Arquitectura, conclusiones y consideraciones

Entrega 5 – Despliegue en PaaS

Migración de una aplicación web a una plataforma como servicio en la nube pública

Integrantes: María Camila Gómez Hernández (202011050), Andrés Felipe Lugo Saavedra (202012538), Juan Sebastián Alegría Zúñiga (202011282)

Arquitectura de la aplicación

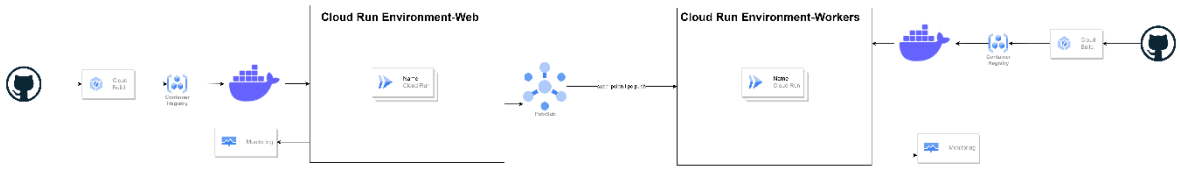

Ilustración 1. Link del diagrama: https://drive.google.com/file/d/1yDi4eFnIOoAr1i4vSl_e_Nz2RLkiJh6n/view?usp=sharing

Para esta entrega, migramos toda nuestra infraestructura de servidores a Cloud Run, tanto los workers como el servidor web. Cloud Run toma las imágenes generadas por Cloud Build para construir los servidores automáticamente y se irá actualizando cada vez que el repositorio del código de la aplicación sea actualizado.

Para el ambiente que tenemos en Cloud Run, tenemos registrado Cloud Monitoring para vigilar la salud de las cosas que tenemos y el tráfico generado.

Como en las entregas anteriores, utilizamos pub/sub para agendar las tareas para el worker, el worker está suscrito a las notificaciones de pub/sub e irá haciendo las tareas encoladas.

Despliegue de la aplicación

Despliegue de la base de datos - Cloud SQL

En primer lugar, cree una instancia de Cloud SQL en la misma región en la que creó las instancias de Compute Engine. Asegurese que utilice PostgreSQL 14.

Luego, chequee la opción de asignar IP privada, esta es la que va a utilizar para hacer llamado en la base de datos. Verifique que la maquina pertenezca a la misma VPC que las demás maquinas desplegadas.

Luego de haber chequeado la opción, en caso de que su proyecto no tenga activadas las APIs necesarias, GCP lo redirigirá a un tutorial para activarlas y asignar correctamente esa IP privada a la VPC correspondiente.

Una vez la instancia se haya creado, reemplace la IP pública de la base de datos contenida en los archivos worker/app y __init__.py. Solo debe reemplazar la parte contenida después del @ y antes del /.

Despliegue del sistema de Cloud Storage

Para comenzar, se creó un bucket en Cloud Storage y se otorgaron los permisos necesarios para que las instancias puedan acceder a él. En los containers del worker

y de la aplicación, se configuraron un par de parámetros adicionales para realizar la conexión con el bucket de Cloud Storage.

Configuración del Secret Manager

Recuerde crear un secret con las credenciales de autenticación en formato JSON, para que los contenedores puedan utilizar servicios, como por ejemplo Cloud Storage.

Configuración de Pub/Sub

En primer lugar, se debe crear un tópico al que llegaran eventos con el nombre de "file_system_notification". Luego de esto, asocie una subsripción, llamada "worker_subscription" de tipo push hacia el endpoint del worker que más tarde configurará en Cloud Run.

Creación de servicios en Cloud Run

En Cloud Run, seleccione "Create Service".

Seleccione la opción "Continuously deploy new revisions from a source repository", y siga los pasos para conectar un repositorio con Google Cloud Build. Esto creará un trigger que actualizará automáticamente los contenedores del servico.

Configure reglas de autoscaling, capacidad y puertos.

Añada un secret que enlace el creado para las credenciales. Debe estar montado como volumen en /app/credentials/, con el nombre de google-credentials.json.

Opcionalmente, podrá crear un health check del servicio.

Cree una conexión a la base de datos en Cloud SQL.

Conclusiones de pruebas de estrés

Una vez finalizadas las pruebas de estrés, logramos detectar que los cuellos de botella pasaron a ser un poco más implícitos que en otros casos. La cantidad de usuarios que se pueden atender simultáneamente logro aumentar de forma considerable, y los porcentajes de error fueron casi inexistentes en este caso. Por lo que podemos concluir que el atributo de disponibilidad fue exitosamente satisfactorio.

Consideraciones

Durante el desarrollo de la aplicación, logramos identificar varias limitantes que pueden afectar el correcto funcionamiento de nuestra aplicación. Lo principal es que, al estar usando un servicio que crece bajo demanda, puede que en algunos casos se presenten cuellos de botella a la hora de manejar un número elevado de solicitudes, ya que en el proceso de encendido de los cloud-runs, puede que pase tiempo y las solicitudes tarden más en ser respondidas. Además, el auto-escalado puede generar costos imprevistos para el negocio. Finalmente, se tiene que tener en cuenta la velocidad de desarrollo del equipo y el tiempo para entregar un producto final, ya que estos factores pueden tener impacto en la aplicación entregada.