Logic For First Submission

TASK -1: Write a job to consume clickstream data from Kafka and ingest to Hadoop

1. create a python file (spark_kafka_to_local.py) using vi command. This file will contain the code which will ingest the relevant data from Kafka into hadoop

   vi spark_kafka_to_local.py

2. Explaining the code in spark_kafka_to_local.py

   # Importing the modules

```
import sys
import os
os.environ["PYSPARK_PYTHON"] = "/opt/cloudera/parcels/Anaconda/bin/python"
os.environ["JAVA_HOME"] = "/usr/java/jdk1.8.0_232-cloudera/jre"
os.environ["SPARK_HOME"]="/opt/cloudera/parcels/SPARK2-2.3.0.cloudera2-
1.cdh5.13.3.p0.316101/lib/spark2/"
os.environ["PYLIB"] = os.environ["SPARK_HOME"] + "/python/lib"
sys.path.insert(0, os.environ["PYLIB"] +"/py4j-0.10.6-src.zip")
sys.path.insert(0, os.environ["PYLIB"] +"/pyspark.zip")


from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *
from pyspark.sql.functions import from_json
from pyspark.sql.window import Window


# Initializing Spark Session

spark = SparkSession \
    .builder \
    .master("local") \
    .appName("CapstoneProject") \
    .getOrCreate()
spark.sparkContext.setLogLevel('ERROR')

# Creating Dataframe from Kafka Data
```

```
clickstream_df = spark \
    .readStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers", "18.211.252.152:9092") \
    .option("subscribe", "de-capstone5") \
    .option("startingOffsets", "earliest") \
    .load()

clickstream_df.printSchema()

# Transform dataframe by dropping few columns and changing value column data type

clickstream_df = clickstream_df \
    .withColumn('value_str',
clickstream_df['value'].cast('string').alias('key_str')).drop('value') \
    .drop('key','topic','partition','offset','timestamp','timestampType')

# writing the dataframe to local file directory and keep it running until terminated

clickstream_df.writeStream \
    .outputMode("append") \
    .format("json") \
    .option("truncate","false") \
    .option("path", "clickstream_data") \
    .option("checkpointLocation","clickstream_checkpoint") \
    .start() \
    .awaitTermination()
```

