**1. Spark JSON Analysis:**

**Tools Used**: (Flume, Kafka, HDFS, Spark Sql, SBT)

Spark SQL provides a natural syntax for querying JSON data along with automatic inference of JSON schemas for both reading and writing data. Spark SQL understands the nested fields in JSON data and allows users to directly access these fields without any explicit transformations.

To query a JSON dataset in Spark SQL, one only needs to point Spark SQL to the location of the data. The schema of the dataset is inferred and natively available without any user specification. In the programmatic APIs, it can be done through jsonFile and jsonRDD methods provided by SQLContext. With these two methods, we can create a SchemaRDD for a given JSON dataset and then we can register the SchemaRDD as a table.

The result of a SQL query can be used directly and immediately by other data analytic tasks, for example a machine learning pipeline. Also, JSON datasets can be easily cached in Spark SQL’s built in in-memory columnar store and be save in other formats such as Parquet or Avro.

In Spark SQL, SchemaRDDs can be output in JSON format through the toJSON method. Because a SchemaRDD always contains a schema (including support for nested and complex types), Spark SQL can automatically convert the dataset to JSON without any need for user-defined formatting. SchemaRDDs can themselves be created from many types of data sources, including Apache Hive tables, Parquet files, JDBC, Avro file, or as the result of queries on existing SchemaRDDs. This combination means users can migrate data into JSON format with minimal effort, regardless of the origin of the data source.

**Example**:







