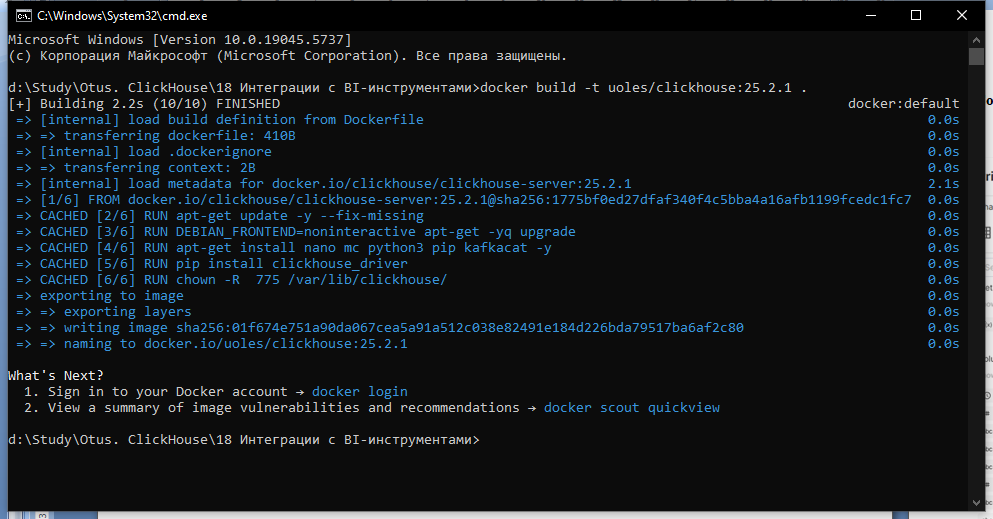
**17 ДЗ - Делаем НЕ больно с Apache Kafka**

**Собираем Dockerfile для clickhouse.**

Dockerfile:  
  
FROM clickhouse/clickhouse-server:25.2.1   
MAINTAINER Maksim Kulikov [max.uoles@rambler.ru](mailto: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

build: .

image: uoles/clickhouse:25.2.1

environment:

CLICKHOUSE\_DB: my\_database

CLICKHOUSE\_USER: username

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;

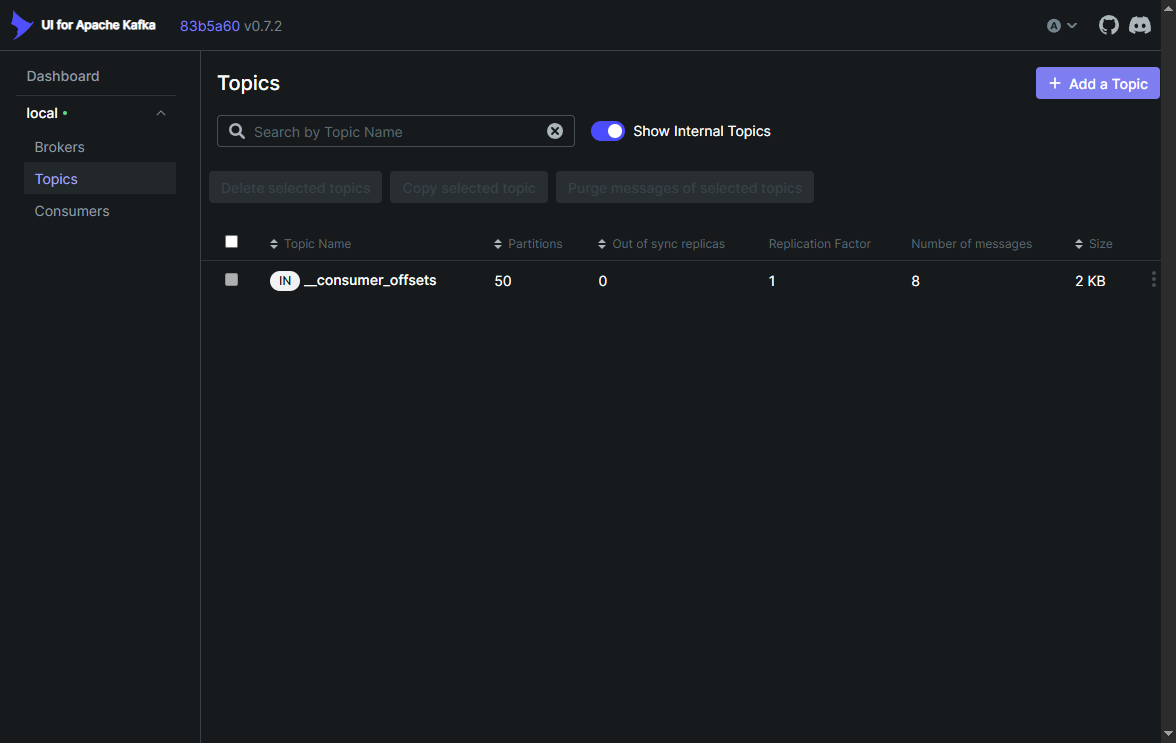
CREATE MATERIALIZED VIEW kafka\_test\_queue\_mv TO kafka\_test(

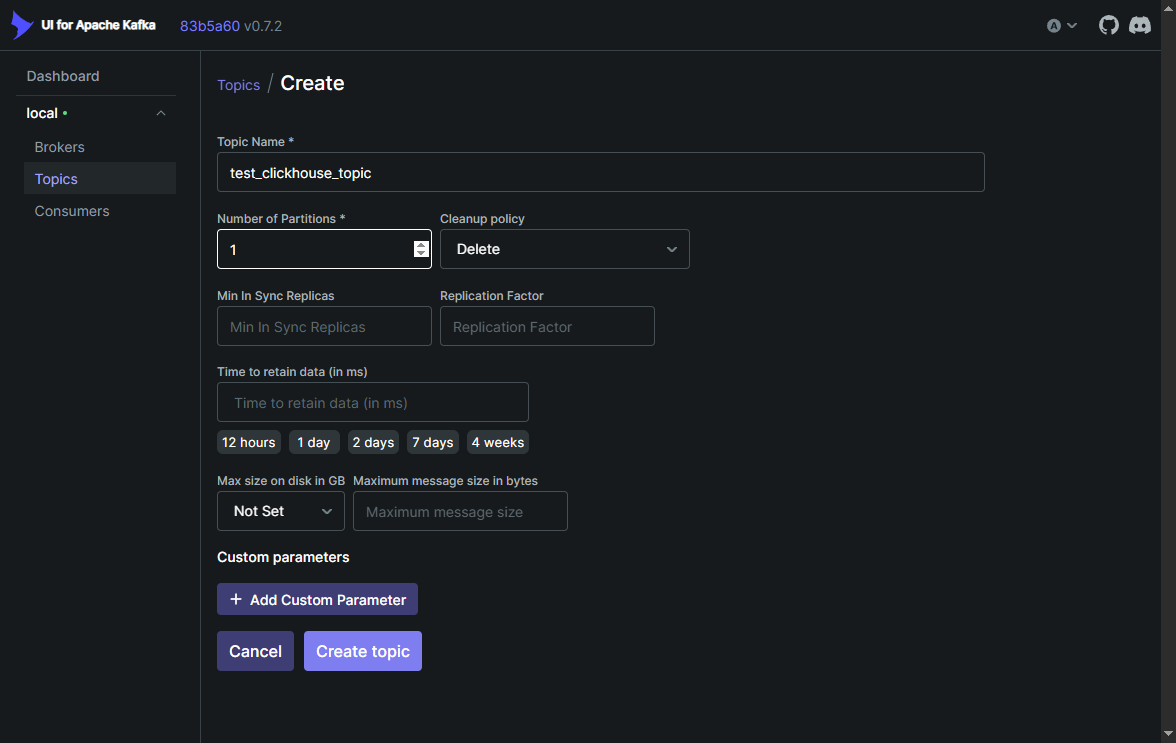
message String

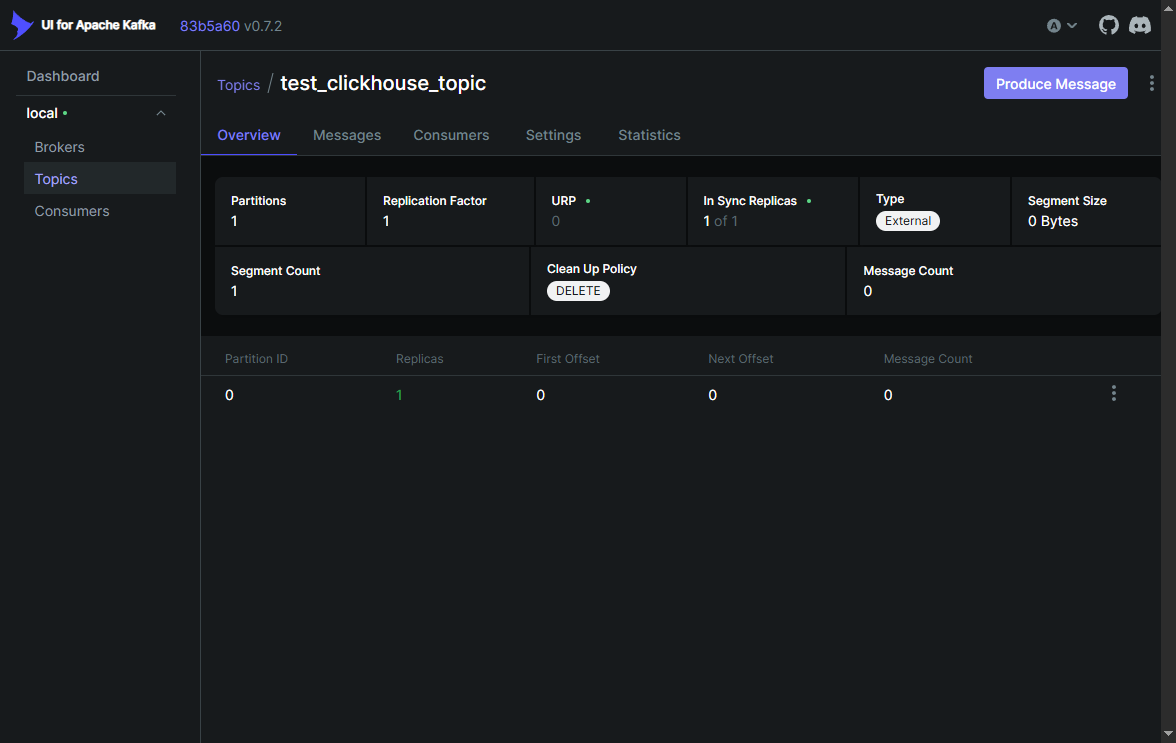
) AS SELECT message FROM kafka\_test\_queue;

**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:

