

Липецкий государственный технический университет

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине «Операционная система Linux»

Контейнеризация.

Студент

Титов В. А.

Группа АС-20

Руководитель

Кургасов В. В.

к.п.н.

Липецк 2022 г.

Цель работы

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

Задание кафедры

С помощью Docker Compose на своем компьютере поднять сборку nginx+php-fpm+postgres, продемонстрировать ее работоспособность, запустив внутри контейнера демо-проект на symfony.

Ход работы

1. Установка Docker

Установим Docker посредством следующих команд:

1) Обновим пакеты системы при помощи команды apt-get update:

```
root@debian:/home/vlad# apt-get update
Пол:1 http://security.debian.org/debian-security bullseye-security InRelease [48,4 kB]
Пол:2 http://deb.debian.org/debian bullseye InRelease [116 kB]
Пол:3 http://deb.debian.org/debian bullseye-updates InRelease [44,1 kB]
Пол:4 http://deb.debian.org/debian bullseye/main Sources [8 634 kB]
Пол:5 http://security.debian.org/debian-security bullseye-security/main Sources [176 kB]
Пол:6 http://security.debian.org/debian-security bullseye-security/main amd64 Packages [214 kB]
Пол:7 http://security.debian.org/debian-security bullseye-security/main Translation-en [139 kB]
Пол:8 http://deb.debian.org/debian bullseye/main amd64 Packages [8 183 kB]
Пол:9 http://deb.debian.org/debian bullseye/main Translation-en [6 240 kB]
Получено 23,7 MB за 13с (1 823 kB/s)
Чтение списков пакетов... 99%
```

Рис.1. Обновление пакетов системы

2) Установка пакетов для использования репозитория через HTTPS-протокол

```
root@debian:/home/vlad# apt-get install ca-certificates curl gnupg lsb-release
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Уже установлен пакет ca-certificates самой новой версии (20210119).
Уже установлен пакет curl самой новой версии (7.74.0-1.3+deb11u3).
Уже установлен пакет gnupg самой новой версии (2.2.27-2+deb11u2).
gnupg помечен как установленный вручную.
Уже установлен пакет lsb-release самой новой версии (11.1.0).
lsb-release помечен как установленный вручную.
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 12 пак
бновлено.
root@debian:/home/vlad# _
```

Рис.2. Установка нужных пакетов

3) Добавление официального GPG-ключа

```
root@debian:/home/vlad# curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o /
usr/share/keyrings/docker-archive-keyring.gpg
```

Рис.3. Добавление GPG-ключа

4) Добавление репозитория Докера в sources.list, для его установки

```
root@debian:/etc/apt/sources.list.d# echo "deb [arch=$(dpkg --print-architecture)
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/debian $(lsb_release -cs) stable" | tee /etc/apt/sources.list.d/do
cker.list > /dev/null
```

Рис.4. Добавление репозитория Докера в sources.list

5) Установка Docker

```
root@debian:/etc/apt/sources.list.d# apt-get update
Сущ:1 http://security.debian.org/debian-security bullseye-security InRelease
Сущ:2 http://deb.debian.org/debian bullseye InRelease
Сущ:3 http://deb.debian.org/debian bullseye-updates InRelease
Пол:4 https://download.docker.com/linux/debian bullseye InRelease [43,3 kB]
Пол:5 https://download.docker.com/linux/debian bullseye/stable amd64 Packages [16,6 kB]
Получено 59,9 kB за 1с (43,3 kB/s)
Чтение списков пакетов... Готово
```

