**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Лабораторная работа № 5

по OS Linux

Контейнеризация

Студент Комаричев А. В.

Группа АИ-19

Руководитель Кургасов В. В.

Липецк 2021г.

Цель работы

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

Ход работы

Для начала работы необходимо установить php8.0, composer, symphony, sqlite.


Рисунок 1 – php

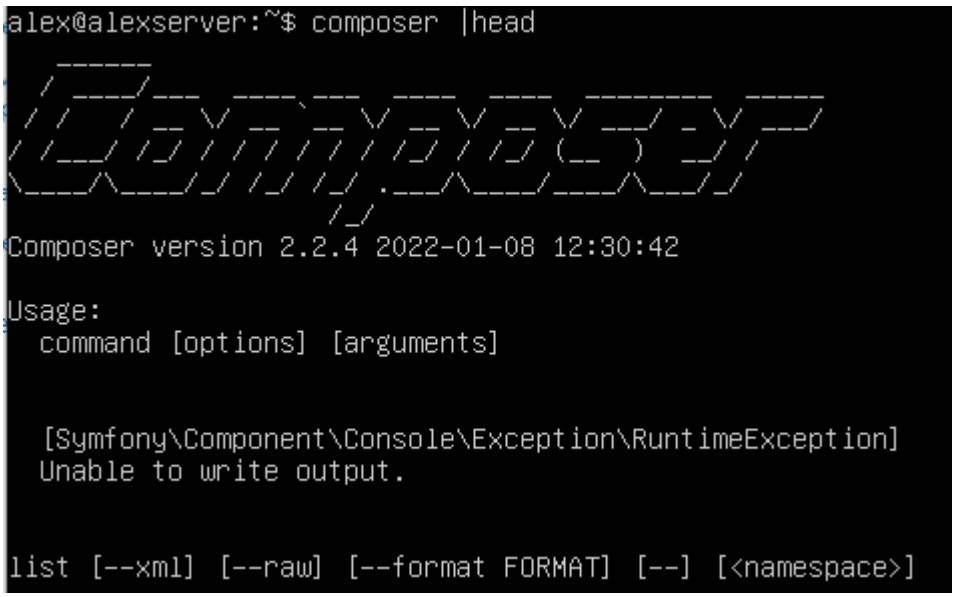



Рисунок 2 – composer

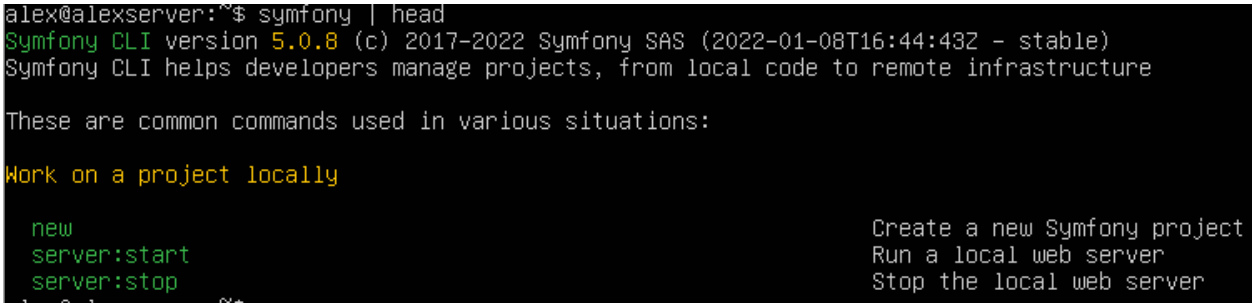



Рисунок 3 – symfony

Клонируем demo проект командой git clone https://github/com/symfony/demo

После того как проект клонирован и окружение настроено, перейдем в папку проекта.

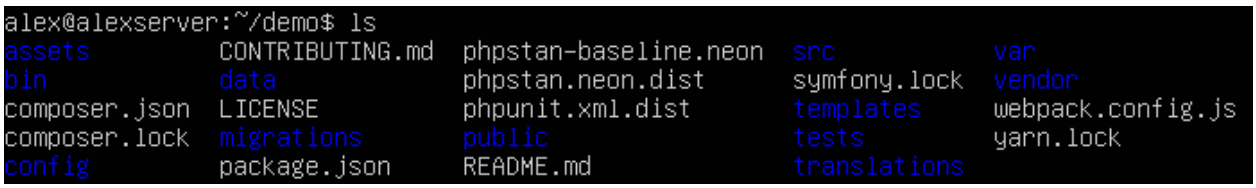
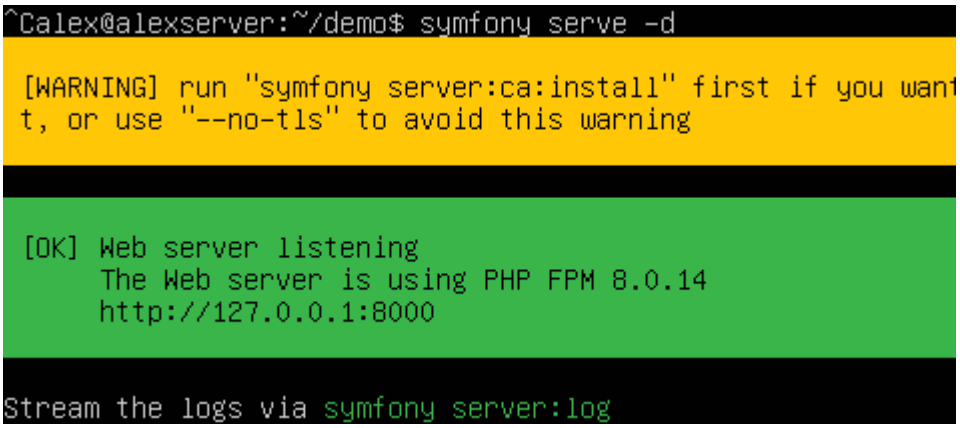



Рисунок 4 – demo проект

Запускаем проект symfony serve




Запущенный проект можно видеть в браузере по адресу localhost. Адрес у меня в браузере отличается, так как проект запущен на гостевой машине, а браузер установлен на основной ос.

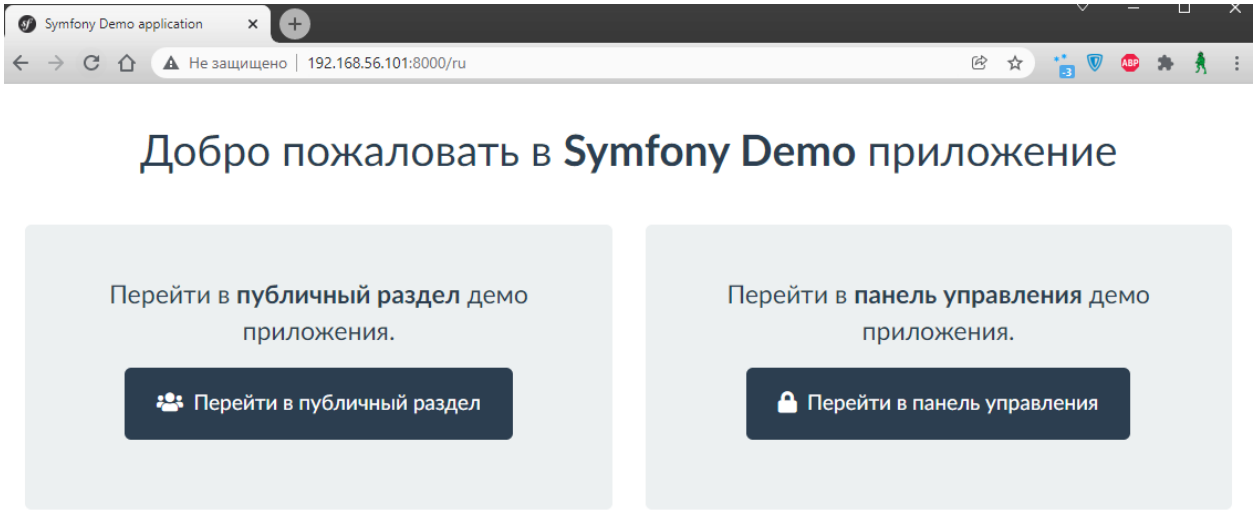



Рисунок 5 – Запуск проекта

Установим docker и docker-compose. Пакет установки Docker, доступный в официальном репозитории Ubuntu, может содержать не самую последнюю версию. Чтобы точно использовать самую актуальную версию, мы будем устанавливать Docker из официального репозитория Docker. Для этого мы добавим новый источник пакета, ключ GPG от Docker, чтобы гарантировать

загрузку рабочих файлов, а затем установим пакет. Для установки docker-compose используем команду

sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

```
alex@alexserver:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2022-01-10 14:55:17 UTC; 1min 0s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 8713 (dockerd)
      Tasks: 7
     Memory: 27.5M
     CGroup: /system.slice/docker.service
             └─8713 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Рисунок 6 – Docker установлен и работает

```
alex@alexserver:~$ docker-compose --version
docker-compose version 1.29.2, build 5becea4c
```

Рисунок 7 – docker-compose установлен

Установим postgresql, чтобы заменить sqlite базу (sudo apt -y install postgresql).

```
postgres=# \l
                                 List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 mydb      | postgres | UTF8     | ru_RU.UTF-8 | ru_RU.UTF-8 | =Tc/postgres         +
           |          |          |             |             | postgres=CTc/postgres+
           |          |          |             |             | myuser=CTc/postgres
 postgres  | postgres | UTF8     | ru_RU.UTF-8 | ru_RU.UTF-8 |
 template0 | postgres | UTF8     | ru_RU.UTF-8 | ru_RU.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
 template1 | postgres | UTF8     | ru_RU.UTF-8 | ru_RU.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
(4 rows)
```

Рисунок 8 – Создаем базу данных (mydb)

Редактируем файл .env. В DATABASE_URL указываем postgresql базу, указываем адрес, порт, название базы данных, имя пользователя и его пароль.

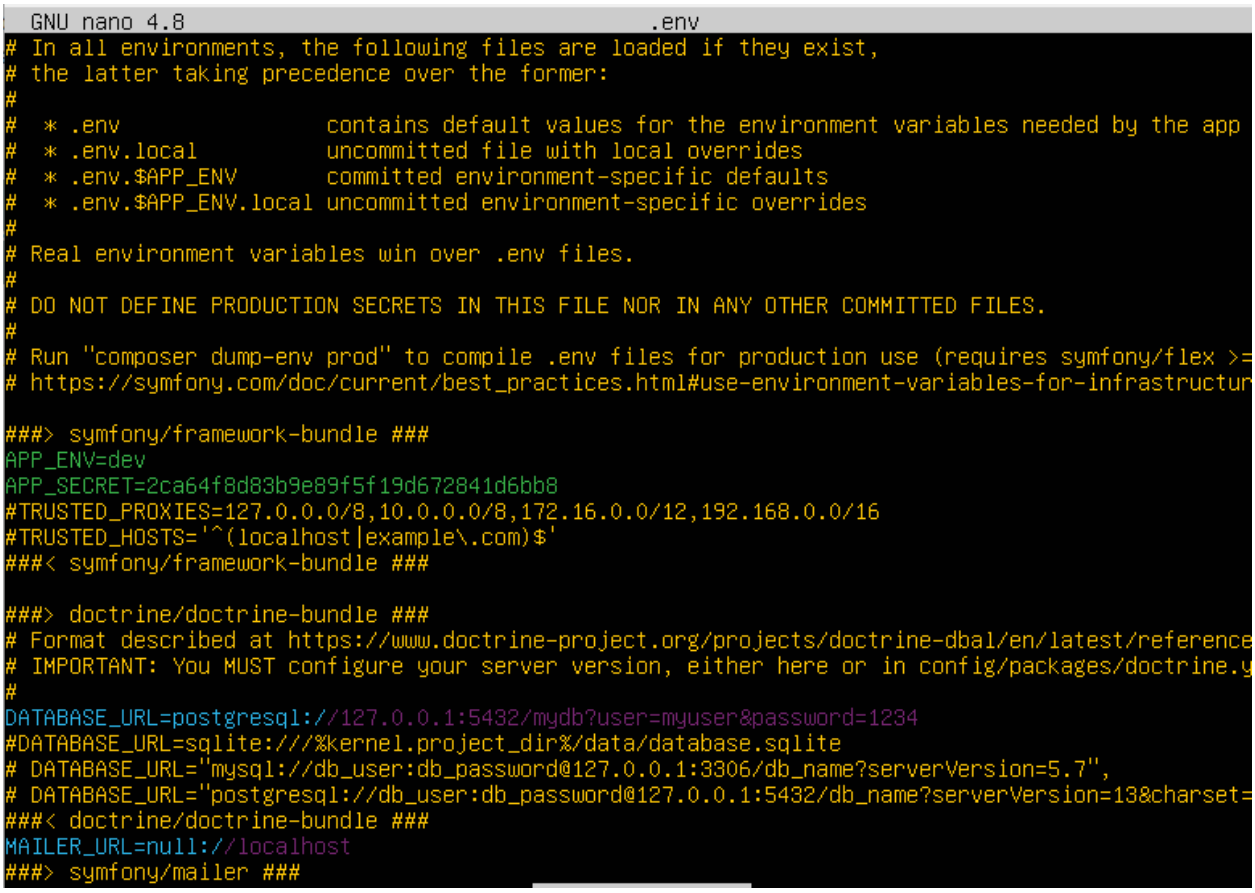



Рисунок 9 – Изменение фала .env

Заполняем базу данных данными из фикстур. Команды:

php bin/console doctrine:schema:create,

php bin/console doctrine:fixtures:load.

Для этого необходимо также установить драйвер pdo-pgsql (sudo apt install php-pgsql).

Для проверки работоспособности новый базы данных, добавим туда новую запись.

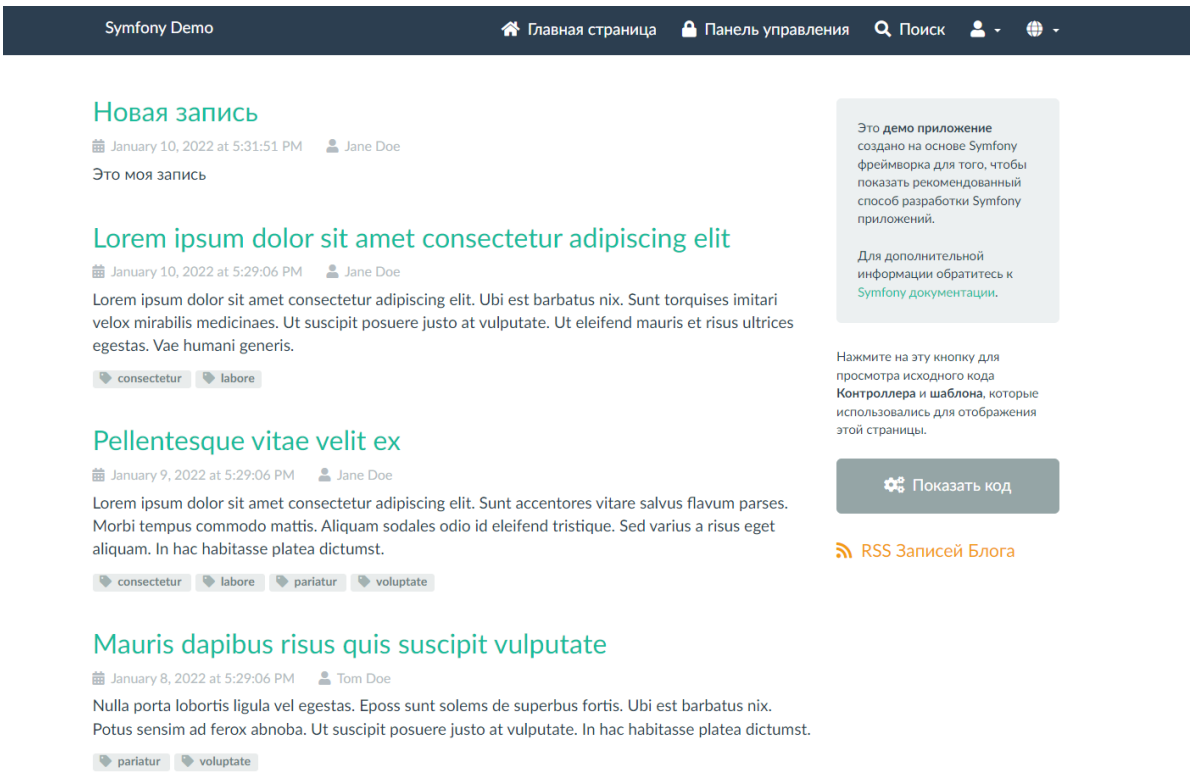



Рисунок 10 – Добавление новой записи

Проект работает, перейдем к настройке контейнеров.

Создадим папку docker внутри проекта. В этой папке создадим файл docker-compose.yaml в этом файле будет конфигурация всех контейнеров. Создадим папку nginx и Dockerfile в ней с конфигурацией контейнера nginx, папку php-fpm и Dockerfile в ней с конфигурацией контейнера php-fpm, папку postgres и Dockerfile в ней с конфигурацией контейнера postgres.

```
  GNU nano 4.8                                    Dockerfile
FROM nginx
COPY ./config.conf /etc/nginx/conf.d/default.conf
WORKDIR /var/www
CMD ["nginx"]
EXPOSE 80 443
```

Рисунок 11 – Dockerfile nginx

Для конфигурации nginx также создадим файл config.conf

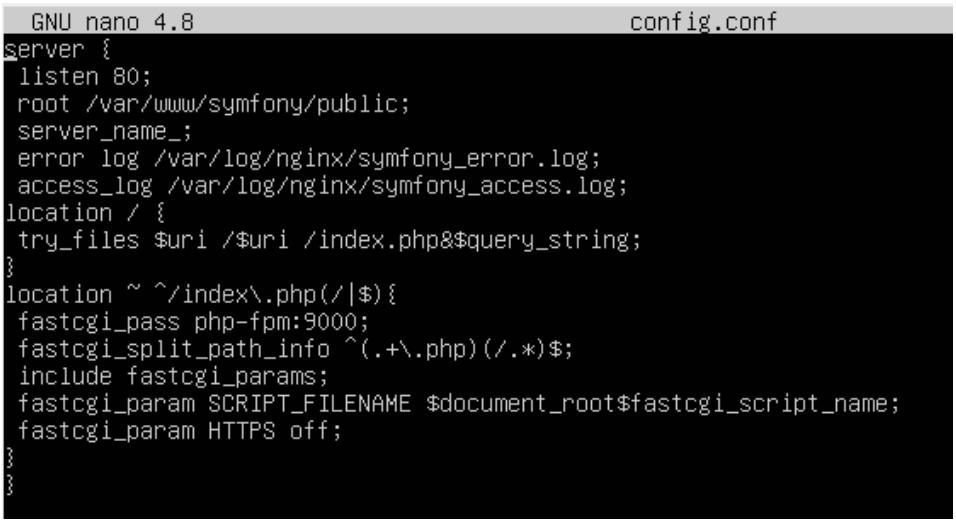



Рисунок 12 – config.conf

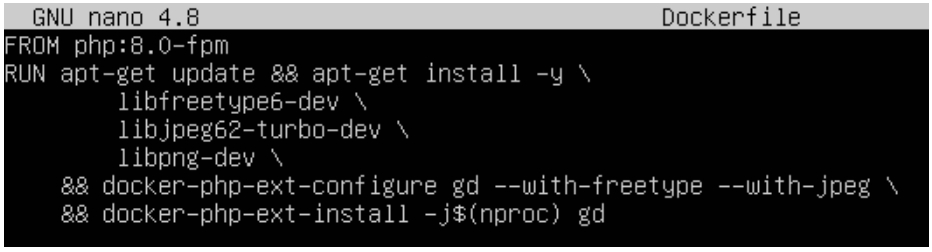



Рисунок 13 – Dockerfile php-fpm

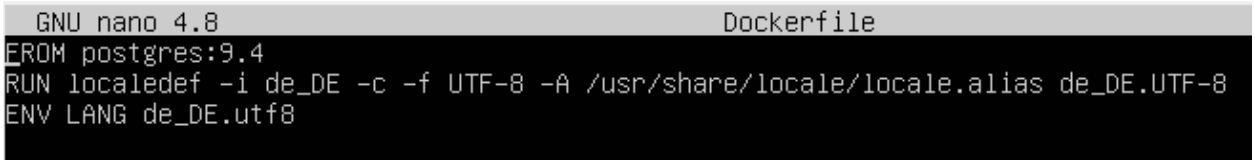



Рисунок 14 – Dockerfile postgres

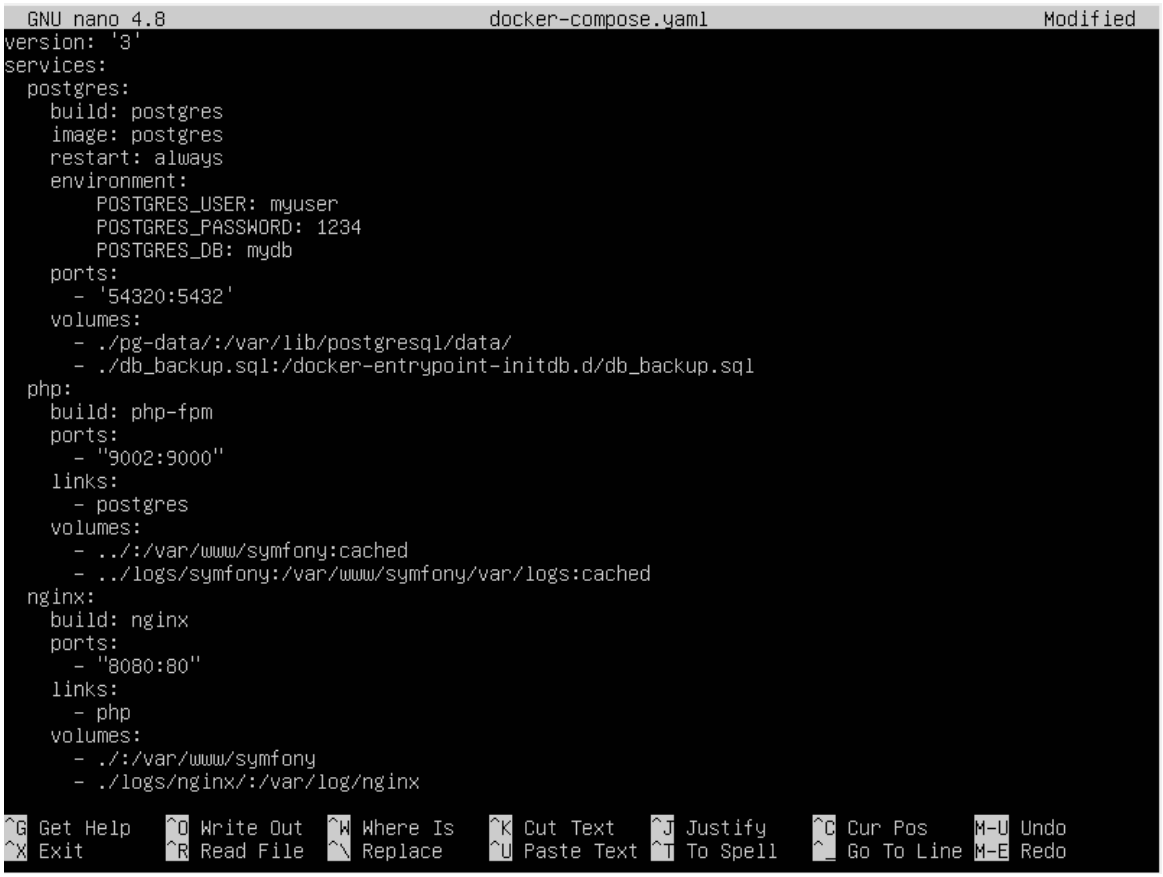



Рисунок 15 – Файл docker-compose.yaml

Далее редактируем файл .env. Вместо 127.0.0.1 указываем имя контейнера.

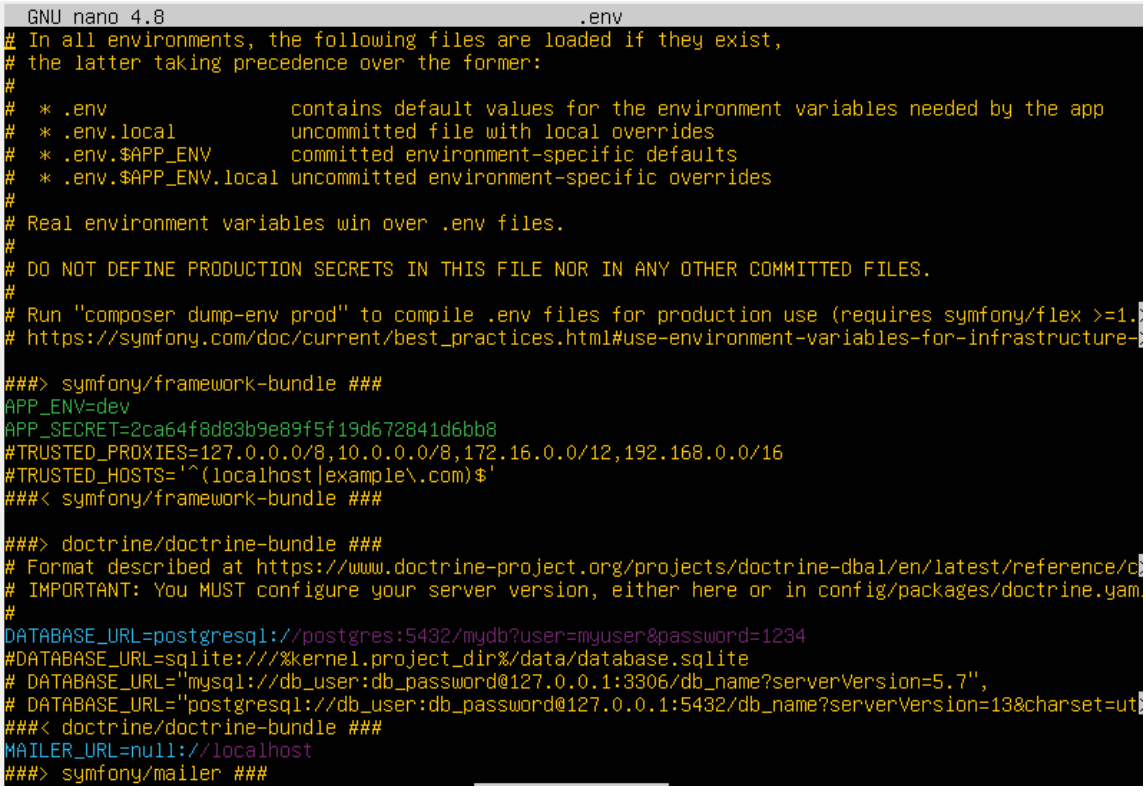



Рисунок 16 – Файл .env

Запускаем проект командой docker-compose up.

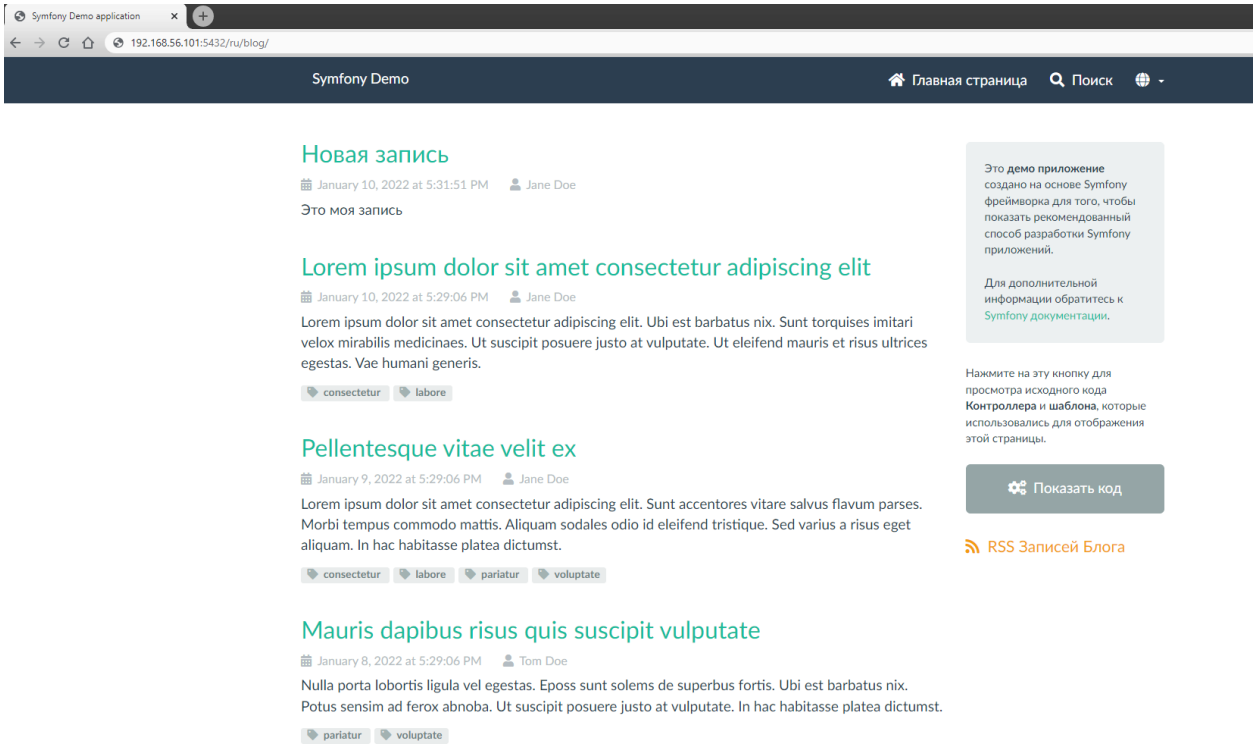



Рисунок 17 – Проект, запущенный из контейнера

Контрольные вопросы

1. Назовите отличия использования контейнеров по сравнению с виртуализацией.

Основное отличие контейнеров и виртуальных машин заключается в том, что виртуальные машины виртуализируют весь компьютер вплоть до аппаратных уровней, а контейнеры — только программные уровни выше уровня операционной системы. Следовательно,
Меньшие накладные расходы на инфраструктуру.

2. Основные компоненты Docker

Контейнеры.

3. Какие технологии используются для работы с контейнерами?

Контрольные группы.

4. Соответствие между компонентом и его описанием

| | |
|---|---|
| контейнеры | изолированные при помощи технологий операционной системы пользовательские окружения, в которых выполняются приложения. |
| образы | доступные только для чтения шаблоны приложений |
| реестры | сетевые хранилища образов |

5. В чем отличие контейнеров от виртуализации?

Виртуальная машина.
Требуется гипервизор, для каждой ВМ используется собственная гостевая ОС. Позволяет создавать неоднородные вычислительные среды на одном компьютере. Из-за собственной ОС ВМ может занимать несколько ГБ, а запуск ОС и всех приложений занимает какое-то время.
Контейнер.
Даже несколько контейнеров используют ядро одной хостовой ОС. Позволяет

создавать на одном компьютере только однородные вычислительные среды. Намного легче ВМ, размер измеряется в Мб. Способен запускаться почти мгновенно.

6. Перечислите основные команды утилиты Docker с их кратким описанием.

| Команда | Описание |
|---|---|
| docker login (logout) | Вход (выход) в реестр |
| docker search nginx | Поиск образа |
| docker pull (push) nginx | Выгрузка из реестра образа (загрузка в реестр) |
