Contents

[Purpose 2](#_Toc55677153)

[General Diagram 2](#_Toc55677154)

[Prerequisites 2](#_Toc55677155)

[Lab 8A: DynamoDB Streams with lambda to notify updates using CLI 2](#_Toc55677156)

[Create Table and able Streams 2](#_Toc55677157)

[Create SNS Topic and subscribe an email 3](#_Toc55677158)

[Create Lambda Execution role and apply policy to it 4](#_Toc55677159)

[Create Lambda Function and Test it 5](#_Toc55677160)

[Vinculate Lamdba source as DynamoDB Streams and put an item on DynamoDB 7](#_Toc55677161)

[Review Cloudwatch (Optional) 9](#_Toc55677162)

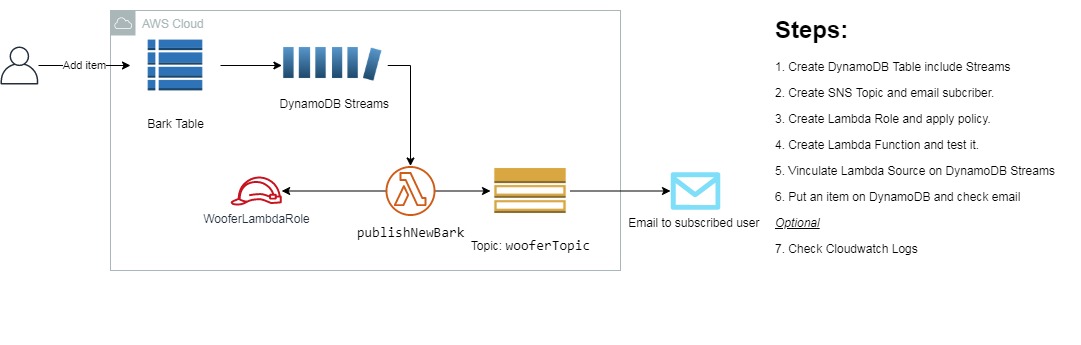
[Clean Resources 10](#_Toc55677163)

[Evidences to send 11](#_Toc55677164)

# Purpose

Use DynamoDB Streams feature to call a Lambda function and then, use the notification system using email.

# General Diagram



Create a DynamoDB Table with Streams Feature to send an email notification when a new item is created. It uses DynamoDB Streams, Lambda Function, IAM Role to Lambda and SNS.

# Prerequisites

Labs1c1 have to be done and the context for Administrative user have to activated on Command Line Session.

This lab was adapted from <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.Lambda.Tutorial.html>

To acquire knowledge about: DynamoDB, DynamoDB Streams, AWS SNS, AWS Lambda and Cloudwatch Logs.

# Lab 8A: DynamoDB Streams with lambda to notify updates using CLI

## Create Table and able Streams

rem SECCION CREAR TABLA PARA DYNAMODB STREAM

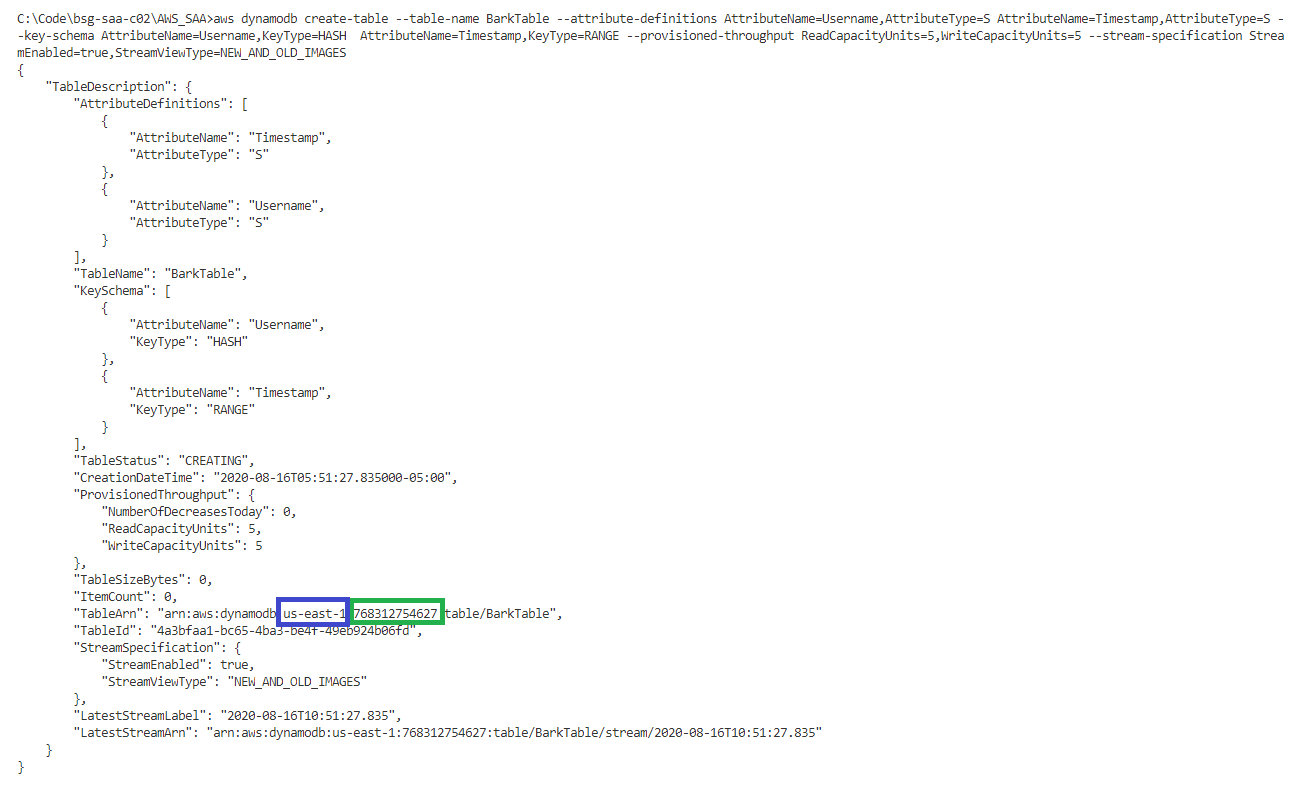
rem Crear la base de datos original, asignarle RCU y WCU bajos para este caso son 5 pero si quieres hacer mas pruebas puedes llegar hasta 25 en la capa gratuita.

rem Crearle Partition and Sort Key, recordar que se llaman HASH y RANGE key dentro AWS tambien. Recordar que la llave compuesta no permite escribir ambos valores iguales para un registro

rem Activarle el Stream a la tabla para realizarle las acciones de eventos al Lambda.

aws dynamodb create-table --table-name BarkTable --attribute-definitions AttributeName=Username,AttributeType=S AttributeName=Timestamp,AttributeType=S --key-schema AttributeName=Username,KeyType=HASH  AttributeName=Timestamp,KeyType=RANGE --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 --stream-specification StreamEnabled=true,StreamViewType=NEW\_AND\_OLD\_IMAGES

rem Obtener la region y el AccountID. Aqui se almacenan la region y el AccountID en mi caso es el us-east-1 y 768312754627



## Create SNS Topic and subscribe an email

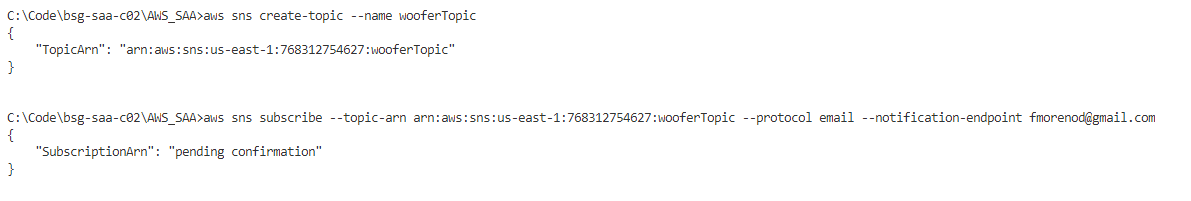
rem Crear SNS Topic. Es necesario crear un topic para al enviar un evento se publicado en los correos suscritos.

aws sns create-topic --name wooferTopic

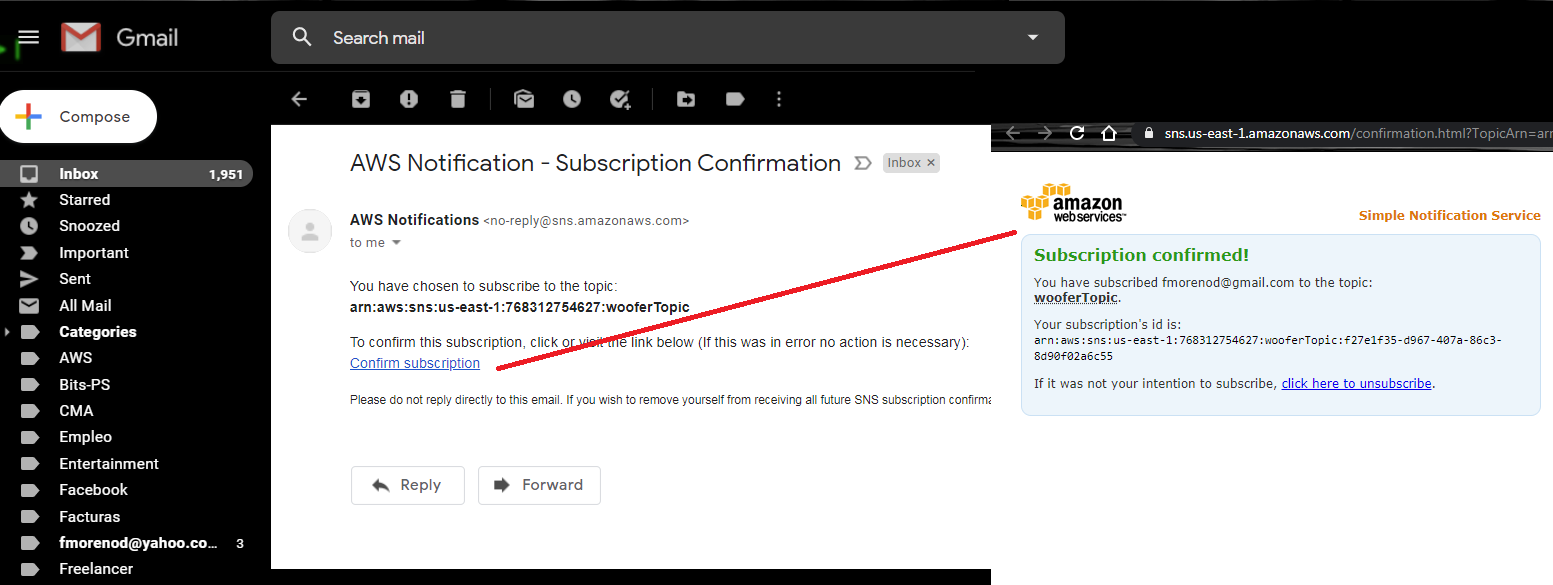
rem Subscribir el correo electronico al SNS Topic. Modificar region, accountID y el correo electronico a su correo personal.

