

P254 MVP de Rekognition en AWS

Para

LEN

Versión 1.0

Presentado por:

Applying Consulting S.A.C

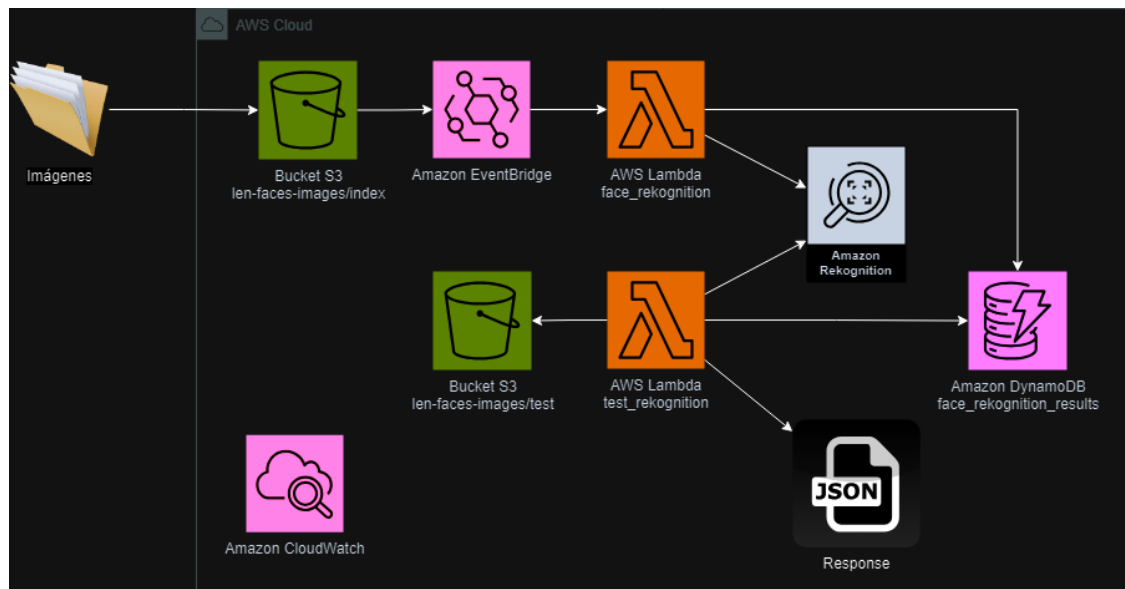
2024

Contenido

1. Diagrama de arquitectura.....	3
2. Implementación de Rekognition.....	5
2.1. Función Lambda de creación de colección	5
2.2. Bucket S3 para almacenar las imágenes	7
2.3. Función Lambda de indexado de rostros	9
2.4. Tabla en DynamoDB.....	10
3. Pruebas de función Lambda.....	12
3.1. Subida de archivo a S3.....	12
3.2. Función Lambda de reconocimiento de rostro.....	13
4. Infraestructura como código (IaC)	15
4.1. Plantilla CloudFormation	15

1. Diagrama de arquitectura

Todos los servicios y recursos implementados siguen el siguiente diagrama de arquitectura:



N°	Nombre del servicio	Descripción
1	AWS Lambda	Es un servicio serverless utilizado para crear 3 funciones: una para crear una colección de Rekognition, para realizar el indexado de los rostros dentro de las imágenes y la ultima para realizar el análisis en base a un rostro que se quiera identificar.
2	Amazon EventBridge	Es un servicio utilizado para que ejecute la función lambda de forma automática cada vez que se suban imágenes al bucket S3.
3	Amazon CloudWatch	Es un servicio utilizado para monitorear el funcionamiento de las funciones lambda.
4	Amazon S3	Es utilizado como servicio de almacenamiento en la nube de las imágenes que servirán tanto para el indexado de rostros como para el análisis.
5	Amazon Rekognition	Es un servicio de análisis de imágenes y videos basado en machine learning que se usará para identificar rostros de personas en base a las imágenes cargadas en S3 para su posterior análisis.

6	Amazon DynamoDB	Se utilizará una tabla de DynamoDB para almacenar ciertos datos relevantes al realizar el indexado de rostros en las imágenes.
---	-----------------	--

Flujo de trabajo:

- Se deben cargar las imágenes en el bucket S3 que se ha definido como almacén de datos.
- Una vez finalizado la carga, la función lambda de indexado de rostros se ejecutará de forma automática.
- Los resultados se podrán visualizar en la tabla de DynamoDB con la información relevante del indexado de rostros, esta información se puede ver casi de forma inmediata al terminar la carga de imágenes.
- Para realizar las pruebas se debe ejecutar de forma manual la función lambda de testeo y analizar el rostro de una persona al comparar con todos los rostros indexados.
- Los resultados mostrarán el Faceld con su porcentaje de confianza en la coincidencia de rostros. Además de la ruta S3 donde se encuentran dichas imágenes y las dimensiones de los rostros encontrados.

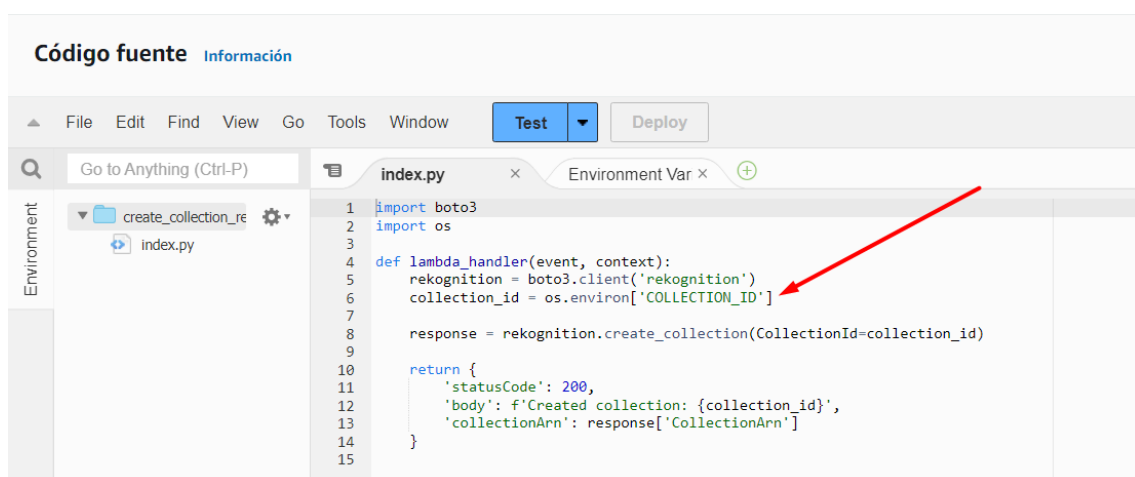
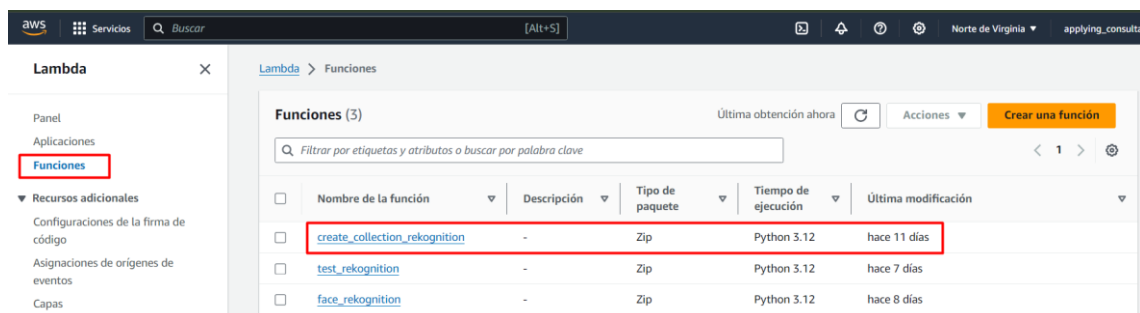
2. Implementación de Rekognition

2.1. Función Lambda de creación de colección

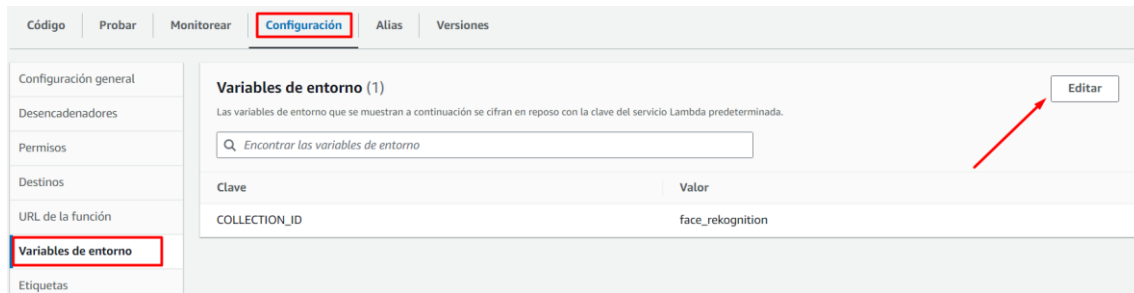
Esta función lambda se encarga de crear una nueva colección de Rekognition para almacenar información de los rostros detectados para luego ser comparados con los rostros que se quiera identificar.

Nota: Esta función lambda se debe ejecutar previamente a la carga de imágenes en S3, ya que, si no existe una colección de Rekognition, la función lambda de indexado devolverá un error de que necesita el nombre de la colección.

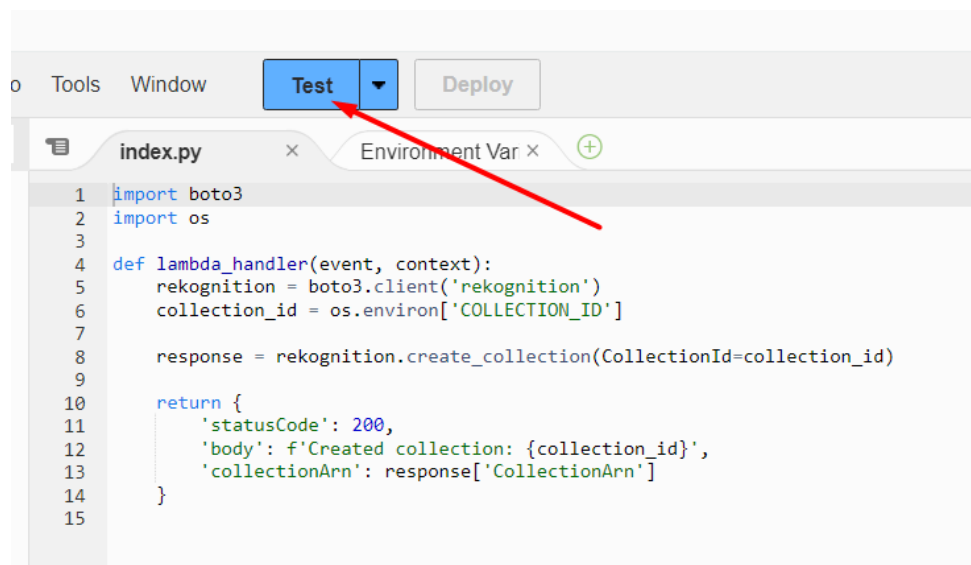
- Características
 - Nombre: **create_collection_rekognition**
 - Lenguaje: Python 3.12
 - Arquitectura: x86_64
- Variables de entorno
 - COLLECTION_ID: nombre de la colección que se va a crear para almacenar información de los rostros detectados.



Para futuras ejecuciones, si se desea crear una nueva colección, se debe actualizar la variable de entorno "COLLECTION_ID".



Una vez actualizada la variable de entorno, puede ejecutar la función lambda haciendo clic en la opción **“Test”**.



Si es la primera vez que se va a ejecutar la función, aparecerá una ventana para configurar un evento de prueba, debe colocar un nombre al evento (test_create_rekognition) y luego seleccionar “Guardar”

Un evento de prueba es un objeto JSON que imita la estructura de las solicitudes emitidas por los servicios de AWS para invocar una función de Lambda. Se utiliza para ver el resultado de la invocación de la función.

Para invocar la función sin guardar un evento, configure el evento JSON, y luego elija Probar.

Acción de evento de prueba

☒ Crear un nuevo evento ☐ Editar evento guardado

Nombre del evento

test_create_recognition

Un máximo de 25 caracteres compuestos por letras, números, puntos, guiones y guiones bajos.

Configuración de uso compartido de eventos

☒ Privado
Este evento solo está disponible en la consola de Lambda y para el creador del evento. Puede configurar un total de 10. [Más información](#)

☐ Compartible
Este evento está disponible para los usuarios de IAM de la misma cuenta que tienen permisos para obtener acceso y utilizar eventos compartibles. [Más información](#)

Plantilla - Opcional

test_recognition

Evento JSON Formato JSON

```
1 {  
2   "key1": "value1",  
3   "key2": "value2",  
4   "key3": "value3"  
5 }
```

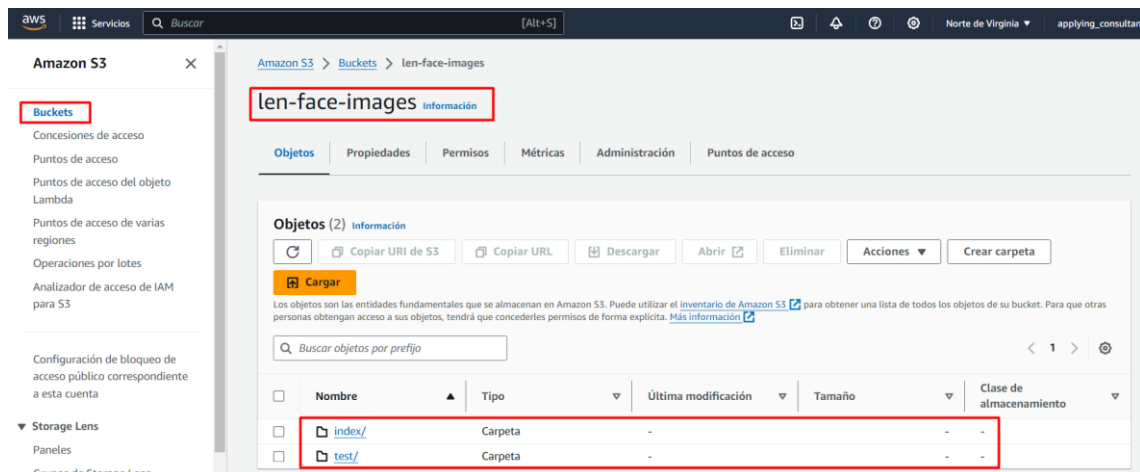
Cancelar

Por último, vuelva a seleccionar la opción “test” y la función lambda se ejecutará correctamente devolviendo como respuesta el nombre de la colección creada, que sería la variable de entorno COLLECTION_ID.

2.2. Bucket S3 para almacenar las imágenes

Este bucket S3 se usará para cargar las imágenes donde aparezcan las personas que se desee reconocer. Posteriormente cuando se quiera consultar las fotos en donde aparece un rostro en particular, como resultado obtendrá todas las rutas de S3 donde estén dichas imágenes.

Nombre del bucket S3: **len-face-images**

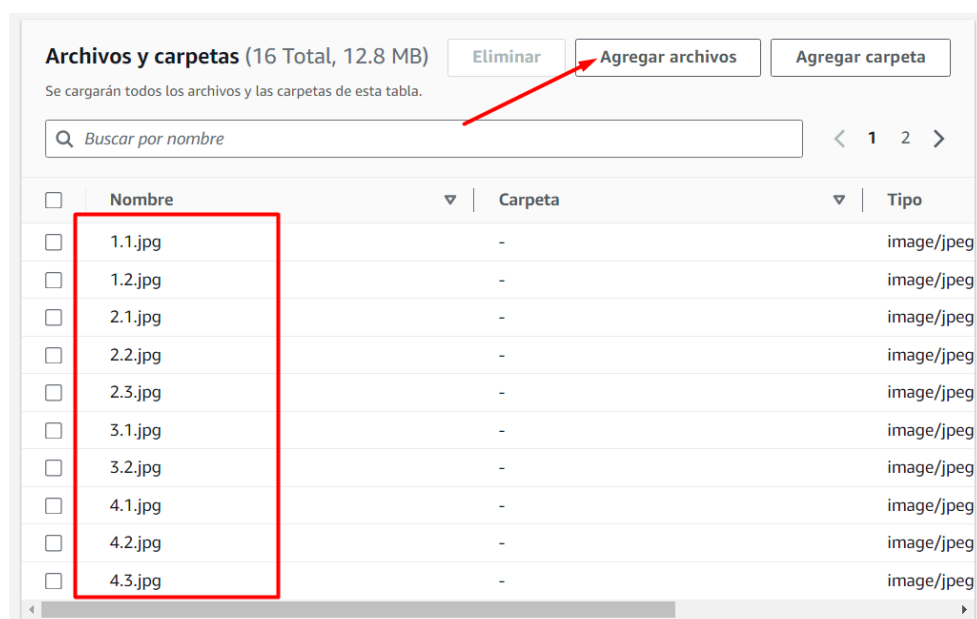


Para que la función lambda de indexado se ejecute, se deben cargar las imágenes en el bucket mencionado siguiendo la siguiente estructura:

ID Usuario/ID Álbum/ID Fotografía



Luego seleccione “**Agregar archivos**” y seleccione las imágenes que desea cargar a S3.



Finalmente seleccione “**Cargar**” y los archivos se subirán a S3, esto hará que se ejecute automáticamente la función lambda de indexado de rostros.

Destino Información

Destino
s3://len-face-images/1/100/3005/

► **Detalles del destino**
Los ajustes del bucket que afectan a los objetos nuevos almacenados en el destino especificado.

► **Permisos**
Conceder acceso público y acceso a otras cuentas de AWS.

► **Propiedades**
Especifique la clase de almacenamiento, los ajustes de cifrado, las etiquetas y mucho más.

Cancelar Cargar

2.3. Función Lambda de indexado de rostros

Esta función lambda se encarga de realizar el indexado de todos los rostros que se encuentren en las imágenes que se carguen previamente al bucket S3 y almacenar dicha información tanto en la colección de Rekognition como en una tabla de DynamoDB.

Nota: Esta función lambda se ejecuta automáticamente después de cargar las imágenes a S3.

- Características
 - Nombre: **face_rekognition**
 - Arn: `arn:aws:lambda:us-east-1:306683712254:function:face_rekognition`
 - Lenguaje: Python 3.12
 - Arquitectura: x86_64
 - Desencadenadores: S3
- Variables de entorno
 - `COLLECTION_ID`: Nombre de la colección de Rekognition que se creó previamente.
 - `DYNAMO_TABLE_NAME`: Nombre de la tabla de DynamoDB donde se almacenará los datos de los rostros indexados.
 - `S3_BUCKET_NAME`: Nombre del bucket S3 donde se subirán las imágenes.

The screenshot shows the AWS Lambda console. On the left, the 'Funciones' (Functions) tab is selected in the sidebar. The main area displays a list of functions. The function 'face_recognition' is highlighted with a red box. The table below shows the details of the functions:

Nombre de la función	Descripción	Tipo de paquete	Tiempo de ejecución	Última modificación
create_collection_recognition	-	Zip	Python 3.12	hace
test_recognition	-	Zip	Python 3.12	hace
face_recognition	-	Zip	Python 3.12	hace

The screenshot shows the source code of the 'face_recognition' function. The code is in a file named 'index.py'. The code is as follows:

```

1 import boto3
2 import json
3 import urllib
4 import os
5 from io import BytesIO
6 from datetime import datetime, timedelta
7 from PIL import Image
8
9 print('Loading function')
10 dynamodb = boto3.client('dynamodb')
11 s3 = boto3.client('s3')
12 rekognition = boto3.client('rekognition')
13
14 collection_id = os.environ['COLLECTION_ID']
15 table_name = os.environ['DYNAMO_TABLE_NAME']
16
17 # ----- Helper Functions -----
18
19 def index_faces(bucket, key):
20     response = rekognition.index_faces(
21         Image={'S3Object':
22             {'Bucket': bucket,
23              'Name': key}},
24         CollectionId=collection_id)
25     return response
26
27 def update_index(table_name, faceId, imageId, personBucketUrl, dateload, box_width, box_height, box_left, box_top):
28     response = dynamodb.put_item(
29         TableName=table_name,
30         Item={
31             'RekognitionId': {'S': faceId},
32             'ImageId': {'S': imageId},
33             'BucketUrl': {'S': personBucketUrl},
34             'Date': {'S': dateload},
35         })
36
37 
```

Para futuras ejecuciones si el nombre de colección de Rekognition y la base de datos en DynamoDB han cambiado, se deben actualizar las variables de entorno de la función.

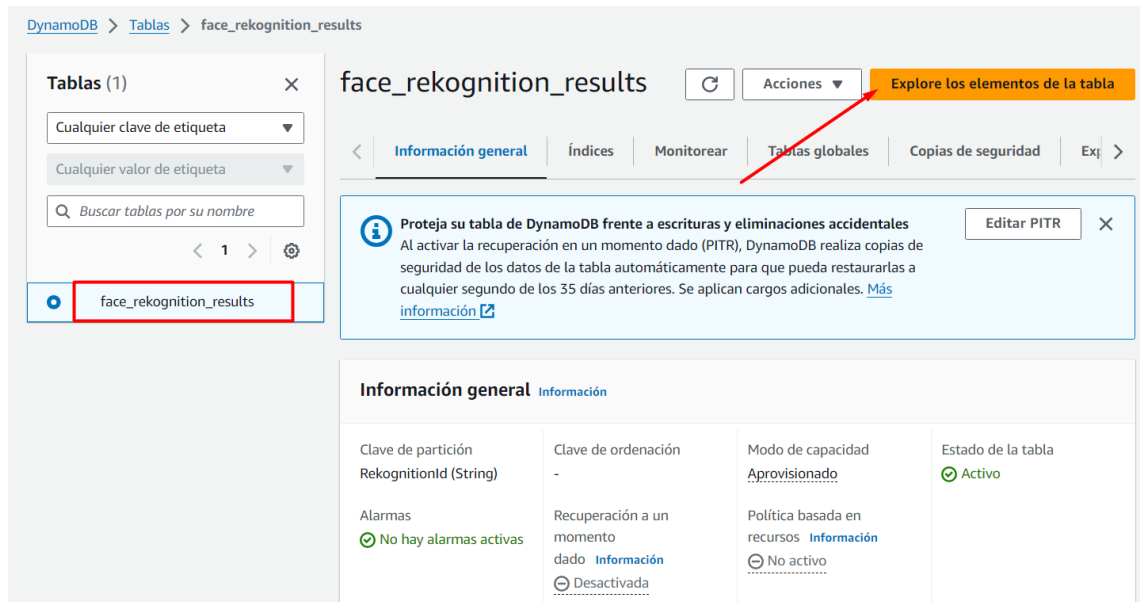
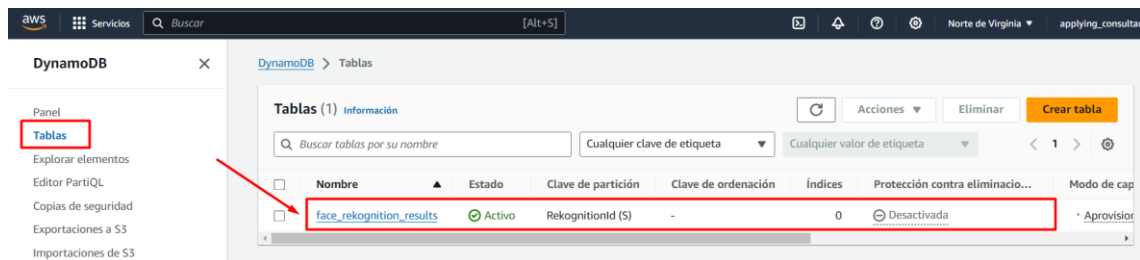
The screenshot shows the 'Configuración' (Configuration) tab of the AWS Lambda console. The 'Variables de entorno' (Environment Variables) section is highlighted with a red box. The table below shows the environment variables:

Clave	Valor
COLLECTION_ID	face_recognition
DYNAMO_TABLE_NAME	face_recognition_results
S3_BUCKET_NAME	len-face-images

2.4. Tabla en DynamoDB

Esta tabla de DynamoDB se usará como un almacenamiento de clave-valor simple para almacenar el FaceId devuelto por Amazon Rekognition y otras variables que se requieran.

Nombre de la tabla: **face_recognition_results**



Al finalizar la subida de imágenes a S3, se ejecuta la función lambda y finalmente se almacenan algunos datos relevantes de los rostros indexados por Amazon Rekognition, por ejemplo:

- **RekognitionId:** Hace referencia al FaceId de los rostros indexados.
- **BoundingBox:** Hace referencia a las dimensiones de los rostros indexados, muestra las dimensiones de ancho, alto, izquierda y derecha para ubicar el rostro de las personas dentro de las imágenes.
- **BucketUrl:** Hace referencia a la ruta s3 donde se encuentran almacenadas las imágenes que coincide con el rostro que se desea analizar.
- **ImageId:** Hace referencia al id que por defecto tiene una imagen, este dato viene por defecto en la metadata de la imagen al subirla a S3.

Elementos devueltos (44)