| docker create [options] image [command] [arg…] | Создание контейнера |
| docker run [OPTIONS] IMAGE [COMMAND] [ARG...] | Запуск контейнера |
| docker rename | Переименование контейнера |
| docker rm | Удаление контейнера |
| docker update | Обновление контейнера |
| docker stop (start) container | Остановка (запуск) контейнера |
| docker restart nginx | Перезагрузка контейнера |
| docker kill –s HUP nginx | Отправка сигнала контейнеру |
| docker ps | Работающие контейнеры |
| docker images | Список образов |
| docker build … | Создание образов |
| docker rmi | Удаление образа |

7. Каким образом осуществляется поиск образов контейнеров?

Docker проверяет локальный репозиторий на наличие искомого образа, если образ не был найден, производится поиск в репозитории Docker hub.

8. Каким образом осуществляется запуск контейнера?

Docker выполняет инициализацию и запускает контейнер. Контейнер собирается из образа.

9. Что значит управлять состоянием контейнеров?

В любой момент времени можно запустить, остановить или выполнить команду внутри контейнера.

10. Как изолировать контейнер?

Контейнеры являются изолированными.

11. Опишите последовательность создания новых образов, назначение Dockerfile?

Для создания нового образа необходимо выбрать основу на Docker Hub, затем произвести конфигурацию Dockerfile. Dockerfile – это текстовый файл с инструкциями для создания образа контейнера. Инструкции включают идентификацию существующего образа, команды, выполняемые в процессе создания образа и команды, выполняемые при развертывании новых экземпляров этого контейнера.

12. Возможно ли работать с контейнерами Docker без одноименного движка?

Да возможно, в среде виртуализации Kubernetes.

13. Опишите назначение системы оркестрации контейнеров Kubernetes. Перечислите основные объекты Kubernetes?

Kubernetes – это открытое программное обеспечение для автоматизации развертывания, масштабирования контейнеризированных приложений и управления ими. Поддерживает основные технологии контейнеризации.

Основные объекты:

- Кластеры: пул для вычислений, хранения и сетевых ресурсов.
- Ноды: хост-машины, работающие в кластере.
- Пространства имен: логические разделы кластера.
- Поды: единицы развертывания.

- Метки и селекторы: пары «ключ-значение» для идентификации и обнаружения сервисов.
- Сервисы: коллекция подов, принадлежащих одному и тому же приложению.
- Набор реплик: обеспечивает доступность и масштабируемость.
- Развертывание: управляет жизненным циклом приложения.