

# Installation de Laravel & Docker

ROTHEUDT Thomas et MOUCHAMPS Antoine - 29 août 2023





# Table des matières

Table des matières			2
I	Intro 1 2 3	Oduction  Qu'est-ce que Docker?  Pourquoi Docker?  Qu'est-ce que Laravel Sail?	3 3 3
11	Insta 1 2 3 4 5	Installation de Docker Installation Linux Installation Windows  2.1 Prérequis 2.2 Installation de Docker Desktop Installation MacOS Aperçu VS Code	5 7 7 8 9 9
Ш	Utili 1 2	Démarrage du container	10 10 10
IV	Créa 1 2	Création d'un nouveau projet  1.1 Création du dossier  1.2 Ajout PhpMyAdmin  1.3  1.4 Localhosts  Ajout d'un projet depuis GitHub  2.1 Setup	11 11 12 12 13 14 14
V	Ensı	uite 1	16





# I. Introduction

## 1 Qu'est-ce que Docker ♥?

Docker est un outil open-source créé en 2012 par des français. Cet outil permet de créer, de déployer et de lancer des applications tournant dans un conteneur. Ces conteneurs sont en réalité des environnements dans lesquels les applications tournent. Ils sont créés grâce à des images qui sont des fichiers Docker , ainsi n'importe quel conteneur est assuré de tourner sur n'importe quel machine sans se soucier des configurations locales. L'outil est tellement populaire que les mainteneurs de librairies ou de logiciels maintiennent des images Docker (dans notre cas on verra que



Laravel Sail N possède une image Docker Q qui est maintenue à jour régulièrement) Il permet donc de "centraliser" l'environment sur lequel une application est développée.

## 2 Pourquoi Docker ?

Un conteneur Docker opossède de nombreux avantages comparé aux machines virtuels. Premièrement, il utilise les ressources d'un système plus éfficacement qu'une VM tout en gardant ses avantages (isolation et reproductibilité).



Il garanti d'être identique quel que soit le système, ainsi chaque membres d'un projet est certain de travailler sur le même environment sans se soucier de la portabilité de l'application sur l'environment final.

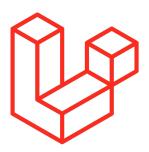
Les conteneurs sont isolés de la machine hôte, ainsi vous pouvez avoir plusieurs versions différentes de dépendances pour plusieurs projets.

Enfin, la modifications d'un conteneurs est extrêmement simplifié comparé à une VM où les mises à jour sont souvent complexes et laborieuse.

Pour résumer, un conteneur Docker 🕑 est flexible, léger, portable.

# 3 Qu'est-ce que Laravel Sail • ?

Laravel Sail 1 est une interface de ligne de commande légère pour interagir avec l'environnement de développement Docker 2 par défaut de Laravel 4.



Sail est un excellent point de départ pour construire une application Laravel en utilisant PHP, MySQL et Redis sans avoir besoin d'une expérience préalable de Docker.

Au cœur de Sail se trouve le fichier docker-compose.yml et le script sail qui est stocké à la racine de votre projet. Le script sail fournit une CLI (command line interface) avec des méthodes pratiques pour interagir avec les conteneurs Docker définis par le fichier docker-compose.yml.



Laravel Sail → est pris en charge sur MacOS ♠, Linux △ et Windows ➡ (via WSL2).

Pour résumer, le scrip sail permet de gérer le projet Laravel 👺 à l'intérieur du conteneur du projet. De plus Laravel Sail 🕦 est complètement compatible avec Docker 🔾, ce qui permet des interactions plus simples avec les conteneurs.





# II. Installation de Docker

Choisissez la section correspondant à votre système d'exploitation :



Linux: SECTION 1



Windows: SECTION 2



MacOS: SECTION 3

#### 1 Installation Linux 🐧

L'Installation Linux A nécéssite de suivre quelques étapes de plus que pour l'Installation MacOS de ou Windows .

1. Mettez à jour l'index des paquets apt et des paquets d'installation pour permettre à apt d'utiliser un dépôt sur HTTPS :

```
sudo apt—get update
```

2. Installez les dépendances de Docker 📀 :

```
sudo apt—get install \
    ca—certificates \
    curl \
    gnupg \
    lsb—release
```

3. Ajoutez la clé GPG officielle de Docker 📀 :

```
sudo mkdir —m 0755 —p /etc/apt/keyrings

curl —fsSL https://download.docker.com/linux/ubuntu/gpg |
sudo gpg ——dearmor —o /etc/apt/keyrings/docker.gpg
```

4. Configurez le répertoire :

 $\$(lsb\_release - cs)$  doit dans quelques cas être remplacé par votre version de distribution Linux  $\Delta$ . Dans l'installation de référence on a remplacé par jammy.

```
echo \ "deb [arch=\$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \ \(lsb_release -cs) stable"
```





| sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

5. Verifiez que le répertoire est bien configurer :

sudo apt-get update

6. Installez Docker Engine, conteneurd, et Docker Compose :

sudo apt—get install docker—ce docker—ce—cli conteneurd.io docker—buildx—plugin docker—compose—plugin

Vous pouvez tester votre installation avec la commande : sudo docker run hello-world

7. Installez Docker Desktop 🔮 :

Téléchargez le Docker Destop et l'installez avec la commande :

 $sudo\ apt-get\ install\ ./docker-desktop-<\!version>-<\!arch>.deb$ 

L'Installation de référence pour ce guide a été faite sur une distribution Linux 🐧 Mint (distibution basé sur Ubuntu 🌖





#### 2 Installation Windows



#### 2.1 Prérequis: WSL2

Pour utiliser Docker Desktop , il est fortement conseillé d'utiliser WSL2. C'est une application Windows 💶 permettant d'installer un environnement Linux 🚨 avec la distribution de votre choix nativement sous Windows =

Pour l'installer, il suffit de lancer wsl --install dans le terminal de Windows € de base ou Powershell<sup>1</sup>. Par défaut, cette commande installera Ubuntu **1** comme système Linux **1**. Pour d'avantage d'informations concernant l'installation de WSL2, consultez le lien vers la page officielle de Microsoft. Si tout se passe bien vous devriez recevoir le message suivant :

```
Windows Subsystem for Linux has been installed.
Installing: Übuntu
Jbuntu has been installed.
```

Après avoir redémarer votre PC, la fenêtre suivante devrait apparaitre. Si ce n'est pas le cas, chercher wsl.exe dans la barre de recherche windows et lancez le terminal.

```
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username
For more information visit: https://aka.ms/wslusers
```

lci, vous devez créer un utilisateur ainsi qu'ajouter un mot de passe <sup>2</sup>. Attention, vous devrez utiliser votre mot de passe de temps en temps donc ne l'oubliez pas!

Une fois ces étapes effectuées sans accros, vous pouvez passer à l'installation de Docker Desktop 📀 à proprement parlé.

#### 2.2 Installation de Docker Desktop 🔮

- 1. Téléchargez Docker Desktop
- 2. Exécutez le fichier d'installation . exe
- 3. Vérifiez l'installation en lançant votre environment wsl et en tapant la commande suivante :

docker

Docker cet wsl2 sont totalement compatibles. C'est pourquoi vous devez cocher l'option "use WSL2" pendant l'installation.

- 1. Attention, le terminal choisi doit être exécuté en tant qu'administrateur!
- 2. Lorsque vous tapez votre mot de passe, il est normal que les charactères ne s'affichent pas.



## 3 Installation MacOS

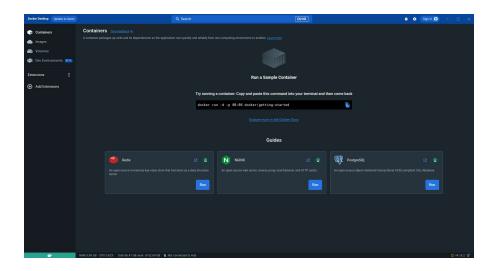
- 1. Téléchargez Docker Desktop Apple Chip ou Intel Chip
- 2. Installez Docker Desktop odans le dossier application





## 4 Aperçu

Si tout s'est bien passé vous êtes en mesure de lancer Docker Desktop ②. Vous avez donc le panneau de contrôle de Docker Desktop ② qui devrait ressembler à ça :



Vous pouvez voir différents onglets sur la gauche :

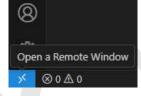
- Conteneurs : Cet onglet va contenir tous les conteneurs que vous avez créés. Les conteneurs sont en quelque sorte des images déployées et exécutées lorsque Docker tourne.
- Images : Cet onglet va contenir toutes les images d'installation des services installés dans nos conteneurs. Les images décrivent l'environnement du serveur, les packages installés et toutes les configurationde celui-ci.
- Volumes : Cet onglet contient les volumes liés aux services qui contiennent les data de nos applications. Les volumes servent à conserver les données des base de donnée lorsque nous éteignons un container et qu'on le relance.
- Dev Environnements : une feature très intéressante permettant de créer des environnements de développement (plus d'informations).

#### 5 Utilisez VS Code < ♥!!

Enfin, pour utiliser tout ce que nous venons d'installer, il faut un éditeur de code efficace. VS Code permet, via ses extensions d'interagir avec Docker et WSL2 en parfaite harmonie, c'est pourquoi nous le recommandons CHAUDEMENT.

Une fois VS Code **X** installé, ajouter les extensions WSL (de Microsoft) et Docker (de Microsoft également). Une fois cela fait, cliquez sur le petit carré montré ci-joint.

Ensuite, sélectionner Connect to WSL. C'est ici que vous devrez effectuer les prochaines commandes.





# III. Utilisation

Avant de créer votre premier projet, nous allons voir les commandes de base pour utiliser Laravel Sail 1.

Pour utiliser le script, il vous faudra utiliser la commande suivante dans le dossier du projet (que vous créerez à la section suivante) :

./vendor/bin/sail [command] [options] [arguments]

Vous pouvez créer un alias pour sail en ajoutant la ligne suivante à votre .bashrc, .zshrc, ect :

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

Ce qui vous permettra de ne plus devoir taper ./vendor/bin/sail à chaque commande mais simplement sail.

## 1 Démarrage du container

Pour lancer le container et ainsi pouvoir travailler sur le projet, entrez dans le dossier de votre projet via un terminal et lancer la commande :

sail up  $-\mathsf{d}$ 

Cette commande lancera le container en mode détaché pour garder l'accès au terminal (pratique puisque vous devrez exécuter de nombreuses commandes, même une fois le container démaré) <sup>3</sup>

C'est une fois cette commande faite que vous pourrez accéder à votre site en allant sur <a href="http://localhost/">http://localhost/</a> ou votre URL personnalisée.

#### 2 Arrêt du container

Pour éteintre le container lorsque vous avez finis de travailler, tapez la commande

sail down

3. On peut aussi lancer dans un terminal avec sail up



# IV. Création Projet Laravel 👺

- 1. Si vous souhaitez créer un nouveau projet vierge, allez à la SECTION 1.
- 2. Si vous souhaitez importer un projet depuis GitHub ②, allez à la SECTION 2.

## 1 Création d'un nouveau projet

#### 1.1 Création du dossier

Pour créer un nouveau projet Laravel & dans un dossier "exemple-app" il suffit juste d'entrer la ligne de commande :

```
curl —s https://laravel.build/example—app | bash
```

Avec cette commande votre projet sera installé avec plusieurs services de base (mysql, redis, et d'autres). PS :la création peut prendre du temps (510 minutes).

il est possible de choisir les services à installer avec le mot clé *with*. Dans le cadre de ce tutoriel, nous aurons besoin de 3 services : MySQL, mailpit et PhpMyAdmin . Les 2 premiers peuvent être téléchargés avec laravel comme suit :

```
curl —s "https://laravel.build/example—app?with=mysql,mailpit" | bash
```

une fois la commande lancée le projet va se créer (cela peut prendre un certain temps)



Une fois le dossier du projet créé vous allez rentrer dans le dossier (File et ensuite Open Folder...) et ouvrir le fichier docker-compose.yml.

Ce fichier représente la configuration générale de votre conteneur, c'est ici que l'on va définir les services à installer, les versions, les images, ect. Ce fichier est très important, sans lui, pas de conteneur.



Vous pouvez voir qu'il est composé de plusieurs sections, Version, services, network, volumes. Chaque section à son utilitée. Nous allons nous concentrer sur la section service du fichier. Plus précisement, nous allons ajouter un service.

### 1.2 Ajout PhpMyAdmin <table-cell-columns>

On va ajouter un service à notre projet qui est nécessaire à la gestion de notre base de donnée, PhpMyAdmin ...

Pour ce faire, rendez-vous à la fin de la section services, et ajoutez y PhpMyAdmin Acomme suit :

Une fois cela fait, PhpMyAdmin hest intégré au projet.

#### 1.3 Setup

Il y'a plusieurs choses à faire avant de pouvoir accéder à votre site. Premièrement, configurer le fichier .env :

- APP NAME : changer le nom de votre application.
- APP\_URL : changer l'URL pour accéder à votre site, attention, cette url doit être conforme à celle écrite dans le fichier hosts de la section suivante. Par example, http://example.
- DB\_DATABASE : changez le nom de votre base de donnée comme bon vous semble.

Les autres champs peuvent à priori rester inchangés.

Une fois cela fait, vous pouvez démarer le container avec sail up -d.

Ensuite, installez les librairies Javascript € telle que Vite ▼ avec sail npm install

Enfin, afin de pouvoir accéder à votre site avec l'URL fournie dans le .env, il est nécessaire d'ajouter un localhost.





### **1.4** Localhosts

La méthode diffère selon le système d'exploitation :

— Si vous êtes sous Windows €, allez dans C:\Windows\System32\drivers\etc\ et ouvrez le fichier hosts 4. Dedans, ajoutez deux lignes comme ceci :

127.0.0.1 example ::1 example

- Si vous êtes sous Linux 🐧, pareil mais le fichier se trouve dans /etc/.
- Si vous êtes sous MacOS €, pareil mais le fichier se trouve dans /private/etc/.

Une fois cela fait, redémarez votre navigateur et allez sur l'URL que vous avez choisie, vous devriez voir la page d'acceuil de Laravel 👺.

<sup>4.</sup> ATTENTION, vous devez ouvrir ce fichier avec les droits d'administrateur, sinon vous ne pourrez pas sauvegarder vos modifications.



## 2 Ajout d'un projet depuis GitHub ®

Lorsque vous clonez un repo sur votre machine, le site manque des composants essentiels à son fonctionnements : ses packages. Pour installer ces packages, il est nécessaire d'effectuer la commande composer install, sauf que vous n'avez pas composer d'installé. En revanche, composer est utilisable grâce à Laravel Sail 1 par sail composer install!

Le petit hick, c'est que Laravel Sail • est un package... à installer avec composer install 5.

Heureusement, Laravel & a la sollution pour nous. Il suffit de se rendre dans le dossier de votre projet fraichement installé et de taper :

```
\begin{array}{c} \mathsf{docker}\;\mathsf{run}\;--\mathsf{rm}\;\backslash\\ -\mathsf{u}\;\mathsf{"}(\mathsf{id}\;-\mathsf{u}):\mathsf{"}(\mathsf{id}\;-\mathsf{g})\;\;\backslash\\ -\mathsf{v}\;\mathsf{"}(\mathsf{pwd}):/\mathsf{var}/\mathsf{www}/\mathsf{html}\;\;\backslash\\ -\mathsf{w}\;/\mathsf{var}/\mathsf{www}/\mathsf{html}\;\;\backslash\\ \mathsf{laravelsail}/\mathsf{php82}-\mathsf{composer}:\mathsf{latest}\;\;\backslash\\ \mathsf{composer}\;\mathsf{install}\;--\mathsf{ignore}-\mathsf{platform}-\mathsf{reqs} \end{array}
```

de cette manière, les packages de composer seront installés. Plus de renseignements sur la Doc Laravel .

Avec les packages (et donc Laravel Sail 1) d'installés, vous pouvez démarer le container avec sail up -d.

En ce qui concerne les librairies Javascript<sup>©</sup>, vous pouvez executer

```
sail npm install
```

pour installer, entres autres, ViteV.

#### 2.1 Setup

Enfin, il faut setup les paramètres du site en créant un fichier . env <sup>6</sup>. Vous pouvez modifier :

- APP NAME : changer le nom de votre application.
- APP\_URL : changez l'url pour accéder à votre site. Attention, cette url doit être conforme à celle écrite dans le fichier hosts de la SECTION 1.4
- DB DATABASE : changer le nom de votre base de donnée comme bon vous semble.
- DB\_HOST : à changer en mysql.

A priori, les autres champs doivent rester inchangés.

Enfin, il ne vous reste plus qu'à exécuter

<sup>5.</sup> Le serpent qui se mort la queue, l'oeuf ou la poule, appellez ce paradoxe comme vous le voulez.

<sup>6.</sup> ps : vous pouvez copier et renommer le fichier .env example, mais ne supprimez pas l'original!



sail artisan key: generate

et qu'à ajouter l'URL de votre site dans la liste de vos localhosts comme décrit dans la  ${\tt SECTION}$  1.4.

#### 2.2 Dernières étapes

Enfin, les dernière étapes consistent à taper ces deux commandes, que nous expliquerons dans la formation suivante :

- 1. sail artisan migrate:fresh --seed pour lancer la base de donnée.
- 2. sail npm run dev pour compiler CSS v et Javascript.

Une fois cela fait, redémarez votre navigateur et allez sur l'URL que vous avez choisie, vous devriez voir la page d'acceuil de Laravel &.





# V. Ensuite

Bien, avec votre site déployé et opérationnel, vous êtes prêts pour passer à la formation suivante dans laquelle vous créerez véritablement votre site.

