**Data Science full-stack developer – Case Study**

Note : I have used Cloudera VM to resolve this Case Study

* 1. **Hadoop directory Structure to be created.**

HDFS DFS -MKDIR Emirates/Raw/data/raw

HDFS DFS –MKDIR Emirates/Decomposed/data/decomposed

HDFS DFS –MKDIR Emirates/Modelled/data/modelled

HDFS DFS –MKDIR Emirates/Schema/data/schema

* 1. **Source data details**

Download the stats for 2008 and Supplemental data for Kafka topics

* 1. **Data preparation**

Create Kafka Cluster with 2 Broker

***Create folder for log file & Edit server. Properties & Server1.Properties***

Broker.id=0, Listeners=PLAINTEXT://:9091, log.dirs=/path/kafka-log1

Broker.id=1, Listeners=PLAINTEXT://:9092, log.dirs=/path/kafka-log2

***Started zookeeper:***

bin/zookeeper-server-start.sh config/zookeeper.properties

***Started Server***

bin/kafka-server-start.sh config/server.properties

bin/kafka-server-start.sh config/server1.properties

***Created topics named "Airports",” Carriers”,” Planedate”,” OTP” with replication-factor 2 and partitions 2***

bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 2 --partitions 2 --topic Airports

bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 2 --partitions 2 --topic Carriers

bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 2 --partitions 2 --topic Planedate

bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 2 --partitions 2 --topic OTP

***Ran the Producer for all the Topics which I downloaded and save in my local disc***

kafka-console-producer.sh --broker-list localhost:9092, localhost:9091 --topic Airports < Airports.csv

kafka-console-producer.sh --broker-list localhost:9092, localhost:9091 --topic Airports < Carriers.csv

kafka-console-producer.sh --broker-list localhost:9092, localhost:9091 --topic Airports < Planedate.csv

***Listened by the Consumer:***

bin/kafka-console-consumer.sh --bootstrap-server localhost:9091, localhost:9092 --topic Airports --from-beginning

Note: For loading the data to Kafka topics, we can also use Kafka connect

* 1. **Batch Ingestion (HDFS)**
* **Raw layer (Store data AS-IS)**

***Use* Apache Flume *to consume messages from Airports & Planedate Kafka Topic to HDFS Raw folder***

bin/flume-ng agent -n $agent\_name -c conf -f conf/flume-conf.properties.templats

bin/flume-ng agent --conf-file kafka.conf --name agent1

agent1.sources = kafka-source

agent1.channels = memory-channel

agent1.sinks = hdfs-sink

agent1.sources.kafka-source.type = org.apache.flume.source.kafka.KafkaSource

agent1.sources.kafka-source.zookeeperConnect = localhost:2181

agent1.sources.kafka-source.topic = Airports

agent1.sources.kafka-source.channels = memory-channel

agent1.sources.kafka-source.interceptors = i1

agent1.sources.kafka-source.interceptors.i1.type = timestamp

agent1.sources.kafka-source.kafka.consumer.timeout.ms = 100

agent1.channels.memory-channel.type = memory

agent1.channels.memory-channel.capacity = 10000

agent1.channels.memory-channel.transactionCapacity = 1000

agent1.sinks.hdfs-sink.type = hdfs

agent1.sinks.hdfs-sink.hdfs.path = \*\*\*PATH\*\*Emirates/Raw/data/raw

agent1.sinks.hdfs-sink.hdfs.rollInterval = 5

agent1.sinks.hdfs-sink.hdfs.rollSize = 0

agent1.sinks.hdfs-sink.hdfs.rollCount = 0

agent1.sinks.hdfs-sink.hdfs.fileType = DataStream

agent1.sinks.hdfs-sink.channel = memory-channel

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |