

IMPLEMENTACIÓN EN AWS

Sunglasses Detection Project

Nahuel Montes de Oca



"Detecting the use of sunglasses in images using Amazon Rekognition Image"

supervisado por
SA Roberto LANDA

Diciembre 21, 2022

Contents

1	Presentación	2
2	Marco Teórico	3
2.1	Amazon Web Services	3
2.2	Informática en la nube con AWS	3
2.3	Servicios utilizados	3
3	Proyecto	5
3.1	Como surgió	5
3.2	Funcionamiento	6
3.3	Sobre el código	6
3.4	Pruebas	7
4	Conclusiones	11
5	Fuentes	12

1 Presentación

Mi nombre es Nahuel Montes de Oca, actualmente me encuentro estudiando para la certificación de AWS Solutions Architect (SAA-C03), dado que el año pasado cursé el programa AWS re/Start el cual me permitió certificarme como Cloud Practitioner y aún más importante, me abrió las puertas a la computación en la nube, la cual encontré muy interesante, siendo que ya venía interiorizado en el área de desarrollo por haber empezado en Ingeniería de Software y cursos anteriores (de programación y desarrollo web en su mayoría).

En cuanto a este proyecto personal, tiene como objetivo no solo como implementación que podría aplicarse en muchos casos de uso, sino también para empezar a combinar los servicios de AWS y entender como se pueden crear las arquitecturas en la nube. En este caso utilicé cuatro servicios especialmente, pero se puede expandir a más si la situación lo requiere. Más adelante se explica en detalle los servicios involucrados y como se van a comunicar entre ellos.



Contacto:

A continuación adjunto una lista de enlaces donde me pueden encontrar, ya sea para saber más sobre el proyecto o cualquier otra cuestión relacionada, también son bienvenidas las sugerencias o dudas:

- [LinkedIn](#)
- [GitHub](#)
- [email](#)

2 Marco Teórico

2.1 Amazon Web Services



Amazon Web Services (AWS) es la plataforma en la nube más adoptada y completa en el mundo, que ofrece más de 200 servicios integrales de centros de datos a nivel global. Millones de clientes, incluso las empresas emergentes que crecen más rápido, las compañías más grandes y los organismos gubernamentales líderes, están usando AWS para reducir los costos, aumentar su agilidad e innovar de forma más rápida.

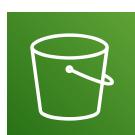
2.2 Informática en la nube con AWS

¿Qué es la informática en la nube?

La informática en la nube es la distribución de recursos de TI bajo demanda a través de Internet mediante un esquema de pago por uso. En lugar de comprar, poseer y mantener servidores y centros de datos físicos, puede acceder a servicios tecnológicos, como capacidad informática, almacenamiento y bases de datos, en función de sus necesidades a través de un proveedor de la nube como Amazon Web Services (AWS).

2.3 Servicios utilizados

Amazon Simple Storage Service (S3)



Amazon S3 es un servicio de almacenamiento de objetos creado para almacenar y recuperar cualquier volumen de datos desde cualquier ubicación. Es un servicio de almacenamiento sencillo que ofrece excelente durabilidad, disponibilidad, rendimiento, seguridad y escalabilidad prácticamente ilimitada a costos muy reducidos. Más info [aquí](#)

Amazon Rekognition Image



Rekognition Image es un servicio de reconocimiento de imágenes con tecnología de aprendizaje profundo que detecta objetos, escenas y rostros; extrae texto, reconoce a personas famosas e identifica contenido inapropiado en imágenes. También le permite realizar búsquedas y comparar rostros. Rekognition Image se basa en la misma tecnología de aprendizaje profundo demostrada y altamente escalable desarrollada por los científicos de visión informática de Amazon para analizar miles de millones de imágenes al día para Prime Photos. El servicio devuelve una puntuación de fiabilidad de todos los elementos que identifica para que pueda tomar decisiones bien informadas acerca de cómo utilizar los resultados. Además, todos los rostros detectados se devuelven con coordenadas de un cuadro delimitador, un marco rectangular que abarca todo el rostro y que se puede utilizar para localizar la posición del rostro en la imagen. Más info [aquí](#)

