2024

Building a Data Ingestion Pipeline

Instructor:
Eng. Ahmed Jamal
Eng. Salma Hegazy
Eng. Tawfik Yasser

Facilitator:
Eng. Ahmed Abdelnasser

1 INTRODUCTION

1.1 OVERVIEW

This project involves building a data ingestion pipeline that integrates both batch data migration and real-time streaming data ingestion using Apache Sqoop, Apache Flume, and Apache Kafka. The goal is to create a robust and scalable pipeline that efficiently handles both historical and real-time data.

1.2 OBJECTIVE

The objective is to design and implement a pipeline that can:

- Migrate historical data from a relational database to Hadoop's HDFS.
- Capture and ingest real-time streaming data, simulating a continuous flow of data.

1.3 TOOLS AND TECHNOLOGIES

1. **Apache Sqoop**: For batch data migration from MySQL to HDFS.
2. **Apache Flume**: For real-time data ingestion from a local directory to Kafka.
3. **Apache Kafka**: For buffering and streaming real-time data.
4. **Hadoop HDFS**: As the storage layer for both batch and streaming data.

1.4 USE CASE

You need to set up a robust data ingestion pipeline to handle both historical data and simulated real-time data streams on your local machine. The pipeline will involve migrating data from a local relational database to Hadoop's HDFS, handling real-time streaming data from local files, and ensuring seamless data flow using Apache Kafka.

2 BATCH DATA INGESTION WITH APACHE SQOOP

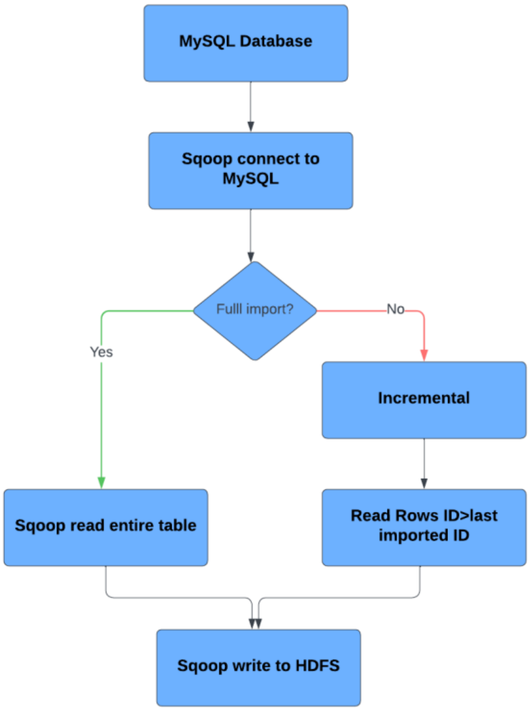
2.1 INTRODUCTION TO BATCH DATA

Batch data ingestion involves transferring large amounts of structured data from one system to another in bulk. This process is typically scheduled at regular intervals and is suitable for scenarios where data does not need to be processed in real-time.

2.2 ENVIRONMENT SETUP

1. **Hadoop**: Installed on a local machine or a cluster to provide the HDFS storage.
2. **MySQL**: Used as the relational database where historical data is stored.
3. **Sqoop**: Installed to facilitate the data transfer between MySQL and HDFS.

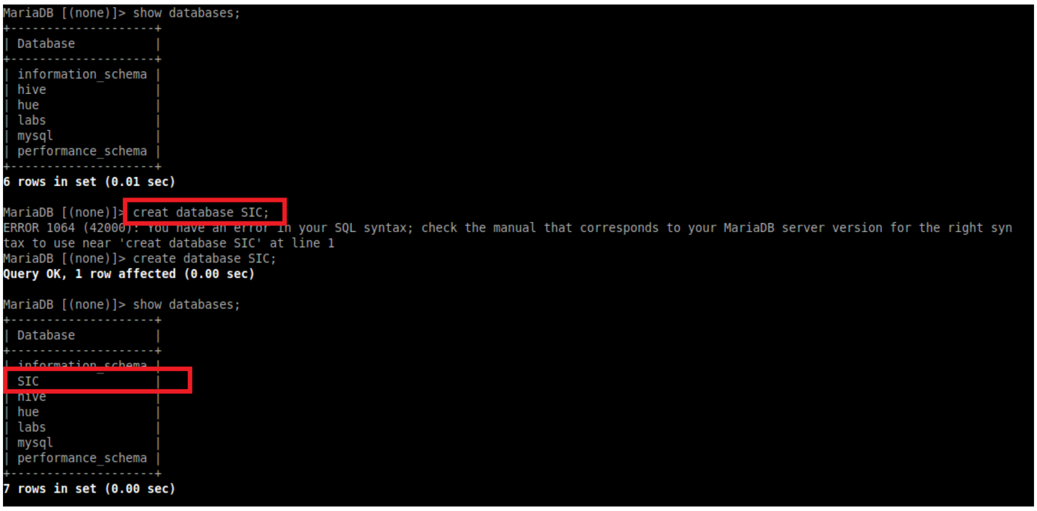
2.3 FLOW CHART



2.4 DATABASE PREPARATION

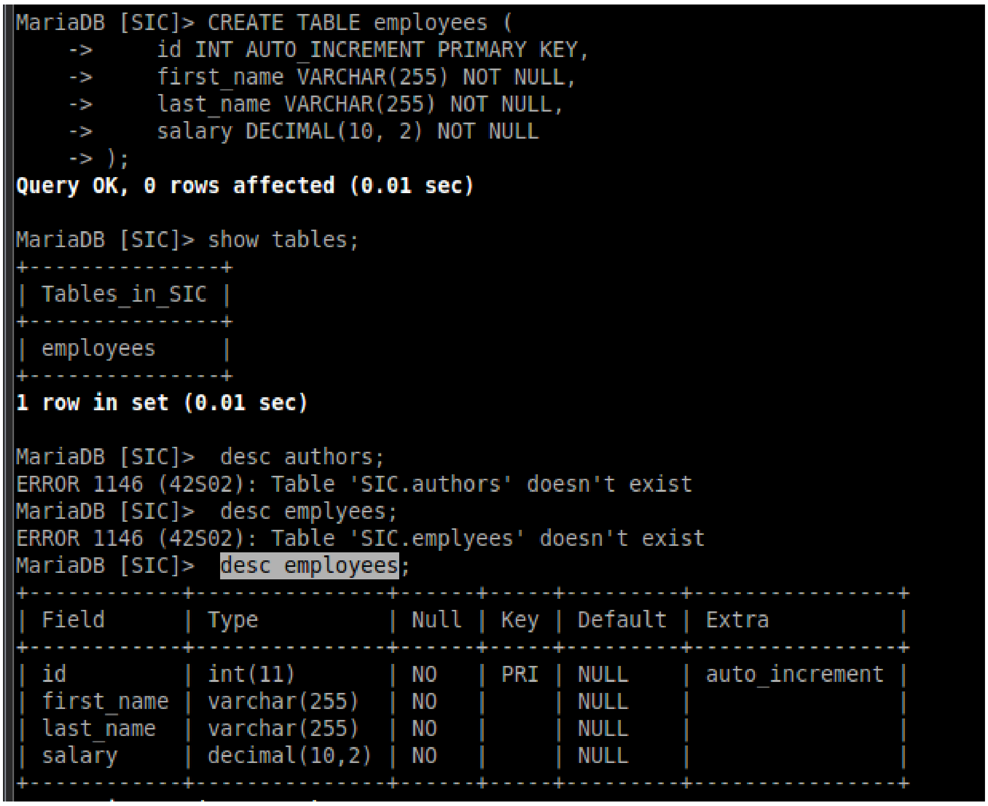
Database: The SIC database in MySQL contains an employees table with the following structure:

1. Create database “SIC”



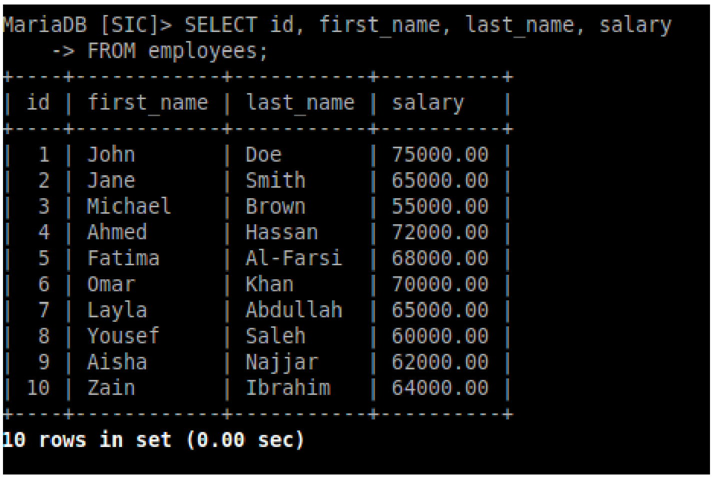
```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| hive               |
| hue                |
| labs               |
| mysql              |
| performance_schema |
+--------------------+
6 rows in set (0.01 sec)

MariaDB [(none)]> creat database SIC;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syn
tax to use near 'creat database SIC' at line 1
MariaDB [(none)]> create database SIC;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| SIC                |
| hive               |
| hue                |
| labs               |
| mysql              |
| performance_schema |
+--------------------+
7 rows in set (0.00 sec)
```

