P1 Spark Kafka setup on VM

https://github.com/apache/kafka/releases/tag/0.8.2.2
http://kafka.apache.org/downloads.html
We are using our CentOS 6.7 VM with JDK 1.8 and Spark 1.6.2

[cloudera@localhost Marina]$ cp /mnt/hgfs/VM_shared/kafka_2.11-0.8.2.2.tgz .
[cloudera@localhost Marina]$ file kafka_2.11-0.8.2.2.tgz
kafka_2.11-0.8.2.2.tgz: POSIX tar archive
[cloudera@localhost Marina]$ tar -xvf kafka_2.11-0.8.2.2.tgz kafka_2.11-0.8.2.2/

...

```
[cloudera@localhost Marina]$ ls -l
total 25424
-rw-r-----. 1 cloudera cloudera  8491533 Feb 20 12:27 apache-maven-3.3.9-bin.tar.gz
-rw-r-----. 1 cloudera cloudera     9128 Feb  7 18:46 cloudera-cdh-5-0.x86_64.rpm
drwxr-xr-x. 5 cloudera cloudera     4096 Sep  2  2015 kafka_2.11-0.8.2.2
-rw-r--r--. 1 cloudera cloudera 17489920 Mar  6 20:12 kafka_2.11-0.8.2.2.tgz
drwxrwxr-x. 2 cloudera cloudera     4096 Mar  6 08:24 kafka-data
-rw-r--r--. 1 cloudera cloudera       61 Feb 20 13:04 local_simple_text.txt
drwxr-xr-x. 5 cloudera cloudera     4096 Feb 24 13:28 mini-examples-scala
-rw-r--r--. 1 cloudera cloudera    14576 Feb 20 14:07 spark-cscie63-jdk7.jar
drwxrwxr-x. 4 cloudera cloudera     4096 Feb 20 16:34 spark_project
[cloudera@localhost Marina]$
```

```
[cloudera@localhost Marina]$ cd kafka_2.11-0.8.2.2
[cloudera@localhost kafka_2.11-0.8.2.2]$ ls
bin  config  libs  LICENSE  NOTICE
[cloudera@localhost kafka_2.11-0.8.2.2]$ cd config/
[cloudera@localhost config]$ ls
consumer.properties  producer.properties  test-log4j.properties   zookeeper.properties
log4j.properties     server.properties    tools-log4j.properties
[cloudera@localhost config]$
```

```
[cloudera@localhost Marina]$ pwd
/home/cloudera/Marina
[cloudera@localhost Marina]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        36G  6.2G   28G  19% /
tmpfs           2.0G  164K  2.0G   1% /dev/shm
/dev/sda1       283M   41M  227M  16% /boot
.host:/         465G  303G  163G  66% /mnt/hgfs
[cloudera@localhost Marina]$
```

adjust Kafka's server.properties if needed:

```
num.io.threads=8

# The send buffer (SO_SNDBUF) used by the socket server
socket.send.buffer.bytes=102400

# The receive buffer (SO_RCVBUF) used by the socket server
socket.receive.buffer.bytes=102400

# The maximum size of a request that the socket server will accept (protection
gainst OOM)
socket.request.max.bytes=104857600


############################# Log Basics #############################

# A comma seperated list of directories under which to store log files
#log.dirs=/tmp/kafka-logs
log.dirs=/home/cloudera/Marina/kafka-data/kafka-logs

# The default number of log partitions per topic. More partitions allow greate
# parallelism for consumption, but this will also result in more files across
# the brokers.
num.partitions=1
```

A comma seperated list of directories under which to store log files
#log.dirs=/tmp/kafka-logs
log.dirs=/home/cloudera/Marina/kafka-data/kafka-logs

and zookeeper properties:

```
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
# the directory where the snapshot is stored.
#dataDir=/tmp/zookeeper
dataDir=/home/cloudera/Marina/kafka-data/zookeeper

# the port at which the clients will connect
clientPort=2181
# disable the per-ip limit on the number of connections since this is a noi
uction config
maxClientCnxns=0
                                                              18,0-1
```

Start Zookeeper first:

/home/cloudera/Marina/kafka_2.11-0.8.2.2/bin/zookeeper-server-start.sh /home/cloudera/Marina/kafka_2.11-0.8.2.2/config/zookeeper.properties

[cloudera@localhost Marina]$ /home/cloudera/Marina/kafka_2.11-0.8.2.2/bin/zookeeper-server-start.sh /home/cloudera/Marina/kafka_2.11-0.8.2.2/config/zookeeper.properties
[2016-03-06 20:25:19,560] INFO Reading configuration from: /home/cloudera/Marina/kafka_2.11-0.8.2.2/config/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2016-03-06 20:25:19,566] INFO autopurge.snapRetainCount set to 3 (org.apache.zookeeper.server.DatadirCleanupManager)
….
[2016-03-06 20:25:19,613] INFO Server environment:user.home=/home/cloudera (org.apache.zookeeper.server.ZooKeeperServer)

[2016-03-
06 20:25:19,613] INFO Server environment:user.dir=/home/cloudera/Marina (org.apache.zookeeper.server.ZooKeeperServer)
[2016-03-06 20:25:19,634] INFO tickTime set to 3000 (org.apache.zookeeper.server.ZooKeeperServer)
[2016-03-06 20:25:19,634] INFO minSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2016-03-06 20:25:19,634] INFO maxSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2016-03-06 20:25:19,651] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory)

Start Kafka server:

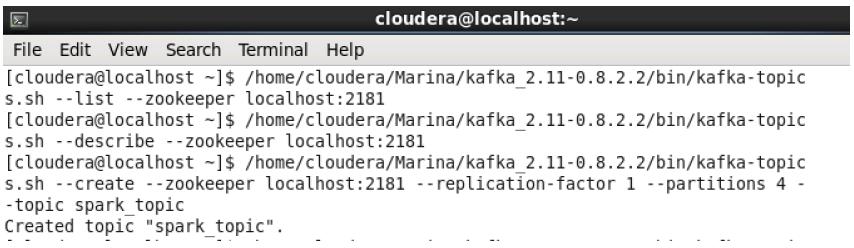
/home/cloudera/Marina/kafka_2.11-0.8.2.2/bin/kafka-server-start.sh /home/cloudera/Marina/kafka_2.11-0.8.2.2/config/server.properties

....
[2016-03-06 20:27:23,010] INFO Log directory '/home/cloudera/Marina/kafka-data/kafka-logs' not found, creating it. (kafka.log.LogManager)
[2016-03-06 20:27:23,014] INFO Loading logs. (kafka.log.LogManager)
[2016-03-06 20:27:23,025] INFO Logs loading complete. (kafka.log.LogManager)
[2016-03-06 20:27:23,026] INFO Starting log cleanup with a period of 300000 ms. (kafka.log.LogManager)
[2016-03-06 20:27:23,028] INFO Starting log flusher with a default period of 9223372036854775807 ms. (kafka.log.LogManager)
[2016-03-06 20:27:23,080] INFO Awaiting socket connections on 0.0.0.0:9092. (kafka.network.Acceptor)
[2016-03-06 20:27:23,083] INFO [Socket Server on Broker 0], Started (kafka.network.SocketServer)
[2016-03-06 20:27:23,180] INFO Will not load MX4J, mx4j-tools.jar is not in the classpath (kafka.utils.Mx4jLoader$)
[2016-03-06 20:27:23,241] INFO 0 successfully elected as leader (kafka.server.ZookeeperLeaderElector)
[2016-03-06 20:27:23,325] INFO Registered broker 0 at path /brokers/ids/0 with address localhost:9092. (kafka.utils.ZkUtils$)
[2016-03-06 20:27:23,342] **INFO [Kafka Server 0], started (kafka.server.KafkaServer)**
[2016-03-06 20:27:23,437] **INFO New leader is 0** (kafka.server.ZookeeperLeaderElector$LeaderChangeListener)

Describe Kafka cluster:
/home/cloudera/Marina/kafka_2.11-0.8.2.2/bin/kafka-topics.sh --describe --zookeeper localhost:2181

Check which topics are already created: (none so far)
/home/cloudera/Marina/kafka_2.11-0.8.2.2/**bin/kafka-topics.sh --list --zookeeper localhost:2181**
<< nothing so far - since we did not create any topics yet>>

Create new topics:
/home/cloudera/Marina/kafka_2.11-0.8.2.2/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 4 --topic spark_topic



