

Docker

Documentation pour nouvelle version du Docker Compose

Prérequis

- Docker installé sur votre machine (voir la [documentation officielle de Docker](#))
- WSL 2 installé sur votre machine
- Docker Compose installé (inclus dans les installations de Docker Desktop pour Windows et Mac)
- Connaissances de base en manipulation de CLI (Command Line Interface)

Structure du fichier docker-compose.yml

Ce fichier définit les services nécessaires pour faire fonctionner l'ensemble de l'application. Ci-dessous, un fichier docker-compose.yml pour citeo-pro :

```
1 #####  
2 # Generated on docker-local.io #  
3 #####  
4 version: '3.1'  
5 services:  
6   wordpress:  
7     build: .  
8     env_file:  
9       - path: ./docker-local/.env  
10      required: true # default  
11    working_dir: /usr/share/nginx/html  
12    container_name: citeopro-wordpress  
13    volumes:  
14      - './site/wp-content/plugins/ac-addon-acf:/usr/share/nginx/html/wp-content/plugins/ac-addon-acf'  
15      - './site/wp-content/plugins/citeoforms:/usr/share/nginx/html/wp-content/plugins/citeoforms'  
16      - './site/wp-content/plugins/wpforms:/usr/share/nginx/html/wp-content/plugins/wpforms'  
17      - './site/wp-content/plugins/polylang:/usr/share/nginx/html/wp-content/plugins/polylang'  
18      - './site/wp-content/plugins/admin-columns-pro:/usr/share/nginx/html/wp-content/plugins/admin-column  
19      - './site/wp-content/plugins/advanced-custom-fields-pro:/usr/share/nginx/html/wp-content/plugins/adv  
20      - './site/wp-content/themes:/usr/share/nginx/html/wp-content/themes'  
21    ports:  
22      - '9081:80'  
23   mysql:  
24     image: 'mysql:8.0'  
25     working_dir: /usr/share/nginx/html  
26     container_name: citeopro-mysql  
27     volumes:  
28       - './:/usr/share/nginx/html'  
29     environment:  
30       - MYSQL_ROOT_PASSWORD=root  
31       - MYSQL_DATABASE=local_db_citeopro  
32     ports:  
33       - '9041:3306'  
34   phpmyadmin:  
35     image: phpmyadmin/phpmyadmin  
36     container_name: citeopro-phpmyadmin  
37     links:  
38       - mysql
```

```
39      ports:
40          - "8081:80"
41      environment:
42          UPLOAD_LIMIT: 1024M
43          PMA_ARBITRARY: 1
```

Installation et Démarrage ↗

Clonage ou création du fichier docker-compose:

- Si disponible, clonez le dépôt (sur [le git ici](#))

Récupération du fichier .env à placer dans docker-local/

```
1 # Valeur a recuperer via un autre dev
2 ENVIRONMENT=
3 DB_HOST=
4 DB_NAME=
5 DB_PSWD=
6 DB_USER=
7 SQL_TABLES_PREFIX=
8 FRONTEND_ACCESS_PSWD=
9 FRONTEND_ACCESS_USER=
10 BID=
```

Lancement des containers:

Exécutez la commande suivante depuis le répertoire contenant le fichier docker-compose.yml :

```
1 docker-compose up -d
```

Accès aux services ↗

Wordpress :

Accessible via <http://localhost:9081>

phpMyAdmin :

Accessible via <http://localhost:8081>

Utilisez **root** et **MYSQL_ROOT_PASSWORD** pour vous connecter.

i Vous pouvez retrouver le bon port a utilisé dans le docker-compose.yml à **ports** dans chaque container

Gestion des volumes ↗

Les données pour MySQL et Wordpress sont stockées dans des volumes Docker pour persistance.

Ces volumes sont définis dans le fichier docker-compose sous volumes.

On se sert des volumes pour définir le thème et les plugins dans les volumes du wordpress car nous envoyons sur le git seulement les contenus et non plus tout le wordpress.

Exemple :

- - './site/wp-content/plugins/ac-addon-acf:/usr/share/nginx/html/wp-content/plugins/ac-addon-acf'
 - La première partie avant les : est le chemin vers le plugin sur notre poste
 - La deuxième partie après les : est le chemin vers le plugin dans le container

Rentrer dans le container

Pour rentrer dans un container, il suffit de lancer la commande

```
1 docker-compose exec [Container] bash
```

Cela lance le bash du container

Afficher les logs

Pour afficher les logs en continu, il faut lancer la commande

```
1 docker-compose exec wordpress tail -f /var/log/php82/error.log
```

[WIP] Makefile

Changelog

- 14/08/24 : Ajout du squelette et information sur le .env