2. Create table “employees”



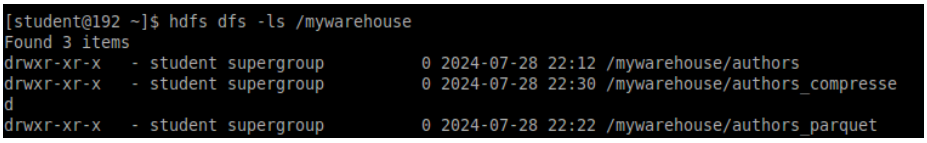
```
MariaDB [SIC]> CREATE TABLE employees (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     first_name VARCHAR(255) NOT NULL,
    ->     last_name VARCHAR(255) NOT NULL,
    ->     salary DECIMAL(10, 2) NOT NULL
    -> );
Query OK, 0 rows affected (0.01 sec)

MariaDB [SIC]> show tables;
+---------------+
| Tables_in_SIC |
+---------------+
| employees     |
+---------------+
1 row in set (0.01 sec)

MariaDB [SIC]>  desc authors;
ERROR 1146 (42S02): Table 'SIC.authors' doesn't exist
MariaDB [SIC]>  desc emplyees;
ERROR 1146 (42S02): Table 'SIC.emplyees' doesn't exist
MariaDB [SIC]>  desc employees;
+------------+---------------+------+-----+---------+----------------+
| Field      | Type          | Null | Key | Default | Extra          |
+------------+---------------+------+-----+---------+----------------+
| id         | int(11)       | NO   | PRI | NULL    | auto_increment |
| first_name | varchar(255)  | NO   |     | NULL    |                |
| last_name  | varchar(255)  | NO   |     | NULL    |                |
| salary     | decimal(10,2) | NO   |     | NULL    |                |
+------------+---------------+------+-----+---------+----------------+
```

3. Data samples



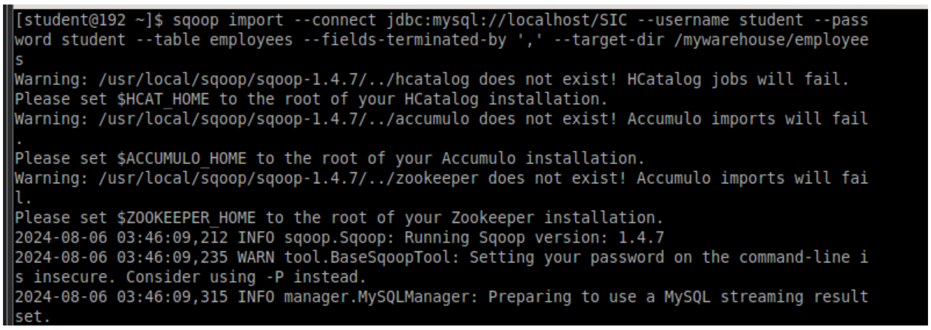
```
MariaDB [SIC]> SELECT id, first_name, last_name, salary
    -> FROM employees;
+----+------------+-----------+----------+
| id | first_name | last_name | salary   |
+----+------------+-----------+----------+
|  1 | John       | Doe       | 75000.00 |
|  2 | Jane       | Smith     | 65000.00 |
|  3 | Michael    | Brown     | 55000.00 |
|  4 | Ahmed      | Hassan    | 72000.00 |
|  5 | Fatima     | Al-Farsi  | 68000.00 |
|  6 | Omar       | Khan      | 70000.00 |
|  7 | Layla      | Abdullah  | 65000.00 |
|  8 | Yousef     | Saleh     | 60000.00 |
|  9 | Aisha      | Najjar    | 62000.00 |
| 10 | Zain       | Ibrahim   | 64000.00 |
+----+------------+-----------+----------+
10 rows in set (0.00 sec)
```

