Apache Kafka’da Veri İşleme Uygulaması

Kafka servislerini başlatalım:

```
sudo systemctl start zookeeper
sudo systemctl start kafka
```

▼ 1 - Kafka üzerinde 2 parçaya sahip ve replikasyon faktörü 1 olan `atscale` adında bir topic oluşturunuz.

```
kafka-topics.sh --bootstrap-server localhost:9092 --create --topic atscale --partitions 2 --replication-factor 1
```

▼ 2 - Kafka üzerindeki mevcut `topic` leri listeyiniz.

```
kafka-topics.sh --bootstrap-server localhost:9092 --list
```

▼ 3 - `atscale` topic özelliklerini ekrana yazdırınız.

```
kafka-topics.sh --bootstrap-server localhost:9092 --describe --topic atscale
```

▼ 4 - data-generator kullanarak

`https://raw.githubusercontent.com/erkansirin78/datasets/master/Churn_Modelling.csv` veri setini Kafka `churn topic`’e aşağıdaki koşulları sağlayarak produce/consume ediniz.

- Kafkaya gönderdiğiniz mesajın anahtarı `CustomerId` olsun.

- Üç farklı consumer'a sahip `churn_group` adındaki bir consumer group altında mesajları consume ediniz.
  - Her bir consumer için farklı bir terminal kullanınız.
  - Consumer olarak `kafka-console-consumer.sh` kullanınız.

- Tüm veri setinin bitmesini beklemeyiniz. yaklaşık 500 satırı gözlemleyiniz.

Cevap:

- Topic oluştur.

```
kafka-topics.sh --bootstrap-server localhost:9092 \
--create --topic churn \
--partitions 3 \
--replication-factor 1
```

data-generator indir ve dizin değiştir.

```
git clone https://github.com/erkansirin78/data-generator.git
cd data-generator
```

Virtualenv aktif et.

```
source ~/venvspark/bin/activate
```

Veri setini indir.

```
wget -P ~/datasets https://raw.githubusercontent.com/erkansirin78/datasets/master/Churn_Modelling.csv
```

Consumer'lar için 3 adet terminal aç. Her birinde; çalıştır.

```
kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic churn --group churn_group
```

- Virtualenv aktif ettiğin terminalde data-generator ile veri setini produce etmeye başla

```
python dataframe_to_kafka.py -i ~/datasets/Churn_Modelling.csv -t churn -k 1
```

▼ 5 - `atscale` ve `churn` topic'lerini siliniz.

```
kafka-topics.sh --bootstrap-server localhost:9092 --delete --topic  atscale,churn
```

▼ 6 - Python Kafka kütüphanesini kullanarak aşağıdaki işleri yapınız.

- Türkiye'nin coğrafi bölgelerinin adlarını her birinin başında belirleyeceğiniz rakamları key olarak kullanarak belirlediğiniz bir topic'e produce ediniz. Örneğin 1 Marmara, 2 Ege şeklinde.
- Consumer ile ekrana key, value, partition, timestamp bilgilerini aşağıdaki gibi yazdırınız.

```
Key: 1, Value: Marmara, Partition: 0, TS: 1613224639352
Key: 4, Value: İç Anadolu, Partition: 1, TS: 1613224654849
Key: 3, Value: Akdeniz, Partition: 2, TS: 1613224661486
Key: 2, Value: Ege, Partition: 2, TS: 1613224667044
```

- İki tane terminal açınız ve virtualenv aktif hale getirin, python shell'i açın.

```
source ~/venvspark/bin/activate
python
```

```
- Consumer terminalinde.

from kafka import KafkaConsumer
consumer = KafkaConsumer('test1',group_id='group1',bootstrap_servers=['localhost:9092'])

for message in consumer:
  print("Key: {}, Value: {}, Partition: {}, TS: {} ".format(message.key.decode(), message.value.decode(), message.partition, message.timestamp))
```

```
- Producer terminalinde

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

producer.send("test1", key="1".encode(), value="Marmara".encode())

producer.send("test1", key="2".encode(), value="Ege".encode())

producer.send("test1", key="3".encode(), value="Akdeniz".encode())

producer.send("test1", key="4".encode(), value="İç Anadolu".encode())
```