

Installation Docker avec Apache, PHP et MySQL

Sur une machine Debian



Pour commencer, nous allons passer en root pour avoir les droits administrateurs

```
emersonn@debian-EMERSON:~$ su -  
Mot de passe :
```

Nous allons installer docker en utilisant un repository

Pour ce faire nous allons faire un script Shell "dockerInstall.sh" qui permet d'automatiser

```
root@debian-EMERSON:~# nano dockerInstall.sh
```



Cette commande nous ouvrira un fichier, sur lequel nous pourrons commencer à faire notre script :

Tout d'abord nous faisons la mise à jour de l'index des paquets apt et des paquets d'installation pour permettre à apt d'utiliser un dépôt via HTTPS

```
GNU nano 5.4 dockerInstall.sh *  
#!/bin/sh  
  
apt-get update  
  
apt-get install \  
ca-certificates \  
curl \  
gnupg
```



Ensuite nous ajoutons la clé GPG officielle de Docker :

```
#!/bin/sh

apt-get update

apt-get install \
ca-certificates \
curl \
gnupg

mkdir -m 0755 -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | \
gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```



Ensuite il faut configurer le référentiel :

```
apt-get install \
ca-certificates \
curl \
gnupg

mkdir -m 0755 -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | \
gpg --dearmor -o /etc/apt/keyrings/docker.gpg

echo \
"deb [arch="$(dpkg --print-architecture)" \
signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/debian \
"${(. /etc/os-release && echo "$VERSION_CODENAME")}" stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Maintenant, nous pouvons commencer l'installation de docker engine

Pour ce faire, nous pouvons commencer par mettre à jour les paquets :

```
apt-get update
```



Si vous avez un problème lors de la mise à jour, faite la commande suivante :

```
chmod a+r /etc/apt/keyrings/docker.gpg  
apt-get update
```



Ensuite, il faut installer Docker Engine, containerd et Docker Compose.

```
apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin \  
docker-compose-plugin
```



Et enfin, vérifiez que l'installation de Docker Engine est réussie en exécutant hello-world :

```
docker run hello-world
```



Ensuite, il faut faire en sorte que l'utilisateur (user) de votre machine doit pouvoir lancer la commande docker sans "sudo".

```
groupadd docker
usermod -aG docker emersonn
chown "emersonn":"emersonn" /home/"emersonn"/.docker -R
chmod g+rx /home/emersonn/.docker -R
```



Puis activer le lancement automatique de(s) service(s) Docker au démarrage.

```
systemctl enable docker.service
systemctl enable containerd.service
```



Ensuite, nous pouvons installer Docker en utilisant le script "dockerInstall.sh".

```
root@debian-EMERSON:~# ./dockerInstall.sh
```



Maintenant, nous pouvons vérifier que docker fonctionne correctement avec la commande suivante :

```
root@debian-EMERSON:~#  
root@debian-EMERSON:~# docker run hello-world  
Hello from Docker!
```



Ensuite, nous allons télécharger le conteneur "ubuntu/apache2:2.4-20.04_beta"

```
root@debian-EMERSON:~# docker pull ubuntu/apache2:2.4-20.04_beta
```



Nous pourrions ensuite créer le répertoire /var/www/html_docker

```
root@debian-EMERSON:~# mkdir -p /var/www/html_docker
```



Nous pouvons ensuite, lancer le conteneur avec la commande suivante :

```
docker run --name serverphp -d -v /var/www/html_docker:/var/www/html -p  
<Votre@IP>:8001:80 ubuntu/apache2:2.4-20.04_beta /usr/sbin/apache2ctl -DFOREGROUND
```

```
root@debian-EMERSON:~# docker run --name serverphp -d -v /var/www/html  
_docker:/var/www/html -p 192.168.31.85:8001:80 ubuntu/apache2:2.4-20.0  
4_beta /usr/sbin/apache2ctl -DFOREGROUND  
284c645076a44d04facca18f5a5b44aaf657fe3c98de2fd5227d6189d070078d
```



Par la suite, nous allons ajouter **ServerName 127.0.0.1** à la fin du fichier **/etc/apache2/apache2.conf** (de serverphp), pour éviter les bugs

Pour se faire, il faut se connecter au conteneur avec la commande :
docker exec -it serverphp /bin/bash

```
root@debian-EMERSON:~# docker exec -it serverphp /bin/bash  
root@284c645076a4:/# ls
```



Nous pouvons donc maintenant, accéder au fichier et ajouter la **ServerName 127.0.0.1** à la fin du fichier grâce aux double « >> »

Nous pouvons cat le fichier, pour vérifier que la ligne soit bien ajoutée

```
root@284c645076a4:/etc/apache2# echo "ServerName 127.0.0.1" >>apache2.  
conf  
root@284c645076a4:/etc/apache2# cat apache2.conf
```



Ensuite, il faut générer une image de serverphp

```
root@debian-EMERSON:~# docker commit serverphp severphpimg:01
sha256:ec3aacdd74565567835bdde7b89c00f13c16001619817e610521732a0015428
0
root@debian-EMERSON:~# █
```



Enfin, nous pouvons maintenant lancer le nouveau conteneur

```
root@debian-EMERSON:~# docker run --name serverphp01 -d -v /var/www/html_docker:
/var/www/html -p 192.168.31.85:8001:80 severphpimg:01 /usr/sbin/apache2ctl -DFOR
EGROUND
```



Nous sommes maintenant aptes à installer les paquets php8.2 et php8.2-mysql.

Pour ce faire, nous commençons par se connecter au conteneur

```
root@debian-EMERSON:~# docker exec -it serverphp01 /bin/bash
root@06c551b1e2d2:/# █
```



Ensuite nous mettons à jour et à niveau les paquets

```
root@06c551b1e2d2:/# apt update && apt upgrade -y
```




Ensuite nous allons redémarrer le conteneur avec les commandes suivantes :

```
root@06c551b1e2d2:/# exit
exit
root@debian-EMERSON:~# docker stop serverphp01
serverphp01
root@debian-EMERSON:~# docker exec -it serverphp01 /bin/bash
Error response from daemon: Container 06c551b1e2d2e24edd7791644a970e6b96891c42722a210793e162d09e1f2007 is not running
```



Maintenant, nous procédons à l'installation de php avec les

Commandes suivantes :

```
root@06c551b1e2d2:/# apt install software-properties-common
root@06c551b1e2d2:/# add-apt-repository ppa:ondrej/php
root@06c551b1e2d2:/# apt update
root@06c551b1e2d2:/# apt install php8.2 -y
root@06c551b1e2d2:/# php --version
PHP 8.2.4 (cli) (built: Mar 16 2023 14:34:24) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.2.4, Copyright (c) Zend Technologies
with Zend OPcache v8.2.4, Copyright (c), by Zend Technologies
```



Ensuite, il faut redémarrer le conteneur :

```
root@debian-EMERSON:~# docker stop serverphp01
root@debian-EMERSON:~# docker start serverphp01
serverphp01
```



Maintenant, nous allons installer le conteneur Mysql avec la commande suivante :

```
root@debian-EMERSON:~# docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
d98f79840351: Pull complete
e1e1bcf71068: Pull complete
1ae5de5bc10f: Pull complete
e7be806b0533: Pull complete
efe4db00091d: Pull complete
cdc80326f5be: Pull complete
2e8b64cd8427: Downloading 26.95MB/56.21MB
6175ffb8d824: Download complete
2321d4015453: Downloading 22.37MB/50.17MB
```



Maintenant, nous créons les répertoires : /u01/mysql/datadir

```
|root@debian-EMERSON:~# mkdir -p /u01/mysql/datadir
```



Ensuite, nous pourrions donc lancer ce conteneur avec la commande suivante :

```
root@debian-EMERSON:~# docker run --name mysqlDB -e MYSQL_ROOT_PASSWORD=prevert
-d -v /u01/mysql/datadir:/var/lib/mysql -p 192.168.31.85:3306:3306 mysql
1ef99456ca8d5fd1f13a33a1261564b91d9059f46600bda7b02106f8d5a4f3bc
```



Nous pouvons maintenant nous connecter au conteneur :

```
root@debian-EMERSON:~# docker exec -it mysqlDB /bin/bash
bash-4.4# _
```



Ensuite nous nous connectons à mysql :

```
bash-4.4# mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.32 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```



Ensuite nous nous connectons à mysql :

```
mysql> GRANT ALL PRIVILEGES ON *.* TO "|root"@"%" with GRANT OPTION;
```

Nous pouvons voir les bdd avec la commande suivante :

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql       |
| performance_schema |
| sys        |
+-----+
4 rows in set (0.00 sec)
```

Puis sélectionner mysql :

```
mysql> use mysql
```



Nous pouvons donc maintenant créer la table user01 :

```
mysql> CREATE TABLE user01 (id int, identifiant varchar(30), password varchar(30));
Query OK, 0 rows affected (0.02 sec)
```



J'ai par la suite ajouté un utilisateur à notre table user01 :

```
mysql> insert into user01 (id, identifiant, password) values (1, 'user', 'prevert');
Query OK, 1 row affected (0.01 sec)
```



Ensuite, nous allons copier un répertoire myapps dans le chemin /var/www/html_docker avec la commande suivante :

```
cp /chemin/vers/myapps/* /var/www/html_docker/myapps
```

Une fois le répertoire à disposition, nous pouvons l'exploiter à partir du navigateur.



Vous avez maintenant un environnement de développement Docker avec Apache, PHP et MySQL sur une machine Debian

