Kafka

Kafka – a distributed streaming platform

This technology is best suited for a scenario where data has to be published or subscribed to. It is more of a similar to message queue or enterprise messaging system. Can be consumed appropriately into two broad type of applications:

1. Building real time streaming data pipelines between several interfaces and or applications
2. Building real time services to perform analysis or data transformation before exchanging between consumers.

Three major aspects of kafka are: Producer, Consumer and the cluster itself.

Kafka Producer:

The entity under kafka world which is essential to generate or source all the data or messages to be floated over.

Kafka Consumer:

The individual entity which is responsible to ping kafka cluster on time interval basis to look for any new messages or data. The segregation happens on the basis of one concept called ‘Topic’ used extensively by producer and consumers.

At the core of apache kafka is the cluster. This cluster can run in two different modes namely: standalone, distributed.

Producer Producer Producer

Kafka Cluster

Consumer Consumer Consumer

Idea:

Considering the image of expedia among the masses. There is a huge possibility that users will be submitting the reviews or user ratings of their experience with one particular hotel or their stay. At the same time if there is a user who has been booking the same hotel or place, then it is necessary that the visiting user should know about people’s experience with the hotel they are booking.

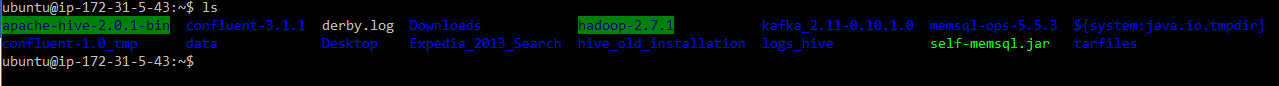
For this very reason, we have implemented this idea of uploading or producing the reviews on real-time basis which gets uploaded onto kafka producer with its timestamp, hotel id and user review. Now, we need to consume this review, apply sentiment analysis on this review and get the score of this review to better understand whether the experience was positive or negative. We use Stanford NLP to calculate this score on real-time basis and update the already available score of this hotel with necessary aggregation and normalization.

We have utilized kafka’s java api and used the documentation extensively to write the code for designing kafka producer and kafka consumer utility. We have hosted our kafka cluster which comprises of zookeeper server and kafka server on Amazon EC2’s powerful instance.

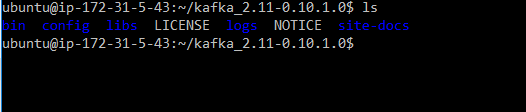
The concept of topic applies to producer and consumer. We try to segregate the messages or identify them rather based on their origination property. While generating or producing a message, producer assigns a property to that message or data, that property is called topic. We can think of this topic as a #tag concept with twitter.

Process of initiating kafka service:

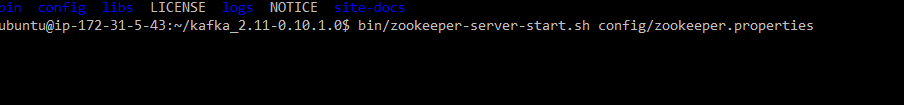
1. Inititate Zookeepers’ server with specific properties where in we mention the server has to be up on one specific ip address and port. We chose this properties as localhost:2181



