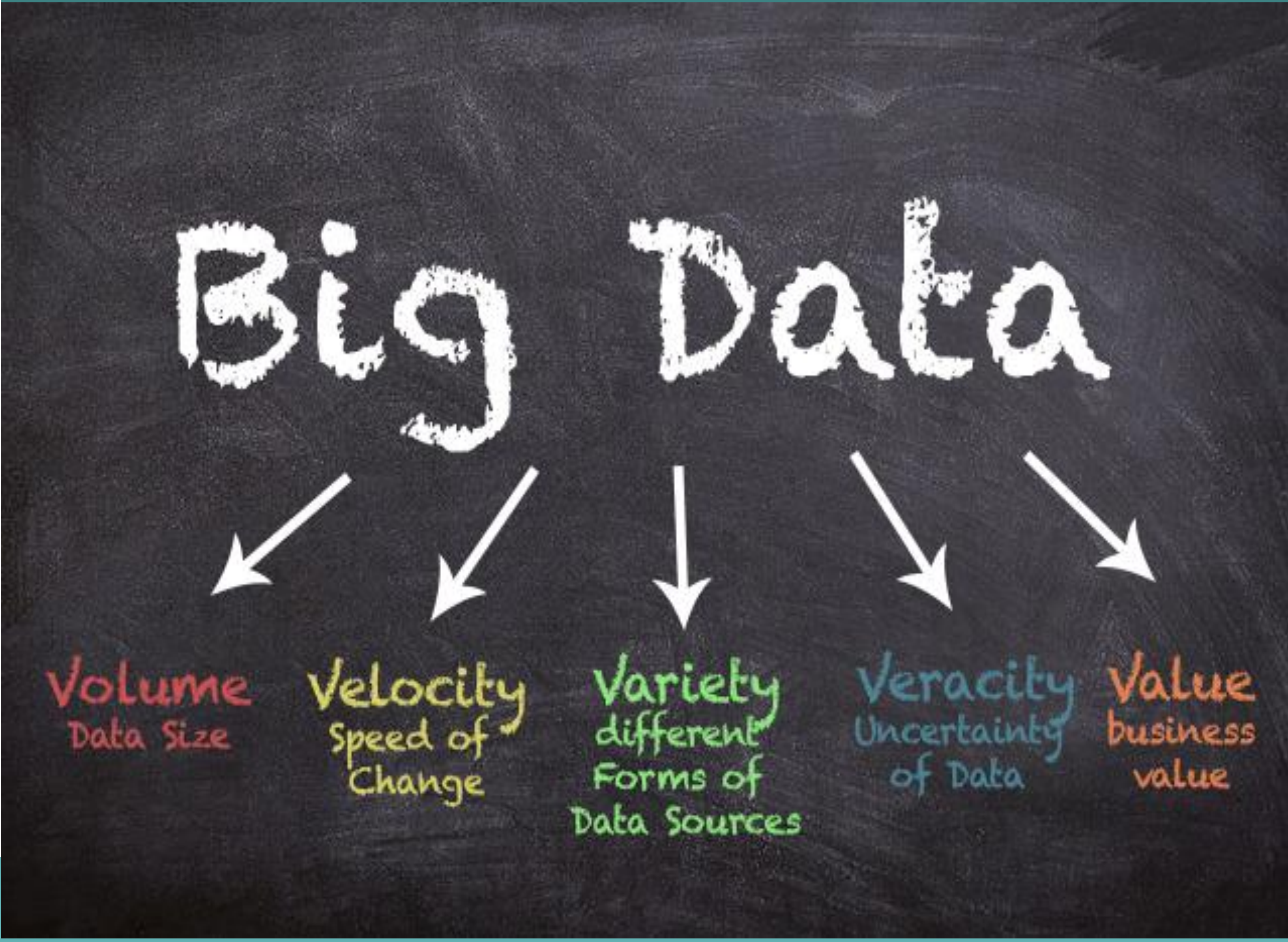
09/25/2018 | By: Ajit K Prasad



Big Data Engineering with Hadoop & Spark

Kafka Introduction

APACHE kafka®
A distributed streaming platform

APACHE Spark™

Acadgild

Session 22: Assignment 22.1

This assignment is aimed at consolidating the concepts that was learnt during the Apache Kafka session of the course.

Initial Definitions:

Topic: A topic is a category or feed name to which records are published. Topics in Kafka are always multi-subscriber; that is, a topic can have zero, one, or many consumers that subscribe to the data written to it.

