

ENTREGA 5 - DESPLIEGUE EN PAAS MIGRACIÓN DE UNA APLICACIÓN WEB A UN PLATAFORMA COMO SERVICIO EN LA NUBE PÚBLICA

Grupo 3

C. Camilo Baquero Gómez, Franklin A. Pinto Carreño, Julian Yamid Torres Torres
Desarrollo de Aplicaciones Cloud

Universidad de los Andes, Bogotá, Colombia

c.baquero@uniandes.edu.co, f.pintoc@uniandes.edu.co, jy.torres@uniandes.edu.co

Fecha de presentación: Mayo 28 de 2023

LINK APLICACIÓN WEB:

<https://crunbuid-webappcomprimemelo-imm52zicba-uc.a.run.app>

LINK componentes backend:

<https://cloud-run-backoffice-comprimemelo-imm52zicba-uc.a.run.app/api>

Github frontend: <https://github.com/camilooob/comprimemelo.com>

Github Backend: <https://github.com/JulianTorrest/Comprimemelo-Backoffice>

Video: <https://www.youtube.com/watch?v=TGW-4ybs7Eg>

1. Arquitectura de Aplicación

La aplicación web de compresión de archivos se encuentra implementada bajo el modelo desacoplamiento utilizando una instancia web que se encarga de desplegar el front en cloud run , y procesar los archivos de compresión en un Worker (backend) desplegado en un cloud run, comunicado mediante una cola de Pubsub, el front y un worker que procesa la compresión de archivos. El modelo implementa un conjunto de métodos para crear, modificar, eliminar, consultar, comprimir y descomprimir archivos y tiene acceso directo al motor de persistencia.

La vista está implementada en formato html para los formularios y páginas de presentación en capa web, y para las api rest, se utiliza el formato json, para capturar y responder las peticiones web. El controlador es el intermediario entre el modelo y la vista para interpretar las peticiones y entregar una respuesta a cada petición web realizada por un usuario.

Diagrama de arquitectura

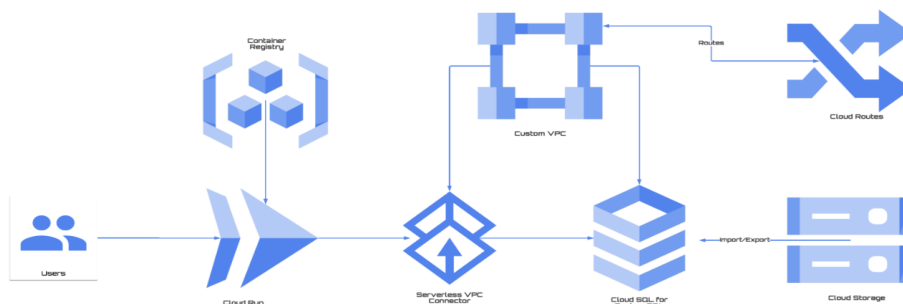


Figura 1. Diagrama de arquitectura aplicación de compresión de archivos

Login

Registrar

Login

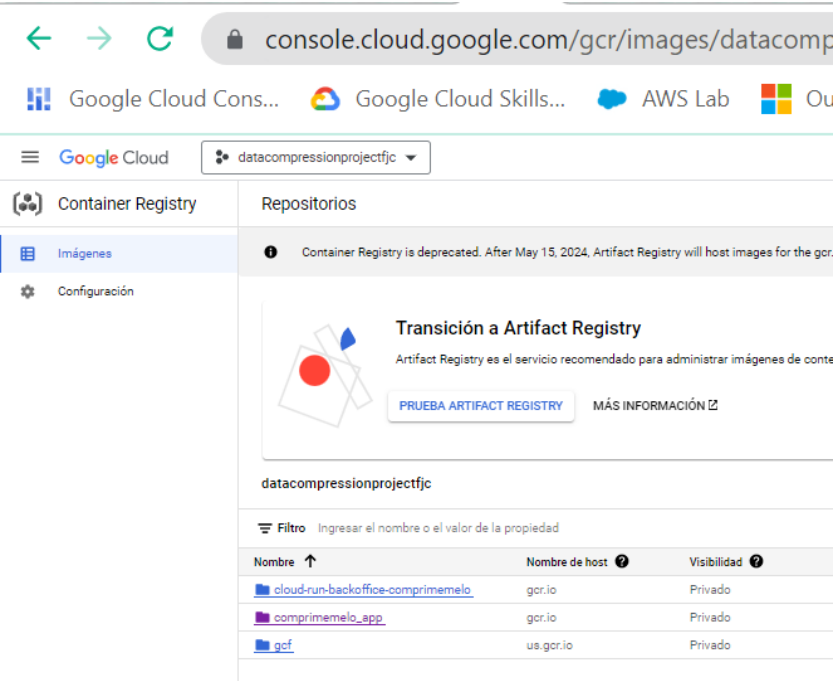
Username

Password

Login

Configurar y desplegar la aplicación frontend en Cloud Run

Se realiza la dockerización de la imagen de la aplicación web en python flask, la cual se registra en el container Registry de google cloud, con el nombre de comprimemelo_app.



Dentro del nombre de imagen se encuentran distintas versiones generadas las cuales se usarán para crear el servicio de cloud run

Al autenticarse a través de la pantalla de login, el servicio internamente debe conectarse a la base de datos en la tabla de users y comparar las credenciales ingresadas.

The screenshot shows a web browser window with the URL `crunbuid-webappcompprimemelo-imm52zicba-uc.a.run.app/auth/login`. The page has a header with navigation links: "Login" and "Regístrate". The main content area is titled "Login" and contains a form with two input fields: "Username" (containing `steffanypavi@gmail.com`) and "Password" (containing masked characters). Below the fields is a blue "Login" button.

Al validar correctamente las credenciales de acceso, la aplicación permite el ingreso a la página principal del usuario autenticado

The screenshot shows the main page of the web application. The URL is `crunbuid-webappcompprimemelo-imm52zicba-uc.a.run.app`. The header includes navigation links: "Home", "Comprimir", and a dropdown menu for the user `steffanypavi@gmail.com`. The main content area features a large graphic with the word "COMPRESIÓN" and the website URL "COMPRIMELO.COM". Below the text are several 3D blocks representing different file compression formats: ZIP, RAR, 7Z, CAB, ACE, GZ, and BZ2.

De aquí en adelante el usuario ya puede navegar sobre la aplicación web publicada en google cloud run, por ejemplo puede comprimir un archivo:

The screenshot shows the "Comprimir" (Compress) page of the web application. The URL is `crunbuid-webappcompprimemelo-imm52zicba-uc.a.run.app/eventos/comprimir`. The header includes navigation links: "Home", "Comprimir", and a dropdown menu for the user `steffanypavi@gmail.com`. The main content area has a blue button labeled "Comprimir Archivo". Below the button is a section titled "Compresión de archivos". It contains a dropdown menu labeled "Seleccionar archivo" with the text "Ninguno archivo selec." and a list of file formats: zip, z7, rar, bzip, tar, gz, tar.gz, and tar.bz2. The "tar.gz" option is currently selected. To the right of the list is a button labeled "Comprimir".

En caso que el usuario no posea credenciales de acceso, se puede registrar por el link de registro y creación de usuario.

Registrar Usuario

Nombre
Franklin

Apellido
Pinto

Username
franklinpinto20@hotmail.com

Correo
f.pintoc@uniandes.edu.co

Celular
3187678086

Nueva contraseña

Confirmación de contraseña

Registrarme

2. Configurar y desplegar la aplicación backend en Cloud Run

Cloud Build

Google Cloud datacompressionprojectjc trigg Search 15 ? J

Cloud Build Triggers + CREATE TRIGGER → CONNECT REPOSITORY MANAGE REPOSITORIES LEARN

Region
global (non-regional)

Filter Enter property name or value

Name	Description	Repository	Event	Build configuration	Build type
cloud-run-backoffice	-	JulianTorrest/Comprimemelo-Backoffice	Push to branch	cloudbu	RUN
mgpgab-gcrun-comprimemelo-app-us-central1-camilooob-comprimde	Build and d...	camilooob/comprimemelo.com	Push to branch	Inline	RUN

VPC Serverless

Databases

HELP ASSISTANT

All instances > mysqlcomprime

mysqlcomprime

MySQL 8.0

CREATE DATABASE

Name	Collation	Character set	Type
comprimelodb	utf8mb4_0900_ai_ci	utf8mb4	User
information_schema	utf8_general_ci	utf8	System
mysql	utf8_general_ci	utf8	System
performance_schema	utf8mb4_0900_ai_ci	utf8mb4	System
sys	utf8mb4_0900_ai_ci	utf8mb4	System

Container Registry

Container Registry

Images

Settings

← Images

DELETE

Container Registry is deprecated. After May 15, 2024, Artifact Registry will host images for the gcr.io domain in projects without previous Container Registry usage. [Learn more](#)

cloud-run-backoffice-comprimemelo

gcr.io > datacompressionprojectjc > cloud-run-backoffice-comprimemelo

Filter Enter property name or value

Name	Tags	Virtual Size	Created	Uploaded	Vulnerabilities
cs757893a66f	latest	449.8 MB	4 days ago	4 days ago	Never scanned
c4152a865df9		449.8 MB	4 days ago	4 days ago	Never scanned

Cloud Run

cloud-run-backoffice-comprimemelo-00002-m7m

Deployed by juliantorres@usantotomas.edu.co using Cloud Console

CONTAINERSVOLUMESNETWORKINGSECURITYYAML

General

CPU allocation	CPU is only allocated during request processing
Startup CPU boost	Disabled
Concurrency	80
Request timeout	300 seconds
Execution environment	First generation (Default)

Autoscaling

Max instances	100
---------------	-----

Image URL

gcr.io/datacompressionprojectjc/cloud-run-backoffice...

Port

4000

Build

(no build information available)

Source

(no source information available)

Command and args

(container endpoint)

CPU limit

1

Memory limit

512MiB

3. Configurar un servicio de almacenamiento SQL en GCP

Cloud SQL

Google Cloud

datacompressionprojectjc

sql

Search

15

HELP ASSISTANT

SQL

Overview

EDIT

IMPORT

EXPORT

RESTART

STOP

DELETE

CLONE

HELP ASSISTANT

PRIMARY INSTANCE

Overview

Query insights

Connections

Users

Databases

Backups

Replicas

Operations

Connect to this instance

Public IP address

34.136.110.133

Private IP address

10.59.192.3

Associated networking

projects/datacompressionprojectjc/global/networks/comprimemelo-vpc

Connection name

datacompressionprojectjc:us-central1:mysqlcomprime

Need help connecting?

Review the documentation to learn about the many ways to connect to your instance.

Learn more

To connect using gcloud,

OPEN CLOUD SHELL

To learn about connecting with a Compute Engine VM,

START TUTORIAL

Configuration

vCPUs

2

Memory

8 GB

SSD storage

20 GB

Database version is MySQL 8.0.26

Auto storage increase is enabled

Automated backups are disabled

Point-in-time recovery is disabled

Instance deletion protection is disabled

Located in us-central-1-b

Not highly available (zonal)

No database flags set

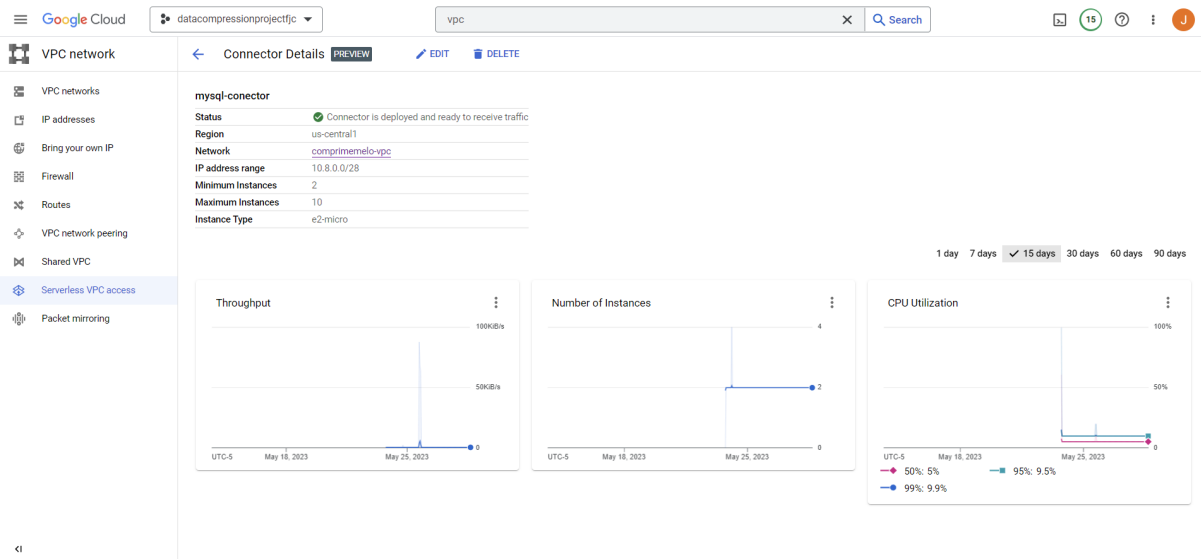
No labels set

Edit configuration

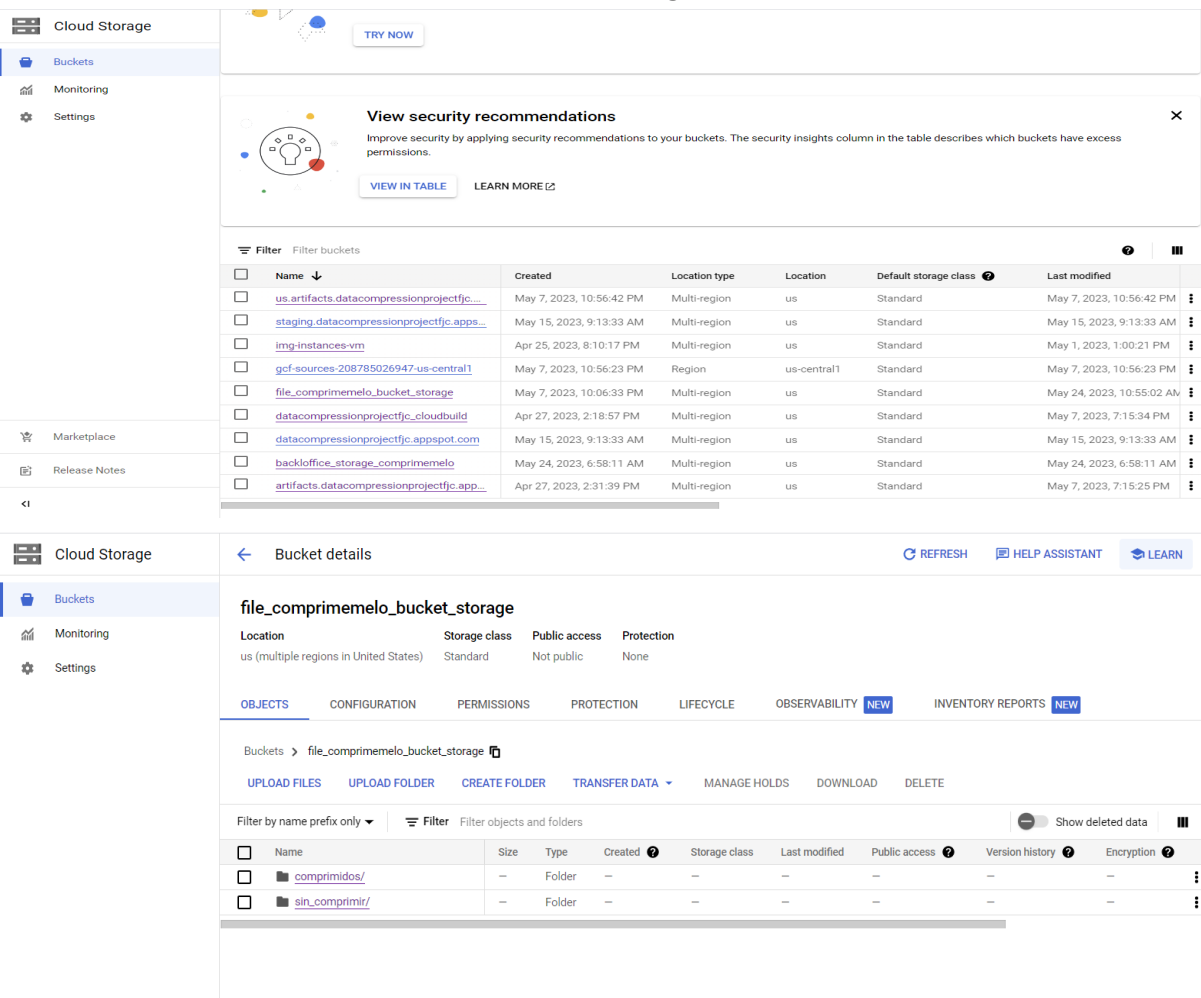
Maintenance

Maintenance window

...



Cloud Storage



4. Configurar un servicio de mensajería en GCP

Pubsub y Secret manager configurado

En el Cloud Pub/Sub utilizamos uno, con el Topic ID pubsubcomprimemelofiles, con el topic name projects/datacompressionprojectfjc/topics/pubsubcomprimemelofiles, de esa

forma nosotros diseñamos el sistema de comunicación entre los servidores web y los procesos workers permitiendo que se comunicaran entre ellos. Esto es muy importante porque permite que se añadan o se creen las diferentes solicitudes para procesar nuevos archivos y los workers pueden procesar dicha cola.

Topics [+ CREATE TOPIC](#) [DELETE](#)

LIST					METRICS	
<div>Filter Filter topics</div>						
<input type="checkbox"/>	Topic ID ↑	Encryption key	Topic name	Retention		
<input type="checkbox"/>	pubsubcomprimemelofiles	Google-managed	projects/datacompressionprojectfjc/topics/pubsubcomprimemelofiles	7 days		

A continuación podemos ver como se encuentra configurado el servicio de pub/sub

pubsubcomprimemelofiles

EDITAR + ACTIVAR CLOUD FUNCTION IMPORTAR BORRAR

Las opciones de exportación se trasladaron al menú desplegable Crear suscripción en la pestaña Suscripciones que aparece a continuación.

ENTENDIDO

Nombre del tema

projects/datacompressionprojectfjc/topics/pubsubcomprimemelofiles

Exportar a BigQuery

Exportar datos a una tabla de BigQuery.

EXPORTAR A BIGQUERY

Cómo exportar a Cloud Storage

Crea un trabajo de Dataflow para exportar datos a texto o a un archivo de Avro en Cloud Storage.

EXPORTAR A TEXTO EXPORTAR A AVRO

SUSCRIPCIONES INSTANTÁNEAS MÉTRICAS DETALLES MENSAJES

Solo se muestran las suscripciones vinculadas a este tema. Una suscripción captura la transmisión de mensajes publicados en un tema específico. También puedes transmitir mensajes a BigQuery o Cloud Storage creando una suscripción desde un trabajo de Cloud Dataflow. Más información

CREAR SUSCRIPCIÓN EXPORTAR

Filter Filtrar las suscripciones

ID de la suscripción ↑	Nombre de la suscripción	Proyecto
pubsubcomprimemelofiles-sub	projects/datacompressionprojectfjc/subscriptions/pubsubcomprimemelofiles-sub	datacompressionprojectfjc

pubsubcomprimemelofiles

PERMISOS

Edita o borra los permisos que aparecen a continuación. Selecciona "Agregar" para otorgar acceso.

Mostrar

Filter Ingresar el nombre o el valor de la propiedad

Función/Principio

Agente de seguridad

Agente de seguridad

Agente de seguridad

Agente de seguridad

Cuenta de servicio

Editor (3)

Propietario

Google Cloud

datacompressionprojectfjc

secret

Detalles del secreto

EDITAR SECRET BORRAR

Secret: "secretos_comprimemelo"

projects/208785026947/secrets/secretos_comprimemelo

DESCRIPCIÓN GENERAL VERSIONES PERMISOS REGISTROS

Versiones

+ VERSIÓN NUEVA HABILITAR INHABILITAR DESTRUIR

Filter Ingresar el nombre o el valor de la propiedad

<input type="checkbox"/>	Versión	Alias	Estado	Encriptación	Fecha de creación ↓	Acciones
<input type="checkbox"/>	2	-	Habilitada	Administrada por Google	7/5/23, 21:44	
<input type="checkbox"/>	1	-	Habilitada	Administrada por Google	7/5/23, 19:44	

No se seleccionaron versiones

5. Análisis de capacidad

Prueba escenario 1. Obtener token de autenticación

Para realizar el proceso de obtener el token de autenticación se requiere previamente haber creado una cuenta de usuario, para este caso de prueba se ha considerado un usuario existente (generado en la entrega 4)

Ejecución con un grupo de 1000 hilos concurrentes

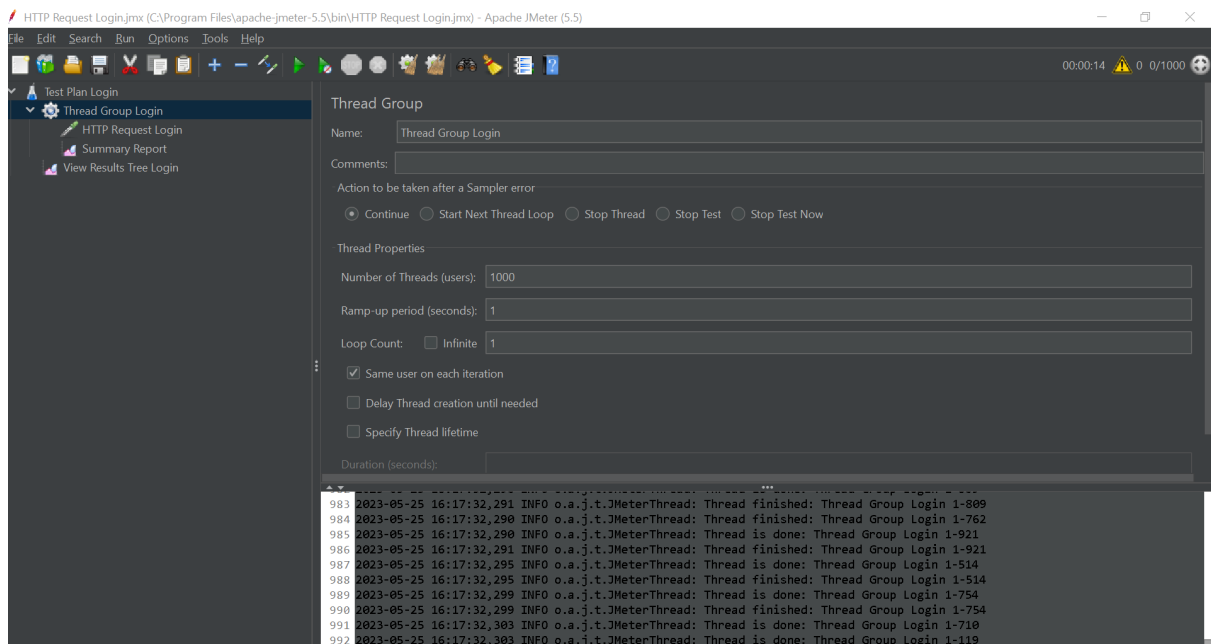
Se ejecuta el plan de pruebas desde JMeter para un grupo de 1000 hilos a la siguiente url del servicio de cloud run del backend, Método POST

<https://cloud-run-backoffice-comprimemelo-imm52zicba-uc.a.run.app/api/auth/login>

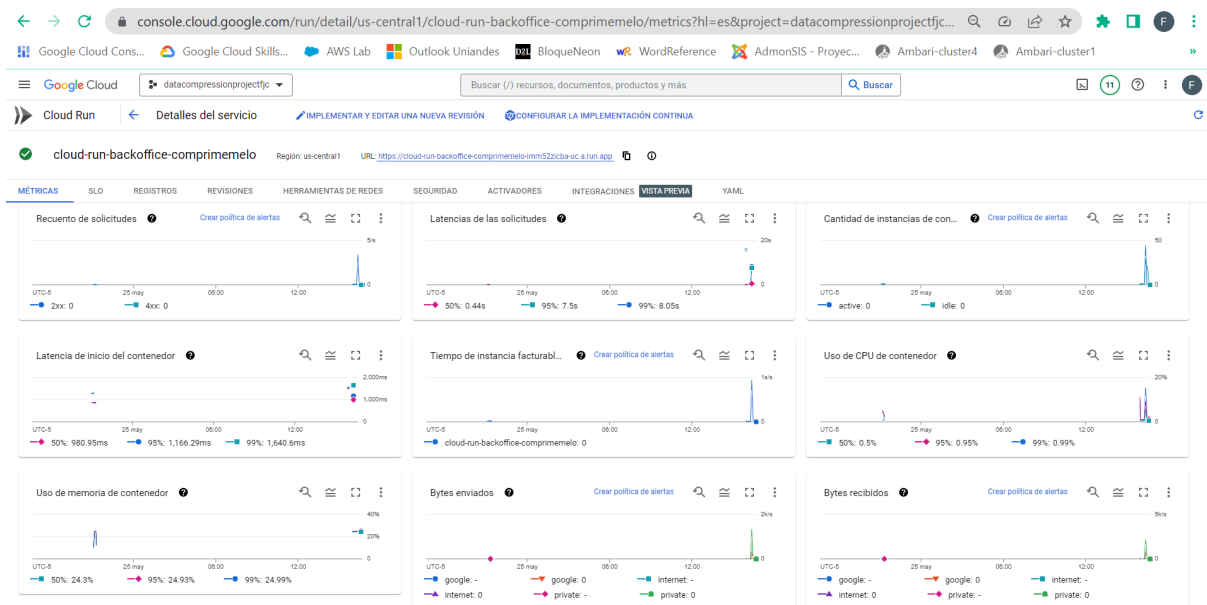
con los parámetros:

email *f.pintoc@uniandes.edu.co*

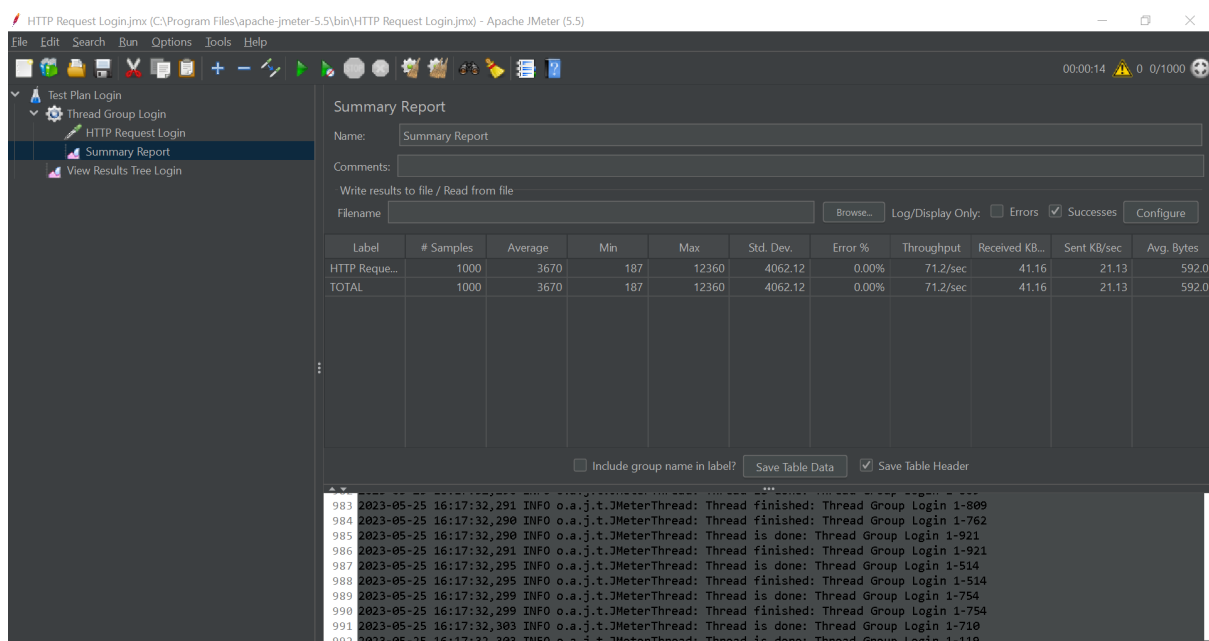
password *pwd*



Para este escenario se comienza a identificar en las métricas del servicio de cloud run del backend que se aumenta la cantidad de recursos de asignación automática, por ejemplo la cpu, cantidad de instancias del contenedor y memoria