Рис.5. Обновление пакетов

```
root@debian:/etc/apt/sources.list.d# apt-get install docker-ce docker-ce-cli containerd.io
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
E: Невозможно найти пакет docker-ce-cli
root@debian:/etc/apt/sources.list.d# apt-get install docker-ce docker-ce-cli containerd.io
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  dbus-user-session docker-ce-rootless-extras docker-scan-plugin git git-man iptables
  liberror-perl libip6tc2 libltdl7 libnetfilter-contrack3 libnfnetlink0 libslirp0 patch pigz
  slirp4netns
Предлагаемые пакеты:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-el
  git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn firewallld ed diffutils-doc
Следующие НОВЫЕ пакеты будут установлены:
  containerd.io dbus-user-session docker-ce docker-ce-cli docker-ce-rootless-extras
  docker-scan-plugin git git-man iptables liberror-perl libip6tc2 libltdl7 libnetfilter-contrack3
  libnfnetlink0 libslirp0 patch pigz slirp4netns
Обновлено 0 пакетов, установлено 18 новых пакетов, для удаления отмечено 0 пакетов, и 12 пакетов не
обновлено.
Необходимо скачать 110 МБ архивов.
После данной операции объём занятого дискового пространства возрастёт на 425 МБ.
Хотите продолжить? [Д/н] _
```

Рис.6. Установка Docker

2. Установка Docker-Compose

Самым быстрым способом установить Docker-Compose является установка через pip.

1) Установим pip

```
root@debian:/# apt install pip
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Заметьте, вместо «pip» выбирается «python3-pip»
Будут установлены следующие дополнительные пакеты:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-10 dpkg-dev fakeroot
  g++ g++-10 gcc gcc-10 javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libatomic1 libbinutils libc-dev-bin libc-devtools libc6-dev
  libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot
  libfile-fcntllock-perl libgcc-10-dev libgomp1 libisl23 libitm1 libjs-jquery libjs-sphinxdoc
  libjs-underscore liblsan0 libmpc3 libmpfr6 libns1-dev libpython3-dev libpython3.9
  libpython3.9-dev libquadmath0 libstdc++-10-dev libtirpc-dev libtsan0 libubsan1 linux-libc-dev
  make manpages-dev python-pip-whl python3-dev python3-distutils python3-lib2to3
  python3-setuptools python3-wheel python3.9-dev zlib1g-dev
Предлагаемые пакеты:
  binutils-doc cpp-doc gcc-10-locales debian-keyring g++-multilib g++-10-multilib gcc-10-doc
  gcc-multilib autoconf automake libtool flex bison gdb gcc-doc gcc-10-multilib glibc-doc bsr
  libstdc++-10-doc make-doc python-setuptools-doc
Следующие НОВЫЕ пакеты будут установлены:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-10 dpkg-dev fakeroot
  g++ g++-10 gcc gcc-10 javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libatomic1 libbinutils libc-dev-bin libc-devtools libc6-dev
  libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot
  libfile-fcntllock-perl libgcc-10-dev libgomp1 libisl23 libitm1 libjs-jquery libjs-sphinxdoc
  libjs-underscore liblsan0 libmpc3 libmpfr6 libns1-dev libpython3-dev libpython3.9
  libpython3.9-dev libquadmath0 libstdc++-10-dev libtirpc-dev libtsan0 libubsan1 linux-libc-dev
  make manpages-dev python-pip-whl python3-dev python3-distutils python3-lib2to3 python3-pip
  python3-setuptools python3-wheel python3.9-dev zlib1g-dev
Обновлено 0 пакетов, установлено 61 новых пакетов, для удаления отмечено 0 пакетов, и 12 пакетов не
обновлено.
Необходимо скачать 74,2 МБ архивов.
После данной операции объем занятого дискового пространства возрастёт на 268 МБ.
Хотите продолжить? [Д/н]
```

