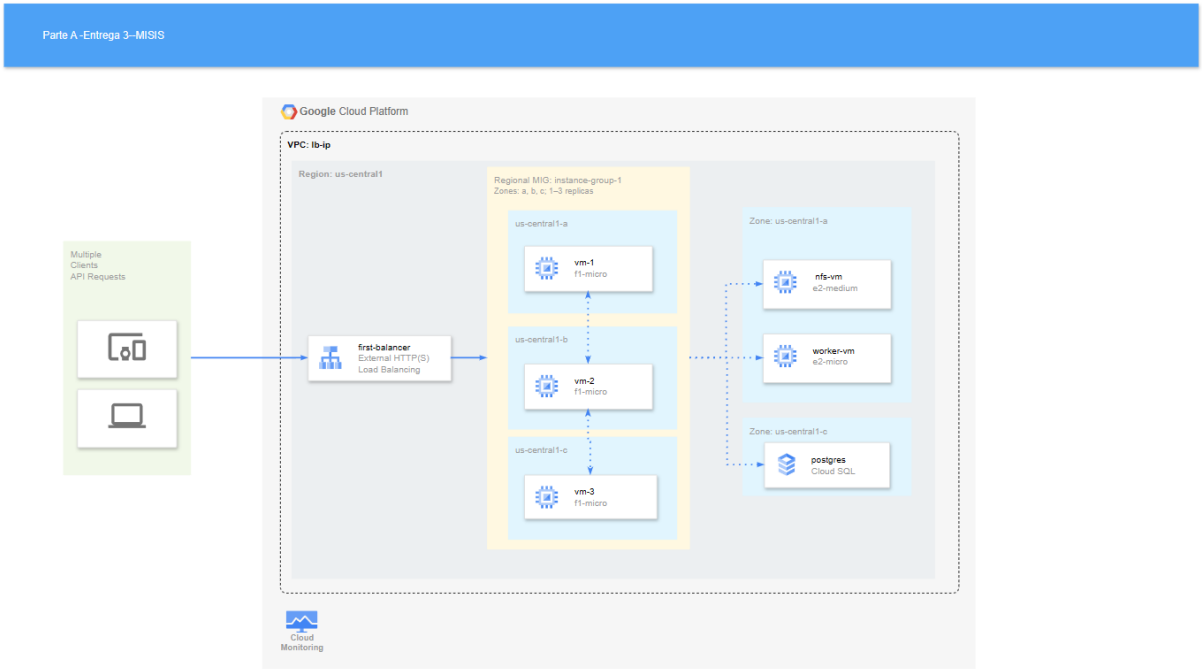
Desarrollo de Soluciones Cloud – ISIS4426
Marcos Rodrigo España Cuaran – 202124714
Juan Camilo Rodríguez Fonseca – 202514404
Camilo Andrés Cáceres Fontecha – 201812935

Arquitectura de Aplicación

Parte A

En la nueva arquitectura todo el tráfico HTTPs de los clientes entra por el Load Balancer externo. Este balanceador distribuye las solicitudes al Managed Instance Group, que levanta réplicas f1-micro en us-central1-a, us-central1-b y us-central1-c, escalando entre 1 y 3 instancias según la CPU y comprobando salud vía /healthz.



Para el almacenamiento de documentos se usa la nds-vm que expone un sistema de ficheros compartido por las instancias web y el worker. El worker consume nuevos trabajos, procesa los documentos desde NDS y actualiza su estado.

La gestión de usuarios y metadatos la realiza Cloud SQL, donde se almacenan credenciales, registros de actividad y resultados del procesamiento. Por otra parte, Cloud Monitoring se encarga de recopilar métricas de CPI, tamaño del MIG y estado de los health checks.

Desarrollo de Soluciones Cloud – ISIS4426
Marcos Rodrigo España Cuaran – 202124714
Juan Camilo Rodríguez Fonseca – 202514404
Camilo Andrés Cáceres Fontecha – 201812935

Componentes

- **External HTTP(S) Load Balancer**

  Este servicio global recibe todo el tráfico HTTPs de los clientes y lo distribuye entre las réplicas del Web Tier.

- **Web Tier – Managed Instance Group**
  - Front: contenedor que sirve la interfaz HTML/CSS/JS a navegadores y móviles.
  - Back: contenedor Flask que expone los endpoints (autenticación, subida, descarga, estado).
  - Gemini API: llamada desde el backend para análisis de texto y generación de resúmenes.
  - Health checks en el puerto 5000 garantizan que sólo instancias saludables reciban tráfico.

- **Worker-VM (Compute Engine)**
  - Worker: Servicio en contenedor que consulta documentos pendientes en la base de datos, accede a los archivos compartidos y realiza el procesamiento de texto. Al finalizar, actualiza el estado del documento.

  Esta separación permite liberar carga de la Web-Server-VM y facilita el escalamiento de procesos de análisis en paralelo.

- **NFS-VM (Compute Engine)**
  - Provee un sistema de archivos compartido utilizando NFS, exportando el directorio /mnt/nfs/files.

  Tanto el backend como el worker montan este volumen, garantizando un almacenamiento común y persistente para los documentos subidos.

- **Cloud SQL**
  - Servicio administrado de base de datos SQL que almacena la información estructurada: usuarios, metadatos de los documentos, estados de procesamiento.

Es accedido tanto por el backend (para registrar y consultar solicitudes) como por el worker (para cambiar estados de documentos).

Análisis de Capacidad

1. Escenario de Prueba de Capa Web

El escenario 1 de capa web debe medir la eficiencia de transacciones críticas, es decir, registro, inicio de sesión y carga de documento. Para este escenario se simularán 200 usuarios.

Simulación:

- 200 usuarios simultáneos realizando registro, login y carga de documentos pequeños (~1MB) durante un periodo de ramp-up de 1s.

Métricas Clave y Criterios de aceptación:

- Tiempo de respuesta (registro/login): < 500ms.
- Tiempo de carga de documentos: < 1000ms.
- Throughput (transacciones por minuto): ≥ 20. (0,33 por segundo)
- Errores HTTP (%): < 2%.

Resultados

Los resultados obtenidos para la prueba del escenario 1 fueron los siguientes,

Summary Report

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Registro Usuario	200	16487	2224	42647	10463	11,0%	4,63/sec	0,99	1,61	219,4
Inicio de Sesión	200	17058	254	33008	8174	16,0%	4,69/sec	2,17	1,36	473,8
Carga Documento	200	7220	346	30473	7553	16,5%	5,39/sec	1,38	253,35	261,4
TOTAL	600	13588	254	42647	9905	14,5%	13,26/sec	4,12	210,61	318,2

Marcos Rodrigo España Cuaran – 202124714
Juan Camilo Rodríguez Fonseca – 202514404
Camilo Andrés Cáceres Fontecha – 201812935

Agregate Report

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Received KB/sec	Sent KB/sec
Registro Usuario	200	16487	13163	31641	32668	34404	2224	42647	0,99	1,61
Inicio de Sesión	200	17058	15465	29844	30565	32776	254	33008	2,17	1,36
Carga Documento	200	7220	4861	17364	24561	27803	346	30473	1,38	253,35
TOTAL	600	13588	12050	29872	31220	33991	254	42647	4,12	210,61

Análisis

Tiempos de Respuesta: Bajo una condicion de 200 usuarios en sólo 1s de ramp-up, los endpoints de registro e inicio de sesión presentan latencias medias superiores a 16s, muy por encima del umbral deseado de 500 ms. Este retraso indica que las nuevas instancias tardan demasiado en arrancar y atender las conexiones, y que la base de datos puede estar actuando como cuello de botella. Además, los picos máximos de 42s sugieren “cold starts” frecuentes en los contenedores

Errores HTTP: Se presentan errores entre 11 % y 16,5 %, por encima del umbral de 2 %. Lo cual sugiere una saturación de las instancias. La mayoría de estos errores probablemente son fallos por timeouts o rechazos de nuevas conexiones cuando el grupo de instancias no está suficientemente dimensionado al inicio de la prueba.

Throughput: Aun así, el sistema logra procesar unos 780 req/min lo que supera el umbral mínimo, es decir que hay capacidad de procesar más peticiones. Esto demuestra que, una vez estabilizadas las instancias, el servicio web tiene capacidad de cómputo suficiente para un mayor volumen de peticiones.

2. Escenario de Capa de Procesamiento por Lotes

Para el escenario de capa de procesamiento por lotes se quiere medir la eficiencia de transacciones que ocurren en segundo plano, como el procesamiento de documentos en términos de uso del modelo LLM como generación de resúmenes y respuestas.

Procesamiento en segundo plano

Se simulará 50 usuarios que suben documentos grandes (PDF o DOCX) al mismo tiempo y solicitan la generación de resumen y respuestas.

Métricas: Se medirá el rendimiento del modelo LLM en tiempos de procesamiento y utilización de recursos en el servidor (CPU y memoria).

Escenario 2: Capa de Procesamiento por Lotes - Generación de Resúmenes y Respuestas

Descripción**:** Medición del rendimiento en el procesamiento de documentos mediante el servicio de API de Gemini, evaluando tiempos de respuesta y consumo de recursos.

Simulación:

- 50 usuarios simultáneos subiendo documentos grandes (5MB-10MB) y solicitando resúmenes y respuestas del modelo LLM.

Métricas Clave:

- Tiempo de respuesta de IA: < 2000ms.
- Throughput (procesos por minuto): ≥ 10.
- Errores en generación de resumen (%): < 2%

Resultados

Los resultados obtenidos para la prueba del escenario 2 fueron los siguientes:

Summary Report

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Registro Usuario	200	4661	2151	11613	2221	0,0%	4,13/sec	0,82	1,3	219,4
Inicio de Sesión	200	4671	388	7713	1562	0,0%	4,84/sec	2,44	1,39	473,8
Carga Documento	200	106	104	112	2	11,5%	7,51/sec	19,18	253,35	261,4

Agregate Report

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Received KB/sec	Sent KB/sec
Registro Usuario	200	17272	14626	31649	32578	34817	2587	35418	0,82	1,3
Inicio de Sesión	200	18274	16324	30182	30234	31324	228	32152	2,44	1,39
Carga Documento	200	6256	4051	15818	21045	25919	292	27654	19,18	226,64

Desarrollo de Soluciones Cloud – ISIS4426
Marcos Rodrigo España Cuaran – 202124714
Juan Camilo Rodríguez Fonseca – 202514404
Camilo Andrés Cáceres Fontecha – 201812935

Análisis

Tiempo de Respuesta de IA: La media de procesamiento de cada documento es de 6256ms más de tres veces el objetivo de 2000ms. Estos picos pueden deberse a cold starts del contenedor del worker o al invocar Gemini API.

Throughput: El throughput medido en el summary report es de 7,51 procesos/s, lo que equivale a unos 450 procesos/min, muy por encima del mínimo de 10/min. Lo que significa que el sistema tiene capacidad para atender el volumen, es decir el cuello de botella no está en el número de workers, sino en la latencia de la llamada a la LLM.

Errores de Generación: Se observa un 11,5% de errores en la carga de documentos, muy por encima del umbral del 2%. Estos errores pueden ser timeouts o fallos de la Gemini API cuando el worker está colapsado o cuando la VM receptor alcanza límites de concurrencia.

Conclusiones

- La arquitectura cumple los requisitos de escalabilidad y tolerancia a fallos en la capa web, el Managed Instance Group y el balanceador de carga garantizan una respuesta adecuada bajo picos de carga.
- El patrón de separación entre Web Tier, NFS y Worker permite soportar el procesamiento en segundo plano, evitando la saturación de la capa de frontend.
- Cloud SQL proporciona facilidad para la persistencia de usuarios sin preocuparse por la replicación.

Consideraciones

- Sustituir la NFS por un bucket como se tenia planeado, aportará mayor rendimiento y simplicidad en el mantenimiento.
- La conversión del worker a un Managed Instance Group permitirá escalar el procesamiento de documentos de manera desacoplada.
- Mantener un mínimo número de réplicas permitirá reducir latencias iniciales en momentos de picos de usuarios.

Video: https://youtu.be/yIA_pL5DmYQ