LAB Stream 1-1 Kafka Word Count Example

Welcome to the session Stream 1-1 lab. The work for this lab is done in `~/kafka-training/stream-lab1`. In this lab, you are going to run the Kafka Word Count Example. You create two new Kafka topics, one called `streams-plaintext-input` and the other `streams-plaintext-output` You will then add some data to the input topic, start a consumer on the output topic so you can watch it, and then run the Word Count Demo

ACTION - START ZooKeeper and Kafka Broker if needed.

Only 1 broker is necessary.

Create Kafka Topics

- Create a topic named `streams-plaintext-input` with 1 partition and a replication factor of 1.
- Create a topic named `streams-wordcount-output` with 1 partition and a replication factor of 1.

ACTION - REVIEW `create-topics.sh`

~/kafka-training/stream-lab1/bin/create-topics.sh

```
#!/usr/bin/env bash
cd ~/kafka-training

## Create input topic
kafka/bin/kafka-topics.sh --create \
    --replication-factor 1 \
    --partitions 1 \
    --topic streams-plaintext-input \
    --zookeeper localhost:2181

## Create output topic
kafka/bin/kafka-topics.sh --create \
    --replication-factor 1 \
    --partitions 1 \
    --topic streams-wordcount-output \
    --zookeeper localhost:2181

## List created topics
kafka/bin/kafka-topics.sh --list \
    --zookeeper localhost:2181
```

ACTION - RUN `create-topics.sh` as follows:

```
$ cd ~/kafka-training/stream-lab1/bin
$ ./create-topics.sh
Created topic "streams-plaintext-input".
Created topic "streams-wordcount-output".
streams-plaintext-input
streams-wordcount-output
```

Add data to the topic

Now add data to the input topic by starting a console producer and entering data.

ACTION - REVIEW `start-producer-console-input.sh`

~/kafka-training/stream-lab1/bin/start-producer-console-input.sh

```
#!/usr/bin/env bash
cd ~/kafka-training

## Producer
kafka/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic streams-
plaintext-input
```

ACTION - RUN `start-producer-console-input.sh`

```
$ cd ~/kafka-training/stream-lab1
$ ./start-producer-console-input.sh
>
```

ACTION - Enter Data

Enter the following 3 lines of data. Ctrl-C to stop the console.

```
> stock streams MSFT
> stock data AAPL
> stock streams RHT
```

ACTION - Validate Data

Run a console consumer against the input topic to verify the data is there.

~/kafka-training/stream-lab1/start-consumer-console-input.sh

```
#!/usr/bin/env bash
cd ~/kafka-training

## Input Consumer
kafka/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic streams-
plaintext-input --from-beginning
```

ACTION - RUN `start-consumer-console-input.sh`

```
$ cd ~/kafka-training/stream-lab1/bin
$ ./start-consumer-console-input.sh
stock streams MSFT
stock data AAPL
stock streams RHT
```

```
<Ctrl-C>
Processed a total of 3 messages
```

Above, stops the consumer.

Run a console consumer against the output.

Run another consumer against the output topic so we can see the work that the Word Count Demo does. Notice a few things about this consumer. These match the way the topic is populated by the demo.

- It has a message formatter specified.
- It has key and value deserializers.

ACTION - EDIT `~/kafka-training/stream-lab1/bin/start-consumer-console-output.sh`, follow instructions in file.

~/kafka-training/stream-lab1/start-consumer-console-output.sh

```
#!/usr/bin/env bash
cd ~/kafka-training

## Output Consumer
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
```

ACTION - RUN `start-consumer-console-output.sh`

```
$ cd ~/kafka-training/stream-lab1/bin
$ ./start-consumer-console-output.sh
```

There will be no output until the demo code runs. Keep this window open so you can see it as the demo runs.

Run the Word Count demo

Kafka provides examples, one of which is the Word Count demo. It consumes, via the KStream dsl (java api) data from the input topic and writes to the output topic.

> *See the documentation and the java code for the Word Count demo in github*

Since the demo is included in the Kafka libraries, we can use Kafka's `kafka-run-class.sh` script to run it.

ACTION - REVIEW `run-word-count-demo.sh`

~/kafka-training/stream-lab1/run-word-count-demo.sh

```
#!/usr/bin/env bash
cd ~/kafka-training

kafka/bin/kafka-run-class.sh org.apache.kafka.streams.examples.wordcount.WordCountDemo
```

ACTION - RUN `run-word-count-demo.sh`

```
$ cd ~/kafka-training/stream-lab1/bin
$ ./run-word-count-demo.sh
```

Consumer Console Output after running run-word-count-demo.sh

```
stock   1
streams 1
msft    1
stock   2
data    1
aapl    1
stock   3
streams 2
rht     1
```

Conclusion Word Count example

The word count is the simplest example of stream processing of an input topic and writing results to a different then you created Kafka Producer in Java that uses the Kafka replicated topic to send records. You sent records with the Kafka Producer using async and sync send methods.

Review Word Count example

Related content

- What is Kafka?
- Kafka Architecture
- Kafka Topic Architecture
- Kafka Consumer Architecture
- Kafka Producer Architecture
- Kafka Architecture and low level design
- Kafka and Schema Registry
- Kafka and Avro
- Kafka Ecosystem
- Kafka vs. JMS
- Kafka versus Kinesis
- Kafka Tutorial: Using Kafka from the command line
- Kafka Tutorial: Kafka Broker Failover and Consumer Failover
- Kafka Tutorial
- Kafka Tutorial: Writing a Kafka Producer example in Java
- Kafka Tutorial: Writing a Kafka Consumer example in Java

- Kafka Architecture: Log Compaction

About Cloudurable