Рис.7. Установка pip

2) Установка Docker-Compose через pip

```
root@debian:/# pip install docker-compose
Collecting docker-compose
  Downloading docker-compose-1.29.2-py2.py3-none-any.whl (114 kB)
    |#####| 114 kB 1.0 MB/s
Collecting dockerpty<1,>=0.4.1
  Downloading dockerpty-0.4.1.tar.gz (13 kB)
Requirement already satisfied: requests<3,>=2.20.0 in /usr/lib/python3/dist-packages (from docker-co
mpose) (2.25.1)
Collecting docker[ssh]>=5
  Downloading docker-6.0.1-py3-none-any.whl (147 kB)
    |#####| 147 kB 8.2 MB/s
Collecting texttable<2,>=0.9.0
  Downloading texttable-1.6.7-py2.py3-none-any.whl (10 kB)
Collecting docopt<1,>=0.6.1
  Downloading docopt-0.6.2.tar.gz (25 kB)
Collecting PyYAML<6,>=3.10
  Downloading PyYAML-5.4.1-cp39-cp39-manylinux1_x86_64.whl (630 kB)
    |#####| 630 kB 29.1 MB/s
Collecting python-dotenv<1,>=0.13.0
  Downloading python-dotenv-0.21.0-py3-none-any.whl (18 kB)
Collecting websocket-client<1,>=0.32.0
  Downloading websocket-client-0.59.0-py2.py3-none-any.whl (67 kB)
    |#####| 67 kB 2.1 MB/s
Collecting jsonschema<4,>=2.5.1
  Downloading jsonschema-3.2.0-py2.py3-none-any.whl (56 kB)
    |#####| 56 kB 1.8 MB/s
Collecting distro<2,>=1.5.0
  Downloading distro-1.8.0-py3-none-any.whl (20 kB)
```

Рис.8. Установка Docker-Compose

3. Установка PHP

- Для установки PHP нужно добавить пакет PPA.

1) Установка ПО для работы с пакетом PPA

```
root@debian:/# apt install apt-transport-https lsb-release ca-certificates wget -y
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Уже установлен пакет apt-transport-https самой новой версии (2.2.4).
Уже установлен пакет ca-certificates самой новой версии (20210119).
Уже установлен пакет lsb-release самой новой версии (11.1.0).
Уже установлен пакет wget самой новой версии (1.21-1+deb11u1).
Следующие пакеты устанавливались автоматически и больше не требуются:
  libsodium23 php-common
Для их удаления используйте «apt autoremove».
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 12 пакетов не о
бновлено.
root@debian:/# _
```

Рис.9. Установка ПО

```
root@debian:/# wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg
--2023-01-17 15:35:00-- https://packages.sury.org/php/apt.gpg
Распознаётся packages.sury.org (packages.sury.org)... 93.189.63.147
Подключение к packages.sury.org (packages.sury.org) [93.189.63.147]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: 1769 (1,7K) [application/octet-stream]
Сохранение в: «/etc/apt/trusted.gpg.d/php.gpg»

/etc/apt/trusted.gpg.d/p 100%[=====] 1,73K --.-KB/s за 0s

2023-01-17 15:35:00 (28,4 MB/s) - «/etc/apt/trusted.gpg.d/php.gpg» сохранён [1769/1769]

root@debian:/#

root@debian:/# sh -c 'echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" > /etc/apt/s
ources.list.d/php.list'
root@debian:/#
```

Рис.10-11. Добавление PPA в качестве источника пакетов, который содержит PHP

```
root@debian:/# apt update
Сущ:1 http://deb.debian.org/debian bullseye InRelease
Пол:2 https://packages.sury.org/php bullseye InRelease [6 841 B]
Сущ:3 https://download.docker.com/linux/debian bullseye InRelease
Сущ:4 http://deb.debian.org/debian bullseye-updates InRelease
Сущ:5 http://security.debian.org/debian-security bullseye-security InRelease
Пол:6 https://packages.sury.org/php bullseye/main amd64 Packages [375 kB]
Игн:7 http://ppa.launchpad.net/ondrej/php/ubuntu lunar InRelease
Ошб:8 http://ppa.launchpad.net/ondrej/php/ubuntu lunar Release
  404 Not Found [IP: 185.125.190.52 80]
Чтение списков пакетов... Готово
```

Рис.12. Обновление пакетов

- Установка PHP с библиотеками

1) Установка PHP восьмой версии

```

root@debian:/# apt install php8.2
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  libapache2-mod-php8.2 libpcre2-8-0 php8.2-cli php8.2-opcache php8.2-readline
Предлагаемые пакеты:
  php-pear
Следующие НОВЫЕ пакеты будут установлены:
  libapache2-mod-php8.2 php8.2 php8.2-cli php8.2-opcache php8.2-readline
Следующие пакеты будут обновлены:
  libpcre2-8-0
Обновлено 1 пакетов, установлено 5 новых пакетов, для удаления отмечено 0 пакетов, и 16 пакетов не о
бновлено.
Необходимо скачать 3 987 кВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 12,5 МВ.
Хотите продолжить? [Д/Н]

```

Рис.13. Установка PHP8.2

2) Установка sqlite3

```

root@debian:/# apt-get install php-sqlite3
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Следующий пакет устанавливался автоматически и больше не требуется:
  libsodium23
Для его удаления используйте «apt autoremove».
Будут установлены следующие дополнительные пакеты:
  php8.2-common php8.2-sqlite3
Следующие НОВЫЕ пакеты будут установлены:
  php-sqlite3 php8.2-common php8.2-sqlite3
Обновлено 0 пакетов, установлено 3 новых пакетов, для удаления отмечено 0 пакетов, и 17 пакетов не о
бновлено.
Необходимо скачать 699 кВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 8 779 кВ.
Хотите продолжить? [Д/Н]

```

Рис.14. Установка sqlite3

3) Установка xml

```

root@debian:/# apt-get install php8.2-xml
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Следующие НОВЫЕ пакеты будут установлены:
  php8.2-xml
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 16 пакетов не о
бновлено.
Необходимо скачать 107 кВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 476 кВ.
Пол:1 https://packages.sury.org/php bullseye/main amd64 php8.2-xml amd64 8.2.1-2+0~20230113.12+debia
n11~1.gbpa67f25 [107 кВ]
Получено 107 кВ за 1с (97,9 кВ/с)
Выбор ранее не выбранного пакета php8.2-xml.
(Чтение базы данных ... на данный момент установлено 48262 файла и каталога.)
Подготовка к распаковке .../php8.2-xml_8.2.1-2+0~20230113.12+debian11~1.gbpa67f25_amd64.deb ...
Распаковывается php8.2-xml (8.2.1-2+0~20230113.12+debian11~1.gbpa67f25) ...
Настраивается пакет php8.2-xml (8.2.1-2+0~20230113.12+debian11~1.gbpa67f25) ...

```

Рис.15. Установка xml

4) Установка mbstring

```
root@debian:/# apt-get install php8.2-mbstring
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  libonig5
Следующие НОВЫЕ пакеты будут установлены:
  libonig5 php8.2-mbstring
Обновлено 0 пакетов, установлено 2 новых пакетов, для удаления отмечено 0 пакетов, и 16 пакетов не о
бновлено.
Необходимо скачать 624 кВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 1 883 кВ.
Хотите продолжить? [Д/н] _
```

Рис.16. Установка mbstring

5) Установка pgsql

```
root@debian:/# apt-get install php8.2-pgsql
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  libpq5
Следующие НОВЫЕ пакеты будут установлены:
  libpq5 php8.2-pgsql
Обновлено 0 пакетов, установлено 2 новых пакетов, для удаления отмечено 0 пакетов, и 16 пакетов не о
бновлено.
Необходимо скачать 238 кВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 1 034 кВ.
Хотите продолжить? [Д/н] _
```

Рис.17. Установка pgsql

6) Установка postgresql

```
root@debian:/# apt install postgresql postgresql-contrib
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  libl1vm11 libsensors-config libsensors5 libz3-4 postgresql-13 postgresql-client-13
  postgresql-client-common postgresql-common sysstat
Предлагаемые пакеты:
  lm-sensors postgresql-doc postgresql-doc-13 libjson-perl isag
Следующие НОВЫЕ пакеты будут установлены:
  libl1vm11 libsensors-config libsensors5 libz3-4 postgresql postgresql-13 postgresql-client-13
  postgresql-client-common postgresql-common postgresql-contrib sysstat
Обновлено 0 пакетов, установлено 11 новых пакетов, для удаления отмечено 0 пакетов, и 16 пакетов не
обновлено.
Необходимо скачать 42,7 МВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 167 МВ.
Хотите продолжить? [Д/н] _
```

Рис.18. Установка postgresql

4. Установка Composer

1) Устанавливаем Composer в систему Debian

```
root@debian:/# wget -O composer-setup.php https://getcomposer.org/installer
--2023-01-17 16:09:59-- https://getcomposer.org/installer
Распознаётся getcomposer.org (getcomposer.org)... 54.36.53.46, 2001:41d0:302:1100::8:104f
Подключение к getcomposer.org (getcomposer.org)[54.36.53.46]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: 58140 (57K) [application/octet-stream]
Сохранение в: «composer-setup.php»

composer-setup.php      100%[=====] 56,78K  --.-KB/s   за 0,07s
2023-01-17 16:10:00 (818 KB/s) - «composer-setup.php» сохранён [58140/58140]
root@debian:/# _
```

Рис.19. Установка Composer

2) Чтобы установить Composer глобально внутри каталога, выполнив следующую команду:/usr/local/bin

```
root@debian:/# php composer-setup.php --install-dir=/usr/local/bin --filename=composer
All settings correct for using Composer
Downloading...

Composer (version 2.5.1) successfully installed to: /usr/local/bin/composer
Use it: php /usr/local/bin/composer
root@debian:/# _
```

Рис.20. Установка Composer глобально внутри каталога

5. Установка Symfony

1) Устанавливаем Symfony в систему Debian

```
root@debian:/# wget https://get.symfony.com/cli/installer -O - | bash
--2023-01-17 16:19:03-- https://get.symfony.com/cli/installer
Распознаётся get.symfony.com (get.symfony.com)... 108.156.22.112, 108.156.22.52, 108.156.22.17, ...
Подключение к get.symfony.com (get.symfony.com)[108.156.22.112]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: 6100 (6,0K) [binary/octet-stream]
Сохранение в: «STDOUT»

100%[=====] 5,96K --.-KB/s за 0,001s

/2023-01-17 16:19:03 (8,46 MB/s) - записан в stdout [6100/6100]

Symfony CLI installer

Environment check
[*] cURL is installed
[*] Tar is installed
[*] Git is installed
[*] Your architecture (amd64) is supported

Download
Downloading https://github.com/symfony-cli/symfony-cli/releases/latest/download/symfony-cli_linux_
amd64.tar.gz...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:--:-- 0:00:01 --:--:-- 0
0 0 0 0 0 0 0 0 --:--:-- 0:00:01 --:--:-- 0
```

Рис.21. Установка Symfony

2) Переместим Symfony в /usr/local/bin

```
root@debian:/# mv /root/.symfony5/bin/symfony /usr/local/bin/symfony
root@debian:/# _
```

Рис.22. Перемещение Symfony в /usr/local/bin

6. Настройка проекта на сервере

1) Создание демо проекта

```
root@debian:/home/vlad/lr# symfony new --demo project
* Creating a new Symfony Demo project with Composer
  (running /usr/local/bin/composer create-project symfony/symfony-demo /home/vlad/lr/project --no-interaction)

* Setting up the project under Git version control
  (running git init /home/vlad/lr/project)

[OK] Your project is now ready in /home/vlad/lr/project

root@debian:/home/vlad/lr# cd project/
root@debian:/home/vlad/lr/project# ls
assets          data            public          translations
bin             migrations     README.md      var
composer.json  package.json   src            vendor
composer.lock  phpstan-baseline.neon  symfony.lock  webpack.config.js
config         phpstan.neon.dist  templates      yarn.lock
CONTRIBUTING.md  phpunit.xml.dist  tests
```

Рис.23. Демо проект

2) Запуск демо проекта командой symfony serve

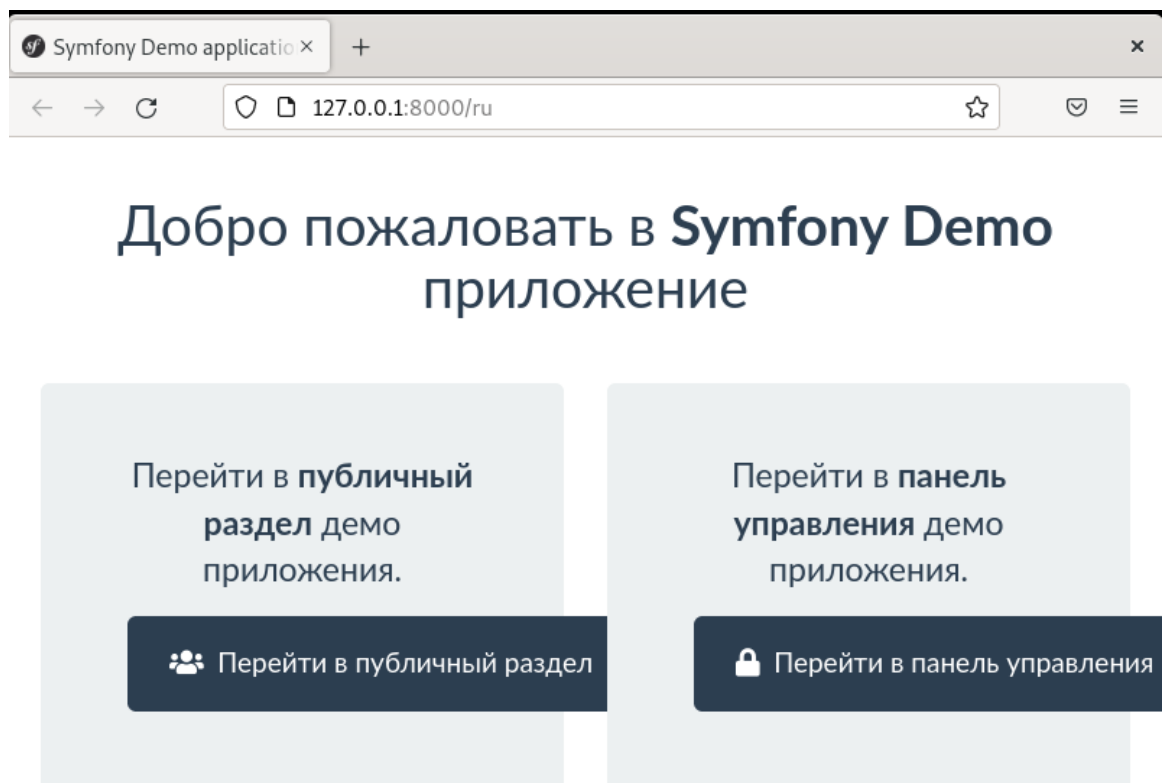


Рис.24. Запуск демо проекта

7. Настройка конфигурации Docker-Compose для запуска мультиконтейнерного приложения

1) Создание docker-compose.yml, для сборки приложения

```
GNU nano 5.4 docker-compose.yml
version: '3.8'

services:
  db:
    container_name: db
    image: postgres:12
    restart: always
    environment:
      POSTGRES_PASSWORD: 12345
      POSTGRES_DB: lr_db
    ports:
      - 5432:5432
    volumes:
      - ./data/postgresql:/var/lib/postgresql/data
      - ./lr.db_backup.sql:/docker-entrypoint-initdb.d/lr_db_backup.sql
  php-fpm:
    container_name: php-fpm
    build:
      context: ./php-fpm
    depends_on:

    depends_on:
      - APP_SECRET=${APP_SECRET}
      - DATABASE_URL=${DATABASE_URL}
    volumes:
      - ../../symfony:/var/www

  nginx:
    container_name: nginx
    build:
      context: ./nginx
    volumes:
      - ../../symfony:/var/www
      - ./nginx/nginx.conf:/etc/nginx/nginx.conf
      - ./nginx/sites:/etc/nginx/sites-available
      - ./nginx/conf.d:/etc/nginx/conf.d
    depends_on:
      - php-fpm
    ports:
      - "8080:80"
      - "443:443"
```

Рис.25-26. Создание docker-compose.yml

2) Для базы данных изменим конфигурацию в .env

```
# Run "composer dump-env prod" to compile .env files for production use (require
# https://symfony.com/doc/current/best_practices.html#use-environment-variables>

###> 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
# IMPORTANT: You MUST configure your server version, either here or in config/p
#
#DATABASE_URL=sqlite:///kernel.project_dir%/data/database.sqlite
# DATABASE_URL="mysql://db_user:db_password@127.0.0.1:3306/db_name?serverVersion
# DATABASE_URL="postgresql://db_user:db_password@127.0.0.1:5432/db_name?serverV
DATABASE_URL="postgresql://postgres:12345@127.0.0.1:5432/lr_db"
###< doctrine/doctrine-bundle ###
```

Рис.27. Изменение конфигурации в .env

3) Создадим и заполним Dockerfile в php-fpm

```
GNU nano 5.4 Dockerfile
FROM php:8.2-fpm

RUN apt-get update && \
    apt-get install -y --no-install-recommends libssl-dev zlib-dev curl git u
    apt-get 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 apcu pdo_pgsql sodium && \
    apt-get clean && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*

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

WORKDIR /var/www

CMD composer i -o ; php-fpm

EXPOSE 9000
```

Рис.28. Dockerfile

4) Создаем и заполняем Dockerfile в nginx

```
GNU nano 5.4 Dockerfile
FROM nginx:alpine

WORKDIR /var/www

CMD ["nginx"]

EXPOSE 80 443
```

Рис.29. Dockerfile

- 5) Выполним команду `docker-compose up` иждемся выполнения скачивания образов и установки зависимостей composer

```
|root@debian:/home/vlad/1r/project# docker-compose up
```

Рис.30. `docker-compose up`

- 6) Смотрим результаты запуска

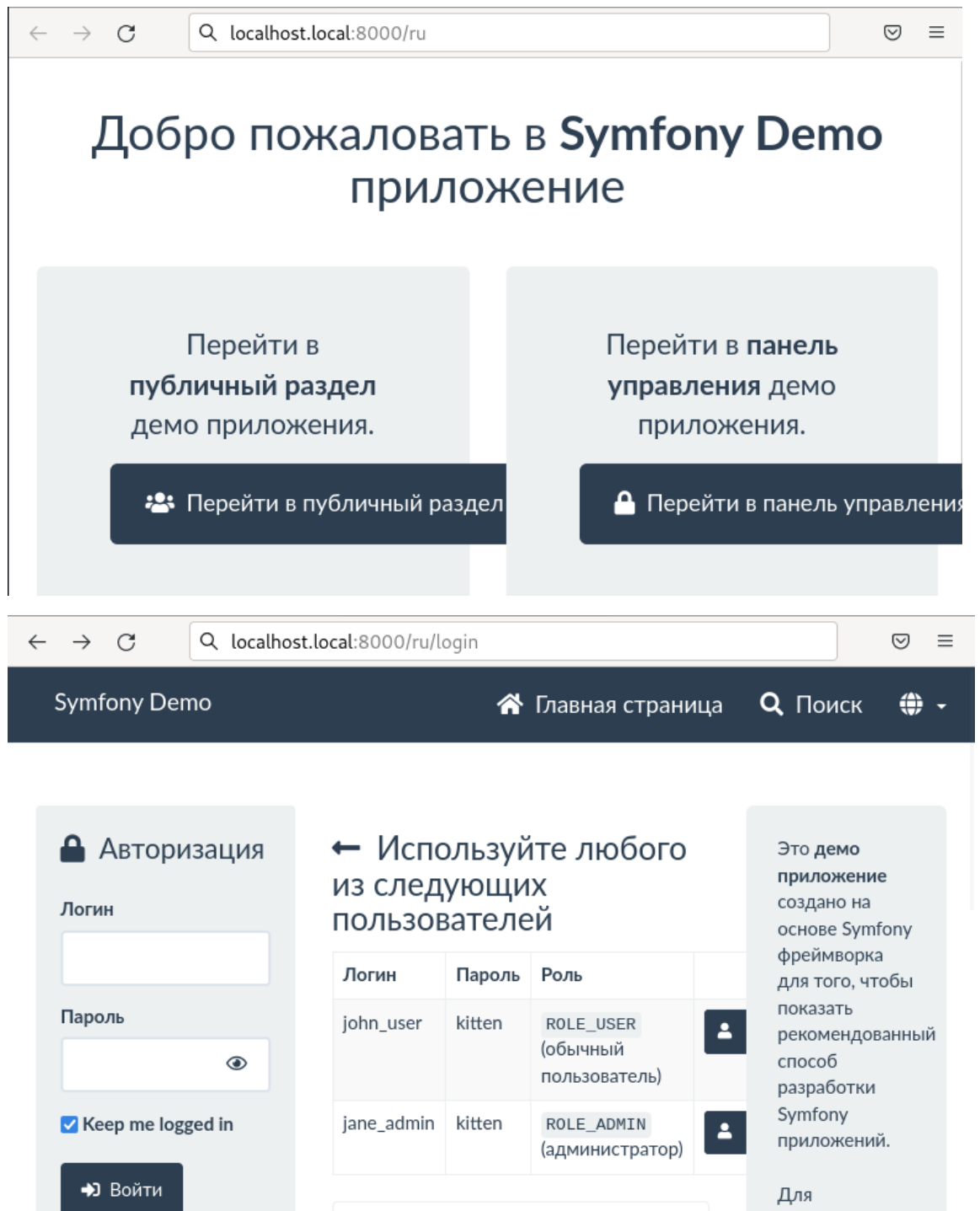


Рис.31-32. Результаты запуска `docker-compose`

8. Контрольные вопросы

- 1) Назовите отличия использования контейнеров по сравнению с виртуализацией.

A. Меньшие накладные расходы на инфраструктуру

B. Время старта приложений больше

C. Невозможность запуска GNU/Linux- и Windows-приложений на одном хосте

D. Обязательное использование гипервизора KVM

- 2) Назовите основные компоненты Docker.

A. Гипервизор

B. Контейнеры

C. Образы виртуальных машин

D. Реестры

- 3) Какие технологии используются для работы с контейнерами?

A. Пространства имен (Linux Namespaces)

B. Подключаемые модули аутентификации (PAM)

C. Контрольные группы (cgroups)

D. Аппаратная поддержка виртуализации

- 4) Найдите соответствие между компонентом и его описанием:

— образы – доступные только для чтения шаблоны приложений;

— контейнеры – изолированные при помощи технологий операционной системы пользовательские окружения, в которых выполняются приложения;

— реестры (репозитории) – сетевые хранилища образов;

- 5) В чем отличие контейнеров от виртуализации?

Контейнер обеспечивает виртуализацию на уровне операционной системы, а аппаратная виртуализация обеспечивает виртуализацию на уровне машины.