Acciones ▼ Crear elemento

< 1 > ⚙️

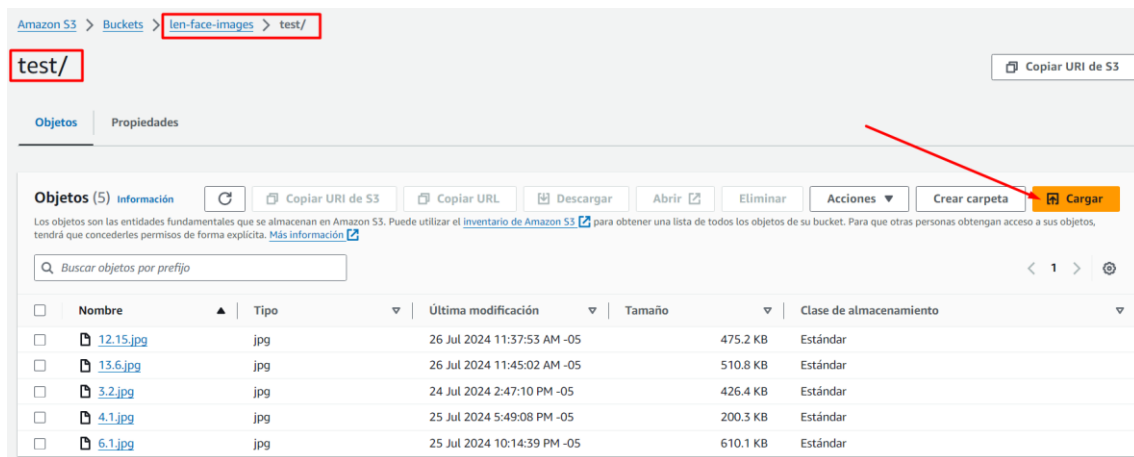
<input type="checkbox"/>	RekognitionId (Cadena) ▼	BoundingBox ▼	BucketUrl ▼	Date ▼	ImageId ▼
<input type="checkbox"/>	ebaa667b-91b8-4501-b3ae-4794d6e76bce	{"Height": { "...	len-face-im...	2024-07-2...	e0d78838-31ed-3b93-9a2e-ed...
<input type="checkbox"/>	efdb55a8-5c9d-4bdb-9d04-db239e0eb357	{"Height": { "...	len-face-im...	2024-07-2...	4653a431-92c6-3966-bf59-ffa...
<input type="checkbox"/>	dc6426cd-3798-49e8-8c55-136224e046fa	{"Height": { "...	len-face-im...	2024-07-2...	d0e30e1c-1d18-3971-a460-d7...
<input type="checkbox"/>	75164b6c-9503-4805-a467-eb5aafab85d1	{"Height": { "...	len-face-im...	2024-07-2...	d0e30e1c-1d18-3971-a460-d7...
<input type="checkbox"/>	a0ce5e2e-cdc7-4e96-b501-27dc6ecaba03	{"Height": { "...	len-face-im...	2024-07-2...	efa32a34-e0e9-3569-8885-e3...
<input type="checkbox"/>	cf9e1524-f712-4689-a25e-16062f410ba1	{"Height": { "...	len-face-im...	2024-07-2...	ec2a2455-256d-359b-b4d5-84...
<input type="checkbox"/>	606a81e1-62eb-40c2-9525-02cea26fd45f	{"Height": { "...	len-face-im...	2024-07-2...	ec2a2455-256d-359b-b4d5-84...
<input type="checkbox"/>	6c87ed42-6c1c-43b5-885d-957a3a2cf10c	{"Height": { "...	len-face-im...	2024-07-2...	f1c876f5-a85f-3761-b554-4da...
<input type="checkbox"/>	add7541a-d1be-44de-89d5-b3cb672d6781	{"Height": { "...	len-face-im...	2024-07-2...	eead4bab-d6e7-3f49-9a23-54...
<input type="checkbox"/>	09ea6b44-3833-4b5a-80e3-edd7b0b868ae	{"Height": { "...	len-face-im...	2024-07-2...	46775f2f-3d6f-32ff-bcd8-e09...
<input type="checkbox"/>	e1e005aa-36d0-4e03-8e46-4d606f284ae1	{"Height": { "...	len-face-im...	2024-07-2...	86f0a80f-a7cb-335d-8333-84f...

3. Pruebas de función Lambda

3.1. Subida de archivo a S3

Para realizar las pruebas se deben subir imágenes a siguiente bucket s3: **len-face-images/test**.

Nota: Si no se quiere usar este método, podrían copiar el script de la función lambda en un notebook de jupyter y realizar pruebas con las imágenes de manera local. Para esto deben configurar sus credenciales de AWS en su entorno para poder ejecutarlo.

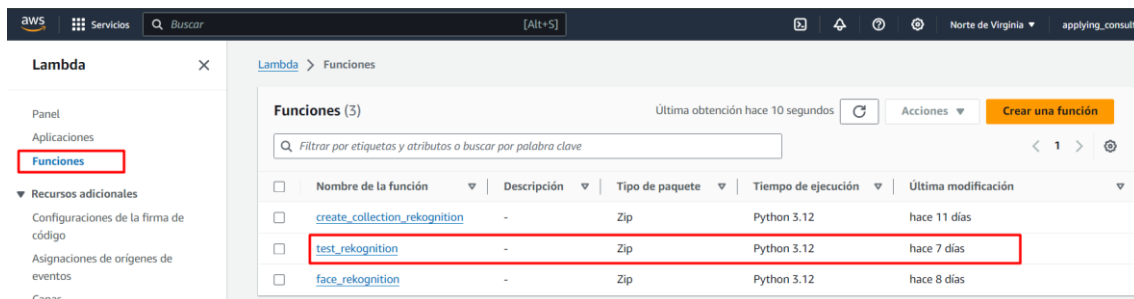


3.2. Función Lambda de reconocimiento de rostro

Esta función lambda se ha implementado para realizar pruebas al detectar las imágenes en donde se encuentra el rostro de una persona con cierto grado de confianza.

Nota: Para realizar las pruebas para el análisis y reconocimiento de rostros, las imágenes que se utilicen deben ser personales (selfies) o en todo caso si la imagen contiene varios rostros, Rekognition por defecto tomará como rostro para el análisis el rostro más prominente dentro de la imagen.

Nombre de la función: **test_rekognition**



Para realizar el análisis se debe actualizar la variable **image_key** con el nombre de la imagen que se encuentra en la ruta S3 que se menciona en el punto anterior.

Código fuente Información

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

test_rekognition - lambda_function.py

```

1 import boto3
2 import json
3
4 s3 = boto3.client('s3')
5 rekognition = boto3.client('rekognition')
6 dynamodb = boto3.client('dynamodb')
7
8 bucket_name = 'len-face-images'
9 image_key = 'test/13.6.jpg'
10 collection_id = 'face_rekognition'
11 table_name = 'face_rekognition_results'
12
13 def lambda_handler(event, context):
14
15     # Paso 2: Usar search_faces_by_image
16     response = rekognition.search_faces_by_image(
17         CollectionId=collection_id,
18         Image={'S3Object': {'Bucket': bucket_name, 'Name': image_key}},
19     )
20
21     # Paso 3: Consultar DynamoDB con los FaceIds obtenidos
22     for match in response['FaceMatches']:
23         print('-----')
24         print(f" FaceId: {match['Face']['FaceId']} - Confidence: {match['Face']['Confidence']}")
25
26     # Consulta la tabla de DynamoDB
27     face = dynamodb.get_item(
28         TableName=table_name,
29         Key={'RekognitionId': {'S': match['Face']['FaceId']}}
30     )

```

Por último, se debe ejecutar la función lambda seleccionando la opción “Test” y ver los resultados que nos devuelve como el porcentaje de confianza de coincidencia y la ruta S3 de las imágenes donde se encuentra el rostro que se analizó.

Tools Window Test Deploy

lambda_function x Execution results x

Execution results

Test Event Name
test_rekognition

Response

```

{
  "statusCode": 200,
  "body": "\"Script ejecutado correctamente!\""
}

```

Function Logs

START RequestId: ce43a3f8-4575-406a-bca2-0d0de35f0b05 Version: \$LATEST

```

FaceId: 6858184f-0636-466d-a163-642ea7388c5a - Confidence: 99.99549865722656
Ruta S3: len-face-images/index/13.4.jpg
Dimensiones de la caja del rostro:
Width: 183.88270810246468 px - Height: 245.89987182617188 px - Left: 555.2538925409317 px - Top: 713.38623046875 px
-----
FaceId: 22484896-e9b9-4b8d-8e2e-4453c37de648 - Confidence: 99.98189544677734
Ruta S3: len-face-images/index/13.1.jpg
Dimensiones de la caja del rostro:
Width: 79.17400360107422 px - Height: 107.38150242716074 px - Left: 927.1900634765625 px - Top: 274.23782028257847 px
-----
FaceId: 4e3c557f-154d-452c-b106-95951006c3e4 - Confidence: 99.96880340576172
Ruta S3: len-face-images/index/13.3.jpg
Dimensiones de la caja del rostro:
Width: 77.91556549072266 px - Height: 115.45223459601402 px - Left: 804.90185546875 px - Top: 393.2924263179302 px
-----
FaceId: 09ea6b44-3833-4b5a-80e3-edd7b0b868ae - Confidence: 99.943603515625
Ruta S3: len-face-images/index/13.2.jpg
Dimensiones de la caja del rostro:
Width: 77.91781616210938 px - Height: 85.29518550261855 px - Left: 1017.0242919921875 px - Top: 462.9610414803028 px
-----
END RequestId: ce43a3f8-4575-406a-bca2-0d0de35f0b05
REPORT RequestId: ce43a3f8-4575-406a-bca2-0d0de35f0b05 Duration: 1950.21 ms Billed Duration: 1951 ms Memory Size: 128 MB

```

4. Infraestructura como código (IaC)

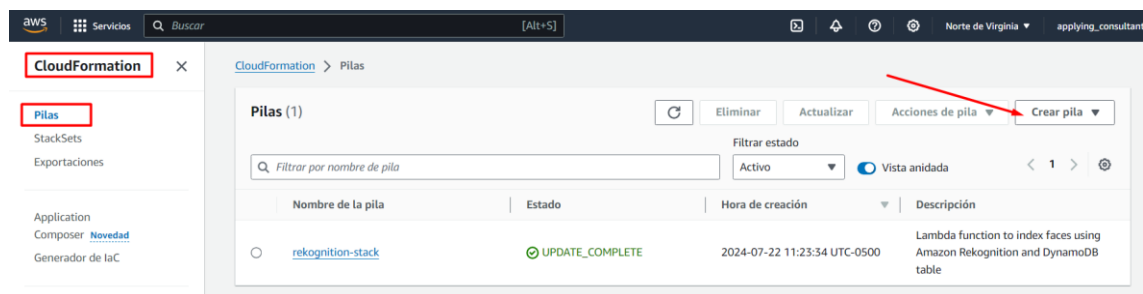
4.1. Plantilla CloudFormation

Todos los recursos implementados dentro de la cuenta de AWS se han realizado mediante una plantilla de CloudFormation, ya que es una buena practica para desplegar recursos de una forma más fácil y rápida.

Esta plantilla se va a adjuntar en el correo de la documentación técnica.

Para desplegar la plantilla de CloudFormation se debe seguir los siguientes pasos:

- Debe dirigirse al servicio de Cloudformation, en el panel izquierdo seleccione la opción **Pilas** (Stacks) y haga clic en **“Crear pila”**.



- En esta sección debe seleccionar las opciones **“Seleccione una plantilla existente”** y **“Cargar un archivo de plantilla”** para elegir la plantilla (rekognition.yaml), luego haga clic en **“Siguiente”**.

The screenshot shows the 'Crear pila' (Create Stack) wizard. The first section, 'Preparar la plantilla', has 'Seleccione una plantilla existente' selected. The second section, 'Especificar plantilla', has 'Cargar un archivo de plantilla' selected. Below this, the 'Elegir archivo' button is highlighted with a red arrow. At the bottom right, the 'Siguiente' button is highlighted with a red arrow.

