

Documento de Diseño - Proyecto AWS Anuncios

1. Introducción

El proyecto **AWS Anuncios** es una API serverless diseñada para la gestión de anuncios y comentarios, utilizando **AWS Lambda**, **API Gateway** y **DynamoDB**. Este documento detalla la arquitectura, el diseño de la base de datos, la lógica de negocio y las configuraciones necesarias para su implementación.

2. Arquitectura del Sistema

El sistema sigue una arquitectura **serverless** que permite escalabilidad y mantenimiento reducido. Los principales componentes incluyen:

- **AWS Lambda:** Funciones para la gestión de anuncios y comentarios.
- **Amazon API Gateway:** Exposición de los endpoints RESTful.
- **Amazon DynamoDB:** Base de datos NoSQL para almacenar anuncios y comentarios.
- **AWS IAM Roles y Políticas:** Permisos para la ejecución de funciones Lambda y acceso a DynamoDB.

Diagrama de Arquitectura

Usuario → API Gateway → AWS Lambda → DynamoDB

3. Diseño de Base de Datos

Se utilizan dos tablas en **Amazon DynamoDB**:

Tabla: `Anuncios_iad`

Atributo	Tipo	Descripción
anuncioId	String	Identificador único del anuncio
title	String	Título del anuncio
description	String	Descripción del anuncio

Tabla: `Comentarios_iad`

Atributo	Tipo	Descripción
comentarioId	String	Identificador único del comentario
anuncioId	String	Referencia al anuncio relacionado
autor	String	Nombre del autor del comentario
contenido	String	Contenido del comentario

4. Diseño de Funciones Lambda

Función: `getAnunciosiad`

Descripción: Recupera todos los anuncios almacenados en la tabla `Anuncios_iad`.

- **Entrada:** Ninguna.
- **Salida:** Lista de anuncios en formato JSON.

Función: `createAnuncioiad`

Descripción: Agrega un nuevo anuncio a `Anuncios_iad`.

- **Entrada:** JSON con `anuncioId`, `title`, `description`.
- **Salida:** Mensaje de éxito o error.

Función: `getComentariosiad`

Descripción: Obtiene todos los comentarios asociados a un anuncio específico.

- **Entrada:** `anuncioId` en los parámetros de consulta.
- **Salida:** Lista de comentarios en JSON.

Función: `createComentarioiad`

Descripción: Agrega un comentario a la tabla `Comentarios_iad`.

- **Entrada:** JSON con `comentarioId`, `anuncioId`, `autor`, `contenido`.
- **Salida:** Mensaje de éxito o error.

5. API Gateway - Endpoints

Anuncios

Método	Endpoint	Función Lambda
GET	<code>/anunciosia</code>	<code>getAnunciosiad</code>
POST	<code>/anunciosia</code>	<code>createAnuncioiad</code>

Comentarios

Método	Endpoint	Función Lambda
GET	<code>/comentariosia?anuncioId=xxx</code>	<code>getComentariosiad</code>
POST	<code>/comentariosia</code>	<code>createComentarioiad</code>

6. Seguridad y Permisos

- **CORS habilitado** en API Gateway para permitir solicitudes desde diferentes orígenes.
- **Roles de IAM** configurados para permitir a Lambda interactuar con DynamoDB.
- **Validaciones en la API** para evitar inyección de datos maliciosos.

7. Pruebas y Validación

Se recomienda probar los endpoints utilizando **cURL** o herramientas como **Postman**:

Obtener anuncios

```
curl -X GET "https://tu-api-gateway.execute-api.eu-west-1.amazonaws.com/v1/anunciosia"
```

Crear un anuncio

```
curl -X POST "https://tu-api-gateway.execute-api.eu-west-1.amazonaws.com/v1/anunciosia" \
  -H "Content-Type: application/json" \
  -d '{"anuncioId":"123", "title":"Nuevo anuncio",
    "description":"Esto es un test"}'
```

Obtener comentarios de un anuncio

```
curl -X GET "https://tu-api-gateway.execute-api.eu-west-1.amazonaws.com/v1/comentariosia?anuncioId=123"
```

Crear un comentario

```
curl -X POST "https://tu-api-gateway.execute-api.eu-west-1.amazonaws.com/v1/comentariosia" \
  -H "Content-Type: application/json" \
  -d '{"comentarioId":"cmt-001", "anuncioId":"123", "autor":"Juan",
    "contenido":"Buen anuncio!"}'
```

8. Consideraciones Finales

- **Escalabilidad:** La arquitectura serverless permite escalado automático sin gestión de servidores.
- **Seguridad:** Se recomienda habilitar autenticación mediante **AWS Cognito o IAM Authorizers**.
- **Monitoreo:** Utilizar **AWS CloudWatch Logs** para registrar eventos y analizar problemas.