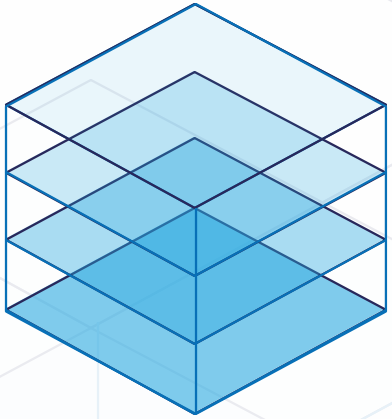


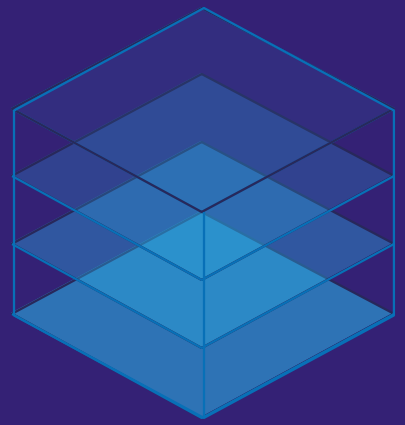
MISO

Maestría en Ingeniería de Software

Entrega 5 Sistema Conversión Cloud

Sistema de Conversión Cloud – Despliegue PaaS

Camilo Ramírez Restrepo
Laura Daniela Molina Villar
Leidy Viviana Osorio Jiménez
Tim Ulf Pambor
Shadit Perez

