1. Init confluent folders

docker run -v $PWD/confluent-hub-components:/share/confluent-hub-components confluentinc/ksqldb-server:0.8.0 confluent-hub install --no-prompt confluentinc/kafka-connect-datagen:0.4.0

docker run -v $PWD/confluent-hub-components:/share/confluent-hub-components confluentinc/ksqldb-server:0.8.0 confluent-hub install --no-prompt confluentinc/kafka-connect-elasticsearch:10.0.2

1. Updated **.env**:

PWD=D:\projects\Mentoring\data\_mentoring\hw\_04\_kafka\docker\_kafka\_tutorial

1. Launch tutorial

docker-compose up -d

1. Check docker images:

docker-compose ps

Изображение выглядит как текст, черный, электроника, снимок экрана

Автоматически созданное описание

1. Launch the ksqlDB CLI

docker-compose exec ksqldb-cli ksql <http://ksqldb-server:8088>

Изображение выглядит как текст

Автоматически созданное описание

1. Ensure the ksqlDB server is ready to receive requests by running the following until it succeeds:

show topics; Изображение выглядит как текст

Автоматически созданное описание

1. Run scripts

RUN SCRIPT '/scripts/create-connectors.sql'

Изображение выглядит как текст

Автоматически созданное описание

1. Run

print clickstream limit 3;

Изображение выглядит как текст, на открытом воздухе

Автоматически созданное описание

1. Run

print clickstream\_codes limit 3;

