

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS



Gerber David Colindres Monterroso 201800586

Erick Alexander Alvarado Guerra 201800546

Osman Alejandro Pérez López 201801229

Tecnologías Utilizadas

Linkerd: es un proxy de red open source diseñado para ser desplegado como Servicemesh, su finalidad es hacer de link entre las distintas piezas de sistemas distribuidos y es de ayuda para las arquitecturas de microservicios.

Docker: es una herramienta diseñada para centrar código sin preocuparse si dicho código funcionará en la máquina en la que se ejecutará.

Kubernetes: Es una plataforma portátil y extensible de código para administrar cargas de trabajo y servicios. Esta puede ser vista como una plataforma de contenedores, microservicios o una plataforma portable de nube.

Kafka: 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.

RabbitMQ: Es un software de encolado de mensajes llamado broker de mensajería o gestor de colas. En este se pueden definir colas y las aplicaciones pueden conectarse a dichas colas.

React: Es una biblioteca de javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página.

Node js: es un entorno de tiempo de ejecución de javascript, este utiliza un modelo de entrada y salida sin bloqueo controlado por eventos que lo hace ligero y eficiente. Puede referirse a cualquier operación de leer o escribir archivos de cualquier tipo hasta hacer una solicitud HTTP.

WebSocket: Es una tecnología que proporciona un canal de comunicación bidireccional sobre un único socket TCP. Está diseñada para ser implementada en navegadores y servicios Web, pero puede utilizarse en cualquier aplicación cliente/servidor.

Nginx: está diseñado para ofrecer un bajo uso de memoria y alta concurrencia. En lugar de crear nuevos procesos para cada solicitud web usa un enfoque sincrónico basado en eventos donde las solicitudes se manejan en un solo hilo.

Golang: Es un lenguaje de programación compilado que toma ideas de los lenguajes interpretados para tener una compilación más rápida. Por ejemplo, Go permite compilar cross-platform, si estás en Linux puede compilar para Windows u otro sistema operativo, sin necesidad de irte a esa máquina específica para compilar ahí.

GRPC: Es un sistema de llamada por procedimiento remoto de código abierto desarrollado por google.

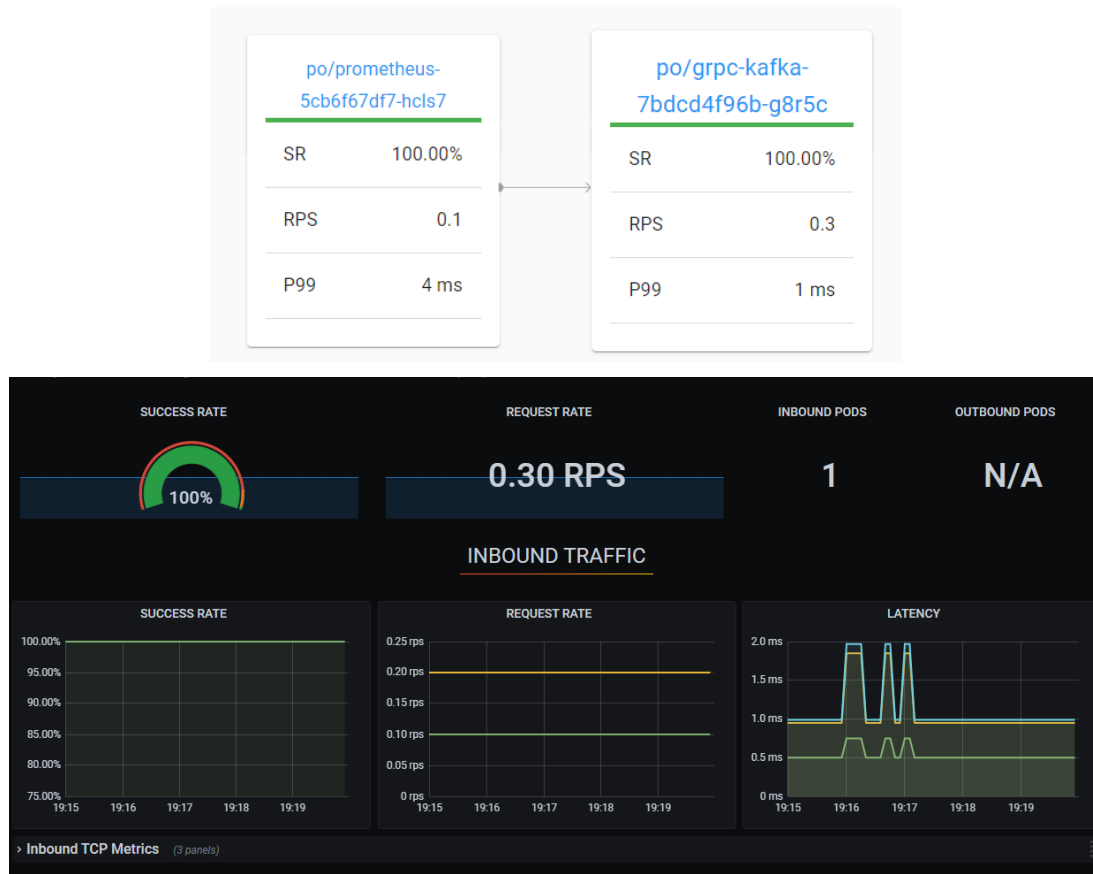
Redis: Es un sistema de base de datos en memoria, basado en el almacenamiento de tablas de hashes, pero que opcionalmente puede ser usada como una base de datos durable o persistente.

MongoDB: Es un sistema de bases de datos NoSQL orientado a documentos y de código abierto.

Preguntas

- Cómo funcionan las métricas de oro, cómo puedes interpretar estas 7 pruebas de faulty traffic, usando como base los gráficos y métricas que muestra el tablero de Linkerd Grafana.

Las métricas oro funcionan para medir el costo y rendimiento de un pod en kubernetes.



PubSub, Kafka y Rabbit

- ¿Qué sistema de mensajería es más rápido?

Kafka es más rápido que los demás debido a su arquitectura para almacenar los eventos o mensajes recibidos, distribuye los topics entre los nodos mediante particiones.

- ¿Cuáles son las ventajas y desventajas de cada sistema?

Kafka

Ventajas: buena escalabilidad, tolerancia a fallos y en tiempo real.

Desventajas: depende de un zookeeper.

Rabbit

Ventajas: Adecuado para muchos protocolos de mensajería, flexibilidad y *plugins* disponibles, herramientas de desarrollo varias.

Desventajas: El desarrollo de Erlang es difícil de entender el código fuente. Tiene un rendimiento más bajo debido al mecanismo de implementación más pesado.

- ¿Cuál es el mejor sistema?

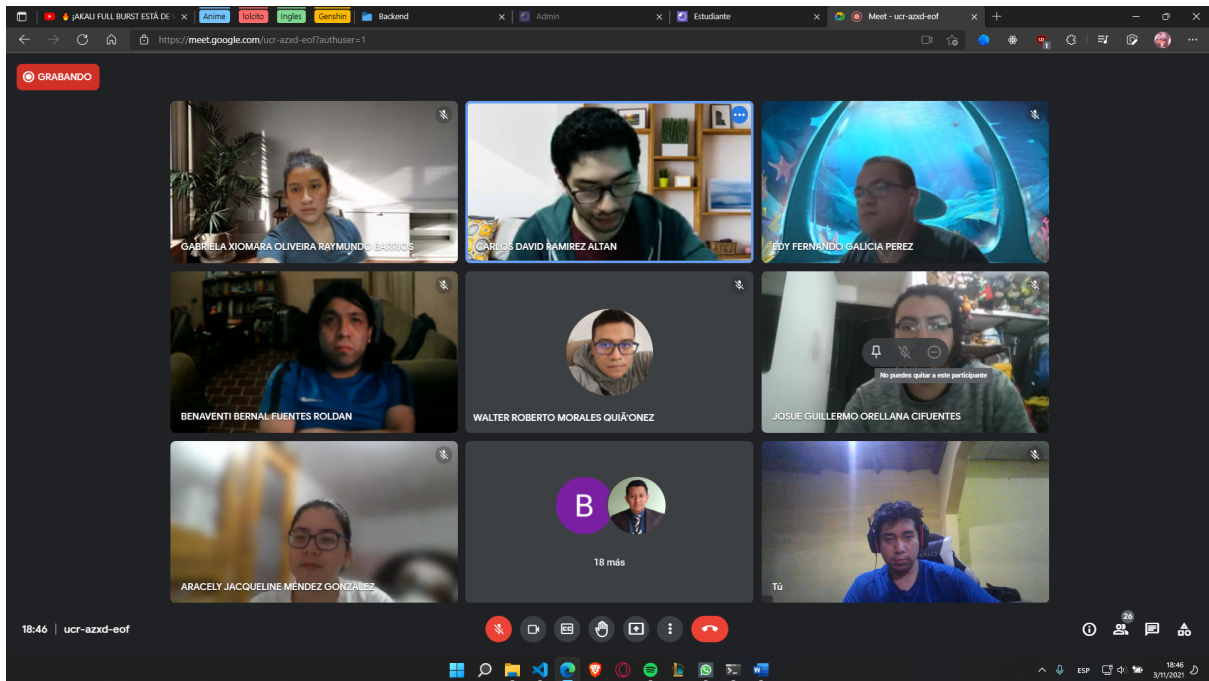
Kafka debido a su característica principal de Kafka es procesar el consumo de mensajes en función del modo Pull y perseguir un alto rendimiento.

Bases

- ¿Cuál de las dos bases se desempeña mejor y por qué?

La base de datos que tiene un mejor desempeño es mongodb, debido a su eficiencia y persistencia de datos. Esto claramente depende del uso que se le dará a la información, ya que si la persistencia no es de suma importancia redis posee mejores tiempos de lectura debido a su almacenamiento en memoria caché.

