MODULO 09 - EJERCICIO 07-B

ALEXIS YURI M.

Diseñe una pequeña API sin servidor para dar de alta, listar y eliminar registros de una base DynamoDB. Deberá contar con al menos 3 funciones Lambda y estar conectada con API Getaway.

1. Definición de la Arquitectura.

La solución de gestión de datos se basa en la arquitectura FaaS (Función como Servicio) para lograr escalabilidad ilimitada y bajo costo.



2. Diseño de Endpoints (CRUD).

El diseño utiliza principios RESTful para mapear las operaciones de datos a métodos y recursos HTTP, permitiendo que API Gateway sepa qué función Lambda invocar.



Asociación de cada Endpoint a una función Lambda.



Cada recurso HTTP es configurado en Amazon API Gateway para actuar como un trigger (disparador) directo de la función Lambda correspondiente.

Mapping: API Gateway es responsable de realizar el mapeo de la solicitud. Por ejemplo, en el caso de GET /productos/{id}, API Gateway extrae el valor de la variable de ruta ({id}) y lo pasa como parte del payload de entrada (el evento) a la función lambda-get-product-by-id.

Separación de Responsabilidades: Utilizar una función Lambda por cada operación CRUD (una función, una tarea) es una buena práctica serverless. Esto simplifica el código de cada función, facilita el aislamiento de errores y permite optimizar individualmente los recursos (memoria, timeout) para cada tipo de operación.

3. Conexión entre AWS Lambda y DynamoDB.

La conexión de las funciones AWS Lambda con Amazon DynamoDB es un proceso que se gestiona completamente a través del servicio de AWS Identity and Access Management (IAM). La seguridad en AWS se basa en el principio del menor privilegio, asegurando que Lambda sólo tenga los permisos exactos necesarios para interactuar con la tabla de DynamoDB.

La conexión no requiere claves de acceso dentro del código de Lambda, sino que utiliza un Rol de Ejecución (Execution Role) que se asume al invocar la función.

- El Rol de Ejecución de AWS Lambda

Cuando se crea una función Lambda, se le debe asignar un Rol de Ejecución de IAM. Este rol define las acciones que la función puede realizar sobre otros servicios de AWS.

Identidad de la Función: La función Lambda utiliza este rol como su identidad al intentar

conectarse a DynamoDB.

- La Política de Permisos Específica

La política adjunta al Rol de Ejecución es la que otorga el acceso a la base de datos. Para mantener la seguridad estricta, esta política debe ser lo más restrictiva posible.

a. Acciones (Permissions): Se deben otorgar solo las acciones de DynamoDB que la API necesite para sus operaciones CRUD.

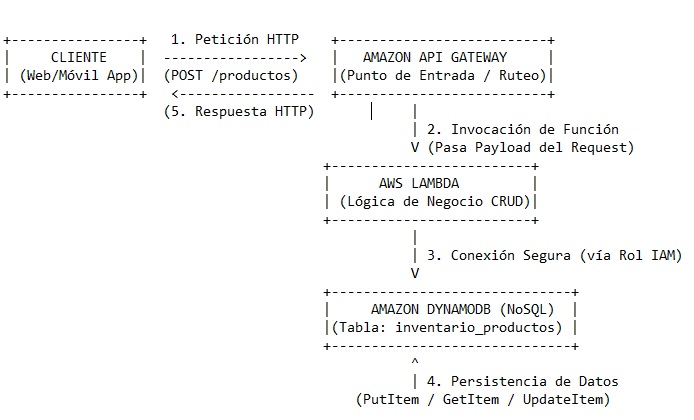


b. Recurso (Resource): El permiso debe limitarse a la tabla específica de DynamoDB, en lugar de a todas las tablas de la cuenta, utilizando el Nombre de Recurso de Amazon (ARN) de la tabla:

Formato de ARN: arn:aws:dynamodb:region:account-id:table/inventario\_productos

4. Flujo General del sistema.

Se diagrama la ruta completa que sigue una solicitud HTTP desde el cliente hasta la base de datos, y cómo el servicio de IAM habilita la seguridad.



5. Ventajas del Diseño Serverless frente a una API Tradicional.

La principal ventaja de una API Serverless frente a una API tradicional (basada en máquinas virtuales o servidores dedicados) se centra en la eliminación de la administración de infraestructura, lo que conduce a una enorme optimización de costos, escalabilidad automática y alta disponibilidad nativa.

La diferencia fundamental reside en quién gestiona los servidores y cómo se paga por el cómputo.

En resumen, la API Serverless transforma la arquitectura de una inversión de capital y esfuerzo de mantenimiento a un costo variable puramente funcional y una arquitectura inherentemente escalable y disponible.