3. Run Spark Submit command, to ingest the relevant data from Kafka into hadoop.

```
spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5
spark_kafka_to_local.py 18.211.252.152 9092 de-capstone5
```

```
[hadoop@ip-172-31-70-125 ~]$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 spark_kafka_to_local.py 18.211.252.152 9092 de-capstone5
Ivy Default Cache set to: /home/hadoop/.ivy2/cache
The jars for the packages stored in: /home/hadoop/.ivy2/jars
:: loading settings :: url = jar:file:/usr/lib/spark/jars/ivy-2.4.0.jar!/org/apache/ivy/core/settings/ivysettings.xml
org.apache.spark#spark-sql-kafka-0-10_2.11 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-efc7cc12-9ed5-42ad-8027-bbef3c4ef515;1.0
        confs: [default]
        found org.apache.spark#spark-sql-kafka-0-10_2.11;2.4.5 in central
        found org.apache.kafka#kafka-clients;2.0.0 in central
        found org.lz4#lz4-java;1.4.0 in central
        found org.xerial.snappy#snappy-java;1.1.7.3 in central
        found org.slf4j#slf4j-api;1.7.16 in central
        found org.spark-project.spark#unused;1.0.0 in central
downloading https://repo1.maven.org/maven2/org/apache/spark/spark-sql-kafka-0-10_2.11/2.4.5/spark-sql-kafka-0-10_2.11-2.4.5.jar ...
        [SUCCESSFUL ] org.apache.spark#spark-sql-kafka-0-10_2.11;2.4.5!spark-sql-kafka-0-10_2.11.jar (29ms)
downloading https://repo1.maven.org/maven2/org/apache/kafka/kafka-clients/2.0.0/kafka-clients-2.0.0.jar ...
        [SUCCESSFUL ] org.apache.kafka#kafka-clients;2.0.0!kafka-clients.jar (74ms)
downloading https://repo1.maven.org/maven2/org/spark-project/spark/unused/1.0.0/unused-1.0.0.jar ...
        [SUCCESSFUL ] org.spark-project.spark#unused;1.0.0!unused.jar (4ms)
downloading https://repo1.maven.org/maven2/org/lz4/lz4-java/1.4.0/lz4-java-1.4.0.jar ...
        [SUCCESSFUL ] org.lz4#lz4-java;1.4.0!lz4-java.jar (15ms)
downloading https://repo1.maven.org/maven2/org/xerial/snappy/snappy-java/1.1.7.3/snappy-java-1.1.7.3.jar ...
        [SUCCESSFUL ] org.xerial.snappy#snappy-java;1.1.7.3!snappy-java.jar(bundle) (84ms)
downloading https://repo1.maven.org/maven2/org/slf4j/slf4j-api/1.7.16/slf4j-api-1.7.16.jar ...
        [SUCCESSFUL ] org.slf4j#slf4j-api;1.7.16!slf4j-api.jar (5ms)
:: resolution report :: resolve 1272ms :: artifacts dl 218ms
        :: modules in use:
        org.apache.kafka#kafka-clients;2.0.0 from central in [default]
        org.apache.spark#spark-sql-kafka-0-10_2.11;2.4.5 from central in [default]
        org.lz4#lz4-java;1.4.0 from central in [default]
        org.slf4j#slf4j-api;1.7.16 from central in [default]
        org.spark-project.spark#unused;1.0.0 from central in [default]
        org.xerial.snappy#snappy-java;1.1.7.3 from central in [default]
        ---------------------------------------------------------------------
        |                  |            modules            ||   artifacts   |
        |       conf       | number| search|dwnlded|evicted|| number|dwnlded|
        ---------------------------------------------------------------------
        |      default     |   6   |   6   |   6   |   0   ||   6   |   6   |
        ---------------------------------------------------------------------
:: retrieving :: org.apache.spark#spark-submit-parent-efc7cc12-9ed5-42ad-8027-bbef3c4ef515
        confs: [default]
        6 artifacts copied, 0 already retrieved (4749kB/17ms)
24/04/26 19:02:21 INFO SparkContext: Running Spark version 2.4.5-amzn-0
```

```
[hadoop@ip-172-31-70-125 ~]$ hadoop fs -ls
Found 2 items
drwxr-xr-x   - hadoop hadoop          0 2024-04-26 19:02 clickstream_checkpoint
drwxr-xr-x   - hadoop hadoop          0 2024-04-26 19:05 clickstream_data
[hadoop@ip-172-31-70-125 ~]$ hadoop fs -ls clickstream_data
Found 119 items
drwxr-xr-x   - hadoop hadoop          0 2024-04-26 19:05 clickstream_data/_spark_metadata
-rw-r--r--   1 hadoop hadoop      71318 2024-04-26 19:04 clickstream_data/part-00000-0a33d63f-7bc6-462c-9a2a-3bf9c6130041-c000.json
-rw-r--r--   1 hadoop hadoop     261191 2024-04-26 19:04 clickstream_data/part-00000-0b5cda10-cc0f-4c43-858a-c24567585f8c-c000.json
-rw-r--r--   1 hadoop hadoop     342657 2024-04-26 19:04 clickstream_data/part-00000-0dc80654-3de0-4498-9f70-aa8b49ebdd0b-c000.json
-rw-r--r--   1 hadoop hadoop     716242 2024-04-26 19:04 clickstream_data/part-00000-0e0b9a97-3d32-43d1-940d-ba90690a6d5b-c000.json
-rw-r--r--   1 hadoop hadoop      59583 2024-04-26 19:04 clickstream_data/part-00000-13094622-db98-4729-8b23-ca567fba830c-c000.json
-rw-r--r--   1 hadoop hadoop     346359 2024-04-26 19:04 clickstream_data/part-00000-136aa6f6-473c-4056-8f6d-6bcf0058c639-c000.json
-rw-r--r--   1 hadoop hadoop      82369 2024-04-26 19:04 clickstream_data/part-00000-165ec041-bd71-4f17-b971-ea23c77a8af6-c000.json
-rw-r--r--   1 hadoop hadoop     238378 2024-04-26 19:04 clickstream_data/part-00000-1707dc6f-b70e-4db0-9e35-6824925c6a18-c000.json
```

```
[hadoop@ip-172-31-70-125 ~]$ hadoop fs -cat clickstream_data/part-00000-d8df90ac-1ec9-4014-b633-a20d66f44a32-c000.json
2453
[hadoop@ip-172-31-70-125 ~]$
```

