Самостоятельное задание

Обработка данных датчиков в реальном времени с помощью Kafka и Python

Цель:

Создать конвейер данных с использованием Apache Kafka и Python для обработки данных в реальном времени от нескольких датчиков. Данные будут разделены на основе идентификаторов датчиков для параллельной обработки.

Описание задания:

Задание выполняется самостоятельно, не требует проверки, потренируйтесь в отправке данных через Apache Kafka, перед тем, как приступать к заданию по проекту.

Необходимо настроить Kafka и код в Python следующим образом:

1. Настройка Kafka:

Запустите брокер Kafka и создайте топик. Топик имеет название «`sensor-data`» с 3-мя партициями.

2. Настройка producer

Напишите скрипты Python, которые будут выступать в качестве продюсеров, имитирующих различные датчики. То есть необходимо генерировать рандомные данные **температуры в Цельсиях, атмосферного давления в миллиметрах ртутного столба и влажности в процентах.** Каждый продюсер должен генерировать данные по каждому датчику, отправлять их в топик `sensor-data` с соответствующим датчику (температуры, давления или влажности) key_id в сообщении в бесконечном цикле с указанием времени (аналогично тому, как в лекции).

Для примера один продюсер, для последующих меняется значение параметра sensor_id на 2, 3 итд

```
…
topic = 'sensor-data'

sensor_id = 1

if __name__=='__main__':
    while True:
        temperature = random.uniform(20, 30)
        message = f'Sensor ID: {sensor_id}, Temperature: {temperature}'
        producer.send(topic, key=str(sensor_id).encode('utf-8'), value=message.encode('utf-8'))
        producer.flush()
```

3. Настройка consumer

Консьюмер должен подписаться на топик sensor-data и обрабатывать входящие сообщения.

Реализуйте логику для извлечения данных так, чтобы каждый в консьюмер группе соотносился с одной партицией (по одному датчику на каждый). Пример кода:

```
…

consumer = KafkaConsumer(topic,
                         bootstrap_servers=['localhost:9092'],
                         group_id='sensor-consumer-group',
                         auto_offset_reset='earliest',
                         enable_auto_commit=True,
                         consumer_timeout_ms=1000)
…
```

Если вы запустите несколько экземпляров этого скрипта с одним и тем же group_id, они будут работать вместе как часть одной и той же группы потребителей для обработки сообщений из темы данных датчиков. Необходимо чтобы каждый экземпляр считывал ассоциированную с одним из датчиков партицию. Делается это так:

```
from kafka import KafkaConsumer, TopicPartition

topic = 'sensor-data'

partitions = TopicPartition(topic, 1)

...

consumer.assign([partition])

...
```

Инструменты, которые пригодятся для выполнения:

Python версии не меньше 3.8

Библиотека kafka-python

Образ Kafka https://hub.docker.com/r/confluentinc/confluent-local

Docker https://www.docker.com/get-started/

UI для Kafka https://www.geeksforgeeks.org/how-to-install-configure-conduktor-tool-for-apache-kafka/

Для подключения к Conductor используйте данные контейнера: порт прописан в docker-compose.yml файле, ip адрес можно получить как в инструкции здесь:

https://www.freecodecamp.org/news/how-to-get-a-docker-container-ip-address-explained-with-examples/

По итогу выполнения задания вам необходимо проверить, что 1 продюсер с 3-мя метриками, отправляет данные в топик с 3-мя партициями по этим метрикам соответственно и 3 консьюмера в одной консьюмер группе (с одинаковым `group_id`) получают данные каждый по одному датчику (партиции), запущенные в разных терминалах.

Как вам урок? 😠 😐 🔥

Изучил, далее »