МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт (факультет) Институт компьютерных наук

Кафедра Автоматизированные системы управления

ЛАБОРАТОРНАЯ РАБОТА №6

По дисциплине: «Операционные системы»

На тему: «Контейнеризация»

Студент АИ-23 (Группа) _____ (подпись, дата) Жданов М.С. (фамилия, инициалы)

Руководитель

кандидат наук (ученая степень, ученое звание) _____ (подпись, дата) Кургасов В.В. (фамилия, инициалы)

Липецк 2025

Задание:

1. С помощью Docker Compose на своем компьютере поднять сборку nginx+php-fpm+postgres, продемонстрировать ее работоспособность, запустив внутри контейнера демо-проект на symfony (Исходники взять отсюда https://github.com/symfony/demo /ссылка на github/. По умолчанию проект работает с sqlite-базой. Нужно заменить ее на postgres. (Для этого: 1. Создать новую БД в postgres;

2. Заменить DATABASE_URL в .env на строку подключения к postgres;

3. Создать схему БД и заполнить ее данными из фикстур, выполнив в консоли (php bin/console doctrine:schema:create php bin/console doctrine:fixtures:load)). Проект должен открываться по адресу http://demo-symfony.local/ (Код проекта должен располагаться в папке на локальном хосте) контейнеры с fpm и nginx должны его подхватывать. Для компонентов nginx, fpm есть готовые docker-образы, их можно и нужно использовать.

Нужно расшарить папки с локального хоста, настроить подключение к БД. В .env переменных для постгреса нужно указать путь к папке, где будет лежать база, чтобы она не удалялась при остановке контейнера.

На выходе должен получиться файл конфигурации docker-compose.yml и env файл с настройками переменных окружения.

Дополнительные требования: Postgres также должен работать внутри контейнера. В .env переменных нужно указать путь к папке на локальном хосте, где будут лежать файлы БД, чтобы она не удалялась при остановке контейнера.

Ход работы:

Для начала я установил docker на свою виртуальную машину. Для этого первым делом нужно обновить систему командами:

```
sudo apt update

sudo apt upgrade -y
```

Далее нужно было поставить зависимости:

```
sudo apt install ca-certificates curl gnupg lsb-release -y
```

Следующим шагом шло добавление ключей и репозитория Docker:

```
sudo mkdir -p /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] \

https://download.docker.com/linux/ubuntu \

$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

После чего можно было приступить к установке Docker и Compose:

```
sudo apt update

sudo apt install docker-ce docker-ce-cli containerd.io docker-compose-plugin -y
```

После этого проведём проверку установки командами:

```
docker --version

docker compose version
```

Статус установки Docker можно видеть на рисунке 1:

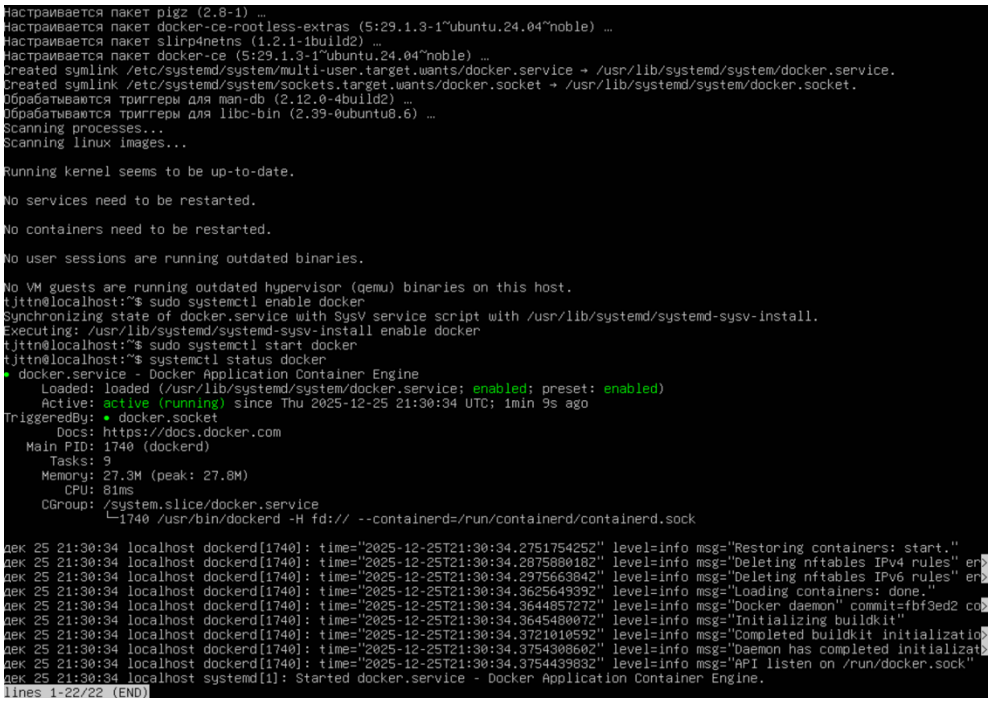

Рисунок 1 – Итог установки Docker

Проверка установки путём запуска «hello world» - рисунок 2:

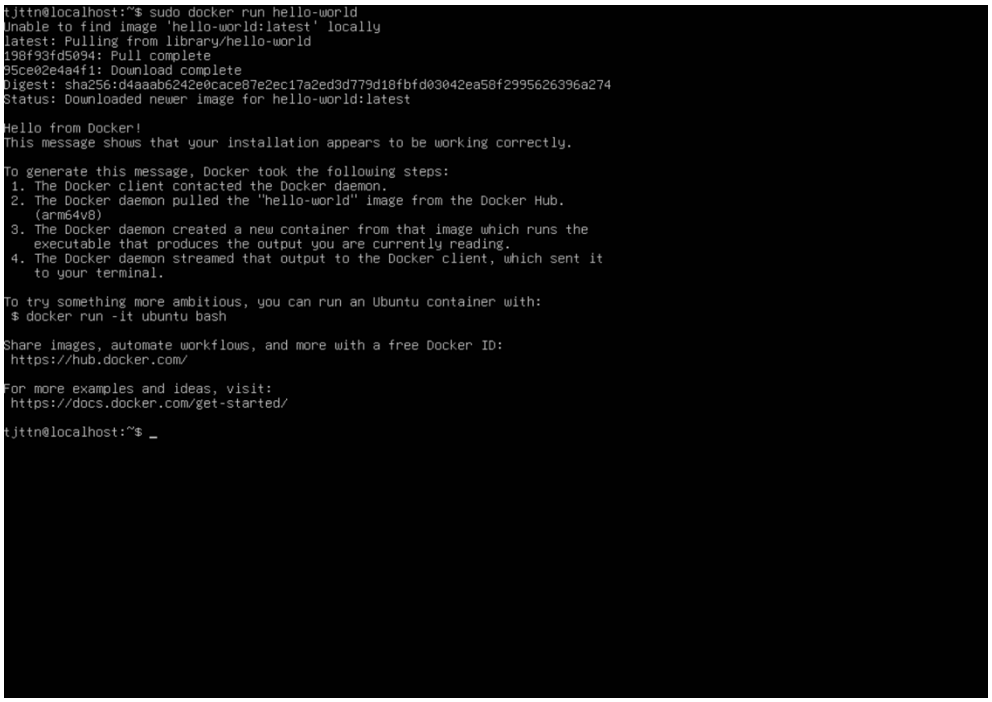

Рисунок 2 – Проверка Docker

Далее для теста я создал простой yml файл и html файл и запустил, чтобы удостовериться, что всё работает – рисунок 3:

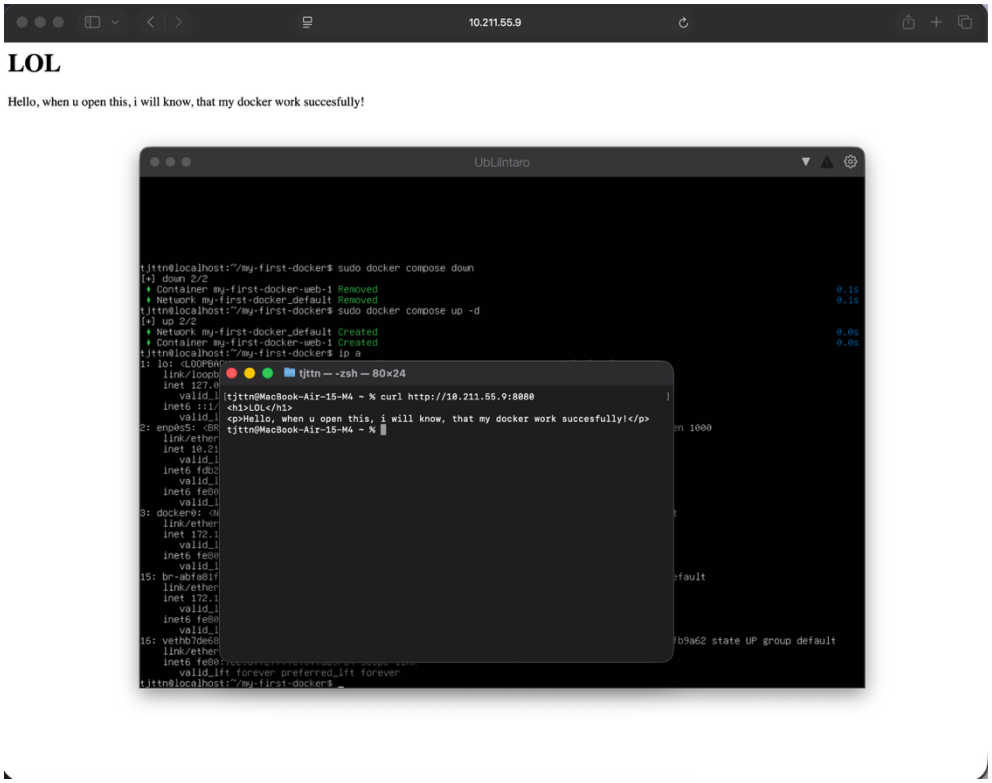

Рисунок 3 – Тестовый запуск

Установка symphony:

Для установки symphony я клонировал репозиторий с github на виртуальную машину – рисунок 4:

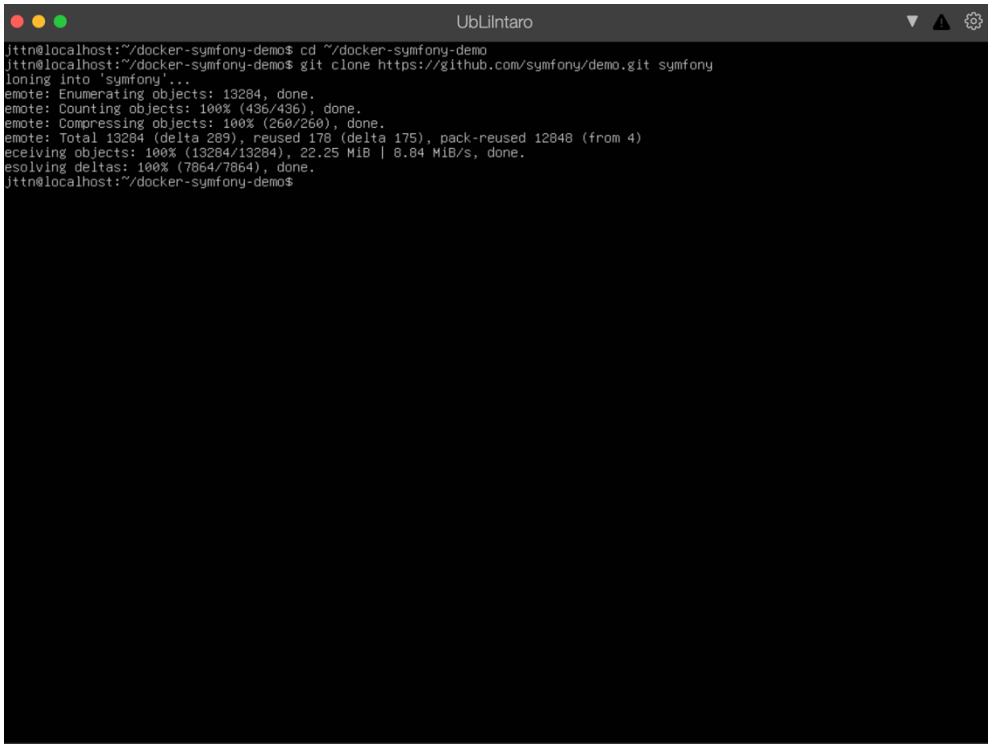

Рисунок 4 – Клонирование репозитория

Далее нужно создать правильную структуру проекта – рисунок 5:

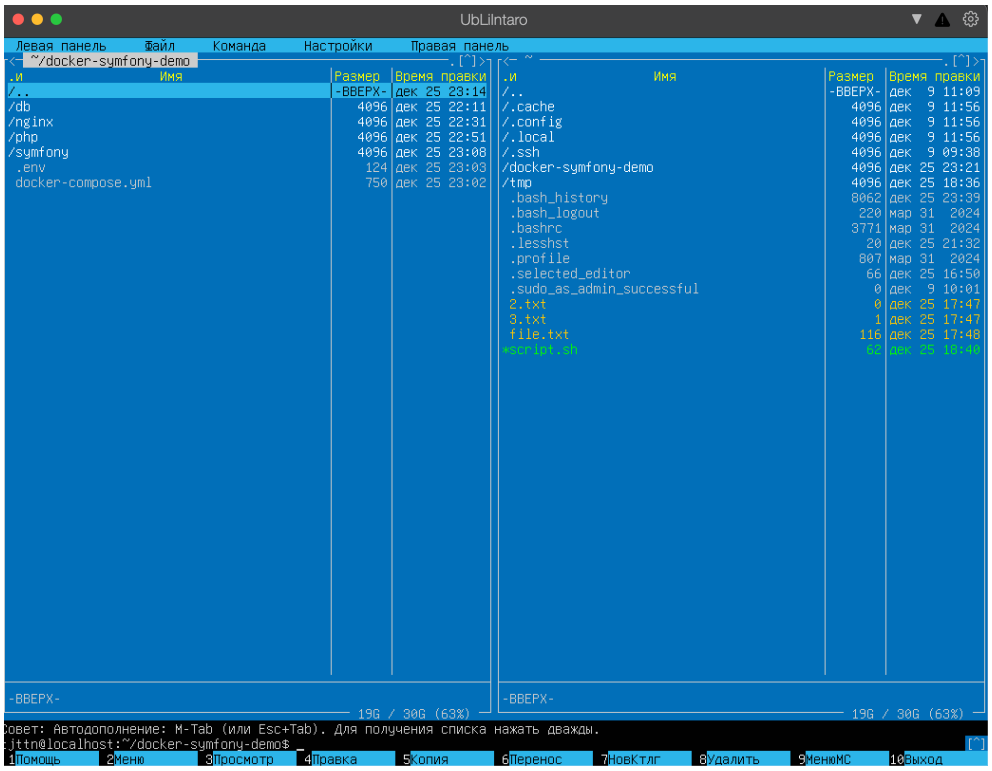

Рисунок 5 – Структура докера

А также нужно написать Dockerfile, который находится в php – для работы с Composer и расширениями:

```
FROM php:8.2-fpm

RUN apt-get update && apt-get install -y git unzip libpq-dev libicu-dev libzip-dev
\

    && docker-php-ext-install pdo pdo_pgsql intl zip opcache

COPY --from=composer:2 /usr/bin/composer /usr/bin/composer

WORKDIR /var/www/symfony
```

В docker-compose.yml содержатся три сервиса:

- Php – symphony и composer;
- Nginx – прокси, монтирует symphony;
- Postgres - бд;

Текст .env файла для PostgreSQL и окружения Symfony выглядит так:

```
POSTGRES_USER=demo

POSTGRES_PASSWORD=demo

POSTGRES_DB=demo

POSTGRES_DATA_PATH=./db

APP_ENV=dev
```

Чтобы удостовериться что всё корректно установлено и написано выполним команду с рисунка 6:

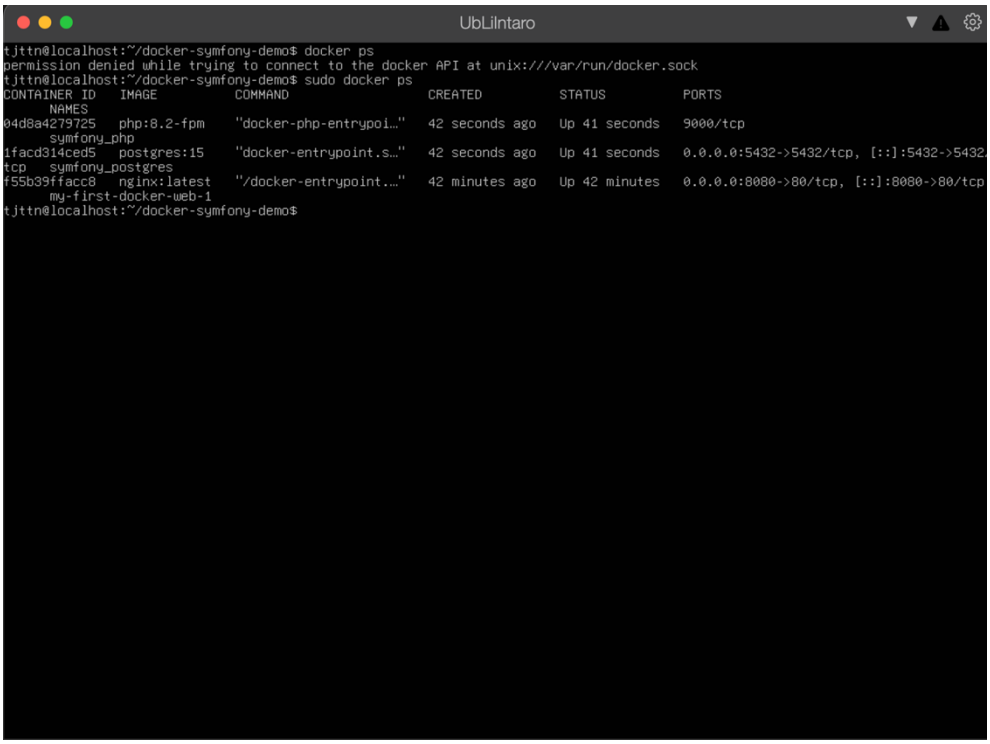

Рисунок 6 – Финальная проверка

Далее последовала сборка контейнеров:

```
sudo docker compose build

sudo docker compose up -d
```

Установка зависимостей Symfony внтури контейнера php выглядит следующим образом:

```
sudo docker exec -it symfony_php bash

composer install

php bin/console doctrine:schema:create

php bin/console doctrine:fixtures:load
```

А результатом будет – рисунок 7:

Рисунок 7 – Результат установки зависимостей

После чего можно проверять сайт на работоспособность – рисунок 8:

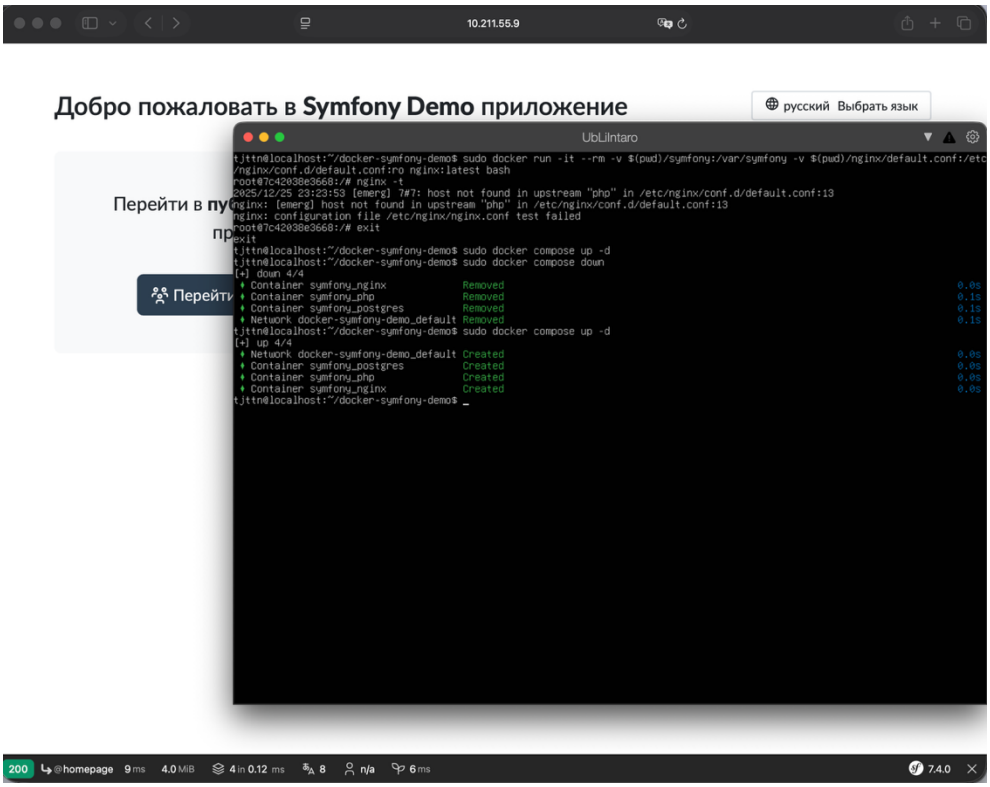

Рисунок 8 – Итог работы

Вывод:

В ходе работы я изучил современные методы разработки ПО в динамических и распределенных средах на примере контейнеров Docker.