**Homework Assignment "Spark Streaming"**

**1. 1\_create\_topics.py**

* **Призначення**: Створення двох Kafka-топіків: iot\_sensors\_data (для даних датчиків) і iot\_alerts (для повідомлень про алерти).
* **Основні етапи:**
  1. Підключення до Kafka Admin Client із використанням захищених облікових даних (username, password).
  2. Визначення параметрів топіків: 2 партиції та 1 репліка.
  3. Створення топіків або логування, якщо вони вже існують.
  4. Закриття клієнта Kafka Admin Client.
* **Скріншот (topics\_created\_1.jpg):**
  1. Успішно створено топіки iot\_sensors\_data та iot\_alerts. Клієнт закрито без помилок.

A screenshot of a computer program

Description automatically generated

**2. 2\_producer.py**

* **Призначення:** Генерація випадкових даних із сенсорів IoT і відправлення їх у Kafka-топік iot\_sensors\_data.
* **Основні етапи:**
  1. Створення Kafka Producer із конфігурацією для авторизації.
  2. Генерація випадкових даних сенсорів (id, temperature, humidity, timestamp) кожну секунду.
  3. Відправлення даних у JSON-форматі до топіка iot\_sensors\_data.
  4. Логування успішно відправлених повідомлень (зеленим кольором для видимості).
  5. Обробка переривань користувача та закриття продюсера.
* **Скріншот (producer\_messages\_2.jpg):**
  1. Показано згенеровані та відправлені дані сенсорів у топік iot\_sensors\_data

A screenshot of a computer program

Description automatically generated

**3. 3\_spark\_kafka.py**

* **Призначення:** Споживання даних сенсорів із Kafka, їх агрегація, перевірка умов алертів та відправка алертів у Kafka-топік iot\_alerts.
* **Основні етапи:**
  1. Зчитування даних із топіка iot\_sensors\_data.
  2. Парсинг і трансформація JSON-даних, конвертація поля timestamp у тип Spark TIMESTAMP.
  3. Агрегація даних із використанням *Sliding window* (1 хвилина, з перекриттям у 30 секунд).
  4. Об’єднання агрегованих даних із умовами алертів (CSV-файл) та фільтрація за заданими порогами.
  5. Стримінг повідомлень алертів у консоль і відправлення їх у топік iot\_alerts.
* **Скріншоти (spark\_kafka\_1.jpg і spark\_kafka\_2.jpg):**
  1. **spark\_kafka\_1.jpg:** Показано, як Spark успішно споживає дані, агрегує їх і генерує алерти.
  2. **spark\_kafka\_2.jpg:** Демонструє результат обробки: алерти, виведені у консоль та подальші оброблені батчі.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated