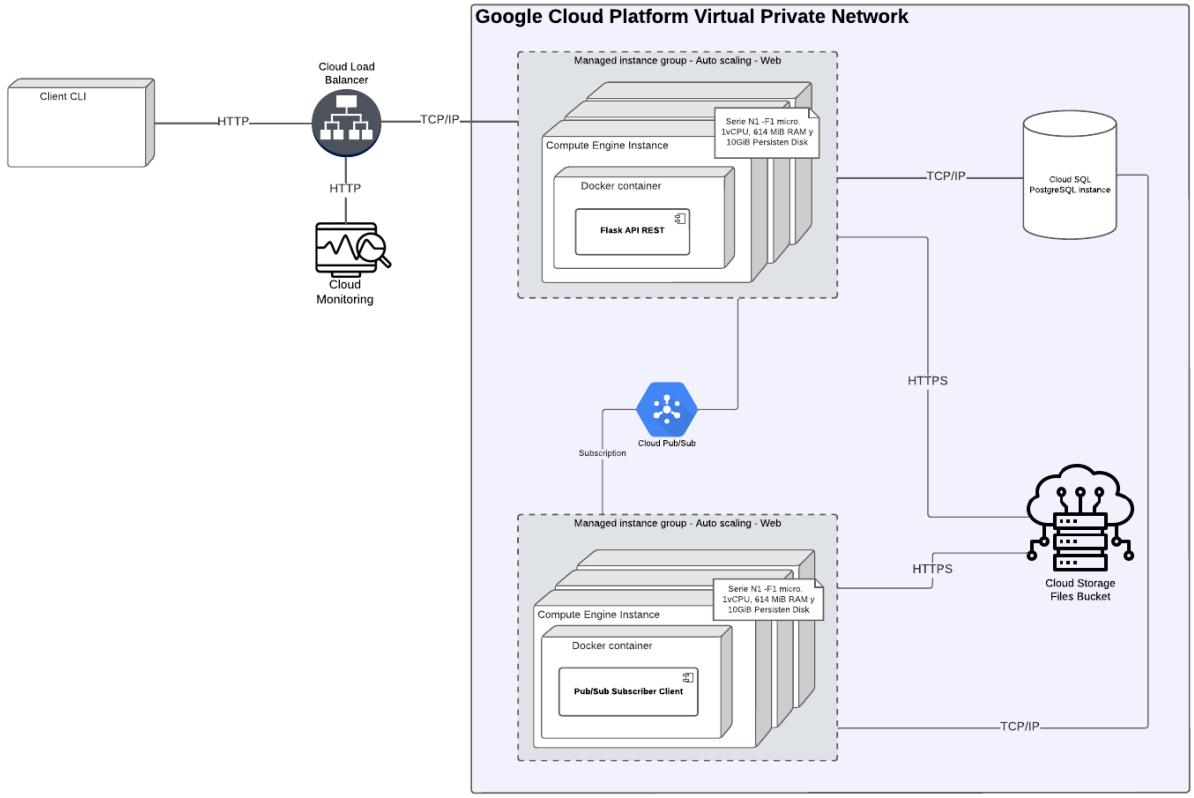
Arquitectura, conclusiones y consideraciones

Entrega 4 - Escalabilidad En El Backend Diseño E Implementación De Una Aplicación Web Escalable En La Nube Pública

Diego Alejandro Camelo Giraldo, Juan Sebastián Alegría Zúñiga, María Camila Gómez Hernández, Andrés Felipe Lugo Saavedra

Grupo 6

Arquitectura de la aplicación



Para esta entrega, a comparación de la anterior, no enfocamos en la escalabilidad automática de nuestro sistema de conversión de archivos para una sobrecarga de peticiones en la misma. Para lograr esto, como grupo implementamos un grupo auto escalable de instancias de Compute Engine por parte de la API REST y del worker, de esta manera aseguramos que cada aplicación va a porder soportar un gran número de peticiones simultaneas, cada programa utilizado está desplegado en un contenedor de Docker. Además, se implementó un balanceador

de carga que lograr distribuir el gran número de peticiones entrantes, y un servicio de monitoreo conectado al mismo. Por último, se reemplazó el Network File System por un Bucket de Cloud Storage. Por último, se creó el sisema de comunicación Pub/Sub de Google Cloud que permite notificar que se subió un nuevo archivo que debe ser procesado, desde la API REST, al worker. Este nuevo componente nos permite simplificar el código realizado, debido a que en esta entrega no tenemos que implementar Celery para el manejo de colas (ya lo hace Cloud Pub/Sub).

Conclusiones identificadas de las pruebas de estrés

Una vez concluidas las pruebas de estrés, se pudo concluir que la arquitectura logro reducir satisfactoriamente los tiempos de espera entre cada solicitud. Sin embargo, su mayor aporte fue reducir el porcentaje de errores que solía presentarse cuando se llegaba a altas demandas en el servicio. Principalmente en el escenario 1 en donde se atendieron 100000 solicitudes concurrentes sin que estas fallaran o se desconectaran. Por lo que la disponibilidad fue uno de los atributos más beneficiados de esta propuesta de arquitectura.

Limitaciones

Durante la construcción de la aplicación, logramos identificar varias limitantes que pueden afectar el correcto funcionamiento de nuestra aplicación. La principal limitante o punto de observación son los grupos de auto-scaling, se dice esto debido a que manejar un elevado número de instancias para soportar la gran cantidad de peticiones de los usuarios puede generar unos costos añadidos no previstos para el negocio. Además, la inclusión de un balanceador de carga puede generar mayor latencia durante el procesamiento de peticiones. Por último, se debe tomar en cuenta el tiempo que tenemos para desarrollar la aplicación y la velocidad de desarrollo del equipo, pues esto puede afectar la capacidad de soluciones que se pueden realizar.