## Python Examples

### Demo – Orders Data

**Terminal #1**

su - hdoop

ssh localhost

cd $HADOOP\_HOME/sbin

start-dfs.sh

start-yarn.sh

cd $KAFKA\_HOME

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

**Terminal #2**

su - hdoop

cd $KAFKA\_HOME

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

**Terminal #3**

su - hdoop

cd $KAFKA\_HOME

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

**Terminal #4**

su - hdoop

cd $KAFKA\_HOME

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

**Terminal #5**

su - hdoop

cd SparkSamples/OrderData

mkdir /tmp/spark-events

spark-submit --packages org.apache.spark:spark-streaming-kafka-0-8-assembly\_2.11:2.4.4 kafka\_streaming\_json\_demo.py

OR

spark-submit --packages org.apache.spark:spark-sql-kafka-0-10\_2.12:3.1.2 kafka\_streaming\_json\_demo.py

**Terminal #3**

cd SparkSamples/OrderData

python3 kafka\_producer.py

### Demo – Orders Data – AVRO

All steps remain the same, except running the producer and streaming (consumer) programs:

**New Terminal**

su - hdoop

cd SparkSamples/OrderData

mkdir /tmp/spark-events

spark-submit --packages org.apache.spark:spark-sql-kafka-0-10\_2.12:3.1.2,org.apache.spark:spark-avro\_2.12:3.1.2 kafka\_streaming\_avro\_demo.py

**New Terminal**

cd SparkSamples/OrderData

python3 kafka\_producer\_avro.py

### Demo – Orders Data - CSV

All steps remain the same, except running the producer and streaming (consumer) programs:

**New Terminal**

su - hdoop

cd SparkSamples/OrderData

mkdir /tmp/spark-events

spark-submit --packages org.apache.spark:spark-sql-kafka-0-10\_2.12:3.1.2,org.apache.spark:spark-avro\_2.12:3.1.2 kafka\_streaming\_csv\_demo.py

**New Terminal**

cd SparkSamples/OrderData

python3 kafka\_producer\_csv.py