Single Node Apache Kafka 0.10 Setup and Configuration on Ubuntu 14.04

Data Science Lab, The Department of Computer Science,
KSKV Kachchh University.

Web: http://cs.kutchuni.edu.in
Github: https://github.com/dskskv

1 Introduction to Kafka

Kafka is a distributed, partitioned, replicated commit log service. It provides the functionality of a messaging system, but with a unique design.
What does all that mean?

First let's review some basic messaging terminology:

Kafka maintains feeds of messages in categories called topics.
We'll call processes that publish messages to a Kafka topic producers.
We'll call processes that subscribe to topics and process the feed of published messages consumers.
Kafka is run as a cluster comprised of one or more servers each of which is called a broker.

Messaging

In comparison to most messaging systems Kafka has better throughput, built-in partitioning, replication, and fault-tolerance which makes it a good solution for large scale message processing applications.

Kafka-Storm Pipeline

Kafka can be used with Apache Storm to handle data pipeline for high speed filtering and pattern matching on the fly.

Metrics

Kafka is often used for operation monitoring data pipelines. This involves aggregating statistics from distributed applications to produce centralized feeds of operational data.

Website Activity Tracking

Website activity (page views, searches, or other actions users may take) is published to central topics with one topic per activity type. These feeds are available for subscription for a range of use cases including real-time processing, real-time monitoring, and loading into Hadoop or offline data warehousing systems for offline processing and reporting.

Kafka is it's main use - low overhead short term persistence stream processing

Apache Kafka is the pub-sub, distributed, partitioned, replicated messaging system, Kafka publisher subscriber model would be used for low latency, nearly real time events/log aggregations, monitoring via distributed systems etc. With combination with Apache Storm or any distributed processing system, Kafka would be used as full end to end bigdata log/events aggregations and processing framework, or speed layer in Lambda architecture.

2. Kafka Setup & Configuration steps.

1. Download Apache Kafka version 0.10 with build using scala 2.11
   wget http://mirror.fibergrid.in/apache/kafka/0.10.0.0/kafka_2.11-0.10.0.0.tgz

2. Uncompress it using command:
   tar xvzf kafka_2.11-0.10.0.0.tgz

3. Move Uncompressed Kafka to user local directory where usually all big data services are there.
   sudo mv kafka_2.11-0.10.0.0 /usr/local/kafka

4. create logs directory at kafka root directory
   mkdir logs
   [Note: your current pwd(current directory) should be at root directory of kafka]

5. Edit server properties file
   sudo gedit server.properties

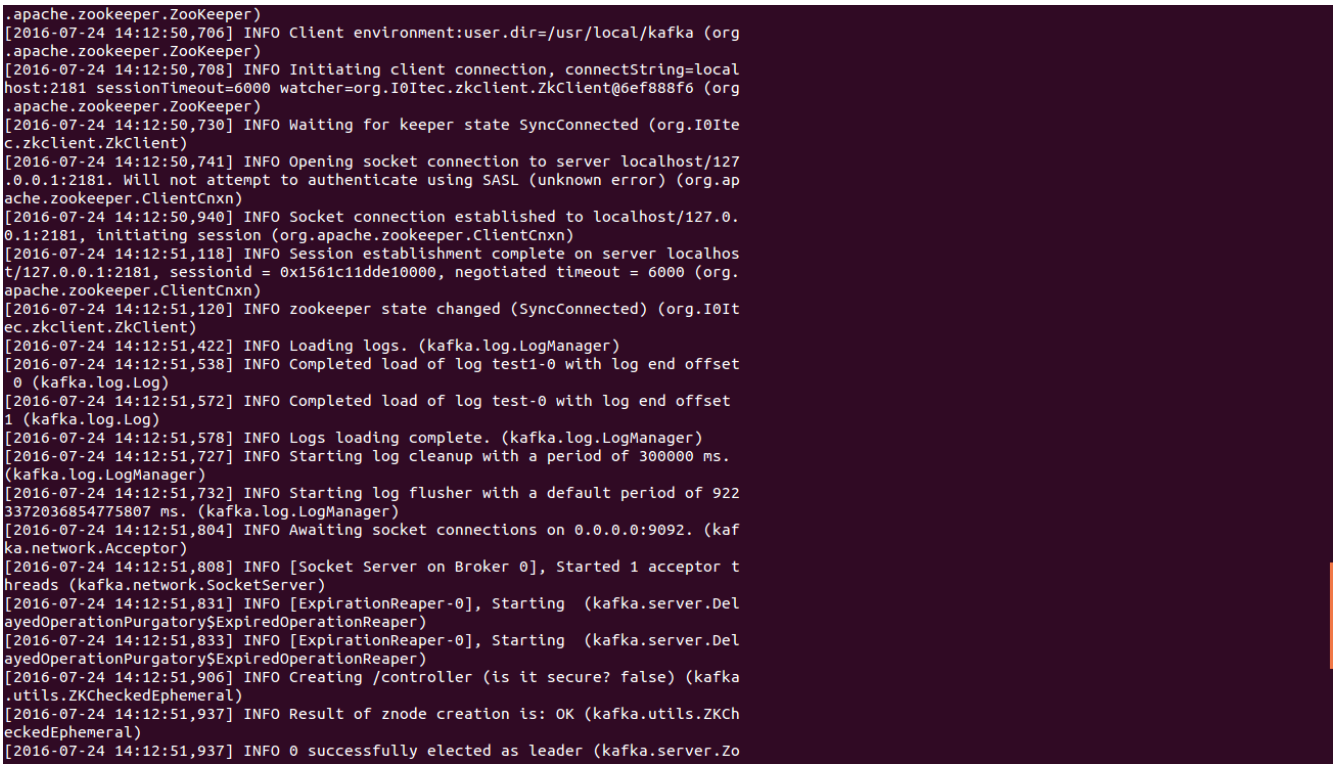
6. Change path : log.dirs=/usr/local/kafka/logs

7. go to Kafka root directory in one Terminal:
chetan306@chetan306:/usr/local/kafka$
bin/zookeeper-server-start.sh config/zookeeper.properties

```
[2016-07-24 14:12:28,017] INFO Server environment:java.home=/usr/lib/jvm/java-8-oracle/jre (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 14:12:28,017] INFO Server environment:java.class.path=.:/usr/lib/jvm/java-8-oracle/lib:/home:/usr/local/kafka/bin/../libs/aopalliance-repackaged-2.4.0-b34.jar:/usr/local/kafka/bin/../libs/argparse4j-0.5.0.jar:/usr/local/kafka/bin/../libs/connect-api-0.10.0.0.jar:/usr/local/kafka/bin/../libs/connect-file-0.10.0.0.jar:/usr/local/kafka/bin/../libs/connect-json-0.10.0.0.jar:/usr/local/kafka/bin/../libs/connect-runtime-0.10.0.0.jar:/usr/local/kafka/bin/../libs/guava-18.0.jar:/usr/local/kafka/bin/../libs/hk2-api-2.4.0-b34.jar:/usr/local/kafka/bin/../libs/hk2-locator-2.4.0-b34.jar:/usr/local/kafka/bin/../libs/hk2-utils-2.4.0-b34.jar:/usr/local/kafka/bin/../libs/jackson-annotations-2.6.0.jar:/usr/local/kafka/bin/../libs/jackson-core-2.6.3.jar:/usr/local/kafka/bin/../libs/jackson-databind-2.6.3.jar:/usr/local/kafka/bin/../libs/jackson-jaxrs-base-2.6.3.jar:/usr/local/kafka/bin/../libs/jackson-jaxrs-json-provider-2.6.3.jar:/usr/local/kafka/bin/../libs/jackson-module-jaxb-annotations-2.6.3.jar:/usr/local/kafka/bin/../libs/javassist-3.18.2-GA.jar:/usr/local/kafka/bin/../libs/javax.annotation-api-1.2.jar:/usr/local/kafka/bin/../libs/javax.inject-1.jar:/usr/local/kafka/bin/../libs/javax.inject-2.4.0-b34.jar:/usr/local/kafka/bin/../libs/javax.servlet-api-3.1.0.jar:/usr/local/kafka/bin/../libs/javax.ws.rs-api-2.0.1.jar:/usr/local/kafka/bin/../libs/jersey-client-2.22.2.jar:/usr/local/kafka/bin/../libs/jersey-common-2.22.2.jar:/usr/local/kafka/bin/../libs/jersey-container-servlet-2.22.2.jar:/usr/local/kafka/bin/../libs/jersey-container-servlet-core-2.22.2.jar:/usr/local/kafka/bin/../libs/jersey-guava-2.22.2.jar:/usr/local/kafka/bin/../libs/jersey-media-jaxb-2.22.2.jar:/usr/local/kafka/bin/../libs/jersey-server-2.22.2.jar:/usr/local/kafka/bin/../libs/jetty-continuation-9.2.15.v20160210.jar:/usr/local/kafka/bin/../libs/jetty-http-9.2.15.v20160210.jar:/usr/local/kafka/bin/../libs/jetty-io-9.2.15.v20160210.jar:/usr/local/kafka/bin/../libs/jetty-security-9.2.15.v20160210.jar:/usr/local/kafka/bin/../libs/jetty-server-9.2.15.v20160210.jar:/usr/local/kafka/bin/../libs/jetty-servlet-9.2.15.v20160210.jar:/usr/local/kafka/bin/../libs/jetty-servlets-9.2.15.v20160210.jar:/usr/local/kafka/bin/../libs/jetty-util-9.2.15.v20160210.jar:/usr/local/kafka/bin/../libs/jopt-simple-4.9.jar:/usr/local/kafka/bin/../libs/kafka_2.11-0.10.0.0.jar:/usr/local/kafka/bin/../libs/kafka_2.11-0.10.0.0-sources.jar:/usr/local/kafka/bin/../libs/kafka_2.11-0.10.0.0-test-sources.jar:/usr/local/kafka/bin/../libs/kafka-clients-0.10.0.0.jar:/usr/local/kafka/bin/../libs/kafka-log4j-appender-0.10.0.0.jar:/usr/local/kafka/bin/../libs/kafka-streams-0.10.0.0.jar:/usr/local/kafka/bin/../libs/kafka-streams-examples-0.10.0.0.jar:/usr/local/kafka/bin/../libs/kafka-tools-0.10.0.0.jar:/usr/local/kafka/bin/../libs/log4j-1.2.17.jar:/usr/local/kafka/bin/../libs/lz4-1.3.0.jar:/usr/local/kafka/bin/../libs/metrics-core-2.2.0.jar:/usr/local/kafka/bin/../libs/osgi-resource-locator-1.0.1.jar:/usr/local/kafka/bin/../libs/reflections-0.9.10.jar:/usr/local/kafka/bin/../libs/rocksdbjni-4.4.1.jar:/usr/local/kafka/bin/../libs/scala-library-2.11.8.jar:/usr/local/kafka/bin/../libs/scala-parser-combinators_2.11-1.0.4.jar:/usr/local/kafka/bin/../libs/slf4j-api-1.7.21.jar:/usr/local/kafka/bin/../libs/slf4j-log4j12-1.7.21.jar:/usr/local/kafka/bin/../libs/snappy-java-1.1.2.4.jar:/usr/local/kafka/bin/../libs/validation-api-1.1.0.Final.jar:/usr/local/kafka/bin/../libs/zkclient-0.8.jar:/usr/local/kafka/bin/../libs/zookeeper-3.4.6.jar (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 14:12:28,017] INFO Server environment:java.library.path=/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 14:12:28,017] INFO Server environment:java.io.tmpdir=/tmp (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 14:12:28,017] INFO Server environment:java.compiler=<NA> (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 14:12:28,017] INFO Server environment:os.name=Linux (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 14:12:28,017] INFO Server environment:os.arch=amd64 (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 14:12:28,017] INFO Server environment:os.version=3.19.0-25-generic (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 14:12:28,017] INFO Server environment:user.name=chetan306 (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 14:12:28,017] INFO Server environment:user.home=/home/chetan306 (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 14:12:28,017] INFO Server environment:user.dir=/usr/local/kafka (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 14:12:28,028] INFO tickTime set to 3000 (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 14:12:28,028] INFO minSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 14:12:28,029] INFO maxSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 14:12:28,054] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory)
```

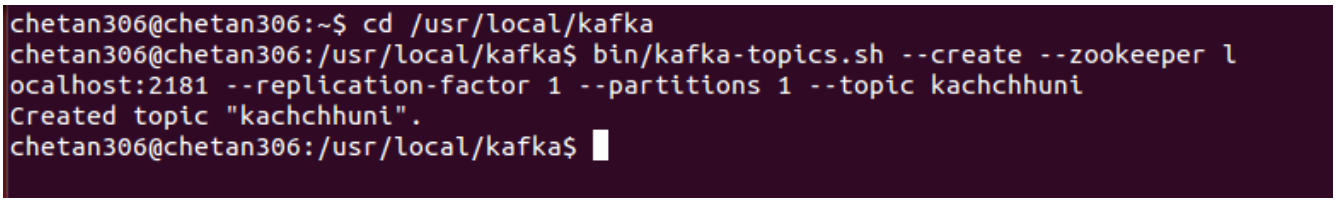
[Figure 1]: Zookeeper service has started

8. open another Terminal and hit below command:
chetan306@chetan306:/usr/local/kafka$
bin/kafka-server-start.sh config/server.properties



[Figure 2]: Kafka Server service has started

9. Open another Terminal and hit below command for creating Topic from Kafka Root directory:
bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic kachchuni

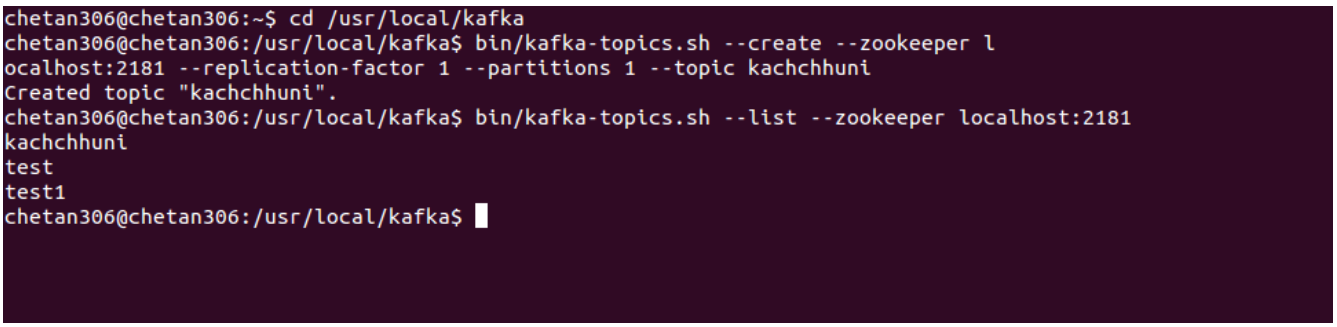


[Figure 3]: Kafka Topic creation

Note: --replication-factor parameter used to provide number of broker(Kafka server's, cluster is combination of servers and in kafka individual server is known as broker, kafka client has port number 2181 (Kakfa Producer) and kafka server has port number 9092 (Kafka Consumer)).

