Nombre: César Armando Cruz Mendoza

Entregable: Práctica del módulo Big Data Arquitectura

Diseño del DAaas

**Definición de la estrategia del DAaaS**

Servicio de notificación a usuarios del Sistema de Transporte Metro de la CDMX, que informe mediante la cuenta de Twitter @MetroCDMX y pantallas de andenes, sobre el estado del servicio y anticipe eventuales retrasos que se puedan presentar en el servicio de la Línea 1 (Rosa), con el objetivo de optimizar los tiempos de desplazamiento de los usuarios y mejorar la percepción de la calidad del servicio, complementado con un dashboard de monitoreo para análisis de la alta Dirección del STC Metro.

La propuesta de implementar el servicio en la Línea 1 obedece a los siguientes factores:

* La Línea 1 cuenta con más de 180 sensores IoT que detectan el paso del tren en ambos sentidos (Pantitlán – Observatorio) y (Observatorio – Pantitlán)
* Los sensores IoT notifican a un servicio en JavaScript vía API REST, su estado (1 tren presente, 0 tren ausente)
* La Línea 1 es la de mayor afluencia con registros de brindar servicio en 2020 a 141,606,476 pasajeros.
* Los usuarios de la CDMX utilizan la cuenta de Twitter @MetroCDMX para manifestar inconformidad en el servicio e informarse de lo que sucede con el servicio.

**Arquitectura DDaaS**

* Servicio que concentra la información de los sensores IoT de la Línea 1
* Servicio en JavaScript para generar suscripción a la información de los sensores IoT
* MongoDB para almacenar la información de los sensores y que será la fuente de datos para los procesos de análisis
* Implementación de un servicio para extraer información de los tweets de la cuenta @MetroCDMX y colocarlos en Kafka.
* Servicio de Kafka
* Máquina virtual con servicio para analizar frecuencia de palabras claves que ayuden a identificar reportes de usuarios por problemas en el servicio mediante el hashtag #MetroCDMX.
* Máquina virtual con servicio que analice el desplazamiento de los trenes y estime los tiempos de desplazamiento por estación
* Máquina virtual con servicio que analice la información de trenes y sentimiento de mensajes en Twitter
* Máquina virtual con servicio para enviar avisos a la cuenta @MetroCDMX
* Máquina virtual con servicio para desplegar informe ejecutivo en dashboard (sitio web)

**DDaas Operating Model Design and Rollout**

1. Habilitar un acceso vía VPN para realizar el proceso de suscripción del microservicio de la MV – Consumidor IoT desde Google Cloud a la infraestructura interna de la red del STC Metro.
2. Realizar el aprovisionamiento todas las máquinas virtuales en Google Cloud para controla la optimización de costos, siendo las siguientes:
   1. Máquina virtual para el microservicio que extraerá la información de Twitter, de acuerdo al hashtag establecido. Mismo que se encargará de suministrar la información al clúster de Kafka.
   2. Máquina virtual para el microservicio que extraerá la información del estado de los sensores IoT de la Línea 1. Mismo que se encargará de suministrar la información al clúster de Kafka.
   3. Máquina virtual con MongoDB.
   4. Máquina virtual con clúster de Kafka.
   5. Máquina virtual para los microservicios que consumirán la información de Kafka y la almacenará en MongoDB, así mismo, la proveerá para que el microservicio de análisis de tweets y operación de los trenes.
   6. Máquina virtual para el microservicio que se encargará de publicar los tweets necesarios, en función de los análisis de información que se realicen.
   7. Máquina virtual para el servicio del sitio web que despliega el dashboard ejecutivo para Dirección del STC Metro.
3. En la sección de Compute Engine / Instancias de VM, crear un programa para gestionar automáticamente el proceso de encendido / apagado de las máquinas virtuales y optimizar el consumo de recursos. Se tomará en cuenta que el servicio de transporte es de 5am a 12pm.
4. Las especificaciones del programa de encedido / apagado son:
   1. Establecer la zona horaria para la CDMX.
   2. Como inicio de la máquina virtual, establecer las 4:00 hrs.
   3. Como hora para detener el servicio, establecer las 23:59 hrs.
   4. Dejar fecha de inicio y finalización en blanco.
5. En el programa de instancias creado, incorporar todas las máquinas del punto 2.
6. Máquina virtual para el microservicio que extraerá la información de Twitter.
   1. Este servicio realizará la obtención de los tweets asociados a la cuenta @MetroCDMX y el hastag #MetroCDMX.
   2. Se considera que el servicio estará desarrollado en Python.
   3. La definición del hashtag se podrá ajustar mediante la actualización del archivo de búsqueda localizado entro de la MV, el cual será actualizado por el usuario manualmente en el caso de requerir.
   4. Se encargará de entregar la información al clúster de Kafka.
7. Máquina virtual para el microservicio que extraerá la información del estado de los sensores IoT.
   1. Este servicio se encargará de obtener el flujo de eventos del estado de los diferentes sensores de la Línea 1.
   2. Se considera que el servicio sea desarrollado con Javascript para facilitar la conexión vía SocketIO con microservicio que gestiona la distribución de información.
   3. Se encargará de entregar la información al clúster de Kafka.
8. Máquina virtual con clúster de Kafka.
   1. Contiene el clúster de Kafka que se dimensionará en capacidad a partir de pruebas con los flujos de datos que se reciban.
9. Máquina virtual para los consumidores de Kafka.
   1. Contendrá el microservicio que consumirá la información de los tweets y que los entrega al microservicio de análisis.
   2. Contendrá el microservicio que consumirá la información de los sensores IoT y los entrega al microservicio de análisis de operación de trenes.
   3. En ambos casos, la información se almacena en MongoDB para efectos de tener datos estructurados que faciliten un análisis posterior de mayor temporalidad.
10. Máquina virtual con MongoDB.
11. Máquina virtual para los microservicios de análisis de sentimiento y notificaciones.
    1. Contendrá el microservicio que ejecuta el análisis del contenido de los tweets por la ventana de tiempo establecida. La información de salida del análisis deberá permitir:
       1. Identificar inconformidad en los usuarios en la Línea 1.
       2. Identificar el sentido de vías afectadas: vía 1, vía 2 o ambos.
       3. Aislar el problema a nivel de estación.
       4. Identificar la hora de inicio y final del problema.
       5. Concluir si existe un problema que deba reportarse.
    2. Contendrá el microservicio que deberá:
       1. Llevar un registro de la frecuencia de paso de los trenes por estación en cada una de las vías (1 y 2).
       2. Identificar tendencias de incremento de los tiempos de frecuencia.
       3. Identificar cuando los trenes se encuentren en alto total
       4. Habilitar la publicación en tiempo real del estado de los trenes para el dashboard
       5. Integrar el resultado de análisis de los tweets con el análisis de avance de trenes, para generar las recomendaciones que informen:
          1. Anticipar a estaciones posteriores al problema, que el sistema presenta retraso en avance.
          2. Identificar el alcance de la notificación para saber qué estaciones son afectadas.
          3. Generar la notificación vía Twitter.
          4. Generar la notificación al microservicio que gestiona mensajes a las pantallas disponibles en los andenes de las estaciones de la Línea 1.
12. Máquina virtual para el dashboard ejecutivo.
    1. Contendrá el aplicativo para el despliegue del dashboard con la información en tiempo real proporcionada por los microservicios de análisis de sentimiento y notificaciones.
    2. El desarrollo se considerará con NodeJS, Javascript, Express
    3. El usuario podrá ver un concentrado ejecutivo del estado de la Línea 1 del Metro.

**Diagrama de solución**

