

177- [JAWS] - Lab - [Reto] Ejercicio de AWS Lambda

# **Datos Generales:**

Nombre: Tomás Alfredo Villaseca Constantinescu

País: Chile

Fecha: 06/11/2023

Contacto: tomas.villaseca.c@gmail.com

Después de completar este laboratorio, usted será capaz de hacer lo siguiente:

- Crear una función Lambda para contar el número de palabras de un archivo de texto.
- Configurar un bucket de Amazon S3 para invocar una función de Lambda cuando se cargue un archivo de texto en el bucket de S3.
- Crear un Topic de Amazon SNS para informar del recuento de palabras en un email.

# Desafío

- (1) Crear una función Lambda para contar el número de palabras en un archivo de texto.
  - Utilizar la AWS Management Console para desarrollar una función Lambda en Python y crear los recursos necesarios de la función.
  - Informe del recuento de palabras en un correo electrónico utilizando un tema SNS.
  - Formato del mensaje de respuesta:

The word count in the <textFileName> file is nnn.

- Asunto del mensaje de correo electrónico → Resultado del recuento de palabras.
- Invoca automáticamente la función cuando el archivo de texto se cargue en un bucket de S3.
- (2) Pruebe la función cargando algunos archivos de texto de muestra con diferentes recuentos de palabras en el bucket de S3.

#### Paso 1: S3 → Create Bucket

- Bucket Name = labbucket224466
- AWS Region = us-west-2

#### Bucket name

#### labbucket224466

Bucket name must be unique within the global namespace and fol

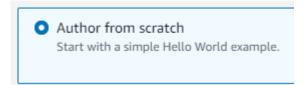
#### AWS Region

US West (Oregon) us-west-2

	Name	▲ AWS Region
0	labbucket224466	US West (Oregon) us-west-2

### Paso 2: Lambda → Functions → Create Function → Basic Information

Author from scratch



Function Name = WordCount

# Function name

Enter a name that describes the purpose of your function.

WordCount

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime → Python 3.9

#### Runtime Info

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.9

#### Paso 3: Create Function → Permissions

Use an existing role → LambdaAccessRole

#### Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console 🔀

- O Create a new role with basic Lambda permissions
- Use an existing role
- O Create a new role from AWS policy templates

#### Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

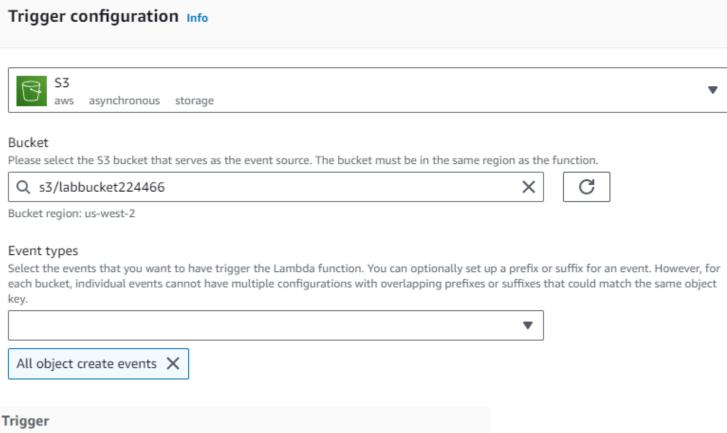
LambdaAccessRole

View the LambdaAccessRole role 🔀 on the IAM console.

<b>~</b>	Function name
<b>✓</b>	WordCount

# Paso 4: Lambda → Create Trigger → Add Trigger

- Select Source → S3 → labbucket224466
- Event Types → All object create events





# S3: labbucket224466

arn:aws:s3:::labbucket224466

#### ▼ Details

Bucket arn: arn:aws:s3:::labbucket224466

Event types: s3:ObjectCreated:\*

Notification name: e9a0872c-cd60-4aeb-a3a2-89454d0957fc

Service principal: s3.amazonaws.com

Source account: 712310588139

Statement ID: lambda-c551ed8c-e595-4969-94a2-8f78388dca15

# Paso 5: SNS → Topic → Create Topic

Type → Standard

# Standard

- · Best-effort message ordering
- At-least once message delivery
- · Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints
- Name = WordCountTopic

#### Name

0

#### WordCountTopic

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (\_).

Name WordCountTopic

# Paso 6: SNS → Topic → WordCountTopic → Create Subscription

- Topic ARN → WordCountTopic
- Protocol → Email
- Endpoint → tomas.villaseca.c@gmail.com

#### Topic ARN

Q arn:aws:sns:us-west-2:712310588139:WordCountTopic

#### Protocol

The type of endpoint to subscribe

Email

### Endpoint

An email address that can receive notifications from Amazon SNS.

tomas.villaseca.c@gmail.com

You have chosen to subscribe to the topic:

arn:aws:sns:us-west-2:712310588139:WordCountTopic

\*\*\*

To confirm this subscription, click or visit the link below (If this was in error no action is necessary): Confirm subscription



Simple Notification Service

# Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:

arn:aws:sns:us-west-2:712310588139:WordCountTopic:f02e2117-aaba-4ac9-af59afbbc8781846

If it was not your intention to subscribe, click here to unsubscribe.

Paso 7: Lambda → Functions → WordCount → Code → Code Source

```
8
     lambda function × Environment Vari ×
    import boto3
 1
 2
     def lambda_handler(event, context):
 3
         # Obtener el nombre del archivo de S3
 4
         bucket_name = event['Records'][0]['s3']['bucket']['name']
 5
         file_name = event['Records'][0]['s3']['object']['key']
 6
         # Crear una instancia del cliente de S3
         s3 = boto3.client('s3')
 7
         # Leer el archivo de S3
 8
 9
         response = s3.get object(Bucket=bucket name, Key=file name)
         content = response['Body'].read().decode('utf-8')
10
11
         # Contar la cantidad de palabras en el archivo
         word_count = len(content.split())
12
         # Crear una instancia del cliente de SNS
13
         sns = boto3.client(|'sns'|)
14
15
         # Enviar el correo usando el tema de SNS
         topic_arn = 'arn:aws:sns:us-west-2:712310588139:WordCountTopic'
16
17
         message = f"El archivo {file name} tiene {word_count} palabras."
         sns.publish(TopicArn=topic_arn, Message=message)
18
```

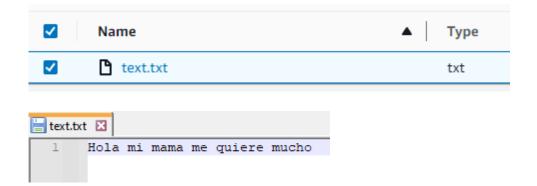
Paso 8: S3 → labbucket224466 → Upload

# No objects

You don't have any objects in this bucket.

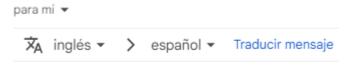


Add files → test.txt



Paso 9: Revisar correo electrónico → Notificación SNS

### **AWS Notifications**



El archivo text.txt tiene 6 palabras.

# Laboratorio Completado

