Apache Kafka Workshop Handout

Introduction to Apache Kafka

- **What is Apache Kafka?**
    - A distributed streaming platform, originating from LinkedIn, used for building real-time data pipelines and streaming apps.
    - It's horizontally scalable, fault-tolerant, and incredibly fast.
- **Key Concepts:**
    - **Producers:** Applications that publish (write) events to Kafka topics.
    - **Consumers:** Applications that subscribe to (read) events from Kafka topics.
    - **Brokers:** Kafka servers that store data and serve clients.
    - **Topics:** Categories or feeds to which records are published.
    - **Partitions:** How Kafka topics are split for scalability and parallel processing.
    - **Offsets:** Position of a consumer in a partition.

Kafka Architecture

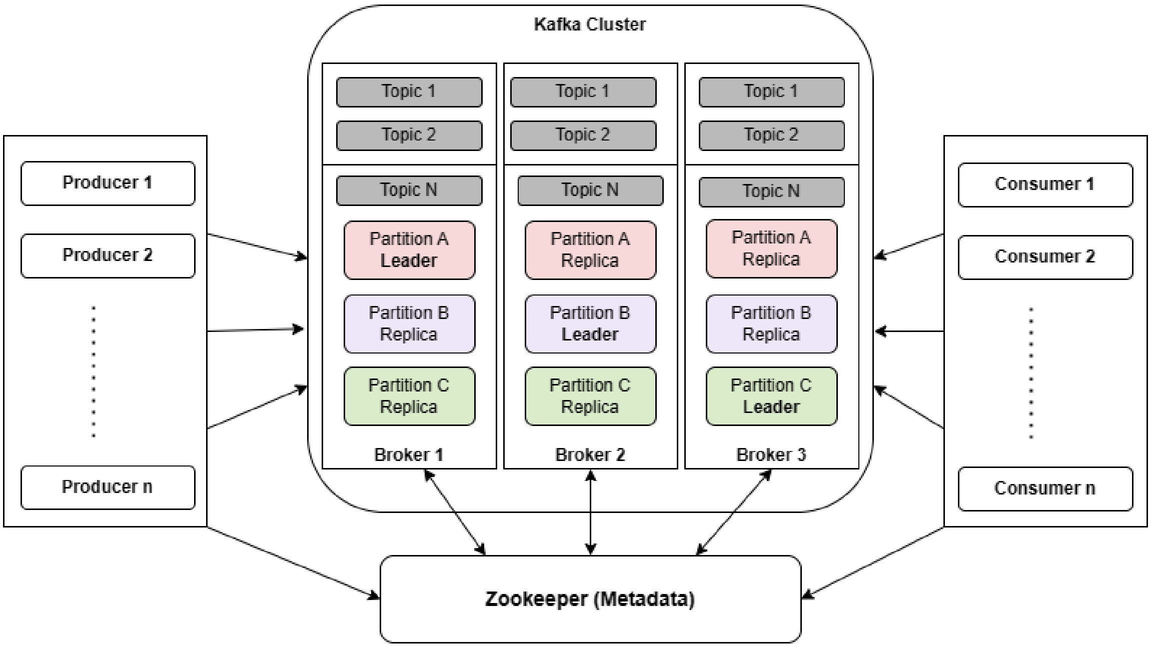
- **Kafka Brokers and Clusters**
    - A single Kafka server is called a broker.
    - A cluster is a group of brokers working together.
- **Role of Zookeeper**
    - Manages cluster metadata and coordinates the Kafka brokers.
    - **Note:** Recent Kafka versions (2.8.0+) can run in KRaft mode without Zookeeper.
- **Transition to KRaft:**
    - Kafka has traditionally used Zookeeper for cluster coordination and metadata management.
    - The new KRaft mode (Kafka Raft Metadata mode) eliminates the need for Zookeeper and simplifies Kafka's architecture.
    - KRaft provides Kafka with its own consensus mechanism for metadata management, making Kafka deployments more straightforward and efficient.

Kafka Producers and Consumers

- **Message Serialization:**
    - Data can be serialized as strings, JSON, Avro, etc.

Kafka Topics and Partitions

- **Understanding Partitions:**
    - Each partition is an ordered, immutable sequence of records.
    - Partitions allow Kafka to parallelize processing.
- **Replication for Fault Tolerance:**
    - Topics can be replicated across multiple brokers for fault tolerance.



Kafka Stream Processing

- **Basics of Kafka Streams API:**
  - A client library for building applications and microservices to process and analyze data stored in Kafka.
- **Building Streaming Applications:**
  - Example of a simple stream processing application using Kafka Streams.

Kafka Connect and Integrations

- **Overview of Kafka Connect:**
  - A tool for scalably and reliably streaming data between Apache Kafka and other systems.
- **Setting Up Connectors:**
  - Configuring connectors for importing data from databases, logs, etc., into Kafka topics, and exporting data from Kafka topics to external systems.

Kafka Security

- **Encryption and Authentication:**
  - SSL/TLS for encryption.
  - SASL/SCRAM or Kerberos for authentication.
- **Authorization:**
  - Using Kafka’s ACLs (Access Control Lists) for authorization.

Resources for this handout and further learning

https://howtodoinjava.com/kafka/apache-kafka-tutorial/

https://www.tutorialspoint.com/apache_kafka/index.htm

https://medium.com/@patelharshali136/apache-kafka-tutorial-kafka-for-beginners-a58140cef84f