

DATA FLOW DIAGRAM

// ------------ PART 1: PREPARE

I. Install Kafka

1. Download Kafka:

- Latest version: https://kafka.apache.org/downloads

- Project version: https://archive.apache.org/dist/kafka/2.4.1/kafka\_2.12-2.4.1.tgz

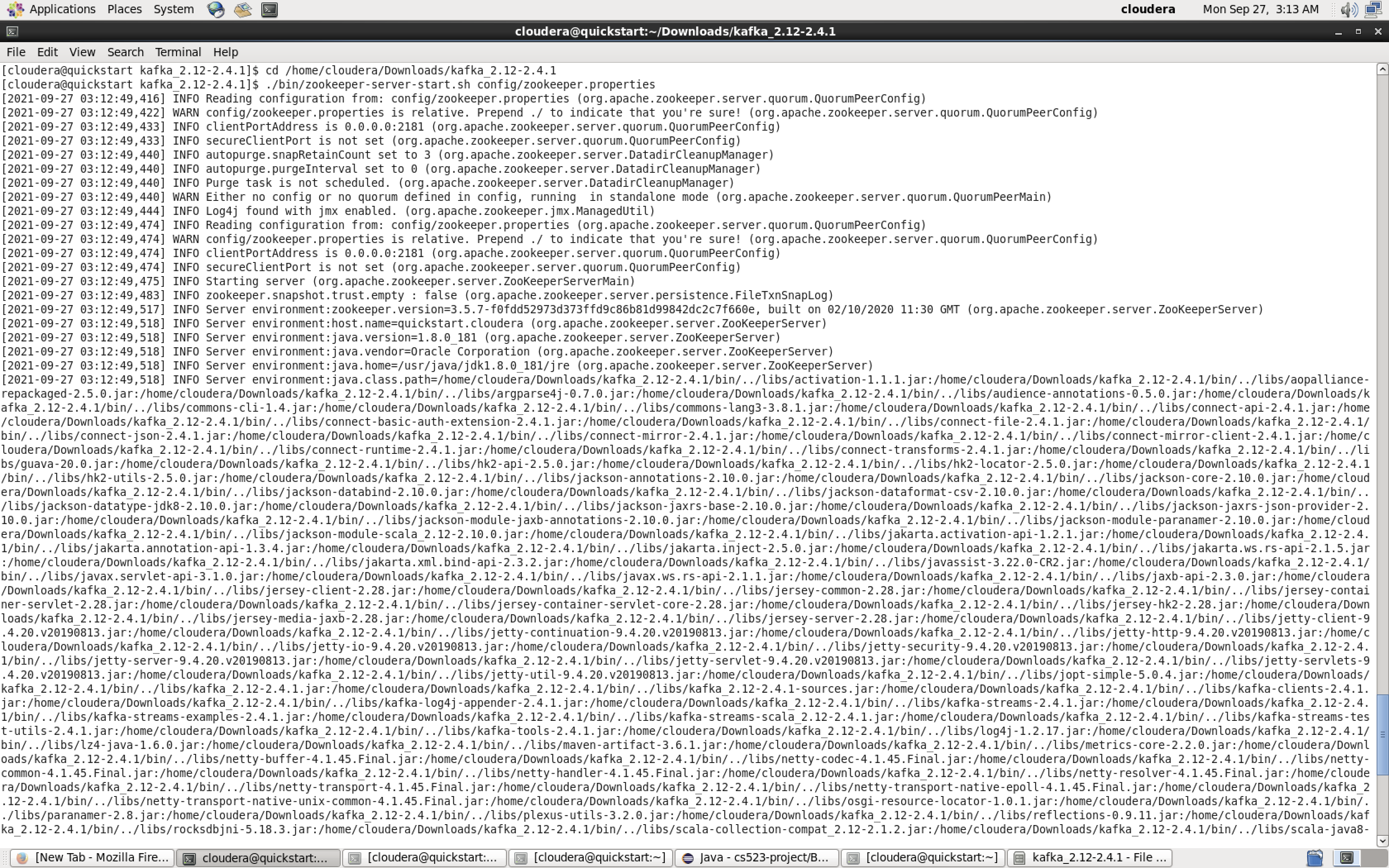
2. Extract and change current directory to

$ cd /home/cloudera/Downloads/kafka\_2.12-2.4.1

II. Start Zookeeper and Kafka

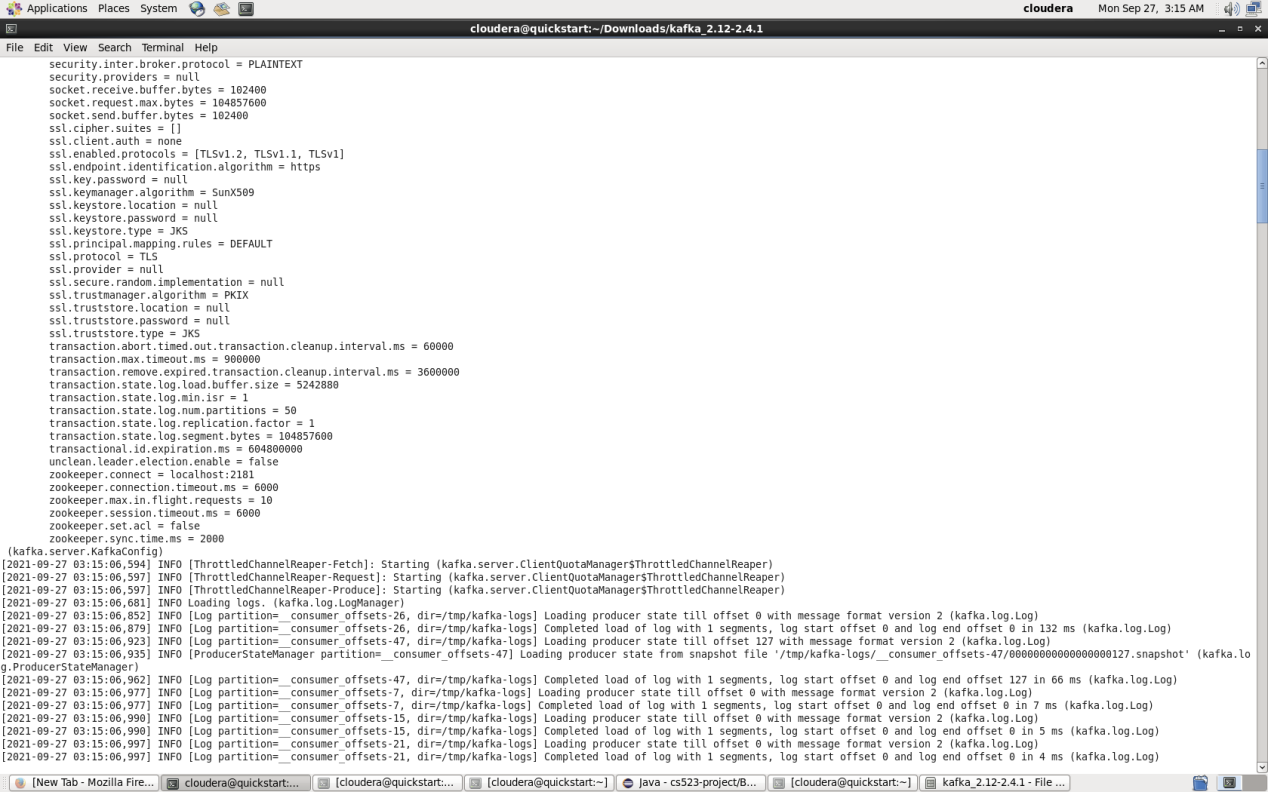
1. Run properties files:

1.1 $ ./bin/zookeeper-server-start.sh config/zookeeper.properties



Screen: Start Zookeeper

1.2 $ ./bin/kafka-server-start.sh config/server.properties



Screen: Start Kafka Server

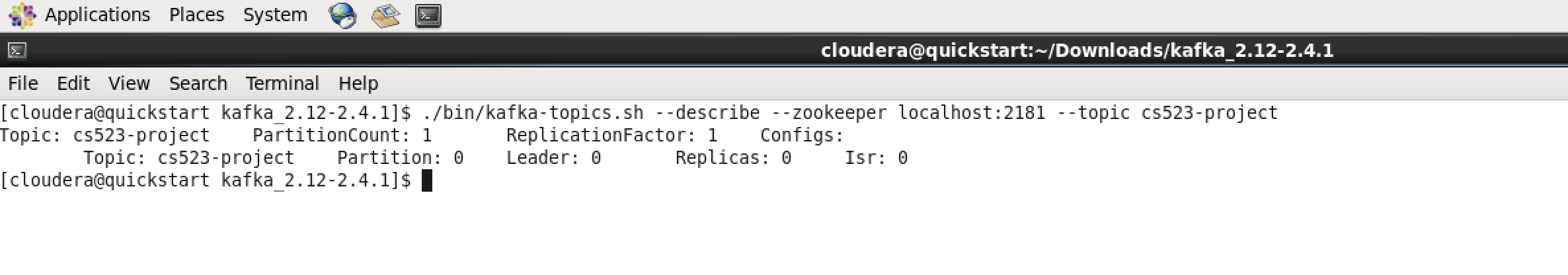
2. Create topic

$ ./bin/kafka-topics.sh --create --zookeeper localhost:2181 --topic cs523-project --replication-factor 1 --partitions 1

3. Check topic existed: cs523-project

3.1 $ ./bin/kafka-topics.sh --list --zookeeper localhost:2181

3.2 $ ./bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic cs523-project



III. Run Kafka Producer (Terminal 1)

1. Start console shell (Terminal)

2. Run command

$ ./bin/kafka-console-producer.sh --broker-list localhost:9092 --topic cs523-project

IV. Run Kafka Consumer (Terminal 2)

1. Start console shell (Terminal)

2. Run command

$ ./bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic cs523-project --from-beginning

