Data Engineer Challange

Federico De Grazia

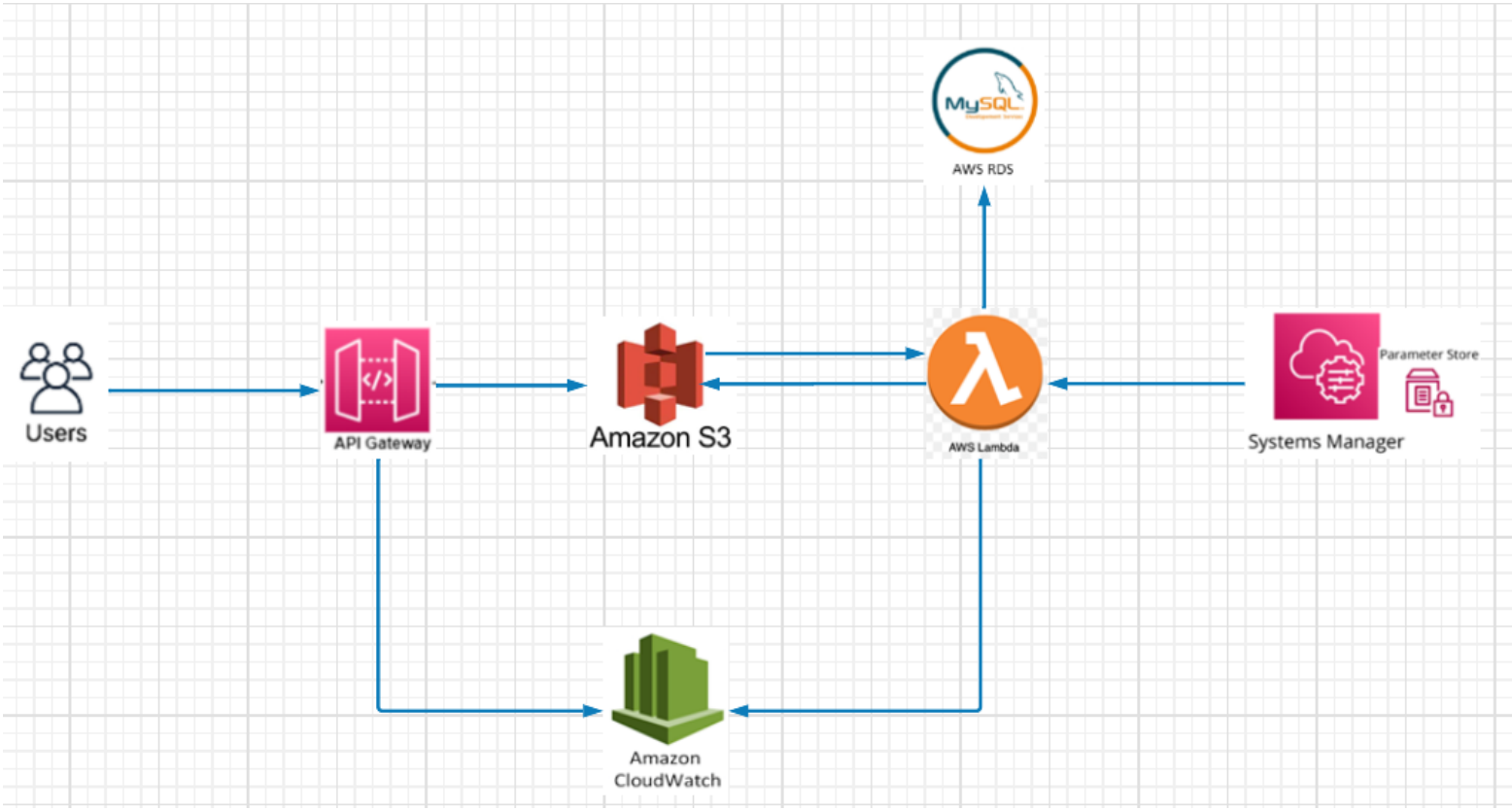
Enero 2023

Introducción

El objetivo del presente documento es explicar cómo está compuesto el entregable y brindar una explicación de la solución propuesta, detallando cada uno de sus componentes.

Arquitectura

El siguiente diagrama presenta una propuesta de solución en la plataforma de aws:



Funcionalidades de servicios:

AWS API Getway:

Se implementó una API, “https://9cev9nfpo4.execute-api.us-east-1.amazonaws.com/dev”, con el método “Put” para la carga de archivos .csv en el bucket S3 que será el área de destino para los mismos.

La invocación de la API debe hacerse según la siguiente estructura:

curl --request PUT -H "Content-Type: */csv" --data-binary "@<Fullpath to csv>" https://9cev9nfpo4.execute-api.us-east-1.amazonaws.com/dev/<bucket> /<filename>

Un ejemplo usando el archivo de empleados seria:

curl --request PUT -H "Content-Type: */csv" --data-binary "@C:\Users\Usuario\Desktop\hired_employees.csv" https://9cev9nfpo4.execute-api.us-east-1.amazonaws.com/dev/globant-landing/hired_employees.csv.

En el archivo “apiOpen3.json”, el resutlado de exportar dicha API como un OpenAPI3.

AWS S3:

Es el filesystem donde se almacenarán los archivos que involucran esta solución.
Por un lado, tenemos al bucket “globant-landing” como lugar de almacenamiento de los archivos csv provenientes del servicio de API Getway, esta acción de creación de objetos tipo csv dispara una notificación de evento que funciona como disparador de una función programa en AWS Lambda, encargada del procesamiento del archivo depositado.
Otro bucket utilizado es el “globant-invalid-data” donde se escribe un archivo nuevo tipo csv con los datos del archivo original que fueron descartados en la lógica de la Lambda por no cumplir las reglas definidas en el enunciado, para su posterior revisión.

AWS Lambda:

Es el servicio encargado de ejecutar la lógica de procesamiento de los archivos csv suministrados. La lógica de la misma consiste en:

1. Establecer conexión con los servicios de aws a interactuar mediante la SDK de Python, para nuestro caso aws s3 y aws system manager.
2. Conexión y ejecución de consultas contra la base de datos.
3. Procesamiento de la información del csv, armado de los datasets a impactar, tanto en aws s3 como en la base destino.

El paquete con la función principal y librería utilizada para mysql se pueden encontrar en el archivo: “deploy_package.zip”.
Adicionalmente se le agrego la librería de pandas por medio de la funcionalidad Layer.

AWS CloudWatch:

En él son almacenados los logs de ejecución de los servicios AWS API Getway y AWS Lambda, posibilitando el seguimiento, debugging y registro histórico de los mismos.
Los siguientes archivos son ejemplos de logging en este servicio:

1. “api_getway_log_example.txt”
2. “lambda_log_example.txt”

AWS Parameter Store:

Funciona como almacenamiento, potencialmente compartido entre varios recursos, de información clave valor.
Para nuestro caso, aws lambda consulta este servicio para conocer los datos de conexión a la base de datos destino, en específico estos son: host, nombre de la db, usuario y contraseña. De los cuales los últimos dos están encriptados por seguridad.

AWS RDS:

Es la base de datos utilizada en esta PoC como almacenamiento final de los datos procesados. Se trata de una base MySQL relacional donde fueron creadas las tablas y las posteriores queries de la segunda parte del desafío.
Entre la configuración de la misma, se estableció la generación automática de backups por medio de snapshots cada semana.
La lógica de las ddls de las tablas y las queries se pueden observar en los archivos “DDL.sql” y “challange2.sql” respectivamente.