Producers: Producers publish data to the topics of their choice. The producer is responsible for choosing which record to assign to which partition within the topic. This can be done in a round-robin fashion simply to balance load or it can be done according to some semantic partition function.

Consumers: Consumers label themselves with a consumer group name, and each record published to a topic is delivered to one consumer instance within each subscribing consumer group.

- If all the consumer instances have the same consumer group, then the records will effectively be load balanced over the consumer instances.
- If all the consumer instances have different consumer groups, then each record will be broadcast to all the consumer processes.

Zoopkeeper: ZooKeeper is a service to which clients a can connect to. It provides access to clients to a tree like structure or a hierarchical namespace.

Task 1:

Create a Kafka topic named KeyLessTopic.

- Start Zookeeper

```
$   $KAFKA_HOME/bin/zookeeper-server-start.sh
    $KAFKA_HOME/config/zookeeper.properties
```

- Start Kafka Broker

```
$   $KAFKA_HOME/bin/kafka-server-start.sh
    $KAFKA_HOME/config/server.properties
```

- Create a topic named KeyLessTopic

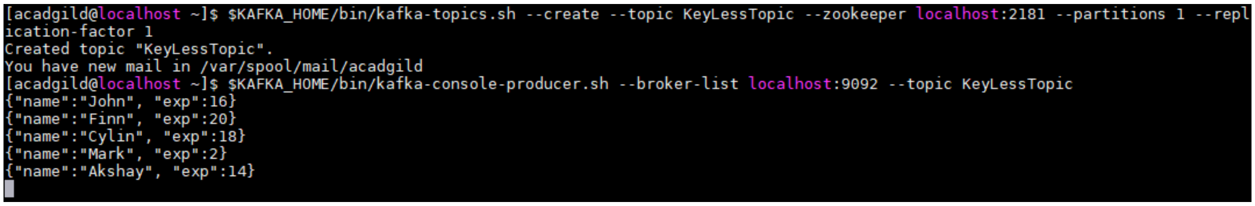
```
$   $KAFKA_HOME/bin/kafka-topics.sh --create --topic KeyLessTopic --
    zookeeper localhost:2181 --partitions 1 --replication-factor 1
```

- Inside KeyLessTopic insert following data

```
$   $KAFKA_HOME/bin/kafka-console-producer.sh            --broker-list
    localhost:9092 --topic KeyLessTopic
```

- Insert the Data below

```
{"name":"John", "exp":16}
{"name":"Finn", "exp":20}
{"name":"Cylin", "exp":18}
{"name":"Mark", "exp":2}
{"name":"Akshay", "exp":14}
```



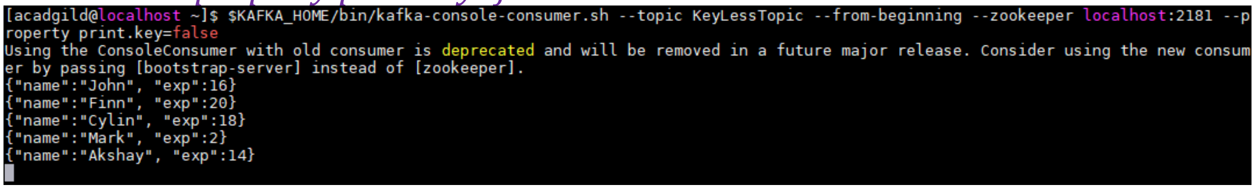
```
[acadgild@localhost ~]$ $KAFKA_HOME/bin/kafka-topics.sh --create --topic KeyLessTopic --zookeeper localhost:2181 --partitions 1 --repl
ication-factor 1
Created topic "KeyLessTopic".
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ $KAFKA_HOME/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic KeyLessTopic
{"name":"John", "exp":16}
{"name":"Finn", "exp":20}
{"name":"Cylin", "exp":18}
{"name":"Mark", "exp":2}
{"name":"Akshay", "exp":14}
```

Task 2:

Create a console consumer that reads KeyLessTopic from beginning

- Running a console consumer to read the type KeyLessTopic sent by the producer.

```
$   $KAFKA_HOME/bin/kafka-console-consumer.sh                --topic
    KeyLessTopic  --from-beginning  --zookeeper  localhost:2181  --
    property print.key=false
```



```
[acadgild@localhost ~]$ $KAFKA_HOME/bin/kafka-console-consumer.sh --topic KeyLessTopic --from-beginning --zookeeper localhost:2181 --p
roperty print.key=false
Using the ConsoleConsumer with old consumer is deprecated and will be removed in a future major release. Consider using the new consum
er by passing [bootstrap-server] instead of [zookeeper].
{"name":"John", "exp":16}
{"name":"Finn", "exp":20}
{"name":"Cylin", "exp":18}
{"name":"Mark", "exp":2}
{"name":"Akshay", "exp":14}
```

Task 3:

Create a Kafka topic named KeyedTopic. Inside KeyedTopic insert following data:

- The part before comma(,) should be treated as key and after comma(,) should be treated as value

  {"name":"John"},{"exp":16}
  {"name":"Finn"},{"exp":20}
  {"name":"Cylin"},{"exp":18}
  {"name":"Mark"},{"exp":2}
  {"name":"Akshay"},{"exp":14}

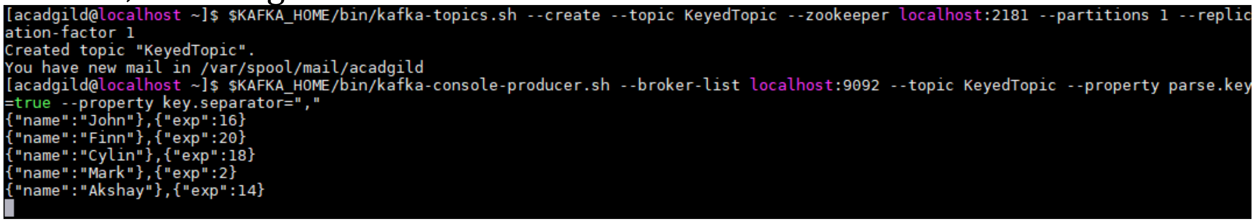
- Creating Topic

  $ $KAFKA_HOME/bin/kafka-topics.sh --create --topic KeyedTopic --zookeeper localhost:2181 --partitions 1 --replication-factor 1

- Creating Producer

  $ $KAFKA_HOME/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic KeyedTopic --property parse.key=true --property key.separator=","

- Now, inserting the above-mentioned data



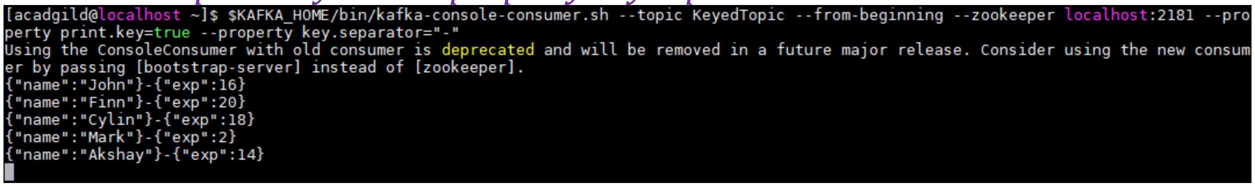
```
[acadgild@localhost ~]$ $KAFKA_HOME/bin/kafka-topics.sh --create --topic KeyedTopic --zookeeper localhost:2181 --partitions 1 --replic
ation-factor 1
Created topic "KeyedTopic".
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ $KAFKA_HOME/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic KeyedTopic --property parse.key
=true --property key.separator=","
{"name":"John"},{"exp":16}
{"name":"Finn"},{"exp":20}
{"name":"Cylin"},{"exp":18}
{"name":"Mark"},{"exp":2}
{"name":"Akshay"},{"exp":14}
```

Task 4:

Create a console consumer that reads KeyedTopic from beginning. The key and value should be separated by '-'

- Creating Comsumer

  $ $KAFKA_HOME/bin/kafka-console-consumer.sh --topic KeyedTopic --from-beginning --zookeeper localhost:2181 --property print.key=true --property key.separator="-"



```
[acadgild@localhost ~]$ $KAFKA_HOME/bin/kafka-console-consumer.sh --topic KeyedTopic --from-beginning --zookeeper localhost:2181 --pro
perty print.key=true --property key.separator="-"
Using the ConsoleConsumer with old consumer is deprecated and will be removed in a future major release. Consider using the new consum
er by passing [bootstrap-server] instead of [zookeeper].
{"name":"John"}-{"exp":16}
{"name":"Finn"}-{"exp":20}
{"name":"Cylin"}-{"exp":18}
{"name":"Mark"}-{"exp":2}
{"name":"Akshay"}-{"exp":14}
```