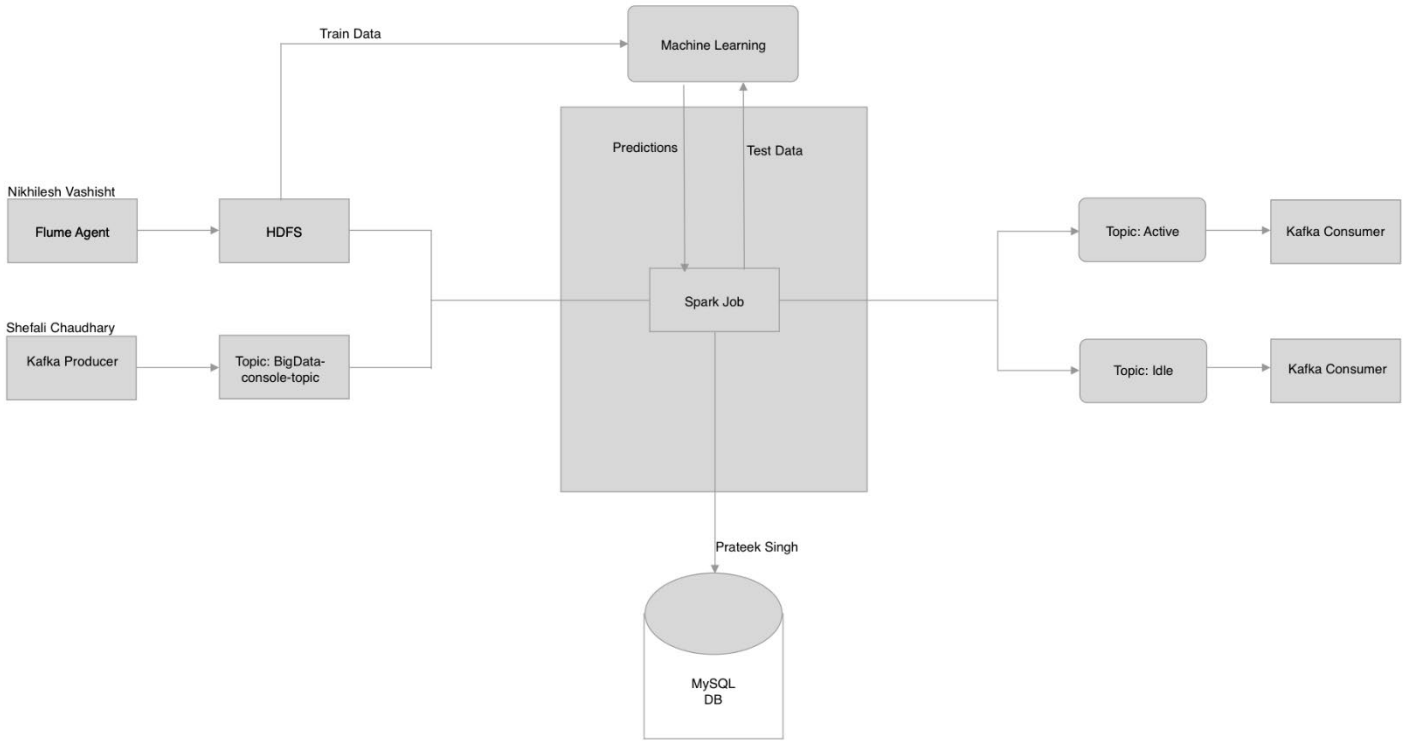
APACHE

Architecture



The above flowchart shows the architecture of the system we have designed and which group member was responsible for which part.

Flume – Nikhilesh Vashisht

"Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows. It is robust and fault tolerant with tunable reliability mechanisms and many failover and recovery mechanisms. It uses a simple extensible data model that allows for online analytic application"[1].

There are 3 steps involved in this:

1. Setting up the kafka topics.
   We have setup 2 kafka topics that is idle and active which receives the data from spark write stream.

```
SSH-in-browser                                                    UPLOAD FILE   DOWNLOAD FILE
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$ bin/kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 3 --topic idle
Error while executing topic command : Topic 'idle' already exists.
[2022-12-16 07:48:19,485] ERROR org.apache.kafka.common.errors.TopicExistsException: Topic 'idle' already exists.
 (kafka.admin.TopicCommand$)
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$ bin/kafka-topics.sh --list --zookeeper localhost:2181
-bash: bin/kafka-topics.sh: No such file or directory
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$ bin/kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 3 --topic idlee
Created topic "idlee".
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$ bin/kafka-console-consumer --bootstrap-server localhost:9092 --topic idle
```

```
SSH-in-browser                                                    UPLOAD FILE   DOWNLOAD FILE
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$ nohup bin/kafka-server-start etc/kafka/server.properties > /dev/null 2>&1 &
[1] 29624
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$ bin/kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 3 --topic active
Created topic "active".
[1]+  Exit 1                  nohup bin/kafka-server-start etc/kafka/server.properties > /dev/null 2>&1
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$ bin/kafka-console-consumer --bootstrap-server localhost:9092 --topic active
```

2. Setting up flume to send data.

   We are using Flume's **SpoolDir** as exec type, which keeps checking for new files in the directory mentioned in the configuration file and publishes it to the sink mentioned in the configuration file. We have mentioned HDFS as our sink. All older files are renamed by appending. COMPLETED to the file name.

   Data before running flume:

```
nikilesh_r98@dcc-l3-m:~/iot_data$ ls -ltrh
total 68K
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.8K Dec 16 03:31 1-stand.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 6.2K Dec 16 03:31 2-sit.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 4.4K Dec 16 03:31 3-staircsdown.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 4.3K Dec 16 03:32 4-bike.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.4K Dec 16 03:32 5-sit.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.8K Dec 16 03:32 6-staircsup.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 4.5K Dec 16 03:32 7-walk.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 2.8K Dec 16 03:32 8-mixed.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.0K Dec 16 03:32 9-mixed.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.0K Dec 16 03:32 10-mixed.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 4.0K Dec 16 03:32 11-mixed.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.8K Dec 16 03:32 12-mixed.json
-rw-r--r-- 1 root         root         1.9K Dec 17 00:21 test.COMPLETED
nikilesh_r98@dcc-l3-m:~/iot_data$
```

Run Flume:

```
nikilesh_r98@dcc-13-m:~$ /home/nikilesh_r98/apache-flume-1.9.0-bin/bin/flume-ng agent --conf /home/nikilesh_r98/configs/ -f /home/nikilesh_r98/configs/co
nfig.conf -Dflume.root.logger=DEBUG,console -n agent
Info: Including Hadoop libraries found via (/usr/lib/hadoop/bin/hadoop) for HDFS access
/usr/lib/hadoop/libexec/hadoop-functions.sh: line 2365: HADOOP_ORG.APACHE.FLUME.TOOLS.GETJAVAPROPERTY_USER: invalid variable name
/usr/lib/hadoop/libexec/hadoop-functions.sh: line 2460: HADOOP_ORG.APACHE.FLUME.TOOLS.GETJAVAPROPERTY_OPTS: invalid variable name
Info: Including Hive libraries found via () for Hive access
+ exec /usr/lib/jvm/temurin-8-jdk-amd64/bin/java -Xmx20m -Dflume.root.logger=DEBUG,console -cp '/home/nikilesh_r98/configs:/home/nikilesh_r98/apache-flum
e-1.9.0-bin/lib/*:/etc/hadoop/conf:/usr/lib/hadoop/lib/*:/usr/lib/hadoop/.//*:/usr/lib/hadoop-hdfs/./:/usr/lib/hadoop-hdfs/lib/*:/usr/lib/hadoop-hdfs/.//
*:/usr/lib/hadoop-mapreduce/.//*:/usr/lib/hadoop-yarn/lib/*:/usr/lib/hadoop-yarn/.//*:/usr/local/share/google/dataproc/lib/gcs-connector-hadoop3-2.2.8.ja
r:/usr/local/share/google/dataproc/lib/gcs-connector.jar:/usr/local/share/google/dataproc/lib/ranger_gcs_plugin_client.jar:/usr/local/share/google/datapr
oc/lib/spark-metrics-listener-1.0.1.jar:/usr/local/share/google/dataproc/lib/spark-metrics-listener.jar:/lib/*' -Djava.library.path=:/usr/lib/hadoop/lib/
native org.apache.flume.node.Application -f /home/nikilesh_r98/configs/config.conf -n agent
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/nikilesh_r98/apache-flume-1.9.0-bin/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-reload4j-1.7.35.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2022-12-17 18:23:48,715 INFO node.PollingPropertiesFileConfigurationProvider: Configuration provider starting
2022-12-17 18:23:48,719 INFO node.PollingPropertiesFileConfigurationProvider: Reloading configuration file:/home/nikilesh_r98/configs/config.conf
2022-12-17 18:23:48,723 INFO conf.FlumeConfiguration: Processing:tail-source
2022-12-17 18:23:48,724 INFO conf.FlumeConfiguration: Processing:tail-source
2022-12-17 18:23:48,724 INFO conf.FlumeConfiguration: Processing:tail-source
2022-12-17 18:23:48,724 INFO conf.FlumeConfiguration: Processing:hdfs-sink
2022-12-17 18:23:48,724 INFO conf.FlumeConfiguration: Processing:memory-channel
2022-12-17 18:23:48,724 INFO conf.FlumeConfiguration: Processing:tail-source
2022-12-17 18:23:48,724 INFO conf.FlumeConfiguration: Added sinks: hdfs-sink Agent: agent
2022-12-17 18:23:48,724 INFO conf.FlumeConfiguration: Processing:hdfs-sink
2022-12-17 18:23:48,724 INFO conf.FlumeConfiguration: Processing:hdfs-sink
2022-12-17 18:23:48,724 INFO conf.FlumeConfiguration: Processing:hdfs-sink
2022-12-17 18:23:48,724 WARN conf.FlumeConfiguration: Agent configuration for 'agent' has no configfilters.
2022-12-17 18:23:48,741 INFO conf.FlumeConfiguration: Post-validation flume configuration contains configuration for agents: [agent]
2022-12-17 18:23:48,741 INFO node.AbstractConfigurationProvider: Creating channels
2022-12-17 18:23:48,750 INFO channel.DefaultChannelFactory: Creating instance of channel memory-channel type memory
```

