Project Milestone - Data Ingestion Software - Kafka Clusters

Student: Alex 100599423

Google Drive Folder (videos inside):
https://drive.google.com/drive/folders/1HBc4QRKMGfRz12eH9tkgMaCyU94588aB?usp=sharing

1. **What is EDA? What are its advantages and disadvantages?**

   EDA stands for event driven architecture. This means whenever there is a change of state in a system an event occurs and some action may or may not take place.

   Advantages

   - Systems using EDA are generally loosely coupled due to the separation between producers and consumers
   - Performant due to the need to not constantly poll for messages. In other words, it can be used to stream real time data cost effectively.
   - Fault tolerant as publishers and subscribers are decoupled.
   - Scalability, distributed nature of event handling allows a solution like Kafka to be used for web scale applications

   Disadvantages

   - Duplicated events depending on the implementation may need to be delt with, creating additional work.
   - Hard to debug and troubleshoot due to the massive amount of data constantly being passed around
   - Unclear event processing pipeline, since 1 event may trigger another event and so on.

2. **In Kafka, what's meant by cluster, broker, topic, replica, partition, zookeeper, controller, leader, consumer, producer, and consumer group?**
   - Cluster – contains one or more brokers
   - Broker – Handles messages from clients that produce and consume data. Also, it keeps the data replicated within the cluster.
   - Topic - A place where a category of messages are stored/retrieved from by clients
   - Replica – A "backup" copy of a partition
   - Partition – A data store for messages often distributed over many brokers
   - Zookeeper – Maintains the cluster by tracking status of nodes, topics and enforcing access control lists and quotas.
   - Controller – a single broker in a cluster that is responsible for state management of partitions and replicas
   - Leader – The leader handles all read/write requests and replicas, replicate the changes.
   - Consumer – A client that reads messages
   - Producer - A client that produces messages
   - Consumer Group – a group of consumers that reads messages from a topic at different offsets so parallel processing can happen.

3. **With the provided YAML file Kafka data is not persistent why?**
   Updating the docker-compose.yaml file to include volumes for the Kafka data directories makes the Kafka data persistent. The provided docker-compose.yaml file does not make use of volumes so data is stored within the container and when it's destroyed, the data is lost.

4. **Kafka in Confluent Cloud**

   Creating Topic
   - confluent kafka topic create lab2 –partitions 1 –if-not-exists

   Creating Consumer
   - Confluent kafka topic consume lab2

   Creating Producer
   - confluent kafka topic produce –delimiter : --value-format string lab2