aws sns subscribe --topic-arn arn:aws:sns:us-east-1:768312754627:wooferTopic --protocol email --notification-endpoint fmorenod@gmail.com

rem Ir a su correo electronico y aceptarlo la suscripcion.



rem Evidencia: Copiar el correo electronico de la suscripcion.



## Create Lambda Execution role and apply policy to it

rem SECCION CREAR ROLE, LA FUNCION LAMBDA Y PROBARLA

rem Crear el Role para ejecucion del Lambda y su alcance basado en Trusted Entity. El archivo json debe ser visualizado desde la consola.

aws iam create-role --role-name WooferLambdaRole --path "/service-role/" --assume-role-policy-document file://trust-relationship.json

rem Asignarle permisos al role. Modificar el archivo json role-policy.json para ponerle region y account ID: en mi caso era us-east-1 y 768312754627

rem En este caso asignarles los permisos especificos al role de los recursos a usar: DynamoDB Streams, SNS - Notificaciones, y ejecucion del lambda.

aws iam put-role-policy --role-name WooferLambdaRole --policy-name WooferLambdaRolePolicy --policy-document <file://role-policy.json>



## Create Lambda Function and Test it

rem Modificar el archivo publishNewBark.js con la region y accountID correcto y luego crear el archivo comprimido, aqui uso "zip" que viene incluido en mi windows.

zip publishNewBark.zip publishNewBark.js

rem Confirmar que el role fue creato y obtener el role ARN para crear la funcion con el role adecuado.

aws iam get-role --role-name WooferLambdaRole

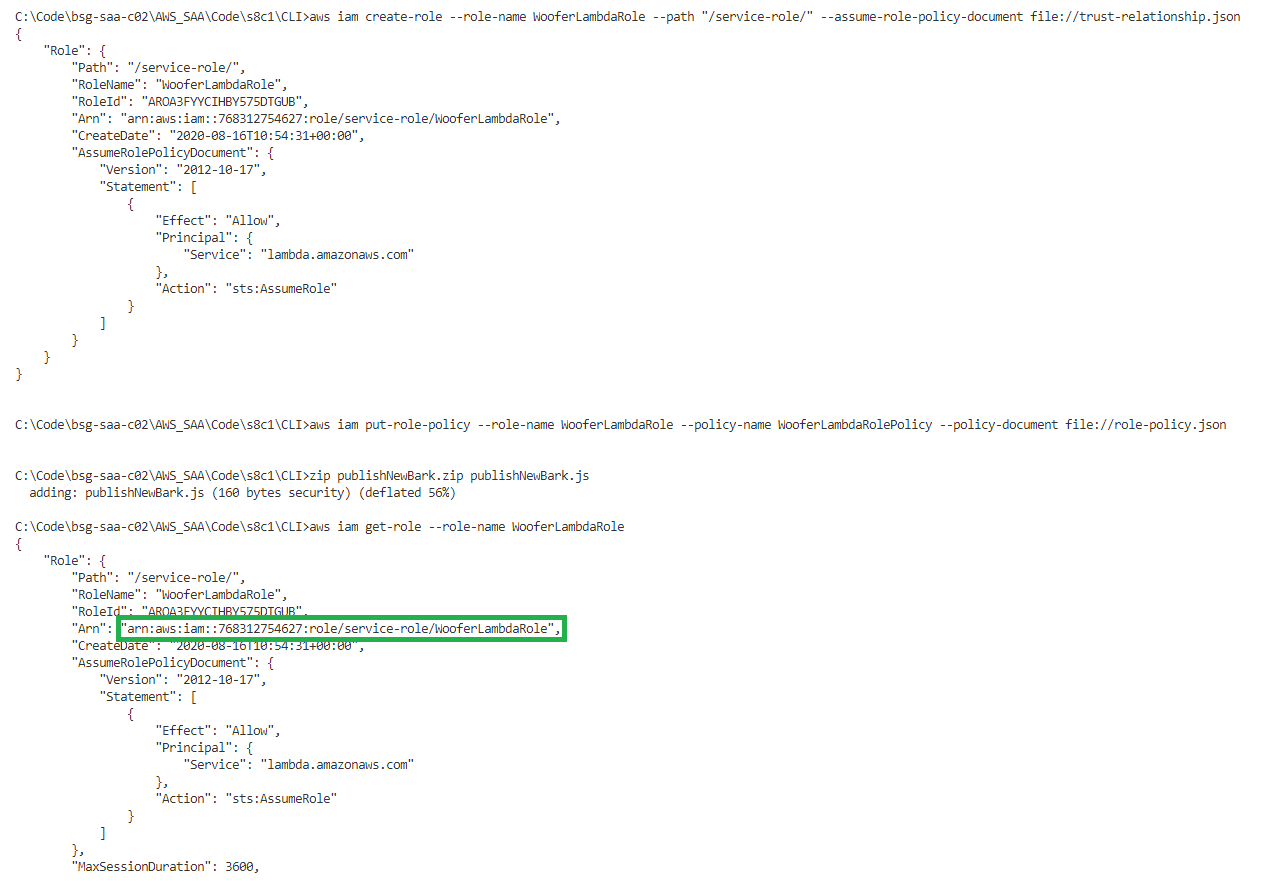
rem Crear la funcion lambda: con codigo, Role ARN (obtenido del paso anterior) y enviar el codigo comprimido, version de lenguaje nodejs.10 y funcion a ejecutar

