

Projet Docker

WORDPRESS DEPLOYMENT

Amandine CHARLES | Docker | 23/04/2021

Table des matières

Présentation du projet 2

Mise en place du projet 2

Résultats 3

Présentation du projet

J'ai décidé pour ce projet de mettre un place un wordpress de base avec une base de données avec mysql et phpmyadmin. L'idées est d'avoir notre wordpress qui communique avec la base de données.

Mise en place du projet

J'ai alors créé un docker compose :

```
1. version: "3.7"
2. services:
3.
4.   #database
5.   mysql:
6.     image: mysql:5.7
7.     environment:
8.       MYSQL_ROOT_PASSWORD: toor
9.       MYSQL_USER: wordpress
10.      MYSQL_DATABASE: wordpress
11.      MYSQL_PASSWORD: wordpress
12.      restart: always
13.      volumes:
14.        - ./db:/var/lib/mysql
15.      ports:
16.        - 3306:3306
17.      networks:
18.        wordpress-net:
19.
20.   #phpmyadmin
21.   phpmyadmin:
22.     depends_on:
23.       - mysql
24.     image: phpmyadmin/phpmyadmin:5.0
25.     environment:
26.       PMA_HOST: mysql
27.       PMA_PORT: 3306
28.     volumes:
29.       - ./php:/etc/phpmyadmin
30.     restart: always
31.     ports:
32.       - 8080:80
33.     networks:
34.       wordpress-net:
35.
36.   #wordpress
37.   wordpress:
38.     depends_on:
39.       - mysql
40.     image: wordpress:5.4.0-php7.2-apache
41.     environment:
```

```

42.     WORDPRESS_DB_HOST: mysql
43.     WORDPRESS_DB_USER: wordpress
44.     WORDPRESS_DB_PASSWORD: wordpress
45.     WORDPRESS_DB_NAME: wordpress
46.     volumes:
47.       - ./wordpress:/var/www/html
48.     restart: always
49.     ports:
50.       - 80:80
51.     networks:
52.       wordpress-net:
53.
54. volumes:
55.   db:
56.   php:
57.   wordpress:
58.
59. networks:
60.   wordpress-net:

```

Dans ce dernier, on lui instancie différents services, comme le service 'MySQL' qui utilise l'image de MySQL avec des variables d'environnement pour définir les différents paramètres de connexion à la base.

Puis on utilise le service phpMyAdmin pour visualiser et administrer la base de données. C'est pour cela que j'ai rajouté dans ce service le paramètre 'depends_on' pour indiquer que ce service dépend aussi du service précédent. Il est important de lui attribuer un numéro de port afin d'avoir accès à l'interface web.

Et enfin, j'ai créé le service WordPress afin d'avoir notre page WordPress, connecté à notre MySQL, bien évidemment.

Après avoir créé le docker compose, il suffit de lancer la commande : `docker-compose up -d`.

Résultats

Après avoir lancer la commande dites précédemment, nous différentes images vont se télécharger et les différents conteneurs vont se créer.

```

Creating projet_docker_mysql_1 ... done
Creating projet_docker_phpmyadmin_1 ... done
Creating projet_docker_wordpress_1 ... done

```

On aperçoit bien les conteneurs lancés :

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
aeb34fc9dca3	wordpress:5.4.0-php7.2-apache	"docker-entrypoint.s..."	18 minutes ago	Up 18 minutes	0.0.0.0:80->80/tcp, :::80->80/tcp	projet_docker
wordpress_1	phpmyadmin/phpmyadmin:5.0	"/docker-entrypoint..."	18 minutes ago	Up 18 minutes	0.0.0.0:8080->80/tcp, :::8080->80/tcp	projet_docker
fd08147d1c0b	mysql:5.7	"docker-entrypoint.s..."	18 minutes ago	Up 18 minutes	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp	projet_docker
mysql_1						

Ainsi, pour accéder à notre phpMyAdmin, il nous suffit d'aller sur <http://localhost:8080>, pour notre WordPress sur <http://localhost:80> et enfin

Il suffit par la suite de rentrer les identifiants de connexion, définis dans le docker-compose.