Describe the cluster again:
/home/cloudera/Marina/kafka_2.11-0.8.2.2/bin/kafka-topics.sh --describe --zookeeper localhost:2181

```
[cloudera@localhost ~]$ /home/cloudera/Marina/kafka_2.11-0.8.2.2/bin/kafka-topics.sh --describe -
zookeeper localhost:2181
Topic:spark_topic       PartitionCount:4        ReplicationFactor:1     Configs:
        Topic: spark_topic      Partition: 0    Leader: 0       Replicas: 0     Isr: 0
        Topic: spark_topic      Partition: 1    Leader: 0       Replicas: 0     Isr: 0
        Topic: spark_topic      Partition: 2    Leader: 0       Replicas: 0     Isr: 0
        Topic: spark_topic      Partition: 3    Leader: 0       Replicas: 0     Isr: 0
[cloudera@localhost ~]$
```

Test new topic:
/home/cloudera/Marina/kafka_2.11-0.8.2.2/**bin/kafka-console-producer.sh --broker-list localhost:9092 --topic** spark_topic

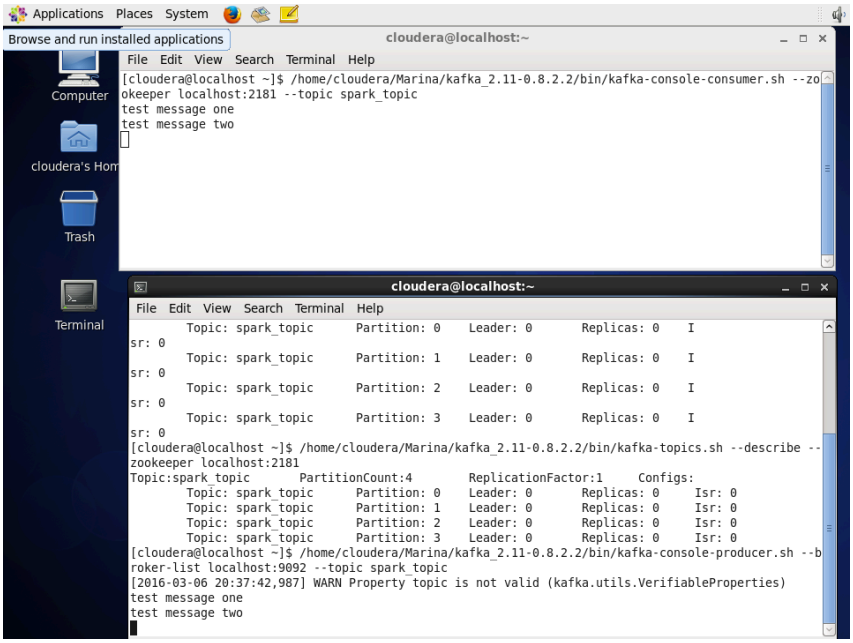
```
[cloudera@localhost ~]$ /home/cloudera/Marina/kafka_2.11-0.8.2.2/bin/kafka-console-producer.sh --broker-
list localhost:9092 --topic spark_topic
[2016-03-06 20:37:42,987] WARN Property topic is not valid (kafka.utils.VerifiableProperties)
```

NOTE: this warning is Ok - Kafka community fixed it in Kafka 0.9

Use simple consumer to verify messages are being sent/consumed:

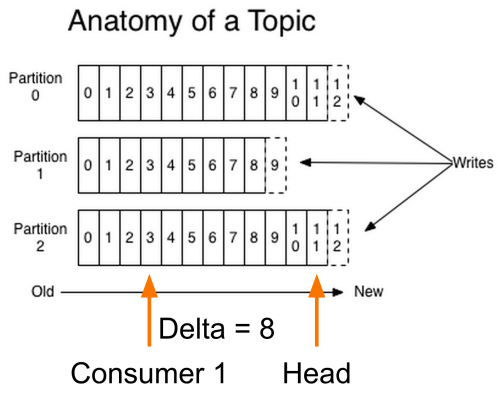
/home/cloudera/Marina/kafka_2.11-0.8.2.2/**bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic** spark_topic **--from-beginning**

type a message in the producer (one message = one line)
observe it received in the consumer :



To stop:
Very important - always do it in this order!
First, stop the Kafka server - Cntr+C in the terminal
Second, stop Zookeeper server - Cntrl+C in the terminal

Exploring Kafka logs



Look into your specified kafka log directory (they one you set in kafka's server.properties):

```
cloudera@localhost:~/Marina
File  Edit  View  Search  Terminal  Help
kafka_2.11-0.8.2.2.tgz          spark-cscie63-jdk7.jar
[cloudera@localhost Marina]$ ls kafka-data/
kafka-logs  zookeeper
[cloudera@localhost Marina]$ ls kafka-data/kafka-logs/
__consumer_offsets-0    __consumer_offsets-26  __consumer_offsets-43
__consumer_offsets-1    __consumer_offsets-27  __consumer_offsets-44
__consumer_offsets-10   __consumer_offsets-28  __consumer_offsets-45
__consumer_offsets-11   __consumer_offsets-29  __consumer_offsets-46
__consumer_offsets-12   __consumer_offsets-3   __consumer_offsets-47
__consumer_offsets-13   __consumer_offsets-30  __consumer_offsets-48
__consumer_offsets-14   __consumer_offsets-31  __consumer_offsets-49
__consumer_offsets-15   __consumer_offsets-32  __consumer_offsets-5
__consumer_offsets-16   __consumer_offsets-33  __consumer_offsets-6
__consumer_offsets-17   __consumer_offsets-34  __consumer_offsets-7
__consumer_offsets-18   __consumer_offsets-35  __consumer_offsets-8
__consumer_offsets-19   __consumer_offsets-36  __consumer_offsets-9
__consumer_offsets-2    __consumer_offsets-37  recovery-point-offset-checkpoint
__consumer_offsets-20   __consumer_offsets-38  replication-offset-checkpoint
__consumer_offsets-21   __consumer_offsets-39  spark_topic-0
__consumer_offsets-22   __consumer_offsets-4   spark_topic-1
__consumer_offsets-23   __consumer_offsets-40  spark_topic-2
__consumer_offsets-24   __consumer_offsets-41  spark_topic-3
__consumer_offsets-25   __consumer_offsets-42
[cloudera@localhost Marina]$
```

Inspect content:

/home/cloudera/Marina/kafka_2.11-0.8.2.2/bin/kafka-run-class.sh kafka.tools.DumpLogSegments --files /home/cloudera/Marina/kafka-data/kafka-logs/spark_topic-0/00000000000000000000.log --print-data-log

[cloudera@localhost kafka-logs]$ ls -l spark_topic-1
total 4
-rw-rw-r--. 1 cloudera cloudera 0 Mar 10 20:26 00000000000000000000.index
-rw-rw-r--. 1 cloudera cloudera 84 Mar 6 20:39 00000000000000000000.log
[cloudera@localhost kafka-logs]$ ls -l spark_topic-2
total 0
-rw-rw-r--. 1 cloudera cloudera 0 Mar 10 20:26 00000000000000000000.index
-rw-rw-r--. 1 cloudera cloudera 0 Mar 6 20:33 00000000000000000000.log

This partition (3) seems to have more data: (green ones are the messages)
[cloudera@localhost kafka-logs]$ ls -l spark_topic-3
total 4
-rw-rw-r--. 1 cloudera cloudera 0 Mar 10 20:26 00000000000000000000.index
-rw-rw-r--. 1 cloudera cloudera 210 Mar 10 19:59 00000000000000000000.log

[cloudera@localhost kafka-logs]$ **/home/cloudera/Marina/kafka_2.11-0.8.2.2/bin/kafka-run-class.sh kafka.tools.DumpLogSegments --files /home/cloudera/Marina/kafka-data/kafka-logs/spark_topic-3/00000000000000000000.log --print-data-log**
Dumping /home/cloudera/Marina/kafka-data/kafka-logs/spark_topic-3/00000000000000000000.log
Starting offset: 0
offset: 0 position: 0 isvalid: true payloadsize: 8 magic: 0 compresscodec: NoCompressionCodec crc: 1953334871 payload: one line
offset: 1 position: 34 isvalid: true payloadsize: 9 magic: 0 compresscodec: NoCompressionCodec crc: 3903235078 payload: two lines
offset: 2 position: 69 isvalid: true payloadsize: 11 magic: 0 compresscodec: NoCompressionCodec crc: 2297784979 payload: three lines
offset: 3 position: 106 isvalid: true payloadsize: 8 magic: 0 compresscodec: NoCompressionCodec crc: 282985463 payload: line one
offset: 4 position: 140 isvalid: true payloadsize: 9 magic: 0 compresscodec: NoCompressionCodec crc: 4260965638 payload: message 1
offset: 5 position: 175 isvalid: true payloadsize: 9 magic: 0 compresscodec: NoCompressionCodec crc: 1693482172 payload: message 2
[cloudera@localhost kafka-logs]$