UNIVERSIDAD EAFIT
DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS
ST1612 SISTEMAS INTENSIVOS EN DATOS

2021-2

Lab 5 - AWS Kinesis + S3 + DynamoDB

Contexto del laboratorio:

- Tenemos una empresa retail ACME que vende una serie de productos en sus tiendas.
- El servidor principal donde tiene su sistema de información principal de ventas, genera Logs por cada compra que realizan sus clientes.
- Las transacciones son simuladas desde el archivo: OnlineRetail.csv y son generadas como Logs por el programa python: LogGenerator.py
- En el servidor principal, se instalará un Agente de Software para AWS Kinesis, que envía a Kinesis Firehose y Kinesis Data Streams los datos de los logs.
- Los logs que son recolectados por Firehose son Almacenados en AWS S3 particionados por fecha/hora, para posteriormente ser catalogados por AWS Glue y consultados por AWS Athena como tablas SQL.
- Los logs que son recolectados por Data Streams son consumidos por 2 servicios: 1) un programa standalone llamado ‘Consumer.py’ que lee de la API AWS los datos de Data Streams, los imprime en consola y los almacena en AWS DynamoDB. 2) una función lambda, que incluye gran parte del código ‘Consumer.py’, y que se activa mediante un Trigger de Kinesis Data Stream cuando llegan datos, una vez ejecuta la función lambda, lee los datos y son almacenados en una base de datos DynamoDB.
- Queda para un procesamiento futuro el procesamiento de los datos en DynamoDB para visualización o para notificaciones.

Código fuente e instrucciones:

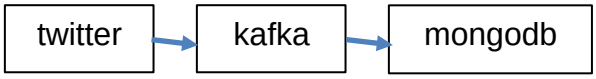
- Todas las indicaciones del lab 5 se encuentran en el github de la materia:
- https://github.com/st1612eafit/st1612_20212.git:
  - https://github.com/st1612eafit/st1612_20212/tree/main/kinesis
- Además, puede consultar la presentación de clase sobre AWS Kinesis donde se explica la fundamentación del servicio y se ilustra una serie de

casos de uso de kinesis, dos de los cuales están incluidos en este laboratorio. Consultar la presentación en Interactiva Virtual.

- El Lab 5 se puede realizar completamente en AWS Educate.

RETO: crear una desde la API de twitter lectura de tuits, enviarlos vía Kafka (elegir MSK o LOCALHOST) y en el consumidor almacenarlos en una base de datos mongodb.

Arquitectura propuesta:



realizar ejemplos de consultas en mongodb de algunos tuits.