



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

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

Институт (факультет)

Компьютерных наук

Кафедра

Автоматизированных систем управления

**Лабораторная работа №6**

**По дисциплине: «INTARO»**

**На тему: «Контейнеризация»**

Студент

АИ-23-1

Группа

Подпись, дата

Пронин В.О.

Фамилия, инициалы

Руководитель

Канд. Техн. Наук, Доцент

Ученая степень, ученое звание

Подпись, дата

Качановский Ю.П.

Фамилия, инициалы

Липецк 2025 г.

## 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 установим 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
```

```
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2025-12-26 17:27:30 MSK; 5h 13min ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
    Main PID: 1898 (dockerd)
      Tasks: 14
     Memory: 99.7M
        CPU: 1min 39.844s
    CGroup: /system.slice/docker.service
           └─1898 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Рисунок 1 – статус docker

Дальше клонируем тестовый проект на symphony в папку /var командой:

```
Git clone https://github.com/symfony/demo
```

После этого переходим в папку demo командой :cd demo и запускаем комадной:

```
php -S localhost:8000 -t public/
```

после этого, по адресу <http://localhost:8000> отобразится проект.

Теперь сделаем тоже самое, но в docker, для этого нам понадобится dockerfile и docker-compose.yml

В dockerfile пишем следующее:

```
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
```

И в docker-compose.yml:

```
version: "3"
services:
  app:
    container_name: docker-node-mongo
    restart: always
    build: .
    ports:
```

```
    - "8000:8000"
  depends_on:
    - postgres
  environment:
    DATABASE_URL:
"postgresql://user:password@postgres:5432/dbname"
    postgres:
      container_name: postgres
      image: postgres:15-alpine
      restart: always
      ports:
        - "5432:5432"
      environment:
        POSTGRES_DB: dbname
        POSTGRES_USER: user
        POSTGRES_PASSWORD: password
      volumes:
        - postgres_data:/var/lib/postgresql/data
volumes:
  postgres_data:
```

**После чего запускаем командой:**

```
docker compose up
```

```

root@captain-HP-Laptop-15-rb0xx:/var/apps/demo# docker compose up
WARN[0000] /var/apps/demo/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 12/12
  ✓ postgres Pulled
    ✓ 1074353ee0d Already exists
    ✓ d2e5feb410e8 Pull complete
    ✓ ff7ac73e977d Pull complete
    ✓ 25fbel3bb576 Pull complete
    ✓ 060e3656c186 Pull complete
    ✓ 9fa3a17171bc Pull complete
    ✓ 1de01a2b564f Pull complete
    ✓ 141979064465 Pull complete
    ✓ 86a96880bc80 Pull complete
    ✓ c5fd49a04d70 Pull complete
    ✓ 90b1b1a4157a Pull complete
[+] Running 3/3
  ✓ Volume "demo_postgres_data" Created
  ✓ Container docker-node-mongo Recreated
  ✓ Container postgres Recreated
Attaching to docker-node-mongo, postgres
postgres | The files belonging to this database system will be owned by user "postgres".
postgres | This user must also own the server process.
postgres |
postgres | The database cluster will be initialized with locale "en_US.utf8".
postgres | The default database encoding has accordingly been set to "UTF8".
postgres | The default text search configuration will be set to "english".
postgres |
postgres | Data page checksums are disabled.
postgres |
postgres | fixing permissions on existing directory /var/lib/postgresql/data ... ok
postgres | creating subdirectories ... ok
postgres | selecting dynamic shared memory implementation ... posix
postgres | selecting default max_connections ... 100
postgres | selecting default shared_buffers ... 128MB
postgres | selecting default time zone ... UTC
postgres | creating configuration files ... ok
docker-node-mongo | [Fri Dec 26 21:04:36 2025] PHP 8.2.30 Development Server (http://0.0.0.0:8000) started
postgres | running bootstrap script ... ok
postgres | sh: locale: not found
postgres | 2025-12-26 21:04:36.503 UTC [34] WARNING: no usable system locales were found
postgres | performing post-bootstrap initialization ... ok
postgres | syncing data to disk ... ok
postgres |
postgres | initdb: warning: enabling "trust" authentication for local connections
postgres | initdb: hint: You can change this by editing pg_hba.conf or using the option -A, or --auth-local and --auth-host, the next time you run initdb.
postgres |
postgres | Success. You can now start the database server using:
postgres |
postgres |     pg_ctl -D /var/lib/postgresql/data -l logfile start
postgres |
postgres | waiting for server to start...2025-12-26 21:04:38.603 UTC [40] LOG: starting PostgreSQL 15.15 on x86_64-pc-linux-musl, compiled by gcc (Alpine 15.2.0) 15.2.0, 64-bit
postgres | 2025-12-26 21:04:38.605 UTC [40] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
postgres | 2025-12-26 21:04:38.612 UTC [43] LOG: database system was shut down at 2025-12-26 21:04:37 UTC
postgres | 2025-12-26 21:04:38.619 UTC [40] LOG: database system is ready to accept connections
postgres | done
postgres | server started
postgres | CREATE DATABASE
postgres |
postgres | /usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/*
postgres |
postgres | waiting for server to shut down...2025-12-26 21:04:38.813 UTC [40] LOG: received fast shutdown request
postgres | 2025-12-26 21:04:38.814 UTC [40] LOG: aborting any active transactions
postgres | 2025-12-26 21:04:38.822 UTC [40] LOG: background worker "logical replication launcher" (PID 46) exited with exit code 1
postgres | 2025-12-26 21:04:38.822 UTC [41] LOG: shutting down
postgres | 2025-12-26 21:04:38.823 UTC [41] LOG: checkpoint starting: shutdown immediate
postgres | 2025-12-26 21:04:39.001 UTC [41] LOG: checkpoint complete: wrote 921 buffers (5.6%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.018 s, sync=0.154 s, total=0.17
postgres | 2025-12-26 21:04:39.010 UTC [40] LOG: database system is shut down

```

## Рисунок 2 – вывод

И теперь наш проект запущенный в docker, доступен по адресу <http://localhost:8000>

## Вывод:

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