The Word Count tutorial for Kafka Streams comes from Apache Kafka build. Unfortunately, there are some changes that need to be made to ensure these tutorials work correctly with HDF 3.3. Following are the steps.

* We are going to use a Kafka Streams Maven Archetype for creating a Streams project structure with the following commands

|  |
| --- |
| mvn archetype:generate \  -DarchetypeGroupId=org.apache.kafka \  -DarchetypeArtifactId=streams-quickstart-java \  -DarchetypeVersion=2.1.0 \  -DgroupId=streams.examples \  -DartifactId=streams.examples \  -Dversion=0.1 \  -Dpackage=myapps |

* The pom.xml file included in the project already has the Streams dependency defined, and there are already several example programs written with Streams library under src/main/java.

|  |
| --- |
| > cd streams.examples |

* Edit WordCount.java from

|  |
| --- |
| Properties props = new Properties(); props.put(StreamsConfig.APPLICATION\_ID\_CONFIG, "streams-pipe"); props.put(StreamsConfig.BOOTSTRAP\_SERVERS\_CONFIG, "localhost:9092"); |

to below. Use Broker’s hostname here.

|  |
| --- |
| Properties props = new Properties(); props.put(StreamsConfig.APPLICATION\_ID\_CONFIG, "streams-pipe"); props.put(StreamsConfig.BOOTSTRAP\_SERVERS\_CONFIG, "<hostname>:6667"); |

* Create 2 Topics streams-plaintext-input. Substitute your host name or IP

|  |
| --- |
| /usr/hdf/current/kafka-broker/bin/kafka-topics.sh --create \  --zookeeper 13.56.151.119:2181 \  --replication-factor 1 \  --partitions 1 \  --topic streams-plaintext-input |

and streams-wordcount-output. Substitute your host name or IP

|  |
| --- |
| /usr/hdf/current/kafka-broker/bin/kafka-topics.sh --create \  --zookeeper 13.56.151.119:2181 \  --replication-factor 1 \  --partitions 1 \  --topic streams-wordcount-output \  --config cleanup.policy=compact |

* Describe the topics. Substitute your host name or IP

|  |
| --- |
| /usr/hdf/current/kafka-broker/bin/kafka-topics.sh --zookeeper 13.56.151.119:2181 --describe |

* Set up Maven.

|  |
| --- |
| sudo yum install maven |

* Check Maven version

|  |
| --- |
| mvn --version |

* There are some dependencies that need to be added right at the end to pom.xml

|  |
| --- |
| <**dependencies**>  <!-- Apache Kafka dependencies -->  <**dependency**>  <**groupId**>org.apache.kafka</**groupId**>  <**artifactId**>kafka-streams</**artifactId**>  <**version**>${kafka.version}</**version**>  </**dependency**> <!-- https://mvnrepository.com/artifact/org.slf4j/slf4j-simple --> <**dependency**>  <**groupId**>org.slf4j</**groupId**>  <**artifactId**>slf4j-simple</**artifactId**>  <**version**>1.7.7</**version**>  <**scope**>test</**scope**> </**dependency**> <!-- logging dependencies -->  <**dependency**>  <**groupId**>ch.qos.logback</**groupId**>  <**artifactId**>logback-classic</**artifactId**>  <**version**>1.1.1</**version**>  <**exclusions**>  <**exclusion**>  <!-- Defined below -->  <**artifactId**>slf4j-api</**artifactId**>  <**groupId**>org.slf4j</**groupId**>  </**exclusion**>  </**exclusions**>  </**dependency**> <!-- https://mvnrepository.com/artifact/org.slf4j/slf4j-api --> <**dependency**>  <**groupId**>org.slf4j</**groupId**>  <**artifactId**>slf4j-api</**artifactId**>  <**version**>1.7.7</**version**> </**dependency**>  </**dependencies**> |

While you are in the pom.xml directory run the following

|  |
| --- |
| mvn clean package mvn exec:java -Dexec.mainClass=myapps.WordCount |

if you run into any dependency issues, make sure to remove .m2 directory, fix your issue and then run the previous steps again.

Start Producer

|  |
| --- |
| /usr/hdf/current/kafka-broker/bin/kafka-console-producer.sh --broker-list 54.215.225.243:6667 --topic streams-plaintext-input |

Start Consumer

|  |
| --- |
| /usr/hdf/current/kafka-broker/bin/kafka-console-consumer.sh --bootstrap-server 54.215.225.243:6667 \  --topic streams-wordcount-output \  --from-beginning \  --formatter kafka.tools.DefaultMessageFormatter \  --property print.key=true \  --property print.value=true \  --property key.deserializer=org.apache.kafka.common.serialization.StringDeserializer \  --property value.deserializer=org.apache.kafka.common.serialization.LongDeserializer |

While you enter messages in the Producer terminal, words and count associated with each is displayed. As words keep coming, word count increments hence acting as a stream of data and the counts are from the start of the queue. Here is a short video.

<https://youtu.be/fB9zaRkAUPc>

Image below shows 3 screens.

1 shows the kafka streams application started.

2. Producer generating messages.

3. Consumer showing words with word count. Count is maintained from the start of topic creation time.



WordCount example on CDH 1 node cluster.

|  |
| --- |
| [centos@ip-172-31-7-203 libs]$ pwd  /opt/cloudera/parcels/CDH-6.2.0-1.cdh6.2.0.p0.967373/lib/kafka/libs/kafka-streams-examples-2.1.0-cdh6.2.0.jar |

Create 2 Topics streams-plaintext-input. Substitute your host name or IP

|  |
| --- |
| /opt/cloudera/parcels/CDH-6.2.0-1.cdh6.2.0.p0.967373/lib/kafka/bin/kafka-topics.sh --create \  --zookeeper localhost:2181 \  --replication-factor 1 \  --partitions 1 \  --topic streams-plaintext-input |

and streams-wordcount-output. Substitute your host name or IP

|  |
| --- |
| /opt/cloudera/parcels/CDH-6.2.0-1.cdh6.2.0.p0.967373/lib/kafka/bin/kafka-topics.sh --create \  --zookeeper localhost:2181 \  --replication-factor 1 \  --partitions 1 \  --topic streams-wordcount-output \  --config cleanup.policy=compact |

./.ivy2/cache/org.apache.kafka/kafka-clients/jars/kafka-clients-0.8.2.1.jar

Run word count example

|  |
| --- |
| Run an example application from the standalone jar. Here: `WordCountLambdaExample`.  $ export CLASSPATH = $CLASSPATH:./.ivy2/cache/org.apache.kafka/kafka-clients/jars/kafka-clients-0.8.2.1.jar:.:  $ java -cp /opt/cloudera/parcels/CDH-6.2.0-1.cdh6.2.0.p0.967373/lib/kafka/libs/kafka-streams-examples-2.1.0-cdh6.2.0.jar \  org.apache.kafka.streams.examples.wordcount.WordCountDemo |

Start Producer

|  |
| --- |
| /opt/cloudera/parcels/CDH-6.2.0-1.cdh6.2.0.p0.967373/lib/kafka/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic streams-plaintext-input |

Start Consumer

|  |
| --- |
| /opt/cloudera/parcels/CDH-6.2.0-1.cdh6.2.0.p0.967373/lib/kafka/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 \  --topic streams-wordcount-output \  --from-beginning \  --formatter kafka.tools.DefaultMessageFormatter \  --property print.key=true \  --property print.value=true \  --property key.deserializer=org.apache.kafka.common.serialization.StringDeserializer \  --property value.deserializer=org.apache.kafka.common.serialization.LongDeserializer |