

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

"Российский технологический университет"

МИРЭА

Институт «Кибербезопасности и цифровых технологий»

Кафедра КБ-3 «Разработка программных решений и системного программирования»

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1**

**по дисциплине**

**«Проектирование Архитектуры программного обеспечения»**

**Выполнил**

студент группы БСБО-02-20 Рулев Д.О.

*ФИО*

**Принял**

к.т.н., доцент Волович М. Е.

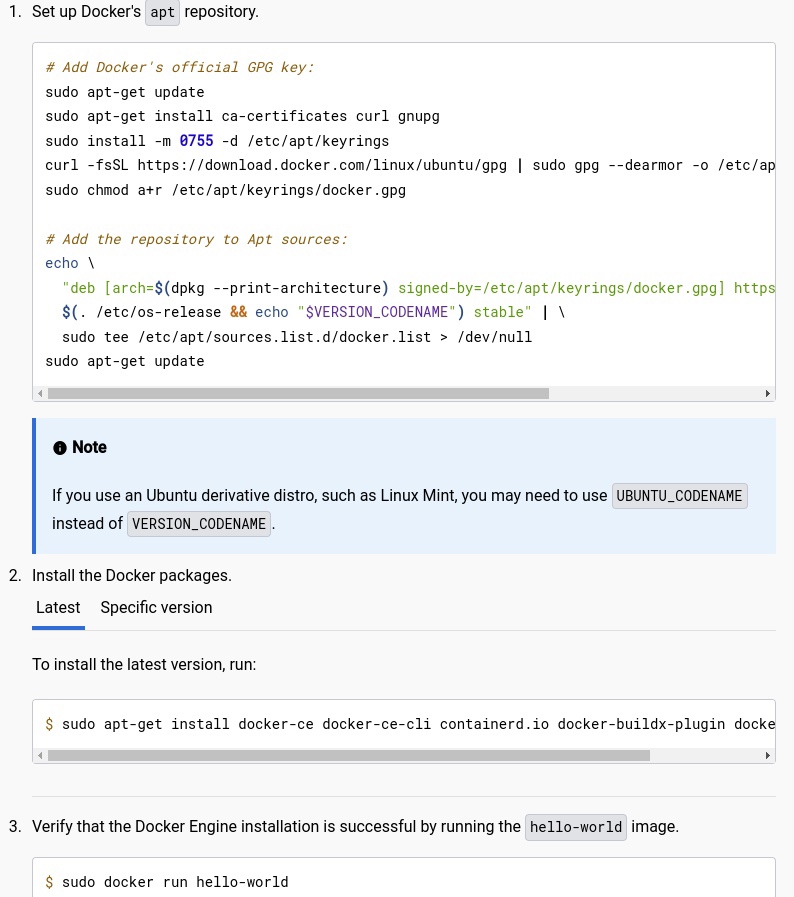
**Работа выполнена**

« » февраля 2023 г.