V. Notes - If Can not create HBase table

1. Check HBase Service Start.

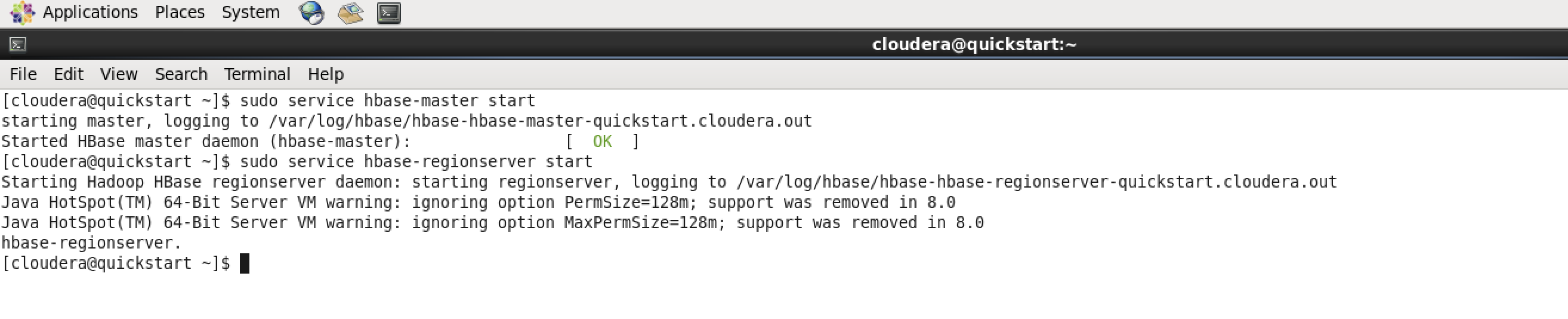
- $ service --status-all

2. If not start --> Need re-start

// --Start HBase Service.