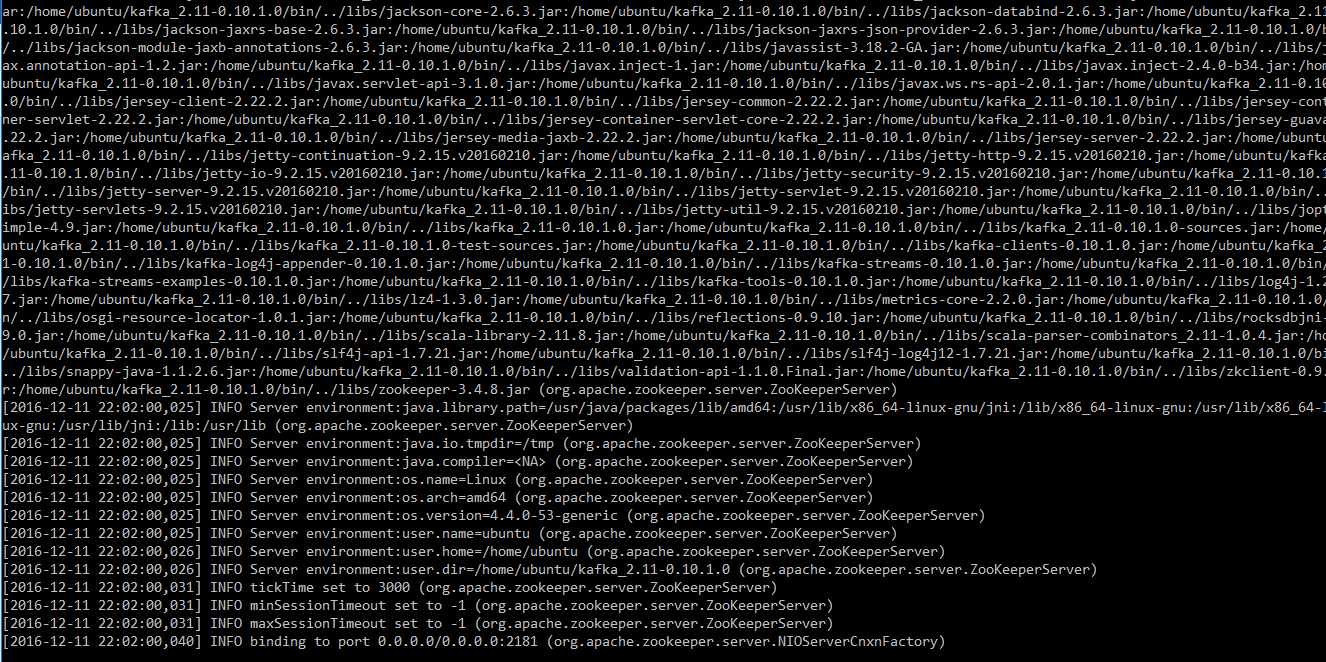
Cd kafka\_2.11-0.10.1.0



Command to start zookeeper service(keep in mind this has to be in its own instance):

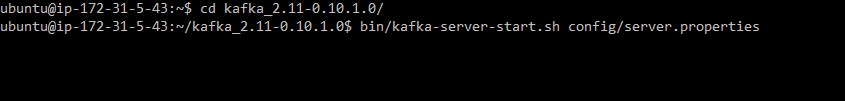


Service started:

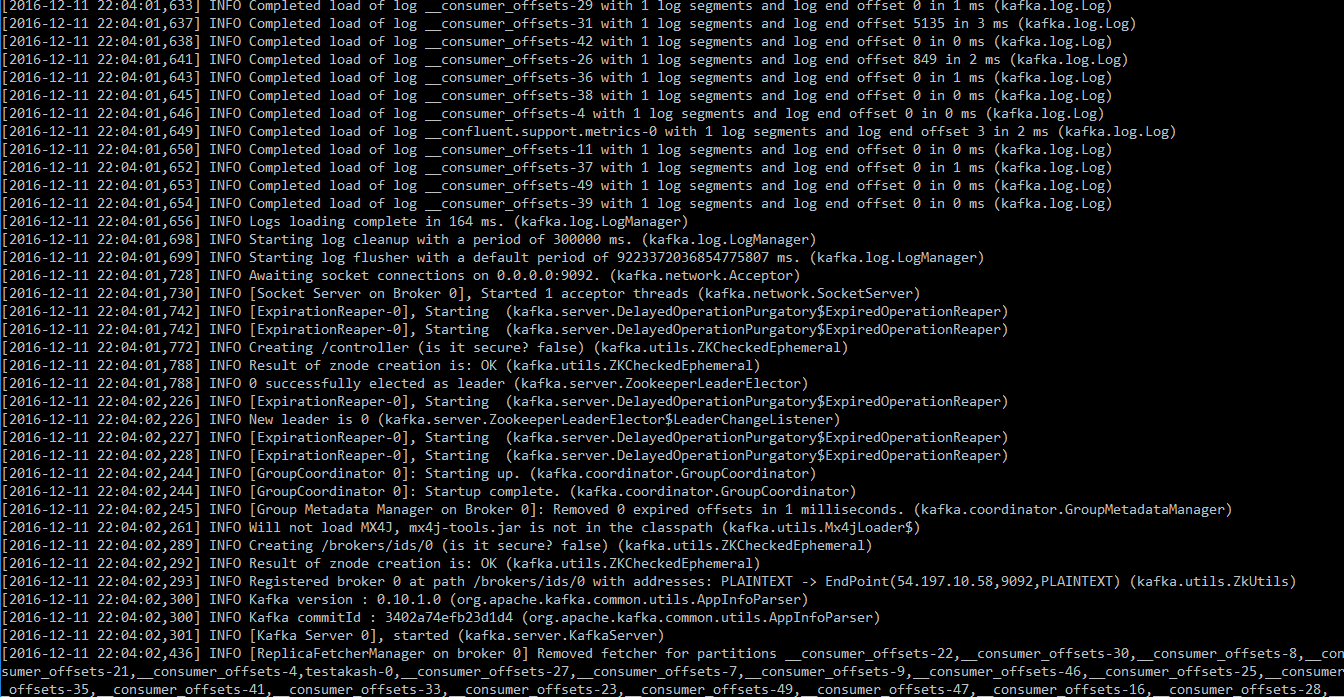


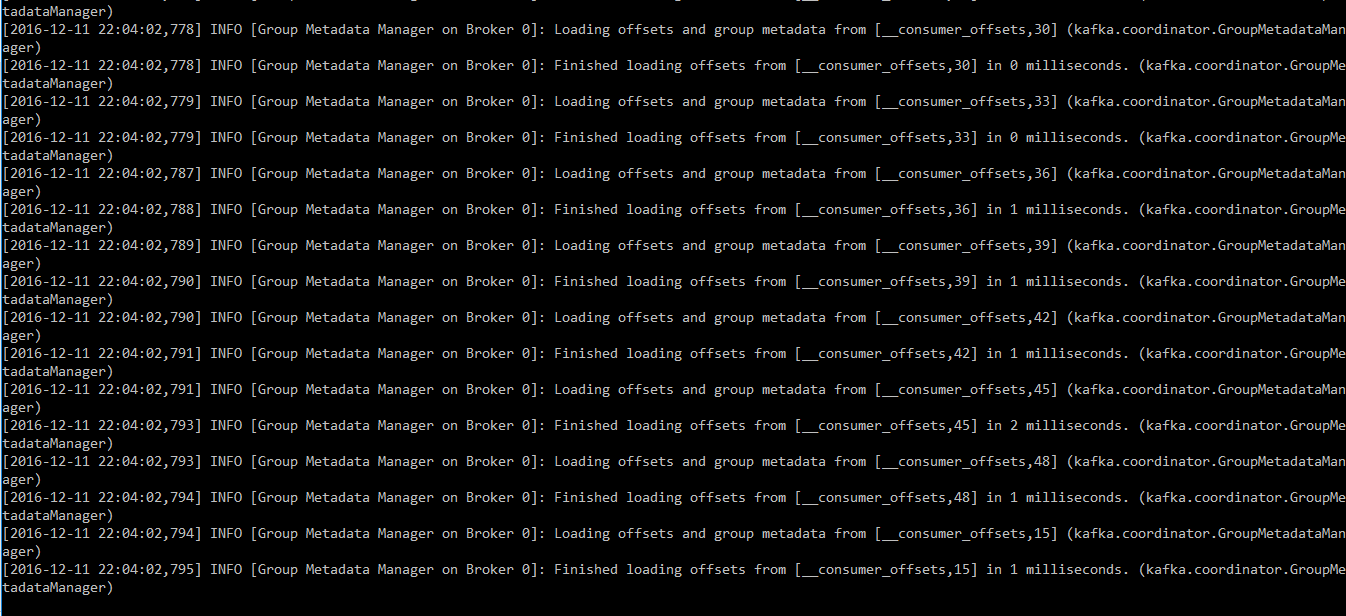
1. Initiate kafka server service:

Command to start kafka service(keep in mind this has to be in its own instance):

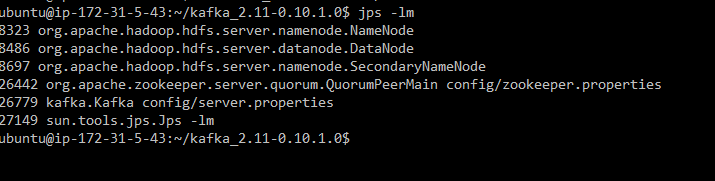


Service started:

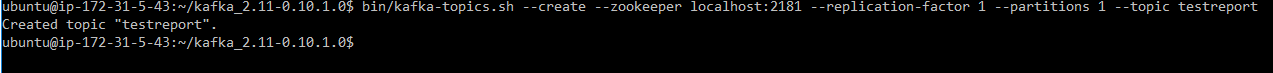




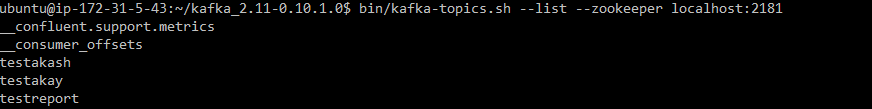
Make sure zookeeper and kafka service is working:



1. Create topic to be used as a common attribute between producer and consumer

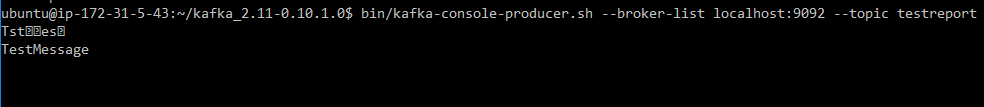


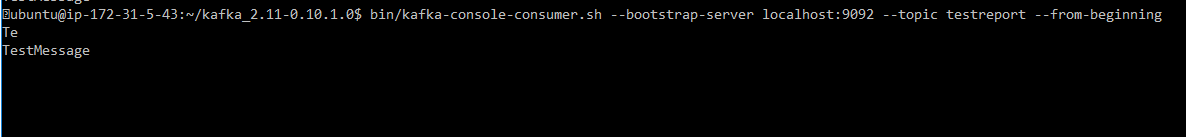
1. Check if your topic has been created by listing all the topics available with kafka right now



1. We can utilize producer and consumer to exchange messages on real-time basis.

Producer – to test whether producer and consumer are in sync and send over some messages:

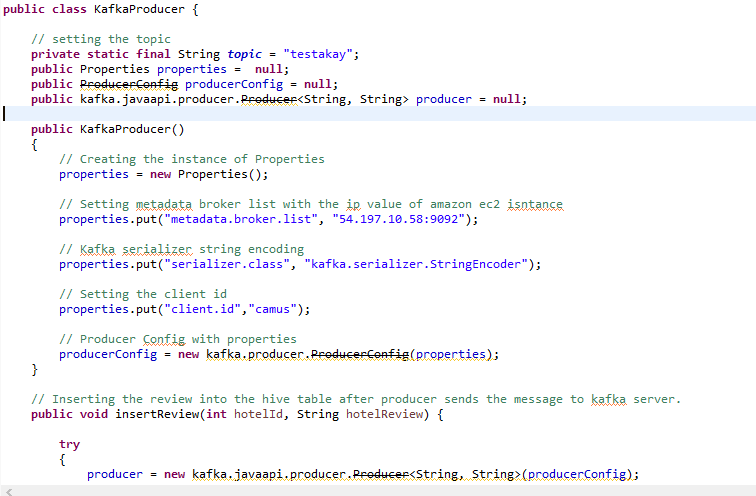


1. Consumer – to check if messages sent by producer are same received here: 

After completing these 6 steps, we are assured that kafka and zookeeper service are running properly.

Project Implementation:

KafkaProducer-



Properties a utility function provided by java. Used to configure to point to specific ip and port to push messages to.

ProducerConfig a class defined under kafka-core java library which enables kafka to consume the properties java’s utility function and evaluate configuration parameters.

Creating instance of kafka producer utilizing producer configuration instance which has all the configuration parameters loaded.

Functions:

1. InsertReview

This method is to get reviews from user interface and publish the reviews on kafka producer, for consumer to consume it.

1. GeneratePositivieReviews

Because one positive or negative review won’t make much of a difference for user to evaluate the real time feed experience. We have created a method where in at a time 30 positive reviews will be generated for that particular hotel to observe the change in review score and to simulate it to the user.

1. GenerateNegativeReviews

Because one positive or negative review won’t make much of a difference for user to evaluate the real time feed experience. We have created a method where in at a time 30 negative reviews will be generated for that particular hotel to observe the change in review score and to simulate it to the user.

Kafka Consumer Service:

There is a separate utility service working on another machine trying to pull Kafka server continuously to check if there is any new review available for it to consume.

Implementation:

