19 DE ABRIL DE 2022

MANUAL TECNICO

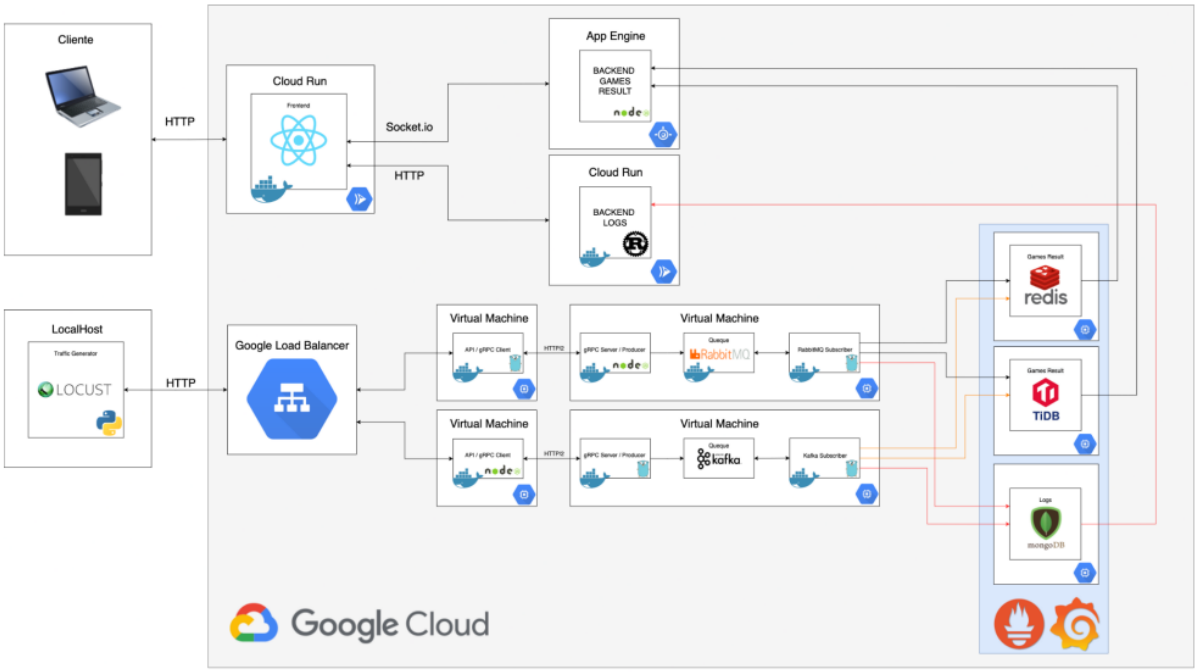
PROYECTO FASE 2

ARIEL RUBELCE MACARIO CORONADO
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Herramientas utilizadas.

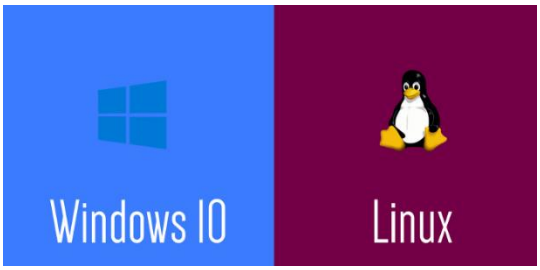
- Sistemas Operativos.
    - Ubuntu.
    - Windows.
- Lenguajes
    - Go
    - Node
    - Rust
    - Python
- Frameworks.
    - React js.
    - Locust
- Bases de datos.
    - Mongo DB
    - Redis
    - TidisDB
- Brokers.
    - RabbitMQ
    - Kafka.
- GCP
- Docker
- GRCP
- Monitoreo
    - Grafana
    - Prometheus

Arquitectura.



Sistemas Operativos.

Se utilizaron dos sistemas operativos para realizar el proyecto, ya que realizar la programación de los distintos módulos algunos se realizaron en Windows y otros en Linux, también se utilizó Linux en las maquinas virtuales ya que se encuentra mayor facilidad para realizar los distintos comandos.



Lenguajes.

Se utilizaron lenguajes de alta demanda laboral como lo es go, rust y node, se utilizo rust para crear un pequeño api para consumir una base de datos.

Se usó Go y Node en la mayoría de los módulos del proyecto, en algunos casos para realizar backend y en otro casos para realizar frontend.

Python se uso para realizar un tester de api para generar trafico para la aplicación.

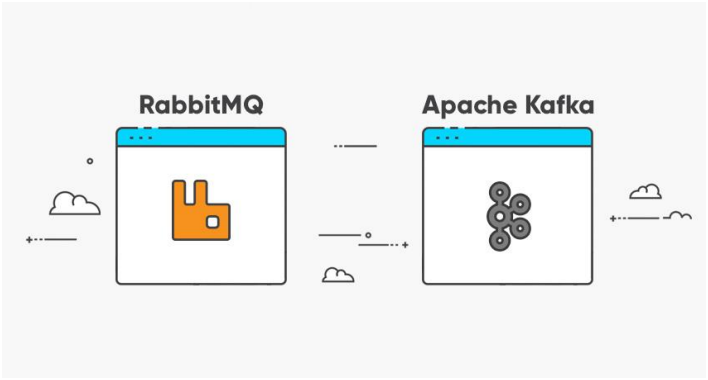
Bases de datos.

Se usaron 3 bases de datos las cuales son, mongo db, redis y tidis, las ultimas dos mencionadas son clave – valor y como bien conocemos mongo es documental, mongo se uso para guardar logs o un bitácora de juegos y las clave valor para mostrar datos en tiempo real.

Brokers.

Se usó rabbit MQ y Kafka como broker para recibir y encolar mensajes, como bien sabemos que rabbit es un borker de mensajería de código abierto distribuido y escalabre, que sirve como intermediario para la comunicación eficiente entre productores y consumidores.

Kafka que es una plataforma distribuida de transmisión de datos que permite publicar, almacenar y procesar flujos de registros, así como suscribirse a ellos de forma inmediata.



GCP.

Se trata de la suite de infraestructuras y servicios que Google utiliza a nivel interno y ahora y disponible para cualquier empresa de tal forma que sea aplicable a multitud de proceso empresariales.

Básicamente nos aparta todas las herramientas necesarias para diseñar hacer testing y lanzar aplicaciones desde gcloud con mucha mas seguridad y escalabilidad que cualquiera herramienta gracias a la propia infraestructura con la que Google cuenta.

Google Cloud

Docker.

Docker es una plataforma de software que le permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución. Con Docker, puede implementar y ajustar la escala de aplicaciones rápidamente en cualquier entorno con la certeza de saber que su código se ejecutará.

docker

Grafana.

Es una solución de código abierto que sirve para ejecutar análisis de datos, extraer métricas que dan sentido ante enormes cantidades de datos y monitorear aplicaciones y recursos hardware con la ayuda de atractivos paneles de control personalizables.

Prometheus

Prometheus es un software especializado como sistema de monitorización y alertas escrito en el lenguaje de programación Go. Todos los datos y métricas se almacenan en la base de datos como series temporales (junto al instante de tiempo en el que el valor se ha registrado). También es posible añadir etiquetas de tipo clave-valor junto a estas métricas.

Las métricas que almacena Prometheus pueden ser de cualquier tipo, y dependerán de la naturaleza de la aplicación o del sistema que se quiera monitorizar. Por ejemplo, puede ser el uso de CPU o de memoria, número de conexiones, número de peticiones o cantidad de sesiones activas. Todas las mediciones y métricas recogidas ayudarán a diagnosticar errores o problemas de servicio en los sistemas y aplicaciones que se monitorizan.