17 ДЗ - Делаем НЕ больно с Apache Kafka

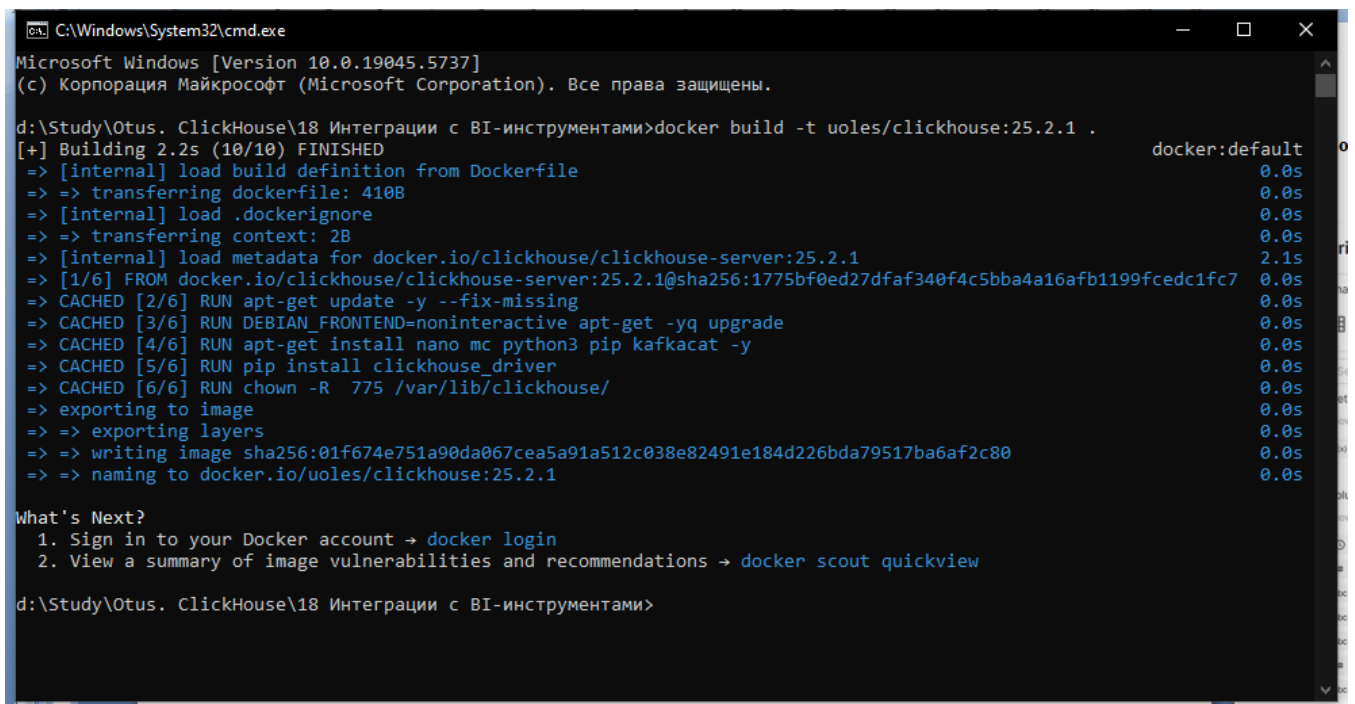## Собираем Dockerfile для clickhouse.

Dockerfile:

```
FROM clickhouse/clickhouse-server:25.2.1
MAINTAINER Maksim Kulikov max.uoles@rambler.ru

RUN apt-get update -y --fix-missing
RUN DEBIAN_FRONTEND=noninteractive apt-get -yq upgrade
RUN apt-get install nano mc python3 pip kafkacat –y
RUN pip install clickhouse_driver

EXPOSE 8123 9000
ENTRYPOINT ["/entrypoint.sh"]
```

Собираем образ командой:

```
docker build -t uoles/clickhouse:25.2.1 .
```


Собираем docker-compose для запуска ClickHouse, Kafka, Kafka-ui, Zookeeper

За основу брал статью https://habr.com/ru/articles/753398/

Docker-compose.yml:

```
version: "3.6"

name: kafka
services:
  clickhouse-server:
    container_name: uoles-clickhouse-udf-25
    image: uoles/clickhouse:25.2.1
    environment:
      CLICKHOUSE_DB: my_database
      CLICKHOUSE_USER: username
```

```yaml
    CLICKHOUSE_DEFAULT_ACCESS_MANAGEMENT: 1
    CLICKHOUSE_PASSWORD: password
  ports:
   - "18123:8123"
   - "19000:9000"
  volumes:
   - ./clickhouse/config.xml:/etc/clickhouse-server/config.xml
   - ./clickhouse/zookeeper-servers.xml:/etc/clickhouse-server/conf.d/zookeeper-servers.xml
  ulimits:
   nofile:
    soft: 262144
    hard: 262144
  depends_on:
   - kafka
  links:
   - kafka

zookeeper:
  image: confluentinc/cp-zookeeper:6.2.4
  healthcheck:
   test: [ "CMD", "nc", "-vz", "localhost", "2181" ]
   interval: 10s
   timeout: 3s
   retries: 3
  environment:
   ZOOKEEPER_CLIENT_PORT: 2181
   ZOOKEEPER_TICK_TIME: 2000
  ports:
   - 22181:2181

kafka:
  image: confluentinc/cp-kafka:6.2.4
  depends_on:
   zookeeper:
    condition: service_healthy
  ports:
   - 29092:29092
  healthcheck:
   test: [ "CMD", "nc", "-vz", "localhost", "9092" ]
   interval: 10s
   timeout: 3s
   retries: 3
  environment:
   KAFKA_BROKER_ID: 1
   KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
   KAFKA_LISTENERS: OUTSIDE://:29092,INTERNAL://:9092
   KAFKA_ADVERTISED_LISTENERS: OUTSIDE://localhost:29092,INTERNAL://kafka:9092
   KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: INTERNAL:PLAINTEXT,OUTSIDE:PLAINTEXT
   KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
   KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1

kafka-ui:
  image: provectuslabs/kafka-ui
  container_name: kafka-ui
  ports:
   - "8089:8080"
  restart: always
```
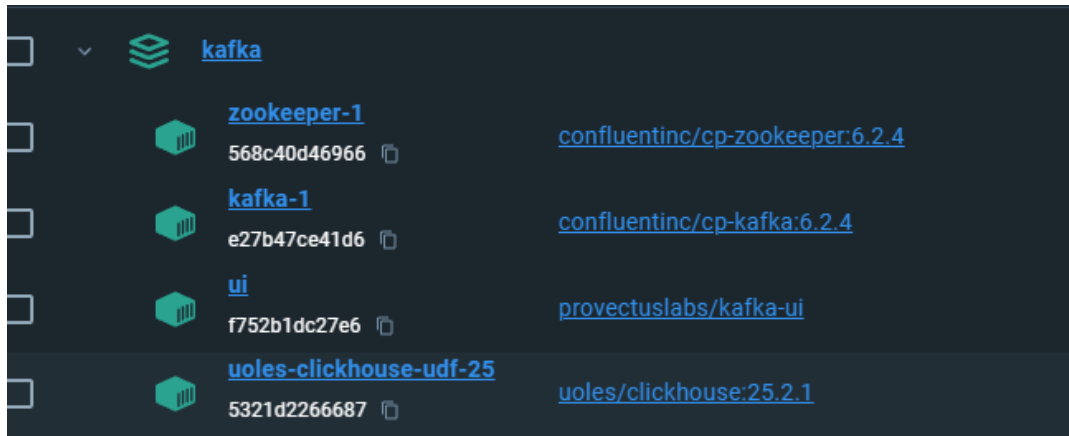
```
    depends_on:
      kafka:
        condition: service_healthy
    environment:
      KAFKA_CLUSTERS_0_NAME: local
      KAFKA_CLUSTERS_0_BOOTSTRAPSERVERS: kafka:9092
```

Поднимаем приложения командой docker-compose up –d:


Создаем таблицы и мат.вью.

```
CREATE TABLE kafka_test(
        message String
)
ENGINE = MergeTree
ORDER BY message;

CREATE TABLE kafka_test_queue(
        message String
)
ENGINE = Kafka
SETTINGS
kafka_broker_list = 'kafka:9092',
kafka_topic_list = 'test_clickhouse_topic',
kafka_group_name = 'test_group',
kafka_format = 'CSV',
kafka_num_consumers = 1,
kafka_skip_broken_messages = 10,
kafka_thread_per_consumer = 0;

CREATE MATERIALIZED VIEW kafka_test_queue_mv TO kafka_test(
        message String
) AS SELECT message FROM kafka_test_queue;
```
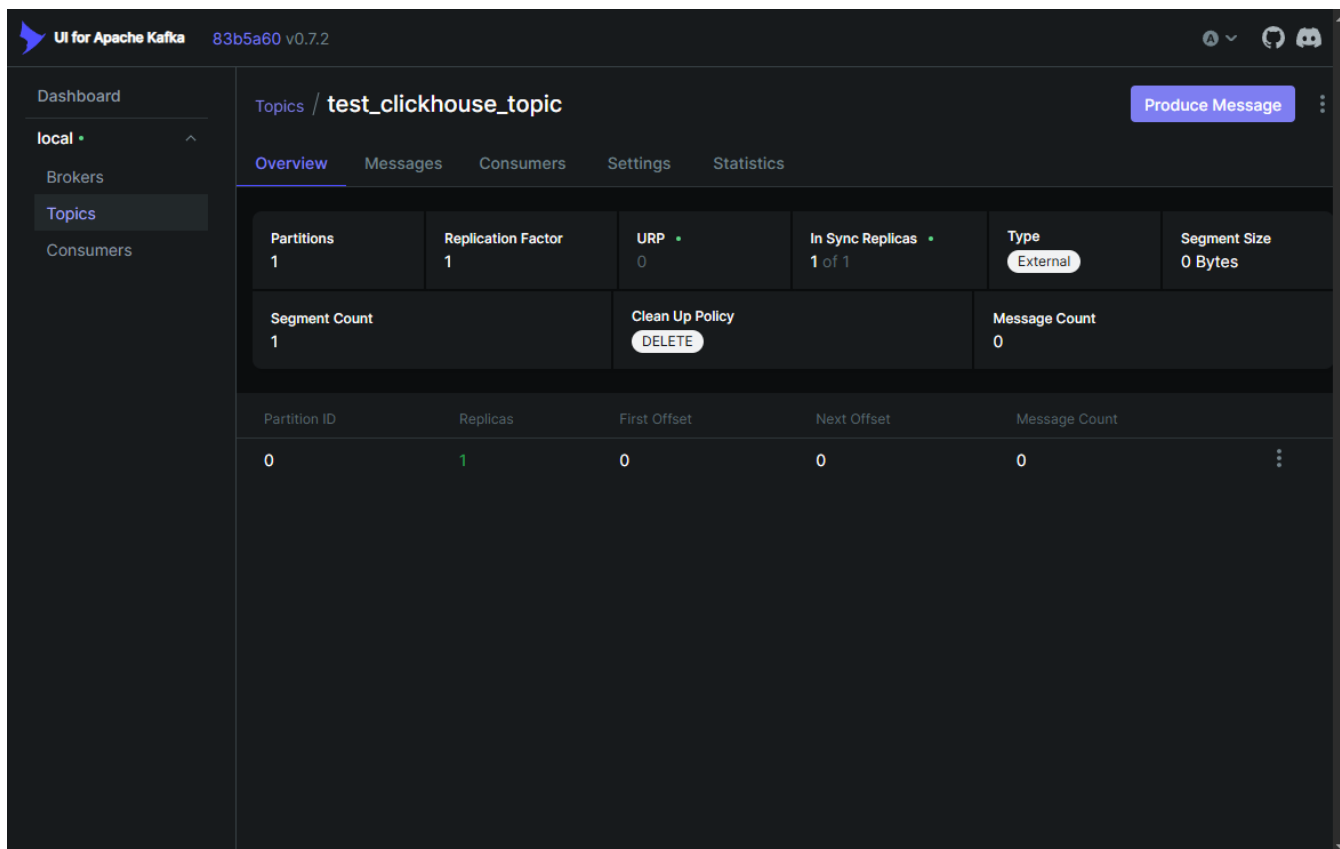
Настраиваем топик.

kafka-ui страничка - http://localhost:8089/ Приложения для мониторинга кафки. Заходим и вручную
создаем топик - test_clickhouse_topic.
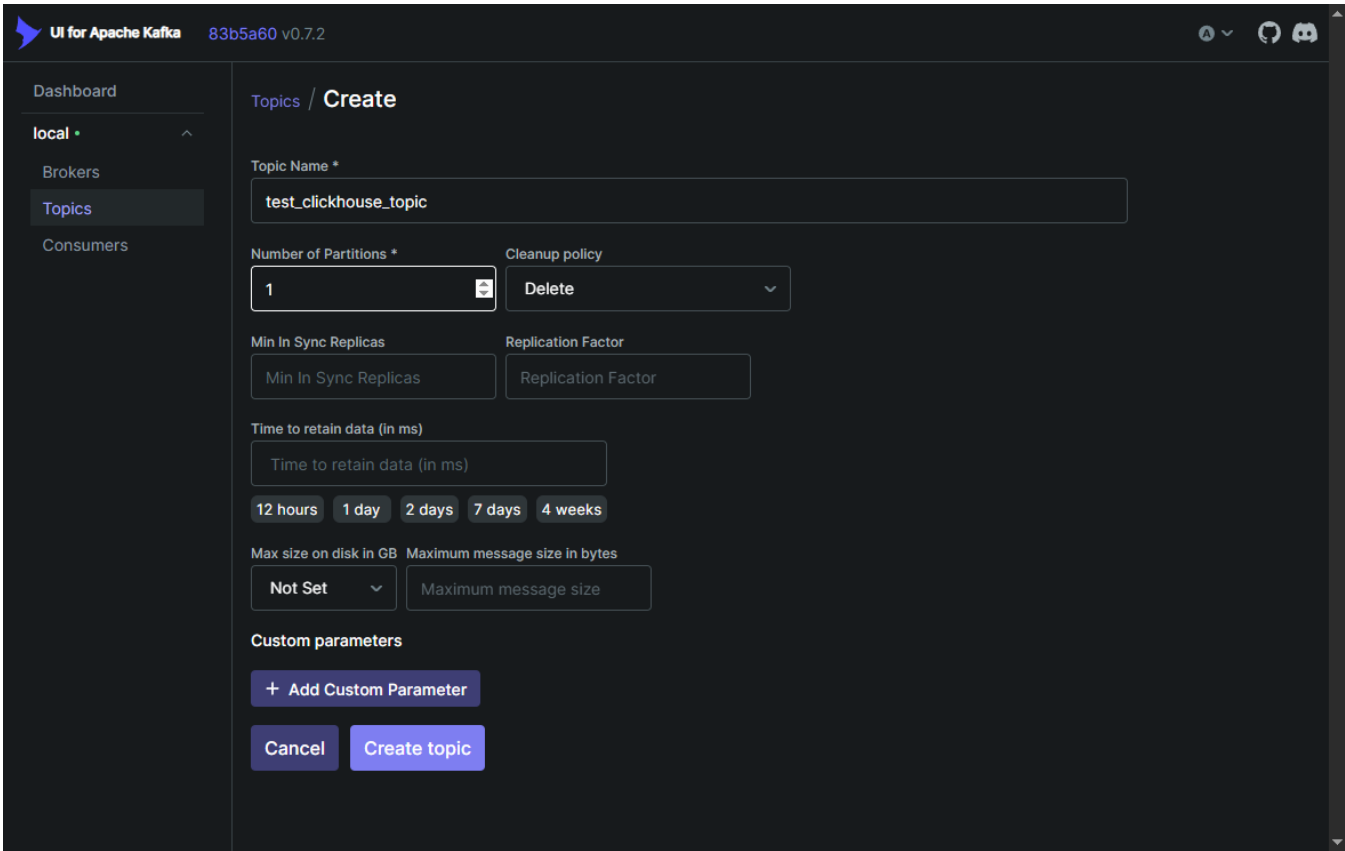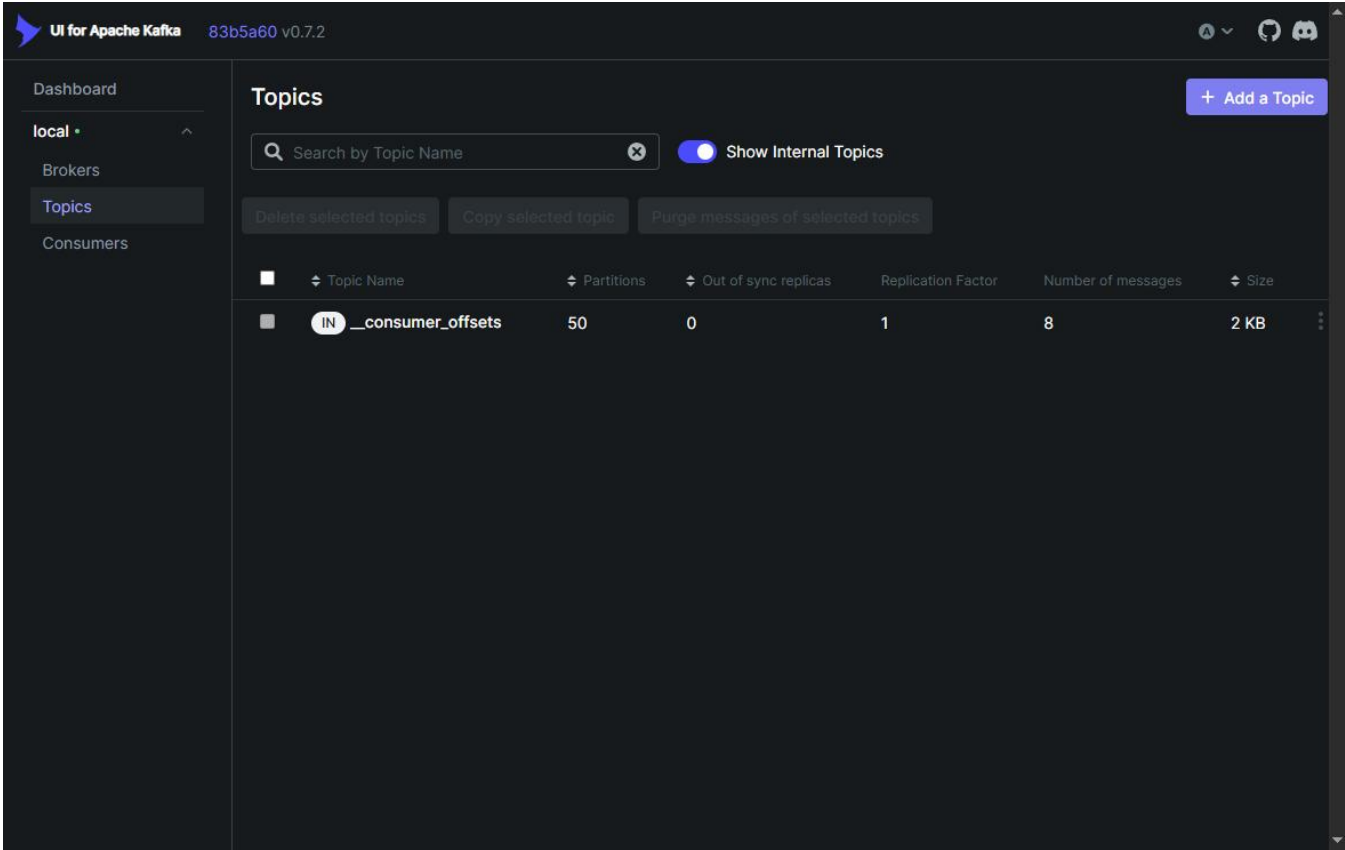
Посылаем сообщения в кафку.

Заходим в контейнер clickhouse
docker exec -it uoles-clickhouse-udf-25 bash

отправляем сообщения командой
echo "test message 12" | kafkacat -P -b kafka:9092 -t test_clickhouse_topic –Z

слушаем командой
kafkacat -C -b kafka:9092 -t test_clickhouse_topic -Z -K:

Dashboard

local ·

Brokers

Topics

Consumers

Topics / **test_clickhouse_topic**

Overview   Messages   Consumers   Settings   Statistics

Partitions	Replication Factor	URP ·	In Sync Replicas ·	Type
1	1	0	**1** of 1	External

Segment Count	Clean Up Policy	Message Count
1	DELETE	4

Partition ID	Replicas	First Offset	Next Offset	Message Count
0	1	0	4	4

```
27
28
29  ⊖ select *
30    from kafka_test_queue_mv
31
32
```

kafka_test_queue_mv 1 ✕

select * from kafka_test_queue_mv   Введите SQL выражение чтобы отф

	A-Z message ▼
1	test message
2	test message 12
3	test message 123
4	test message 1234

Таблица

Текст