



**Arquitectura Orientada a Servicios aplicada a
sistemas emergentes**

CE-5508

Grupo 01

Proyecto 1

Image Analyzer Infrastructure

Documentación

Prof. Alejandra Bolaños Murillo

Alumnos:

Mario Araya Chacón - 2018319178

Fabián Ramírez Arrieta - 2018099536

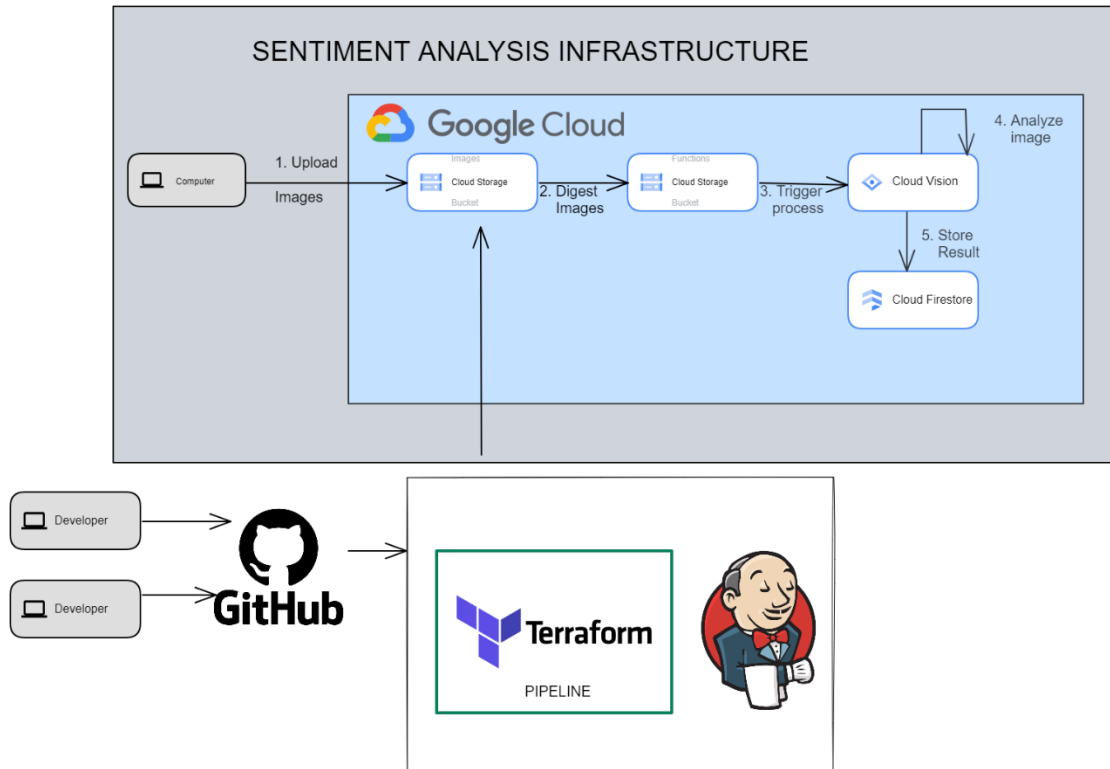
Mariana Vargas Ramírez - 2018086985

II Semestre 2022

Índice

Índice.....	2
Diagrama de arquitectura	3
Componentes y conectores	4
Bucket de funciones:	4
Bucket de imágenes:	4
GitHub:	4
Pipeline:	4
Cloud Vision:	5
Cloud Firestore:	5
Set de tecnologías	6
GitHub.....	6
Python	6
Google Cloud.....	6
Jenkins.....	7
GCP Vision API.....	7
GCP Cloud Firestore.....	7
Bibliografía	8

Diagrama de arquitectura



Se puede acceder de manera más detallada al diagrama de arquitectura por medio del siguiente [link](#).

Componentes y conectores

Bucket de funciones:

Almacena las funciones utilizadas por el sistema. Son las encargadas de ejecutar el análisis de las imágenes con el archivo seleccionado para comenzar con el proceso de reconocimiento de emociones.

Bucket de imágenes:

Componente de almacenamiento encargado de guardar las imágenes que van ingresando al sistema. Como parte de los requerimientos, se necesita guardar cada imagen que se ingrese al sistema y esto se hará en este bucket.

GitHub:

Controlador de versiones seleccionado. Acá se almacenarán las distintas ediciones del código. Este estará implementado con los conectores para poder aplicar los cambios de desarrollo en producción de una manera eficiente. Será la forma como los desarrolladores podrán darle mantenimiento y mejorar el código constantemente.

Pipeline:

Serie de procesos que hará que verificará la sanidad del código. Esto permitirá una integración continua para poder mejorar el código con regularidad. Su función será de hacer una revisión del código, en este caso por medio de Lint, ejecución de pruebas unitarias, un plan de Terraform y finalmente la ejecución de Terraform apply. El encargado de conectar todo de una manera eficiente y sencilla para los encargados es por medio de Jenkins, el cual funciona como conector entre desarrollo y producción.

Cloud Vision:

Esta API será la encargada de realizar todo el procesamiento de la imagen, luego de ser seleccionada por las funciones. Será por medio de la API publicada por Google, la cual tiene una eficiencia sumamente elevada, lo cual permitirá obtener resultados fiables.

Cloud Firestore:

Una vez analizada la imagen, se procederá a guardar el resultado del análisis en una base de datos de Firestore. Este análisis estará relacionado a la imagen seleccionada inicialmente para que de esta forma cumplir con el requerimiento de guardar las imágenes ingresadas y su análisis respectivo.

Set de tecnologías

A continuación se procederá a enumerar cada una de las tecnologías utilizadas y cuál es su función dentro de la solución implementada, además del porqué se decidió esta alternativa sobre las otras opciones que hay en el mercado.

GitHub

Utilizado como controlador de versiones para albergar el código. Es la opción más popular del mercado y con la que el equipo tiene mayor familiaridad, por estos motivos se optó por utilizar este producto. Además de poseer una muy buena integración con las herramientas que se verán a continuación.

Python

Se eligió Python como lenguaje de alto nivel para implementar la lógica del proyecto. Es uno de los lenguajes más populares a nivel mundial con una comunidad de desarrolladores sumamente extensa. Esto hace que su utilización sea más sencilla debido a la vasta documentación disponible. Además de la simpleza en su sintaxis.

Google Cloud

Herramienta para hospedar los distintos componentes de la aplicación y poder almacenarlos en la nube. Se eligió esta herramienta ya que al pertenecer a uno de los gigantes de la industria tal y como es Google hace que sea un producto sumamente pulido y de alta fiabilidad. Además que la documentación proveída por el fabricante es de gran calidad, lo que permite una guía de calidad durante el desarrollo de la aplicación.

Jenkins

Se decidió por usar Jenkins por encima de Cloud Build debido a su mayor popularidad. Se hizo más sencillo implementar el Pipeline con esta herramienta ya que al ser la herramienta líder en su sector, se encontraron gran cantidad de ayudas en la web. Jenkins posee muchos más usuarios en comparación a Cloud Build. [1]

GCP Vision API

Se eligió utilizar esta API ya que, similar a la herramienta anterior, al pertenecer al gigante de Google es una excelente alternativa para lograr el objetivo de la aplicación, la cual es reconocer las emociones de personas en las imágenes que se le provean. Esta inteligencia artificial, la cual funciona por algoritmos de Machine Learning, entre mejores sean los datos, mejores serán los resultados obtenidos y al ser esta API propiedad de Google se puede dar por un hecho que el conjunto de datos con el cual alimentan a la inteligencia artificial está sumamente depurado.

GCP Cloud Firestore

Finalmente, como herramienta para almacenar información, especialmente las imágenes procesadas y el análisis posterior se eligió Cloud Firestore. Nuevamente, esta herramienta pertenece al kit de herramientas creadas por Google, por lo cuál luego de haber seleccionado varias herramientas de esta compañía, el hecho de seguir dentro del mismo ecosistema hará que el desarrollo sea más uniforme.

Bibliografía

[1] Google Cloud Build vs Jenkins. (2022). Online. Available on:

<https://stackshare.io/stackups/google-cloud-build-vs-jenkins>