

Amazon Gateway y Lambda

Arquitectura Empresarial

Nicolás D. Cárdenas G.¹

¹ Bogotá D.C., Escuela de ingenieros, Colombia

Fecha: 30/01/2020

Resumen— Utilizando dos servicios de AWS implementaremos una arquitectura que permite desarrollar de forma mas ágil, reduciendo el tiempo de implementación de funcionalidades concretas.

Palabras clave— Gateway, Lambda, Apache spark, pojo, java, maretialize

Abstract— Expanding the server of the previous paper, additional functionality was added that allows POJOS to be used. We will deepen the implementation of the architecture that supports the wells in addition to defining topics such as annotations and reflection.

Keywords— Gateway, Lambda, Apache spark, pojo, java, maretialize

INTRODUCCIÓN

A Continuación desarrollaremos a detalle los componentes desarrollados en este experimento, Utilizamos 3 herramientas para esta ejecución, Spark (Wendel, 2020), Api Gateway y Lambda funciones.

CONTENIDO

En el primer paso es desarrollar un POJO (Plain Old Java Object Oracle (2020b)) en el lenguaje Java (Oracle, 2020a) con una funcionalidad especifica, en nuestro caso implementaremos una función que recibe un numero y retorna su cuadrado.

En el segundo paso compilaremos nuestro proyecto java en un jar o war para posteriormente ejecutarlo con lambda function, para eso usamos el servicio que provee AWS con este mismo nombre. Probaremos nuestra funcion.

Datos de contacto: Nicolás D. Cárdenas G., nicolas.cardenas@mail.escuelaing.edu.co

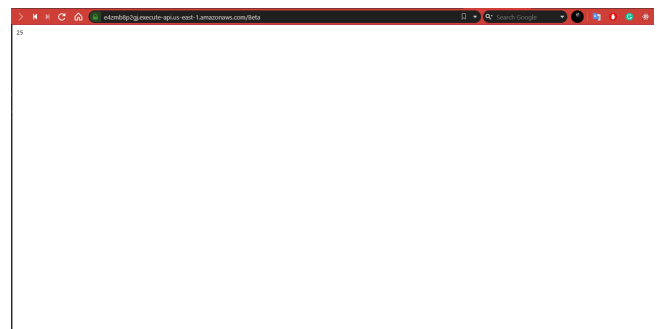


Fig. 1: Diagrama de la arquitectura

Posteriormente para el siguiente paso usaremos Api Gateway de AWS, este servicio lo conectaremos con nuestra función lambda. Para ilustrar el contexto general a continuación podemos ver un diagrama de despliegue, 2 a continuación.

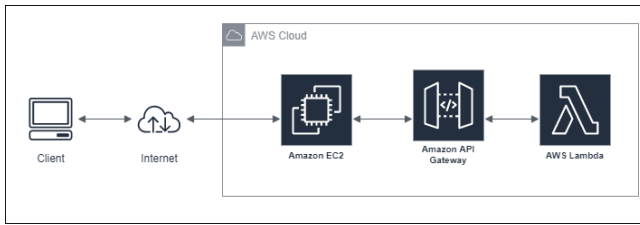


Fig. 2: Diagrama de despliegue

En este diagrama podemos ver como interactúan todos los componentes de la arquitectura. En el EC2 tenemos un servidor web con un formulario que recibirá un valor numérico que será procesado por el lambda function.

A continuación mostraremos el diagrama de componentes, se ilustra en la figura 3

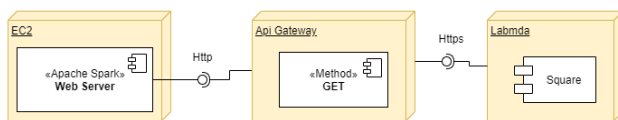


Fig. 3: Diagrama de componentes

PRUEBAS

Html renderizado en un navegador 3.

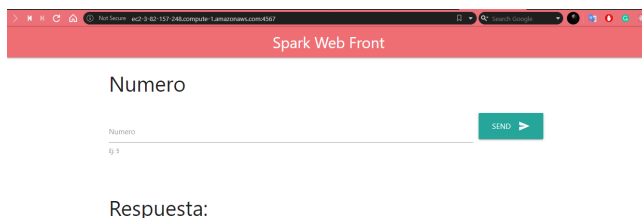


Fig. 4: Imagen del index del servidor EC2

este es el index con el formulario que recoge y manda el valor a los servicios de AWS y posteriormente los muestra.

CONCLUSIONES

Los servicios que provee AWS permiten una desagregación del mantenimiento de funciones concretas con las cuales no tenemos que soportar en nuestro servidor, esto permite que sin importar que le suceda al servidor la función lambda seguirá estable y respondiendo. Además de tener la posibilidad de escalamiento mas fácil.

Por otro la el servicio de Api gateway permite generar un endpoint visible en la red con poca configuración, esto permite que los servicios que implementemos en lambda function tengan salidad a la red por un api que mantiene AWS.

Esto genera que el desarrollador solo se enfoque en desarrollar funcionalidad, se olvide de la implementación y/o configuración de un servidor.

REFERENCIAS

- [1] Oracle (2020a). "Java™ platform, standard edition 8 api specification". Tomado de <https://docs.oracle.com/javase/8/docs/api/> (30/01/2020).
- [2] Oracle (2020b). "what is a socket?" Tomado de https://es.wikipedia.org/wiki/Plain_Old_Java_Object (30/01/2020).
- [3] Wendel, P. (2020). "Documentation - spark framework: An expressive web framework for kotlin and java". Tomado de <http://sparkjava.com/documentation> (30/01/2020).