High Level Design Document

Movie analytics Using PySpark

Table of Contents

1. **Introduction……………………………………………….1**

1.1 Architecture ………………………………………….1

2. **Implementation ……………………………………...2**

2.1 Loading Data to Hadoop ……………………………2

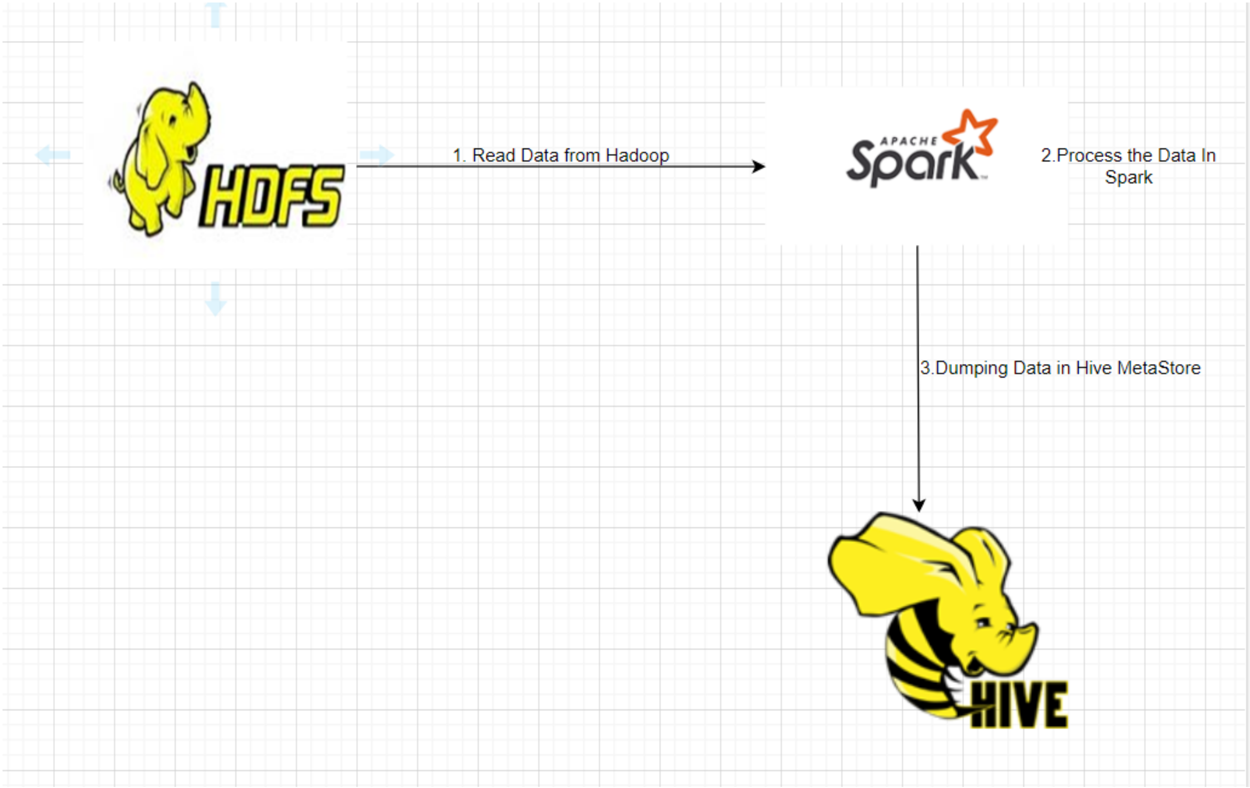
2.2 Running a Spark Job ………...……………………...3

3 Hive Metastore……………………..…………………...4

1. Introduction

The aim of this project is to perform analytical queries and extract meaningful insights from Movies_analytics Dataset. This Dataset contains three datafiles(users.dat , ratings.dat and movies.dat). We store data in HDFS and process the data using Pyspark.

1.1 Architecture



1.2 Explanation

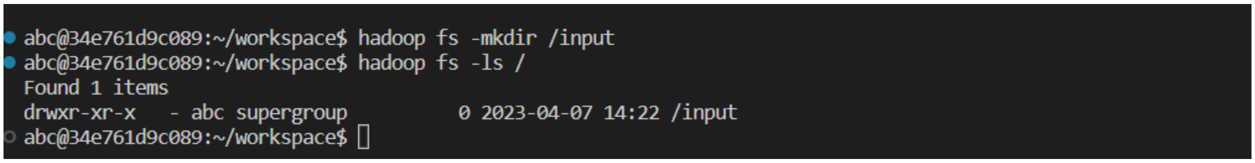
The process starts with downloading the Dataset from **https://grouplens.org/datasets/movielens/1m/** . In the dataset we have three datafiles(i.e. users.dat , ratings.dat , movies.dat). We store the data in Hadoop later , processes using spark. We performed some analytical queries and store in Hadoop and table schema is stored in hive metastore.

2. Implementation

2.1 Loading Data in hadoop

The Download data needs to be stored in Hadoop. The below command will create a directory in HDFS.

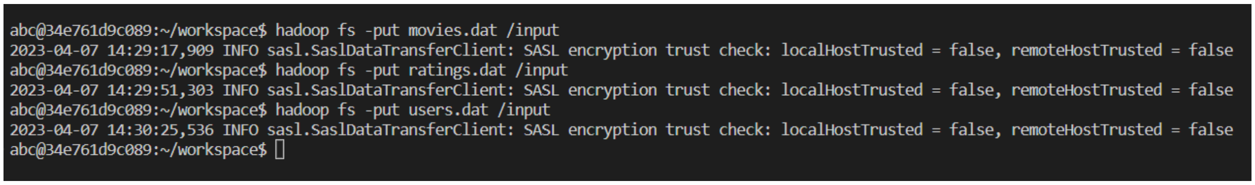
Hadoop fs -mkdir /dir_name



```
abc@34e761d9c089:~/workspace$ hadoop fs -mkdir /input
abc@34e761d9c089:~/workspace$ hadoop fs -ls /
Found 1 items
drwxr-xr-x   - abc supergroup          0 2023-04-07 14:22 /input
abc@34e761d9c089:~/workspace$ 
```

The below command will put the data from local to HDFS.

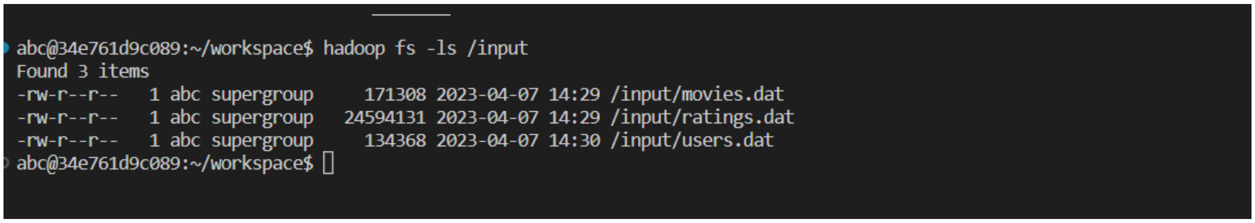
Hadoop fs -put /path_of_local_file /path_of_HDFS_file



```
abc@34e761d9c089:~/workspace$ hadoop fs -put movies.dat /input
2023-04-07 14:29:17,909 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
abc@34e761d9c089:~/workspace$ hadoop fs -put ratings.dat /input
2023-04-07 14:29:51,303 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
abc@34e761d9c089:~/workspace$ hadoop fs -put users.dat /input
2023-04-07 14:30:25,536 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
abc@34e761d9c089:~/workspace$ 
```

To list the files in HDFS directory , we need to use below command

Hadoop fs -ls /dir_name



```
abc@34e761d9c089:~/workspace$ hadoop fs -ls /input
Found 3 items
-rw-r--r--   1 abc supergroup     171308 2023-04-07 14:29 /input/movies.dat
-rw-r--r--   1 abc supergroup   24594131 2023-04-07 14:29 /input/ratings.dat
-rw-r--r--   1 abc supergroup     134368 2023-04-07 14:30 /input/users.dat
abc@34e761d9c089:~/workspace$ 
```

2.2 Running a Spark Job

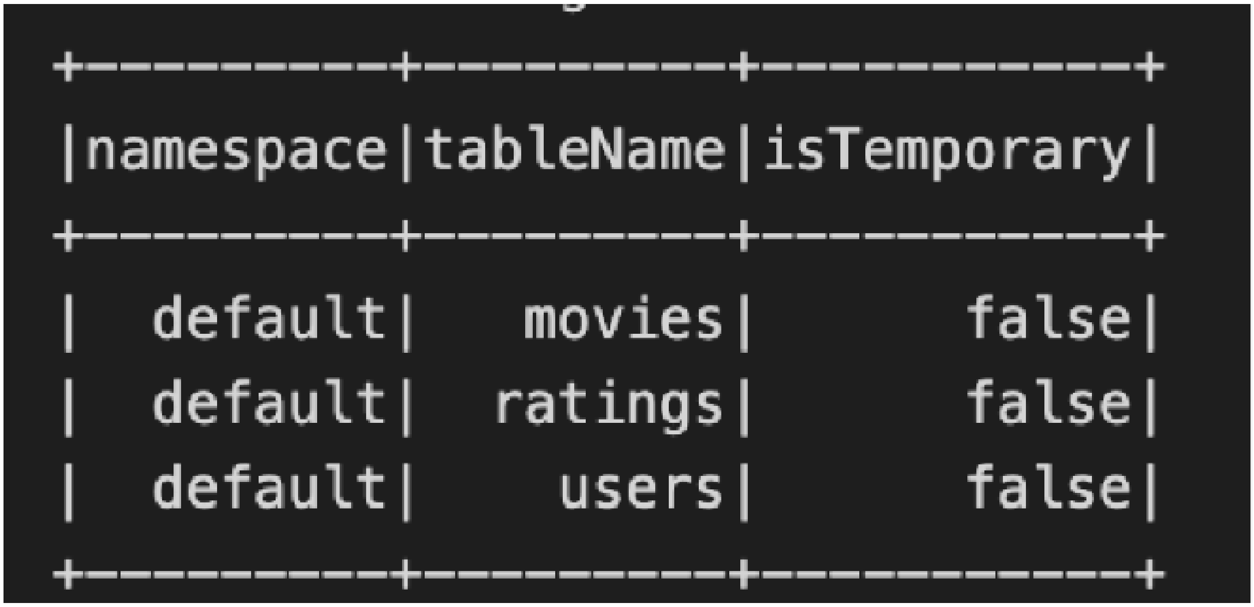
The spark code is written in the main.py file in the folder. The below command is used to run the spark job.

spark-submit main.py &> output.txt

- Spark-submit (To submit the spark Application)
- main.py (Spark code is written in this file)
- Output.txt (Save output in separate file)

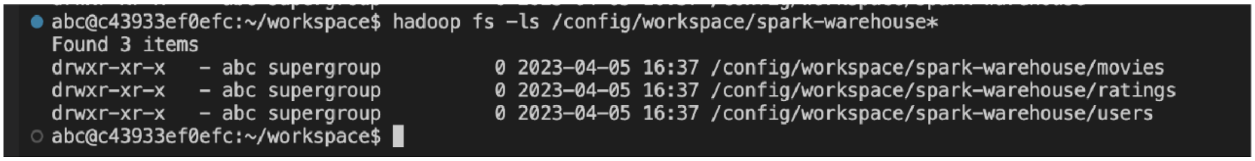
Hive metastore

The processed data is stored in hadoop and table schema is stored in hive metastore.
The below image shows spark storing table schema in hive metastore (i.e. , derby)



```
+---------+---------+-----------+
|namespace|tableName|isTemporary|
+---------+---------+-----------+
|  default|   movies|      false|
|  default|  ratings|      false|
|  default|    users|      false|
+---------+---------+-----------+
```

The original data is stored in Hadoop as parquet files.



```
● abc@c43933ef0efc:~/workspace$ hadoop fs -ls /config/workspace/spark-warehouse*
  Found 3 items
  drwxr-xr-x   - abc supergroup          0 2023-04-05 16:37 /config/workspace/spark-warehouse/movies
  drwxr-xr-x   - abc supergroup          0 2023-04-05 16:37 /config/workspace/spark-warehouse/ratings
  drwxr-xr-x   - abc supergroup          0 2023-04-05 16:37 /config/workspace/spark-warehouse/users
○ abc@c43933ef0efc:~/workspace$ █
```