MODULO 09 - EJERCICIO 07-A

ALEXIS YURI M.

Diseña una solución basada en AWS Lambda que se dispare automáticamente cuando se suba un archivo a un bucket S3. La función debe procesar los datos (por ejemplo extraer campos relevantes o convertir el formato) y dejar un archivo transformado en otro bucket.

Diseño del Flujo Serverless: Ingesta y Validación de Datos para un Data Lake.

Se necesita un sistema que, cada vez que un archivo (por ejemplo, un CSV de ventas) sea subido al bucket data\_lake\_inicial, lo valide, lo comprima en formato Parquet, lo mueva al bucket data\_lake\_procesado y notifique el resultado.

Paso 1: Arquitectura Serverless y Disparador (Trigger).

Se utiliza la arquitectura orientada a eventos (event-driven) donde la función Lambda sólo se ejecuta en respuesta a una acción específica.

Componente Principal: Función AWS Lambda (en Python).

Disparador (Trigger): Se configura el bucket de origen (data\_lake\_inicial) para que emita un evento de creación de objeto en S3 cada vez que se carga un nuevo archivo.

Flujo: El evento S3 llama a la función Lambda, pasándole como payload la ubicación exacta del archivo recién cargado (bucket y key).

Paso 2: Lógica de la Función Lambda (Procesamiento).

La función debe encapsular la lógica de Extracción y Transformación para la ingesta de datos:

- Extracción: La función utiliza el módulo boto3 para descargar el archivo de datos desde el bucket de origen S3 al entorno de ejecución temporal (/tmp) de Lambda.

- Validación: Se valida el archivo (ej. se comprueba el número de columnas o la presencia de encabezados). Si la validación falla, se mueve el archivo a un bucket de errores (data\_lake\_errores) y se notifica al equipo.

- Transformación: Si la validación es exitosa, se usa una biblioteca de procesamiento de datos (como Pandas dentro de Lambda) para:

- Aplicar limpieza básica (ej. eliminar valores nulos).

- Convertir los datos al formato Parquet, optimizado para el Data Lake.

- Carga: El archivo Parquet transformado se sube al bucket de destino (data\_lake\_procesado).

Paso 3: Optimización de Rendimiento y Costo-

Para garantizar que el procesamiento de archivos sea rápido y costo-eficiente, se aplican las siguientes prácticas serverless:

- Asignación de Memoria y Cómputo: En AWS Lambda, la memoria asignada está directamente ligada a la potencia de CPU. Para archivos grandes, se debe aumentar la memoria (por ejemplo, a 1.024 MB o 2.048 MB). Esto permite que las bibliotecas de procesamiento de datos (como Pandas) se ejecuten en paralelo y más rápido, lo que reduce el tiempo total de ejecución y, a menudo, el costo total (Costo = Memoria × Tiempo de Ejecución).

- Compresión y Formato: La transformación a formato Parquet (columnar y comprimido) optimiza el uso de lectura/escritura de disco en S3 y mejora el rendimiento.

- Gestión del Entorno: Usar Variables de Entorno para definir los nombres de los buckets de origen y destino. Esto hace que el código sea más portable y no requiere re-despliegue para cambios de configuración.

- Monitoreo: Se configura Amazon CloudWatch para monitorear las métricas de Duration (duración de la función) y Errors con el objetivo de establecer una alarma que notifique al equipo si la duración promedio de la función excede un límite (por ejemplo, 5 segundos), indicando un problema de rendimiento o un archivo inusualmente grande.