- $ sudo service hbase-master start

- $ sudo service hbase-regionserver start



Screen: Start HBase

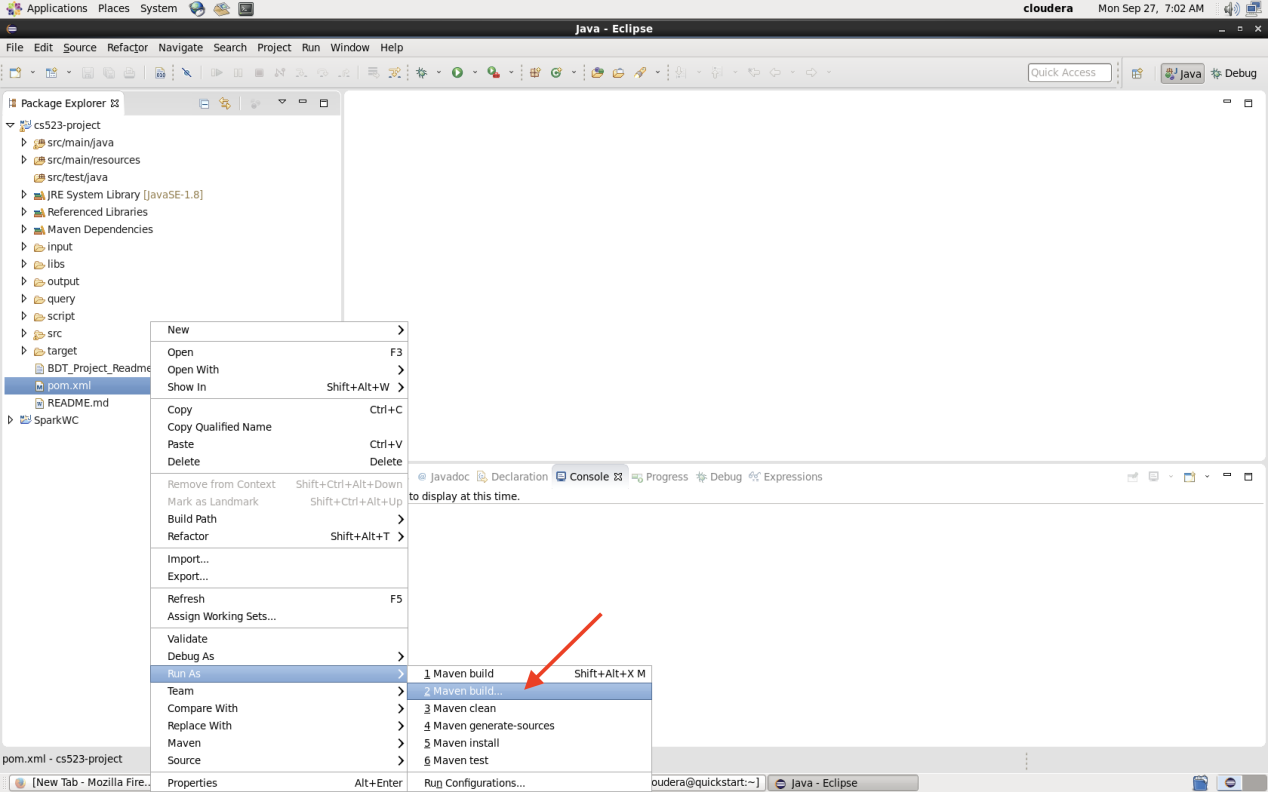
// ------------ PART 2: Run pseudo Mode

I. Build jar file: In Eclipse

1. In eclipse project

- Right click on pom.xml file

- Choose: Run As --> Maven Build …



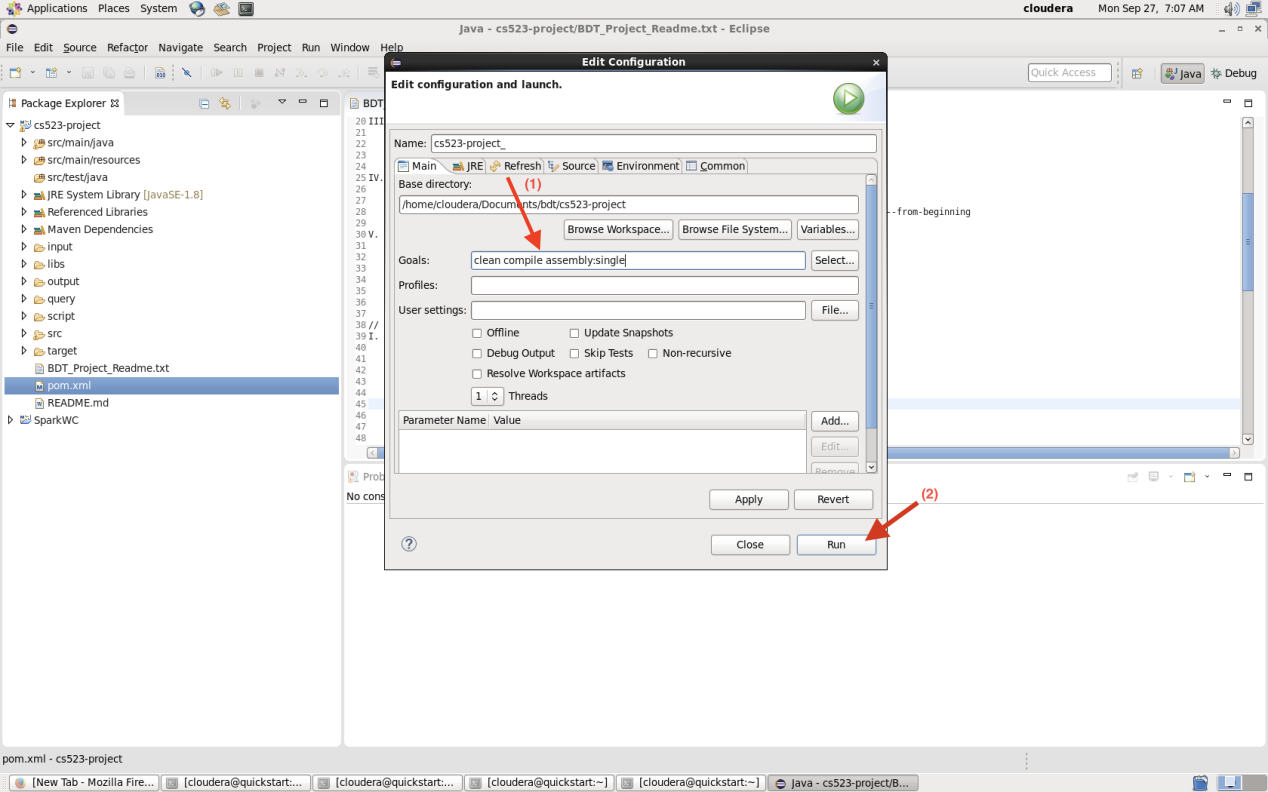
Screen: Choose Maven Build

2. in Edit Configuration dialog:

- tab Main

- Set field Goals = clean compile assembly:single

- Click Run

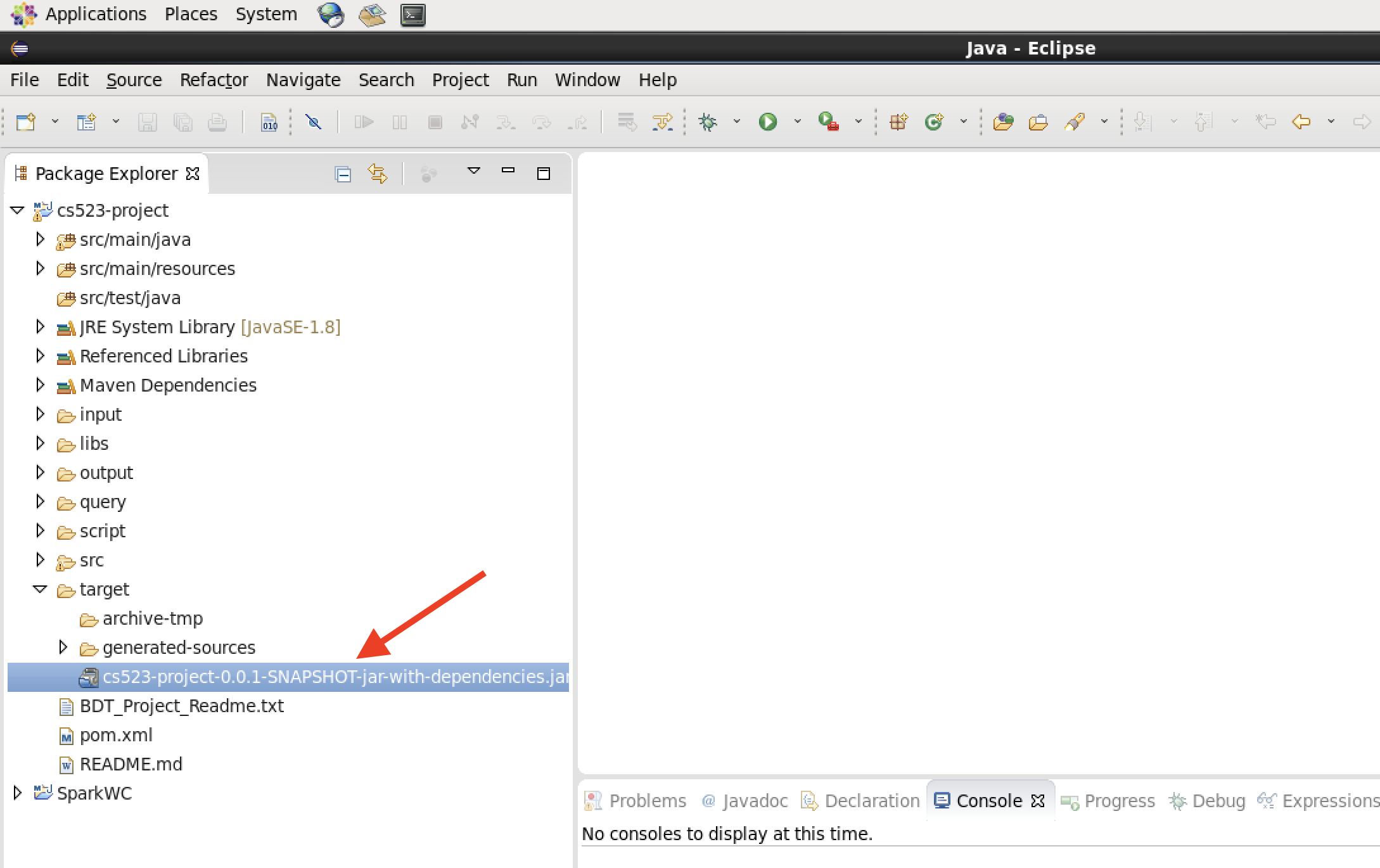


Screen: Edit Configuration

3. After build successful:

- Refresh project

- See file target/cs523-project-0.0.1-SNAPSHOT-jar-with-dependencies.jar



Screen: After build Jar Runable

II. Setup to Run Kafka Listener Producer and Spark Streaming

1. Start Terminal and change directory to:

$ cd /home/cloudera/Documents/bdt/cs523-project

2. Run Kafka Listener Producer: in Terminal 1

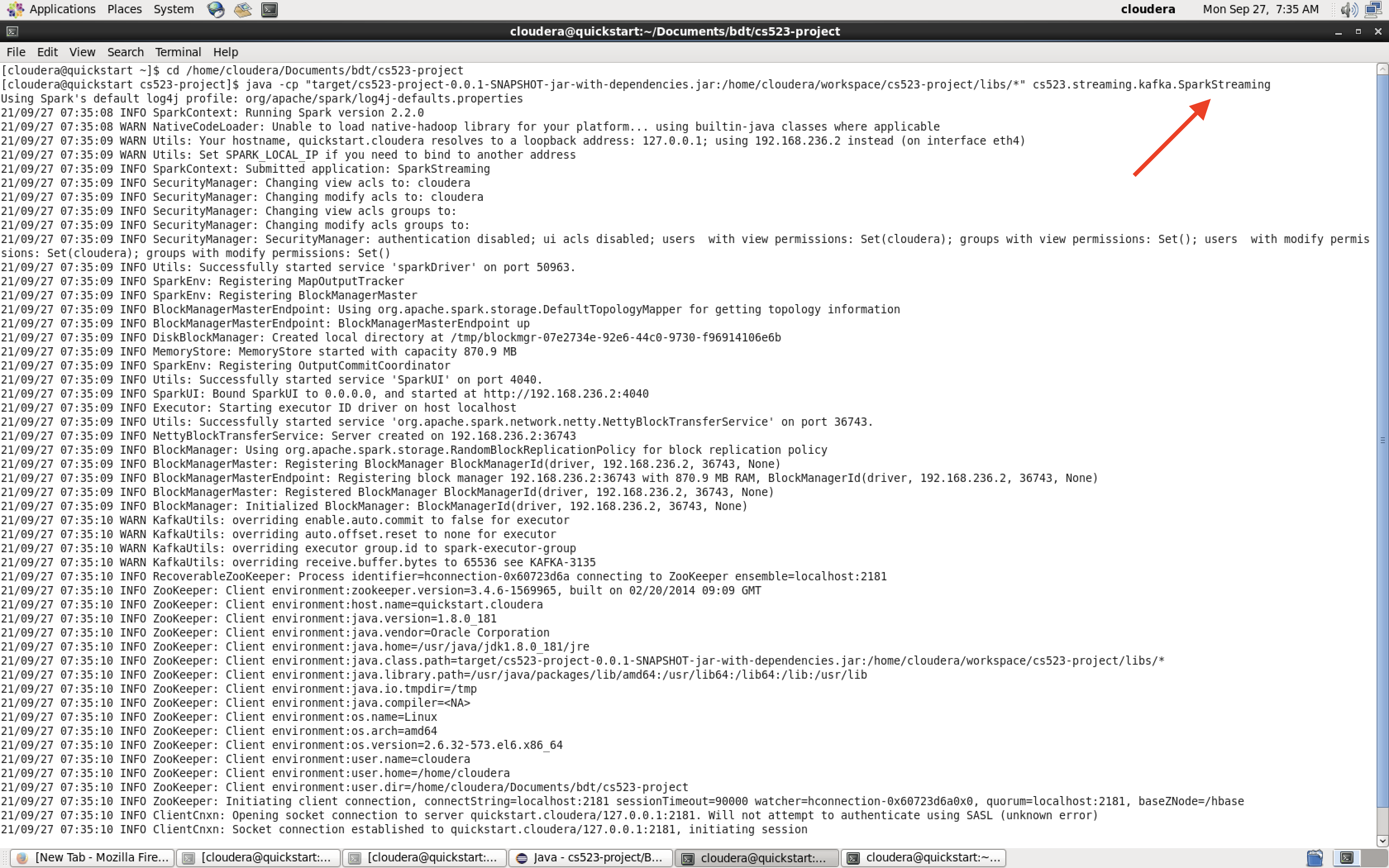
$ java -cp "target/cs523-project-0.0.1-SNAPSHOT-jar-with-dependencies.jar:/home/cloudera/workspace/cs523-project/libs/\*" cs523.utils.DataListener



Scree: Kafka Listener Producer

3. Run Spark Streaming Consume: in Terminal 2

$ java -cp "target/cs523-project-0.0.1-SNAPSHOT-jar-with-dependencies.jar:/home/cloudera/workspace/cs523-project/libs/\*" cs523.streaming.kafka.SparkStreaming



Screen: Start Spark Streaming Consumer

4. After step II: System now ready for receiving data and processing data:

The data flow as below:

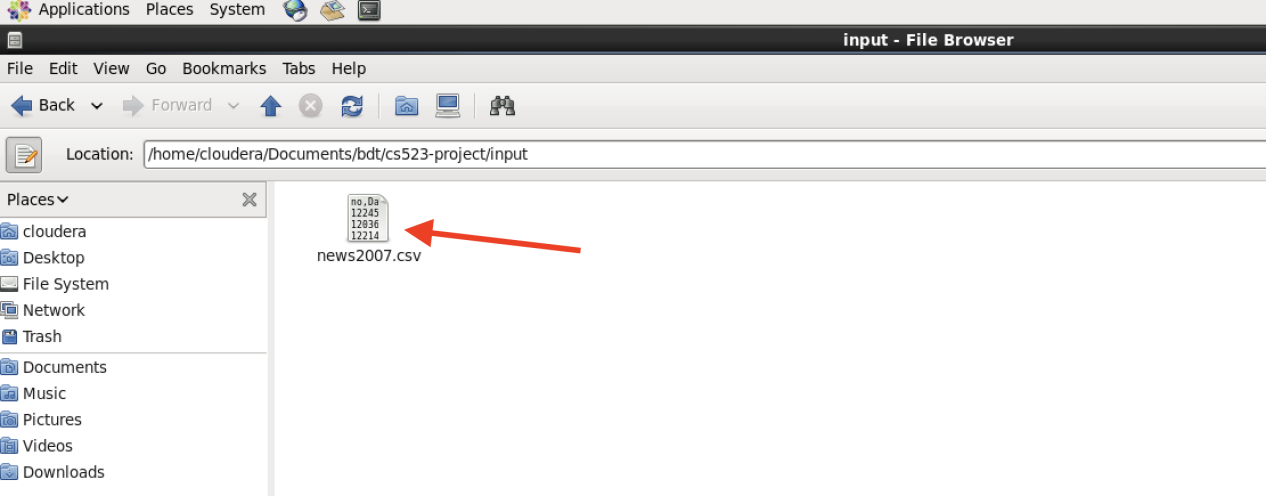
- Input(data file) --> Tunnel (Topic) --> Streaming ---> Hive Table

III. Step to Process Data.

1. Copy data to <input> folder

- This data maybe from real time web-site: Such as Forex Factory or Yahoo Final

- In this project was used static data set from Forex Factory web-site. File format is .csv

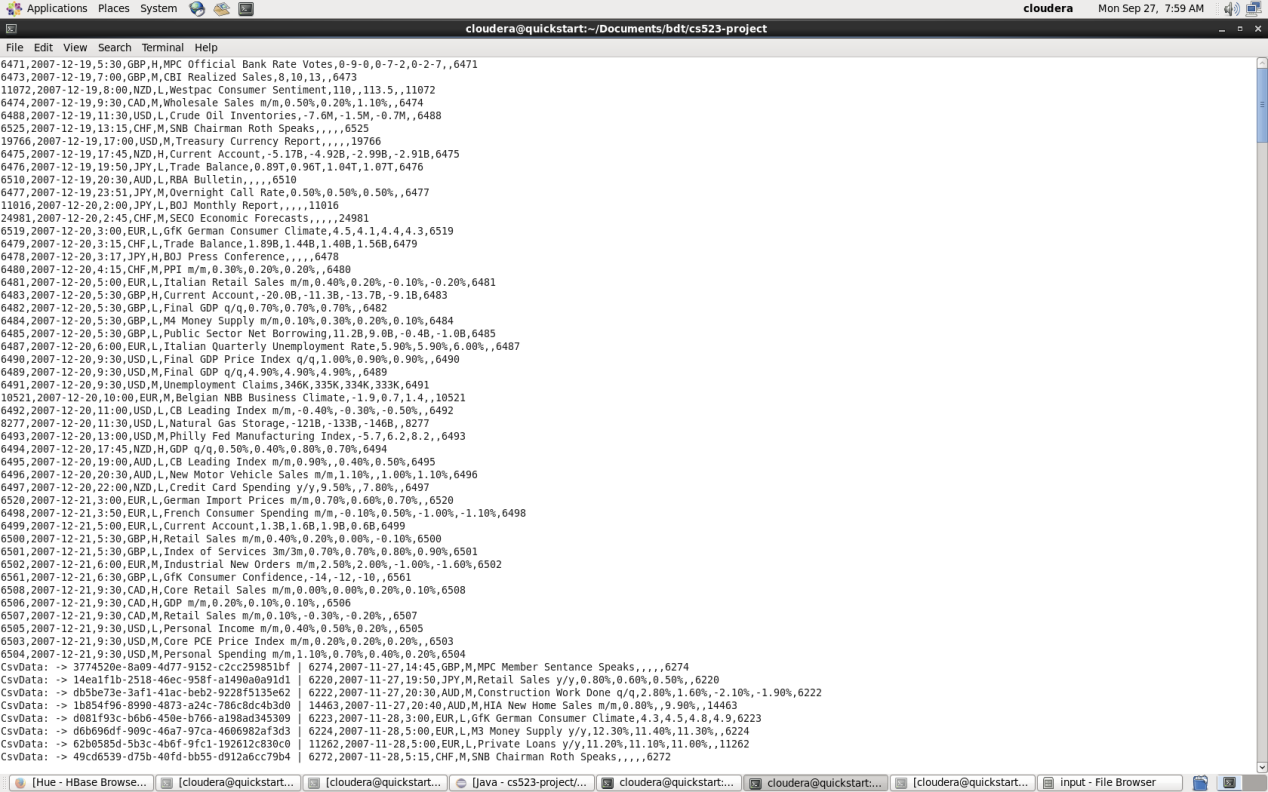


Screen: Data in <input> folder

2. Look at Terminal 1 - Kafka Listener Producer

- Any data change in <input> folder

- The KafkaListenerProducer application will read, parse and produce new data



Screen: Kafka Listener Producer

3. Look at Terminal 2 - Spark Streaming Consumer

- Base on the topic name.

- This application will communicate with Kafka Producer application to get the data