4. Data on my warehouse before importing



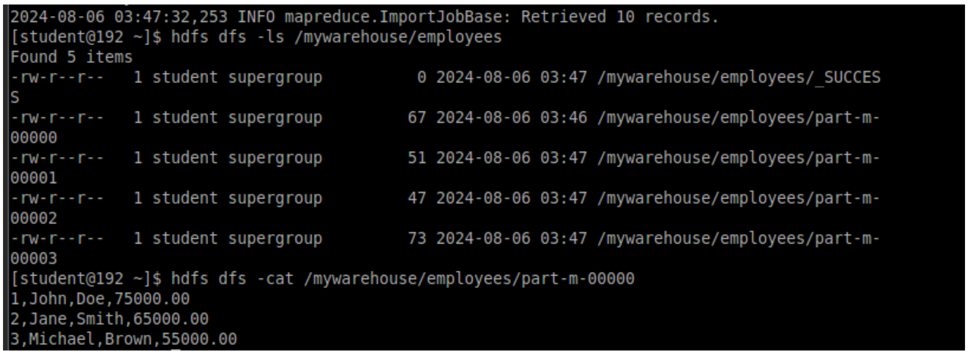
```
[student@192 ~]$ hdfs dfs -ls /mywarehouse
Found 3 items
drwxr-xr-x   - student supergroup          0 2024-07-28 22:12 /mywarehouse/authors
drwxr-xr-x   - student supergroup          0 2024-07-28 22:30 /mywarehouse/authors_compresse
d
drwxr-xr-x   - student supergroup          0 2024-07-28 22:22 /mywarehouse/authors_parquet
```

5. Import data to HDFS



```
[student@192 ~]$ sqoop import --connect jdbc:mysql://localhost/SIC --username student --pass
word student --table employees --fields-terminated-by ',' --target-dir /mywarehouse/employee
s
Warning: /usr/local/sqoop/sqoop-1.4.7/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/sqoop-1.4.7/../accumulo does not exist! Accumulo imports will fail
.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/sqoop-1.4.7/../zookeeper does not exist! Accumulo imports will fai
l.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2024-08-06 03:46:09,212 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2024-08-06 03:46:09,235 WARN tool.BaseSqoopTool: Setting your password on the command-line i
s insecure. Consider using -P instead.
2024-08-06 03:46:09,315 INFO manager.MySQLManager: Preparing to use a MySQL streaming result
set.
```

6. Data transferred to HDFS



```
2024-08-06 03:47:32,253 INFO mapreduce.ImportJobBase: Retrieved 10 records.
[student@192 ~]$ hdfs dfs -ls /mywarehouse/employees
Found 5 items
-rw-r--r--   1 student supergroup          0 2024-08-06 03:47 /mywarehouse/employees/_SUCCES
S
-rw-r--r--   1 student supergroup         67 2024-08-06 03:46 /mywarehouse/employees/part-m-
00000
-rw-r--r--   1 student supergroup         51 2024-08-06 03:47 /mywarehouse/employees/part-m-
00001
-rw-r--r--   1 student supergroup         47 2024-08-06 03:47 /mywarehouse/employees/part-m-
00002
-rw-r--r--   1 student supergroup         73 2024-08-06 03:47 /mywarehouse/employees/part-m-
00003
[student@192 ~]$ hdfs dfs -cat /mywarehouse/employees/part-m-00000
1,John,Doe,75000.00
2,Jane,Smith,65000.00
3,Michael,Brown,55000.00
```

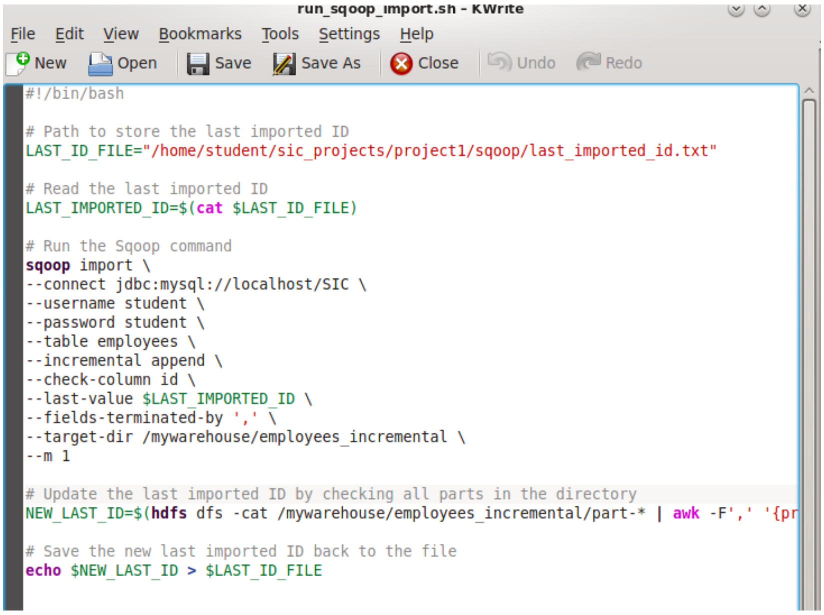
7. Incremental Import Using Append Mode

   This approach will allow you to import only the new rows based on the unique id column, ensuring that you don't re-import data that has already been imported.

- Create a File to Track the Last Imported ID

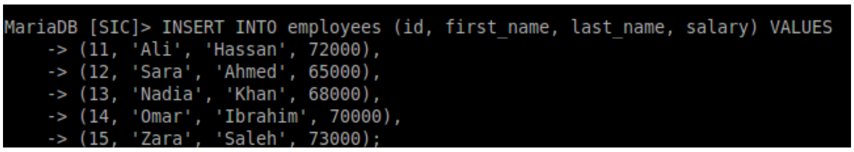
  "/home/student/sic_projects/project1/sqoop/last_imported_id.txt"

- Create a Shell Script


run_sqoop_import.sh - KWrite
File Edit View Bookmarks Tools Settings Help
New Open Save Save As Close Undo Redo

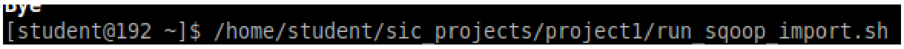
```
#!/bin/bash

# Path to store the last imported ID
LAST_ID_FILE="/home/student/sic_projects/project1/sqoop/last_imported_id.txt"

# Read the last imported ID
LAST_IMPORTED_ID=$(cat $LAST_ID_FILE)

# Run the Sqoop command
sqoop import \
--connect jdbc:mysql://localhost/SIC \
--username student \
--password student \
--table employees \
--incremental append \
--check-column id \
--last-value $LAST_IMPORTED_ID \
--fields-terminated-by ',' \
--target-dir /mywarehouse/employees_incremental \
--m 1

# Update the last imported ID by checking all parts in the directory
NEW_LAST_ID=$(hdfs dfs -cat /mywarehouse/employees_incremental/part-* | awk -F',' '{pr

# Save the new last imported ID back to the file
echo $NEW_LAST_ID > $LAST_ID_FILE
```

- Insert more data

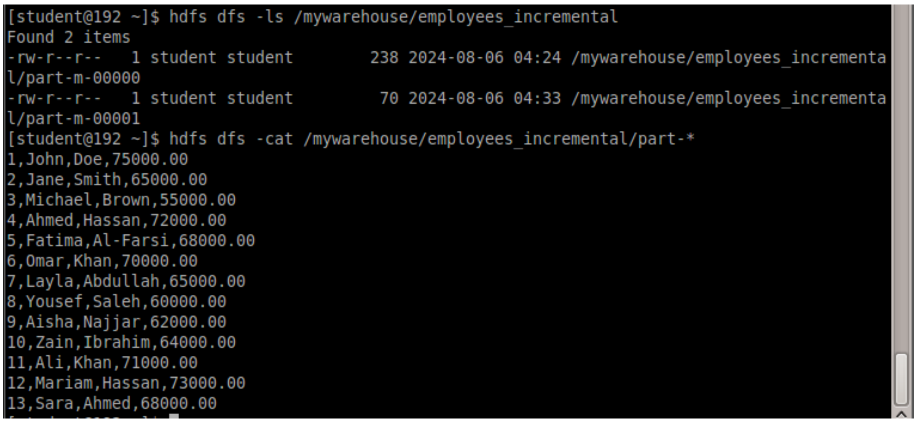


```
MariaDB [SIC]> INSERT INTO employees (id, first_name, last_name, salary) VALUES
    -> (11, 'Ali', 'Hassan', 72000),
    -> (12, 'Sara', 'Ahmed', 65000),
    -> (13, 'Nadia', 'Khan', 68000),
    -> (14, 'Omar', 'Ibrahim', 70000),
    -> (15, 'Zara', 'Saleh', 73000);
```

- Run script



```
Bye
[student@192 ~]$ /home/student/sic_projects/project1/run_sqoop_import.sh
```



```
[student@192 ~]$ hdfs dfs -ls /mywarehouse/employees_incremental
Found 2 items
-rw-r--r--   1 student student        238 2024-08-06 04:24 /mywarehouse/employees_incrementa
l/part-m-00000
-rw-r--r--   1 student student         70 2024-08-06 04:33 /mywarehouse/employees_incrementa
l/part-m-00001
[student@192 ~]$ hdfs dfs -cat /mywarehouse/employees_incremental/part-*
1,John,Doe,75000.00
2,Jane,Smith,65000.00
3,Michael,Brown,55000.00
4,Ahmed,Hassan,72000.00
5,Fatima,Al-Farsi,68000.00
6,Omar,Khan,70000.00
7,Layla,Abdullah,65000.00
8,Yousef,Saleh,60000.00
9,Aisha,Najjar,62000.00
10,Zain,Ibrahim,64000.00
11,Ali,Khan,71000.00
12,Mariam,Hassan,73000.00
13,Sara,Ahmed,68000.00
```

3 REAL-TIME DATA INGESTION WITH APACHE FLUME AND KAFKA

3.1 INTRODUCTION TO REAL-TIME DATA

3.1.1 Real-Time Data: Refers to the continuous processing of data as it is generated, allowing immediate analysis and action.

3.1.2 **Importance:** Enables timely insights, enhanced decision-making, and improved customer experiences.

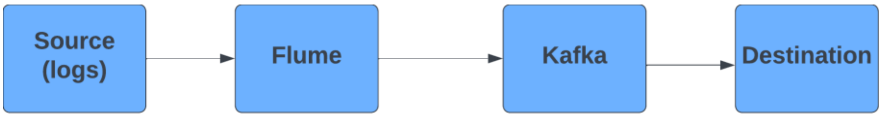
3.1.3 Technologies:

- **Apache Flume**: Collects and moves large amounts of log data to various destinations.
- **Apache Kafka**: A distributed streaming platform for building real-time data pipelines and streaming applications.

3.2 ENVIRONMENT SETUP

- **Apache Flume**: Download, extract, and set environment variables.
- **Apache Kafka**: Download, extract, and start Zookeeper and Kafka server.

3.3 FLOW CHART: DATA FLOW OVERVIEW



3.4 DATA LOGS

In the context of setting up a real-time data ingestion pipeline with Apache Flume and Kafka, creating a logs directory and a log file is a crucial step. This directory and file serve as the source of data that Flume will monitor and forward to Kafka

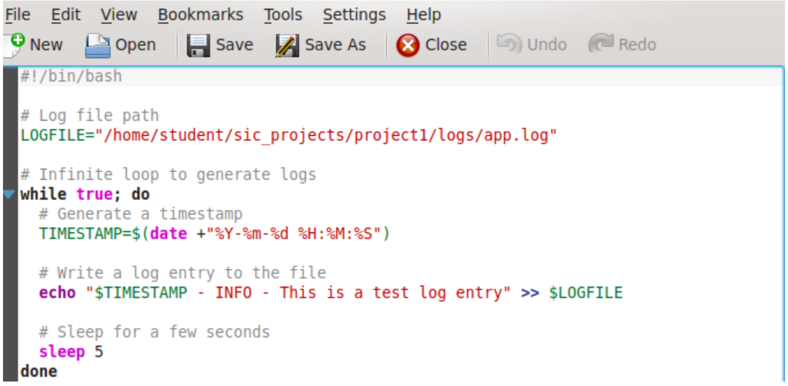
Logs Directory: To store log files that Apache Flume will monitor for real-time

Command: “mkdir -p /home/student/sic_projects/project1/logs”

Log File (app.log): To serve as the source file where log entries are written, which Flume will capture and forward to Kafka.

Command: “touch /home/student/sic_projects/project1/logs/app.log”

Create a Log Generation Script:


File Edit View Bookmarks Tools Settings Help
New Open Save Save As Close Undo Redo

```
#!/bin/bash

# Log file path
LOGFILE="/home/student/sic_projects/project1/logs/app.log"

# Infinite loop to generate logs
while true; do
  # Generate a timestamp
  TIMESTAMP=$(date +"%Y-%m-%d %H:%M:%S")

  # Write a log entry to the file
  echo "$TIMESTAMP - INFO - This is a test log entry" >> $LOGFILE

  # Sleep for a few seconds
  sleep 5
done
```