Amazon Lambda



AWS Lambda es un servicio informático sin servidor que ejecuta código como respuesta a eventos y administra automáticamente los recursos informáticos subyacentes. Estos eventos pueden ser cambios de estado o una actualización, como que un usuario coloque un artículo en un carrito de la compra en un sitio web de comercio electrónico. Puede utilizar AWS Lambda para ampliar otros servicios de AWS con lógica personalizada o crear sus propios servicios backend que funcionen a escala, rendimiento y seguridad de AWS. Más info [aquí](#)

Amazon Simple Notification Service (SNS)



Amazon Simple Notification Service (Amazon SNS) es un servicio web que facilita las tareas de configuración, utilización y envío de notificaciones desde la nube. Ofrece a los desarrolladores una funcionalidad muy escalable, flexible y rentable para publicar mensajes desde una aplicación y entregarlos inmediatamente a suscriptores o a otras aplicaciones. Está diseñado para simplificar a los desarrolladores el uso de recursos informáticos con escala web. Amazon SNS sigue el paradigma de mensajería “publicación-suscripción” (pub-sub), con notificaciones que se envían a los clientes con un mecanismo “push” que elimina la necesidad de verificar o “sondear” regularmente información o actualizaciones nuevas. Más info [aquí](#)

Nota: La información detallada anteriormente fue completamente extraída de la página web de [AWS](#)

3 Proyecto

3.1 Como surgió

Durante el curso de AWS re/Start, fui conociendo los servicios basicos, tales como S3, Lambda y SNS, entre otros. Vi tambien como se pueden combinar para realizar tareas automaticas o solucionar problemas cotidianos. Sumado a esto, apareció un servicio muy interesante, Amazon Rekognition, que como fue explicado anteriormente, utiliza inteligencia artificial para el reconocimiento y análisis de imágenes o videos. Me llamó mucho la atención dado que cuenta con una gran cantidad de casos de uso, lo cual me podía servir para realizar algún proyecto personal.

Me gustaría hacer una mención especial a mi profesor Roberto Landa, quién me compartió anteriormente una implementación similar con el uso de Amazon Rekognition (en este proyecto se detectaba el uso de casco de seguridad, más [info aquí](#)) , que fue uno de los casos el cual me incentivó a aplicarlo por mi cuenta, buscando información en la documentación de AWS y recursos de la comunidad en YouTube y StackOverflow.

Casos de uso:

Con el uso de Amazon Rekognition, en especial con esta implementación, se puede aplicar en varios escenarios, donde podemos mencionar:

- Control de vestimenta en un local
- Página web interactiva
- Tienda de ropa o accesorios
- Búsqueda en gran cantidad de archivos
- Filtrado de imágenes

Comunicación entre los servicios:

Aquí se puede ver una imagen que muestra, a modo de esquema, como funciona la arquitectura y cual es el funcionamiento que se detallará posteriormente, donde se pueden apreciar dos buckets de S3, uno donde se coloca la imagen a testear, el cual activa la primer función Lambda, que a su vez utiliza la IA de Amazon Rekognition. Si encuentra lentes de sol en el individuo, la función copia la imagen al otro bucket, activando así la otra Lambda, la cual envía una notificación por mail al usuario.

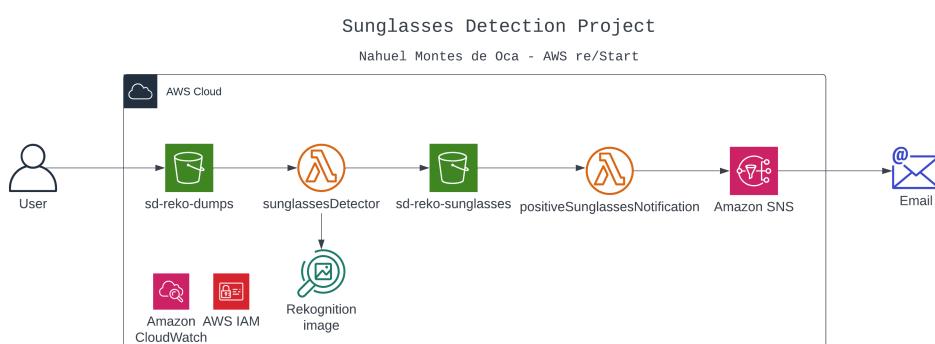


Figure 1: Arquitectura del proyecto

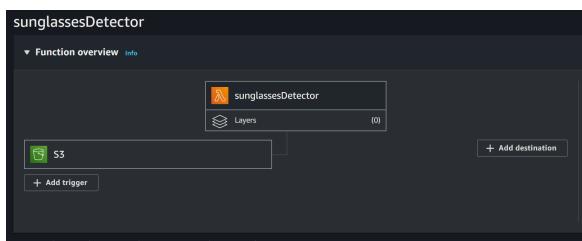
3.2 Funcionamiento

Cómo es el procedimiento

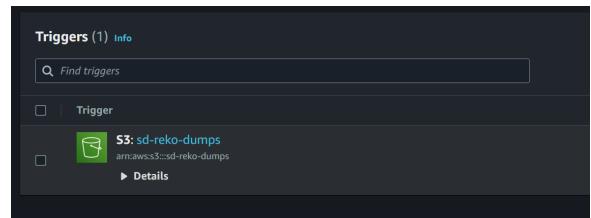
1. Se realiza la subida de la imagen (preferentemente con extensión .png) al bucket "sd-reko-dumps"
2. El evento put en el bucket S3 activa la función Lambda *sunglassesDetector*, la cual analizará el archivo (imagen) involucrada en este evento.
3. La función, con el uso de Amazon Rekognition Image, analiza la imagen y determina si el individuo porta lentes de sol.
4. Si la respuesta a lentes de sol es positiva, procede a copiar el archivo al otro bucket llamado "sd-reko-sunglasses"
5. Esa copia genera el evento put en ese bucket
6. Se activa la otra función Lambda *positiveSunglassesNotification* y envía el mail, usando SNS, al usuario registrado
7. Se puede verificar el mail de confirmación y la copia del archivo en la Management Console

3.3 Sobre el código

En este [artículo](#) tenemos un ejemplo de un llamado a Lambda a través del trigger s3-put, donde podemos con la subida del archivo al bucket, activar la función para hacer dentro del código lo que nos parezca necesario para el uso de Rekognition:



(a) Lambda Function



(b) S3 trigger

Figure 2: *sunglassesDetector* function

En este [enlace](#) podemos ver un poco de cómo se puede utilizar Rekognition, a través del SDK, en este caso usaremos el lenguaje de programación **python** para trabajar dentro de la función Lambda.

Aquí tenemos un ejemplo de como es la respuesta de la función **detect-faces**, la cual utilizaremos para determinar si en nuestra imagen se encuentra una cara, y consecuentemente si el individuo detectado cuenta con **lentes de sol**, de ser así procedemos a la parte del copiado a otro S3 Bucket. Según la [documentación](#) de la CLI Command Reference:

"*detect-faces* detects the 100 largest faces in the image. For each face detected, the operation returns face details. These details include a bounding box of the face, a confidence value (that the bounding box contains a face), and a fixed set of attributes such as facial landmarks (for example, coordinates of eye and mouth), presence of beard, **sunglasses**, and so on."

Luego de haber resuelto el uso de Rekognition, en caso positivo, podemos copiar esta imagen al otro Bucket, como podemos ver en este [artículo](#). Finalmente, la función lambda termina su ejecución.

Luego de haber copiado nuestra imagen con respuesta positiva, este nuevo evento S3-put activará la otra función Lambda que funciona muy similar a la primera en la parte del trigger, pero en vez de utilizar las funciones de Rekognition, va a trabajar con SNS para enviar un mail al usuario registrado con un mensaje a modo de aviso por la respuesta positiva de la imagen en cuestión, con el nombre del archivo subido.

