KAFKA

Kafka Producer and Consumer in Python

By Mahesh Mogal

Till now we have seen basics of Apache Kafka and created Producer and Consumer using Java. In this tutorial, we are going to build Kafka Producer and Consumer in Python. Along with that, we are going to learn about how to set up configurations and how to use group and offset concepts in Kafka.

Set up

We should have python installed on our machine for this tutorial. Also, we need to have access to Apache Kafka running on our device or some server. You can check how to install Apache Kafka on windows. Apart from this, we need python's *kafka* library to run our code. To fix this, on system run following command

```
pip install kafka
```

Kafka Producer

Let us start creating our own Kafka Producer. We have to import KafkaProducer from kafka library. We also need to give broker list of our Kafka server to Producer so that it can connect to the Kafka server. We also need to provide a topic name to which we want to publish messages. That is the minimal configuration that we need to give to create a Producer.

```
from kafka import KafkaProducer
bootstrap_servers = ['localhost:9092']
topicName = 'myTopic'
producer = KafkaProducer(bootstrap_servers = bootstrap_servers)
producer = KafkaProducer()
```

We can start sending messages to this topic using the following code.

```
ack = producer.send(topicName, b'Hello World!!!!!!!!')
metadata = ack.get()
print(metadata.topic)
print(metadata.partition)
```

The above code sends a message to the topic named 'myTopic' in Kafka server. However, what if that topic is not already present in the Kafka server? In such a case, Kafka creates a new topic with this name and publish messages to it. Convenient isn't it? But you should remember to check for any spelling mistakes in topic names.

If you want to set some more properties for your Producer or change its serialization format you can use the following lines of code.

```
producer = KafkaProducer(bootstrap_servers = bootstrap_servers, retries = 5,value_serializer=lambda m: json.dumps(m).
```

Kafka Consumer

As we are finished with creating Producer, let us now start building Consumer in python and see if that will be equally easy. After importing KafkaConsumer, we need to set up provide bootstrap server id and topic name to establish a connection with Kafka server.

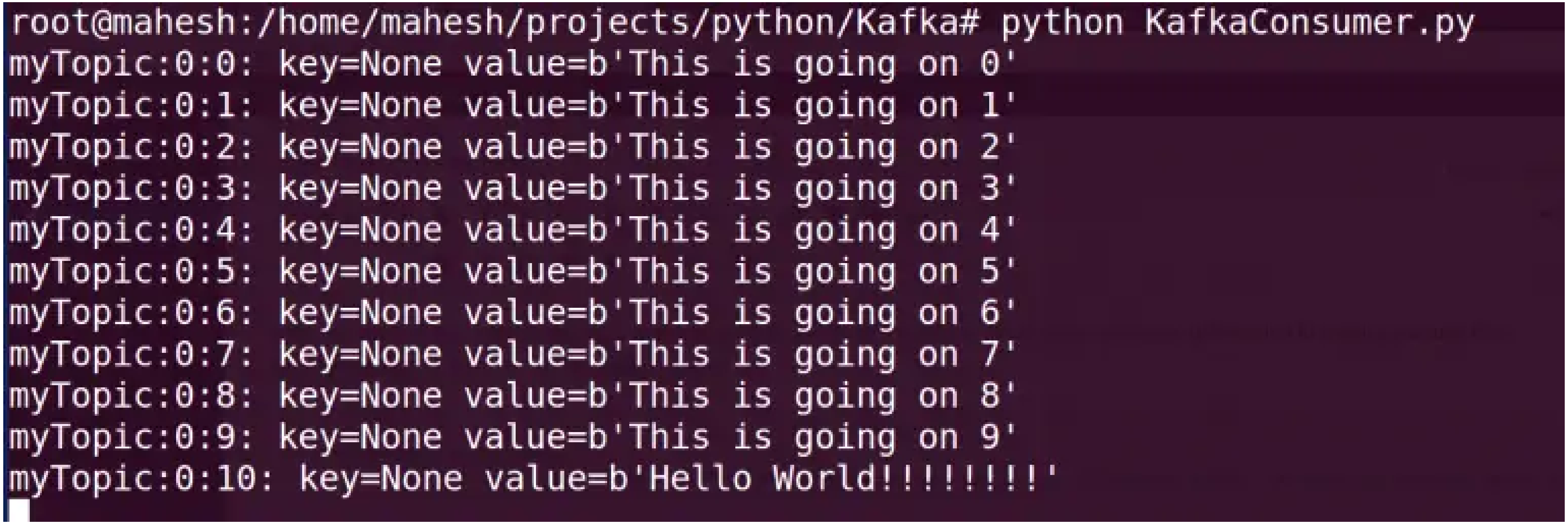
```
from kafka import KafkaConsumer
import sys
bootstrap_servers = ['localhost:9092']
topicName = 'myTopic'
consumer = KafkaConsumer (topicName, group_id = 'group1',bootstrap_servers = bootstrap_servers,
auto_offset_reset = 'earliest')
```

As we can see we need to set up which group consumer belongs to. Also, we need to specify offset from which this consumer should read messages from the topic. In the above case, we have specified auto_offset_reset to earliest which means this consumer will start reading messages from the beginning of the topic.

After this, we can start reading messages from a topic. Along with each message, we get some additional information like which partition that message belongs to, its offset in that partition and its key.

```
try:
    for message in consumer:
        print ("%s:%d:%d: key=%s value=%s" % (message.topic, message.partition,message.offset, message.key,message.va
except KeyboardInterrupt:
    sys.exit()
```

This will print output in the following format.



```
root@mahesh:/home/mahesh/projects/python/Kafka# python KafkaConsumer.py
myTopic:0:0: key=None value=b'This is going on 0'
myTopic:0:1: key=None value=b'This is going on 1'
myTopic:0:2: key=None value=b'This is going on 2'
myTopic:0:3: key=None value=b'This is going on 3'
myTopic:0:4: key=None value=b'This is going on 4'
myTopic:0:5: key=None value=b'This is going on 5'
myTopic:0:6: key=None value=b'This is going on 6'
myTopic:0:7: key=None value=b'This is going on 7'
myTopic:0:8: key=None value=b'This is going on 8'
myTopic:0:9: key=None value=b'This is going on 9'
myTopic:0:10: key=None value=b'Hello World!!!!!!!'
```

This is it. We have created our first Kafka consumer in python. We can see this consumer has read messages from the topic and printed it on a console.

Conclusion

We have learned how to create Kafka producer and Consumer in python. In the next articles, we will learn the practical use case when we will read live stream data from Twitter. Until then, keep learning.

#Kafka #Python #Streaming

← PREVIOUS
Kafka Consumer in Java

NEXT →
How to set up Cloudera Quickstart VM on windows

Similar Posts

Kafka topics – Create, List, Configure, Delete

By Mahesh Mogal December 7, 2019

Set, Get and Delete AWS S3 bucket policies

By Mahesh Mogal April 12, 2020

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name *

Email *

□ Save my name, email, and website in this browser for the next time I comment.

Post Comment

One Comment

Peter Ryan says:
April 1, 2021 at 8:14 AM

Unable to get this code to run!
I am using WIN 10 and Python 3.8 with IDLE.
I get timeouts since the code hangs
I installed kafka-python library – kafka did not work

Reply

STAY UPDATED WITH LATEST BLOGS

Email Address

Subscribe