

Кроки виконання завдання:

1. Створення topics. Було створено три топіка: oie_building_sensors, oie_humidity_alerts, oie_temperature_alerts на локальному сервері. Для цього створено контейнер kafka через файл docker-compose.yml

```

34     print("My topics: ---")
35     my_name = "oie"
36     [print(topic) for topic in admin_client.list_topics() if my_name in topic]
37     print("-----")
38
39     # Закриваємо зв'язку з клієнтом
40     admin_client.close()

```

```

(spark-env) oie@oie-KOP9YG7LWT kafka % docker-compose ps
(spark-env) oie@oie-KOP9YG7LWT kafka % python Goit_HW_topic5_create_topic.py
An error occurred: [Error 36] TopicAlreadyExistsError: Request 'CreateTopicsRequest_v3:create_topic_requests=[(topic='oie_building_sensors', num_partitions=3, replication_factor=1, config=[{}]), (topic='oie_humidity_alerts', num_partitions=3, replication_factor=1, config=[{}]), (topic='oie_temperature_alerts', num_partitions=3, replication_factor=1, config=[{}])]' failed with response 'CreateTopicsResponse_v3(throttle_time_ms=0, topic_errors=[(topic='oie_building_sensors', error_code=36, error_message="Topic 'oie_building_sensors' already exists."), (topic='oie_temperature_alerts', error_code=36, error_message="Topic 'oie_temperature_alerts' already exists."), (topic='oie_humidity_alerts', error_code=36, error_message="Topic 'oie_humidity_alerts' already exists.")])'.
(spark-env) oie@oie-KOP9YG7LWT kafka %

```

2. Створено та запущено в роботу spark.readStream (підписаний на oie_building_sensors). Що приймає сповіщення від producer з даними, отриманими від датчиків, та обробляє отримані дані (рахує середні значення). Також spark.stream завантажує з таблиці alerts умови фільтрування.

```

30     # Відправлення повідомлення в топік
31     try:
32         data = {
33             "timestamp": time.time(), # Часова мітка
34             "sensor_building_id": sensor_building_id, # Випадковий ідентифікатор будівлі
35             "temperature": round(random.uniform(25.0, 45.0), 2), # Випадкова температура
36             "humidity": round(random.uniform(15.0, 85.0), 2) # Випадкова вологість
37         }
38         key_bytes = str(sensor_building_id).encode('utf-8')
39         producer.send(building_sensors, key=key_bytes, value=data)
40         producer.flush()
41         print("Message {} sent to topic '{}(building_sensors)' successfully.".format(time.sleep(2)))
42     except Exception as e:
43         print("An error occurred: {}".format(e))
44
45     producer.close() # Закриваємо producer

```

```

"latestOffset": {
    "oie_building_sensors": {
        "2": 118,
        "3": 24,
        "4": 216
    }
},
"maxInputRows": 0,
"inputRowsPerSecond": 0.0,
"processedRowsPerSecond": 0.0,
"maxOffset": "0.0",
"avgOffsetBehindLatest": "0.0",
"maxOffsetBehindLatest": "0.0",
"minOffsetBehindLatest": "0.0"
}
}
"sink": {
    "description": "ForeachBatchSink",
    "numOutputRows": 1
}
}

```

У разі відповідності параметрів завантаженім умовам фільтрування, формується alert сповіщення та записується в kafka в топіки (oie_humidity_alerts, oie_temperature_alerts) в залежності від того, який з параметрів відповідає заданим умовам.