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
2022-12-17 00:19:11,410 INFO instrumentation.MonitoredCounterGroup: Component type: SOURCE, name: tail-source started
2022-12-17 00:19:11,681 INFO avro.ReliableSpoolingFileEventReader: Last read took us just up to a file boundary. Rolling to the next file, if there is on
e.
2022-12-17 00:19:11,682 INFO avro.ReliableSpoolingFileEventReader: Preparing to move file /home/nikilesh_r98/iot_data/12-mixed.json to /home/nikilesh_r98
/iot_data/12-mixed.json.COMPLETED
2022-12-17 00:19:11,691 INFO hdfs.HDFSDataStream: Serializer = TEXT, UseRawLocalFileSystem = false
2022-12-17 00:19:11,818 INFO hdfs.BucketWriter: Creating /iot_data//FlumeData.1671236351692.tmp
2022-12-17 00:19:14,018 INFO hdfs.BucketWriter: Closing /iot_data//FlumeData.1671236351692.tmp
2022-12-17 00:19:14,042 INFO hdfs.BucketWriter: Renaming /iot_data/FlumeData.1671236351692.tmp to /iot_data/FlumeData.1671236351692
2022-12-17 00:19:14,069 INFO hdfs.BucketWriter: Creating /iot_data//FlumeData.1671236351693.tmp
2022-12-17 00:19:14,106 INFO hdfs.BucketWriter: Closing /iot_data//FlumeData.1671236351693.tmp
2022-12-17 00:19:14,120 INFO hdfs.BucketWriter: Renaming /iot_data/FlumeData.1671236351693.tmp to /iot_data/FlumeData.1671236351693
2022-12-17 00:19:14,137 INFO hdfs.BucketWriter: Creating /iot_data//FlumeData.1671236351694.tmp
2022-12-17 00:19:14,175 INFO hdfs.BucketWriter: Closing /iot_data//FlumeData.1671236351694.tmp
2022-12-17 00:19:14,187 INFO hdfs.BucketWriter: Renaming /iot_data/FlumeData.1671236351694.tmp to /iot_data/FlumeData.1671236351694
2022-12-17 00:19:14,209 INFO hdfs.BucketWriter: Creating /iot_data//FlumeData.1671236351695.tmp

2022-12-17 00:19:44,232 INFO hdfs.HDFSEventSink: Writer callback called.
2022-12-17 00:19:44,232 INFO hdfs.BucketWriter: Closing /iot_data//FlumeData.1671236351695.tmp
2022-12-17 00:19:44,245 INFO hdfs.BucketWriter: Renaming /iot_data/FlumeData.1671236351695.tmp to /iot_data/FlumeData.1671236351695
2022-12-17 00:21:39,853 INFO hdfs.HDFSDataStream: Serializer = TEXT, UseRawLocalFileSystem = false
2022-12-17 00:21:39,854 INFO avro.ReliableSpoolingFileEventReader: Last read took us just up to a file boundary. Rolling to the next file, if there is on
e.
2022-12-17 00:21:39,854 INFO avro.ReliableSpoolingFileEventReader: Preparing to move file /home/nikilesh_r98/iot_data/test to /home/nikilesh_r98/iot_data
/test.COMPLETED
2022-12-17 00:21:39,872 INFO hdfs.BucketWriter: Creating /iot_data//FlumeData.1671236499854.tmp
2022-12-17 00:21:39,915 INFO hdfs.BucketWriter: Closing /iot_data//FlumeData.1671236499854.tmp
2022-12-17 00:21:39,925 INFO hdfs.BucketWriter: Renaming /iot_data/FlumeData.1671236499854.tmp to /iot_data/FlumeData.1671236499854
2022-12-17 00:21:39,945 INFO hdfs.BucketWriter: Creating /iot_data//FlumeData.1671236499855.tmp
2022-12-17 00:22:09,965 INFO hdfs.HDFSEventSink: Writer callback called.
2022-12-17 00:22:09,965 INFO hdfs.BucketWriter: Closing /iot_data//FlumeData.1671236499855.tmp
2022-12-17 00:22:09,977 INFO hdfs.BucketWriter: Renaming /iot_data/FlumeData.1671236499855.tmp to /iot_data/FlumeData.1671236499855
```

```
nikilesh_r98@dcc-13-m:~/iot_data$ ls -ltrh
total 68K
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.8K Dec 16 03:31 1-stand.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 6.2K Dec 16 03:31 2-sit.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 4.4K Dec 16 03:31 3-stairsdown.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 4.3K Dec 16 03:32 4-bike.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.4K Dec 16 03:32 5-sit.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.8K Dec 16 03:32 6-stairsup.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 4.5K Dec 16 03:32 7-walk.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 2.8K Dec 16 03:32 8-mixed.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.0K Dec 16 03:32 9-mixed.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.0K Dec 16 03:32 10-mixed.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 4.0K Dec 16 03:32 11-mixed.json
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.8K Dec 16 03:32 12-mixed.json
-rw-r--r-- 1 root         root         1.9K Dec 17 00:21 test.COMPLETED
nikilesh_r98@dcc-13-m:~/iot_data$ ls -ltrh
total 68K
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.8K Dec 16 03:31 1-stand.json.COMPLETED
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 6.2K Dec 16 03:31 2-sit.json.COMPLETED
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 4.4K Dec 16 03:31 3-stairsdown.json.COMPLETED
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 4.3K Dec 16 03:32 4-bike.json.COMPLETED
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.4K Dec 16 03:32 5-sit.json.COMPLETED
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.8K Dec 16 03:32 6-stairsup.json.COMPLETED
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 4.5K Dec 16 03:32 7-walk.json.COMPLETED
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 2.8K Dec 16 03:32 8-mixed.json.COMPLETED
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.0K Dec 16 03:32 9-mixed.json.COMPLETED
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.0K Dec 16 03:32 10-mixed.json.COMPLETED
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 4.0K Dec 16 03:32 11-mixed.json.COMPLETED
-rw-r--r-- 1 nikilesh_r98 nikilesh_r98 3.8K Dec 16 03:32 12-mixed.json.COMPLETED
-rw-r--r-- 1 root         root         1.9K Dec 17 00:21 test.COMPLETED
nikilesh_r98@dcc-13-m:~/iot_data$
```

3. Writing spark-streaming code to process it.
    a. We have created a schema.
    b. Used readStream function to read the json file from hdfs location which was transferred by flume.
    c. Selected which columns we need to print in the kafka topics and converted them to key:value pairs. Applied a filter to filter out between idle and active data.
    d. With append mode in spark writeStream, we are appending all the new data received by flume onto the sink.
    e. The spark-streaming job will continue to stream to the kafka topics until we terminate the spark job.

    We have used different sessions for different kafka topics and different spark job for to write to each kafka topics

    Spark Job to push to **idle** kafka topic

```
scala> :paste
// Entering paste mode (ctrl-D to finish)

val iot_active = spark.readStream.format("json").schema(userSchema).option("path", "hdfs:///iot_data/").load()

// Exiting paste mode, now interpreting.

iot_active: org.apache.spark.sql.DataFrame = [Arrival_Time: string, Device: string ... 1 more field]

scala> :paste
// Entering paste mode (ctrl-D to finish)

val iot_key_val = iot_active.withColumn("key", lit(100))
 .select(col("key").cast("string"), concat(col("Arrival_Time"), lit("  "), col("Device"), lit(" "), col("gt")).alias("value")).filter($"gt" === "sit" ||
$"gt" === "stand")

// Exiting paste mode, now interpreting.

iot_key_val: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [key: string, value: string]

scala> :paste
// Entering paste mode (ctrl-D to finish)

val stream = iot_key_val.writeStream.format("kafka").option("kafka.bootstrap.servers", "localhost:9092").option("topic", "idle").option("checkpointLocati
on", "/home/nikilesh_r98/checkpoint").outputMode("append").start()

// Exiting paste mode, now interpreting.

22/12/17 00:46:33 WARN org.apache.spark.sql.streaming.StreamingQueryManager: spark.sql.adaptive.enabled is not supported in streaming DataFrames/Datasets
 and will be disabled.
stream: org.apache.spark.sql.streaming.StreamingQuery = org.apache.spark.sql.execution.streaming.StreamingQueryWrapper@1ab433a7
```

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
1424687114434  nexus4_1 sit
1424687114825  nexus4_1 sit
1424687115137  nexus4_1 sit
1424779593164  nexus4_1 sit
1424779593365  nexus4_1 sit
1424779593568  nexus4_1 sit
1424779593769  nexus4_1 sit
1424779593980  nexus4_1 sit
1424694147486  nexus4_2 sit
1424694147690  nexus4_1 sit
1424694147888  nexus4_2 sit
1424699115186  nexus4_2 sit
1424699115385  nexus4_2 sit
1424699420947  nexus4_1 sit
1424699421148  nexus4_2 sit
1424699421345  nexus4_1 sit
1424699421549  nexus4_2 sit
1424699114185  nexus4_1 sit
1424699114380  nexus4_2 sit
1424699114581  nexus4_2 sit
1424699114783  nexus4_1 sit
1424699114983  nexus4_2 sit
1424776359825  nexus4_1 stand
1424776360026  nexus4_1 stand
1424776360234  nexus4_2 stand
1424776360428  nexus4_2 stand
1424776360636  nexus4_1 stand
1424776360841  nexus4_1 stand
1424779593164  nexus4_1 sit
1424779593365  nexus4_1 sit
1424779593568  nexus4_1 sit
1424779593769  nexus4_1 sit
```

Spark Job to push to **active** kafka topic

```
// Exiting paste mode, now interpreting.

import org.apache.spark.sql._
import org.apache.spark.sql.types._
userSchema: org.apache.spark.sql.types.StructType = StructType(StructField(Arrival_Time,StringType,true), StructField(Device,StringType,true), StructFiel
d(gt,StringType,true))
iot: org.apache.spark.sql.DataFrame = [Arrival_Time: string, Device: string ... 1 more field]
iot_key_val: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [key: string, value: string]

scala> iot_key_val.printSchema
root
 |-- key: string (nullable = false)
 |-- value: string (nullable = true)


scala> :paste
// Entering paste mode (ctrl-D to finish)

val stream = iot_key_val.writeStream
 .format("kafka")
 .option("kafka.bootstrap.servers", "localhost:9092")
 .option("topic", "active") .option("checkpointLocation", "/home/nikilesh_r98/checkpoint-active")
 .outputMode("append")
 .start()

// Exiting paste mode, now interpreting.

22/12/17 18:42:50 WARN org.apache.spark.sql.streaming.StreamingQueryManager: spark.sql.adaptive.enabled is not supported in streaming DataFrames/Datasets
 and will be disabled.
stream: org.apache.spark.sql.streaming.StreamingQuery = org.apache.spark.sql.execution.streaming.StreamingQueryWrapper@3ab3a232
```

```
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$ bin/kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 3 --topic active
Created topic "active".
[1]+  Exit 1                  nohup bin/kafka-server-start etc/kafka/server.properties > /dev/null 2>&1
nikilesh_r98@dcc-l3-m:~/confluent-4.1.4$ bin/kafka-console-consumer --bootstrap-server localhost:9092 --topic active




1424777758749  nexus4_2 stairsdown
1424777758955  nexus4_1 stairsdown
1424777759151  nexus4_2 stairsdown
1424777759358  nexus4_1 stairsdown
1424777759558  nexus4_1 stairsdown
1424777759761  nexus4_2 stairsdown
1424777758749  nexus4_2 stairsdown
1424777758955  nexus4_1 stairsdown
1424777759151  nexus4_2 stairsdown
1424777759358  nexus4_1 stairsdown
1424777759558  nexus4_1 stairsdown
1424777759761  nexus4_2 stairsdown
1424688481462  nexus4_1 stairsdown
1424688481668  nexus4_2 stairsdown
1424688481868  nexus4_2 stairsdown
1424688482063  nexus4_1 stairsdown
1424688482269  nexus4_1 stairsdown
1424688482468  nexus4_1 stairsdown
1424688481462  nexus4_1 stairsdown
```

Kafka

Idle topic created in Kafka:

```
shefali_gc_1994@shefali-m:~/confluent-4.1.4$ bin/kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 3 --topic idle
Created topic "idle".
shefali_gc_1994@shefali-m:~/confluent-4.1.4$ bin/kafka-console-consumer --bootstrap-server localhost:9092 --topic idle
```

Spark session started to filter out the activities based on their type. Only sit and stand were considered as idle activities.

```
scala> :paste
// Entering paste mode (ctrl-D to finish)

import org.apache.spark.sql._
import org.apache.spark.sql.types._

// data is in json format so we provide the schema

val userSchema = new StructType()
 .add("Arrival_Time", "string")
 .add("Device", "string")
 .add("gt", "string")



val iot = spark.readStream.format("json")
 .schema(userSchema)
 .option("path", "hdfs:///BigData/").load()

// to output the data to a Kafka sink, our data should confirm to a key-value pair
// so we are adding a column called "key" and giving the value of the key to 100
// for the value, we are sending the arrival time, device and action user performed
// we are concatenating all three columns and adding a space between columns for differentiating purposes

val iot_key_val = iot.withColumn("key", lit(100))
 .select(col("key").cast("string"), concat(col("Arrival_Time"), lit("  "), col("Device"), lit(" "), col("gt")).alias("value")).filter($"gt" === "sit" || $"gt" === "stand")

// Exiting paste mode, now interpreting.

import org.apache.spark.sql._
import org.apache.spark.sql.types._
userSchema: org.apache.spark.sql.types.StructType = StructType(StructField(Arrival_Time,StringType,true), StructField(Device,StringType,true), StructField(gt,StringType,true))
iot: org.apache.spark.sql.DataFrame = [Arrival_Time: string, Device: string ... 1 more field]
iot_key_val: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [key: string, value: string]
```

Trigger is set in the below command and non-aggregation mode - append is used.

```
scala> :paste
// Entering paste mode (ctrl-D to finish)

val stream = iot_key_val.writeStream
 .format("kafka")
 .option("kafka.bootstrap.servers", "localhost:9092")
 .option("topic", "idle") .option("checkpointLocation", "file:////home/shefali_gc_1994/chkpt")
 .outputMode("append")
 .start()

// Exiting paste mode, now interpreting.

22/12/17 13:01:05 WARN org.apache.spark.sql.streaming.StreamingQueryManager: spark.sql.adaptive.enabled is not supported in streaming DataFrames/Datasets and will be disabled.
stream: org.apache.spark.sql.streaming.StreamingQuery = org.apache.spark.sql.execution.streaming.StreamingQueryWrapper@7ddb114d
```

Another topic, active is created in another SSH session for active activities.

```
shefali_gc_1994@shefali-m:~/confluent-4.1.4$ bin/kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 3 --topic active
Created topic "active".
shefali_gc_1994@shefali-m:~/confluent-4.1.4$ bin/kafka-console-consumer --bootstrap-server localhost:9092 --topic active
```

Spark session started to filter out the active activities. Stairs up and down, bike and walk were considered as active activities.

```
scala> :paste
// Entering paste mode (ctrl-D to finish)

import org.apache.spark.sql._
import org.apache.spark.sql.types._

// data is in json format so we need to provide the schema

val userSchema = new StructType()
 .add("Arrival_Time", "string")
 .add("Device", "string")
 .add("gt", "string")



val iot = spark.readStream.format("json")
 .schema(userSchema)
 .option("path", "hdfs:///BigData/").load()

// to output the data to a Kafka sink, our data should confirm to a key-value pair
// so we are adding a column called "key" and giving the value of the key to 100
// for the value, we are sending the arrival time, device and action user performed
// we are concatenating all three columns and adding a space between columns so we can tell the difference

//val iot_key_val = iot.withColumn("key", lit(100))
// .select(col("key").cast("string"), concat(col("Arrival_Time"), lit("  "), col("Device"), lit(" "), col("gt")).alias("value")).filter($"gt" != "sit" & $"gt" != "stand")



val iot_key_val = iot.withColumn("key", lit(100))
 .select(col("key").cast("string"), concat(col("Arrival_Time"), lit("  "), col("Device"), lit(" "), col("gt")).alias("value")).filter($"gt" === "stairsdown" || $"gt" === "stairsup" || $"g
t" === "bike" || $"gt" === "walk")

// Exiting paste mode, now interpreting.

import org.apache.spark.sql._
import org.apache.spark.sql.types._
userSchema: org.apache.spark.sql.types.StructType = StructType(StructField(Arrival_Time,StringType,true), StructField(Device,StringType,true), StructField(gt,StringType,true))
iot: org.apache.spark.sql.DataFrame = [Arrival_Time: string, Device: string ... 1 more field]
iot_key_val: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [key: string, value: string]
```

Again, a trigger is created for the active activities and append mode is used.

```
scala> :paste
// Entering paste mode (ctrl-D to finish)

val stream = iot_key_val.writeStream
 .format("kafka")
 .option("kafka.bootstrap.servers", "localhost:9092")
 .option("topic", "active") .option("checkpointLocation", "file:////home/shefali_gc_1994/chkpt2")
 .outputMode("append")
 .start()

// Exiting paste mode, now interpreting.

22/12/17 13:04:19 WARN org.apache.spark.sql.streaming.StreamingQueryManager: spark.sql.adaptive.enabled is not supported in streaming DataFrames/Datasets and will be disabled.
stream: org.apache.spark.sql.streaming.StreamingQuery = org.apache.spark.sql.execution.streaming.StreamingQueryWrapper@1e71b70d
```

File moved from local system to the BigData folder in HDFS

```
shefali_gc_1994@shefali-m:~$ hadoop fs -copyFromLocal 8-mixed.json /BigData/.
shefali_gc_1994@shefali-m:~$
```

Output in active topic

```
shefali_gc_1994@shefali-m:~/confluent-4.1.4$ bin/kafka-console-consumer --bootstrap-server localhost:9092 --topic active
1424698515873  nexus4_2 bike
1424698516079  nexus4_1 bike
1424698516273  nexus4_1 bike
1424698516477  nexus4_2 bike
1424698516679  nexus4_1 bike
1424698516880  nexus4_2 bike
1424698517082  nexus4_2 bike
1424698517284  nexus4_1 bike
```

Output in idle topic

```
shefali_gc_1994@shefali-m:~/confluent-4.1.4$ bin/kafka-console-consumer --bootstrap-server localhost:9092 --topic idle
1424699114185  nexus4_1 sit
1424699114380  nexus4_2 sit
1424699114581  nexus4_2 sit
1424699114783  nexus4_1 sit
1424699114983  nexus4_2 sit
1424699115186  nexus4_2 sit
1424699115385  nexus4_2 sit
```

MySQL & MLStreaming

We have moved the IOT data to MySQl and we are using the cancer dataset to train a ML model and use it for ML streaming with another kafka topic called cancer-prod.

Start the kafka topic **cancer-prod**

```
https://ssh.cloud.google.com/v2/ × https://ssh.cloud.google.com/v2/ × +
ssh.cloud.google.com/v2/ssh/projects/inbound-domain-371802/zones/us-central1-f/instances/cluster-ac20-m?authuser=0&hl=en_US&projectNumber=484499882236&useAdminProxy=tr...
SSH-in-browser                                                    UPLOAD FILE   DOWNLOAD FILE

prateeksingh21031994@cluster-ac20-m:~/confluent-4.1.4$ bin/kafka-console-consumer --bootstrap-server localhost:9092 --topic cancer-pred
```

Copy cancer.csv from local to hdfs

```
SSH-in-browser

prateeksingh21031994@cluster-ac20-m:~$ hadoop fs -copyFromLocal cancer.csv /BigData/
```

