

Развертка сервиса

Установка Docker

Для MacOS

Colima является бесплатной альтернативой Docker Desktop для MacOS и Linux и представляет из себя инструмент для управления контейнерами Docker.

1. Необходимо удалить Docker Desktop, если он установлен.
2. Для непосредственной установки Colima будем использовать Homebrew командой консоли (далее мы будем всё время использовать консоль для управления):

```
/bin/bash -c "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

3. Таким же образом установите сам Docker и Colima:

```
brew install docker  
brew install docker-compose  
brew install --HEAD colima
```

4. После установки запустите виртуальную машину:
Для процессоров Apple Silicon (M1/M2/ARM), Lima >=0.14, MacOS >=13.0:

```
colima start --arch aarch64 --vm-type=vz --vz-rosetta  
Для процессоров Intel:  
colima start --vm-type=vz (последний аргумент для Lima>=0.14, MacOS >= 13.0)
```

5. Убедитесь, что Colima запущена:

```
colima status  
INFO[0000] colima is running using macOS Virtualization.Framework  
<...>
```

6. Откройте командную строку и выполните команду:

```
docker -v
```

Если отобразится версия Docker, это означает, что установка Docker прошла успешно.

7. Через терминал создайте хранилище для контейнеров:

```
docker volume create pgdata
```

8. Теперь можно использовать Docker.

Для Astra Linux

Установка должна выполняться от имени пользователя, являющегося администратором системы (при включенном МКЦ - пользователя с высоким уровнем целостности).

1. Для установки Docker требуется выполнить следующие команды:

```
sudo apt update  
sudo apt install docker.io
```

2. После установки Docker рекомендуется предоставить администратору право работать с контейнерами не используя sudo:

```
sudo usermod -aG docker $USER
```

3. Откройте командную строку и выполните команду:

```
docker -v
```

Если отобразится версия Docker, это означает, что установка Docker прошла успешно.

4. Через терминал создайте хранилище для контейнеров:

```
docker volume create pgdata
```

5. Теперь можно использовать Docker.

Для Windows

Рассмотрим установку Docker Desktop for Windows — это Community-версия Docker для систем Microsoft Windows.

1. Включите функции Hyper-V Containers Window. Для этого перейдите в панель управления - установка и удаление программ - включение или отключение компонентов Windows. Активируйте пункт Hyper-V, который включает Hyper-V Managment Tools, Hyper-V Platform.

Также это можно выполнить через powershell или dism (все команды необходимо выполнять с правами администратора).

Powershell:

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
```

DISM:

```
DISM /Online /Enable-Feature /All /FeatureName:Microsoft-Hyper-V
```

2. Скачайте установщик Docker (Docker Desktop Installer) с Docker Hub:
<https://docs.docker.com/docker-for-windows/install>
3. Запустите установщик Docker Desktop Installer.exe и ожидайте, пока он скачает все необходимые компоненты.
4. После установки система потребует перезагрузки. Перезагрузитесь и войдите в систему.
5. После входа может возникнуть запрос на установку дополнительного компонента WSL2. Перейдите по ссылке и скачайте необходимый пакет с официального сайта Microsoft.
6. После скачивания выполните установку WSL2, после которой снова потребуется перезагрузка.
7. Откройте командную строку и выполните команду:

```
docker -v
```

Если отобразится версия docker, это означает, что установка docker в Windows прошла успешно.

8. Через терминал создайте хранилище для контейнеров:

```
docker volume create pgdata
```

9. Теперь можно использовать Docker.

Установка сервиса

Подготовка

1. Скачайте архив проекта из репозитория GitHub. Для этого в репозитории необходимо нажать зелёную кнопку `<> Code` и в появившемся меню нажать `Download ZIP`.
<https://github.com/Krebkel/DocumentMasterDetail>
2. Разархивируйте скачанный архив проекта.
3. Откройте в терминале директорию с проектом `DocumentMasterDetail-master`, в которой лежит файл `Dockerfile`. Все следующие команды необходимо выполнять из этой директории - корня проекта.
4. В терминале необходимо сбилдить проект командой:

```
docker build -t document .
```

Обратите внимание, что в конце команды есть точка.

В терминале должна через некоторое время появиться строка:

```
Successfully tagged document:latest
```

5. После успешного билда запустите сервис вместе с базой данных командой:

```
docker-compose -f docker-compose.yml up
```

6. В терминале выполните команду:

```
docker ps
```

Должны появиться строки в виде таблицы, в которой в столбце IMAGE есть следующие записи:

```
document:latest  
postgres:13
```

7. Сервис готов к работе.

Запуск и использование сервиса

1. Откройте браузер
2. Откройте страницу localhost:80. Появится форма списка документов.
3. Для создания нового документа нажмите кнопку “Добавить документ”. Появится строка с редактируемыми полями нового документа.
 - 3.1.1. Заполните их, затем нажмите “Сохранить”. Документ успешно создан и добавлен в базу.
 - 3.1.2. Создание документа можно отменить, нажав кнопку “Удалить” в строке создаваемого документа.

4. Для добавления, редактирования или удаления позиций в документ нажмите кнопку “Выбрать” в строке нужного документа. Появится форма списка позиций для данного документа.
 - 4.1.1. Добавление позиции в документе происходит по кнопке “Добавить позицию”, аналогично созданию документа. При этом сумма по документу автоматически будет рассчитана по добавленным в базу позициям и отразится в форме списка документов.
 - 4.1.2. Редактирование позиции происходит путём изменения данных позиции и нажатия кнопки “Сохранить” в соответствующей строке.
 - 4.1.3. Удаление позиции происходит путём нажатия кнопки “Удалить” в соответствующей строке. Строка при этом исчезнет.
5. При попытке добавления документа с номером, уже существующим в базе данных, будет вызвана ошибка: “Ошибка: Не удалось добавить новый документ. Документ №_ уже существует!”

При этом в базу данных будет добавлен лог ошибки. Просмотреть логи ошибок можно выполнив в терминале следующие команды:

Подключаемся к базе данных

```
docker exec -it documentmasterdetail-master-postgres-1 psql -ULocalUser -dpostgres
```

Выводим лог последних 10 ошибок через SQL-запрос

```
select * from "documentMasterDetail"."ErrorLogs" limit 10;
```