

# DOCUMENTATION

## Récupérer son environnement de développement PHP

---



## INTRODUCTION

Dans ce document je vais vous expliquer comment récupérer votre environnement de développement pour pouvoir déployer et tester vos futures applications PHP.

## Structure

- Un container avec apache2 / PHP 7-0 / mysql-client
- Un container avec mysql-server-5.7

## Fonctionnement

Dans cette partie je vais vous décrire le fonctionnement de cet environnement.

### 1. Partie serveur web et code

Lancée depuis un dockerfile build manuellement, ce container va en fait inclure la partie serveur-web avec sa configuration et son service (000-default.conf). Mais il inclura aussi la partie php-7 / php-mysql car il possédera le web directory dans son /var/www. Mais comme un développeur a souvent besoin d'une base de donnée, via le package php-mysql, le container pourra communiquer via le port 3306 avec le container hébergeant la partie bdd.

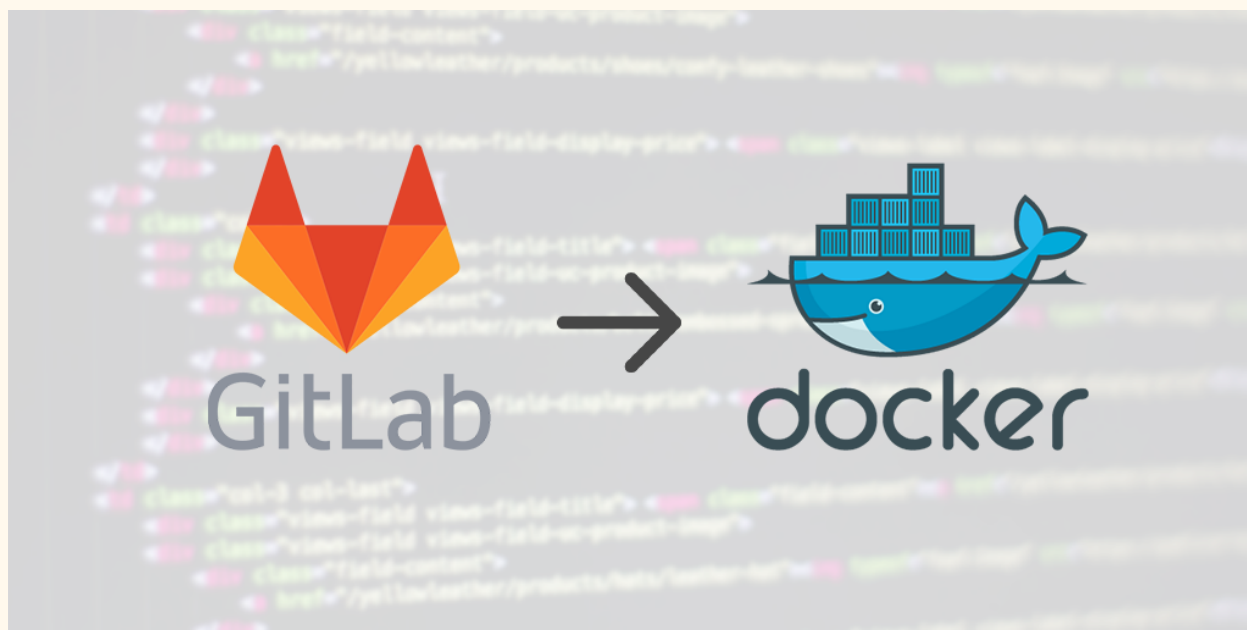
### 2. Partie serveur de base de donnée

Inclue un container mysql-server-5.7 permettant d'avoir une base de donnée accessible via le port 3306.

Informations de connexion à la base de donnée :

```
ports:
  - "3306:3306"
environment:
  - "MYSQL_ROOT_PASSWORD=passe"
  - "MYSQL_USER=your_user"
  - "MYSQL_PASSWORD=passe"
  - "MYSQL_DATABASE=db"
```

# Récupération de l'environnement



Pour la récupération de l'environnement il vous faudra d'abord avoir quelques notions d'utilisation de docker et de git mais aussi avoir certaines paquets/utilitaires présents sur votre poste :

- paquet "git" **obligatoire**
- utilitaire "docker" **obligatoire**
- utilitaire "docker-compose" **obligatoire**
- client "ftp" possiblement (type : FileZilla, WinSCP etc..)

Si vous ne possédez pas les paquets/utilitaires dits <**obligatoires**> alors veuillez vous rendre dans la partie pré-requis.

## Pré-requis

### **GIT :**

- **CentOS** : yum install git -y
- **Debian/Ubuntu** : apt-get install git -y
- **Windows** : <https://git-scm.com/download/win>
- **ArchLinux** : pacman -S git

### **DOCKER :**

- Suivez la documentation docker officielle :  
<https://docs.docker.com/install/linux/docker-ce/debian/>

### **DOCKER-COMPOSE :**

- Suivez la documentation docker officielle :  
<https://docs.docker.com/compose/install/>

## Installation

**A l'aide de “git”, vous allez pouvoir récupérer le projet et ses fichiers**

- git clone <https://vps591692.ovh.net/zweeking/myproject.git>  
(se connecter avec son compte gitlab)

## Il faudra récupérer le certificat .crt permettant la connexion en HTTPS

### Contenu du fichier vps591692.ovh.net :

```
-----BEGIN CERTIFICATE-----
MIIDjzCCAnegAwIBAgIJAKbIZpZyuUuhMA0GCSqGSIb3DQEBCwUAMF4xCzAJBgNV
BAYTAihYMRUwEwYDVQQHDAxEZWZhdWx0IENpdHkxHDAaBgNVBAoME0RIZmF1bHQg
Q29tcGFueSBMdGQxGjAYBgNVBAMMEXZwcU5MTY5Mi5vdmgubmV0MB4XDTE4MTAx
NjEwNDcxNloXDTE5MTAxNjEwNDcxNlowXjELMAkGA1UEBhMCWFgxFTATBgNVBACM
DERIZmF1bHQgQ2I0eTEcMBoGA1UECgwTRGVmYXVsdCBDb21wYW55IEx0ZDEaMBgG
A1UEAwwRdnBzNTkxNjkyLm92aC5uZXQwggeiMA0GCSqGSIb3DQEBAQUAA4IBDwAw
ggEKAoIBAQCgQ3Ay+EsFEY33oHUGmEoCd1wWosbvt5AegfGjqMdi6tKLtKGEU/IW
trjftcwbxSbSZL+37Rr6D0skBdQU+3uJ6pByGaZ3bm04lfQBG/+C/VrQh+ztMzw5
u3DU2+W3CDjVIUxx4vcKhNtuGwkvf2Jr5cd/0qldX0XbnGL4ZwNoH7JtgCY2HQgZ
NxGkThqvxEYwRn1dLGmMKLyAO/EPnDyfrZKfOYWucFKkoqPknYa0qXvNVdjYV2P
/8qXfgn/sAy9zQ5jvSfmce9zzj91ICbLa5iiM634YP7EisqJNeSvknreZgmOd5b4
/rr2jCpmCkGOL1peHQRGWWfgoIM1cZAgMBAAGjUDBOMB0GA1UdDgQWBQBQ9AYNM
OqVDSH1IWbrr8CEMVpT1vzAfBgNVHSMEGDAWgBQ9AYNMOqVDSH1IWbrr8CEMVpT1
vzAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQBsZIJqx0XfV3OAacqE
8fz/4oJ7QETUnU2vL/xv1/cCwqnAlM5P2srWQnfZhDeHCAjMXFyNMLkk2TvGEUk1
Ke5Ozk0RvNONYk+3KgKtjioENseTPWErUZ62dKHKq3pYflqfE2IA0VevCCimOYsA
zCPMmJx9YRiaYInP+n2fiK8+94u7MltjZgk37jlbaji8e4nTG/9CU6MgLvt0s3M/
MvKli4gpjDkhMs92pIFQjJpvLYhk9bHROf6k5dOFBcT4IL/XSMdAyNPdAYUFRUEH
DP9vGXtd4evH9dAkD9HnsUq4xnIkOENvXPWGxUHNfWYvr/tFMX+gUVRcX3tWTI8g
rMDq
-----END CERTIFICATE-----
```

### Sous Linux :

- placer le vps591692.ovh.net.crt dans  
/etc/pki/ca-trust/source/anchors/vps591692.ovh.net.crt

### Sous Windows :

- certutil -addstore -f ROOT C:\your\path\to\the\cert

**Vous allez ensuite pouvoir accéder au dossier myproject/**

- `cd myproject/`
- `docker-compose up -d`

**Après la mise en route des deux containers**

- `docker ps`
- `docker exec -ti container_id_apachephp bash`

**Ne pas oublier de modifier les privilèges de l'utilisateur root bdd après sa création dans le container bdd**

- `docker exec -it 63c8c138e0fb mysql -uroot -p -e "CREATE USER 'root'@'%' IDENTIFIED BY 'urpass';GRANT ALL PRIVILEGES ON *.* TO 'root'@'%;FLUSH PRIVILEGES;"`

## Comment modifier le code PHP présent de base?

Si vous avez bien compris comment tout cela fonctionne, alors cela n'a rien de très compliqué. Il va juste falloir exécuter un shell dans le conteneur :

- **`docker exec -ti container_id_apache bash`**

Ensuite on peut modifier les fichiers directement dedans à l'aide d'éditeurs de texte... si vous êtes un dev vous allez rire là. Et je comprends, donc sinon il y a une autre façon. On peut ensemble choisir un répertoire dans lequel vous allez push votre code et on va le monter dans le conteneur à chaque `docker-compose up -d`. Dans une future version 2.0 il en sera question.

Il va ensuite falloir lancer sur votre navigateur une page <http://votreIP:8081> ! Et voilà, si votre code PHP est correct, et que votre connexion PHP to MYSQL est réussie (new PDO), alors le code sera interprété !