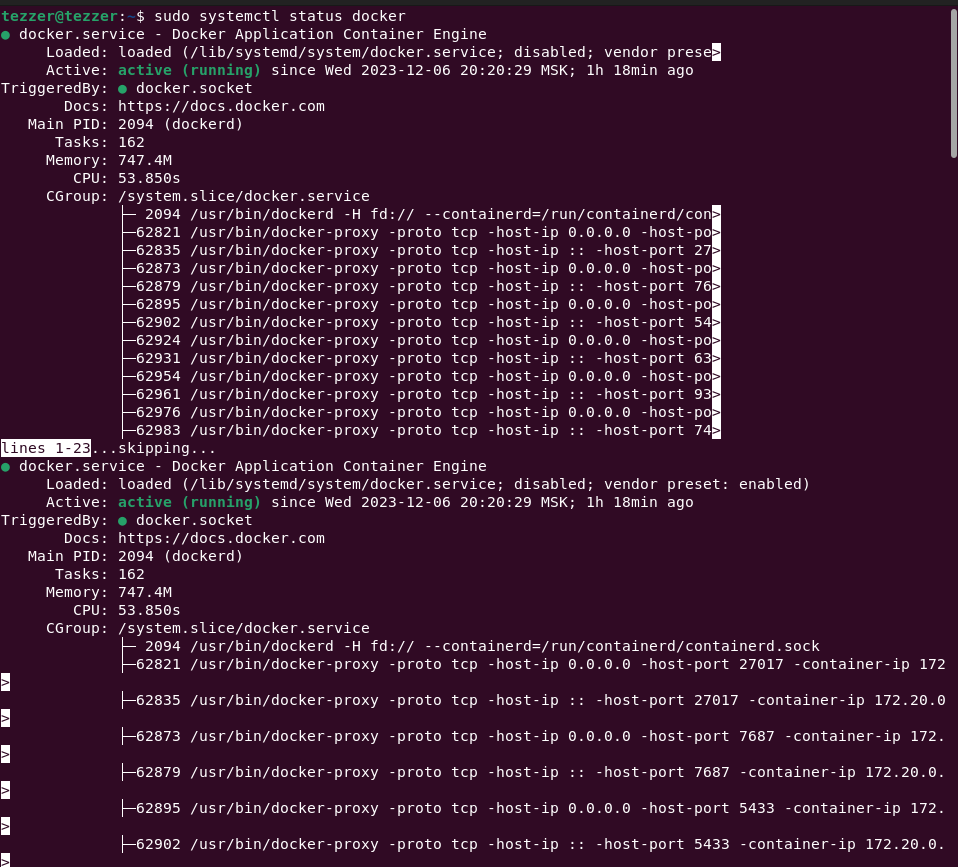
Москва 2023

1. установка docker и docker-compose

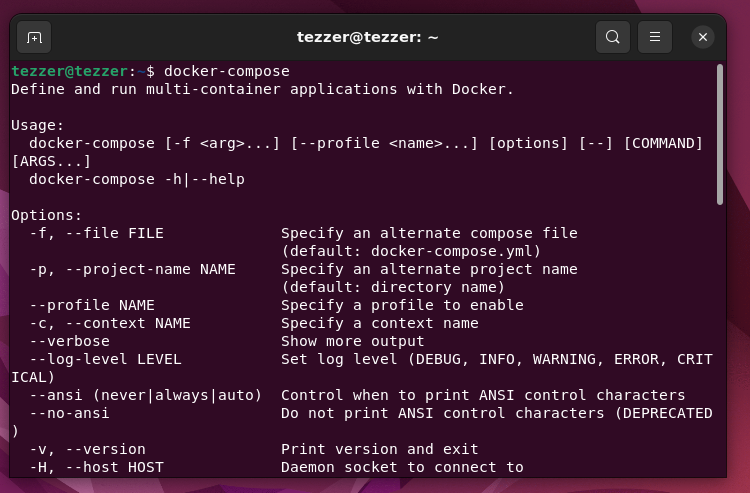
Для установки на Ubuntu ОС была использована следующая команда:



После этого необходимо включить службу докера и проверить её статус:



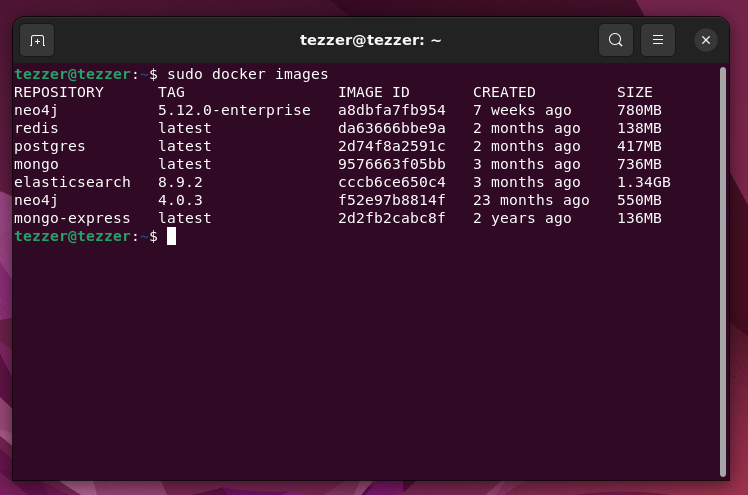
Для проверки docker-compose была введена следующая команда:



На этом установка завершена.

1. установка СУБД

С официального сайта были загружены пять СУБД: postgresql, mongodb, redis, neo4j, elasticsearch. Ниже представлен список установленных пакетов:



1. проверка работоспособности СУБД

Был создан файл с настройками docker-compose. Это позволило упростить конфигурирование системы: автоматический запуск контейнера, папки для хранения данных. Ниже представлено содержание данного файла.

version: '3.8'

services:

  postgres:

    image: postgres

    container\_name: postgres-server

    restart: always

    environment:

      - POSTGRES\_USER=postgres

      - POSTGRES\_PASSWORD=12345

    ports:

      - '5433:5432'

    volumes:

      - ./postgres/data:/var/lib/postgresql/data

  mongo:

    image: mongo

    container\_name: mongo-server

    restart: always

    volumes:

      - ./mongo/data:/data/db

    environment:

      MONGO\_INITDB\_ROOT\_USERNAME: admin

      MONGO\_INITDB\_ROOT\_PASSWORD: 12345

    ports:

      - 27017:27017

  mongo-express:

    image: mongo-express

    container\_name: mongo-express

    restart: always

    ports:

      - 8081:8081

    environment:

      ME\_CONFIG\_MONGODB\_ADMINUSERNAME: admin

      ME\_CONFIG\_MONGODB\_ADMINPASSWORD: 12345

      ME\_CONFIG\_MONGODB\_URL: mongodb://admin:12345@mongo:27017/

  neo4j:

    image: neo4j:5.12.0-enterprise

    hostname: neo4j

    container\_name: neo4j-server

    restart: always

    ports:

      - "7474:7474"

      - "7687:7687"

    volumes:

      - ./neo4j/plugins:/plugins

      - ./neo4j/data:/var/lib/neo4j/data

    environment:

      NEO4J\_AUTH: neo4j/root12345

      NEO4J\_dbms\_logs\_debug\_level: DEBUG

      NEO4J\_ACCEPT\_LICENSE\_AGREEMENT: "yes"

  redis:

    image: redis:latest

    container\_name: redis-server

    restart: always

    ports:

      - "6379:6379"

    volumes:

      - ./redis/dаta:/root/redis

      - ./redis/redis.conf:/usr/local/etc/redis/redis.conf

    environment:

      - REDIS\_PASSWORD=12345

      - REDIS\_PORT=6379

      - REDIS\_DATABASES=16

  elasticsearch:

    image: elasticsearch:8.9.2

    container\_name: elastic-server

    restart: always

    volumes:

      - ./elasticsearch/data:/usr/share/elasticsearch/data

    environment:

      ES\_JAVA\_OPTS: "-Xmx512m -Xms512m"

      ELASTIC\_USERNAME: "elasticB"

      ELASTIC\_PASSWORD: "12345"

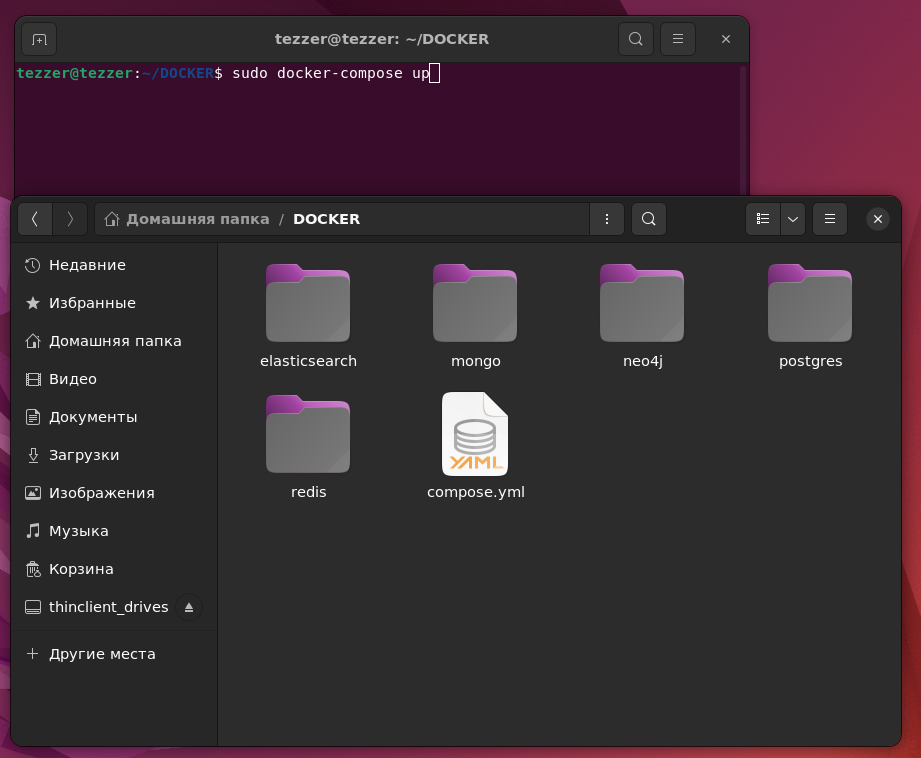
      discovery.type: single-node

    ports:

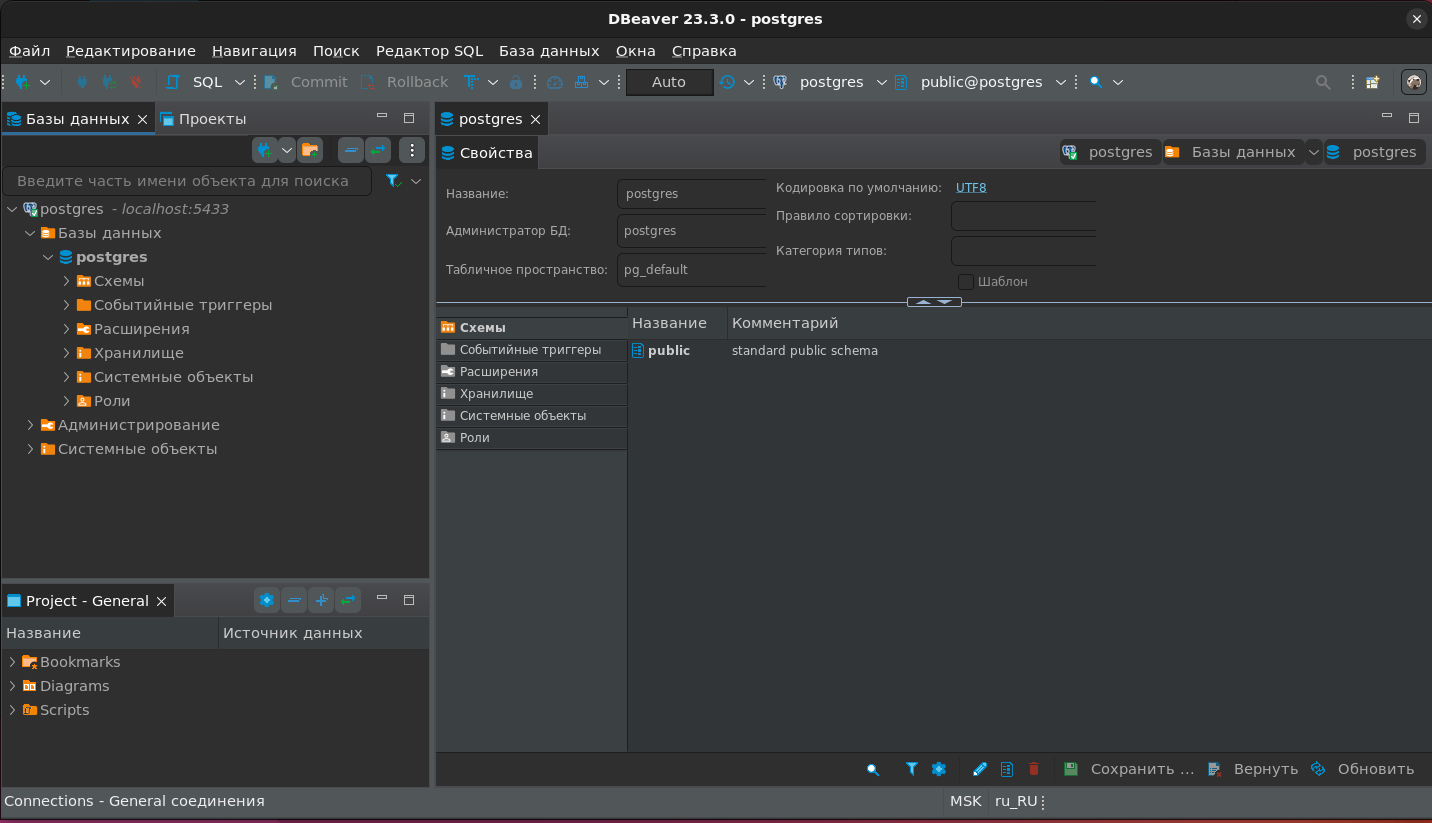
      - "9200:9200"

      - "9300:9300"

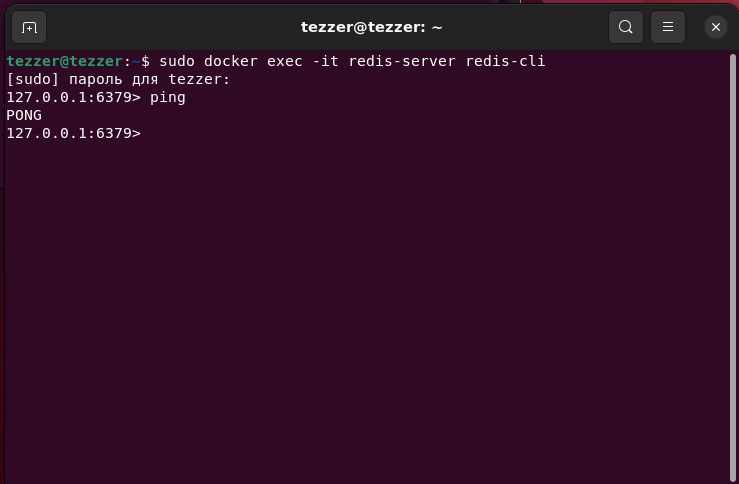
Для запуска файла настроек была применена следующая команда:



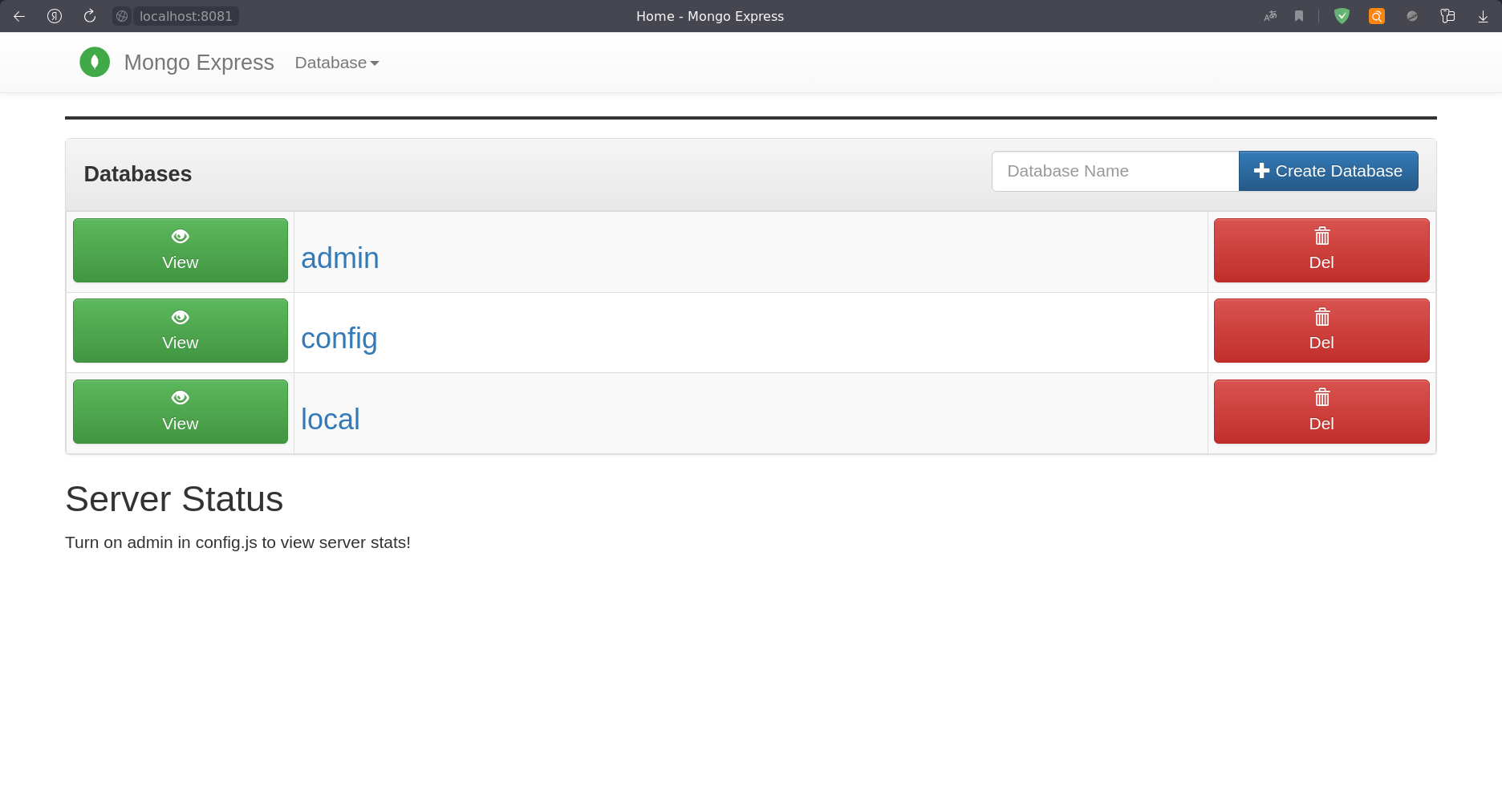
Для проверки состояния СУБД было решено подключиться к каждому контейнеру и выполнить команду.

Для postgres:

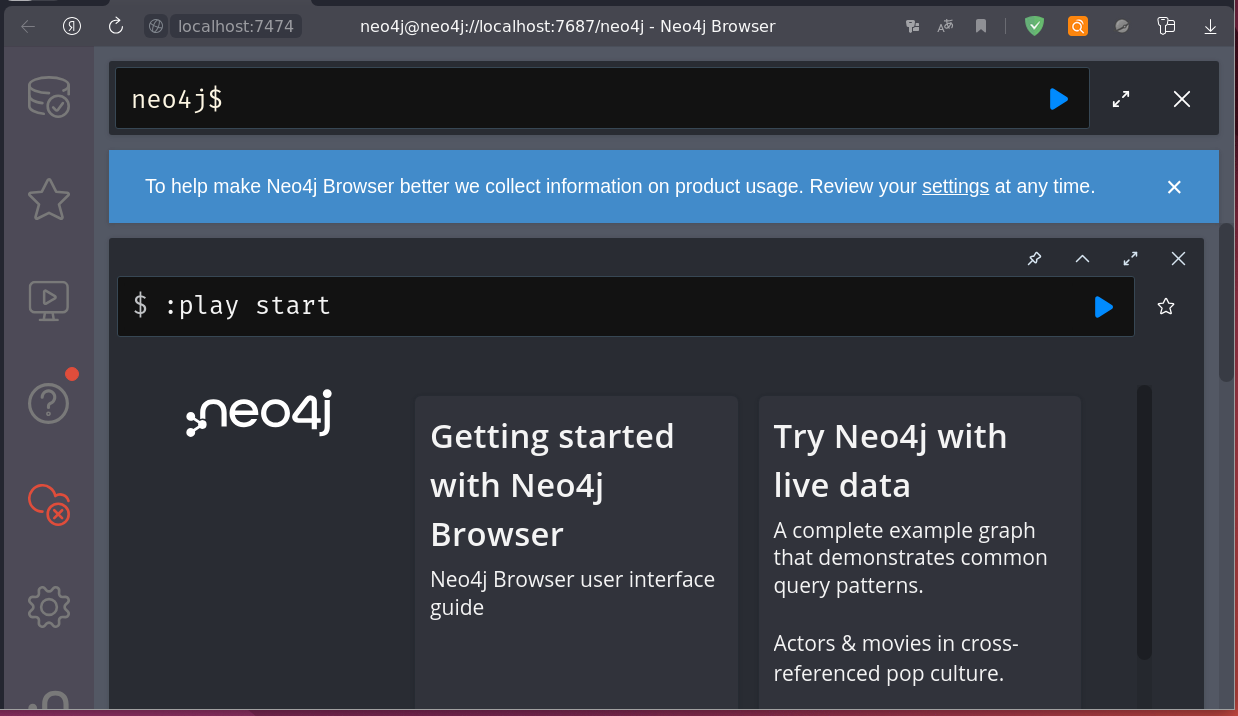
Для redis:



Для mongodb:



Для neo4j:



Для ElasticSearch:

