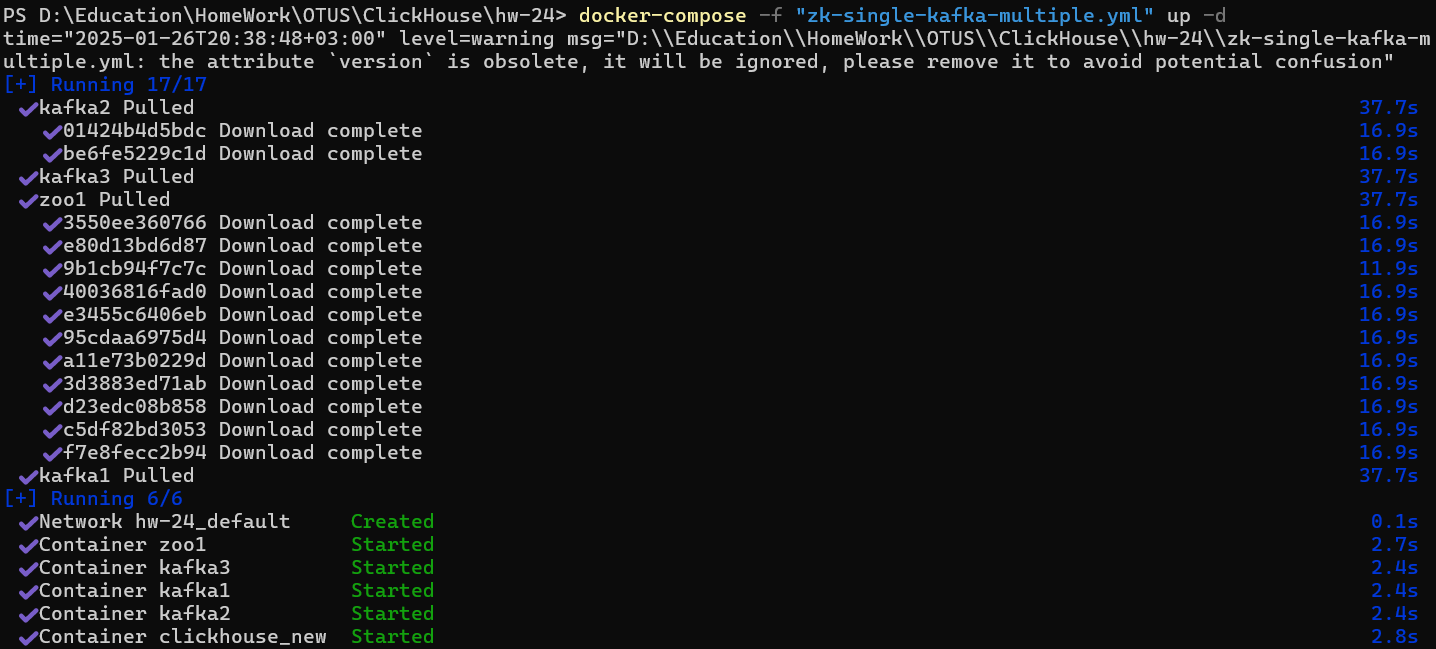
* Установка Kafka с помощью Docker

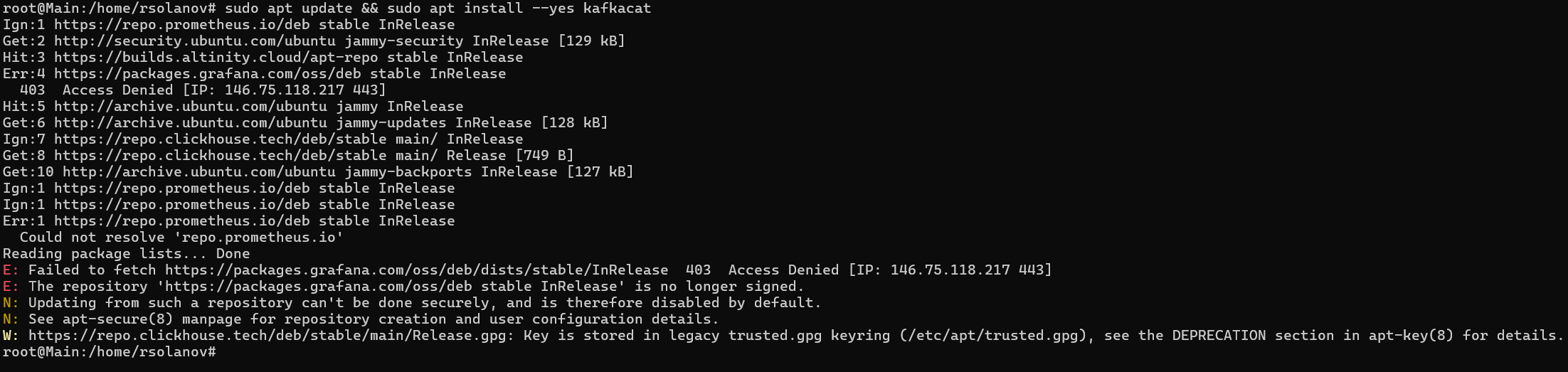
В docker-compose файл добавлен ClickHouse

docker-compose -f "zk-single-kafka-multiple.yml" up -d



* Установка kafkacat

sudo apt update && sudo apt install --yes kafkacat



* Создание таблицы с движком Kafka

**CREATE** **TABLE** kafka\_table

(

key **UInt64**,

value **String**

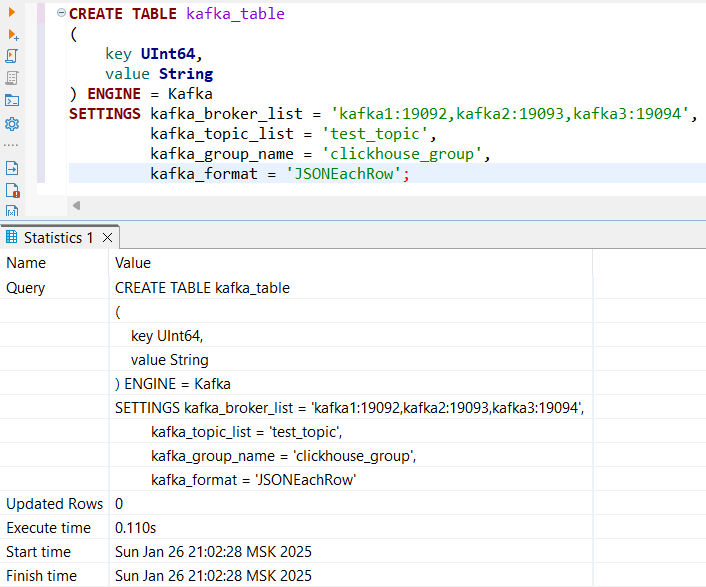
) **ENGINE** = Kafka

**SETTINGS** kafka\_broker\_list = 'kafka1:19092,kafka2:19093,kafka3:19094',

kafka\_topic\_list = 'test\_topic',

kafka\_group\_name = 'clickhouse\_group',

kafka\_format = 'JSONEachRow';



* Создание таблицы с движком MergeTree

**CREATE** **TABLE** mergetree\_table

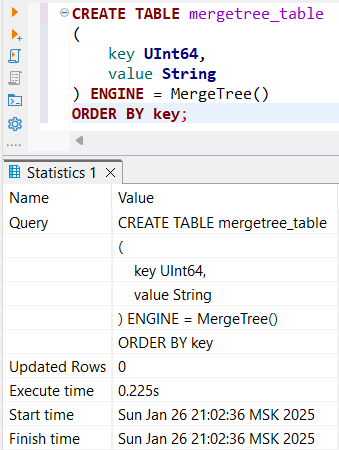
(

key **UInt64**,

value **String**

) **ENGINE** = MergeTree()

**ORDER** **BY** **key**;

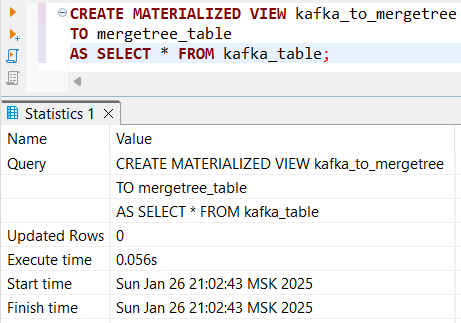


* Создание материализованного представления

**CREATE** **MATERIALIZED** **VIEW** kafka\_to\_mergetree

**TO** mergetree\_table

**AS** **SELECT** \* **FROM** kafka\_table;



* Создание топика Kafka

docker exec -it kafka1 /bin/bash

cd /usr/bin

kafka-topics --create --topic test-topic --bootstrap-server kafka1:19092 --partitions 3 --replication-factor 3



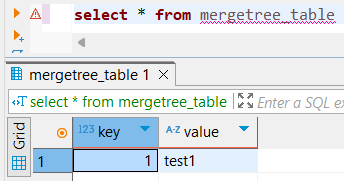
* Отправка сообщений в Kafka

echo '{"key": 1, "value": "test1"}' | kafkacat -b host.docker.internal:9092 -t test\_topic



* Проверка получения сообщений от kafka и запись в ClickHouse

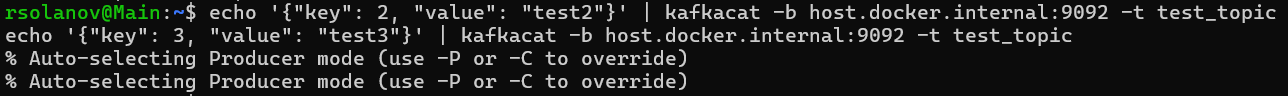
**select** \* **from** mergetree\_table



* Отправка в Kafka новых сообщений

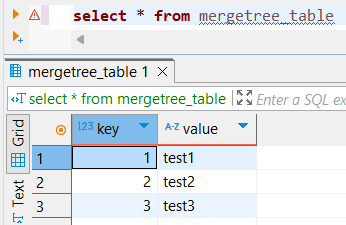
echo '{"key": 2, "value": "test2"}' | kafkacat -b host.docker.internal:9092 -t test\_topic

echo '{"key": 3, "value": "test3"}' | kafkacat -b host.docker.internal:9092 -t test\_topic



* Проверка получения сообщений в ClickHouse

**select** \* **from** mergetree\_table



Записать данные в кафку с помощью ClickHouse Kafka Engine

* Создание таблицы для записи в Kafka

**CREATE** **TABLE** kafka\_writer

(

id **UInt32**,

name **String**,

timestamp **DateTime**

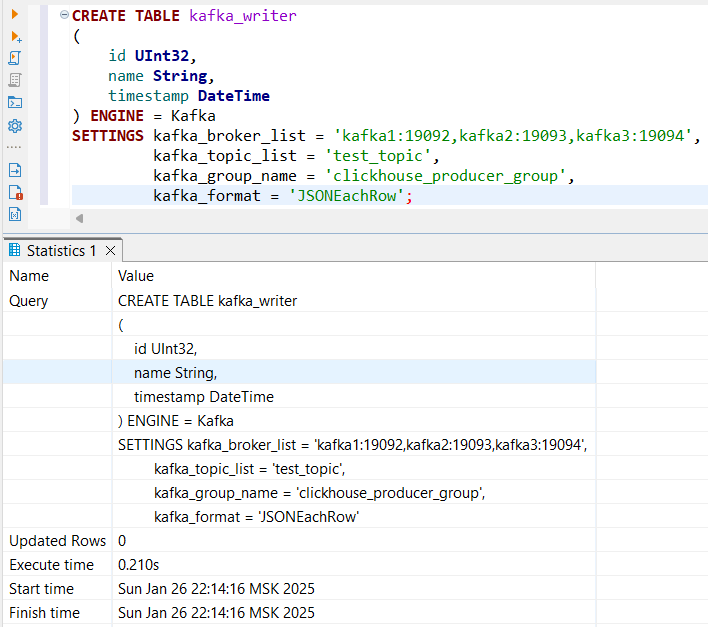
) **ENGINE** = Kafka

**SETTINGS** kafka\_broker\_list = 'kafka1:19092,kafka2:19093,kafka3:19094',

kafka\_topic\_list = 'test\_topic',

kafka\_group\_name = 'clickhouse\_producer\_group',

kafka\_format = 'JSONEachRow';



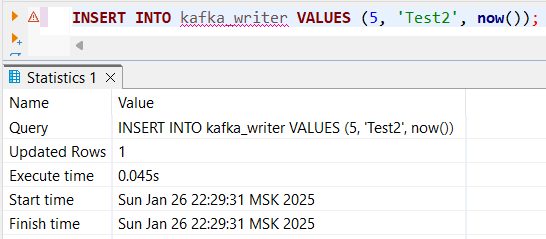
* Пересоздание топика

kafka-topics --bootstrap-server kafka1:19092 --delete --topic test\_topic

kafka-topics --bootstrap-server kafka1:19092 --create --topic test\_topic --partitions 3 --replication-factor 3

* Вставка данных в таблицу

**INSERT** **INTO** kafka\_writer **VALUES** (5, 'Test2', **now**());



* Чтение данных из топика

kafka-console-consumer --bootstrap-server kafka1:19092 --topic test\_topic --from-beginning



Построить тот же пайплайн обработки данных НЕ через Kafka Engine

* Пересоздаем топик, очищаем таблицу «mergetree\_table» и добавляем сообщение

echo '{"key": 1, "value": "test1"}' | kafkacat -b host.docker.internal:9092 -t test\_topic

* Запускаем скрипт Python

from confluent\_kafka import Consumer  
import clickhouse\_connect  
import json  
  
  
def main():  
 consumer = Consumer({  
 'bootstrap.servers': 'host.docker.internal:9092',  
 'group.id': 'python\_consumer\_group',  
 'auto.offset.reset': 'earliest'  
 })  
 consumer.subscribe(['test\_topic'])  
  
 clickhouse\_client = clickhouse\_connect.get\_client(host='host.docker.internal', port=8123, username='default', password='default')  
  
 while True:  
 msg = consumer.poll(1.0)  
 if msg is None:  
 continue  
 if msg.error():  
 print(f"Ошибка потребителя: {msg.error()}")  
 continue  
  
 data = json.loads(msg.value().decode('utf-8'))  
 print(json.dumps(data, indent=2))  
 clickhouse\_client.command(f"INSERT INTO mergetree\_table FORMAT JSONEachRow {json.dumps(data)}")  
  
 consumer.close()  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

* Проверка данных таблицы

**select** \* **from** mergetree\_table

