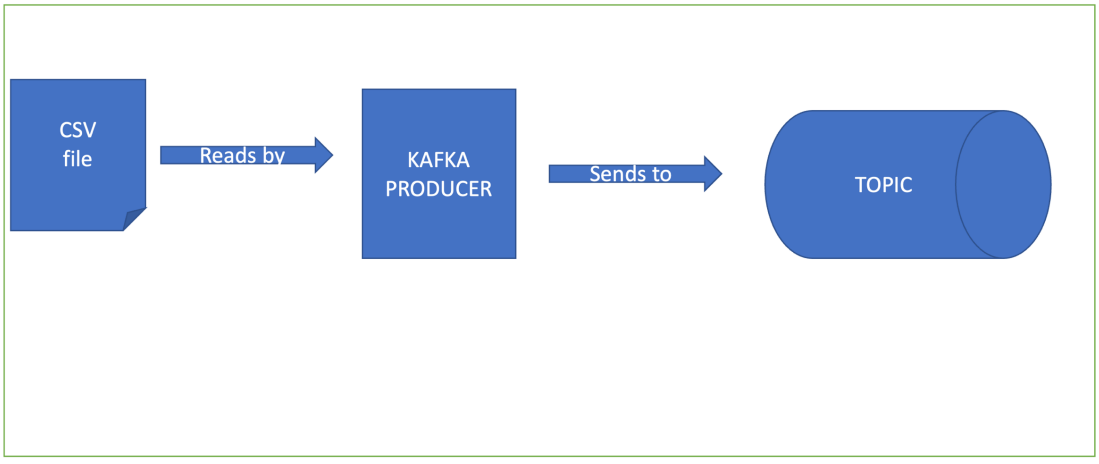
# Producer

- ❑ Producer reads data from CSV file which contains Hospital Information
- ❑ All the text information read from CSV file is converted into list of hospital class.
- ❑ At a time one hospital data is sent to the topic of Kafka
- ❑ After sending each data some delay is made to simulate the real time streaming

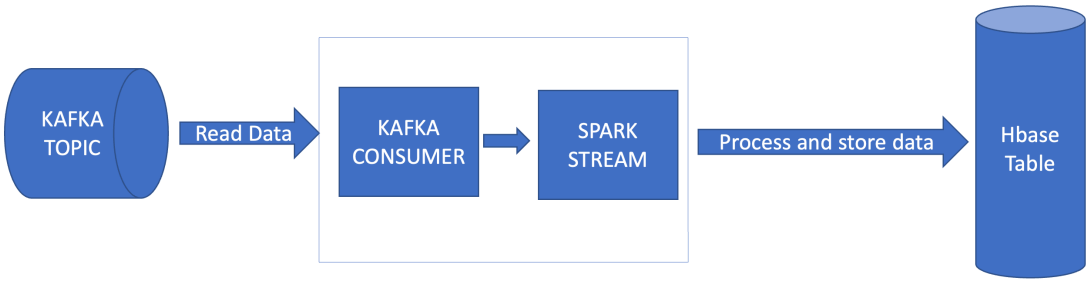
## Flow Diagram



# Consumer

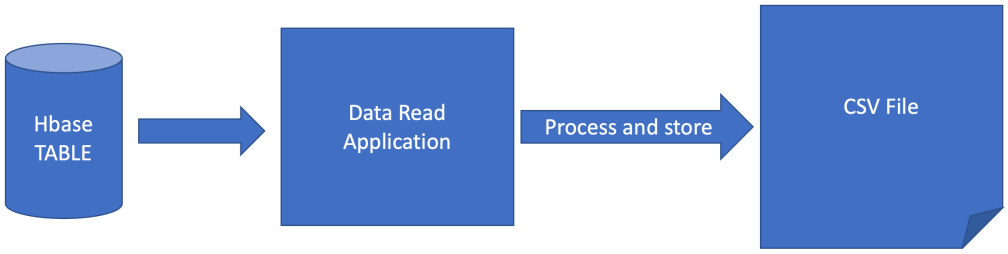
- Consumer is subscribed to the topics and waiting for the data.
- Consumer gets the data from Kafka topics which is produced by the producer.
- We can pass the state from command line parameters while running the consumer. It is optional parameter. So, if we passed the state then it will create two table and in one table it will store the hospital information belongs in that state and in another table, it will store other hospital information.
- Inside the consumer I have used the Spark Streaming to process hospital information and store into the Hbase Table

# Flow Diagram



Data Read Application

- ❑ Read the  data from Hbase Table
- ❑Write data in the CSV file and store it into the /user/cloudera location




# Commands:

**1. Zookeeper Server:** bin/zookeeper-server-start.sh config/zookeeper.properties

**2. Kafka Server:** bin/kafka-server-start.sh config/server.properties

**3. Create Topic:**

bin/kafka-topics.sh --create --topic info_hospital --bootstrap-server localhost:9092

**4. Running Applications:**

Consumer Application:

Building the project using maven and creating jar file and run it using command:

java –jar {filename}.jar {state-name(optional)}

Producer Application:

java –jar {filename}.jar { location of file (required))}

Fetch Data Application:

java –jar {filename}.jar { table-name(required))}