- After that processing data and update the data to Hive table

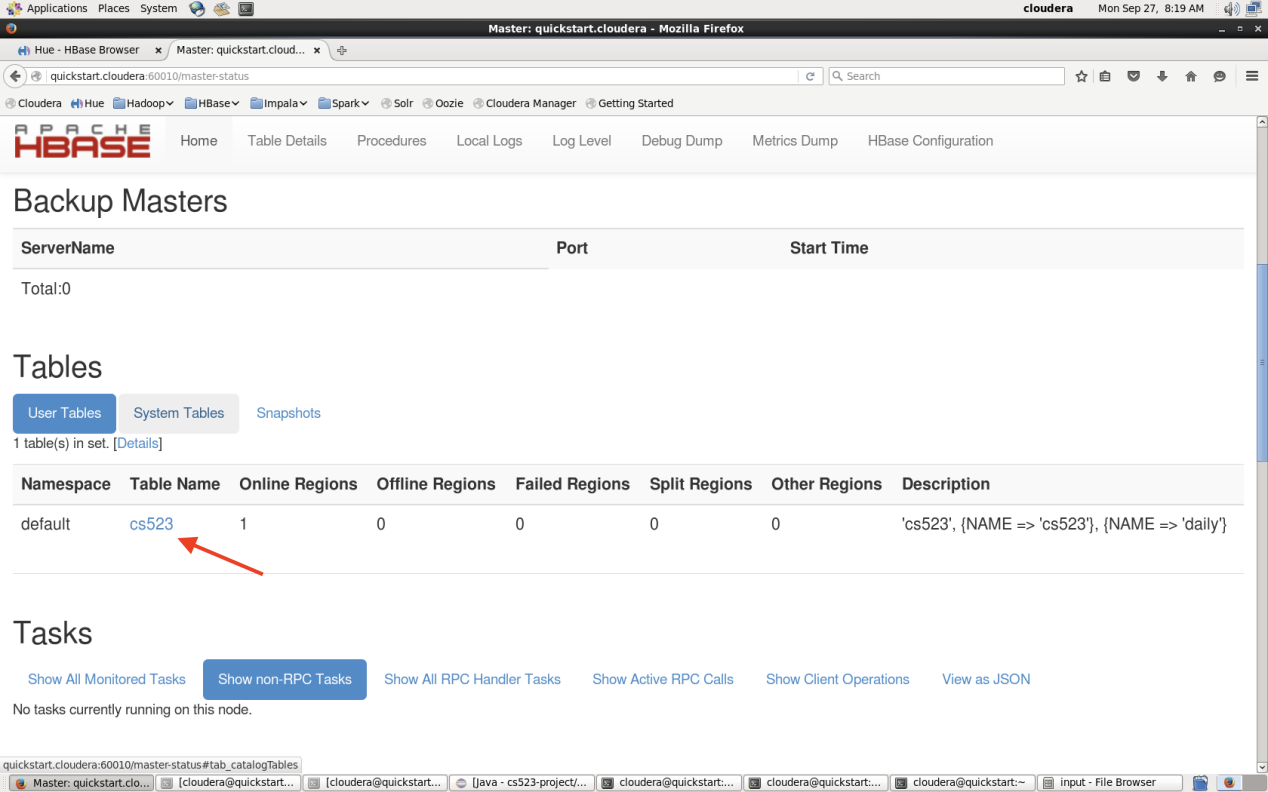


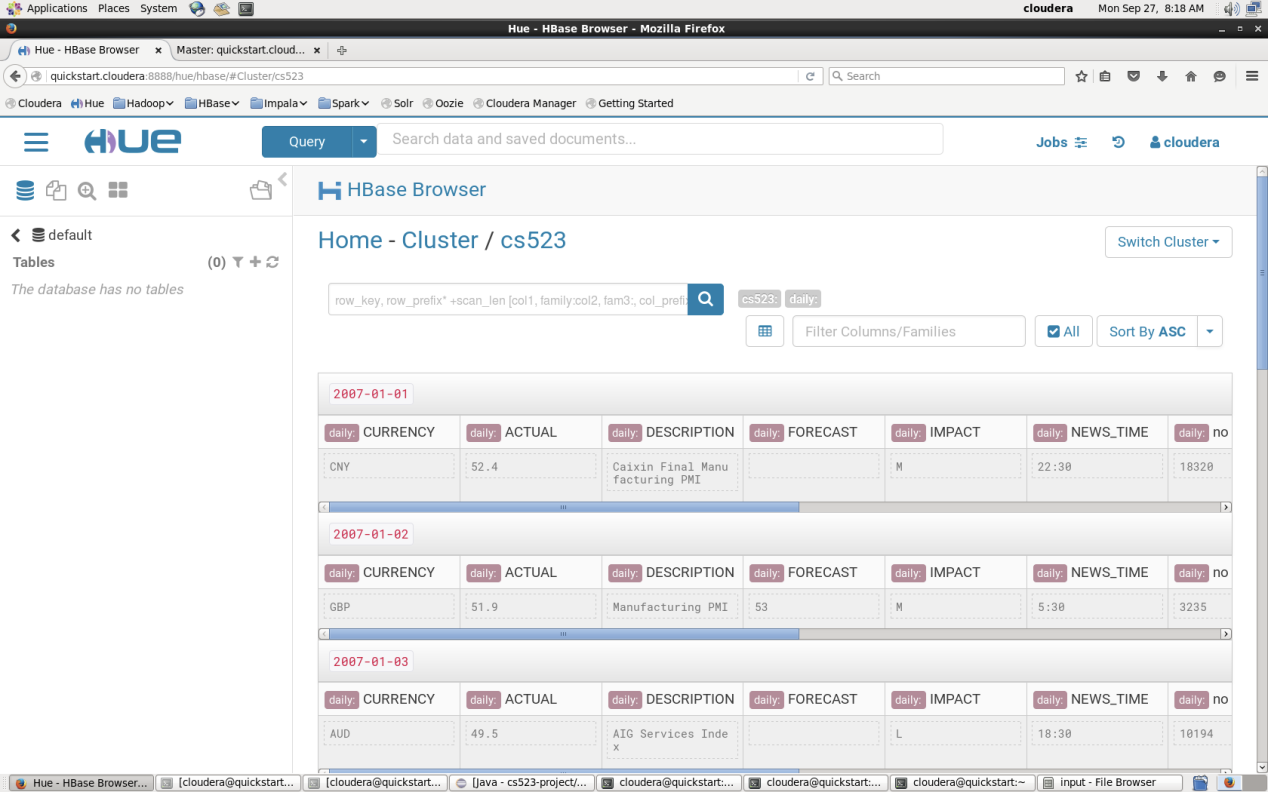
Screen: Spark Streaming Producer

4. Look at HBase Server and Hive Table.

- At first time the Spark Streaming Application will create table name <cs523>

- Refresh the cs523 table in Hive to see the data





Screen: Data in HBase and Hive table.

IV. What is the using of this data.

- This data very useful for technical analyze.

- Traders can base on that to make decision for trading.

- And base on these techniques I can apply to another project with real time data

and another scope such as predict weather, populations, ...

END.