**Below steps describes how to Create Kafka topics, how to write kafka producers for topic and how to consume data.**

* Start ZooKeeper and Kafka services in Ambar
* Open putty.

1. CREATE A NEW TOPIC

Use the kafka-topics.sh script (which should be in your PATH), create a new topic named Test:

./kafka-topics.sh --create --zookeeper sandbox-hdp.hortonworks.com:2181 - -replication-factor 1 --partitions 1 --topic Test

1. If the kafka-topics.sh script is not in your PATH and you get a command not found error, then change directories to where the Kafka scripts are installed:

cd /usr/hdp/current/kafka-broker/bin

1. Also note that sometimes ZooKeeper does not listen on localhost, so you may need to use 127.0.0.1 or the Sandbox’s IP address instead.
2. Now it will create the topic called Test.
3. Check if topic Test was created successfully with the following command:

./kafka-topics.sh --list --zookeeper sandbox-hdp.hortonworks.com:2181

1. Now producer will send messages to a topic. Where everything you type is sent to the kafka topic.

./kafka-console-producer.sh --broker-list sandbox-hdp.hortonworks.com:6667 –-topic Test.

1. Now run kafka consumer console.

* cd /usr/hdp/current/kafka-broker/bin
* ./kafka-console-consumer.sh --zookeeper 192.168.127.130:2181 --topic log- Test --from-beginning
* Now messages will start coming in consumer console.

**REAL TIME EVENT STREAM:**

* Before producer writing data to the topics, need to perform some configurations.

1. Cd ..
2. There is a conf folder in the kafka folder

* cd conf #Change the directory to conf  
  ls #List the files
* Sample config properties that come with kafka
* Connect standalone - use to setup network env for standalone connector
* Connect file sync - how to store the contents of stream as comes in
* Connect file source - setup to listen to changes of a file and publish as producer on a topic on a stream
* Since these are part of kafka we don't want to modify them so let's copy them into home
* cp connect-standalone.properties ~/  
  cp connect-file-sink.properties ~/  
  cp connect-file-source.properties ~/

1. cd ~

#It will redirect to your home directory

1. Now let's edit them!

* For connect standalone properties

Open connect-standlone.properties

* vi connect-standalone.properties
* bootstrap.servers=sandbox.hortonworks.com:6667 Or

sandbox-hdp.hortonworks.com:6667

* For Connect file sink

Open connect-file-sink.properties

* change the file name
* file=/home/maria\_dev/logout.txt
* topics=log-Test
* For Connect file source

Open connect-file-source.properties

* file=/home/maria\_dev/access\_log\_small.txt
* topic=log-test

1. Let’s Download the file

wget <http://media.sundog-soft.com/hadoop/access_log_small.txt>

* less access\_log\_small.txt
* Then :q to come out

1. Before we start publishing data from that log, let's setup a consumer to listen to that data

* Besides to publish and consume that into the file, we also want a consumer to listen and dump everything.
* In our other terminal we do:

./kafka-console-consumer.sh --bootstrap-server sandbox.hortonworks.com:6667 --topic log-bharathtest --zookeeper localhost:2181

1. Now back to our previous terminal to register the producer.

cd /usr/hdp/current/kafka-broker/bin

./connect-standalone.sh ~/connect-standalone.properties ~/connect-file-source.properties ~/connect-file-sink.properties #we are setting up the consumer connector to consume the data.

1. This is a bit closer to a real life scenario.

Let's open a NEW terminal, and run the following

* cd ~ [go to home directory]
* ls
* less logout.txt #To Display the log file.

1. We setup our standalone connector and publish it to our logout.txt.

* We're also listening to it from the other consumer
* We can show that it's writing, let's add a line to access log
* echo "this is a new line" >> access\_log\_small.txt

1. At last, We can now shut down the kafka service