- 6) Перечислите основные команды утилиты Docker с их кратким описанием

— `docker ps` — показывает список запущенных контейнеров;

- `docker pull` — скачать определённый образ или набор образов (репозиторий);
- `docker build` — эта команда собирает образ Docker из Dockerfile и «контекста»;
- `docker run` — запускает контейнер, на основе указанного образа;
- `docker logs` — эта команда используется для просмотра логов указанного контейнера;
- `docker volume ls` — показывает список томов, которые являются предпочитаемым механизмом для сохранения данных, генерируемых и используемых контейнерами Docker;
- `docker rm` — удаляет один и более контейнеров;
- `docker rmi` — удаляет один и более образов;
- `docker stop` — останавливает один и более контейнеров;
- `docker exec -it ...` - выполняет команду в определенном контейнере

7) Каким образом осуществляется поиск образов контейнеров?

Сначала проверяется локальный репозиторий на наличия нужного контейнера, если он не найден локально, то поиск производится в репозитории Docker Hub

8) Каким образом осуществляется запуск контейнера?

Команда `docker run (container_name)`

9) Что значит управлять состоянием контейнеров?

Это означает, что в любой момент времени есть возможность запустить, остановить или выполнить команды внутри контейнера.

10) Как изолировать контейнер?

Контейнеры уже по сути своей являются изолированными единицами, поэтому достаточно без ошибок сконфигурировать файлы Dockerfile и/или docker-compose.yml.

11) Опишите последовательность создания новых образов, назначение Dockerfile?

Производится выбор основы для нового образа на Docker Hub, далее производится конфигурация Dockerfile, где описываются все необходимые пакеты, файлы, команды и т.п. Dockerfile — это текстовый файл с инструкциями, необходимыми для создания образа контейнера. Эти инструкции включают идентификацию существующего образа, используемого в качестве основы, команды, выполняемые в процессе создания образа, и команду, которая будет выполняться при развертывании новых экземпляров этого образа контейнера.

- 12) Возможно ли работать с контейнерами Docker без одноименного движка?

Да. С помощью Kubernetes.