignment - 5

Name	Banner Id	Mail id
Bommera Geetanjali	B00881511	gt584369@dal.ca

Table of Contents

1. Part A

1. Introduction

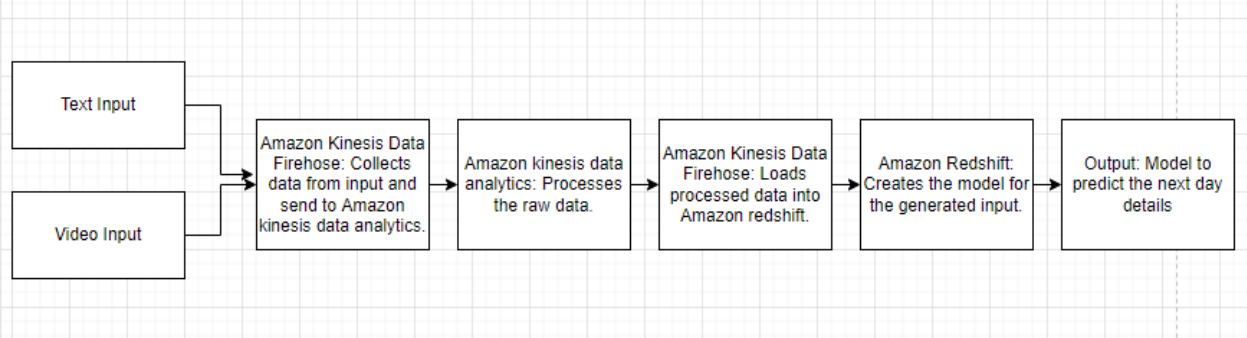
AWS Kinesis is particularly used to collect, process, and analyze real-time data that is continuously streaming. It is mainly used in the case of real-time applications where scalability is a major concern. Data that is collected from different resources is by kinesis video streams or kinesis data streams based on the type of data received. Once data is processed it is stored in a data firehouse or Amazon S3 buckets etc. Kinesis data analytics process the data streams in real-time SQL or Apache Flink to process the data that helps further to extract knowledge. Amazon AI services use this data as the input for different algorithms to bring insights. Output from Kinesis data analytics is used as input for creating models using different technologies such as AWS Lambda, Amazon Redshift.

2. Use case

Considering the pandemic last year, there is a great change in symptoms every now and then. The number of cases was increasing each day, and no one was able to predict the situation. In such a situation, it is difficult to collect whole data, analyze and predict the situation. Also, the insights might not be useful by the time model is built. In such a case, analyzing and processing the data as it comes is very essential.

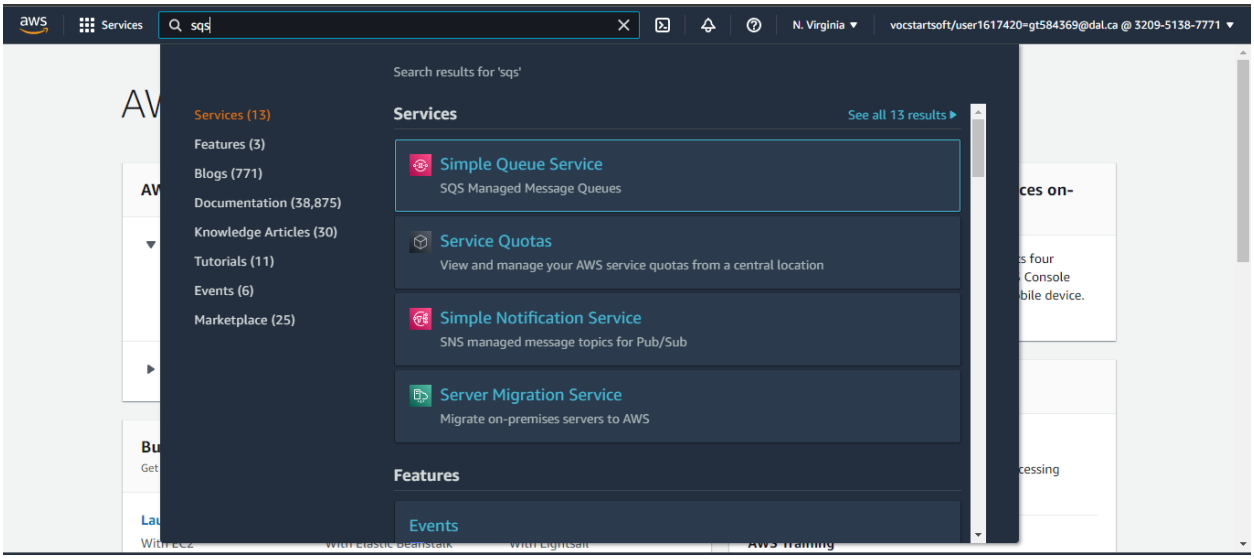
For instance, predicting the number of patients for the next day so that number of beds or rooms can be arranged. Hence data that belongs to the number of patients and type of patients received should be collected. This data can be collected from excel sheets or by scanning cameras while they check-in. Amazon Kinesis data firehouse collects raw data from different inputs and sends it to Amazon kinesis data analytics. Data might consist of patient name, age, health issues, symptoms, medicines currently taken, etc. Once Amazon's kinesis data analytics service receives data it processes data in real-time. Amazon redshift then receives the processed data to build the model from amazon kinesis data analytics through amazon kinesis data firehouse. Once the model is built, users can use this model to predict the number of patients to come to the hospital.

