Департамент образования и науки города Москвы

Государственное автономное образовательное учреждение высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики, управления и технологий

ДИСЦИПЛИНА:

Интеграция и развертывание программного обеспечения с помощью контейнеров

**Лабораторная работа 1.1**

**Установка и настройка Docker. Работа с контейнерами в Docker**

Выполнила: Сергеева А. И., группа: АДЭУ-211

Преподаватель: Босенко Т.М.

Москва

2025

**Цель работы:** освоить процесс установки и настройки Docker, научиться работать с контейнерами и образами Docker.

**Ход работы:**

Для удобства работы сначала нужно подключиться через удаленный рабочий стол к виртуальной машине. Далее была выполнена установка Docker: обновлён список доступных пакетов с помощью sudo apt update, установлены пакеты для работы с HTTPS-репозиториями (apt-transport-https), набор сертификатов для проверки подлинности (ca-certificates), утилита для скачивания файлов (curl), для работы с GPG-файлами (gnupg), пакет для получения информации о дистрибутиве (lsb-release), скачен ключ GPG, установлены основная утилита Docker и пакет для работы с контейнерами, что продемонстрировано на рисунках 1-5.



Рисунок 1 – Обновление списка доступных пакетов

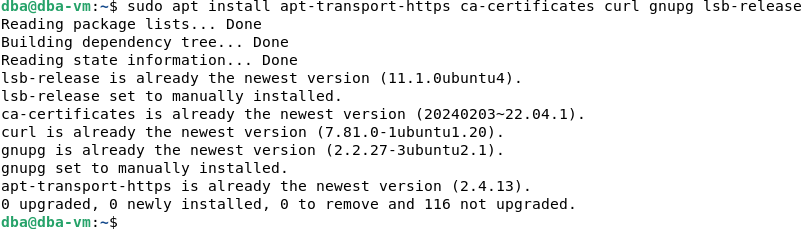


Рисунок 2 – Установка необходимых пакетов

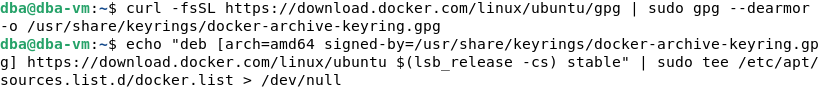


Рисунок 3 – скачивание и добавление ключа GPG и репозитория Docker для установки Docker



Рисунок 4 – Повторное обновление списка доступных пакетов

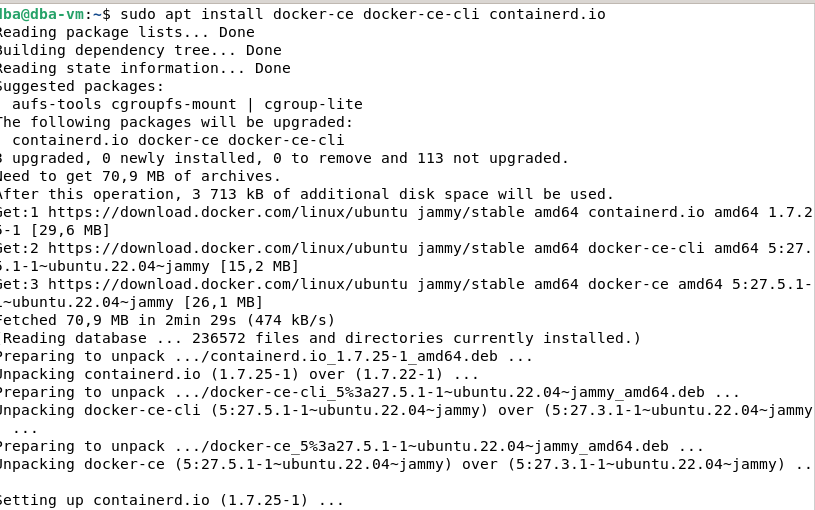


Рисунок 5 – Установка основных пакетов Docker (Docker Engine, командная строка Docker и пакет для управления контейнерами)

После завершения установки пользователь dba, в котором выполнен вход в систему был добавлен в группу docker и затем командой «newgrp docker» активируется членство в группе, которое было добавлено, данные изменения позволят выполнять команды без sudo. Данные операции показаны на рисунке 6.



Рисунок 6 – Добавление текущего пользователя в группу docker

Для проверки установки Docker была выведена его версия и запущен Docker-контейнер на основе образа hello-world, который выводит приветственное сообщение «Hello from Docker!», что можно проследить на рисунке 7.

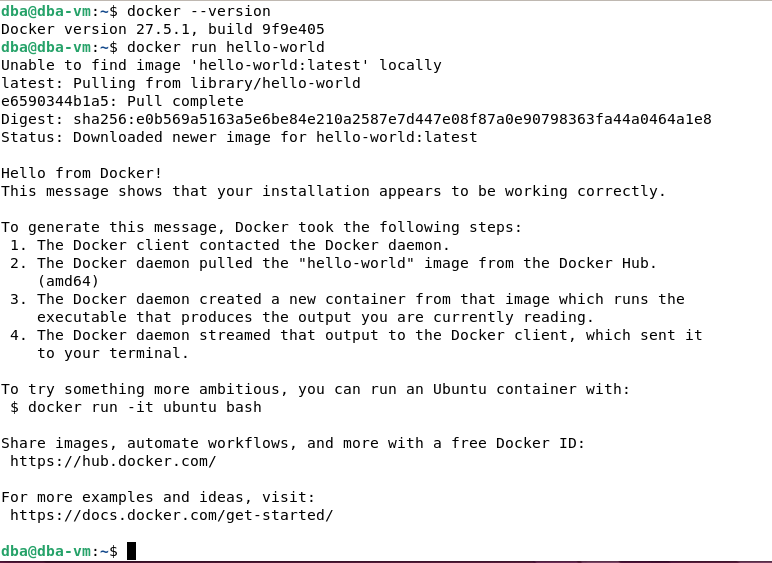


Рисунок 7 – Проверка версии Docker и запуск контейнера на основе образа hello-world

Для ознакомления с основными командами Docker CLI с помощью docker images был просмотрен список локальных образов Docker, docker ps список запущенных контейнеров, на данный момент ничего не запущено и docker ps –a для списка всех контейнеров, в том числе и осатновленных, где зафиксирован и недавно запущенный на основе образа hello-world. Рисунок 8 демонстрирует все перечисленные команды.

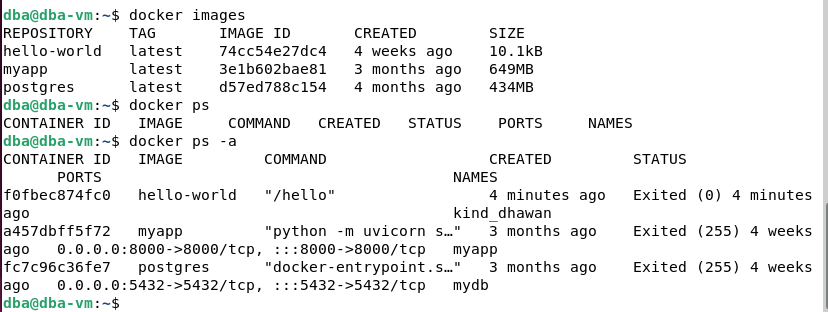


Рисунок 8 – Выполнение команд на просмотр локальных образов и контейнеров

**Индивидуальное задание**

**Вариант 12.** Загрузить образ grafana, запустить контейнер, настроить маршрутизацию портов и добавить новый источник данных.

Grafana — программная система визуализации данных, ориентированная на данные систем ИТ-мониторинга с открытым исходным кодом.

Для начала необходимо загрузить образ grafana последней версии. Сделать это можно с помощью команды «docker pull», показанной на рисунке 9.

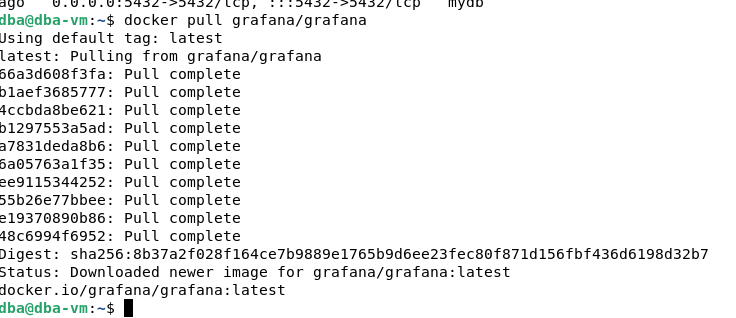


Рисунок 9 – Загрузка образа grafana

Далее образ был запущен в контейнере Docker. Команда запускает контейнер Grafana в автономном режиме и сопоставляет порт 3000 контейнера с портом 3000 хост-машины. В рамках порта хоста машины можно было выбрать любой свободный, проверка, свободен ли порт 3000 была выполнена. С результатами можно ознакомиться на рисунке 10.

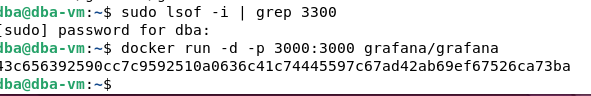


Рисунок 10 – Запуск контейнера на основе образа Grafana

Далее была проведена авторизация по ссылке <http://localhost:3000> с логином и паролем admin на рисунке 11.

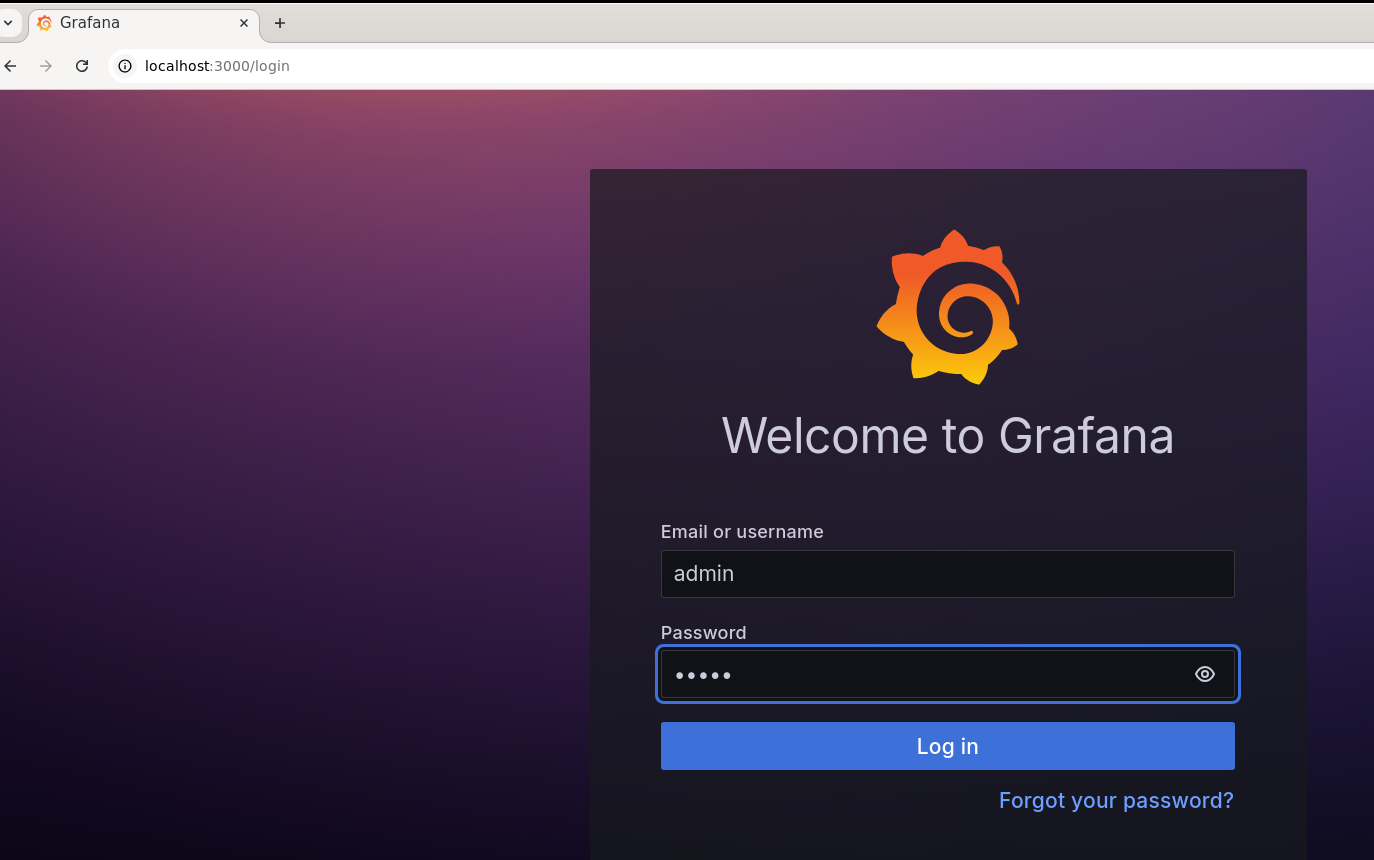


Рисунок 11 – Вход в Grafana через браузер

Далее был настроен новый пароль для входа, как видно на рисунке 12.

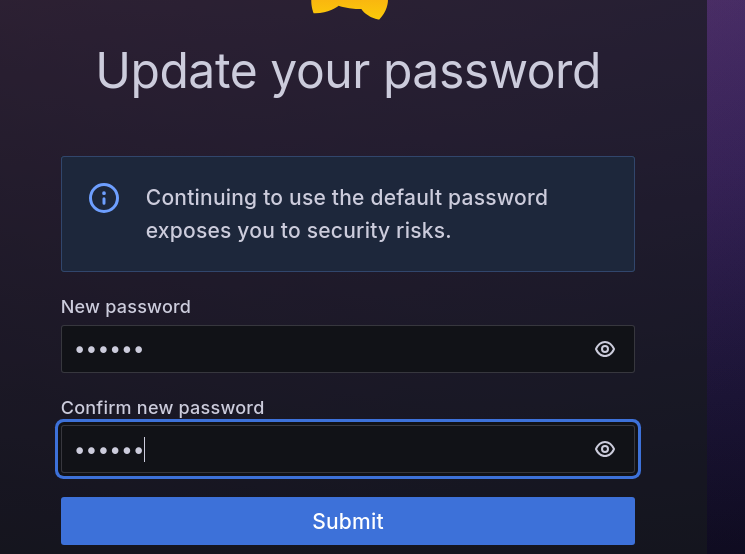


Рисунок 12 – Изменение пароля в Grafana

В рамках нового источника данных был выбран CSV-файл о супергероях, скачанный с пособия по работе с Yandex DataLens, а также установлен нужный плагин, данные шаги показаны на рисунках 12-15.

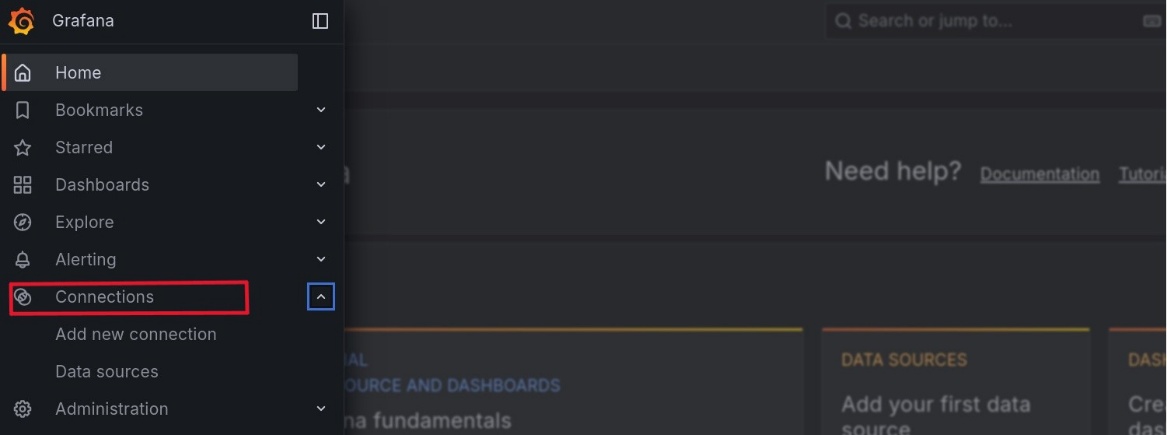


Рисунок 12 – Выбор в меню пункта о добавлении нового источника данных

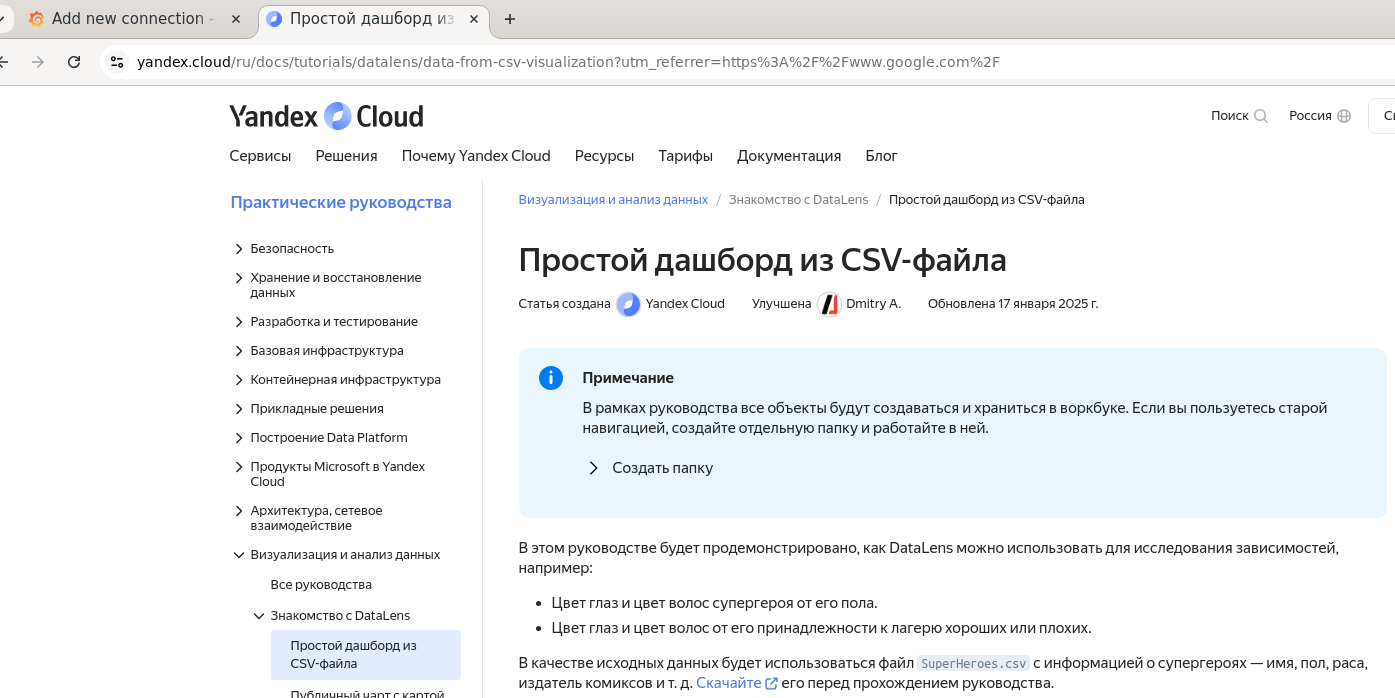


Рисунок 13 – Подготовка файла для подключения к новому источнику данных

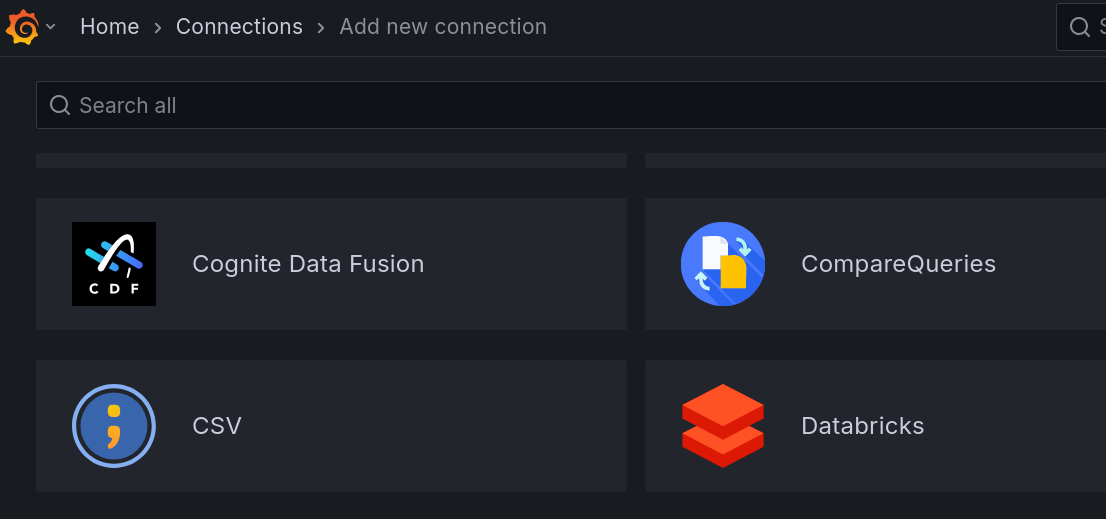


Рисунок 14 – Выбор CSV файла как типа для создания подключения

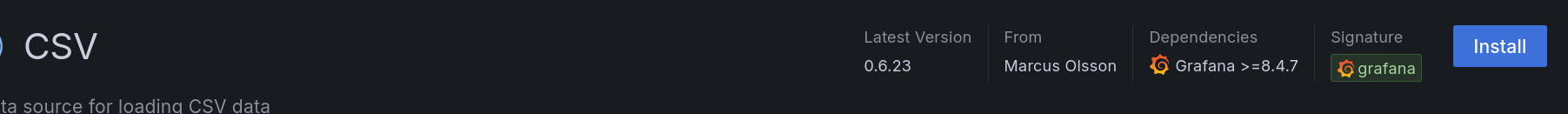


Рисунок 15 – Установка плагина для работы с CSV файлами

Файл был успешно загружен, появилось уведомление о возможности визуализации данных по этому файлу, было задано имя новому подключению «SuperHeroessssss», как показано на рисунках 16-17.

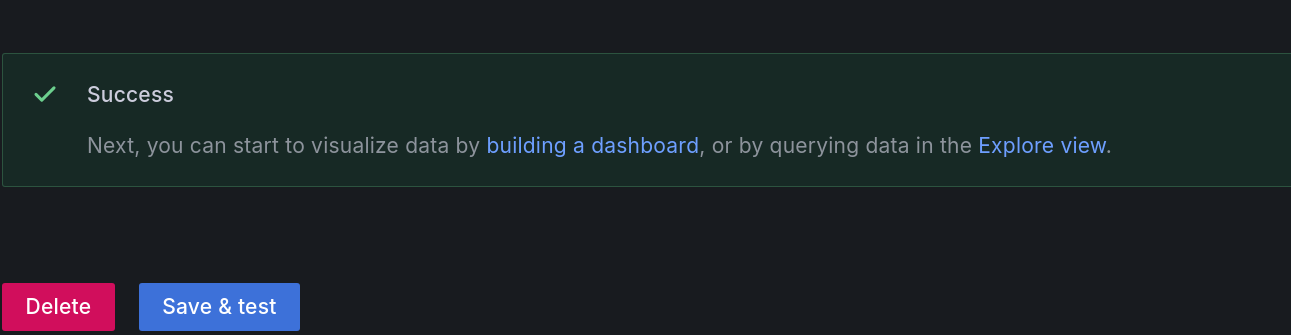


Рисунок 16 – Успешная загрузка файла

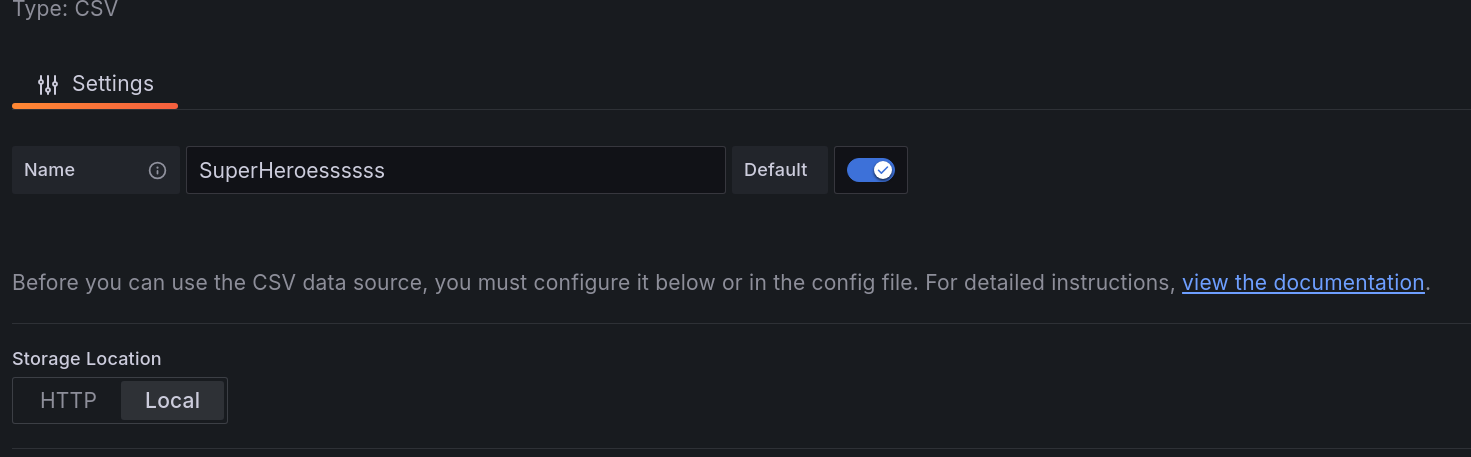


Рисунок 17 – Выбор локальной загрузки файла и обозначение наименования подключения

Чтобы остановить работу grafana, необходимо узнать имя контейнера, для этого была выполнена команда на просмотр всех запущенных контейнеров, его имя «loving\_fermat», что можно увидеть на рисунке 18 и командой с рисунка 19 grafana была остановлена.

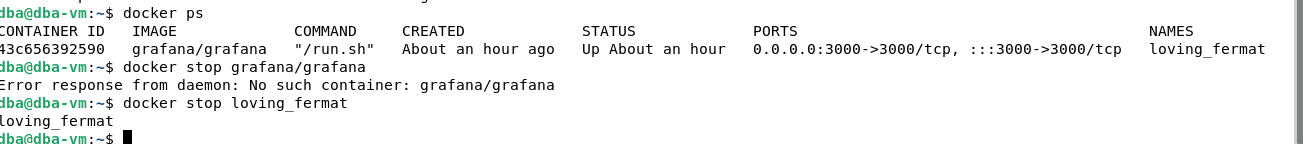


Рисунок 18 – Просмотр всех запущенных контейнеров

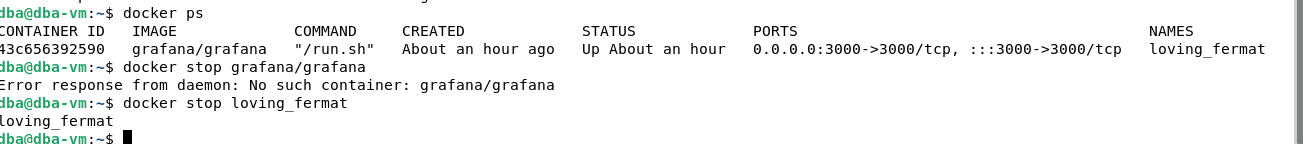


Рисунок 19 – Остановка работы контейнера

**Общий вывод**:

По результатам работы был освоен процесс загрузки и запуска Docker-образов, настроек портов для обеспечения внешнего доступа к веб-интерфейсу Grafana, а также добавления и конфигурации источников данных. В рамках ознакомления были выполнены основные команды при работе с Docker CLI. Хотя CSV-файл — простой и не самый эффективный источник для Grafana, его использование помогло изучить принципы настройки плагинов для Grafana, а также попробовать создать новый источник данных, с которым можно в дальнейшем работать. Развёртывание Grafana оказалось значительно быстрым благодаря возможностям Docker, что позволило запустить и протестировать Grafana без дополнительных настроек, ведь Docker управляет всеми зависимостями Grafana внутри контейнера. Нет необходимости беспокоиться о правильной установке и конфигурации. Изучив меню Grafana, было отмечено огромное количество плагинов, позволяющее работать с более расширенным списком типов данных и визуализаций. Также данное решение с открытым исходным кодом и имеет активное сообщество, делая его довольно популярным решением для визуализации. Цель работы, связанная с ознакомлением работы с Docker, образами и контейнерами, была выполнена.

**Ответы на контрольные вопросы:**

1. Что такое Docker и для чего он используется?

Docker — это платформа c открытым исходным кодом для автоматизации развертывания, масштабирования и управления приложениями в контейнерах. Контейнеры позволяют упаковывать приложение и все его зависимости в единый, легковесный пакет, который можно запускать на любой системе, поддерживающей Docker. Это обеспечивает согласованность среды разработки и продакшена, а также упрощает развертывание и управление приложениями.

1. Какие преимущества дает использование контейнеров Docker по сравнению с виртуальными машинами?

* Эффективность:

Контейнеры требуют меньше накладных расходов, чем ВМ, поскольку они совместно используют ОС хоста. Это приводит к более быстрому запуску и лучшему использованию ресурсов.

* Масштабируемость:

Контейнеры можно легко масштабировать вверх или вниз в зависимости от спроса. Вы можете быстро развернуть новые экземпляры контейнеров для обработки возросшей нагрузки.

* Согласованность в разных средах:

Контейнеры инкапсулируют все зависимости, гарантируя, что приложения будут работать одинаково при разработке, тестировании и производстве. Это устраняет проблему «работает на моей машине».

* Упрощенное развертывание:

С контейнерами развертывание приложений становится простым. Вы можете упаковать свое приложение и его зависимости в один образ контейнера, который можно легко распространять и развертывать.

* Изоляция:

Контейнеры обеспечивают уровень изоляции между приложениями.

* Быстрая разработка и тестирование:

Разработчики могут быстро создавать, тестировать и итерировать приложения в изолированных средах, не беспокоясь о конфликтах с другими приложениями.

1. Что такое образ Docker и как он связан с контейнерами?

Образ Docker — это неизменяемый файл для развертывания контейнера. Он содержит все необходимые инструкции по созданию среды для приложения, включая систему, библиотеки и зависимости. Контейнер — это запущенный образ, который работает в изолированной среде. Существуют разные версии образов, и в процессе создания контейнера создается новая инстанция образа, которая может изменяться независимо от оригинала.

1. Какие основные команды Docker CLI вы узнали в ходе выполнения лабораторной работы?

docker images - список локальных образов Docker;

docker ps - список запущенных контейнеров;

docker ps –a - список всех контейнеров, включая остановленные;

docker run — для запуска контейнера из образа;

docker stop — для остановки запущенного контейнера;

docker rm — для удаления остановленного контейнера.

1. Как можно настроить маршрутизацию портов при запуске контейнера Docker?

docker run -p <порт хоста>:<порт контейнера> <наименование образа>