

## Подключение Laravel Sail к существующему проекту для взаимодействия со средой разработки Docker

*Стрельцова Марина Николаевна*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

### Аннотация

Для разработчиков важным аспектом при написании различных приложений является использование средств контейнеризации. Данный подход помогает обезопасить основную операционную систему от вредоносного кода, ускоряет процесс разработки и развертывания приложений, а также имеет гибкую и адаптивную структуру для общей работы над каким-либо проектом. В данной статье описано подключение Laravel Sail к существующему проекту для взаимодействия со средой разработки Docker.

**Ключевые слова:** laravel, sail, docker, контейнеризация

## Connecting Laravel Sail to an existing project to interact with the Docker development environment

*Streltsova Marina Nikolaevna*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

For developers, an important aspect when writing various applications is the use of containerization tools. This approach helps to protect the main operating system from malicious code, speeds up the process of developing and deploying applications, and has a flexible and adaptive structure for general work on any project. This article describes how to connect Laravel Sail to an existing project to interact with the Docker development environment.

**Keywords:** laravel, sail, docker, containerization

## 1. Введение

### 1.1 Актуальность исследования

Для разработчиков важным аспектом при написании различных приложений является использование средств контейнеризации. Данный подход помогает обезопасить основную операционную систему от вредоносного кода, ускоряет процесс разработки и развертывания приложений, а также имеет гибкую и адаптивную структуру для общей работы над каким-либо проектом. Существует различное программное обеспечение контейнеризации, которое позволяет изолировать все процессы,

при котором код приложения, среда запуска, библиотеки и зависимости упаковываются в единую «капсулу» — контейнер.

### **1.2 Обзор исследований**

В своем исследовании М. К. Солонько и К. А. Краснов рассматривают программную платформу виртуализации под названием Docker. Также данными авторами проведено сравнение с виртуальными машинами и целесообразность применения контейнеров, созданных на платформе Docker, для решения различных задач. Рассмотрен процесс создания и запуска контейнера на конкретном примере. Сделан вывод, что пользователям привычнее работать с виртуальными машинами, а платформа Docker удобна разработчикам [1]. И. А. Тарасов и Д. В. Борисенков в своей статье разработали веб-приложение для непрерывной интеграции микросервисов на платформе контейнеризации Docker [2]. В работе Б. И. Закамскова рассматривается контейнеризация проекта с возможностью дальнейшего горизонтального масштабирования [3].

### **1.3 Цель исследования**

Целью данного исследования является описание процесса подключения Laravel Sail [4] к существующему проекту для взаимодействия со средой разработки Docker [5].

## **2. Методы исследования**

Для подключения Laravel Sail к существующему проекту понадобится инструмент Docker. Docker - это проект с открытым исходным кодом для автоматизации развертывания приложений в виде переносимых автономных контейнеров, выполняемых в облаке или локальной среде.

Требования к аппаратной и программной части для подключения Laravel Sail (в рамках данной статьи):

1. Операционная система (ОС) Windows 10 версии 1904 или выше;
2. Фреймворк Laravel [6] с созданным проектом (Laravel 8x);
3. Установленный PHP (PHP 7.4) [7];
4. Менеджер пакетов Composer [8];
5. IDE с поддержкой командной строки (PHPStorm) [9];
6. Процессор, поддерживающий средства виртуализации.

## **3. Результаты исследования**

Для начала нужно включить Windows Subsystem for Linux (WSL) с помощью командной строки (Рис. 1). Более подробное описание компонента можно прочитать на официальном сайте Microsoft [10].

C:\> Администратор: Командная строка

```
Microsoft Windows [Version 10.0.19043.1706]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Windows\system32>wsl --install_
```

Рисунок 1 – Включение WSL

После данных действий нужно перезагрузить ПК и зайти в BIOS или UEFI, проверив, что включена функция виртуализации (Рис. 2).



Рисунок 2 – Пример включения функции виртуализации

Далее необходимо скачать Docker с официального сайта и установить его. После установки рекомендуется проверить наличие включенной опции «Use the WSL 2 based engine» (Рис. 3).