Изображение выглядит как текст

Автоматически созданное описание

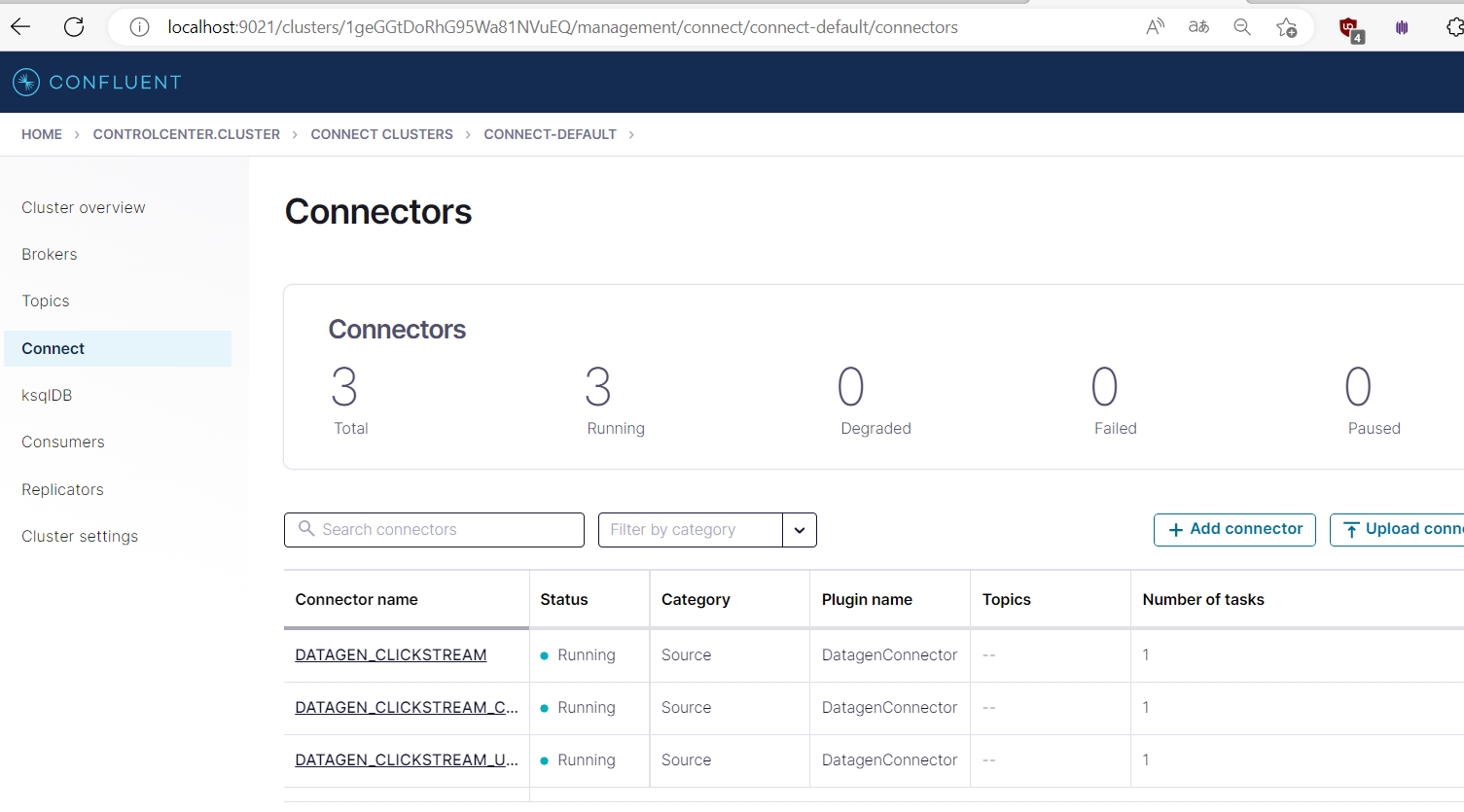
1. Run

print clickstream\_users limit 3;

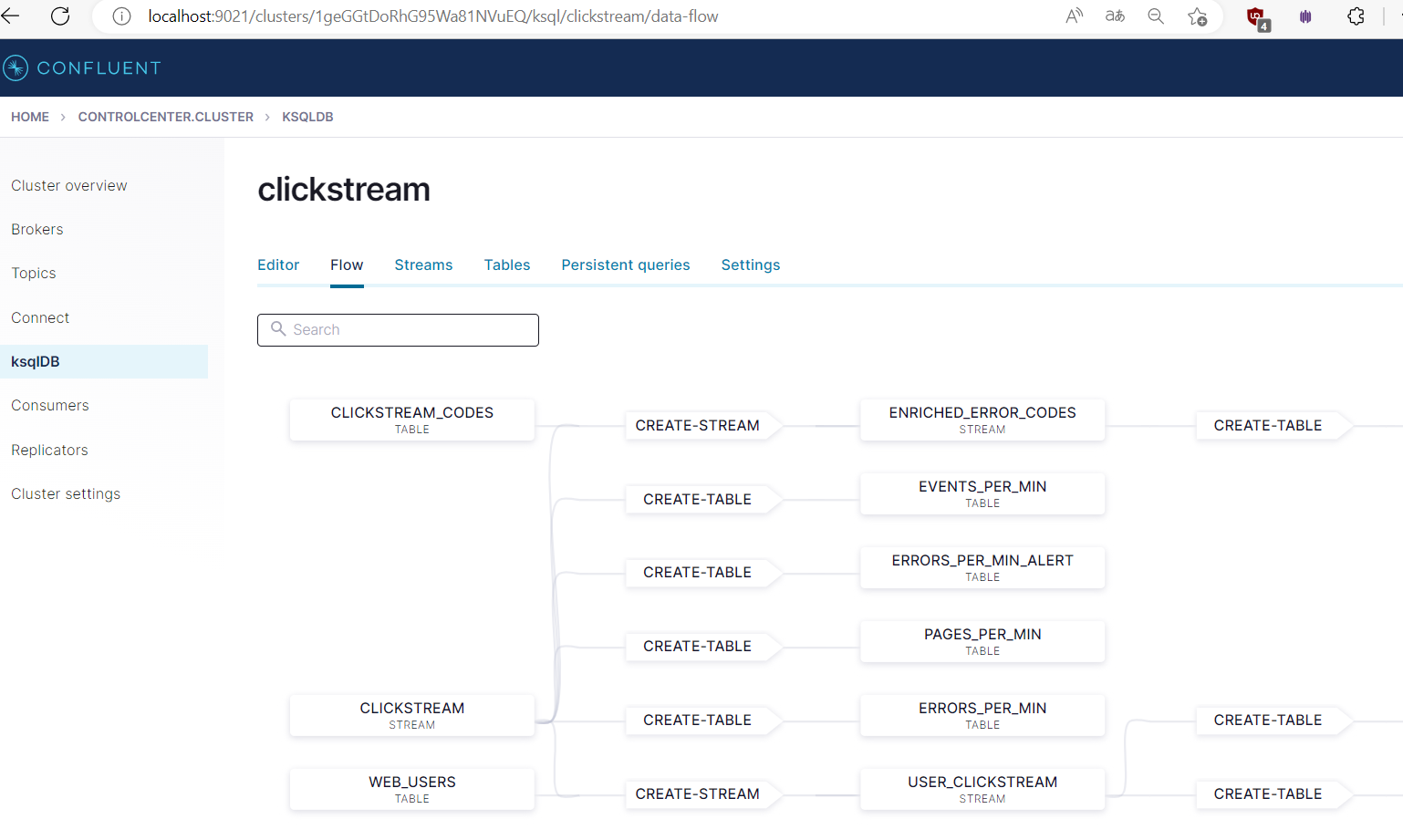
Изображение выглядит как текст

Автоматически созданное описание

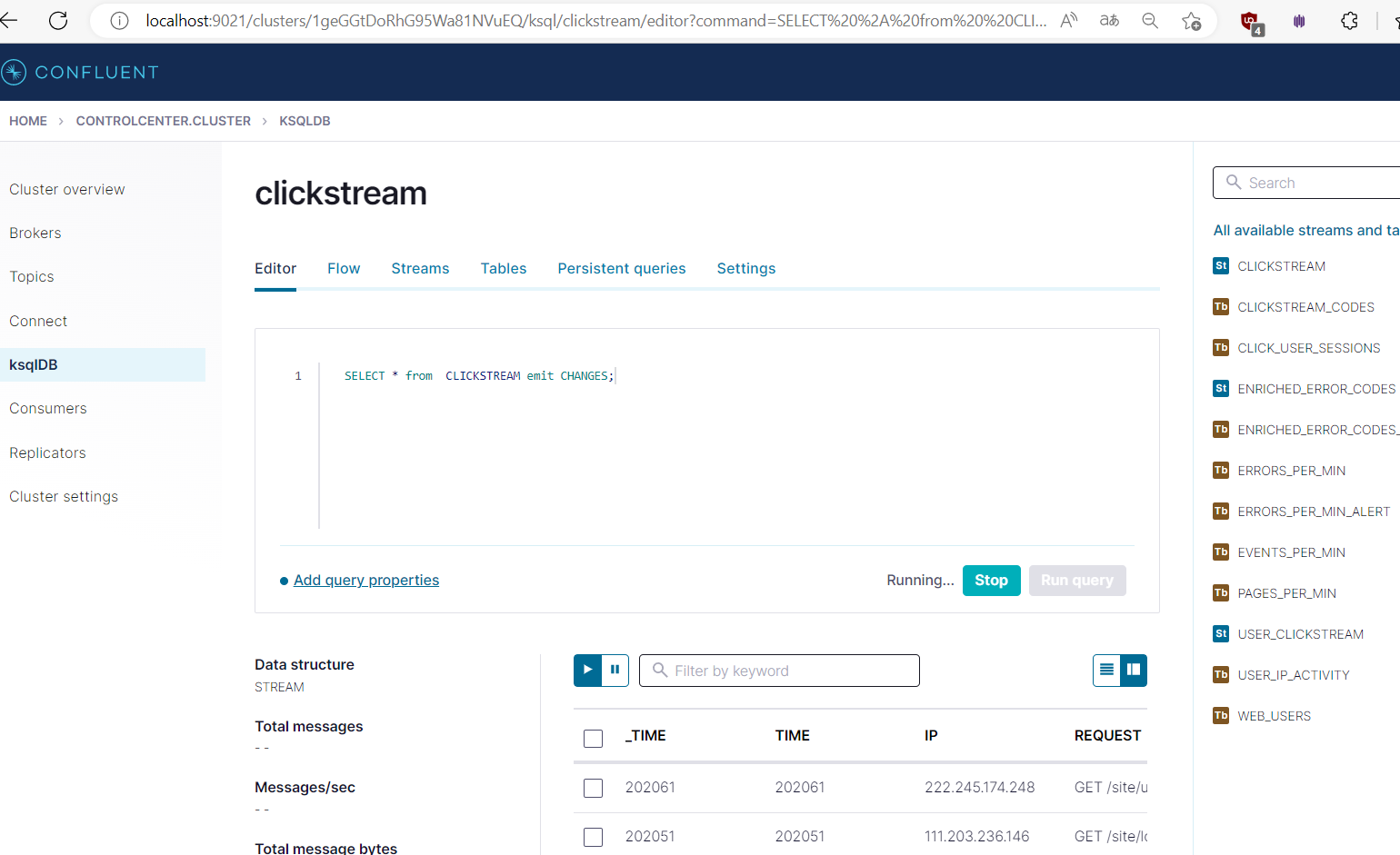
1. Go to |c3| UI at <http://localhost:9021>