Spark code for the ML streaming

```
SSH-in-browser                                                    UPLOAD FILE   DOWNLOAD FILE

scala> :paste
// Entering paste mode (ctrl-D to finish)

import org.apache.spark.sql.functions._
import org.apache.spark.sql.expressions.Window
import org.apache.spark.ml.feature.{VectorAssembler, StringIndexer}
import org.apache.spark.ml.Pipeline
import org.apache.spark.ml.classification.{RandomForestClassificationModel, RandomForestClassifier}
import org.apache.spark.ml.tuning.{CrossValidator, CrossValidatorModel, ParamGridBuilder}
import org.apache.spark.ml.evaluation.{MulticlassClassificationEvaluator}
import org.apache.spark.ml.param.ParamMap
import org.apache.spark.sql.types._
import org.apache.spark.sql._

val schema = new StructType()
.add("id", "integer")
.add("Clump Thickness", "integer")
.add("UofCSize", "integer")
.add("UofCShape", "integer")
.add("Marginal Adhesion", "integer")
.add("SECSize", "integer")
.add("Bare Nuclei", "integer")
.add("Bland Chromatin", "integer")
.add("Normal Nucleoli", "integer")
.add("Mitoses", "integer")
.add("Class", "integer")

// Exiting paste mode, now interpreting.

import org.apache.spark.sql.functions._
import org.apache.spark.sql.expressions.Window
import org.apache.spark.ml.feature.{VectorAssembler, StringIndexer}
import org.apache.spark.ml.Pipeline
import org.apache.spark.ml.classification.{RandomForestClassificationModel, RandomForestClassifier}
import org.apache.spark.ml.tuning.{CrossValidator, CrossValidatorModel, ParamGridBuilder}
import org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator
import org.apache.spark.ml.param.ParamMap
import org.apache.spark.sql.types._
import org.apache.spark.sql._
schema: org.apache.spark.sql.types.StructType = StructType(StructField(id,IntegerType,true), StructField(Clump Thickness,IntegerType,true), StructField(UofCSize,IntegerType,true), StructField(
UofCShape,IntegerT...

scala>
```

```
scala> val streamContent_key_val = streamContent.withColumn("key", lit(100)).select(col("key").cast("string"), concat(col("id"), lit("| class: "), col("class").cast("string"), lit("| predition
: "), col("prediction").cast("string")).alias("value"))
streamContent_key_val: org.apache.spark.sql.DataFrame = [key: string, value: string]

scala> val stream_data = streamContent_key_val.writeStream.format("kafka")
stream_data: org.apache.spark.sql.streaming.DataStreamWriter[org.apache.spark.sql.Row] = org.apache.spark.sql.streaming.DataStreamWriter@7d1cb447

scala>  .option("kafka.bootstrap.servers", "localhost:9092")
res0: org.apache.spark.sql.streaming.DataStreamWriter[org.apache.spark.sql.Row] = org.apache.spark.sql.streaming.DataStreamWriter@7d1cb447

scala>  .option("topic", "cancer-pred") .option("checkpointLocation", "file:////home/prateeksingh21031994/chkpt")
res1: org.apache.spark.sql.streaming.DataStreamWriter[org.apache.spark.sql.Row] = org.apache.spark.sql.streaming.DataStreamWriter@7d1cb447

scala>  .outputMode("update")
res2: org.apache.spark.sql.streaming.DataStreamWriter[org.apache.spark.sql.Row] = org.apache.spark.sql.streaming.DataStreamWriter@7d1cb447

scala>  .start()
22/12/17 21:53:13 WARN org.apache.spark.sql.streaming.StreamingQueryManager: spark.sql.adaptive.enabled is not supported in streaming DataFrames/Datasets and will be disabled.
res3: org.apache.spark.sql.streaming.StreamingQuery = org.apache.spark.sql.execution.streaming.StreamingQueryWrapper@21bdcdc
```

Predicitions on the kafka, **cancer-pred**

SSH-in-browser

```
prateeksingh21031994@cluster-ac20-m:~/confluent-4.1.4$ bin/kafka-console-consumer --bootstrap-server localhost:9092 --topic cancer-pred

1000025| class: 2| predition: 2.0
1002945| class: 2| predition: 2.0
1015425| class: 2| predition: 2.0
1016277| class: 2| predition: 2.0
1017023| class: 2| predition: 2.0
1017122| class: 4| predition: 4.0
1018099| class: 2| predition: 2.0
1018561| class: 2| predition: 2.0
1033078| class: 2| predition: 2.0
1033078| class: 2| predition: 2.0
1035283| class: 2| predition: 2.0
1036172| class: 2| predition: 2.0
1041801| class: 4| predition: 2.0
1043999| class: 2| predition: 2.0
1044572| class: 4| predition: 4.0
1047630| class: 4| predition: 4.0
1048672| class: 2| predition: 2.0
1049815| class: 2| predition: 2.0
1050670| class: 4| predition: 4.0
1050718| class: 2| predition: 2.0
1054590| class: 4| predition: 4.0
1054593| class: 4| predition: 4.0
1056784| class: 2| predition: 2.0
1059552| class: 2| predition: 2.0
1065726| class: 4| predition: 4.0
1066373| class: 2| predition: 2.0
1066979| class: 2| predition: 2.0
1067444| class: 2| predition: 2.0
1070935| class: 2| predition: 2.0
1070935| class: 2| predition: 2.0
1071760| class: 2| predition: 2.0
1072179| class: 4| predition: 4.0
1074610| class: 2| predition: 2.0
1075123| class: 2| predition: 2.0
1079304| class: 2| predition: 2.0
1080185| class: 4| predition: 4.0
1081791| class: 2| predition: 2.0
1084584| class: 4| predition: 4.0
1091262| class: 4| predition: 4.0
1099510| class: 4| predition: 4.0
1100524| class: 4| predition: 4.0
```

Writing the IOT data to MySQL using SPARK

VM instances – Compute Engi | https://ssh.cloud.google.com/v | https://ssh.cloud.google.com/v | https://ssh.cloud.google.com/v | https://ssh.cloud.google.com/v | https://ssh.cloud.google.com/v

ssh.cloud.google.com/v2/ssh/projects/inbound-domain-371802/zones/us-central1-f/instances/cluster-ac20-m?authuser=0&hl=en_US&projectNumber=484499882236&use...

SSH-in-browser

UPLOAD FILE DOWNLOAD FILE

```
                                .option("password", "hello")
                                .mode("append")
                                .save()
      }
saveToMySql: (org.apache.spark.sql.Dataset[org.apache.spark.sql.Row], Long) => Unit

scala> :paste
// Entering paste mode (ctrl-D to finish)

 staticDf.writeStream
      .outputMode("append")
      .foreachBatch(saveToMySql)
      .start()
      .awaitTermination()


// Exiting paste mode, now interpreting.

22/12/17 05:06:37 WARN org.apache.spark.sql.streaming.StreamingQueryManager: Temporary checkpoint location created which is deleted normally when the query didn't fail: /tmp/temporary-bb087cfc-38bf-4b
5e-a1d1-1635159862 70. If it's required to delete it under any circumstances, please set spark.sql.streaming.forceDeleteTempCheckpointLocation to true. Important to know deleting temp checkpoint folder
 is best effort.
22/12/17 05:06:37 WARN org.apache.spark.sql.streaming.StreamingQueryManager: spark.sql.adaptive.enabled is not supported in streaming DataFrames/Datasets and will be disabled.
-------------------------------------------
Batch: 1
-------------------------------------------
+-------------+--------+----+
| Arrival_Time|  Device|  gt|
+-------------+--------+----+
|1424779591955|nexus4_2| sit|
|1424779592156|nexus4_2| sit|
|1424779592359|nexus4_2| sit|
|1424779592559|nexus4_1| sit|
|1424779592763|nexus4_1| sit|
|1424779592963|nexus4_1| sit|
|1424779593164|nexus4_1| sit|
|1424779593365|nexus4_1| sit|
|1424779593568|nexus4_1| sit|
|1424779593769|nexus4_1| sit|
|1424779593980|nexus4_1| sit|
|1424779676612|nexus4_2|walk|
|1424779676810|nexus4_1|walk|
|1424779677012|nexus4_1|walk|
|1424779677226|nexus4_2|walk|
|1424779677414|nexus4_1|walk|
|1424779677614|nexus4_1|walk|
|1424779677816|nexus4_1|walk|
|1424779678022|nexus4_1|walk|
|1424779678219|nexus4_1|walk|
+-------------+--------+----+
```

View the data on MySQL:

```
prateeksingh21031994@cluster-ac20-m:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1018
Server version: 5.7.40 MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| finalProject       |
| metastore          |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.00 sec)

mysql> USE finalProject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+------------------------+
| Tables_in_finalProject |
+------------------------+
| iotData                |
| test                   |
+------------------------+
2 rows in set (0.00 sec)

mysql> select * from iotData;
+---------------+----------+------------+---------+
| Arrival_Time  | Device   | gt         | batchId |
+---------------+----------+------------+---------+
| 1424687114037 | nexus4_1 | sit        |       0 |
| 1424687114434 | nexus4_1 | sit        |       0 |
| 1424687114825 | nexus4_1 | sit        |       0 |
| 1424687115137 | nexus4_1 | sit        |       0 |
| 1424687115338 | nexus4_1 | sit        |       0 |
| 1424687115540 | nexus4_2 | sit        |       0 |
```

References:

1. https://flume.apache.org/

- https://www.conduktor.io/kafka/kafka-topics-cli-tutorial
- https://www.conduktor.io/kafka/kafka-topics-cli-tutorial
- https://www.conduktor.io/kafka/kafka-consumers-in-group-cli-tutorial
- https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html
- https://spark.apache.org/docs/2.3.0/sql-programming-guide.html
- https://spark.apache.org/docs/2.3.0/sql-programming-guide.html
- https://spark.apache.org/docs/2.3.0/sql-programming-guide.html
- https://spark.apache.org/docs/2.3.0/sql-programming-guide.html
- https://spark.apache.org/docs/2.2.0/streaming-flume-integration.html#:~:text=Flume%20pushes%20data%20into%20the,and%20replicated%20by%20Spark%20Streaming
- https://towardsdatascience.com/learn-how-to-use-spark-ml-and-spark-streaming-3a731485d052
- https://youtu.be/UuQz7G2Eux8