Note: Diagram is drawn using draw.io

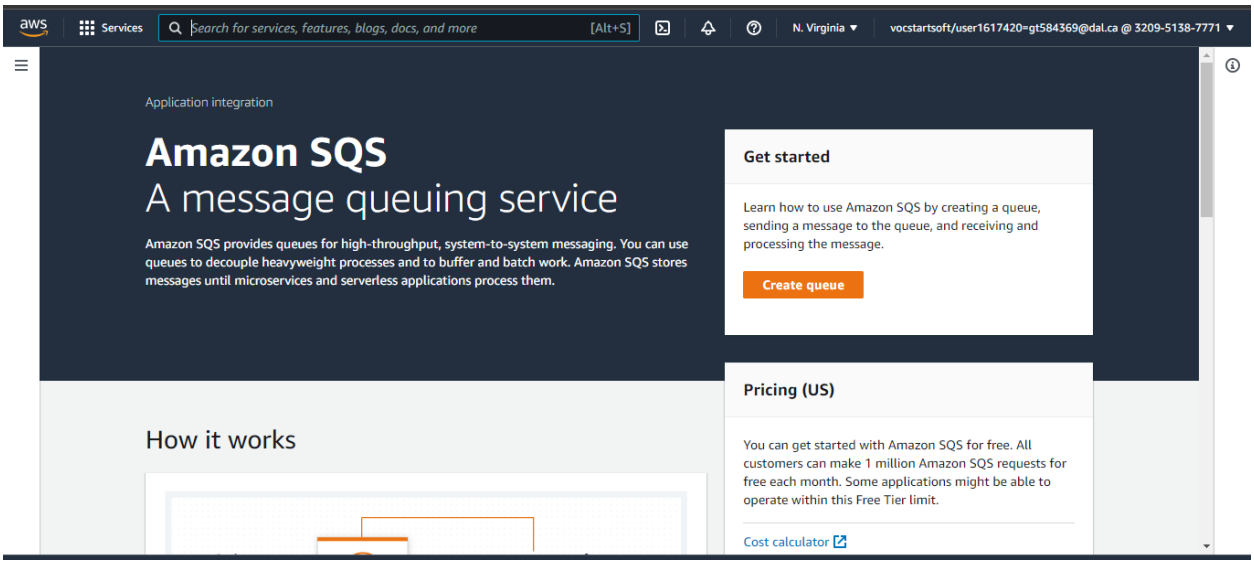


2. Part B

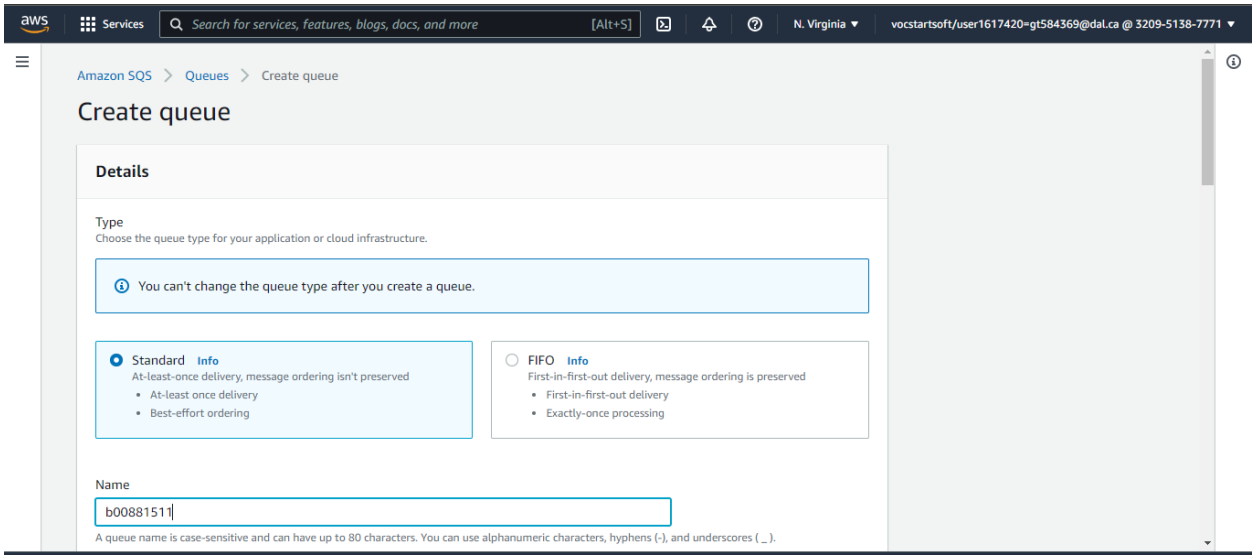
1. Go to AWS console and search for SQS. Click on Simple Query Service [1].



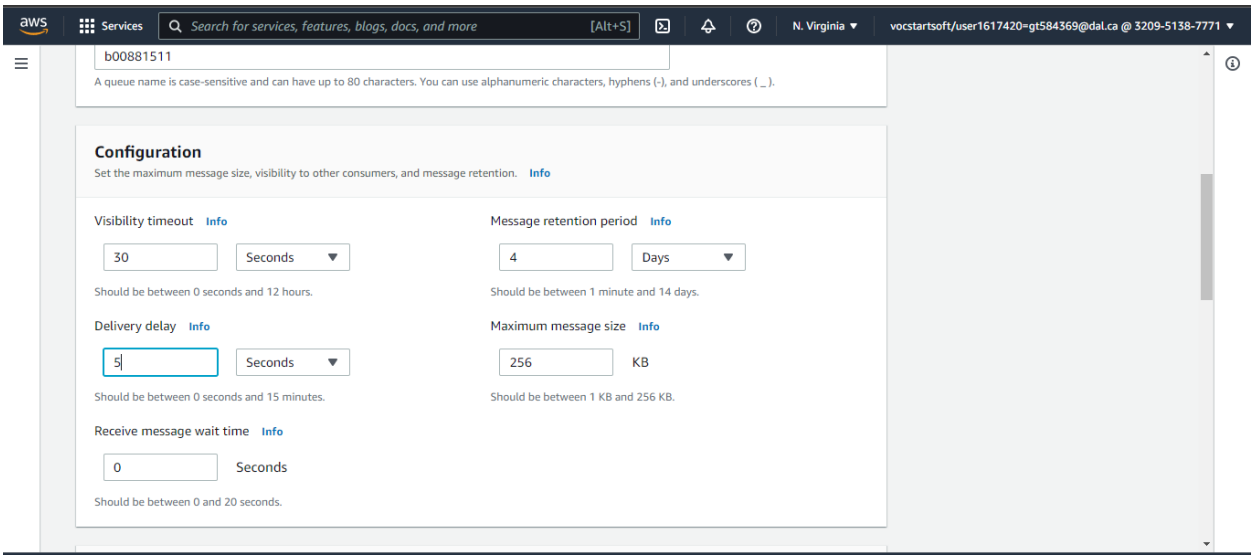
2. Click on create queue to create the queue from the console.



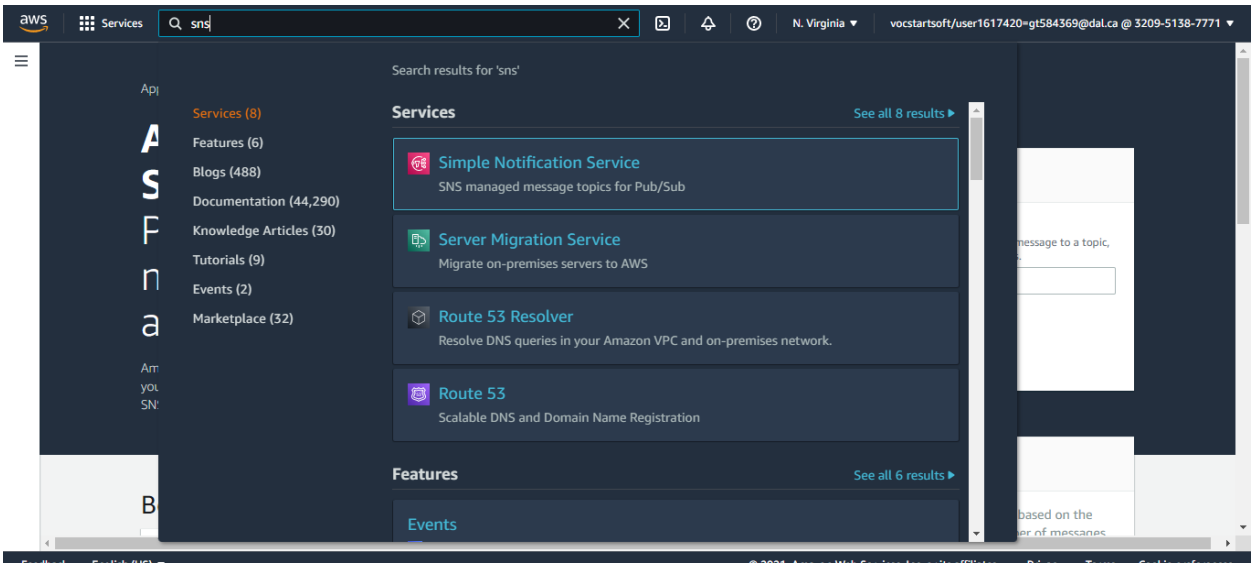
3. Give the name of the queue.



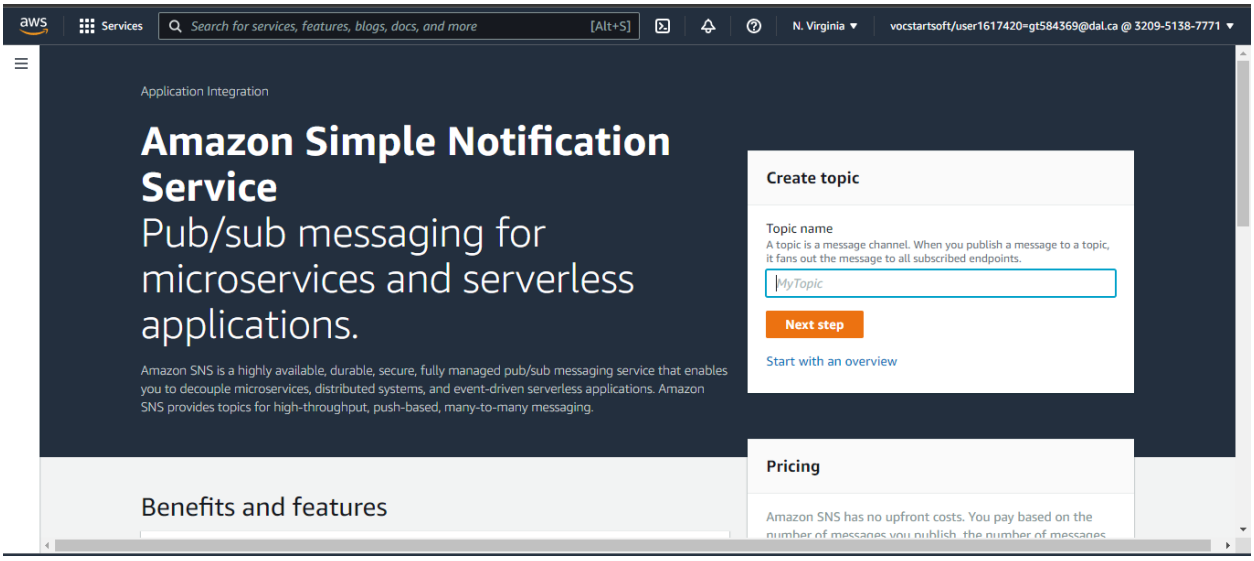
4. Give the delivery delay as 5 seconds and click on create a queue.



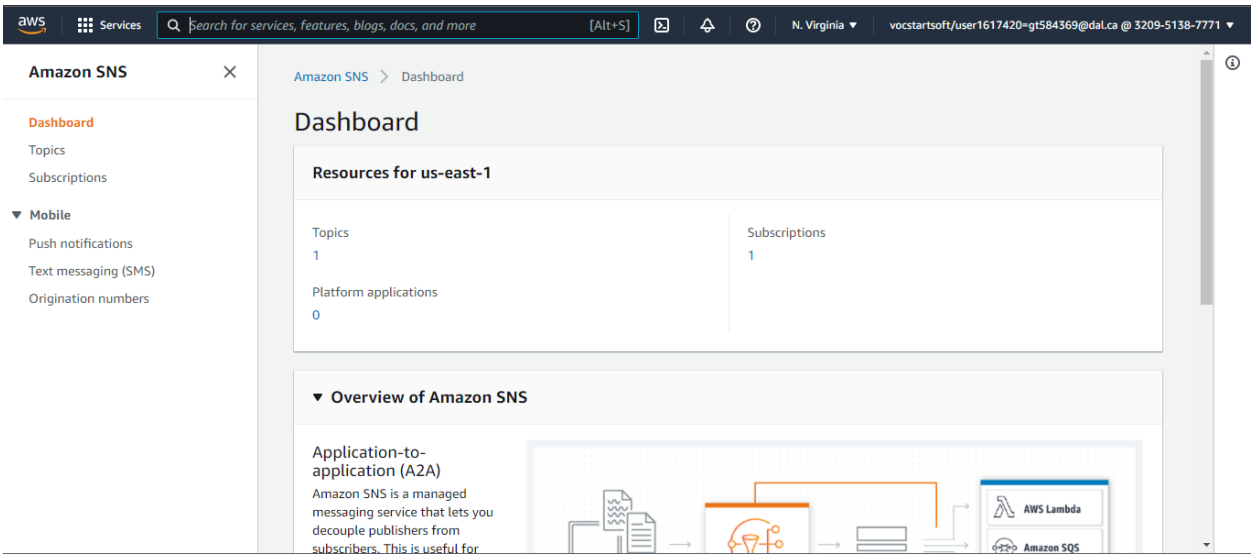
5. Go to the AWS console once again to create a Simple Notification Service. Click on Simple Notification Service [2].



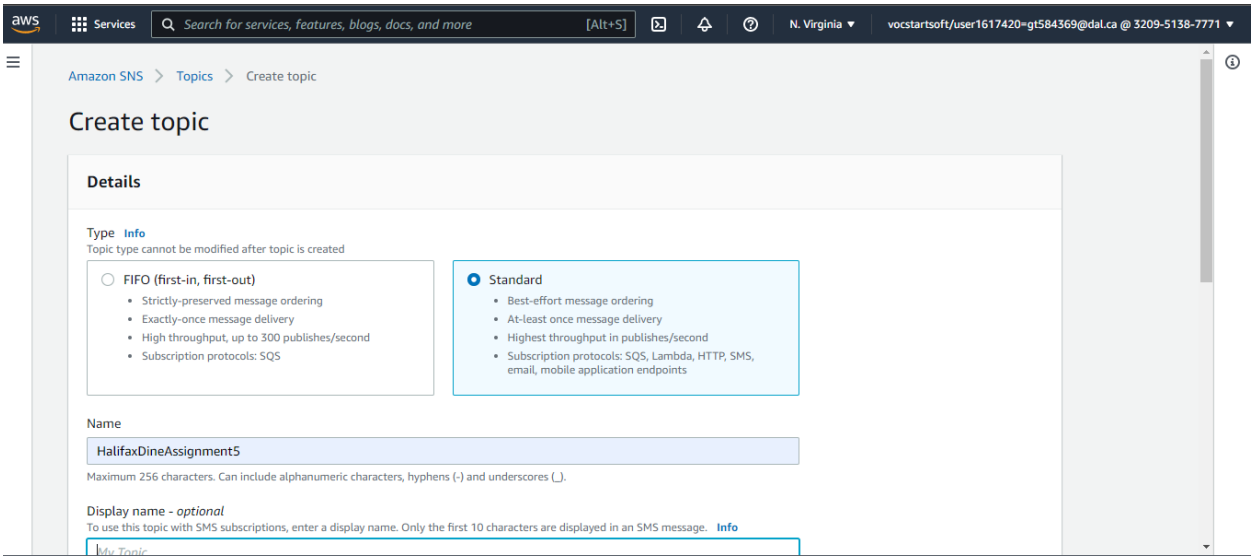
6. Click on start with an overview.