1. RUN SCRIPT '/scripts/statements.sql';
2. Exit out of the **ksqldb-cli** with CTRL+D
3. Verify the data
   1. Go to |c3| UI at http://localhost:9021, and view the ksqlDB view ``Flow``



* 1. Run query “SELECT \* from CLICKSTREAM emit CHANGES;”



1. Send the ksqlDB tables to Elasticsearch and Grafana.
   1. docker-compose exec elasticsearch bash -c '/scripts/elastic-dynamic-template.sh'

Изображение выглядит как текст

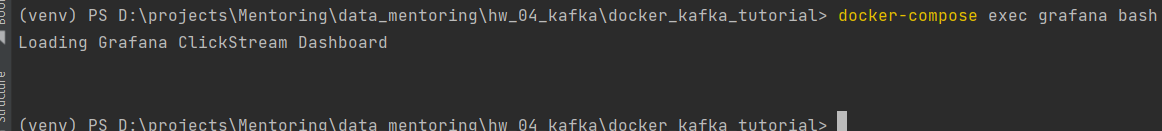
Автоматически созданное описание

* 1. docker-compose exec ksqldb-server bash -c '/scripts/ksql-tables-to-grafana.sh'

Изображение выглядит как текст

Автоматически созданное описание

* 1. docker-compose exec grafana bash -c '/scripts/clickstream-analysis-dashboard.sh'