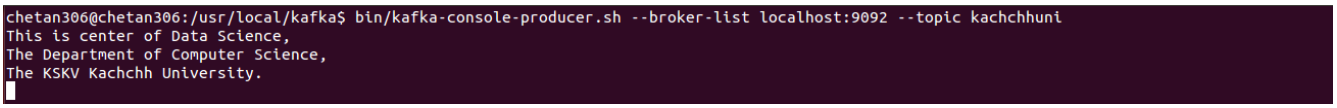
10. To check all the created topics:

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



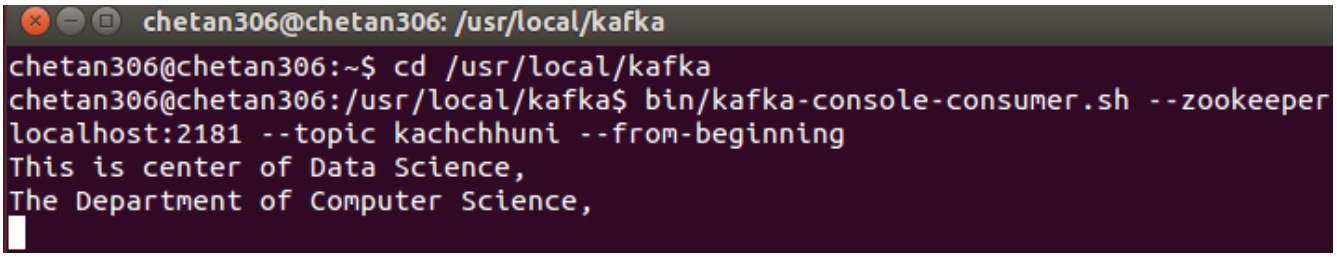
[Figure 4]: Listing out Kafka Topics

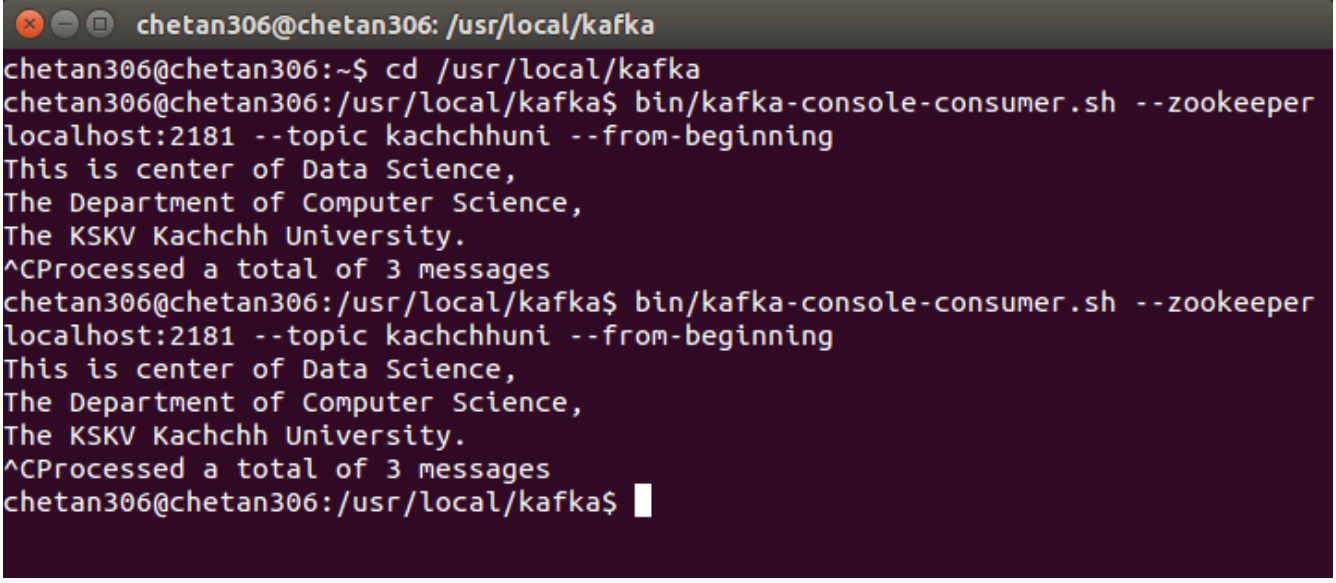
11. Insert Message to Topic, by using console based kafka producer
    bin/kafka-console-producer.sh --broker-list localhost:9092 --topic kachchhuni



[Figure 4]: Producing Message for Kafka Topic

12. Read Message from Topic, by using console based kafka consumer
    bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic kachchhuni –from-beginning

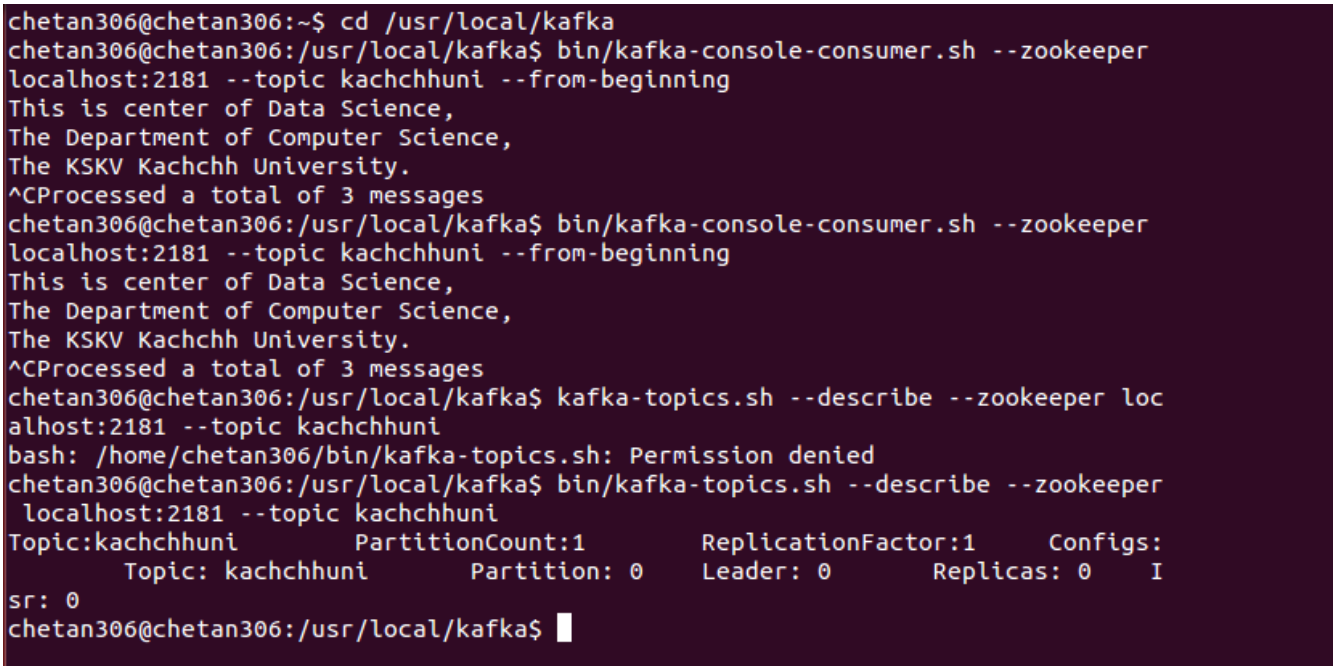




[Figure 5]: Consuming message from Kafka topic produced by Kafka Producer.

13. Describe Kafka Topic

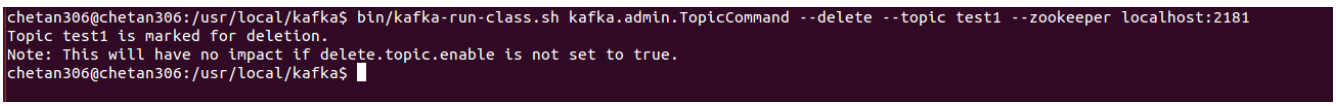
bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic kachchhuni



[Figure 6]: Describe Kafka Topic

14. Delete Kafka Topic

bin/kafka-run-class.sh kafka.admin.TopicCommand --delete --topic test1 --zookeeper localhost:2181



[Figure 7]: Describe Kafka Topic