Setting up Kafka and Publishing some Data.

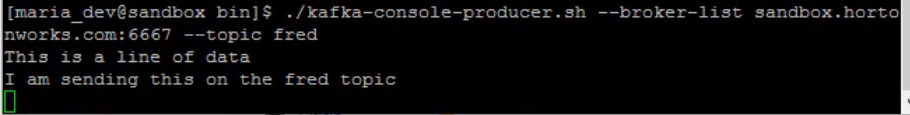
Start Kafka on our sandbox

1. Start Hortonworks Docker Sandbox
2. Go to Ambari web application and you will see Kafka Preinstalled and start the Kafka Service.
3. Login to Putty
   1. cd /usr/hdp/current/kafka-broker
   2. cd bin
   3. ls

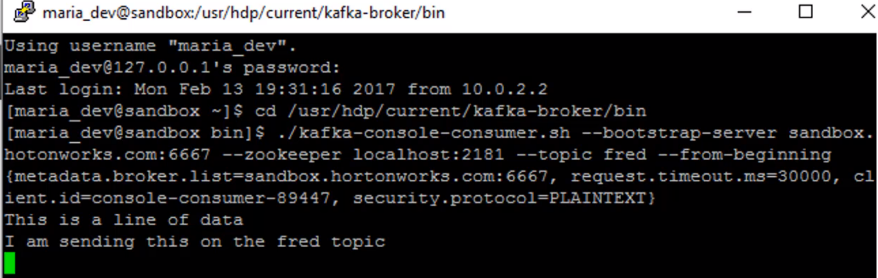
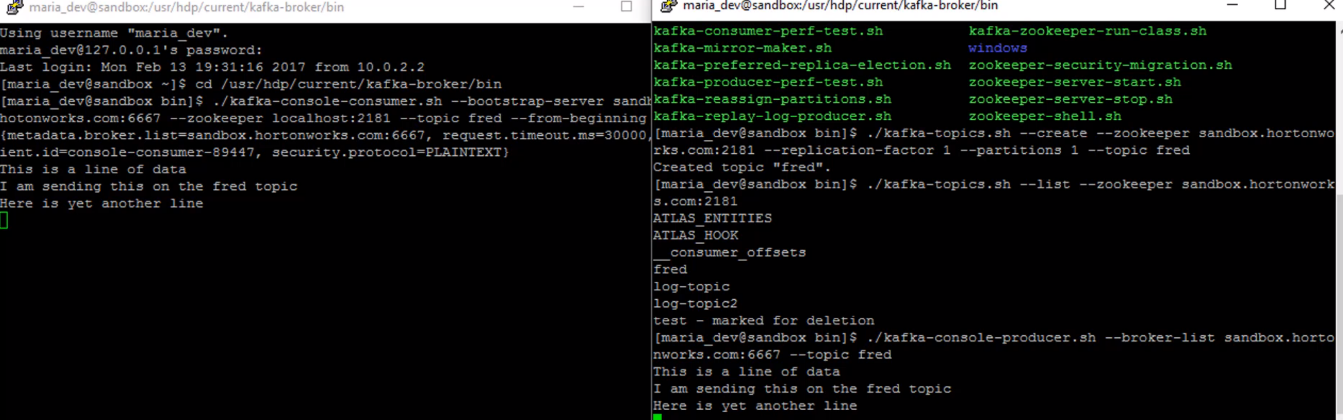
Set up a topic

1. ./kafka-topics.sh --create --zookeeper sandbox.hortonworks.com:2181 --replication-factor 1 --partitions 1 --topic <name>
2. ./kafka-topics.sh --list --zookeeper sandbox.hortonworks.com:2181

Publish some data

1. ./kafka-console-producer.sh --broker-list sandbox.hortonworks.com:6667 --topic fred(sample producer app, you might have your own app)
   1. It will see any lines that’s on standard input on the fred topic through Kafka
2. Then you can test and publish the data. Kafka is holding them onto it.

Get the messages back

1. Login to another session of Putty and change directory
   1. cd /usr/hdp/current/kafka-broker/bin
   2. ./kafka-console-consumer.sh --bootstrap-server sandbox.hortonworks.com:6667 --zookeeper localhost:2181 --topic fred --from-beginning
   3. You should see the two lines of data
2. You will see that its connected. It is very much scalable

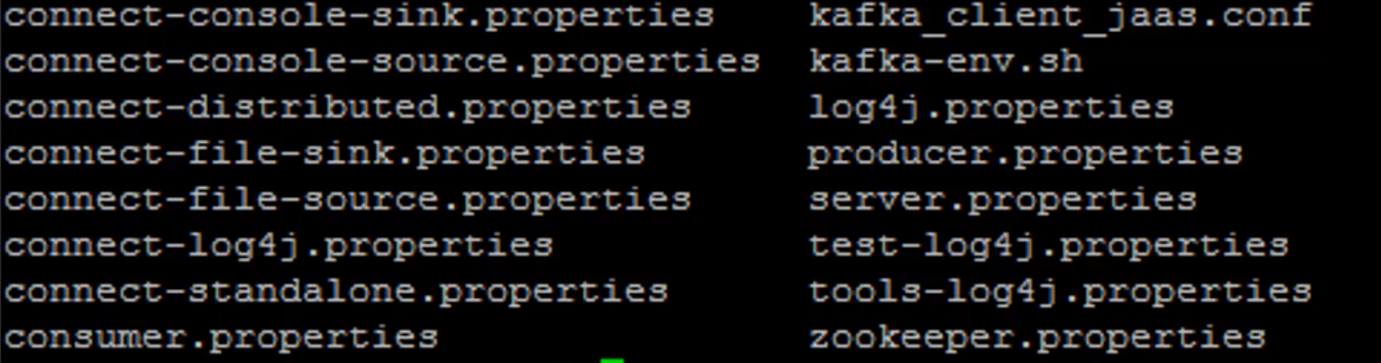
To quit, you Ctrl+C

Part 2: Closer to reality, use a Kafka connector to monitor a file and publish new lines on that file. Log Processing. Error Logs and Access logs. Publish those New Lines of Logs into one service and store it somewhere else.

It has file connector.

Pick one of our session.

1. Go out of the bin folder into the kafka-broker folder
2. cd into conf and ls to see built in Kafka connectors

connect-standlone.properties: set up the network environment for our standalone server

connect-file-sink.properties: store the content of the stream, consumer of that stream and dumping the output into a file of our choise

connect-file-source.properties: Listen to the changes to the file and publish that as a producer

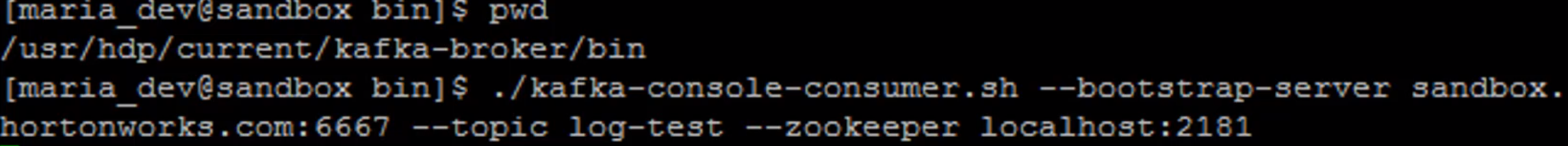
If you want to change with them you copy them into local disk

* + 1. cp connect-standalone.properties ~/
    2. cp connect-file-sink.properties ~/
    3. cp connect-file-source.properties ~/
    4. cd ~
    5. vi connect-standalone.properties
  1. main thing we change is the bootstrap.servies =sandbox.hortonworks.com:6667

1. vi connect-file-sink.properties
   1. file = /home/maria\_dev/logout.txt
   2. topics = log-test
2. vi connect-file-source.properties
   1. file = /home/maria\_dev/access\_log\_small.txt
   2. topic= log\_test
3. and Log out, by :wq
4. wget <http://media.sundog-soft.com/hadoop/access_log_small.txt>
5. less access\_log\_small.txt

Setup a consumer to listen on the side. Connector to publish and consume and watch it in real time.

1. Login to a new Putty
2. ./kafka-console-consumer.sh –bootstrap-server sandbox.hortonworks.com:6667 –topic log-test –zookeeper localhost:2181



Kick off our connecter that will listen to changes to the file and publish

1. cd /user/hdp/current/kafka-broker/bin

2. ./connect-standalone.sh ~/connect-standalone.properties ~/connect-file-sources.properties ~/connect-file-sink.properties

Open up a 3rd Window putty

1. cd ~ and ls

2. less logout.txt (Connector consumer that is publishing data to a file somewhere)

Add a line to our access log

echo ‘This is a new line’ >> access\_log\_small.txt