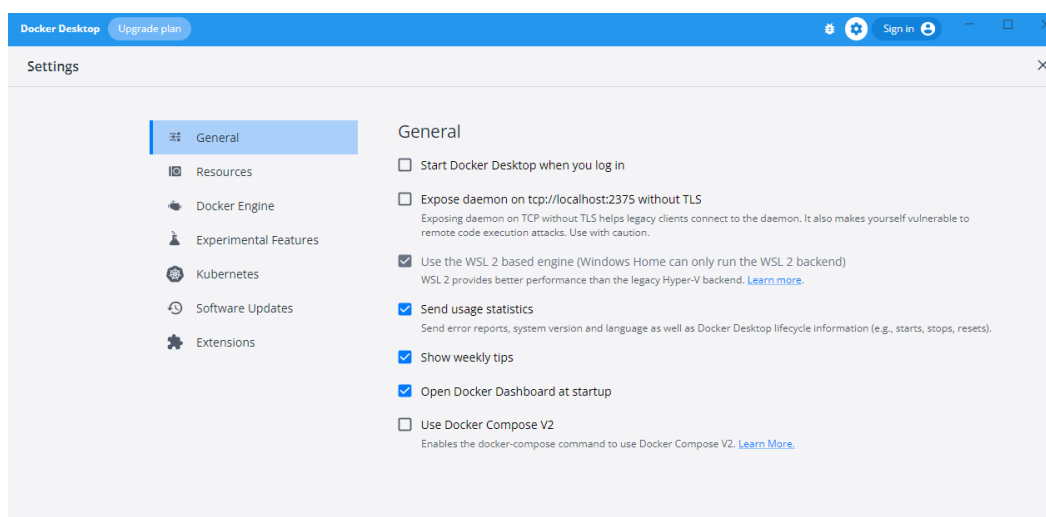


Рисунок 3 – Настройка Docker

Следующим шагом следует зайти в Microsoft Store, скачать и установить подсистему Ubuntu для Windows, необходимую для взаимодействия с Laravel Sail (Рис. 4).

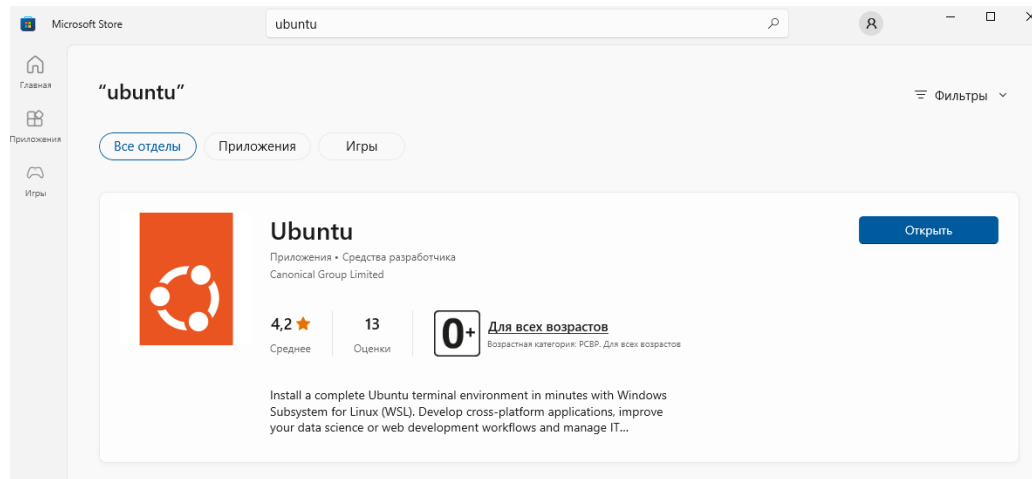


Рисунок 4 – Скачивание и установка Ubuntu

Далее нужно снова зайти в Docker и включить ранее скачанную подсистему (Рис. 5).

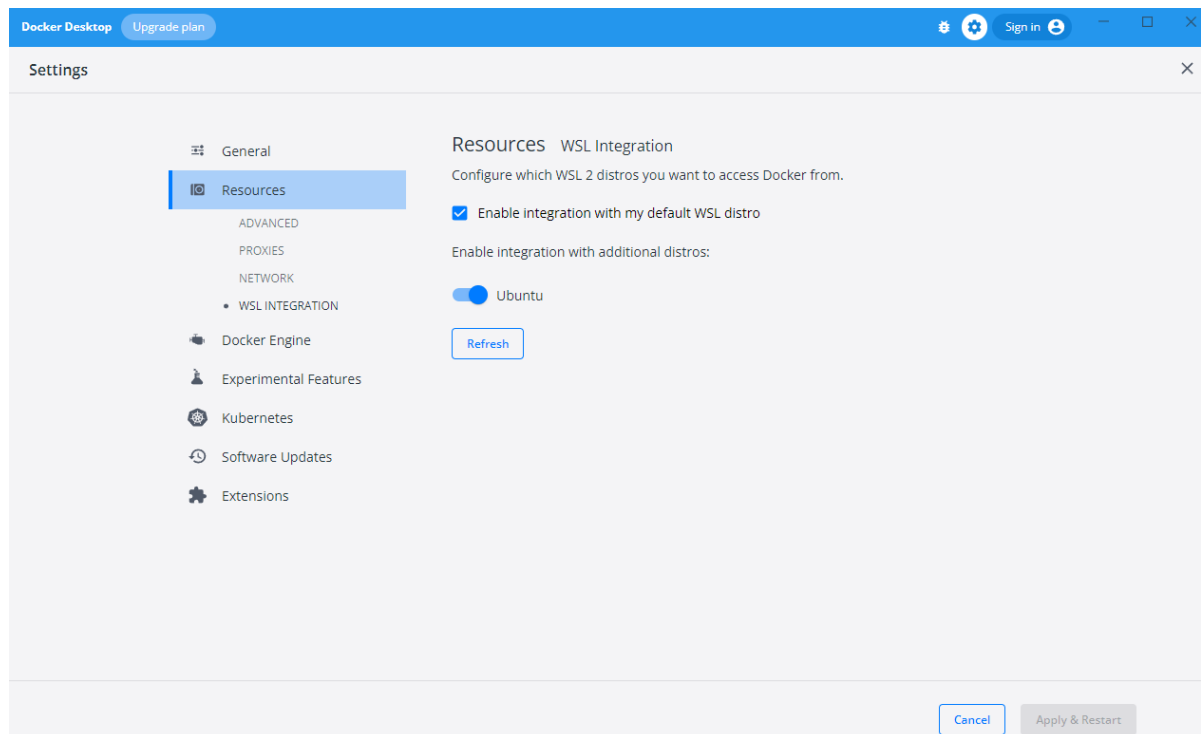
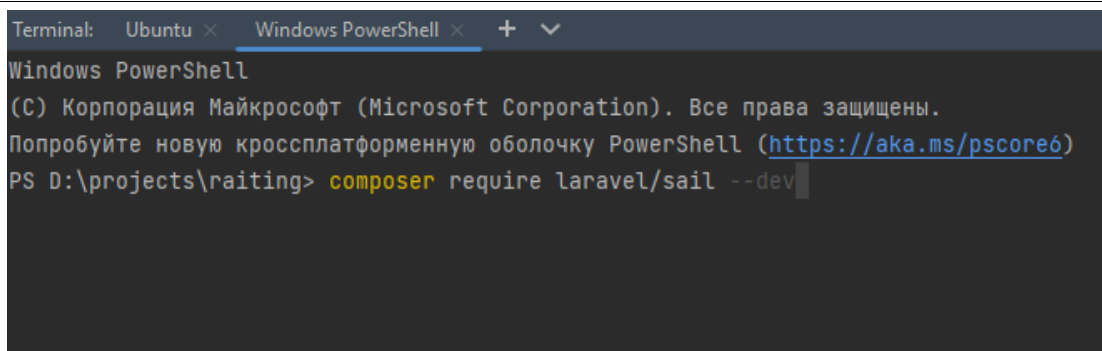


Рисунок 5 – Включение подсистемы в Docker

Теперь запустим IDE, перейдем в папку с созданным проектом Laravel и запустим следующую команду для установки Sail с помощью Composer (Рис. 6).



```
Terminal: Ubuntu x Windows PowerShell x + v
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.
Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)
PS D:\projects\raiting> composer require laravel/sail --dev
```

Рисунок 6 – Установка Sail

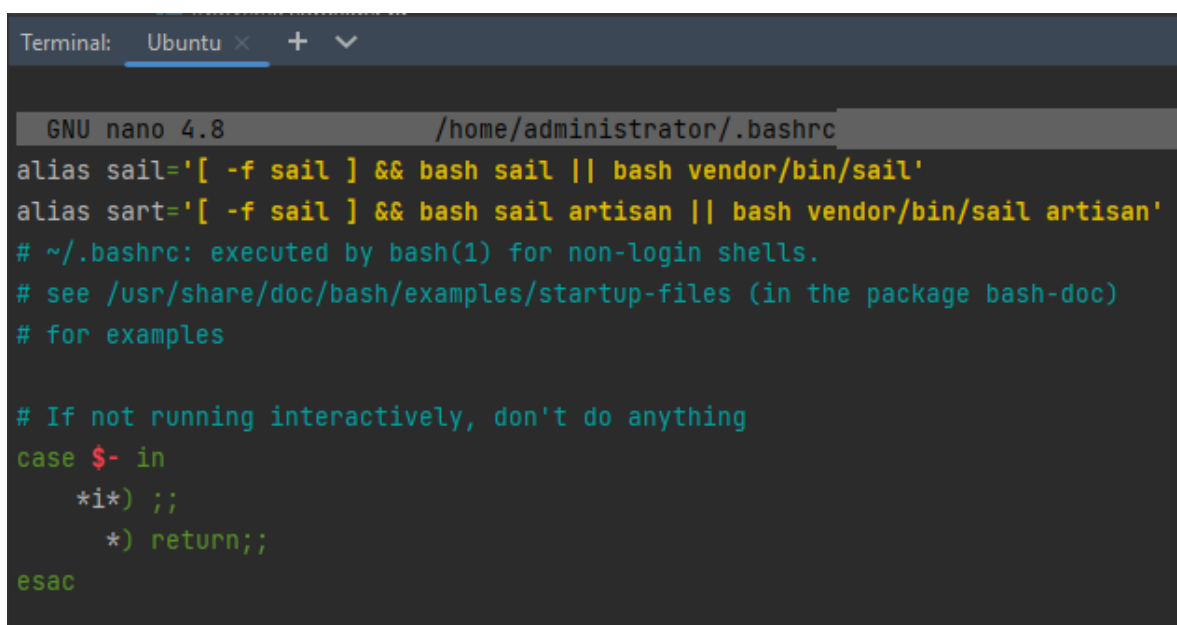
Далее перейдем в командную строку Ubuntu и выполним последовательно команды:

1. wsl
2. nano ~/.bashrc

Вторая команда откроет Bash файл для создания алиасов, которые позволяют сократить временные затраты на ввод определенных команд, заменяя путь к исполняемому файлу командой, задаваемой пользователем. Например, следующая команда заменит ввод vendor/bin/sail на sail:

- alias sail='[ -f sail ] && bash sail || bash vendor/bin/sail'

Данную команду необходимо вставить в открытый Bash файл и сохранить его (Рис. 7).

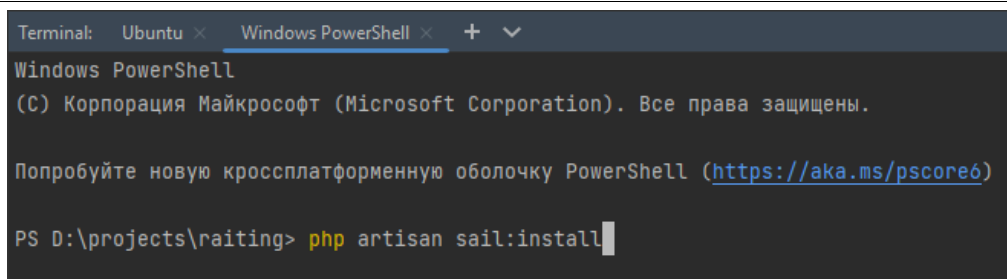


```
Terminal: Ubuntu x + v
GNU nano 4.8 /home/administrator/.bashrc
alias sail='[ -f sail ] && bash sail || bash vendor/bin/sail'
alias sart='[ -f sail ] && bash sail artisan || bash vendor/bin/sail artisan'
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *i*) ;;
  *) return;;
esac
```

Рисунок 7 – Создание алиасов

Следующим шагом нужно вернуться в локальную командную строку и ввести команду как на скриншоте 8. Эта команда опубликует файл Sail docker-compose.yml в корень приложения.



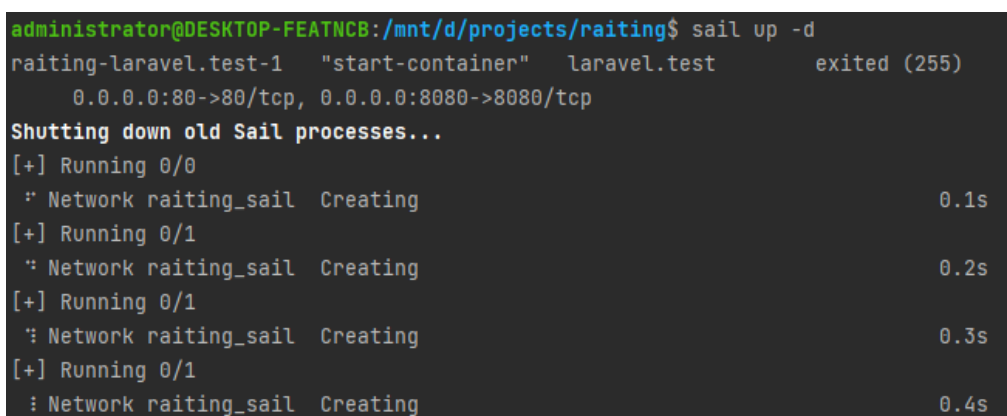
```
Terminal: Ubuntu x Windows PowerShell x + v
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

PS D:\projects\raiting> php artisan sail:install
```

Рисунок 8 – Публикация файла docker-compose.yml

Последним шагом в командной строке необходимо ввести `sail up -d` для запуска контейнера в фоновом режиме и начала работы с проектом на Laravel. Также статус контейнеров можно отследить в Docker (Рис. 9-10).



```
administrator@DESKTOP-FEATNCB:/mnt/d/projects/raiting$ sail up -d
raiting-laravel.test-1 "start-container" laravel.test exited (255)
0.0.0.0:80->80/tcp, 0.0.0.0:8080->8080/tcp
Shutting down old Sail processes...
[+] Running 0/0
 * Network raiting_sail Creating 0.1s
[+] Running 0/1
 * Network raiting_sail Creating 0.2s
[+] Running 0/1
 * Network raiting_sail Creating 0.3s
[+] Running 0/1
 * Network raiting_sail Creating 0.4s
```

Рисунок 9 – Запуск контейнера

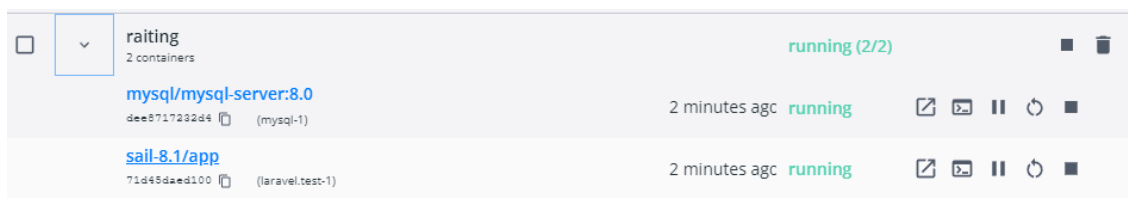


Рисунок 10 – Отображение контейнеров в Docker

Для проверки работоспособности проекта достаточно перейти по адресу `http://localhost`.

#### 4. Выводы

Laravel Sail — это удобный инструмент командной строки для взаимодействия со средой разработки Docker, обеспечивающий отличную отправную точку для создания приложения Laravel с использованием PHP, MySQL и Redis. К тому же для использования Sail не нужен опыт работы с Docker.

В рамках данной статьи описан процесс подключения Laravel Sail к существующему проекту для взаимодействия со средой разработки Docker. Были также рассмотрены необходимые инструменты, аппаратные и программные условия для начала работы с Laravel Sail. Выявлены преимущества использования инструмента контейнеризации — Docker.

**Библиографический список**

1. Солонько М. К., Краснов К. А. DOCKER-контейнеры //Телекоммуникации и информационные технологии. 2020. Т. 7. №. 1. С. 65-69.
2. Тарасов И. А., Борисенков Д. В. Разработка веб-приложения для непрерывной интеграции микросервисов на платформе контейнеризации Docker //Актуальные проблемы прикладной математики, информатики и механики. 2021. С. 632-638.
3. Закамсков Б. И. Контейнеризация и масштабирование веб сервиса// Информатика и вычислительная техника. 2021. С. 110-114.
4. Laravel Sail URL: <https://laravel.su/docs/8.x/sail> (дата обращения: 13.06.2022).
5. Docker URL: <https://www.docker.com/> (дата обращения: 13.06.2022).
6. Laravel URL: <https://laravel.com/> (дата обращения: 13.06.2022).
7. PHP URL: <https://www.php.net/> (дата обращения: 13.06.2022).
8. Composer URL: <https://getcomposer.org/> (дата обращения: 13.06.2022).
9. PHPStorm URL: <https://www.jetbrains.com/ru-ru/phpstorm/> (дата обращения: 13.06.2022).
10. WSL URL: <https://docs.microsoft.com/en-us/windows/wsl/install> (дата обращения: 13.06.2022).