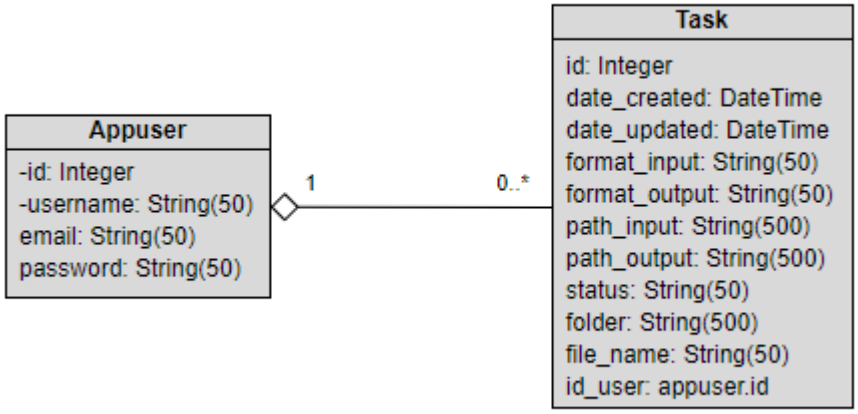
Entrega 4 - Arquitectura, conclusiones y consideraciones

A. Arquitectura

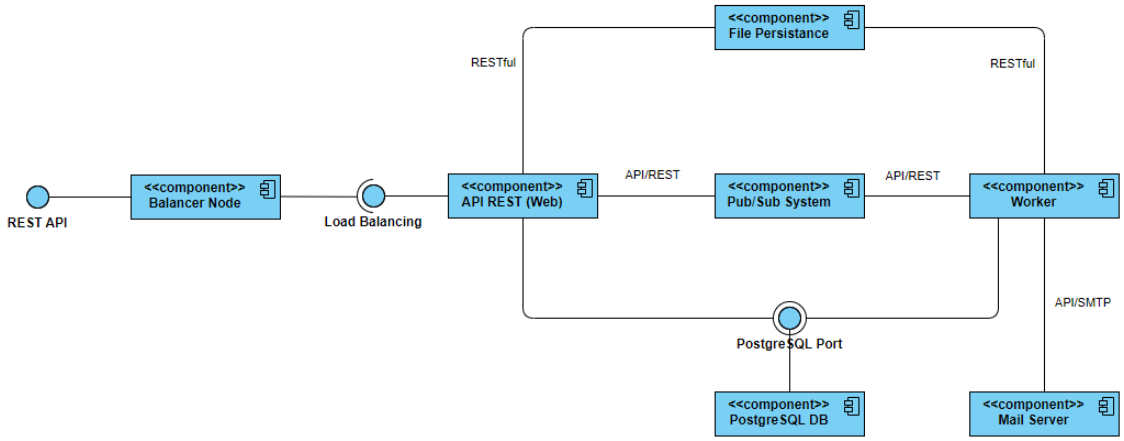
Se diagrama la arquitectura desde Vista de Diseño (Diagramas de entidades, y de componentes), para explicar la estructura de tablas SQL definidas para gestionar seguimiento a las tareas a encolar en el worker, y una Vista de Despliegue (Diagrama de despliegue) para que se identifiquen las máquinas virtuales, sus imágenes de software montadas, y demás componentes tipo SaaS.

a. Diagrama de entidades (PostgreSQL)



Esto se refleja en las tablas Appuser y Task creadas en el entorno Cloud SQL, sabor PostgreSQL.

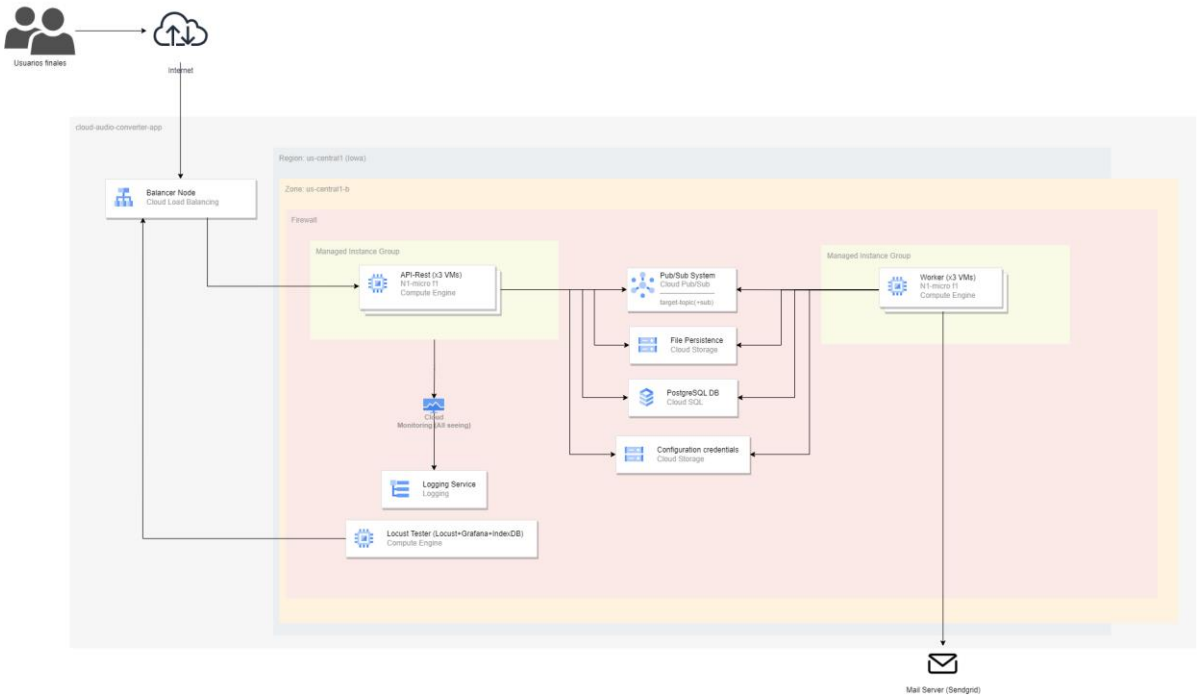
b. Diagrama de componentes



A grandes rasgos, el diagrama de componentes expone brevemente la solución migrada y mejorada por el equipo de desarrollo. Se tiene el Balancer node que es una abstracción del servidor de balanceo de carga, quien a su vez se conecta al API REST asociado a la capa Web de la solución (pues está será balanceada y autoescalada), el componente Worker que procesa los trabajos de conversión, quien ya no emplea una instancia local de Redis sino que se suscribe a un tópico de una instancia de Google Cloud Pub/Sub (modo PULL). Como componentes compartidos se tiene la base de datos en PostgreSQL DB, el servidor de envió de correos como Mail Server, y

un Cloud Storage que se encarga de guardar los archivos recibidos por Web a convertir, y los archivos convertidos por el Converter Worker (conforme consume mensajes de petición de conversión en el tópico Pub/Sub).

c. Diagrama de Despliegue



Enlace: [Diagrama de despliegue Cloud - Converter + Pub-Sub-Page-1.drawio.png]

El diagrama de despliegue, mediante las cajas de color identifica las distintas fronteras:

- Rojo: Componentes protegidos por firewall
- Verde claro: Instanced groups que gestionan ciertas VMs (API Rest en este caso).
- Amarillo: Zona donde se crean los componentes
- Azul: Región que alberga zona para despliegue.
- Gris: Frontera del proyecto contenedor de los componentes de la solución.

Las flechas indican las dependencias o uso de recursos explícitos desde cada componente de Google Cloud Platform. Hay que resaltar que Cloud Storage y PostgreSQL reflejan componentes tipo SaaS que emplea la solución actual, de tal forma que están completamente desacopladas del dominio de las VMs levantadas en la solución, pero que de todas formas sólo existen dentro del contexto del proyecto cloud cloud-audio-converter-app. En particular, se abstraen los nombres del diagrama de componentes sobre este diagrama para homologar los componentes identificados inicialmente en el diseño abstraído de la solución, es decir, sin indicar proveedores de nube puntuales:

- Balancer node emplea Cloud Load Balancing exponiendo un endpoint público.
- API REST (Web) se replica en un Compute Engine, hasta 3 VMs que existen simultáneamente, y que se escalan o desescalan según la demanda; en este caso, se escala cuando se supera el 90% de capacidad de CPU para todas las máquinas en el managed instance group.
- Worker se replica en un Compute Engine, hasta 3 VMs que existen simultáneamente, y que escalan o desescalan según la cantidad de tareas de conversión en cola en el componente

Cloud Pub/Sub; en este caso, se escala cuando se supera el 80% de capacidad de CPU para todas las máquinas en el managed instance group.

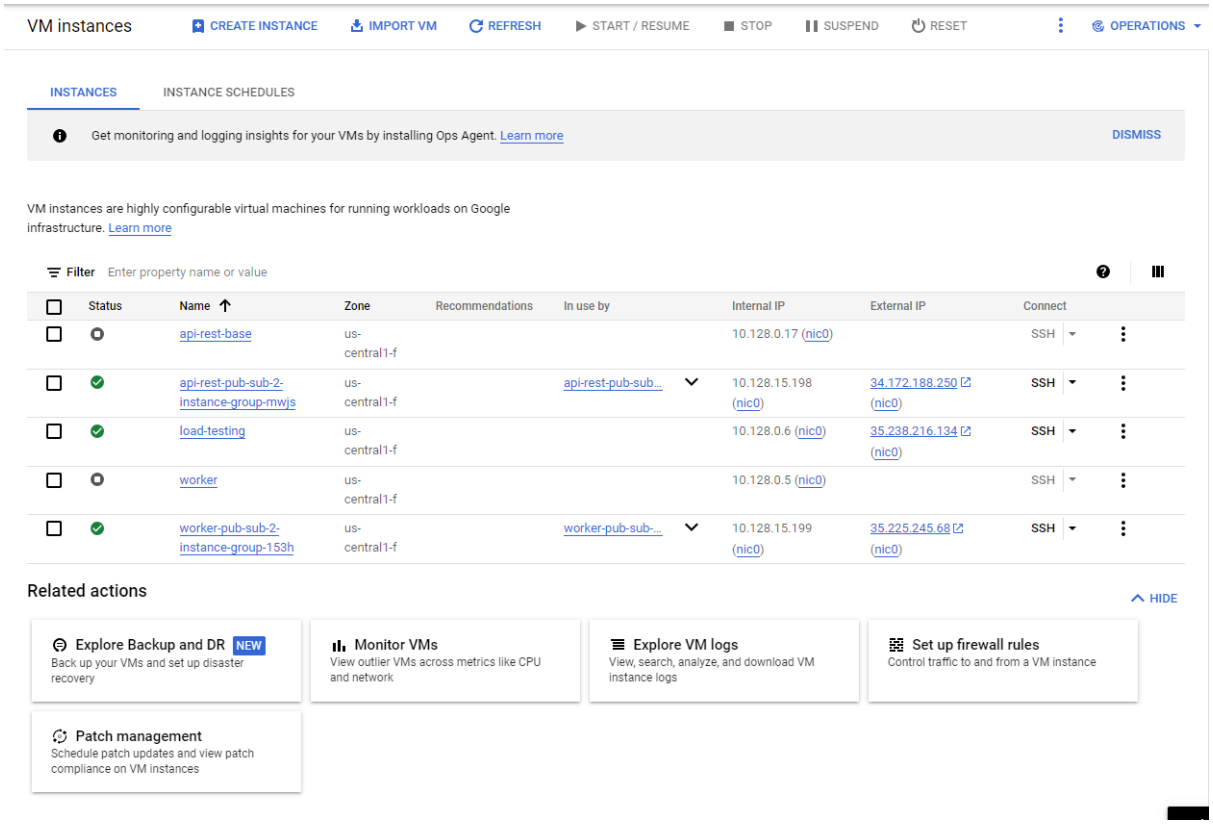
- Los Cloud Storage usa modo Standard debido a que se accede con frecuencia a los archivos de audio a procesar, y procesados. Adicionalmente el componente Configuration Credentials se emplea para que cuando los MIGs creen nuevas VMs para controlar la demanda tengan los componentes requeridos para que la imagen a desplegar en cada nodo API o Worker se levante.
- PostgreSQL DB se aprovisiona con una instancia SaaS de Cloud SQL “sabor” PostgreSQL.
- El Mail Server lo provee Sendgrid y existe afuera de la solución. No se aprovisiona desde GCP directamente.

Como nota clave, para esta versión se abandona el empleo de Celery apoyado en instancia Redis, en cambio se conecta a la cola de mensajería ofrecida por una instancia de Google Cloud Pub/Sub. Esto permite autoescalar al Worker, el cual ahora se centra en convertir los archivos de audio, actualizar los estados asociados en la base de datos SQL que usa el API REST. Ya no tiene obligaciones de gestionar el sistema de colas de conversión, sino meramente consumir los mensajes de petición de conversión mediante suscripción: Siendo así, se puede escalar fácilmente con un MIG exclusivamente centrado en escalar según consumo de CPU cada nodo Worker.

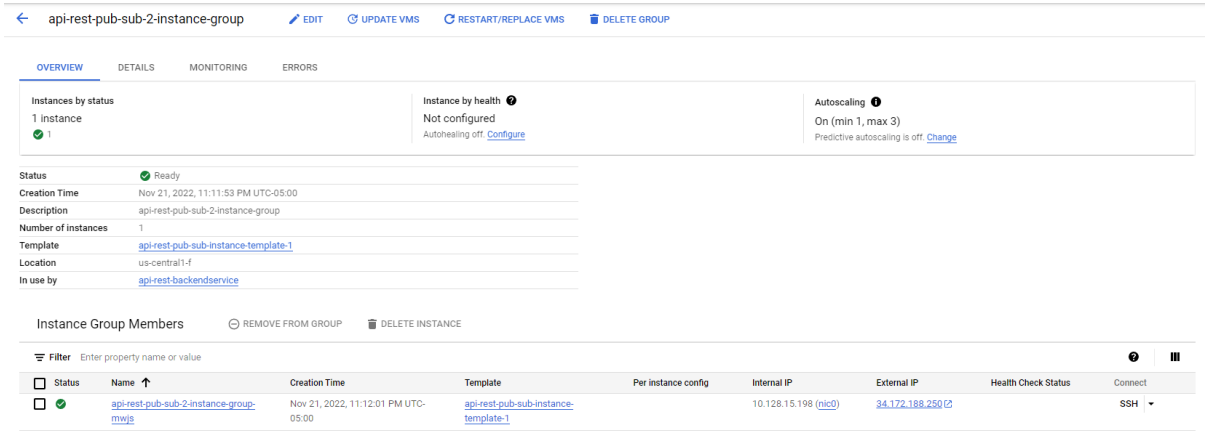
Adicionalmente se levanta un Compute Engine con la aplicación Locust empleada en las pruebas de estrés, denominado Locust Tester.

Los servicios de Monitoring y Logging vienen por defecto en el proyecto de Cloud, simplemente se configuran las VMs para que envien la data necesaria por estos servicios para hacer su trabajo.

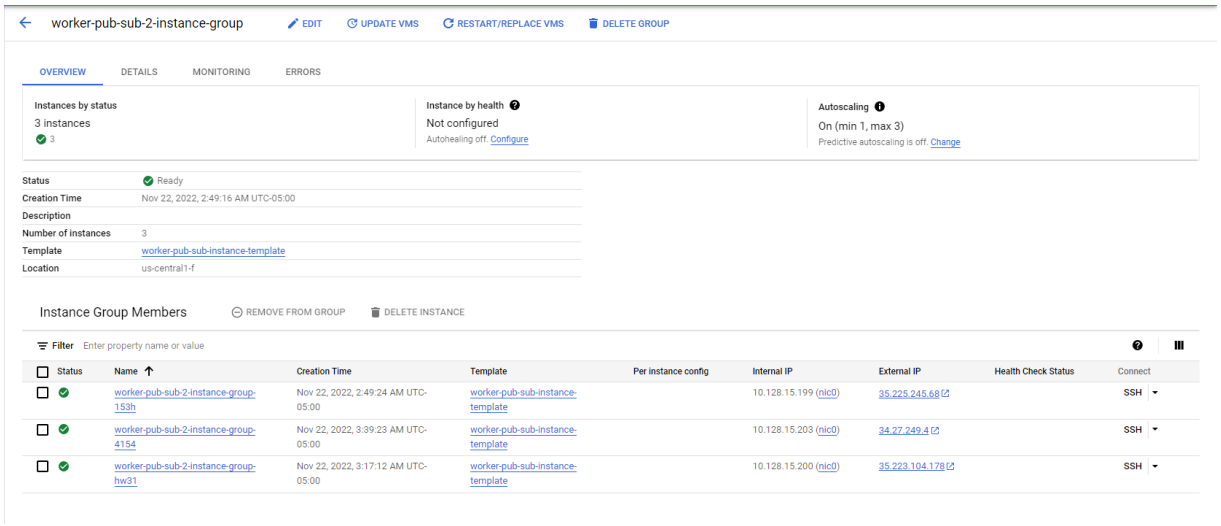
Para todos los Compute Engine (API REST, Worker, Locust Tester), se emplean VMs de tipo Serie N1, F1 Micro (con 1 vCPU, 614 MiB en RAM y 10 GiB).



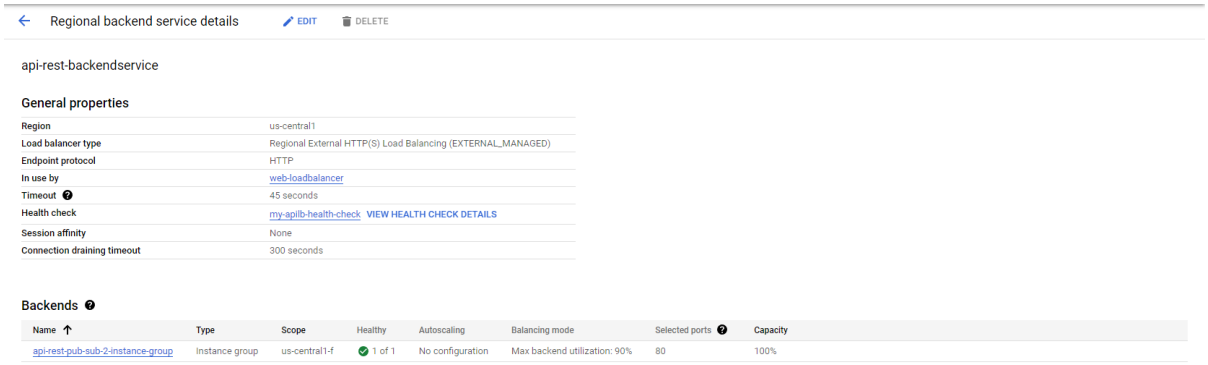
Acá, la configuración de Managed Instance Group para API REST:



Acá, la configuración de Managed Instance Group para Workers (acabadas pruebas de estrés)



Backend Service conectado al Load balancer:



Acá, la configuración de Load Balancer

Load balancing + CREATE LOAD BALANCER REFRESH DELETE HELP ASSISTANT

Get real-time analytics with Network Intelligence Center

Use Network Intelligence Center for comprehensive monitoring and troubleshooting. Learn more

- Visualize your network resources
- Diagnose and prevent connectivity issues
- View packet loss and latency metrics
- Keep your firewall rules strict and efficient

GO TO NETWORK INTELLIGENCE CENTER REMIND ME LATER

LOAD BALANCERS BACKENDS FRONTENDS

Faster web performance and improved web protection with Cloud CDN and Cloud Armor. Learn more DISMISS

Filter Enter property name or value

Name	Load balancer type	Protocols	Region	Backends
web-loadbalancer	HTTP(S)	HTTP	us-central1	1 regional backend service (1 instance group, 0 network endpoint groups)

Load balancer details EDIT DELETE HELP ASSISTANT

web-loadbalancer

Region
us-central1

Frontend

Protocol	IP:Port	Network Tier	Certificate	SSL Policy	DNS name
HTTP	35.209.134.18:80	Standard	-		

Host and path rules

Hosts	Paths	Backend
All unmatched (default)	All unmatched (default)	api-rest-backendservice

Backend

Backend services

1. api-rest-backendservice

Endpoint protocol	Named port	Timeout	Health check	Logging
HTTP	http	45 seconds	my-apilb-health-check	Disabled

ADVANCED CONFIGURATIONS

Name	Type	Scope	Healthy	Autoscaling	Balancing mode	Selected ports	Capacity
api-rest-pub-sub-2-instance-group	Instance group	us-central1-f	1 of 1	On: Target CPU utilization 90%	Max backend utilization: 90%	80	100%

Buckets configurados:

Buckets CREATE REFRESH HELP ASSISTANT LEARN

Preview The New Cloud Storage Monitoring Dashboard

Check out the new Cloud Storage monitoring dashboard and Bucket Observability pages! Powered by Cloud Operations, you can customize these dashboards for each project.

TRY NOW

View security recommendations

Improve security by applying security recommendations to your buckets. The security insights column in the table describes which buckets have excess permissions.

VIEW IN TABLE LEARN MORE

Filter Filter buckets

Name	Created	Location type	Location	Default storage class	Last modified	Public access	Access control	Protection	Lif
credencials-app-music	Nov 13, 2022, 6:09:45 PM	Region	us-central1	Standard	Nov 13, 2022, 6:09:45 PM	Not public	Uniform	None	No
music-converter-prueba-1	Nov 11, 2022, 11:06:15 PM	Region	us-central1	Standard	Nov 11, 2022, 11:06:15 PM	Not public	Uniform	None	No

Bucket details REFRESH HELP ASSISTANT LEARN

credencials-app-music

Location	Storage class	Public access	Protection
us-central1 (Iowa)	Standard	Not public	None

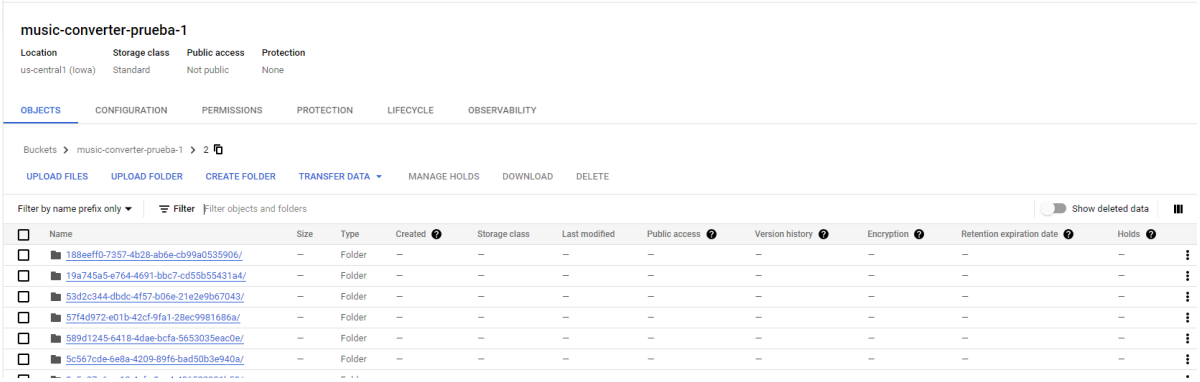
OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE OBSERVABILITY

Buckets > credencials-app-music

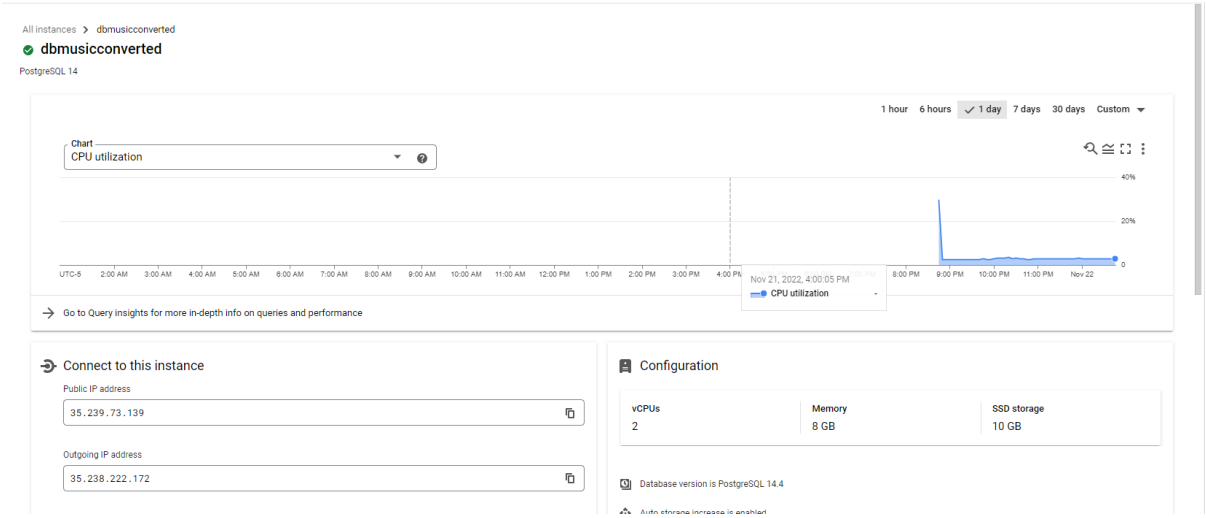
UPLOAD FILES UPLOAD FOLDER CREATE FOLDER TRANSFER DATA MANAGE HOLDS DOWNLOAD DELETE

Filter by name prefix only Filter Filter objects and folders Show deleted data

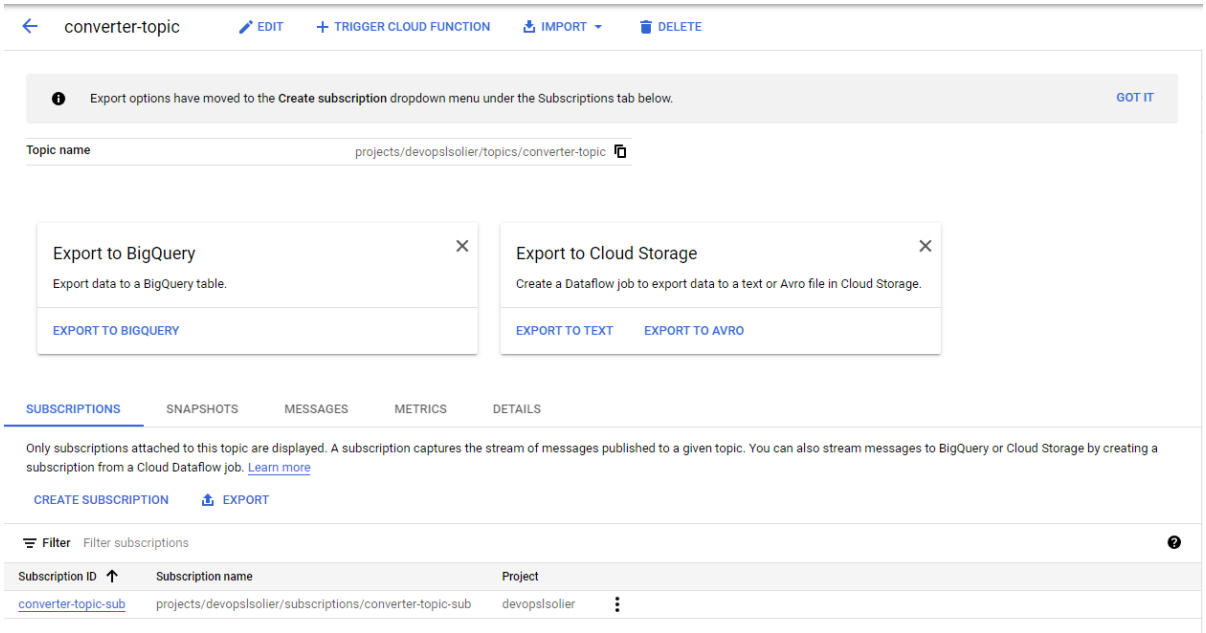
Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption	Retention expiration date
.env.cloud	584 B	application/octet-stream	Nov 21, 2022, 10:27:22 PM	Standard	Nov 21, 2022, 10:27:22 PM	Not public	–	Google-managed key	–
api-rest-startup.sh	343 B	text/x-sh	Nov 21, 2022, 10:36:14 PM	Standard	Nov 21, 2022, 10:36:14 PM	Not public	–	Google-managed key	–
path_to_your_.json_credential_file	2.3 KB	application/octet-stream	Nov 21, 2022, 10:28:34 PM	Standard	Nov 21, 2022, 10:28:34 PM	Not public	–	Google-managed key	–



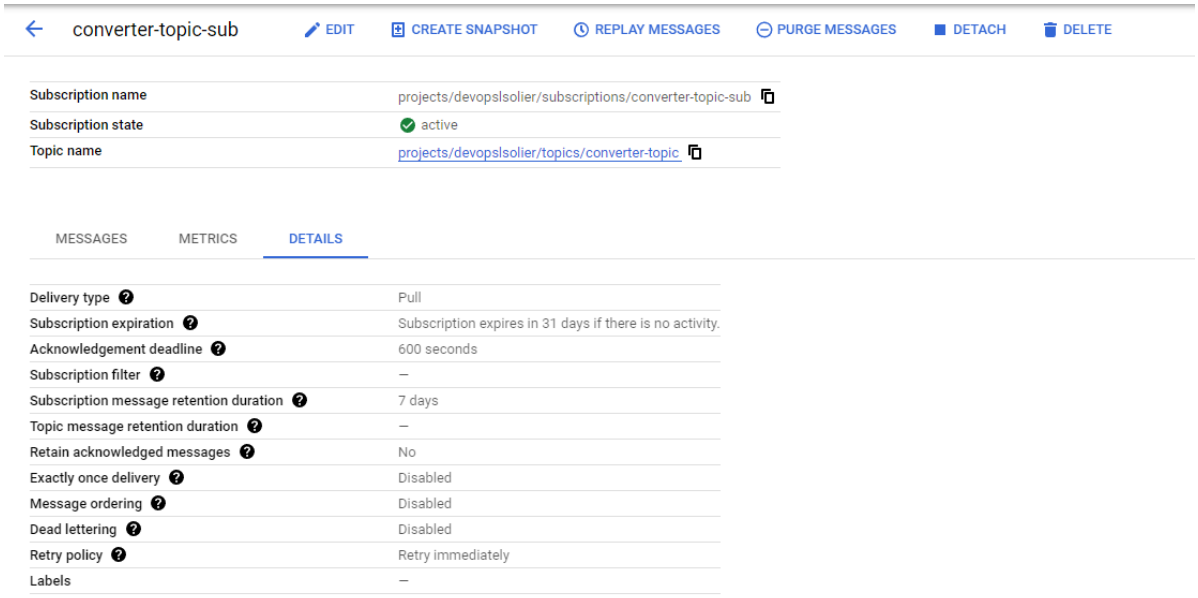
Cloud SQL configuración:



Tópico Pub/Sub:



Suscripción para consumir del Tópico Pub/Sub:



IP pública de solución migrada a nube: http://35.209.134.18/

Reemplazar en URL base de Postman Collections disponible en: https://github.com/edgar-melara-uniandes/proyecto_cloud/tree/entrega_3/documentation

B. Conclusiones de las pruebas de estrés

Con las nuevas mejoras del MIG con la nueva imagen de instancia de API REST, el API se mantiene más estable. Ahora solo escala a 2 nodos, de forma más elegante. Para el caso del worker, se ha visto que escala hasta los 3 nodos y completa todas las conversiones, algo bienvenido. Esto puede ser debido a que ffmpeg como librería de conversión ya es bastante eficiente, así que no desbordará la CPU por cualquier conversión a hacer.

Desafortunadamente, un bug descubierto mostró que, si se hace una conversión hacia el mismo tipo, no ocurre ningún procesamiento de audio, esto debido a que Celery estaba ofuscando el error que entregaba ffmpeg donde indicaba que si se hacía una conversión al mismo tipo de archivo pedía confirmación para avanzar de todas formas. En esta versión ya se solucionó y se compensó pidiendo conversiones MP3 a AAC. Esto dificulta ver como realmente se estaba comportando la solución en las entregas previas y el rendimiento en terminos de conversiones.

Siguiendo el diseño dado por el curso, y conociendo que el código fuente es lo más eficiente posible, solo queda mejorar el hardware de los nodos VM para mejorar la cantidad de personas potenciales a atender

Más detalles de las métricas de desempeño las pueden encontrar en el documento “Entrega 4 - Escenario y Pruebas de Estrés API REST y Batch”.

C. Consideraciones para el paso a entorno en nube, masivo

Al escalar tanto los API REST y los Worker nodes con Instance Groups, la solución alcanza la máxima capacidad permitida por hardware gestionado virtualmente. Mejoras de desempeño ya requiere escalar a nivel vertical, por ende se debe mejorar el tier de los instance templates que usan los Managed Instance Groups tanto de API como Worker. Si es así, hasta no se necesitaría extender el timeout para peticiones seteadas en en Load Balancer cuando haya sobrecarga de peticiones de clientes. Aunque como siempre, todo depende la demanda real de la aplicación en ambientes en vivo.

Acá entran a brillar los servicios de Monitoring y Logging, llegado el momento, como se dijo en la versión pasada, eso sí, son difíciles de analizar en rangos cortos de tiempo.

Al haber previamente descargado las dependencias que se requieren para armar una imagen API REST (y la misma mejora para las máquinas Worker), se mejoran los tiempos de arranque de nuevas instancias cuando recibe incremento de demanda el Instance Group. Definitivamente optimizar el proceso de levantamiento de nuevas instancias es vital para atender de forma rápida, y escalar de forma elegante los componentes computacionales de la aplicación.

Pub/Sub opera con política de límite de tiempo de ACK para un mensaje de máximo 600 segundos, esto implica que si no se llega a procesar/confirmar el mensaje en dicho tiempo se corre el riesgo de que entre en reprocesamiento al Pub/Sub reencolar dicho mensaje por la falta de ACK. Esto, en teoría, con el diseño propuesto puede generar retrasos desde la percepción del usuario final al ver que su archivo "sigue" convirtiéndose, aunque la realidad sea que se reencolo por dicha pólitica y entro en "reconversión". Aunque se necesita investigar más, hay desarrolladores que dicen que las librerías provistas por Google se encargan automáticamente de extender este tiempo de ACK en casos normales, por lo que para este caso no hay necesidad de preocuparse (https://medium.com/google-cloud/things-i-wish-i-knew-about-google-cloud-pub-sub-part-2-b037f1f08318)

Aún después de hacer todas la configuraciones recomendadas y solicitadas, es probable que se pueda mejorar la solución a nivel de escalabilidad. Definitivamente se debe disponer de máquinas virtuales de mayor calibre, quizás empezar con tipo N2D.