**Tech stack** : apache Kafka, apache spark , python , pyspark , cmd , sparksql , mongodb and elastic search

**Problem statement** : Create a data pipeline for ingesting clickstream data that will be stored in mongodb in a particular schema of unique row identifier and column familiar and then we will process the data using spark and store it in elasticsearch.

**Assumption:**

1. Ingesting the data using a python script with the help of producer
2. Using python to store the ingested data in mongodb ( for larger dataset we can use pyspark for the same.
3. Assuming the data processing will be done every hour using the latest offset in the data so as to know how much data to be processed
4. All the servers were running in my local system.

**Approach:**

For ingesting the data I used a python script which takes the data from a list which ,in real world scenario we will be using api’s , or other data sources like sensors or logs directly from the web ui so the script can be modified for specific use cases.

Storing the data in mongodb from kafka consumer was also done using python script. The unique row identifier is the offset of each event which is unique and the data is stored in index and collection where the index is the unique identifier and we have 3 collections in the database of 3 column families.

The last step is to read the data from mongodb, process it and load it to elasticsearch. For this pyspark is used the data is read and transformed using spark sql and the using elasticsearch sdk we send the data with the help of mapping to elastic search as desired

For scheduling the job we can use cronjob or apache airflow which can have a dependency tree on the consumermongodb script and we can schedule the DAG to run in an hourly manner or according to the use case.

**Note:**

1. We can directly use spark streaming to process the ingested data in kafka and store it directly in elastic search , removing the data store/data lake issue hence reducing the cost and infrastructure setup of the datastore or lake
2. The code snippet i provided are py scripts which will need to be running the producer and consumer script so as to live stream the data to mongodb.
3. Then using airflow we can schedule the processing script.
4. We can also use docker-compose for creating the servers and pipeline.
5. The code can be modularized using a functional approach or can be converted to object oriented using class.