|  |  |
| --- | --- |
|  |  |
| МИНОБРНАУКИ РОССИИ |  |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |  |

**Институт информационных технологий**

КАФЕДРА ИНСТРУМЕНТАЛЬНОГО И ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (ИиППО)

Практическая работа 1

по дисциплине «Разработка серверных частей интернет-ресурсов»

Выполнил студент группы ИКБО-01-20 *Клочек А.М.*

Принял преподаватель *Волков М.Ю.*

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2022г. *подпись студента*

«Зачтено» «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022г. *подпись преподавателя*

Москва 2022

# ОГЛАВЛЕНИЕ

[Цель работы 3](#_Toc113206000)

[Теоретическое введение 3](#_Toc113206001)

[Задание на практическую работу 3](#_Toc113206002)

[Выполнение практической работы 4](#_Toc113206003)

[Вывод 6](#_Toc113206004)

[Ссылка на удалённый репозиторий проекта 6](#_Toc113206005)

[Ответы на вопросы: 6](#_Toc113206006)

[Список использованной литературы 8](#_Toc113206007)

# 

# Цель работы

Получить навыки создания сервера.

# Теоретическое введение

Существует проблема разработки того или иного приложения и его развертывания на других машинах. Самыми частыми решениями данной проблемы является установочные скрипты, облачные сервисы и виртуальные машины. Описанные подходы не являются оптимальными что раздувает техническую поддержку до максимума, а также медленны и тяжеловесны. Одним из вариантов решения данной задачи является докер, который представляет технологию контейнеризации.

Подобно виртуальной машине докер запускает свои процессы в собственной, заранее настроенной операционной системе. Но при этом все процессы докера работают на физическом host-сервере, деля все процессоры и всю доступную память со всеми другими процессами, запущенными в host-системе. Подход, используемый докером находится посередине между запуском всего на физическом сервере и полной виртуализацией, предлагаемой виртуальными машинами. Этот подход называется контейнеризацией.

# Задание на практическую работу

Создать свою конфигурацию серверного программного обеспечения, в которой должны присутствовать веб-сервер, операционная система, язык программирования и база данных.

# Выполнение практической работы

Выполнение данной практической работы происходит на Windows 10 при помощи WSL2 и docker desktop.

Для начала опишем конфигурацию контейнера веб-сервера, которая показана на рисунке1.

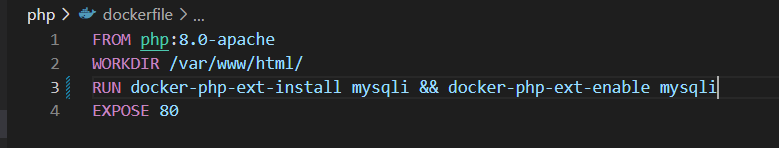


Рис. 1 Содержание Dockerfile веб сервера

Далее опишем конфигурацию взаимодействия контейнеров между собой и хостом в файле docker-compose.yml, которая показана на рисунке 2.

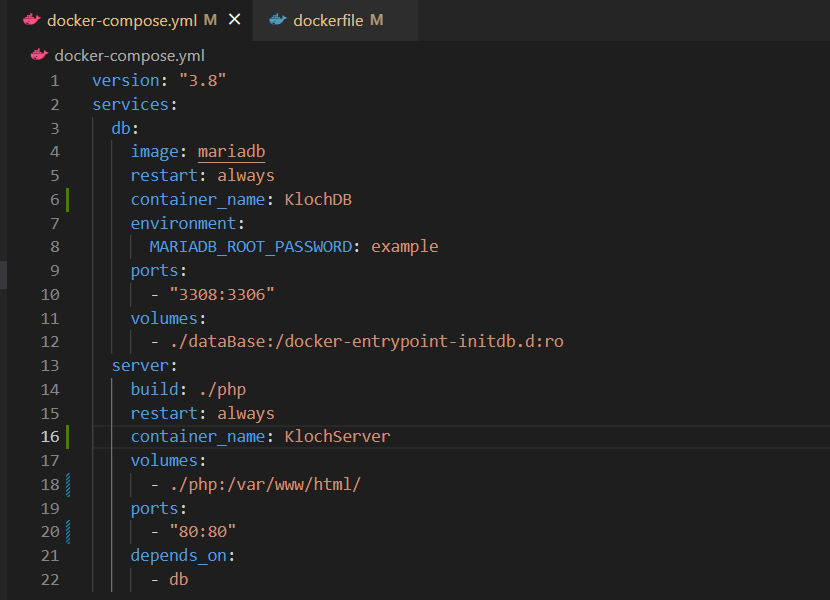


Рис. 2 Содержание файла docker-compose

Запустим контейнеры с помощью команды docker-compose up (рисунок 3).

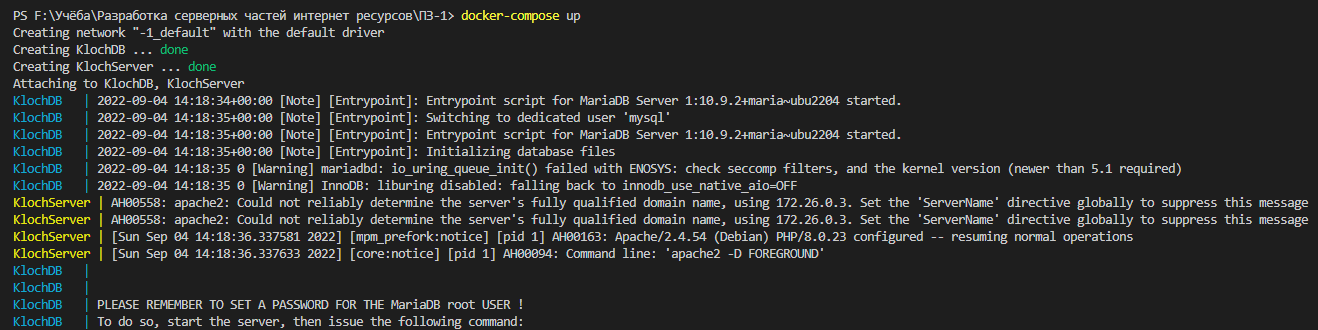


Рис. 3 Выполнение команды docker-compose up

Для демонстрации того, что контейнеры отработали правильно перейдём по адресу localhost:8081/index.php (рисунок 4).

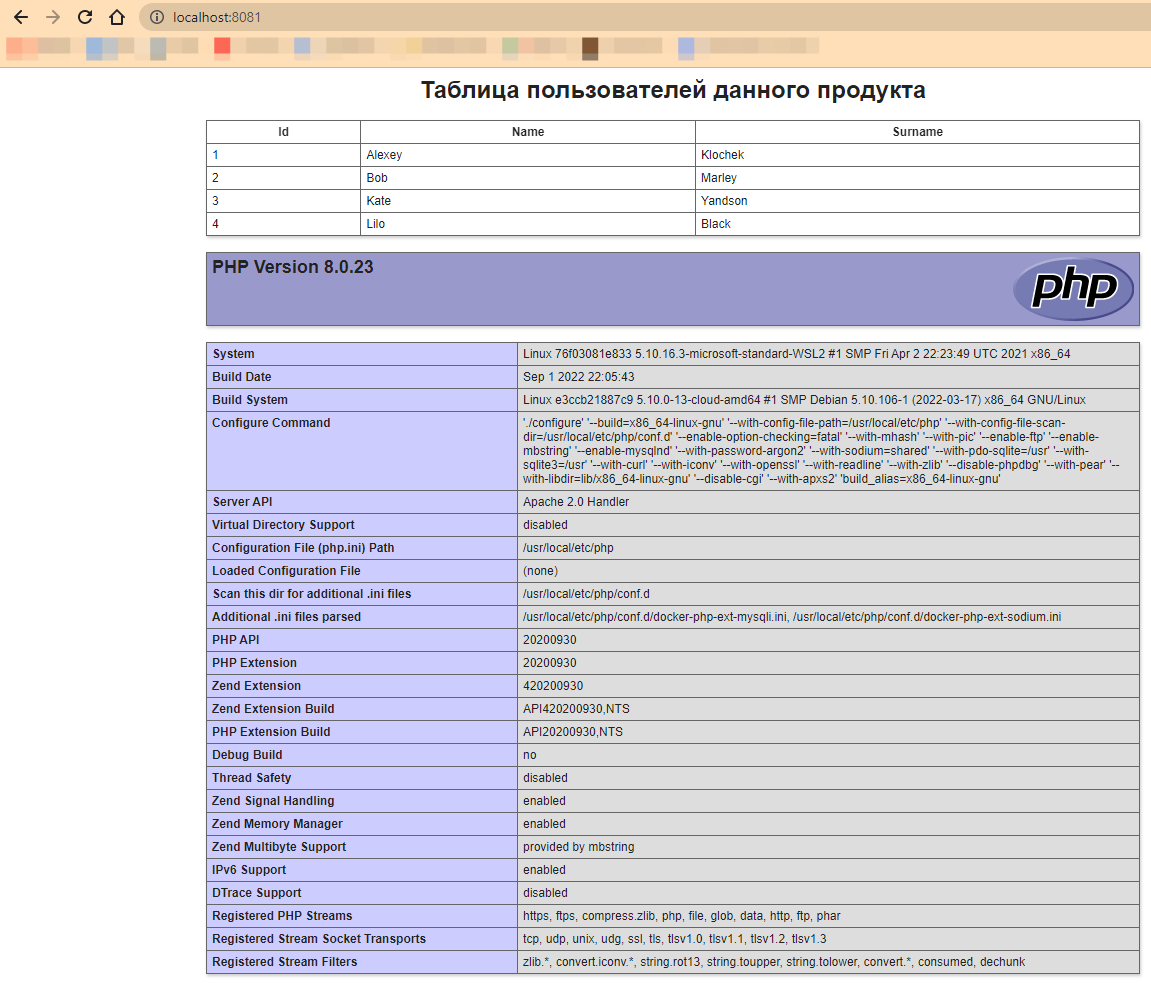


Рис. 4 Результат работы

# Вывод

В ходе выполнения данной практической работы были получены навыки настройки и запуска сервера на стеке LAMP, а также навыки развёртывания и связывания контейнеров с помощью Docker и docker-compose.

# Ссылка на удалённый репозиторий проекта

Ссылка на репозиторий: <https://github.com/Agser228/RSCHIR>

# 

# Ответы на вопросы:

1) Сервер – это сторона, которая принимает запросы от клиента и предпринимает соответствующие действия с данными необходимые по запросу.

2) База данных — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

3) API (программный интерфейс приложения, интерфейс прикладного программирования) — описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

4) Сервис - одна ограниченная функция сервера.

5) «Клиент — сервер» — архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.

6) Веб-серверы, Почтовые серверы, Файл-серверы, Прокси-серверы, Почтовые серверы, Веб-серверы, Серверы приложений, Серверы баз данных.

7) Вертикальная масштабируемость – увеличение производительности серверной системы. Горизонтальная масштабируемость – разбиение на более мелкие структурные компоненты и разнесение их (распараллеливание).

8) Протокол передачи данных — набор определённых инструкций интерфейса логического уровня, которые определяют способы обмена информации между различными программами или их компонентами.

9) Тонкий клиент – частный случай клиент-серверной архитектуры, в которой большая часть логики выполняется на сервере. Толстый же клиент, наоборот, когда логика выполняется на стороне клиента.

10) Паттерн MVC: общие тезисы. MVC – это способ организации кода, при котором выделяются компоненты, модель данных приложения, представление и контроллер. Эти компоненты независимы, модификации одного из них, не влияют на друга, что прощает процесс разработки приложения.

11) Паттерн MVC: Model-View-Presenter. Производный от MVC паттерн проектирования. Отличие от MVC в том, что элемент Presenter берёт на себя обработку событий пользовательского интерфейса, в то время как в MVC за это отвечает представление.

12) Паттерн MVC: Model-View-View Model. ModelView – модель представления, обёртка над моделью, преобразованная для прямого использования представлению, содержит команды, которые использует представление для работы с моделью.

13) MVC: Модель представляет собой логику работы с данными и описание фундаментальных данных, необходимых для работы приложения. Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели. Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели.

14) Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации

15) Dockerfile – это сценарий, который состоит из последовательности команд и аргументов, необходимых для создания образа.

16) Docker Compose — это инструментальное средство, входящее в состав Docker. Оно предназначено для решения задач, связанных с развёртыванием проектов.

17) LAMP — Linux, Apache, MariaDB/MySQL, PHP.

# Список использованной литературы

1. Официальный репозиторий образов Docker, URL: <https://hub.docker.com/> (дата последнего обращения: 02.09.2022)
2. Docker самый простой и понятный туториал. Изучаем докер, так, если бы он был игровой приставкой, URL: <https://badcode.ru/docker-tutorial-dlia-novichkov-rassmatrivaiem-docker-tak-iesli-by-on-byl-ighrovoi-pristavkoi/amp/> (дата последнего обращения: 02.09.2022)
3. Шпаргалка с командами Docker: сайт. – <URL:https://habr.com/ru/company/flant/blog/336654/> (дата обращения: 02.01.2022). – Текст: электронный.
4. Официальная документация языка PHP, URL: <https://www.php.net/manual/ru/index.php> (дата последнего обращения: 02.09.2022)
5. Документация PHP: сайт. – URL: <https://www.php.net/docs.php> (дата обращения: 02.01.2022).
6. Medium blog. Статья «Apache Airflow in 10 minutes» [Электронный ресурс]. – URL: (дата последнего обращения: 02.09.2022)