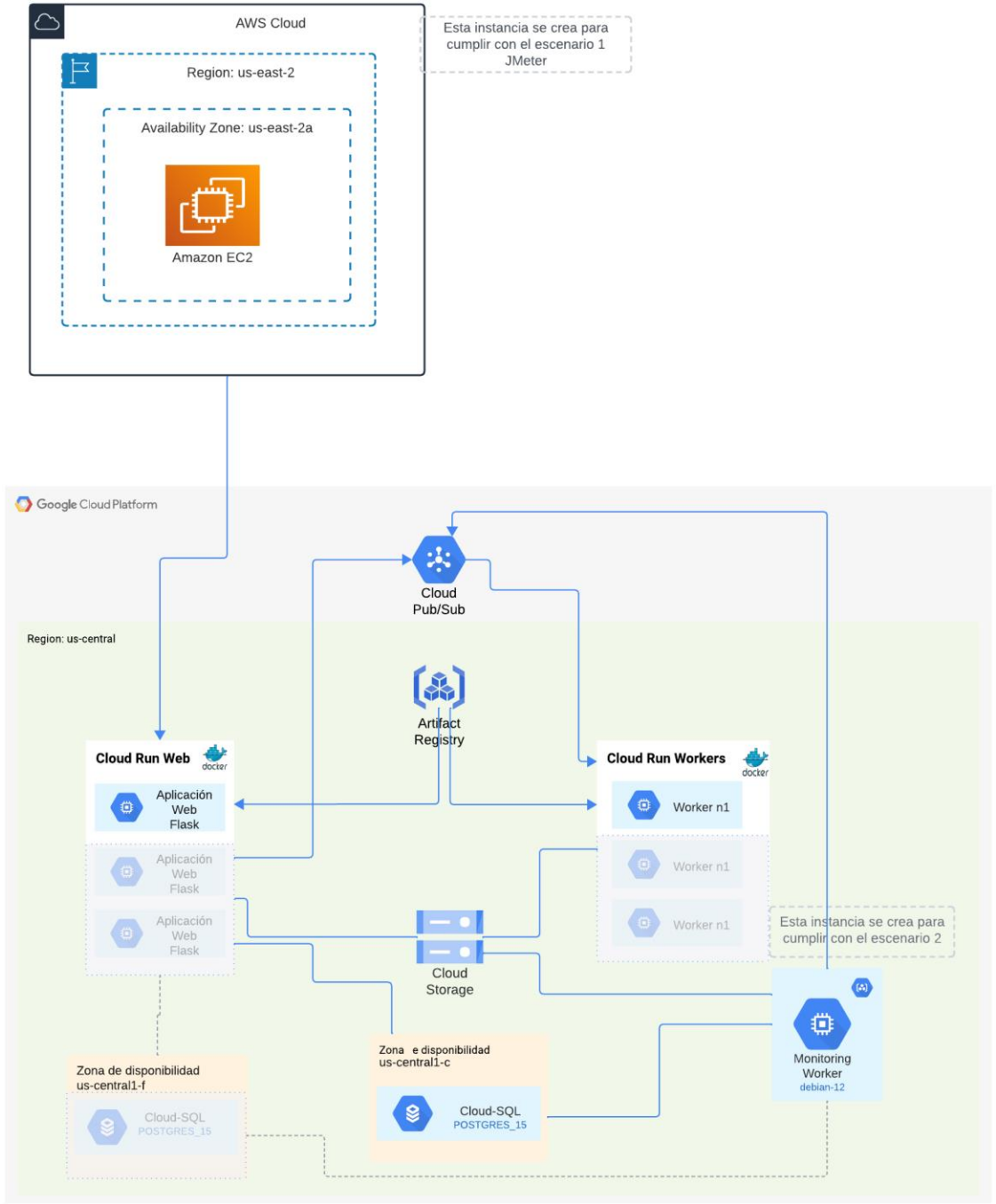


MISO

Maestría en Ingeniería de Software

Arquitectura

Arquitectura



Configuración del sistema

Entorno GCP

- Google Cloud Monitoring: Utilizamos Google Cloud Monitoring para supervisar y gestionar nuestro entorno en la nube.
- Cloud Storage: Para el almacenamiento de videos, hemos integrado Cloud Storage, aprovechando su robusta capacidad de almacenamiento en la nube.
- Cloud SQL Postgres: Para implementar una base de datos Postgres con 1 vCPU y 4 GB de memoria, garantizando un rendimiento eficiente y escalable. Trabaja en dos zonas de disponibilidad para garantizar una alta disponibilidad.
- Pub/Sub: Es un servicio de mensajería y publicación/suscripción en GCP. Se utiliza para la comunicación asíncrona entre la capa web y la capa batch.
- Cloud Run: Se usa Cloud Run para ejecutar los contenedores de la aplicación web y del worker. Los contenedores contienen todo lo necesario para ejecutar la aplicación. Cloud Run puede escalar automáticamente en función de la carga de tráfico HTTP.
    - Cloud Run Service "API": Recursos del contenedor 2 CPUs, 2 GB de memoria (igual a los recursos de las máquinas virtuales en las entregas pasadas para facilitar la comparación de resultados)
    - Cloud Run Service "Worker": Recursos del contenedor 4 CPUs, 2 GB de memoria (igual a los recursos de las máquinas virtuales en las entregas pasadas para facilitar la comparación de resultados)
- Artifact Registry: Se utiliza para gestionar las imágenes de los contenedores Docker
- Compute Engine: Se ha implementado una instancia Compute Engine llamada "monitoring-worker", utilizando una instancia e2-highcpu-2, para ejecutar las pruebas de estrés del escenario 2.

Experimento 1

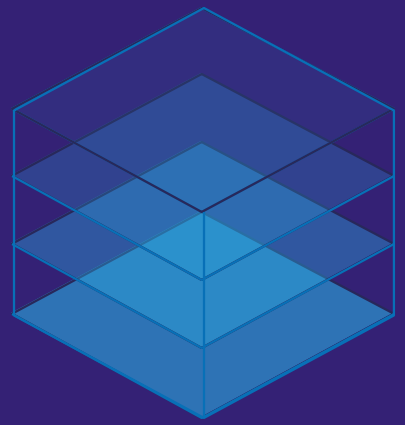
El objetivo de este plan es evaluar la capacidad de la aplicación Cloud conversión tool y su infraestructura de soporte en un entorno Cloud con despliegue PaaS y alta disponibilidad en la capa web y batch, para determinar sus máximos aceptables. El objetivo es comprender cómo la aplicación responde a diferentes niveles de carga de usuarios y cuál es su capacidad máxima.

• Configuración AWS y JMeter: Documento

Esta arquitectura combina servicios de AWS y Google Cloud para crear un entorno distribuido y escalable que satisface los requisitos del Escenario 1, con pruebas de rendimiento realizadas desde una instancia de JMeter en AWS y la infraestructura principal en Google Cloud.

Despliegue Arquitectura GCP

Configuración GCP infraestructura como código: Documento de configuración
Documento Escenario y pruebas de estrés Api Rest y Batch



MISO

Maestría en Ingeniería de Software

Experimento 1 – Agregar tarea a la cola

Configuración del sistema

Servidores EC2

- Capacidad de CPU: Cada instancia t2.medium ofrece 4 vCPUs, permitiendo la ejecución simultánea de múltiples instancias para simular usuarios concurrentes.
- Sistema Operativo: Utilizaremos Debian 12 en las instancias EC2 para llevar a cabo las pruebas.
- Configuración de Red: Las instancias EC2 proporcionan una conexión de red de hasta 10 Gbps para satisfacer los requisitos de ancho de banda necesarios para la transmisión de videos.
- Implementación sobre la instancia AWS: Se llevará a cabo la instalación de JMeter sobre la instancia para facilitar y gestionar las pruebas de rendimiento.

Experimento 1

Métricas

Throughput: cantidad de peticiones procesadas por minuto.

Tiempo de respuesta (P95): percentil 95% del tiempo máximo que tarda la aplicación en procesar una petición

Tiempo de respuesta (P99): percentil 99% del tiempo máximo que tarda la aplicación en procesar una petición

Utilización de recursos: monitoreo de la CPU, memoria y uso de red durante las pruebas.

Criterios de aceptación

Throughput: La aplicación debe ser capaz de procesar 100 peticiones por minuto.

Tiempo de respuesta:

- El tiempo de respuesta de la aplicación en todos los escenarios de prueba no debe superar los 0.5 segundos en el 95% de las transacciones, por petición.
- El tiempo de respuesta en ningún escenario de prueba debe superar los 4 segundos en el 99% de las transacciones.