Screenshots



Gerber David Colindres Monterroso

Manual de usuario

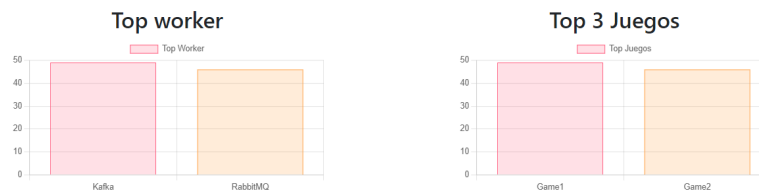
Reportes Mongo

Página principal de mongo donde se ve varios reportes entre ellos tenemos:

- Top worker: Este muestra el top de cuantos registros hay de cada pubsub enviado a mongo.
- Top 3: Este muestra el top de mas juegos ingresados a mongo.
- Datos Almacenados: tabla donde muestra todos los datos almacenados en mongo.



Reportes Mongo



Datos almacenados

id	Game Name	Winner	Players	Worker
1	Game1	8	25	RabbitMQ
2	Game2	5	12	Kafka
2	Game2	0	1	Kafka
2	Game2	4	6	Kafka
1	Game1	4	8	RabbitMQ
1	Game1	2	11	RabbitMQ
2	Game2	13	23	Kafka
2	Game2	4	11	Kafka
2	Game2	7	7	Kafka
2	Game2	3	3	Kafka

Reportes Redis

Página principal de redis donde se ve varios reportes entre ellos tenemos:

- Ultimos Juegos: Este reporte muestra en una tabla los ultimos juegos insertados en redis.
- Mejores Jugadores: En este reporte muestra una tabla donde hay un top de los jugadores que más han ganado en los juegos.
- Estadísticas en tiempo real: Este es un reporte donde se pedirá un jugador y mostrará en tiempo real el último juego y si ganó dicho jugador.



Estadísticas en tiempo real del jugador:

id	Game Name	State
		Lost

Reportes Redis

Ultimos juegos

id	Game Name	Winner	Players	Worker
----	-----------	--------	---------	--------

Mejores Jugadores

Player	Wins
--------	------