3.5 CONFIGURE FLUME TO CAPTURE LOG DATA

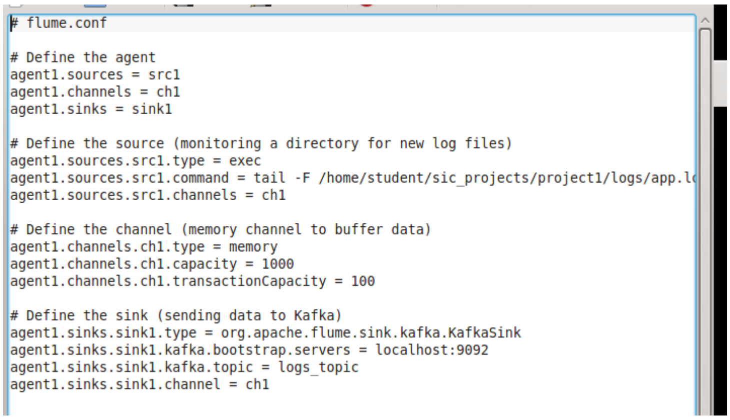
- **Create the Configuration Directory:**

  “mkdir -p /home/student/sic_projects/project1/flume”

- **Create and Edit the flume.conf File:**

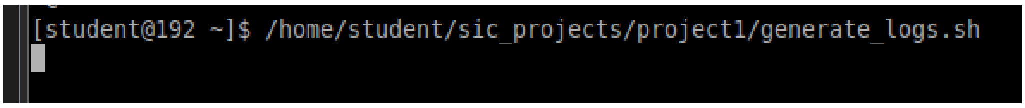
  “vi /home/student/sic_projects/project1/flume/flume.conf”

- **Flume agent**

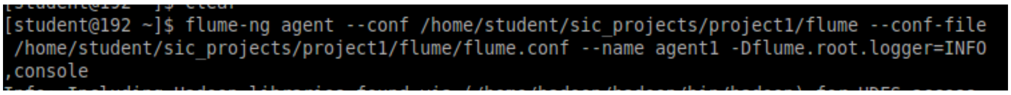


```
# flume.conf

# Define the agent
agent1.sources = src1
agent1.channels = ch1
agent1.sinks = sink1

# Define the source (monitoring a directory for new log files)
agent1.sources.src1.type = exec
agent1.sources.src1.command = tail -F /home/student/sic_projects/project1/logs/app.l
agent1.sources.src1.channels = ch1

# Define the channel (memory channel to buffer data)
agent1.channels.ch1.type = memory
agent1.channels.ch1.capacity = 1000
agent1.channels.ch1.transactionCapacity = 100

# Define the sink (sending data to Kafka)
agent1.sinks.sink1.type = org.apache.flume.sink.kafka.KafkaSink
agent1.sinks.sink1.kafka.bootstrap.servers = localhost:9092
agent1.sinks.sink1.kafka.topic = logs_topic
agent1.sinks.sink1.channel = ch1
```

- **Start Flume and the Log Generation Script**



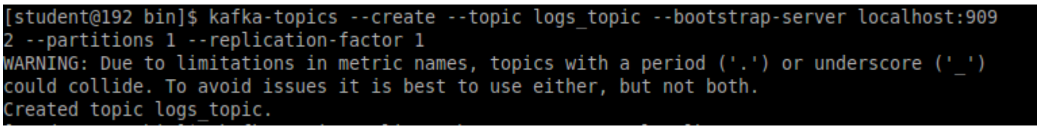
```
[student@192 ~]$ /home/student/sic_projects/project1/generate_logs.sh
```



```
[student@192 ~]$ clear
[student@192 ~]$ flume-ng agent --conf /home/student/sic_projects/project1/flume --conf-file
 /home/student/sic_projects/project1/flume/flume.conf --name agent1 -Dflume.root.logger=INFO
,console
Info: Including Hadoop libraries found via (/home/hadoop/hadoop/bin/hadoop) for HDFS access
```

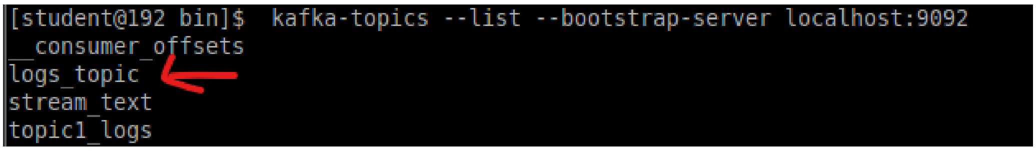
3.6 CONFIGURE KAFKA TO STORE AND MANAGE THE INCOMING LOG DATA

- Create a Kafka Topic



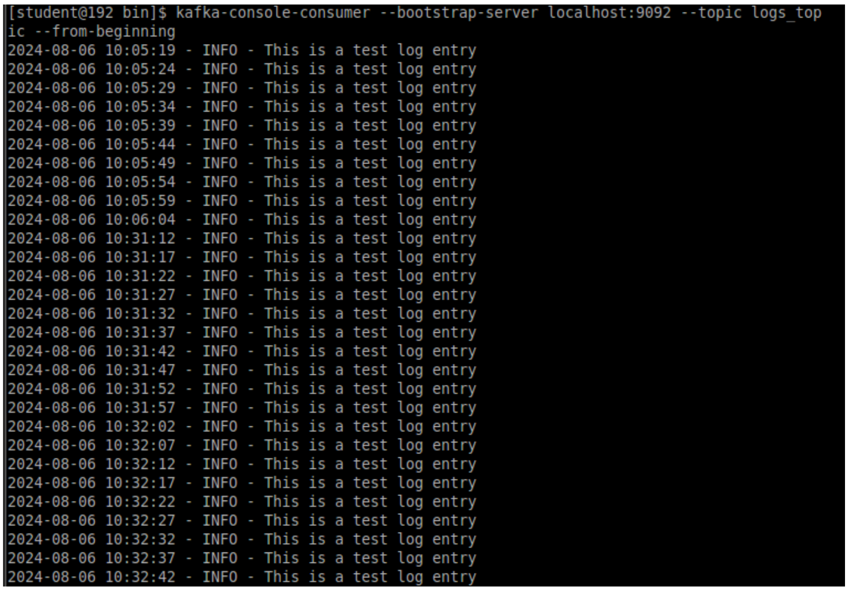
```
[student@192 bin]$ kafka-topics --create --topic logs_topic --bootstrap-server localhost:909
2 --partitions 1 --replication-factor 1
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_')
could collide. To avoid issues it is best to use either, but not both.
Created topic logs_topic.
```

- Verify the Topic Creation



```
[student@192 bin]$  kafka-topics --list --bootstrap-server localhost:9092
__consumer_offsets
logs_topic
stream_text
topic1_logs
```

- Consume Data from Kafka to Verify Setup



```
[student@192 bin]$ kafka-console-consumer --bootstrap-server localhost:9092 --topic logs_top
ic --from-beginning
2024-08-06 10:05:19 - INFO - This is a test log entry
2024-08-06 10:05:24 - INFO - This is a test log entry
2024-08-06 10:05:29 - INFO - This is a test log entry
2024-08-06 10:05:34 - INFO - This is a test log entry
2024-08-06 10:05:39 - INFO - This is a test log entry
2024-08-06 10:05:44 - INFO - This is a test log entry
2024-08-06 10:05:49 - INFO - This is a test log entry
2024-08-06 10:05:54 - INFO - This is a test log entry
2024-08-06 10:05:59 - INFO - This is a test log entry
2024-08-06 10:06:04 - INFO - This is a test log entry
2024-08-06 10:31:12 - INFO - This is a test log entry
2024-08-06 10:31:17 - INFO - This is a test log entry
2024-08-06 10:31:22 - INFO - This is a test log entry
2024-08-06 10:31:27 - INFO - This is a test log entry
2024-08-06 10:31:32 - INFO - This is a test log entry
2024-08-06 10:31:37 - INFO - This is a test log entry
2024-08-06 10:31:42 - INFO - This is a test log entry
2024-08-06 10:31:47 - INFO - This is a test log entry
2024-08-06 10:31:52 - INFO - This is a test log entry
2024-08-06 10:31:57 - INFO - This is a test log entry
2024-08-06 10:32:02 - INFO - This is a test log entry
2024-08-06 10:32:07 - INFO - This is a test log entry
2024-08-06 10:32:12 - INFO - This is a test log entry
2024-08-06 10:32:17 - INFO - This is a test log entry
2024-08-06 10:32:22 - INFO - This is a test log entry
2024-08-06 10:32:27 - INFO - This is a test log entry
2024-08-06 10:32:32 - INFO - This is a test log entry
2024-08-06 10:32:37 - INFO - This is a test log entry
2024-08-06 10:32:42 - INFO - This is a test log entry
```