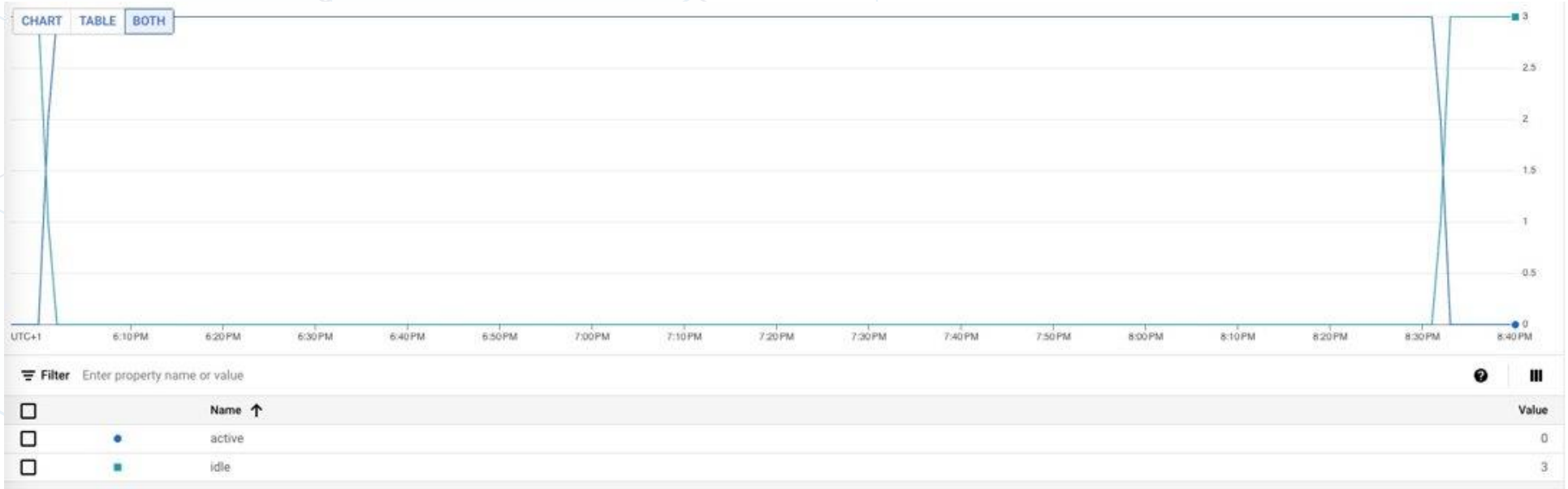
Utilización de recursos: Durante las pruebas con 100 peticiones concurrentes, la CPU del servidor alcanza un pico de 80% y la memoria se mantendrá < 80% de uso.

Resultados

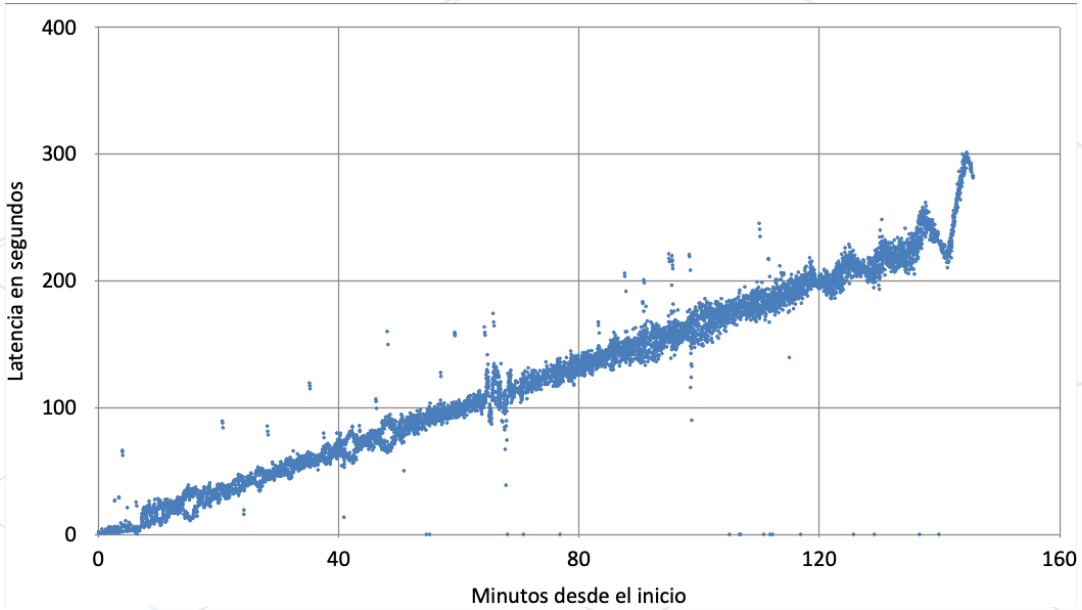
El experimento se inició con 1 usuario concurrente. Cada 30 segundos se agregó un nuevo usuario concurrente, y a lo largo del tiempo se observaron los siguientes eventos clave:

- **Inicio**: Inicio del experimento a las 5:59 pm. Se activaron tres contenedores.
- **Colapso:** El sistema colapsó completamente con 286 usuarios concurrentes a las 8:22 pm.
- Fin de pruebas de estrés con JMeter a las 8:29 pm.
- Inactivación: A las 8:33 pm los tres contendores vuelven a estar inactivos (idle).

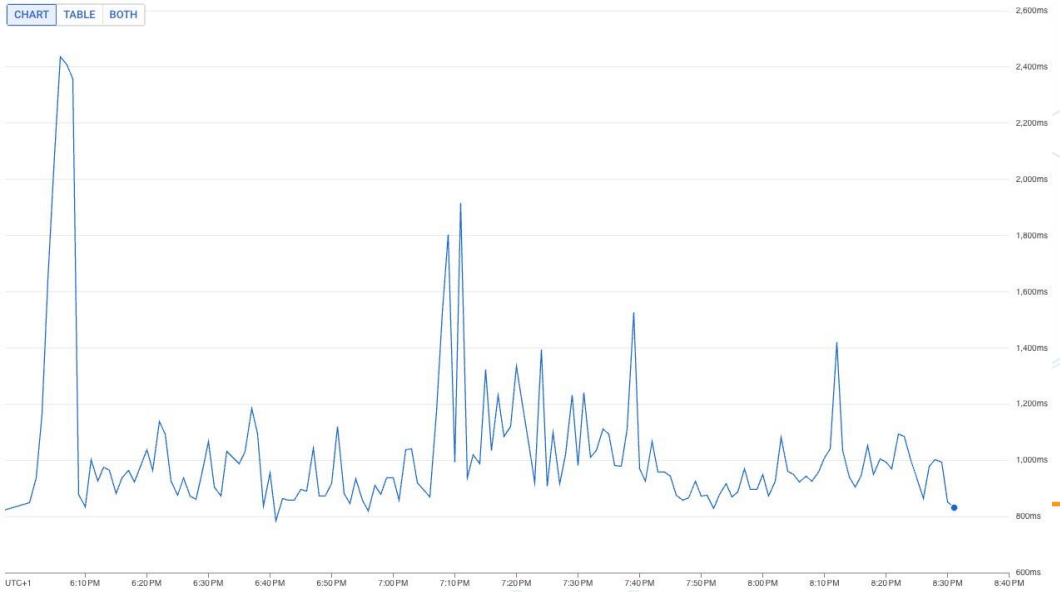
Se generaron gráficas para visualizar diferentes aspectos del sistema, incluyendo el uso de CPU, uso de RAM, la latencia y el tráfico de red.



Resultados - Latencia



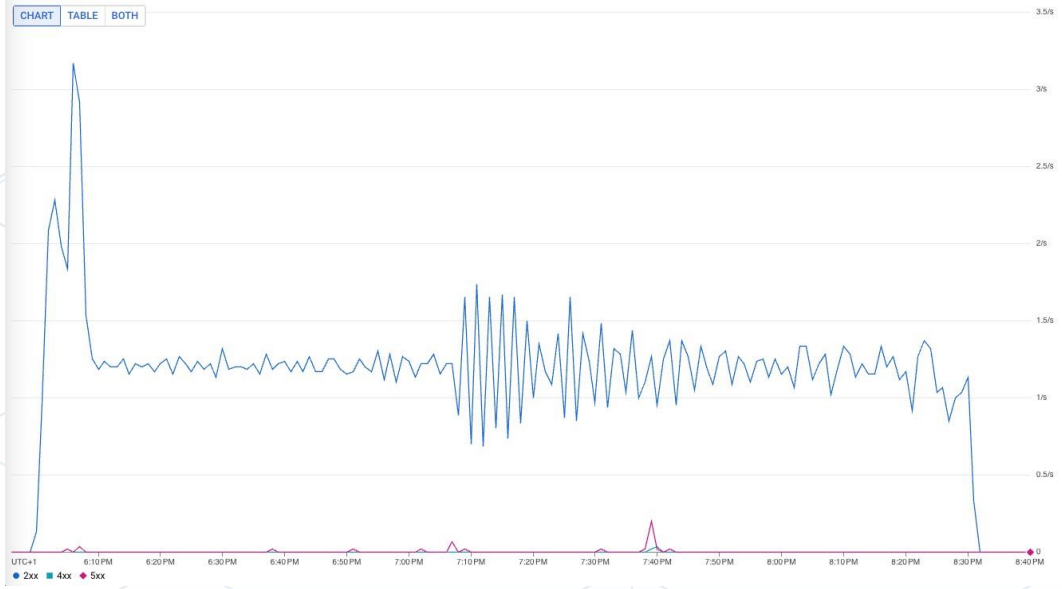
Resultados - Latencia Cloud Run



La latencia media aumenta linealmente con el número de usuarios. Esto se puede atribuirse al hecho de que el ancho de banda de red entre la máquina en AWS y GCP es fijo y no aumenta con los usuarios, lo que resulta en menos ancha de banda por usuario con un número creciente de usuarios.

Esto también se puede confirmar consultando la latencia Cloud Run, que sólo representa la latencia de responder una petición sin el componente de red entre GCP y AWS. Permanece constante en un segundo y así confirma que el sistema colapsa debido a la sobrecarga de red.

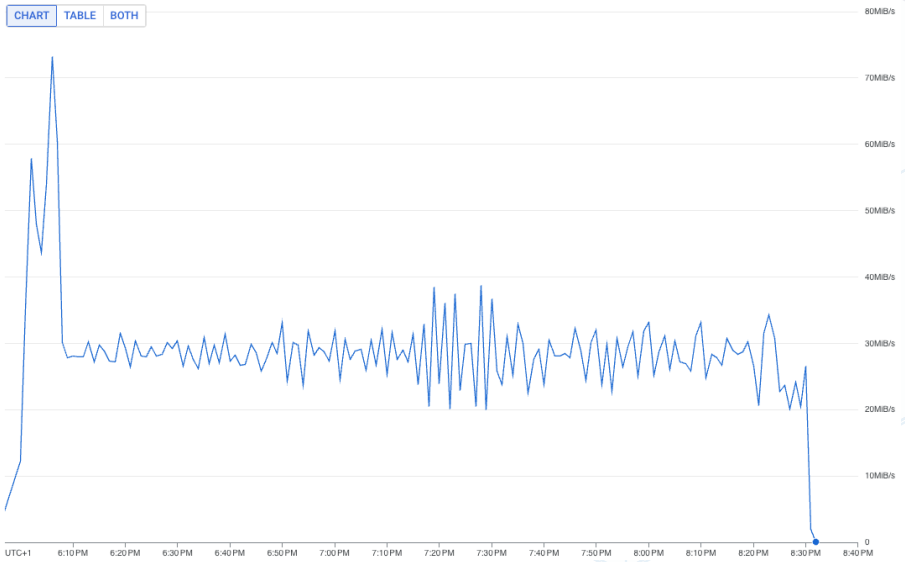
Resultados - Peticiones por segundo



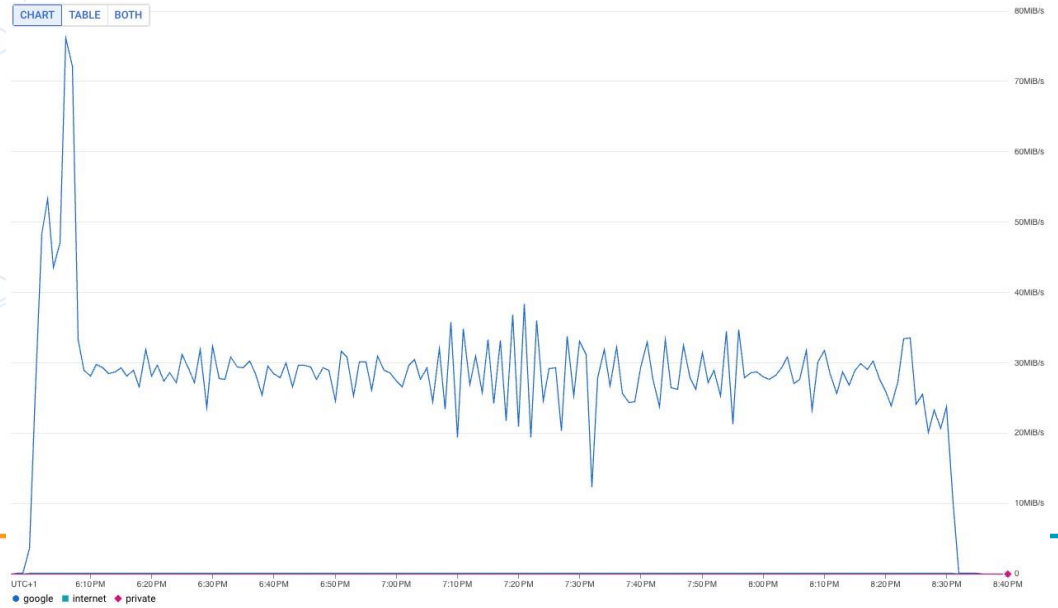
El rendimiento (throughput) se mantiene estable y constante a lo large de todo el experimento. Al inicio del experimento se observa un pico debido a la capacidad de burst (aumento de recursos durante un breve periodio de tiempo).

La utilización de Cloud Storage y red se mantiene constante debido al ancho de banda fijo entre la máquina en AWS y GCP.

Resultados - Cloud Storage



Resultados - Recursos utilizados Red



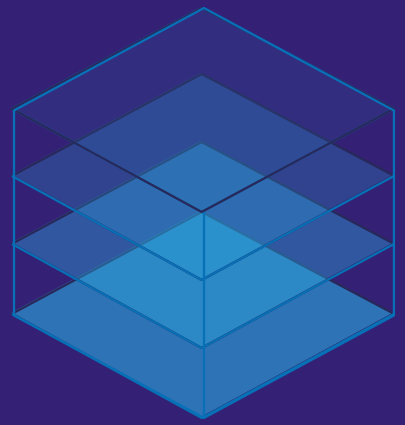
Resultados - Recursos utilizados CPU

Resultados - Recursos utilizados Memoria

La utilización de la CPU se mantuvo entre un 15% y un 30%, mientras que la utilización de memoria fue del 8% a 12%. No se observó ninguna limitación de la CPU ni de la memoria, porque el sistema ya estaba limitado por la red.

Reflexiones y conclusiones

- El experimento evidenció una mayor capacidad de escalabilidad del sistema al pasar de 1 a 286 usuarios concurrentes. En el experimento anterior con despliegue IaaS el sistema colapsó con 226 usuarios concurrentes.
- El procesamiento alcanzo niveles superiores sin colapsar debido a limitaciones de memoria. Cloud Run ha proporcionado una infraestructura más eficientemente optimizada para este propósito. Anteriormente, el servidor Nginx se veía obligado a almacenar temporalmente las solicitudes hasta que se recibieran por completo y pudieran procesarse con la aplicación Flask. Ahora, gracias a Cloud Run, esta operación se realiza de manera más eficiente, reduciendo considerablemente el consumo de recursos.
- El rendimiento de Throughput con 1.22 req/s es comparable a las soluciones IaaS con 1.25 req/s. Se recomienda evaluar y ajustar la capacidad del sistema para cumplir con los criterios de aceptación establecidos en términos de tiempo de respuesta y utilización de recursos, dado que el objetivo es procesar 1.66 req/s (100 peticiones por minuto).
- Aunque la carga de usuarios aumento, el rendimiento del sistema se mantuvo estable.
- El despliegue de contenedores se realiza mucho más rápido (6 segundos) en comparación con instancias de cómputo que necesitaron 3 minutos para iniciar. Esto permite que el sistema puede ajustarse más rápido a la carga.
- A pesar del colapso de la red, la CPU y la memoria no presentaron limitaciones, con un rango entre el 15% y el 30% para CPU y con 8% al 12% para la memoria.



MISO

Maestría en Ingeniería de Software

Experimento 2 – Conversión de videos

Experimento 2

Análisis de desempeño del componente encargado de convertir videos, actualizar el estado en la base de datos y guardar el video convertido, de manera que el video quede listo para ser solicitado por el usuario.

• Configuración: Documento

Experimento 2

Métricas

Throughput: Cantidad de peticiones por minuto en la conversión de videos.

Tiempo de respuesta promedio: Tiempo promedio que tarda la aplicación en procesar las tareas de conversión**.**

Tiempo de respuesta (P95): Percentil 95% de tiempo máximo que tarda la aplicación en procesar una tarea.

Criterios de aceptación

Throughput: Capacidad de procesar 100 peticiones por minuto.

Tiempo de respuesta:

- El tiempo de respuesta no debe superar los 40 segundos.
- No debe superar los 120 segundos en el 95% de las transacciones, por petición.

Utilización de recursos: Durante las pruebas con 100 peticiones concurrentes, la CPU del servidor alcanza un pico de 80% y la memoria se mantendrá < 80% de uso.

Métricas obtenidas

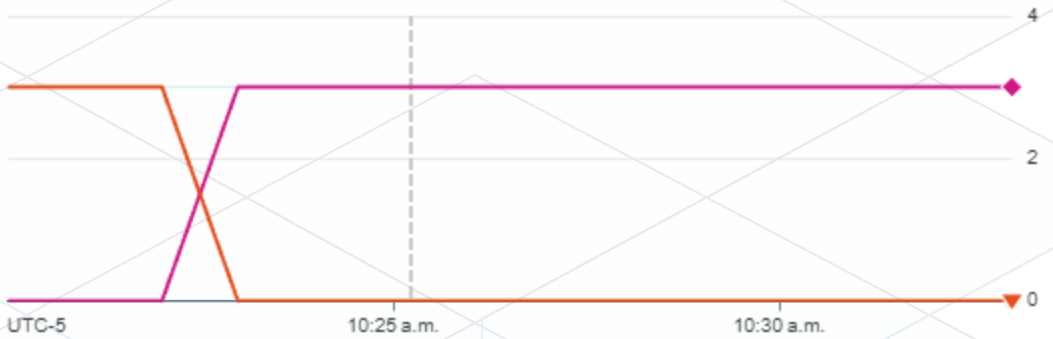
Instancias mínimas y máximas: 3

Datos	Caso 1	Caso 2	Caso 3
Peticiones concurrentes	5	10	20
Total peticiones	10	20	40
Tiempo respuesta promedio por petición (sec)	403,192	650,166	1327,79
Tiempo respuesta P95 (sec)	713,07	1145,64	2408,132
Peticiones por minuto (*Throughput*):	0,838	1,024	0,94

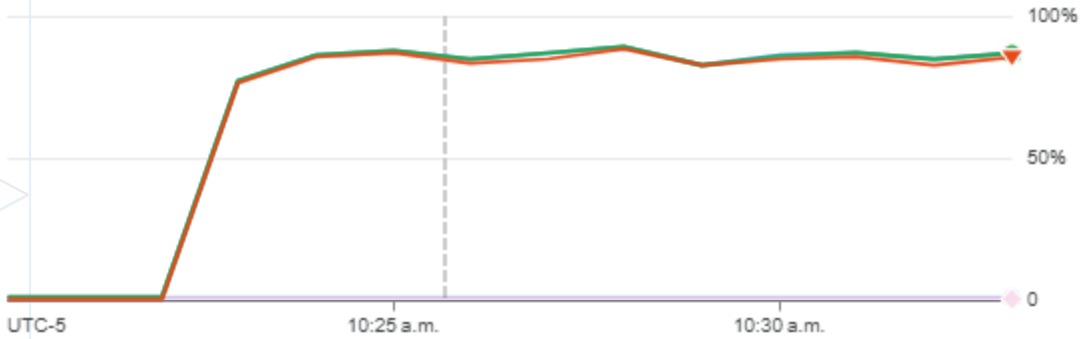
Revisar resultados obtenidos: Resultados

Resultados – Caso 1

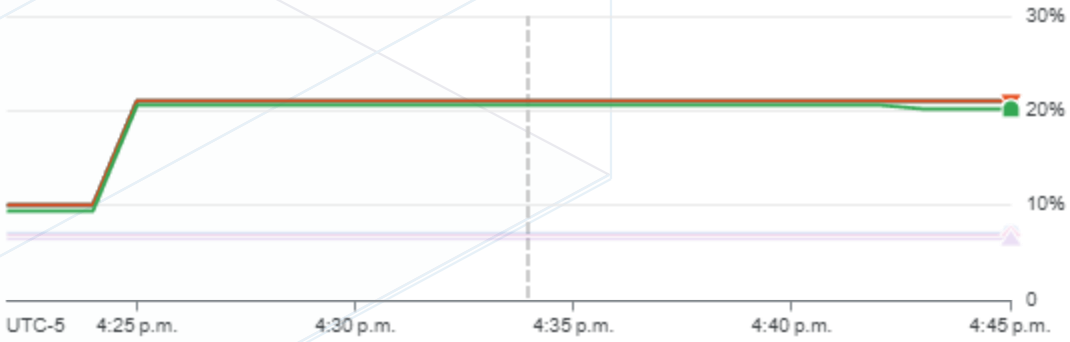
Cantidad de instancias de contenedor



Uso de CPU de contenedor



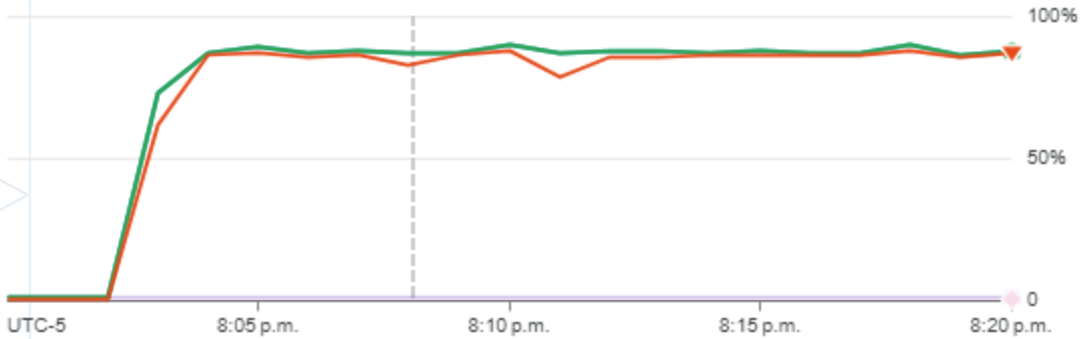
Uso de memoria de contenedor



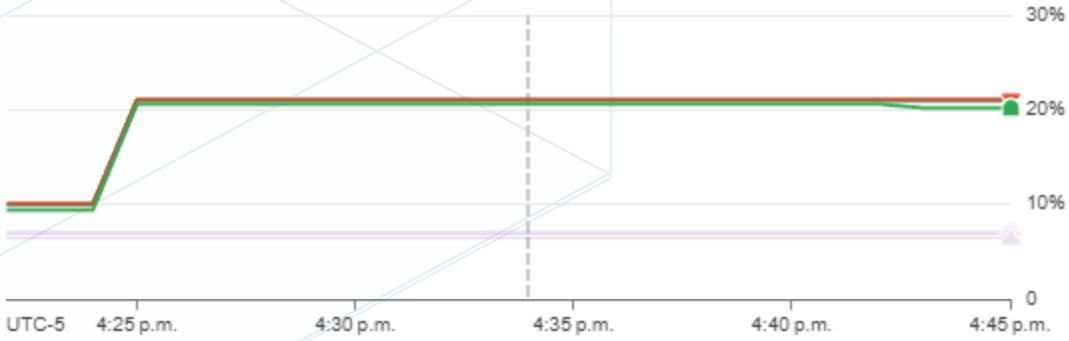
Resultados – Caso 2

Cantidad de instancias de contenedor

Uso de CPU de contenedor

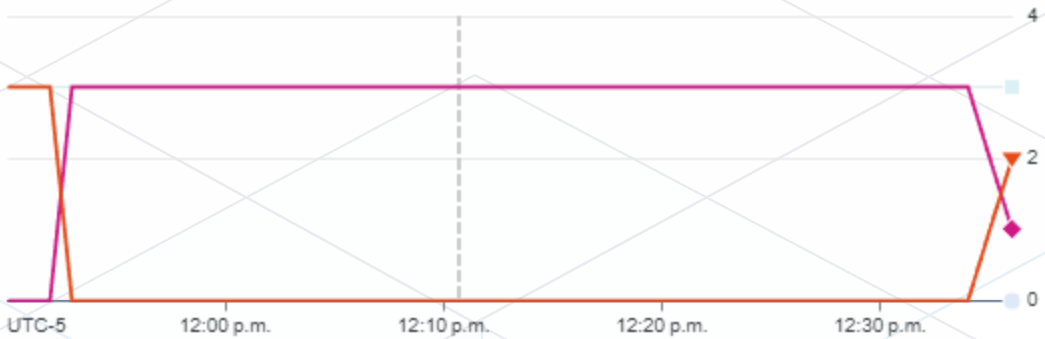


Uso de memoria de contenedor

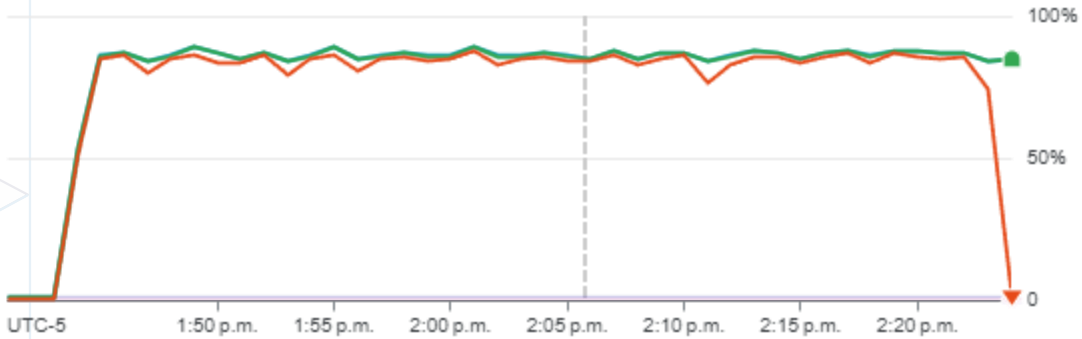


Resultados – Caso 3

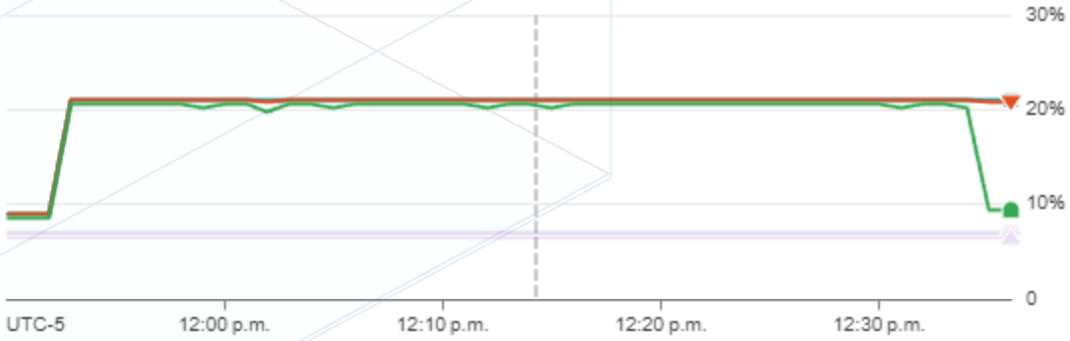
Cantidad de instancias de contenedor



Uso de CPU de contenedor



Uso de memoria de contenedor



Comparación con resultados anteriores

Caso	Métrica	Primera entrega	Segunda entrega	Tercera Entrega	Cuarta Entrega
Caso 1	Tiempo respuesta promedio por petición (sec)	408,6708	410,49342	300,0921733	403,192
	Tiempo respuesta P95 (sec)	707,3242	713,68513	451,6734433	713,07
	Peticiones por minuto (*Throughtput*):	0,81	0,81	1,3	0,838
Caso 2	Tiempo respuesta promedio por petición (sec)	803,567	776,08173	464,9051167	650,166
	Tiempo respuesta P95 (sec)	1462,7432	1406,7539	707,7307167	1145,64
	Peticiones por minuto (*Throughtput*):	0,798	0,812	1,64	1,024
Caso 2	Tiempo respuesta promedio por petición (sec)	1521,9354	1549,6144	716,09512	1327,79
	Tiempo respuesta P95 (sec)	2825,5324	2869,7137	1161,92874	2408,132
	Peticiones por minuto (*Throughtput*):	0,82	0,794	1,946666667	0,94

Análisis de resultados

- **La utilización máxima de CPU superó el 80% en todos los casos,** por lo cual no se cumple con dicho criterio de aceptación.

- **En ninguno de los 3 casos se cumplió con el criterio de aceptación planteado para el tiempo de respuesta de cada petición** (menor o igual a 40 segundos), por el contrario, aumentó más a medida que se realizaban más peticiones, pero en definitiva disminuyó mucho esta métrica con respecto a **algunas** entregas anteriores (entrega 1 y 2 donde no había autoescalamiento).

