

# Краткое руководство по настройке WSL 2 и Docker для Windows

---

## Предварительные требования

- Windows 10 версии 2004 и выше (сборка 19041 и выше) или Windows 11
- Подключение к интернету для загрузки компонентов
- Ваши файлы проекта (Dockerfile, main.py, requirements.txt)

## Шаг 1: Включение функции WSL в Windows

### 1. Откройте PowerShell от имени администратора:

- Нажмите Win + X и выберите "Windows PowerShell (администратор)".
- Или введите "PowerShell" в поиск Windows, нажмите правой кнопкой и выберите "Запуск от имени администратора".

### 2. Введите команду для включения WSL:

```
wsl --install
```

Эта команда автоматически установит WSL и скачает дистрибутив Ubuntu по умолчанию.

### 3. Если команда выше не работает, выполните ручную:

- `dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart`
- `dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart`

Перезагрузите компьютер после выполнения команд.

## Шаг 2: Установка и настройка дистрибутива Linux

1. После перезагрузки откроется окно Ubuntu для начальной настройки.
2. Придумайте имя пользователя и пароль для вашего WSL-окружения.

*При вводе пароля символы не отображаются — это нормально.*

### 3. Обновите пакеты в Ubuntu (в терминале WSL):

```
sudo apt update && sudo apt upgrade -y
```

### 4. Установите Docker и Docker Compose

Следуйте официальным инструкциям для установки Docker на Ubuntu. Выполните следующие команды в терминале WSL:

```
# Обновляем индекс пакетов
sudo apt update

# Устанавливаем необходимые пакеты для работы с репозиториями по HTTPS
sudo apt install -y ca-certificates curl

# Создаем директорию для ключей и добавляем официальный GPG-ключ Docker
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Добавляем репозиторий Docker в источники APT
echo \
"deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Снова обновляем индекс пакетов после добавления нового репозитория
sudo apt update

# Устанавливаем последние версии Docker Engine, Containerd и Docker Compose
Plugin
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```

## Шаг 3: Установка Docker Desktop

### 1. Скачайте Docker Desktop для Windows:

- Перейдите на официальный сайт: <https://www.docker.com/products/docker-desktop/>
- Нажмите "Download for Windows".

### 2. Установите Docker Desktop:

- Запустите скачанный установщик **Docker Desktop Installer.exe**.
- Следуйте инструкциям мастера установки (все параметры по умолчанию).
- **Обязательно поставьте галочку "Install required Windows components for WSL 2".**
- Перезагрузите компьютер, когда установка завершится.

### 3. Настройте Docker Desktop для работы с WSL 2:

- Запустите Docker Desktop (из меню "Пуск").

- Перейдите в **Settings** (Настройки) → **General** (Общие).
- Убедитесь, что стоит галочка **Use the WSL 2 based engine**.
- Перейдите в **Settings** → **Resources** → **WSL Integration**.
- Включите интеграцию с вашим установленным дистрибутивом Ubuntu (передвиньте переключатель).

## Шаг 4: Перенос файлов проекта в WSL

Ваши файлы находятся в Windows, но Docker в WSL должен иметь к ним доступ. Есть два способа:

### Способ 1: Прямой доступ через файловую систему WSL (Проще)

- В WSL ваши диски Windows уже подключены.
- Перейдите в папку с вашим проектом. Например, если проект в `C:\Users\ВашеИмя\my_project`:

```
cd /mnt/c/Users/ВашеИмя/my_project
```

- `C:\` становится `/mnt/c/`
- `D:\` становится `/mnt/d/` и т.д.

Обратите внимание, буква диска пишется с **маленькой буквы!!**

### Способ 2: Копирование файлов в домашнюю директорию WSL

#### 1. Перейдите в домашнюю директорию WSL:

```
cd ~
```

#### 2. Создайте папку для проекта:

```
mkdir my_project  
cd my_project
```

#### 3. Скопируйте файлы из Windows-директории (например, с рабочего стола):

```
cp -r /mnt/c/Users/ВашеИмя/Desktop/my_project/* .
```

Замените путь на актуальный.

## Шаг 5: Работа с Docker в WSL

#### 1. Откройте терминал WSL (Ubuntu):

- Можно через Docker Desktop (вкладка WSL) или просто найдя "Ubuntu" в меню "Пуск".

#### 2. Перейдите в директорию вашего проекта (используя один из способов выше).

### 3. Убедитесь, что Docker работает:

```
docker --version
# Должна отобразиться версия Docker
docker ps
# Должен показать пустой список контейнеров без ошибок

# Проверяем версию Docker Compose Plugin
docker compose version

# Запускаем тестовый контейнер
docker run hello-world
```

Если команда hello-world выполнялась и вывела приветственное сообщение, установка прошла успешно.



## Шаг 6: Сборка и запуск вашего FastAPI-приложения

### 1. Соберите Docker-образ (находясь в директории проекта):

```
docker compose up -d --build
```

Флаг --build принудительно пересобирает образ вашего приложения.

Флаг -d запускает контейнеры в фоновом режиме (daemon).

### 2. Проверьте, что контейнер запустился:

```
docker ps
```

В списке должны быть контейнеры (`service_users`, `service_orders` и `api_gateway`) со статусом "Up".



## Шаг 7: Проверка работы API

### 1. Проверьте логи контейнера (убедитесь, что нет ошибок):

```
docker compose logs api_gateway docker compose logs service_users docker compose logs service_orders
```

В логах должно быть сообщение, что Uvicorn запущен на 0.0.0.0:8000.

### 2. Протестируйте эндпоинт /status:

Из терминала WSL:

```
curl http://localhost:8000/status
```

Или из браузера/Postman на вашей Windows:

- Откройте браузер и перейдите по адресу: `http://localhost:8000/status`

Ожидаемый ответ: {"status":"API Gateway is running"}