3. Створено та запущено в роботу producer. Що імітує роботу двох датчиків (два запуски у різних терміналах), генерує сповіщення з даними, отриманими від датчиків , та відправляє сповіщення в топік oie_building_sensors, який слухає spark.readStream (підписаний на oie_building_sensors).

```

30     # Відправлення позначення в топік
31     try:
32         data = {
33             "timestamp": time.time(), # Часова мітка
34             "sensor_building_id": sensor_building_id, # Випадковий ідентифікатор будівлі
35             "temperature": round(random.uniform(25.0, 45.0), 2), # Випадкова температура
36             "humidity": round(random.uniform(15.0, 85.0), 2) # Випадкова вологість
37         }
38
39         key_bytes = str(sensor_building_id).encode('utf-8')
40         producer.send(building_sensors, key=key_bytes, value=data)
41         producer.flush() # Очикування, поки всі позначення будуть відправлені
42         print("Message (%d) sent to topic '%s' successfully." % (i, building_sensors))
43     except Exception as e:
44         print("An error occurred: (%e)" % e)
45
46     producer.close() # Закриття producer

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

(spark-env) oie@oie-OptiPlex-7040:~/kafka_streaming % /users/oie/miniforge3/envs/spark-env/bin/python /Users/oie/Documents/my_pyspark_repository/kafka_streaming/Goit_HW_topic6_create_topic.py
(spark-env) oie@oie-OptiPlex-7040:~/kafka_streaming % /users/oie/miniforge3/envs/spark-env/bin/python /Users/oie/Documents/my_pyspark_repository/kafka_streaming/Goit_HW_topic6_producer.py
Kafka Producer created successfully.
Message 0 sent to topic 'oie_building_sensors' successfully.
Message 1 sent to topic 'oie_building_sensors' successfully.
Message 2 sent to topic 'oie_building_sensors' successfully.
Message 3 sent to topic 'oie_building_sensors' successfully.
Message 4 sent to topic 'oie_building_sensors' successfully.
Message 5 sent to topic 'oie_building_sensors' successfully.
Message 6 sent to topic 'oie_building_sensors' successfully.
Message 7 sent to topic 'oie_building_sensors' successfully.
Message 8 sent to topic 'oie_building_sensors' successfully.
Message 9 sent to topic 'oie_building_sensors' successfully.
Message 10 sent to topic 'oie_building_sensors' successfully.
Message 11 sent to topic 'oie_building_sensors' successfully.
Message 12 sent to topic 'oie_building_sensors' successfully.
Message 13 sent to topic 'oie_building_sensors' successfully.
Message 14 sent to topic 'oie_building_sensors' successfully.
Message 15 sent to topic 'oie_building_sensors' successfully.

```

4. Створено та запущено в роботу ще один consumer (підписаний на oie_humidity_alerts, oie_temperature_alerts), що приймає сповіщення від kafka з alert сповіщеннями за заданим умовами завдання форматом, та виводить їх у консоль.

```

30     # Відправлення позначення в топік
31     try:
32         data = {
33             "timestamp": time.time(), # Часова мітка
34             "sensor_building_id": sensor_building_id, # Випадковий ідентифікатор будівлі
35             "temperature": round(random.uniform(25.0, 45.0), 2), # Випадкова температура
36             "humidity": round(random.uniform(15.0, 85.0), 2) # Випадкова вологість
37         }
38
39         key_bytes = str(sensor_building_id).encode('utf-8')
40         producer.send(building_sensors, key=key_bytes, value=data)
41         producer.flush() # Очикування, поки всі позначення будуть відправлені

```

Average Temperature: 34.51133333333334°C
Average Humidity: 49.10866666666666%
Alert Code: 109
Message: It's too dry

ALERT: Humidity out of range
Topic: oie_humidity_alerts
Window: [2025-11-14T21:10:00.000Z, 2025-11-14T21:11:00.000Z]
Sensor ID: 2
Average Temperature: 34.51133333333334°C
Average Humidity: 49.10866666666666%
Alert Code: 109
Message: It's too wet

ALERT: Humidity out of range
Topic: oie_humidity_alerts
Window: [2025-11-14T21:10:30.000Z, 2025-11-14T21:10:30.000Z]
Sensor ID: 2
Average Temperature: 35.375000000000001°C
Average Humidity: 51.11166666666666%
Alert Code: 109
Message: It's too wet

ALERT: Temperature out of range
Topic: oie_temperature_alerts
Window: [2025-11-14T21:09:30.000Z, 2025-11-14T21:10:00.000Z]
Sensor ID: 2
Average Temperature: 35.375000000000001°C
Average Humidity: 51.11166666666666%
Alert Code: 102
Message: It's too wet

```

30     # Відправлення позначення в топік
31     try:
32         data = {
33             "timestamp": time.time(), # Часова мітка
34             "sensor_building_id": sensor_building_id, # Випадковий ідентифікатор будівлі
35             "temperature": round(random.uniform(25.0, 45.0), 2), # Випадкова температура
36             "humidity": round(random.uniform(15.0, 85.0), 2) # Випадкова вологість
37         }
38
39         key_bytes = str(sensor_building_id).encode('utf-8')
40         producer.send(building_sensors, key=key_bytes, value=data)
41         producer.flush() # Очикування, поки всі позначення будуть відправлені
42         time.sleep(2)
43     except Exception as e:
44         print("An error occurred: (%e)" % e)
45
46     producer.close() # Закриття producer

```

ALERT: Humidity out of range
Topic: oie_humidity_alerts
Window: [2025-11-14T21:05:30.000Z, 2025-11-14T21:06:00.000Z]
Sensor ID: 8
Average Temperature: 35.23199999999999°C
Average Humidity: 57.45699999999994%
Alert Code: 102
Message: It's too wet

ALERT: Humidity out of range
Topic: oie_humidity_alerts
Window: [2025-11-14T21:05:00.000Z, 2025-11-14T21:06:00.000Z]
Sensor ID: 8
Average Temperature: 36.953999999999995°C
Average Humidity: 46.85240000000001%
Alert Code: 102
Message: It's too wet