- En la siguiente sección se debe colocar un nombre a la pila (rekognition-stack), luego hay una serie de parámetros que se pueden dejar por defecto o cambiarlos

con otros nombres según lo requieran. Al finalizar en la parte inferior se debe seleccionar **“Siguiente”**.

Especificar los detalles de la pila

Proporcionar un nombre de pila

Nombre de la pila

rekognition-stack2

El nombre de la pila debe tener entre 1 y 128 caracteres, comenzar con una letra y contener solo caracteres alfanuméricos. Recuento de caracteres: 18/128.

Parámetros

Los parámetros se definen en la plantilla y le permiten introducir valores personalizados al crear o actualizar una pila.

Lambda Function Configuration

LambdaFunctionName

The name of the lambda function

face_rekognition

LambdaRoleName

The role name for the lambda function

lambda_face_rekognition_role

LambdaPythonVersion

The python version that uses the lambda function

3.12

LambdaTimeout

The S3 prefix name

300

LambdaMaxMemory

The S3 prefix name

512

S3BucketName

The S3 Bucket name

len-face-images

S3PrefixName

The S3 prefix name

index/

Rekognition and DynamoDB

CollectionId

face_rekognition

DynamoTableName

The name of the table where the index faces are stored

face_rekognition_results

Cancelar Anterior **Siguiente**

- En esta siguiente sección deje todo tal y como esta por defecto, en la parte inferior debe seleccionar “**Siguiente**”.

Configurar opciones de pila

Etiquetas — opcional

Las etiquetas (pares clave-valor) se utilizan para aplicar metadatos a los recursos de AWS, lo que puede ayudar a organizar, identificar y categorizar esos recursos. Puede añadir hasta 50 etiquetas únicas para cada pila.

No hay etiquetas asociadas a la pila.

Agregar etiqueta nueva

Puede agregar 50 etiqueta(s) más

Permisos — opcional

Especifique un rol de servicio de AWS Identity and Access Management (IAM) existente que CloudFormation pueda asumir.

Rol de IAM: opcional

Elija el rol de IAM para CloudFormation que se utilizará para todas las operaciones realizadas en la pila.

Nombre de rol de IAM Sample-role-name Eliminar

↻

Opciones avanzadas

Puede establecer opciones adicionales para la pila, como opciones de notificación y una política de pilas. [Más información](#)

► **Política de pila — opcional**

Define los recursos que desea proteger frente a actualizaciones involuntarias durante una actualización de pila.

► **Configuración de restauración — opcional**

Especifique alarmas para el monitoreo de CloudFormation al crear y actualizar la pila. Si la operación supera un umbral de alarma, CloudFormation lo restaura.

► **Opciones de notificación — opcional**

Especifique un tema nuevo o existente de Amazon Simple Notification Service al que se envíen las notificaciones sobre los eventos de la pila.

► **Opciones de creación de la pila — opcional**

Especifique las opciones de protección de tiempo de espera y terminación para la creación de pilas.

Cancelar Anterior **Siguiente**

- En esta última sección se muestra un resumen de todas las configuraciones que se han realizado, en la parte inferior se debe seleccionar la opción indicada para crear recursos IAM y finalmente seleccionar “**Enviar**” para desplegar la plantilla con todos los recursos necesarios de la solución.

Revisar y crear

Paso 1: Especificar plantilla

Editar

Requisito previo: preparar la plantilla


Plantilla
La plantilla está lista

Plantilla

URL de plantilla
<https://s3.us-east-1.amazonaws.com/cf-templates-mqg1b3849q5v-us-east-1/2024-08-03T025810.572Z77m-rekognition.yaml>

Descripción de la pila
Lambda function to index faces using Amazon Rekognition and DynamoDB table

Capacidades

 The following resource(s) require capabilities: [AWS::IAM::Role]
Esta plantilla contiene recursos de Identity and Access Management (IAM). Verifique que desea crear cada uno de estos recursos y que estos tengan los permisos mínimos necesarios. Además, tienen nombres personalizados. Verifique que los nombres personalizados sean únicos dentro de su cuenta de AWS. [Más información](#)

☒ Confirmando que AWS CloudFormation podría crear recursos de IAM con nombres personalizados.

Crear conjunto de cambios

Cancelar

Anterior

Enviar

- Una vez finalizado el despliegue de la plantilla CloudFormation, se podrá visualizar todos los recursos que se han desplegado con éxito en la cuenta de AWS con el estado **“CREATE_COMPLETE”** o **“UPDATE_COMPLETE”**.

CloudFormation > Pilas > rekognition-stack

Pilas (1)

Filtrar por nombre de pila

Filtrar estado: Activo, Vista anidada

Pilas

rekognition-stack
2024-07-22 11:23:34 UTC-0500
UPDATE_COMPLETE

rekognition-stack

Eliminar

Actualizar

Acciones de pila

Crear pila

Información de la pila **Eventos** Recursos Salidas Parámetros Plantilla

Eventos (65)

Detectar causa raíz

Buscar eventos

Marca temporal	ID lógico	Estado	Estado detallado
2024-07-25 17:37:49 UTC-0500	rekognition-stack	UPDATE_COMPLETE	-
2024-07-25 17:37:49 UTC-0500	rekognition-stack	UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	-
2024-07-25 17:37:46 UTC-0500	MyLambdaFunction	UPDATE_COMPLETE	-
2024-07-25 17:37:39 UTC-0500	MyLambdaFunction	UPDATE_IN_PROGRESS	-
2024-07-25 17:37:35 UTC-0500	rekognition-stack	UPDATE_IN_PROGRESS	-