Project: Building a Data Ingestion Pipeline with Apache Sqoop, Flume, and Kafka on a Local Machine

Part 1: Data Migration with Apache Sqoop

- ## Overview

  This part of project demonstrates data migration from a local relational database (MySQL or SQLite) to Hadoop's HDFS using Apache Sqoop. The process involves setting up a local database, populating it with sample data, importing the data to HDFS, and configuring incremental imports to handle new data entries efficiently.

- ## Steps to Create the Local Database

  1. Create a Database (labs):
  2. Create a Table (authors) and Insert Sample Data:

- ## Create a target directory in HDFS to import table data into

  Command**:** $hdfs dfs -mkdir /mywarehouse

- ## Import Data to HDFS with Sqoop

  ### 1. Full Data Import

  - **Sqoop Command:**

    ```
    $ sqoop import --connect jdbc:mysql://localhost/labs \
    --username student --password student \
     --table authors --fields-terminated-by '\t' \
    --target-dir /mywarehouse/authors_table
    ```

```
[student@192 ~]$ hdfs dfs -ls /mywarehouse/authors_table
Found 5 items
-rw-r--r--   1 student supergroup          0 2024-08-08 08:30 /mywarehouse/authors_table/_SUCCESS
-rw-r--r--   1 student supergroup     189746 2024-08-08 08:29 /mywarehouse/authors_table/part-m-00000
-rw-r--r--   1 student supergroup     190603 2024-08-08 08:29 /mywarehouse/authors_table/part-m-00001
-rw-r--r--   1 student supergroup     190637 2024-08-08 08:30 /mywarehouse/authors_table/part-m-00002
-rw-r--r--   1 student supergroup     190593 2024-08-08 08:29 /mywarehouse/authors_table/part-m-00003
[student@192 ~]$
```

2. **Incremental Data Import**

   Note: I inserted new rows in ‘authors’ table for testing

   - **Sqoop Command:**

     $ sqoop import --connect jdbc:mysql://localhost/labs \
     --username student --password student \
     --table authors --fields-terminated-by '\t' \
     --target-dir /mywarehouse/authors_table \
     --incremental append \
     --check-column id --last-value

- Validation and Testing

- **Verify Data in HDFS**

```
[student@192 ~]$ hdfs dfs -ls /mywarehouse/authors_table
Found 5 items
-rw-r--r--   1 student supergroup          0 2024-08-08 08:30 /mywarehouse/authors_table/_SUCCESS
-rw-r--r--   1 student supergroup     189746 2024-08-08 08:29 /mywarehouse/authors_table/part-m-00000
-rw-r--r--   1 student supergroup     190603 2024-08-08 08:29 /mywarehouse/authors_table/part-m-00001
-rw-r--r--   1 student supergroup     190637 2024-08-08 08:30 /mywarehouse/authors_table/part-m-00002
-rw-r--r--   1 student supergroup     190593 2024-08-08 08:29 /mywarehouse/authors_table/part-m-00003
[student@192 ~]$ hdfs dfs -ls /mywarehouse/authors_table
Found 9 items
-rw-r--r--   1 student supergroup          0 2024-08-08 08:30 /mywarehouse/authors_table/_SUCCESS
-rw-r--r--   1 student supergroup     189746 2024-08-08 08:29 /mywarehouse/authors_table/part-m-00000
-rw-r--r--   1 student supergroup     190603 2024-08-08 08:29 /mywarehouse/authors_table/part-m-00001
-rw-r--r--   1 student supergroup     190637 2024-08-08 08:30 /mywarehouse/authors_table/part-m-00002
-rw-r--r--   1 student supergroup     190593 2024-08-08 08:29 /mywarehouse/authors_table/part-m-00003
-rw-r--r--   1 student student           229 2024-08-08 08:38 /mywarehouse/authors_table/part-m-00004
-rw-r--r--   1 student student           152 2024-08-08 08:39 /mywarehouse/authors_table/part-m-00005
-rw-r--r--   1 student student           168 2024-08-08 08:39 /mywarehouse/authors_table/part-m-00006
-rw-r--r--   1 student student           235 2024-08-08 08:39 /mywarehouse/authors_table/part-m-00007
[student@192 ~]$
```

- Future Enhancements

- **Automate Incremental Imports**: Use a cron job or another scheduler to automate the incremental import process.
- **Data Transformation**: Implement data transformation using Apache Hive or Apache Spark before storing the data in HDFS.

Part 2: Real-Time Data Ingestion with Apache Flume and Apache Kafka

- Overview

This project demonstrates the setup and configuration of a real-time data ingestion pipeline using Apache Flume and Apache Kafka. The objective is to capture log data from a local directory using Flume, send it to a Kafka topic, and store/manage the incoming log data using Kafka.

- Environment Preparation

1. Stop any running services (HBase, Kafka, Zookeeper):

   ```
   sudo stop-hbase.sh
   sudo systemctl stop kafka
   sudo systemctl stop zookeeper
   ```

2. Start Zookeeper and Kafka:

   ```
   sudo systemctl start zookeeper
   sudo systemctl start kafka
   ```

3. Configure Apache Flume

   ```
   agent1.sources = streaming-txt-source
   agent1.sinks = kafka-sink logger-sink
   agent1.channels = memory-channel
   agent1.sources.streaming-txt-source.type = spooldir
   agent1.sources.streaming-txt-source.spoolDir = /home/student/Labs/C3U4/spool
   agent1.sinks.kafka-sink.type = org.apache.flume.sink.kafka.KafkaSink
   agent1.sinks.kafka-sink.topic = stream_text
   agent1.sinks.kafka-sink.brokerList = localhost:9092
   agent1.sinks.kafka-sink.batchSize = 5
   agent1.channels.memory-channel.type = memory
   agent1.channels.memory-channel.capacity = 10000
   agent1.channels.memory-channel.transactionCapacity = 100
   agent1.sinks.logger-sink.type = logger
   agent1.sources.streaming-txt-source.channels = memory-channel
   agent1.sinks.kafka-sink.channel = memory-channel
   agent1.sinks.logger-sink.channel = memory-channel
   ```

4. Configure Apache Kafka

- Create a new Kafka topic:
  Command:
  kafka-topics --create \
  --bootstrap-server localhost:9092 \
  --replication-factor 1 \
  --partitions 1 \
  --topic stream_text

5. Prepare the Spool Directory

cd /home/student/Labs/C3U4
mkdir spool

- Running the Real-Time Data Ingestion Pipeline

1. Start the Flume Agent
flume-ng agent --conf $FLUME_HOME/conf \
--conf-file /home/student/Labs/C3U4/spooldir.conf \
--name agent1 -Dflume.root.logger=INFO,console

2. Generate Log Data

- Open another terminal and run a script to generate log data:

python ./spool_stream.py ./spool 5000 ~/Data/alice_in_wonderland.txt

3. Consume Data from Kafka

- Open a new terminal (consumer) and consume the data from the Kafka topic:

kafka-console-consumer \
--bootstrap-server localhost:9092 \
--topic stream_text \
--from-beginning

- **Validation and Testing**
    - **Check Kafka Topic:** Ensure that data is being consumed by the Kafka topic stream_text.
    - **Monitor Flume Logs:** Check Flume logs to ensure that data is being captured and forwarded correctly.