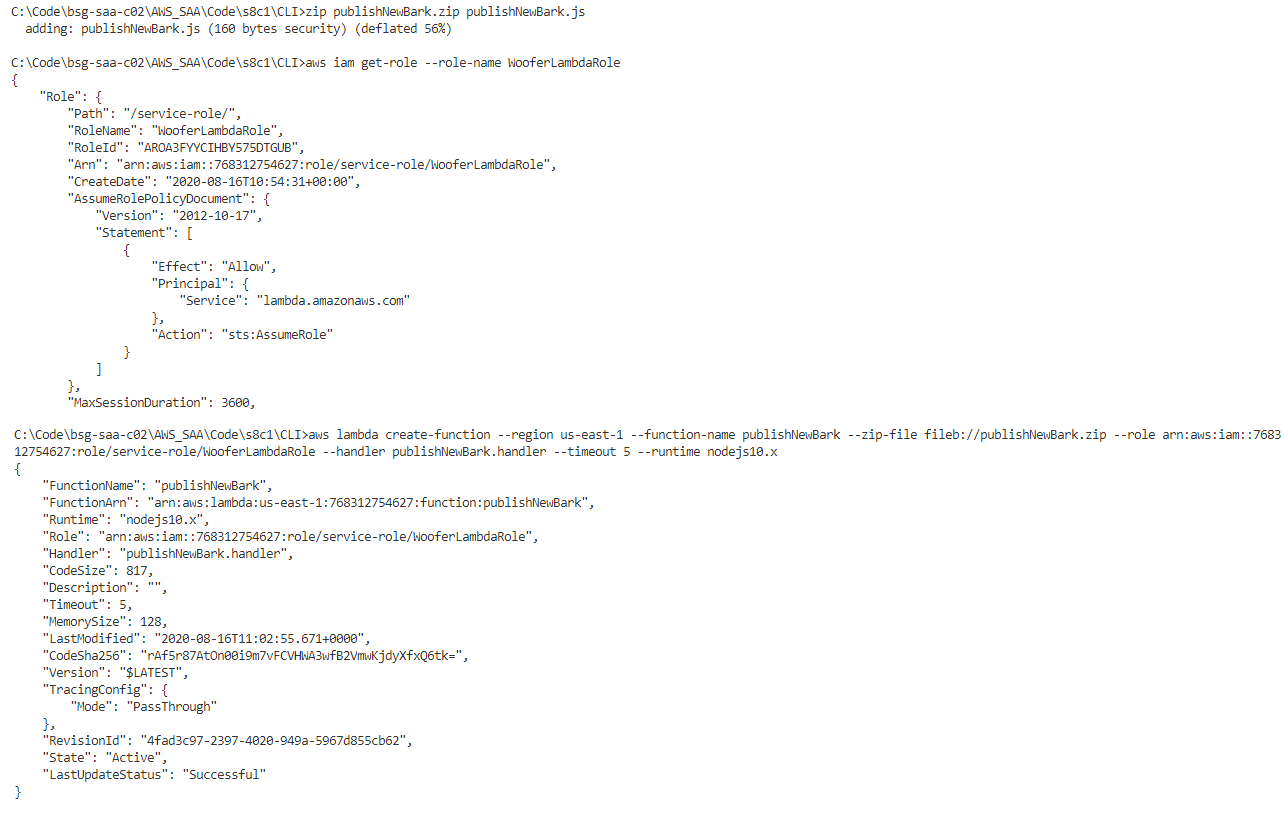
aws lambda create-function --region us-east-1 --function-name publishNewBark --zip-file fileb://publishNewBark.zip --role arn:aws:iam::768312754627:role/service-role/WooferLambdaRole --handler publishNewBark.handler --timeout 5 --runtime nodejs10.x

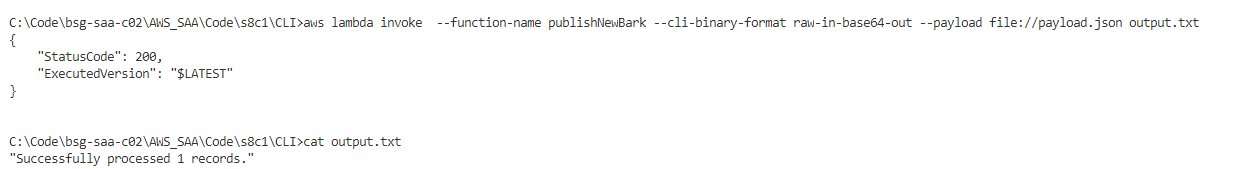
rem Probar la funcion creada enviando un json (payload.json) abriendo el achivo de salida (output.txt)

aws lambda invoke --function-name publishNewBark --cli-binary-format raw-in-base64-out --payload file://payload.json output.txt

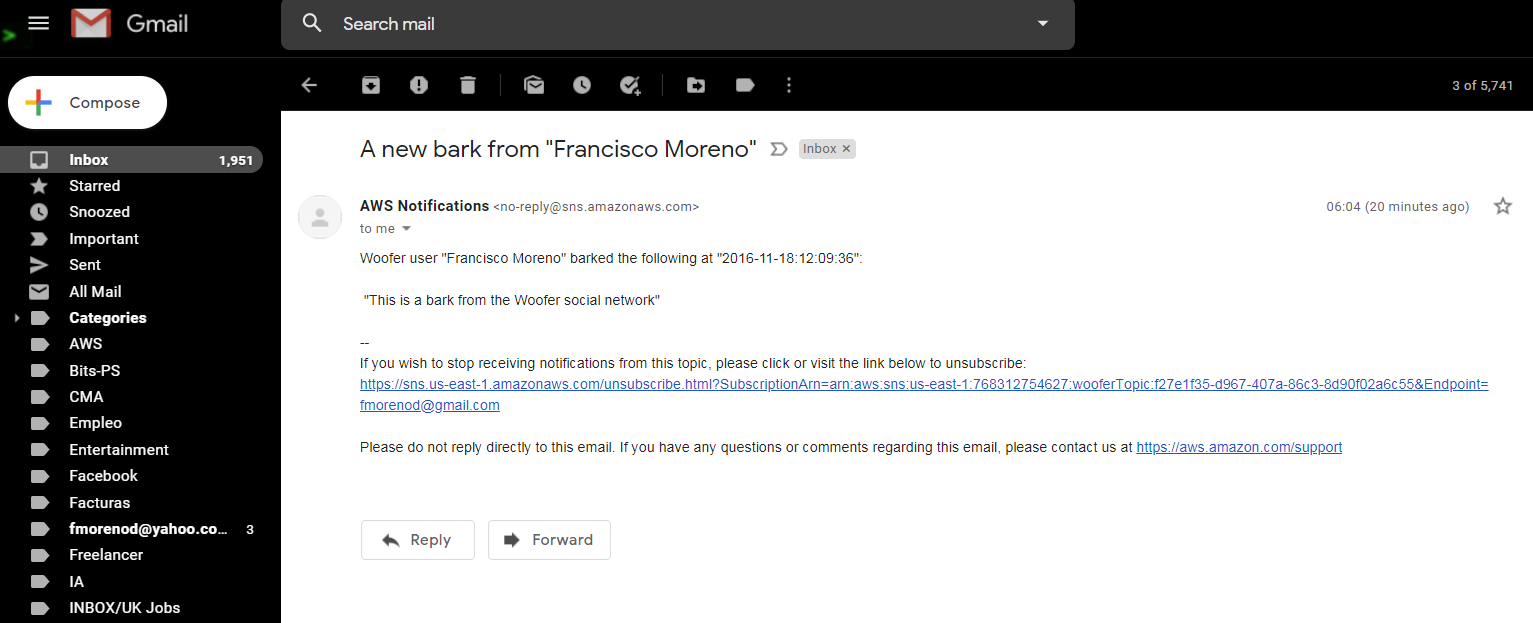
rem Comprobar la respuesta 200 en el StatusCode y que el mensaje del archivo output.json asi como el correo electronico con la notificacion







rem Evidencia: Copiar el correo electrónico de la prueba.



## Vinculate Lamdba source as DynamoDB Streams and put an item on DynamoDB

rem SECCION VINCULAR FUENTE DEL LAMBDA COMO DYNAMODB STREAMS Y ALMACENAR UN REGISTRO AL DYNAMOD

rem Obtener el Latest Stream ARN (Ultimo eventos Stream generado del campo LatestStreamArn)

aws dynamodb describe-table --table-name BarkTable

rem Crear el origen de llamada al Lambda, al reemplazar event-source con el LatestStreamArn obtenido del paso anterior.

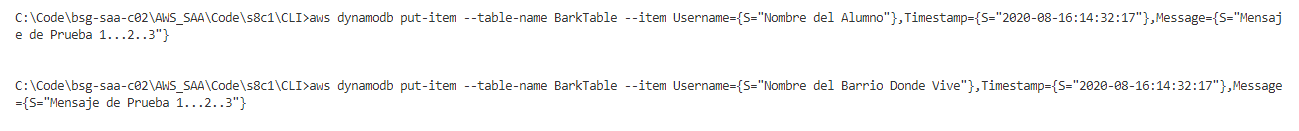
aws lambda create-event-source-mapping --region us-east-1 --function-name publishNewBark --event-source arn:aws:dynamodb:us-east-1:768312754627:table/BarkTable/stream/2020-08-16T10:51:27.835  --batch-size 1 --starting-position TRIM\_HORIZON

rem Crear un registro en DynamoDB

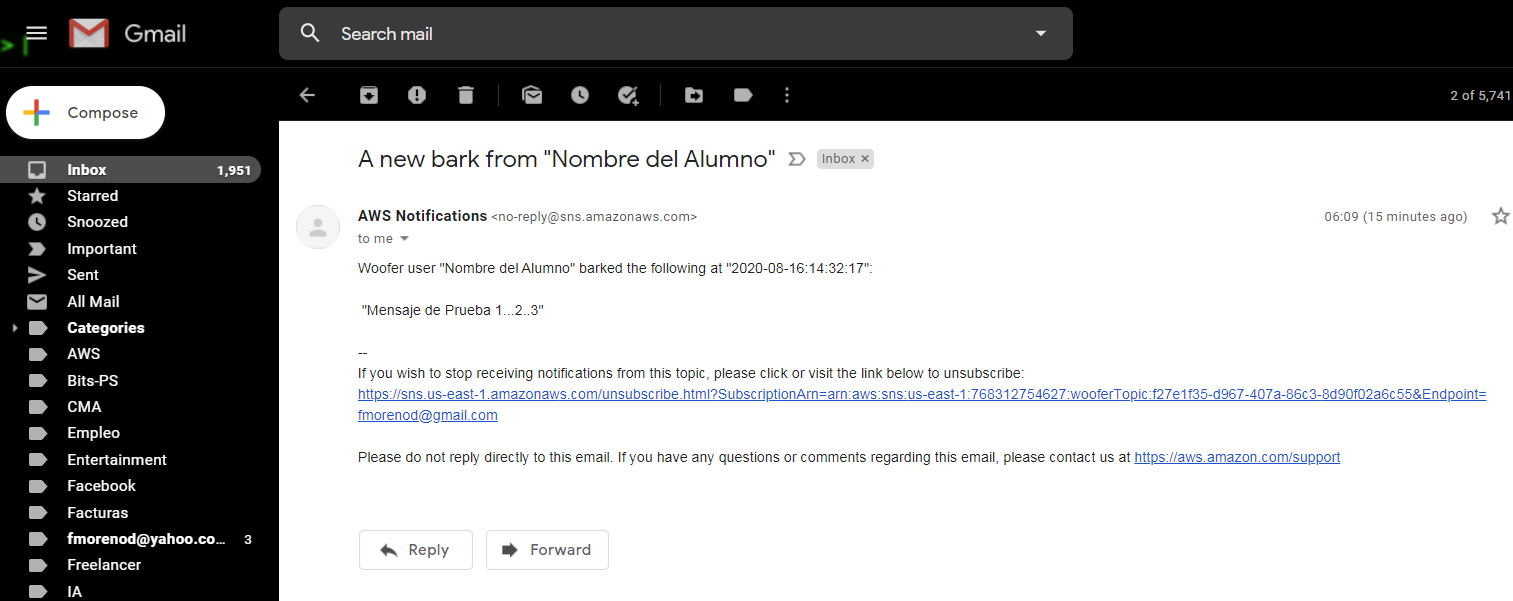
aws dynamodb put-item --table-name BarkTable --item Username={S="Nombre del Alumno"},Timestamp={S="2020-08-16:14:32:17"},Message={S="Mensaje de Prueba 1...2..3"}

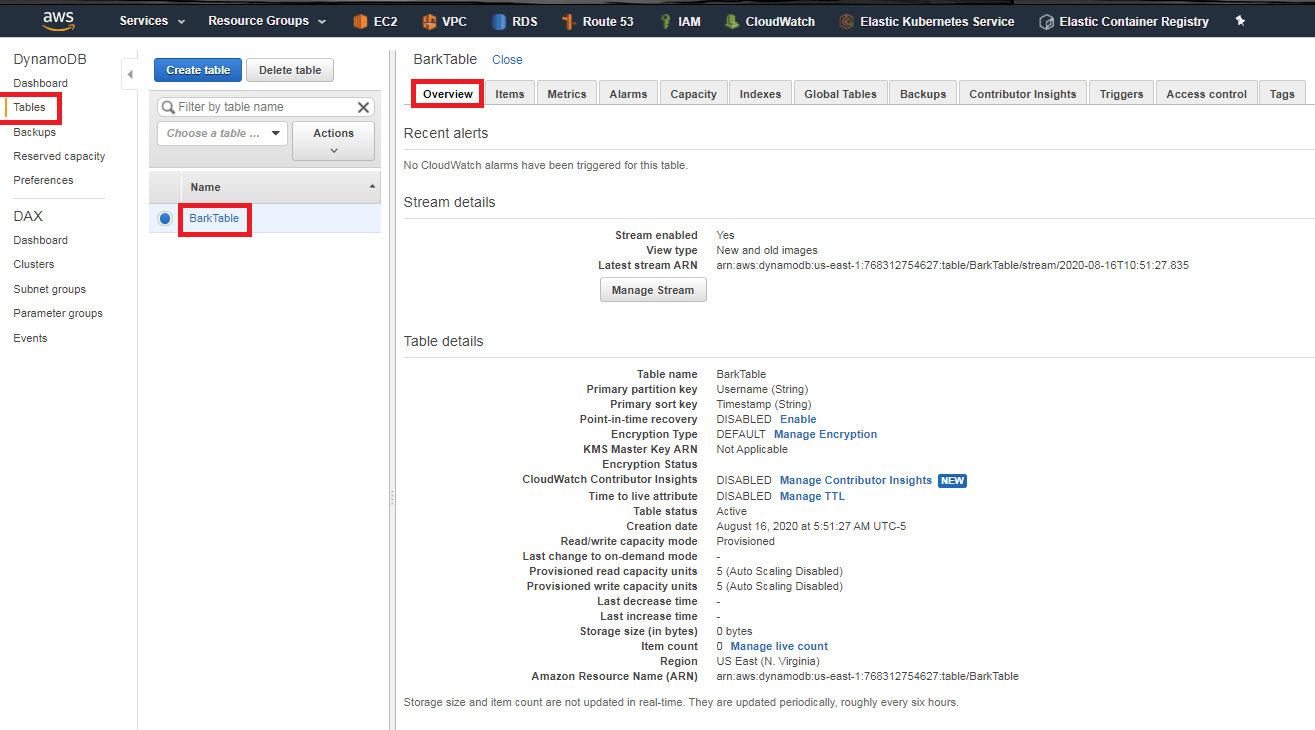
aws dynamodb put-item --table-name BarkTable --item Username={S="Nombre del Barrio Donde Vive"},Timestamp={S="2020-08-16:14:32:17"},Message={S="Mensaje de Prueba 1...2..3"}





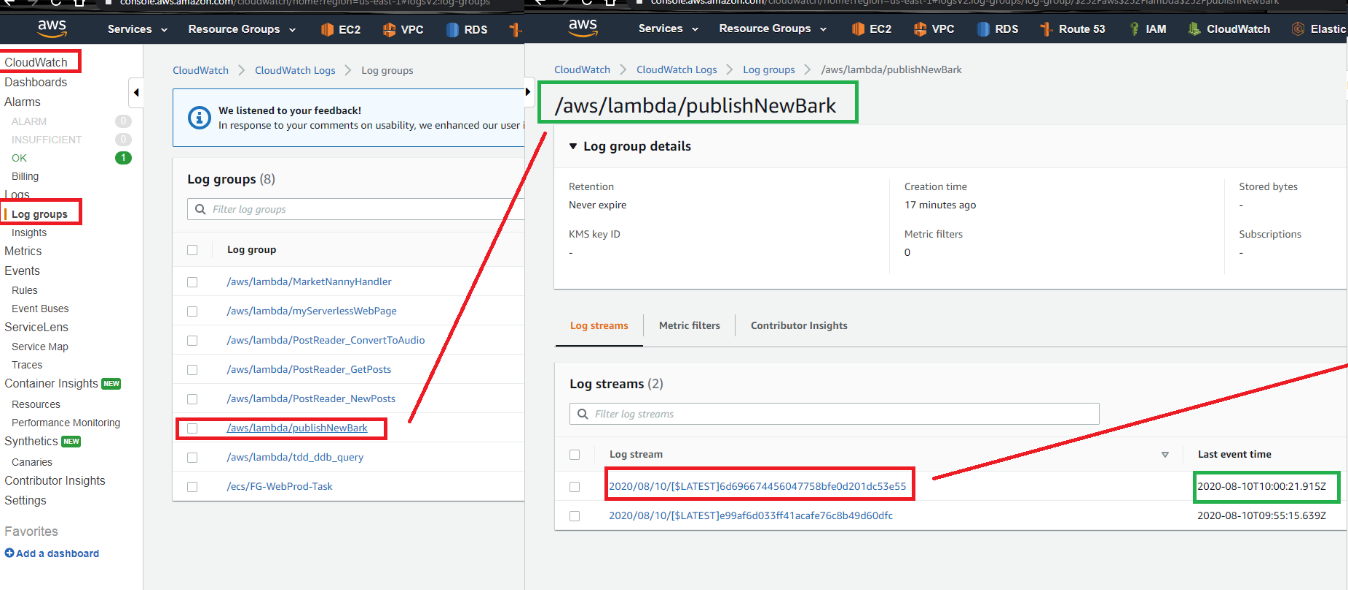
rem Evidencia: Copiar el correo electronico de la prueba y e ir a AWS Console y obtener un pantallazo de la configuración del DynamoDB Table como se vé en el ejemplo.

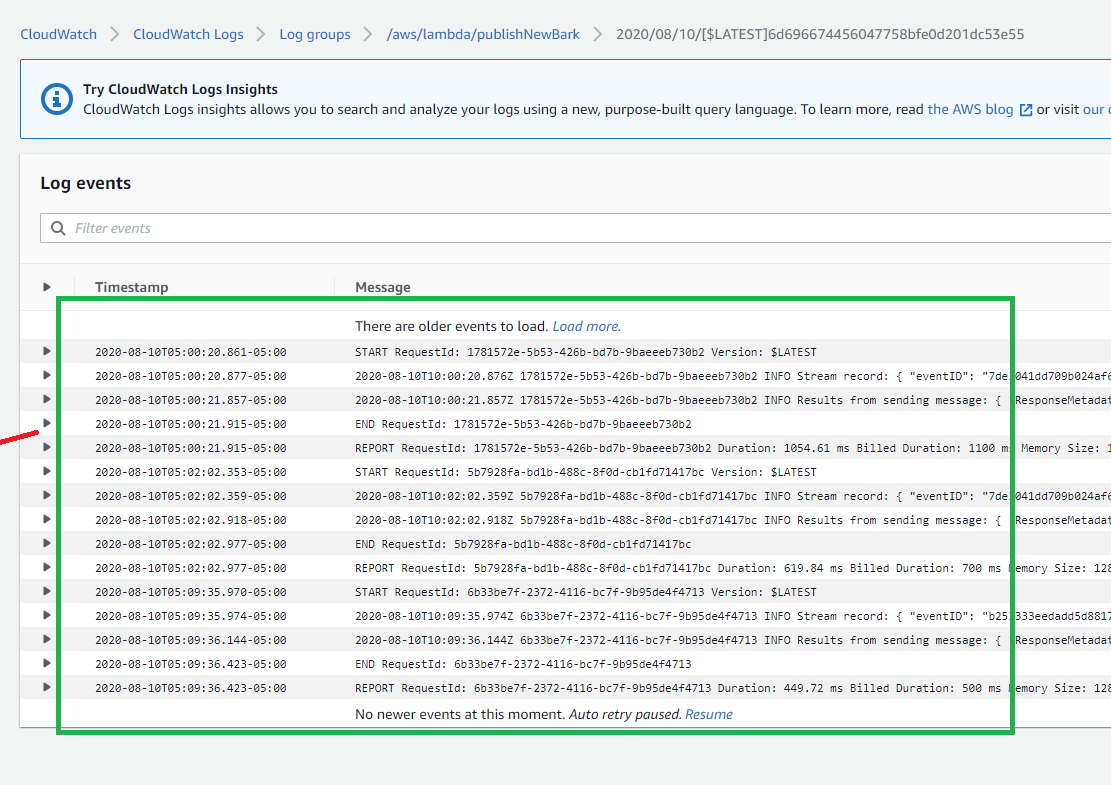




## Review Cloudwatch (Optional)

You can visit Cloudwatch to have more information about Lambda compsuntiom





## Clean Resources

All resources are serverless, but if you like to delete:

DynamoDB: Table

Lambda: Function

IAM: Role

SNS: Topic

# Evidences to send

To have a review, the student has to send some screenshots to instructor email:

1. Subscription Email and acceptance form from SNS Topic. It’s similar to email confirmation picture from [Create SNS Topic and subscribe an email](#_Create_SNS_Topic) section.
2. Email from Lambda Test. It’s similar picture from [Create Lambda Function and Test it](#_Create_Lambda_Function) section.
3. Email from Lambda Execution and DynamoDB Table Configuration. Those pictures are similar to last 2 picture from [Vinculate Lamdba source as DynamoDB Streams and put an item on DynamoDB](#_Vinculate_Lamdba_source) section.