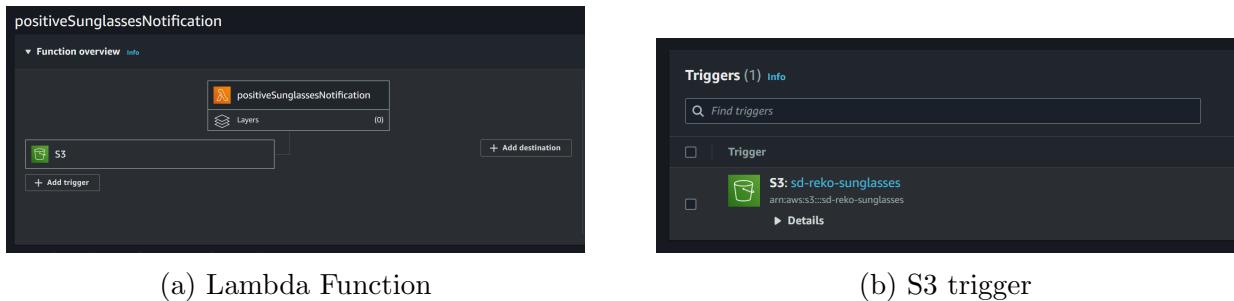


Figure 3: *positiveSunglassesNotification* function

3.4 Pruebas

Comenzamos eligiendo un par de sujetos de prueba, en este caso utilicé dos personas con una foto positiva y otra negativa a lentes de sol, para ver si la función cumple su propósito:



Figure 4: Sujetos de prueba

Procedo a subir las imágenes al bucket de pruebas llamado "sd-reko-dumps", y esperamos a que se active la primer función Lambda, lo podemos verificar en la consola de CloudWatch o la de Lambda, luego veremos si las imágenes fueron copiadas al otro bucket "sd-reko-sunglasses" si es que la persona tiene lentes de sol:

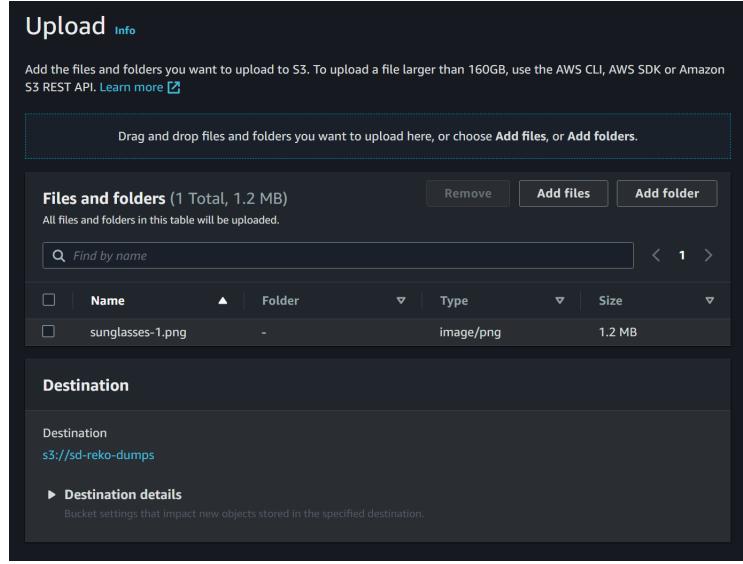


Figure 5: Primer prueba

Luego de completar la subida, podemos ver en la consola de monitoreo la llamada a nuestra función Lambda y su respuesta:

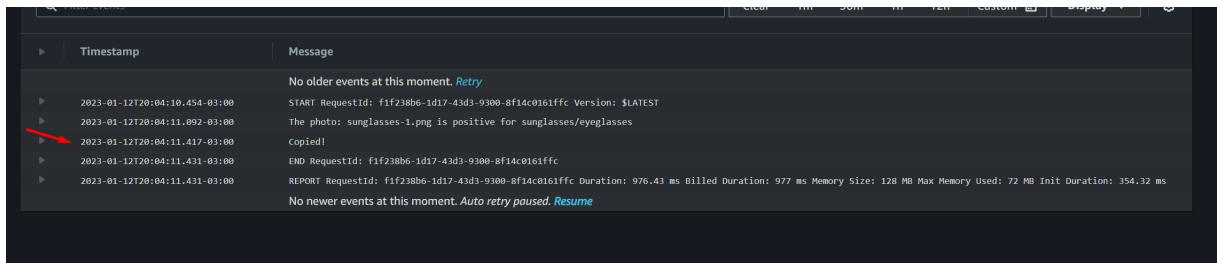


Figure 6: Primer respuesta

Como muestran los logs, nuestra primera imagen contiene una persona utilizando lentes de sol, por lo cual fue copiada en el bucket "sd-reko-sunglasses" al instante:

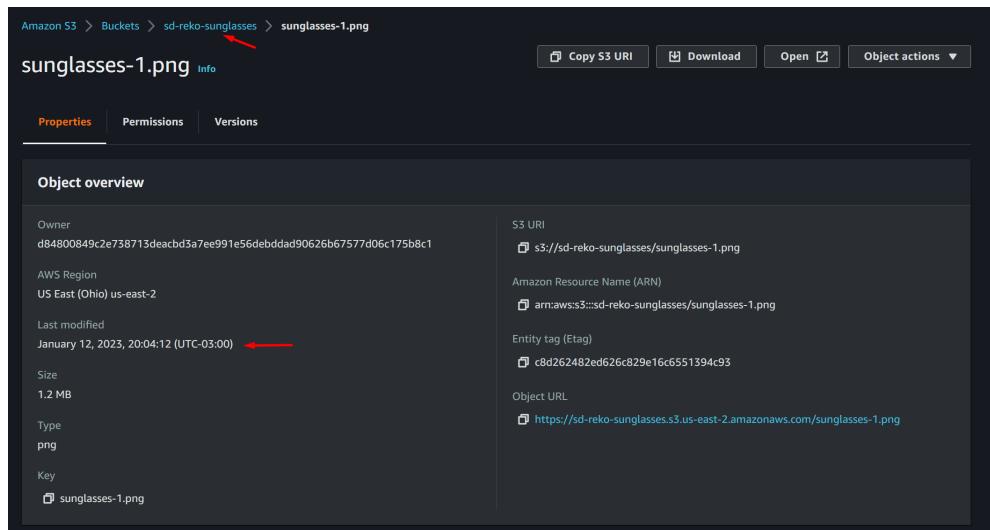


Figure 7: Primer copiado

Continuamos con la subida de la siguiente imagen, la cual no cuenta con lentes de sol:

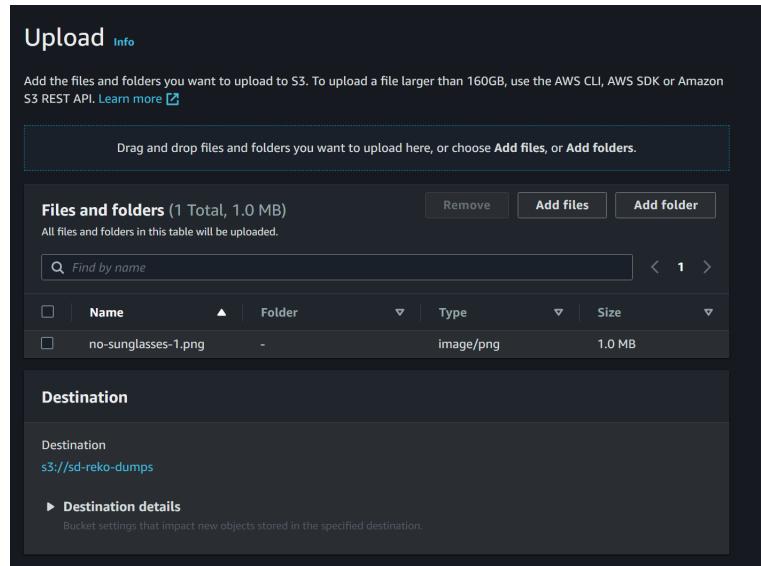


Figure 8: Segunda prueba

Luego de completar la subida, podemos ver en la consola de monitoreo la llamada a nuestra función Lambda y su respuesta:

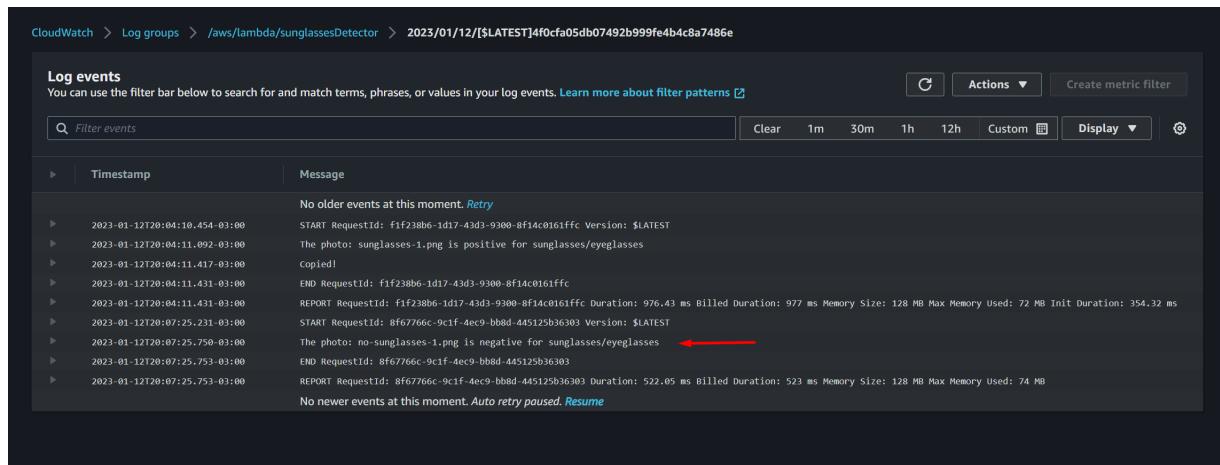
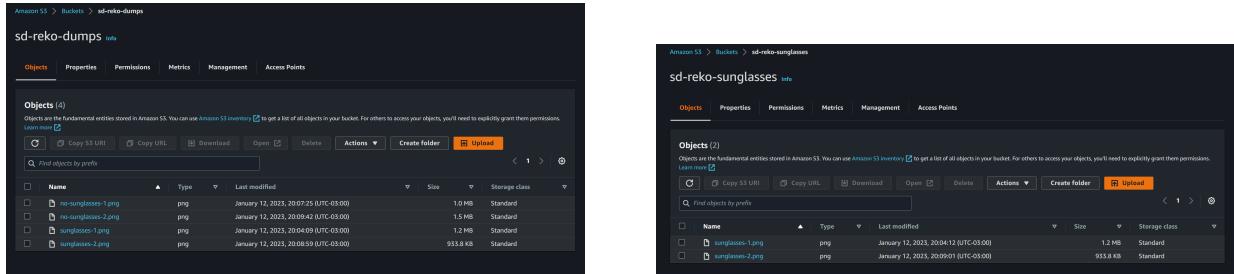


Figure 9: Segunda respuesta

Como muestran los logs, nuestra segunda imagen no contiene una persona utilizando lentes de sol, por lo cual no fue copiada en el otro bucket.

Procedo a subir las 2 imágenes restantes donde veremos un resumen de los llamados a la función *sunglassesDetector* en la consola de CloudWatch a través de los registros más adelante.

Segun las respuestas mostradas anteriormente, podemos ver que solo fueron copiadas las imágenes donde el individuo se encuentra con lentes de sol, lo cual nos muestra el correcto funcionamiento del código en las dos funciones Lambda, ya que tambien se puede ver el mail de confirmación:



(a) sd-reko-dumps bucket

(b) sd-reko-sunglasses bucket

Figure 10: Buckets al final de la prueba

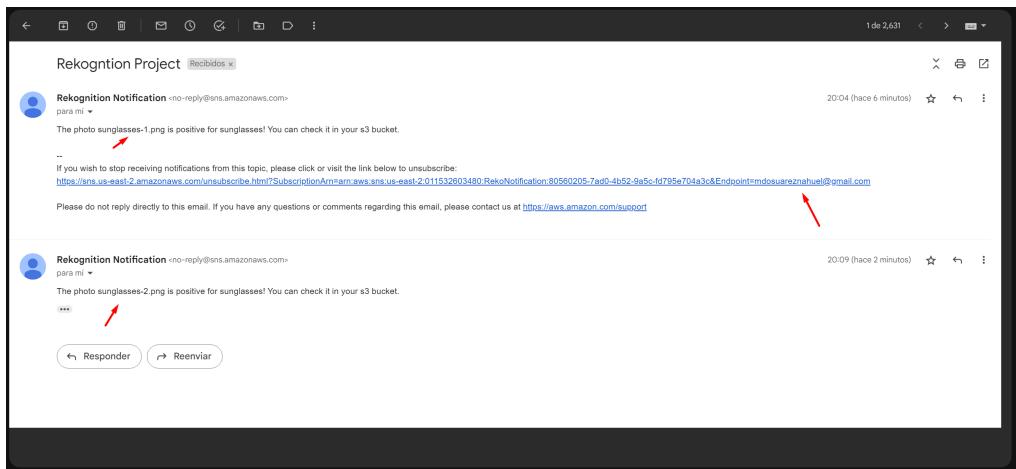


Figure 11: Mails de confirmación

A continuación podemos ver los llamados a la función en la consola de CloudWatch, donde se detallan los tiempos de ejecución y el momento en que fueron accionadas:

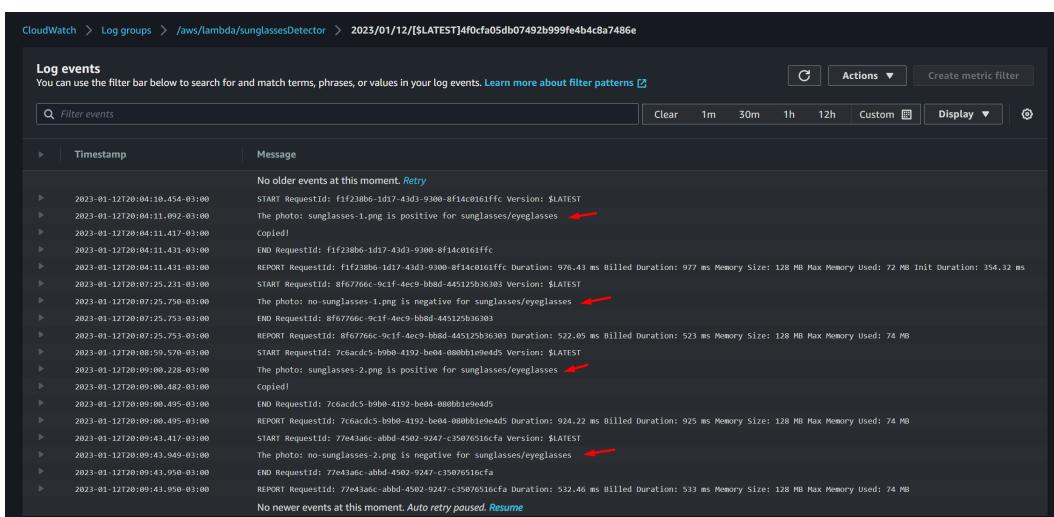


Figure 12: Llamados a la función

4 Conclusiones

Como pudimos ver en las pruebas, la implementación funciona correctamente, cumple con su propósito de organizar los archivos según cumplan con nuestra condición o no. Como mencioné anteriormente, este proyecto es una arquitectura básica, que puede escalar en complejidad si tenemos un objetivo más específico o también lo podemos combinar con otras arquitecturas que tengan propósitos más grandes.

En este repositorio están los códigos de las funciones Lambda y los sujetos de prueba. Por cualquier duda o sugerencia me pueden contactar a través de los links de la sección 1.

5 Fuentes

Sujetos de prueba:

- 4a : Dua Lipa con lentes
- 4b : Harry Styles con lentes
- 4c : Dua Lipa sin lentes
- 4d : Harry Styles sin lentes

Documentación:

Amazon Rekognition Image
AWS

Videos:

Youtube, Implementación de Amazon Rekognition Image: Rishab Teaches Tech
Youtube, Implementación de Amazon Rekognition Image: edureka!