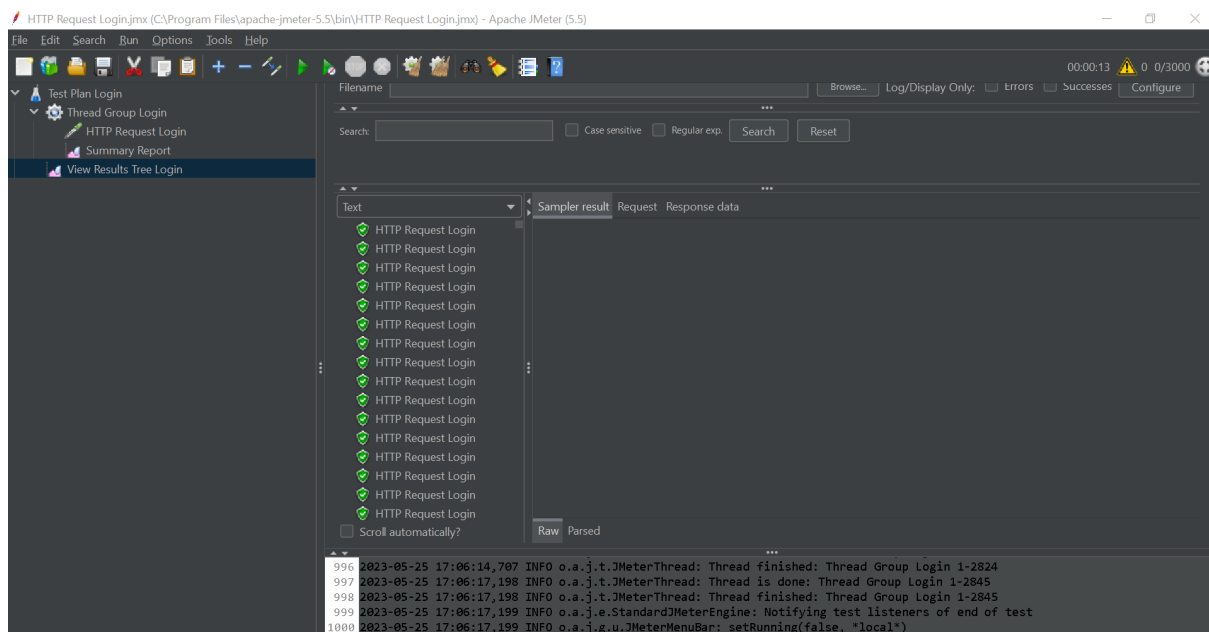


En el reporte de resultados en jmeter se visualiza que no se produjeron errores y el throughput equivale a 71.2 peticiones por segundo, en total tardó 14 segundos en procesar 1000 peticiones concurrentes.



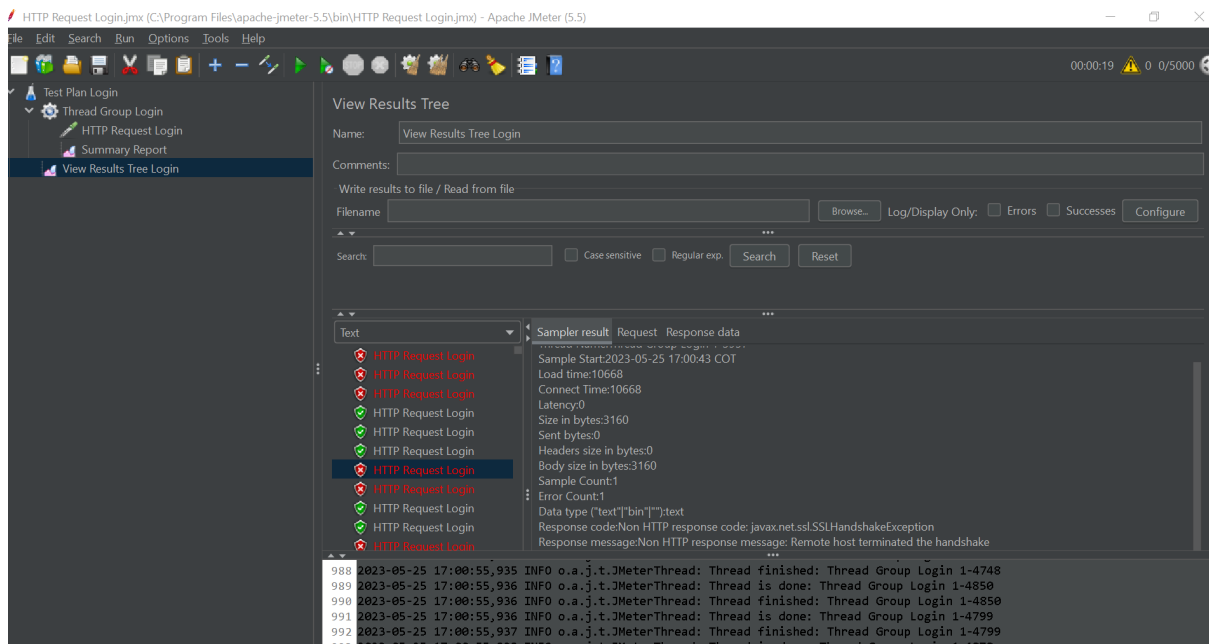
Ejecución con un grupo de 3000 hilos concurrentes

En una prueba realizada con 3000 hilos se evidencia que el servidor sigue respondiendo a todas la peticiones concurrentes y completa la operación en 13 segundos.



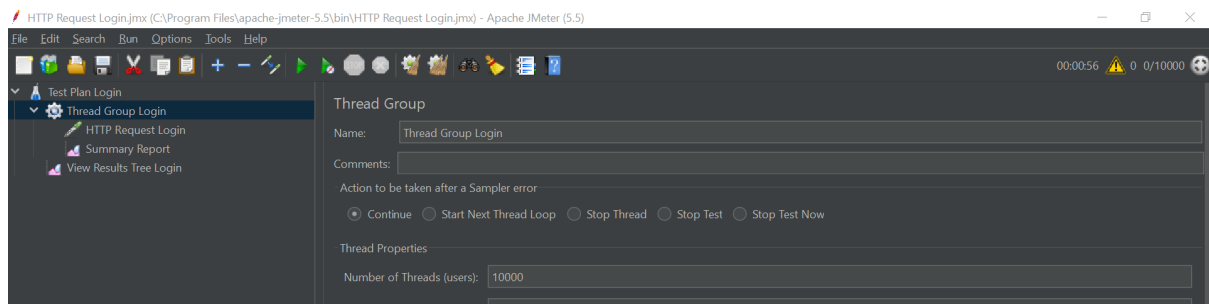
Ejecución con un grupo de 5000 hilos concurrentes

En la prueba realizada con 5000 hilos se evidencia que el servidor comienza a presentar errores en la respuesta. Aparecen errores de SSL, esto se debe a que el host remoto cerró la conexión durante la comunicación de protocolo de enlace al realizar la petición HTTPS de un servicio web de cloud run.

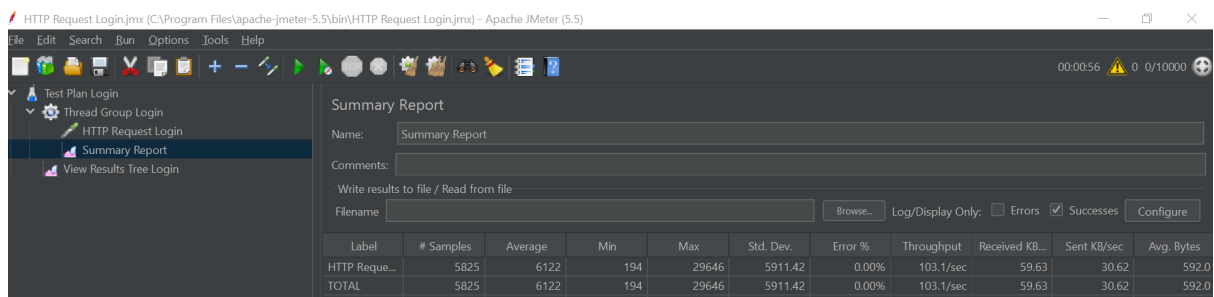


Ejecución con un grupo de 10000 hilos concurrentes

El tiempo de ejecución total de las 10.000 peticiones es de 56 segundos

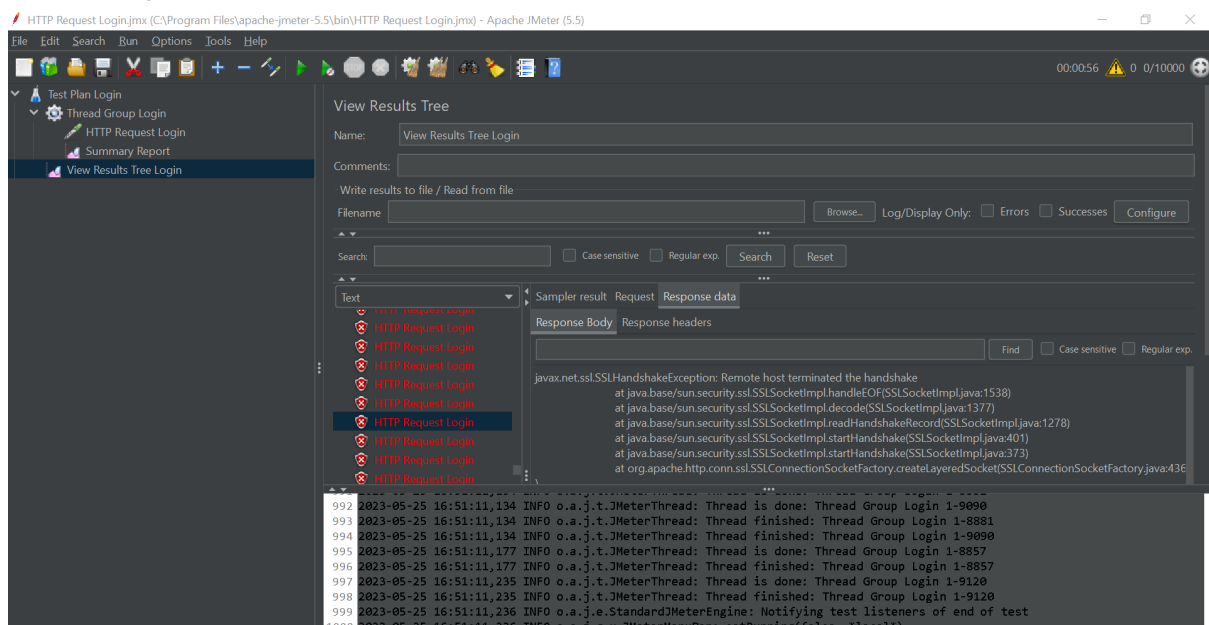


Se evidencia que aparecen errores de comunicación durante las peticiones al servicio web. El throughput es de 103.1 por segundo, pero esto ocurre porque el servidor comienza a cerrar las conexiones entrantes y de esta manera se ejecutan más rápido las peticiones aunque la respuesta sea fallida.

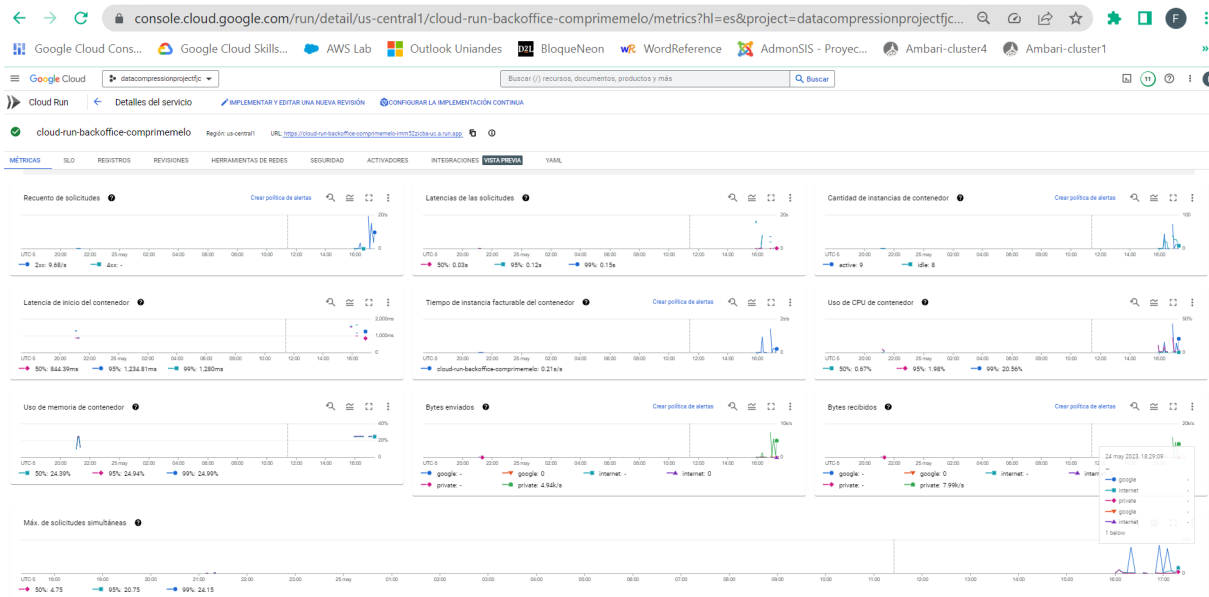


Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
HTTP Reque...	5825	6122	194	29646	5911.42	0.00%	103.1/sec	59.63	30.62	592.0
TOTAL	5825	6122	194	29646	5911.42	0.00%	103.1/sec	59.63	30.62	592.0

El detalle del error es Remote host terminated the handshake por SSL, esto ocurrió porque el host remoto cerró la conexión durante la comunicación de protocolo de enlace al realizar la petición HTTPS hacía el servicio web de cloud run.



La instanciación automática de contenedores se aumentó hasta 72 cuando se realizó las 10 mil peticiones, así mismo el uso de cpu y memoria.

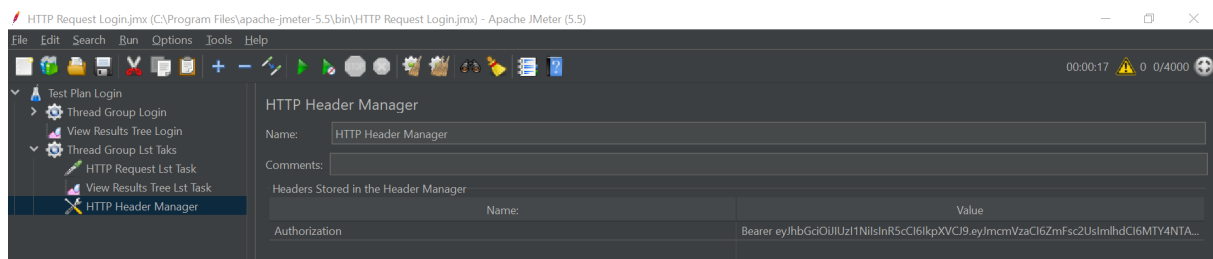


Se concluye que para el servicio de obtener token de autenticación, con la infraestructura configurada, la lógica implementada en python y la administración automática de cloud run, se pueden procesar concurrentemente 4000 peticiones sin obtener errores en la respuesta del servicio, este sería el valor que supera el SLO que se tenía propuesto, dado que se había considerado procesar 2000 peticiones concurrentes.

Prueba escenario 2. *Listar todas las tareas de conversión de un usuario*

El servicio entrega el identificador de la tarea, el nombre y la extensión del archivo original, a qué extensión desea convertir y si está disponible o no. El usuario debe proveer el token de autenticación para realizar dicha operación.

Dentro de los parámetros de consulta, se debe adicionar únicamente el token de autorización, este se agrega en el componente de tipo http header manager

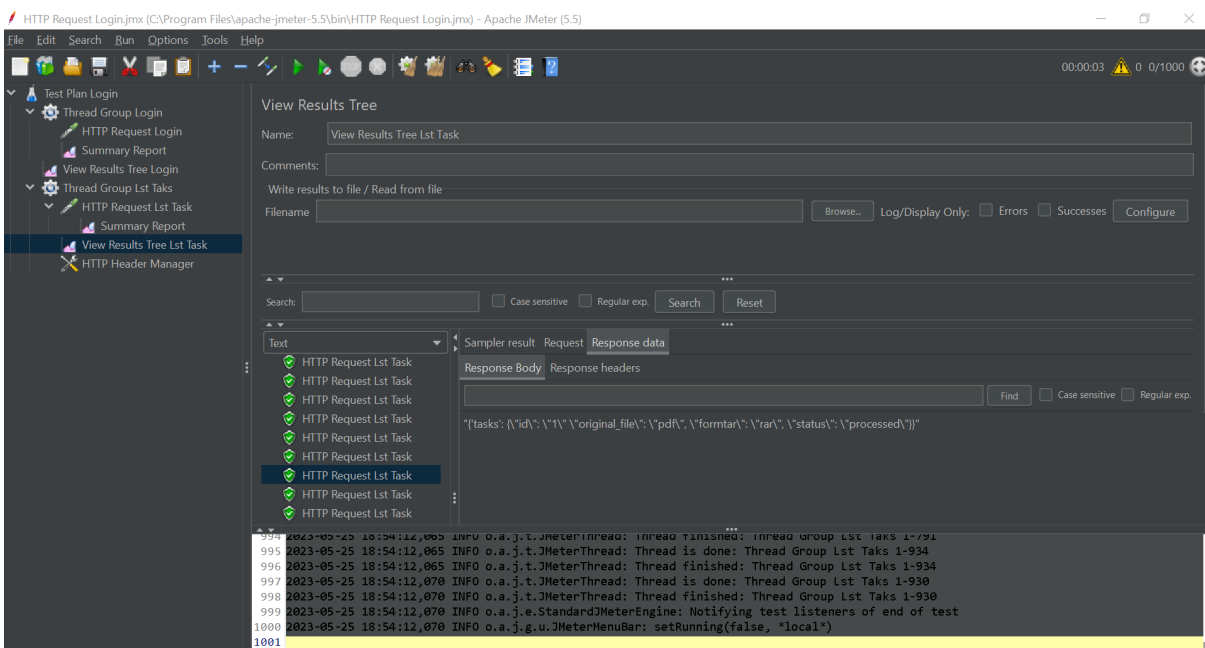


Ejecución con un grupo de 1000 hilos concurrentes

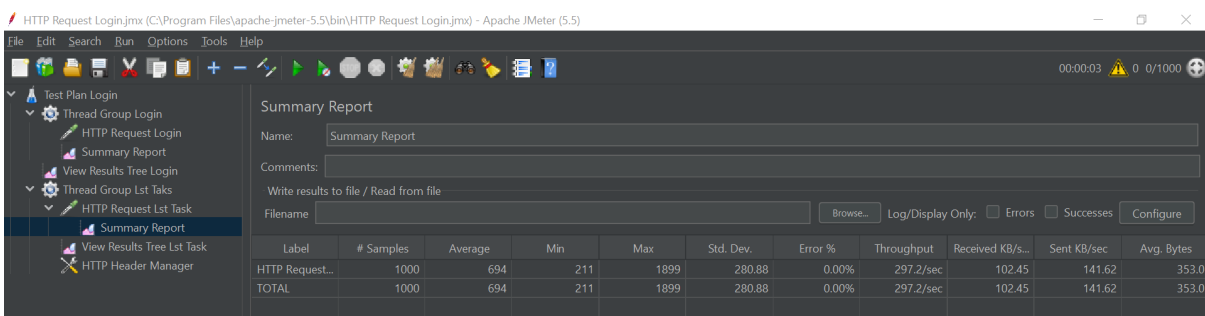
Se ejecuta el plan de pruebas desde JMETER para un grupo de 1000 hilos a la siguiente url del servicio de cloud run del backend, método GET

<https://cloud-run-backoffice-comprimemelo-imm52zicba-uc.a.run.app/api/tasks>

El procesamiento de 1000 hilos se realiza en un tiempo de 3 segundos, donde solo se encuentra una sola tarea pendiente de procesar. No se presentan errores en las respuestas



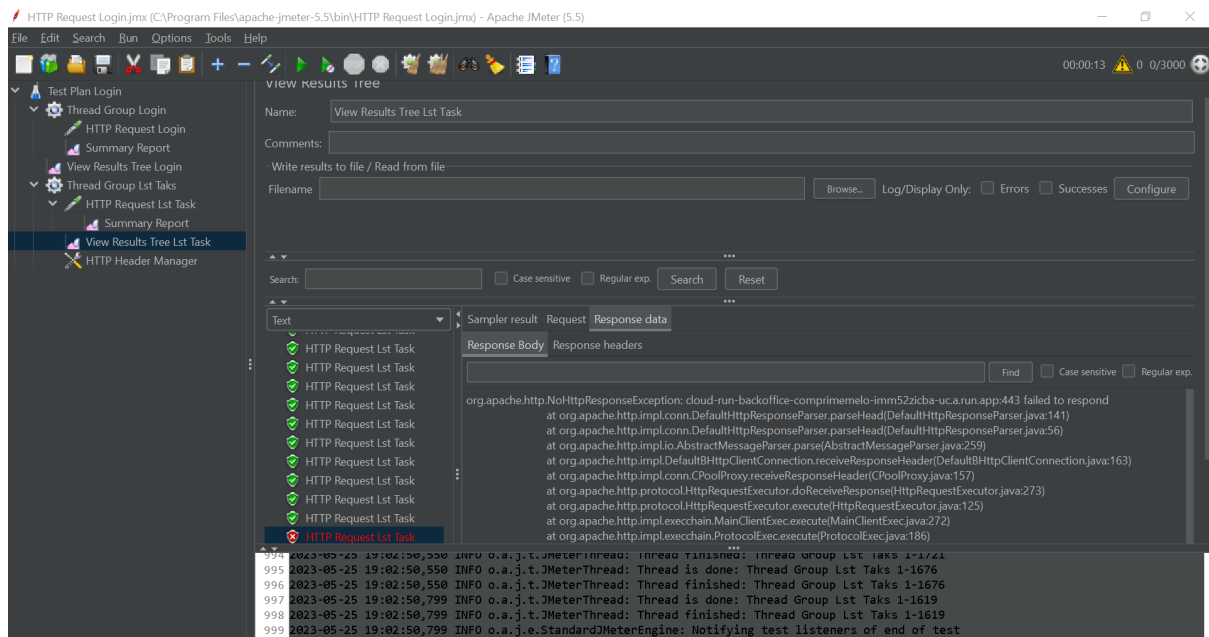
El throughput representa que se ejecutaron 297.7 peticiones por segundo, siendo este un valor óptimo según la cantidad de peticiones



Ejecución con un grupo de 3000 hilos concurrentes

Para esta cantidad de peticiones se presentó un error en la respuesta del servicio, el detalle de error es porque ocurrió un error interno en el servidor que hizo que el servicio de cloud run no pudiese responder.

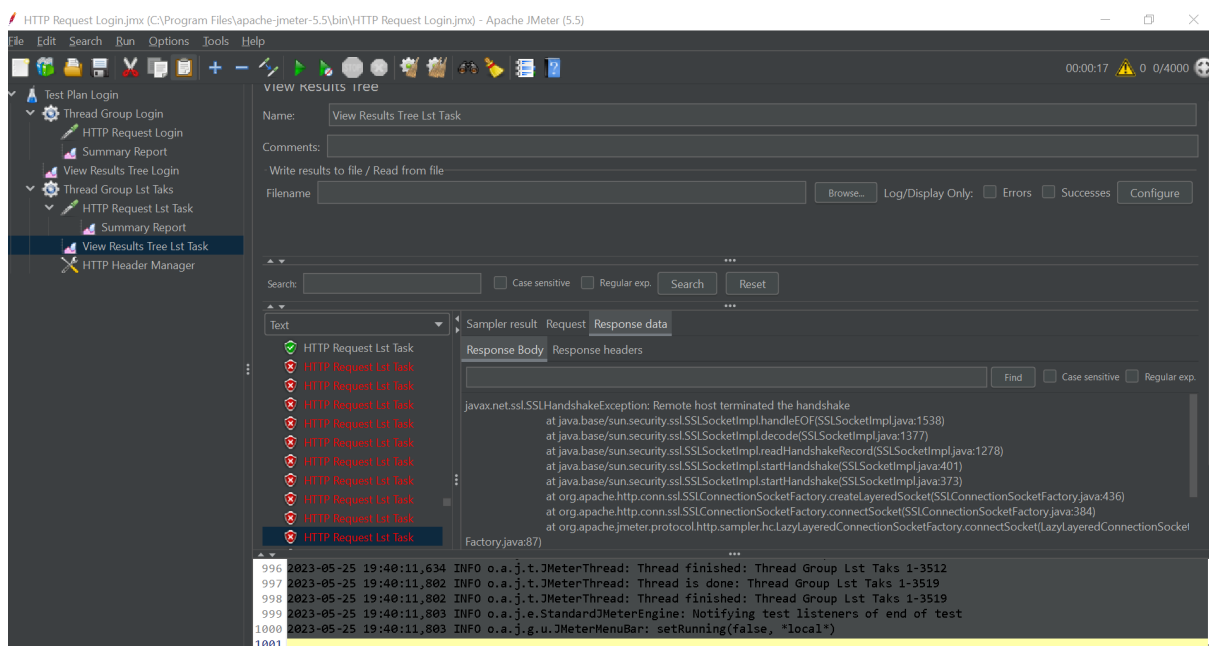
org.apache.http.NoHttpResponseException:
cloud-run-backoffice-comprimemelo-imm52zicba-uc.a.run.app:443 failed to respond



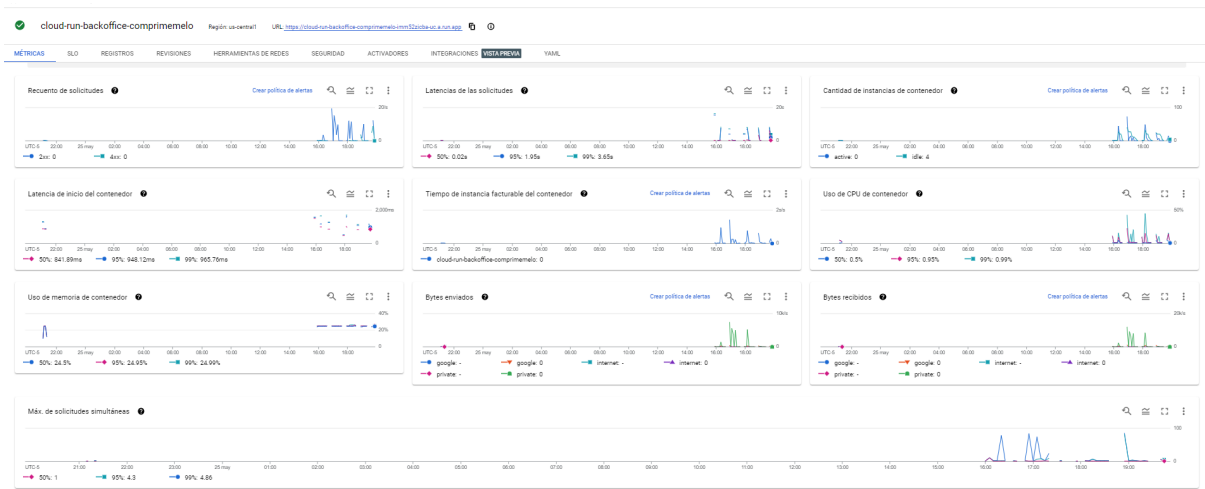
Ejecución con un grupo de 4000 hilos concurrentes

Se evidencia que después de las 3000 mil peticiones comienza a aparecer un error de socket cerrado.

javax.net.ssl.SSLHandshakeException: Remote host terminated the handshake
at java.base/sun.security.ssl.SSLSocketImpl.handleEOF(SSLSocketImpl.java:1538)



En las métricas de uso de recursos del servicio de cloud run se evidencia que después del proceso de pruebas de carga, aumentó significativamente todas las estadísticas de uso de CPU, Memoria, instancias automáticas de cloud run, etc.



En conclusión para el servicio de consulta de tareas el valor máximo permitido para peticiones concurrentes, acorde al actual infraestructura son 2800 peticiones concurrentes. Sin embargo se considera un valor que se encuentra dentro de los parámetros definidos en el acuerdo de nivel de servicio.

6. Bibliografía

<https://docs.celeryq.dev/en/stable/>

<https://flask.palletsprojects.com/en/2.2.x/>

<https://flask-jwt-extended.readthedocs.io/en/stable/>