

# **Липецкий государственный технический университет**

Факультет автоматизации и информатики

Автоматизированные системы управления

## **ЛАБОРАТОРНАЯ РАБОТА №5 – ЧАСТЬ 1**

по курсу ДПО Интаро - Linux

«Контейнеризация»

Студент

Митина М. В.

Группа     ПИ-20-1

Руководитель  
доц.

Кургасов В.В.

Липецк 2022 г.

## Цель работы

Изучить современные методы разработки ПО в динамических и распределенных средах на примере контейнеров Docker.

## Задание кафедры

1. С помощью Docker Compose на своем компьютере поднять сборку nginx+phpfpm+postgres, продемонстрировать ее работоспособность, запустив внутри контейнера демо-проект на symfony (Исходники взять отсюда <https://github.com/symfony/demo>). 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.

## Оглавление

1. Ход работы.....	4
Установка необходимых пакетов .....	4
Установка Docker и docker compose:.....	4
Установка php v8.1. ....	4
Установка symfony: .....	5
Установка Composer.....	5
Настройка проекта.....	6
Запуск сервера symfony и проекта demo.....	6
Настройка docker, сборка проекта.....	10
Вывод.....	16

## 1. Ход работы

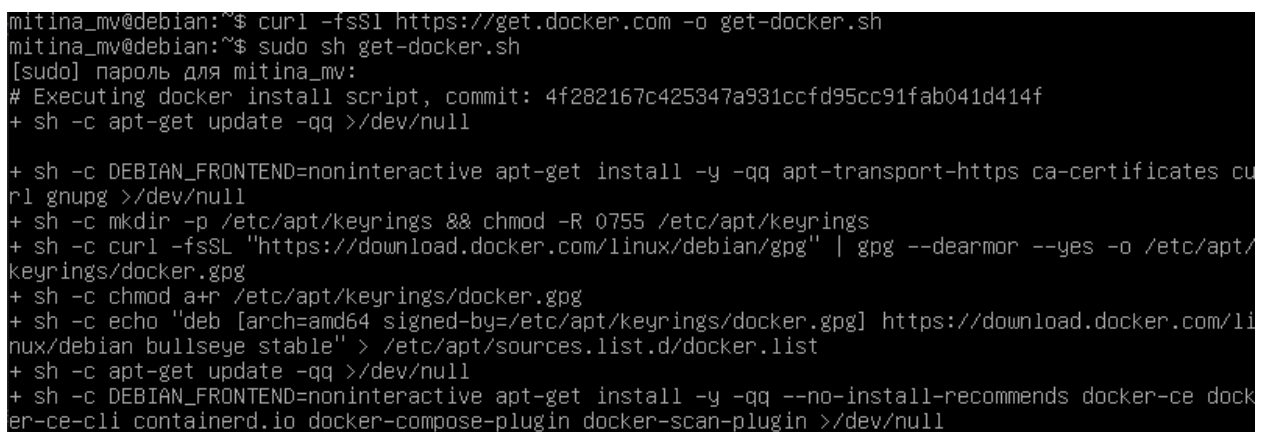
### Установка необходимых пакетов

#### Установка Docker и docker compose:

```
curl -fsSL https://get.docker.com -o get-docker.sh
```

```
DRY_RUN=1 sudo sh ./get-docker.sh
```

Мы скачиваем файл со скриптом для установки docker и всех необходимых пакетов, а после запускаем этот скрипт на выполнение.



```
mitina_mv@debian:~$ curl -fsSL https://get.docker.com -o get-docker.sh
mitina_mv@debian:~$ sudo sh get-docker.sh
[sudo] пароль для mitina_mv:
# Executing docker install script, commit: 4f282167c425347a931ccfd95cc91fab041d414f
+ sh -c apt-get update -qq >/dev/null

+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificates curl gnupg >/dev/null
+ sh -c mkdir -p /etc/apt/keyrings && chmod -R 0755 /etc/apt/keyrings
+ sh -c curl -fsSL "https://download.docker.com/linux/debian/gpg" | gpg --dearmor --yes -o /etc/apt/keyrings/docker.gpg
+ sh -c chmod a+r /etc/apt/keyrings/docker.gpg
+ sh -c echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debian bullseye stable" > /etc/apt/sources.list.d/docker.list
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq --no-install-recommends docker-ce docker-ce-cli containerd.io docker-compose-plugin docker-scan-plugin >/dev/null
```

Рисунок 1 — скачивание скрипта и ход установки docker

#### Установка php v8.1.

По умолчанию на дебиан ставится php7.4, потому что версия >8.0 в официальных репозиториях дебиан еще нет. Но под новую версию демо проекта symfony версия 7.4 не подходит (это выяснилось гораздо позже этапа загрузки), поэтому необходимо установить именно 8.0 или выше.

```
sudo apt install -y lsb-release ca-certificates apt-transport-https software-properties-common gnupg2
echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" | sudo tee /etc/apt/sources.list.d/sury-php.list
wget -q0 - https://packages.sury.org/php/apt.gpg | sudo apt-key add -
sudo apt -y install php8.1
```

Большинство необходимых пакетов под php будут установлены автоматически, но мы догрузим некоторые пакеты, которые будут необходимы для корректной работы symfony: sqlite, xml, psql, mbstring и т.д.

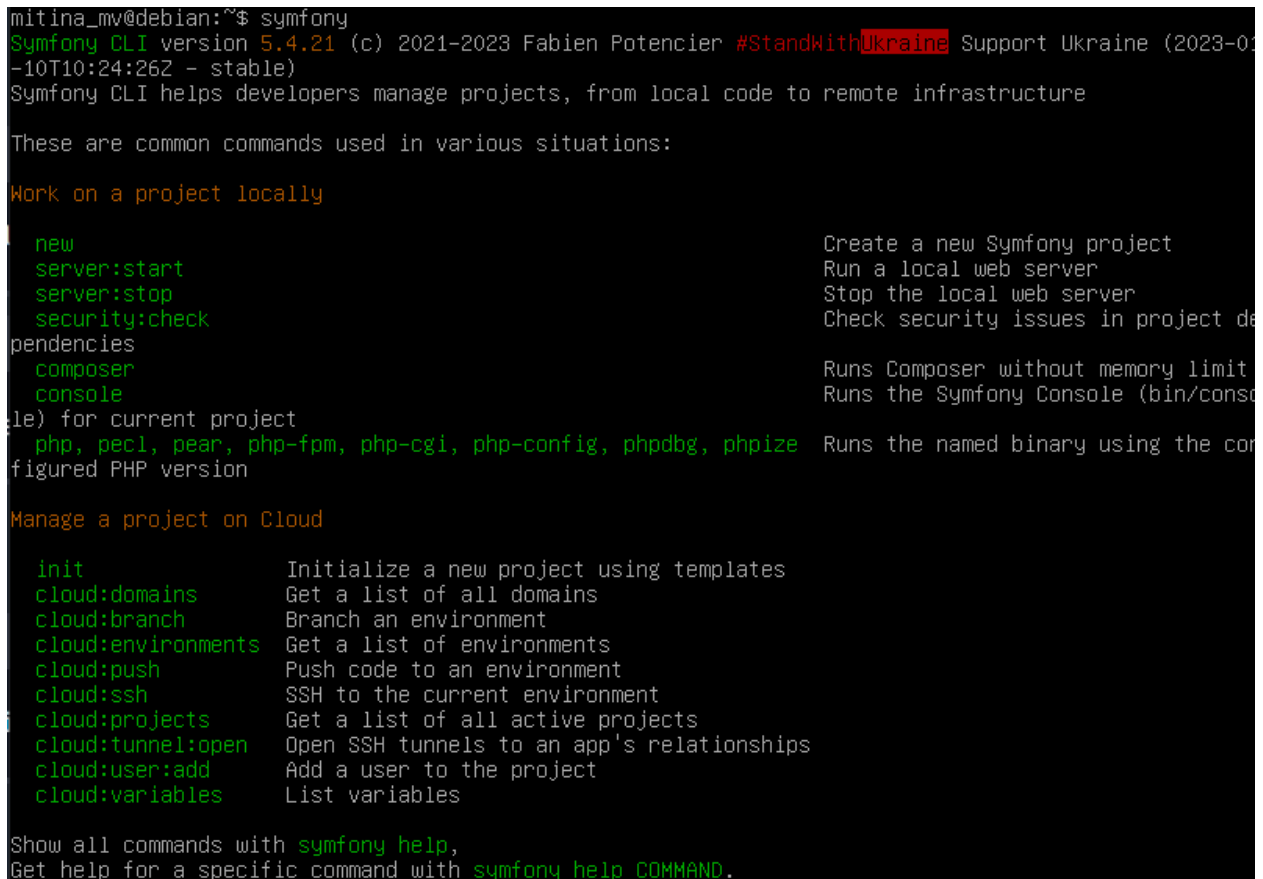
## Установка symfony:

```
curl -1sLf
```

```
'https://dl.cloudsmith.io/public/symfony/stable/setup.deb.sh' |
```

```
sudo -E bash
```

```
sudo apt install symfony-cli
```



```
mitina_mv@debian:~$ symfony
Symfony CLI version 5.4.21 (c) 2021-2023 Fabien Potencier #StandWithUkraine Support Ukraine (2023-03-10T10:24:26Z - stable)
Symfony CLI helps developers manage projects, from local code to remote infrastructure

These are common commands used in various situations:

Work on a project locally

  new                                Create a new Symfony project
  server:start                       Run a local web server
  server:stop                       Stop the local web server
  security:check                    Check security issues in project dependencies
  composer                          Runs Composer without memory limit
  console                            Runs the Symfony Console (bin/console) for current project
  php, pecl, pear, php-fpm, php-cgi, php-config, phpdbg, phpize  Runs the named binary using the configured PHP version

Manage a project on Cloud

  init                               Initialize a new project using templates
  cloud:domains                     Get a list of all domains
  cloud:branch                      Branch an environment
  cloud:environments                Get a list of environments
  cloud:push                        Push code to an environment
  cloud:ssh                         SSH to the current environment
  cloud:projects                   Get a list of all active projects
  cloud:tunnel:open                 Open SSH tunnels to an app's relationships
  cloud:user:add                   Add a user to the project
  cloud:variables                  List variables

Show all commands with symfony help,
Get help for a specific command with symfony help COMMAND.
```

Рисунок 2 – результат установки symfony.

## Установка Composer

Скачиваем файл composer-setup.php с официального сайта первой командой и устанавливаем глобально второй:

```
wget -O composer-setup.php https://getcomposer.org/installer
```

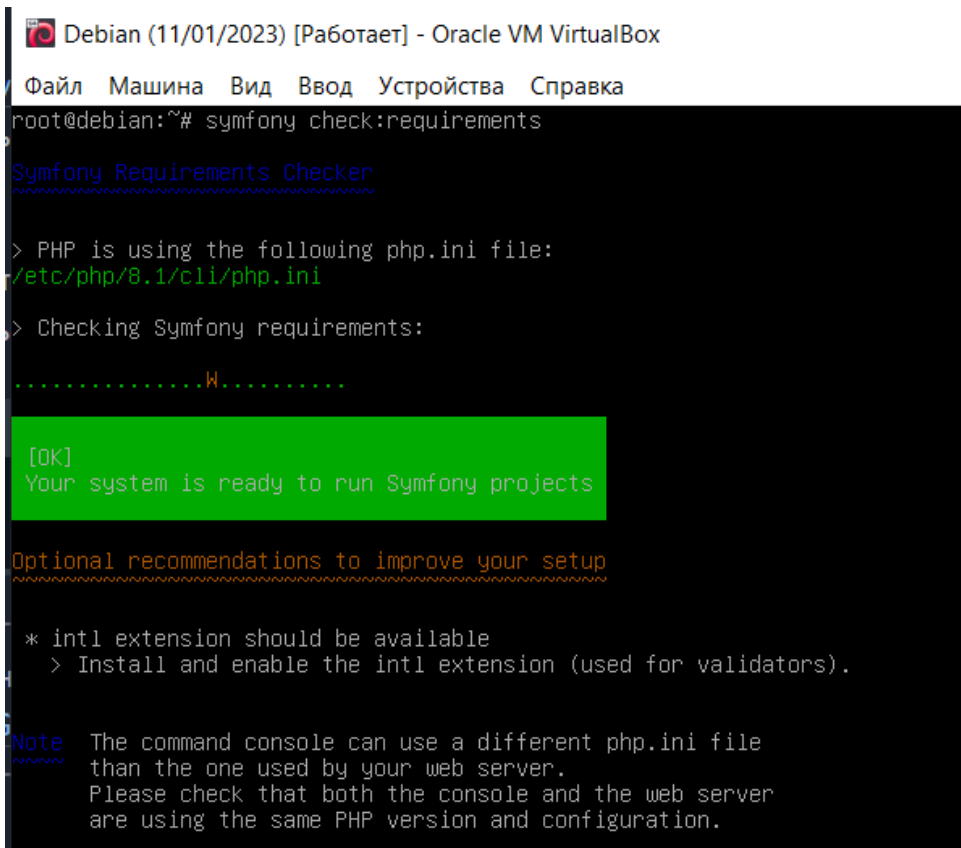
```
php composer-setup.php --install-dir=/usr/local/bin --
```

```
filename=composer
```

## Настройка проекта

### Запуск сервера symfony и проекта demo

Перед установкой проекта запускаем проверку от symfony на соответствие системы для запуска проекта: `symfony check:requirements`

A screenshot of a terminal window titled "Debian (11/01/2023) [Работает] - Oracle VM VirtualBox". The terminal shows the command `symfony check:requirements` being executed. The output includes the "Symfony Requirements Checker" header, the PHP configuration file path (`/etc/php/8.1/cli/php.ini`), and a confirmation that the system is ready to run Symfony projects. A green box highlights the message "[OK] Your system is ready to run Symfony projects". Below this, there are optional recommendations to improve the setup, such as installing the intl extension. A note at the bottom advises checking the PHP version and configuration for both the console and the web server.

```
Debian (11/01/2023) [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
root@debian:~# symfony check:requirements

Symfony Requirements Checker
~~~~~

> PHP is using the following php.ini file:
/etc/php/8.1/cli/php.ini

> Checking Symfony requirements:

.....W.....

[OK]
Your system is ready to run Symfony projects

Optional recommendations to improve your setup
~~~~~

* intl extension should be available
  > Install and enable the intl extension (used for validators).

Note: The command console can use a different php.ini file
      than the one used by your web server.
      Please check that both the console and the web server
      are using the same PHP version and configuration.
```

Рисунок 3 – Проверка на соответствие – система готова к запуску проекта на symfony.

Скачиваем проект с официального репозитория <https://github.com/symfony/demo> - он загрузится в папку с названием demo. Необходимо установить git и выполнить:

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

После того, как все было установлено, переходим в папку с проектом и запускаем командой: `symfony serve` или `symfony server:start`.

```
mitina_mv@debian:~/15-2$ cd demo/
mitina_mv@debian:~/15-2/demo$ symfony serve

[WARNING] run "symfony server:ca:install" first if you want to run the web server with TLS support, or use "--p12" or "--no-tls" to avoid this warning

Following Web Server log file (/home/mitina_mv/.symfony5/log/7368a69efa046e25d59e369be53acfa18432512c.log)
Following PHP log file (/home/mitina_mv/.symfony5/log/7368a69efa046e25d59e369be53acfa18432512c/7daf403c7589f4927632ed3b6af762a992f09b78.log)
WARNING the current dir requires PHP 8.1.0 (composer.json from current dir: /home/mitina_mv/15-2/demo/composer.json), but this version is not available

[WARNING] The local web server is optimized for local development and MUST never be used in a production setup.

[OK] Web server listening
The Web server is using PHP CLI 8.1.14
http://127.0.0.1:8000

[Web Server ] Jan 11 22:37:36 |DEBUG | PHP Reloading PHP versions
[Web Server ] Jan 11 22:37:36 |WARN | PHP the current dir requires PHP 8.1.0 (composer.json from current dir: /home/mitina_mv/15-2/demo/composer.json), but this version is not available
[Web Server ] Jan 11 22:37:36 |DEBUG | PHP Using PHP version 8.1.14 (from default version in $PATH)
[PHP ] [Wed Jan 11 22:37:36 2023] PHP 8.1.14 Development Server (http://127.0.0.1:37427) started
[Web Server ] Jan 11 22:37:36 |INFO | PHP listening path="/usr/bin/php8.1" php="8.1.14" port=37427
```

Рисунок 4 – успешный запуск проекта.

Чтобы открыть страницу в браузере, необходимо пробросить порты из виртуальной машины на хостовую. Открываем настройки виртуальной машины, переходим на вкладку Сеть. Выбираю сеть «NAT», после чего кликаем по «Дополнительно» и настраиваем пути:

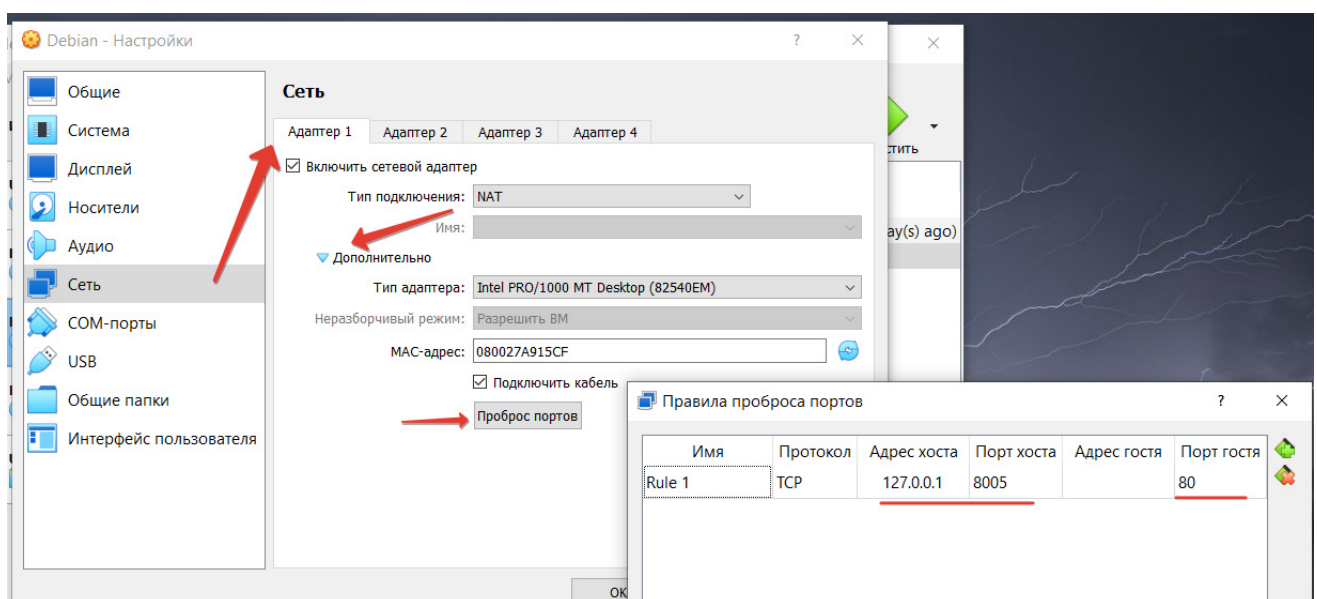


Рисунок 5 – проброс портов с ВМ на хостовую.

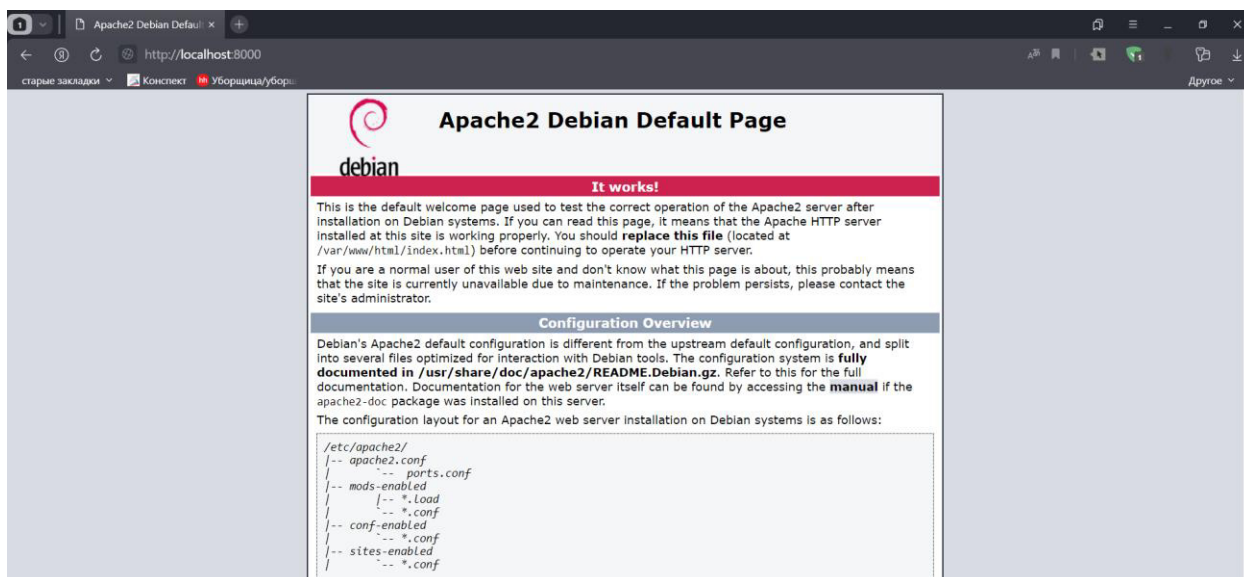


Рисунок 6 – порт 80 виртуальной машины.

Меняем порт 80 на 8000, чтобы увидеть страницу приложения symfony. После того, как порты были проброшены, я получила ответ на странице – ошибка 500, система посоветовала установить intl, что я и сделала на рисунке \_\_, однако все пакеты установлены. Я пыталась пофиксить ошибку 500 несколько дней, но все было бесполезно. Если знаете, как это исправить, напишите мне в telegram @mitina\_mv – буду ждать советов.

Сначала страница в браузере вообще не отображалась, но логи говорили о том, что на нее осуществляются запросы, которые сервер сбрасывает при 500 ошибке. Я установила очень много всяких пакетов, фиксила runtime ошибки при скачивании пакетов symfony через установку в composer.json в разделе config отключения timeout, но пакеты были установлены, однако починить сервер это не помогало.



```
Debian (11/01/2023) [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка

[PHP] ] [Sat Jan 14 21:42:55 2023] 127.0.0.1:53016 Accepted
[PHP] ] [Sat Jan 14 21:42:55 2023] 127.0.0.1:53016 [500]: GET /ru
[PHP] ] [Sat Jan 14 21:42:55 2023] 127.0.0.1:53016 Closing
[Application] Jan 14 21:49:56 |INFO| REQUES Matched route "homepage". method="GET" request_uri="http://127.0.0.1:8005/ru" route="homepage" route_parameters={"_controller":"Symfony\\Bundle\\FrameworkBundle\\Controller\\TemplateController::templateAction", "_locale":"ru", "_route":"homepage", "template":"default/homepage.html.twig"}
[Application] Jan 14 21:49:56 |DEBUG| PHP User Notice: Install the curl extension or run "composer require amphp/http-client:^4.2.1" to perform async HTTP operations, including full HTTP/2 support
[Application] Jan 14 21:49:56 |CRITICAL| REQUES Uncaught PHP Exception Symfony\\Polyfill\\Intl\\Icu\\Exception\\MethodNotImplementedException: "The Symfony\\Polyfill\\Intl\\Icu\\Locale::setDefault() is not implemented. Please install the "intl" extension for full localization capabilities." at /home/mitina_mv/15-2/demo/vendor/symfony/polyfill-intl-icu/Locale.php line 305
[Application] Jan 14 21:49:56 |CRITICAL| REQUES Exception thrown when handling an exception (Symfony\\Polyfill\\Intl\\Icu\\Exception\\MethodNotImplementedException: The Symfony\\Polyfill\\Intl\\Icu\\Locale::setDefault() is not implemented. Please install the "intl" extension for full localization capabilities. at /home/mitina_mv/15-2/demo/vendor/symfony/polyfill-intl-icu/Locale.php line 305)
[Application] Jan 14 21:49:56 |CRITICAL| PHP Uncaught Exception: The Symfony\\Polyfill\\Intl\\Icu\\Locale::setDefault() is not implemented. Please install the "intl" extension for full localization capabilities.
[Application] Jan 14 21:49:56 |CRITICAL| REQUES Uncaught PHP Exception Symfony\\Polyfill\\Intl\\Icu\\Exception\\MethodNotImplementedException: "The Symfony\\Polyfill\\Intl\\Icu\\Locale::setDefault() is not implemented. Please install the "intl" extension for full localization capabilities." at /home/mitina_mv/15-2/demo/vendor/symfony/polyfill-intl-icu/Locale.php line 305
[Application] Jan 14 21:49:56 |CRITICAL| REQUES Exception thrown when handling an exception (Symfony\\Polyfill\\Intl\\Icu\\Exception\\MethodNotImplementedException: The Symfony\\Polyfill\\Intl\\Icu\\Locale::setDefault() is not implemented. Please install the "intl" extension for full localization capabilities. at /home/mitina_mv/15-2/demo/vendor/symfony/polyfill-intl-icu/Locale.php line 305)
[Application] Jan 14 21:49:56 |CRITICAL| PHP Uncaught Exception: The Symfony\\Polyfill\\Intl\\Icu\\Locale::setDefault() is not implemented. Please install the "intl" extension for full localization capabilities.
[Web Server] ] Jan 14 21:49:56 |ERROR| SERVER GET (500) /ru ip="10.0.2.2"
[PHP] ] [Sat Jan 14 21:49:56 2023] 127.0.0.1:50294 Accepted
[PHP] ] [Sat Jan 14 21:49:56 2023] 127.0.0.1:50294 [500]: GET /ru
[PHP] ] [Sat Jan 14 21:49:56 2023] 127.0.0.1:50294 Closing
```

Рисунок 7 – логи сервера

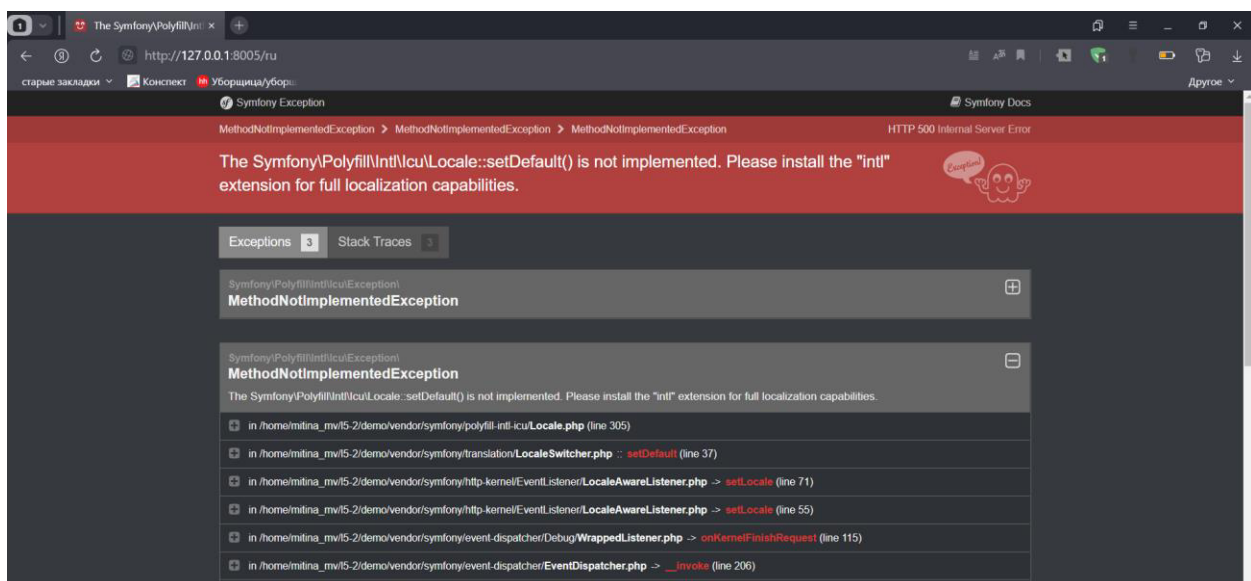


Рисунок 8 – страница symfony

```

mitina_mv@debian:~/15-2/demo$ composer require symfony/intl
./composer.json has been updated
Running composer update symfony/intl
Loading composer repositories with package information
Info from https://repo.packagist.org: #StandWithUkraine
Updating dependencies
Nothing to modify in lock file
Writing lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating autoload files
31 packages you are using are looking for funding.
Use the `composer fund` command to find out more!

Run composer recipes at any time to see the status of your Symfony recipes.

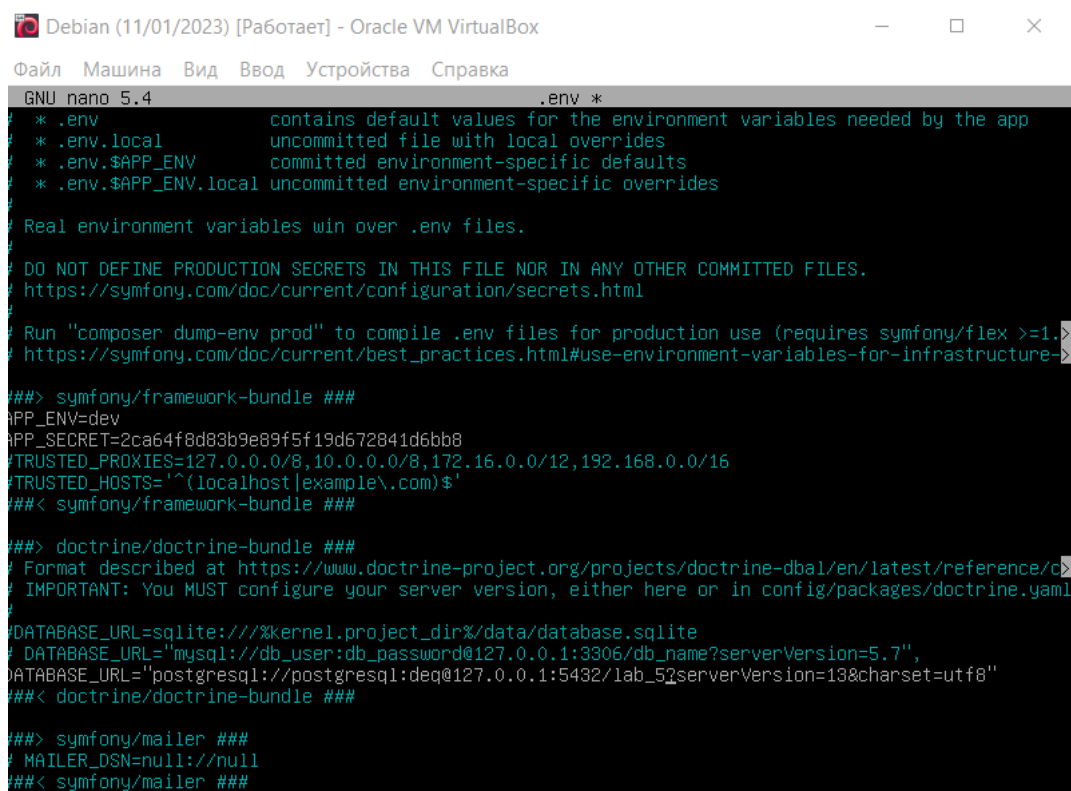
https://repo.packagist.org could not be fully loaded (The "https://repo.packagist.org/p2/symfony/mime.json" file could not be downloaded: php_network_getaddresses: getaddrinfo for repo.packagist.org failed: Temporary failure in name resolution
Failed to open stream: php_network_getaddresses: getaddrinfo for repo.packagist.org failed: Temporary failure in name resolution), package information was loaded from the local cache and may be out of date
No security vulnerability advisories found

```

Рисунок 9 – установка пакета symfony/intl

## Настройка docker, сборка проекта

Меняем файл .env переменную DATABASE\_URL, чтобы подключить базу данных на postgres. Нам нужно указать пользователя:пароль и название БД.



```

Debian (11/01/2023) [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
GNU nano 5.4 .env *
# * .env contains default values for the environment variables needed by the app
# * .env.local uncommitted file with local overrides
# * .env.$APP_ENV committed environment-specific defaults
# * .env.$APP_ENV.local uncommitted environment-specific overrides
#
# Real environment variables win over .env files.
#
# DO NOT DEFINE PRODUCTION SECRETS IN THIS FILE NOR IN ANY OTHER COMMITTED FILES.
# https://symfony.com/doc/current/configuration/secrets.html
#
# Run "composer dump-env prod" to compile .env files for production use (requires symfony/flex >=1.
# https://symfony.com/doc/current/best_practices.html#use-environment-variables-for-infrastructure-
##> symfony/framework-bundle ###
APP_ENV=dev
APP_SECRET=2ca64f8d83b9e89f5f19d672841d6bb8
#TRUSTED_PROXIES=127.0.0.0/8,10.0.0.0/8,172.16.0.0/12,192.168.0.0/16
#TRUSTED_HOSTS='^(localhost|example\.com)$'
##< symfony/framework-bundle ###

##> doctrine/doctrine-bundle ###
# Format described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/c
# IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml
#
DATABASE_URL=sqlite://%kernel.project_dir%/data/database.sqlite
# DATABASE_URL="mysql://db_user:db_password@127.0.0.1:3306/db_name?serverVersion=5.7",
DATABASE_URL="postgresql://postgres:lab_5@127.0.0.1:5432/lab_5?serverVersion=13&charset=utf8"
##< doctrine/doctrine-bundle ###

##> symfony/mailer ###
# MAILER_DSN=null://null
##< symfony/mailer ###

```

Рисунок 10 – Переключаем бд с sqlite на postgres

По умолчанию пароль для пользователя postgres не установлен, поэтому нам нужно переключиться на пользователя postgres и через утилиту psql установить пароль для пользователя:

```
sudo -i -u postgres psql
```

```
\password
```

Теперь здесь же, в psql, создадим базу данных:

```
mitina_mv@debian:~/15-2/demo$ sudo -i -u postgres
[sudo] пароль для mitina_mv:
postgres@debian:~$ createdb lab_5
postgres@debian:~$ psql
psql (13.9 (Debian 13.9-0+deb11u1))
Введите "help", чтобы получить справку.

postgres=# \l

```

Список баз данных					
Имя	Владелец	Кодировка	LC_COLLATE	LC_CTYPE	Права доступа
lab_5	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8	
postgres	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8	
template0	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8	=c/postgres +
template1	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8	postgres=CtC/postgres +
(4 строки)					
postgres=#					

### Создание базы данных

Для работы doctrine тоже нужно установить пакеты через composer. Они просто громадные, поэтому, если интернет медленный, нужно отключить временное ограничение, чтобы composer не подумал, что процесс умер и не сбросил загрузку. Когда пакеты установятся, можно запускать:

```
mitina_mv@debian:~/15-2/demo$ php bin/console doctrine:schema:create
!
! [CAUTION] This operation should not be executed in a production environment!
!

Creating database schema...

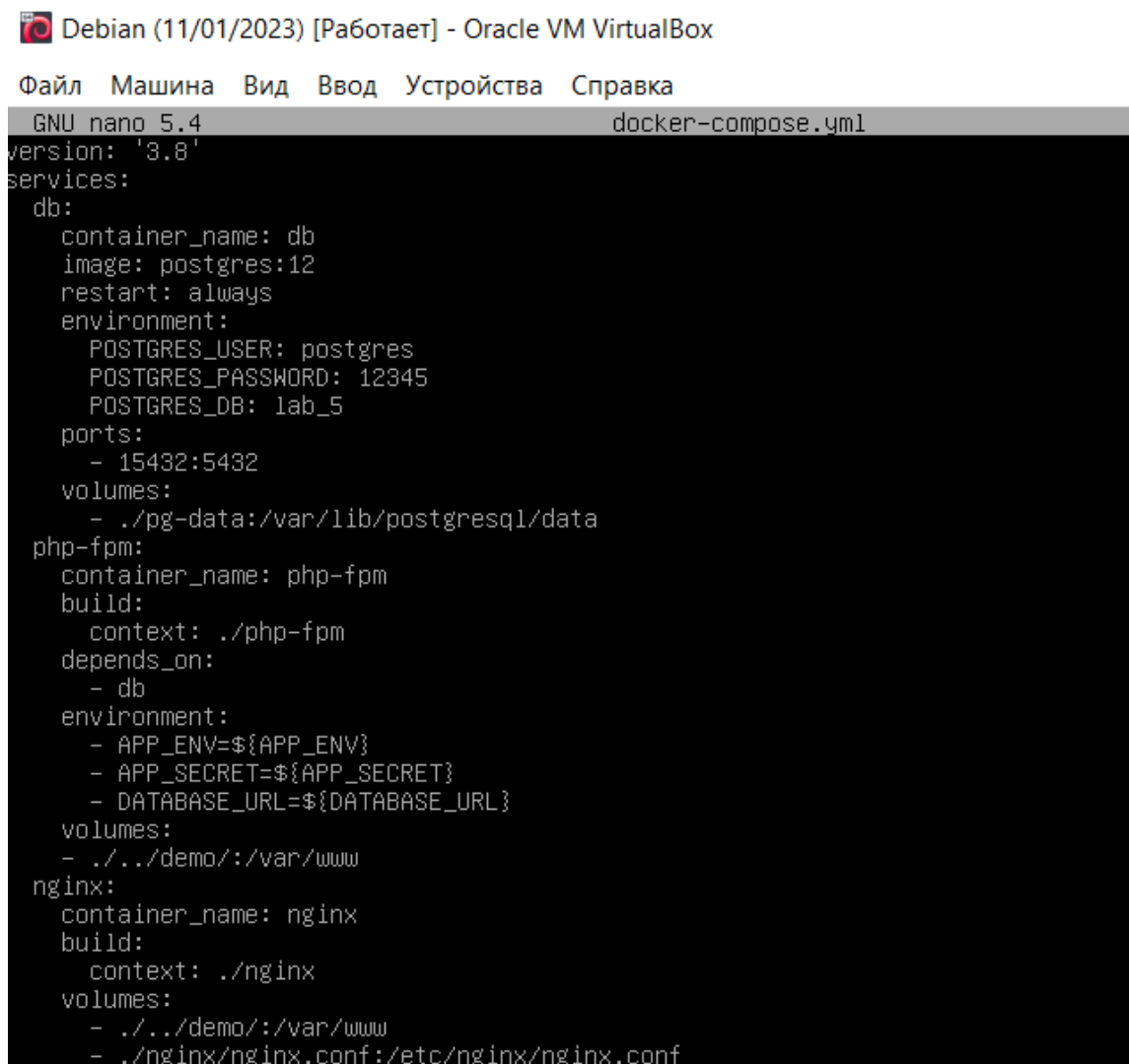
[OK] Database schema created successfully!

mitina_mv@debian:~/15-2/demo$ php bin/console doctrine:fixtures:load
Careful, database "lab_5" will be purged. Do you want to continue? (yes/no) [no]:
> Y
> purging database
> loading App\DataFixtures\AppFixtures
```

Скажу еще, что через doctrine можно было создать и саму базу данных, как мы сделали это через psql, но задания такого не было )

Пока все, что мы делали, изменяло только проект demo на symfony, поэтому проект должен был бы запуститься и показать начальную страницу без изменений. Но так как у меня сервер лежал с 500 ошибкой, проверить у меня не было возможности.

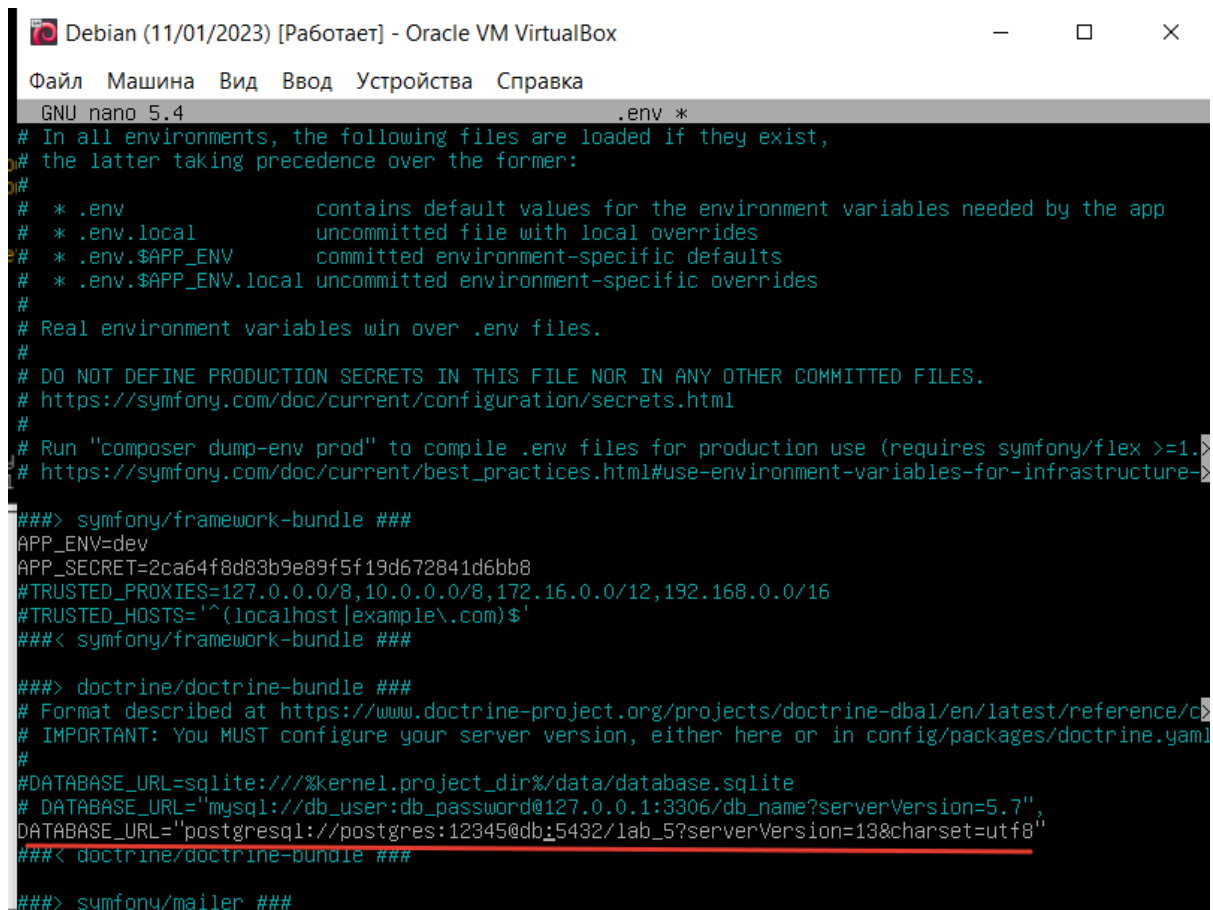
Теперь мы собираем проект под docker, прописываем файл docker-compose.yml:



```
Debian (11/01/2023) [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
GNU nano 5.4                                docker-compose.yml
version: '3.8'
services:
  db:
    container_name: db
    image: postgres:12
    restart: always
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: 12345
      POSTGRES_DB: lab_5
    ports:
      - 15432:5432
    volumes:
      - ./pg-data:/var/lib/postgresql/data
  php-fpm:
    container_name: php-fpm
    build:
      context: ./php-fpm
    depends_on:
      - db
    environment:
      - APP_ENV=${APP_ENV}
      - APP_SECRET=${APP_SECRET}
      - DATABASE_URL=${DATABASE_URL}
    volumes:
      - ../../demo:/var/www
  nginx:
    container_name: nginx
    build:
      context: ./nginx
    volumes:
      - ../../demo:/var/www
      - ./nginx/nginx.conf:/etc/nginx/nginx.conf
```

Рисунок 11 – docker-compose

В наш проект копируем .env из проекта /demo, меняем 127.0.0.1 на название нашего контейнера:



```
Debian (11/01/2023) [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
GNU nano 5.4 .env *
# In all environments, the following files are loaded if they exist,
# the latter taking precedence over the former:
#
# * .env contains default values for the environment variables needed by the app
# * .env.local uncommitted file with local overrides
# * .env.$APP_ENV committed environment-specific defaults
# * .env.$APP_ENV.local uncommitted environment-specific overrides
#
# Real environment variables win over .env files.
#
# DO NOT DEFINE PRODUCTION SECRETS IN THIS FILE NOR IN ANY OTHER COMMITTED FILES.
# https://symfony.com/doc/current/configuration/secrets.html
#
# Run "composer dump-env prod" to compile .env files for production use (requires symfony/flex >=1.
# https://symfony.com/doc/current/best_practices.html#use-environment-variables-for-infrastructure-

###> symfony/framework-bundle ###
APP_ENV=dev
APP_SECRET=2ca64f8d83b9e89f5f19d672841d6bb8
#TRUSTED_PROXIES=127.0.0.0/8,10.0.0.0/8,172.16.0.0/12,192.168.0.0/16
#TRUSTED_HOSTS='^(localhost|example\.com)$'
###< symfony/framework-bundle ###

###> doctrine/doctrine-bundle ###
# Format described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/c
# IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml
#
#DATABASE_URL=sqlite://%%kernel.project_dir%/data/database.sqlite
# DATABASE_URL="mysql://db_user:db_password@127.0.0.1:3306/db_name?serverVersion=5.7",
DATABASE_URL="postgresql://postgres:12345@db:5432/lab_5?serverVersion=13&charset=utf8"
###< doctrine/doctrine-bundle ###

###> symfony/mailer ###
```

.env для нашего проекта на docker.

Чтобы удобнее было писать код, я подключила общую папку для виртуальной машины, где создала необходимые Dockerfile и копировала их в свой проект docker.

```
1 FROM php:8.1-fpm
2
3 RUN apt-get update && \
4     apt-get install -y --no-install-recommends libssl-dev zlib1g-dev && \
5     curl git unzip netcat libxml2-dev libpq-dev libzip-dev && \
6     pecl install apcu && \
7     docker-php-ext-configure pgsql -with-pgsql=/usr/local/pgsql && \
8     docker-php-ext-install -j$(nproc) zip opcache intl pdo_pgsql pgsql && \
9     docker-php-ext-enable apcu pdo_pgsql sodium && \
10    apt-get clean && rm -rf /var/lib/apt/lists/* /tmp/* /var tmp/*
11 COPY --from=composer /usr/bin/composer /usr/bin/composer
12 WORKDIR /var/www
13 CMD composer i -o ; php-fpm
14 EXPOSE 9000
```

Рисунок 12 – /php-fpm/Dockerfile

```

mitina_mv@debian:~/15-2/docker1$ docker compose build
[+] Building 27.6s (6/13)
=> [docker1-nginx internal] load build definition from Dockerfile      0.2s
=> => transferring dockerfile: 102B                                   0.0s
=> [docker1-nginx internal] load .dockerignore                        0.2s
=> => transferring context: 2B                                         0.0s
=> [docker1-php-fpm internal] load build definition from Dockerfile   0.2s
=> => transferring dockerfile: 637B                                    0.0s
=> [docker1-php-fpm internal] load .dockerignore                      0.2s
=> => transferring context: 2B                                         0.0s
=> [docker1-nginx internal] load metadata for docker.io/library/nginx:alpine 12.8s
=> [docker1-php-fpm internal] load metadata for docker.io/library/php:8.1-fpm 9.3s
=> [docker1-php-fpm stage-0 1/4] FROM docker.io/library/php:8.1-fpm@sha256:bdac671e3ca663c0e9dc7008f28b3ee8dc2cc5b2d 17.9s
=> => resolve docker.io/library/php:8.1-fpm@sha256:bdac671e3ca663c0e9dc7008f28b3ee8dc2cc5b2d 0.0s
=> => sha256:bdac671e3ca663c0e9dc7008f28b3ee8dc2cc5b2d11fcde83d2142ab0369f45 1.86kB / 1.86kB 0.0s
=> => sha256:b9e4e67f8b13639b015c0bd0406aae414476416f603a741f0968944106703 11.71kB / 11.71kB 0.0s
=> => sha256:1873be8582646883d25d54eaa9b9afa7d778d378e04063e1f616cbdec129b9f4 224B / 224B 2.8s
=> => sha256:7ce6a163d8c1d0c95101cdc4fa31cc6ef5846353b802f3885bc2756afb42d7f8 0B / 91.63MB 17.9s
=> => sha256:92b09bb98bb9ad33d5ea72942b98924117ac190f51e6257754263be91fd5050 2.41kB / 2.41kB 0.0s
=> => sha256:8740c948ffd4c816ea7ca963f99ca52f4788baa23f228da9581a9ea2edd3f 1.05MB / 31.40MB 17.9s
=> => sha256:008a172010ba0e6c008f04b19dec142a3b41ef2bcbd59901c88a2a356f61b766 270B / 270B 4.6s
=> => sha256:2cb716163fb0abb23e557d3f38d58da0b45d4ebdc1dc89c7ac2d48f7312ac92d 0B / 12.07MB 17.9s
=> [docker1-php-fpm] FROM docker.io/library/composer:latest          17.9s
=> => resolve docker.io/library/composer:latest                       17.9s
=> [docker1-nginx 1/2] FROM docker.io/library/nginx:alpine@sha256:659610aadb34b7967dea76869 14.5s
=> => resolve docker.io/library/nginx:alpine@sha256:659610aadb34b7967dea7686926f6dcf08d588a71 0.0s
=> => sha256:659610aadb34b7967dea7686926f6dcf08d588a71c5121edb094ce0e4cdbc45e 1.65kB / 1.65kB 0.0s
=> => sha256:c1b9fe3c0c015486cf1e4a0ecabe78d05864475e279638e9713eb55f013f907 1.78kB / 1.78kB 0.0s
=> => sha256:c433c51bbd66153269da1c592105c9c19bf353e9d7c3d1225ae2bbb888c 16.35kB / 16.35kB 0.0s

```

Рисунок 13 – сборка проекта командой docker compose build

```

solve host: libzip-dev
-----
failed to solve: executor failed running [/bin/sh -c apt-get update && apt-get install -y --no-i
install-recommends libssl-dev zlib1g-dev && curl git unzip netcat libxml2-dev libpq-dev libzip-de
v && pecl install apcu && docker-php-ext-configure pgsql -with-pgsql=/usr/local/pgsql &&
docker-php-ext-install -j$(nproc) zip opcache intl pdo_pgsql pgsql && docker-php-ext-enable apc
u pdo_pgsql sodium && apt-get clean && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*]: exit code
: 6
mitina_mv@debian:~/15-2/docker1$

```

Рисунок 14 – ошибка сборки с exit code 6

Предполагаю, что эта ошибка фиксируется через изменение строки 5 на:

`apt install -y git unzip netcat libxml2-dev libpq-dev libzip-dev`

Кстати, везде, где я искала способы фикса этой ошибки, было написано по сути «я не знаю, почему она возникает, но попробуйте удалить docker». Я предполагаю, что ошибка с кодом 6 возникает из-за curl, так как эта утилита стучится на адрес, а мы в 5 строчке адрес не указываем – сама утилита ругается с ошибкой 6, и докер потом тоже показывает ошибку 6. Итак, после того, как мы поменяли строчку 5, вновь собираем проект. По свойству docker, предыдущие слои будут подгружены из кеша и заново собирать их не нужно:

```
Debian (11/01/2023) [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
#0 99.30 checking for suffix of executables...
#0 99.33 checking whether we are cross compiling... no
#0 99.37 checking for suffix of object files... o
#0 99.40 checking whether we are using the GNU C compiler... yes
#0 99.41 checking whether cc accepts -g... yes
#0 99.44 checking for cc option to accept ISO C89... none needed
#0 99.47 checking how to run the C preprocessor... cc -E
#0 99.52 checking for icc... no
#0 99.53 checking for suncc... no
#0 99.54 checking for system library directory... lib
#0 99.54 checking if compiler supports -Wl,-rpath,... yes
#0 99.59 checking build system type... x86_64-pc-linux-gnu
#0 99.60 checking host system type... x86_64-pc-linux-gnu
#0 99.60 checking target system type... x86_64-pc-linux-gnu
#0 99.63 checking for PHP prefix... /usr/local
#0 99.64 checking for PHP includes... -I/usr/local/include/php -I/usr/local/include/php/main -I/usr/
local/include/php/TSRM -I/usr/local/include/php/Zend -I/usr/local/include/php/ext -I/usr/local/inclu
de/php/ext/date/lib
#0 99.64 checking for PHP extension directory... /usr/local/lib/php/extensions/no-debug-non-zts-2021
0902
#0 99.64 checking for PHP installed headers prefix... /usr/local/include/php
#0 99.64 checking if debug is enabled... no
#0 99.65 checking if zts is enabled... no
#0 99.67 checking for gawk... no
#0 99.67 checking for nawk... nawk
#0 99.67 checking if nawk is broken... no
#0 99.68 checking for PostgreSQL support... yes, shared
#0 99.68 checking for pg_config... not found
#0 99.68 configure: error: Cannot find libpq-fe.h. Please specify correct PostgreSQL installation pa
th
-----
failed to solve: executor failed running [/bin/sh -c apt-get update -y && apt-get install -y --r
p-install-recommends libssl-dev zlib1g-dev && pecl install apcu && docker-php-ext-configure
pgsql -with-pgsql=/usr/local/pgsql && docker-php-ext-install -j$(nproc) zip opcache intl pdo_pgs
ql pgsql && docker-php-ext-enable apcu pdo_pgsql sodium && apt-get clean && rm -rf /var/lib/
apt/lists/* /tmp/* /var/tmp/*]: exit code: 1
```

Рисунок 15 – ошибка новой сборки

```
mitina_mv@debian:/usr/lib/postgresql$ cd /usr/include/postgresql/
mitina_mv@debian:/usr/include/postgresql$ ls
internal  libpq-events.h  pg_config_ext.h  pg_config_manual.h  postgres_ext.h
libpq     libpq-fe.h      pg_config.h      pg_config_os.h
```

Рисунок 16 – поиск указанного файла по расположению.

Указанный пакет– libpq – я устанавливала и вручную, но ошибка не устранялась. Везде написано, что нужно установить эту библиотеку и все полетит, но переустановка postgres и всех пакетов для него, перезагрузка компьютера, добавления текущего пользователя в группу докера и многое другое – ничего не помогало.

Дальше этого пункта я не ушла, к сожалению. Василий Владимирович разрешил попробовать выполнить эту лабораторную на Ubuntu, что я и сделаю во второй части лабораторной.



## Вывод

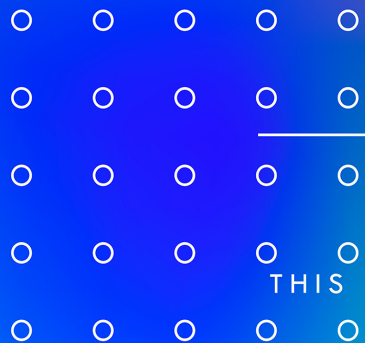
Суммарно в попытках выполнить эту лабораторную работу я ghjdtlf неделю. В ходе выполнения мне довелось исправить массу ошибок, это научило меня очень многому и позволило понять больше linux. Я попробую реализовать задания данной лабораторной работы на Ubuntu, о чем напишу во второй части отчета.

Здесь изображены мои впечатления от выполнения лабораторной ))



Кроме того, я прошла курс по Docker, прилагаю в конце этого отчета сертификат, чтобы подтвердить свои навыки.





# CERTIFICATE OF COMPLETION

THIS CERTIFICATE IS PROUDLY PRESENTED TO

**МАРИЯ МИТИНА**

COMPLETED THE COURSE

Docker для начинающих + практический опыт

28659-1673453405

Саваров Андрей

2023-01-11

ROTORO



CLOUD