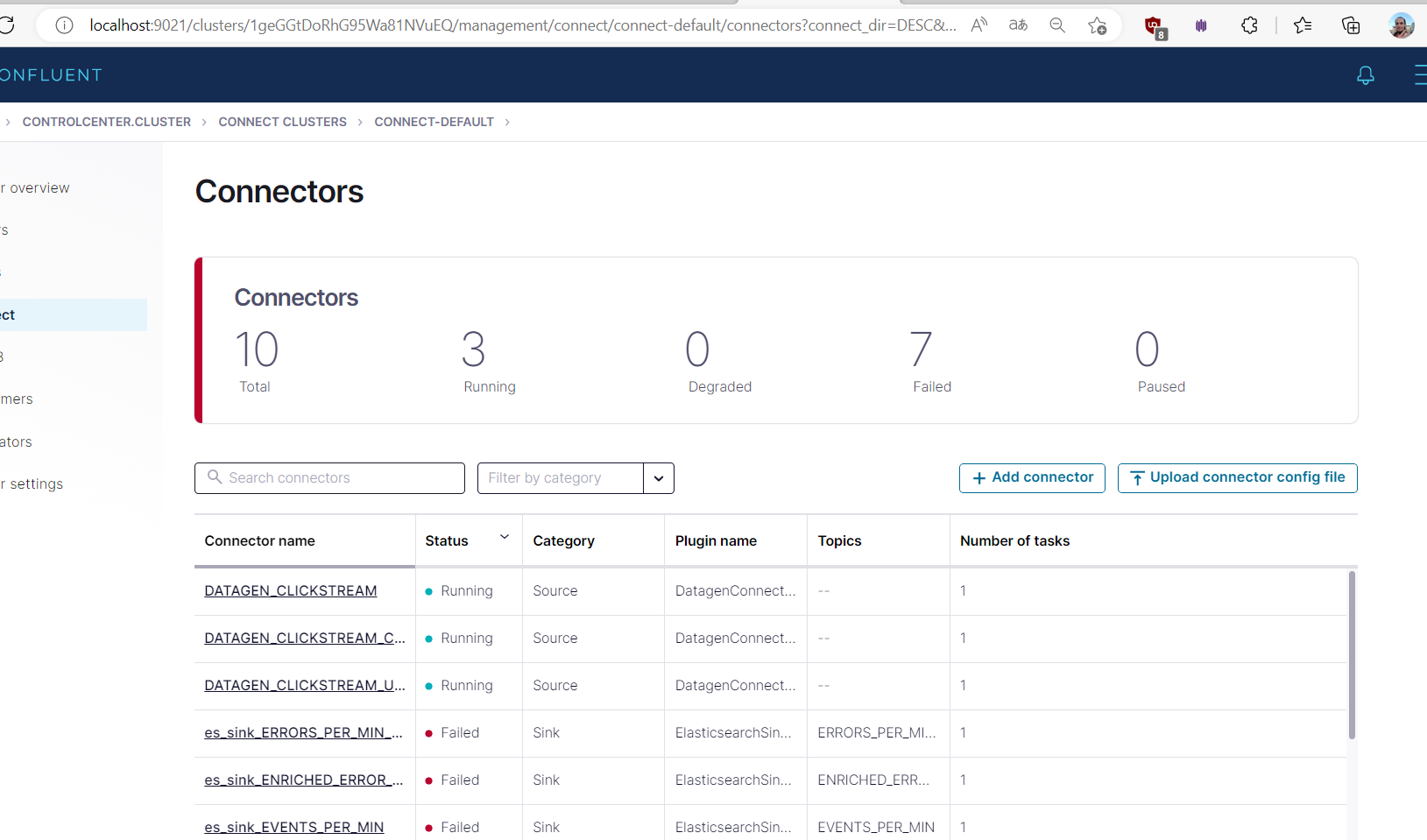


* 1. Navigate to the Grafana dashboard at <http://localhost:3000> user/user and navigate to **Clickstream Analysis Dashboard**

Изображение выглядит как текст, монитор, экран, телевидение

Автоматически созданное описание

* 1. In the |c3| UI at <http://localhost:9021>, again view the running connectors. The three kafka-connect-datagen source connectors were created with the ksqlDB CLI, and the seven Elasticsearch sink connectors were created with the ksqlDB REST API.



1. ./sessionize-data.sh

Изображение выглядит как текст

Автоматически созданное описание

1. Stop the container(s) using the following command:

docker-compose down

1. Delete all containers using the following command:

docker rm -f $(docker ps -a -q)

1. Delete all volumes using the following command:

docker volume rm $(docker volume ls -q)

* index.rst
* <https://www.youtube.com/watch?v=DgqWFJ4ZOmg> 17:30