Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

специальность 09.02.07 «Информационные системы и программирование»

Квалификация: Программист

ПРАКТИЧЕСКАЯ РАБОТА

По теме «Bitrix»

ПО МДК 04.02 «Обеспечение качества функционирования компьютерных систем»

|  |  |
| --- | --- |
| Выполнил студент  группы П50-4-21  Игошев Р. В. | Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_ Р.Д. Степаньков  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2025 года |

Москва 2025

Цель работы: создать файл docker-compose.yml для запуска CMS Bitrix с использованием необходимых сервисов. Конфигурация должна быть адаптирована для работы в режиме разработки или тестирования.

Начать следует с установки дистрибутива Bitrix, который и следует в рамках практической работы развернуть локально. Можно выбрать «Старт» или «Стандарт», в рамках задания принципиальных различий нет.

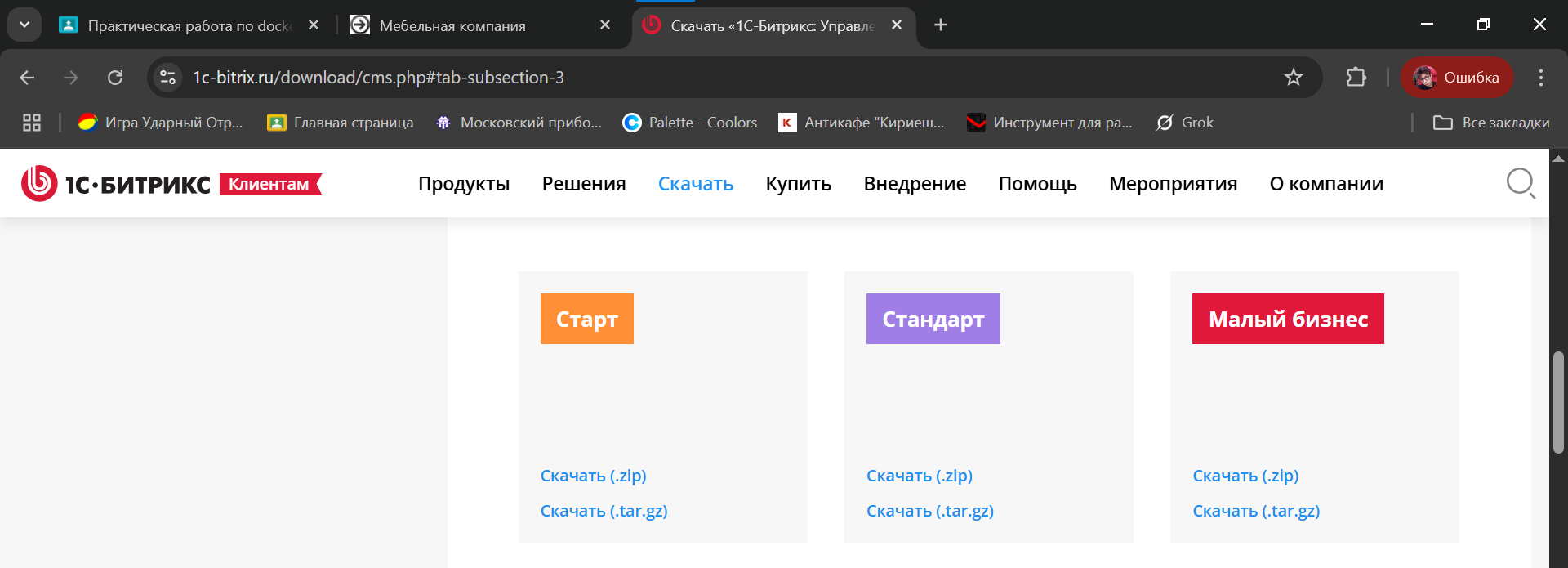


Рисунок 1 – Скачивание дистрибутива bitrix

Далее, следует перейти к самой настройке развертывания. Иерархия готового проекта изображена на рисунке 2 и содержит в себе следующие папки:

* bitrix – содержит в себе дистрибутив Bitrix для дальнейшей его установки.
* mysql – папка, содержащая в себе базу данных приложения и файл инициализации БД.
* nginx – эта папка содержит в себе конфигурацию сервера nginx и папки с логами доступа и ошибок сайта.
* php – содержит в себе файл инициализации самого php а также конфигурацию статуса pm, необходимую для сбора метрик.
* prometheus – содержит в себе конфигурационный файл prometheus в формате yml-файла.
* telegraf – папка содержит в себе одноименный конфигурационный файл, необходимый для настройки подключения телеграфа и дальнейшей его передачи данных между сервисами.

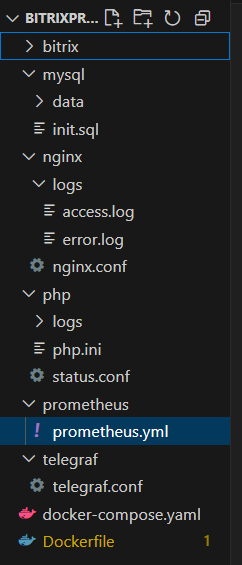


Рисунок 2 – Иерархия проекта

Теперь можно приступить и к самим файлам развертывания. Первый на очереди – докерфайл, где прописаны настройки для предоставления доступа к папке сессий php.

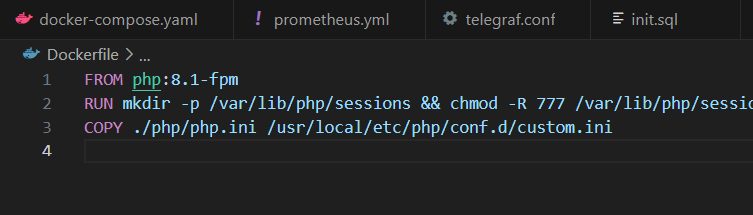


Рисунок 3 – Докерфайл

Ниже, в рисунках 4-9 изображено содержимое файла docker-compose.yaml и настройка сервисов, используемых в проекте.

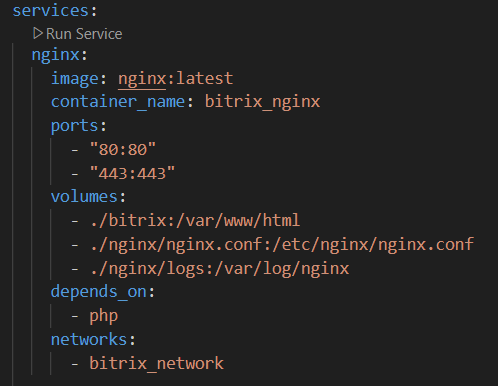


Рисунок 4 – Сервис nginx

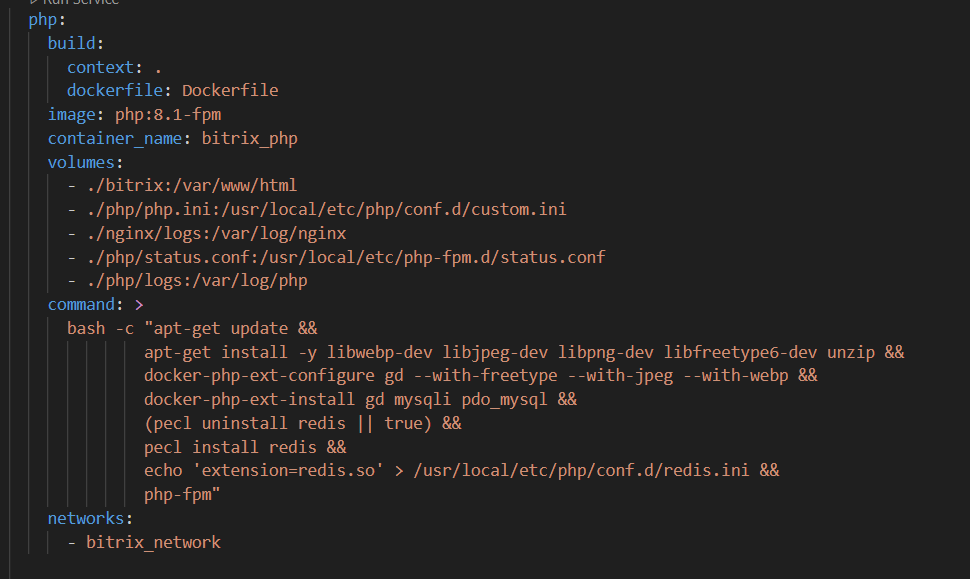


Рисунок 5 – Сервис php

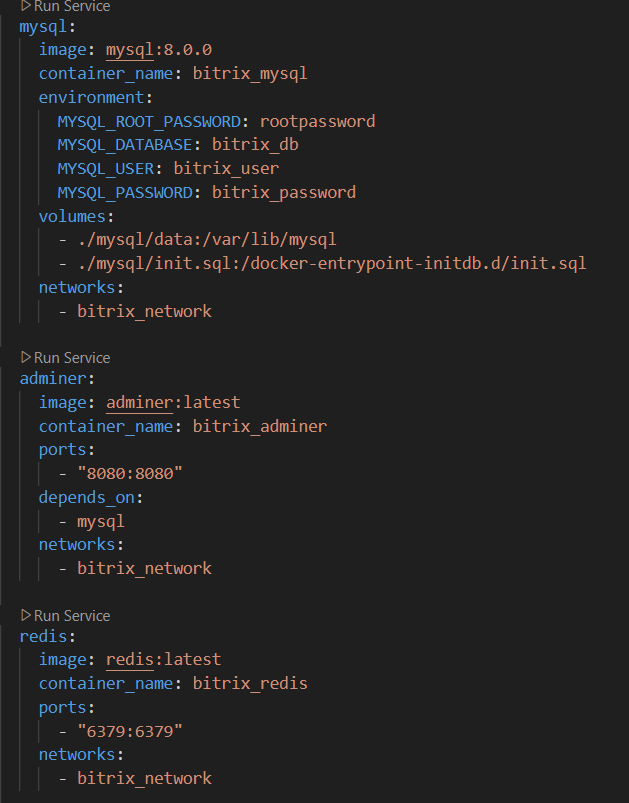


Рисунок 6 – Сервисы mysql, adminer и redis

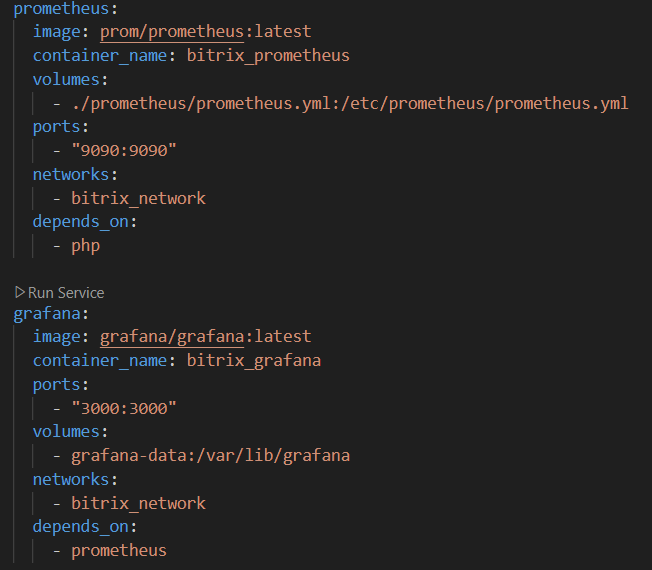


Рисунок 7 – Сервисы Prometheus и Grafana

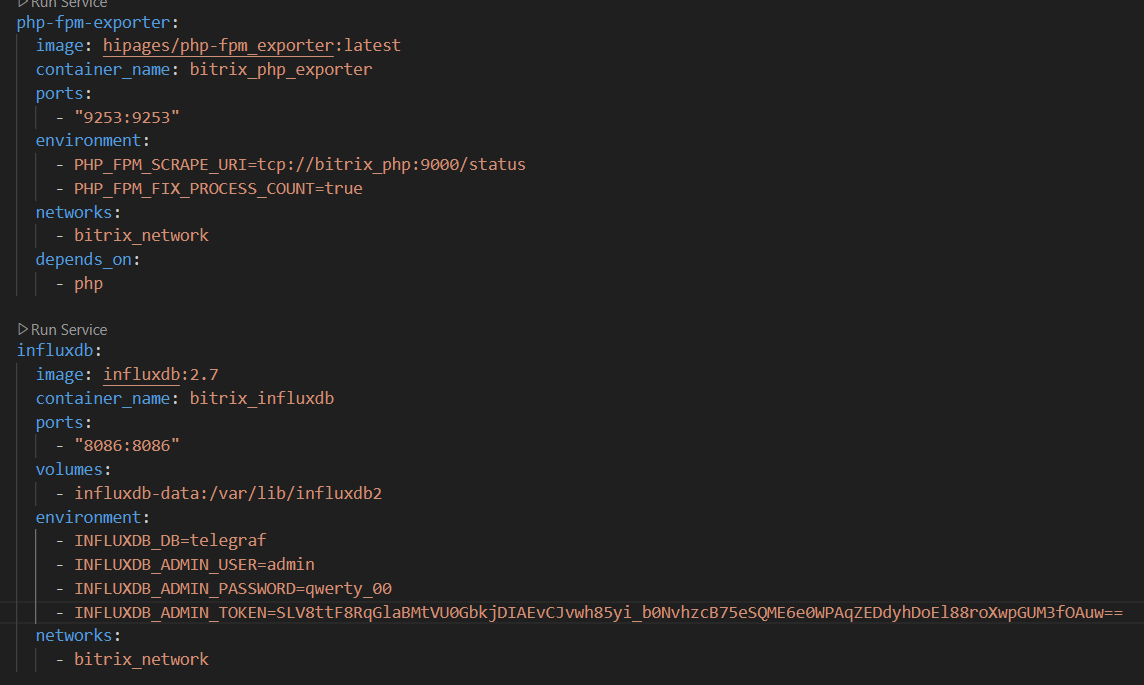


Рисунок 8 – Сервисы php-fpm-exporter и influxdb

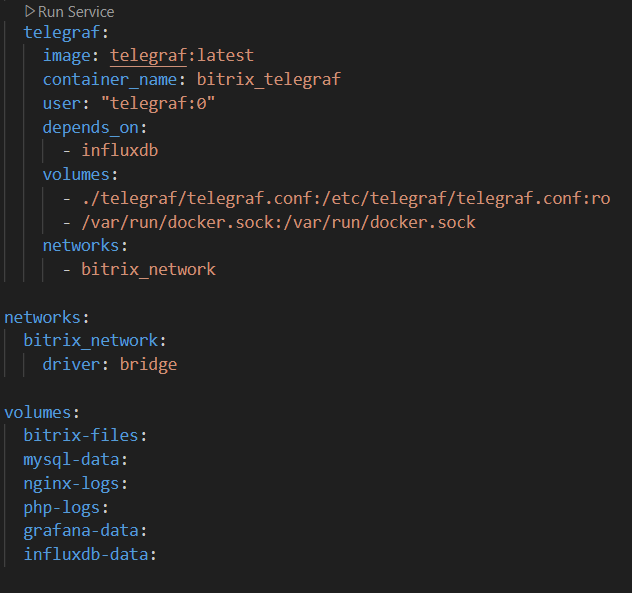


Рисунок 9 – Сервис telegraf и настройки volumes и networks

Конфигурация телеграфа содержит в себе адрес подключения, ключ, и другие параметры подключения и передачи данных.



Рисунок 10 – Конфигурация Telegraf

Конфигурация Prometheus для сбора метрик:



Рисунок 11 – Конфигурация Prometheus

Дополнительная настройка в виде конфигурации статуса php:

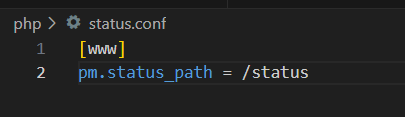


Рисунок 12 – Конфигурация статуса php

Конфигурация инициализатора php, в котором отображены его связи и основные настройки сайта.

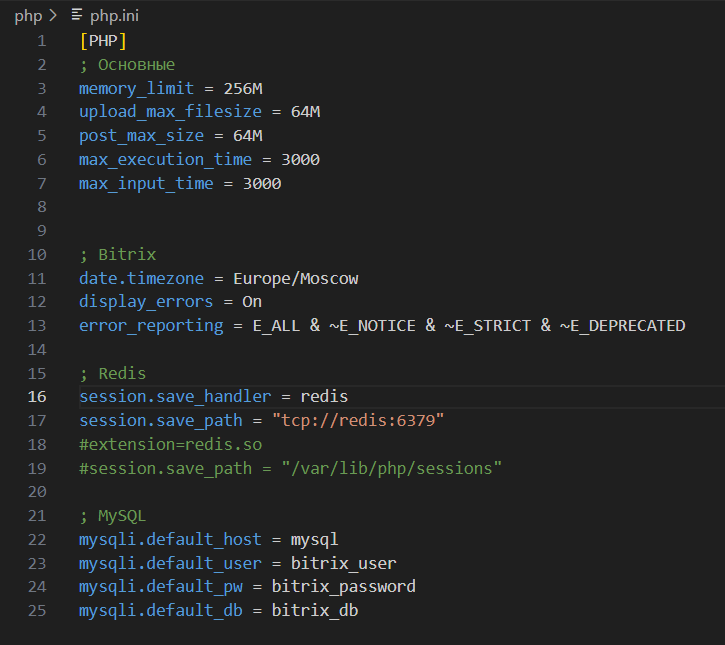


Рисунок 13 – Конфигурация php и его подключений

Конфигурация веб-сервера nginx:

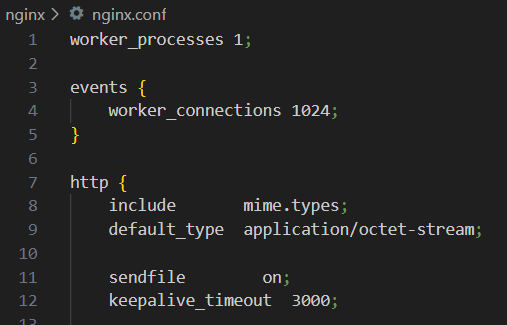


Рисунок 14 – Конфигурация nginx №1



Рисунок 15 – Конфигурация nginx №2

Логи nginx, прописанные в отдельных файлах:

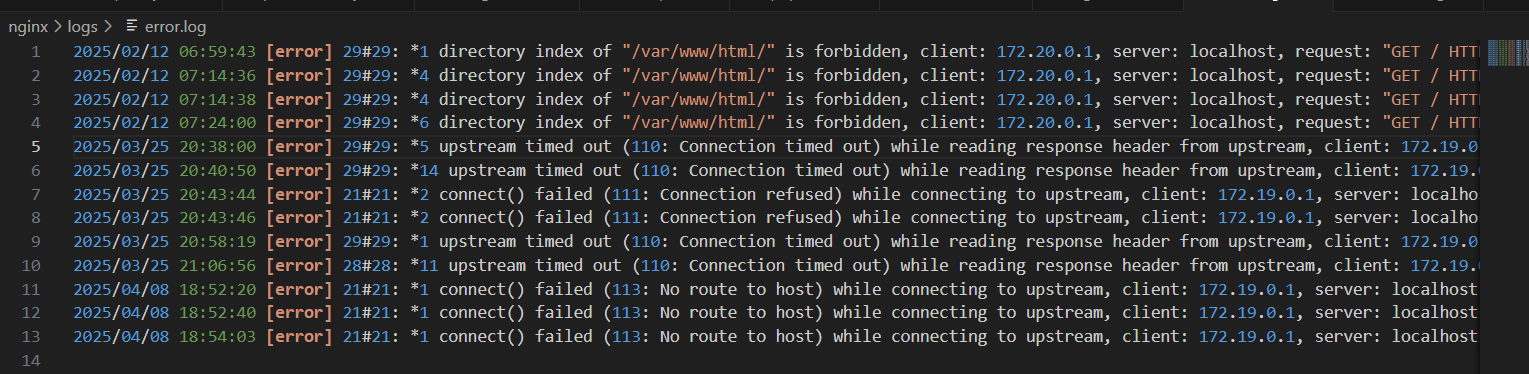


Рисунок 16 – Логи nginx об ошибках

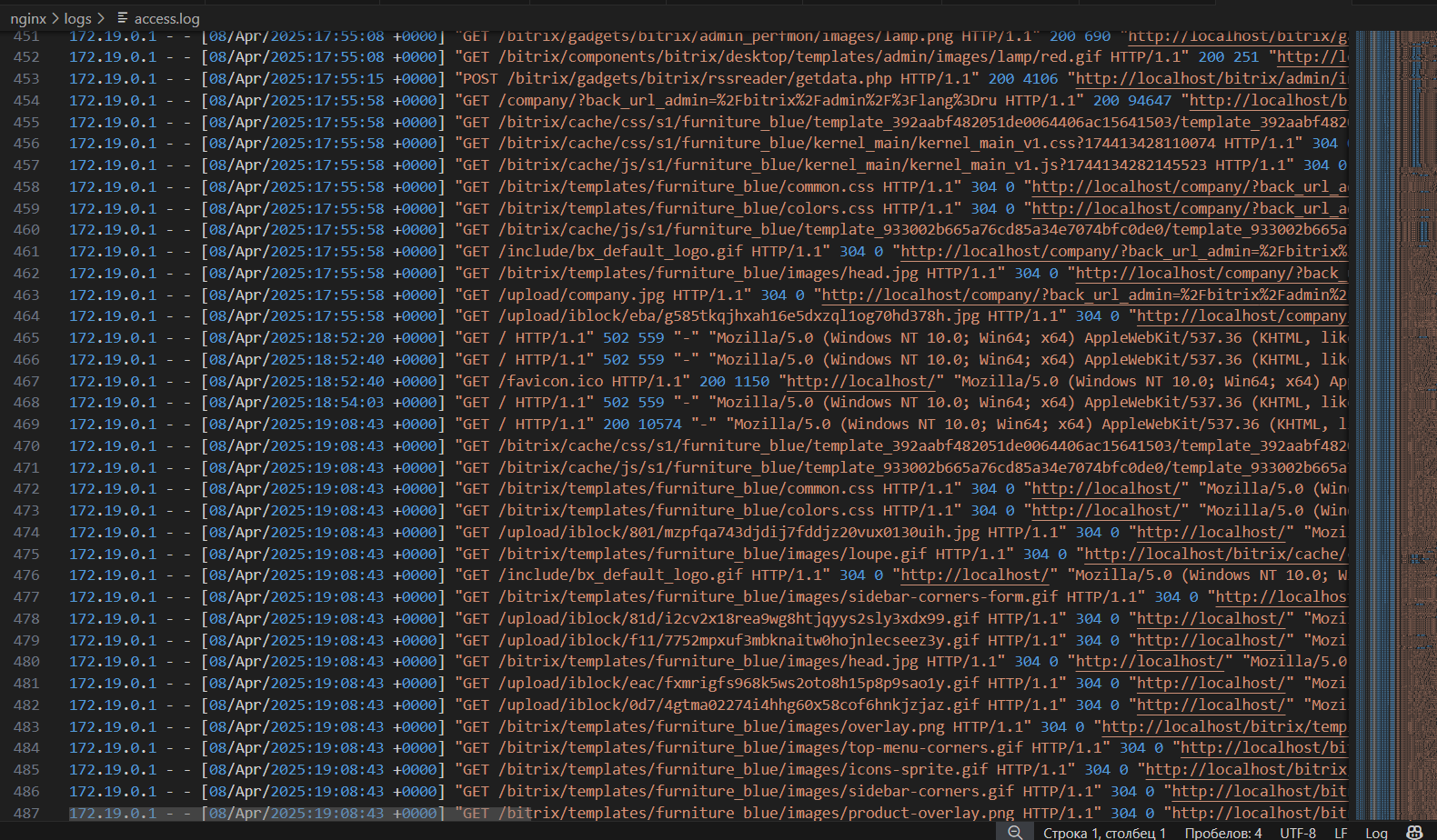


Рисунок 17 – Логи nginx обо всех запросах, их статусах и времени

Файл инициализации баз данных, необходимый для создания суперпользователя и создании БД (в случае ее отсутствия).

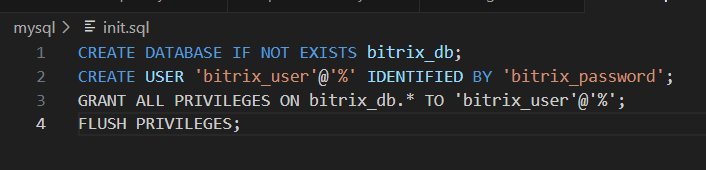


Рисунок 18 – Файл инициализации базы данных

Результат работы основной страницы:



Рисунок 19 – Основная страница

Результат работы страницы новостей:

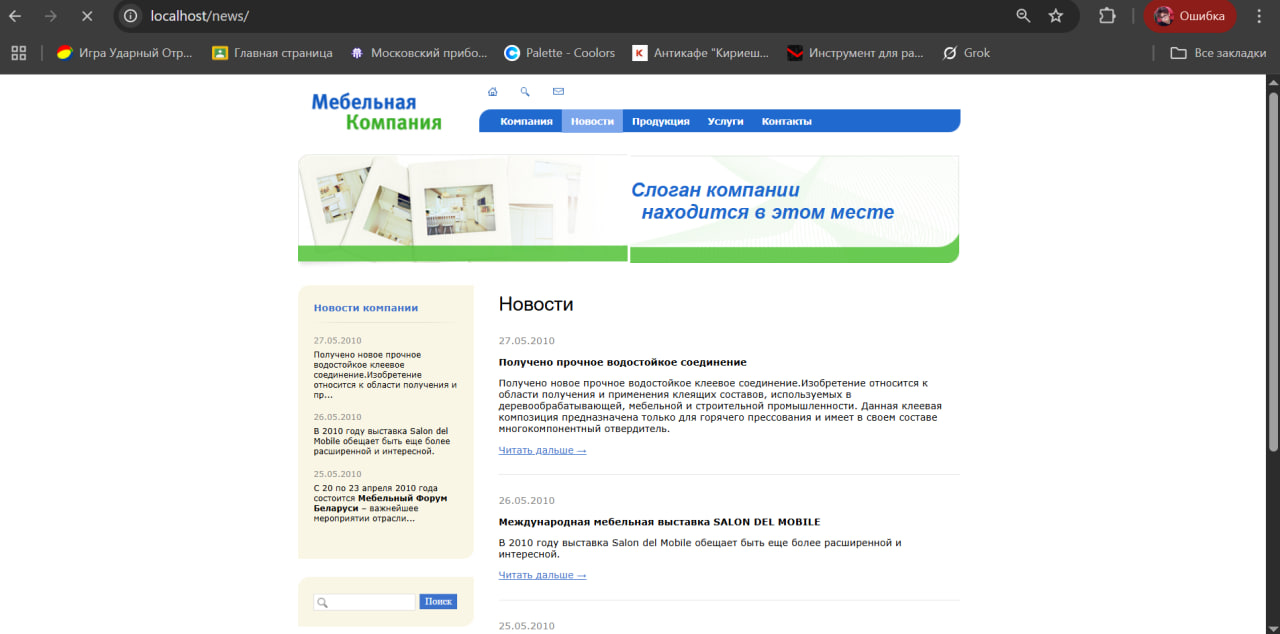


Рисунок 20 – Страница новостей

Ввод данных на странице авторизации:



Рисунок 21 – Страница авторизации

Сообщение об успешной авторизации на следующей странице:

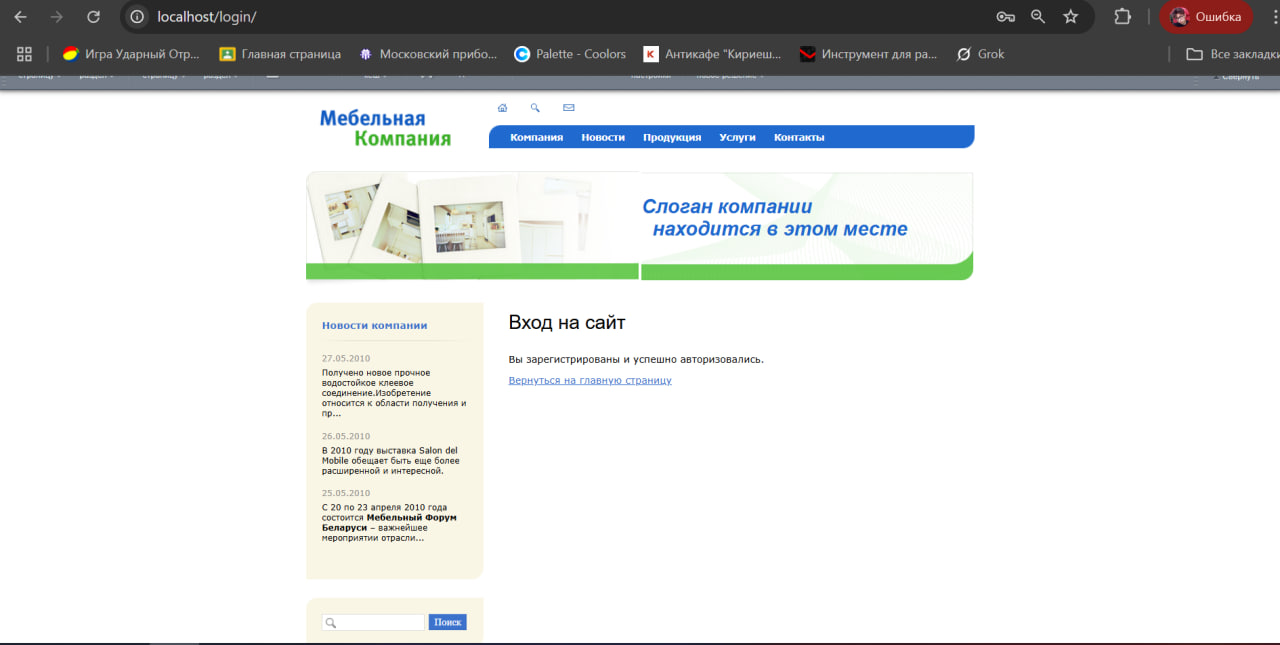


Рисунок 22 – Страница успешной авторизации

Отображение панели администрирования Bitrix:

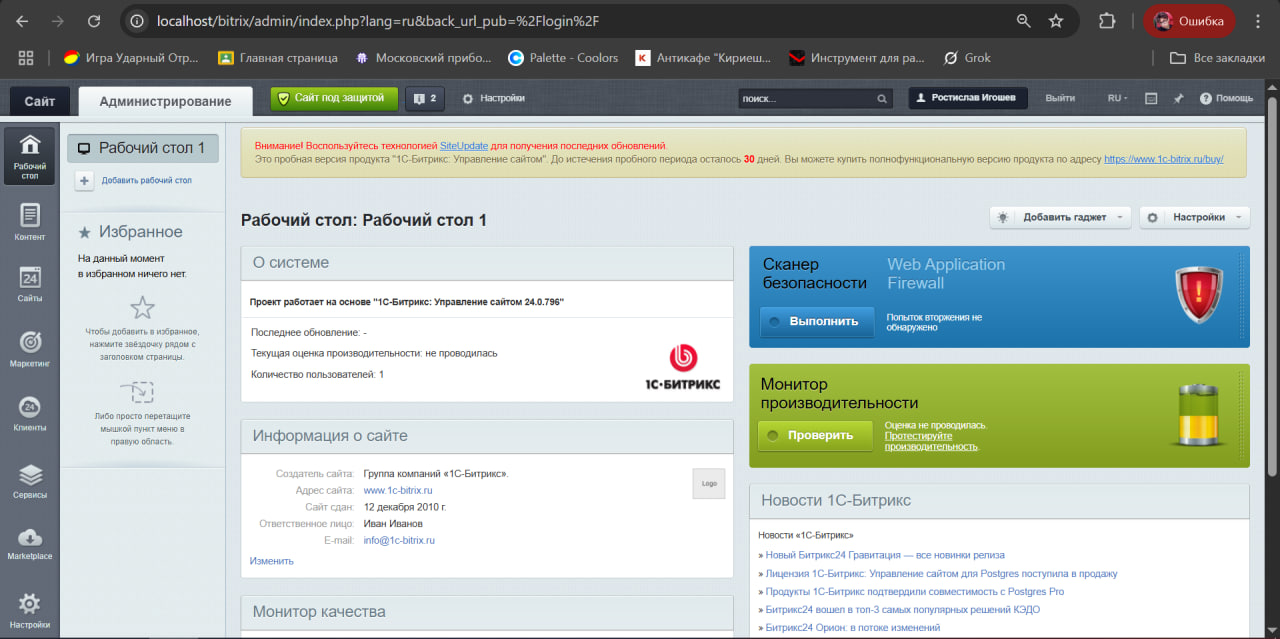


Рисунок 23 – Страница администрирования CMS Bitrix

Вход в Adminer по данным БД Bitrix:

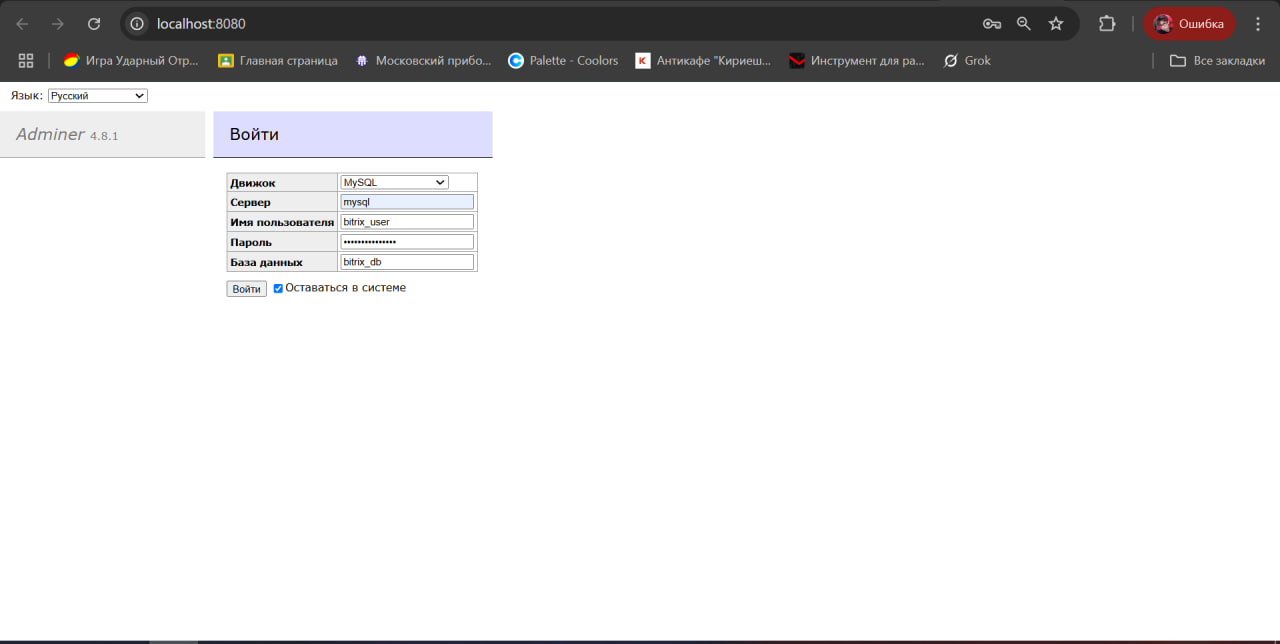


Рисунок 24 – Подключение к Adminer

Успешное подключение к Adminer и отображение таблиц БД Bitrix:

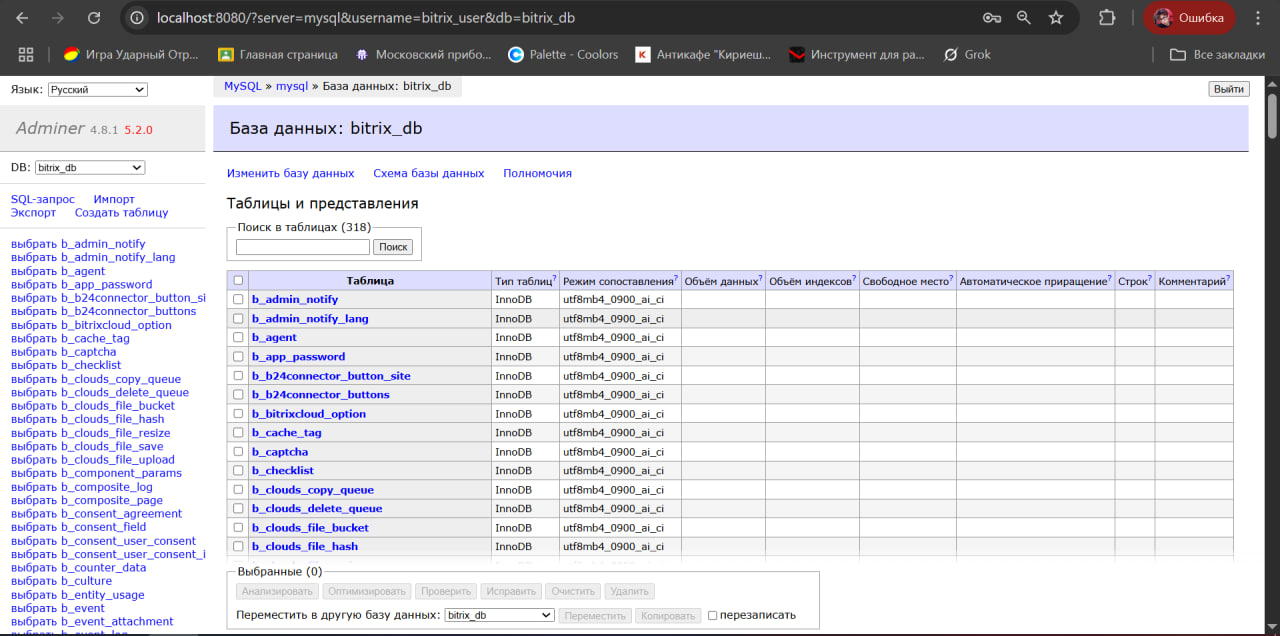


Рисунок 25 – Вывод таблиц сайта после успешного подключения

Далее, стоит проверить работу Grafana, Prometheus и InfluxDB:

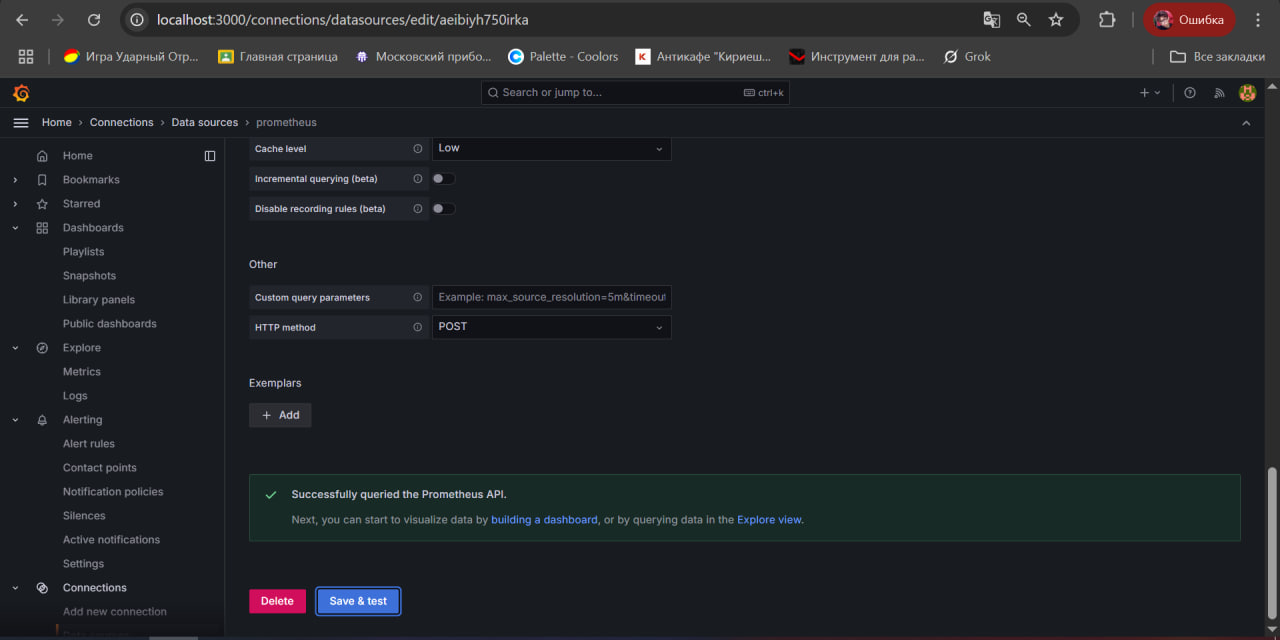


Рисунок 26 – Успешное подключение к Prometheus внутри Grafana

Результат работы графического отображения метрик в Grafana:

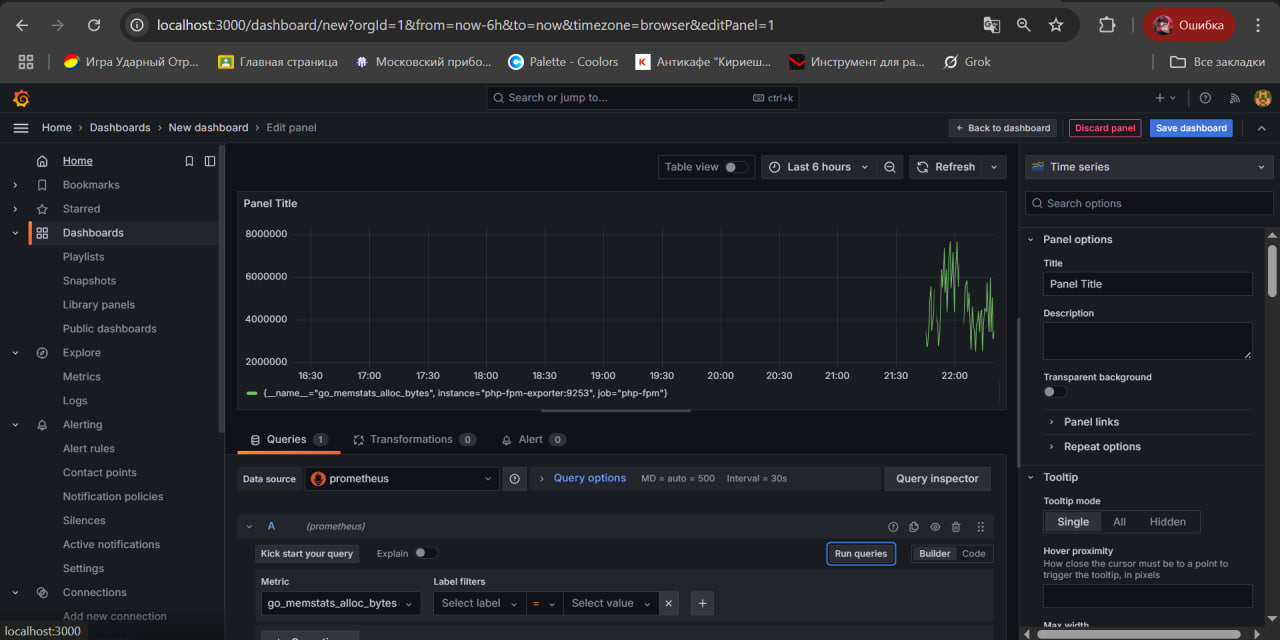


Рисунок 27 – Вывод метрик, получаемых из Prometheus в Grafana

Теперь нужно приступить к InfluxDB, введя данные для подключения:

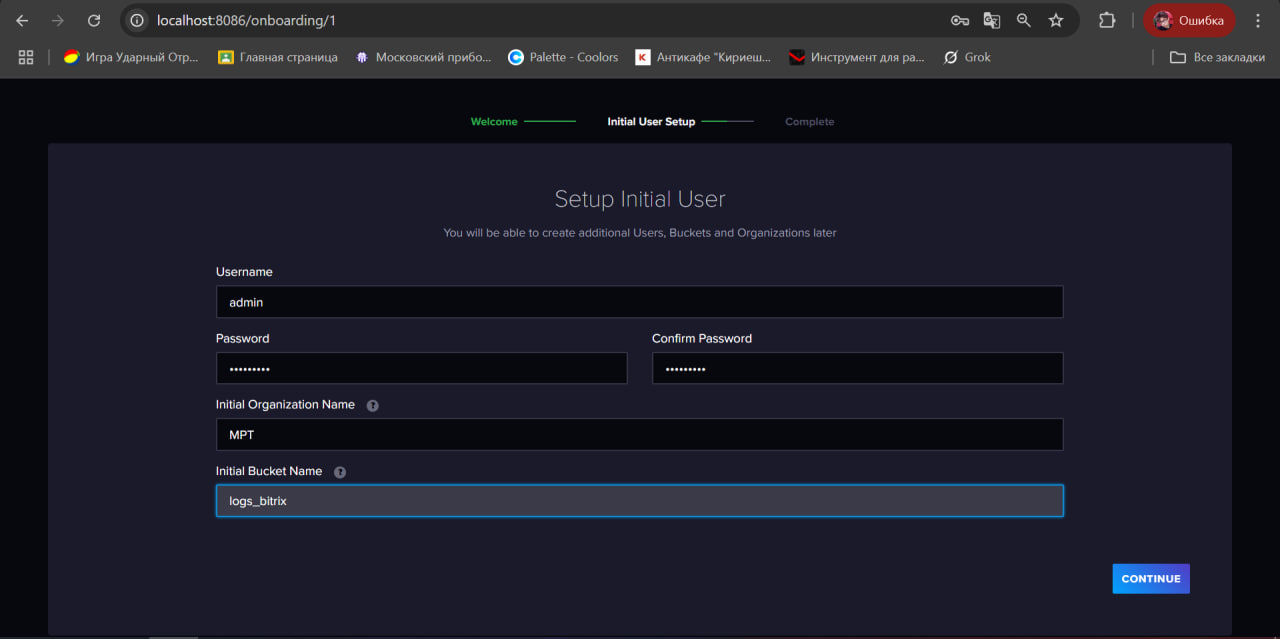


Рисунок 28 – Инициализация InfluxDB

Инициализировав InfluxDB, будет выдан API-ключ:

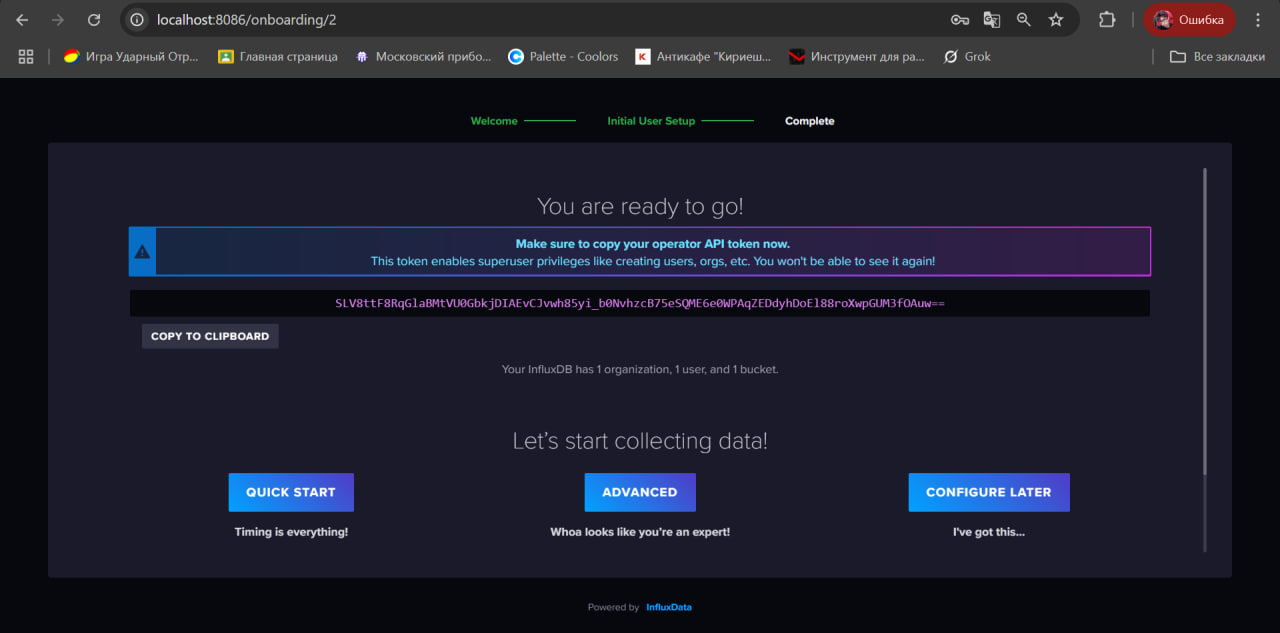


Рисунок 29 – Сообщение об успешной инициализации и выдача ключа для запросов

Далее, следует подключить телеграф к логам докера:

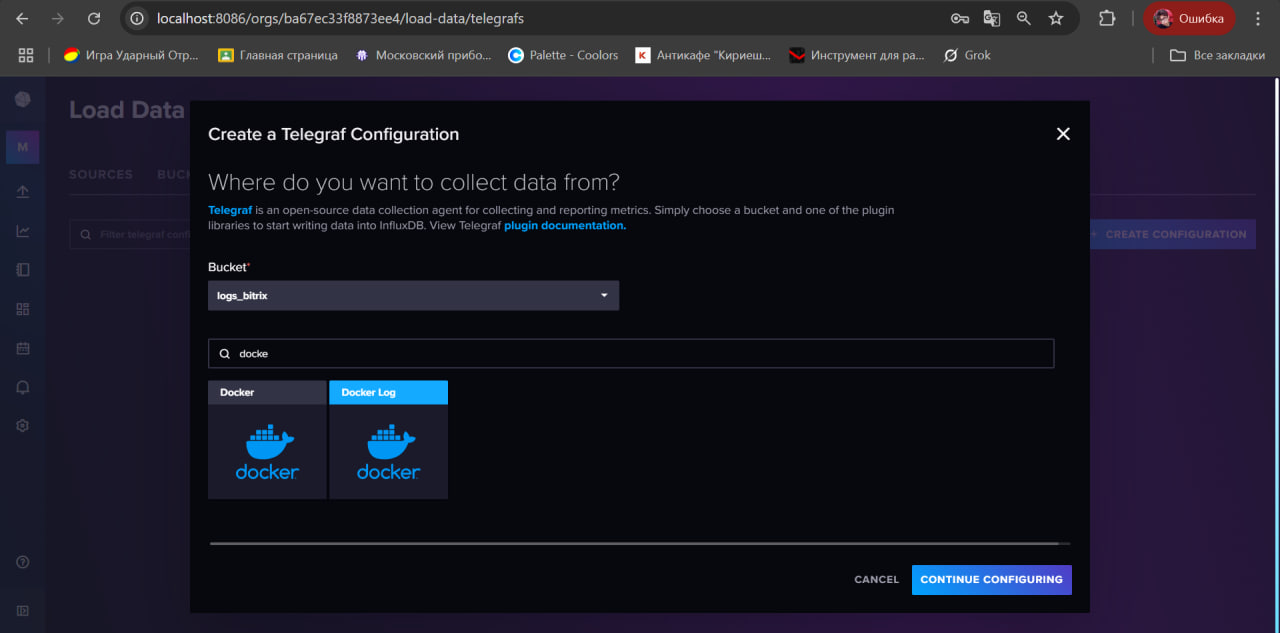


Рисунок 30 – Создание подключения телеграфа для логов докера

После этого предстоит настроить конфигурацию телеграфа:



Рисунок 31 – Настройка подключения телеграфа

Попытавшись вывести данные в InfluxDB, выведется следующая ошибка, связанная с тем, что в InfluxDB нет подходящего отображения:

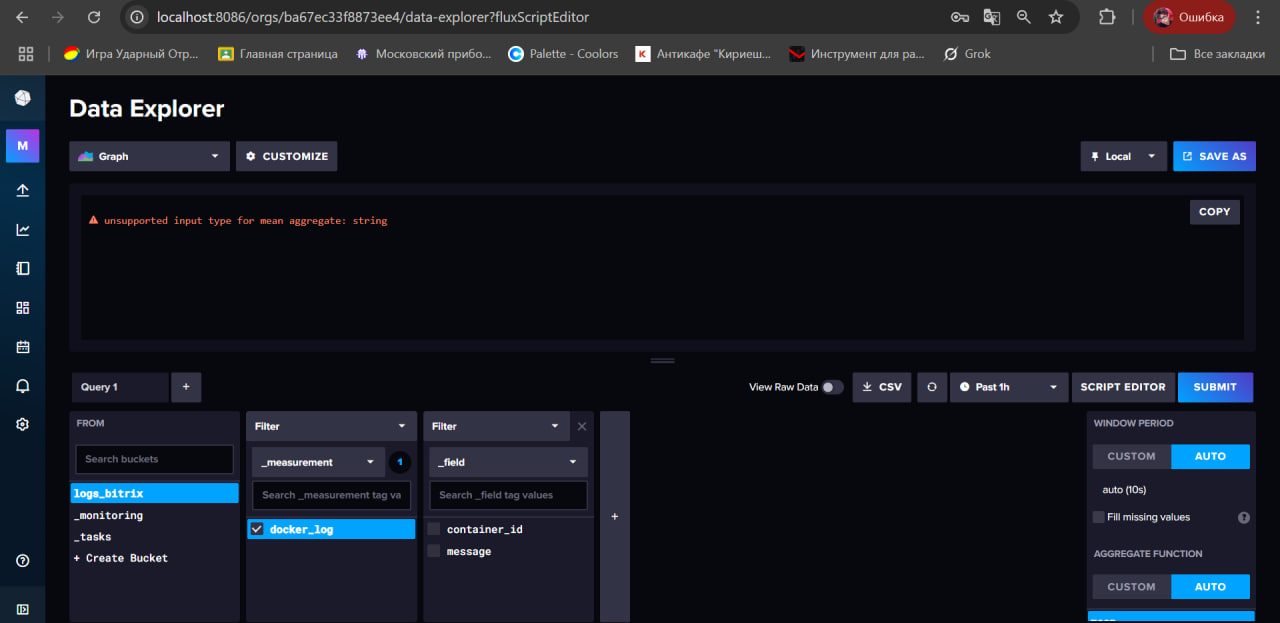


Рисунок 32 – Попытка вывести полученные данные в InfluxDB

Для отображения, необходимо подключить InfluxDB к Grafana:

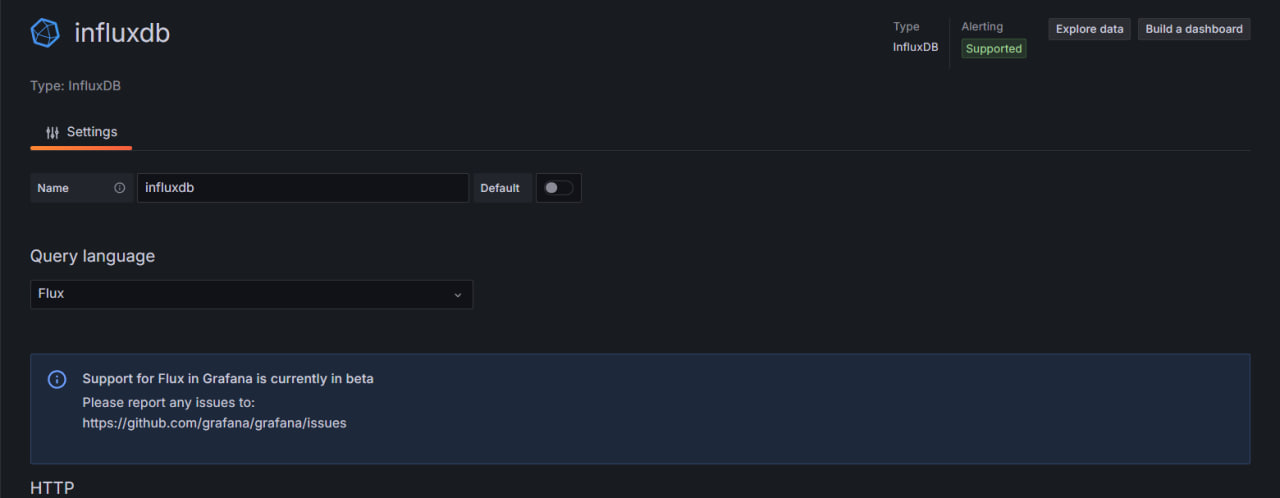


Рисунок 33 – Создание подключения InfluxDB к локальной Grafana

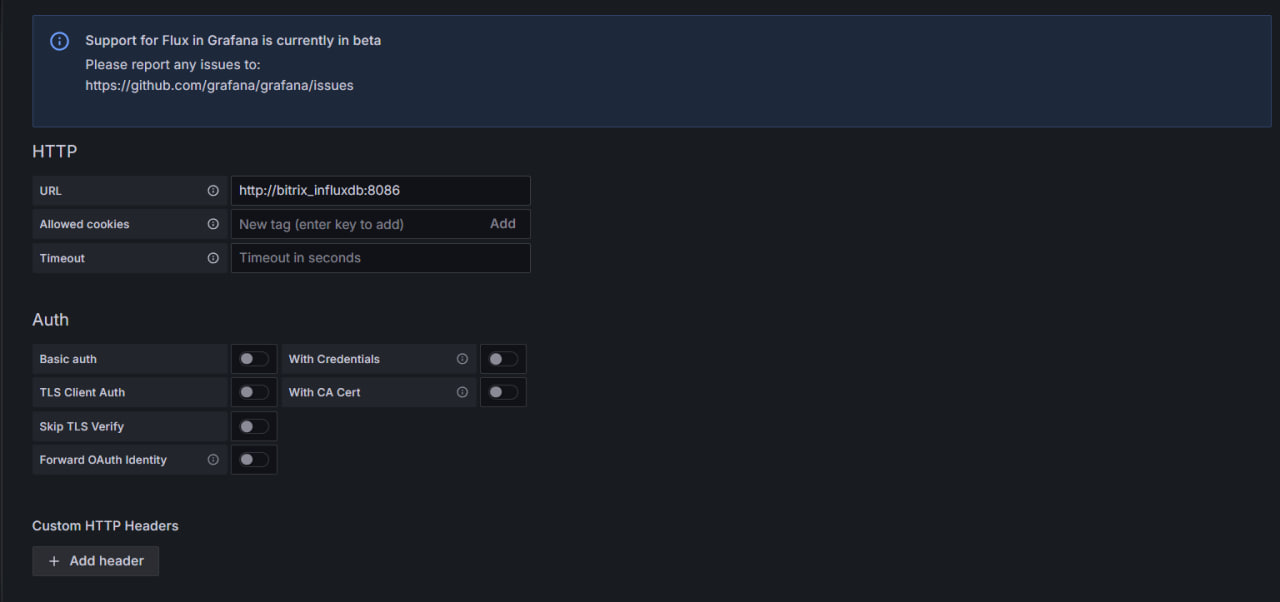


Рисунок 34 – Ввод адреса InfluxDB

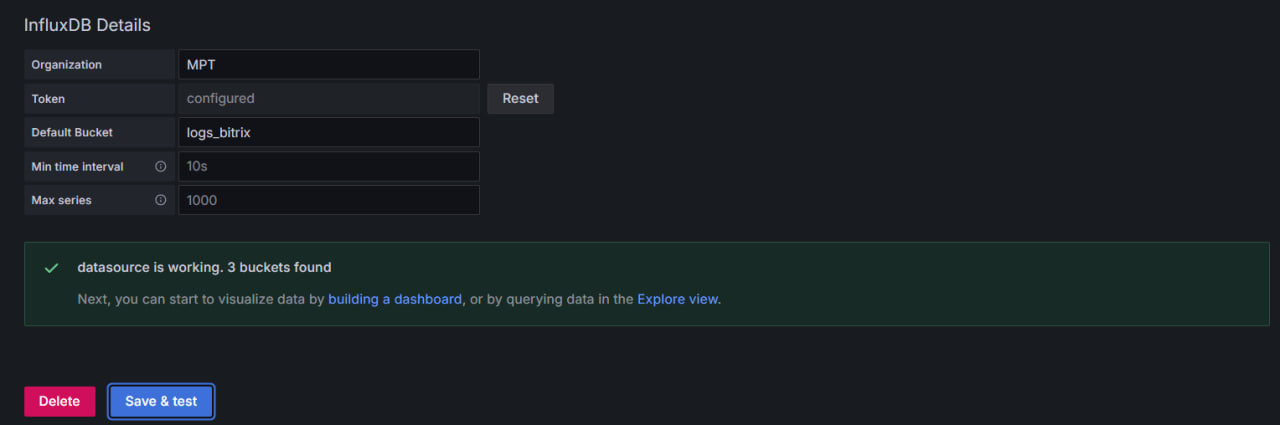


Рисунок 35 – Успешное подключение к InfluxDB

Результат работы метрик из InfluxDB в локальной Grafana:



Рисунок 36 – Успешный вывод полученных логов докера на дашборде Grafana

Вывод: в ходе выполнения практической работы был создан файл docker-compose.yml для запуска CMS Bitrix с использованием необходимых сервисов. Конфигурация была адаптирована для работы в режиме разработки.