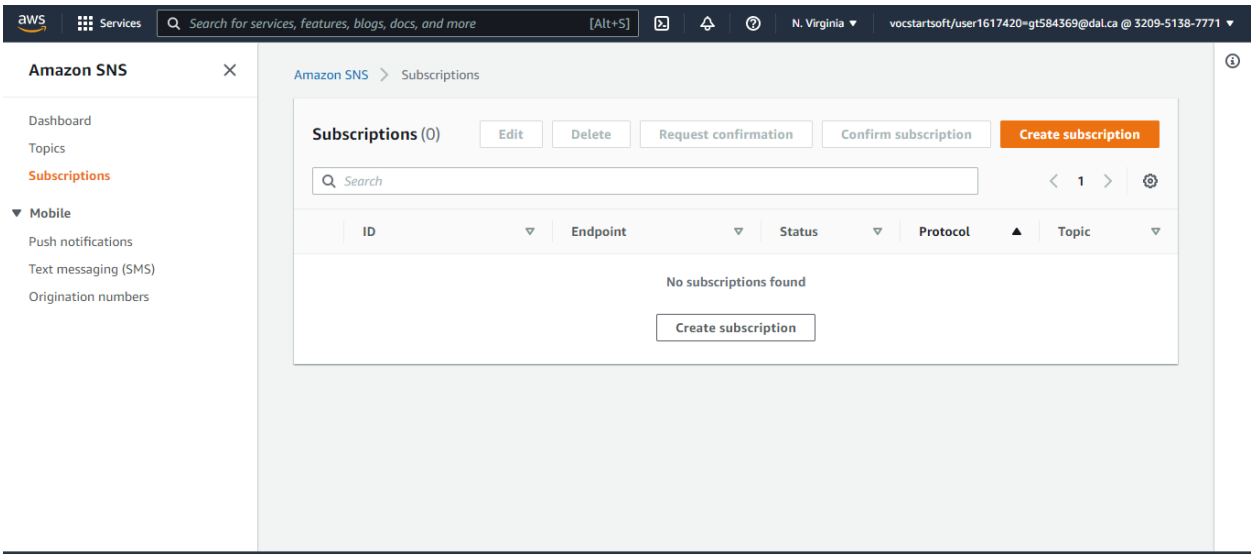
7. Click on topics and then create a topic.



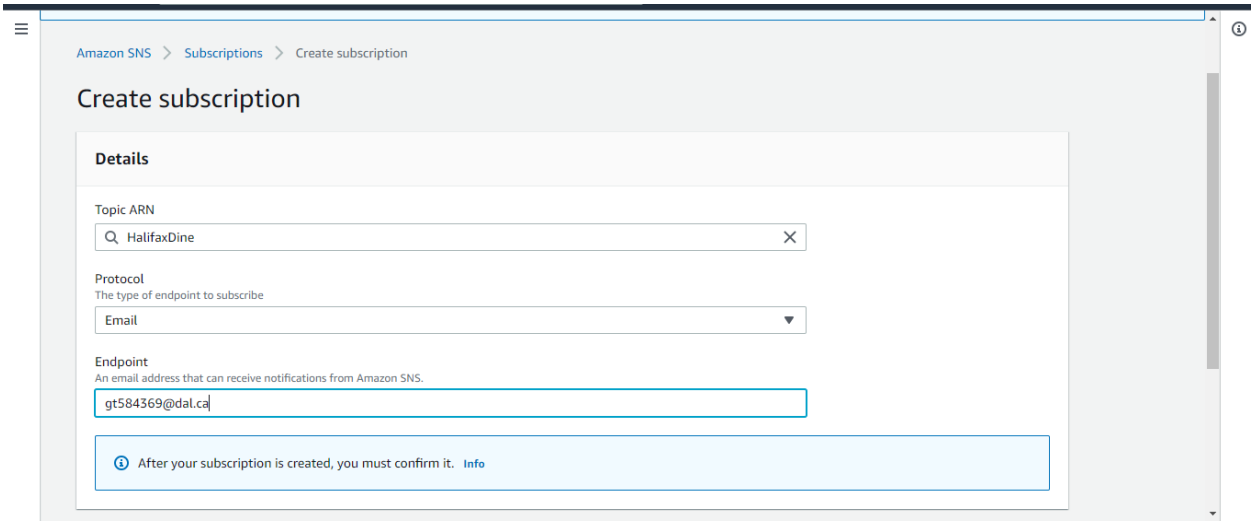
8. Give the name for the topic and choose type as standard. Click on create a topic to save this topic.



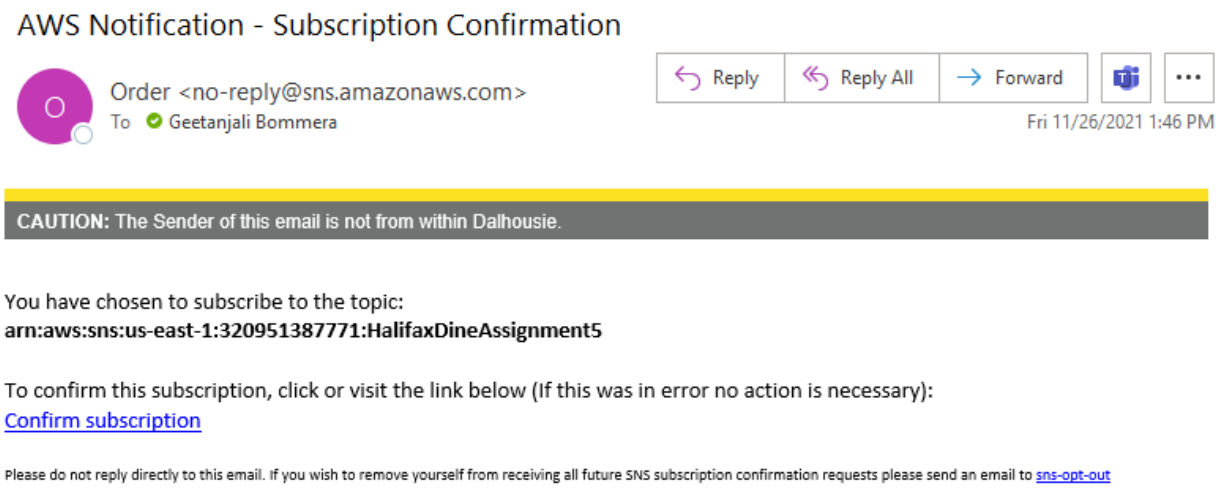
9. Click on subscriptions and then on create a subscription.



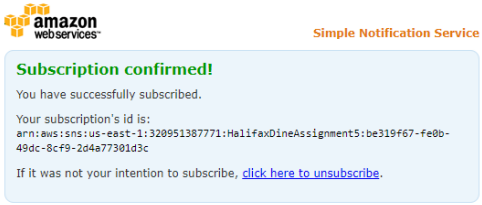
10. Give the topic as the topic created in the above step. Give protocol as email as mail is the medium of notification. Give endpoint as the mail id to which notification needs to be received.



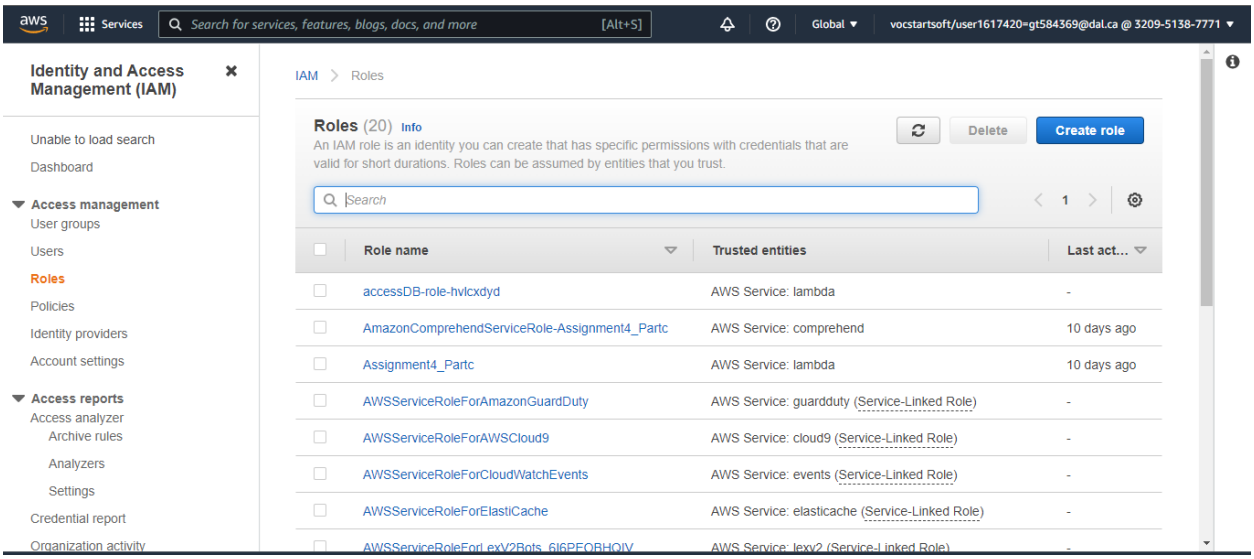
11. Once create subscription button is hit, an email confirmation is sent to the given mail id. Click on confirm subscription in the mail.



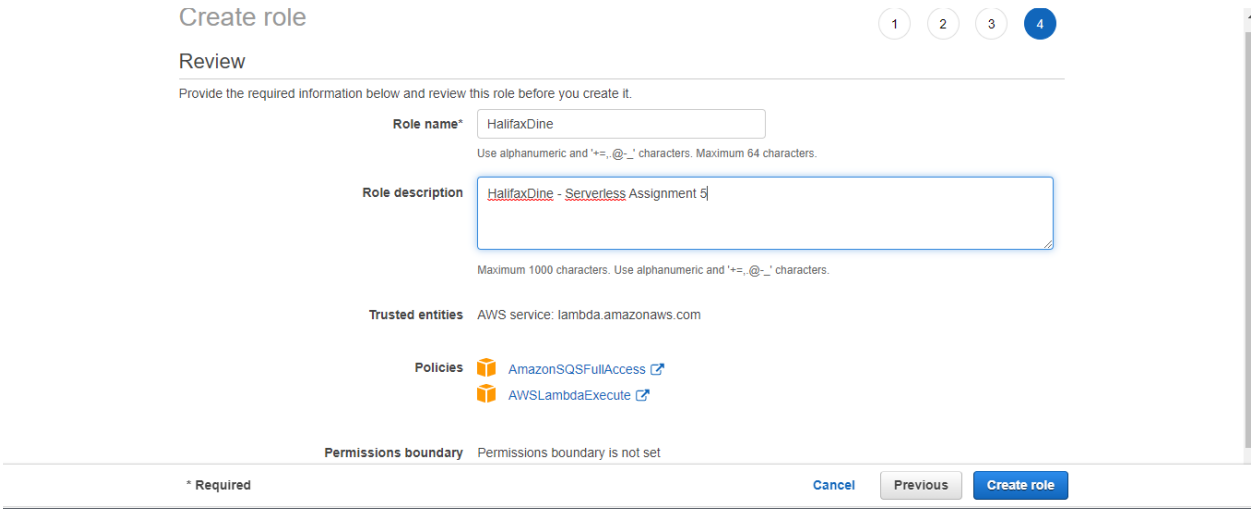
12. Pop up the tab in the browser is displayed as shown below.



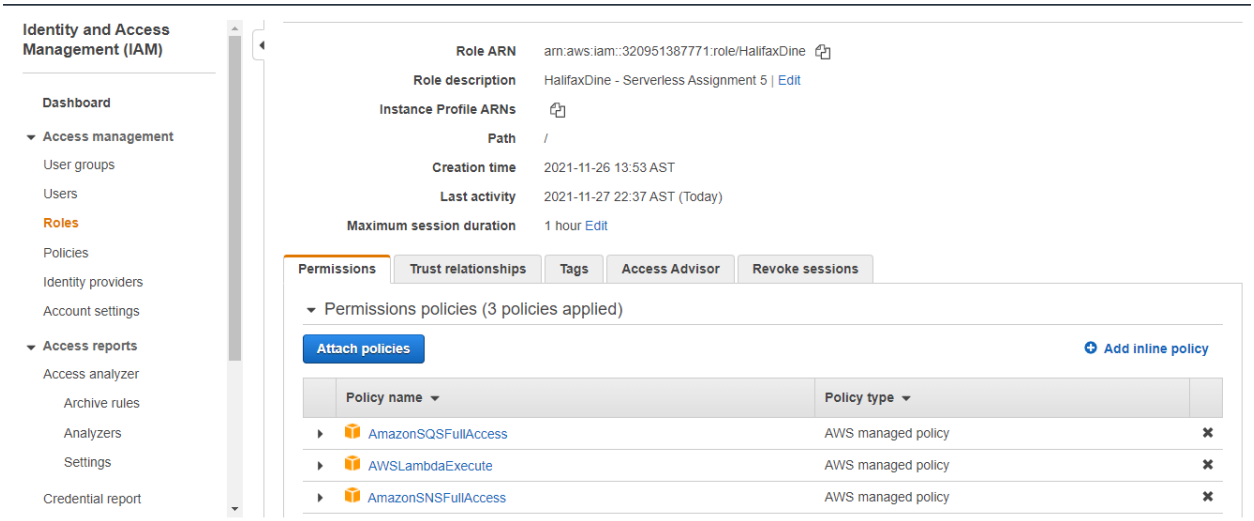
13. Go to IAM from the AWS console [3]. Create the role to have access to the SNS, SQS, and Lambda. Click on create the role.



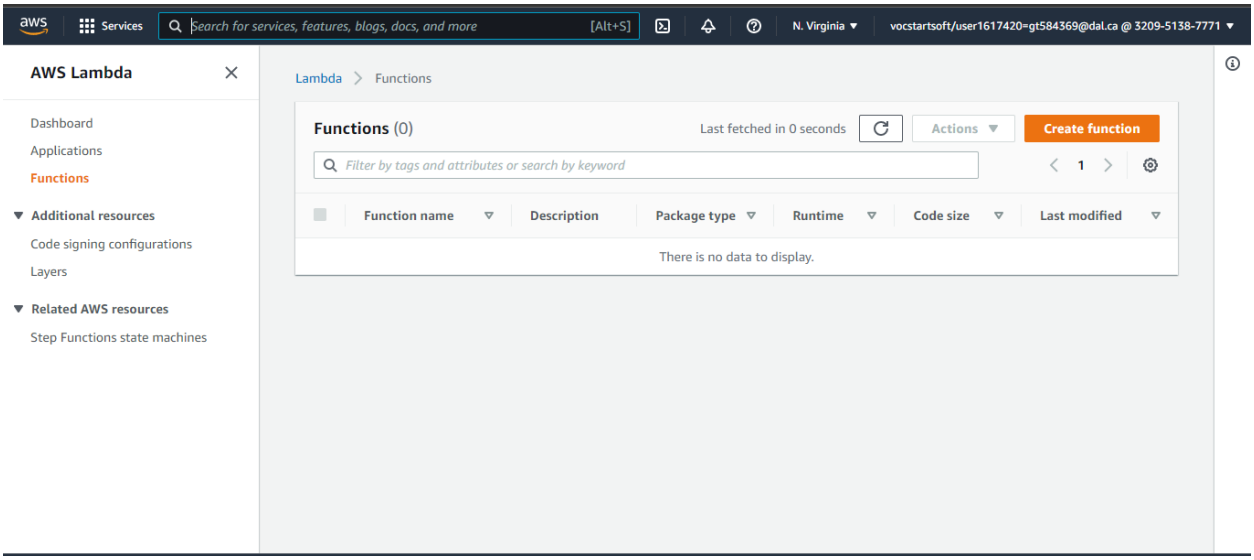
14. Give the name to the role and attach the policies required.



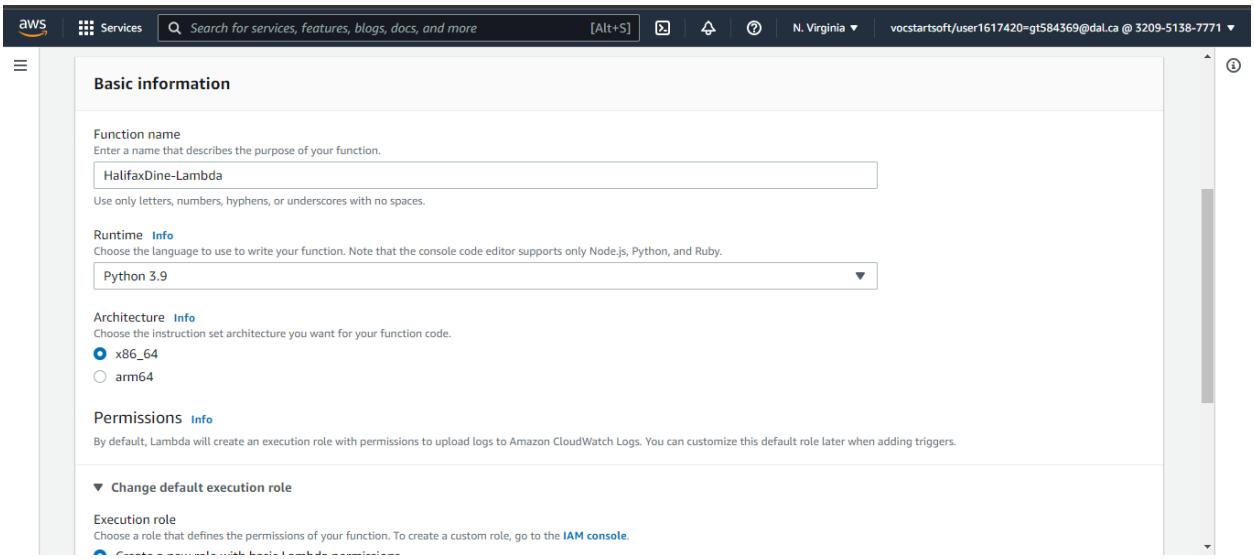
15. Full access policies for all three services are given for the role.



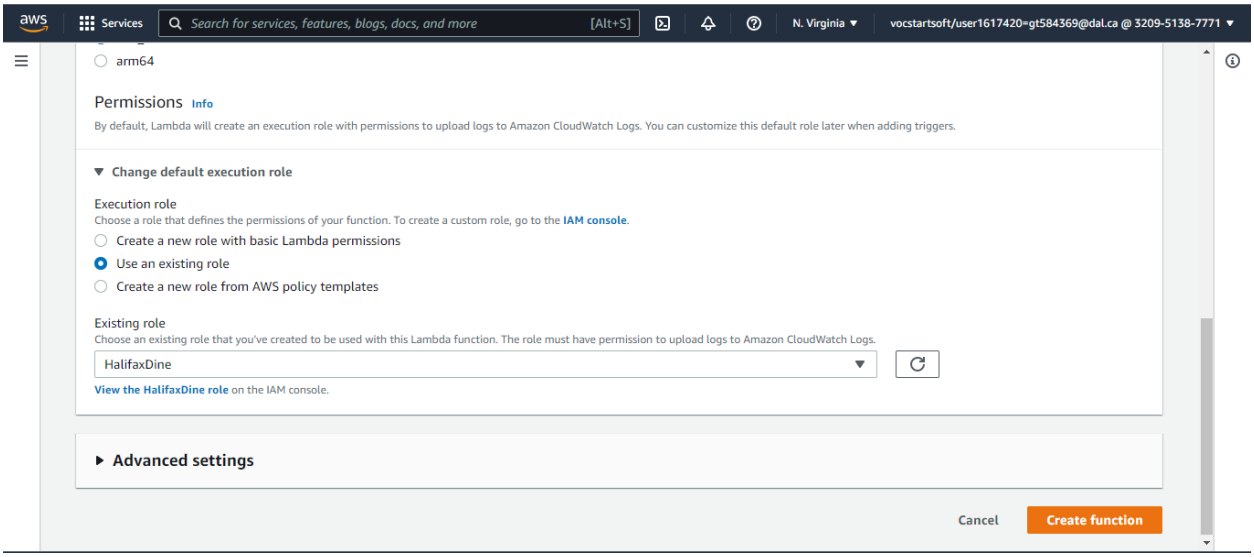
16. Go to AWS console and search for lambda [4]. Click on lambda and create the function.



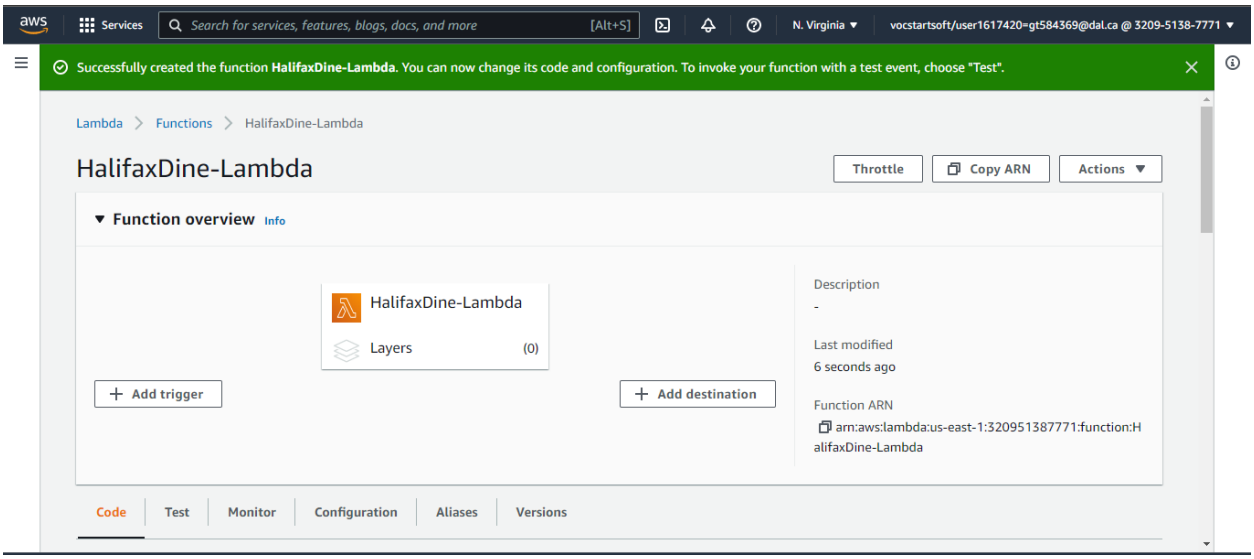
17. Give lambda function name and select the language as python 3.9.



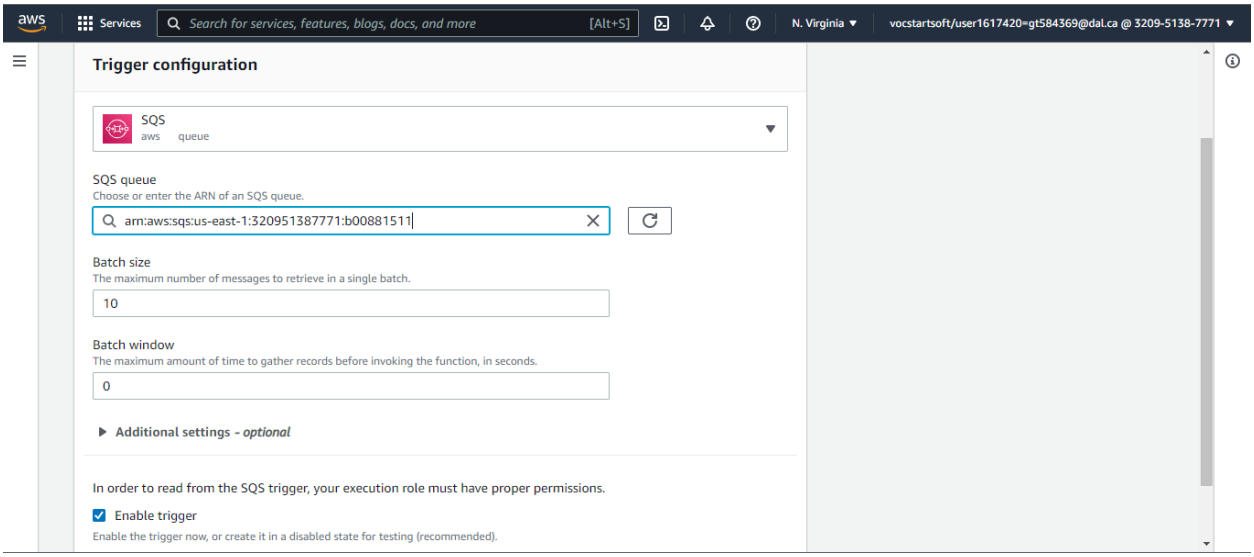
18. Select the existing role option and choose the role created in the above steps. Click on create function.



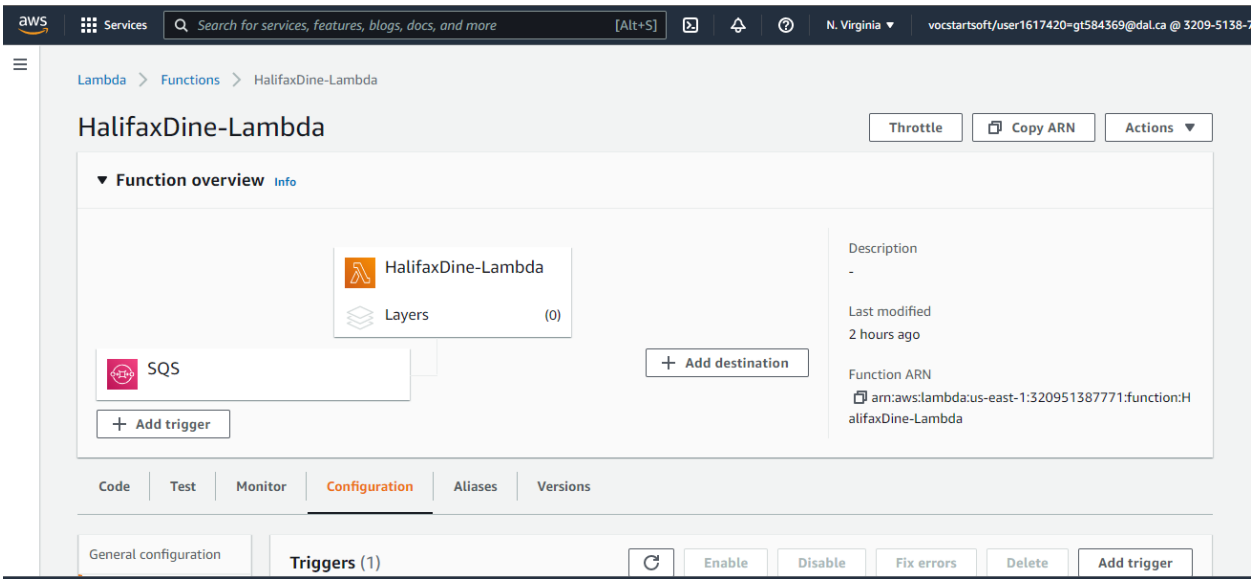
19. Click on add trigger to add SQS as a trigger to this lambda function.



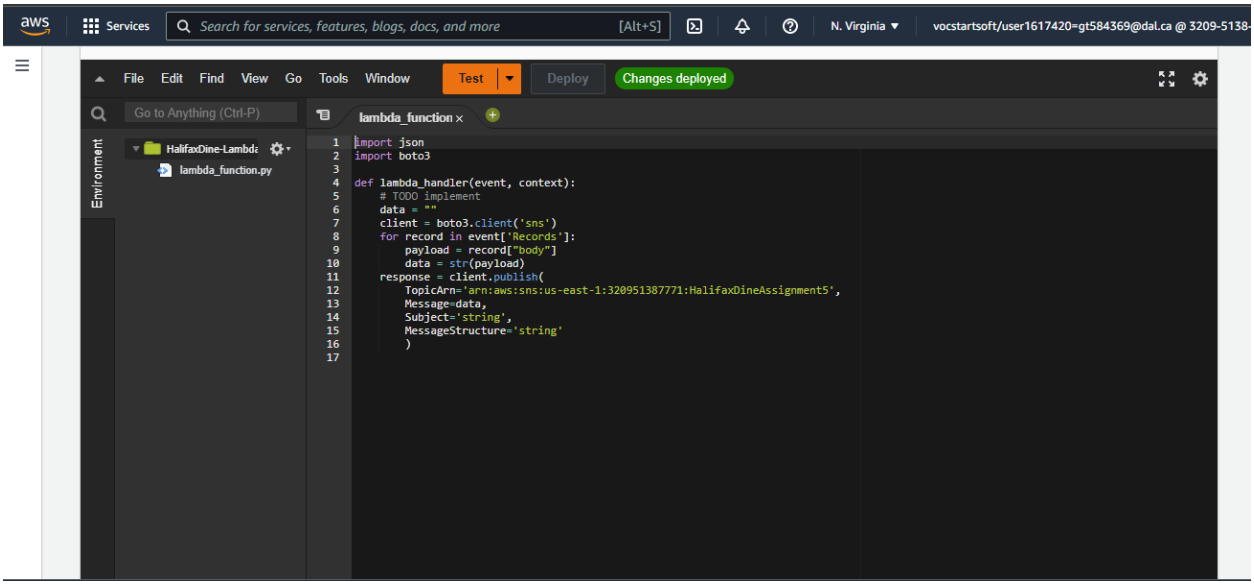
20. Search for SQS and select SQS. Select the SQS queue that is already created and save it.



21. Once the trigger is added, SQS is shown as the trigger for the lambda function.



22. Click on the code to write the code that sends the notification to the mail [5].



23. Once the code is ready. Click on deploy. Run the java program that inserts the data into the queue [6].

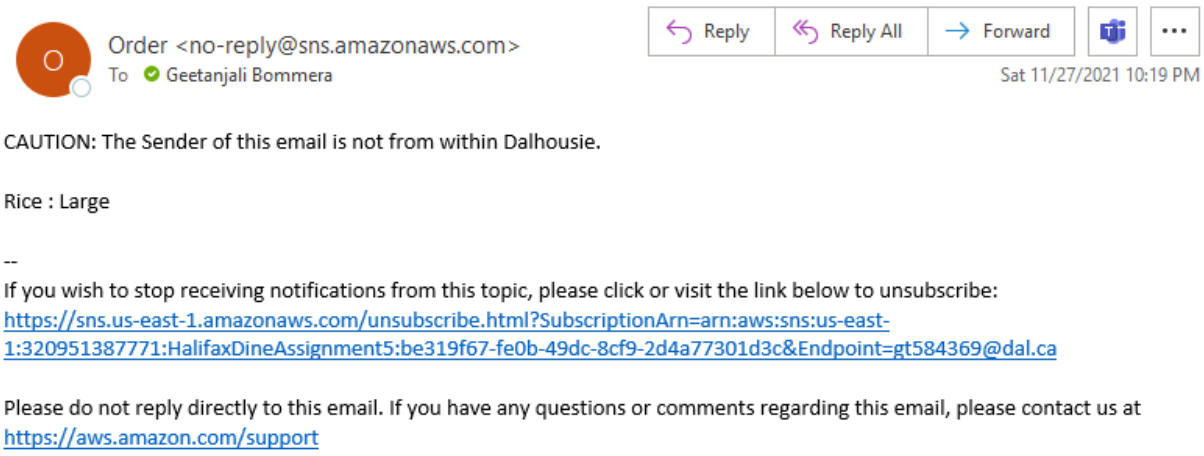
```
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.sqs.AmazonSQS;
import com.amazonaws.services.sqs.AmazonSQSClientBuilder;
import com.amazonaws.services.sqs.model.SendMessageRequest;
import java.util.Random;
import java.util.ArrayList;

public class HalifaxDine {
    public static void main(String[] args) {
        String access_key_id = "ASIAUVORRQZ55L3CQJWH";
        String secret_key_id = "kKgmFI0qJLN8zpBzUWLuoDcQQ/mHThRQETrCFz7a";
        String session_token =
"FwoGZXIvYXdzEPv//////////wEaDJZc0zEIS2QXWtYMsSK/AQ3g+Zx7wq+vQdSc6f6kBXqkP
CSaSy5O8ltq/1Sip20lBI/zCb7lZ6hEDdVaYrp6tI8m09yMmNcN7T9jodyYAECdnnQfwNILHGr
QEv3CHNV/f1tFwu/WX6zkwGY1MUHl9LMRkpf9L/2XA29+GWnrebaexmziTTQEJepOk7vvhiOY9
GghFF5toJ0nvlbGGPFDUu3NQbYk+ApO8w9PbAIS8w2uASxOELCohtNW6i6KkUhVgcrVe7qHQ6K
BI4zwiV1YKIiti40GMi1xhafOf8byM5D9rA8TYjXfpp62zuNH87wZ6zh8cGbNc7sUehXci04tz
izdNIs=";
        BasicSessionCredentials sessionCredentials = new
BasicSessionCredentials(
                access_key_id, secret_key_id, session_token);
        AmazonSQS sqs =
AmazonSQSClientBuilder.standard().withRegion(Regions.US_EAST_1)
                .withCredentials(new
AWSStaticCredentialsProvider(sessionCredentials))
                .build();
        createQueue(sqs);
    }

    private static void createQueue( AmazonSQS sqs) {
//        https://docs.aws.amazon.com/sdk-for-java/v1/developer-
guide/examples-sqs-message-queues.html
```

```
        String QUEUE_NAME = "b00881511";
        String queueUrl = sqs.getQueueUrl(QUEUE_NAME).getQueueUrl();
        ArrayList<String> quantity = new ArrayList<>();
        quantity.add("Small");
        quantity.add("Medium");
        quantity.add("Large");
        ArrayList<String> food_items = new ArrayList<>();
        food_items.add("Pizza");
        food_items.add("Salad");
        food_items.add("Rice");
        Random rand = new Random();
        for(int i =0;i<2;i++) {
            int index = rand.nextInt(3);
            String item = food_items.get(index);
            String size = quantity.get(index);
            String body = item + " : " + size;
            SendMessageRequest send_msg_request = new SendMessageRequest()
                    .withQueueUrl(queueUrl)
                    .withMessageBody(body)
                    .withDelaySeconds(300);
            sqs.sendMessage(send_msg_request);
        }
    }
}
```

24. Once the data is inserted the mail notifications are sent with a delay of 5 seconds every time.



3. References

[1] Amazon, "Amazon Simple Queue Service," [Online]. Available: https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-configure-create-queue.html. [Accessed 26 November 2021].

[2] Amazon, "Amazon Simple Notification Service," [Online]. Available: https://console.aws.amazon.com/sns/v3/home?region=us-east-1#/homepage. [Accessed 26 November 2021].

[3] Amazon, "Identity and Access Management (IAM)," [Online]. Available: https://console.aws.amazon.com/iam/home#/roles/HalifaxDine?section=permissions. [Accessed 26 November 2021].

[4] Amazon, "AWS Lambda," [Online]. Available: https://console.aws.amazon.com/lambda/home?region=us-east-1#/functions. [Accessed 26 November 2021].

[5] Amazon, "Sample Amazon SQS function code," [Online]. Available: https://docs.aws.amazon.com/lambda/latest/dg/with-sqs-create-package.html#with-sqs-example-deployment-pkg-python. [Accessed 26 November 2021].

[6] Amazon, "Creating an Amazon SQS queue," [Online]. Available: https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-configure-create-queue.html. [Accessed 26 November 2021].

4. Git Link

https://git.cs.dal.ca/bommera/csci-5410-f2021-b00881511-geetanjali-bommera