- **El número de peticiones atendidas por minuto fue se mantuvo en un rango de 0.83 a 1,** por lo que, a pesar de que son mejores resultados que las entregas 1 y 2, está lejos de las 100 peticiones por minuto solicitadas.

Reflexiones y conclusiones

El empleo de contenedores a través de Cloud Run evidenció mejoras respecto a las primeras dos entregas, aunque **no alcanzó un nivel de rendimiento superior al obtenido en la tercera entrega**. Esta disparidad puede atribuirse a la capacidad de configuración detallada que ofrecían las máquinas virtuales personalizadas en la entrega 3. Dichas VM permitían ajustar no solo la cantidad de memoria y CPU, sino también otros parámetros como el sistema operativo y el tipo de máquina, en este caso, e2-highcpu-2, aspectos que influyeron significativamente en el desempeño del worker.

Un aspecto destacado es el **ahorro económico** que Cloud Run aportó al experimento. A lo largo de la prueba, los costos se mantuvieron por debajo de los 10 dólares, incluso al dejar Cloud Run activo durante períodos de inactividad. En contraste, la utilización de máquinas virtuales sin momentos de inactividad generaba un gasto de aproximadamente 18 dólares.

En resumen, aunque Cloud Run mostró ciertas mejoras en comparación con las primeras entregas, la capacidad de personalización ofrecida por las máquinas virtuales en la tercera entrega sigue siendo un factor crucial. No obstante, el evidente ahorro monetario brinda a Cloud Run una ventaja considerable, sugiriendo que, dependiendo de las prioridades del proyecto, podría ser una opción más eficiente desde el punto de vista financiero.