Awesome! Let’s set up **Kafka 0.8.2.2** step-by-step on your Cloudera QuickStart VM 5.13.0.

Step-by-Step: Install & Run Kafka 0.8.2.2 on Cloudera QuickStart VM

Step 1: Download Kafka 0.8.2.2

Open a terminal in your VM and run:

```bash
wget https://archive.apache.org/dist/kafka/0.8.2.2/kafka_2.10-0.8.2.2.tgz
```

Step 2: Extract Kafka package

```bash
tar -xvzf kafka_2.10-0.8.2.2.tgz
cd kafka_2.10-0.8.2.2
```

Step 3: Start Zookeeper (required by Kafka)

Kafka needs Zookeeper running.

Run Zookeeper:

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

Open a **new terminal** window/tab and continue with the next steps.

Step 4: Start Kafka Server

In the new terminal inside Kafka folder, run:

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

Kafka will start and listen on default port 9092.

Step 5: Create Kafka Topic

Create a topic named **`wordcount_topic`**:

```
bash

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

Verify the topic creation:

```
bash

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

You should see:

```
nginx

wordcount_topic
```

Step 6: Test Kafka Producer

Start a Kafka console producer to send messages to the topic:

```
bash

bin/kafka-console-producer.sh --broker-list localhost:9092 --topic wordcount_topic
```

Type some messages like:

```
kotlin

hello kafka
this is kafka and spark
kafka wordcount example
```

Press Enter after each line to send it.

Step 7: Test Kafka Consumer (optional)

Open a new terminal and run:

```
bash

bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic wordcount_topic --from-beginning
```

You will see the messages you typed in producer terminal.

Step 8: Run Spark Streaming Kafka WordCount Consumer

Now that Kafka is up and running, you can follow the earlier provided Spark Streaming wordcount example to consume from `wordcount_topic`.

Summary of running services in separate terminals:

Terminal	Command	Purpose
1	`bin/zookeeper-server-start.sh config/zookeeper.properties`	Start Zookeeper
2	`bin/kafka-server-start.sh config/server.properties`	Start Kafka server
3	`bin/kafka-console-producer.sh --broker-list localhost:9092 --topic wordcount_topic`	Send messages to Kafka
4	Spark Streaming wordcount app (your Scala/Java program)	Run your consumer job

If you want, I can help you write a ready-to-run **Scala Spark Streaming application** or show how to **build and submit** it with correct Kafka dependencies.

Ready for the next step? Or want help running the producer/consumer now?
❞ Ask ChatGPT