4. Create another python file (spark_local_flatten.py) using vi command. This file will clean the loaded Kafka data to a more structured format

   vi spark_local_flatten.py

5. Explaining the code in spark_local_flatten.py

```
# Importing the modules

from pyspark.sql import SparkSession
from pyspark.sql.functions import *

# Initializing Spark Session

spark = SparkSession.builder \
    .master("local") \
    .appName("CapstoneProject") \
    .getOrCreate()

# Reading json data into dataframe

clickstream_df = spark.read.json('clickstream_data/part-00000-d8df90ac-1ec9-4014-
b633-a20d66f44a32-c000.json')


# Extrating columns from json value in dataframe and create new dataframe with new
cloumns

clickstream_df = clickstream_df.select(\
        get_json_object(clickstream_df["value_str"],"$.customer_id").alias("customer_id"),\
        get_json_object(clickstream_df["value_str"],"$.app_version").alias("app_version"),\
        get_json_object(clickstream_df["value_str"],"$.OS_version").alias("OS_version"),\
        get_json_object(clickstream_df["value_str"],"$.lat").alias("lat"),\
        get_json_object(clickstream_df["value_str"],"$.lon").alias("lon"),\
        get_json_object(clickstream_df["value_str"],"$.page_id").alias("page_id"),\
        get_json_object(clickstream_df["value_str"],"$.button_id").alias("button_id"),\
        get_json_object(clickstream_df["value_str"],"$.is_button_click").alias("is_button_click"),\
        get_json_object(clickstream_df["value_str"],"$.is_page_view").alias("is_page_view"),\
        get_json_object(clickstream_df["value_str"],"$.is_scroll_up").alias("is_scroll_up"),\
        get_json_object(clickstream_df["value_str"],"$.is_scroll_down").alias("is_scroll_down"),\
        get_json_object(clickstream_df["value_str"],"$.timestamp\n").alias("timestamp"),\
        )


# Printing the Dataframe schema
```

print(clickstream_df.schema)

Printing 10 records from the dataframe

clickstream_df.show(10)

Saving the dataframe to csv file with headers in local file directory

clickstream_df.coalesce(1).write.format('com.databricks.spark.csv').mode('overwrite').save('clickstream_data_flatten/', header = 'true')

6. Run Spark Submit command

   spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 spark_local_flatten.py

```
[hadoop@ip-172-31-70-125 ~]$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 spark_local_flatten.py
Ivy Default Cache set to: /home/hadoop/.ivy2/cache
The jars for the packages stored in: /home/hadoop/.ivy2/jars
:: loading settings :: url = jar:file:/usr/lib/spark/jars/ivy-2.4.0.jar!/org/apache/ivy/core/settings/ivysettings.xml
org.apache.spark#spark-sql-kafka-0-10_2.11 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-d16b5d34-20e9-49aa-ac3b-f443eff7679f;1.0
        confs: [default]
        found org.apache.spark#spark-sql-kafka-0-10_2.11;2.4.5 in central
        found org.apache.kafka#kafka-clients;2.0.0 in central
        found org.lz4#lz4-java;1.4.0 in central
        found org.xerial.snappy#snappy-java;1.1.7.3 in central
        found org.slf4j#slf4j-api;1.7.16 in central
        found org.spark-project.spark#unused;1.0.0 in central
:: resolution report :: resolve 419ms :: artifacts dl 13ms
        :: modules in use:
        org.apache.kafka#kafka-clients;2.0.0 from central in [default]
        org.apache.spark#spark-sql-kafka-0-10_2.11;2.4.5 from central in [default]
        org.lz4#lz4-java;1.4.0 from central in [default]
        org.slf4j#slf4j-api;1.7.16 from central in [default]
```

7. Make a directory using mkdir command

   hadoop fs -mkdir clickstream_data_flatten

8. Loading the data from local file system to hadoop file system

   hadoop fs- put ~/clickstream_data_flatten clickstream_data_flatten

9. Checking the data file in hadoop

   hadoop fs -ls clickstream_data_flatten

   hadoop fs -cat clickstream_data_flatten/part-00000-bb423f13-4963-4dd7-8afb-0630877df998-c000.csv | wc -l

```
[hadoop@ip-172-31-70-125 ~]$ hadoop fs -ls clickstream_data_flatten
Found 2 items
-rw-r--r--   1 hadoop hadoop          0 2024-04-26 19:12 clickstream_data_flatten/_SUCCESS
-rw-r--r--   1 hadoop hadoop     376742 2024-04-26 19:12 clickstream_data_flatten/part-00000-bb423f13-4963-4dd7-8afb-0630877df998-c000.csv
[hadoop@ip-172-31-70-125 ~]$ hadoop fs -cat clickstream_data_flatten/part-00000-bb423f13-4963-4dd7-8afb-0630877df998-c000.csv | wc -l
2454
[hadoop@ip-172-31-70-125 ~]$
```

Task 2: Write a script to ingest the relevant bookings data from AWS RDS to Hadoop

1. Setting up MySql connector with commands

a. Install the MySQL connector jar file:

wget https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz

```
[hadoop@ip-172-31-70-125 ~]$ wget https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz
--2024-04-26 19:16:35--  https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz
Resolving de-mysql-connector.s3.amazonaws.com (de-mysql-connector.s3.amazonaws.com)... 52.216.136.236, 3.5.22.74,
Connecting to de-mysql-connector.s3.amazonaws.com (de-mysql-connector.s3.amazonaws.com)|52.216.136.236|:443... co
HTTP request sent, awaiting response... 200 OK
Length: 4079310 (3.9M) [application/x-gzip]
Saving to: 'mysql-connector-java-8.0.25.tar.gz'

100%[=========================================================================================================================

2024-04-26 19:16:35 (37.7 MB/s) - 'mysql-connector-java-8.0.25.tar.gz' saved [4079310/4079310]

[hadoop@ip-172-31-70-125 ~]$
```

b. Extract the MySQL connector tar file

tar -xvf mysql-connector-java-8.0.25.tar.gz

```
[hadoop@ip-172-31-70-125 ~]$ tar -xvf mysql-connector-java-8.0.25.tar.gz
mysql-connector-java-8.0.25/
mysql-connector-java-8.0.25/src/
mysql-connector-java-8.0.25/src/build/
mysql-connector-java-8.0.25/src/build/java/
mysql-connector-java-8.0.25/src/build/java/documentation/
mysql-connector-java-8.0.25/src/build/java/instrumentation/
mysql-connector-java-8.0.25/src/build/misc/
mysql-connector-java-8.0.25/src/build/misc/debian.in/
mysql-connector-java-8.0.25/src/build/misc/debian.in/source/
mysql-connector-java-8.0.25/src/demo/
mysql-connector-java-8.0.25/src/demo/java/
mysql-connector-java-8.0.25/src/demo/java/demo/
mysql-connector-java-8.0.25/src/demo/java/demo/x/
mysql-connector-java-8.0.25/src/demo/java/demo/x/devapi/
mysql-connector-java-8.0.25/src/generated/
mysql-connector-java-8.0.25/src/generated/java/
mysql-connector-java-8.0.25/src/generated/java/com/
mysql-connector-java-8.0.25/src/generated/java/com/mysql/
mysql-connector-java-8.0.25/src/generated/java/com/mysql/cj/
mysql-connector-java-8.0.25/src/generated/java/com/mysql/cj/x/
mysql-connector-java-8.0.25/src/generated/java/com/mysql/cj/x/protobuf/
mysql-connector-java-8.0.25/src/legacy/
mysql-connector-java-8.0.25/src/legacy/java/
```

c. Getting into MySQL Connector directory

cd mysql-connector-java-8.0.25/

d. Copying it to the Sqoop library

sudo cp mysql-connector-java-8.0.25.jar /usr/lib/sqoop/lib/ library

```
[hadoop@ip-172-31-70-125 ~]$ cd mysql-connector-java-8.0.25/
[hadoop@ip-172-31-70-125 mysql-connector-java-8.0.25]$ sudo cp mysql-connector-java-8.0.25.jar /usr/lib/sqoop/lib
[hadoop@ip-172-31-70-125 mysql-connector-java-8.0.25]$ cd ..
[hadoop@ip-172-31-70-125 ~]$
```

2. Run the Sqoop import command to import data from AWS RDS to Hadoop

```
sqoop import \
--connect jdbc:mysql://upgraddetest.cyaielc9bmnf.us-east-1.rds.amazonaws.com/testdatabase \
--table bookings \
--username student --password STUDENT123 \
--null-string '\\N' --null-non-string '\\N' \
--target-dir bookings_data \
-m 1
```

```
[hadoop@ip-172-31-70-125 ~]$ sqoop import \
> --connect jdbc:mysql://upgraddetest.cyaielc9bmnf.us-east-1.rds.amazonaws.com/testdatabase \
> --table bookings \
> --username student --password STUDENT123 \
> --null-string '\\N' --null-non-string '\\N' \
> --target-dir bookings_data \
> -m 1
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
24/04/26 19:18:21 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerB
SLF4J: Found binding in [jar:file:/usr/share/aws/redshift/jdbc/redshift-jdbc42-1.2.37.1061.jar!/org/slf4j/im
SLF4J: Found binding in [jar:file:/usr/lib/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerB
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
24/04/26 19:18:21 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider u
24/04/26 19:18:21 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
24/04/26 19:18:21 INFO tool.CodeGenTool: Beginning code generation
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver
river class is generally unnecessary.
24/04/26 19:18:22 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `bookings` AS t LIMIT 1
24/04/26 19:18:22 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `bookings` AS t LIMIT 1
24/04/26 19:18:22 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-hadoop/compile/a0c93eb6f9c198f04964e03175b1989a/bookings.java uses or overrides a deprecate
Note: Recompile with -Xlint:deprecation for details.
24/04/26 19:18:27 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-hadoop/compile/a0c93eb6f9c198f04
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=89
                CPU time spent (ms)=3780
                Physical memory (bytes) snapshot=283848704
                Virtual memory (bytes) snapshot=3287175168
                Total committed heap usage (bytes)=243793920
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=165678
24/04/26 19:18:58 INFO mapreduce.ImportJobBase: Transferred 161.7949 KB in 29.2515 seconds (5.5312 KB/sec)
24/04/26 19:18:58 INFO mapreduce.ImportJobBase: Retrieved 1000 records.
[hadoop@ip-172-31-70-125 ~]$
```

3. Use this command to check the imported data

   hadoop fs -ls bookings_data

   hadoop fs -cat bookings_data/part-m-00000 | wc -l

```
[hadoop@ip-172-31-70-125 ~]$ hadoop fs -ls bookings_data
Found 2 items
-rw-r--r--   1 hadoop hadoop          0 2024-04-26 19:18 bookings_data/_SUCCESS
-rw-r--r--   1 hadoop hadoop     165678 2024-04-26 19:18 bookings_data/part-m-00000
[hadoop@ip-172-31-70-125 ~]$ hadoop fs -cat bookings_data/part-m-00000 | wc -l
1000
[hadoop@ip-172-31-70-125 ~]$
```

Task 3: Create aggregates for finding date-wise total bookings using the Spark script

1. create a python file (datewise_bookings_aggregates_spark.py) using vi editor. This file will contain the code which will aggregate the booking data for finding datewise total bookings

   vi datewise_bookings_aggregates_spark.py

2. Explaining the code in datewise_bookings_aggregates_spark.py

```
# Importing the modules

from pyspark.sql.types import *
from pyspark.sql.functions import *
from pyspark.sql import functions as F
from pyspark.sql import SparkSession
from pyspark.sql.functions import from_json
from pyspark.sql.types import TimestampType, IntegerType, FloatType,
ArrayType,LongType
from pyspark.sql import functions as func

# Initializing Spark Session

spark = SparkSession  \
    .builder  \
    .appName("StructuredSocketRead")  \
    .getOrCreate()
spark.sparkContext.setLogLevel('ERROR')

# Defining the schema

schema1 = StructType([
        StructField("booking_id", StringType()),
        StructField("customer_id",  LongType()),
        StructField("driver_id",  LongType()),
```

```
StructField("customer_app_version", StringType()),
StructField("customer_phone_os_version", StringType()),
StructField("pickup_lat", DoubleType()),
StructField("pickup_lon", DoubleType()),
StructField("drop_lat", DoubleType()),
StructField("drop_lon", DoubleType()),
StructField("pickup_timestamp", TimestampType()),
StructField("drop_timestamp", TimestampType()),
StructField("trip_fare", IntegerType()),
StructField("tip_amount", IntegerType()),
StructField("currency_code", StringType()),
StructField("cab_color", StringType()),
StructField("cab_registration_no", StringType()),
StructField("customer_rating_by_driver", IntegerType()),
StructField("rating_by_customer", IntegerType()),
StructField("passenger_count", IntegerType())
])

# Reading the file "part-m-00000" in bookings_data folder in hadoop

df=spark.read.csv("bookings_data/part-m-00000", schema=schema1)

# Creating data from column "pickup_date" and "pickup_timestamp"

df = df.withColumn("pickup_date", func.to_date(func.col("pickup_timestamp")))

# Group the data by "pickup_date"

date = df.groupBy('pickup_date').count()

# Saving the datewise total booking data in .csv format

date.coalesce(1).write.format('com.databricks.spark.csv').mode('overwrite').save("datewise_aggregated_data/")

# Saving the booking data in .csv format

df.coalesce(1).write.format('com.databricks.spark.csv').mode('overwrite').save('booking_data_csv/', header = 'true')
```

3. Run Spark Submit command

```
spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 datewise_bookings_aggregates_spark.py
```

```
[hadoop@ip-172-31-70-125 ~]$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 datewise_bookings_aggregates_spark.py
Ivy Default Cache set to: /home/hadoop/.ivy2/cache
The jars for the packages stored in: /home/hadoop/.ivy2/jars
:: loading settings :: url = jar:file:/usr/lib/spark/jars/ivy-2.4.0.jar!/org/apache/ivy/core/settings/ivysettings.xml
org.apache.spark#spark-sql-kafka-0-10_2.11 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-493bceb8-1a92-42ff-b9df-44438c544a85;1.0
        confs: [default]
        found org.apache.spark#spark-sql-kafka-0-10_2.11;2.4.5 in central
        found org.apache.kafka#kafka-clients;2.0.0 in central
        found org.lz4#lz4-java;1.4.0 in central
        found org.xerial.snappy#snappy-java;1.1.7.3 in central
        found org.slf4j#slf4j-api;1.7.16 in central
        found org.spark-project.spark#unused;1.0.0 in central
:: resolution report :: resolve 424ms :: artifacts dl 11ms
        :: modules in use:
        org.apache.kafka#kafka-clients;2.0.0 from central in [default]
        org.apache.spark#spark-sql-kafka-0-10_2.11;2.4.5 from central in [default]
        org.lz4#lz4-java;1.4.0 from central in [default]
        org.slf4j#slf4j-api;1.7.16 from central in [default]
        org.spark-project.spark#unused;1.0.0 from central in [default]
        org.xerial.snappy#snappy-java;1.1.7.3 from central in [default]
        ---------------------------------------------------------------------
        |                  |            modules            ||   artifacts   |
        |       conf       | number| search|dwnlded|evicted|| number|dwnlded|
        ---------------------------------------------------------------------
        |      default     |   6   |   0   |   0   |   0   ||   6   |   0   |
        ---------------------------------------------------------------------
```

```
24/04/26 19:35:49 INFO BlockManager: external shuffle service port = 7337
24/04/26 19:35:49 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, ip-172-31-70-125.ec2.internal, 42897, None)
24/04/26 19:35:49 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /metrics/json.
24/04/26 19:35:50 INFO EventLoggingListener: Logging events to hdfs:/var/log/spark/apps/application_1714157635183_0002
24/04/26 19:35:50 INFO Utils: Using initial executors = 50, max of spark.dynamicAllocation.initialExecutors, spark.dynamicAllocation.
24/04/26 19:35:50 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredReso
24/04/26 19:35:51 INFO SharedState: loading hive config file: file:/etc/spark/conf.dist/hive-site.xml
24/04/26 19:35:51 INFO SharedState: Setting hive.metastore.warehouse.dir ('null') to the value of spark.sql.warehouse.dir ('hdfs:///u
24/04/26 19:35:51 INFO SharedState: Warehouse path is 'hdfs:///user/spark/warehouse'.
24/04/26 19:35:51 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL.
24/04/26 19:35:51 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL/json.
24/04/26 19:35:51 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL/execution.
24/04/26 19:35:51 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL/execution/json.
24/04/26 19:35:51 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /static/sql.
24/04/26 19:35:52 INFO StateStoreCoordinatorRef: Registered StateStoreCoordinator endpoint
[hadoop@ip-172-31-70-125 ~]$
```

4. Make a directory using mkdir command

   hadoop fs -mkdir datewise_aggregated_data

5. Loading the data from local file system to hadoop file system

   hadoop fs- put ~/ datewise_aggregated_data datewise_aggregated_data

6. Checking the data file in Hadoop

   hadoop fs -ls datewise_aggregated_data

hadoop fs -cat datewise_aggregated_data/part-00000-20429a3a-dc5a-4539-9557-abbea1bf7616-c000.csv | wc -l

hadoop fs -ls booking_data_csv

hadoop fs -cat booking_data_csv/part-00000-42a51088-74e1-4e61-a9fb-66a412006b78-c000.csv | wc -l

```
[hadoop@ip-172-31-70-125 ~]$ hadoop fs -ls datewise_aggregated_data
Found 2 items
-rw-r--r--   1 hadoop hadoop          0 2024-04-26 19:36 datewise_aggregated_data/_SUCCESS
-rw-r--r--   1 hadoop hadoop       3758 2024-04-26 19:36 datewise_aggregated_data/part-00000-20429a3a-dc5a-4539-9557-abbea1bf7616-c000.csv
[hadoop@ip-172-31-70-125 ~]$ hadoop fs -cat datewise_aggregated_data/part-00000-20429a3a-dc5a-4539-9557-abbea1bf7616-c000.csv | wc -l
289
[hadoop@ip-172-31-70-125 ~]$ hadoop fs -ls booking_data_csv
Found 2 items
-rw-r--r--   1 hadoop hadoop          0 2024-04-26 19:36 booking_data_csv/_SUCCESS
-rw-r--r--   1 hadoop hadoop     182968 2024-04-26 19:36 booking_data_csv/part-00000-42a51088-74e1-4e61-a9fb-66a412006b78-c000.csv
[hadoop@ip-172-31-70-125 ~]$ hadoop fs -cat booking_data_csv/part-00000-42a51088-74e1-4e61-a9fb-66a412006b78-c000.csv | wc -l
1001
[hadoop@ip-172-31-70-125 ~]$
```

Task 4: Create Hive Managed Tables

1. First create a database

   create database if not exists cab_booking_data ;

   use cab_booking_data ;

```
[hadoop@ip-172-31-70-125 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> show databases;
OK
default
Time taken: 0.749 seconds, Fetched: 1 row(s)
hive> create database if not exists cab_booking_data ;
OK
Time taken: 0.388 seconds
hive>
```

```
hive> show databases;
OK
database_name
cab_booking_data
default
Time taken: 0.026 seconds, Fetched: 2 row(s)
hive> use cab_booking_data;
OK
Time taken: 0.03 seconds
hive>
```

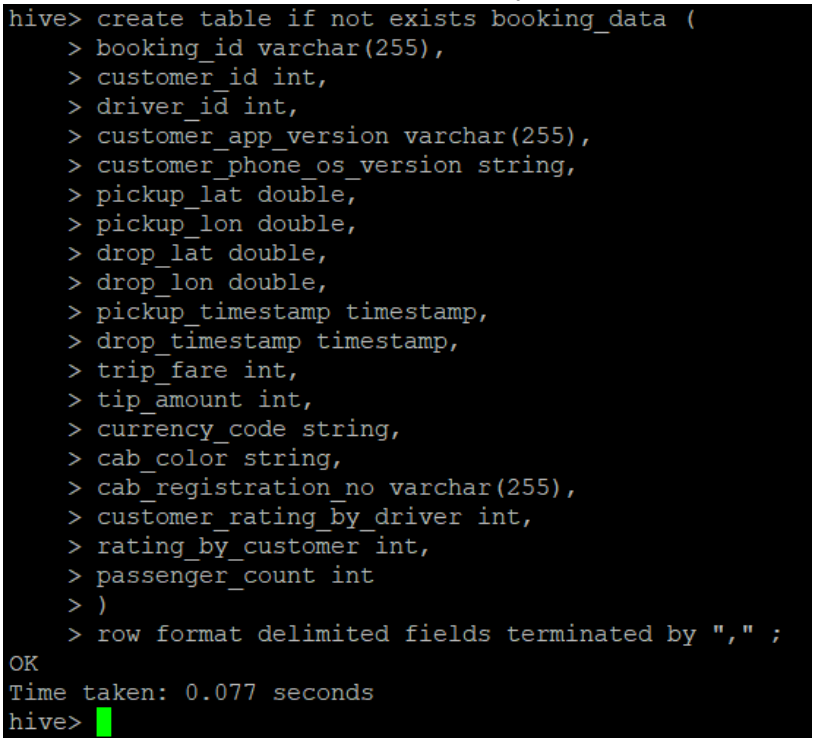
2. Creating a Hive-managed table for clickstream data

```
create table if not exists clickstream_data (
customer_id int ,
app_version varchar(255),
os_version string,
lat varchar(255),
lon varchar(255),
page_id varchar(255),
button_id varchar(255),
is_button_click string,
is_page_view string,
is_scroll_up string,
is_scroll_down string,
`timestamp` timestamp
)
row format delimited fields terminated by "," ;
```

```
hive> create table if not exists clickstream_data (
    > customer_id int ,
    > app_version varchar(255),
    > os_version string,
    > lat varchar(255),
    > lon varchar(255),
    > page_id varchar(255),
    > button_id varchar(255),
    > is_button_click string,
    > is_page_view string,
    > is_scroll_up string,
    > is_scroll_down string,
    > `timestamp` timestamp
    > )
    > row format delimited fields terminated by "," ;
OK
Time taken: 0.6 seconds
hive>
```

3. Creating a Hive-managed table for bookings data

```
create table if not exists booking_data (
booking_id varchar(255),
customer_id int,
driver_id int,
customer_app_version varchar(255),
customer_phone_os_version string,
pickup_lat double,
pickup_lon double,
drop_lat double,
drop_lon double,
pickup_timestamp timestamp,
drop_timestamp timestamp,
trip_fare int,
tip_amount int,
currency_code string,
cab_color string,
cab_registration_no varchar(255),
customer_rating_by_driver int,
rating_by_customer int,
passenger_count int
)
row format delimited fields terminated by "," ;
```



4. Creating a Hive-managed table for aggregated data in Task 3

```
create table if not exists datewise_aggregated_data (
`date` string,
count int
)
row format delimited fields terminated by "," ;
```

```
hive> create table if not exists datewise_aggregated_data (
    > `date` string,
    > count int
    > )
    > row format delimited fields terminated by "," ;
OK
Time taken: 0.071 seconds
hive>
```

5. Command to load the data into Hive tables

1. load data inpath 'clickstream_data_flatten/part-00000-bb423f13-4963-4dd7-8afb-0630877df998-c000.csv' into table clickstream_data ;

2. load data inpath 'booking_data_csv/part-00000-42a51088-74e1-4e61-a9fb-66a412006b78-c000.csv' into table booking_data ;

3. load data inpath 'datewise_aggregated_data/part-00000-20429a3a-dc5a-4539-9557-abbea1bf7616-c000.csv' into table datewise_aggregated_data ;

```
hive> load data inpath 'clickstream_data_flatten/part-00000-bb423f13-4963-4dd7-8afb-0630877df998-c000.csv' into table clickstream_data;
Loading data to table cab_booking_data.clickstream_data
OK
Time taken: 1.029 seconds
hive> load data inpath 'booking_data_csv/part-00000-42a51088-74e1-4e61-a9fb-66a412006b78-c000.csv' into table booking_data ;
Loading data to table cab_booking_data.booking_data
OK
Time taken: 0.629 seconds
hive> load data inpath 'datewise_aggregated_data/part-00000-20429a3a-dc5a-4539-9557-abbea1bf7616-c000.csv' into table datewise_aggregated_data
Loading data to table cab_booking_data.datewise_aggregated_data
OK
Time taken: 0.525 seconds
hive>
```

4. Verify the data in hive tables

select count(*) from clickstream_data;

```
hive> select count(*) from clickstream_data;
Query ID = hadoop_20240426195557_ff2a8956-1a8f-40ab-812b-91700bbbbdf1
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1714157635183_0004)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 6.02 s
--------------------------------------------------------------------------------
OK
_c0
2454
Time taken: 15.434 seconds, Fetched: 1 row(s)
hive>
```

select count(*) from booking_data;

```
hive> select count(*) from booking_data;
Query ID = hadoop_20240426195737_12c3659a-5273-4844-bfde-140ecedf493a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1714157635183_0004)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 5.77 s
--------------------------------------------------------------------------------
OK
_c0
1001
Time taken: 6.398 seconds, Fetched: 1 row(s)
hive>
```

select count(*) from datewise_aggregated_data;

```
hive> select count(*) from datewise_aggregated_data;
Query ID = hadoop_20240426195927_3ba355d3-40da-414f-bf64-c563797b8a39
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1714157635183_0004)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 5.30 s
--------------------------------------------------------------------------------
OK
_c0
289
Time taken: 5.867 seconds, Fetched: 1 row(s)
hive>
```