Control N°3

De acuerdo con los diferentes escenarios de escalabilidad existen diversas formas en que las aplicaciones pueden conectarse a servicios en la nube y poder contar con muchas funcionalidades, una solución a esta problemática es la implementación de API y funciones serverless.

Las arquitecturas basadas en la alta escalabilidad en la actualidad en su mayoría han sido montadas sobre arquitecturas orientadas a servicios con backend serverless, ya sea para soportar trivias en los grandes superbowl (Más de 100 M de transacciones), como en los típicos cybermonday o cyberfriday, por lo que es muy apetecido a la hora de crear soluciones que requieran muchos recursos escalables en poco tiempo y los costos sean bajos.

Diagrama

Descripción generada automáticamente

En este diagrama de componentes podemos observar que cualquier host ya sea una aplicación móvil, web e incluso cajeros automáticos tótems, etc. Se pueden conectar a las API.

El API Gateway se compone como una forma de hacer llamadas REST con diferentes métodos HTTP, en general los más usados son GET (SELECT), POST(INSERT), PUT(UPDATE), DELETE(DELETE),

En general los recursos son llamados como url/PathParam, en donde la url es el recurso que fue publicado.

FAAS es donde cada vez que es llamado un GET, POST, PUT o DELETE es creada una instancia transaccional (CPU, MEM y STORAGE virtual es reunido como respuesta ante la transacción), aquí es donde esta función se convierte en el método de integración para poder realizar acciones sobre las transacciones.

Persistence para que las funciones serverless no sean cuello de botella, también es necesario pensar en como esto se debe distribuir y escalar de manera serverless, para eso existen diferentes soluciones en los clouds más conocidos como Azure, AWS, Google Cloud, Heroku, Digital Ocean, etc.

Un ejemplo de esto es AWS

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

En este control se solicita a los estudiantes de arquitectura que construyan una API que realice las operaciones CRUD básicas de persistencia de objetos, para esto se debe realizar el siguiente proceso:

1. Crear un Recurso en el API Gateway del Cloud Seleccionado /Recurso
2. Sobre el recurso agregar los métodos GET, POST, PUT, DELETE

Ejemplo en AWS

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Por cada método del API crear una función que atienda a la solicitud.

Ejemplo en AWS

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Por cada función de integración serverless agregar la acción de consulta o manipulación de datos de la base de datos NO-SQL serverless

Ejemplo en AWS

<https://docs.aws.amazon.com/es_es/amazondynamodb/latest/developerguide/GettingStarted.NodeJs.03.html>

Post

Captura de pantalla de computadora

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

PUT

Captura de pantalla de computadora

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

GET

Captura de pantalla de computadora

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Captura de pantalla con la imagen de una pantalla

Descripción generada automáticamente

DELETE

Captura de pantalla de computadora

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Captura de pantalla de un celular

Descripción generada automáticamente

API

Una captura de pantalla de una computadora

Descripción generada automáticamente

Una captura de pantalla de una computadora

Descripción generada automáticamente

Una captura de pantalla de una computadora

Descripción generada automáticamente

Una captura de pantalla de una computadora

Descripción generada automáticamente

Una captura de pantalla de una computadora

Descripción generada automáticamente